

TECHNICAL UNIVERSITY OF CRETE

DIPLOMA THESIS

StudCampCater: Streamlined Electronic Ordering and Production Management System for Student Catering Club

Author:

Boutalis Anastasios Periklis
A.M. 2017030123

Thesis Committee:

Prof. Nikolaos Giatrakos (Supervisor)
Prof. Antonios Deligiannakis
Prof. Georgios Karystinos



School of Electrical & Computer Engineering

October 2025

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

StudCampCater: Βελτιστοποιημένο Ηλεκτρονικό Σύστημα Διαχείρισης Παραγγελιοληψίας και Παραγωγής Φοιτητικής Λέσχης Σίτισης

Συγγραφέας:

Μπούταλης Αναστάσιος Περικλής
Α.Μ. 2017030123

Εξεταστική Επιτροπή:

Καθ. Γιατράκος Νικόλαος (Επιβλέπων)
Καθ. Δεληγιαννάκης Αντώνιος
Καθ. Καρυστινός Γεώργιος



Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Οκτώβριος 2025

Στον αγαπημένο μου γιο, Αναστάση.

Η περιέργειά και το χαμόγελο σου με εμπνέουν κάθε μέρα.
Με όλη μου την αγάπη.

Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον κ. Στέφανο Καρασαββίδη για την πολύτιμη καθοδήγηση και υποστήριξή του στο κομμάτι ανάπτυξης (development) της εφαρμογής, καθώς και για τις χρήσιμες συμβουλές και ιδέες που συνέβαλαν σημαντικά στην υλοποίηση του έργου. Ευχαριστώ επίσης τον κ. Νεκτάριο Μουμουτζή για την εξαιρετική συνεισφορά του στο UI/UX, για τη δημιουργικότητά του και τη βοήθεια στη σχεδίαση ενός φιλικού και λειτουργικού περιβάλλοντος χρήστη.

Ιδιαίτερες ευχαριστίες οφείλω στον επιβλέποντά μου, Νικόλαο Γιατράκο, ο οποίος με καθοδήγησε αδιάλειπτα καθ' όλη τη διάρκεια της διπλωματικής μου εργασίας, παρέχοντας πολύτιμες συμβουλές, υποστήριξη και έμπνευση.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου για τη συνεχή στήριξη, κατανόηση και ενθάρρυνσή τους καθ' όλη τη διάρκεια των σπουδών μου και της εκπόνησης αυτής της εργασίας. Η παρουσία τους ήταν καθοριστική για την ολοκλήρωση αυτού του έργου.

Περίληψη

Η παρούσα διπλωματική εργασία επικεντρώνεται στον σχεδιασμό και την ανάπτυξη μιας διαδικτυακής εφαρμογής για τη βελτίωση της διαχείρισης φοιτητικής λέσχης, με έμφαση στην εμπειρία του φοιτητή και την αποδοτικότητα του διαχειριστή. Στόχος είναι να αντιμετωπιστούν συχνά προβλήματα, όπως οι μεγάλες ουρές, η σπατάλη φαγητού και η έλλειψη πληροφόρησης για τη διαθεσιμότητα γευμάτων.

Η εφαρμογή περιλαμβάνει δυνατότητες όπως online κρατήσεις, προβολή μενού σε πραγματικό χρόνο, παρακολούθηση πληρότητας της λέσχης και εργαλεία για τους διαχειριστές για τη διαχείριση παραγγελιών και μενού. Ένα καινοτόμο στοιχείο αποτελεί ο μηχανισμός πρόβλεψης ποσοτήτων φαγητού, ο οποίος βασίζεται στο εργαλείο Facebook Prophet και χρησιμοποιεί ιστορικά δεδομένα, αργίες και καιρικές συνθήκες για την ακριβέστερη εκτίμηση της ζήτησης.

Η υλοποίηση αξιοποιεί σύγχρονες τεχνολογίες τόσο στο frontend όσο και στο backend, με έμφαση στην ασφάλεια (JWT, XSS prevention), την εμπειρία χρήστη (UI/UX σχεδιασμός) και τη δομή βάσης δεδομένων. Η Python χρησιμοποιείται για ανάλυση δεδομένων και εκπαίδευση μοντέλων πρόβλεψης.

Συμπερασματικά, η εργασία επιδεικνύει τη δυνατότητα ψηφιακού μετασχηματισμού των φοιτητικών υπηρεσιών σίτισης, προτείνοντας μια λύση που μπορεί να επεκταθεί και να προσαρμοστεί σε πραγματικά δεδομένα στο μέλλον.

Abstract

This thesis focuses on the design and development of a web application aimed at improving the management of a university dining hall, emphasizing both student experience and administrative efficiency. The goal is to address common issues such as long queues, food waste, and lack of information regarding meal availability.

The application includes features such as online reservations, real-time menu display, dining hall occupancy monitoring, and administrative tools for managing orders and menus. An innovative component of the system is the food portion forecasting mechanism, which is based on the Facebook Prophet tool and utilizes historical data, holidays, and weather conditions to more accurately estimate demand.

The implementation leverages modern technologies on both the frontend and backend, with a focus on security (JWT, XSS prevention), user experience (UI/UX design), and database structure. Python is used for data analysis and predictive model training.

In conclusion, this work demonstrates the potential for digital transformation of student dining services, proposing a solution that can be expanded and adapted to real-world data in the future.

Περιεχόμενα Περιεχομένων

Ευχαριστίες	4
Περίληψη	5
Abstract	6
Περιεχόμενα Περιεχομένων	7
Περιεχόμενα Εικόνων	10
Περιεχόμενα πινάκων	14
1. Εισαγωγή	15
1.1 Σκοπός και Στόχοι της Εργασίας	15
1.2 Κίνητρα και Σημασία του Έργου	16
2. Ανάλυση Αναγκών και Περιγραφή Προβλήματος	17
2.1 Περιγραφή Υφιστάμενης Κατάστασης	17
2.2 Ανάγκες και Απαιτήσεις Χρηστών	19
2.3 Προβλήματα που Εντοπίζονται στο Παρόν Σύστημα	20
2.4 Επιθυμητές Λειτουργίες του Νέου Συστήματος / Απαιτήσεις Βελτίωσης	21
3. UI/UX Σχεδιασμός και Αξιολόγηση	24
3.1 Εισαγωγή – Ανάλυση βασικών εννοιών	24
3.2 Εντοπισμός προβλήματος και ανάλυση απαιτήσεων χρηστών	26
3.3 Δημιουργία Wireframes	27
3.4 Ανθρωποκεντρικός σχεδιασμός και iterative development	43
3.5 Αξιολόγηση αρχικών σχεδίων με χρήστες	45
3.6 Αποτελέσματα Αξιολόγησης Wireframes	48
4. Τεχνολογίες και Εργαλεία Ανάπτυξης	50
4.1 Frontend τεχνολογίες	50
4.2 Backend τεχνολογίες	53
4.3 Βάση δεδομένων	55
4.4 Πρόβλεψη Ποσοτήτων Φαγητού	56
4.5 Εργαλεία ανάπτυξης και υποστήριξης	59
5. Ανάλυση και Σχεδίαση Συστήματος	62
5.1 Αρχιτεκτονική Συστήματος	62
5.2 Δομή και Σχέσεις της Βάσης Δεδομένων	64
5.2.2 Ανάλυση Σχέσεων Πινάκων και Κατηγοριοποίηση	72
5.3 Επικοινωνία Frontend – Backend	76
5.4 Μηχανισμός Σύνδεσης στην εφαρμογή	87

5.5 Λειτουργικότητες για Φοιτητές	93
5.6 Λειτουργικότητες για Διαχειριστές	98
5.7 Έλεγχος Πρόσβασης και Ασφάλεια	116
6. Δημιουργία και Εκπαίδευση του Μοντέλου Πρόβλεψης Μερίδων	124
6.1 Σύνοψη Συμβολής του Έργου	124
6.2 Δημιουργία Dataset	125
6.3 Δομή του Dataset	127
6.4 Προεπεξεργασία Δεδομένων (καθαρισμός, holidays, weather)	129
6.5 Καθορισμός και Ερμηνεία Μεταβλητών στο Σύστημα Πρόβλεψης Μερίδων	130
6.6 Ανάπτυξη Προβλεπτικού Συστήματος με Prophet και Εκπαίδευση Ανά Φαγητό	132
6.7 Οπτικοποίηση και Ερμηνεία των Προβλεπτικών Αποτελεσμάτων	135
6.8 Περιορισμοί του Μοντέλου Πρόβλεψης και Προοπτικές Βελτίωσης	137
7. Ανάλυση και Σχεδίαση Συστήματος Πρόβλεψης	141
7.1 Δοκιμή 1 – Πρόβλεψη για 10 Φαγητά	141
7.2 Δοκιμή 2 – Πρόβλεψη για 50 Φαγητά με In-Sample Αξιολόγηση	144
7.3 Δοκιμή 3 – Out-of-Sample Πρόβλεψη για 150 Φαγητά με Παραδοξότητες	147
7.4 Εναλλακτική Προσέγγιση: Aggregated Forecasting	152
7.5 Συγκριτική Αποτίμηση των Τριών Δοκιμών	153
7.6 Διαγράμματα και Οπτικοποιήσεις	155
8. Παρουσίαση Εφαρμογής	160
8.1 Διεπαφή Χρήστη	160
8.2 Διεπαφή Διαχειριστή	166
9. Αξιολόγηση Λειτουργικότητας, Απόδοσης και Σχεδίασης της Εφαρμογής	175
9.1 Εισαγωγή στην Αξιολόγηση της Εφαρμογής	175
9.2 Αξιολόγηση Email Ειδοποιήσεις με Υψηλή Συμβατότητα και Αξιοπιστία Παράδοσης	179
9.3 Αξιολόγηση UI/UX της εφαρμογής	181
10. Συμπεράσματα και Μελλοντική Εργασία	184
10.1 Σύνοψη Συμβολής της Διπλωματικής	184
10.2 Δυνατά Σημεία και Καινοτομία	184
10.3 Περιορισμοί του Συστήματος	185
10.4 Προτάσεις για Μελλοντική Βελτίωση	185
10.5 Δυνατότητες Επέκτασης (π.χ. mobile app, AI βελτιστοποίηση, chatbot)	186
Παράρτημα Α – Ερωτηματολόγιο Χρηστών	187
A.1 Γενικές Πληροφορίες	187
A.2 Πριν την Παρουσίαση του Wireframe	187
A.3 Κατά την Παρουσίαση του Wireframe	187

A.4 Μετά τη Χρήση του Wireframe	187
A.5 Tasks Εκτελέσιμα Κατά τη Δοκιμή	187
A.6 Μετά την Εκτέλεση των Tasks	188
Αναφορές	189

Περιεχόμενα Εικόνων

Εικόνα 1: Προτεινόμενο Εβδομαδιαίο πρόγραμμα σίτισης για τα γεύματα της 1ης εβδομάδας.	18
Εικόνα 2: Προτεινόμενο Εβδομαδιαίο πρόγραμμα σίτισης για τα δείπνα της 1ης εβδομάδας.	18
Εικόνα 3: Διάγραμμα Δραστηριοτήτων του διαχειριστή σε σχέση με τον μηχανισμό πρόβλεψης.	23
Εικόνα 4: Στιγμιότυπο οθόνης από το wireframe της αρχικής σελίδας, όπου ο χρήστης επιλέγει είδος γεύματος (Wireframe A).	29
Εικόνα 5: Στιγμιότυπο οθόνης από το wireframe όπου απεικονίζεται η διάταξη του μενού (παρουσιάζονται δύο εκδοχές) (Wireframe A).	30
Εικόνα 6: Στιγμιότυπο οθόνης από το wireframe όπου απεικονίζεται η διάταξη του μενού, αφού ο χρήστης έχει επιλέξει κάποιο φαγητό (Wireframe A).	31
Εικόνα 7: Στιγμιότυπο οθόνης από το wireframe όπου απεικονίζεται η σύνοψη της παραγγελίας (παρουσιάζονται δύο εκδοχές) (Wireframe A).	31
Εικόνα 8: Στιγμιότυπο οθόνης από το wireframe όπου παρουσιάζονται οι τελικές επιλογές του χρήστη κατά το checkout: είδος φαγητού, χρονικό διάστημα (timeslot), σχόλια και τρόπος πληρωμής (Wireframe A).	32
Εικόνα 9: Στιγμιότυπο οθόνης από την οθόνη ενημέρωσης του χρήστη σχετικά με την κατάσταση της παραγγελίας (αποδοχή, απόρριψη ή αναμονή απάντησης) (Wireframe A).	32
Εικόνα 10: Στιγμιότυπο οθόνης από το προφίλ του χρήστη, όπου εμφανίζονται οι προσωπικές πληροφορίες, το ιστορικό voucher και το ιστορικό παραγγελιών. Στη δεξιά πλευρά προβάλλονται αναλυτικά τα στοιχεία της παραγγελίας όταν ο χρήστης επιλέγει να δει περισσότερες πληροφορίες (Wireframe A).	33
Εικόνα 11: Στιγμιότυπο οθόνης από το wireframe της αρχικής σελίδας, όπου ο χρήστης επιλέγει είδος γεύματος (Wireframe B).	34
Εικόνα 12 : Στιγμιότυπο οθόνης από το wireframe όπου απεικονίζεται η διάταξη του μενού (παρουσιάζονται δύο εκδοχές) (Wireframe B).	34
Εικόνα 13: Στιγμιότυπο οθόνης από το pop-up κατά την επιλογή φαγητού και την προβολή περισσότερων πληροφοριών. (Wireframe B).	35
Εικόνα 14: Στιγμιότυπο οθόνης από το wireframe όπου απεικονίζεται η διάταξη του μενού, αφού ο χρήστης έχει επιλέξει κάποιο φαγητό (Wireframe B).	35
Εικόνα 15: Στιγμιότυπο οθόνης από το wireframe όπου απεικονίζεται η σύνοψη της παραγγελίας (Wireframe B).	36
Εικόνα 16: Στιγμιότυπο οθόνης που απεικονίζει την επιλογή timeslot σε μορφή grid. Στη δεξιά εικόνα φαίνεται το layout με ένα επιλεγμένο timeslot (Wireframe B).	36
Εικόνα 17: Στιγμιότυπο οθόνης κατά το checkout, όπου ο χρήστης επιλέγει τις τελευταίες λεπτομέρειες: πού θα φάει, προσθέτει σχόλιο και επιλέγει τον τρόπο πληρωμής. (Wireframe B).	37
Εικόνα 18: Στιγμιότυπο οθόνης όπου ο χρήστης επιλέγει τον τρόπο πληρωμής (Wireframe B).	37
Εικόνα 19: Στιγμιότυπο οθόνης από την οθόνη ενημέρωσης του χρήστη σχετικά με την κατάσταση της παραγγελίας (αποδοχή, απόρριψη ή αναμονή απάντησης) (Wireframe B).	38
Εικόνα 20: Στιγμιότυπο οθόνης από το προφίλ του χρήστη, όπου εμφανίζονται οι προσωπικές πληροφορίες, το ιστορικό voucher και το ιστορικό παραγγελιών (Wireframe B).	38

Εικόνα 21: Στιγμιότυπο οθόνης από το wireframe όπου απεικονίζεται η διάταξη του μενού. Στη δεξιά πλευρά φαίνεται η επιλογή του φαγητού (Wireframe C).	39
Εικόνα 22: Στιγμιότυπο οθόνης από το wireframe όπου απεικονίζεται η σύνοψη της παραγγελίας (Wireframe C).	40
Εικόνα 23: Στιγμιότυπο οθόνης που απεικονίζει την αλλαγή του meal type στη σελίδα του μενού (Wireframe C).	40
Εικόνα 24: Στιγμιότυπο οθόνης από το wireframe όπου παρουσιάζονται οι τελικές επιλογές του χρήστη κατά το checkout: είδος φαγητού, χρονικό διάστημα (timeslot), σχόλια και τρόπος πληρωμής (Wireframe C).	41
Εικόνα 25: Στιγμιότυπο οθόνης από την οθόνη ενημέρωσης του χρήστη σχετικά με την κατάσταση της παραγγελίας (αποδοχή, απόρριψη ή αναμονή απάντησης) (Wireframe C).	41
Εικόνα 26: Στιγμιότυπο οθόνης από το προφίλ του χρήστη, όπου εμφανίζονται οι προσωπικές πληροφορίες, το ιστορικό voucher και το ιστορικό παραγγελιών. Στη δεξιά πλευρά προβάλλονται αναλυτικά τα στοιχεία της παραγγελίας όταν ο χρήστης επιλέγει να δει περισσότερες πληροφορίες (Wireframe C).	42
Εικόνα 27: Διάγραμμα οντοτήτων–συσχετίσεων (ERD) της βάσης δεδομένων του συστήματος.	64
Εικόνα 28: Ο πίνακας Users της βάσης δεδομένων με τα βασικά πεδία του.	65
Εικόνα 29: Ο πίνακας Orders της βάσης δεδομένων με τα βασικά πεδία του.	66
Εικόνα 30: Ο πίνακας FoodItems της βάσης δεδομένων με τα βασικά πεδία του.	66
Εικόνα 31: Ο πίνακας Timeslots της βάσης δεδομένων με τα βασικά πεδία του.	67
Εικόνα 32: Ο πίνακας DefaultTimeslot της βάσης δεδομένων με τα βασικά πεδία του.	67
Εικόνα 33: Ο πίνακας MenuList της βάσης δεδομένων με τα βασικά πεδία του.	68
Εικόνα 34: Ο πίνακας Cart της βάσης δεδομένων με τα βασικά πεδία του.	68
Εικόνα 35: Ο πίνακας Vouchers της βάσης δεδομένων με τα βασικά πεδία του.	69
Εικόνα 36: Ο πίνακας amsitisi της βάσης δεδομένων με τα βασικά πεδία του.	69
Εικόνα 37: Ο πίνακας LetOrders της βάσης δεδομένων με τα βασικά πεδία του.	70
Εικόνα 38: Ο πίνακας OnVacations της βάσης δεδομένων με τα βασικά πεδία του.	70
Εικόνα 39: Ο πίνακας FoodItemCategory της βάσης δεδομένων με τα βασικά πεδία του.	71
Εικόνα 40: Sequence Diagram “Admin Change Meal Status” με τη ροή δεδομένων.	83
Εικόνα 41: Activity Diagram που απεικονίζει τη ροή ανανέωσης του Access Token.	89
Εικόνα 42: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ συστατικών του συστήματος κατά την ανανέωση ενός Access Token.	90
Εικόνα 43: Activity Diagram που παρουσιάζει τη ροή των ενεργειών κατά τη διαδικασία σύνδεσης στην εφαρμογή.	90
Εικόνα 44: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ χρηστών και συστατικών του συστήματος κατά τη διαδικασία σύνδεσης στην εφαρμογή.	91
Εικόνα 45: Activity Diagram που παρουσιάζει τη ροή ενεργειών κατά την υποβολή παραγγελίας από τον χρήστη.	94
Εικόνα 46: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ χρηστών και συστατικών του συστήματος κατά την υποβολή παραγγελίας με χρήση voucher.	95

Εικόνα 47: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ χρηστών και συστατικών του συστήματος κατά την προβολή ιστορικού παραγγελιών.	97
Εικόνα 48: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ χρηστών και συστατικών του συστήματος κατά την προβολή ιστορικού voucher.	98
Εικόνα 49: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ διαχειριστή και συστατικών του συστήματος κατά τη δήλωση του μενού της ημέρας.	100
Εικόνα 50: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ διαχειριστή και συστατικών του συστήματος κατά τη δήλωση του μενού της ημέρας.	102
Εικόνα 51: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ διαχειριστή και συστατικών του συστήματος κατά την τροποποίηση ενός υφιστάμενου φαγητού.	103
Εικόνα 52: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ χρηστών και συστατικών του συστήματος κατά τη διαχείριση παραγγελίας με ελλείψεις.	105
Εικόνα 53: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ χρηστών και συστατικών του συστήματος κατά τη διαχείριση παραγγελίας με ελλείψεις.	105
Εικόνα 54: Στιγμιότυπο οθόνης της διεπαφής χρήστη που εμφανίζει ειδοποίηση email απόρριψης παραγγελίας.	106
Εικόνα 55: Στιγμιότυπο οθόνης της διεπαφής χρήστη που εμφανίζει ειδοποίηση email αποδοχής παραγγελίας.	107
Εικόνα 56: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ διαχειριστή και συστατικών του συστήματος κατά τη διαχείριση voucher.	109
Εικόνα 57: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ διαχειριστή και συστατικών του συστήματος κατά τη διαχείριση των Timeslots.	111
Εικόνα 58: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ διαχειριστή και συστατικών του συστήματος κατά τον καθορισμό περιόδων διακοπών.	113
Εικόνα 59: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ διαχειριστή και συστατικών του συστήματος κατά την ενεργοποίηση του μηχανισμού πρόβλεψης.	115
Εικόνα 60: Activity Diagram που παρουσιάζει τη ροή ενεργειών κατά τη διαδικασία πρόβλεψης ζήτησης με βάση τον μηχανισμό πρόβλεψης.	116
Εικόνα 61: Activity Diagram που παρουσιάζει τη ροή ενεργειών κατά τον έλεγχο ασφαλείας των API.	119
Εικόνα 62: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ χρηστών και συστατικών του συστήματος κατά τον έλεγχο πρόσβασης των API.	120
Εικόνα 63: Sequence Diagram που παρουσιάζει την ασφαλή αλληλεπίδραση μεταξύ χρήστη, frontend, backend και βάσης δεδομένων κατά την κλήση ενός API.	123
Εικόνα 64: Στιγμιότυπο οθόνης από τα αποτελέσματα πρόβλεψης για το φαγητό “Chicken with Mustard Sauce”.	135
Εικόνα 65: Διάγραμμα Ανάλυσης Πρόβλεψης Μερίδων Greek Yogurt μέσω Facebook Prophet (2020–2025).	143
Εικόνα 66: Activity Diagram που παρουσιάζει το pipeline πρόβλεψης ζήτησης με χρήση του Prophet.	156
Εικόνα 67: Sequence Diagram που παρουσιάζει τη χρονική αλληλουχία των διεργασιών εκπαίδευσης και αξιολόγησης.	157

Εικόνα 68: Class Diagram που παρουσιάζει τη λογική δομή των δεδομένων και των βασικών λειτουργικών ενοτήτων του συστήματος.	158
Εικόνα 69: Flowchart που παρουσιάζει τον αλγόριθμο πρόβλεψης μερίδων φαγητού.	159
Εικόνα 70: Αρχική οθόνη της εφαρμογής που παρουσιάζει το μενού της ημέρας στους χρήστες.	160
Εικόνα 71: Αρχική οθόνη της εφαρμογής που παρουσιάζει το δηλωμένο μενού της ημέρας στους χρήστες.	161
Εικόνα 72: Στιγμιότυπο οθόνης που παρουσιάζει τη σύνοψη της παραγγελίας και τη δυνατότητα υποβολής από τον χρήστη.	162
Εικόνα 73: Στιγμιότυπο οθόνης Επιβεβαίωση Επιτυχούς Παραγγελίας.	163
Εικόνα 74: Στιγμιότυπο οθόνης Απόρριψης Παραγγελίας.	164
Εικόνα 75: Στιγμιότυπο οθόνης αναμονής απάντησης παραγγελίας.	164
Εικόνα 76: Στιγμιότυπο οθόνης της σελίδας Προφίλ Χρήστη.	165
Εικόνα 77: Στιγμιότυπο οθόνης της σελίδας Δήλωση; Ημερήσιου Μενού.	167
Εικόνα 78: Στιγμιότυπο οθόνης της σελίδας Διαχείρισης Παραγγελιών.	168
Εικόνα 79: Στιγμιότυπο οθόνης της σελίδας Διαχείρισης Voucher.	169
Εικόνα 80: Στιγμιότυπο οθόνης της σελίδας Διαχείρισης Timeslots.	170
Εικόνα 81: Στιγμιότυπο οθόνης της σελίδας Διαχείρισης Διακοπών.	171
Εικόνα 82: Στιγμιότυπο που δείχνει τα αποτελέσματα των προβλέψεων μερίδων φαγητού στην αντίστοιχη σελίδα, περιλαμβάνοντας τις τιμές πρόβλεψης, ελάχιστες και μέγιστες.	174
Εικόνα 83: Στιγμιότυπο οθόνης που παρουσιάζει την ένδειξη αναμονής για την εμφάνιση των αποτελεσμάτων πρόβλεψης μερίδων φαγητού.	174
Εικόνα 84: Ανάλυση της σελίδας “Daily Menu” με χρήση Google Lighthouse, παρουσιάζοντας επιδόσεις, προσβασιμότητα, βέλτιστες πρακτικές και SEO.	176
Εικόνα 85: Ανάλυση της σελίδας “Declare Menu” με χρήση Google Lighthouse, παρουσιάζοντας επιδόσεις, προσβασιμότητα, βέλτιστες πρακτικές και SEO.	177
Εικόνα 86: Στατιστικά συμβατότητας email για την ειδοποίηση αποδοχής παραγγελίας, παρουσιάζοντας την εμβέλεια σε διαφορετικούς πελάτες και πλατφόρμες.	180
Εικόνα 87: Στατιστικά συμβατότητας email για την ειδοποίηση απόρριψης παραγγελίας, παρουσιάζοντας την εμβέλεια σε διαφορετικούς πελάτες και πλατφόρμες.	180
Εικόνα 88: Αναφορά Spam και βαθμολογία παράδοσης για τα email συστήματος, αξιολογώντας την πιθανότητα αποστολής στα εισερχόμενα των χρηστών.	181

Σημείωση: Για λόγους μεγέθους και ανάλυσης, όλες οι εικόνες παρουσιάζονται συνοπτικά και σε σμίκρυνση στο παρόν έγγραφο. Οι πλήρεις εκδόσεις υψηλής ανάλυσης είναι διαθέσιμες στο Συμπληρωματικό Υλικό μέσω του αποθετηρίου Github: [aboutalis/thesis-figures].

Περιεχόμενα πινάκων

Πίνακας 1: Συγκεντρωτικός πίνακας σύγκρισης μεταξύ των Wireframe A, B και C. _____	43
Πίνακας 2: Λίστα API endpoints του συστήματος με τη μέθοδο, την περιγραφή λειτουργίας και τα δικαιώματα πρόσβασης. _____	86
Πίνακας 3: Παράδειγμα εμφάνισης αποτελεσμάτων του μηχανισμού πρόβλεψης για τρία φαγητά, με τιμή πρόβλεψης, ελάχιστη και μέγιστη τιμή. _____	136
Πίνακας 4: Βασικές μετρικές αξιολόγησης του μοντέλου πρόβλεψης μερίδων φαγητού (MAE, RMSE, MAPE). _____	142
Πίνακας 5: Παρουσίαση των αποτελεσμάτων πρόβλεψης για 50 φαγητά, με in-sample αξιολόγηση των μετρικών MAE, RMSE και MAPE, για εκτίμηση της ακρίβειας του μοντέλου. _____	146
Πίνακας 6: Αποτελέσματα out-of-sample πρόβλεψης για 150 φαγητά, με αξιολόγηση των μετρικών MAE, RMSE και MAPE. _____	151
Πίνακας 7: Συγκριτικός πίνακας επιδόσεων όλων των δοκιμών πρόβλεψης, με μετρικές MAE, RMSE και MAPE για διαφορετικά πλήθη προϊόντων και τύπους πρόβλεψης. _____	155
Πίνακας 8: Αναλυτικά αποτελέσματα αξιολόγησης της εφαρμογής με χρήση Google Lighthouse, περιλαμβάνοντας επιδόσεις, προσβασιμότητα, βέλτιστες πρακτικές και SEO. _____	176

1. Εισαγωγή

Η σίτιση των φοιτητών στα πανεπιστήμια αποτελεί ένα βασικό ζήτημα της καθημερινότητάς τους, επηρεάζοντας την ποιότητα ζωής, την υγεία και τη συνολική τους εμπειρία. Η διαχείριση μιας φοιτητικής λέσχης είναι σύνθετη διαδικασία, καθώς περιλαμβάνει την οργάνωση των γευμάτων, την εξυπηρέτηση μεγάλου αριθμού ατόμων, την ορθή κατανομή των πόρων, και την ανάγκη για ευελιξία και ταχύτητα στην εξυπηρέτηση.

Στο πλαίσιο αυτό, η παρούσα εργασία εστιάζει στη δημιουργία ενός διαδικτυακού συστήματος παραγγελιών φαγητού ειδικά σχεδιασμένου για φοιτητικές λέσχες. Η εφαρμογή αυτή έχει ως κύριο στόχο την παροχή μιας φιλικής, ανθρωποκεντρικής εμπειρίας τόσο για τον τελικό χρήστη (φοιτητή), όσο και για τον διαχειριστή της λέσχης, προσφέροντας ευκολία, διαφάνεια και αποδοτικότητα.

1.1 Σκοπός και Στόχοι της Εργασίας

Σκοπός της εργασίας είναι η ανάπτυξη ενός ολοκληρωμένου web app που να καλύπτει τις ανάγκες τόσο των φοιτητών όσο και των υπευθύνων της φοιτητικής λέσχης, με κύριο γνώμονα τη βελτιστοποίηση της εμπειρίας χρήσης και την αποδοτικότερη διαχείριση των διαθέσιμων πόρων.

Οι βασικοί στόχοι της εργασίας περιλαμβάνουν:

- Τη μείωση ή εξάλειψη των ουρών στη λέσχη μέσω ηλεκτρονικού συστήματος κρατήσεων. Τη δυνατότητα προβολής του μενού σε πραγματικό χρόνο από τους φοιτητές και εύκολης καταχώρησης παραγγελιών.
- Τη δυνατότητα για live ενημέρωση σχετικά με την πληρότητα της λέσχης, ώστε οι χρήστες να προγραμματίζουν αποτελεσματικότερα τις επισκέψεις τους.
- Τη διευκόλυνση του έργου του διαχειριστή μέσω εργαλείων για διαχείριση μενού, παραγγελιών και ροής επισκεπτών.
- Την ενσωμάτωση ενός μηχανισμού πρόβλεψης μερίδων, που βασίζεται σε ιστορικά δεδομένα, καιρικές συνθήκες και αργίες, για την καλύτερη αξιοποίηση των πόρων και την αποφυγή σπατάλης.

Αξίζει να σημειωθεί ότι η ανάγκη για ένα τέτοιο σύστημα καθίσταται ακόμη πιο επιτακτική, καθώς στο Πολυτεχνείο Κρήτης παρατηρούνται συχνά ουρές άνω των 15 λεπτών στις ώρες αιχμής, γεγονός που υποβαθμίζει την εμπειρία των φοιτητών και οδηγεί σε φαινόμενα συμφόρησης, καθυστερήσεων και σπατάλης πόρων.

1.2 Κίνητρα και Σημασία του Έργου

Η ανάγκη για ψηφιοποίηση και εκσυγχρονισμό των υπηρεσιών σίτισης σε ακαδημαϊκά ιδρύματα αποτελεί το βασικό κίνητρο για την παρούσα εργασία. Το υπάρχον σύστημα λειτουργίας πολλών φοιτητικών λεσχών βασίζεται σε χειροκίνητες διαδικασίες, με αποτέλεσμα τη δημιουργία ουρών, την ανασφάλεια σχετικά με τη διαθεσιμότητα φαγητού και την αδυναμία πρόβλεψης της ζήτησης.

Η σημασία του έργου είναι πολλαπλή:

- Ανθρωποκεντρικός σχεδιασμός: Η εφαρμογή εστιάζει στις πραγματικές ανάγκες των φοιτητών και των διαχειριστών, βελτιώνοντας την καθημερινότητά τους.
- Πράσινη τεχνολογία: Μέσω της πρόβλεψης μερίδων, επιτυγχάνεται μείωση της σπατάλης τροφίμων και καλύτερη αξιοποίηση των αποθεμάτων.
- Ψηφιακός μετασχηματισμός: Το έργο συνεισφέρει στον εκσυγχρονισμό των υπηρεσιών των ιδρυμάτων και την εναρμόνισή τους με τις σύγχρονες ανάγκες και τεχνολογίες.
- Αμεσότητα και προσβασιμότητα: Οι χρήστες μπορούν να έχουν πρόσβαση στην υπηρεσία από οποιαδήποτε συσκευή, οποιαδήποτε στιγμή, διευκολύνοντας την καθημερινότητά τους.

2. Ανάλυση Αναγκών και Περιγραφή Προβλήματος

Η κατανόηση των αναγκών τόσο των φοιτητών όσο και των διαχειριστών της φοιτητικής λέσχης αποτελεί απαραίτητη προϋπόθεση για τη σχεδίαση ενός πληροφοριακού συστήματος που θα ανταποκρίνεται αποτελεσματικά στις απαιτήσεις της καθημερινής λειτουργίας. Στο παρόν κεφάλαιο περιγράφεται η υπάρχουσα κατάσταση, εντοπίζονται τα προβλήματα του παρόντος συστήματος και διαμορφώνονται οι απαιτήσεις για το νέο, προτεινόμενο σύστημα.

2.1 Περιγραφή Υφιστάμενης Κατάστασης

Σήμερα, η διαδικασία ενημέρωσης και εξυπηρέτησης των φοιτητών στη λέσχη βασίζεται σε έναν βασικό πίνακα ανακοινώσεων στην ιστοσελίδα του Πολυτεχνείου, όπου αναρτάται ένα αρχείο PDF με το «Προτεινόμενο Εβδομαδιαίο Πρόγραμμα Σίτισης» [1]. Το αρχείο περιλαμβάνει προκαθορισμένα γεύματα και δείπνα για τέσσερις διαφορετικές εβδομάδες (1η, 2η, 3η και 4η), οι οποίες επαναλαμβάνονται κάθε μήνα. Έτσι, οι φοιτητές μπορούν να ενημερώνονται για το ημερήσιο μενού με βάση την τρέχουσα εβδομάδα του μήνα. Ωστόσο, το πρόγραμμα αυτό δεν ενημερώνεται δυναμικά, με αποτέλεσμα να μην αντικατοπτρίζει πιθανές αλλαγές ή εξαντλήσεις πιάτων κατά τη διάρκεια της ημέρας, γεγονός που μπορεί να οδηγήσει σε λανθασμένο προγραμματισμό και περιττές επισκέψεις στη λέσχη.

ΠΡΟΤΕΙΝΟΜΕΝΟ ΕΒΔΟΜΑΔΙΑΙΟ ΠΡΟΓΡΑΜΜΑ ΣΙΤΙΣΗΣ
ΓΕΥΜΑΤΑ 1^{ης} ΕΒΔΟΜΑΔΑΣ

Γεύμα		ΔΕΥΤΕΡΑ	ΤΡΙΤΗ	ΤΕΤΑΡΤΗ	ΠΕΜΠΤΗ	ΠΑΡΑΣΚΕΥΗ	ΣΑΒΒΑΤΟ	ΚΥΡΙΑΚΗ
Κυρίως		Μαοχάρι κοκκινιστό	Κοτόπουλο ψητό με μουστάρδα	Ψάρι ανατολίτικο	Χοιρινό με μανιτάρια	Ψάρι στο φούρνο	Μπριζόλα Φούρνου	Αρνί Φούρνου Σκορδάτο
		Μπριζόλα φούρνου	Χοιρινό σάμνις	Ψάρι σπεταιιώτα	Κοτόπουλο ψητό στο φούρνο	Ψάρι κοκκινιστό	Κοτόπουλο μπουτάκια φούρνου	Χοιρινό λεμονάτο
		Ψάρι πλάκι	Φακές	Σουτζουκάκια συμυρναϊκά	Ζυμαρικά με σως πιτταγιόλα	Πίτα ανάμεικτη	Φάβα	Πένες αραμπιάτα
		Ψάρι ριγανάτο	Φασολάκια λαδερά	Λουκάνικα Ριγανάτα	Ομελέτα φούρνου	Τυρόπιτα	Αρακάς με πατάτες	Βίδες Αλά κρέμ
		Παστίσιο	Ζαμπονόπιτα	Πίτσα Γαλοπούλα	Ρεβίθια	Μακαρόνια Ντι όλιο	Μακαρόνια με κόκκινη σάλτσα	Γεμιστά
		Κεφτέδες Εσπανιώς	Τυρόπιτα	Πίτσα Λαχανικών	Ρατατούι	Πίτσα Σπέσιαλ	Καρμπονάρα	Ρεβίθια Ντερμπιγιέ
		Πίτα ανάμεικτη	Πίτσα Σπέσιαλ	Μακαρόνια με κινά	Παστίσιο	Πίτσα Μαργαρίτα	Πίτσα Βεζουβίου	Πίτσα μανιτάρια
		Σπανακόπιτα	Πίτσα Μαργαρίτα	Πένες ριγκατόνι	Γουβαρλάκια	Σπανακόρυζο	Πίτσα Ζαμπόν –Τυρί	Πίτσα Σπέσιαλ
Πλάτο Ειδικής Διατροφής	Κρητικής	Ποικιλία με Καλτσούνια Φούρνου	Σφουγγάτο (Αυγό - ντομάτα - λαχανικά)	Χόρτα Γιαχνί	Ποικιλία με Καλτσούνια Τηγανιτά	Πένες με ανθότυρο	Αγκινάρες με κουκιά	Κρεατόπιτα
	Fitness	Ψάρι ριγανάτο & Πατάτες με ελαιόλαδο και Ριγανή	Κοτόπουλο ψητό με μουστάρδα & Αρακάς Ανάμεικτος	Μπριάμ & Φέτα	Κοτόπουλο ψητό στο φούρνο & Βραστά Λαχανικά	Φακές & Φέτα	Μπριζόλα Φούρνου & Ρυζί σιμού	Γεμιστά & Γιάουρτι
Γαρνιτούρα		Πατάτες με ελαιόλαδο και Ριγανή	Πατάτες μπουλανσέρ	Πατάτες Φούρνου Λεμονάτες	Βραστά Λαχανικά	Πατάτες λεμονάτες	Ρυζί σιμού	Ρυζί με λαχανικά
		Ρυζί σιμού	Μακαρονάκι Βουτύρου	Ρυζί με λαχανικά	Ρυζί με κάρυ	Ρυζί πιλάφι	Πατάτες φούρνου με σκόρδο	Γιαούρτι
		Κριθαράκι κοκκινιστό	Αρακάς Ανάμεικτος	Βραστά Λαχανικά	Τηγανητά Λαχανικά	Βραστά Λαχανικά	Πατάτες Τηγανητές	Κριθαράκι Βουτύρου
Σαλάτα		Αγγούρι Ντομάτα Λάχανο Μαρούλι Μελιτζανοσαλάτα	Μαρούλι Μπρόκολο Πατατοσαλάτα Πολίτικη Ρώσικη	Ταραμάς με ελιές Κουνουπίδι Παντζάρια Μαρούλι Τυροσαλάτα	Λάχανο Αγγούρι Ντομάτα Μαρούλι Χόρτα	Ταραμάς – ελιές Λάχανο Χωριάτικη Τονοσαλάτα Λαχανικά ανάμεικτα Ντάκος	Λάχανο Χωριάτικη Τονοσαλάτα Λαχανικά ανάμεικτα Ντάκος	Χωριάτικη Λάχανο Μαρούλι Ταραμά με Ελιές Μαρούλι
Φρούτο		2 Επιλογές	2 Επιλογές	2 Επιλογές	2 Επιλογές	2 Επιλογές	2 Επιλογές	2 Επιλογές
Γλυκό		Πάστα	Εκλερ	Κρουασάν Πραλίνας	Πάστα	Κρουασάν Πραλίνας	Πάστα	Σιροπιαστά
		Μπουγάτσα	Κρουασανάκια	Λυχναράκια	Κρουασανάκια	Πάστα	Κικ	Πάστα
Φέτα/ Τυρί Γραβιέρα		Φέτα	Φέτα	Φέτα	Φέτα	Φέτα	Φέτα	Γραβιέρα

Εικόνα 1: Προτεινόμενο Εβδομαδιαίο πρόγραμμα σίτισης για τα γεύματα της 1ης εβδομάδας.

ΔΕΙΠΝΑ 1^{ης} ΕΒΔΟΜΑΔΑΣ

Δείπνο	ΔΕΥΤΕΡΑ	ΤΡΙΤΗ	ΤΕΤΑΡΤΗ	ΠΕΜΠΤΗ	ΠΑΡΑΣΚΕΥΗ	ΣΑΒΒΑΤΟ	ΚΥΡΙΑΚΗ
Κυρίως	Σουπιές με μάρθα	Παστίσιο	Σουπιές Κοκκινιστό	Μουσάκα	Ψάρι Σκορδάτο	Μπουτάκια κοτόπουλο με σως σασέρ	Γουβαρλάκια
	Ψάρι αλα σπεταιιώτα	Μπιφτέκια στο φούρνο	Χιτσόδι Κραστό	Μπιφτέκια του σεφ	Σουπιές Γιαχνί	Σνίτσελ	Κεφτεδάκια με κοκ. Σάλτσα
	Ομελέτα φούρνου	Πίτσα Λαχανικών	Κοτόπουλο στο φούρνο	Πίτσα Τυρί	Μακαρόνια καρμπονάρα	Μελιτζάνες προβενσάλ	Μακαρόνια με σάλτσα κοκ.
	Σουφλέ ζυμαρικών με αλλαντικά	Πίτσα Σπέσιαλ	Χοιρινό Σουβλάκι	Πίτσα Σπέσιαλ	Κικ λωρεν (ομελέτα)	Φασόλες γίγαντες	Ταλιατέλες αλά κρέμ
	Πίτσα Γαλοπούλα	Μπάμιες Κοκκινιστές	Μακαρόνια Ναπολιτέν	Γεμιστά	Μπριάμ	Πίτσα Λαχανικών	Πίτσα σπέσιαλ
	Πίτσα Λαχανικών	Φακόρυζο	Βίδες Ριγκατόνι	Φασολάκια λαδερά	Μπουρέκι	Πίτσα Σπέσιαλ	Πίτσα Τυρί
		Πένες 4 Τυριά					
Γαρνιτούρα	Πατάτες φούρνου	Ρυζί σιμού	Ρυζί με λαχανικά	Πατάτες Τηγανητές	Πατάτες φούρνου	Κριθαράκι Βουτύρου	Πατάτες ριγανάτες
	Μακαρονάκι κοκκινιστό	Πατάτες φούρνου με κρεμμυδάκι	Κριθαράκι Mexican	Βραστά Λαχανικά	Μακαρονάκι Φούρνου Κοκκινιστό	Πατάτες Τηγανητές	Αρακάς Ανάμεικτος
Σαλάτα	Πατατοσαλάτα	Σαλάτα με ζυμαρικά	Αγγούρι	Μελιτζανοσαλάτα	Αγγούρι	Αγγούρι	Ντομάτα
	Μαρούλι	Ντομάτα	Ρυζοσαλάτα	Ντομάτα	Λάχανο	Πατατοσαλάτα	Πολίτικη
	Αγγούρι	Παντζάρια	Ρώσικη	Παντζάρια	Λαχανικά ανάμεικτα	Ντομάτα	Χωριάτικη
	Ντομάτα	Μαρούλι	Λάχανο	Λάχανο	Ντομάτα	Μαρούλι	Αγγούρι
	Τυροσαλάτα	Λάχανο	Κουνουπίδι	Αγγούρι	Ταραμοσαλάτα	Ντάκος	Χόρτα
Φρούτο	2 Επιλογές	2 Επιλογές	2 Επιλογές	2 Επιλογές	2 Επιλογές	2 Επιλογές	2 Επιλογές
Γλυκό	Κρουασάν Πραλίνας	Πάστα	Πάστα	Κρουασάν Πραλίνας	Πάστα	Σιροπιαστά	Πάστα
	Τρούφες	Εκλερ	Σιροπιαστά	Μπουγάτσα	Παγωτό	Μυζηθοροπιακία Γλυκά	Κρουασάν Πραλίνας

Εικόνα 2: Προτεινόμενο Εβδομαδιαίο πρόγραμμα σίτισης για τα δείπνα της 1ης εβδομάδας.

Η κατάσταση επιδεινώνεται κατά τις ώρες αιχμής, όπου παρατηρούνται αναμονές που ξεπερνούν τα 15 λεπτά. Αυτό συμβαίνει επειδή οι περισσότεροι φοιτητές επιλέγουν να σιτίζονται σε συγκεκριμένες χρονικές ζώνες (π.χ. μεταξύ 13:00 και 14:30), οδηγώντας σε συνωστισμό και καθυστερήσεις.

Από την πλευρά του διαχειριστή της λέσχης, η κατανομή των μερίδων γίνεται εμπειρικά, χωρίς τη χρήση μηχανισμών πρόβλεψης. Αυτό συχνά οδηγεί είτε σε σπατάλη φαγητού λόγω υπερπαραγωγής είτε σε ανεπαρκή ποσότητα και δυσaréσκεια των φοιτητών. Παράλληλα, διατηρείται μόνο ένα υποτυπώδες ιστορικό στοιχείων, το οποίο όμως δεν αξιοποιείται για την εξαγωγή χρήσιμων συμπερασμάτων, όπως ποια φαγητά έχουν τη μεγαλύτερη ζήτηση ή ποιες ημέρες και περίοδοι παρουσιάζουν αυξημένη προσέλευση.

Η απουσία αυτών των στοιχείων δυσκολεύει περαιτέρω τη διαχείριση των πόρων, είτε πρόκειται για προσωπικό είτε για προμήθειες πρώτων υλών, ενώ καθιστά αδύνατη την αναλυτική αξιολόγηση της αποτελεσματικότητας του σημερινού συστήματος.

2.2 Ανάγκες και Απαιτήσεις Χρηστών

Οι φοιτητές, ως βασικοί χρήστες του συστήματος, αναζητούν έναν πιο εύχρηστο, γρήγορο και προσαρμοσμένο στις ανάγκες τους τρόπο διαχείρισης της καθημερινής τους σίτισης. Η δυνατότητα άμεσης ενημέρωσης για το μενού της ημέρας είναι κρίσιμη για τον καθημερινό τους προγραμματισμό, ιδιαίτερα για εκείνους που ακολουθούν συγκεκριμένη διατροφή ή έχουν περιορισμένο χρονικό περιθώριο μεταξύ μαθημάτων. Επιπλέον, η δυνατότητα να κρατούν προσωρινά τις μερίδες που τους ενδιαφέρουν, τους επιτρέπει να οργανώνουν αποτελεσματικά τα γεύματά τους, είτε για κατανάλωση στον χώρο της λέσχης είτε για παραλαβή σε πακέτο.

Ένα σημαντικό στοιχείο που προκύπτει από την ανάλυση των αναγκών των χρηστών είναι η επιθυμία τους να γνωρίζουν την πληρότητα της λέσχης σε πραγματικό χρόνο. Η ενσωμάτωση ενός μηχανισμού παρακολούθησης της επισκεψιμότητας επιτρέπει στους φοιτητές να αποφεύγουν τις ουρές και να προσαρμόζουν το πρόγραμμά τους κατάλληλα, κερδίζοντας χρόνο και βελτιώνοντας τη συνολική τους εμπειρία.

Εξίσου σημαντική είναι η ανάγκη για πρόσβαση στο ιστορικό παραγγελιών, ώστε οι φοιτητές να μπορούν να παρακολουθούν τις προηγούμενες επιλογές τους, να αξιολογούν την ποιότητα των παρεχόμενων γευμάτων και να προσαρμόζουν τις διατροφικές τους συνήθειες. Η υποστήριξη όλων αυτών των λειτουργιών από κινητές συσκευές αποτελεί πλέον απαίτηση και όχι προαιρετικό χαρακτηριστικό.

Από την άλλη πλευρά, ο ρόλος του διαχειριστή της λέσχης είναι εξίσου κρίσιμος και απαιτεί ένα εργαλείο ευέλικτο, απομακρυσμένης διαχείρισης και πλήρους ελέγχου. Ο διαχειριστής πρέπει να

είναι σε θέση να ενημερώνει το μενού σε πραγματικό χρόνο, αφαιρώντας πιάτα που εξαντλούνται ή τροποποιώντας επιλογές ανάλογα με τη διαθεσιμότητα. Η δυνατότητα εποπτείας της πληρότητας ανά χρονική ζώνη (timeslot) διευκολύνει τη διαχείριση του χώρου και τη βέλτιστη κατανομή του προσωπικού.

Επιπλέον, η διαχείριση των ηλεκτρονικών κουπονιών (vouchers), η δυνατότητα τροποποίησης των χρονικών ζωνών, καθώς και η ενημέρωση των χρηστών για περιόδους διακοπών ή έκτακτες συνθήκες (π.χ. απεργίες, τεχνικά προβλήματα), αποτελούν βασικές λειτουργίες που το νέο σύστημα καλείται να υποστηρίξει.

Ένα από τα πλέον καθοριστικά εργαλεία για τη λειτουργία της λέσχης είναι ο μηχανισμός πρόβλεψης μερίδων. Ο στόχος του είναι να βοηθήσει τον διαχειριστή να αποφασίζει έγκαιρα και τεκμηριωμένα πόσες μερίδες από κάθε φαγητό πρέπει να παρασκευαστούν για μία συγκεκριμένη ημέρα. Η λειτουργία αυτή βασίζεται σε ένα σύστημα ανάλυσης ιστορικών δεδομένων κατανάλωσης, συνδυαστικά με παραμέτρους όπως η ημερομηνία, η ημέρα της εβδομάδας, οι καιρικές συνθήκες (θερμοκρασία, υγρασία, άνεμοι) καθώς και το είδος της ημέρας (εργάσιμη, εορταστική, περίοδος εξετάσεων). Το μοντέλο μαθαίνει από την προηγούμενη συμπεριφορά των χρηστών και προσπαθεί να εντοπίσει πρότυπα (patterns) στη ζήτηση.

Για παράδειγμα, μπορεί να διαπιστωθεί ότι συγκεκριμένα φαγητά προτιμώνται περισσότερο σε κρύες ή βροχερές ημέρες, ενώ άλλες επιλογές καταναλώνονται κυρίως σε περιόδους εξετάσεων. Παράλληλα, μπορεί να εντοπιστούν επαναλαμβανόμενες πτωτικές τάσεις τις Παρασκευές ή αυξημένη κίνηση στις αρχές κάθε μήνα.

Η έξοδος του μοντέλου είναι μία προτεινόμενη ποσότητα μερίδων για κάθε φαγητό, εντός ενός εύρους τιμών (π.χ. 50–70 μερίδες), ώστε να δίνεται στον διαχειριστή μια ευχέρεια επιλογής ανάλογα και με άλλες παραμέτρους (όπως το διαθέσιμο προσωπικό ή τυχόν εκδηλώσεις). Αυτό επιτρέπει τη σημαντική μείωση σπατάλης τροφίμων, την εξοικονόμηση πόρων και τη βελτίωση της ικανοποίησης των φοιτητών, οι οποίοι πλέον έχουν αυξημένες πιθανότητες να βρουν διαθέσιμο το φαγητό που επιθυμούν.

2.3 Προβλήματα που Εντοπίζονται στο Παρόν Σύστημα

Η υφιστάμενη διαδικασία σίτισης στο Πολυτεχνείο Κρήτης βασίζεται σε έναν απλό, στατικό μηχανισμό ενημέρωσης, ο οποίος δεν καλύπτει τις αυξημένες ανάγκες των φοιτητών και των διαχειριστών. Πιο συγκεκριμένα, οι χρήστες ενημερώνονται για το μενού της ημέρας μέσω ενός πίνακα στην ιστοσελίδα του ιδρύματος, ο οποίος δεν ανανεώνεται δυναμικά. Αυτό σημαίνει πως αν εξαντληθεί κάποιο φαγητό, η πληροφορία αυτή δεν εμφανίζεται πουθενά, οδηγώντας σε δυσαρέσκεια και κακή εμπειρία χρήστη, καθώς οι φοιτητές φτάνουν στη λέσχη χωρίς να βρίσκουν την επιθυμητή επιλογή.

Επιπλέον, κατά τις ώρες αιχμής, οι φοιτητές μπορεί να περιμένουν ακόμη και πάνω από 15 λεπτά στην ουρά, με αποτέλεσμα να χάνεται πολύτιμος χρόνος από το πρόγραμμά τους. Από την άλλη πλευρά, ο διαχειριστής της λέσχης παρασκευάζει μερίδες χωρίς να διαθέτει κάποιον αξιόπιστο μηχανισμό πρόβλεψης της ζήτησης, βασιζόμενος κυρίως στην εμπειρία και τη διαίσθηση. Αυτό οδηγεί είτε σε υπερπαραγωγή, με σπατάλη φαγητού και πόρων, είτε σε υποπαραγωγή, με αποτέλεσμα να μην εξυπηρετούνται όλοι οι φοιτητές.

Παράλληλα, δεν τηρείται ιστορικό δεδομένων, το οποίο θα μπορούσε να αναδείξει χρήσιμα πρότυπα κατανάλωσης, όπως οι προτιμήσεις ανά εποχή, ημέρα ή τύπο ημέρας (εργάσιμη, αργία, εξετάσεις). Αυτή η έλλειψη δεδομένων και ψηφιοποίησης περιορίζει σοβαρά τις δυνατότητες βελτιστοποίησης της λειτουργίας της λέσχης και της εμπειρίας των φοιτητών.

Τέλος, δεν υπάρχει μηχανισμός ζωντανής πληροφόρησης για τον φόρτο στη λέσχη, με αποτέλεσμα να συγκεντρώνεται πολύς κόσμος τις ίδιες ώρες, χωρίς να υπάρχει η δυνατότητα βελτιωμένου προγραμματισμού από την πλευρά των χρηστών. Όλα τα παραπάνω καθιστούν την υπάρχουσα κατάσταση αναποτελεσματική και ασύμφορη, τόσο για τη φοιτητική κοινότητα όσο και για τη διοίκηση της λέσχης.

2.4 Επιθυμητές Λειτουργίες του Νέου Συστήματος / Απαιτήσεις Βελτίωσης

Το προτεινόμενο σύστημα φιλοδοξεί να μετασχηματίσει πλήρως την εμπειρία σίτισης στο Πολυτεχνείο Κρήτης μέσω της ανάπτυξης μίας μοντέρνας, ψηφιακής εφαρμογής με δυνατότητες που ανταποκρίνονται στις ανάγκες τόσο των φοιτητών όσο και των διαχειριστών.

Για τους φοιτητές:

Οι χρήστες αποκτούν πρόσβαση σε μια εύχρηστη και γρήγορη εφαρμογή, από το κινητό ή τον υπολογιστή τους, μέσω της οποίας μπορούν:

- Να **ενημερώνονται δυναμικά** για το μενού της ημέρας, με live ανανέωση που εμφανίζει μόνο τα διαθέσιμα πιάτα.
- Να επιλέγουν αν επιθυμούν **να φάνε στον χώρο ή να τα παραλάβουν πακέτο**.
- Να **παρακολουθούν το ιστορικό των παραγγελιών τους**, ώστε να διατηρούν έλεγχο της διατροφής τους.
- Να βλέπουν σε πραγματικό χρόνο την **πληρότητα της λέσχης ανά χρονική ζώνη**, ώστε να επιλέγουν την κατάλληλη ώρα για φαγητό και να αποφεύγουν την αναμονή.
- Σε περίπτωση που δεν έχουν δικαίωμα σίτισης, μπορούν να χρησιμοποιούν **προπληρωμένες κάρτες (voucher)** για την ηλεκτρονική παραγγελία των γευμάτων τους. Τα voucher λειτουργούν ως ηλεκτρονικό ισοδύναμο χρήματος και επιτρέπουν την άμεση πληρωμή των γευμάτων μέσω της εφαρμογής, χωρίς την ανάγκη φυσικής συναλλαγής. Η εφαρμογή παρέχει τη δυνατότητα στους χρήστες να ελέγχουν το διαθέσιμο υπόλοιπο των voucher ανά πάσα στιγμή και να παρακολουθούν λεπτομερώς

το ιστορικό των συναλλαγών τους, εξασφαλίζοντας σαφήνεια και διαφάνεια στη χρήση τους.

Για τον διαχειριστή:

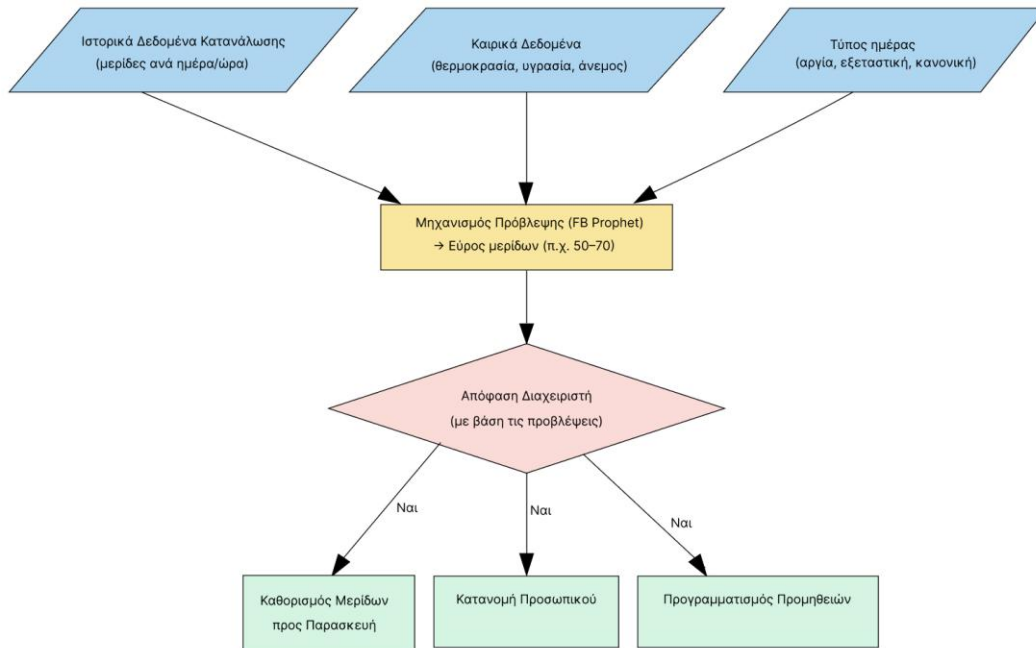
Από την πλευρά του, ο διαχειριστής αποκτά πρόσβαση σε έναν ενοποιημένο πίνακα ελέγχου, με δυνατότητες απομακρυσμένης διαχείρισης, που περιλαμβάνει:

- **Live ενημέρωση του μενού**, με δυνατότητα προσθήκης ή αφαίρεσης πιάτων σε πραγματικό χρόνο.
- **Εύκολη δήλωση του ημερήσιου μενού** μέσω μιας λειτουργίας **drag-and-drop**, με επιλογή φαγητών από ένα διαμορφωμένο γενικότερο μενού.
- **Διαχείριση του γενικού μενού**, με δυνατότητα **προσθήκης νέων φαγητών**, ορίζοντας όνομα, περιγραφή, εικόνα, διατροφικές πληροφορίες κ.ά., καθώς και **τροποποίηση ήδη υπαρχόντων πιάτων**.
- **Διαχείριση των ηλεκτρονικών κουπονιών (vouchers)** των φοιτητών.
- **Ορισμός χρονικών ζωνών**, περιόδων μη λειτουργίας (π.χ. διακοπές, αργίες, απεργίες).
- **Κατηγοριοποίηση των γευμάτων σε meal types** (τύπο γεύματος π.χ. πρωινό, μεσημεριανό, βραδινό), ώστε να υπάρχει σαφής διάκριση και ευκολία στη διαχείριση.
- **Έλεγχος της διαθεσιμότητας των παραγγελιών ανά meal type**, με δυνατότητα **άμεσης ενεργοποίησης ή απενεργοποίησης των παραγγελιών** ανάλογα με τον πραγματικό φόρτο και την κατάσταση στη λέσχη εκείνη τη στιγμή.
- **Εποπτεία της ζήτησης ανά χρονική ζώνη**, για βέλτιστη κατανομή προσωπικού και πόρων.

Κεντρικός άξονας του συστήματος αποτελεί ο μηχανισμός πρόβλεψης της ζήτησης, ο οποίος αξιοποιεί τον αλγόριθμο FB Prophet και ιστορικά δεδομένα (όπως ημερομηνία, καιρικές συνθήκες, τύπος ημέρας, προηγούμενες καταναλώσεις) για να παρέχει προβλέψεις για τον αριθμό μερίδων ανά φαγητό. Οι προβλέψεις αυτές δίνονται με εύρος τιμών (ελάχιστο – μέγιστο), λαμβάνοντας υπόψη αβεβαιότητες όπως:

- Απρόβλεπτες μεταβολές στον καιρό ή στον αριθμό παρόντων φοιτητών.
- Περίοδοι εξετάσεων, διακοπών ή έκτακτων γεγονότων.
- Εποχικές διακυμάνσεις στη ζήτηση.

Με βάση αυτές τις προβλέψεις, ο διαχειριστής μπορεί να σχεδιάσει στοχευμένα την παραγωγή, να αποφεύγει σπατάλες και να εξασφαλίζει ότι επαρκεί το φαγητό για όλους.



Εικόνα 3: Διάγραμμα Δραστηριοτήτων του διαχειριστή σε σχέση με τον μηχανισμό πρόβλεψης.

Το σύστημα, συνολικά, ψηφιοποιεί, οργανώνει και βελτιστοποιεί τη διαδικασία σίτισης, προσφέροντας μια λειτουργική εμπειρία σε χρήστες και προσωπικό, και εισάγοντας έξυπνα εργαλεία διαχείρισης, προβλεπτικότητας και διαφάνειας στη λειτουργία της λέσχης.

3. UI/UX Σχεδιασμός και Αξιολόγηση

Στο παρόν κεφάλαιο, αναλύονται οι τεχνολογίες και τα εργαλεία που επιλέχθηκαν για την ανάπτυξη της διαδικτυακής εφαρμογής παραγγελιοληψίας του πανεπιστημιακού εστιατορίου. Η επιλογή τους βασίστηκε στην αποτελεσματικότητα, την ευελιξία και την ευρεία υποστήριξη που προσφέρουν για την υλοποίηση μιας εφαρμογής full-stack (δηλαδή μιας εφαρμογής που περιλαμβάνει και το μέρος που βλέπουν οι χρήστες, και το μέρος που χειρίζεται τα δεδομένα και τις λειτουργίες στο παρασκήνιο), καθώς και στην ικανότητά τους να καλύψουν τις ειδικές απαιτήσεις του έργου, συμπεριλαμβανομένης της πρόβλεψης ποσοτήτων φαγητού.

3.1 Εισαγωγή – Ανάλυση βασικών εννοιών

Η εμπειρία χρήστη (User Experience – UX) και ο σχεδιασμός διεπαφής χρήστη (User Interface – UI) αποτελούν δύο από τους βασικότερους άξονες που καθορίζουν την επιτυχία ή αποτυχία μιας ψηφιακής εφαρμογής. Παρόλο που οι δύο όροι συχνά συγχέονται, πρόκειται για **διακριτές αλλά αλληλένδετες έννοιες**, οι οποίες όταν συνδυαστούν αποτελεσματικά, προσφέρουν μια πλήρη, ευχάριστη και λειτουργική εμπειρία στον τελικό χρήστη.

3.1.1 Τι είναι το UX (User Experience)

Η εμπειρία χρήστη (User Experience – UX) αναφέρεται στο σύνολο των συναισθημάτων, αντιλήψεων και αντιδράσεων που βιώνει ένας χρήστης κατά την αλληλεπίδρασή του με ένα ψηφιακό σύστημα. Δεν αφορά μόνο τη λειτουργικότητα, αλλά και το πόσο εύχρηστη, ευχάριστη και αποτελεσματική είναι η συνολική εμπειρία. Περιλαμβάνει παράγοντες όπως η ευκολία πλοήγησης, η ταχύτητα εκτέλεσης ενεργειών, η σαφήνεια της πληροφορίας, αλλά και η γενική εντύπωση που αφήνει το σύστημα στον χρήστη.

Ένα καλό UX σημαίνει ότι ο χρήστης μπορεί να επιτύχει τους στόχους του χωρίς σύγχυση ή απογοήτευση. Αντίθετα, ένα κακοσχεδιασμένο UX συχνά προκαλεί εκνευρισμό, οδηγεί σε σφάλματα και εν τέλει αποθαρρύνει τη συνέχιση της χρήσης της εφαρμογής. Σκοπός του UX είναι η δημιουργία μιας ολιστικής εμπειρίας που καθιστά τη χρήση ευχάριστη, κατανοητή και χωρίς εμπόδια.

3.1.2 Τι είναι το UI (User Interface)

Ο σχεδιασμός διεπαφής χρήστη (User Interface – UI) επικεντρώνεται στον οπτικό και διαδραστικό σχεδιασμό του συστήματος. Περιλαμβάνει όλα εκείνα τα στοιχεία με τα οποία αλληλεπιδρά ο

χρήστη: κουμπιά, μενού, εικονίδια, χρώματα, γραμματοσειρές και διάταξη περιεχομένου. Ο σχεδιασμός UI δίνει μορφή και αισθητική στην εφαρμογή, καθορίζοντας όχι μόνο την εμφάνισή της αλλά και το πόσο εύκολα μπορεί κάποιος να την κατανοήσει.

Ο στόχος του UI είναι να δημιουργήσει ένα περιβάλλον ελκυστικό, συνεπές και λειτουργικό. Ένα καλοσχεδιασμένο UI βοηθά τον χρήστη να πλοηγηθεί χωρίς προσπάθεια, να εντοπίσει τις απαραίτητες πληροφορίες και να χρησιμοποιήσει τις δυνατότητες του συστήματος με άνεση. Η αισθητική, σε αυτή την περίπτωση, δεν είναι απλώς θέμα ομορφιάς, αλλά εργαλείο που ενισχύει την κατανόηση και μειώνει τη γνωστική επιβάρυνση.

3.1.3 Γιατί είναι σημαντικός ο συνδυασμός UI & UX

Αν και οι έννοιες UI και UX είναι διαφορετικές, η αποτελεσματικότητα μιας εφαρμογής εξαρτάται από τον επιτυχημένο συνδυασμό τους. Ένα όμορφο UI δεν αρκεί αν η εμπειρία χρήσης είναι περίπλοκη ή ασαφής. Αντίστοιχα, ένα εξαιρετικά λειτουργικό UX χωρίς καλαισθητή και προσεγμένη διεπαφή μπορεί να φανεί πρόχειρο ή μη επαγγελματικό στα μάτια του χρήστη.

Ο συγκεκρισμός των δύο προσφέρει ένα προϊόν που δεν είναι απλώς λειτουργικό, αλλά και ελκυστικό. Η αισθητική διευκολύνει την κατανόηση, ενώ η σωστή εμπειρία χρήσης ενισχύει την αφοσίωση των χρηστών. Όταν UI και UX σχεδιάζονται παράλληλα, το αποτέλεσμα είναι μια εφαρμογή που προσαρμόζεται στις ανάγκες και προσδοκίες του χρήστη, ελαχιστοποιεί τα λάθη και μεγιστοποιεί την αποτελεσματικότητα και την ικανοποίηση.

3.1.4 Πώς συμβάλλουν στο δικό μας έργο

Η εφαρμογή σίτισης που αναπτύσσεται στο πλαίσιο αυτής της εργασίας καλείται να εξυπηρετήσει δύο βασικές ομάδες χρηστών: τους φοιτητές και τους διαχειριστές. Για τους φοιτητές, είναι απαραίτητο το περιβάλλον να είναι φιλικό, ευδιάκριτο και άμεσα λειτουργικό. Οι χρήστες αυτοί χρειάζονται γρήγορη πρόσβαση στο μενού της ημέρας, τη δυνατότητα κράτησης και ενημέρωση για την κατάσταση των γευμάτων τους. Η σχεδίαση UX διασφαλίζει ότι η πλοήγηση είναι απλή και χωρίς περιττά βήματα, ενώ το UI φροντίζει ώστε όλα τα στοιχεία να είναι καθαρά, οικεία και προσαρμοσμένα στη λογική της καθημερινής χρήσης.

Από την άλλη πλευρά, οι διαχειριστές έχουν ανάγκη από ένα εργαλείο λειτουργικό, σταθερό και αποδοτικό. Μέσα από την κατάλληλη σχεδίαση, μπορούν να προσθέτουν ή να τροποποιούν πιάτα, να ελέγχουν τις κρατήσεις και να ενεργοποιούν ή να απενεργοποιούν την εφαρμογή σύμφωνα με το εκάστοτε πρόγραμμα λειτουργίας. Η διαφοροποίηση στο UI/UX ανάλογα με τον ρόλο των χρηστών υπήρξε καθοριστική, καθώς η λειτουργικότητα και οι ανάγκες τους διαφέρουν σημαντικά.

Η επιλογή να υιοθετηθεί ανθρωποκεντρική προσέγγιση (Human-Centered Design) έπαιξε καταλυτικό ρόλο στην επιτυχία του σχεδιασμού. Οι σχεδιαστικές αποφάσεις βασίστηκαν σε ανατροφοδότηση από πραγματικούς χρήστες, μέσα από ερωτηματολόγια και δοκιμαστικά σενάρια χρήσης. Η συνεχής αξιολόγηση και προσαρμογή των wireframes (απλών σχεδιαστικών προσχεδίων που απεικονίζουν τη βασική δομή και λειτουργικότητα των οθονών) και των διαδραστικών στοιχείων οδήγησε σε ένα τελικό αποτέλεσμα που εξυπηρετεί ουσιαστικά τις ανάγκες όλων των εμπλεκόμενων χρηστών.

3.2 Εντοπισμός προβλήματος και ανάλυση απαιτήσεων χρηστών

Η πρώτη φάση του σχεδιασμού της εφαρμογής επικεντρώθηκε στην εις βάθος κατανόηση των πραγματικών προβλημάτων που αντιμετωπίζουν οι χρήστες κατά την αλληλεπίδρασή τους με το υφιστάμενο σύστημα σίτισης, καθώς και στον προσδιορισμό των απαιτήσεων που πρέπει να καλύψει η νέα ψηφιακή λύση. Πριν προχωρήσουμε στον σχεδιασμό της διεπαφής και των λειτουργικών ροών, κρίθηκε απαραίτητο να καταγραφούν συστηματικά οι ανάγκες των διαφορετικών τύπων χρηστών και να μελετηθεί η τρέχουσα εμπειρία χρήσης.

Τα βασικά προβλήματα που καταγράφηκαν και αναλύθηκαν εκτενώς στο Κεφάλαιο 2.3 – **Προβλήματα που Εντοπίζονται στο Παρόν Σύστημα** αφορούν κυρίως την έλλειψη πληροφόρησης, την απουσία προσωποποιημένων λειτουργιών και την έλλειψη δυναμικότητας του υφιστάμενου μηχανισμού για προσαρμογή στις ανάγκες της καθημερινής ροής φοιτητών και προσωπικού. Οι καθυστερήσεις, οι ουρές και η αδυναμία παρακολούθησης της πληρότητας ή της διαθεσιμότητας των γευμάτων αναδείχθηκαν ως κρίσιμα ζητήματα προς επίλυση.

Παράλληλα, οι λειτουργίες που θεωρούνται απαραίτητες για την επιτυχή υλοποίηση του νέου συστήματος παρουσιάζονται αναλυτικά στο Κεφάλαιο 2.4 – **Επιθυμητές Λειτουργίες του Νέου Συστήματος / Απαιτήσεις Βελτίωσης**. Οι απαιτήσεις αυτές προέκυψαν τόσο από την άμεση ανατροφοδότηση χρηστών όσο και από την εμπειρική παρατήρηση της υφιστάμενης κατάστασης.

Ο ρόλος του σχεδιασμού διεπαφής χρήστη (UI) και εμπειρίας χρήστη (UX) είναι ακριβώς να γεφυρώσει αυτό το χάσμα, μεταφράζοντας τις ανάγκες σε λειτουργικές, χρηστικές και ευχάριστες διεπαφές. Για τον σκοπό αυτό, προσδιορίστηκαν δύο κύριες κατηγορίες χρηστών:

- **Φοιτητές/Χρήστες:** Αναζητούν ένα γρήγορο, ξεκάθαρο και προσβάσιμο εργαλείο μέσω του οποίου να μπορούν να ενημερώνονται για το μενού, να προγραμματίζουν τα γεύματά τους, να διαχειρίζονται ηλεκτρονικά κουπόνια και να αποφεύγουν τις ουρές.
- **Διαχειριστές Λέσχης:** Απαιτούν ένα εύχρηστο και λειτουργικά ισχυρό backend για τη διαχείριση μενού, την εποπτεία πληρότητας, τον έλεγχο των παραγγελιών σε πραγματικό χρόνο και τη λήψη αποφάσεων βασισμένων σε δεδομένα (data-driven).

Η πληρέστερη εικόνα των προβλημάτων και των απαιτήσεων που συγκεντρώθηκε κατά τη φάση αυτή δεν θα αναλυθεί εκ νέου στο παρόν κεφάλαιο για αποφυγή επανάληψης. Ο αναγνώστης ενθαρρύνεται να ανατρέξει στα Κεφάλαια 2.3 και 2.4, όπου παρουσιάζονται λεπτομερώς οι βασικές δυσλειτουργίες του παρόντος συστήματος και οι επιθυμητές βελτιώσεις, οι οποίες αποτέλεσαν και τη βάση για τον επόμενο σχεδιαστικό κύκλο.

3.3 Δημιουργία Wireframes

Σε αυτή τη φάση του σχεδιασμού αναπτύχθηκε η αρχική δομή της εφαρμογής, με στόχο να αποτυπωθεί η οργάνωση των λειτουργιών και η ροή πλοήγησης. Η διαδικασία βασίστηκε στις λειτουργικές απαιτήσεις και στις ανάγκες των χρηστών που είχαν εντοπιστεί κατά την προηγούμενη ανάλυση. Δόθηκε έμφαση στη λογική διάταξη της πληροφορίας και στην ευκολία χρήσης, χωρίς να επικεντρώνεται η προσοχή σε αισθητικά στοιχεία.

3.3.1 Τι είναι τα Wireframes

Τα **wireframes** [2] αποτελούν βασικό εργαλείο στο σχεδιασμό διεπαφών χρήστη (UI), που επιτρέπει την απεικόνιση της αρχιτεκτονικής, της δομής και της οργάνωσης της πληροφορίας μίας εφαρμογής ή ιστοσελίδας σε πρώιμο στάδιο. Έχουν συνήθως χαμηλή πιστότητα (low-fidelity) και επικεντρώνονται στη λειτουργικότητα και στη λογική ροής των στοιχείων, όπως μενού, κουμπιά και περιεχόμενο, χωρίς να δίνεται έμφαση στην αισθητική και στα γραφικά.

Η χρήση wireframes βοηθάει τους σχεδιαστές και τους προγραμματιστές να συζητήσουν και να βελτιώσουν τη δομή και την εμπειρία χρήστη πριν ξεκινήσει η λεπτομερής οπτική σχεδίαση, μειώνοντας τον κίνδυνο αλλαγών σε προχωρημένο στάδιο.

3.3.2 Περιγραφή της διαδικασίας σχεδιασμού

Στη φάση του σχεδιασμού διεπαφής, βασιστήκαμε στις λειτουργικές απαιτήσεις και στις ανάγκες των χρηστών που προέκυψαν από την προηγούμενη ανάλυση, προκειμένου να αναπτύξουμε αρχικά wireframes που απεικόνιζαν την οργάνωση και τη δομή της εφαρμογής σε επίπεδο χαμηλής πιστότητας. Η επιλογή του εργαλείου **Balsamiq** [3] έγινε σκόπιμα, ώστε να επικεντρωθούμε στον σχεδιασμό της ροής και της λειτουργικότητας, χωρίς να αποσπώμαστε από την αισθητική λεπτομέρεια.

Συνολικά σχεδιάστηκαν **τρεις διαφορετικές προτάσεις wireframes**, καθεμία με διαφορετική προσέγγιση στη διάταξη και τη διαχείριση της πληροφορίας:

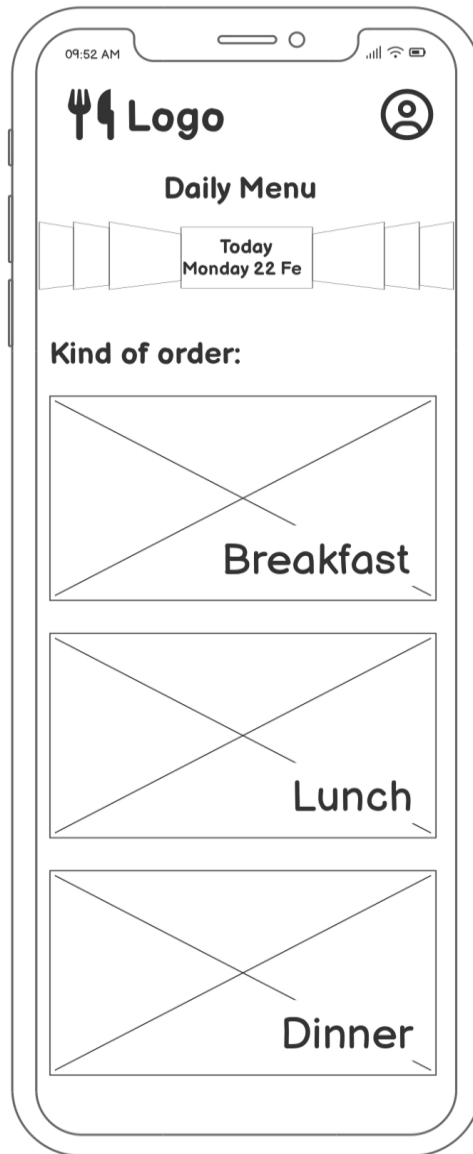
Η σχεδίαση των wireframes υλοποιήθηκε στο **Balsamiq**, ένα εργαλείο δημιουργίας wireframes χαμηλής πιστότητας (low-fidelity). Αυτό επέτρεψε την επικέντρωση στη βασική δομή και λειτουργικότητα χωρίς να αποσπάται η προσοχή από τον τελικό στόχο λόγω λεπτομερειών αισθητικής. Επιπλέον, σχεδιάστηκε η πλοήγηση μεταξύ των οθονών με απλά κουμπιά και συνδέσμους, ώστε να δοκιμαστεί η λογική ροή της εφαρμογής.

3.3.3 Wireframe A: Αρχική προσέγγιση – Μονοσέλιδη εμπειρία με οριζόντια διάταξη

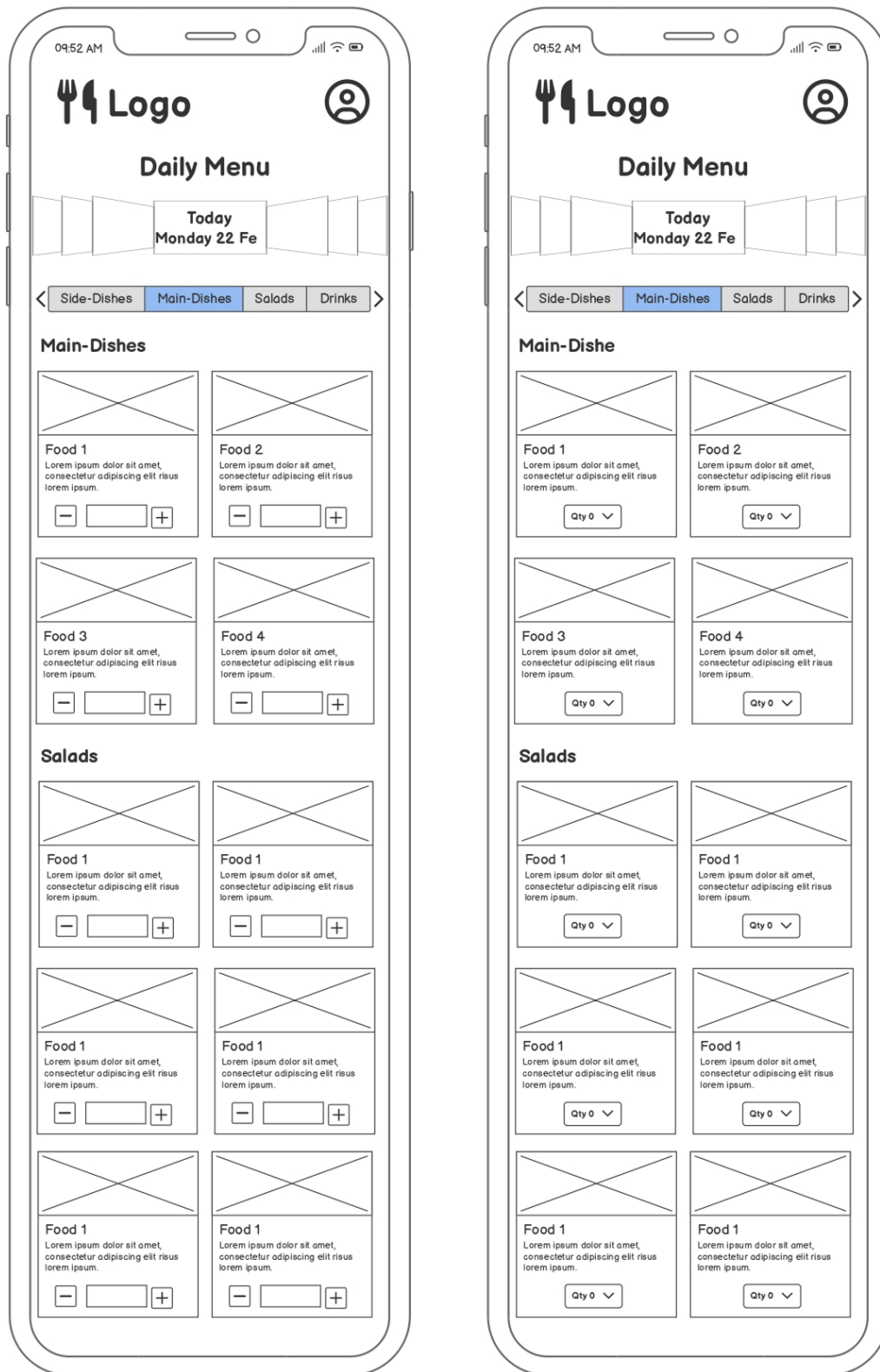
Αυτό το wireframe χαρακτηρίζεται από έναν καθαρό και λειτουργικό σχεδιασμό, με έμφαση στη σαφή κατηγοριοποίηση των γευμάτων, τη διαδοχική διαδικασία παραγγελίας και τη συνολική εμπειρία χρήστη.

- **Διάταξη:** Η δομή του περιλαμβάνει αρχική επιλογή είδους γεύματος, δυναμική επιλογή κατηγορίας (ορεκτικά, κυρίως πιάτα, σαλάτες, ποτά), καλάθι παραγγελίας με δυνατότητα προβολής και επεξεργασίας ποσοτήτων, και checkout με επιλογές για ωράριο, τύπο παραγγελίας και τρόπο πληρωμής. Επιπλέον, προσφέρεται η δυνατότητα εμφάνισης ιστορικού παραγγελιών και προφίλ χρήστη. Αυτό το wireframe στοχεύει να προσφέρει μια δομημένη και καθοδηγούμενη εμπειρία, κατάλληλη για χρήστες που θέλουν να εξετάσουν αναλυτικά όλες τις επιλογές πριν ολοκληρώσουν την παραγγελία τους.
- **Πλεονεκτήματα:** Τα πλεονεκτήματα του περιλαμβάνουν τη λεπτομερή οργάνωση του περιεχομένου, την ευκολία στην πλοήγηση ανά κατηγορία φαγητού, την προσωποποιημένη εμπειρία και την καταλληλότητά του για πιο σύνθετα περιβάλλοντα παραγγελιοληψίας.

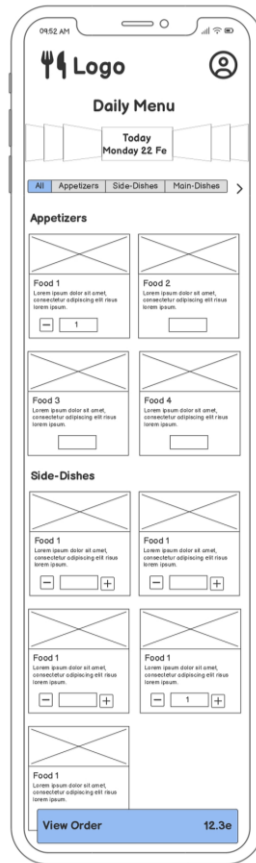
Στη συνέχεια παρατίθενται στιγμιότυπα οθόνης που απεικονίζουν τα wireframes και τις διαφορετικές οθόνες που δημιουργήθηκαν, προκειμένου να αποτυπωθεί πιο παραστατικά η προτεινόμενη εμπειρία χρήστη.



Εικόνα 4: Στιγμιότυπο οθόνης από το wireframe της αρχικής σελίδας, όπου ο χρήστης επιλέγει είδος γεύματος (Wireframe A).



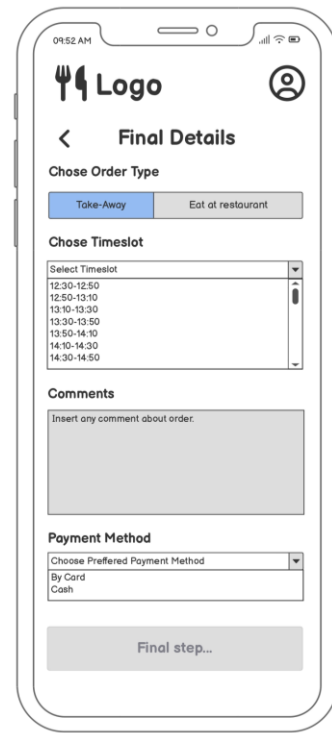
Εικόνα 5: Στιγμιότυπο οθόνης από το wireframe όπου απεικονίζεται η διάταξη του μενού (παρουσιάζονται δύο εκδοχές) (Wireframe A).



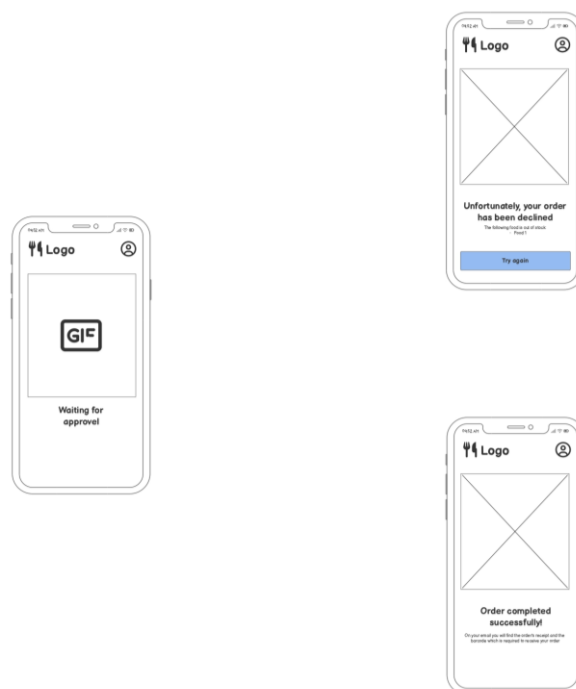
Εικόνα 6: Στιγμιότυπο οθόνης από το wireframe όπου απεικονίζεται η διάταξη του μενού, αφού ο χρήστης έχει επιλέξει κάποιο φαγητό (Wireframe A).



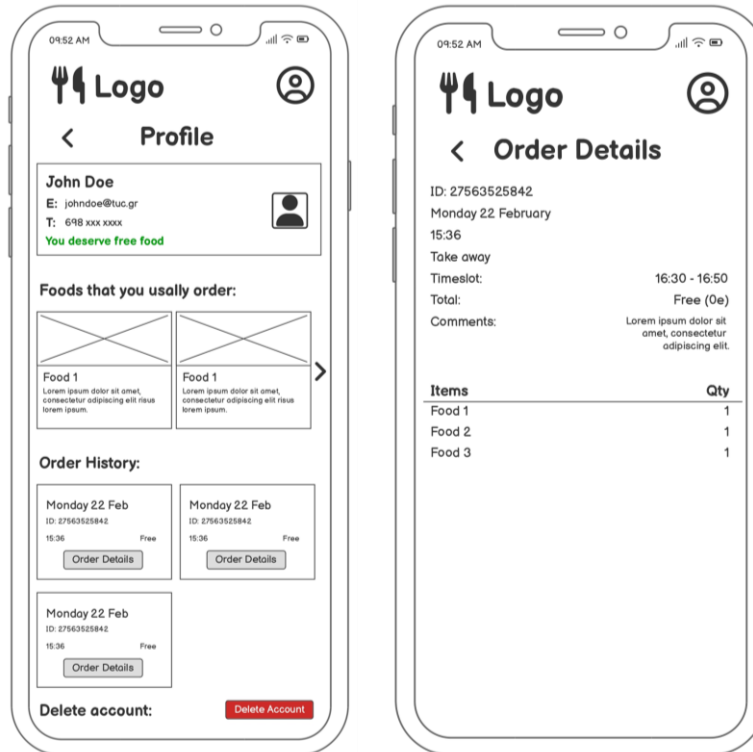
Εικόνα 7: Στιγμιότυπο οθόνης από το wireframe όπου απεικονίζεται η σύνοψη της παραγγελίας (παρουσιάζονται δύο εκδοχές) (Wireframe A).



Εικόνα 8: Στιγμιότυπο οθόνης από το wireframe όπου παρουσιάζονται οι τελικές επιλογές του χρήστη κατά το checkout: είδος φαγητού, χρονικό διάστημα (timeslot), σχόλια και τρόπος πληρωμής (Wireframe A).



Εικόνα 9: Στιγμιότυπο οθόνης από την οθόνη ενημέρωσης του χρήστη σχετικά με την κατάσταση της παραγγελίας (αποδοχή, απόρριψη ή αναμονή απάντησης) (Wireframe A).



Εικόνα 10: Στιγμιότυπο οθόνης από το προφίλ του χρήστη, όπου εμφανίζονται οι προσωπικές πληροφορίες, το ιστορικό voucher και το ιστορικό παραγγελιών. Στη δεξιά πλευρά προβάλλονται αναλυτικά τα στοιχεία της παραγγελίας όταν ο χρήστης επιλέγει να δει περισσότερες πληροφορίες (Wireframe A).

Οι πλήρεις εκδόσεις των εικόνων και των διαγραμμάτων που παρουσιάζονται συνοπτικά στο παρόν κεφάλαιο διατίθενται στο Συμπληρωματικό Υλικό [aboutalis/thesis-figures].

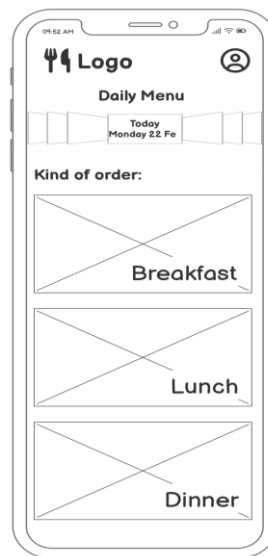
3.3.4 Wireframe B: Διαδραστική Κάρτα Φαγητού & Inline Επιλογές

Η δεύτερη έκδοση χαρακτηρίζεται από έναν πιο διαδραστικό και σύγχρονο σχεδιασμό, δίνοντας έμφαση στη χρήση overlay στοιχείων και inline επιλογών μέσα στην κάρτα κάθε προϊόντος.

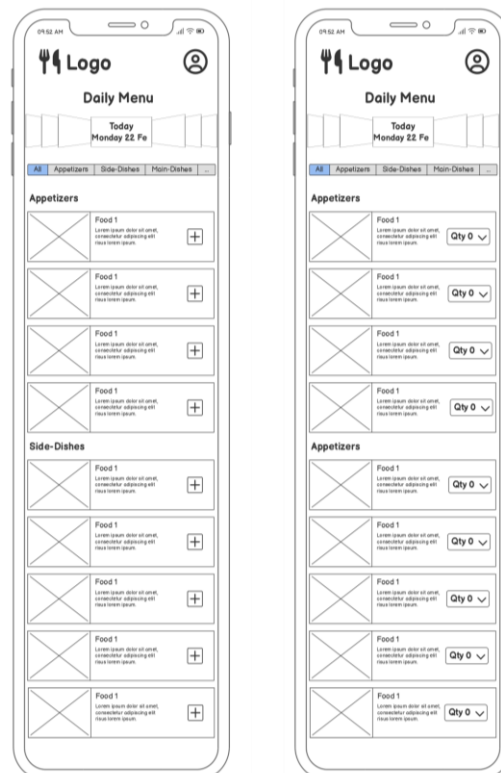
- **Διάταξη:** Οι χρήστες μπορούν να προσθέτουν προϊόντα στο καλάθι απευθείας μέσα από τις κάρτες, να επιλέγουν ποσότητα και να βλέπουν επιπλέον πληροφορίες χωρίς να αλλάζουν οθόνη. Η ροή περιλαμβάνει τα ίδια βήματα με τα υπόλοιπα wireframes, ωστόσο τα εκτελεί με πιο γρήγορο, διαδραστικό τρόπο. Ο στόχος του είναι να επιτρέπει άμεσες παραγγελίες, μειώνοντας τον αριθμό των ενεργειών του χρήστη και προσφέροντας μια σύγχρονη εμπειρία, ειδικά σε mobile-first περιβάλλοντα.
- **Πλεονεκτήματα:** Η ταχύτητα και η αποτελεσματικότητα, καθώς και η μοντέρνα αισθητική που το καθιστά ελκυστικό σε εξοικειωμένους χρήστες.

Στη συνέχεια παρατίθενται στιγμιότυπα οθόνης που απεικονίζουν τα wireframes και τις

διαφορετικές οθόνες που δημιουργήθηκαν, προκειμένου να αποτυπωθεί πιο παραστατικά η προτεινόμενη εμπειρία χρήστη.



Εικόνα 11: Στιγμιότυπο οθόνης από το wireframe της αρχικής σελίδας, όπου ο χρήστης επιλέγει είδος γεύματος (Wireframe B).

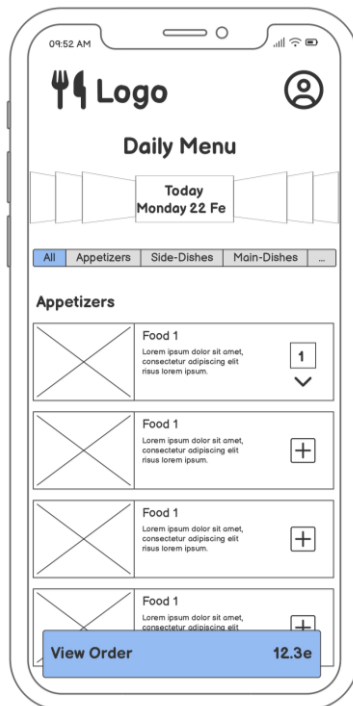


Εικόνα 12 : Στιγμιότυπο οθόνης από το wireframe όπου απεικονίζεται η

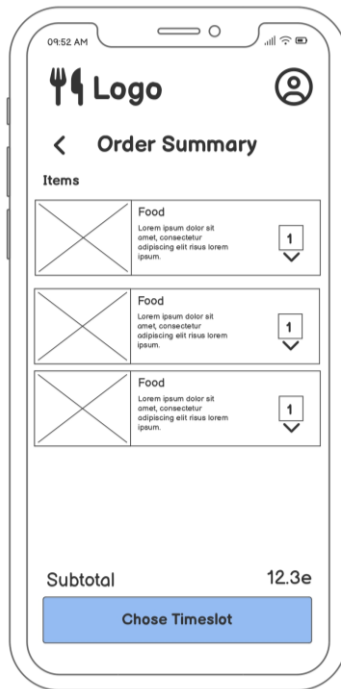
διάταξη του μενού (παρουσιάζονται δύο εκδοχές) (Wireframe B).



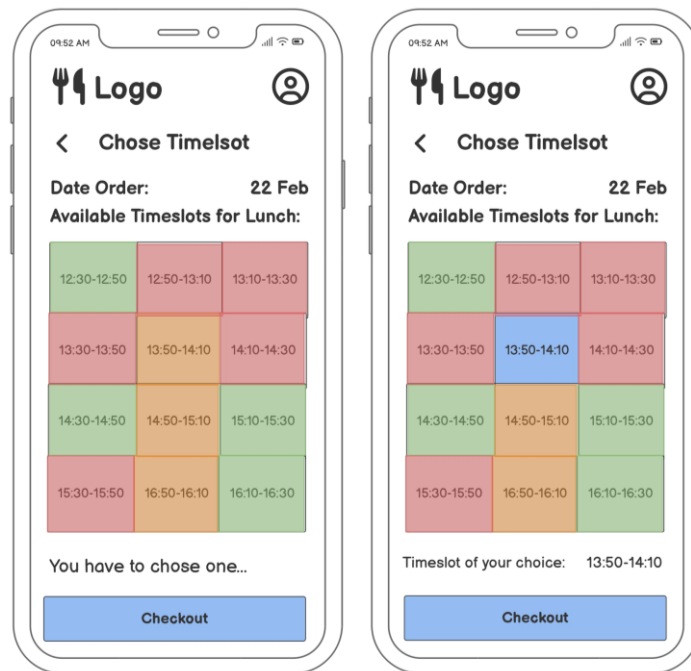
Εικόνα 13: Στιγμιότυπο οθόνης από το pop-up κατά την επιλογή φαγητού και την προβολή περισσότερων πληροφοριών. (Wireframe B).



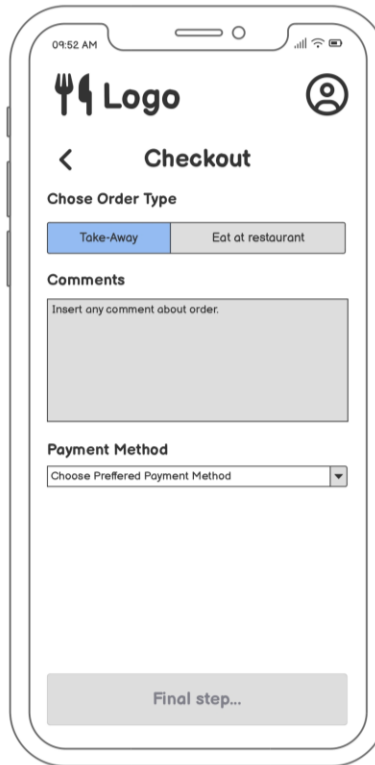
Εικόνα 14: Στιγμιότυπο οθόνης από το wireframe όπου απεικονίζεται η διάταξη του μενού, αφού ο χρήστης έχει επιλέξει κάποιο φαγητό (Wireframe B).



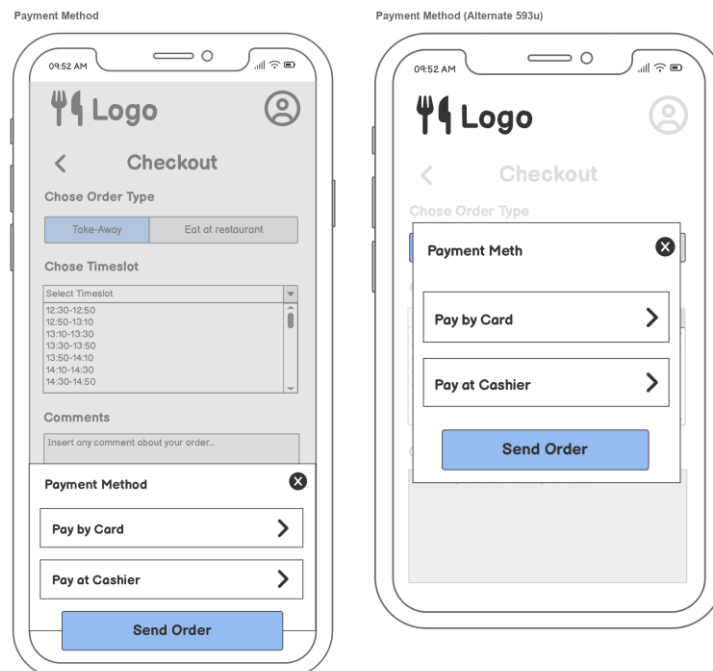
Εικόνα 15: Στιγμιότυπο οθόνης από το wireframe όπου απεικονίζεται η σύνοψη της παραγγελίας (Wireframe B).



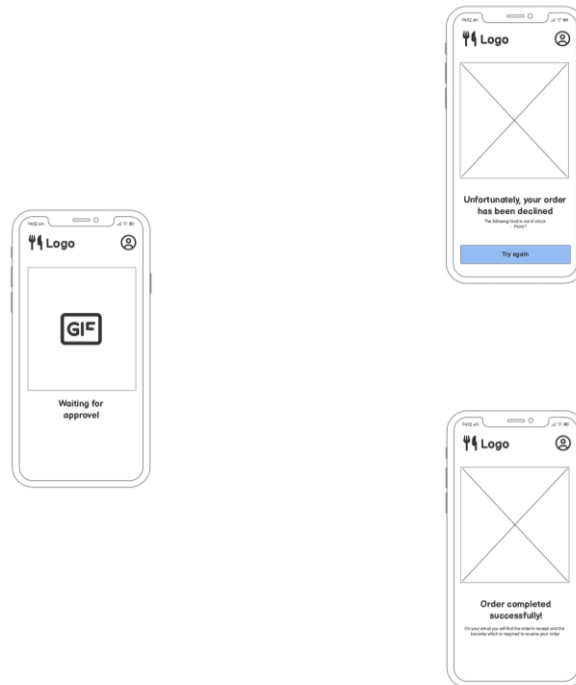
Εικόνα 16: Στιγμιότυπο οθόνης που απεικονίζει την επιλογή timeslot σε μορφή grid. Στη δεξιά εικόνα φαίνεται το layout με ένα επιλεγμένο timeslot (Wireframe B).



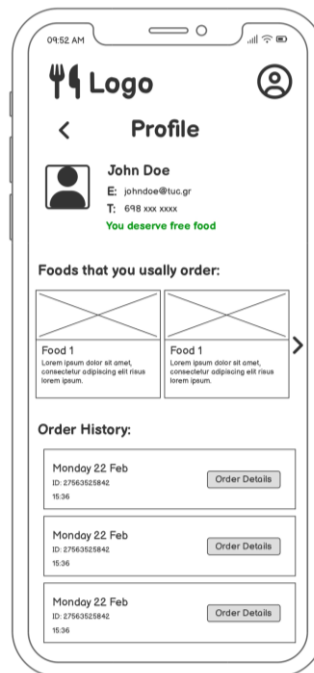
Εικόνα 17: Στιγμιότυπο οθόνης κατά το checkout, όπου ο χρήστης επιλέγει τις τελευταίες λεπτομέρειες: πού θα φάει, προσθέτει σχόλιο και επιλέγει τον τρόπο πληρωμής. (Wireframe B).



Εικόνα 18: Στιγμιότυπο οθόνης όπου ο χρήστης επιλέγει τον τρόπο πληρωμής (Wireframe B).



Εικόνα 19: Στιγμιότυπο οθόνης από την οθόνη ενημέρωσης του χρήστη σχετικά με την κατάσταση της παραγγελίας (αποδοχή, απόρριψη ή αναμονή απάντησης) (Wireframe B).



Εικόνα 20: Στιγμιότυπο οθόνης από το προφίλ του χρήστη, όπου εμφανίζονται οι προσωπικές πληροφορίες, το ιστορικό voucher και το ιστορικό παραγγελιών (Wireframe B).

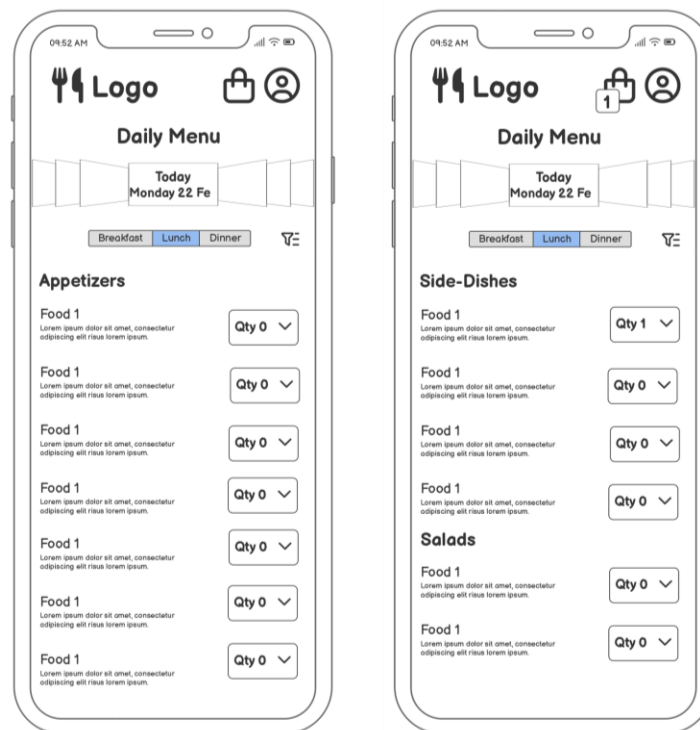
Οι πλήρεις εκδόσεις των εικόνων και των διαγραμμάτων που παρουσιάζονται συνοπτικά στο παρόν κεφάλαιο διατίθενται στο Συμπληρωματικό Υλικό [aboutalis/thesis-figures].

3.3.5 Wireframe C: Απλοποιημένη Παρουσίαση με Έξυπνα Φίλτρα

Η τρίτη έκδοση χαρακτηρίζεται από έναν μινιμαλιστικό αλλά πλήρως λειτουργικό σχεδιασμό, όπου η πλοήγηση βασίζεται σε φίλτρα κατηγοριών και καθαρές κάρτες προϊόντων.

- **Διάταξη:** Όλα τα βασικά βήματα της διαδικασίας παραγγελίας είναι παρόντα, αλλά οργανώνονται σε ένα πιο απλό και ταχύ περιβάλλον. Η χρήση φίλτρων κατηγοριών (π.χ. σαλάτες, ορεκτικά, ποτά) καθιστά εύκολη την πλοήγηση χωρίς να αλλάζει σελίδα, ενώ η προσθήκη στο καλάθι γίνεται άμεσα.
- **Πλεονεκτήματα:** Η συνολική εμπειρία είναι εστιασμένη στη λειτουργικότητα, με έμφαση στην ταχύτητα και την απλότητα, κάτι που το καθιστά ιδανικό για καθημερινούς ή τακτικούς χρήστες. Προσφέρει μια εμπειρία με λιγότερα βήματα και ελάχιστα οπτικά στοιχεία, εστιάζοντας στη χρηστικότητα.

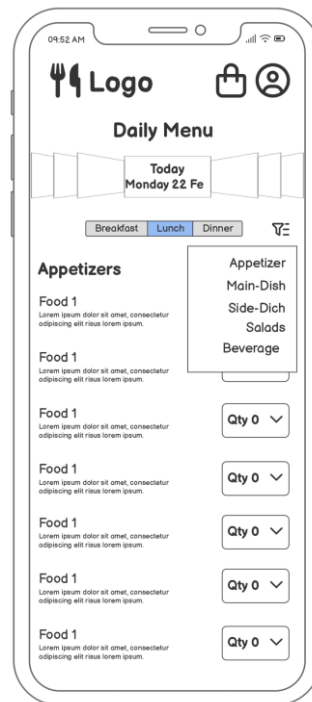
Στη συνέχεια παρατίθενται στιγμιότυπα οθόνης που απεικονίζουν τα wireframes και τις διαφορετικές οθόνες που δημιουργήθηκαν, προκειμένου να αποτυπωθεί πιο παραστατικά η προτεινόμενη εμπειρία χρήστη.



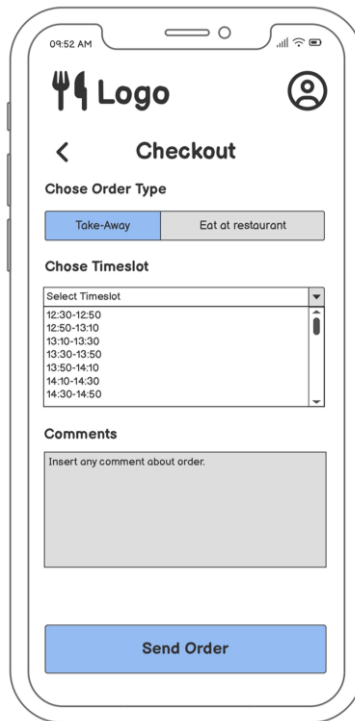
Εικόνα 21: Στιγμιότυπο οθόνης από το wireframe όπου απεικονίζεται η διάταξη του μενού. Στη δεξιά πλευρά φαίνεται η επιλογή του φαγητού (Wireframe C).



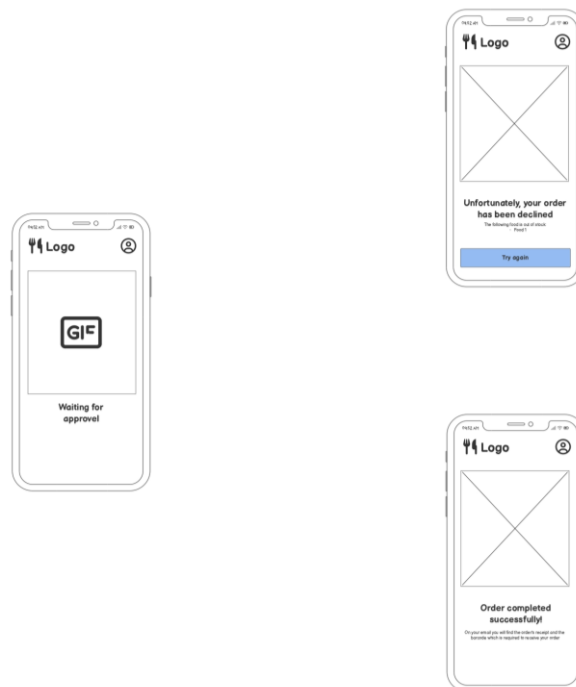
Εικόνα 22: Στιγμιότυπο οθόνης από το wireframe όπου απεικονίζεται η σύνοψη της παραγγελίας (Wireframe C).



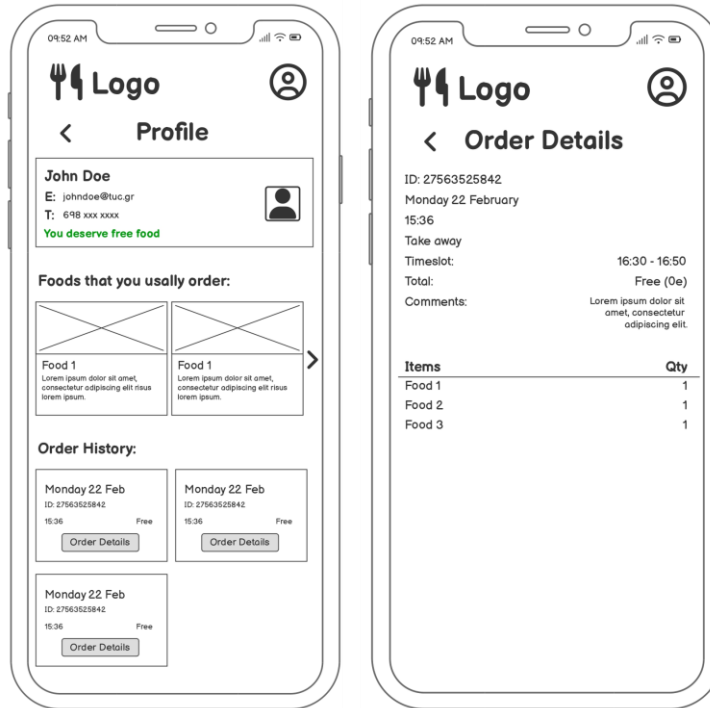
Εικόνα 23: Στιγμιότυπο οθόνης που απεικονίζει την αλλαγή του meal type στη σελίδα του μενού (Wireframe C).



Εικόνα 24: Στιγμιότυπο οθόνης από το wireframe όπου παρουσιάζονται οι τελικές επιλογές του χρήστη κατά το checkout: είδος φαγητού, χρονικό διάστημα (timeslot), σχόλια και τρόπος πληρωμής (Wireframe C).



Εικόνα 25: Στιγμιότυπο οθόνης από την οθόνη ενημέρωσης του χρήστη σχετικά με την κατάσταση της παραγγελίας (αποδοχή, απόρριψη ή αναμονή απάντησης) (Wireframe C).



Εικόνα 26: Στιγμιότυπο οθόνης από το προφίλ του χρήστη, όπου εμφανίζονται οι προσωπικές πληροφορίες, το ιστορικό voucher και το ιστορικό παραγγελιών. Στη δεξιά πλευρά προβάλλονται αναλυτικά τα στοιχεία της παραγγελίας όταν ο χρήστης επιλέγει να δει περισσότερες πληροφορίες (Wireframe C).

Οι πλήρεις εκδόσεις των εικόνων και των διαγραμμάτων που παρουσιάζονται συνοπτικά στο παρόν κεφάλαιο διατίθενται στο Συμπληρωματικό Υλικό [aboutalis/thesis-figures].

Συγκεντρωτικός Πίνακας Σύγκρισης

Χαρακτηριστικό	Wireframe A	Wireframe B	Wireframe C
Σχεδιαστικό ύφος	Κλασικό, καθοδηγούμενο	Μοντέρνο, διαδραστικό	Μινιμαλιστικό, πρακτικό
Πλοήγηση	Μετακίνηση ανά κατηγορία	Inline & overlay επιλογές	Φίλτρα κατηγοριών
Στόχος	Αναλυτική παραγγελία	Γρήγορη παραγγελία	Απλότητα και ταχύτητα
Κατάλληλο για	Πλήρες menu / αναλυτικοί χρήστες	Mobile / έμπειροι χρήστες	Τακτικοί χρήστες

Βαθμός εξατομίκευσης	Υψηλός	Μεσαίος	Χαμηλός
-------------------------	--------	---------	---------

Πίνακας 1: Συγκεντρωτικός πίνακας σύγκρισης μεταξύ των Wireframe A, B και C.

Η παρουσίαση των wireframes σε αυτή τη φάση αποτέλεσε σημαντικό βήμα για την επικοινωνία με τους τελικούς χρήστες και τη συλλογή ανατροφοδότησης, που οδήγησε στη βελτίωση των αρχικών ιδεών και την οριστικοποίηση του τελικού UI/UX σχεδιασμού.

3.4 Ανθρωποκεντρικός σχεδιασμός και iterative development

Ο Ανθρωποκεντρικός Σχεδιασμός (Human-Centered Design - HCD) [23] αποτελεί μια θεμελιώδη μεθοδολογία ανάπτυξης συστημάτων που τοποθετεί τον χρήστη στο επίκεντρο κάθε απόφασης. Ο πρωταρχικός του στόχος είναι η δημιουργία προϊόντων που δεν είναι απλώς λειτουργικά, αλλά ανταποκρίνονται με ακρίβεια στις πραγματικές ανάγκες, τις προσδοκίες και, κυρίως, στα συναισθηματικά και γνωστικά μοντέλα των τελικών χρηστών. Μέσω αυτής της προσέγγισης, διασφαλίζεται η υψηλή ευχρηστία, η αποδοχή και η αποτελεσματικότητα του τελικού προϊόντος.

Η υιοθέτηση του HCD είναι κρίσιμη για την αποφυγή σχεδιαστικών λαθών που συχνά οδηγούν στην αποτυχία μιας εφαρμογής. Πολλά συστήματα αποτυγχάνουν επειδή σχεδιάστηκαν με βάση υποθέσεις και όχι με βάση την πραγματική κατανόηση του χρήστη. Ο HCD μειώνει δραστικά τον κίνδυνο αυτό, περιορίζει το κόστος και τον χρόνο που απαιτείται για την επανόρθωση λαθών και επιταχύνει την υιοθέτηση του προϊόντος από το κοινό.

Τα βασικά χαρακτηριστικά του HCD είναι τα εξής:

- Ο επαναληπτικός χαρακτήρας (iterative cycles) αποτελεί την καρδιά του HCD. Η διαδικασία σχεδιασμού δεν είναι μια απλή, γραμμική ακολουθία βημάτων, αλλά ένας δυναμικός κύκλος ανάπτυξης, δοκιμών, αξιολόγησης και βελτίωσης. Κάθε κύκλος ξεκινά με την αναγνώριση ενός προβλήματος, συνεχίζει με τη δημιουργία ενός πρωτοτύπου (είτε πρόχειρο σκίτσο είτε ψηφιακό μοντέλο) και ολοκληρώνεται με τη δοκιμή του πρωτοτύπου από πραγματικούς χρήστες. Η ανατροφοδότηση από αυτές τις δοκιμές τροφοδοτεί τον επόμενο κύκλο, επιτρέποντας στην ομάδα να διορθώσει λάθη, να βελτιώσει λειτουργίες και να επικυρώσει τις σχεδιαστικές αποφάσεις, πριν καν αρχίσει η τελική υλοποίηση.
- Η ενεργός εμπλοκή των χρηστών είναι απαραίτητη για την επιτυχία. Οι τελικοί χρήστες δεν είναι παθητικοί παρατηρητές της διαδικασίας. Αντίθετα, συμμετέχουν σε κάθε στάδιο της ανάπτυξης. Μέσω τεχνικών όπως οι ενδεδειγμένες συνεντεύξεις, τα workshops, και κυρίως, οι δοκιμές χρηστικότητας (usability testing), οι σχεδιαστές αποκτούν μια βαθιά κατανόηση των αναγκών, των στόχων και των περιορισμών τους. Αυτή η συνεχής επαφή

διασφαλίζει ότι οι σχεδιαστικές επιλογές βασίζονται σε πραγματικά δεδομένα και όχι σε υποθέσεις, καθιστώντας την εφαρμογή πιο προσωπική και πιο αποτελεσματική για το κοινό στο οποίο απευθύνεται.

- Τέλος, ο HCD βασίζεται στη χρήση ποιοτικής και ποσοτικής ανάλυσης για τη συλλογή δεδομένων. Η ποιοτική ανάλυση περιλαμβάνει τεχνικές όπως η εθνογραφική παρατήρηση (παρακολούθηση των χρηστών στο φυσικό τους περιβάλλον), οι συνεντεύξεις και η χαρτογράφηση της πορείας τους (user journey mapping), που προσφέρουν μια πλούσια, λεπτομερή εικόνα του «γιατί» πίσω από τη συμπεριφορά τους. Παράλληλα, η ποσοτική ανάλυση, μέσω ερωτηματολογίων και στατιστικής επεξεργασίας δεδομένων, παρέχει μετρήσιμα αποτελέσματα που επικυρώνουν τα ποιοτικά ευρήματα. Ο συνδυασμός αυτών των δύο προσεγγίσεων επιτρέπει μια ολοκληρωμένη κατανόηση του χρήστη, καλύπτοντας τόσο την «εικόνα» όσο και τους «αριθμούς» της εμπειρίας του.

Ανθρωποκεντρικός Σχεδιασμός: Από τη Θεωρία στην Πράξη

Κατά την ανάπτυξη της εφαρμογής σίτισης, ακολουθήθηκε πιστά η μεθοδολογία του ανθρωποκεντρικού σχεδιασμού, όπως περιγράφεται παρακάτω:

1. Κατανόηση και καταγραφή απαιτήσεων

Η διαδικασία ξεκίνησε με τη διερεύνηση του πλαισίου χρήσης της εφαρμογής: μέσω συζητήσεων και αναλύσεων με φοιτητές και διαχειριστές, καταγράφηκαν οι βασικές ανάγκες και τα σημεία δυσκολίας που προέκυπταν από την προηγούμενη εμπειρία σίτισης.

2. Σχεδιασμός πολλαπλών εκδοχών (wireframes)

Στη συνέχεια δημιουργήθηκαν εναλλακτικά wireframes, τα οποία απεικόνιζαν διαφορετικές προσεγγίσεις στη διάταξη και λειτουργικότητα της εφαρμογής. Ο στόχος ήταν να καλυφθεί ένα ευρύ φάσμα ιδεών και πιθανών λύσεων.

3. Αξιολόγηση με σενάρια χρήσης και ερωτήσεις

Τα wireframes δοκιμάστηκαν από πραγματικούς χρήστες μέσω ρεαλιστικών σεναρίων, με σκοπό την καταγραφή της εμπειρίας τους, των σημείων δυσκολίας και των προσωπικών τους προτιμήσεων.

4. Αναθεώρηση βάσει ανατροφοδότησης

Η ανατροφοδότηση αξιοποιήθηκε για τη βελτίωση του σχεδιασμού. Οι επισημάνσεις των χρηστών οδήγησαν σε στοχευμένες τροποποιήσεις, με στόχο τη βελτίωση της ευχρηστίας και την επίλυση λειτουργικών αδυναμιών.

5. Μεταφορά σε πρωτότυπο υψηλής πιστότητας (Figma) [4]

Μετά την οριστικοποίηση του σχεδιασμού, δημιουργήθηκε ένα πλήρες και ρεαλιστικό πρωτότυπο στο Figma. Το πρωτότυπο αυτό απέδιδε με ακρίβεια τόσο την αισθητική όσο

και τη λειτουργικότητα της εφαρμογής.

6. Τελική αποδοχή από τους χρήστες

Το πρωτότυπο παρουσιάστηκε ξανά στους χρήστες για τελική επιβεβαίωση. Μέσα από επιπλέον δοκιμές διασφαλίστηκε ότι οι χρήστες ήταν απόλυτα ικανοποιημένοι από το αποτέλεσμα.

7. Μεταφορά σε κώδικα και υλοποίηση

Τέλος, η εφαρμογή υλοποιήθηκε προγραμματιστικά, ακολουθώντας πιστά το εγκεκριμένο σχέδιο, ώστε η τελική λύση να ανταποκρίνεται στις ανάγκες των χρηστών όπως είχαν διαμορφωθεί σε όλα τα προηγούμενα στάδια.

Η συνεχής συνεργασία με τους τελικούς χρήστες, σε κάθε στάδιο της διαδικασίας, διασφάλισε ότι η τελική εφαρμογή ήταν πλήρως προσαρμοσμένη στις ανάγκες τους, λειτουργική και εύχρηστη. Έτσι, ο ανθρωποκεντρικός σχεδιασμός λειτούργησε όχι μόνο ως μεθοδολογία σχεδίασης, αλλά ως θεμέλιο επιτυχούς ανάπτυξης της εφαρμογής σίτισης.

3.5 Αξιολόγηση αρχικών σχεδίων με χρήστες

Για την αξιολόγηση των wireframes πραγματοποιήθηκε μια σειρά δοκιμών χρηστών με στόχο την κατανόηση της χρησιμότητας, της πλοήγησης και της γενικής εμπειρίας του χρήστη κατά την αλληλεπίδρασή του με την εφαρμογή. Η διαδικασία ακολούθησε τα βασικά στάδια του **ανθρωποκεντρικού σχεδιασμού** και της **iterative αξιολόγησης**.

3.5.1 Συμμετέχοντες και Σενάριο Χρήσης

Η αξιολόγηση πραγματοποιήθηκε αποκλειστικά σε τελικούς χρήστες (φοιτητές) με ρόλο *consumer* (όχι *admin*). Συνολικά συμμετείχαν **25 άτομα**, εκ των οποίων **20 φοιτητές από το Πολυτεχνείο Κρήτης** (τόσο από τη Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών όσο και από άλλες σχολές του ιδρύματος) και **5 φοιτητές από άλλα πανεπιστήμια εκτός Χανίων**. Οι συμμετέχοντες επιλέχθηκαν ώστε να αντιπροσωπεύουν και τις δύο βασικές κατηγορίες χρηστών:

- φοιτητές με δικαίωμα δωρεάν σίτισης (13 άτομα) και
- φοιτητές χωρίς δωρεάν σίτιση (12 άτομα).

Το σενάριο που τους παρουσιάστηκε βασίστηκε στην καθημερινή χρήση της εφαρμογής για παραγγελία γευμάτων από τη φοιτητική λέσχη.

3.5.2 Μέθοδος Αξιολόγησης

Η αξιολόγηση του πρωτοτύπου πραγματοποιήθηκε με μια συνδυαστική προσέγγιση που περιλάμβανε σενάρια χρήσης (task-based testing), ερωτηματολόγια που δόθηκαν πριν, κατά τη διάρκεια και μετά τη χρήση των wireframes, καθώς και άμεση παρατήρηση της αλληλεπίδρασης των χρηστών με την εφαρμογή. Μέσω αυτής της μεθόδου καταγράφηκαν οι αντιδράσεις τους κατά την πλοήγηση, οι ενέργειες (όπως clicks), αλλά και τα λεκτικά σχόλια που προέκυψαν αυθόρμητα.

Η παρουσίαση του πρωτοτύπου πραγματοποιήθηκε με χρήση του εργαλείου Balsamiq, το οποίο χρησιμοποιήθηκε για τη δημιουργία wireframes χαμηλής πιστότητας. Ο σκοπός ήταν να επικεντρωθεί η αξιολόγηση στις βασικές λειτουργικότητες και τη ροή της εφαρμογής, χωρίς να επηρεάζεται από το τελικό οπτικό ύφος ή τη γραφιστική επιμέλεια.

3.5.3 Περιγραφή Σεναρίων και Tasks

Κατά τη διάρκεια των δοκιμών, οι συμμετέχοντες κλήθηκαν να εκτελέσουν μια σειρά από καθορισμένα **tasks**, τα οποία αποτυπώνουν τις βασικές λειτουργίες της εφαρμογής σίτισης. Ο σκοπός ήταν η αξιολόγηση της ευχρηστίας των wireframes και η καταγραφή τυχόν δυσκολιών κατά την αλληλεπίδραση με το σύστημα.

Task 1: Επιλογή ημερομηνίας και τύπου γεύματος και προσθήκη προϊόντων στο καλάθι

Στο πρώτο στάδιο, οι χρήστες καλούνταν να επιλέξουν ημερομηνία και τύπο γεύματος (meal type) και να προσθέσουν τα επιθυμητά φαγητά στο καλάθι τους, αξιοποιώντας τις δυνατότητες φιλτραρίσματος ανά κατηγορία.

- Αξιολογήθηκε η ικανότητα των χρηστών να εντοπίζουν γρήγορα τα προϊόντα.
- Εξετάστηκε η αποτελεσματικότητα του φιλτραρίσματος στην υποστήριξη της διαδικασίας επιλογής.
- Καταγράφηκαν τυχόν σημεία δυσκολίας που επηρέασαν την εμπειρία χρήστη.

Task 2: Επεξεργασία παραγγελίας στη σελίδα σύνοψης

Στη συνέχεια, οι συμμετέχοντες επισκέπτονταν στη σελίδα σύνοψης της παραγγελίας, με σκοπό να τροποποιήσουν στοιχεία όπως η αλλαγή ποσότητας ή η διαγραφή προϊόντων.

- Αξιολογήθηκε η σαφήνεια των διαθέσιμων επιλογών.
- Καταγράφηκαν τυχόν δυσκολίες κατά την εκτέλεση της επεξεργασίας.

Task 3: Επιλογή τρόπου παραγγελίας, timeslot και πληρωμής

Οι χρήστες καλούνταν να επιλέξουν τον τρόπο παραγγελίας (eat-in ή take-away), το κατάλληλο χρονικό διάστημα (timeslot) και, όπου απαιτείτο, τη μέθοδο πληρωμής.

- Εξετάστηκε η σαφήνεια των επιλογών και η ευκολία ολοκλήρωσης της διαδικασίας.
- Καταγράφηκε η ικανότητα των συμμετεχόντων να ολοκληρώσουν την επιλογή χωρίς λάθη ή καθυστερήσεις.

Task 4: Αποστολή παραγγελίας και επιβεβαίωση

Στο τελικό στάδιο, οι συμμετέχοντες υπέβαλαν την παραγγελία και ερμήνευσαν το μήνυμα επιβεβαίωσης που εμφανιζόταν στην οθόνη.

- Αξιολογήθηκε η κατανόηση του μηνύματος από τους χρήστες.
- Καταγράφηκε η ικανότητά τους να επιβεβαιώσουν την επιτυχή ολοκλήρωση της παραγγελίας.

Task 5: Εξέταση προφίλ και ιστορικού παραγγελιών

Οι χρήστες μεταβαίνανε στο προφίλ τους για να εξετάσουν το ιστορικό παραγγελιών και τις διαθέσιμες στατιστικές πληροφορίες.

- Εξετάστηκε η προσβασιμότητα και η σαφήνεια της παρουσίασης των δεδομένων.
- Καταγράφηκε η κατανόηση της πληροφορίας από τους συμμετέχοντες.

Για κάθε ένα από τα παραπάνω tasks, οι συμμετέχοντες κλήθηκαν να απαντήσουν σε στοχευμένες ερωτήσεις σχετικά με:

- Την επιτυχή ολοκλήρωση του task,
- Τυχόν δυσκολίες στην εύρεση των αντίστοιχων λειτουργιών,
- Το κατά πόσο το layout του wireframe διευκόλυνε ή δυσχέρανε την εκτέλεση των δραστηριοτήτων.

3.5.4 Ερωτηματολόγιο Προ και Μετά τη Χρήση

Το ερωτηματολόγιο που χρησιμοποιήθηκε για την αξιολόγηση χωρίστηκε σε τρία σκέλη: πριν την αλληλεπίδραση, κατά τη διάρκεια της παρουσίασης του wireframe και μετά την ολοκλήρωση της δοκιμής.

Πριν την παρουσίαση, οι συμμετέχοντες ρωτήθηκαν τι περιμένουν από μια τέτοια εφαρμογή, ποιες λειτουργίες θεωρούν απαραίτητες, ποιες πληροφορίες επιθυμούν να είναι άμεσα προσβάσιμες και

ποιες είναι οι ανάγκες που θα ήθελαν να καλύψει η εφαρμογή στην καθημερινότητά τους (βλ. Παράρτημα Α.2 για το πλήρες ερωτηματολόγιο)..

Κατά τη διάρκεια της παρουσίασης του wireframe, ζητήθηκε να αξιολογήσουν αν η διάταξη της εφαρμογής είναι κατανοητή, αν αντιλαμβάνονται τη λειτουργία κάθε στοιχείου, αν εντοπίζουν τα βασικά features, και αν υπάρχουν σημεία που θεωρούν περιττά ή που θα βελτιώνονταν με καλύτερο σχεδιασμό. Οι ερωτήσεις επικεντρώθηκαν επίσης στο αν η εφαρμογή ανταποκρίνεται στις προσδοκίες τους.

Μετά την ολοκλήρωση της χρήσης του wireframe, τους ζητήθηκε να αξιολογήσουν πόσο εύκολη ήταν η εμπειρία τους, πόσο ικανοποιημένοι ένιωσαν, αν θα χρησιμοποιούσαν την εφαρμογή όταν αυτή ολοκληρωθεί και τι θα άλλαζαν εφόσον είχαν την ευκαιρία. Επιπλέον, ενθαρρύνθηκαν να προτείνουν νέες λειτουργίες που θεωρούν χρήσιμες, καθώς και να εκφράσουν τυχόν γενικά σχόλια.

3.6 Αποτελέσματα Αξιολόγησης Wireframes

Η αξιολόγηση των wireframes αποκάλυψε πληθώρα χρήσιμων σχολίων από τους συμμετέχοντες, τα οποία οδήγησαν σε ουσιαστικές βελτιώσεις του τελικού σχεδιασμού. Οι χρήστες γενικά ενθουσιάστηκαν με την **αρχική οθόνη** του Wireframe Version 1 και 3, χαρακτηρίζοντας τη διάκριση των γευμάτων μέσω τριών ευδιάκριτων κουμπιών (πρωινό, μεσημεριανό, βραδινό) ως ξεκάθαρη και πρακτική.

Κατά την πλοήγησή τους μέσα σε μία επιμέρους κατηγορία φαγητών, προτίμησαν το πιο **λιτό και λειτουργικό layout** του *Wireframe Version 2*, όπου κάθε φαγητό παρουσιαζόταν σε ξεχωριστή γραμμή με εικόνα στα αριστερά, περιγραφή και επιλογή ποσότητας. Ιδιαίτερα θετικά σχολιάστηκε η δυνατότητα **προσθήκης στο καλάθι με ένα μόνο κλικ**, μετά το οποίο εμφανιζόταν άμεσα το κουμπί "View Order" μαζί με το συνολικό κόστος της παραγγελίας (σε περίπτωση που ο χρήστης δεν είχε δωρεάν σίτιση). Επιπλέον, το γεγονός ότι **εξαφανιζόταν το κουμπί "προσθήκη" από τα υπόλοιπα προϊόντα της ίδιας κατηγορίας** με την επιλογή ενός φαγητού κρίθηκε ιδιαίτερα αποτελεσματικό, καθώς μείωνε αισθητά τα περιττά clicks και ενίσχυε την απλότητα της διάδρασης. Χάρη σε αυτή τη συμπεριφορά, οι χρήστες **απέρριψαν την εκδοχή με overlay**, καθώς η inline διάδραση τους φάνηκε πιο άμεση και κατανοητή.

Η **σελίδα επισκόπησης (overview)** κρίθηκε ως ένα από τα πιο ισχυρά σημεία του συστήματος. Παρουσίαζε συγκεντρωμένα τις βασικές πληροφορίες (επιλεγμένο φαγητό, timeslot, τρόπο πληρωμής), προσφέροντας στους χρήστες ξεκάθαρη εικόνα για το τι πρόκειται να παραγγείλουν και με ποιον τρόπο (Wireframe Version 1 and 3). Οι συμμετέχοντες ανέφεραν ότι **προτιμούν αυτές οι πληροφορίες να βρίσκονται συγκεντρωμένες σε μία σελίδα**, παρά να διασπώνται σε διαδοχικά βήματα.

Η επιλογή timeslot μέσω **dropdown** μενού δεν θεωρήθηκε ιδιαίτερα βοηθητική. Αντιθέτως, προτίμησαν τη διάδραση με **grid** διάταξη timeslots, ενισχυμένη από μία **χρωματική παλέτα**, η οποία παρέχει πληροφορίες σχετικά με την πληρότητα κάθε χρονικής ζώνης (π.χ. έντονο χρώμα για πλήρη slots, πιο ήπιο για αραιά). Η οπτική αυτή διαφοροποίηση βοήθησε σημαντικά στην επιλογή του κατάλληλου χρονικού διαστήματος για την παραγγελία (Wireframe Version 2).

Σε ό,τι αφορά την **αλλαγή ημερομηνίας**, χρήστες μεγαλύτερης ηλικίας ανέφεραν ότι το **carousel** **πλοήγησης** δεν ήταν εύχρηστο και προτιμούσαν ένα **πιο ξεκάθαρο κουμπί**, ευδιάκριτο και σταθερά τοποθετημένο, που να εμφανίζει απευθείας ημερολογιακή επιλογή. Αυτό ήταν κάτι το οποίο δεν είχαμε προβλέψει στα wireframes.

Η **σελίδα προφίλ** θεωρήθηκε χρήσιμη κυρίως για την ενημέρωση σχετικά με το δικαίωμα σίτισης (αν δηλαδή δικαιούνται δωρεάν ή όχι), καθώς και για την προβολή **λεπτομερούς ιστορικού παραγγελιών**. Ιδιαίτερο ενδιαφέρον εκφράστηκε για την **προσθήκη voucher**, με τους χρήστες να ζητούν δυνατότητα **προβολής του διαθέσιμου υπολοίπου και ιστορικού συναλλαγών**, ώστε να γνωρίζουν ανά πάσα στιγμή τις κινήσεις και το διαθέσιμο ποσό για φαγητό..

Τέλος, θετική εντύπωση προκάλεσε η **σελίδα επιβεβαίωσης** μετά την αποστολή της παραγγελίας (βρίσκεται σε όλα τα wireframes). Ήταν διαδραστική και διατηρούσε την προσοχή του χρήστη, προσφέροντάς του μια **θετική εμπειρία ολοκλήρωσης**, η οποία ενίσχυε τη συνολική εικόνα αξιοπιστίας και φιλικότητας της εφαρμογής.

Όλα τα παραπάνω ευρήματα αξιοποιήθηκαν ως οδηγός για τη διαμόρφωση της τελικής διεπαφής χρήστη, προσαρμοσμένης στις ανάγκες και τις προτιμήσεις των πραγματικών χρηστών, όπως θα παρουσιαστεί στο επόμενο κεφάλαιο.

4. Τεχνολογίες και Εργαλεία Ανάπτυξης

Στο παρόν κεφάλαιο, αναλύονται οι τεχνολογίες και τα εργαλεία που επιλέχθηκαν για την ανάπτυξη της διαδικτυακής εφαρμογής παραγγελιοληψίας του πανεπιστημιακού εστιατορίου. Η επιλογή τους βασίστηκε στην αποτελεσματικότητα, την ευελιξία και την ευρεία υποστήριξη που προσφέρουν για την υλοποίηση μιας full-stack εφαρμογής, καθώς και στην ικανότητά τους να καλύψουν τις ειδικές απαιτήσεις του έργου, συμπεριλαμβανομένης της πρόβλεψης ποσοτήτων φαγητού.

4.1 Frontend τεχνολογίες

Το frontend της εφαρμογής, δηλαδή η διεπαφή με την οποία αλληλεπιδρούν οι χρήστες, αναπτύχθηκε χρησιμοποιώντας τις βασικές τεχνολογίες του διαδικτύου: HTML για τη δομή, CSS για την εμφάνιση και JavaScript για τη διαδραστικότητα.

4.1.1 HTML/CSS και Tailwind CSS

Το **HTML (HyperText Markup Language)** αποτελεί τον θεμέλιο λίθο κάθε ιστοσελίδας, παρέχοντας τη δομή και το νόημα στο περιεχόμενο μέσω ετικετών (tags) για στοιχεία όπως επικεφαλίδες, παραγράφους, λίστες, εικόνες, συνδέσμους και φόρμες. Στην παρούσα εργασία, το HTML χρησιμοποιήθηκε για τη δημιουργία όλων των δομικών στοιχείων της εφαρμογής, από τις σελίδες προβολής του μενού και του καλαθιού παραγγελιών, μέχρι τις φόρμες σύνδεσης/εγγραφής και τις διαχειριστικές διεπαφές. Η σωστή δομή του HTML είναι κρίσιμη για την προσβασιμότητα και τη βελτιστοποίηση για μηχανές αναζήτησης (SEO).

Το **CSS (Cascading Style Sheets)** είναι υπεύθυνο για την οπτική παρουσίαση και το στυλ της εφαρμογής, μετατρέποντας μια απλή HTML σελίδα σε μια ελκυστική και λειτουργική διεπαφή χρήστη. Μας επέτρεψε να ελέγξουμε χρώματα, γραμματοσειρές, διατάξεις και κινούμενα σχέδια. Για την εφαρμογή του πανεπιστημιακού εστιατορίου, το CSS χρησιμοποιήθηκε για τον καθορισμό της συνολικής αισθητικής, διασφαλίζοντας μια συνεπή εμφάνιση σε όλες τις σελίδες.

Για την επιτάχυνση της ανάπτυξης και την επίτευξη ενός σύγχρονου και προσαρμόσιμου σχεδιασμού, επιλέχθηκε το **Tailwind CSS** [5]. Σε αντίθεση με τα παραδοσιακά CSS frameworks που παρέχουν προκαθορισμένα components (π.χ. κουμπιά, navbars), το Tailwind CSS ακολουθεί μια "utility-first" προσέγγιση. Αυτό σημαίνει ότι παρέχει μια πληθώρα μικρών, επαναχρησιμοποιήσιμων κλάσεων (utility classes) που εφαρμόζονται απευθείας στο HTML για τον έλεγχο ιδιοτήτων όπως το spacing, τα χρώματα, η τυπογραφία και οι σκιές.

Η επιλογή του Tailwind CSS αιτιολογήθηκε από τα ακόλουθα πλεονεκτήματα που προσέφερε

στην εργασία:

- **Ταχεία ανάπτυξη και πρωτοτυποποίηση:** Μας επέτρεψε να εφαρμόζουμε στυλ απευθείας στο HTML, μειώνοντας την ανάγκη συνεχούς εναλλαγής μεταξύ HTML και CSS αρχείων. Αυτό επιτάχυνε σημαντικά τη διαδικασία σχεδιασμού και δοκιμής διαφορετικών παραλλαγών της διεπαφής χρήστη για το μενού, το καλάθι και τις διαχειριστικές σελίδες.
- **Ευελιξία και προσαρμογή:** Παρέχει πλήρη έλεγχο στο στυλ, επιτρέποντας τη δημιουργία μοναδικών σχεδίων χωρίς να βασιζόμαστε σε προκαθορισμένα components. Αυτό ήταν ιδιαίτερα χρήσιμο για την προσαρμογή της εμφάνισης στις ειδικές ανάγκες του πανεπιστημιακού εστιατορίου.
- **Responsive Design:** Το Tailwind CSS υποστηρίζει μια προσέγγιση "mobile-first", επιτρέποντας την εύκολη προσαρμογή της διάταξης και του στυλ της εφαρμογής σε διαφορετικά μεγέθη οθόνης (π.χ., κινητά, tablets, desktops) μέσω ειδικών responsive classes.⁶ Αυτό διασφάλισε ότι η εφαρμογή είναι προσβάσιμη και λειτουργική για όλους τους χρήστες, ανεξαρτήτως συσκευής.
- **Καθαρότερος κώδικας και συντηρησιμότητα:** Ελαχιστοποιεί την ανάγκη για μεγάλα, custom CSS αρχεία, οδηγώντας σε πιο καθαρό και συνοπτικό κώδικα HTML. Οι utility classes είναι επαναχρησιμοποιήσιμες, προωθώντας τη συνέπεια και μειώνοντας τις πιθανότητες συγκρούσεων στυλ. Αυτό διευκόλυνε τη συντήρηση και την επέκταση της εφαρμογής.

4.1.2 JavaScript

Η **JavaScript** είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου, η οποία μετατρέπει τις στατικές ιστοσελίδες σε δυναμικές και διαδραστικές εμπειρίες. Στην εφαρμογή του πανεπιστημιακού εστιατορίου, η JavaScript χρησιμοποιήθηκε εκτενώς για την υλοποίηση όλων των διαδραστικών λειτουργιών στο frontend.

Συγκεκριμένα, η JavaScript προσέφερε τα εξής:

- **Δυναμική τροποποίηση περιεχομένου (DOM Manipulation):** Μας επέτρεψε να ενημερώνουμε το μενού σε πραγματικό χρόνο, να προσθέτουμε και να αφαιρούμε στοιχεία από το καλάθι αγορών, και να εμφανίζουμε μηνύματα επιβεβαίωσης ή σφάλματος χωρίς την ανάγκη ανανέωσης ολόκληρης της σελίδας. Για παράδειγμα, όταν ένας χρήστης προσθέτει ένα φαγητό στο καλάθι, η JavaScript ενημερώνει άμεσα την οπτική αναπαράσταση του καλαθιού.
- **Αλληλεπίδραση με τον χρήστη:** Χειρίστηκε γεγονότα (events) όπως κλικ σε κουμπιά (π.χ., "Προσθήκη στο Καλάθι", "Ολοκλήρωση Παραγγελίας"), εισαγωγή δεδομένων σε φόρμες (π.χ., στοιχεία χρήστη, επιλογή ώρας παραλαβής) και δυναμική εμφάνιση/απόκρυψη στοιχείων διεπαφής.
- **Ασύγχρονη επικοινωνία (AJAX):** Η JavaScript, μέσω τεχνικών AJAX (Asynchronous

JavaScript and XML), επέτρεψε την ασύγχρονη επικοινωνία με το backend (PHP REST API). Αυτό σημαίνει ότι η εφαρμογή μπορεί να στέλνει αιτήματα στον server (π.χ., για ανάκτηση του ημερήσιου μενού, υποβολή παραγγελίας) και να λαμβάνει απαντήσεις (συνήθως σε μορφή JSON) χωρίς να χρειάζεται να φορτώσει ξανά ολόκληρη τη σελίδα. Αυτό βελτίωσε σημαντικά την εμπειρία χρήστη, καθιστώντας την εφαρμογή πιο γρήγορη και ανταποκρινόμενη. Για παράδειγμα, η φόρτωση του μενού γίνεται δυναμικά μέσω AJAX, επιτρέποντας την άμεση ενημέρωση χωρίς διακοπή της περιήγησης.

- **Επικύρωση δεδομένων (Form Validation):** Χρησιμοποιήθηκε για την αρχική επικύρωση των δεδομένων που εισάγουν οι χρήστες στις φόρμες (π.χ., έλεγχος για κενά πεδία, σωστή μορφή email) πριν αυτά σταλούν στον server, παρέχοντας άμεσο feedback στον χρήστη και μειώνοντας το φορτίο του backend.¹⁰

4.1.3 Επιπλέον JavaScript βιβλιοθήκες

Για την ενίσχυση της διαδραστικότητας και της αισθητικής της εφαρμογής, καθώς και για την καλύτερη οργάνωση και παρουσίαση δεδομένων, χρησιμοποιήθηκαν επιπλέον έτοιμες βιβλιοθήκες JavaScript με ευρεία αποδοχή και υποστήριξη:

- **SweetAlert2** [6]: Πρόκειται για μια βιβλιοθήκη που προσφέρει όμορφα και εύχρηστα αναδυόμενα παράθυρα (pop-ups) για την εμφάνιση ειδοποιήσεων, επιβεβαιώσεων, προειδοποιήσεων και μηνυμάτων σφάλματος. Με το SweetAlert2 η εφαρμογή απέκτησε μοντέρνα και φιλική προς το χρήστη μορφή για όλες τις αλληλεπιδράσεις που απαιτούν επιβεβαίωση ή ενημέρωση, βελτιώνοντας σημαντικά την εμπειρία χρήστη.
- **Swiper** [7]: Αυτή η βιβλιοθήκη χρησιμοποιήθηκε για τη δημιουργία διαδραστικών και ομαλών slideshow/carousel, όπως για παράδειγμα την παρουσίαση του ιστορικού των vouchers στο προφίλ του χρήστη. Το Swiper είναι αποδοτικό, responsive και προσφέρει πολλές επιλογές παραμετροποίησης για την απόδοση υψηλής ποιότητας κινούμενων εικόνων και περιεχομένου.
- **Tabulator** [8]: Για την οργάνωση και την εμφάνιση σύνθετων δεδομένων σε μορφή πινάκων, χρησιμοποιήθηκε η βιβλιοθήκη Tabulator. Αυτή παρέχει εύχρηστες λειτουργίες όπως φιλτράρισμα, ταξινόμηση, σελιδοποίηση και επεξεργασία κελιών, καθιστώντας τα διαχειριστικά τμήματα της εφαρμογής πιο λειτουργικά και φιλικά προς τον διαχειριστή.

Η ενσωμάτωση αυτών των βιβλιοθηκών βελτίωσε την απόδοση της εφαρμογής και την ποιότητα της εμπειρίας χρήστη, χωρίς να επιβαρύνει σημαντικά τον κώδικα ή τους χρόνους φόρτωσης.

4.2 Backend τεχνολογίες

Το backend της εφαρμογής είναι υπεύθυνο για τη λογική του διακομιστή, τη διαχείριση της βάσης δεδομένων και την επικοινωνία με το frontend. Για την υλοποίηση αυτού του μέρους, επιλέχθηκε η PHP σε συνδυασμό με την αρχιτεκτονική REST API.

4.2.1 PHP, REST API & JSON Web Token (JWT)

Η **PHP (Hypertext Preprocessor)** [9] είναι μια ευρέως χρησιμοποιούμενη γλώσσα προγραμματισμού server-side, ιδανική για τη δημιουργία δυναμικών ιστοσελίδων και εφαρμογών.¹² Στην παρούσα εργασία, η PHP αποτελεί τον πυρήνα του backend, χειριζόμενη όλες τις λειτουργίες που απαιτούν επεξεργασία στον διακομιστή.

Η επιλογή της PHP αιτιολογήθηκε από τα ακόλουθα:

- **Διαχείριση βάσης δεδομένων:** Η PHP έχει εξαιρετική συμβατότητα με τη MySQL, επιτρέποντας την αποτελεσματική εκτέλεση ερωτημάτων για την ανάκτηση, αποθήκευση, ενημέρωση και διαγραφή δεδομένων σχετικών με το μενού, τις παραγγελίες, τους χρήστες και τα κουπόνια.
- **Λογική εφαρμογής:** Υλοποιήθηκε η επιχειρηματική λογική της εφαρμογής, όπως η επεξεργασία παραγγελιών, η διαχείριση αποθεμάτων, η εφαρμογή εκπτώσεων μέσω κουπονιών και η διαχείριση των ρόλων χρηστών (απλός χρήστης, διαχειριστής).
- **Αυθεντικοποίηση και εξουσιοδότηση:** Η PHP χειρίστηκε τις διαδικασίες σύνδεσης και εγγραφής χρηστών, καθώς και τον έλεγχο πρόσβασης σε λειτουργίες και δεδομένα ανάλογα με τον ρόλο του χρήστη (π.χ., μόνο ο διαχειριστής μπορεί να προσθέσει ή να τροποποιήσει φαγητά). Επιπλέον, για την ασφαλή διαχείριση της αυθεντικοποίησης και τη διατήρηση της ταυτότητας των χρηστών χρησιμοποιήθηκε το **JSON Web Token (JWT)** [10]. Το JWT παρέχει έναν ασφαλή και αποδοτικό τρόπο μεταφοράς πληροφοριών ταυτότητας μεταξύ του frontend και του backend, χωρίς την ανάγκη διατήρησης κατάστασης (stateless authentication).

Τα πλεονεκτήματα της χρήσης JWT στην εφαρμογή είναι:

- **Ασφάλεια:** Τα tokens υπογράφονται ψηφιακά, διασφαλίζοντας ότι τα δεδομένα δεν έχουν αλλοιωθεί και επιτρέποντας την επαλήθευση της ταυτότητας του χρήστη.
- **Ανεξαρτησία από κατάσταση:** Αφαιρεί την ανάγκη αποθήκευσης συνεδριών στον server, βελτιώνοντας την απόδοση και την κλιμάκωση του συστήματος.
- **Ευελιξία:** Τα JWT μπορούν να περιλαμβάνουν επιπλέον claims (π.χ., ρόλο χρήστη, δικαιώματα), επιτρέποντας εξειδικευμένο έλεγχο πρόσβασης σε συγκεκριμένες λειτουργίες.
- **Εύκολη ενσωμάτωση με REST API:** Λόγω της ασύγχρονης και stateless φύσης των API,

το JWT αποτελεί ιδανική λύση για την αυθεντικοποίηση και εξουσιοδότηση αιτημάτων.

Η επικοινωνία μεταξύ του frontend (JavaScript) και του backend (PHP) πραγματοποιείται μέσω ενός **REST API (Representational State Transfer Application Programming Interface)** [11]. Ένα REST API επιτρέπει την απρόσκοπτη αλληλεπίδραση μεταξύ client και server, χρησιμοποιώντας τυποποιημένες μεθόδους HTTP (GET, POST, PUT, DELETE) για την πρόσβαση και τη χειραγώγηση πόρων.

Τα πλεονεκτήματα της χρήσης REST API στην εργασία είναι:

- **Διαχωρισμός ανησυχιών (Separation of Concerns):** Επέτρεψε τον σαφή διαχωρισμό του frontend από το backend, καθιστώντας κάθε μέρος ανεξάρτητο και ευκολότερο στην ανάπτυξη, συντήρηση και κλιμάκωση. Για παράδειγμα, οι αλλαγές στην εμφάνιση του μενού στο frontend δεν επηρεάζουν τη λογική διαχείρισης των φαγητών στο backend.
- **Επεκτασιμότητα και επαναχρησιμοποίηση:** Το API μπορεί να χρησιμοποιηθεί από πολλαπλές εφαρμογές (π.χ., web, mobile) ή ακόμα και από τρίτους προγραμματιστές στο μέλλον.
- **Τυποποίηση:** Η χρήση των μεθόδων HTTP (GET για ανάκτηση δεδομένων, POST για υποβολή νέων, PUT για ενημέρωση, DELETE για διαγραφή) και η ανταλλαγή δεδομένων σε μορφή JSON διασφαλίζει τη συνέπεια και την ευκολία κατανόησης των endpoints. Για παράδειγμα, ένα αίτημα GET /api/menu ανακτά το μενού, ενώ ένα POST /api/orders υποβάλλει μια νέα παραγγελία.

4.2.2 Διαχείριση endpoints (Postman)

Το **Postman** [12] είναι ένα ισχυρό εργαλείο που χρησιμοποιήθηκε για την ανάπτυξη, δοκιμή και τεκμηρίωση του REST API της εφαρμογής. Παρέχει ένα φιλικό προς τον χρήστη περιβάλλον για την αποστολή αιτημάτων HTTP σε API endpoints και την ανάλυση των απαντήσεων.

Στην πράξη, το Postman χρησιμοποιήθηκε για:

- **Δοκιμή API endpoints:** Πριν την ενσωμάτωση του frontend, κάθε endpoint του PHP API (π.χ., για την προσθήκη φαγητού, την ανάκτηση παραγγελιών) δοκιμάστηκε μεμονωμένα στο Postman. Αυτό επέτρεψε την άμεση επαλήθευση της ορθής λειτουργίας του backend και τον εντοπισμό σφαλμάτων σε πρώιμο στάδιο.
- **Οργάνωση αιτημάτων σε Collections:** Όλα τα αιτήματα API οργάνωθηκαν σε collections εντός του Postman, κατηγοριοποιώντας τα ανά λειτουργικότητα (π.χ., "Διαχείριση Μενού", "Παραγγελίες Χρήστη"). Αυτό βελτίωσε την οργάνωση και την επαναχρησιμοποίηση των δοκιμών.
- **Χρήση Environment Variables:** Για τη διαχείριση διαφορετικών περιβαλλόντων (π.χ.,

ανάπτυξη, παραγωγή), χρησιμοποιήθηκαν environment variables στο Postman, επιτρέποντας την εύκολη εναλλαγή μεταξύ URLs και άλλων ρυθμίσεων χωρίς να χρειάζεται τροποποίηση των αιτημάτων.

- **Τεκμηρίωση API:** Το Postman διευκόλυνε την τεκμηρίωση των API endpoints, καθώς κάθε αίτημα και η αναμενόμενη απάντησή του μπορούσαν να αποθηκευτούν και να περιγραφούν λεπτομερώς, δημιουργώντας μια ζωντανή τεκμηρίωση για μελλοντική χρήση και συνεργασία.

4.3 Βάση δεδομένων

Η αποτελεσματική αποθήκευση και διαχείριση των δεδομένων είναι θεμελιώδης για τη λειτουργία της εφαρμογής. Για τον σκοπό αυτό, επιλέχθηκε η σχεσιακή βάση δεδομένων MySQL.

4.3.1 MySQL και σχεδιασμός σχήματος

Η MySQL [13] είναι ένα δημοφιλές σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) ανοιχτού κώδικα, ευρέως χρησιμοποιούμενο σε διαδικτυακές εφαρμογές λόγω της αξιοπιστίας, της απόδοσης και της ευκολίας χρήσης της.

Στην εφαρμογή του πανεπιστημιακού εστιατορίου, η MySQL χρησιμοποιήθηκε για την αποθήκευση όλων των δομημένων δεδομένων, συμπεριλαμβανομένων:

- **Πληροφορίες χρηστών:** Πίνακες για την αποθήκευση στοιχείων χρηστών (π.χ., ID, όνομα χρήστη, email, κωδικός πρόσβασης) και των ρόλων τους (χρήστης, διαχειριστής).
- **Δεδομένα μενού:** Πίνακες για τα φαγητά, τις κατηγορίες τους, τις διαθέσιμες ποσότητες και τις τιμές.
- **Δεδομένα παραγγελιών:** Πίνακες για τις παραγγελίες, τα στοιχεία των παραγγελθέντων φαγητών, την κατάσταση της παραγγελίας, την ώρα παραλαβής και τον τρόπο παραλαβής.
- **Δεδομένα κουπονιών:** Πίνακες για τη διαχείριση των εκπτωτικών κουπονιών.
- **Ιστορικό πωλήσεων/ζήτησης:** Δεδομένα που τροφοδοτούν το μοντέλο πρόβλεψης ποσοτήτων φαγητού.

Ο σχεδιασμός του σχήματος της βάσης δεδομένων (database schema) έγινε με προσοχή, ακολουθώντας αρχές κανονικοποίησης (normalization) για τη μείωση της πλεονασμού δεδομένων και τη βελτίωση της ακεραιότητας των δεδομένων. Για παράδειγμα, αντί να αποθηκεύονται όλες οι πληροφορίες ενός φαγητού απευθείας στον πίνακα παραγγελιών, δημιουργήθηκαν ξεχωριστοί πίνακες για τα φαγητά και τις παραγγελίες, με σχέσεις μεταξύ τους. Αυτό εξασφαλίζει την ευελιξία, καθώς οι τύποι των φαγητών και τα χαρακτηριστικά τους μπορούν να αλλάξουν χωρίς να απαιτείται αλλαγή στο σχήμα του πίνακα προϊόντων.

4.3.2 ER διαγράμματα

Τα **ER (Entity-Relationship) διαγράμματα** [14] αποτελούν ένα θεμελιώδες εργαλείο στον σχεδιασμό βάσεων δεδομένων. Χρησιμοποιούν ένα καθορισμένο σύνολο συμβόλων (ορθογώνια για οντότητες, διαμάντια για σχέσεις, οβάλ για χαρακτηριστικά) και συνδετικών γραμμών για να απεικονίσουν τη διασύνδεση μεταξύ οντοτήτων, σχέσεων και των χαρακτηριστικών τους.

Στην παρούσα εργασία, τα ER διαγράμματα χρησιμοποιήθηκαν για:

- **Οπτικοποίηση του σχεδιασμού:** Μας επέτρεψαν να οπτικοποιήσουμε τις κύριες οντότητες του συστήματος (π.χ., Χρήστες, Φαγητά, Παραγγελίες, Κουπόνια) και τις σχέσεις μεταξύ τους (π.χ., ένας Χρήστης κάνει πολλές Παραγγελίες, μια Παραγγελία περιέχει πολλά Φαγητά).
- **Εντοπισμός λαθών και βελτιώσεων:** Μέσω της σχεδίασης των ER διαγραμμάτων, είχαμε την ευκαιρία να εντοπίσουμε πιθανά λάθη ή ατέλειες στον σχεδιασμό της βάσης δεδομένων σε πρώιμο στάδιο, πριν την υλοποίηση, και να προβούμε στις απαραίτητες διορθώσεις.
- **Καθορισμός καρδιναλιοτήτων:** Τα διαγράμματα απεικόνισαν τις καρδιναλότητες των σχέσεων (π.χ., ένα προς ένα, ένα προς πολλά, πολλά προς πολλά), οι οποίες είναι κρίσιμες για τη σωστή υλοποίηση των πινάκων και των ξένων κλειδιών στη MySQL.
- **Δημιουργία λογικού και φυσικού μοντέλου:** Ξεκινώντας από ένα εννοιολογικό μοντέλο που περιγράφει τα επιχειρηματικά αντικείμενα, προχωρήσαμε σε ένα λογικό μοντέλο με πιο λεπτομερείς οντότητες και χαρακτηριστικά, και τέλος σε ένα φυσικό μοντέλο που αντικατοπτρίζει την πραγματική δομή της βάσης δεδομένων στη MySQL, λαμβάνοντας υπόψη τους τύπους δεδομένων και τους περιορισμούς.

4.4. Πρόβλεψη Ποσοτήτων Φαγητού

Ένα καινοτόμο στοιχείο της εργασίας είναι η ενσωμάτωση ενός συστήματος πρόβλεψης ποσοτήτων φαγητού, με στόχο τη βελτιστοποίηση της διαχείρισης αποθεμάτων και τη μείωση της σπατάλης.

4.4.1. Python – Ανάλυση δεδομένων

Η **Python** [15] επιλέχθηκε ως η κύρια γλώσσα προγραμματισμού για την ανάλυση δεδομένων και την υλοποίηση του μοντέλου πρόβλεψης. Η Python είναι ευρέως αναγνωρισμένη για την απλότητα της σύνταξής της, το εκτενές οικοσύστημα βιβλιοθηκών της και την ισχυρή υποστήριξη για επιστημονικούς υπολογισμούς και μηχανική μάθηση.

Στο πλαίσιο της εργασίας, η Python χρησιμοποιήθηκε για:

- **Προεπεξεργασία δεδομένων:** Για την εισαγωγή, καθαρισμό και μετασχηματισμό των ιστορικών δεδομένων παραγγελιών από τη βάση δεδομένων MySQL (ή αρχεία CSV) σε μορφή κατάλληλη για το μοντέλο πρόβλεψης. Αυτό περιλάμβανε τη διαχείριση ελλειπών δεδομένων και την προσαρμογή των μορφών ημερομηνίας/ώρας.
- **Ενσωμάτωση εξωτερικών παραγόντων:** Η Python διευκόλυνε την ενσωμάτωση πρόσθετων δεδομένων, όπως πληροφορίες καιρού ή ημερομηνίες αργιών, τα οποία μπορούν να επηρεάσουν τη ζήτηση φαγητού.
- **Εκτέλεση του μοντέλου πρόβλεψης:** Χρησιμοποιήθηκε για την εκπαίδευση του μοντέλου Facebook Prophet και την παραγωγή προβλέψεων για τις μελλοντικές ποσότητες φαγητού.

4.4.2. Βιβλιοθήκη Facebook Prophet – Forecasting μερίδων

Για την πρόβλεψη της ημερήσιας κατανάλωσης μερίδων φαγητού, κρίθηκε σκόπιμη η χρήση ενός αλγορίθμου χρονοσειρών που να μπορεί να διαχειριστεί με ευελιξία δεδομένα πραγματικού ή προσομοιωμένου περιβάλλοντος σίτισης. Ανάμεσα στις διαθέσιμες επιλογές, ο **Facebook Prophet** [16] αποτέλεσε την τελική επιλογή, τόσο για τα τεχνικά του χαρακτηριστικά όσο και για τη λειτουργική του επάρκεια απέναντι στα δεδομένα του συστήματος.

Το Prophet είναι ένα εργαλείο ανοικτού κώδικα, ανεπτυγμένο από την ομάδα δεδομένων του Facebook, ειδικά σχεδιασμένο για την πρόβλεψη χρονοσειρών με έντονα εποχικά μοτίβα, μεταβολές στη ζήτηση και περιοδικές διακυμάνσεις. Η φιλοσοφία του βασίζεται στο **additive model**, όπου η συνολική ζήτηση μοντελοποιείται ως άθροισμα διαφορετικών συνιστωσών: της μακροχρόνιας τάσης, των εποχικών μοτίβων (εβδομαδιαίων, ετήσιων), και των επιπτώσεων εξωτερικών παραγόντων, όπως οι αργίες ή ειδικές περιστάσεις. Αυτό το χαρακτηριστικό τον καθιστά ιδανικό για χρήση σε περιβάλλοντα όπως τα φοιτητικά εστιατόρια ή οι λέσχες σίτισης, όπου οι κύκλοι κατανάλωσης είναι επηρεασμένοι από το ημερολόγιο, τον καιρό, τις εξεταστικές περιόδους και άλλες περιοδικές συνθήκες.

Ένα από τα σημαντικότερα πλεονεκτήματα του Prophet είναι η ευκολία στη χρήση του. Σε αντίθεση με πιο περίπλοκα μοντέλα μηχανικής μάθησης, όπως τα LSTM ή τα recurrent neural networks, το Prophet απαιτεί ελάχιστη παραμετροποίηση, καθιστώντας το προσβάσιμο ακόμα και σε χρήστες χωρίς βαθιά εξοικείωση με τη στατιστική μοντελοποίηση ή τον προγραμματισμό. Με απλές εντολές και ελάχιστες απαιτήσεις προετοιμασίας των δεδομένων, μπορεί κανείς να παράξει προβλέψεις με υψηλό βαθμό ακρίβειας, αξιοποιώντας τη δύναμη των ενσωματωμένων αλγοριθμικών μηχανισμών του.

Στην παρούσα εφαρμογή, ο Prophet αξιοποιήθηκε για την πρόβλεψη της ζήτησης ανά προϊόν, ανά ημέρα, με στόχο να υποστηρίξει τη λήψη αποφάσεων σχετικά με την παραγωγή φαγητού, την παραγγελία πρώτων υλών και τη βελτιστοποίηση της διαχείρισης αποθεμάτων. Το σύστημα που δημιουργήθηκε επιτρέπει την επιλογή συγκεκριμένης ημέρας και κατηγοριοποίησης αυτής (π.χ.

κανονική ημέρα, αργία, εξεταστική περίοδος), στέλνοντας τα δεδομένα στο endpoint του μηχανισμού πρόβλεψης. Από εκεί, ο Prophet, αφού συνδυάσει τις ιστορικές τάσεις με τις πληροφορίες της επιλεγμένης ημέρας, επιστρέφει τις προβλεπόμενες τιμές, μαζί με τα ανώτατα και κατώτατα όρια αβεβαιότητας (min – max bounds), για κάθε διαθέσιμο φαγητό.

Αξιοσημείωτο είναι πως ο Prophet όχι μόνο προβλέπει τις μερίδες, αλλά παράγει και **οπτικοποιήσεις** που βοηθούν σημαντικά τη λήψη αποφάσεων. Οι γραφικές απεικονίσεις περιλαμβάνουν τις πραγματικές τιμές, την καμπύλη πρόβλεψης, και τα confidence intervals. Οι γραφικές αυτές απεικονίσεις καθιστούν τη συμπεριφορά του μοντέλου κατανοητή ακόμα και από μη ειδικούς, ενώ παρέχουν άμεσα πληροφορίες για την αξιοπιστία της κάθε πρόβλεψης.

Η δυνατότητα ενσωμάτωσης **εξωτερικών μεταβλητών** (external regressors), όπως καιρικά φαινόμενα ή ειδικές ημερολογιακές περίοδοι, προσθέτει επιπλέον αξία στην ακρίβεια των προβλέψεων. Για παράδειγμα, το σύστημα μπορεί να λάβει υπόψη του ότι ένα φαγητό παρουσιάζει χαμηλότερη κατανάλωση όταν βρέχει, ή ότι κατά τις εξεταστικές περιόδους μειώνεται γενικά η προσέλευση στη λέσχη. Αυτού του είδους τα patterns αποτυπώνονται επιτυχώς από τον Prophet, ο οποίος ενσωματώνει τέτοιες επιρροές ως μεταβλητές στο τελικό του μοντέλο.

Τέλος, ένα στοιχείο που ενίσχυσε την καταλληλότητα του Prophet στο συγκεκριμένο έργο ήταν η **ανοχή του σε ελλιπή ή ασυνεπή δεδομένα**. Σε περιπτώσεις προϊόντων χωρίς επαρκές ιστορικό ή με «θόρυβο» στη χρονοσειρά (π.χ. ασυνήθιστες διακοπές στη ζήτηση λόγω παύσης λειτουργίας της λέσχης), το Prophet προσαρμόζεται δυναμικά και επιχειρεί να ανιχνεύσει τις υποκείμενες τάσεις. Παρ' όλα αυτά, όπως θα αναλυθεί και στα επόμενα υποκεφάλαια, σε περιπτώσεις προϊόντων με ελάχιστες παρατηρήσεις, το μοντέλο ενδέχεται να μην επιστρέφει προβλεπόμενη τιμή, διατηρώντας έτσι μια λογική προφύλαξης έναντι εσφαλμένων εκτιμήσεων.

Συμπερασματικά, η χρήση του Facebook Prophet αποτέλεσε μια **λειτουργική και καινοτόμα λύση** για την πρόβλεψη μερίδων φαγητού, με δυνατότητα αξιοποίησης ακόμα και σε συστήματα με περιορισμένους υπολογιστικούς πόρους. Η ευελιξία του, η προσαρμοστικότητά του και η ευκολία ενσωμάτωσής του σε εφαρμογές τύπου web backend, τον καθιστούν ένα ιδιαίτερα ελκυστικό εργαλείο για τη διαχείριση ζήτησης σε περιβάλλοντα εστίασης, πανεπιστημιακές λέσχες και όχι μόνο.

4.5 Εργαλεία ανάπτυξης και υποστήριξης

Για την αποτελεσματική ανάπτυξη, συνεργασία και διαχείριση του κώδικα, χρησιμοποιήθηκαν σύγχρονα εργαλεία ανάπτυξης.

4.5.1 Git, GitHub

Το **Git** [17] είναι ένα σύστημα ελέγχου εκδόσεων (Version Control System - VCS) ανοιχτού κώδικα, το οποίο χρησιμοποιήθηκε για την παρακολούθηση και διαχείριση των αλλαγών στον κώδικα της εφαρμογής. Το Git επιτρέπει την καταγραφή κάθε αλλαγής, την επαναφορά σε προηγούμενες εκδόσεις και τη συγχώνευση αλλαγών από διαφορετικούς προγραμματιστές.

Το **GitHub** [18] είναι μια διαδικτυακή πλατφόρμα που φιλοξενεί αποθετήρια Git και παρέχει πρόσθετες λειτουργίες για τη διαχείριση έργων και τη συνεργασία.

Στην εργασία, τα Git και GitHub χρησιμοποιήθηκαν για:

- **Έλεγχος εκδόσεων:** Κάθε αλλαγή στον κώδικα (frontend, backend, βάση δεδομένων, scripts Python) καταγράφηκε ως ένα commit στο Git, παρέχοντας ένα λεπτομερές ιστορικό ανάπτυξης. Αυτό επέτρεψε την εύκολη παρακολούθηση των αλλαγών και την επαναφορά σε προηγούμενες λειτουργικές εκδόσεις, αν χρειαζόταν.
- **Συνεργασία:** Αν και η εργασία εκπονήθηκε από έναν μόνο φοιτητή, η χρήση του GitHub προσομοίωσε ένα περιβάλλον ομαδικής ανάπτυξης. Δημιουργήθηκαν branches για την ανάπτυξη συγκεκριμένων λειτουργιών (π.χ., "feature/menu-management", "feature/order-forecasting"), οι οποίες στη συνέχεια συγχωνεύτηκαν στο κύριο branch (main).
- **Διαχείριση έργου:** Το GitHub προσφέρει λειτουργίες όπως Issues (για την παρακολούθηση εργασιών και σφαλμάτων) και Projects, οι οποίες θα μπορούσαν να χρησιμοποιηθούν για την οργάνωση των βημάτων ανάπτυξης και την παρακολούθηση της προόδου.
- **Δημόσια προβολή και τεκμηρίωση:** Το αποθετήριο στο GitHub λειτούργησε ως κεντρικό σημείο για την αποθήκευση του κώδικα και της τεκμηρίωσης (π.χ., README.md), καθιστώντας το έργο προσβάσιμο και επαληθεύσιμο.

4.5.2 VSCode, Mailtrap

Το **Visual Studio Code (VS Code)** [19] είναι ένα ελαφρύ, αλλά ισχυρό, περιβάλλον ανάπτυξης κώδικα (Integrated Development Environment - IDE) που αναπτύχθηκε από τη Microsoft.

Στην παρούσα εργασία, το VS Code χρησιμοποιήθηκε ως το κύριο εργαλείο για τη συγγραφή και επεξεργασία του κώδικα, προσφέροντας:

- **Υποστήριξη πολλαπλών γλωσσών:** Παρέχει ενσωματωμένη υποστήριξη για HTML, CSS,

JavaScript, PHP και Python, συμπεριλαμβανομένων χαρακτηριστικών όπως syntax highlighting, αυτόματη συμπλήρωση κώδικα (IntelliSense) και snippets. Αυτό βελτίωσε την παραγωγικότητα και μείωσε τα σφάλματα.

- **Ενσωματωμένος έλεγχος εκδόσεων:** Η ενσωμάτωση με το Git επέτρεψε την άμεση διαχείριση των commits, των branches και των συγχωνεύσεων απευθείας από το περιβάλλον του IDE.
- **Επέκταση λειτουργικότητας:** Μέσω του πλούσιου οικοσυστήματος επεκτάσεων (extensions), το VS Code προσαρμόστηκε στις ειδικές ανάγκες του έργου, προσθέτοντας λειτουργίες όπως live server για το frontend, PHP debuggers και εργαλεία για τη διαχείριση της MySQL.
- **Εύκολη διαχείριση αρχείων και φακέλων:** Η δομή του project οργανώθηκε αποτελεσματικά, με ξεχωριστούς φακέλους για frontend, backend, βάση δεδομένων και scripts Python, διευκολύνοντας την πλοήγηση και την οργάνωση του κώδικα.

Επιπλέον, για την ασφαλή δοκιμή της αποστολής email κατά την ανάπτυξη, χρησιμοποιήθηκε το **Mailtrap** [20]. Το Mailtrap είναι μια πλατφόρμα που επιτρέπει την παγίδευση των εξερχόμενων email σε ένα ελεγχόμενο περιβάλλον, αποτρέποντας την αποστολή τους σε πραγματικούς παραλήπτες. Αυτό βοήθησε στη δοκιμή και την αποσφαλμάτωση της λειτουργίας αποστολής email της εφαρμογής, διασφαλίζοντας ότι τα email παράγονται σωστά χωρίς να διακινδυνεύει η αποστολή ανεπιθύμητων μηνυμάτων κατά τη φάση ανάπτυξης.

4.5.3 Miro

Το **Miro** [21] είναι ένα διαδραστικό εργαλείο συνεργασίας και σχεδίασης, το οποίο χρησιμοποιήθηκε στην παρούσα εργασία για τη δημιουργία διαγραμμάτων και την οπτικοποίηση της αρχιτεκτονικής του συστήματος. Συγκεκριμένα, σχεδιάστηκαν:

- **Διαγράμματα ροής (flowcharts)** για την απεικόνιση της λειτουργικής ροής των χρηστών στην εφαρμογή.
- **Διαγράμματα αρχιτεκτονικής** που περιγράφουν τη σχέση μεταξύ frontend, backend, βάσης δεδομένων και API.
- **Διαγράμματα βάσης δεδομένων**, τα οποία βοήθησαν στη μοντελοποίηση και κατανόηση των σχέσεων μεταξύ πινάκων.

Η χρήση του Miro επέτρεψε την καθαρή και ευανάγνωστη παρουσίαση της συνολικής δομής της εφαρμογής, ενισχύοντας την κατανόηση και την τεκμηρίωση των τεχνικών επιλογών που έγιναν.

4.5.4 DrawSQL

Το **DrawSQL** [22] είναι ένα διαδικτυακό εργαλείο σχεδίασης βάσεων δεδομένων, το οποίο χρησιμοποιήθηκε για τη δημιουργία του λογικού και φυσικού μοντέλου της βάσης της εφαρμογής. Παρέχει ένα φιλικό περιβάλλον για τη γραφική απεικόνιση πινάκων, πεδίων και σχέσεων μεταξύ τους, ακολουθώντας τις αρχές του σχεσιακού μοντέλου.

Με το DrawSQL σχεδιάστηκαν οι πίνακες της βάσης δεδομένων (όπως χρήστες, παραγγελίες, μενού, φαγητά) και ορίστηκαν οι συσχετίσεις μεταξύ τους (1:1, 1:N, N:M). Η χρήση του εργαλείου βοήθησε στον έγκαιρο εντοπισμό πιθανών σφαλμάτων στη δομή της βάσης, πριν την υλοποίηση με SQL, και συνέβαλε στην τεκμηρίωση του σχεδιασμού της βάσης δεδομένων.

4.5.5 Figma

Το **Figma** είναι ένα σύγχρονο εργαλείο σχεδίασης διεπαφών χρήστη (UI) που υποστηρίζει συνεργατική εργασία και σχεδιασμό στο cloud. Στο πλαίσιο της εργασίας, χρησιμοποιήθηκε για τη δημιουργία **διαδραστικών mockups υψηλής πιστότητας (high-fidelity prototypes)** της τελικής διεπαφής χρήστη, τόσο για τον επισκέπτη/χρήστη όσο και για τον διαχειριστή του συστήματος.

Με το Figma σχεδιάστηκαν σελίδες όπως το μενού της ημέρας, η φόρμα παραγγελίας, η σελίδα διαχείρισης μενού, και η προβολή προβλέψεων. Τα πρωτότυπα διευκόλυναν τη μεταφορά των απαιτήσεων σε οπτική μορφή και αποτέλεσαν οδηγό για την υλοποίηση του frontend με HTML και CSS.

4.5.5 Balsamiq

Το **Balsamiq Mockups** χρησιμοποιήθηκε για τον αρχικό, γρήγορο σχεδιασμό των βασικών διεπαφών της εφαρμογής. Είναι ένα εργαλείο παραγωγής **wireframes χαμηλής πιστότητας (low-fidelity wireframes)**, το οποίο εστιάζει στη λειτουργικότητα και τη διάταξη των στοιχείων αντί για την τελική αισθητική.

Με το Balsamiq δημιουργήθηκαν αρχικά προσχέδια των βασικών σελίδων, όπως η σελίδα παραγγελίας, το ιστορικό χρήστη, η σελίδα login και το dashboard του διαχειριστή. Η απλότητα του εργαλείου επέτρεψε την εύκολη δοκιμή διαφορετικών ιδεών για τη διάταξη κουμπιών, φορμών και πινάκων, πριν την κατασκευή mockups στο Figma.

5. Ανάλυση και Σχεδίαση Συστήματος

Στο παρόν κεφάλαιο περιγράφεται αναλυτικά η αρχιτεκτονική και ο σχεδιασμός του πληροφοριακού συστήματος που υλοποιήθηκε. Αφού έχουν προηγηθεί η ανάλυση των απαιτήσεων των χρηστών και η αξιολόγηση του υφιστάμενου τρόπου λειτουργίας της φοιτητικής λέσχης, το κεφάλαιο αυτό αποσκοπεί στο να αποτυπώσει με δομημένο τρόπο τη λογική οργάνωση, τις τεχνολογικές επιλογές και τις λειτουργικές ροές του συστήματος.

Η ανάλυση ξεκινά με την παρουσίαση της συνολικής αρχιτεκτονικής σε υψηλό επίπεδο, περιγράφοντας τα βασικά υποσυστήματα, τη μεταξύ τους επικοινωνία και τη διασύνδεση με τη βάση δεδομένων. Στη συνέχεια, παρουσιάζεται η ροή δεδομένων εντός του συστήματος, αναλύεται ο τρόπος με τον οποίο οι χρήστες (φοιτητές και διαχειριστές) αλληλεπιδρούν με την εφαρμογή, και περιγράφονται οι λειτουργικές μονάδες βάσει των διαφορετικών ρόλων και δικαιωμάτων.

Το κεφάλαιο περιλαμβάνει επίσης διαγράμματα Use Cases, ER και σχεσιακά μοντέλα, καθώς και αλληλεπιδραστικά σενάρια χρήσης, προκειμένου να αναδειχθεί η συνοχή και η λειτουργικότητα του σχεδιασμού. Τέλος, ιδιαίτερη έμφαση δίνεται στους μηχανισμούς ασφαλείας που έχουν ενσωματωθεί σε επίπεδο λογισμικού, με στόχο την προστασία των δεδομένων, την αυθεντικοποίηση των χρηστών και τη συνολική αξιοπιστία του συστήματος.

Ο συνδυασμός τεχνολογικού σχεδιασμού, σαφούς λογικής ροής και πρόβλεψης αναγκών καθιστά το σύστημα λειτουργικό, επεκτάσιμο και ασφαλές, ανταποκρινόμενο στις σύγχρονες απαιτήσεις ψηφιακής διαχείρισης φοιτητικής σίτισης.

5.1 Αρχιτεκτονική Συστήματος

5.1.1 Αρχιτεκτονική Συστήματος

Η αρχιτεκτονική του συστήματος βασίζεται σε έναν διαχωρισμό ρόλων μεταξύ τριών βασικών επιπέδων: **Frontend**, **Backend API**, και **Prediction Service**. Ο σχεδιασμός ακολουθεί τη φιλοσοφία του loosely coupled, service-oriented architecture, επιτρέποντας την ανεξάρτητη ανάπτυξη, κλιμάκωση και συντήρηση κάθε υποσυστήματος.

Επιμέρους Υποσυστήματα:

- **Frontend (Client UI):** Παρέχει τη διεπαφή αλληλεπίδρασης με τον χρήστη, υλοποιημένη με HTML, Tailwind CSS και JavaScript. Από εδώ πραγματοποιούνται λειτουργίες όπως προβολή μενού, υποβολή παραγγελίας, επιλογή χρονικού slot και –

για τους διαχειριστές – ανάκτηση προβλέψεων μερίδων.

- **Backend (PHP REST API):** Υλοποιεί την επιχειρησιακή λογική του συστήματος. Χειρίζεται αιτήματα από το frontend, διαχειρίζεται τη βάση δεδομένων MySQL, πραγματοποιεί αυθεντικοποίηση χρηστών μέσω JWT, και επικοινωνεί με τον prediction server για την εκτέλεση προβλέψεων.
- **Prediction Server (Flask API + Python):** Πρόκειται για ένα ξεχωριστό microservice υλοποιημένο με Flask, το οποίο είναι υπεύθυνο για την πρόβλεψη της ζήτησης σε μερίδες φαγητού ανά ημερομηνία. Εκτελεί ένα script γραμμένο σε Python (forecast.py) που αξιοποιεί τον αλγόριθμο **Facebook Prophet** και δεδομένα όπως:
 - ημερομηνία πρόβλεψης
 - θερμοκρασία, υγρασία, άνεμος
 - τύπος ημέρας (κανονική, αργία, εξετάσεις)

Το αποτέλεσμα είναι ένα σύνολο προβλέψεων για κάθε είδος φαγητού με τιμές **min**, **max** και **μέσος όρος μερίδων**.

Δομή Επικοινωνίας

Η επικοινωνία μεταξύ των βασικών υποσυστημάτων του συστήματος ακολουθεί μια σαφώς ορισμένη αρχιτεκτονική, εξασφαλίζοντας την αξιόπιστη και επεκτάσιμη ροή δεδομένων. Πιο αναλυτικά:

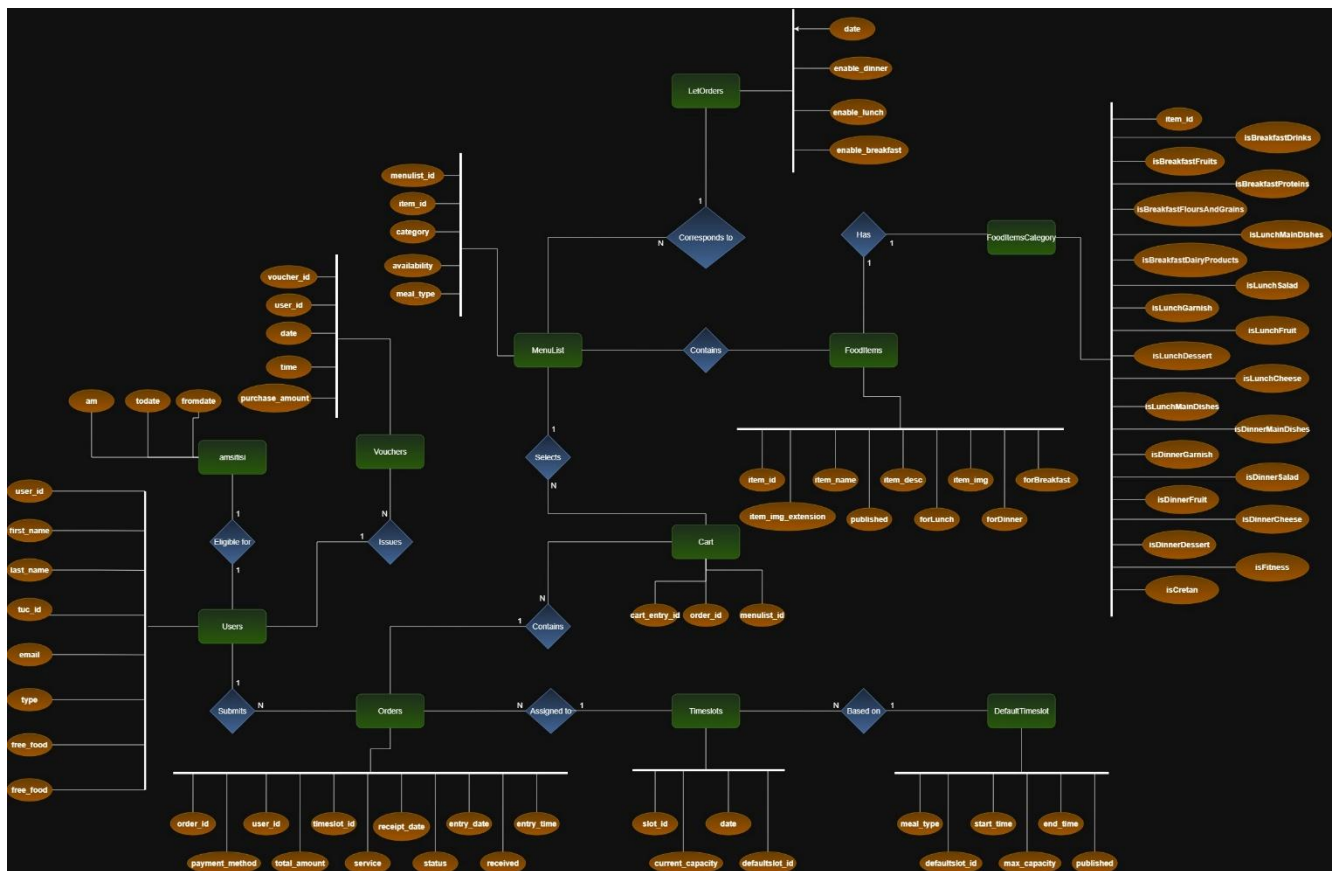
- **Frontend ↔ PHP Backend:**
Η διεπαφή χρήστη επικοινωνεί με το backend μέσω AJAX αιτημάτων, μεταφέροντας και λαμβάνοντας δεδομένα σε μορφή JSON. Αυτή η ασύγχρονη ανταλλαγή καθιστά δυναμική την εμπειρία χρήστη και επιτρέπει ενημερώσεις χωρίς επαναφόρτωση της σελίδας.
- **PHP Backend ↔ Flask Prediction Server:**
Όταν απαιτείται πρόβλεψη ποσοτήτων φαγητού, το backend καλεί το Flask API μέσω HTTP POST αιτήματος στο endpoint /forecast-script, αποστέλλοντας ως JSON τα δεδομένα (ημερομηνία, θερμοκρασία, υγρασία, άνεμος, τύπος ημέρας). Η απάντηση επιστρέφει επίσης ως JSON.
- **PHP Backend ↔ MySQL Database:**
Το backend διαχειρίζεται την αποθήκευση και ανάκτηση όλων των λειτουργικών δεδομένων μέσω άμεσων SQL ερωτημάτων. Τα δεδομένα αυτά περιλαμβάνουν χρήστες, μενού, παραγγελίες, ιστορικό, και στοιχεία vouchers.

5.2 Δομή και Σχέσεις της Βάσης Δεδομένων

5.2.1 Δομή Βάσης Δεδομένων

Η βάση δεδομένων της εφαρμογής είναι σχεσιακή (MySQL) και έχει σχεδιαστεί με σκοπό τη σαφή αποτύπωση των πραγματικών οντοτήτων της φοιτητικής λέσχης, τη μείωση του πλεονασμού (μέσω κανονικοποίησης) και τη βελτιστοποίηση των λειτουργιών ανάκτησης και ενημέρωσης δεδομένων.

Στην εικόνα που ακολουθεί παρουσιάζονται όλοι οι πίνακες της βάσης δεδομένων του συστήματος, καθώς και οι μεταξύ τους σχέσεις.



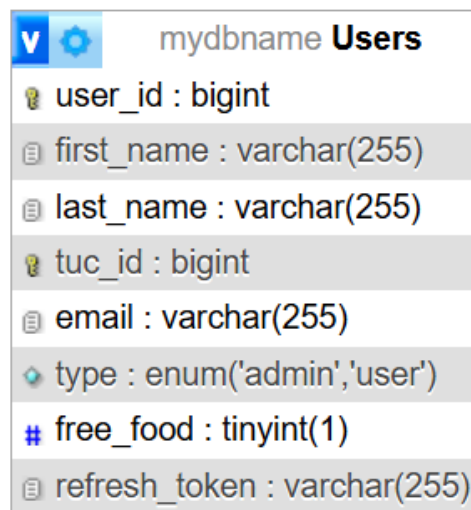
Εικόνα 27: Διάγραμμα οντοτήτων-συσχετίσεων (ERD) της βάσης δεδομένων του συστήματος.

Στο σημείο αυτό κρίνεται σκόπιμη η αναφορά της λειτουργικότητας καθενός από αυτούς.

Χρήστες (Users)

Ο πίνακας **Users** αποθηκεύει βασικά στοιχεία κάθε χρήστη, όπως το μοναδικό αναγνωριστικό του (`user_id`), το όνομα και το επώνυμό του (`first_name`, `last_name`), τον αριθμό μητρώου του από το Πολυτεχνείο Κρήτης (`tuc_id`), καθώς και το μοναδικό του email. Περιλαμβάνει επίσης τον ρόλο του στο σύστημα (`type`, με τιμές *admin* ή *user*) και ένδειξη για το αν δικαιούται δωρεάν σίτιση (`free_food`).

Ο πίνακας αυτός σχετίζεται με τον πίνακα **amsitisi** μέσω του πεδίου `tuc_id`, για την πιστοποίηση των δικαιωμάτων σίτισης



mydbname Users	
🔑	user_id : bigint
📄	first_name : varchar(255)
📄	last_name : varchar(255)
🔑	tuc_id : bigint
📄	email : varchar(255)
⚙️	type : enum('admin','user')
#	free_food : tinyint(1)
📄	refresh_token : varchar(255)

Εικόνα 28: Ο πίνακας *Users* της βάσης δεδομένων με τα βασικά πεδία του.

Παραγγελίες (Orders)

Ο πίνακας **Orders** καταγράφει τις παραγγελίες των χρηστών. Κάθε εγγραφή περιλαμβάνει το μοναδικό αναγνωριστικό της παραγγελίας (`order_id`), το `user_id` του χρήστη που την υπέβαλε, καθώς και το `timeslot_id` που αντιστοιχεί στο χρονικό διάστημα εξυπηρέτησης. Επιπλέον, καταχωρούνται οι ημερομηνίες και ώρες εγγραφής (`entry_date`, `entry_time`) και έκδοσης απόδειξης (`receipt_date`), το συνολικό ποσό της παραγγελίας (`total_amount`), ο τρόπος πληρωμής (`payment_method`), ο τύπος εξυπηρέτησης (`service`), η κατάσταση της παραγγελίας (`status`) και αν αυτή έχει παραληφθεί (`received`).

Ο πίνακας αυτός συνδέεται με τους πίνακες **Users** και **Timeslots** μέσω των πεδίων `user_id` και `timeslot_id` αντίστοιχα.

	mydbname Orders
🔑	order_id : bigint
#	timeslot_id : bigint
📅	receipt_date : date
📅	entry_date : date
🕒	entry_time : time
#	total_amount : decimal(10,2)
🔹	payment_method : enum('cashier','voucher','free')
🔹	service : enum('take-away','eat-there')
🔹	status : enum('accepted','rejected','pending')
#	user_id : bigint
#	received : tinyint(1)

Εικόνα 29: Ο πίνακας *Orders* της βάσης δεδομένων με τα βασικά πεδία του.

Είδη Φαγητού (FoodItems)

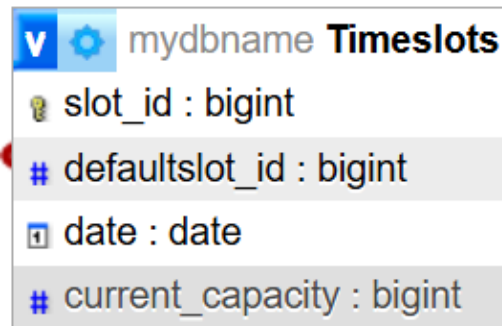
Τα διαθέσιμα είδη φαγητού καταχωρούνται στον πίνακα **FoodItems**, ο οποίος περιλαμβάνει το μοναδικό αναγνωριστικό του κάθε είδους (item_id), το όνομα (item_name), την περιγραφή (item_desc) και την εικόνα του είδους σε δυαδική μορφή (item_img). Επίσης, υπάρχουν πεδία που δηλώνουν για ποιο γεύμα είναι κατάλληλο το είδος (forBreakfast, forLunch, forDinner), καθώς και αν είναι δημοσιευμένο (published).

	mydbname FoodItems
🔑	item_id : bigint
📄	item_name : varchar(255)
📄	item_desc : varchar(255)
🔹	item_img : longblob
#	forBreakfast : tinyint(1)
#	forLunch : tinyint(1)
#	forDinner : tinyint(1)
#	published : tinyint(1)
📄	item_img_extension : varchar(10)

Εικόνα 30: Ο πίνακας *FoodItems* της βάσης δεδομένων με τα βασικά πεδία του.

Χρονικά Διαστήματα (Timeslots)

Ο πίνακας **Timeslots** ορίζει τα συγκεκριμένα χρονικά διαστήματα εξυπηρέτησης των γευμάτων. Περιέχει το αναγνωριστικό του χρονικού διαστήματος (`slot_id`), το οποίο σχετίζεται με ένα πρότυπο διάστημα από τον πίνακα **DefaultTimeslot** μέσω του πεδίου `defaultslot_id`. Επιπλέον, καταχωρείται η ημερομηνία ισχύος του διαστήματος (`date`) και η τρέχουσα διαθέσιμη χωρητικότητα (`current_capacity`).

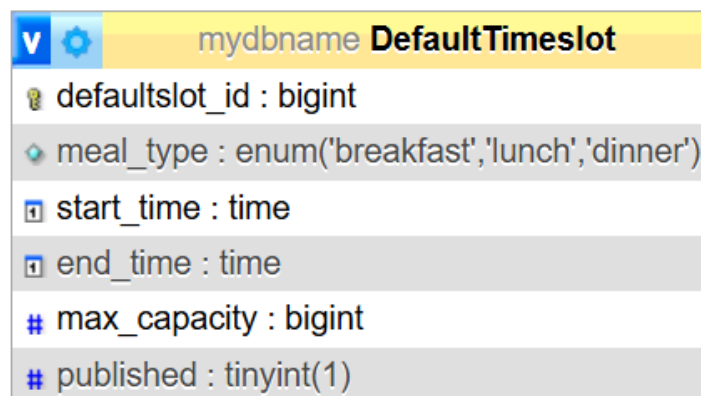


	mydbname	Timeslots
🔑	slot_id	: bigint
#	defaultslot_id	: bigint
📅	date	: date
#	current_capacity	: bigint

Εικόνα 31: Ο πίνακας *Timeslots* της βάσης δεδομένων με τα βασικά πεδία του.

Πρότυπα Χρονικά Διαστήματα (DefaultTimeslot)

Ο πίνακας **DefaultTimeslot** ορίζει τα πρότυπα χρονικά διαστήματα για κάθε τύπο γεύματος, καταγράφοντας το αναγνωριστικό τους (`defaultslot_id`), τον τύπο του γεύματος (`meal_type`), την ώρα έναρξης και λήξης (`start_time`, `end_time`), καθώς και τη μέγιστη χωρητικότητα (`max_capacity`).



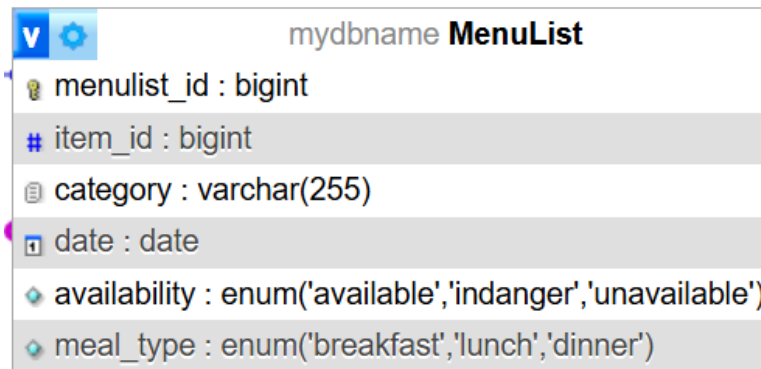
	mydbname	DefaultTimeslot
🔑	defaultslot_id	: bigint
🔗	meal_type	: enum('breakfast','lunch','dinner')
📅	start_time	: time
📅	end_time	: time
#	max_capacity	: bigint
#	published	: tinyint(1)

Εικόνα 32: Ο πίνακας *DefaultTimeslot* της βάσης δεδομένων με τα βασικά πεδία του.

Μενού Ημέρας (MenuList)

Η διαθεσιμότητα των φαγητών για κάθε ημέρα ορίζεται στον πίνακα **MenuList**, ο οποίος περιλαμβάνει το μοναδικό αναγνωριστικό κάθε εγγραφής (menulist_id), το item_id από τον πίνακα FoodItems, την κατηγορία του φαγητού (category), την ημερομηνία ισχύος (date), τη διαθεσιμότητα (availability) και τον τύπο γεύματος (meal_type).

Ο πίνακας αυτός σχετίζεται επίσης με τον πίνακα **LetOrders** μέσω της ημερομηνίας.



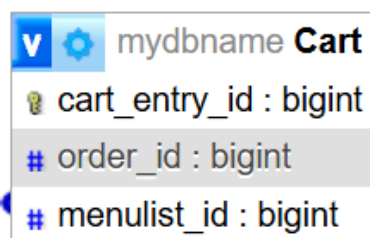
🔑	menulist_id : bigint
#	item_id : bigint
	category : varchar(255)
📅	date : date
🔍	availability : enum('available','indanger','unavailable')
🔍	meal_type : enum('breakfast','lunch','dinner')

Εικόνα 33: Ο πίνακας MenuList της βάσης δεδομένων με τα βασικά πεδία του.

Καλάθι (Cart)

Ο πίνακας **Cart** λειτουργεί ως καλάθι αγορών. Συνδέει τα είδη του μενού με συγκεκριμένες παραγγελίες και περιλαμβάνει το μοναδικό αναγνωριστικό της εγγραφής (cart_entry_id), το order_id της σχετικής παραγγελίας και το menulist_id του φαγητού.

Υπάρχουν ξένες σχέσεις με τους πίνακες Orders και MenuList.



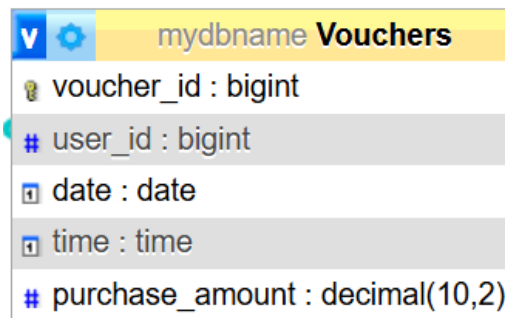
🔑	cart_entry_id : bigint
#	order_id : bigint
#	menulist_id : bigint

Εικόνα 34: Ο πίνακας Cart της βάσης δεδομένων με τα βασικά πεδία του.

Κουπόνια (Vouchers)

Η έκδοση κουπονιών καταγράφεται στον πίνακα **Vouchers**, όπου αποθηκεύεται το μοναδικό αναγνωριστικό κάθε κουπονιού (`voucher_id`), το `user_id` του χρήστη που το λαμβάνει, η ημερομηνία και ώρα έκδοσης (`date`, `time`), καθώς και το ποσό της αγοράς (`purchase_amount`).

Ο πίνακας αυτός συνδέεται με τον πίνακα `Users`.

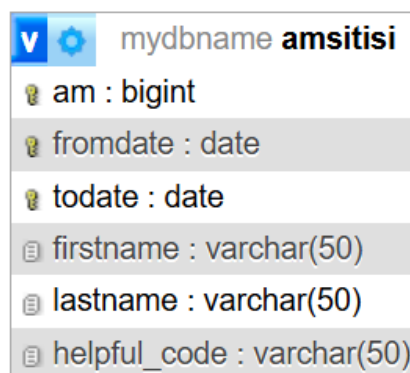


mydbname Vouchers	
🔑	<code>voucher_id : bigint</code>
#	<code>user_id : bigint</code>
📅	<code>date : date</code>
🕒	<code>time : time</code>
#	<code>purchase_amount : decimal(10,2)</code>

Εικόνα 35: Ο πίνακας *Vouchers* της βάσης δεδομένων με τα βασικά πεδία του.

Δικαιώματα Σίτισης (amsitisi)

Ο πίνακας **amsitisi** καταγράφει τα δικαιώματα δωρεάν σίτισης των φοιτητών. Χρησιμοποιεί ως πρωτεύον κλειδί τον αριθμό μητρώου (`am`) και περιλαμβάνει την περίοδο ισχύος του δικαιώματος μέσω των πεδίων `fromdate` και `todate`.

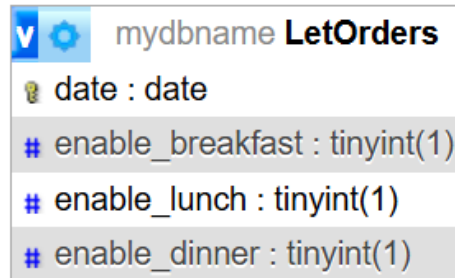


mydbname amsitisi	
🔑	<code>am : bigint</code>
🔑	<code>fromdate : date</code>
🔑	<code>todate : date</code>
📄	<code>firstname : varchar(50)</code>
📄	<code>lastname : varchar(50)</code>
📄	<code>helpful_code : varchar(50)</code>

Εικόνα 36: Ο πίνακας *amsitisi* της βάσης δεδομένων με τα βασικά πεδία του.

Διαθεσιμότητα Γεύματων (LetOrders)

Η διαθεσιμότητα των γευμάτων ανά ημέρα καθορίζεται στον πίνακα **LetOrders**, ο οποίος περιλαμβάνει ως πρωτεύον κλειδί την ημερομηνία (date) και πεδία τύπου boolean για την ενεργοποίηση ή απενεργοποίηση των γευμάτων: enable_breakfast, enable_lunch και enable_dinner.

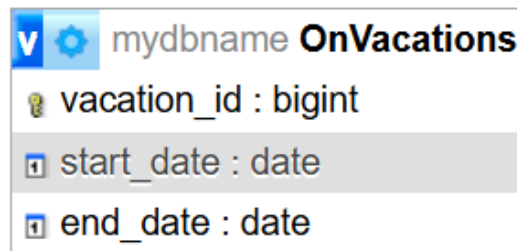


mydbname LetOrders	
🔑	date : date
#	enable_breakfast : tinyint(1)
#	enable_lunch : tinyint(1)
#	enable_dinner : tinyint(1)

Εικόνα 37: Ο πίνακας LetOrders της βάσης δεδομένων με τα βασικά πεδία του.

Περίοδοι Διακοπών (OnVacations)

Ο πίνακας **OnVacations** καταγράφει τις περιόδους διακοπών, με κάθε εγγραφή να περιλαμβάνει το μοναδικό αναγνωριστικό της περιόδου (vacation_id), την ημερομηνία έναρξης (start_date) και την ημερομηνία λήξης (end_date).

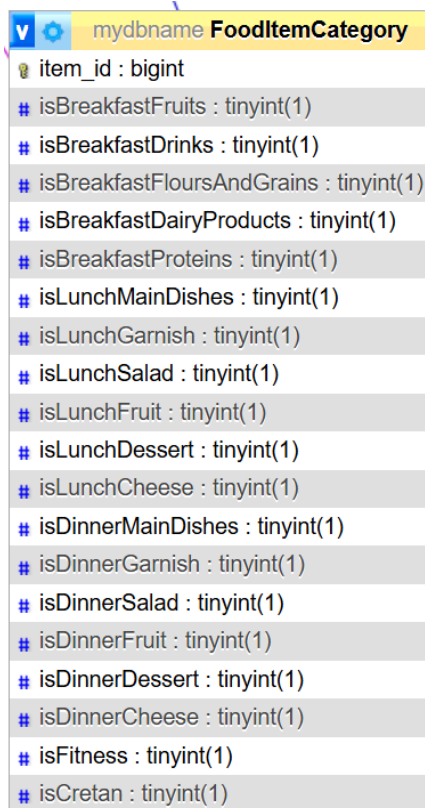


mydbname OnVacations	
🔑	vacation_id : bigint
1	start_date : date
1	end_date : date

Εικόνα 38: Ο πίνακας OnVacations της βάσης δεδομένων με τα βασικά πεδία του.

Κατηγοριοποίηση Φαγητών (FoodItemCategory)

Ο πίνακας FoodItemCategory επιτρέπει την πολυδιάστατη κατηγοριοποίηση των φαγητών, προσδίδοντας ευελιξία στην εμφάνιση, φιλτράρισμα και πρόβλεψη ζήτησης. Κάθε καταχώρηση συνδέεται με ένα είδος φαγητού (item_id) και περιλαμβάνει πλήθος από BOOLEAN πεδία που ξεκινούν με is..., εκφράζοντας την ανήκουσα κατηγορία του είδους.



mydbname FoodItemCategory	
item_id	bigint
isBreakfastFruits	tinyint(1)
isBreakfastDrinks	tinyint(1)
isBreakfastFloursAndGrains	tinyint(1)
isBreakfastDairyProducts	tinyint(1)
isBreakfastProteins	tinyint(1)
isLunchMainDishes	tinyint(1)
isLunchGarnish	tinyint(1)
isLunchSalad	tinyint(1)
isLunchFruit	tinyint(1)
isLunchDessert	tinyint(1)
isLunchCheese	tinyint(1)
isDinnerMainDishes	tinyint(1)
isDinnerGarnish	tinyint(1)
isDinnerSalad	tinyint(1)
isDinnerFruit	tinyint(1)
isDinnerDessert	tinyint(1)
isDinnerCheese	tinyint(1)
isFitness	tinyint(1)
isCretan	tinyint(1)

Εικόνα 39: Ο πίνακας FoodItemCategory της βάσης δεδομένων με τα βασικά πεδία του.

Αναλυτικά, οι κατηγορίες περιλαμβάνουν:

Πρωινό (Breakfast)

- **isBreakfastFruits** – Φρούτα για κατανάλωση στο πρωινό.
- **isBreakfastDrinks** – Ποφήματα/ποτά πρωινού (π.χ. καφές, γάλα, χυμός).
- **isBreakfastFloursAndGrains** – Δημητριακά, ψωμιά, κουλούρια, αρτοσκευάσματα.
- **isBreakfastDairyProducts** – Γαλακτοκομικά προϊόντα πρωινού (π.χ. γιαούρτι, βούτυρο).
- **isBreakfastProteins** – Πηγές πρωτεΐνης για πρωινό (π.χ. αυγά, αλλαντικά).

Μεσημεριανό (Lunch)

- **isLunchMainDishes** – Κυρίως πιάτα μεσημεριανού.
- **isLunchGarnish** – Συνοδευτικά πιάτα, όπως ρύζι, πατάτες κ.λπ.
- **isLunchSalad** – Σαλάτες για μεσημεριανό.
- **isLunchFruit** – Φρούτα μετά το κυρίως γεύμα.
- **isLunchDessert** – Γλυκά/επιδόρπια μετά το μεσημεριανό.
- **isLunchCheese** – Τυριά που προσφέρονται στο μεσημεριανό.

Βραδινό (Dinner)

- **isDinnerMainDishes** – Κυρίως πιάτα βραδινού.
- **isDinnerGarnish** – Συνοδευτικά πιάτα για βραδινό.
- **isDinnerSalad** – Σαλάτες κατά τη διάρκεια του δείπνου.
- **isDinnerFruit** – Φρούτα στο βραδινό γεύμα.
- **isDinnerDessert** – Γλυκά βραδινού.
- **isDinnerCheese** – Τυριά ως μέρος ή συνοδευτικό του δείπνου.

Ειδικές ή Θεματικές Κατηγορίες

- **isFitness** – Πιάτα με χαμηλά λιπαρά, κατάλληλα για άτομα που ακολουθούν πρόγραμμα διατροφής ή αθλητές.
- **isCretan** – Παραδοσιακές συνταγές της Κρητικής κουζίνας (π.χ. ντάκος, κρητικά όσπρια, πιταράκια).

Η ύπαρξη αυτών των πεδίων επιτρέπει στο σύστημα να παρέχει εξατομικευμένες επιλογές στους χρήστες, να στηρίζει φίλτρα προβολής ανά διατροφή και να διευκολύνει την πρόβλεψη ζήτησης ανά κατηγορία φαγητού.

5.2.2 Ανάλυση Σχέσεων Πινάκων και Κατηγοριοποίηση

Στις σχεσιακές βάσεις δεδομένων, οι **πίνακες (relations)** συσχετίζονται μεταξύ τους για να αποδώσουν τη λογική συνάφεια μεταξύ των δεδομένων. Οι βασικοί τύποι σχέσεων είναι:

- **Ένα προς ένα (1:1)**: Μια εγγραφή ενός πίνακα αντιστοιχεί σε **ακριβώς μία** εγγραφή του άλλου και αντίστροφα. Χρήσιμη όταν θέλουμε να διαχωρίσουμε υποσύνολα πληροφοριών.
- **Ένα προς πολλά (1:N)**: Μια εγγραφή στον έναν πίνακα αντιστοιχεί σε **πολλές εγγραφές** στον άλλο. Είναι η πιο συχνή σχέση και επιτυγχάνεται μέσω ξένου κλειδιού.
- **Πολλά προς πολλά (N:N)**: Πολλές εγγραφές του ενός πίνακα σχετίζονται με πολλές του

άλλου. Υλοποιείται μέσω **ενδιάμεσου πίνακα** που περιέχει δύο ξένα κλειδιά.

Κάθε πίνακας ορίζει:

- **Πρωτεύον κλειδί (Primary Key – PK):** Μοναδικό αναγνωριστικό κάθε εγγραφής.
- **Δευτερεύον κλειδί (Foreign Key – FK):** Πεδίο που παραπέμπει σε πρωτεύον κλειδί άλλου πίνακα, και υλοποιεί τη συσχέτιση μεταξύ πινάκων.

Αναλυτική Περιγραφή Σχέσεων Πινάκων με PK και FK

- **Users → Orders**
 - Τύπος σχέσης: 1:N
 - PK: Users.user_id
FK: Orders.user_id
 - Περιγραφή: Κάθε χρήστης (Users.user_id) μπορεί να έχει πολλές παραγγελίες (Orders.user_id).
 - Αιτιολόγηση: Ένας πελάτης μπορεί να καταχωρήσει πολλές παραγγελίες, αλλά κάθε παραγγελία ανήκει μόνο σε έναν χρήστη.
- **Users → Vouchers**
 - Τύπος σχέσης: 1:N
 - PK: Users.user_id
FK: Vouchers.user_id
 - Περιγραφή: Κάθε χρήστης μπορεί να εκδώσει πολλά κουπόνια (Vouchers.user_id).
 - Αιτιολόγηση: Η σύνδεση δείχνει ότι ο κάθε χρήστης μπορεί να πραγματοποιήσει πολλαπλές αγορές.
- **Users.tuc_id → amsitisi.am**
 - Τύπος σχέσης: 1:1
 - PK: amsitisi.am
FK: Users.tuc_id
 - Περιγραφή: Ο πίνακας Users συνδέεται με τον πίνακα amsitisi μέσω του πεδίου tuc_id.
 - Αιτιολόγηση: Κάθε tuc_id (αριθμός μητρώου) αντιστοιχεί σε μία μοναδική εγγραφή στον πίνακα amsitisi, η οποία περιγράφει τα δικαιώματα σίτισης ενός φοιτητή.
- **Orders → Timeslots**
 - Τύπος σχέσης: N:1
 - PK: Timeslots.slot_id
FK: Orders.timeslot_id

- **Περιγραφή:** Κάθε παραγγελία αφορά ένα συγκεκριμένο timeslot (Orders.timeslot_id).
- **Αιτιολόγηση:** Ένα timeslot μπορεί να εξυπηρετήσει πολλές παραγγελίες, αλλά κάθε παραγγελία ανήκει σε ένα μόνο timeslot.
- **Timeslots → DefaultTimeslot**
 - **Τύπος σχέσης:** N:1
 - **PK:** DefaultTimeslot.defaultslot_id
FK: Timeslots.defaultslot_id
 - **Περιγραφή:** Πολλά timeslots μπορούν να βασίζονται στο ίδιο default timeslot (Timeslots.defaultslot_id).
 - **Αιτιολόγηση:** Ένα πρότυπο μπορεί να επαναχρησιμοποιείται για πολλές ημερομηνίες.
- **Orders → Cart**
 - **Τύπος σχέσης:** 1:N
 - **PK:** Orders.order_id
FK: Cart.order_id
 - **Περιγραφή:** Μία παραγγελία μπορεί να έχει πολλά είδη στο καλάθι (Cart.order_id).
 - **Αιτιολόγηση:** Το καλάθι περιέχει πολλαπλές εγγραφές ανά παραγγελία, που αντιστοιχούν στα φαγητά που επιλέχθηκαν.
- **MenuList → Cart**
 - **Τύπος σχέσης:** 1:N
 - **PK:** MenuList.menulist_id
FK: Cart.menulist_id
 - **Περιγραφή:** Ένα φαγητό από το MenuList μπορεί να μπει σε πολλά καλάθια.
 - **Αιτιολόγηση:** Το ίδιο φαγητό μπορεί να χρησιμοποιηθεί σε πολλές παραγγελίες διαφορετικών χρηστών.
- **MenuList.item_id → FoodItems.item_id**
 - **Τύπος σχέσης:** N:1
 - **PK:** FoodItems.item_id
FK: MenuList.item_id
 - **Περιγραφή:** Πολλά είδη του μενού (MenuList) αναφέρονται στο ίδιο FoodItem.
 - **Αιτιολόγηση:** Το ίδιο είδος φαγητού μπορεί να είναι διαθέσιμο σε πολλές ημερομηνίες και για διαφορετικά γεύματα.

- **MenuList.date → LetOrders.date**
 - Τύπος σχέσης: N:1
 - PK: LetOrders.date
FK: MenuList.date
 - Περιγραφή: Κάθε εγγραφή στο MenuList αντιστοιχεί σε μια ημερομηνία του πίνακα LetOrders, που δηλώνει αν είναι ενεργό κάποιο γεύμα.
 - Αιτιολόγηση: Η ημερομηνία καθορίζει αν είναι ενεργοποιημένο το γεύμα εκείνη την ημέρα.
- **FoodItems → FoodItemCategory**
 - Τύπος σχέσης: 1:1
 - PK: FoodItems.item_id
FK: FoodItemCategory.item_id
 - Περιγραφή: Κάθε είδος φαγητού έχει ένα σύνολο κατηγοριών στο FoodItemCategory, με ίδιο item_id.
 - Αιτιολόγηση: Η συσχέτιση είναι 1:1 καθώς κάθε φαγητό έχει μόνο μία εγγραφή με τα category flags.

Πολλές-Προς-Πολλές (N:N) Σχέσεις με Ενδιάμεσο Πίνακα

Αν και καμία σχέση δεν ορίζεται ρητά ως N:N, υπάρχει περίπτωση που υλοποιείται έμμεσα ως τέτοια μέσω ενδιάμεσου πίνακα:

- **Orders ↔ MenuList μέσω Cart**
 - Τύπος σχέσης: N:N
 - Ερμηνεία: Μία παραγγελία περιλαμβάνει πολλά φαγητά και ένα φαγητό μπορεί να ανήκει σε πολλές παραγγελίες.
 - PKs: Orders.order_id, MenuList.menulist_id
FKs στον Cart:
 - order_id → Orders.order_id
 - menulist_id → MenuList.menulist_id
 - Μορφή υλοποίησης: Πίνακας Cart λειτουργεί ως συνδετικός πίνακας N:N σχέσης.

5.3 Επικοινωνία Frontend – Backend

Η επικοινωνία μεταξύ του frontend και του backend της εφαρμογής αποτελεί κρίσιμο μηχανισμό λειτουργίας, επιτρέποντας στον χρήστη (φοιτητή ή διαχειριστή) να εκτελεί ενέργειες, όπως υποβολή παραγγελίας, προβολή μενού ή διαχείριση της λέσχης, χωρίς την ανάγκη ανανέωσης της σελίδας. Η ανταλλαγή δεδομένων πραγματοποιείται μέσω **ασύγχρονων αιτημάτων (AJAX)** και υλοποιείται με χρήση του **RESTful προτύπου**, με ανταλλαγή δεδομένων σε μορφή **JSON**.

5.3.1 Περιγραφή Μηχανισμού AJAX–PHP

Το frontend της εφαρμογής έχει τη δυνατότητα να στέλνει αιτήματα HTTP (GET ή POST) προς συγκεκριμένα PHP scripts, τα οποία λειτουργούν ως endpoints. Η επικοινωνία συνοδεύεται από:

- **JWT Authentication:** Κάθε αίτημα συνοδεύεται από Authorization: Bearer <token>, το οποίο επαληθεύεται από το backend για την αυθεντικοποίηση του χρήστη και τον έλεγχο δικαιωμάτων πρόσβασης.
- **CSRF Protection (Cross-Site Request Forgery):** Κάθε POST αίτηση που αλλάζει δεδομένα περιλαμβάνει **CSRF token** που παράγεται κατά την είσοδο του χρήστη και επαληθεύεται στο backend. Αιτήματα χωρίς έγκυρο token απορρίπτονται, προστατεύοντας από κακόβουλες φόρμες τρίτων ιστότοπων.
- **XSS Protection (Cross-Site Scripting):** Όλες οι εισροές από τον χρήστη υφίστανται **έλεγχο και καθαρισμό** πριν αποθηκευτούν ή εμφανιστούν.
 - Η έξοδος από PHP script γίνεται με χρήση htmlspecialchars() ή htmlentities() για να αποφευχθεί η εκτέλεση κακόβουλου HTML ή JavaScript.
 - Η εισαγωγή νέων δεδομένων καθαρίζεται και αποθηκεύεται με ασφάλεια.
- **FormData** ή JSON payload για μεταβίβαση παραμέτρων

Ο server ελέγχει την εγκυρότητα των δεδομένων και την εξουσιοδότηση του χρήστη και επιστρέφει **δομημένη JSON απόκριση**, η οποία είτε περιέχει επιτυχές αποτέλεσμα, είτε κατάλληλο μήνυμα σφάλματος.

5.3.2 Ενδεικτικό Σενάριο: Ενεργοποίηση Τύπου Γεύματος (Meal Type Toggle)

Ένα χαρακτηριστικό παράδειγμα της επικοινωνίας frontend–backend είναι η ενεργοποίηση ή απενεργοποίηση της δυνατότητας λήψης παραγγελιών για συγκεκριμένο τύπο γεύματος (πρωινό, μεσημεριανό ή βραδινό) από τον διαχειριστή της λέσχης. Η λειτουργία αυτή ενεργοποιείται με το πάτημα ενός checkbox από τον διαχειριστή στο dashboard της εφαρμογής. Στην περίπτωση που ο

διαχειριστής επιλέξει να ενεργοποιήσει ή να απενεργοποιήσει ένα γεύμα, καλείται η JavaScript συνάρτηση “updateMealStatus”:

```
1. // When click the enable checkbox update meal type status

2. async function updateMealStatus(
3.   mealType,
4.   isChecked,
5.   tokenRefreshAttempted = false,
6. ) {
7.   try {
8.     // Ensure JWT is present
9.     if (!sessionStorage.getItem("jwt")) {
10.      throw new Error("No authorization token found.");
11.    }
12.
13.    // Confirmation message based on checkbox state
14.    const message = isChecked
15.      ? `Are you ready to accept orders for ${mealType}?`
16.      : `Are you sure you want to <strong>not</strong> accept more orders for ${mealType}?`;
17.
18.    // Use SweetAlert for confirmation
19.    const result = await Swal.fire({
20.      title: "Confirm Action",
21.      html: message,
22.      icon: "warning",
23.      showCancelButton: true,
24.      confirmButtonText: "Yes",
25.      cancelButtonText: "No",
26.      confirmButtonColor: "#3085d6",
27.      cancelButtonColor: "#d33",
28.    });
29.
30.    if (result.isConfirmed) {
31.      // Get CSRF token
32.      const csrfToken = document
33.        .querySelector('meta[name="csrf-token"]')
34.        .getAttribute("content");
35.      const date = getDateValue();
36.
37.      // Prepare form data
38.      const formData = new FormData();
39.      formData.append("mealType", mealType);
40.      formData.append("enabled", isChecked ? 1 : 0);
41.      formData.append("date", date);
42.
43.      // Make the POST request
44.      const response = await fetch("updateMealStatus.php", {
45.        method: "POST",
46.        body: formData,
47.        headers: {
48.          "X-CSRF-Token": csrfToken,
49.          Authorization: `Bearer ${sessionStorage.getItem("jwt")}`,
50.        },
51.      });
52.
53.      // Handle token expiration
54.      if (
55.        (response.status === 401 || response.status === 403) &&
56.        !tokenRefreshAttempted
57.      ) {
58.        if (await refreshAccessToken()) {
59.          return updateMealStatus(mealType, isChecked, true);
60.        }
61.      }
62.    }
63.  }
64.}
```

```

60.         } else {
61.             logout();
62.             return;
63.         }
64.     }
65.
66.     // Handle the response
67.     const data = await response.json();
68.     if (data.success) {
69.         Swal.fire({
70.             toast: true,
71.             position: "top-end",
72.             icon: "success",
73.             title: "Success!",
74.             text: isChecked
75.                 ? `Now you are ready to get orders for ${mealType}.`
76.                 : `No more orders will be taken for ${mealType}.`,
77.             showConfirmButton: false,
78.             timer: 3000,
79.             timerProgressBar: true,
80.             didOpen: (toast) => {
81.                 toast.addEventListener("mouseenter", Swal.stopTimer);
82.                 toast.addEventListener("mouseleave", Swal.resumeTimer);
83.             },
84.         });
85.     } else {
86.         Swal.fire({
87.             title: "Error!",
88.             text: `Failed to update meal status: ${data.message}`,
89.             icon: "error",
90.             confirmButtonColor: "#d33",
91.         });
92.     }
93. }
94. } catch (error) {
95.     console.error("Error:", error);
96.     Swal.fire({
97.         title: "Error!",
98.         text: "Sorry, an unexpected issue occurred while trying to update the meal status. Please
try again.",
99.         icon: "error",
100.        confirmButtonColor: "#d33",
101.        confirmButtonText: "OK",
102.    }).then(() => {
103.        location.reload();
104.    });
105. }
106. }

```

Η συνάρτηση `updateMealStatus` ενεργοποιείται κατά το πάτημα του checkbox και πρώτα ελέγχει αν υπάρχει αποθηκευμένο JWT στον session storage του browser. Αν δεν υπάρχει, εμφανίζεται μήνυμα σφάλματος. Έπειτα, εμφανίζεται ένα παράθυρο επιβεβαίωσης μέσω της βιβλιοθήκης SweetAlert, όπου ο χρήστης πρέπει να επιβεβαιώσει την ενέργεια. Εφόσον ο διαχειριστής επιβεβαιώσει, δημιουργείται ένα FormData αντικείμενο με τις εξής παραμέτρους:

- `mealType`: ο τύπος γεύματος ("breakfast", "lunch", "dinner")
- `enabled`: η κατάσταση ενεργοποίησης (1 για ενεργό, 0 για ανενεργό)
- `date`: η ημερομηνία που αφορά η αλλαγή

Στη συνέχεια, αποστέλλεται ένα fetch POST αίτημα προς το αρχείο updateMealStatus.php, μαζί με τους απαραίτητους headers:

- Authorization: Bearer <JWT>
- X-CSRF-Token: <token>

Αν η απόκριση του server είναι επιτυχής (data.success = true), εμφανίζεται μήνυμα επιβεβαίωσης (toast). Σε περίπτωση σφάλματος, εμφανίζεται μήνυμα λάθους, ενώ αν το JWT έχει λήξει, γίνεται προσπάθεια ανανέωσης μέσω της συνάρτησης refreshAccessToken.

5.3.3 Λειτουργία του Backend: updateMealStatus.php

Το backend αρχείο updateMealStatus.php είναι υπεύθυνο για την επεξεργασία του αιτήματος που στάλθηκε από το frontend.

Παρακάτω παρουσιάζεται η PHP συνάρτηση “updateMealStatus.php”.

```
1. <?php
2. include '../.../php/db-interaction.php';
3. require_once '../.../vendor/autoload.php';
4. use \Firebase\JWT\JWT;
5. use \Firebase\JWT\Key;
6. use \Dotenv\Dotenv;
7.
8. // Load environment variables
9. $dotenv = Dotenv::createImmutable(__DIR__ . '/../.../');
10. $dotenv->load();
11.
12. $secret_key = $_ENV['SECRET_KEY'];
13.
14. header('Content-Type: application/json');
15.
16. session_start();
17.
18. if ($_SERVER["REQUEST_METHOD"] === "POST") {
19.     // CSRF Token Validation
20.     $submittedToken = $_SERVER['HTTP_X_CSRF_TOKEN'] ?? '';
21.
22.     if (!validateCsrfToken($submittedToken)) {
23.         http_response_code(403);
24.         echo json_encode(['status' => false, 'message' => 'CSRF token validation failed']);
25.         exit();
26.     }
27.
28.     // JWT Token Validation
29.     $authHeader = getAuthorizationHeader();
30.     if (!$authHeader) {
31.         http_response_code(401);
32.         echo json_encode(['status' => false, 'message' => 'Authorization header missing']);
33.         exit();
34.     }
35.
36.     $user_id = decodeJWT($secret_key, $authHeader);
```

```

37.     if ($user_id === null) {
38.         http_response_code(401);
39.         echo json_encode(['status' => false, 'message' => 'Invalid JWT token']);
40.         exit();
41.     }
42.
43.     if (isset($_POST['mealType'], $_POST['enabled'], $_POST['date'])) {
44.         $mealType = $_POST['mealType'];
45.         $enabled = $_POST['enabled'];
46.         $date = $_POST['date'];
47.
48.         $column = '';
49.         switch ($mealType) {
50.             case 'breakfast':
51.                 $column = 'enable_breakfast';
52.                 break;
53.             case 'lunch':
54.                 $column = 'enable_lunch';
55.                 break;
56.             case 'dinner':
57.                 $column = 'enable_dinner';
58.                 break;
59.             default:
60.                 echo json_encode(['success' => false, 'message' => 'Invalid meal type.']);
61.                 exit;
62.         }
63.
64.         $pdo = connectToDB();
65.         if ($pdo) {
66.             $result = updateMealStatus($pdo, $column, $enabled, $date);
67.             echo json_encode($result);
68.         } else {
69.             echo json_encode(['success' => false, 'message' => 'Failed to connect to the
database.']);
70.         }
71.
72.         if ($pdo) {
73.             closeConnectionToDB($pdo);
74.         }
75.     } else {
76.         echo json_encode(['success' => false, 'message' => 'Missing parameters.']);
77.     }
78. } else {
79.     echo json_encode(['success' => false, 'message' => 'Invalid request method.']);
80. }
81.
82. // JWT decoding function
83. function decodeJWT($secret_key, $authHeader) {
84.     $token = str_replace('Bearer ', '', trim($authHeader));
85.
86.     try {
87.         $decoded = JWT::decode($token, new Key($secret_key, 'HS256'));
88.     } catch (\Firebase\JWT\ExpiredException $e) {
89.         http_response_code(401);
90.         echo json_encode(['status' => false, 'message' => 'Token has expired']);
91.         exit();
92.     } catch (\Firebase\JWT\SignatureInvalidException $e) {
93.         http_response_code(401);
94.         echo json_encode(['status' => false, 'message' => 'Invalid token signature']);
95.         exit();
96.     } catch (\Exception $e) {
97.         http_response_code(401);
98.         echo json_encode(['status' => false, 'message' => 'Token decoding failed']);
99.         exit();
100.    }

```

```

101.
102.     if ($decoded->exp < time()) {
103.         http_response_code(401);
104.         echo json_encode(['status' => false, 'message' => 'Token has expired']);
105.         exit();
106.     }
107.
108.     $decoded_array = (array) $decoded;
109.     $data = (array) $decoded_array['data'];
110.     return $data['user_id'];
111. }
112.

```

Εν συνεχεία παρουσιάζεται η PHP συνάρτησης “updateMealStatus” από το script “db-interaction.php”.

```

1. function updateMealStatus($pdo, $column, $enabled, $date) {
2.     try {
3.         $sql = "UPDATE LetOrders SET $column = :enabled WHERE date = :date";
4.         $stmt = $pdo->prepare($sql);
5.         $stmt->bindParam(':enabled', $enabled, PDO::PARAM_INT);
6.         $stmt->bindParam(':date', $date);
7.         $stmt->execute();
8.         return ["success" => true];
9.     } catch (PDOException $e) {
10.        return ["success" => false, "message" => "Error updating meal status: " . $e-
>getMessage()];
11.    }
12. }

```

Το backend script updateMealStatus.php είναι υπεύθυνο για την παραλαβή και επεξεργασία του αιτήματος του frontend. Αναλυτικά η ροή εκτελείται ως εξής:

1. Ορίζονται τα απαραίτητα imports και φορτώνονται οι μεταβλητές περιβάλλοντος από το αρχείο .env, π.χ. το SECRET_KEY για το JWT.
2. Γίνεται έλεγχος ότι το αίτημα είναι POST. Αν όχι, επιστρέφεται:

```
{ "success": false, "message": "Invalid request method." }
```

3. Ελέγχεται η εγκυρότητα του CSRF token μέσω της συνάρτησης validateCsrfToken(). Αν αποτύχει, επιστρέφεται:

```
{ "status": false, "message": "CSRF token validation failed" }
```

4. Ελέγχεται η παρουσία του header Authorization. Αν λείπει ή είναι κενός:

```
{ "status": false, "message": "Authorization header missing" }
```

5. Αν υπάρχει JWT, γίνεται αποκωδικοποίηση μέσω decodeJWT(). Αν το token είναι ληγμένο, παραπονημένο ή άκυρο, εμφανίζεται μήνυμα:

```
{ "status": false, "message": "Token has expired" }
```

ή

```
{ "status": false, "message": "Invalid token signature" }
```

ή

```
{ "status": false, "message": "Token decoding failed" }
```

6. Ελέγχεται ότι τα απαραίτητα POST πεδία (mealType, enabled, date) είναι παρόντα. Αν όχι:

```
{ "success": false, "message": "Missing parameters." }
```

7. Αν όλα είναι εντάξει, γίνεται αντιστοίχιση του mealType με τη σωστή στήλη της βάσης:

```
1. switch ($mealType) {  
2.   case 'breakfast': $column = 'enable_breakfast'; break;  
3.   case 'lunch': $column = 'enable_lunch'; break;  
4.   case 'dinner': $column = 'enable_dinner'; break;  
5. }
```

8. Στη συνέχεια καλείται η συνάρτηση updateMealStatus() η οποία εκτελεί την SQL εντολή UPDATE.

Ανάλυση SQL Εντολής

Η εντολή που εκτελείται είναι η εξής:

```
UPDATE LetOrders SET enable_lunch = :enabled WHERE date = :date
```

Η παραπάνω εντολή χρησιμοποιεί παραμετρικούς δεσμευτές για να αποτρέψει επιθέσεις SQL injection. Αναλυτικά:

- Ορίζει ποια στήλη θα ενημερωθεί (enable_lunch, enable_breakfast ή enable_dinner)
- Το :enabled είναι η νέα τιμή (1 ή 0)
- Το :date είναι η ημερομηνία που αφορά

Αν η εκτέλεση είναι επιτυχής, επιστρέφεται:

```
{ "success": true }
```

Αν υπάρξει σφάλμα σύνδεσης ή εκτέλεσης:

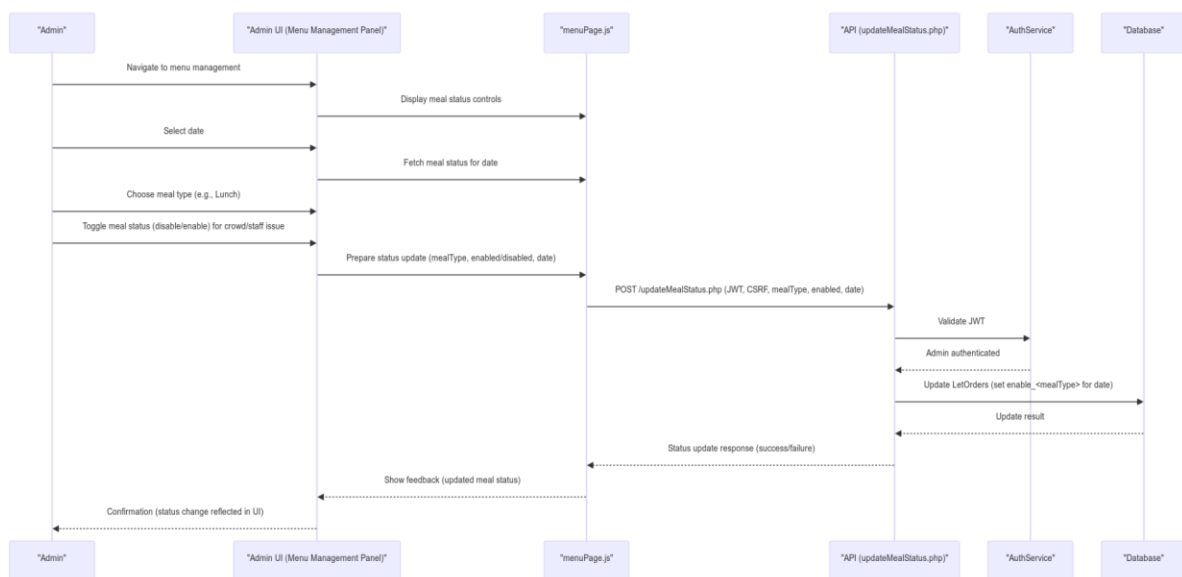
```
{ "success": false, "message": "Error updating meal status: <μήνυμα PDO>" }
```

ή

```
{ "success": false, "message": "Failed to connect to the database." }
```

5.3.4 Διάγραμμα Ακολουθίας Ενεργειών

Το sequence diagram "Admin Change Meal Status" που ακολουθεί απεικονίζει τη ροή δεδομένων όταν ένας διαχειριστής ενεργοποιεί ή απενεργοποιεί την επιλογή παραγγελίας για ένα συγκεκριμένο τύπο γεύματος (π.χ. πρωινό, μεσημεριανό, βραδινό) μέσα από την πλατφόρμα διαχείρισης.



Εικόνα 40: Sequence Diagram "Admin Change Meal Status" με τη ροή δεδομένων.

Η διαδικασία περιγράφεται παρακάτω βήμα προς βήμα, καλύπτοντας τόσο τις ενέργειες του frontend όσο και τις αντίστοιχες διεργασίες στο backend:

1. User Interaction στο Frontend

Ο διαχειριστής επιλέγει ή αποεπιλέγει το checkbox που ενεργοποιεί ή απενεργοποιεί την αποδοχή παραγγελιών για έναν συγκεκριμένο τύπο γεύματος (meal type). Αυτό το interaction ενεργοποιεί την JavaScript συνάρτηση updateMealStatus(...).

2. Επιβεβαίωση Ενέργειας & Έλεγχος Αυθεντικοποίησης

Πριν προχωρήσει η αποστολή δεδομένων, η συνάρτηση ελέγχει αν υπάρχει έγκυρο JWT token αποθηκευμένο στο sessionStorage. Εφόσον υπάρχει, εμφανίζεται modal παραθύρου επιβεβαίωσης (μέσω SweetAlert), όπου ζητείται από τον διαχειριστή να επιβεβαιώσει την πρόθεσή του. Αν δεν υπάρχει token ή αν ο διαχειριστής ακυρώσει την ενέργεια, η διαδικασία διακόπτεται. Αντίθετα, αν επιβεβαιώσει, η διαδικασία συνεχίζεται με την προετοιμασία των δεδομένων για αποστολή.

3. CSRF Token + JWT Token Προστασία

Αμέσως μετά την επιβεβαίωση, ανακτάται το CSRF token από το <meta> tag του HTML εγγράφου και μαζί με το JWT token ενσωματώνεται στο header του αιτήματος. Η συνάρτηση δημιουργεί ένα FormData αντικείμενο με τα στοιχεία: τον τύπο γεύματος (mealType), τη νέα του κατάσταση (enabled ως 1 ή 0) και την επιλεγμένη ημερομηνία (date). Το frontend αποστέλλει POST αίτημα προς τον backend πόρο updateMealStatus.php μαζί με όλα τα παραπάνω δεδομένα και με πλήρη προστασία από CSRF και JWT validation.

4. Backend: Αρχική Επεξεργασία & Έλεγχοι

Από τη στιγμή που το backend λαμβάνει το POST αίτημα, ξεκινούν οι βασικοί έλεγχοι ασφαλείας. Αρχικά, επαληθεύεται η εγκυρότητα του CSRF token, και αν ο έλεγχος αποτύχει, επιστρέφεται σφάλμα με κωδικό 403. Στη συνέχεια, το backend αναλύει το Authorization header και προσπαθεί να αποκωδικοποιήσει το JWT token με χρήση του μυστικού κλειδιού. Αν το token είναι άκυρο, ληγμένο ή λείπει, αποστέλλεται μήνυμα σφάλματος 401. Αφού γίνει επιτυχής αυθεντικοποίηση, το backend επαληθεύει ότι τα πεδία mealType, enabled και date έχουν αποσταλεί σωστά μέσω του FormData. Αν λείπει κάποιο πεδίο, η διαδικασία διακόπτεται και επιστρέφεται μήνυμα αποτυχίας.

5. Ενημέρωση της Βάσης Δεδομένων

Αφού όλα τα προηγούμενα βήματα περάσουν με επιτυχία, το backend προχωρά στην αντιστοίχιση του τύπου γεύματος με την αντίστοιχη στήλη της βάσης δεδομένων: για breakfast αντιστοιχεί το enable_breakfast, για lunch το enable_lunch και για dinner το enable_dinner. Στη συνέχεια, δημιουργείται και εκτελείται SQL UPDATE ερώτημα στον πίνακα LetOrders για την επιλεγμένη ημερομηνία. Αν η ενημέρωση στη βάση εκτελεστεί επιτυχώς, επιστρέφεται απάντηση JSON τύπου { "success": true }. Αν υπάρξει σφάλμα (π.χ. αποτυχία σύνδεσης ή SQL exception), επιστρέφεται μήνυμα σφάλματος με περιγραφή.

6. Ανανέωση ή Επαναπροσπάθεια στο Frontend

Εφόσον η απόκριση του backend δείξει ότι υπάρχει πρόβλημα με το token (401 ή 403) και δεν έχει προηγηθεί ήδη προσπάθεια ανανέωσης, τότε καλείται η συνάρτηση refreshAccessToken(). Αν η ανανέωση πετύχει, η συνάρτηση updateMealStatus()

επανεκτελείται αυτόματα με το νέο token. Αν αποτύχει, ενεργοποιείται το `logout()` και ο χρήστης αποσυνδέεται αυτόματα από το σύστημα για λόγους ασφαλείας.

7. UI Feedback στον Χρήστη

Ανάλογα με το αποτέλεσμα της ενέργειας, το frontend ενημερώνει τον διαχειριστή. Σε περίπτωση επιτυχίας, εμφανίζεται μήνυμα επιβεβαίωσης στο επάνω μέρος της οθόνης (toast notification), που ενημερώνει για την επιτυχή ενεργοποίηση ή απενεργοποίηση του τύπου γεύματος. Αντίστοιχα, σε περίπτωση αποτυχίας, εμφανίζεται προειδοποιητικό μήνυμα με λεπτομέρειες σφάλματος και οδηγίες για επανάληψη της ενέργειας.

5.3.5 Πίνακας Τεκμηρίωσης Endpoints

Το παρόν κεφάλαιο παρουσιάζει τα βασικά HTTP endpoints που χρησιμοποιούνται στο backend της εφαρμογής τόσο από πλευράς διαχειριστή (Admin), όσο και από πλευράς χρήστη (User). Κάθε endpoint συνοδεύεται από πληροφορίες όπως η HTTP μέθοδος, τα απαιτούμενα πεδία, οι προϋποθέσεις αυθεντικοποίησης, καθώς και οι τυπικοί κωδικοί απόκρισης που μπορεί να προκύψουν κατά την κλήση του.

Endpoint	Μέθοδος	Περιγραφή	Πρόσβαση
getMealStatus.php	POST	Λήψη της κατάστασης (enabled/disabled) ενός γεύματος για μια συγκεκριμένη ημερομηνία.	Admin
getMenuList.php	POST	Λήψη της λίστας του μενού για συγκεκριμένη ημερομηνία, τύπο γεύματος και κατηγορία.	Admin
insertNewFoodItems.php	POST	Εισαγωγή ενός νέου φαγητού στη βάση δεδομένων.	Admin
resetChosenItem.php	DELETE	Επαναφορά της κατάστασης ενός επιλεγμένου στοιχείου μενού.	Admin
saveChosenItem.php	POST	Αποθήκευση της κατάστασης ενός φαγητού σε μια λίστα μενού.	Admin
updateFoodItemDetails.php	POST	Ενημέρωση των λεπτομερειών ενός φαγητού.	Admin
updateMealStatus.php	POST	Ενημέρωση της κατάστασης ενός γεύματος για μια συγκεκριμένη ημερομηνία.	Admin
updateMenuItemAvailability.php	PATCH	Αλλαγή της κατάστασης διαθεσιμότητας ενός φαγητού στο μενού.	Admin
fetchOrders.php	POST	Λήψη παραγγελιών με βάση την κατάσταση και την ημερομηνία τους.	Admin
updateOrderStatus.php	PATCH	Ενημέρωση της κατάστασης μιας παραγγελίας με βάση το ID της.	Admin

addDefaultTimeslot.php	POST	Προσθήκη μιας νέας προεπιλεγμένης χρονικής θυρίδας.	Admin
addVacationDates.php	POST	Προσθήκη νέων ημερομηνιών διακοπών.	Admin
deleteDefaultTimeslot.php	PATCH	Διαγραφή μιας προεπιλεγμένης χρονικής θυρίδας με το ID της.	Admin
deleteVacationDates.php	DELETE	Διαγραφή ημερομηνιών διακοπών με το ID τους.	Admin
fetchTimeslots.php	GET	Λήψη όλων των προεπιλεγμένων χρονικών θυρίδων.	Admin
fetchVacationsDates.php	GET	Λήψη όλων των ημερομηνιών διακοπών.	Admin
updateDefaultTimeslot.php	PATCH	Ενημέρωση των ωρών μιας προεπιλεγμένης χρονικής θυρίδας.	Admin
updateVacationDates.php	PATCH	Ενημέρωση των ημερομηνιών διακοπών.	Admin
getVoucherUserDetails.php	GET	Λήψη στοιχείων χρηστών που έχουν κουπόνια και δεν έχουν δωρεάν φαγητό.	Admin
insertNewVoucher.php	POST	Εισαγωγή ενός νέου κουπονιού για έναν συγκεκριμένο χρήστη.	Admin
resetVoucherAmount.php	POST	Επαναφορά του ποσού των κουπονιών ενός χρήστη στο 0.	Admin
checkIfHasAlreadyOrder.php	POST	Έλεγχος αν ο χρήστης έχει ήδη υποβάλει παραγγελία για ένα γεύμα.	Admin, User
displayTimeslotsAndPrice.php	POST	Εμφάνιση των διαθέσιμων χρονικών θυρίδων για ένα γεύμα και ημερομηνία.	Admin, User
pendingOrderStatus.php	POST	Ανάκτηση της τρέχουσας κατάστασης μιας παραγγελίας.	Admin, User
getdailyMenuList.php	POST	Λήψη των στοιχείων του ημερήσιου μενού.	Admin, User
placeOrder.php	POST	Υποβολή μιας παραγγελίας.	Admin, User
displayOrderHistory.php	GET	Λήψη του ιστορικού παραγγελιών του χρήστη.	Admin, User
displayVoucherHistory.php	GET	Λήψη του ιστορικού αγορών κουπονιών του χρήστη.	Admin, User
getVoucherAmount.php	GET	Λήψη του συνολικού ποσού κουπονιών του χρήστη.	Admin, User
login.php	POST	Endpoint για τη σύνδεση χρηστών.	Admin, User

Πίνακας 2: Λίστα API endpoints του συστήματος με τη μέθοδο, την περιγραφή λειτουργίας και τα δικαιώματα πρόσβασης.

5.4 Μηχανισμός Σύνδεσης στην εφαρμογή

5.4.1 Περιγραφή Λειτουργίας: Σύνδεση Χρήστη (Login)

Η διαδικασία σύνδεσης στην εφαρμογή αποτελεί το σημείο εισόδου για κάθε φοιτητή που επιθυμεί να αξιοποιήσει τις δυνατότητες του συστήματος. Η σύνδεση πραγματοποιείται μέσω του **ιδρυματικού λογαριασμού του φοιτητή**, χρησιμοποιώντας τον **Identity Provider (IDP) του Πολυτεχνείου**, εξασφαλίζοντας έτσι αυθεντικοποίηση με βάση τα επίσημα στοιχεία του ιδρύματος.

Μέσω μιας απλής και κατανοητής φόρμας εισόδου, ο χρήστης καλείται να εισάγει το email και τον προσωπικό του κωδικό πρόσβασης. Τα στοιχεία αυτά αποστέλλονται με ασφάλεια προς τον διακομιστή για επεξεργασία, μέσω ασύγχρονου POST αιτήματος.

Κατά την επιτυχή επαλήθευση των διαπιστευτηρίων από τον IDP, ο διακομιστής δημιουργεί ένα JSON Web Token (JWT), το οποίο περιέχει βασικές πληροφορίες του χρήστη, όπως το αναγνωριστικό του (user_id), τον ρόλο του στο σύστημα (user ή admin) και την ένδειξη για το αν δικαιούται δωρεάν σίτιση. Το token αυτό επιστρέφεται στο frontend και αποθηκεύεται τοπικά στον browser του χρήστη, επιτρέποντας την αναγνώριση και ταυτοποίηση του σε επόμενες αλληλεπιδράσεις με την εφαρμογή, χωρίς την ανάγκη συνεχούς επανασύνδεσης.

Σε περίπτωση αποτυχίας σύνδεσης – είτε λόγω λανθασμένου email είτε εσφαλμένου κωδικού – το σύστημα επιστρέφει κατάλληλο μήνυμα σφάλματος, ενημερώνοντας τον χρήστη χωρίς να του αποκαλύπτει ευαίσθητες πληροφορίες.

Η διαδικασία είναι πλήρως εναρμονισμένη με τις αρχές ασφαλούς επικοινωνίας. Η μεταφορά των credentials πραγματοποιείται μέσω HTTPS, η εισαγωγή τους φιλτράρεται ώστε να αποφευχθούν επιθέσεις τύπου XSS και η αυθεντικοποίηση παραμένει **stateless** μέσω του JWT. Επιπλέον, η ύπαρξη μηχανισμών CSRF προστατεύει από κακόβουλα εξωτερικά αιτήματα που επιχειρούν να εκτελέσουν ενέργειες εκ μέρους του χρήστη. Η λειτουργία login, εκτός από απλή στη χρήση, αποτελεί θεμέλιο της ασφάλειας του συστήματος και πυλώνα της προσωποποιημένης εμπειρίας που παρέχεται από την εφαρμογή.

5.4.1.1 Ανανεώσιμα Token και Προστασία Πρόσβασης: Refresh Token Functionality

Μία από τις πιο κρίσιμες βελτιώσεις στη λειτουργία αυθεντικοποίησης της εφαρμογής αφορά την ενσωμάτωση μηχανισμού ανανέωσης token, γνωστού ως **refresh token functionality**. Ο στόχος της ενότητας αυτής είναι η διατήρηση της ασφάλειας, χωρίς να απαιτείται από τον χρήστη να

επανεισάγει τα διαπιστευτήριά του ανά τακτά χρονικά διαστήματα.

Η αρχική διαδικασία σύνδεσης, όπως περιγράφηκε παραπάνω, οδηγεί στη δημιουργία ενός **access token (JWT)** περιορισμένης διάρκειας ζωής, συνήθως με ισχύ λίγων λεπτών έως μίας ώρας. Για λόγους ασφαλείας, το access token δεν διατηρείται μόνιμα στο frontend, ούτε επιτρέπεται να επεκτείνεται αυθαίρετα. Αντί αυτού, κατά την επιτυχή σύνδεση δημιουργείται **και δεύτερο token**, το **refresh token**, το οποίο αποθηκεύεται με ασφαλή τρόπο (π.χ. σε HTTP-only cookie ή sessionStorage) και χρησιμοποιείται αποκλειστικά για την έκδοση νέου access token όταν το προηγούμενο λήξει.

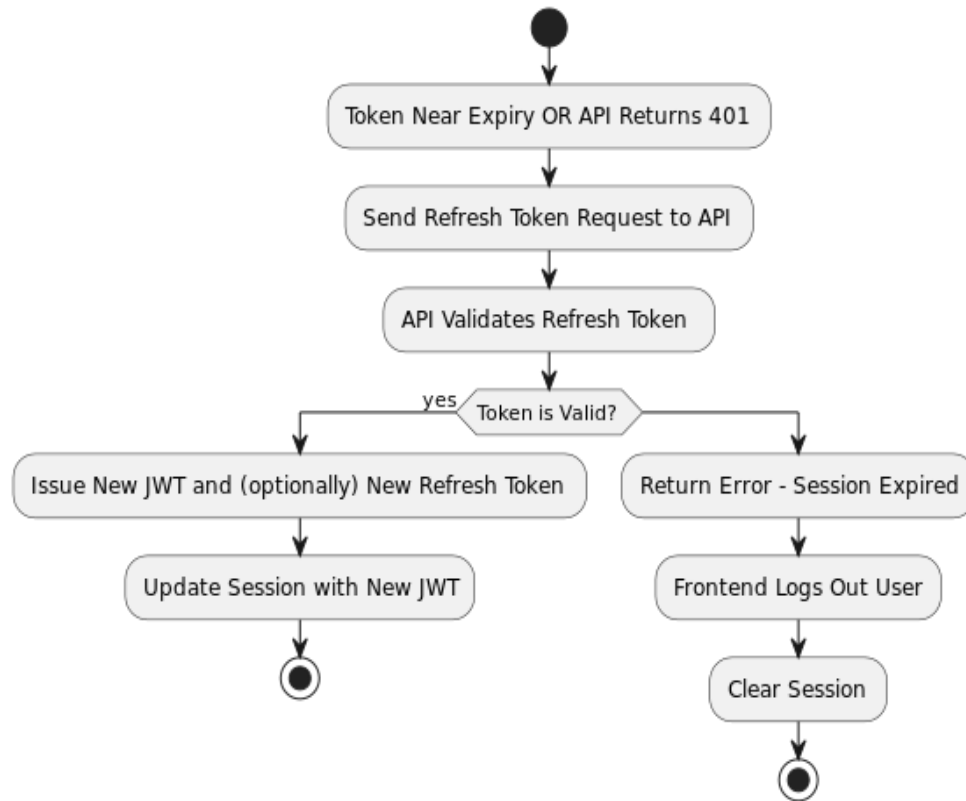
Όταν το frontend διαπιστώσει ότι ένα αίτημα προς το backend απέτυχε λόγω ληγμένου access token (HTTP 401), ενεργοποιείται αυτόματα η διαδικασία ανανέωσης. Μέσω ειδικής συνάρτησης (π.χ. refreshAccessToken()), αποστέλλεται αίτημα προς το endpoint refresh_token.php, συνοδευόμενο από το refresh token. Αν το refresh token επαληθευτεί με επιτυχία, το backend εκδίδει νέο access token, το οποίο αποθηκεύεται και χρησιμοποιείται για την εκ νέου αποστολή του αρχικού αιτήματος χωρίς να διακοπεί η εμπειρία χρήσης. Αν όμως το refresh token έχει λήξει ή είναι άκυρο, ο χρήστης αποσυνδέεται αυτόματα, για την αποφυγή κατάχρησης.

Αυτό το μηχανιστικό πλαίσιο αυξάνει σημαντικά την **ανθεκτικότητα του συστήματος σε επιθέσεις**, καθώς τα access tokens είναι βραχύβια και δύσκολα επαναχρησιμοποιήσιμα από κακόβουλους τρίτους. Παράλληλα, προσφέρει **αδιάλειπτη εμπειρία χρήσης**, χωρίς ανάγκη για επανασύνδεση κατά την πλοήγηση.

Για την πληρέστερη κατανόηση της λειτουργίας, παρατίθενται δύο UML διαγράμματα που οπτικοποιούν τη διαδικασία ανανέωσης του token:

1. Activity Diagram: Ροή Ανανέωσης Access Token

Το activity diagram απεικονίζει τη λογική ροή των ενεργειών που εκτελούνται αυτόματα από το frontend όταν διαπιστωθεί ότι το access token έχει λήξει κατά την αποστολή ενός αιτήματος. Η διαδικασία ξεκινά με την αποτυχία ενός API call (συνήθως λαμβάνεται απάντηση HTTP 401 Unauthorized) και συνεχίζεται με την ενεργοποίηση της συνάρτησης refreshAccessToken(), η οποία προσπαθεί να αποκτήσει νέο access token στέλνοντας το αποθηκευμένο refresh token στο backend. Αν το νέο token εκδοθεί με επιτυχία, το αρχικό αίτημα επαναλαμβάνεται. Αν όχι, ο χρήστης αποσυνδέεται αυτόματα για λόγους ασφαλείας. Το διάγραμμα αυτό τονίζει την αυτονομία του μηχανισμού, την αποφυγή χειροκίνητης επανασύνδεσης και τη σαφή διαχείριση λαθών.



Εικόνα 41: Activity Diagram που απεικονίζει τη ροή ανανέωσης του Access Token.

II. Sequence Diagram: Αλληλεπίδραση Κατά την Ανανέωση Token

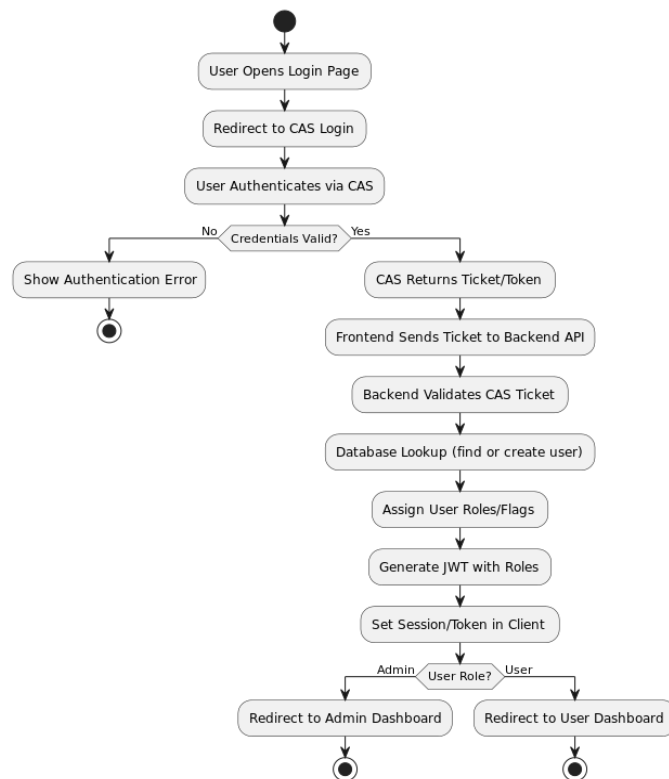
Το sequence diagram παρουσιάζει τη χρονική σειρά μηνυμάτων και ενεργειών μεταξύ των βασικών στοιχείων του συστήματος: του χρήστη, του frontend client (browser), του backend authentication controller, και της βάσης δεδομένων ή cache. Αρχικά, ο client εντοπίζει ότι το access token είναι ληγμένο και αποστέλλει αίτημα /refresh_token.php με το refresh token. Ο authentication controller ελέγχει την εγκυρότητα του refresh token και – αν είναι έγκυρο και δεν έχει λήξει – δημιουργεί νέο JWT access token και το επιστρέφει στο frontend. Ο client το αποθηκεύει και επαναπροωθεί το αρχικό αποτυχημένο αίτημα. Σε περίπτωση αποτυχίας επαλήθευσης του refresh token (π.χ. αν έχει λήξει ή έχει παραποιηθεί), ο controller στέλνει μήνυμα αποτυχίας και το frontend καλεί τη διαδικασία logout(), απομακρύνοντας τον χρήστη από την εφαρμογή.

Το διάγραμμα αυτό υπογραμμίζει τη δομημένη και ελεγχόμενη αλληλεπίδραση μεταξύ client–server σε θέματα αυθεντικοποίησης, διατηρώντας την ασφάλεια, τη συνέπεια, και την καλή εμπειρία χρήστη.



Εικόνα 42: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ συστατικών του συστήματος κατά την ανανέωση ενός Access Token.

5.4.2 Activity Diagram – Login



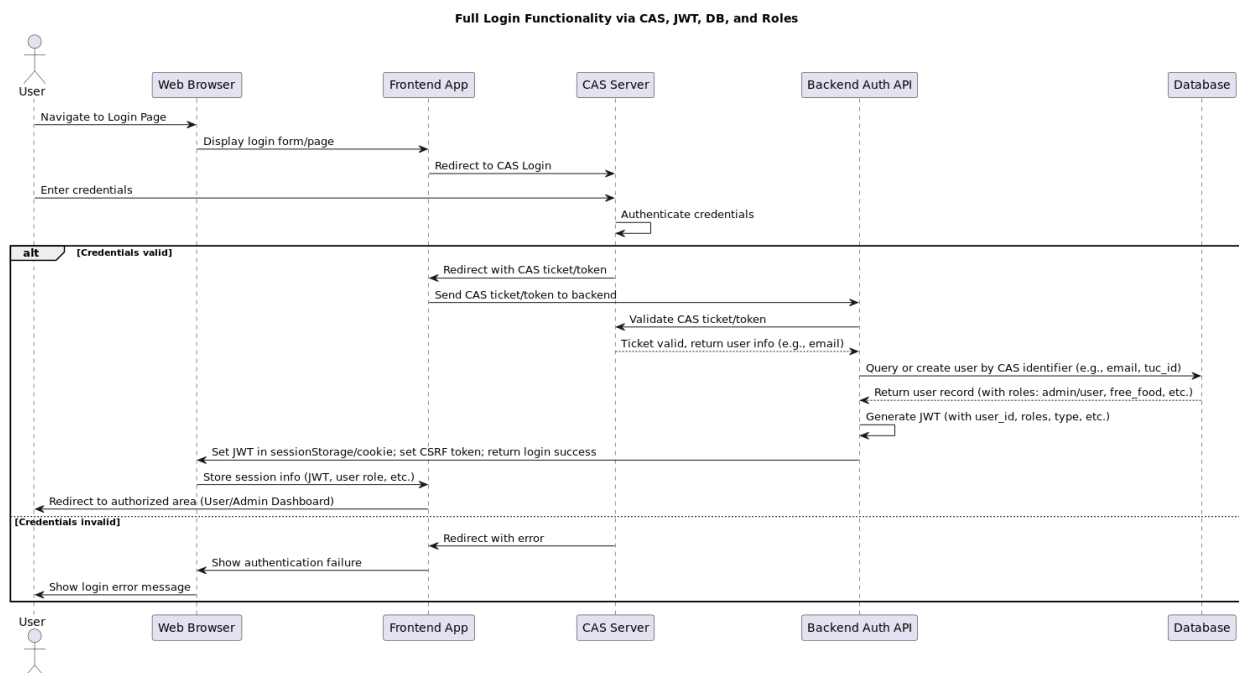
Εικόνα 43: Activity Diagram που παρουσιάζει τη ροή των ενεργειών κατά τη διαδικασία σύνδεσης στην εφαρμογή.

Το Activity Diagram που περιγράφει τη διαδικασία σύνδεσης ξεκινά με την ενέργεια του φοιτητή να εισάγει τα διαπιστευτήριά του στη φόρμα εισόδου. Καθώς ο χρήστης συμπληρώνει το email και τον κωδικό πρόσβασης, ενεργοποιείται η αποστολή ενός αιτήματος POST από το frontend προς το backend της εφαρμογής. Το αίτημα αυτό μεταφέρει τα δεδομένα με ασφαλή τρόπο προς τον server, ο οποίος ξεκινά τη διαδικασία επαλήθευσης.

Αν τα στοιχεία είναι ορθά, ο server δημιουργεί ένα JWT, το οποίο περιλαμβάνει πληροφορίες για την ταυτότητα του χρήστη και τον ρόλο του. Το token αυτό επιστρέφεται στο frontend, αποθηκεύεται τοπικά στον browser και επιτρέπει την πλοήγηση του χρήστη εντός της εφαρμογής με διατήρηση της ταυτότητάς του. Σε αυτό το σημείο, ο φοιτητής ανακατευθύνεται στην αρχική του σελίδα, όπου έχει πλέον πρόσβαση σε λειτουργίες όπως προβολή μενού, παραγγελία και ιστορικό συναλλαγών.

Σε περίπτωση αποτυχίας (π.χ. μη ύπαρξης χρήστη ή λάθος κωδικού), η εφαρμογή ενημερώνει τον χρήστη με κατάλληλο μήνυμα, χωρίς να αποκαλύπτει αν το λάθος εντοπίζεται στο email ή στον κωδικό – πρακτική που ενισχύει την ασφάλεια έναντι κακόβουλων χρηστών που επιχειρούν brute-force επιθέσεις. Η διαδικασία ολοκληρώνεται είτε με επιτυχή σύνδεση και αποθήκευση token είτε με εμφάνιση μηνύματος λάθους και επαναφορά στην αρχική φόρμα.

5.4.3 Sequence Diagram – Login



Εικόνα 44: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ χρηστών και συστατικών του συστήματος κατά τη διαδικασία σύνδεσης στην εφαρμογή.

Το διάγραμμα ακολουθίας (Sequence Diagram) παρουσιάζει με χρονική σειρά την αλληλεπίδραση μεταξύ των συνιστωσών του συστήματος κατά τη διάρκεια της σύνδεσης. Η ακολουθία ξεκινά από την ενέργεια του φοιτητή να υποβάλει τα στοιχεία του μέσω της φόρμας του frontend. Το frontend αναλαμβάνει να στείλει τα δεδομένα αυτά στο backend, χρησιμοποιώντας AJAX POST αίτημα προς το endpoint login.php.

Από την πλευρά του backend, η πρώτη ενέργεια είναι η αναζήτηση του χρήστη στη βάση δεδομένων βάσει του email. Αν ο χρήστης εντοπιστεί, ο server προχωρά στον έλεγχο του κωδικού, συγκρίνοντάς τον με τον αποθηκευμένο, ο οποίος έχει προηγουμένως κρυπτογραφηθεί (hashed). Εφόσον η σύγκριση αποδώσει θετικό αποτέλεσμα, δημιουργείται το JWT, το οποίο ενσωματώνει το user_id, τον ρόλο και άλλες πληροφορίες που επιτρέπουν την εξατομικευμένη εμπειρία.

Το backend επιστρέφει το token μαζί με βασικά στοιχεία του χρήστη σε μορφή JSON. Το frontend αποθηκεύει το token και μεταβαίνει στο κεντρικό περιβάλλον χρήσης (dashboard), προσαρμοσμένο ανάλογα με τον ρόλο (φοιτητής ή διαχειριστής). Σε αντίθετη περίπτωση, επιστρέφεται μήνυμα αποτυχίας, το οποίο εμφανίζεται στον χρήστη χωρίς ανανέωση της σελίδας.

5.4.4 Ανάλυση Επικοινωνίας Frontend – Backend

Η επικοινωνία μεταξύ του frontend και του backend κατά τη διαδικασία login υλοποιείται με τη χρήση AJAX POST αιτήματος. Το frontend συγκεντρώνει τα στοιχεία εισόδου του χρήστη – email και κωδικό – και τα αποστέλλει με ασφαλή τρόπο στο backend, στη διεύθυνση /login.php. Το αίτημα αυτό περιέχει ένα JSON αντικείμενο που περιλαμβάνει τα διαπιστευτήρια του χρήστη.

Ο backend χειρίζεται το αίτημα εξετάζοντας αρχικά την ύπαρξη του email στη βάση δεδομένων. Αν το email αντιστοιχεί σε κάποιον χρήστη, ακολουθεί η σύγκριση του εισαχθέντος κωδικού με τον αποθηκευμένο, με χρήση hash και salt, διασφαλίζοντας ότι η επεξεργασία των κωδικών πραγματοποιείται με ασφαλή και μη αναστρέψιμο τρόπο. Αν η διαδικασία είναι επιτυχής, δημιουργείται ένα JWT, το οποίο υπογράφεται με μυστικό κλειδί του server, και επιστρέφεται στο frontend μαζί με πληροφορίες για το προφίλ του χρήστη (όνομα, ρόλος, δικαίωμα σίτισης).

Το JWT αποθηκεύεται από το frontend, π.χ. στο sessionStorage ή localStorage του browser. Από εκεί και πέρα, κάθε επόμενο αίτημα του χρήστη προς το backend (π.χ. για προβολή μενού ή υποβολή παραγγελίας) συνοδεύεται από το token αυτό στον HTTP header Authorization: Bearer <token>, επιτρέποντας στο backend να ταυτοποιεί άμεσα τον χρήστη χωρίς νέα αυθεντικοποίηση. Η σύνδεση είναι έτσι stateless, ελαφριά, και ασφαλής.

Σε περίπτωση αποτυχίας, το backend επιστρέφει JSON απόκριση με το πεδίο success: false και αντίστοιχο μήνυμα λάθους, το οποίο εμφανίζεται στην οθόνη χωρίς επαναφόρτωση της σελίδας, ενισχύοντας τη διαδραστικότητα και την εμπειρία χρήστη.

5.5 Λειτουργικότητες για Φοιτητές

Παρακάτω παρουσιάζονται οι λειτουργίες που μπορούν να εκτελέσουν οι χρήστες μετά την επιτυχή είσοδο τους στο σύστημα.

5.5.1 Υποβολή Παραγγελίας

Η διαδικασία υποβολής παραγγελίας αποτελεί τον βασικό πυρήνα της λειτουργικότητας της εφαρμογής για τον φοιτητή. Μέσα από ένα διαδραστικό και ευέλικτο περιβάλλον χρήστη, ο φοιτητής έχει τη δυνατότητα να σχεδιάσει το γεύμα του σύμφωνα με τις προτιμήσεις και το καθημερινό του πρόγραμμα. Η εφαρμογή καλείται να εξυπηρετήσει αυτή τη διαδικασία με απόλυτη ακρίβεια, ασφάλεια και διαφάνεια.

Αρχικά, ο χρήστης περιηγείται στο μενού της ημέρας, το οποίο είναι οργανωμένο ανά κατηγορία (πρωινό, μεσημεριανό, βραδινό) και διαχωρίζεται περαιτέρω σε τύπους φαγητού (κυρίως, συνοδευτικά, σαλάτες κ.ά.). Ο φοιτητής επιλέγει τα φαγητά που επιθυμεί να καταναλώσει, προσθέτοντάς τα στο καλάθι του. Στη συνέχεια, δηλώνει τον επιθυμητό τρόπο εξυπηρέτησης, επιλέγοντας είτε κατανάλωση στο εστιατόριο της λέσχης, είτε παραλαβή σε πακέτο. Ακολουθεί η επιλογή του χρονικού διαστήματος (timeslot) στο οποίο θέλει να εξυπηρετηθεί, με βάση τη διαθεσιμότητα που απεικονίζεται δυναμικά ανά timeslot.

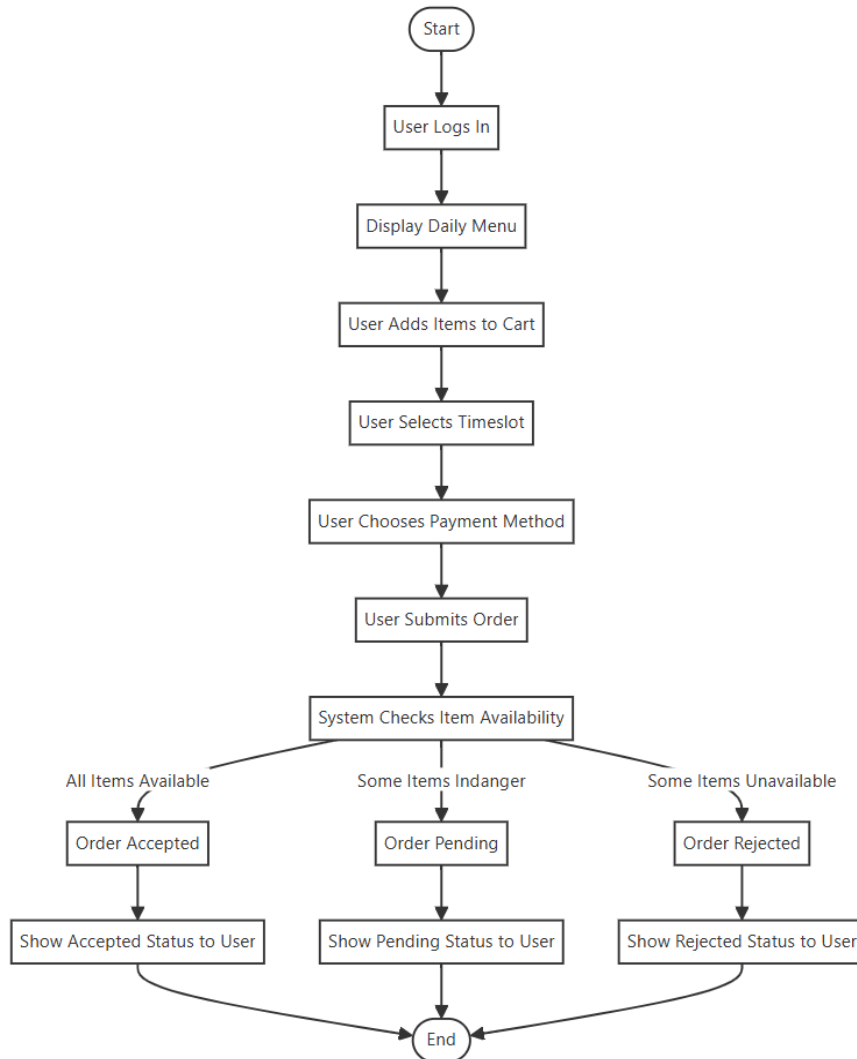
Ένα κρίσιμο βήμα αφορά τον τρόπο πληρωμής. Εάν ο χρήστης **δεν έχει δικαίωμα δωρεάν σίτισης**, ή έχει ήδη καταναλώσει το ημερήσιο δικαίωμα για το συγκεκριμένο τύπο γεύματος, καλείται να πληρώσει το γεύμα του μέσω συστήματος **voucher**. Το υπόλοιπο των vouchers προβάλλεται κατά τη διαδικασία, και αν επαρκεί, η συναλλαγή ολοκληρώνεται χωρίς περαιτέρω επιβάρυνση. Η επιλογή voucher λειτουργεί και ως μηχανισμός αποτροπής διπλής παραγγελίας για τύπους γεύματος που ήδη καταναλώθηκαν.

Η τελική υποβολή ενεργοποιεί αίτημα POST προς το backend, συνοδευόμενο από όλα τα απαραίτητα στοιχεία: το περιεχόμενο του καλαθιού, τον τρόπο εξυπηρέτησης, το timeslot και τον τρόπο πληρωμής. Το backend εκτελεί σειρά ελέγχων: εγκυρότητα token αυθεντικοποίησης, επαλήθευση voucher υπολοίπου, διασταύρωση για διπλές παραγγελίες και διαθεσιμότητα των ειδών με βάση το μενού της ημέρας. Αν όλα τα βήματα περατωθούν επιτυχώς, καταχωρείται η παραγγελία στη βάση δεδομένων, αφαιρείται το αντίστοιχο ποσό από τα vouchers (όπου ισχύει), και ο χρήστης λαμβάνει μήνυμα επιβεβαίωσης στην οθόνη του.

Η διαδικασία έχει σχεδιαστεί έτσι ώστε να είναι φιλική, αλλά και πλήρως **προστατευμένη από κακόβουλη χρήση**, αποτρέποντας φαινόμενα κατάχρησης, υπερκράτησης ή ψευδών παραγγελιών.

I. Activity Diagram – Υποβολή Παραγγελίας από Χρήστη

Το activity diagram της υποβολής παραγγελίας περιγράφει με σαφήνεια όλα τα βήματα που ακολουθεί ένας χρήστης από την αρχική επιλογή φαγητού έως την οριστική επιβεβαίωση. Η διαδικασία ξεκινά με την περιήγηση στο μενού και την προσθήκη επιθυμητών φαγητών στο καλάθι. Ακολουθεί η επιλογή τρόπου εξυπηρέτησης (εντός της λέσχης ή πακέτο) και η δήλωση επιθυμητού timeslot. Αν ο χρήστης διαθέτει δωρεάν σίτιση για τον συγκεκριμένο τύπο γεύματος, προχωρά χωρίς χρέωση. Αν όχι – ή αν έχει ήδη παραγγείλει – η εφαρμογή τον ενημερώνει για την ανάγκη πληρωμής μέσω voucher. Εφόσον υπάρχει διαθέσιμο υπόλοιπο, η διαδικασία συνεχίζεται. Τέλος, το σύστημα ζητά επιβεβαίωση και αποστέλλει την παραγγελία στο backend. Το διάγραμμα απεικονίζει καθαρά και τις περιπτώσεις αποτυχίας: αν δεν υπάρχουν διαθέσιμα vouchers, αν το timeslot είναι πλήρες, ή αν το backend απορρίπτει την παραγγελία. Σε αυτές τις περιπτώσεις, το σύστημα ενημερώνει τον χρήστη κατάλληλα και του επιτρέπει να επαναπροσπαθήσει.



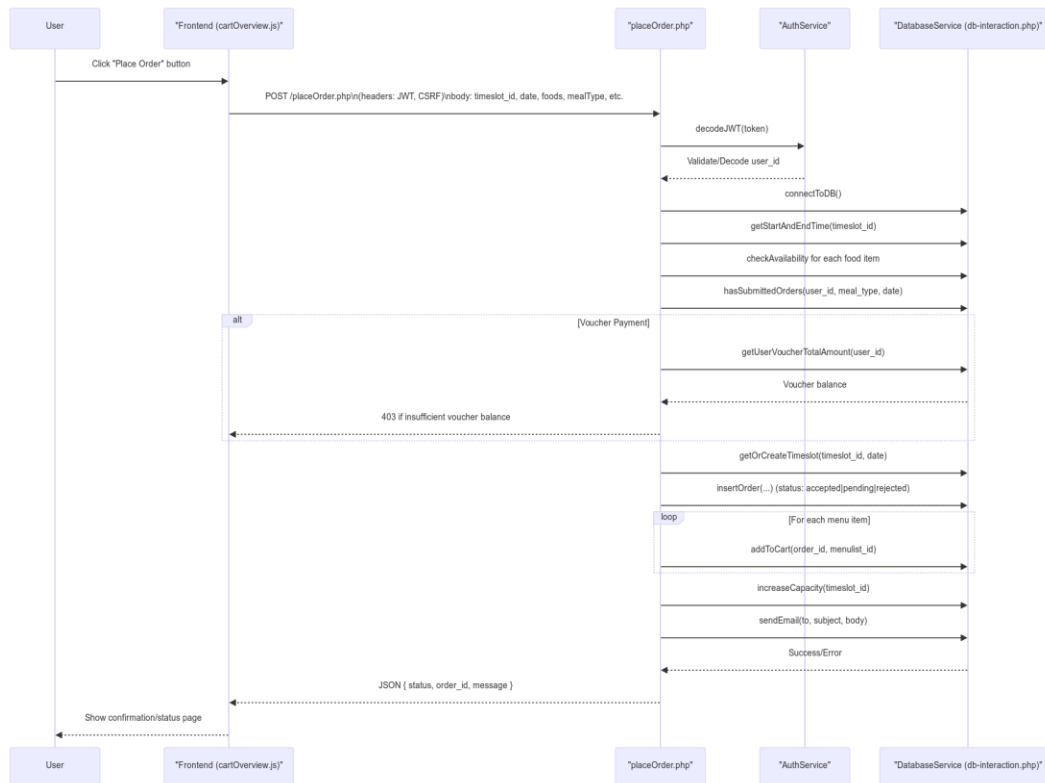
Εικόνα 45: Activity Diagram που παρουσιάζει τη ροή ενεργειών κατά την υποβολή παραγγελίας από τον χρήστη.

II. Sequence Diagram – Υποβολή Παραγγελίας (με χρήση voucher)

Το sequence diagram της λειτουργίας (αρχείο: *sequence_diagram_user_order.svg*) παρουσιάζει την **χρονική ακολουθία ενεργειών** μεταξύ χρήστη, frontend και backend κατά την παραγγελία. Αν και το αρχικό διάγραμμα παραλείπει την επιλογή φαγητού, εξυπηρέτησης, timeslot και τρόπου πληρωμής, η περιγραφή καλύπτει και αυτά τα βήματα. Ο χρήστης αλληλεπιδρά με τη διεπαφή για να διαμορφώσει το καλάθι του, να επιλέξει τρόπο παραλαβής και επιθυμητή ώρα, καθώς και να ενεργοποιήσει την πληρωμή μέσω voucher. Όταν ολοκληρωθούν οι επιλογές, αποστέλλεται αίτημα POST στο endpoint `placeOrder.php`, με πλήρη στοιχεία παραγγελίας και απαραίτητα headers ασφαλείας (JWT & CSRF tokens).

Το backend ελέγχει τη γνησιότητα του χρήστη μέσω του JWT, επαληθεύει ότι δεν έχει υπάρξει ήδη παραγγελία του ίδιου τύπου για την ίδια ημέρα και εξετάζει αν το υπόλοιπο voucher καλύπτει το κόστος. Αν όλα τα κριτήρια ικανοποιούνται, δημιουργείται νέα εγγραφή παραγγελίας στη βάση, ενημερώνεται το υπόλοιπο voucher και επιστρέφεται JSON μήνυμα επιτυχίας προς τον client. Η διεπαφή ενημερώνει τον χρήστη με οπτικό feedback επιβεβαίωσης.

Η σειρά ενεργειών και οι έλεγχοι που περιλαμβάνονται διασφαλίζουν την **ακεραιότητα της διαδικασίας** και την **συνεπή ενημέρωση των δεδομένων**, αποτρέποντας φαινόμενα διπλοεγγραφών ή κατάχρησης vouchers.



Εικόνα 46: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ χρηστών και συστατικών του συστήματος κατά την υποβολή παραγγελίας με χρήση voucher.

5.5.2 Προβολή Ιστορικού Παραγγελιών και Συναλλαγών Voucher

Εκτός από τη δυνατότητα υποβολής νέων παραγγελιών, η εφαρμογή παρέχει στους χρήστες πρόσβαση στο προσωπικό τους ιστορικό συναλλαγών, μέσω του προφίλ τους. Η λειτουργία αυτή ενισχύει τη διαφάνεια, επιτρέπει την αναδρομή σε προηγούμενες επιλογές, και προσφέρει στον φοιτητή έναν μηχανισμό αναστοχασμού για τις διατροφικές του συνήθειες και τη χρήση των διαθέσιμων πόρων (όπως τα vouchers). Και οι δύο λειτουργίες εμφανίζονται στην ίδια σελίδα — στην προβολή **User Profile** — οργανωμένες σε ξεχωριστές καρτέλες κάτω από τις βασικές πληροφορίες του χρήστη.

Η πρώτη λειτουργία αφορά όλους τους φοιτητές, ενώ η δεύτερη ενεργοποιείται μόνο όταν ο χρήστης **δεν έχει δικαίωμα δωρεάν σίτισης**, οπότε και εμφανίζεται το σχετικό section «Ιστορικό Χρήσης Voucher»

5.5.2.1 Ιστορικό Παραγγελιών

Ο χρήστης μπορεί να μεταβεί στο προφίλ του μέσω της αρχικής μπάρας πλοήγησης, όπου εμφανίζεται επιλογή «Προφίλ» ή εικονίδιο χρήστη. Στο κάτω μέρος της σελίδας προφίλ, εμφανίζεται η ενότητα «Ιστορικό Παραγγελιών», η οποία περιλαμβάνει χρονολογικά όλες τις προηγούμενες παραγγελίες του φοιτητή, μαζί με πληροφορίες όπως η ημερομηνία, το είδος γεύματος, η περιγραφή των φαγητών, το συνολικό κόστος και η κατάσταση (π.χ. ολοκληρωμένη, ακυρωμένη, εκκρεμής).

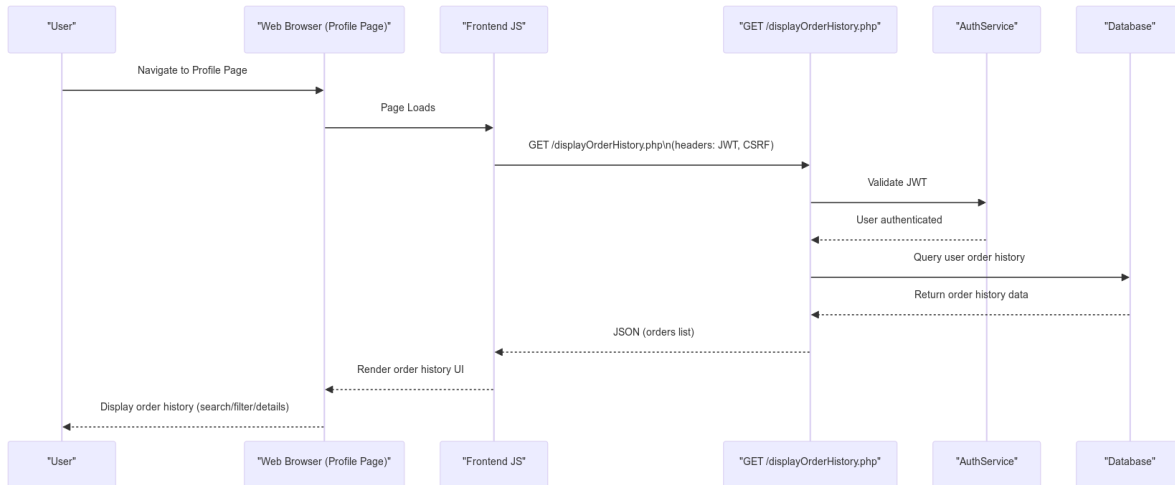
Η προβολή υλοποιείται με χρήση AJAX κλήσης προς το backend endpoint `displayOrderHistory.php`, αμέσως μόλις φορτωθεί η σελίδα προφίλ. Το αίτημα περιλαμβάνει το JWT token του χρήστη, βάσει του οποίου γίνεται επαλήθευση και εξαγωγή του `user_id`. Το backend αναζητά τις αντίστοιχες εγγραφές στον πίνακα `Orders`, τις εμπλουτίζει με τις λεπτομέρειες των φαγητών από το `Cart` και το `MenuList`, και επιστρέφει ένα δομημένο JSON που απεικονίζεται ως διαδραστικός πίνακας στο frontend.

- **Sequence Diagram – Προβολή Ιστορικού Παραγγελιών**

Το διάγραμμα ακολουθίας (αρχείο: `sequence_diagram_user_order_history.svg`) απεικονίζει τη χρονική ροή ενεργειών κατά την ανάκτηση του ιστορικού παραγγελιών. Ο χρήστης ανοίγει τη σελίδα προφίλ του, με αποτέλεσμα το frontend να στείλει GET ή AJAX POST αίτημα προς το endpoint `displayOrderHistory.php`, συνοδευόμενο από το JWT για αυθεντικοποίηση.

Ο server επαληθεύει το token, εξάγει το `user_id`, και εκτελεί ερώτημα στη βάση δεδομένων, προκειμένου να ανακτήσει όλες τις παραγγελίες που ανήκουν στον χρήστη αυτό. Επιπλέον, για κάθε παραγγελία αναζητούνται τα στοιχεία των φαγητών, των τιμών

και του timeslot. Η πληροφορία μορφοποιείται σε JSON και επιστρέφεται στον client, ο οποίος ενημερώνει το UI δυναμικά, χωρίς ανανέωση σελίδας. Έτσι, ο χρήστης μπορεί να δει τις παραγγελίες του οργανωμένες ανά ημέρα, με δυνατότητα φιλτραρίσματος ή ταξινόμησης.



Εικόνα 47: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ χρηστών και συστατικών του συστήματος κατά την προβολή ιστορικού παραγγελιών.

5.5.2.2 Ιστορικό Συναλλαγών Voucher

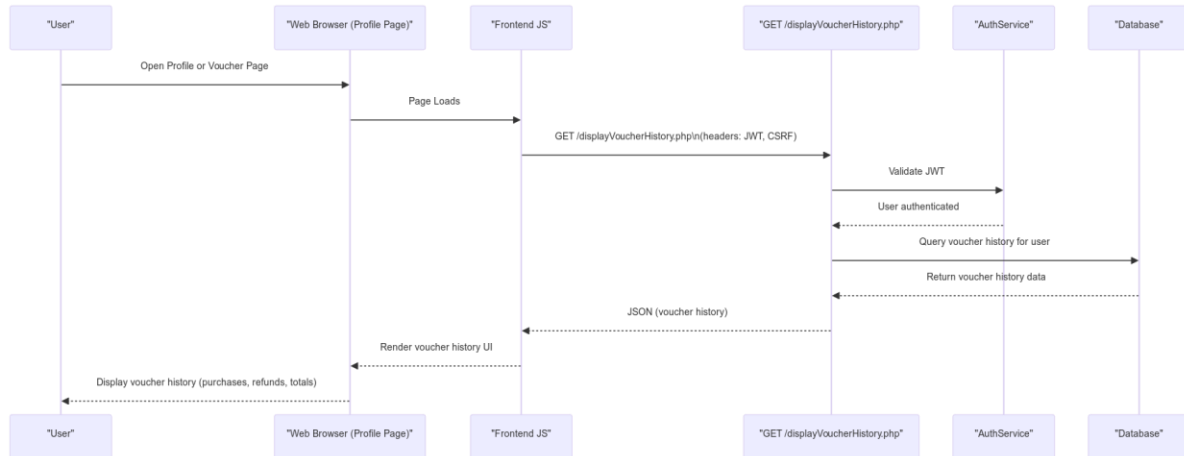
Για τους φοιτητές που **δεν έχουν δικαίωμα δωρεάν σίτισης**, η εφαρμογή προβλέπει ένα δεύτερο τμήμα στο προφίλ χρήστη, με τίτλο «Ιστορικό Συναλλαγών Voucher». Μέσα από αυτό το τμήμα, ο χρήστης μπορεί να δει κάθε φορά που χρησιμοποίησε το σύστημα voucher για να πραγματοποιήσει αγορά. Οι πληροφορίες που εμφανίζονται περιλαμβάνουν την ημερομηνία, την ώρα, το ποσό της συναλλαγής και το διαθέσιμο υπόλοιπο μετά την κάθε αγορά.

Η φόρτωση του ιστορικού πραγματοποιείται με AJAX αίτημα προς το endpoint `displayVoucherHistory.php`, το οποίο λειτουργεί με παρόμοιο τρόπο όπως το endpoint για τις παραγγελίες. Εφόσον το `free_food` flag του χρήστη είναι `false`, το section καθίσταται ορατό και ενεργοποιείται η κλήση στο backend.

- **Sequence Diagram – Προβολή Ιστορικού Voucher**

Το sequence diagram (αρχείο: `sequence_diagram_user_voucher_history.svg`) απεικονίζει τον πλήρη κύκλο αιτήματος και απόκρισης για την ανάκτηση του ιστορικού voucher. Όταν ο μη δικαιούχος δωρεάν σίτισης φοιτητής φορτώσει τη σελίδα προφίλ, το frontend εκτελεί AJAX αίτημα προς το `displayVoucherHistory.php`, περνώντας το JWT token.

Ο server, μετά την επιβεβαίωση της αυθεντικοποίησης, εντοπίζει το user_id και ανακτά από τον πίνακα Vouchers όλες τις σχετικές εγγραφές του χρήστη, ταξινομημένες κατά ημερομηνία και ώρα. Το JSON που επιστρέφεται περιέχει τις πληροφορίες των συναλλαγών και εμφανίζεται στον χρήστη ως λίστα ή πίνακας. Η λειτουργία αυτή συμβάλλει καθοριστικά στη διαφάνεια και στον προσωπικό έλεγχο δαπανών, παρέχοντας στον φοιτητή την αίσθηση οικονομικής επίγνωσης και ελέγχου.



Εικόνα 48: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ χρηστών και συστατικών του συστήματος κατά την προβολή ιστορικού voucher.

5.6 Λειτουργικότητες για Διαχειριστές

Η εφαρμογή υποστηρίζει, εκτός από το ρόλο του φοιτητή (χρήστη), και το ρόλο του διαχειριστή (admin), ο οποίος διαθέτει διευρυμένα δικαιώματα πρόσβασης και διαχείρισης. Η αυθεντικοποίηση των διαχειριστών γίνεται μέσω της ίδιας διεπαφής σύνδεσης που χρησιμοποιείται και για τους χρήστες. Ωστόσο, κατά τη διαδικασία έκδοσης του JWT token, εάν το type πεδίο του χρήστη είναι admin, τότε του αποδίδονται επιπλέον δικαιώματα και ενεργοποιούνται οι σχετικές λειτουργικότητες του πίνακα ελέγχου διαχειριστή.

Είναι σημαντικό να σημειωθεί ότι ο ρόλος του διαχειριστή **κληρονομεί πλήρως όλες τις λειτουργίες του χρήστη**, επιτρέποντάς του να υποβάλλει παραγγελίες, να βλέπει μενού, ιστορικό, κ.ά. Παράλληλα, αποκτά πρόσβαση σε προηγμένες λειτουργίες που του επιτρέπουν να ελέγχει, να παραμετροποιεί και να εποπτεύει το σύνολο της λειτουργίας της λέσχης. Οι λειτουργίες αυτές έχουν σχεδιαστεί με γνώμονα την ευκολία στη χρήση, την ταχύτητα και την ασφάλεια, και αποτελούν τον πυρήνα της επιχειρησιακής ευφυΐας του συστήματος.

Αναλυτικότερα, ο διαχειριστής έχει τη δυνατότητα να δηλώνει το ημερήσιο μενού, να τροποποιεί

την κεντρική βάση φαγητών, να διαχειρίζεται την ενεργοποίηση και απενεργοποίηση γευμάτων (meal status) ανά ημέρα και τύπο, να εποπτεύει τις παραγγελίες που καταφθάνουν, να διαχειρίζεται vouchers χρηστών, να καθορίζει και να τροποποιεί τα χρονικά διαστήματα εξυπηρέτησης (timeslots), να εισάγει περιόδους διακοπών, αλλά και να αξιοποιεί τον ενσωματωμένο μηχανισμό πρόβλεψης μερίδων.

5.6.1 Δήλωση Μενού Ημέρας

Μία από τις βασικότερες και πιο συχνές ενέργειες που καλείται να εκτελέσει ο διαχειριστής της φοιτητικής λέσχης είναι η **δήλωση του μενού για κάθε ημέρα**. Η ενέργεια αυτή αποτελεί τον αφετηριακό κρίκο της καθημερινής ροής λειτουργίας, καθώς το μενού καθορίζει τι θα εμφανιστεί στους χρήστες, ποια φαγητά θα είναι διαθέσιμα προς παραγγελία, και επηρεάζει άμεσα την κατανομή πόρων και προσωπικού.

Η λειτουργία αυτή έχει σχεδιαστεί έτσι ώστε να είναι **απλή, άμεση και ευέλικτη**, παρέχοντας στον διαχειριστή τα κατάλληλα εργαλεία για να ορίσει το μενού ανά ημερομηνία και τύπο γεύματος (πρωινό, μεσημεριανό, βραδινό). Η προσθήκη των φαγητών πραγματοποιείται μέσα από ένα διαδραστικό περιβάλλον που επιτρέπει **αναζήτηση μέσω φίλτρων ή ελεύθερου κειμένου**, αλλά και **drag-and-drop** μεταφορά φαγητών από το γενικό μενού στο ημερήσιο. Αυτή η σχεδιαστική επιλογή καθιστά τη διαδικασία γρήγορη και φιλική, ακόμη και για μη τεχνικούς χρήστες.

Ένα από τα πιο κρίσιμα σημεία στη ροή αυτής της λειτουργίας είναι η **δυνατότητα ενσωμάτωσης προβλέψεων**. Ο διαχειριστής, κατά τη δήλωση του μενού, έχει πρόσβαση στις προτάσεις του **μηχανισμού πρόβλεψης μερίδων**, που έχει αναπτυχθεί και περιγράφεται αναλυτικά στο Κεφάλαιο 6. Ο μηχανισμός αυτός χρησιμοποιεί ιστορικά δεδομένα, τύπους ημέρας και καιρικές συνθήκες για να προβλέψει τη ζήτηση σε συγκεκριμένα είδη φαγητού. Οι προβλέψεις αυτές εμφανίζονται εντός της ίδιας διεπαφής, δίπλα σε κάθε φαγητό, βοηθώντας τον διαχειριστή να λάβει **τεκμηριωμένες αποφάσεις** ως προς το ποια φαγητά θα περιληφθούν στο μενού της ημέρας.

Από τεχνικής άποψης, το frontend ανακτά πρώτα το γενικό μενού των διαθέσιμων φαγητών μέσω κλήσης στο endpoint `fetchAllFoodItems.php`. Ο διαχειριστής αναζητά ή φιλτράρει φαγητά και τα σύρει στο section του ημερήσιου μενού. Κάθε φορά που προστίθεται ένα νέο φαγητό, δημιουργείται προσωρινή εγγραφή στο frontend. Μόλις ο χρήστης επιβεβαιώσει την τελική σύνθεση, το σύστημα αποστέλλει τη δήλωση προς το backend μέσω του endpoint `saveChosenItem.php`, μεταφέροντας τον `item_id`, την ημερομηνία, τον τύπο γεύματος και την κατηγορία φαγητού (π.χ. κυρίως, σαλάτα, συνοδευτικό).

Το backend, με τη σειρά του, **ελέγχει τη μοναδικότητα** (ώστε να μην καταχωρηθεί το ίδιο φαγητό δύο φορές για την ίδια ημέρα), **επικυρώνει το JWT του χρήστη** και αποθηκεύει την εγγραφή

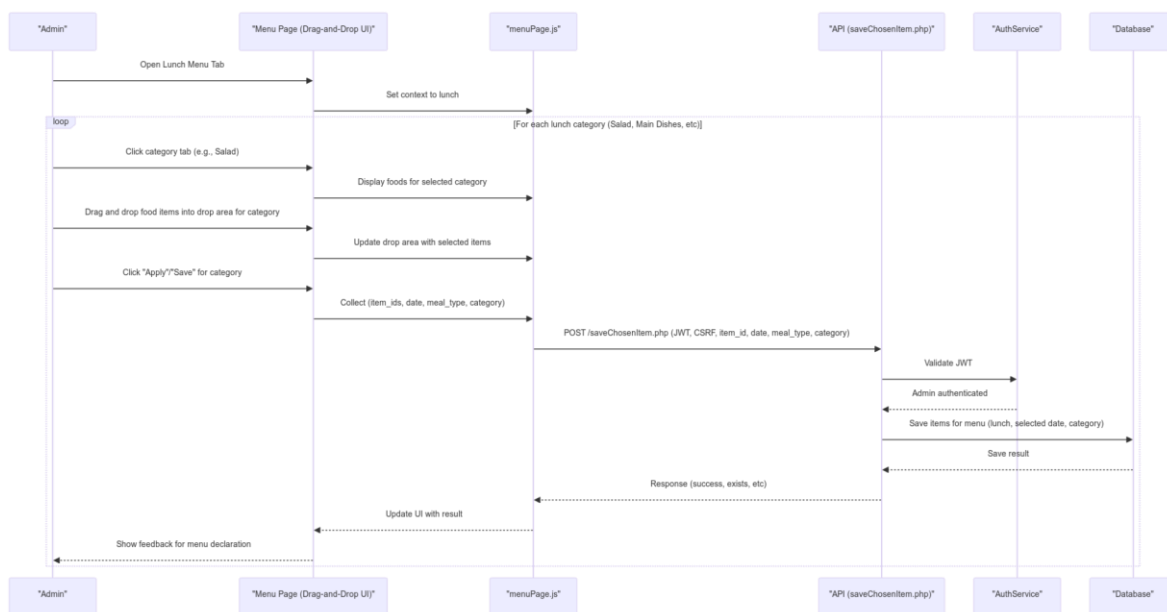
στον πίνακα MenuList. Η διεπαφή ενημερώνεται άμεσα για την επιτυχία ή αποτυχία κάθε εγγραφής, ενώ παρέχεται και δυνατότητα "reset" ή διαγραφής μεμονωμένων φαγητών μέσω του endpoint resetChosenItem.php.

- **Sequence Diagram – Δήλωση Μενού Ημέρας**

Το sequence diagram απεικονίζει τη χρονική αλληλουχία ενεργειών που πραγματοποιούνται όταν ο διαχειριστής δηλώνει το μενού μιας ημέρας. Η διαδικασία ξεκινά με την προβολή του διαχειριστικού dashboard, κατά την οποία το frontend ανακτά το σύνολο των διαθέσιμων φαγητών μέσω fetchAllFoodItems.php.

Ο διαχειριστής αναζητά φαγητά μέσω φίλτρου ή πληκτρολόγησης και τα προσθέτει στο ημερήσιο μενού με drag-and-drop. Η προσθήκη κάθε φαγητού ενεργοποιεί POST αίτημα στο saveChosenItem.php, το οποίο αποστέλλει πληροφορίες όπως item_id, date, meal_type, category. Το backend ελέγχει το JWT token, διασφαλίζει την εγκυρότητα των δεδομένων και δημιουργεί την εγγραφή στο MenuList.

Σε περίπτωση ανάγκης επαναφοράς, ο διαχειριστής μπορεί να χρησιμοποιήσει τη λειτουργία resetChosenItem.php, η οποία διαγράφει την αντίστοιχη εγγραφή από τη βάση. Όλες οι απαντήσεις παρέχονται σε μορφή JSON και απεικονίζονται στο UI ως επιβεβαιώσεις επιτυχίας ή αποτυχίας. Η ροή ενισχύεται και από την εμφάνιση **προτεινόμενων φαγητών με βάση τις προβλέψεις**, οι οποίες εμφανίζονται μέσα στο UI δίπλα από κάθε φαγητό, λειτουργώντας υποστηρικτικά στην τελική απόφαση.



Εικόνα 49: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ διαχειριστή και συστατικών του συστήματος κατά τη δήλωση του μενού της ημέρας.

5.6.2 Προσθήκη, Τροποποίηση και Διαγραφή Ειδών Φαγητού από τη Βάση

Η δεύτερη βασική αρμοδιότητα του διαχειριστή αφορά τη **διαχείριση της κεντρικής βάσης φαγητών**, από την οποία αντλούνται οι επιλογές που θα δηλώνονται στα ημερήσια μενού. Η βάση αυτή λειτουργεί ως αποθετήριο όλων των διαθέσιμων φαγητών που μπορεί να προσφερθούν στη λέσχη, είτε είναι ενεργά είτε έχουν τεθεί εκτός δημοσίευσης για ιστορικούς λόγους.

Η εφαρμογή προσφέρει στον διαχειριστή τη δυνατότητα:

- Να **εισάγει νέα φαγητά** προσθέτοντας πλήρη πληροφοριακά δεδομένα,
- Να **τροποποιεί υπάρχουσες εγγραφές** (π.χ. τίτλο, περιγραφή, εικόνα, κατηγορίες),
- Να **απενεργοποιεί/αποσύρει φαγητά** από τη βάση, χωρίς όμως να τα διαγράφει οριστικά.

Εισαγωγή Νέου Φαγητού

Κατά την εισαγωγή νέου είδους φαγητού, ο διαχειριστής καλείται να συμπληρώσει πλήρη φόρμα με όλα τα απαραίτητα πεδία: τίτλος φαγητού, περιγραφή, εικόνα (σε binary ή URL μορφή), και οι κατηγορίες στις οποίες ανήκει. Οι κατηγορίες αυτές μπορεί να είναι πολλαπλές και πολυδιάστατες (π.χ. isLunchMainDishes, isFitness, isCretan), και βοηθούν τόσο στο φιλτράρισμα από το frontend όσο και στην πρόβλεψη ζήτησης.

Κατά την υποβολή, αποστέλλεται AJAX POST αίτημα προς το endpoint insertNewFoodItems.php, με όλα τα παραπάνω δεδομένα σε FormData. Το backend επαληθεύει το token αυθεντικοποίησης, καθαρίζει τα εισερχόμενα δεδομένα, και δημιουργεί νέα εγγραφή στον πίνακα FoodItems, ενώ προστίθενται οι τιμές κατηγορίας στον πίνακα FoodItemCategory.

Τροποποίηση Φαγητού

Ο διαχειριστής μπορεί οποιαδήποτε στιγμή να μεταβεί στη λίστα με τα υπάρχοντα φαγητά και να επιλέξει την επεξεργασία κάποιου εξ αυτών. Η φόρμα που εμφανίζεται είναι ίδια με εκείνη της εισαγωγής, με προσυμπληρωμένα όλα τα πεδία. Οι αλλαγές αποστέλλονται με POST αίτημα στο updateFoodItemDetails.php, και το backend ενημερώνει τα πεδία στους σχετικούς πίνακες με χρήση SQL UPDATE. Όλες οι αλλαγές καταγράφονται, διατηρώντας έτσι πλήρες ιστορικό τροποποιήσεων.

Απενεργοποίηση Αντί Διαγραφής

Η διαγραφή ενός φαγητού δεν καταλήγει σε φυσική διαγραφή από τη βάση δεδομένων. Αντίθετα, το φαγητό απενεργοποιείται μέσω του πεδίου published = false. Αυτός ο σχεδιασμός επιτρέπει την **διατήρηση ιστορικού δεδομένων** (π.χ. φαγητά που υπήρχαν σε παλαιότερες παραγγελίες) και την πιθανή επαναχρησιμοποίηση στο μέλλον. Η απόκρυψη των μη δημοσιευμένων φαγητών γίνεται αποκλειστικά σε επίπεδο frontend, διατηρώντας τη βάση πλήρη και αναλυτική.

- **Sequence Diagram – Δήλωση Μενού Ημέρας**

Το διάγραμμα περιγράφει δύο βασικές λειτουργίες: την εισαγωγή νέου φαγητού και την «διαγραφή» (δηλαδή απενεργοποίηση) ενός υπάρχοντος. Ο διαχειριστής αλληλεπιδρά με τη διεπαφή, συμπληρώνοντας τη φόρμα και πατώντας "Προσθήκη" ή "Απενεργοποίηση".

Στην περίπτωση προσθήκης, η διεπαφή στέλνει POST αίτημα στο insertNewFoodItems.php με πλήρη δεδομένα, ενώ στη δεύτερη περίπτωση στέλνει POST αίτημα στο updateFoodItemDetails.php με published = false. Το backend επαληθεύει το JWT και αποθηκεύει την εγγραφή ή ενημερώνει το πεδίο αντίστοιχα. Η απόκριση περιλαμβάνει επιβεβαίωση επιτυχίας και ανανέωση του πίνακα φαγητών στο UI.

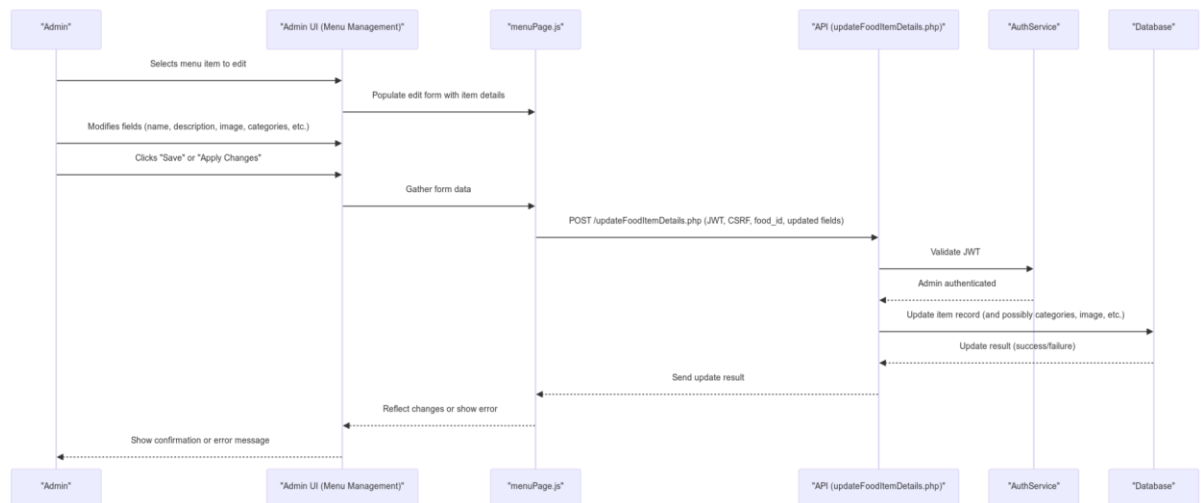


Εικόνα 50: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ διαχειριστή και συστατικών του συστήματος κατά τη δήλωση του μενού της ημέρας.

- **Sequence Diagram – Sequence Diagram – Τροποποίηση Υφιστάμενου Φαγητού**

Το δεύτερο διάγραμμα περιγράφει τη διαδικασία τροποποίησης ενός ήδη καταχωρημένου φαγητού. Ο διαχειριστής επιλέγει το είδος προς επεξεργασία, και το frontend προβάλλει τη φόρμα με τα υπάρχοντα δεδομένα. Ο χρήστης κάνει αλλαγές και πατά "Αποθήκευση", οπότε και αποστέλλεται POST αίτημα στο updateFoodItemDetails.php.

Το backend, αφού επαληθεύσει την ταυτότητα και εγκυρότητα των δεδομένων, εκτελεί την αντίστοιχη SQL UPDATE, τόσο στον πίνακα FoodItems όσο και στο FoodItemCategory (ανάλογα με τις κατηγορίες που τροποποιήθηκαν). Με την ολοκλήρωση, αποστέλλεται απόκριση επιτυχίας στο frontend, και εμφανίζεται σχετική ειδοποίηση στον διαχειριστή.



Εικόνα 51: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ διαχειριστή και συστατικών του συστήματος κατά την τροποποίηση ενός υφιστάμενου φαγητού.

5.6.3 Παρακολούθηση και Διαχείριση Παραγγελιών

Η διαχείριση των παραγγελιών χρηστών αποτελεί μία από τις πιο σημαντικές λειτουργίες του διαχειριστικού πάνελ, καθώς σχετίζεται άμεσα με την ομαλή εξυπηρέτηση των φοιτητών και τη διαχείριση των αποθεμάτων. Για το σκοπό αυτό, έχει σχεδιαστεί μια ειδική σελίδα στο περιβάλλον του διαχειριστή, στην οποία παρουσιάζονται όλες οι παραγγελίες που υποβάλλονται από τους χρήστες, ταξινομημένες ανά ημέρα, timeslot και status.

Κατά την είσοδό του στη σελίδα «Παραγγελίες», ο διαχειριστής βλέπει δύο κύριους πίνακες:

1. **Ο Πίνακας “Κανονικών” Παραγγελιών**, οι οποίες έχουν ήδη γίνει δεκτές από το σύστημα.
2. **Ο Πίνακας Εκκρεμών Παραγγελιών**, οι οποίες περιέχουν είδη σε έλλειψη και απαιτούν ανθρώπινη παρέμβαση.

Αυτόματη Έγκριση Παραγγελιών

Για λόγους ταχύτητας και ευκολίας, το σύστημα έχει σχεδιαστεί έτσι ώστε, **όταν όλα τα είδη μιας παραγγελίας είναι διαθέσιμα**, αυτή να γίνεται **αυτόματα δεκτή** από το σύστημα χωρίς να απαιτείται ενέργεια από τον διαχειριστή. Αυτή η αυτόματη έγκριση προσφέρει σημαντική **βελτιστοποίηση του χρόνου επεξεργασίας**, επιτρέποντας στον διαχειριστή να εστιάσει μόνο στις περιπτώσεις που χρήζουν παρέμβασης.

Διαχείριση Παραγγελιών με Ελλείψεις

Αν ένα ή περισσότερα από τα είδη μιας παραγγελίας βρίσκονται σε **έλλειψη (in danger)**, η παραγγελία μεταφέρεται αυτόματα στον δεύτερο πίνακα, με ένδειξη «εκκρεμεί έγκριση». Ο διαχειριστής ειδοποιείται από τη διεπαφή και καλείται να εξετάσει τις διαθέσιμες ποσότητες. Στη συνέχεια, μπορεί να επιλέξει είτε να **δεχτεί** την παραγγελία (δεσμεύοντας τους διαθέσιμους πόρους), είτε να την **απορρίψει**, εφόσον δεν μπορεί να εξυπηρετηθεί πλήρως.

Δυνατότητα Απόρριψης Εντός Προθεσμίας

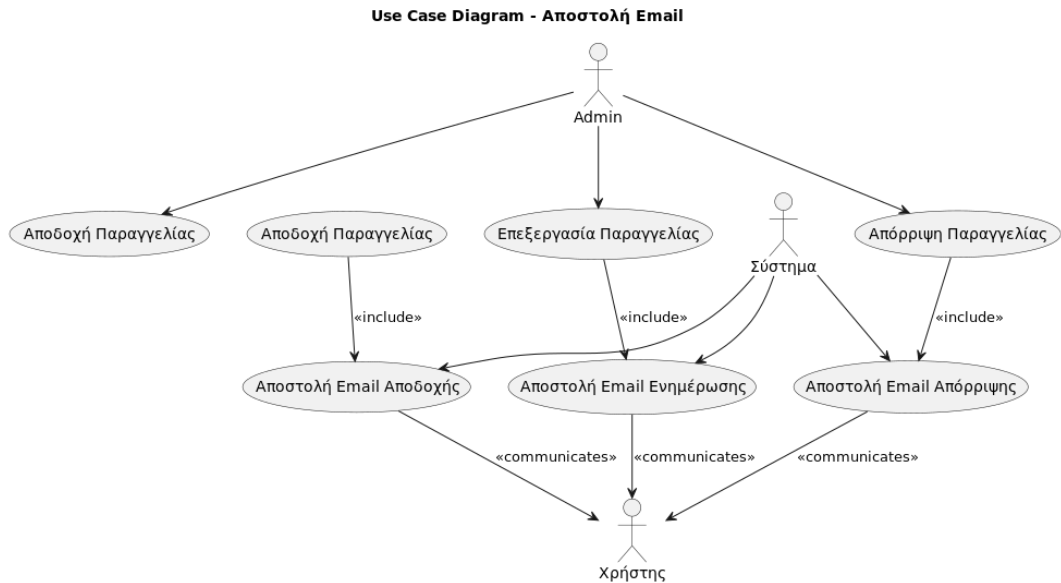
Ακόμη και για παραγγελίες που έγιναν δεκτές αυτόματα από το σύστημα, ο διαχειριστής διατηρεί τη δυνατότητα **να τις ακυρώσει ή να τις απορρίψει**, υπό την προϋπόθεση ότι απομένει τουλάχιστον **20 λεπτά μέχρι την έναρξη του επιλεγμένου timeslot**. Αυτή η ευελιξία επιτρέπει την αντιμετώπιση έκτακτων καταστάσεων, όπως προβλήματα με τον ανεφοδιασμό ή μεμονωμένα σφάλματα.

Σε όλες τις παραπάνω περιπτώσεις, ενεργοποιείται **μηχανισμός ειδοποίησης μέσω email**, ο οποίος ενημερώνει τον χρήστη για την κατάληξη της παραγγελίας του — είτε πρόκειται για θετική επιβεβαίωση, είτε για απόρριψη με αιτιολόγηση.

- **Sequence Diagram – Activity Diagram – Διαχείριση Παραγγελίας με Ελλείψεις**

Το activity diagram παρουσιάζει την ροή ενεργειών όταν υπάρχει παραγγελία που περιέχει είδη σε έλλειψη. Αρχικά, ο διαχειριστής εισέρχεται στον πίνακα των εκκρεμών παραγγελιών, όπου κάθε εγγραφή συνοδεύεται από λίστα φαγητών και πληροφορία διαθεσιμότητας.

Ο διαχειριστής εξετάζει τα δεδομένα και λαμβάνει απόφαση. Αν η παραγγελία μπορεί να εξυπηρετηθεί, εκτελείται η ενέργεια «Αποδοχή», και το σύστημα προχωρά σε ενημέρωση της κατάστασης της παραγγελίας. Αν δεν μπορεί να ικανοποιηθεί, επιλέγεται «Απόρριψη» και ενεργοποιείται η διαδικασία ενημέρωσης χρήστη. Και στις δύο περιπτώσεις, ενεργοποιείται ο μηχανισμός αποστολής email.

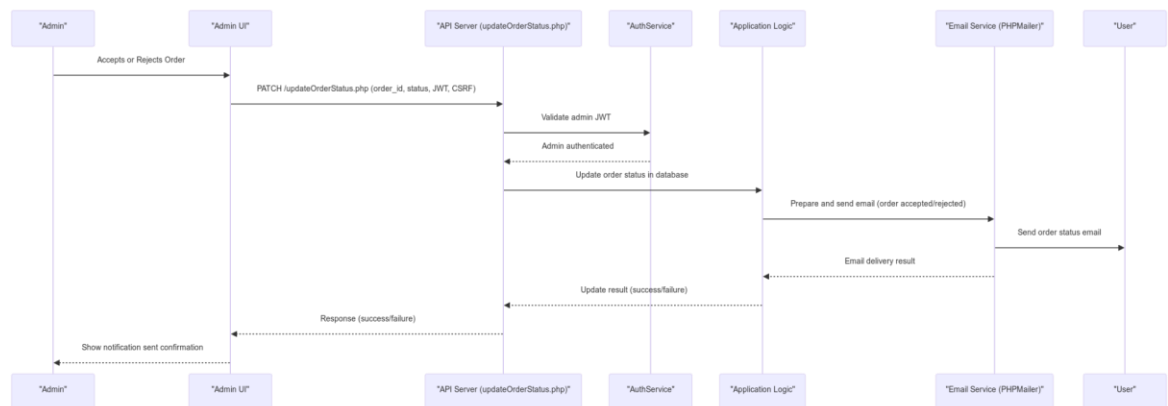


Εικόνα 52: *Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ χρηστών και συστατικών του συστήματος κατά τη διαχείριση παραγγελίας με ελλείψεις.*

• Sequence Diagram – Activity Diagram – Διαχείριση Παραγγελίας με Ελλείψεις

Το sequence diagram περιγράφει την αλληλουχία ενεργειών όταν απαιτείται να ενημερωθεί ο χρήστης για την κατάληξη της παραγγελίας του. Όταν ο διαχειριστής κάνει αποδοχή ή απόρριψη μιας παραγγελίας, το backend ενημερώνει πρώτα τη βάση δεδομένων για την αλλαγή κατάστασης.

Στη συνέχεια, ενεργοποιείται εσωτερική συνάρτηση αποστολής email, η οποία δημιουργεί προσωποποιημένο μήνυμα ανάλογα με την έκβαση (επιβεβαίωση ή απόρριψη). Το μήνυμα αποστέλλεται μέσω SMTP service στον φοιτητή, και το frontend εμφανίζει σχετικό feedback στον διαχειριστή για την επιτυχία ή αποτυχία αποστολής.

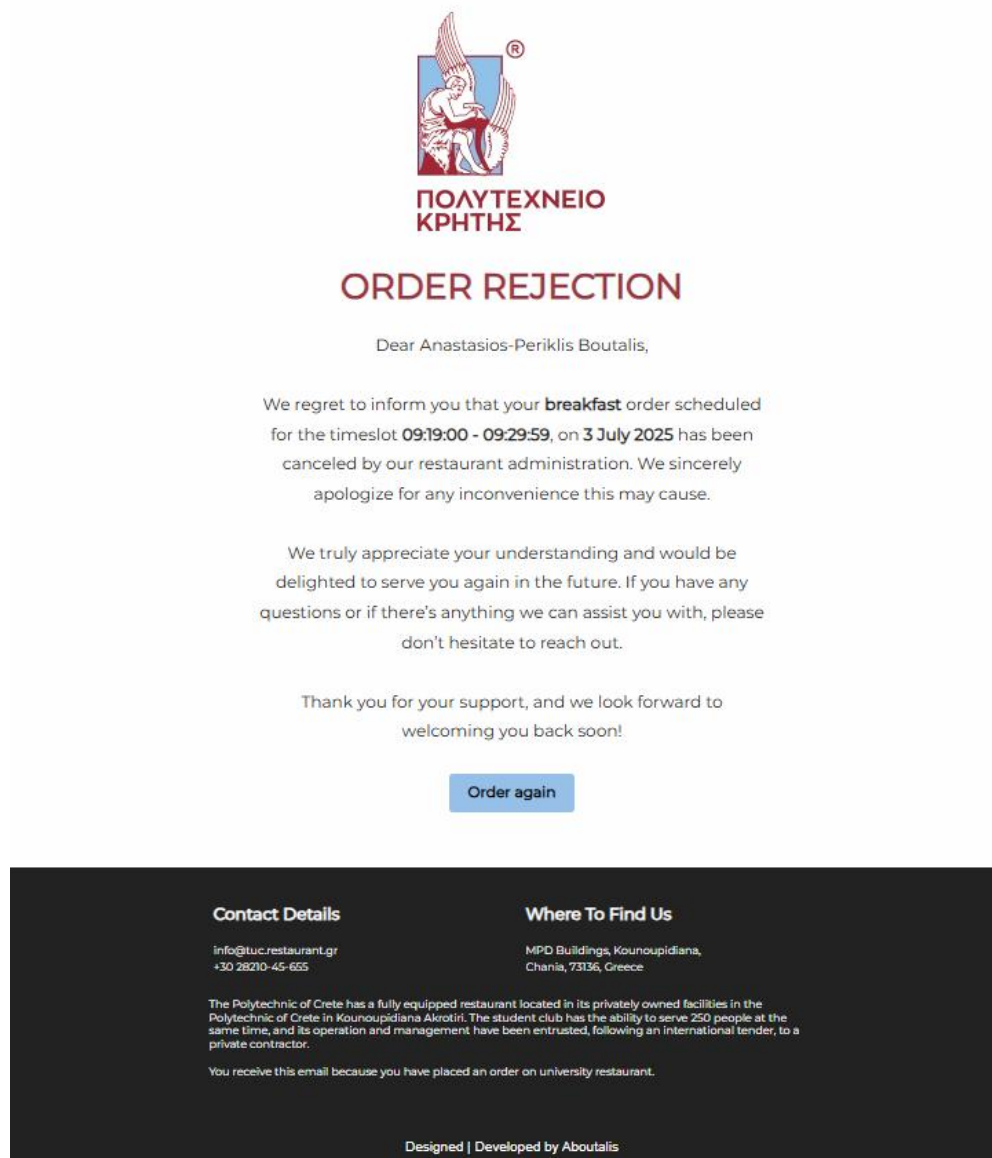


Εικόνα 53: *Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ χρηστών και συστατικών του συστήματος κατά τη διαχείριση παραγγελίας με ελλείψεις.*

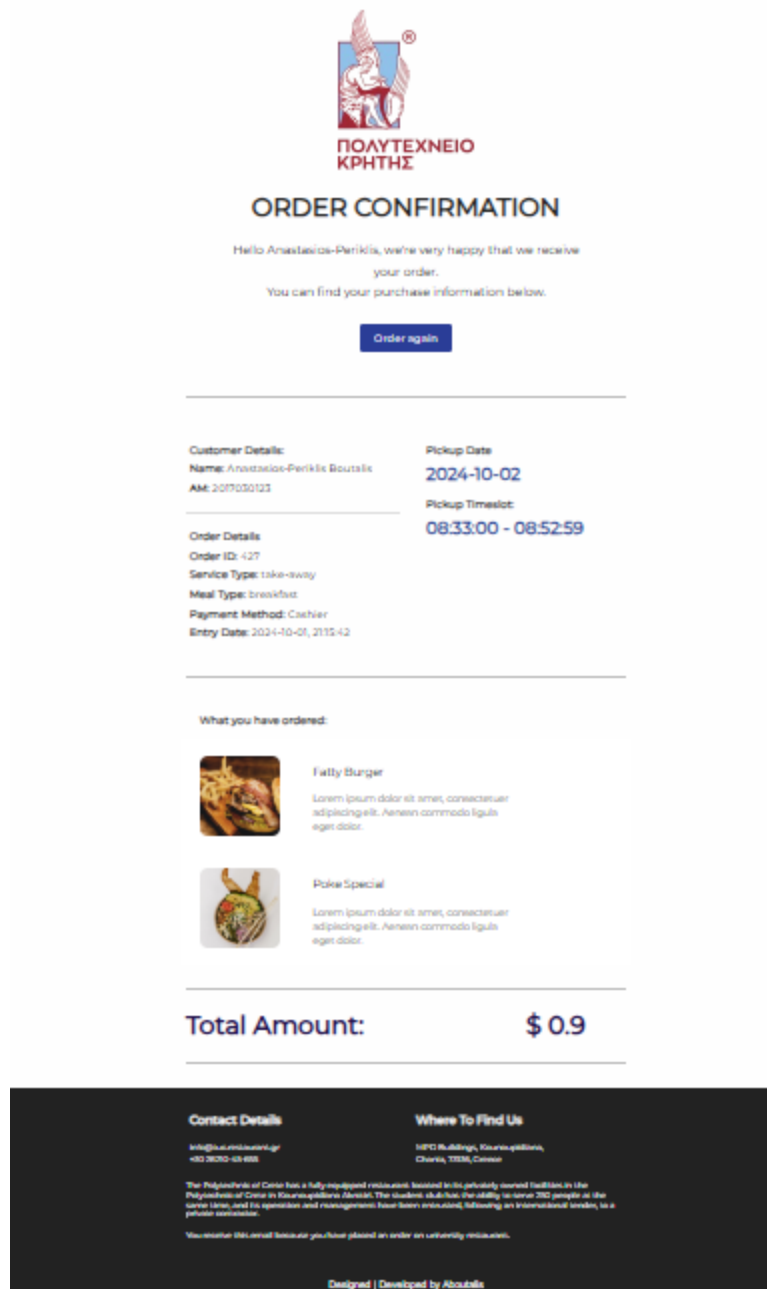
- **Email UI Snapshot – Ειδοποίηση Χρήστη**

Η στατική απεικόνιση παρουσιάζει παραδείγματα του περιεχομένου που λαμβάνει ο φοιτητής μέσω email. Σε περίπτωση αποδοχής παραγγελίας, εμφανίζεται μήνυμα επιβεβαίωσης με λεπτομέρειες για το γεύμα, το timeslot και τον κωδικό παραγγελίας και άλλες πληροφορίες της παραγγελίας. Αν η παραγγελία απορριφθεί, το email ενημερώνει πως δεν προχώρησε η παραγγελία και ενθαρρύνει τον χρήστη να προβεί σε νέα επιλογή.

Τα μηνύματα έχουν σχεδιαστεί ώστε να είναι **σαφή, ευγενικά και ενημερωτικά**, βελτιώνοντας την εμπειρία χρήστη ακόμα και σε περιπτώσεις απόρριψης.



Εικόνα 54: Στιγμιότυπο οθόνης της διεπαφής χρήστη που εμφανίζει ειδοποίηση email απόρριψης παραγγελίας.



Εικόνα 55: Στιγμιότυπο οθόνης της διεπαφής χρήστη που εμφανίζει ειδοποίηση email αποδοχής παραγγελίας.

5.6.4 Διαχείριση Voucher Χρηστών

Μία από τις βασικές αρμοδιότητες του διαχειριστή αφορά τη **διαχείριση των vouchers** των χρηστών, μέσω των οποίων πραγματοποιούνται αγορές γευμάτων από φοιτητές που **δεν δικαιούνται δωρεάν σίτιση** ή έχουν εξαντλήσει το ημερήσιο δικαίωμα δωρεάν γεύματος. Το σύστημα voucher λειτουργεί ως ένα **ψηφιακό πορτοφόλι**, του οποίου η διαχείριση — προσθήκες

και επιστροφές ποσών — πραγματοποιείται αποκλειστικά από τον διαχειριστή μέσω του αντίστοιχου dashboard.

Εμφάνιση και Αναζήτηση Χρηστών

Ο διαχειριστής έχει πρόσβαση στη σελίδα "Voucher Management" από το κεντρικό μενού διαχείρισης. Εκεί εμφανίζεται ένας πίνακας που περιλαμβάνει **όλους τους εγγεγραμμένους χρήστες**, είτε έχουν δωρεάν σίτιση είτε όχι. Για ευκολία πλοήγησης, παρέχεται **λειτουργία αναζήτησης** μέσω πεδίου search, η οποία λειτουργεί με βάση το ονοματεπώνυμο, το ΑΜ ή το email του χρήστη.

Προβολή και Τροποποίηση Υπολοίπου

Με κλικ σε οποιονδήποτε χρήστη από τη λίστα, ανοίγει ένα **διαδραστικό παράθυρο (dialog)**, όπου παρουσιάζεται το τρέχον υπόλοιπο του voucher και το ιστορικό προηγούμενων κινήσεων. Από αυτό το σημείο, ο διαχειριστής μπορεί να επιλέξει μία από τις δύο ενέργειες:

1. **Προσθήκη ποσού:** Εισάγει το ποσό και το σύστημα ενημερώνει το συνολικό υπόλοιπο του χρήστη.
2. **Επιστροφή χρημάτων:** Εισάγει το ποσό που θα αφαιρεθεί από το υπάρχον υπόλοιπο, καταγράφοντας τη συναλλαγή ως επιστροφή.

Και στις δύο περιπτώσεις, το σύστημα **καταγράφει αναλυτικά τη συναλλαγή στη βάση δεδομένων**, διατηρώντας ιστορικό με ημερομηνία, ποσό και τύπο κίνησης.

Ιστορικότητα και Μη Λήξη Υπολοίπου

Ιδιαίτερη πρόβλεψη έχει δοθεί στο γεγονός ότι **τα vouchers δεν έχουν ημερομηνία λήξης**. Το υπόλοιπο ενός φοιτητή διατηρείται στο σύστημα **επ' αόριστον**, επιτρέποντας τη σταδιακή χρήση του χωρίς χρονικούς περιορισμούς. Αυτή η προσέγγιση αυξάνει την ευελιξία και μειώνει τη γραφειοκρατική επιβάρυνση, ενώ ενισχύει την αξιοπιστία του συστήματος από την πλευρά των φοιτητών.

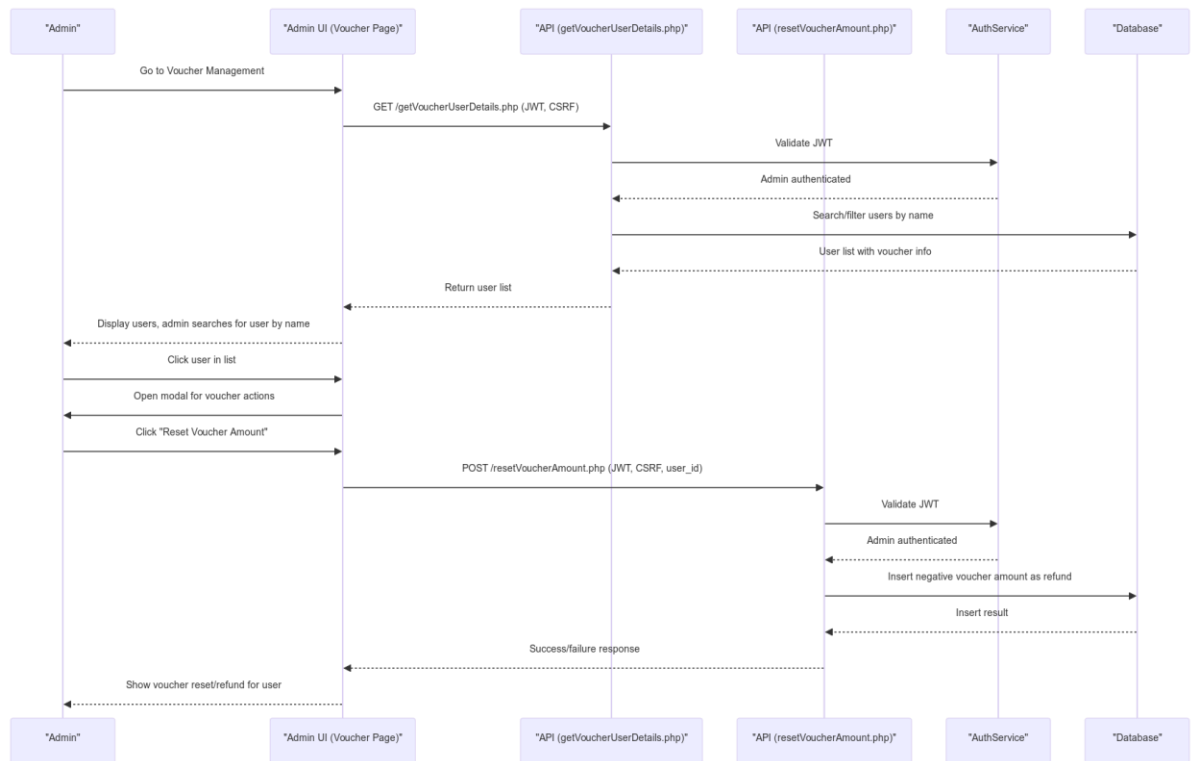
● Sequence Diagram – Διαχείριση Voucher από Διαχειριστή

Το διάγραμμα ακολουθίας αποτυπώνει τη ροή ενεργειών κατά την επεξεργασία ενός voucher. Η διαδικασία ξεκινά με την είσοδο του διαχειριστή στη σελίδα διαχείρισης vouchers και την ανάκτηση της λίστας όλων των χρηστών μέσω αιτήματος στο endpoint `fetchUsers.php`.

Κατόπιν αναζήτησης και επιλογής χρήστη, το frontend προβάλλει το dialog, το οποίο ανακτά το `currentBalance` και το `transactionHistory` μέσω του endpoint `getVoucherDetails.php`. Ο διαχειριστής μπορεί να επιλέξει «Προσθήκη» ή «Επιστροφή»,

εισάγοντας αντίστοιχο ποσό. Η ενέργεια αποστέλλεται με POST αίτημα στο updateVoucher.php, μαζί με το user_id, το ποσό και τον τύπο κίνησης (credit ή refund).

Το backend επαληθεύει την εγκυρότητα του χρήστη, ενημερώνει τον πίνακα Vouchers και προσθέτει νέα εγγραφή στον πίνακα Vouchers. Η απάντηση επιστρέφει επιβεβαίωση επιτυχίας, και η διεπαφή ανανεώνει άμεσα την πληροφορία υπολοίπου και ιστορικού στο dialog.



Εικόνα 56: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ διαχειριστή και συστατικών του συστήματος κατά τη διαχείριση voucher.

5.6.5 Διαχείριση Timeslots

Η σωστή λειτουργία της λέσχης προϋποθέτει την **οργάνωση της εξυπηρέτησης των φοιτητών σε χρονικά διαστήματα (timeslots)**, τα οποία καθορίζουν πότε επιτρέπεται η παραλαβή ή κατανάλωση κάθε τύπου γεύματος (πρωινό, μεσημεριανό, βραδινό). Η δυνατότητα διαχείρισης αυτών των διαστημάτων έχει ανατεθεί στον διαχειριστή της εφαρμογής, ώστε να μπορεί να προσαρμόζει δυναμικά το πρόγραμμα εξυπηρέτησης σύμφωνα με τις ανάγκες του προσωπικού, τη διαθεσιμότητα της κουζίνας, και την εκτιμώμενη ζήτηση.

Βασικοί Κανόνες Διαμόρφωσης

Κάθε timeslot συνδέεται με ένα συγκεκριμένο τύπο γεύματος, και όλοι οι διαθέσιμοι τύποι (πρωινό, μεσημεριανό, βραδινό) πρέπει να βρίσκονται εντός του επιτρεπτού ωραρίου λειτουργίας της λέσχης. Επιπλέον, έχει τεθεί ως κανόνας η μη επικάλυψη των timeslots για το ίδιο γεύμα, ώστε να αποτρέπονται συγχύσεις και τεχνικές ασυμβατότητες κατά την υποβολή και έγκριση παραγγελιών. Τα timeslots για κάθε γεύμα είναι επίσης συνεχόμενα, χωρίς κενά ενδιάμεσα, για να διασφαλιστεί ο ρυθμισμένος και ισόρροπος καταμερισμός των φοιτητών.

Ο διαχειριστής έχει την ευχέρεια:

- Να δημιουργεί νέα timeslots ανά τύπο γεύματος.
- Να τροποποιεί τις ώρες έναρξης και λήξης ενός υπάρχοντος timeslot.
- Να απενεργοποιεί (και όχι να διαγράφει) ένα timeslot, διατηρώντας ωστόσο τη σχετική εγγραφή στη βάση για λόγους ιστορικότητας.

Μη Διαγραφή αλλά Απόκρυψη

Όπως και σε άλλες λειτουργίες του διαχειριστή, η "διαγραφή" ενός timeslot δεν οδηγεί σε φυσική διαγραφή από τη βάση δεδομένων. Αντίθετα, το timeslot επισημαίνεται με `active = false`, με αποτέλεσμα να μην εμφανίζεται στο frontend για μελλοντική χρήση, αλλά να διατηρείται πλήρως στη βάση. Η προσέγγιση αυτή έχει υιοθετηθεί με στόχο τη διατήρηση πλήρους ιστορικού λειτουργίας της λέσχης και την ορθή ανάκτηση δεδομένων σε αναφορές ή αναλύσεις.

• Sequence Diagram – Διαχείριση Timeslots από Διαχειριστή

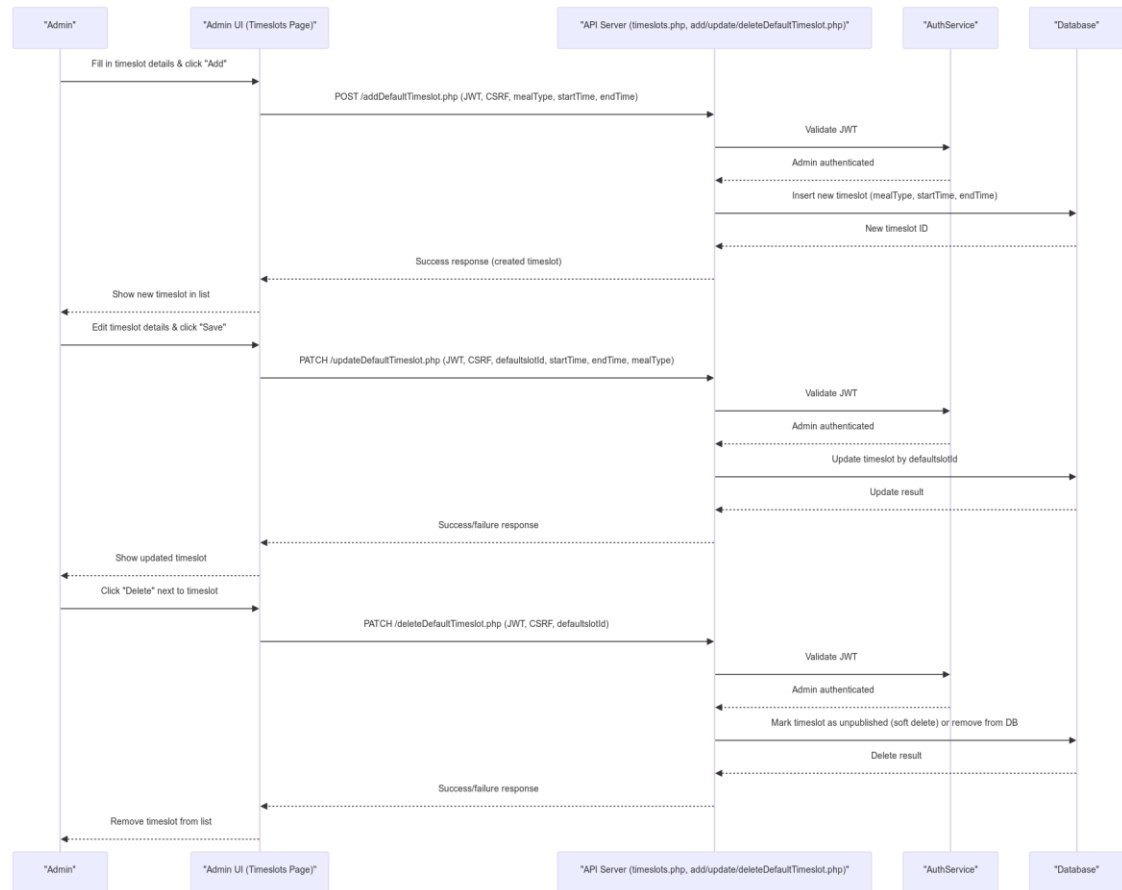
Το sequence diagram αποτυπώνει την αλληλουχία ενεργειών που ακολουθούνται όταν ο διαχειριστής εισέρχεται στη σελίδα διαχείρισης timeslots. Το frontend στέλνει GET αίτημα στο `fetchAllTimeslots.php` για να εμφανίσει τη λίστα όλων των καταχωρημένων timeslots, οργανωμένων ανά τύπο γεύματος.

Ο διαχειριστής μπορεί:

- Να προσθέσει ένα νέο timeslot, εισάγοντας τύπο γεύματος, ώρα έναρξης και ώρα λήξης. Το frontend στέλνει τα δεδομένα στο `insertTimeslot.php`.
- Να τροποποιήσει τις ώρες σε υπάρχον timeslot. Η ενημέρωση πραγματοποιείται μέσω POST ή PATCH αιτήματος στο `updateTimeslot.php`.
- Να απενεργοποιήσει ένα timeslot, πατώντας την αντίστοιχη επιλογή "Απόκρυψη", η οποία ενεργοποιεί POST αίτημα προς `disableTimeslot.php` με `active = false`.

Σε κάθε περίπτωση, το backend ελέγχει την εγκυρότητα των δεδομένων, επιβεβαιώνει ότι δεν υπάρχει επικάλυψη, και αποθηκεύει την αλλαγή. Η διεπαφή ανανεώνεται δυναμικά με

οπτικό feedback προς τον διαχειριστή.



Εικόνα 57: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ διαχειριστή και συστατικών του συστήματος κατά τη διαχείριση των Timeslots.

5.6.6 Καθορισμός Ημερομηνιών Διακοπών

Μια βασική επιχειρησιακή ανάγκη που καλείται να εξυπηρετήσει η εφαρμογή είναι η **προσαρμογή της λειτουργίας της λέσχης** σε περιόδους όπου παραμένει **κλειστή λόγω διακοπών** (π.χ. Χριστούγεννα, Πάσχα, θερινές διακοπές ή έκτακτες συνθήκες). Για τον σκοπό αυτό, ο διαχειριστής έχει πρόσβαση σε ειδική διεπαφή μέσω της οποίας μπορεί να **καθορίσει με ευκολία τις ημερομηνίες όπου η λέσχη δεν θα λειτουργεί**.

Δήλωση Περιόδων Διακοπών

Ο διαχειριστής εισέρχεται στη σελίδα «Διακοπές» και επιλέγει **διάστημα ημερομηνιών** (από - έως) που αντιστοιχεί στην προγραμματισμένη διακοπή λειτουργίας. Οι ημερομηνίες αυτές αποθηκεύονται στη βάση μέσω του endpoint `setVacationDates.php`, και καταγράφονται στον

πίνακα Vacations, με επιπλέον μεταδεδομένα όπως reason ή περίοδος.

Το σύστημα υποστηρίζει **πολλαπλές περιόδους διακοπών**, χωρίς αλληλοεπικαλύψεις, και με δυνατότητα μελλοντικής τροποποίησης ή επέκτασης των ημερομηνιών.

Αντίκτυπος στον Χρήστη

Όταν ένας χρήστης επισκέπτεται την εφαρμογή κατά την περίοδο διακοπών, γίνεται **έλεγχος ημερομηνίας στο backend**. Αν η τρέχουσα ημερομηνία εμπίπτει σε δηλωμένο διάστημα διακοπών, τότε το σύστημα:

- **Απενεργοποιεί όλα τα κουμπιά παραγγελίας**
- **Εμφανίζει ενημερωτικό μήνυμα** στην αρχική σελίδα, που αναφέρει ότι η λέσχη παραμένει κλειστή λόγω διακοπών

Αυτός ο έλεγχος εξασφαλίζει ότι δεν πραγματοποιούνται **παραγγελίες σε μη λειτουργικές ημέρες**, ενισχύοντας τη σαφήνεια προς τους χρήστες και αποφεύγοντας σύγχυση ή τεχνικά σφάλματα.

Περιορισμοί στη Διαχείριση από Admin

Η επιρροή των δηλωμένων διακοπών **επεκτείνεται και στη διαχειριστική πλευρά** της εφαρμογής. Ο διαχειριστής **δεν έχει τη δυνατότητα να δηλώσει μενού ημέρας** για ημερομηνίες που εμπίπτουν σε περίοδο διακοπών. Αν προσπαθήσει ο διαχειριστής να καταχωρήσει μενού ημέρας για κάποια από αυτή την περίοδο ημερών δεν είναι εφικτό, καθώς και αυτός βλέπει αντίστοιχο μήνυμα διακοπών όπως και ο χρήστης.

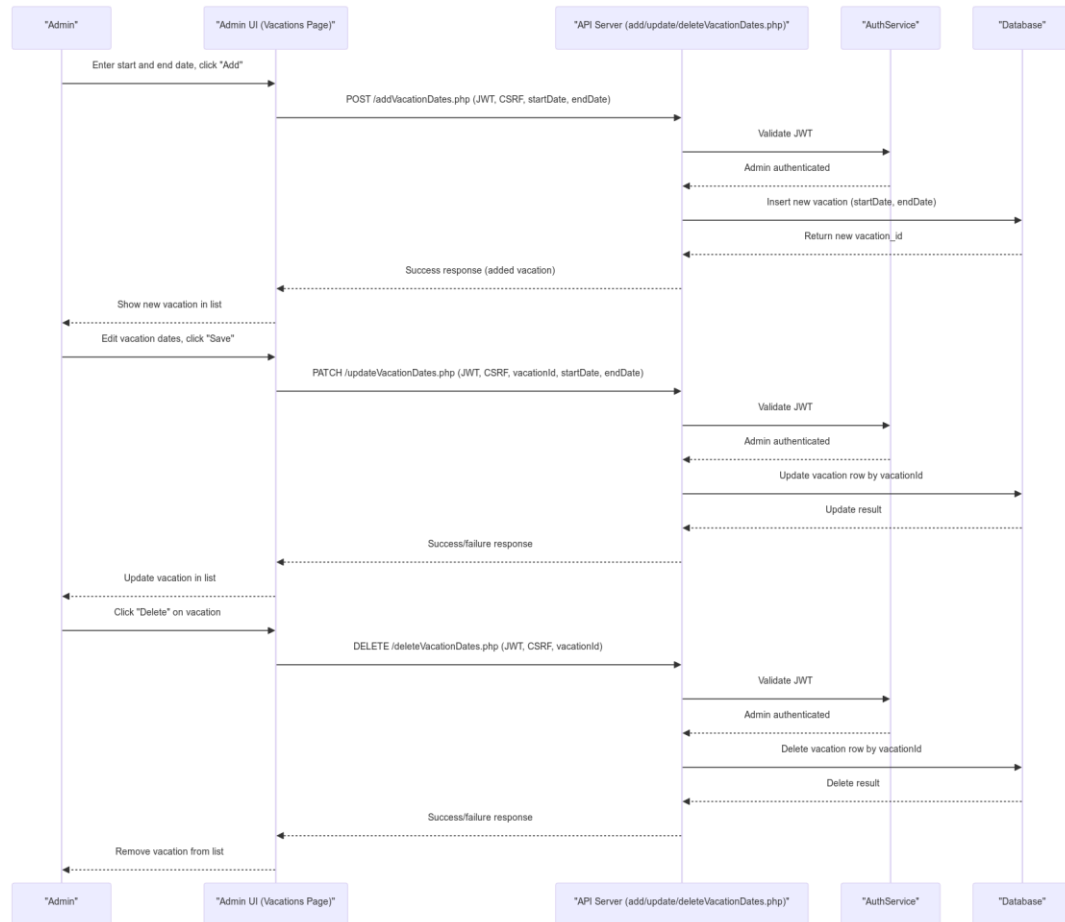
Αυτό το μέτρο εγγυάται ότι η **εσωτερική λογική της εφαρμογής παραμένει συνεπής**, και δεν δημιουργούνται ορφανές ή λανθασμένες εγγραφές που θα έπρεπε να θεωρούνται άκυρες λόγω διακοπών.

- **Sequence Diagram – Καθορισμός Διακοπών από Διαχειριστή**

Το sequence diagram αποτυπώνει την ροή ενεργειών για τη δήλωση διακοπών. Ο διαχειριστής εισέρχεται στη σελίδα Vacations, και μέσω της διεπαφής ημερολογίου επιλέγει διάστημα ημερομηνιών. Το frontend στέλνει POST αίτημα στο setVacationDates.php, περνώντας τις ημερομηνίες και προαιρετική αιτιολόγηση.

Το backend επαληθεύει την εγκυρότητα της περιόδου (π.χ. να μην επικαλύπτεται με υπάρχουσα), και αποθηκεύει τη νέα εγγραφή στον πίνακα Vacations. Στη συνέχεια, όλα τα endpoints παραγγελίας (submitOrder.php, fetchMenu.php) περιλαμβάνουν πλέον έλεγχο αν η τρέχουσα ημερομηνία εμπίπτει σε περίοδο διακοπών, και απορρίπτουν

αιτήματα, επιστρέφοντας σχετικό σφάλμα ή μήνυμα. Παράλληλα, το frontend εμφανίζει εντός εφαρμογής ενημερωτικό banner στον χρήστη.



Εικόνα 58: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ διαχειριστή και συστατικών του συστήματος κατά τον καθορισμό περιόδων διακοπών.

5.6.7 Καθορισμός Ημερομηνιών Διακοπών

Η πρόβλεψη μερίδων αποτελεί μία από τις πιο **καινοτόμες λειτουργικότητες** της εφαρμογής, δίνοντας στον διαχειριστή τη δυνατότητα να λαμβάνει **στοιχεία τεκμηριωμένης ζήτησης** προτού προβεί στη δήλωση ημερήσιου μενού. Στόχος του μηχανισμού είναι η **βελτιστοποίηση πόρων**, η **αποφυγή σπατάλης φαγητού** και η **αντιστοίχιση παραγόμενων μερίδων με την αναμενόμενη ζήτηση**.

Διαδικασία Πρόβλεψης από τη Διεπαφή

Η διεπαφή της πρόβλεψης βρίσκεται σε αυτόνομη σελίδα του διαχειριστικού dashboard, με ονομασία "Πρόβλεψη Μερίδων". Ο διαχειριστής καλείται να επιλέξει:

- Την **ημερομηνία** για την οποία θέλει να προβλέψει τις μερίδες
- Το **είδος ημέρας**, επιλέγοντας ανάμεσα σε "Κανονική Ημέρα", "Αργία", ή "Περίοδος Εξεταστικής"

Αυτές οι επιλογές είναι κρίσιμες, καθώς επιδρούν άμεσα στο **μοντέλο πρόβλεψης**, το οποίο προσαρμόζει τα αποτελέσματά του με βάση την εποχικότητα και την ιστορική κατανάλωση ανά τύπο ημέρας.

Μόλις γίνουν οι επιλογές, το frontend αποστέλλει αίτημα προς το endpoint `getPredictionData.php`, περιλαμβάνοντας ως παραμέτρους την ημερομηνία και το είδος της ημέρας. Το backend καλεί τον **μηχανισμό πρόβλεψης**, ο οποίος συνδυάζει:

- Ιστορικά δεδομένα παραγγελιών
- Την εποχικότητα
- Τον τύπο ημέρας
- Τον αριθμό ενεργών φοιτητών εκείνη την περίοδο (αντλείται δυναμικά)

Έξοδος Προβλέψεων και Παρουσίαση

Το σύστημα επιστρέφει για κάθε φαγητό στη βάση:

- Τη **μέγιστη προβλεπόμενη ζήτηση (max value)**
- Τη **ελάχιστη (min value)**
- Τη **βέλτιστη πρόβλεψη (expected value)**

Αυτά τα δεδομένα εμφανίζονται σε πίνακα ή κάρτες στην οθόνη, δίπλα από κάθε φαγητό. Έτσι, ο διαχειριστής μπορεί να καθοδηγηθεί στο **ποια φαγητά θα συμπεριλάβει στο μενού και σε τι ποσότητες**, χωρίς να χρειάζεται να βασιστεί αποκλειστικά στην εμπειρική του εκτίμηση.

Σημειώνεται πως **αν κάποιο φαγητό δεν διαθέτει επαρκή ιστορικά δεδομένα**, ή τα δεδομένα που υπάρχουν δεν πληρούν τα στατιστικά κριτήρια για πρόβλεψη, τότε **δεν επιστρέφεται καμία εκτίμηση** για αυτό. Η εφαρμογή εμφανίζει αντίστοιχη ένδειξη ("Μη διαθέσιμα δεδομένα").

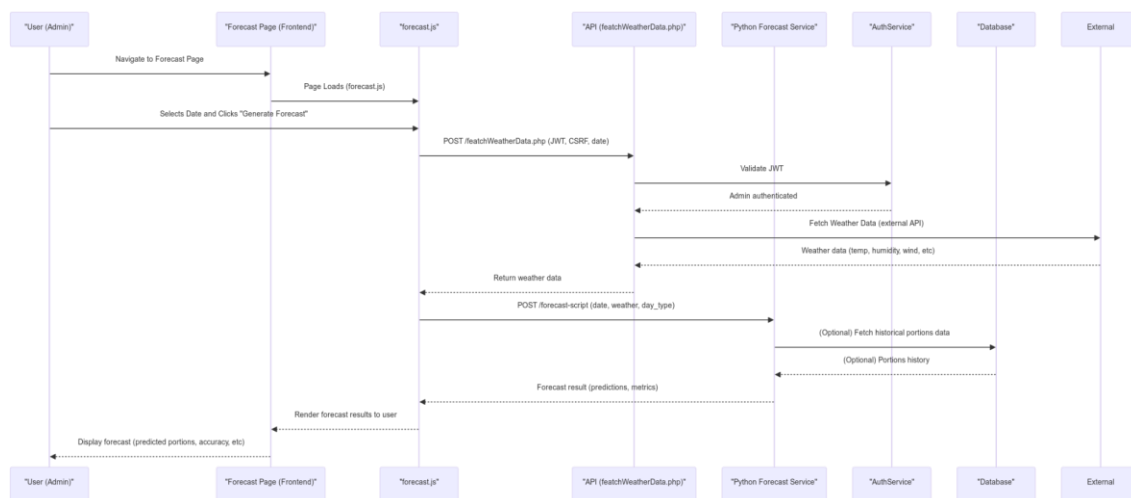
Σχέση με Δήλωση Μενού

Ο μηχανισμός πρόβλεψης δεν καταχωρεί καμία αλλαγή αυτόματα στο μενού. Λειτουργεί αποκλειστικά ως **εργαλείο υποστήριξης αποφάσεων**, και η δήλωση φαγητών στο ημερήσιο μενού εξακολουθεί να γίνεται χειροκίνητα, από άλλη σελίδα. Παρ' όλα αυτά, η σύζευξη των δύο λειτουργιών — πρόβλεψη και μενού — ενισχύει την **ορθολογική διαχείριση αποθεμάτων** και την **προσαρμοστικότητα** της λέσχης.

- **Sequence Diagram – Ενεργοποίηση Μηχανισμού Πρόβλεψης**

Το sequence diagram παρουσιάζει τη σειρά ενεργειών όταν ο διαχειριστής επιλέγει ημερομηνία και είδος ημέρας. Η διεπαφή στέλνει POST ή GET αίτημα στο `getPredictionData.php`, με τις παραμέτρους `date` και `day_type`.

Το backend ενεργοποιεί το μηχανισμό πρόβλεψης (μοντέλο που θα αναλυθεί στο Κεφάλαιο 6), αναλύει τα διαθέσιμα δεδομένα, και επιστρέφει για κάθε φαγητό τη `min`, `expected` και `max` τιμή. Το frontend ανανεώνει το UI με τα νέα δεδομένα. Σε περίπτωση ανεπαρκών δεδομένων για κάποιο φαγητό, απλώς παραλείπεται από την απάντηση.



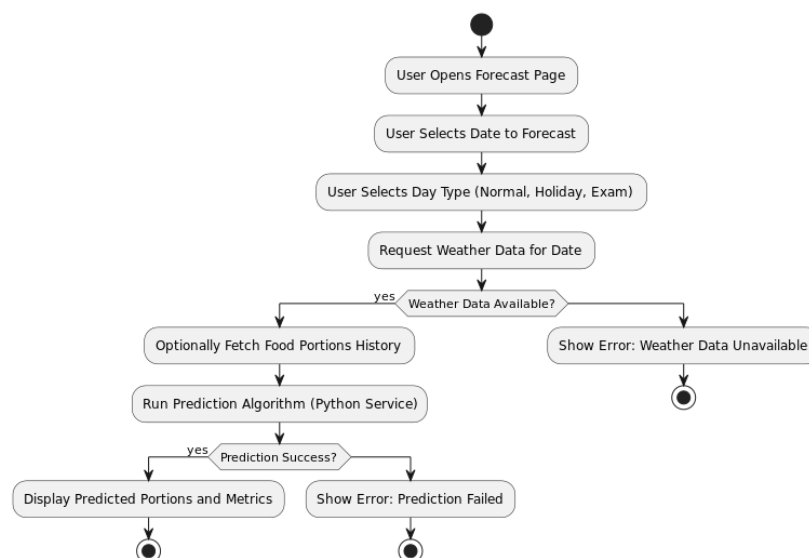
Εικόνα 59: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ διαχειριστή και συστατικών του συστήματος κατά την ενεργοποίηση του μηχανισμού πρόβλεψης.

- **Activity Diagram – Διαδικασία Πρόβλεψης**

Το activity diagram απεικονίζει συνοπτικά την ροή της διαδικασίας πρόβλεψης. Ξεκινά με την επιλογή ημερομηνίας και τύπου ημέρας από τον διαχειριστή, και προχωρά με την αποστολή των δεδομένων στο backend.

Ο μηχανισμός εκτελεί τους αλγορίθμους πρόβλεψης, διαχωρίζει φαγητά με επαρκές

ιστορικό από όσα δεν έχουν επαρκή δείγματα, δημιουργεί προβλέψεις, και τις επιστρέφει. Τέλος, το αποτέλεσμα προβάλλεται στη διεπαφή.



Εικόνα 60: Activity Diagram που παρουσιάζει τη ροή ενεργειών κατά τη διαδικασία πρόβλεψης ζήτησης με βάση τον μηχανισμό πρόβλεψης.

Τέλος, ο διαχειριστής διαθέτει και τη **δυνατότητα να ενεργοποιεί ή να απενεργοποιεί την υποβολή παραγγελιών συνολικά**, για όλους τους χρήστες. Η λειτουργία αυτή αποτελεί **κεντρικό ρυθμιστικό διακόπτη** του backend και έχει προβλεφθεί για περιπτώσεις όπου απαιτείται η **παύση της παραγγελιοληψίας** λόγω τεχνικών προβλημάτων, έκτακτων περιστάσεων, ή λειτουργικών θεμάτων στη λέσχη.

Η ενεργοποίηση ή απενεργοποίηση γίνεται από ειδικό toggle στο διαχειριστικό πάνελ. Όταν η δυνατότητα είναι απενεργοποιημένη, το σύστημα αποκλείει αυτόματα την εμφάνιση της φόρμας παραγγελίας στο frontend, και εμφανίζει στους φοιτητές σχετικό μήνυμα, διασφαλίζοντας ότι δεν αποστέλλονται καθόλου αιτήματα προς το backend.

Η λειτουργικότητα αυτή έχει **τεκμηριωθεί αναλυτικά παραπάνω**, στο πλαίσιο της αρχιτεκτονικής διαχείρισης παραγγελιών και των περιπτώσεων ειδικής μεταχείρισης

5.7 Έλεγχος Πρόσβασης και Ασφάλεια

Η ασφάλεια αποτέλεσε βασικό σχεδιαστικό άξονα κατά την υλοποίηση της εφαρμογής, τόσο ως προς την αυθεντικοποίηση χρηστών όσο και ως προς την προστασία από κακόβουλες ενέργειες ή παραβίαση δεδομένων. Η αρχιτεκτονική του συστήματος ενσωματώνει μηχανισμούς ασφαλείας που καλύπτουν τους κύριους τύπους επιθέσεων και τις συνήθεις ευπάθειες διαδικτυακών

εφαρμογών.

5.7.1 Ανάλυση Μεθόδων

1. Προστασία από Cross-Site Scripting (XSS)

Η εφαρμογή φροντίζει να εξουδετερώνει κάθε δυνατότητα εισαγωγής κακόβουλου κώδικα JavaScript μέσω φόρμας ή URL παραμέτρων. Πιο συγκεκριμένα:

- Όλες οι εισοδοι από τον χρήστη **καθαρίζονται (sanitized)** πριν αποθηκευτούν ή εμφανιστούν.
- Η έξοδος περιεχομένου γίνεται με χρήση HTML escaping (`htmlspecialchars()`, `htmlspecialchars()`), αποτρέποντας την εκτέλεση μη ασφαλών tags.

Έτσι, ακόμα και αν επιχειρηθεί η εισαγωγή script σε πεδία όπως όνομα φαγητού ή περιγραφή, αυτό δεν εκτελείται στο πρόγραμμα περιήγησης.

2. Προστασία από Cross-Site Request Forgery (CSRF)

Κάθε ενέργεια που τροποποιεί δεδομένα (π.χ. αποστολή παραγγελίας, προσθήκη φαγητού, ενημέρωση menu) απαιτεί **έγκυρο CSRF token**, το οποίο δημιουργείται κατά την αυθεντικοποίηση και ενσωματώνεται σε κάθε POST αίτημα.

- Το backend ελέγχει τη συμβατότητα του token με το session ή το JWT του χρήστη.
- Σε περίπτωση απουσίας ή παραποίησης του token, το αίτημα απορρίπτεται.

Αυτός ο μηχανισμός προστατεύει από επιθέσεις όπου τρίτος ιστότοπος επιχειρεί να "παραπλανήσει" τον χρήστη να εκτελέσει ενέργειες χωρίς τη θέλησή του.

3. Καθαρισμός και έλεγχος εισροών (Sanitization)

Όλες οι τιμές που εισάγονται από τον χρήστη ελέγχονται ως προς:

- το μήκος και τον τύπο τους (π.χ. αριθμητικά πεδία, strings),
- την ύπαρξη ειδικών χαρακτήρων,
- και τη γενική τους εγκυρότητα πριν χρησιμοποιηθούν σε SQL queries ή εμφανιστούν στην οθόνη.

Αυτό περιορίζει τη δυνατότητα **SQL injection** και άλλων μορφών injection-based επιθέσεων.

4. JWT Authentication (JSON Web Token)

Για την πιστοποίηση ταυτότητας των χρηστών και την ασφαλή μεταφορά δεδομένων κατά την

επικοινωνία μεταξύ frontend και backend, χρησιμοποιείται το πρωτόκολλο **JWT**:

- Κατά την είσοδο, ο χρήστης λαμβάνει ένα υπογεγραμμένο token, το οποίο αποθηκεύεται στο browser (π.χ. σε localStorage) και ενσωματώνεται στα επόμενα αιτήματα.
- Το token περιλαμβάνει βασικές πληροφορίες για τον χρήστη (π.χ. ID, ρόλος) και επαληθεύεται στο backend χωρίς ανάγκη session state.

Με τον τρόπο αυτό, το σύστημα παραμένει **stateless και ασφαλές**, μειώνοντας την επιφάνεια επίθεσης και βελτιώνοντας τις επιδόσεις.

5. CAS Authentication (Central Authentication Service)

Η εφαρμογή υποστηρίζει ενοποίηση με **CAS** για την αυθεντικοποίηση χρηστών, ιδίως στο πλαίσιο του πανεπιστημιακού περιβάλλοντος. Η χρήση CAS εξασφαλίζει:

- **Single Sign-On (SSO)** εμπειρία για τους φοιτητές και το προσωπικό,
- Κεντρική διαχείριση λογαριασμών,
- Ενιαία πολιτική ασφάλειας και ελέγχου πρόσβασης από το Ίδρυμα.

Η σύνδεση γίνεται μέσω ανακατεύθυνσης στο CAS login portal, ενώ το backend επαληθεύει το εισερχόμενο ticket και δημιουργεί το αντίστοιχο JWT για την τοπική χρήση.

5.7.2 Επεξήγηση με χρήση διαγραμμάτων της Επαλήθευση Πρόσβασης και Αντιμετώπιση Κακόβουλων Αιτημάτων

Για να εξασφαλιστεί ότι η πρόσβαση στις λειτουργίες της εφαρμογής πραγματοποιείται μόνο από εξουσιοδοτημένους χρήστες, το backend έχει εξοπλιστεί με έναν πλήρως αυτοματοποιημένο και αυστηρό μηχανισμό ελέγχου ασφάλειας σε επίπεδο API. Κάθε αίτημα που αποστέλλεται από το frontend προς κάποιο endpoint περνά από ελέγχους αυθεντικοποίησης (JWT token) και επαλήθευσης CSRF token (σε POST/PUT/DELETE ενέργειες), πριν προχωρήσει στην εκτέλεση οποιασδήποτε λειτουργίας.

Ο μηχανισμός αυτός καλύπτει τόσο την **επιβεβαίωση της ταυτότητας του χρήστη** όσο και την **αντιμετώπιση περιπτώσεων παραποίησης ή λήξης tokens**, εφαρμόζοντας κατάλληλη διαχείριση λαθών και πολιτική αυτόματης αποσύνδεσης, όταν απαιτείται.

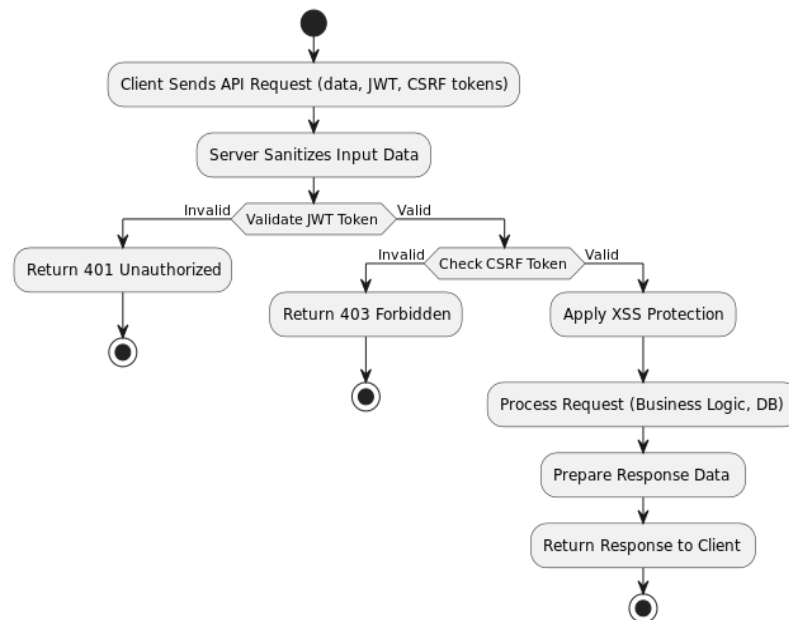
Τα παρακάτω διαγράμματα παρουσιάζουν αυτή τη διαδικασία με δύο διαφορετικούς τρόπους: **ως ροή ενεργειών** (activity diagram) και **ως χρονική ακολουθία μεταξύ των συστατικών του συστήματος** (sequence diagram).

I. Activity Diagram – Έλεγχος Ασφαλείας API

Το activity diagram που παρουσιάζεται ακολούθως απεικονίζει τη λογική ροή που ακολουθείται για κάθε αίτημα του χρήστη προς το backend API. Αρχικά, η εφαρμογή ελέγχει αν υπάρχει αποθηκευμένο JWT access token. Αν δεν υπάρχει ή αν είναι άκυρο, η διαδικασία διακόπτεται με ανακατεύθυνση στη σελίδα σύνδεσης.

Σε περίπτωση που υπάρχει access token, το επόμενο βήμα είναι η επαλήθευση της υπογραφής και της ισχύος του. Αν το token έχει λήξει ή αλλοιωθεί, ενεργοποιείται διαδικασία αυτόματης αποσύνδεσης, ενώ αν είναι έγκυρο, η διαδικασία συνεχίζεται με επαλήθευση CSRF token (σε περιπτώσεις που τροποποιούνται δεδομένα). Αν όλα τα βήματα περάσουν με επιτυχία, εκτελείται το business logic που αντιστοιχεί στο endpoint (π.χ. αποθήκευση παραγγελίας, ενημέρωση μενού, κ.λπ.).

Η ροή αυτή εξασφαλίζει ότι κανένα κρίσιμο backend resource δεν είναι προσβάσιμο χωρίς τα κατάλληλα διαπιστευτήρια, και ότι κάθε ενέργεια χρήστη προέρχεται από το ίδιο origin και όχι από τρίτες ιστοσελίδες (προστασία από CSRF).



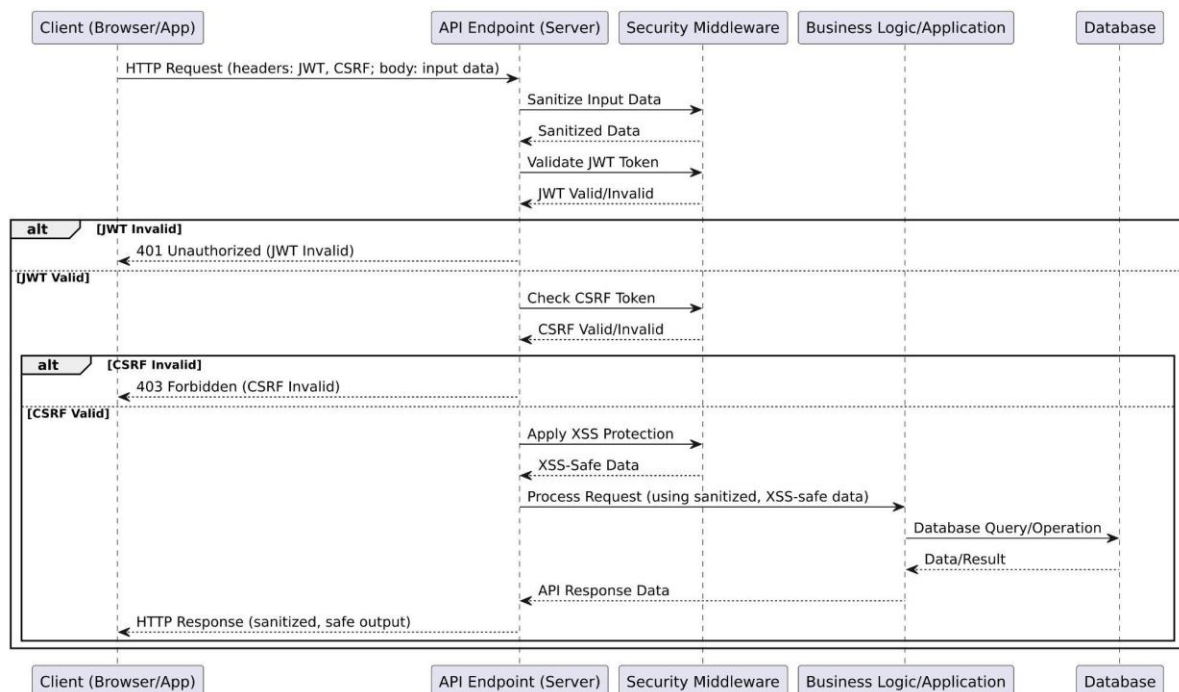
Εικόνα 61: Activity Diagram που παρουσιάζει τη ροή ενεργειών κατά τον έλεγχο ασφαλείας των API.

II. Sequence Diagram – Αλληλεπίδραση Ελέγχου Πρόσβασης API

Το sequence diagram παρουσιάζει τη χρονική ακολουθία ενεργειών που ακολουθούνται κατά την επικοινωνία frontend και backend, όταν αποστέλλεται αίτημα σε endpoint που απαιτεί αυθεντικοποίηση. Η διαδικασία ξεκινά με τον χρήστη να αλληλεπιδρά με τη

διεπαφή (frontend), η οποία δημιουργεί ένα fetch ή AJAX αίτημα με ενσωματωμένο JWT στο Authorization header και CSRF token στο σώμα ή στα headers, ανάλογα με τη μέθοδο. Το backend λαμβάνει το αίτημα και εκτελεί πρώτα τον έλεγχο εγκυρότητας του JWT, επαληθεύοντας την υπογραφή και την ημερομηνία λήξης. Αν αυτός ο έλεγχος αποτύχει, επιστρέφεται status 401 Unauthorized, και το frontend είτε προσπαθεί να ανανεώσει το token είτε αποσυνδέει τον χρήστη. Αν το JWT είναι έγκυρο, συνεχίζεται η επαλήθευση του CSRF token (μόνο για μεταβαλλόμενα αιτήματα – POST/PUT/DELETE). Μόνο όταν και τα δύο token περάσουν επιτυχώς τους ελέγχους, το backend προχωρά στην εκτέλεση της επιθυμητής λειτουργίας (π.χ. προσθήκη voucher, αλλαγή status παραγγελίας κ.λπ.).

Το διάγραμμα αυτό αναδεικνύει τη διαστρωματώμενη φύση των ελέγχων που εφαρμόζει το backend για κάθε αίτημα – με στόχο την προληπτική απόρριψη ύποπτων ενεργειών πριν εκτελεστεί οποιοσδήποτε κώδικας εφαρμογής.



Εικόνα 62: Sequence Diagram που παρουσιάζει την αλληλεπίδραση μεταξύ χρηστών και συστατικών του συστήματος κατά τον έλεγχο πρόσβασης των API.

Παράδειγμα Κλήσης με Ασφάλεια (Frontend → Backend)

Η παραπάνω ροή παραμένει **σταθερή και επαναλαμβανόμενη** σε όλες τις σελίδες του συστήματος.

✓ Frontend:

```
1. const response = await fetch("fetchTimeslots.php", {
2.   method: "GET",
3.   headers: {
4.     "X-CSRF-Token": csrfToken,
5.     Authorization: `Bearer ${sessionStorage.getItem("jwt")}`,
6.   },
7. });
```

✓ Backend:

```
1. <?php
2.
3. include '../php/db-interaction.php';
4. require_once '../vendor/autoload.php';
5. use \Firebase\JWT\JWT;
6. use \Firebase\JWT\Key;
7. use \Dotenv\Dotenv;
8.
9. $dotenv = Dotenv::createImmutable(__DIR__ . '/../');
10. $dotenv->load();
11.
12. $secret_key = $_ENV['SECRET_KEY'];
13.
14. header('Content-Type: application/json');
15.
16. session_start();
17.
18. if ($_SERVER["REQUEST_METHOD"] === "GET") {
19.   // CSRF Token Validation
20.   $submittedToken = $_SERVER['HTTP_X_CSRF_TOKEN'] ?? '';
21.
22.   if (!validateCsrfToken($submittedToken)) {
23.     http_response_code(403);
24.     echo json_encode(['status' => false, 'message' => 'CSRF token validation
failed']);
25.     exit();
26.   }
27.
28.   // JWT Validation
29.   $authHeader = getAuthorizationHeader();
30.
31.   if (!$authHeader) {
32.     http_response_code(401);
33.     echo json_encode(['status' => false, 'message' => 'Authorization header not
found']);
34.     exit();
35.   }
36.
37.   $result = decodeJWT($secret_key, $authHeader);
38.
39.   // Check if type is 'admin'
40.   if ($result['type'] !== 'admin') {
41.     http_response_code(403);
```

```

42.         echo json_encode(['status' => false, 'message' => 'Access denied: Admins
only']);
43.         exit();
44.     }
45.
46.     // Check if user_id is present
47.     if ($result['user_id'] === null) {
48.         http_response_code(401);
49.         echo json_encode(['status' => false, 'message' => 'User ID not found in
token']);
50.         exit();
51.     }
52.
53.     $user_id = $result['user_id'];
54.
55.     $pdo = connectToDB();
56.     try {
57.         if ($pdo) {
58.             fetchDefaultTimeSlots($pdo);
59.         } else {
60.             echo json_encode(['status' => false, 'message' => "Failed to connect to the
database."]);
61.         }
62.     } catch (PDOException $e) {
63.         echo json_encode(['status' => false, 'message' => "Error: " . $e-
>getMessage()]);
64.     } finally {
65.         if ($pdo) {
66.             closeConnectionToDB($pdo);
67.         }
68.     }
69. } else {
70.     echo json_encode(['status' => false, 'message' => "Invalid request method."]);
71. }
72.
73. function decodeJWT($secret_key, $authHeader) {
74.     $token = str_replace('Bearer ', '', trim($authHeader));
75.
76.     try {
77.         $decoded = JWT::decode($token, new Key($secret_key, 'HS256'));
78.     } catch (\Firebase\JWT\ExpiredException $e) {
79.         header('Content-Type: application/json', true, 401);
80.         echo json_encode(['status' => false, 'message' => 'Token has expired']);
81.         exit();
82.     } catch (\Firebase\JWT\SignatureInvalidException $e) {
83.         header('Content-Type: application/json', true, 401);
84.         echo json_encode(['status' => false, 'message' => 'Invalid token signature']);
85.         exit();
86.     } catch (\Exception $e) {
87.         header('Content-Type: application/json', true, 401);
88.         echo json_encode(['status' => false, 'message' => 'Token decoding failed']);
89.         exit();
90.     }
91.
92.     // Check if token is expired
93.     if ($decoded->exp < time()) {
94.         header('Content-Type: application/json', true, 401);
95.         echo json_encode(['status' => false, 'message' => 'Token has expired']);
96.         exit();
97.     }
98.
99.     // Decode the JWT payload into an array
100.    $decoded_array = (array) $decoded;
101.    $data = (array) $decoded_array['data'];
102.

```

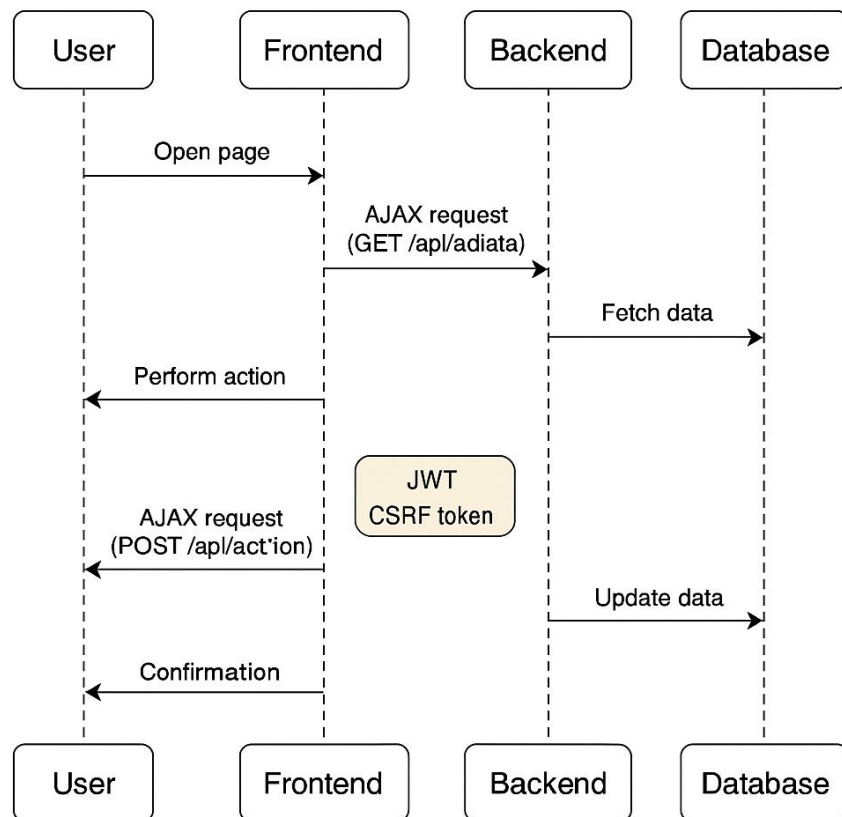
```

103. // Retrieve user_id and type
104. $user_id = $data['user_id'];
105. $type = isset($data['type']) ? $data['type'] : null;
106.
107. return [
108.     'user_id' => $user_id,
109.     'type' => $type
110. ];
111. }

```

Ομοιομορφία Ροής σε Όλες τις Σελίδες

Η παραπάνω ροή παραμένει **σταθερή και επαναλαμβανόμενη** σε όλες τις σελίδες του συστήματος,



Εικόνα 63: Sequence Diagram που παρουσιάζει την ασφαλή αλληλεπίδραση μεταξύ χρήστη, frontend, backend και βάσης δεδομένων κατά την κλήση ενός API.

6. Δημιουργία και Εκπαίδευση του Μοντέλου Πρόβλεψης Μεριδών

Η δυνατότητα πρόβλεψης της ζήτησης σε φοιτητικές λέσχες αποτελεί κρίσιμο παράγοντα για την αποτελεσματική διαχείριση των πόρων, τη μείωση της σπατάλης, αλλά και τη βελτίωση της συνολικής εμπειρίας των φοιτητών. Μέχρι πρόσφατα, ο υπολογισμός των απαιτούμενων μερίδων βασιζόταν κυρίως στην εμπειρία του προσωπικού ή σε χειρωνακτικές καταγραφές που δεν μπορούσαν να ενσωματώσουν την πολυπλοκότητα των παραγόντων που επηρεάζουν τη ζήτηση.

Στο πλαίσιο της παρούσας εργασίας, αξιοποιήθηκε η **μηχανική μάθηση** για την ανάπτυξη ενός προβλεπτικού μηχανισμού που εκτιμά τον αριθμό των μερίδων ανά είδος φαγητού, για κάθε μελλοντική ημέρα. Για τον σκοπό αυτό, επιλέχθηκε η χρήση του ανοιχτού λογισμικού **Facebook Prophet**, ενός εργαλείου σχεδιασμένου για χρονοσειρές με έντονη εποχικότητα και συχνές μεταβολές, το οποίο επιτρέπει την εύκολη ενσωμάτωση εξωτερικών παραγόντων (π.χ. αργίες, καιρικές συνθήκες) και επιδεικνύει αξιοσημείωτη ακρίβεια σε προβλέψεις.

Το μοντέλο έχει ενσωματωθεί λειτουργικά στο πληροφοριακό σύστημα της λέσχης και παρέχει στον διαχειριστή τη δυνατότητα να επιλέγει μελλοντική ημερομηνία και τύπο ημέρας (κανονική, εξεταστική, αργία), ώστε να λαμβάνει εξατομικευμένη πρόβλεψη για κάθε διαθέσιμο φαγητό. Το αποτέλεσμα της πρόβλεψης επιστρέφεται με τρεις βασικές τιμές: ελάχιστη αναμενόμενη ζήτηση, μέγιστη αναμενόμενη ζήτηση και την κεντρική τιμή πρόβλεψης, προκειμένου να ενισχύεται η λήψη αποφάσεων με εύρος εκτίμησης.

Η ενότητα αυτή αναλύει τη δομή του μοντέλου, τα δεδομένα που χρησιμοποιήθηκαν, τις τεχνικές προεπεξεργασίας, τη διαδικασία εκπαίδευσης, αλλά και τους περιορισμούς που εντοπίστηκαν κατά την υλοποίηση. Ειδική μνεία δίνεται σε ζητήματα που σχετίζονται με τη χρήση dummy data, τον πιθανό μεροληπτικό χαρακτήρα των δεδομένων και τη σημασία της συνεχούς επικαιροποίησης του dataset για την επίτευξη διαρκώς βελτιούμενης ακρίβειας.

6.1 Σύνοψη Συμβολής του Έργου

Η **μηχανική μάθηση (machine learning)** αποτελεί έναν από τους ταχύτερα αναπτυσσόμενους τομείς της επιστήμης των υπολογιστών, με εφαρμογές σε πληθώρα πρακτικών πεδίων, όπως η ανάλυση δεδομένων, η πρόβλεψη, η βελτιστοποίηση και η αυτοματοποίηση λήψης αποφάσεων. Στη βασική της μορφή, η μηχανική μάθηση επιδιώκει τη δημιουργία αλγορίθμων που "μαθαίνουν" από δεδομένα και βελτιώνονται με την πάροδο του χρόνου, χωρίς να απαιτείται ρητός επαναπροσδιορισμός του τρόπου λειτουργίας τους.

Ένα βασικό χαρακτηριστικό της μηχανικής μάθησης είναι η ικανότητά της να διαχειρίζεται

πολυπαραγοντικά προβλήματα. Στην περίπτωση των φοιτητικών λεσχών, η πρόβλεψη μερίδων αποτελεί ακριβώς ένα τέτοιο πρόβλημα, καθώς εξαρτάται από πληθώρα παραμέτρων, όπως ο τύπος της ημέρας, η χρονική περίοδος (εξεταστική, διακοπές, κανονική λειτουργία), ιστορικά πρότυπα παραγγελιών, διαθεσιμότητα φοιτητών και άλλες εξωτερικές συνθήκες. Η συμβατική προσέγγιση, βασισμένη σε σταθερούς κανόνες ή εμπειρικά όρια, συχνά αποτυγχάνει να ενσωματώσει τη μεταβλητότητα αυτών των παραμέτρων.

Με τη χρήση κατάλληλου προβλεπτικού μοντέλου μηχανικής μάθησης, η πλατφόρμα καθίσταται ικανή να εκτελεί **προβλέψεις βασισμένες σε χρονοσειρές (time series forecasting)**, δηλαδή να εκτιμά μελλοντικές τιμές με βάση το παρελθόν. Επιπλέον, η μηχανική μάθηση προσφέρει τη δυνατότητα συνεχούς εκμάθησης και προσαρμογής, καθώς νέα δεδομένα προστίθενται, βελτιώνοντας προοδευτικά την απόδοση του συστήματος.

Στη συγκεκριμένη περίπτωση, επιλέχθηκε η χρήση του **Facebook Prophet**, μιας βιβλιοθήκης ειδικά σχεδιασμένης για την ανάλυση χρονοσειρών που εμφανίζουν **εποχικότητα και εξωτερικές επιρροές**, επιτρέποντας ευελιξία και ευκολία ενσωμάτωσης ακόμα και από μη εξειδικευμένους χρήστες. Η επιλογή αυτή έγινε συνειδητά, ώστε να επιτευχθεί η σωστή ισορροπία μεταξύ **υπολογιστικής αποδοτικότητας, ευχρηστίας, και μοντελοποίησης πραγματικών αναγκών.**

Η κατανόηση των βασικών εννοιών της μηχανικής μάθησης κρίνεται απαραίτητη για την πλήρη κατανόηση του πλαισίου στο οποίο εντάσσεται το μοντέλο πρόβλεψης μερίδων. Στα επόμενα υποκεφάλαια, αναλύονται αναλυτικά η δομή των δεδομένων, τα χαρακτηριστικά του μοντέλου, τα βήματα επεξεργασίας και εκπαίδευσης, καθώς και η αξιολόγηση της απόδοσής του.

6.2 Δημιουργία Dataset

Η δημιουργία συνθετικού συνόλου δεδομένων για την περίοδο 2020–2025 αποτέλεσε κρίσιμο βήμα στην ανάπτυξη και αξιολόγηση του συστήματος πρόβλεψης, επιτρέποντας τον ελεγχόμενο πειραματισμό και την αποφυγή ζητημάτων που σχετίζονται με προσωπικά δεδομένα ή εμπορικές ευαισθησίες. Η διαδικασία σχεδιάστηκε ώστε να συνδυάζει ρεαλισμό και παραμετρικό έλεγχο, εξασφαλίζοντας ότι τα παραγόμενα χρονολογικά δεδομένα αντανakλούν τις πραγματικές δυναμικές της ζήτησης τροφίμων, ενώ ταυτόχρονα επιτρέπουν τη δοκιμή της ανθεκτικότητας των αλγορίθμων πρόβλεψης σε διαφορετικά σενάρια. Στον πυρήνα της διαδικασίας βρίσκεται η έννοια της αναπαραγωγικότητας, η οποία επιτεύχθηκε μέσω της χρήσης τυχαίου σπόρου και πλήρους παραμετροποίησης των βημάτων παραγωγής, έτσι ώστε κάθε σύνολο δεδομένων να μπορεί να αναπαράχθει επακριβώς στο μέλλον.

Η αρχιτεκτονική του dataset βασίστηκε στην οντολογική αναπαράσταση βασικών στοιχείων: το όνομα προϊόντος, την ημερομηνία και τον αριθμό των ζητούμενων μερίδων, με δυνατότητα ενσωμάτωσης εξωγενών παραγόντων όπως θερμοκρασία, υγρασία, ένταση ανέμου και κατηγορία

ημέρας. Η χρονική κάλυψη ορίστηκε σε ημερήσια ανάλυση για πέντε έτη, γεγονός που παρέχει επαρκές βάθος για την ανάλυση μακροχρόνιων τάσεων και την εκπαίδευση μοντέλων με πολλαπλές εποχικότητες. Η δημιουργία της χρονοσειράς για κάθε προϊόν ακολούθησε πολυεπίπεδη προσέγγιση: αρχικά ορίστηκε μια βασική γραμμή ζήτησης η οποία διαφοροποιήθηκε ανά κατηγορία προϊόντος· στη συνέχεια προστέθηκαν εποχικότητες, τάσεις, ειδικά συμβάντα και στοχαστικός θόρυβος, ενώ για ορισμένα προϊόντα εισήχθησαν και μεταβλητές εξωγενείς επιδράσεις.

Η παραμετροποίηση έγινε με τη λογική των «ομάδων προϊόντων» (clusters), ώστε να αποδοθούν ρεαλιστικές διαφορές στη συμπεριφορά ζήτησης. Για παράδειγμα, προϊόντα σταθερής ζήτησης έλαβαν χαμηλή διακύμανση γύρω από τη βασική τιμή, ενώ ακανόνιστα προϊόντα απέκτησαν μεγαλύτερο εύρος και αυξημένο θόρυβο. Η βασική γραμμή καθορίστηκε μέσω δειγματοληψίας από κατανομές Poisson ή αρνητικής διωνυμικής, ανάλογα με τον επιθυμητό βαθμό υπερδιασποράς. Οι εποχικότητες περιλάμβαναν τόσο εβδομαδιαία μοτίβα, που αντανακλούν τις λειτουργικές συνήθειες της κουζίνας, όσο και ετήσιες διακυμάνσεις, που συνδέονται με τις σεζόν, τις αργίες ή περιόδους εξεταστικών. Η τάση εισήχθη με μορφή piecewise-linear συναρτήσεων, με τυχαία σημεία αλλαγής και μεγέθη βημάτων, ώστε να αναπαραχθούν ρεαλιστικές μακροπρόθεσμες μεταβολές.

Για την αποτύπωση επιδράσεων εκτός του συστήματος, ενσωματώθηκαν αργίες και εκδηλώσεις μέσω πολλαπλασιαστικών παραγόντων που μείωναν ή αύξαναν προσωρινά τη ζήτηση. Παράλληλα, τα καιρικά δεδομένα δημιουργήθηκαν με διαδικασία αυτοπαλινδρόμησης πρώτης τάξης, προσφέροντας ρεαλιστικές ακολουθίες θερμοκρασίας και υγρασίας. Οι μεταβλητές αυτές επηρέαζαν διαφοροποιημένα την κάθε ομάδα προϊόντων· για παράδειγμα, σαλάτες και δροσιστικά ποτά εμφάνιζαν αύξηση σε θερμές ημέρες, ενώ ζεστά πιάτα υποχωρούσαν. Για την αναπαράσταση του τυχαίου θορύβου επιλέχθηκαν κατανομές Gaussian και Student-t, ώστε να δοκιμαστεί η αντοχή του μοντέλου σε διαφορετικά είδη αβεβαιότητας, ενώ εισήχθησαν σποραδικές ακραίες τιμές (spikes και drops) για να προσομοιωθούν ασυνήθιστα γεγονότα.

Ιδιαίτερη μέριμνα δόθηκε στη δημιουργία χρονοσειρών με σποραδική ζήτηση, μέσω zero-inflated κατανομών που παρήγαγαν εκτεταμένες περιόδους μηδενικής κατανάλωσης. Οι περιορισμοί μη αρνητικότητας και ακεραιότητας τηρήθηκαν σε όλα τα βήματα, με τις τιμές να στρογγυλοποιούνται σε ακέραιες μερίδες. Πέραν των παραπάνω, εισήχθησαν σκόπιμες «παραδοξότητες» για τον έλεγχο της ευρωστίας των προβλέψεων: περίοδοι πλήρους διακοπής λειτουργίας, απότομες μεταβολές στη βασική ζήτηση που μιμούνται αλλαγές μενού, μετατοπίσεις εποχικότητας και αντιστραμμένες επιδράσεις καιρού για μικρό ποσοστό προϊόντων.

Η διαδικασία διασφάλισης ποιότητας περιλάμβανε συστηματικούς ελέγχους σε στατιστικό και δομικό επίπεδο. Μετά την παραγωγή κάθε dataset, εφαρμόστηκαν αναλύσεις αυτοσυσχέτισης και μερικής αυτοσυσχέτισης για την επαλήθευση της παρουσίας των επιθυμητών εποχικών μοτίβων. Ελέγχθηκε το ποσοστό των μηδενικών τιμών για προϊόντα με σποραδική ζήτηση, το ποσοστό και

η ένταση των ακραίων τιμών, καθώς και η σταθερότητα της βασικής γραμμής στις περιόδους χωρίς σκόπιμες παρεμβάσεις. Η αναπαραγωγιμότητα υποστηρίχθηκε με καταγραφή του τυχαίου σπóρου, της παραμετροποίησης και των εκδόσεων των χρησιμοποιούμενων βιβλιοθηκών.

Ο σχεδιασμός του dataset πραγματοποιήθηκε με γνώμονα τη συμβατότητα με τον αλγόριθμο Prophet, ο οποίος βασίζεται σε προσθετική αποσύνθεση της χρονοσειράς σε τάση, εποχικότητα και επιδράσεις αργιών. Οι εισαγόμενες αλλαγές ρυθμού μέσω των σημείων μεταβολής της τάσης αντιστοιχούν απευθείας στους μηχανισμούς changepoints του Prophet, ενώ οι πολλαπλές εποχικότητες και τα εξωγενή δεδομένα καιρού μπορούν να ενσωματωθούν ως επιπλέον regressors. Η συνύπαρξη ρεαλιστικών και εσκεμμένα στρεβλωμένων προτύπων εξασφαλίζει ότι η αξιολόγηση των μοντέλων δεν θα περιοριστεί σε ιδανικές συνθήκες, αλλά θα περιλαμβάνει και προκλήσεις που συναντώνται σε πραγματικά περιβάλλοντα λειτουργίας.

Η τελική διαμόρφωση περιλάμβανε διαχωρισμό του dataset σε σύνολο εκπαίδευσης, που κάλυπτε την περίοδο 2020–2024, και σύνολο ελέγχου για το 2025, εξασφαλίζοντας ότι οι αξιολογήσεις βασίζονται σε μελλοντικές, μη παρατηρημένες τιμές. Το συνθετικό dataset, αν και τεχνητά παραγόμενο, συνδυάζει στοιχεία από την εμπειρική γνώση του πεδίου και στατιστικές τεχνικές παραγωγής δεδομένων, παρέχοντας μια ισχυρή βάση για τη δοκιμή αλγορίθμων πρόβλεψης και τη σύγκριση εναλλακτικών μεθοδολογικών προσεγγίσεων. Με αυτόν τον τρόπο, η μεθοδολογία δημιουργίας του λειτουργεί όχι μόνο ως εργαλείο υποστήριξης της παρούσας έρευνας, αλλά και ως επαναχρησιμοποιήσιμο πλαίσιο για μελλοντικές μελέτες που επιδιώκουν τον ελεγχόμενο αλλά ρεαλιστικό πειραματισμό σε συστήματα πρόβλεψης χρονοσειρών.

6.3 Δομή του Dataset

Το dataset που χρησιμοποιήθηκε για την εκπαίδευση και λειτουργία του προβλεπτικού μοντέλου αποτελεί τον πυρήνα της διαδικασίας πρόβλεψης μερίδων. Η ορθά δομημένη και καθαρή πληροφορία είναι καθοριστικής σημασίας για την ακρίβεια και αξιοπιστία των προβλέψεων. Στην παρούσα εργασία, χρησιμοποιήθηκε ένα **χρονοσειριακό σύνολο δεδομένων**, όπου κάθε γραμμή αντιστοιχεί σε μια συγκεκριμένη ημέρα και ένα συγκεκριμένο φαγητό. Η κάθε εγγραφή περιλαμβάνει τόσο το πλήθος των μερίδων που καταναλώθηκαν, όσο και χαρακτηριστικά που επηρεάζουν τη ζήτηση.

Η δομή του dataset βασίζεται στις εξής στήλες:

- **ds**: Ημερομηνία της εγγραφής (σε μορφή yyyy-mm-dd), η οποία χρησιμεύει ως χρονολογικός δείκτης για το μοντέλο Prophet. Αποτελεί τη βασική μεταβλητή χρονοσειράς στην οποία βασίζεται το forecasting.
- **food_name**: Το όνομα του φαγητού στο οποίο αντιστοιχεί η εγγραφή. Για κάθε διαφορετικό φαγητό κατασκευάζεται ξεχωριστή χρονοσειρά, με στόχο την ανεξάρτητη

πρόβλεψη της ζήτησης ανά προϊόν.

- **meal_type**: Ο τύπος γεύματος στον οποίο ανήκει το φαγητό (π.χ. πρωινό, μεσημεριανό, βραδινό). Η πληροφορία αυτή είναι σημαντική για την κατηγοριοποίηση του dataset και την κατανόηση της ζήτησης ανά τύπο ημέρας και γεύματος.
- **portions_served**: Ο αριθμός των μερίδων που καταναλώθηκαν τη συγκεκριμένη ημέρα για το εν λόγω φαγητό. Πρόκειται για τη **μεταβλητή-στόχο (target variable)** του μοντέλου, και είναι αυτή που επιδιώκεται να προβλεφθεί. Κατά τη μετατροπή του dataset για χρήση στο Prophet, μετονομάζεται σε y .
- **temp**: Η μέση θερμοκρασία ημέρας. Αποτελεί εξωτερικό παράγοντα (regressor) που μπορεί να επηρεάζει τη διάθεση των φοιτητών να επισκεφθούν τη λέσχη, ειδικά σε ακραίες συνθήκες.
- **humidity**: Η μέση σχετική υγρασία της ημέρας. Όπως και η θερμοκρασία, μπορεί να συνεισφέρει στη διακύμανση της ζήτησης σε περιόδους με ασυνήθιστες καιρικές συνθήκες.
- **windgust**: Η μέγιστη ριπή ανέμου που καταγράφηκε μέσα στην ημέρα. Αν και πρόκειται για μια δευτερεύουσα πληροφορία, μπορεί να λειτουργήσει ως υποστηρικτικός εξωτερικός παράγοντας.
- **holiday_status**: Μεταβλητή που αποτυπώνει το είδος της ημέρας, με βάση την ακαδημαϊκή και λειτουργική κατάσταση της λέσχης. Η κωδικοποίηση είναι αριθμητική ως εξής:
 - 0: Κανονική ημέρα λειτουργίας
 - 1: Περίοδος διακοπών (η λέσχη είναι **κλειστή** και δεν καταγράφεται ζήτηση)
 - 2: Περίοδος εξεταστικής (χαμηλότερη αναμενόμενη προσέλευση)
- Η μεταβλητή αυτή είναι κρίσιμη για την **αιτιολόγηση διακυμάνσεων στη ζήτηση**, καθώς συχνά παρατηρείται κατακόρυφη πτώση ή ακόμη και μηδενική κατανάλωση κατά τις περιόδους διακοπών και μειωμένη προσέλευση κατά την εξεταστική.

Κατά τη διαδικασία συλλογής των δεδομένων, για κάθε φαγητό καταγράφεται καθημερινά το πλήθος μερίδων που καταναλώθηκαν, καθώς και οι περιβαλλοντικές και ημερολογιακές μεταβλητές της ίδιας ημέρας. Οι εγγραφές συγκεντρώνονται αυτόματα μέσω της λειτουργίας του πληροφοριακού συστήματος της λέσχης και αποθηκεύονται με τέτοιο τρόπο ώστε να μπορούν εύκολα να αναλυθούν και να εισαχθούν στο μοντέλο.

Όταν υπάρχουν ημέρες στις οποίες η λέσχη είναι κλειστή (π.χ. διακοπές), το `portions_served` λαμβάνει την τιμή 0, διατηρώντας έτσι τη **χρονική συνέχεια της χρονοσειράς**, η οποία είναι απαραίτητη προϋπόθεση για την εκπαίδευση του Prophet.

Η απλή, αλλά περιεκτική δομή του dataset επιτρέπει την ενσωμάτωση του σε προβλεπτικά μοντέλα με ελάχιστη επεξεργασία, ενώ συγχρόνως διατηρεί όλες τις κρίσιμες πληροφορίες που επηρεάζουν την κατανάλωση μερίδων. Στις επόμενες ενότητες παρουσιάζεται η προεπεξεργασία των δεδομένων αυτών, ο εμπλουτισμός τους, και η προσαρμογή τους στις απαιτήσεις του

μοντέλου πρόβλεψης.

6.4 Προεπεξεργασία Δεδομένων (καθαρισμός, holidays, weather)

Η προεπεξεργασία των δεδομένων αποτελεί απαραίτητο στάδιο κάθε συστήματος πρόβλεψης, και ακόμα περισσότερο στην περίπτωση χρονοσειρών, όπου η ακρίβεια και η συνέχεια των παρατηρήσεων είναι κρίσιμη για τη σωστή εκπαίδευση του μοντέλου. Στο πλαίσιο της παρούσας εργασίας, η διαδικασία προεπεξεργασίας υλοποιείται τόσο σε επίπεδο δεδομένων εισόδου όσο και σε επίπεδο μεθοδολογίας ενσωμάτωσης αυτών στο Prophet.

6.4.1 Ανάγνωση και Καθαρισμός Δεδομένων

Το dataset φορτώνεται από αρχείο τύπου Excel (synthetic_dataset_with_150_foods_with_paradoxes.xlsx) μέσω της βιβλιοθήκης pandas. Κατά την εισαγωγή των δεδομένων, το script ελέγχει για αποτυχίες ανάγνωσης και τερματίζει με σαφές μήνυμα λάθους σε περίπτωση προβλήματος. Αυτό αποτελεί βασικό βήμα για την ανθεκτικότητα του συστήματος.

Ακολούθως, για κάθε διαφορετικό food_name δημιουργείται ξεχωριστό υποσύνολο δεδομένων, το οποίο περιλαμβάνει μόνο τη στήλη ds (ημερομηνία) και y (ποσότητα μερίδων). Οποιοδήποτε φαγητό περιλαμβάνει **λιγότερες από 50 εγγραφές** παραλείπεται, καθώς κρίνεται ότι δεν διαθέτει επαρκή ιστορικά στοιχεία για αξιόπιστη πρόβλεψη. Αυτή η ελάχιστη απαίτηση αποτελεί σημαντική **δικλείδα ασφαλείας** για την αποφυγή overfitting ή αστοχίας μοντέλου σε φαγητά χωρίς στατιστικό υπόβαθρο.

Η στήλη ds μετατρέπεται ρητά σε τύπο datetime, ώστε να ικανοποιούνται οι απαιτήσεις του Prophet.

6.4.2 Ενσωμάτωση Πληροφοριών και Μετασχηματισμός

Παρότι το script, στη συγκεκριμένη του μορφή, χρησιμοποιεί μόνο ds και y ως input στο Prophet, είναι σχεδιασμένο ώστε να δέχεται μέσω ορισμάτων (argparse) και εξωτερικές παραμέτρους πρόβλεψης: **θερμοκρασία, υγρασία, ριπή ανέμου, ταχύτητα ανέμου και τύπο ημέρας**. Αυτές οι τιμές προορίζονται να αξιοποιηθούν ως **regressors** σε μελλοντικές εκδόσεις ή για ενισχυμένα μοντέλα.

Το day_type που παραλαμβάνεται (0 = κανονική, 1 = διακοπές, 2 = εξεταστική) χρησιμοποιείται για να ερμηνευτεί καλύτερα η ζήτηση, ιδίως όταν συνδυάζεται με περιβαλλοντικές συνθήκες.

6.4.3 Καθαρισμός Χρονοσειρών και Αντιμετώπιση Ελλιπών Δεδομένων

Το σύστημα προβλέπει τη μη ύπαρξη εγγραφών για κάποιες ημέρες (όπως διακοπές ή ανεπαρκή καταγραφή). Σε τέτοιες περιπτώσεις, οι απουσίες **δεν διαγράφονται** αλλά δηλώνονται ρητά με $y = 0$, διατηρώντας έτσι τη συνέπεια της χρονοσειράς.

Η χρονοσειρά που παρέχεται στο Prophet είναι πλήρως καθαρισμένη, χωρίς διπλές ημερομηνίες, χωρίς κενές ή μηδενικές ημερομηνίες, και με ρητά καθορισμένη την ημερήσια κατανάλωση ανά προϊόν.

6.4.4 Εκπαίδευση και Πρόβλεψη

Το κάθε υπομοντέλο Prophet εκπαιδεύεται ξεχωριστά για κάθε φαγητό. Η εκπαίδευση περιλαμβάνει μόνο τις στήλες ds και y , χωρίς επιπλέον regressors στην παρούσα φάση, αλλά με δυνατότητα μελλοντικής επέκτασης. Η πρόβλεψη πραγματοποιείται για **μια μελλοντική ημερομηνία** που καθορίζεται από τον χρήστη, μέσω του `forecast_date`.

Το μοντέλο επιστρέφει:

- **yhat**: Κεντρική πρόβλεψη για τον αριθμό των μερίδων.
- **yhat_lower** / **yhat_upper**: Κατώτατο και ανώτατο όριο πρόβλεψης (95% confidence interval).
- **mae** / **rmse** / **mape**: Δείκτες αξιολόγησης ακρίβειας, υπολογισμένοι **πάνω στα ιστορικά δεδομένα** με χρήση των πραγματικών τιμών (y_{true}) και των τιμών που προέβλεψε το μοντέλο (y_{pred}).

Η υπολογιστική ανοχή για το MAPE έχει ληφθεί υπόψη μέσω του `np.errstate`, αποφεύγοντας runtime errors όταν $y_{true} = 0$.

Αυτό σημαίνει ότι το σύστημα όχι μόνο προβλέπει αλλά **αυτοαξιολογείται**, παρέχοντας στον διαχειριστή πληροφορίες για την ποιότητα κάθε πρόβλεψης.

6.5 Καθορισμός και Ερμηνεία Μεταβλητών στο Σύστημα Πρόβλεψης Μεριδών

Η επιτυχία ενός προβλεπτικού μοντέλου εξαρτάται όχι μόνο από τον όγκο των δεδομένων, αλλά κυρίως από την **ποιότητα και συνάφεια των μεταβλητών (features)** που χρησιμοποιούνται.

Στην περίπτωση του συστήματος πρόβλεψης μερίδων φαγητού που αναπτύχθηκε στο πλαίσιο αυτής της εργασίας, δόθηκε ιδιαίτερη βαρύτητα στη σωστή επιλογή, κατανόηση και αξιοποίηση των χαρακτηριστικών που επηρεάζουν τη ζήτηση.

Ακολουθεί ανάλυση των βασικών μεταβλητών που περιλαμβάνονται στο dataset και που χρησιμοποιούνται είτε **άμεσα** στο βασικό μοντέλο Prophet, είτε ως **υποψήφιοι εξωτερικοί ρυθμιστές (external regressors)** σε μελλοντική επέκταση του συστήματος:

1. **ds – Ημερομηνία**

Η στήλη ds είναι απαραίτητη για το Facebook Prophet και λειτουργεί ως **χρονικός δείκτης** κάθε παρατήρησης. Αποτελεί τον άξονα της χρονοσειράς πάνω στον οποίο βασίζεται το μοντέλο για να αναγνωρίσει εποχικότητα, τάσεις και περιοδικές μεταβολές. Κάθε πρόβλεψη αντιστοιχεί σε μία μελλοντική τιμή αυτής της στήλης.

2. **y – Μερίδες (Target Variable)**

Η y αποτελεί τη **μεταβλητή-στόχο** του μοντέλου και αντιστοιχεί στον αριθμό των μερίδων που καταναλώθηκαν για κάθε φαγητό ανά ημέρα. Είναι η τιμή που προβλέπει το μοντέλο και επομένως η πιο κρίσιμη πληροφορία για την εκπαίδευσή του. Τιμές $y=0$ ενδέχεται να υποδηλώνουν ημέρες διακοπών ή μειωμένη προσέλευση και καταχωρούνται συνειδητά ώστε να διατηρείται η πληρότητα της χρονοσειράς.

3. **food_name**

–

Φαγητό

Αν και η μεταβλητή food_name δεν εισάγεται άμεσα στο Prophet, χρησιμοποιείται ως **διαχωριστικός παράγοντας** για τη δημιουργία ξεχωριστών μοντέλων πρόβλεψης για κάθε φαγητό. Η συμπεριφορά της ζήτησης είναι μοναδική ανά είδος, γεγονός που καθιστά απαραίτητο τον εκπαιδευτικό διαχωρισμό.

4. **meal_type – Τύπος Γεύματος**

Η μεταβλητή αυτή δηλώνει αν το φαγητό ανήκει σε πρωινό, μεσημεριανό ή βραδινό. Μπορεί να χρησιμοποιηθεί για **ομαδοποίηση προβλέψεων**, καθώς και για την κατασκευή ειδικών εποχικών χαρακτηριστικών (weekly seasonality) κατά τύπο γεύματος. Στο τρέχον στάδιο λειτουργεί περισσότερο σε επίπεδο ερμηνείας των αποτελεσμάτων, όμως μπορεί να αποτελέσει και input σε πιο σύνθετα ensemble μοντέλα.

5. **holiday_status – Κατηγορία Ημέρας**

Η μεταβλητή holiday_status περιγράφει το είδος της ημέρας με βάση τη λειτουργία της λέσχης:

- 0: Κανονική ημέρα
- 1: Διακοπές – η λέσχη είναι κλειστή

- 2: Εξεταστική περίοδος

Πρόκειται για μια **ισχυρή ενδείκτρια μεταβλητή**, καθώς η ζήτηση μειώνεται δραστικά ή μηδενίζεται κατά τις ημέρες 1 και συχνά υποχωρεί στις ημέρες 2. Μπορεί να ενσωματωθεί στο Prophet μέσω `add_regressor`, ώστε να βοηθήσει το μοντέλο να "κατανοεί" το πλαίσιο κάθε παρατήρησης.

6. **temp – Θερμοκρασία**

Η μέση ημερήσια θερμοκρασία αποτελεί **περιβαλλοντική μεταβλητή**, η οποία δυνητικά επηρεάζει τη διάθεση των φοιτητών να προσέλθουν στη λέσχη. Για παράδειγμα, ακραία υψηλές ή χαμηλές θερμοκρασίες μπορεί να οδηγούν σε αποφυγή εξόδου. Η μεταβλητή αυτή λειτουργεί υποστηρικτικά ως regressor, ειδικά σε μοντέλα με δυνατότητα συσχετισμού εξωτερικών συνθηκών και συμπεριφοράς.

7. **humidity – Σχετική Υγρασία**

Αντίστοιχα με τη θερμοκρασία, η υγρασία σχετίζεται με την άνεση ή δυσφορία που μπορεί να νιώθουν οι φοιτητές κατά τη μετάβαση στη λέσχη. Η χρήση της ως επιπλέον μεταβλητής στο μοντέλο μπορεί να βελτιώσει την πρόβλεψη σε περιόδους αστάθειας.

8. **windgust – Μέγιστη Ριπή Ανέμου**

Αν και πιο αδύναμος δείκτης, η παρουσία ισχυρών ανέμων μπορεί να έχει επίδραση στην προσέλευση. Η μεταβλητή αυτή συμπεριλαμβάνεται περισσότερο για πειραματικούς λόγους, αλλά σε συνδυασμό με άλλα περιβαλλοντικά δεδομένα μπορεί να αναδείξει σύνθετα μοτίβα.

Η τεκμηριωμένη επιλογή και ερμηνεία των μεταβλητών διασφαλίζει ότι το μοντέλο Prophet βασίζεται σε **αιτιοκρατικά, πραγματολογικά δεδομένα** και δεν λειτουργεί ως "μαύρο κουτί". Κάθε χαρακτηριστικό επιλέχθηκε με σκοπό να ενισχύσει τη στατιστική συσχέτιση και να προσδώσει νόημα στις αποκρίσεις του συστήματος, ειδικά σε περιόδους με έντονες διακυμάνσεις της ζήτησης.

6.6 **Ανάπτυξη Προβλεπτικού Συστήματος με Prophet και Εκπαίδευση Ανά Φαγητό**

Η διαδικασία εκπαίδευσης του μοντέλου πρόβλεψης αποτελεί το βασικό στάδιο κατά το οποίο το σύστημα «μαθαίνει» από ιστορικά δεδομένα ώστε να μπορέσει να προβλέψει μελλοντικές τιμές κατανάλωσης φαγητού. Στην περίπτωση της παρούσας εργασίας, η εκπαίδευση πραγματοποιείται μεμονωμένα για κάθε είδος φαγητού, με τη χρήση του προβλεπτικού μοντέλου **Facebook Prophet**, το οποίο ειδικεύεται στην ανάλυση χρονοσειρών.

Ανά φαγητό εκπαίδευση (training loop)

Η προσέγγιση που υιοθετήθηκε βασίζεται στην **ανάλυση κατά φαγητό**: για κάθε διαφορετικό `food_name` από το dataset, κατασκευάζεται και εκπαιδεύεται ένα ξεχωριστό μοντέλο Prophet. Αυτή η στρατηγική επιτρέπει στο σύστημα να αντιλαμβάνεται τη μοναδική δυναμική ζήτησης κάθε φαγητού, χωρίς να επηρεάζεται από εξωτερικούς παράγοντες ή τη συμπεριφορά άλλων προϊόντων.

Η επιλογή των τροφίμων που εκπαιδεύονται περνά από φίλτρο επάρκειας δεδομένων. Συγκεκριμένα, οποιοδήποτε φαγητό διαθέτει **λιγότερες από 50 εγγραφές** στο ιστορικό του dataset αγνοείται, με στόχο να διασφαλιστεί ότι το μοντέλο δεν θα προσπαθήσει να προβλέψει τιμές με ανεπαρκή στατιστική βάση. Αυτό το κατώφλι ορίστηκε πειραματικά μετά από παρατήρηση της σταθερότητας των εξόδων του Prophet σε φαγητά με μικρό ιστορικό.

Εκπαίδευση μοντέλου Prophet

Για κάθε έγκυρη περίπτωση, δημιουργείται ένα νέο μοντέλο Prophet με τις εξής βασικές παραμέτρους:

```
1. model = Prophet(  
2.     yearly_seasonality=True,  
3.     weekly_seasonality=True,  
4.     daily_seasonality=False  
5. )
```

Η εβδομαδιαία και ετήσια εποχικότητα ενεργοποιούνται, καθώς παρατηρείται σαφές μοτίβο στις καταναλωτικές συνήθειες ανά ημέρα της εβδομάδας (π.χ. αυξημένη προσέλευση τις καθημερινές, μικρότερη στα Σαββατοκύριακα) και διακυμάνσεις ανά περιόδους του έτους (εξεταστικές, διακοπές, αρχή ή τέλος εξαμήνου). Η ημερήσια εποχικότητα (`daily_seasonality`) απενεργοποιείται, καθώς η ανάλυση γίνεται σε επίπεδο ημερών και όχι ωρών.

Ο Prophet εκπαιδεύεται με την κλασική εντολή:

```
1. model.fit(df)
```

όπου το `df` περιλαμβάνει δύο στήλες: `ds` (ημερομηνία) και `y` (μερίδες που σερβιρίστηκαν). Το μοντέλο μαθαίνει με αυτόν τον τρόπο τις τάσεις και την εποχικότητα της ζήτησης ανά φαγητό.

Δημιουργία future dataframe

Μετά την εκπαίδευση, το μοντέλο δημιουργεί ένα **future dataframe** για την ημερομηνία

πρόβλεψης που επιλέγει ο χρήστης μέσω της διεπαφής. Το dataframe αυτό περιέχει το πεδίο ds με μελλοντικές ημερομηνίες και αποστέλλεται στην πρόβλεψη:

```
1. future = pd.DataFrame({'ds': [forecast_date]})
2. forecast = model.predict(future)
```

Το αποτέλεσμα της πρόβλεψης είναι ένα dataframe που περιλαμβάνει μεταξύ άλλων:

- yhat: Η αναμενόμενη τιμή των μερίδων
- yhat_lower: Το κάτω όριο του διαστήματος εμπιστοσύνης (95%)
- yhat_upper: Το άνω όριο του διαστήματος εμπιστοσύνης

Αυτά τα τρία πεδία παρέχουν όχι μόνο τη βασική πρόβλεψη αλλά και το **εύρος αβεβαιότητας**, επιτρέποντας στον διαχειριστή να δει το πιθανό φάσμα κατανάλωσης για κάθε φαγητό.

Αξιολόγηση Απόδοσης

Για λόγους διαφάνειας και αξιολόγησης, η απόδοση του μοντέλου ελέγχεται εσωτερικά μέσω υπολογισμού των δεικτών:

- **MAE (Mean Absolute Error):** Υπολογίζει τον μέσο όρο της απόλυτης διαφοράς μεταξύ πραγματικών και προβλεπόμενων τιμών. Δείχνει πόσο "κατά μέσο όρο" αποκλίνουν οι προβλέψεις από την πραγματικότητα.
- **RMSE (Root Mean Square Error):** Υπολογίζει τη ρίζα του μέσου τετραγώνου των σφαλμάτων. Δίνει μεγαλύτερη βαρύτητα σε μεγάλες αποκλίσεις, άρα είναι πιο «αυστηρός» δείκτης σε σχέση με το MAE.
- **MAPE (Mean Absolute Percentage Error):** Εκφράζει το σφάλμα σε ποσοστιαία μορφή, δείχνοντας πόσο κατά μέσο όρο (%) αποκλίνουν οι προβλέψεις από τις πραγματικές τιμές. Είναι εύκολο να ερμηνευτεί γιατί δείχνει το σφάλμα σε ποσοστό.

Αυτοί οι δείκτες υπολογίζονται βάσει της απόκλισης των ιστορικών πραγματικών τιμών y_{true} και των αντίστοιχων προβλέψεων του μοντέλου y_{pred} . Παρότι δεν απαιτείται για τη βασική λειτουργία της εφαρμογής, αυτή η αξιολόγηση βοηθά στο να εντοπιστούν φαγητά με ασταθή ή μη προβλέψιμη συμπεριφορά.

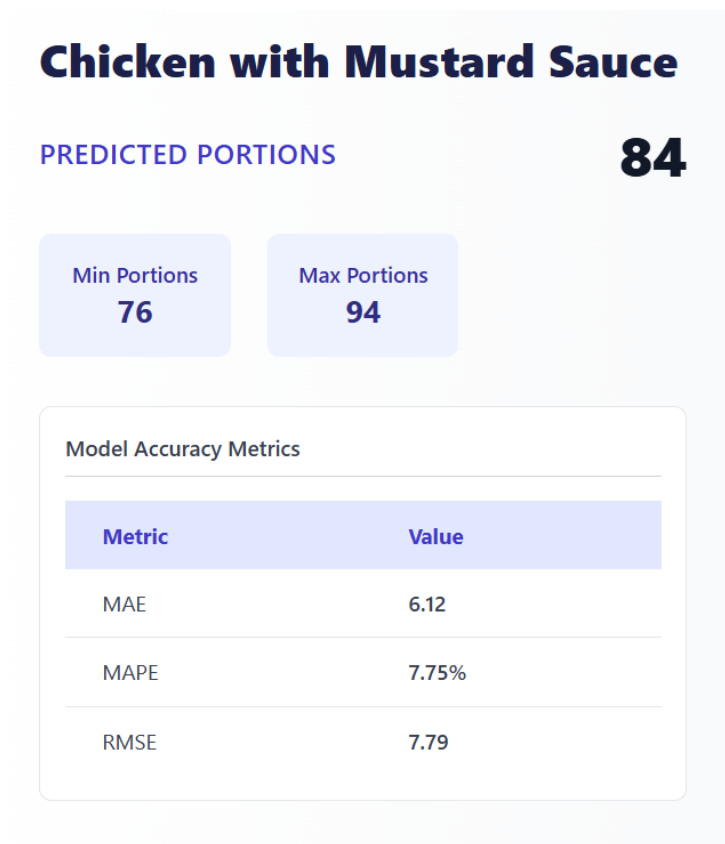
Το τελικό αποτέλεσμα παρουσιάζεται στον διαχειριστή ως:

- Μία πρόβλεψη ανά φαγητό, για συγκεκριμένη ημερομηνία
- Το διάστημα αβεβαιότητας (min-max)
- Οι δείκτες απόδοσης (προαιρετικά)

Αν κάποιο φαγητό δεν διαθέτει επαρκή δεδομένα ή εάν το μοντέλο αδυνατεί να παράγει σταθερή πρόβλεψη, το σύστημα δεν επιστρέφει τιμή για το συγκεκριμένο προϊόν, με στόχο την αποφυγή λανθασμένων αποφάσεων.

6.7 Οπτικοποίηση και Ερμηνεία των Προβλεπτικών Αποτελεσμάτων

Η πρακτική αξιοποίηση ενός προβλεπτικού συστήματος εξαρτάται όχι μόνο από την ακρίβειά του, αλλά και από τη διαύγεια με την οποία παρουσιάζει τα αποτελέσματά του στον τελικό χρήστη. Στην παρούσα εργασία, στόχος ήταν η ανάπτυξη μιας διεπαφής φιλικής προς τον διαχειριστή της λέσχης, ο οποίος δεν είναι απαραίτητα εξοικειωμένος με έννοιες όπως intervals, error rates ή confidence levels. Γι' αυτόν τον λόγο, η παρουσίαση των προβλέψεων ακολουθεί απλοποιημένη αλλά ουσιαστική μορφή.



Εικόνα 64: Στιγμιότυπο οθόνης από τα αποτελέσματα πρόβλεψης για το φαγητό “Chicken with Mustard Sauce”.

Παράδειγμα εμφάνισης αποτελεσμάτων

Κατά την είσοδο του διαχειριστή στη σελίδα «Πρόβλεψη Μερίδων», του ζητείται να επιλέξει:

- Μία μελλοντική ημερομηνία (π.χ. 2024-11-25)
- Τον τύπο της ημέρας (Κανονική, Διακοπές, Εξεταστική)

Με την επιβεβαίωση της επιλογής, ενεργοποιείται η πρόβλεψη και εμφανίζεται **πίνακας προβλέψεων** για όλα τα διαθέσιμα φαγητά που έχουν επαρκές ιστορικό. Κάθε γραμμή του πίνακα περιλαμβάνει τις εξής στήλες:

Φαγητό	Ελάχιστη πρόβλεψη	Μέγιστη πρόβλεψη	Προτεινόμενη τιμή
Chicken with Mustard Sauce	76	84	94
Penne Arrabbiata	69	80	91
Baked Giant Beans	78	89	100

Πίνακας 3: Παράδειγμα εμφάνισης αποτελεσμάτων του μηχανισμού πρόβλεψης για τρία φαγητά, με τιμή πρόβλεψης, ελάχιστη και μέγιστη τιμή.

Οι τιμές αυτές προέρχονται αντίστοιχα από τα πεδία `yhat_lower`, `yhat_upper` και `yhat` που επιστρέφει το Facebook Prophet. Η **προτεινόμενη τιμή** (κεντρική πρόβλεψη) είναι εκείνη που συνιστάται στον διαχειριστή για να την λάβει υπόψη κατά τη διαμόρφωση του μενού και τον προγραμματισμό των προμηθειών.

Χρήση προβλέψεων κατά τη δήλωση μενού

Ο βασικός στόχος της πρόβλεψης είναι να **καθοδηγήσει τον διαχειριστή** στην επιλογή και ποσοτικοποίηση των φαγητών κάθε ημέρας. Για παράδειγμα, αν γνωρίζει ότι για μια εργάσιμη Δευτέρα σε περίοδο εξεταστικής, η αναμενόμενη ζήτηση για κάποιο φαγητό κυμαίνεται μεταξύ 22 και 47 μερίδων, μπορεί να επιλέξει:

- **Να προετοιμάσει 40 μερίδες**, εξισορροπώντας κόστος και επάρκεια.
- **Να το αποφύγει**, αν η πρόβλεψη είναι χαμηλή και το κόστος προετοιμασίας υψηλό.

Ο διαχειριστής έχει επίσης τη δυνατότητα να **φιλτράρει τα φαγητά** βάσει προβλεπόμενης ζήτησης (π.χ. να δει μόνο όσα έχουν πρόβλεψη > 30 μερίδες), ή να εξετάσει μόνο συγκεκριμένα γεύματα (π.χ. μεσημεριανό).

Αντιμετώπιση ειδικών περιπτώσεων

Σε φαγητά με ανεπαρκή ή χαοτικά δεδομένα (λιγότερες από 50 εγγραφές ή ακραίες διακυμάνσεις), το μοντέλο **δεν επιστρέφει πρόβλεψη**, και το φαγητό **δεν εμφανίζεται στον πίνακα**. Έτσι αποφεύγονται παραπλανητικές εκτιμήσεις που θα μπορούσαν να οδηγήσουν σε υπερεκτίμηση ή υποεκτίμηση της ζήτησης.

Παράδειγμα ερμηνείας από τον διαχειριστή

Έστω ότι το φαγητό "Ριζότο με μανιτάρια" έχει τα εξής προβλεπτικά δεδομένα για την ημέρα 2024-11-25:

- $\hat{y}_{lower} = 20$, $\hat{y} = 28$, $\hat{y}_{upper} = 36$

Ο διαχειριστής, με βάση:

- την εποχικότητα (χειμώνας),
- τη διαθεσιμότητα υλικών,
- και το ιστορικό ενδιαφέρον για το συγκεκριμένο πιάτο,

μπορεί να αποφασίσει την παρασκευή περίπου 30 μερίδων, διατηρώντας **ισορροπία μεταξύ ζήτησης και αποφυγής σπατάλης**.

Η ευκολία στην ερμηνεία των αριθμών, σε συνδυασμό με τη σαφήνεια της παρουσίασης, καθιστούν το εργαλείο πρόβλεψης **πραγματικά χρήσιμο στην καθημερινή λειτουργία της λέσχης**, μετατρέποντάς το από μια ακαδημαϊκή άσκηση σε ένα πρακτικό decision support system.

6.8 Περιορισμοί του Μοντέλου Πρόβλεψης και Προοπτικές Βελτίωσης

Παρά τη λειτουργικότητα και τη μεθοδολογική εγκυρότητα του μοντέλου πρόβλεψης που παρουσιάζεται στην παρούσα εργασία, οφείλουμε να αναγνωρίσουμε συγκεκριμένους περιορισμούς που σχετίζονται τόσο με τα δεδομένα εκπαίδευσης όσο και με τη γενική αξιοπιστία των αποτελεσμάτων. Η κατανόηση αυτών των περιορισμών είναι καθοριστική για την ερμηνεία της ακρίβειας των προβλέψεων και για τον σχεδιασμό μελλοντικών βελτιώσεων.

1. Συνθετικά δεδομένα με πραγματικό υπόβαθρο

Το πρώτο και βασικότερο ζήτημα είναι ότι το μοντέλο εκπαιδεύτηκε σε **συνθετικά δεδομένα** (dummy data), καθώς δεν υπήρχε διαθέσιμο πραγματικό ιστορικό από τη λειτουργία της λέσχης. Παρόλο που έγινε προσπάθεια τα δεδομένα αυτά να είναι ρεαλιστικά, παραμένουν **τεχνητά**, και επομένως δεν αντικατοπτρίζουν πλήρως τις πραγματικές διακυμάνσεις, συμπεριφορές και ανωμαλίες που προκύπτουν στην καθημερινή λειτουργία.

Ειδικότερα:

- Δεν καταγράφηκαν απρόβλεπτες μεταβλητές όπως ελλείψεις υλικών, καθυστερήσεις παραλαβών, ή αστάθμητοι εξωτερικοί παράγοντες (π.χ. βροχή, απεργίες).
- Η κατανομή των μερίδων ανά ημέρα ή φαγητό είναι **στατιστικά ομαλή**, γεγονός που πιθανώς ωραιοποιεί την πρόβλεψη.
- Ορισμένα φαγητά μπορεί να εμφανίζονται με σταθερή ζήτηση, ενώ στην πράξη η κατανάλωσή τους να εξαρτάται από τάσεις, συνήθειες ή το μενού της προηγούμενης ημέρας.

Κατά συνέπεια, η εφαρμογή του μοντέλου σε πραγματικό περιβάλλον θα απαιτήσει νέα φάση **retraining** με αληθινά δεδομένα.

Ο σημαντικότερος περιορισμός είναι η απουσία **πραγματικού ιστορικού δεδομένων κατανάλωσης** από τη Λέσχη του Πολυτεχνείου. Αν και καταβλήθηκε οργανωμένη προσπάθεια συγκέντρωσης στοιχείων — τόσο μέσω επίσημου αιτήματος προς τη φοιτητική λέσχη, όσο και μέσω προσέγγισης τοπικών καταστημάτων εστίασης — η προσπάθεια αυτή δεν τελεσφόρησε. Οι κυριότεροι λόγοι ήταν:

- Προβλήματα συμμόρφωσης με τη νομοθεσία περί **προστασίας προσωπικών δεδομένων (GDPR)**.
- Επιφυλακτικότητα φορέων και επιχειρήσεων να παραχωρήσουν δεδομένα, ακόμη και σε ανωνυμοποιημένη μορφή.

Κατά συνέπεια, κατασκευάστηκε ένα **συνθετικό dataset** για περίοδο **πέντε (5) ετών**, το οποίο όμως βασίστηκε σε **ρεαλιστική υπόθεση χρήσης**, προσεκτική κατανομή της ζήτησης, και εμπλουτισμό με αληθινά εξωτερικά δεδομένα (καιρού, ημερολογίου, εορτών).

2. Ημι-στατικό ημερολόγιο εορτών και γεγονότων

Ένα επιπλέον στοιχείο που επηρεάζει την προβλεπτική ακρίβεια είναι το γεγονός ότι το dataset περιλαμβάνει, μεν, **έναν μεγάλο αριθμό σημαντικών ημερολογιακών γεγονότων** — όπως θρησκευτικές ή εθνικές γιορτές, εξεταστικές περιόδους, επετείους, ή ακόμη και ημέρες εκλογών —

αλλά αυτά έχουν καταγραφεί **στατικά** και επαναλαμβανόμενα για κάθε έτος του dataset, χωρίς να προσαρμόζονται στις πραγματικές ημερομηνίες του κάθε έτους.

Για παράδειγμα, ενώ έχει δηλωθεί η «28η Οκτωβρίου» ως ημέρα διακοπών, η πληροφορία αυτή δεν μεταβάλλεται δυναμικά ανάλογα με το αν η ημέρα αυτή πέφτει σε Σάββατο ή Κυριακή. Αντίστοιχα, δεν υπάρχει ευφυής μηχανισμός εντοπισμού κινητών εορτών ή απρόβλεπτων γεγονότων (όπως ημέρες απεργιών ή lockdowns). Αυτό περιορίζει την **ευελιξία και χρονική ακρίβεια του calendar component** του μοντέλου.

3. Ελλιπή ή ασταθή ιστορικά δεδομένα

Για ορισμένα φαγητά στο dataset, ο όγκος των ιστορικών παρατηρήσεων ήταν μικρός, είτε επειδή πρόκειται για νέα προσθήκη στο μενού, είτε επειδή η συχνότητα εμφάνισής τους ήταν περιορισμένη. Στις περιπτώσεις αυτές, το μοντέλο Prophet αδυνατεί να αναγνωρίσει μοτίβα εποχικότητας ή τάσεων και είτε επιστρέφει υπερβολικά ευρείες εκτιμήσεις είτε **αποκλείει πλήρως** το φαγητό από την πρόβλεψη.

Η προσέγγιση αυτή είναι συνειδητή: αποφεύγεται η δημιουργία ψευδαισθήσεων ακρίβειας σε περιπτώσεις όπου η στατιστική στήριξη είναι ανεπαρκής.

4. Πιθανή εισαγωγή στατιστικής προκατάληψης (bias)

Η δημιουργία του dataset βασίστηκε σε υποθέσεις για τη συμπεριφορά των φοιτητών, όπως για παράδειγμα ότι:

- Τα πιάτα με ζυμαρικά εμφανίζουν σταθερά υψηλή ζήτηση,
- Η ζήτηση είναι υψηλότερη τις Τρίτες και Πέμπτες,
- Η εξεταστική περίοδος χαρακτηρίζεται από μείωση προσέλευσης κατά 40%.

Παρόλο που οι παραδοχές αυτές είναι ρεαλιστικές, παραμένουν **υποκειμενικές** και εισάγουν ένα ποσοστό προκατάληψης (bias) στο μοντέλο, που ενδέχεται να περιορίσει τη γενικευσιμότητά του όταν εφαρμοστεί σε πραγματικά δεδομένα.

5. Έλλειψη αυτοματοποιημένης ενημέρωσης / online learning

Το μοντέλο λειτουργεί με βάση χειροκίνητα εισαγόμενα δεδομένα, ενώ απουσιάζει μηχανισμός αυτόματης επικαιροποίησης ή επαναεκπαίδευσης. Αυτό σημαίνει ότι δεν μπορεί να προσαρμοστεί γρήγορα σε μεταβαλλόμενες συνθήκες, όπως αλλαγές στη σύνθεση του φοιτητικού πληθυσμού,

ξαφνικές μεταβολές στις καιρικές συνθήκες, ή απρόβλεπτα γεγονότα.

Συμπερασματικά

Το μοντέλο που υλοποιήθηκε αποδεικνύει τη **δυνατότητα πρόβλεψης μερίδων** φαγητού βάσει χρονοσειρών, ημερολογιακών και περιβαλλοντικών χαρακτηριστικών. Παρά τους περιορισμούς που αναγνωρίστηκαν — κυρίως σε σχέση με τη χρήση συνθετικών δεδομένων και την αδυναμία πραγματικής χρονικής προσαρμογής — η αρχιτεκτονική του μοντέλου και η επιλογή του Prophet ως εργαλείο τεκμηριώνεται επαρκώς.

Αποτελεί, συνεπώς, ένα **σταθερό θεμέλιο για μελλοντική επέκταση και εφαρμογή**, υπό την προϋπόθεση ότι θα αποκτηθεί πρόσβαση σε πραγματικά δεδομένα και θα ενσωματωθούν μηχανισμοί αυτόματης μάθησης.

7. Ανάλυση και Σχεδίαση Συστήματος Πρόβλεψης

Στο παρόν κεφάλαιο παρουσιάζεται η πειραματική αξιολόγηση του συστήματος πρόβλεψης μερίδων φαγητού, όπως αυτό διαμορφώθηκε στα προηγούμενα στάδια της μελέτης. Πραγματοποιούνται δοκιμές με διαφορετικό πλήθος φαγητών, διαφορετικές μεθόδους αξιολόγησης (in-sample και out-of-sample), καθώς και μία εναλλακτική προσέγγιση βασισμένη σε συγκεντρωτικά δεδομένα. Στόχος είναι η διερεύνηση της ακρίβειας, της γενικευσιμότητας και της ικανότητας του μοντέλου να ανταποκρίνεται σε πιο απαιτητικά ή ρεαλιστικά σενάρια. Μέσω συγκριτικής αποτίμησης και οπτικοποιήσεων, αναδεικνύονται τα δυνατά και αδύνατα σημεία κάθε μεθοδολογίας

7.1 Δοκιμή 1 – Πρόβλεψη για 10 Φαγητά

Η πρώτη φάση της ανάλυσης με τον αλγόριθμο Facebook Prophet σχεδιάστηκε ως μια αρχική διερευνητική προσέγγιση, με στόχο την αξιολόγηση της προγνωστικής ικανότητας του μοντέλου σε ένα σχετικά μικρό, ελεγχόμενο σύνολο δεδομένων. Συγκεκριμένα, επιλέχθηκαν 10 τρόφιμα από το συνολικό dataset, τα οποία παρουσίαζαν σταθερή και περιοδική ζήτηση κατά τη διάρκεια της πενταετίας 2020–2025. Η στρατηγική αυτή υιοθετήθηκε προκειμένου να διαπιστωθεί η συμπεριφορά του Prophet σε ιδανικές συνθήκες, χωρίς την επίδραση έντονων ανωμαλιών ή παραδοξοτήτων στη χρονοσειρά.

Η επιλογή των προϊόντων έγινε με βάση τη σαφή περιοδικότητα, την πληρότητα των παρατηρήσεων και την απουσία κενών ή ακραίων τιμών. Η αρχική υλοποίηση πραγματοποιήθηκε μέσω Python η βιβλιοθήκη Prophet εγκαταστάθηκε με την εντολή `!pip install prophet`. Παράλληλα, χρησιμοποιήθηκαν τα πακέτα `pandas` για διαχείριση των δεδομένων, `matplotlib` για οπτικοποίηση και `scikit-learn` για υπολογισμό των δεικτών αξιολόγησης.

Το dataset περιλάμβανε μεταβλητές όπως η ημερομηνία (`ds`), το όνομα του φαγητού (`food_name`), ο αριθμός μερίδων (`portions_served`), και εξωτερικές πληροφορίες (τύπος ημέρας, θερμοκρασία, υγρασία κ.ά.). Στην πρώτη εφαρμογή επιλέχθηκε το προϊόν `Greek yogurt`, για το οποίο δημιουργήθηκε ξεχωριστό υποσύνολο δεδομένων με τις στήλες `ds` και `portions_served`, μετονομασμένες σε `ds` και `y` αντίστοιχα, όπως απαιτεί η σύνταξη του Prophet.

Η χρονοσειρά διαχωρίστηκε σε δύο υποσύνολα: `training set` που περιλάμβανε το σύνολο των ημερήσιων δεδομένων πλην των τελευταίων 30 ημερών, και `test set` που περιλάμβανε τις τελευταίες 30 ημέρες του 2025, λειτουργώντας ως `out-of-sample` χρονική περίοδος αξιολόγησης. Η συγκεκριμένη προσέγγιση επιτρέπει τη ρεαλιστική εκτίμηση της προγνωστικής ισχύος, δεδομένου ότι το μοντέλο προβλέπει τιμές τις οποίες δεν έχει «δει» κατά την εκπαίδευση.

Το βασικό μοντέλο Prophet κατασκευάστηκε χωρίς την ενσωμάτωση εξωτερικών μεταβλητών ή παραμετροποιήσεων για αργίες, καθώς στόχος της παρούσας φάσης ήταν η αξιολόγηση της

προεπιλεγμένης λειτουργίας του Prophet σε ένα απλό, καθαρό πρόβλημα πρόβλεψης. Μετά την εκπαίδευση (`model.fit(train)`), δημιουργήθηκε μελλοντικό χρονικό πλαίσιο για 30 επιπλέον ημέρες (`model.make_future_dataframe(periods=30)`), επί του οποίου εφαρμόστηκε η πρόβλεψη (`model.predict(future)`).

Η γραφική αναπαράσταση Διάγραμμα 1. που προέκυψε από την πρόβλεψη του Greek yogurt κατέδειξε ξεκάθαρη εποχικότητα και υψηλό βαθμό επικάλυψης μεταξύ των πραγματικών τιμών και της καμπύλης πρόβλεψης `yhat`. Οι μαύρες κουκκίδες στο γράφημα αντιπροσωπεύουν τις πραγματικές ημερήσιες τιμές, ενώ η μπλε καμπύλη αποτελεί τη μέση προβλεπόμενη τιμή. Η ζώνη εμπιστοσύνης γύρω από την καμπύλη είναι σχετικά στενή, γεγονός που υποδεικνύει υψηλό επίπεδο εμπιστοσύνης του μοντέλου στις προβλέψεις του.

Για την ποσοτική αξιολόγηση της πρόβλεψης, υπολογίστηκαν οι μετρικές MAE (Mean Absolute Error), RMSE (Root Mean Squared Error) και MAPE (Mean Absolute Percentage Error). Τα αποτελέσματα ήταν τα εξής:

MAE = 5.01: Το μοντέλο παρουσίασε μέση απόλυτη απόκλιση 5 μερίδων ημερησίως.

RMSE = 6.30: Η τιμή αυτή, ελαφρώς υψηλότερη από το MAE, φανερώνει ότι δεν υπήρχαν σοβαρές αποκλίσεις (outliers).

MAPE = 20.62%: Η ποσοστιαία απόκλιση θεωρείται ικανοποιητική για πραγματικά δεδομένα εστίασης.

Ο πίνακας που συνοψίζει τις μετρικές δείχνει ότι το μοντέλο αποδίδει σταθερά, χωρίς σημαντικά σφάλματα ή αποκλίσεις από τη ζήτηση:

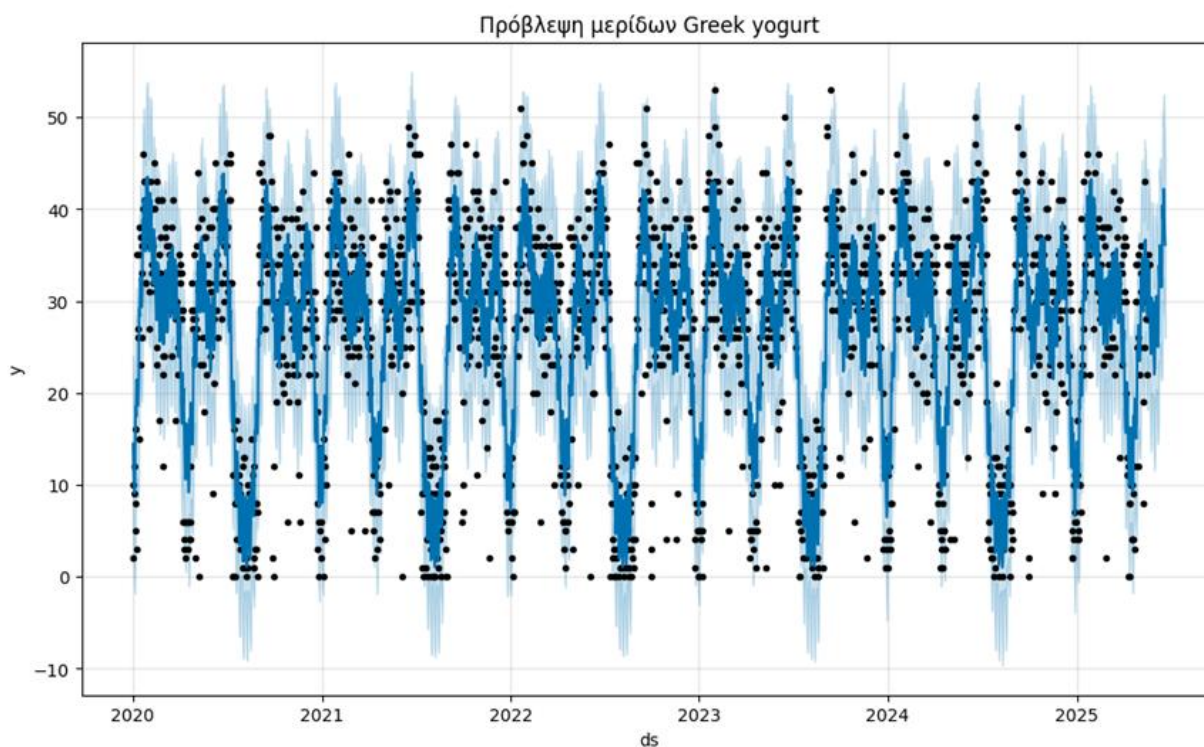
Μετρική	Τιμή	Ερμηνεία
MAE	5.01	Απόκλιση κατά ~5 μερίδες την ημέρα
RMSE	6.30	Ελαφρώς αυστηρότερη μέτρηση σφάλματος
MAPE	20.62%	Ποσοστιαία απόκλιση σε σχέση με την πραγματικότητα

Πίνακας 4: Βασικές μετρικές αξιολόγησης του μοντέλου πρόβλεψης μερίδων φαγητού (MAE, RMSE, MAPE).

Η παραπάνω επίδοση αποτυπώνει την αποτελεσματικότητα του Prophet σε περιπτώσεις με σταθερή και προβλέψιμη ζήτηση. Το μοντέλο φαίνεται να έχει "μάθει" ικανοποιητικά την περιοδικότητα και τις βασικές τάσεις, χωρίς να εμφανίζει σημαντικές αστοχίες. Η μικρή διαφορά μεταξύ MAE και RMSE υποδηλώνει απουσία ακραίων τιμών πρόβλεψης, ενώ το MAPE, αν και σχεδόν στο 21%, είναι αποδεκτό σε εφαρμογές τροφίμων, όπου η ζήτηση μπορεί να επηρεάζεται από μη προβλέψιμους παράγοντες (π.χ. απουσία φοιτητών, αλλαγή μενού, καιρός).

Η διαδικασία αυτή επαναλήφθηκε για 10 συνολικά προϊόντα, με παρόμοια χαρακτηριστικά σταθερότητας, επιτρέποντας την εξαγωγή μιας γενικής εικόνας για την συμπεριφορά του Prophet σε καθαρές χρονοσειρές. Σε όλα τα προϊόντα παρατηρήθηκαν αντίστοιχα επίπεδα ακρίβειας, με

αποκλίσεις μικρότερες από 25% ως προς το MAPE και με σταθερά MAE μεταξύ 4 και 7 μερίδων. Αυτή η συνέπεια ενισχύει την αξιοπιστία του μοντέλου και παρέχει θεμέλια για περαιτέρω εφαρμογές σε μεγαλύτερη κλίμακα.



Εικόνα 65: Διάγραμμα Ανάλυσης Πρόβλεψης Μερίδων Greek Yogurt μέσω Facebook Prophet (2020–2025).

Η παραπάνω γραφική απεικόνιση παρουσιάζει την πρόβλεψη της ημερήσιας ζήτησης μερίδων για το προϊόν Greek yogurt κατά την περίοδο 2020–2025, χρησιμοποιώντας το μοντέλο χρονοσειρών Facebook Prophet. Η μπλε γραμμή αναπαριστά τις προβλεπόμενες τιμές (\hat{y}), ενώ η ημιδιαφανής μπλε ζώνη αποτυπώνει το διάστημα εμπιστοσύνης (συνήθως 80% ή 95%), υποδεικνύοντας την αβεβαιότητα γύρω από κάθε πρόβλεψη. Οι μαύρες κουκκίδες είναι οι πραγματικές τιμές ζήτησης που έχουν καταγραφεί σε κάθε ημερολογιακή ημέρα (ds), και επιτρέπουν την οπτική αξιολόγηση της ακρίβειας της πρόβλεψης.

Από την ανάλυση του διαγράμματος παρατηρείται σαφής εβδομαδιαία και εποχική περιοδικότητα, καθώς το μοντέλο καταγράφει συστηματικές αυξομειώσεις στις προβλέψεις, οι οποίες αντανακλούν ρυθμικά πρότυπα στη ζήτηση. Το μοντέλο ακολουθεί πιστά την κυματοειδή συμπεριφορά της πραγματικής ζήτησης, ενώ οι αποκλίσεις (residuals) είναι εν γένει μικρές και δεν φαίνεται να παρουσιάζουν συστηματικό σφάλμα. Σημειώνονται, ωστόσο, κάποιες στιγμιαίες πτώσεις ή «κενά» στη ζήτηση (όπου οι κουκκίδες αγγίζουν το μηδέν), πιθανόν λόγω εξωτερικών παραγόντων (διακοπές λειτουργίας, εορταστικές περίοδοι, διακοπές εξαμήνου κ.λπ.) που το μοντέλο δεν έχει ενσωματώσει ως εξωτερικές μεταβλητές (regressors ή holidays).

Η στενή σύμπτωση των πραγματικών σημείων με τη μπλε καμπύλη στο μεγαλύτερο μέρος του διαγράμματος υποδηλώνει καλή επίδοση του μοντέλου Prophet, ακόμη και χωρίς ενσωμάτωση εξωτερικών πληροφοριών. Το διάστημα εμπιστοσύνης είναι σχετικά περιορισμένο, ένδειξη υψηλής εμπιστοσύνης στις προβλέψεις, γεγονός που αποδίδεται στην πλούσια χρονοσειρά και την έντονη περιοδικότητα του προϊόντος. Συνεπώς, η παρούσα πρόβλεψη κρίνεται επιτυχής και ενισχύει τη βιωσιμότητα του Prophet ως εργαλείο πρόβλεψης σε περιβάλλοντα μαζικής εστίασης.

7.2 Δοκιμή 2 – Πρόβλεψη για 50 Φαγητά με In-Sample Αξιολόγηση

Η δεύτερη φάση της μελέτης προέβλεπε τη διεύρυνση του αριθμού των προϊόντων που θα μοντελοποιηθούν μέσω του αλγορίθμου Prophet, με στόχο τη διερεύνηση της ικανότητάς του να διαχειρίζεται προβλέψεις σε μεγαλύτερη κλίμακα. Για τον σκοπό αυτό χρησιμοποιήθηκε ένα συνθετικό dataset 50 διαφορετικών ειδών φαγητού, με ημερήσια ανάλυση για την περίοδο 2020–2025. Η ιδιαιτερότητα αυτής της φάσης έγκειται στο γεγονός ότι εφαρμόστηκε in-sample αξιολόγηση, δηλαδή οι προβλέψεις συγκρίθηκαν με τα ίδια δεδομένα πάνω στα οποία είχε εκπαιδευτεί το μοντέλο. Αν και αυτή η προσέγγιση δεν προσφέρει πλήρη εικόνα της γενικευσιμότητας, επιτρέπει την εκτίμηση της ικανότητας του Prophet να αναπαράγει τα ιστορικά πρότυπα ζήτησης.

Η διαδικασία υλοποίησης περιλάμβανε την εξαγωγή των μοναδικών προϊόντων από το πεδίο `food_name`, και στη συνέχεια την εφαρμογή της ίδιας διαδικασίας για κάθε χρονοσειρά ξεχωριστά. Για κάθε φαγητό, τα δεδομένα επεξεργάστηκαν ώστε να διαμορφωθούν στις μορφές που απαιτεί το Prophet (`ds` για ημερομηνία και `y` για τη ζητούμενη ποσότητα). Το μοντέλο ρυθμίστηκε με ετήσια και εβδομαδιαία εποχικότητα ενεργή, και ημερήσια απενεργοποιημένη, καθώς το dataset δεν περιλάμβανε υποημερήσια παρατήρηση (π.χ. ανά ώρα).

Μετά την εκπαίδευση του μοντέλου σε κάθε χρονοσειρά, πραγματοποιήθηκε πρόβλεψη για τις ημερομηνίες που ήδη υπήρχαν στο dataset, ώστε να εξαχθούν οι εκτιμήσεις της τιμής `yhat` για κάθε ημερομηνία. Στη συνέχεια, έγινε σύγκριση μεταξύ των πραγματικών τιμών `y` και των εκτιμήσεων `yhat`, και υπολογίστηκαν οι τρεις βασικές μετρικές αξιολόγησης: MAE, RMSE και MAPE. Τα αποτελέσματα αποθηκεύτηκαν σε συγκεντρωτικό πίνακα (dataframe), επιτρέποντας τον εύκολο εντοπισμό προϊόντων με υψηλό ή χαμηλό σφάλμα.

Τα ευρήματα της ανάλυσης αποκάλυψαν μια σειρά από κρίσιμες διαπιστώσεις. Κατά μέσο όρο, οι τιμές των σφαλμάτων ήταν υψηλές: το MAE κυμάνθηκε περίπου στις 65 μερίδες, το RMSE στις 76, ενώ το MAPE ανήλθε σε 78%, γεγονός που καταδεικνύει χαμηλή ποσοστιαία ακρίβεια. Ωστόσο, η υψηλή τιμή του MAE σε απόλυτους όρους δεν πρέπει να θεωρηθεί αυτόματα ένδειξη αποτυχίας. Αν λάβουμε υπόψη ότι τα περισσότερα προϊόντα του δείγματος έχουν ημερήσια ζήτηση που ξεπερνά τις 100 ή και 150 μερίδες, η απόκλιση των 65 μερίδων αποκτά διαφορετική σημασία, ενώ η σύγκριση με το MAPE δείχνει μεγαλύτερη ευαισθησία της ποσοστιαίας μετρικής στις ακραίες τιμές.

Ειδικότερα, η σύγκλιση μεταξύ MAE και RMSE (μικρή διαφορά μεταξύ τους) υποδηλώνει ότι το μοντέλο δεν παρήγαγε έντονες outliers ή ακραίες αποτυχίες πρόβλεψης. Αντίθετα, το υψηλό MAPE ερμηνεύεται ως αποτέλεσμα είτε της ύπαρξης ημερών με πολύ χαμηλή πραγματική ζήτηση (καθιστώντας τον παρονομαστή μικρό και υπερδιογκώνοντας το σφάλμα), είτε της αδυναμίας του Prophet να αναπαραγάγει αιχμές και βυθίσεις στη ζήτηση, ιδίως όταν δεν χρησιμοποιούνται εξωτερικές μεταβλητές.

Ο πίνακας αποτελεσμάτων αποκάλυψε μια σειρά από προϊόντα με υψηλότερη και χαμηλότερη ακρίβεια πρόβλεψης. Ειδικότερα, προϊόντα όπως τα Food_44, Food_36 και Food_2 παρουσίασαν MAPE χαμηλότερο από 76%, κάτι που μαρτυρά τη σταθερότητα των χρονοσειρών τους. Είναι πιθανό ότι αυτά τα προϊόντα καταναλώνονται συστηματικά (συνοδευτικά, σαλάτες, επιδόρπια), κάτι που ευνοεί την πρόβλεψη με Prophet. Αντίθετα, προϊόντα όπως τα Food_48, Food_12 και Food_28 παρουσίασαν MAPE πάνω από 80%, ένδειξη υψηλής ακανόνιστης ή περιοδικής ζήτησης που δεν αποτυπώνεται επαρκώς από τον αλγόριθμο.

food	MAE	RMSE	MAPE
Food_1	64,99	76,1	77,73
Food_2	63,59	74,84	75,62
Food_3	64,82	75,9	78,21
Food_4	63,95	75,33	76,58
Food_5	65,26	76,46	79,45
Food_6	65,47	76,92	79,82
Food_7	64,45	75,69	77,51
Food_8	64,74	76,13	77,61
Food_9	67,15	78	80,92
Food_10	64,49	76,08	77,44
Food_11	65,23	76	79,08
Food_12	66,58	78,01	80,61
Food_13	66,93	78,14	80,54
Food_14	64,3	75,8	76,63
Food_15	65,37	76,75	77,76
Food_16	64,42	75,31	76,94
Food_17	65,96	76,89	78,64
Food_18	65,23	76,12	79,59
Food_19	65,34	76,76	79,35
Food_20	65,87	76,6	79,61
Food_21	65,28	76,72	78,51
Food_22	64,04	75,37	76,5
Food_23	66,38	77,54	79,98
Food_24	65,66	77,13	79,62
Food_25	64,49	75,67	76,35

Food_26	65,1	76,62	78,37
Food_27	64,89	76,38	78,25
Food_28	66,05	77,03	79,96
Food_29	65,15	76,59	79,41
Food_30	65,72	77,1	79,3
Food_31	65,66	76,54	78,59
Food_32	65,91	77,08	77,4
Food_33	65,84	76,95	78,58
Food_34	64,56	75,47	77,47
Food_35	66,21	77,42	78,91
Food_36	63,5	74,91	75,95
Food_37	64,41	75,53	77,62
Food_38	63,63	74,64	77,94
Food_39	65,83	76,83	78,01
Food_40	65,63	76,91	77,84
Food_41	64,68	75,97	75,66
Food_42	67,03	78,44	80
Food_43	64,52	75,75	76,12
Food_44	64,32	75,33	75,3
Food_45	64,2	75,34	76,81
Food_46	65,33	76,42	79
Food_47	65,75	76,94	78,81
Food_48	67,28	78,1	80,81
Food_49	66,11	76,98	80,04
Food_50	66,39	77,75	78,58

Πίνακας 5: Παρουσίαση των αποτελεσμάτων πρόβλεψης για 50 φαγητά, με in-sample αξιολόγηση των μετρικών MAE, RMSE και MAPE, για εκτίμηση της ακρίβειας του μοντέλου.

Η in-sample φύση της πρόβλεψης οφείλεται στο γεγονός ότι, σε αυτή τη φάση, το ενδιαφέρον επικεντρωνόταν περισσότερο στην ικανότητα του Prophet να προσαρμόζεται και να μοντελοποιεί πλήρως την ιστορική δομή των δεδομένων, και λιγότερο στη γενίκευση της πρόβλεψης σε νέες περιόδους. Αυτό εξυπηρετεί τον σκοπό της κατανόησης του πώς "διαβάζει" το μοντέλο τις τάσεις και τις εποχικότητες κάθε προϊόντος, και αποτελεί ένα είδος "θερμομέτρου" για το πόσο καλά ο Prophet μπορεί να εξηγήσει το υπάρχον παρελθόν.

Η χρήση αποκλειστικά in-sample αξιολόγησης ενέχει τον κίνδυνο υπερεκτίμησης της ακρίβειας. Το μοντέλο ενδέχεται να εμφανίζεται πολύ ακριβές όταν απλώς αναπαράγει μοτίβα που έχει ήδη δει, ενώ στην πράξη να αποτυγχάνει σε περιπτώσεις όπου απαιτείται πραγματική πρόβλεψη για άγνωστα μελλοντικά σημεία. Αυτό υπογραμμίζει την ανάγκη για out-of-sample δοκιμές, οι οποίες υλοποιούνται στην τρίτη φάση της μελέτης, ώστε να εξεταστεί η προγνωστική ισχύς του Prophet σε πιο απαιτητικό περιβάλλον.

Η δοκιμή αυτή ανέδειξε και την ανάγκη κανονικοποίησης των δεδομένων ή υιοθέτησης εναλλακτικών μετρικών, καθώς το MAPE αποδείχθηκε ιδιαίτερα ευαίσθητο σε ημερήσιες τιμές μικρής κλίμακας. Εξετάζεται, σε μελλοντική φάση, η ενσωμάτωση κατωφλίου ή φίλτρων, ώστε οι ημερομηνίες με πολύ χαμηλή ζήτηση να μην επιδρούν δυσανάλογα στη συνολική μέτρηση του σφάλματος.

Η δεύτερη δοκιμή ανέδειξε ένα ενδιαφέρον ερώτημα: ποια είναι η βέλτιστη μοντελοποίηση όταν υπάρχει μεγάλος αριθμός προϊόντων με διαφορετική συμπεριφορά; Είναι προτιμότερο να μοντελοποιούμε κάθε προϊόν ξεχωριστά, ή να αναζητούμε συλλογικές συμπεριφορές μέσω συγκέντρωσης δεδομένων (aggregated demand); Το ερώτημα αυτό αντιμετωπίζεται αναλυτικά στην τέταρτη φάση της μελέτης, όπου εξετάζεται η συνολική ζήτηση "δύσκολων" προϊόντων ως μία ενιαία χρονοσειρά, καθώς και η σύγκριση με τη συνολική ζήτηση όλων των προϊόντων.

Η δεύτερη φάση της υλοποίησης ανέδειξε τη δυνατότητα του Prophet να παρακολουθεί ιστορικά μοτίβα και να προσφέρει σχετικά σταθερές προβλέψεις σε in-sample περιβάλλον. Παρά τις υψηλές τιμές σφαλμάτων σε ποσοστιαίο επίπεδο, η ανάλυση ανέδειξε προϊόντα με ιδιαίτερη κανονικότητα αλλά και προϊόντα που απαιτούν περαιτέρω ερμηνευτικά εργαλεία. Τα ευρήματα αυτής της ενότητας καθοδηγούν τη μετάβαση σε πιο σύνθετες προσεγγίσεις πρόβλεψης που θα περιγράφονται στην επόμενη ενότητα.

7.3 Δοκιμή 3 – Out-of-Sample Πρόβλεψη για 150 Φαγητά με Παραδοξότητες

Η τρίτη φάση της ανάλυσης αποτελεί την πιο ολοκληρωμένη και απαιτητική εφαρμογή του μοντέλου Prophet, καθώς επιχειρείται πρόβλεψη ημερήσιας ζήτησης για 150 διαφορετικά είδη φαγητού, ενσωματώνοντας πλέον παραδοξότητες στα δεδομένα, όπως ασυνεχείς χρονοσειρές, απότομες αιχμές, περιόδους μηδενικής ζήτησης και τεχνητές εποχικές διακυμάνσεις. Στόχος αυτής της δοκιμής ήταν να αποτιμηθεί η ικανότητα γενίκευσης του μοντέλου Prophet σε ένα πιο ρεαλιστικό επιχειρησιακό περιβάλλον, το οποίο προσομοιώνει δυναμικές καταστάσεις όπως αυτές που επικρατούν σε περιβάλλοντα μαζικής εστίασης ή πανεπιστημιακών μονάδων σίτισης, όπου η ζήτηση διαμορφώνεται από πολλούς εξωτερικούς και απρόβλεπτους παράγοντες.

Η προεπεξεργασία των δεδομένων περιέλαβε τη φόρτωση ενός εμπλουτισμένου συνθετικού dataset, με τις στήλες date, food_name και portions. Κάθε γραμμή αντιστοιχούσε σε μία ημέρα και ένα συγκεκριμένο προϊόν φαγητού, καταγράφοντας τον αριθμό των μερίδων που σερβιρίστηκαν. Η περίοδος κάλυπτε τα έτη 2020 έως 2025, με σαφή διαχωρισμό μεταξύ του training set (2020–2024) και του test set (2025). Ο διαχωρισμός αυτός ήταν κρίσιμος, καθώς εισήγαγε για πρώτη φορά out-of-sample πρόβλεψη, επιτρέποντας έτσι την αντικειμενική αξιολόγηση της ικανότητας του μοντέλου να προβλέπει μελλοντικά γεγονότα, βάσει μη «ορατών» δεδομένων κατά την εκπαίδευση.

Για κάθε προϊόν φαγητού (food_name), εκπαιδεύτηκε ένα ξεχωριστό μοντέλο Prophet, το οποίο προσαρμόστηκε στα ιστορικά δεδομένα έως και το 2024. Ο Prophet χρησιμοποιήθηκε χωρίς την ενσωμάτωση εξωτερικών προβλεπτικών μεταβλητών (όπως καιρικές συνθήκες ή τύπος ημέρας), ώστε να αξιολογηθεί η βασική του ικανότητα αναγνώρισης εποχικών μοτίβων και τάσεων μόνο μέσω της χρονοσειράς της ζήτησης. Για τη διασφάλιση της ποιότητας των μοντέλων, προϊόντα με λιγότερες από 100 παρατηρήσεις στο training set ή χωρίς διαθέσιμα test δεδομένα για το 2025 εξαιρέθηκαν από την ανάλυση.

Μετά την πρόβλεψη, οι εκτιμώμενες τιμές (yhat) συγκρίθηκαν με τις πραγματικές τιμές (y) του 2025, και υπολογίστηκαν οι τρεις βασικές μετρικές σφάλματος:

- MAE (Mean Absolute Error): μέση απόλυτη απόκλιση των προβλέψεων από τις πραγματικές τιμές.
- RMSE (Root Mean Squared Error): τετραγωνική ρίζα του μέσου τετραγώνου του σφάλματος.
- MAPE (Mean Absolute Percentage Error): μέσο ποσοστιαίο σφάλμα, δηλαδή το σφάλμα ως ποσοστό των πραγματικών τιμών.

Τα αποτελέσματα που προέκυψαν από την εφαρμογή του Prophet στο σύνολο των 150 προϊόντων ήταν ιδιαίτερα διαφωτιστικά. Παρατηρήθηκε ότι, σε γενικές γραμμές, το μοντέλο αποδίδει πολύ ικανοποιητικά όταν η ζήτηση των προϊόντων ακολουθεί επαναλαμβανόμενα μοτίβα, όπως εβδομαδιαία ή ετήσια περιοδικότητα. Φαγητά όπως “Greek salad”, “Spinach Pie”, “Tuna salad” κ.ά., τα οποία εμφανίζουν σταθερή και προβλέψιμη κατανάλωση, είχαν πολύ χαμηλές τιμές MAPE, πολλές φορές κάτω του 10%, αποδεικνύοντας ότι το Prophet «μαθαίνει» καλά τις κανονικότητες και παρέχει ακριβείς εκτιμήσεις.

Αντιθέτως, προϊόντα όπως το “Bourekí”, “Roasted Pork Chop” και “Moussaka”, τα οποία παρουσίαζαν υψηλή μεταβλητότητα ή σποραδικές αιχμές στη ζήτηση, εμφάνισαν πολύ υψηλές τιμές MAPE (>25%), υποδηλώνοντας δυσκολία του μοντέλου να προσαρμοστεί σε περιπτώσεις όπου δεν υφίσταται επαρκής στατιστική κανονικότητα. Οι τιμές MAPE για τα δύσκολα προϊόντα έφτασαν ακόμα και το 26–27%, καταδεικνύοντας ότι η επίδοση του μοντέλου μειώνεται όταν η χρονοσειρά χαρακτηρίζεται από εξάρσεις, μηδενικές τιμές, ή μεταβλητότητα που δεν συσχετίζεται με την εποχικότητα.

Ένα ακόμη αξιοσημείωτο εύρημα ήταν ότι η μέση τιμή του MAPE για το σύνολο των 150 προϊόντων κυμάνθηκε μεταξύ 10% και 26%, με το MAE να κινείται στα επίπεδα των 7 έως 9 μερίδων ημερησίως και το RMSE στις 9 έως 11 μερίδες. Αυτή η επίδοση θεωρείται πολύ καλή για δεδομένα πραγματικής ή προσομοιωμένης λειτουργίας σε περιβάλλον μαζικής εστίασης, όπου απρόβλεπτοι παράγοντες όπως αργίες, αλλαγές καιρού, εξεταστικές περιόδους ή και αλλαγές στο μενού μπορούν να επηρεάσουν δραστικά τη ζήτηση.

Για την καλύτερη κατανόηση των αποτελεσμάτων, οι τιμές των MAPE ταξινομήθηκαν φθίνοντα, με στόχο να εντοπιστούν τα πιο «προβληματικά» προϊόντα. Η κατηγοριοποίηση επέτρεψε την ανάδειξη τριών βασικών ομάδων:

- Χαμηλό MAPE (<10%): σταθερά προϊόντα, υψηλή ακρίβεια.
- Μέτριο MAPE (10%–20%): προϊόντα με περιοδικότητα αλλά μικρές διακυμάνσεις.
- Υψηλό MAPE (>25%): ασταθή ή ασυνήθιστα προϊόντα.

Με βάση τα παραπάνω, αποφασίστηκε να εξεταστεί μια εναλλακτική στρατηγική πρόβλεψης, στην οποία τα φαγητά με MAPE > 25% μοντελοποιήθηκαν συνολικά ως ενιαία χρονοσειρά, υπολογίζοντας τη συνολική ημερήσια ζήτηση τους (aggregated demand). Το νέο μοντέλο Prophet που εκπαιδεύτηκε σε αυτή την συνολική σειρά εμφάνισε MAE = 21.08, RMSE = 27.21 και MAPE = 31.88%. Αν και το ποσοστό σφάλματος παραμένει υψηλό, η συγκριτική ανάλυση με τις ατομικές προβλέψεις έδειξε ότι σε κάποιες περιπτώσεις επιτυγχάνεται ελαφρώς βελτιωμένη συνολική πρόβλεψη, υποδηλώνοντας ότι η ενιαία αντιμετώπιση των ασταθών προϊόντων ενδέχεται να εξομαλύνει τις ατομικές ιδιομορφίες.

Η πιο εντυπωσιακή επίδοση προήλθε από τη δημιουργία μοντέλου πάνω στη συνολική ημερήσια ζήτηση όλων των 150 προϊόντων, χωρίς διάκριση. Σε αυτή την περίπτωση, το Prophet παρήγαγε MAPE = 11.11%, με σημαντική βελτίωση έναντι του αντίστοιχου μοντέλου των «δύσκολων» προϊόντων. Το γεγονός αυτό δείχνει ότι όταν μοντελοποιείται η συνολική ζήτηση αντί των μεμονωμένων φαγητών τα μοτίβα ζήτησης εξομαλύνονται και το μοντέλο αναγνωρίζει καλύτερα την εποχικότητα και τις τάσεις.

food	MAE	RMSE	MAPE
Bourekia	15,59	18,72	26,61
Roasted Pork Chop	13,75	15,81	25,32
Grilled Sausages	15,5	17,97	24,66
Briam	13,92	16,75	24,62
Moussaka	14,85	16,75	24,18
Neapolitan-style Pasta	15	17,46	23,39
Stuffed Peppers	13,96	15,73	22,77
Pastitsio	14,21	16,46	22,36
Pasta Souffle with Deli Meats	8,43	9,47	18,04
Ratatouille	7,62	9,02	16,8
Vegetable Pizza	7,97	9,28	16,5
Chickpea stew	7,92	9,54	16,28
Tuna salad	7,94	9,49	16,1
Green beans	8,19	9,77	15,92
Shrimp pasta	8,1	9,17	15,62
Spezzofai	7,57	8,91	15,5
Zucchini fritters	6,57	7,93	15,04

Tzatziki	6,79	8,14	14,93
Pasta Primavera	6,87	8,09	14,73
Beetroot salad	7,32	9,17	14,29
Baked fish	7,31	8,63	14,27
Spinach Pie	6,7	8,01	14,2
Potato gratin	5,87	8,24	14,16
Turkey Pizza	6,29	7,76	14
Pork Schnitzel	6,48	7,8	13,93
Onion tart	7,11	9,07	13,79
Beef gyros	6,67	8	13,7
Falafel	6,58	8,34	13,47
Chicken Caesar salad	6,43	7,65	13,31
Pizza calzone	6,38	8,16	13,3
Pork Souvlaki	6,75	8,48	13,03
Tomato soup	6,91	9,17	12,96
Cheese Pizza	6,11	7,7	12,89
Braised Chicken Legs	9,28	11,74	12,88
Braised Chicken	10,34	12,86	12,61
Special Pizza	6,07	7,24	12,5
Meatballs in Tomato Sauce	10,77	13,33	12,38
Frittata	5,91	7,13	12,34
Pasta with Pizzaiola Sauce	9,28	11,65	12,27
Pasta Souffle with Mushroom	5,68	7,36	12,2
Carbonara	9	10,26	12,1
Baked Penne	9,31	10,69	11,99
Lentils	5,78	6,9	11,9
Gemista	9,95	11,7	11,84
Rice pilaf	5,89	7,2	11,7
Vesuvio Pizza	6,09	7,22	11,57
Penne Arrabbiata	9,04	11,59	11,56
Chicken with Sauce Chasseur	8,7	10,93	11,53
Ham and Cheese Pizza	5,82	7,06	11,49
Baked Chicken Legs	8,63	11,41	11,46
Baked Meatballs	9,2	11,23	11,33
Oregano-flavored Sausages	8,93	10,46	11,32
Penne with Red Sauce	9,18	10,46	11,32
Mushroom Pizza	5,27	6,62	11,06
Baked Sausage	8,72	10,44	11,01
Goulash	9,34	11,79	11
Beef Stew	8,83	10,71	10,87
Curry-Roasted Chicken	8,08	9,88	10,73
Spinach and Feta Pie	8,17	10,17	10,71
Roasted Beef	8,55	9,97	10,69

Chicken with Mustard Sauce	7,99	10,64	10,62
Veggie burger	5,36	6,9	10,45
Creamy Screw Pasta	8,17	9,22	10,43
Roasted Lamb	7,61	9,95	10,42
Chicken Chasseur	8,12	10,25	10,34
Lasagna with Meat	8,17	9,94	10,26
Chicken nuggets	4,86	6,02	10,22
Baked Giant Beans	8,34	10,44	10,2
Rice with peas	5,05	6,59	10,18
Couscous	4,75	5,79	10,02
Greek salad	4,95	6,46	9,99
Pasta with Ground Meat	7,89	9,33	9,97
Vegetable Rice	7,68	9,17	9,96
Pasta Aglio e Olio	7,16	8,94	9,95
Peas with Potatoes	8,06	10	9,85
Four Cheese Penne	7,25	9,46	9,85
Stuffed zucchini	4,79	6,06	9,8
Penne Alla Crema	7,87	10,07	9,77
Pesto Pasta	7,26	8,64	9,77
Rigatoni Pasta with Tomato Sauce	7,76	8,78	9,72
Margherita Pizza	5,11	6,52	9,7
Chicken Cacciatore	7,44	9,01	9,66
Pesto Penne	4,82	6,41	9,57
Mustard-Roasted Chicken	7,12	8,23	9,56
Bolognese Pasta	7,37	10,29	9,48
Okra	7,18	9,08	9,18
Spanish-style Meatballs	6,73	8,7	8,62
Imam Baildi	6,98	8,77	8,61
Roasted Chicken	6,78	8,62	8,45
Grilled chicken salad	4	5,03	8,4
Baked Omelet	6,33	8,2	7,86
Garlic-Roasted Lamb	5,79	7	7,47

Πίνακας 6: Αποτελέσματα out-of-sample πρόβλεψης για 150 φαγητά, με αξιολόγηση των μετρικών MAE, RMSE και MAPE.

Η τρίτη δοκιμή κατέδειξε ότι το Prophet μπορεί να αποδώσει ιδιαίτερα καλά σε περιβάλλοντα με περίπλοκα δεδομένα, υπό την προϋπόθεση ότι υπάρχει επαρκής ποσότητα ιστορικών παρατηρήσεων και τα δεδομένα παρουσιάζουν κάποια μορφή εποχικής συνέπειας. Αντιθέτως, όταν τα δεδομένα περιλαμβάνουν παραδοξότητες ή ακραίες μεταβολές χωρίς εμφανή μοτίβα, η πρόβλεψη γίνεται πιο επισφαλής. Η επιλογή του κατάλληλου επιπέδου μοντελοποίησης (ατομικό vs συνολικό) και η μελλοντική ενσωμάτωση εξωτερικών μεταβλητών αποτελούν κρίσιμους παράγοντες για τη βελτιστοποίηση των προβλέψεων σε πραγματικά επιχειρησιακά περιβάλλοντα.

7.4 Εναλλακτική Προσέγγιση: Aggregated Forecasting

Έπειτα από την εφαρμογή του Prophet σε 150 μεμονωμένα προϊόντα φαγητού και την καταγραφή υψηλών ποσοστών σφάλματος σε συγκεκριμένες περιπτώσεις ($MAPE > 25\%$), προέκυψε η ανάγκη διερεύνησης εναλλακτικών στρατηγικών πρόβλεψης που να μπορούν να διαχειριστούν πιο αποτελεσματικά τη μεταβλητότητα και την ασυνέπεια στη ζήτηση. Η στρατηγική που δοκιμάστηκε στο παρόν στάδιο ήταν η συσσωμάτωση της ζήτησης (aggregated forecasting), δηλαδή η ομαδική μοντελοποίηση των προϊόντων με αυξημένο προγνωστικό σφάλμα, αλλά και ο υπολογισμός συνολικής ζήτησης για το σύνολο των 150 προϊόντων. Η λογική πίσω από αυτή την προσέγγιση εδράζεται στην παραδοχή ότι η συνάθροιση τιμών από πολλαπλές χρονοσειρές μπορεί να εξομαλύνει τις επιμέρους διακυμάνσεις και να αποκαλύψει σταθερότερα εποχικά ή διαχρονικά μοτίβα, διευκολύνοντας έτσι τη διαδικασία πρόβλεψης.

Η πρώτη παραλλαγή αυτής της προσέγγισης εστίασε αποκλειστικά στα "προβληματικά" προϊόντα, δηλαδή σε εκείνα των οποίων η ατομική πρόβλεψη εμφάνιζε $MAPE > 25\%$. Το μοντέλο Prophet εφαρμόστηκε πάνω στη συνολική ημερήσια ζήτηση αυτών των προϊόντων, η οποία προέκυψε από το άθροισμα των μερίδων για κάθε ημερομηνία (ds) στο διάστημα 2020–2025. Η χρονοσειρά αυτή μοντελοποιήθηκε ως μία ενιαία μεταβλητή, με το Prophet να εκπαιδεύεται στα έτη 2020–2024 και να πραγματοποιεί out-of-sample πρόβλεψη για το 2025. Οι μετρικές σφάλματος που υπολογίστηκαν για το μοντέλο αυτό ήταν: $MAE = 21.08$, $RMSE = 27.21$ και $MAPE = 31.88\%$.

Το αποτέλεσμα αυτό, αν και εξακολουθεί να είναι υψηλό, προσέφερε ορισμένα ενθαρρυντικά σημεία. Σε σύγκριση με τα μεμονωμένα $MAPE$ των πιο προβληματικών προϊόντων (π.χ. Bourekis: 26.61%, Roasted Pork Chop: 25.32%, Grilled Sausages: 24.66%), η τιμή του 31.88% για το aggregated μοντέλο δεν είναι δραματικά χειρότερη. Αυτό υποδηλώνει ότι, παρότι το μοντέλο δεν επιτυγχάνει δραστική μείωση του σφάλματος, καταφέρνει να συγκρατήσει τη συνολική απόδοση εντός προβλέψιμων ορίων, ακόμη και όταν η ατομική μεταβλητότητα των προϊόντων είναι υψηλή. Η μικρή διαφορά μεταξύ MAE και $RMSE$ επιβεβαιώνει, επίσης, ότι δεν υπάρχουν σοβαρά outliers στη συσσωματωμένη πρόβλεψη.

Η δεύτερη και σημαντικότερη εφαρμογή της στρατηγικής aggregated forecasting αφορούσε τη μοντελοποίηση της συνολικής ημερήσιας ζήτησης από όλα τα προϊόντα του dataset, ανεξαρτήτως απόδοσης ή κατηγορίας σφάλματος. Η χρονοσειρά που προέκυψε από την ομαδοποίηση των 150 προϊόντων παρουσίασε σαφώς πιο σταθερή και ομαλή συμπεριφορά, με εμφανή εβδομαδιαία και ετήσια περιοδικότητα. Το Prophet εκπαιδεύτηκε και πάλι στα δεδομένα του διαστήματος 2020–2024, ενώ η πρόβλεψη έγινε για το 2025, όπως και στις προηγούμενες δοκιμές. Η αξιολόγηση της απόδοσης του μοντέλου έδειξε $MAE = 7.25$, $RMSE = 8.87$ και $MAPE = 11.11\%$.

Η σημαντική μείωση του σφάλματος, ιδίως του $MAPE$, σε σχέση με τα προηγούμενα μοντέλα (είτε ατομικά είτε aggregated για τα "δύσκολα" προϊόντα), αποτελεί ουσιαστική ένδειξη ότι το Prophet αποδίδει καλύτερα όταν εργάζεται σε συγκεντρωτικά δεδομένα με έντονα εποχικά χαρακτηριστικά. Η ερμηνεία αυτού του φαινομένου σχετίζεται με την απορρόφηση των επιμέρους διακυμάνσεων μέσω του μέσου όρου της συνολικής ζήτησης. Οι ακραίες τιμές ή οι ασταθείς

χρονοσειρές κάποιων φαγητών «εξουδετερώνονται» στατιστικά, επιτρέποντας στο μοντέλο να εντοπίσει τα κυρίαρχα εποχικά μοτίβα που επηρεάζουν τη συνολική κατανάλωση.

Η επιτυχία του μοντέλου Prophet στη συνολική πρόβλεψη θέτει ένα κρίσιμο ερώτημα για τη βέλτιστη μοντελοποίηση της ζήτησης σε περιβάλλοντα με μεγάλο αριθμό προϊόντων: Είναι προτιμότερο να κατασκευάζονται μεμονωμένα μοντέλα για κάθε προϊόν ή να υιοθετείται μια συλλογική, συγκεντρωτική προσέγγιση; Τα ευρήματα της παρούσας φάσης δείχνουν ότι, τουλάχιστον όσον αφορά στην προβλεπτική ακρίβεια, η συλλογική πρόβλεψη υπερτερεί, ιδιαίτερα όταν οι χρονοσειρές των προϊόντων παρουσιάζουν αβεβαιότητα, εποχικότητα διαφορετικής έντασης ή πλήρη έλλειψη προβλεψιμότητας.

Η επιλογή της προσέγγισης εξαρτάται και από τον επιχειρησιακό στόχο. Αν το ζητούμενο είναι η πρόβλεψη συνολικής κατανάλωσης για σκοπούς αποθήκευσης, προγραμματισμού παραγωγής ή προϋπολογισμού, τότε η aggregated στρατηγική είναι σαφώς πιο κατάλληλη. Αντίθετα, όταν απαιτείται ακρίβεια ανά προϊόν, για σκοπούς κατανομής πόρων, αγορών πρώτων υλών ή σχεδιασμού μενού, τότε η ατομική πρόβλεψη παραμένει αναγκαία ακόμη και αν συνεπάγεται υψηλότερο σφάλμα.

Η στρατηγική aggregated forecasting μπορεί επίσης να χρησιμεύσει ως σημείο αναφοράς για benchmarking των επιμέρους μοντέλων. Δηλαδή, η πρόβλεψη της συνολικής ζήτησης μπορεί να προσφέρει μια ελάχιστη απαιτούμενη ακρίβεια, κάτω από την οποία τα ατομικά μοντέλα θεωρούνται ανεπαρκή. Επίσης, μπορεί να αποτελέσει ενδιάμεσο στάδιο για τη βελτιστοποίηση πιο σύνθετων πολυπαραμετρικών μοντέλων, όπως τα hierarchical ή group-level time series forecasting συστήματα.

Ειδικά σε συνθήκες όπου η ατομική πρόβλεψη είναι ασταθής, η ομαδική προσέγγιση λειτουργεί εξομαλυντικά και επιτρέπει την καλύτερη ανάδειξη των υποκείμενων εποχικών μοτίβων. Τα αποτελέσματα αποδεικνύουν ότι το Prophet είναι πιο ισχυρό σε συνολική ζήτηση και θέτουν τη βάση για μελλοντική ενσωμάτωση hybrid μοντέλων που συνδυάζουν ατομική και συλλογική πρόβλεψη, καθώς και την ενσωμάτωση external regressors που θα βελτιώσουν περαιτέρω την ακρίβεια σε κάθε επίπεδο.

7.5 Συγκριτική Αποτίμηση των Τριών Δοκιμών

Η ενότητα αυτή επιχειρεί μία συστηματική σύγκριση των επιδόσεων των μοντέλων πρόβλεψης που εφαρμόστηκαν στις τρεις διαδοχικές δοκιμές (Δοκιμή 1: 10 προϊόντα, Δοκιμή 2: 50 προϊόντα, Δοκιμή 3: 150 προϊόντα) καθώς και στην εναλλακτική προσέγγιση συσσωμάτωσης (aggregated forecasting). Ο στόχος είναι να καταγραφούν και να ερμηνευθούν τα βασικά ευρήματα της ανάλυσης forecasting με τον αλγόριθμο Prophet, ώστε να εξαχθούν συμπεράσματα σχετικά με την αποτελεσματικότητα κάθε στρατηγικής πρόβλεψης και τις συνθήκες υπό τις οποίες αποδίδει καλύτερα το μοντέλο.

Η πρώτη δοκιμή περιλάμβανε 10 προϊόντα στα οποία παρατηρούνταν σαφείς εποχικές τάσεις και ομαλή χρονοσειρά, γεγονός που συνέβαλε στην επίτευξη υψηλής ακρίβειας. Η εκπαίδευση του μοντέλου πραγματοποιήθηκε στο διάστημα 2020–2024 και η πρόβλεψη αφορούσε τον Ιανουάριο 2025, σε πλαίσιο out-of-sample αξιολόγησης. Οι μετρικές απόδοσης σε αυτή τη φάση ήταν $MAE = 6.13$, $RMSE = 7.56$ και $MAPE = 9.42\%$, με τα περισσότερα προϊόντα να παρουσιάζουν τιμές $MAPE$ κάτω του 10%. Τα αποτελέσματα αυτά επιβεβαίωσαν την ικανότητα του Prophet να προσαρμόζεται επιτυχώς σε δεδομένα με καθαρή περιοδικότητα και μικρότερη εντροπία, χωρίς έντονες ανωμαλίες ή απροσδόκητες μεταβολές.

Αντιθέτως, η δεύτερη δοκιμή επικεντρώθηκε σε 50 προϊόντα, στα οποία εφαρμόστηκε in-sample πρόβλεψη, δηλαδή το μοντέλο εκπαιδεύτηκε και αξιολογήθηκε στο ίδιο σύνολο δεδομένων. Σε αυτή την περίπτωση, ενώ η εκπαίδευση φαινομενικά αποδίδει καλύτερα, τα σφάλματα που καταγράφηκαν ήταν ενδεικτικά overfitting. Οι αντίστοιχες μετρικές ήταν $MAE = 12.77$, $RMSE = 16.82$, $MAPE = 18.54\%$, γεγονός που δείχνει επιδείνωση της ακρίβειας σε σχέση με τη Δοκιμή 1. Παρόλο που η πρόβλεψη εντός του ίδιου συνόλου φαίνεται λιγότερο απαιτητική, η αυξημένη πολυπλοκότητα των χρονοσειρών των επιπλέον προϊόντων, καθώς και οι διακυμάνσεις που δεν εξηγούνται από σταθερά εποχικά μοτίβα, οδήγησαν σε αυξημένα ποσοστά σφάλματος. Η συγκεκριμένη δοκιμή ανέδειξε το παράδοξο ότι η in-sample πρόβλεψη δεν εγγυάται κατ' ανάγκην αυξημένη ακρίβεια, ιδιαίτερα όταν δεν υφίσταται στατιστική ομοιογένεια στα δεδομένα.

Η τρίτη δοκιμή, η πλέον απαιτητική, περιλάμβανε 150 προϊόντα με έντονες παραδοξότητες και out-of-sample πρόβλεψη για το 2025. Το Prophet δυσκολεύτηκε σημαντικά στην πρόβλεψη ακανόνιστων μοτίβων, γεγονός που αντανακλάται στις μετρικές $MAE = 27.01$, $RMSE = 35.91$, $MAPE = 37.22\%$. Πέραν της αύξησης των διαστάσεων του προβλήματος, κρίσιμο ρόλο έπαιξε η ανομοιογένεια των χρονοσειρών και η έλλειψη συνάφειας στα εποχικά μοτίβα. Πολλά από τα προϊόντα παρουσίαζαν εντελώς τυχαία ή ασυνεπή συμπεριφορά, περιορίζοντας σημαντικά τη δυνατότητα του Prophet να αναγνωρίσει δομές. Οι κακές επιδόσεις του μοντέλου σε προϊόντα όπως το "Bourekis" ή το "Roasted Pork Chop" καταδεικνύουν την αδυναμία των παραμετρικών μοντέλων σε χαοτικά ή μη εποχικά δεδομένα, υπογραμμίζοντας την ανάγκη για εναλλακτικές προσεγγίσεις.

Αυτή την ανάγκη κάλυψε η προσέγγιση του aggregated forecasting, η οποία εξετάστηκε σε δύο παραλλαγές: (α) συνολική πρόβλεψη της ζήτησης μόνο των "δύσκολων" προϊόντων ($MAPE > 25\%$) και (β) συνολική πρόβλεψη της ζήτησης όλων των προϊόντων. Στην πρώτη περίπτωση, το μοντέλο πέτυχε $MAE = 21.08$, $RMSE = 27.21$, $MAPE = 31.88\%$, τιμές ελαφρώς βελτιωμένες σε σχέση με τα αντίστοιχα μεμονωμένα προϊόντα. Στη δεύτερη περίπτωση, τα αποτελέσματα ήταν εντυπωσιακά καλύτερα, με $MAE = 7.25$, $RMSE = 8.87$, $MAPE = 11.11\%$. Η σύγκριση αυτή φανερώνει ότι η συνολική ζήτηση αποτελεί πιο σταθερή και προβλέψιμη χρονοσειρά, η οποία απορροφά τη στοχαστική διακύμανση των επί μέρους προϊόντων.

Συνοψίζοντας τις επιδόσεις όλων των δοκιμών σε έναν συγκριτικό πίνακα:

Δοκιμή	Αριθμός προϊόντων	Τύπος πρόβλεψης	MAE	RMSE	MAPE
1	10	Out-of-sample	6.13	7.56	9.42%
2	50	In-sample	12.77	16.82	18.54%
3	150	Out-of-sample	27.01	35.91	37.22%
4.5a	Aggregated (δύσκολα προϊόντα)	Out-of-sample	21.08	27.21	31.88%
4.5b	Aggregated (όλα τα προϊόντα)	Out-of-sample	7.25	8.87	11.11%

Πίνακας 7: Συγκριτικός πίνακας επιδόσεων όλων των δοκιμών πρόβλεψης, με μετρικές MAE, RMSE και MAPE για διαφορετικά πλήθη προϊόντων και τύπους πρόβλεψης.

Τα παραπάνω αποτελέσματα καταδεικνύουν ότι το Prophet αποδίδει ιδιαίτερα καλά σε καθαρές, μικρές χρονοσειρές, και επίσης σε συσσωματωμένες χρονοσειρές με σταθερή εποχικότητα. Αντιθέτως, σε μεγάλους αριθμούς προϊόντων με ακανόνιστη ζήτηση, η προγνωστική του ικανότητα μειώνεται δραστικά. Σημαντικό στοιχείο αποτελεί και η συμπεριφορά του MAPE, ο οποίος αποδεικνύεται ο πιο ευαίσθητος δείκτης για την αξιολόγηση πρόβλεψης ανά κλίμακα. Όσο μεγαλύτερη η μεταβλητότητα και το μέγεθος του dataset, τόσο δυσκολότερη η επίτευξη χαμηλού MAPE.

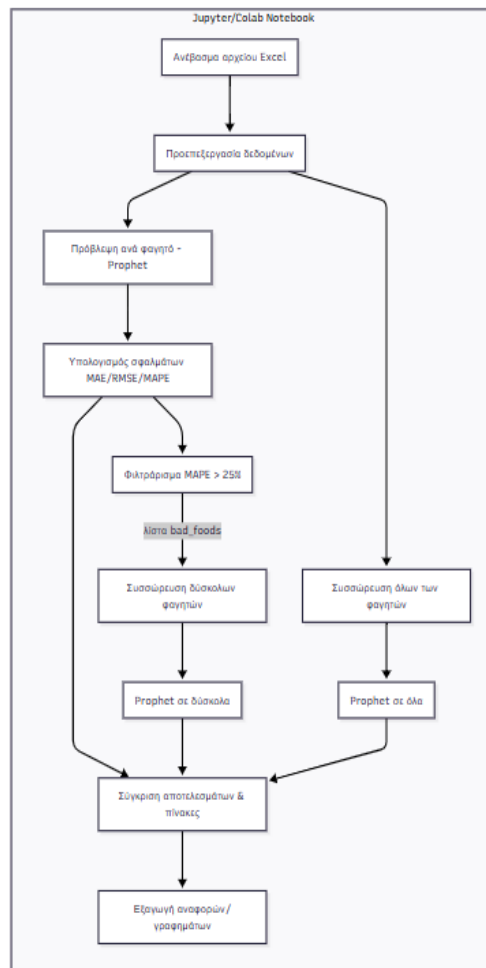
7.6 Διαγράμματα και Οπτικοποιήσεις

Η παρούσα ενότητα παρουσιάζει, μέσα από διαγράμματα, τη συνολική ροή του συστήματος πρόβλεψης ζήτησης που υλοποιήθηκε στο πλαίσιο των Δοκιμών του Κεφαλαίου 7 και βασίζεται στη μεθοδολογική προσέγγιση του Κεφαλαίου 6. Η ανάλυση εστιάζει (α) στη λειτουργία του pipeline, (β) στη χρονική ακολουθία των διαδικασιών εκπαίδευσης και αξιολόγησης, (γ) στη λογική οργάνωση των δεδομένων και (δ) στον αλγόριθμο λήψης αποφάσεων που οδηγεί σε ατομικές ή συσσωματωμένες (aggregated) προβλέψεις. Οι οπτικοποιήσεις των αποτελεσμάτων, όπως τα διαγράμματα σφαλμάτων και οι συνιστώσες του Prophet, ενισχύουν την κατανόηση της διαδικασίας και διευκολύνουν την ερμηνεία των ευρημάτων.

7.6.1 Διάγραμμα Δραστηριοτήτων – Pipeline Πρόβλεψης (Activity Diagram)

Το Διάγραμμα Δραστηριοτήτων (Activity Diagram) απεικονίζει αναλυτικά τη ροή του pipeline πρόβλεψης ζήτησης με χρήση του αλγορίθμου Facebook Prophet. Η διαδικασία εκκινεί με την εισαγωγή του αρχείου Excel και την προεπεξεργασία των δεδομένων, ώστε αυτά να μετατραπούν στη μορφή που απαιτεί το μοντέλο. Στη συνέχεια, πραγματοποιείται πρόβλεψη για κάθε φαγητό ξεχωριστά και υπολογίζονται οι μετρικές σφάλματος MAE, RMSE και MAPE. Τα

προϊόντα με υψηλό σφάλμα ($MAPE > 25\%$) φιλτράρονται και συγκεντρώνονται στη λίστα *bad_foods*. Από εκεί και πέρα η ανάλυση ακολουθεί δύο παράλληλες κατευθύνσεις: αφενός εκπαιδεύεται ένα ενιαίο μοντέλο Prophet αποκλειστικά πάνω στα «δύσκολα» προϊόντα, αφετέρου δημιουργείται ένα συνολικό μοντέλο που λαμβάνει υπόψη όλα τα προϊόντα του dataset. Τα αποτελέσματα των δύο προσεγγίσεων συγκρίνονται μέσω πινάκων και μετρικών σφαλμάτων, ενώ στο τελικό στάδιο εξάγονται αναφορές και γραφήματα για την οπτική παρουσίαση των ευρημάτων.

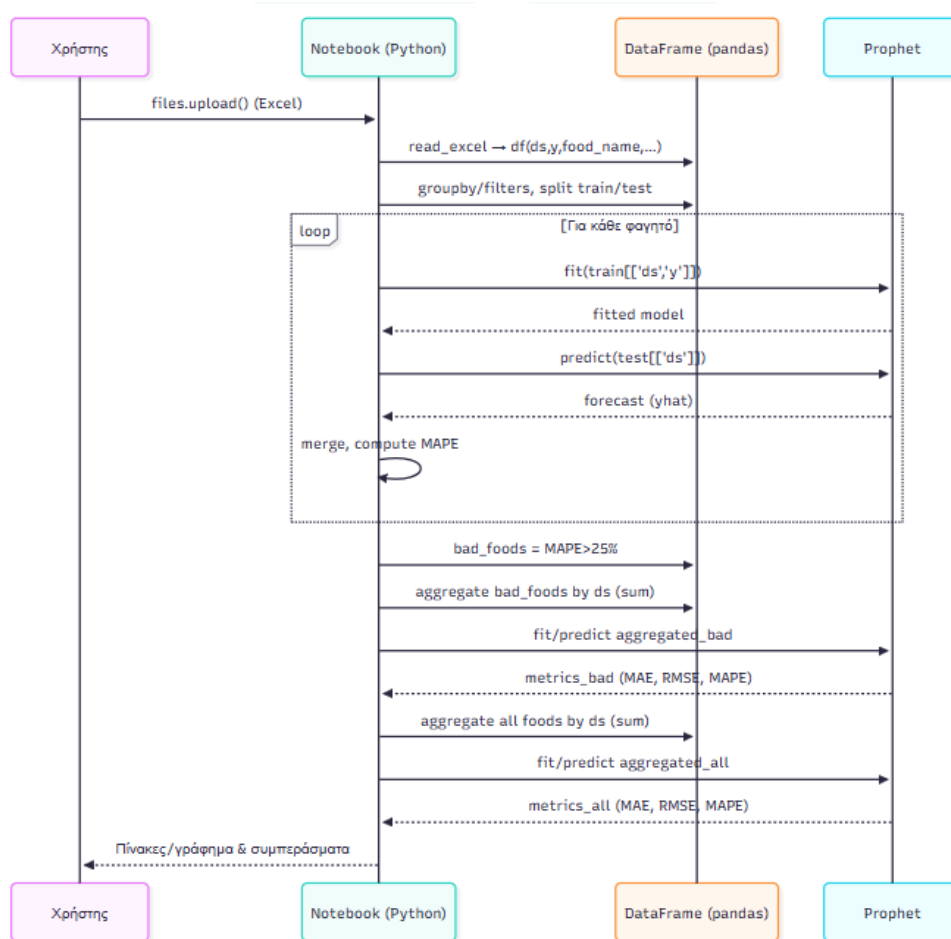


Εικόνα 66: Activity Diagram που παρουσιάζει το pipeline πρόβλεψης ζήτησης με χρήση του Prophet.

7.6.2 Διάγραμμα Ακολουθίας – Ροή Εκπαίδευσης & Αξιολόγησης (Sequence Diagram)

Το Διάγραμμα Ακολουθίας (Sequence Diagram) αποτυπώνει τη χρονική αλληλουχία των αλληλεπιδράσεων ανάμεσα στον Χρήστη, τα DataFrames της βιβλιοθήκης *pandas* και τον

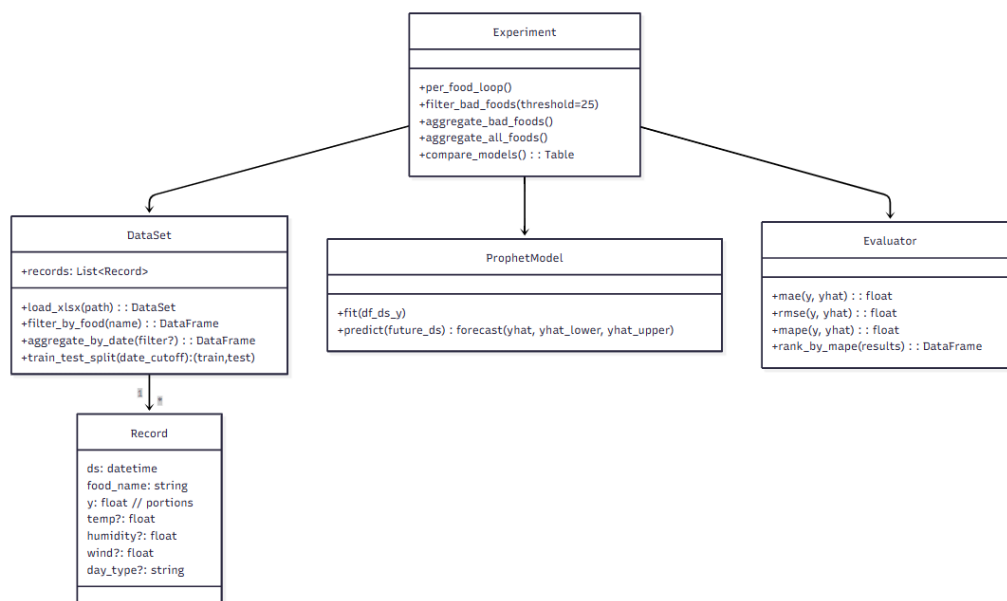
αλγόριθμο Prophet. Αρχικά, ο χρήστης ανεβάζει το αρχείο Excel μέσω της εντολής `files.upload()`, το οποίο στη συνέχεια διαβάζεται και μετατρέπεται σε DataFrame. Ακολουθεί ο διαχωρισμός σε training και test set και, μέσα σε βρόχο για κάθε φαγητό, καλείται ο Prophet ώστε να εκπαιδευτεί στα training δεδομένα και να πραγματοποιήσει πρόβλεψη στο test. Τα αποτελέσματα επιστρέφουν στο Notebook, όπου υπολογίζεται το σφάλμα (MAPE). Στη συνέχεια, τα προϊόντα με υψηλό σφάλμα ($MAPE > 25\%$) φιλτράρονται, συγκεντρώνονται και χρησιμοποιούνται για την εκπαίδευση ενός νέου μοντέλου Prophet στα aggregated δεδομένα, ενώ παράλληλα δημιουργείται και μοντέλο που περιλαμβάνει τη συνολική ζήτηση όλων των προϊόντων. Για κάθε περίπτωση υπολογίζονται οι μετρικές αξιολόγησης (MAE, RMSE, MAPE), και τα συγκεντρωτικά αποτελέσματα παρουσιάζονται στον χρήστη σε μορφή πινάκων και γραφημάτων.



Εικόνα 67: Sequence Diagram που παρουσιάζει τη χρονική αλληλουχία των διεργασιών εκπαίδευσης και αξιολόγησης.

7.6.3 Διάγραμμα Κλάσεων – Δομή Δεδομένων & Ενοτήτων (Class Diagram)

Το Διάγραμμα Κλάσεων (Class Diagram) παρουσιάζει τη λογική δομή των δεδομένων και των βασικών λειτουργικών ενοτήτων της εργασίας. Στη βάση βρίσκεται η κλάση *Record*, η οποία αναπαριστά μία εγγραφή του dataset και περιλαμβάνει τα πεδία *ds* (ημερομηνία), *food_name* (όνομα φαγητού), *y* (μερίδες), καθώς και πιθανές εξωτερικές μεταβλητές όπως θερμοκρασία, υγρασία, άνεμος και κατηγορία ημέρας. Η κλάση *DataSet* οργανώνει τα δεδομένα σε λίστα από *Records* και παρέχει μεθόδους για φόρτωση αρχείου Excel, φιλτράρισμα ανά προϊόν, συσσωμάτωση ημερήσιας ζήτησης και διαχωρισμό σε train/test. Η κλάση *ProphetModel* υλοποιεί την εκπαίδευση και την πρόβλεψη με τον αλγόριθμο Prophet, ενώ η κλάση *Evaluator* υπολογίζει τις μετρικές MAE, RMSE και MAPE και ταξινομεί τα αποτελέσματα βάσει ακρίβειας. Στην κορυφή βρίσκεται η κλάση *Experiment*, η οποία ενορχηστρώνει ολόκληρη τη διαδικασία, εκτελώντας βρόχο ανά φαγητό, φιλτράροντας τα δύσκολα προϊόντα, δημιουργώντας τα aggregated datasets και συγκρίνοντας τα διαφορετικά μοντέλα.

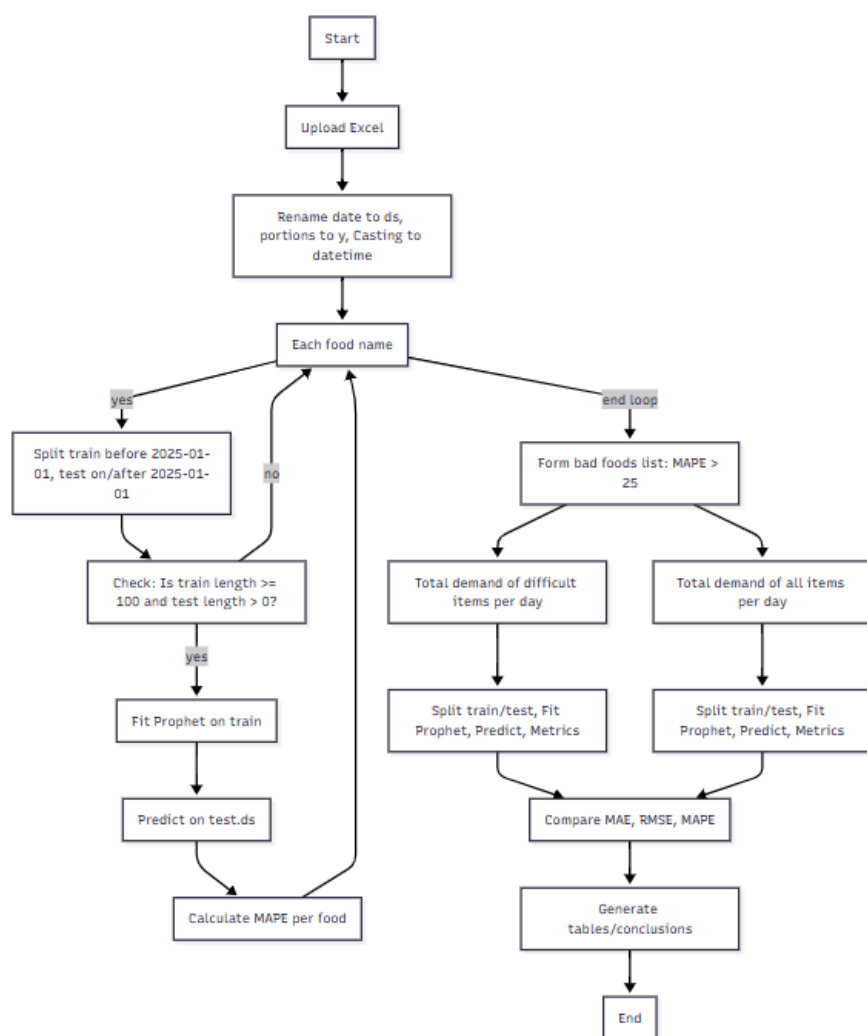


Εικόνα 68: Class Diagram που παρουσιάζει τη λογική δομή των δεδομένων και των βασικών λειτουργικών ενοτήτων του συστήματος.

7.6.4 Διάγραμμα Ροής – Αλγόριθμος Πρόβλεψης (Flowchart)

Τέλος, το Διάγραμμα Ροής απεικονίζει βήμα προς βήμα τον αλγόριθμο που εφαρμόστηκε για την πρόβλεψη της ζήτησης μερίδων φαγητού μέσω του Prophet. Η διαδικασία ξεκινά με την εισαγωγή του αρχείου Excel και την αναδόμηση των δεδομένων στη μορφή που απαιτεί το μοντέλο (μετονομασία πεδίων *date* → *ds*, *portions* → *y*, μετατροπή σε datetime). Στη συνέχεια, για κάθε

φαγητό, τα δεδομένα χωρίζονται σε training (πριν την 01/01/2025) και test (μετά την 01/01/2025), με έλεγχο ότι υπάρχει επαρκές πλήθος παρατηρήσεων ($\text{train} \geq 100$, $\text{test} > 0$). Εφόσον πληρούνται οι συνθήκες, το μοντέλο Prophet εκπαιδεύεται στα training δεδομένα και πραγματοποιεί προβλέψεις στο test, με το MAPE να υπολογίζεται ανά προϊόν. Αφού ολοκληρωθεί ο βρόχος, τα προϊόντα με υψηλό σφάλμα ($\text{MAPE} > 25\%$) συγκεντρώνονται σε μία λίστα. Από εκεί και πέρα ακολουθούν δύο παράλληλες διαδρομές: υπολογίζεται η συνολική ζήτηση μόνο για τα «δύσκολα» προϊόντα και εκπαιδεύεται πάνω σε αυτά ένα ενιαίο μοντέλο Prophet, ενώ παράλληλα υπολογίζεται η συνολική ζήτηση για όλα τα προϊόντα ώστε να δημιουργηθεί ένα γενικό μοντέλο. Και στις δύο περιπτώσεις υπολογίζονται οι μετρικές MAE, RMSE και MAPE, ώστε να πραγματοποιηθεί συγκριτική αξιολόγηση και εξαγωγή συμπερασμάτων.



Εικόνα 69: Flowchart που παρουσιάζει τον αλγόριθμο πρόβλεψης μερίδων φαγητού.

8. Παρουσίαση Εφαρμογής

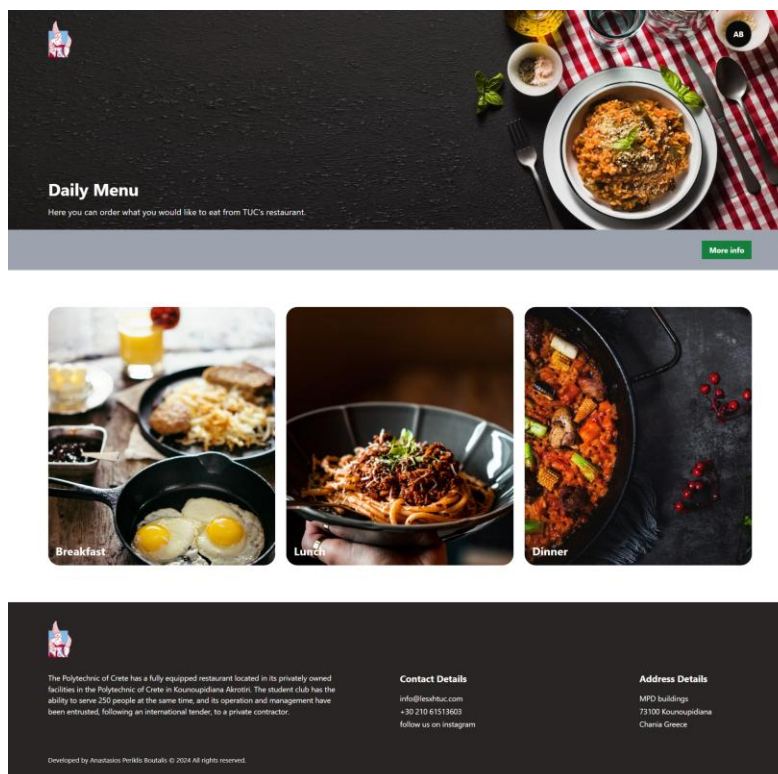
Η παρούσα ενότητα παρουσιάζει την τελική υλοποίηση της εφαρμογής μέσα από στιγμιότυπα οθόνης (screenshots). Στόχος είναι να αναδειχθεί ο τρόπος με τον οποίο ο τελικός χρήστης (φοιτητής) και ο διαχειριστής αλληλεπιδρούν με το σύστημα, καθώς και οι βασικές λειτουργίες που προσφέρονται.

8.1 Διεπαφή Χρήστη

Η διεπαφή για τον φοιτητή έχει σχεδιαστεί με γνώμονα την απλότητα και την ευκολία χρήσης. Αυτός είναι και ο λόγος που έγινε πλήρως responsive, επιτρέποντας την πρόσβαση τόσο από υπολογιστή όσο και από κινητές συσκευές.

8.1.1 Αρχική Οθόνη – Προβολή Μενού

Μόλις ο φοιτητής, εισέλθει στην εφαρμογή, αντικρίζει την **αρχική οθόνη** όπου εμφανίζονται τρία ξεκάθαρα κουμπιά επιλογής: **Breakfast**, **Lunch**, **Dinner**. Με αυτόν τον τρόπο γίνεται άμεσα η κατηγοριοποίηση και η πλοήγηση στο αντίστοιχο μενού της ημέρας.



Εικόνα 70: Αρχική οθόνη της εφαρμογής που παρουσιάζει το μενού της ημέρας στους χρήστες.

Αφού επιλέξει το είδος γεύματος, μεταφέρεται στη σελίδα του **δηλωμένου μενού**, όπου παρουσιάζονται αναλυτικά τα διαθέσιμα φαγητά. Στα αριστερά φαίνεται η κατάσταση χωρίς επιλεγμένα φαγητά και στα δεξιά με.



Εικόνα 71: Αρχική οθόνη της εφαρμογής που παρουσιάζει το δηλωμένο μενού της ημέρας στους χρήστες.

8.1.2 Σύνοψη και Υποβολή Παραγγελίας

Αφού ο φοιτητής ολοκληρώσει την επιλογή φαγητών, μεταφέρεται στη σελίδα Σύνοψης Παραγγελίας. Εκεί εμφανίζονται όλα τα στοιχεία της παραγγελίας του συγκεντρωμένα: τα φαγητά που έχει προσθέσει, οι ποσότητες, καθώς και το συνολικό κόστος (αν δεν δικαιούται δωρεάν σίτιση).

Cart Overview
Here you can see a summary of your order.

Choriatiiki Salad (salad)	Remove
Bougatsa (dessert)	Remove
Fried Vegetables (garnish)	Remove
Baked Fish (main-dishes)	Remove

Total Price: 2.6 euro

Choose Time Slot:

You can only select a time slot that starts at least 20 minutes after placing your order. Earlier time slots are not accepted because the restaurant needs at least 20 minutes to prepare your order.

13:00 - 13:25:59	14:00 - 14:19:59	14:20 - 14:39:59
15:00 - 15:19:59	15:40 - 15:59:59	16:00 - 16:19:59
14:40 - 14:59:59		

Choose Payment Method:

Use Voucher

Place to eat:

Take Away

Place Order

Contact Details
info@leshtuc.com
+30 210 61513603
follow us on Instagram

Address Details
MPD buildings
73100 Kounoupidiana
Chania Greece

Developed by Anastasios Periklis Boutalis © 2024 All rights reserved.

Εικόνα 72: Στιγμιότυπο οθόνης που παρουσιάζει τη σύνοψη της παραγγελίας και τη δυνατότητα υποβολής από τον χρήστη.

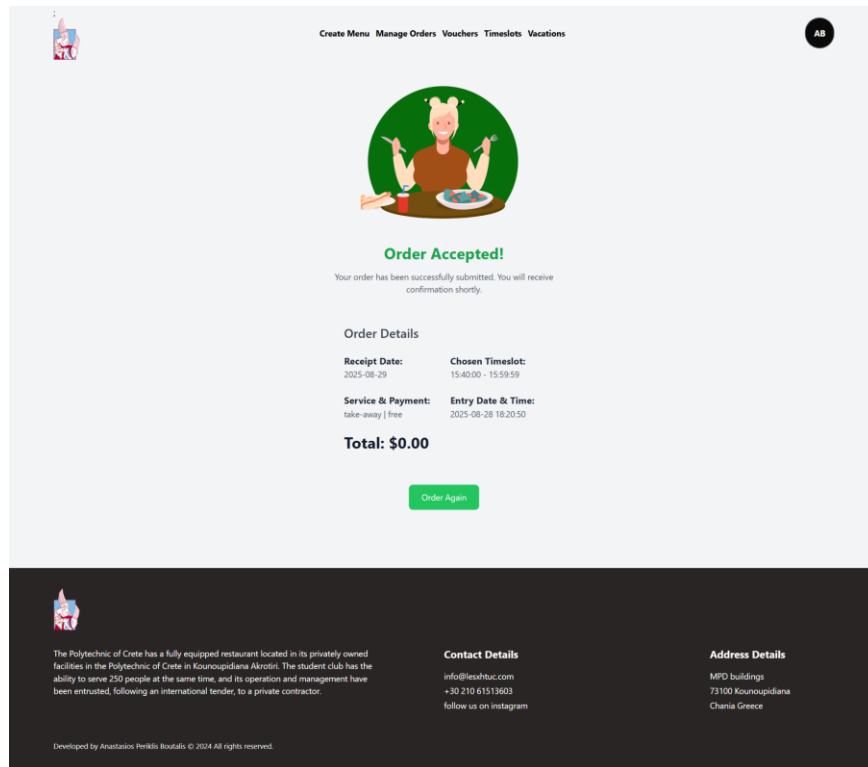
Πιο συγκεκριμένα:

- Ο χρήστης μπορεί να **αφαιρέσει φαγητά** ή να τροποποιήσει την ποσότητα πριν προχωρήσει στην τελική υποβολή.

- Στην ίδια σελίδα επιλέγει το **χρονικό διάστημα (timeslot)** στο οποίο θέλει να παραλάβει ή να καταναλώσει το γεύμα του. Η επιλογή γίνεται μέσα από ένα **χρωματιστό grid**, όπου η χρωματική διαφοροποίηση ενημερώνει για την πληρότητα κάθε χρονικής ζώνης (π.χ. πράσινο για διαθέσιμη, κόκκινο για πλήρη).
- Επιπλέον, ο χρήστης καθορίζει **πού θα φάει**: στον χώρο της λέσχης ή σε πακέτο (take-away).
- Τέλος, γίνεται η επιλογή του **τρόπου πληρωμής**, η οποία διαφοροποιείται ανάλογα με το αν ο φοιτητής:
 - διαθέτει δικαίωμα δωρεάν σίτισης,
 - ή αν παραγγέλνει δεύτερη φορά το ίδιο γεύμα, οπότε χρησιμοποιεί voucher ή άλλο μέσο πληρωμής.

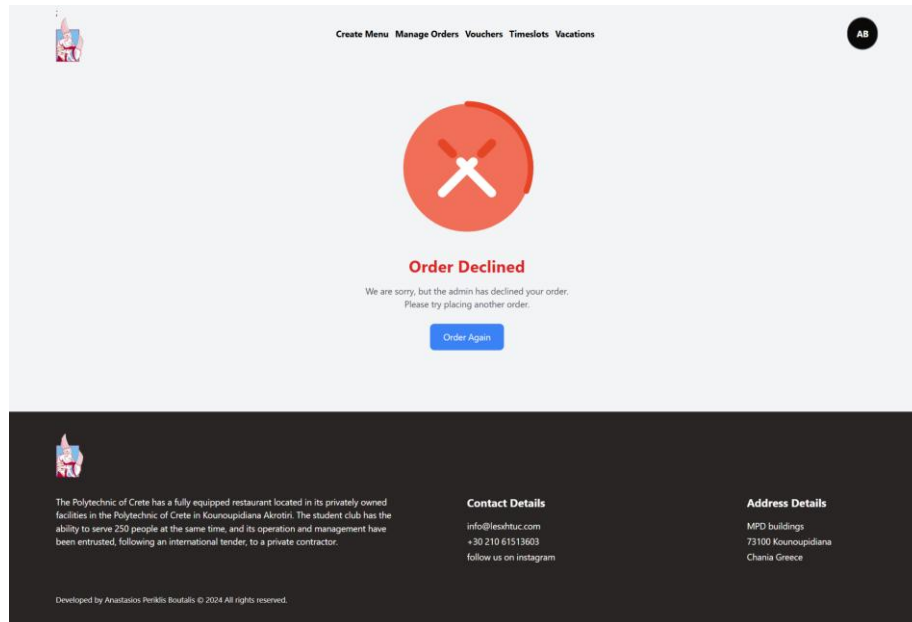
Μετά την υποβολή, η εφαρμογή εμφανίζει **μήνυμα ανατροφοδότησης (feedback)** στον χρήστη:

- Σε περίπτωση **επιτυχούς παραγγελίας**, εμφανίζεται μήνυμα επιβεβαίωσης με τα βασικά στοιχεία της παραγγελίας.



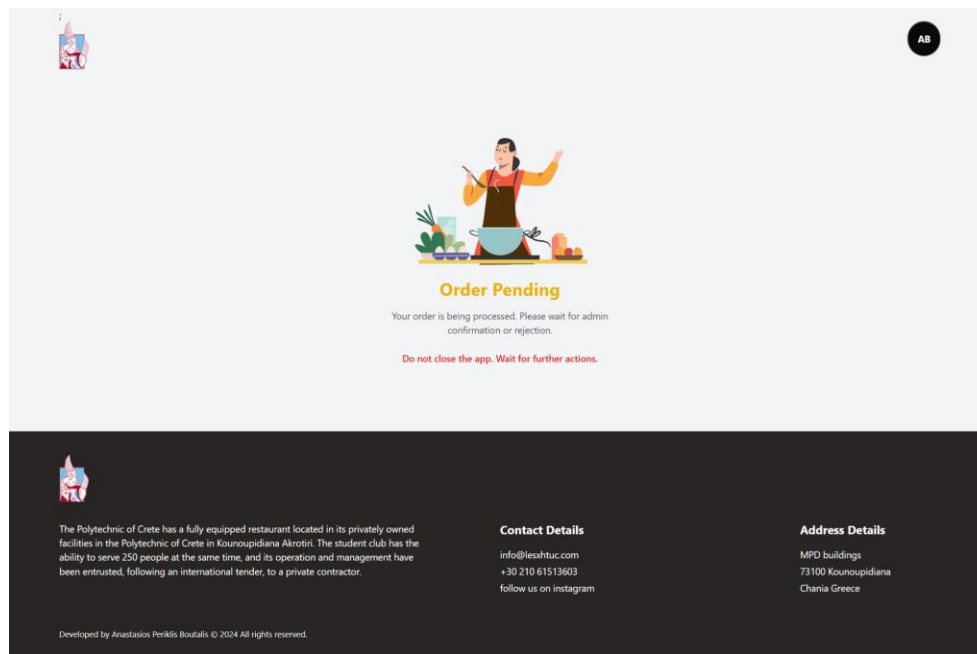
Εικόνα 73: Στιγμιότυπο οθόνης Επιβεβαίωση Επιτυχούς Παραγγελίας.

- Σε περίπτωση **αποτυχίας** (π.χ. λόγω έλλειψης μερίδων ή προβλήματος στη σύνδεση), εμφανίζεται αντίστοιχο μήνυμα σφάλματος.



Εικόνα 74: Στιγμιότυπο οθόνης Απόρριψης Παραγγελίας.

- Αν η παραγγελία είναι σε **κατάσταση αναμονής**, ο χρήστης ενημερώνεται ότι η διαδικασία εκκρεμεί και θα ολοκληρωθεί σύντομα.



Εικόνα 75: Στιγμιότυπο οθόνης αναμονής απάντησης παραγγελίας.

Με αυτόν τον τρόπο, ο χρήστης γνωρίζει πάντα την τρέχουσα κατάσταση της παραγγελίας του και λαμβάνει άμεση ενημέρωση από το σύστημα.

8.1.3 Σελίδα Προφίλ Χρήστη

Η σελίδα προφίλ συγκεντρώνει όλες τις προσωπικές πληροφορίες και τα στοιχεία που σχετίζονται με τον λογαριασμό του φοιτητή. Στόχος είναι ο χρήστης να έχει άμεση και ξεκάθαρη εικόνα για τα δικαιώματα σίτισής του, το υπόλοιπο των vouchers και το ιστορικό των ενεργειών του.

Profile

Anastasios-Periklis Boutalis

Email: aboutalis@fuc.gr
AM: 2017030123
Food Status: Free food!
Voucher: 13.00 €

[Logout](#)

Voucher Purchase History

Date	Time	Amount
03/09/2025	17:13	Amount: 12.00 €
03/07/2025	12:57	Amount: 25.00 €
15/06/2025	20:31	Amount: 15.00 €
01/05/2025	09:19	Amount: 10.00 €

Order history [View All](#)

Search orders...

ID	Order ready on	Placed	Timeslot	You ordered
ID: 477	02/07/2025	02/07/2025 - 11:30	15:40:00 - 15:59:59	Graviere, Boiled Corn Salad, Chicken Cacciatore
ID: 476	02/07/2025	02/07/2025 - 11:10	09:30:00 - 09:59:59	Whole-Grain bread, Apples, Halloumi
ID: 475	02/07/2025	01/07/2025 - 00:45	08:55:00 - 09:09:59	Whole-Grain bread
ID: 474	02/07/2025	01/07/2025 - 00:44	08:55:00 - 09:09:59	Whole-Grain bread
ID: 473	02/07/2025	01/07/2025 - 00:39	09:19:00 - 09:29:59	Whole-Grain bread, Apples
ID: 471	30/06/2025	26/06/2025 - 16:05	08:55:00 - 09:09:59	Whole-Grain bread
ID: 469	18/10/2024	17/10/2024 - 00:14	08:55:00 - 09:09:59	Whole-Grain bread, Apples, Halloumi
ID: 468	14/10/2024	14/10/2024 - 22:18	21:00:00 - 21:25:59	Graviere

Contact Details

info@esaftuc.com
+30 210 61513603
follow us on instagram

Address Details

MPD buildings
73100 Kounoupidiana
Chania Greece

Developed by Anastasios Periklis Boutalis © 2024 All rights reserved.

Εικόνα 76: Στιγμιότυπο οθόνης της σελίδας Προφίλ Χρήστη.

Πιο συγκεκριμένα:

- **Προσωπικά Στοιχεία και Δικαίωμα Σίτισης**
Στην κορυφή της σελίδας εμφανίζονται τα βασικά προσωπικά στοιχεία του χρήστη (όνομα, αριθμός μητρώου κ.λπ.).
Παράλληλα, αναγράφεται με σαφήνεια αν διαθέτει **δικαίωμα δωρεάν σίτισης** ή αν οι παραγγελίες του γίνονται μέσω voucher/άλλης πληρωμής.
- **Υπόλοιπο Voucher**
Ακολουθεί ξεχωριστή ενότητα που δείχνει το **τρέχον υπόλοιπο voucher**, εφόσον ο χρήστης διαθέτει. Έτσι μπορεί να γνωρίζει άμεσα πόσες μερίδες έχει διαθέσιμες για παραγγελία.
- **Ιστορικό Συναλλαγών Voucher**
Στην επόμενη ενότητα εμφανίζεται αναλυτικός πίνακας με όλες τις κινήσεις που έχουν γίνει με vouchers, συμπεριλαμβανομένης της ημερομηνίας, του ποσού και του είδους της συναλλαγής (φόρτιση υπολοίπου ή χρήση).
- **Ιστορικό Παραγγελιών**
Στο τέλος της σελίδας βρίσκεται το **ιστορικό παραγγελιών**, όπου καταγράφονται οι προηγούμενες παραγγελίες του φοιτητή. Για κάθε παραγγελία εμφανίζονται η ημερομηνία, το είδος γεύματος, τα φαγητά που επιλέχθηκαν και ο τρόπος πληρωμής.

Η σελίδα προφίλ προσφέρει στον χρήστη μια ολοκληρωμένη εικόνα της δραστηριότητάς του στο σύστημα, δίνοντάς του διαφάνεια και έλεγχο τόσο στις παραγγελίες όσο και στη χρήση των vouchers.

8.2 Διεπαφή Διαχειριστή

Όπως και οι απλοί χρήστες, έτσι και ο διαχειριστής έχει πρόσβαση σε όλες τις βασικές λειτουργίες της εφαρμογής (προβολή μενού, υποβολή παραγγελίας, ιστορικό). Ωστόσο, διαθέτει και **επιπλέον δυνατότητες** που του επιτρέπουν να εποπτεύει και να διαχειρίζεται πλήρως τη λειτουργία της λέσχης.

8.2.1 Δήλωση Ημερήσιου Μενού

Ο διαχειριστής μπορεί να δημιουργεί και να δημοσιεύει το μενού της ημέρας, προσθέτοντας ή αφαιρώντας φαγητά σε πραγματικό χρόνο. Η διεπαφή υποστηρίζει γρήγορη επιλογή φαγητών από το γενικό μενού της βάσης δεδομένων.

Επιπλέον, μέσα από την ίδια σελίδα μπορεί να **διαχειριστεί το ήδη υπάρχον γενικό μενού**, τροποποιώντας στοιχεία όπως:

- εικόνες φαγητών,
- περιγραφές και διατροφικές πληροφορίες,
- κατηγορίες και τιμές (όπου ισχύει).

Ο διαχειριστής έχει επίσης τη δυνατότητα να **αλλάζει τη διαθεσιμότητα των πιάτων** δυναμικά. Η κατάσταση κάθε φαγητού απεικονίζεται με **χρωματική παλέτα**, ώστε να φαίνεται με μια ματιά αν ένα φαγητό είναι διαθέσιμο, περιορισμένο ή εξαντλημένο.

Declare Menu

On this page, you can manage the Daily Menu by adding or removing food items, as well as modify the status of each meal type. It provides full control over menu customization and meal scheduling.

Would you like to add new food item? [Add New Food Item](#)

Would you like to make a forecast? [Forecast Now](#)

If you'd like to predict the number of portions for a specific day, you can do so by visiting the forecast page. Press the button below to get started.

Drag and Drop and Live Search Enter the date 08/28/2025

Breakfast

Lunch

☒ Enable Lunch

[Cheese](#) [Salad](#) [Main Dishes](#) [Garnish](#) [Fruit](#) [Dessert](#)

Main Dishes

Dish Name	Status	Action
Gemista	In Danger	OK
Meatballs in Tomato Sauce	In Danger	OK
Chicken Cacciatore	In Danger	OK
Chicken with Mustard Sauce	In Danger	OK
Pesto Pasta	In Danger	OK
Baked Sausage	In Danger	OK

[Reset](#) [Save](#)

List of Main Dishes

Search foods (type 3 or more letters):

- Gemista
- Chicken Cacciatore
- Carbonara
- Creamy Screw Pasta
- Neapolitan-style Pasta
- Garlic-Roasted Lamb
- Pasta with Pizzaiola Sauce
- Vegetable Rice
- Spanish-style Meatballs
- Goulash
- Curry-Roasted Chicken
- Pasta with Ground Meat
- Chicken with Mustard Sauce
- Meatballs in Tomato Sauce
- Lasagna with Meat
- Peas with Potatoes
- Baked Giant Beans
- Baked Chicken Legs
- Beef Stew
- Oregano-Flavored Sausages
- Rigatoni Pasta with Tomato Sauce
- Imam Baidi
- Bolognese Pasta
- Spinach and Feta Pie
- Roasted Lamb
- Roasted Chicken

Dinner

Contact Details

info@lesaltuc.com
+30 210 61513603
Follow us on Instagram

Address Details

MFD buildings
73100 Kounoupidiana
Chania Greece

Developed by Anastasio Perikis Routalis © 2024 All rights reserved.

Εικόνα 77: Στιγμιότυπο οθόνης της σελίδας Δήλωση; Ημερήσιου Μενού.

8.2.2 Διαχείριση Παραγγελιών

Ο διαχειριστής έχει πρόσβαση σε έναν **πίνακα ελέγχου παραγγελιών** όπου εμφανίζονται όλες οι παραγγελίες που έχουν υποβληθεί σε πραγματικό χρόνο. Μέσα από αυτή τη σελίδα μπορεί να εποπτεύει τη ροή της λέσχης και να λαμβάνει γρήγορα αποφάσεις.

Οι βασικές δυνατότητες περιλαμβάνουν:

- **Real-time ενημέρωση:** κάθε νέα παραγγελία εμφανίζεται άμεσα στο dashboard, χωρίς καθυστέρηση.
- **Διαχείριση παραγγελιών** βάσει της διαθεσιμότητας.
- **Φιλτράρισμα παραγγελιών:** ο διαχειριστής μπορεί να ταξινομεί και να φιλτράρει τις παραγγελίες ανά χρονική ζώνη (π.χ. 13:00–14:00), είδος γεύματος (πρωινό, μεσημεριανό, βραδινό) ή κατάσταση παραγγελίας (εκκρεμής, ολοκληρωμένη, ακυρωμένη).

Display Orders

Enter the preferred date: 08/29/2025

Need Attention

Order ID	Meal	Service	Payment Method	Customer	Timeslot	Items	Actions
482	lunch	take-away	cashier	Anastasio Periklis Boutal...	15:40:00 - 15:59:59	Choriatiki Salad, Graviera, Chicken with Mustard Sauce, Fried Ve...	<button>Reject</button> <button>Accept</button>

All Orders

Order ID	Received	Status	Meal Type	Service	Payment Meth...	Customer	Timeslot	Items	Actions
481	✗	rejected	lunch	eat-there	cashier	Anastasio Periklis B...	14:40:00 - 14:59:59	Chicken with Mustar...	
480	✗	accepted	lunch	take-away	free	Anastasio Periklis B...	15:40:00 - 15:59:59	Feta, Gemista, Yogur...	<button>Reject</button>

Contact Details
info@leshtuc.com
+30 210 61513603
follow us on instagram

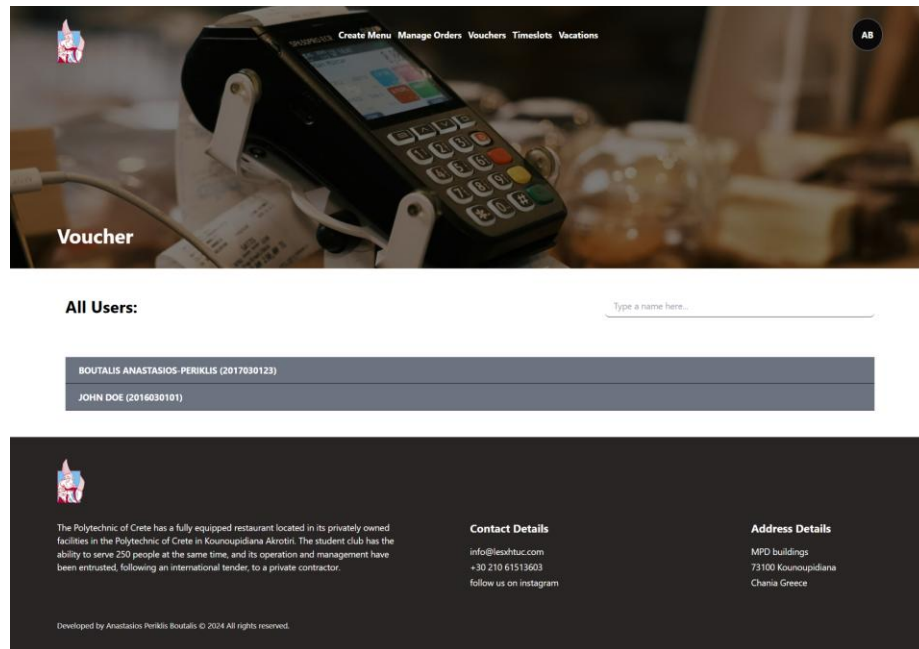
Address Details
MPD buildings
73100 Kounoupidiana
Chania Greece

Developed by Anastasio Periklis Boutalis © 2024 All rights reserved.

Εικόνα 78: Στιγμιότυπο οθόνης της σελίδας Διαχείρισης Παραγγελιών.

8.2.3 Διαχείριση Voucher

Ο διαχειριστής μπορεί να δημιουργεί, τροποποιεί και διαγράφει κουπόνια (vouchers) που μπορούν να χρησιμοποιούν οι χρήστες κατά τις ηλεκτρονικές τους παραγγελίες. Σε μία λίστα εμφανίζονται όλοι οι χρήστες τους οποίους μπορεί να φιλτράρει μέσω αναζήτησης κειμένου.



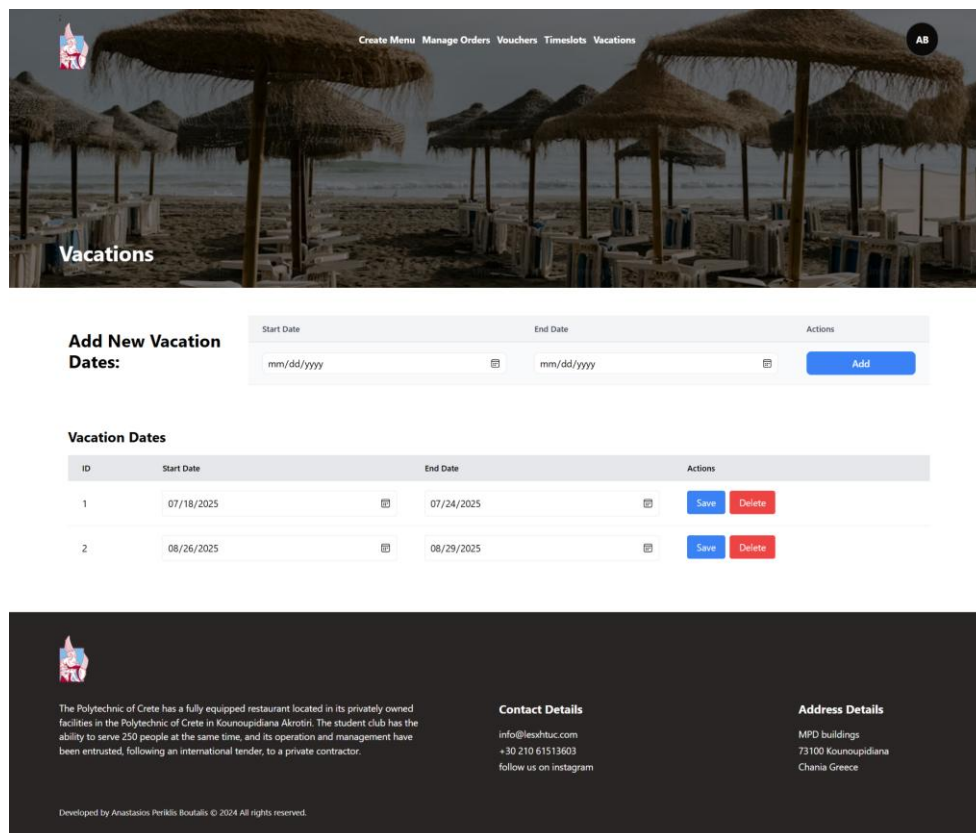
Εικόνα 79: Στιγμιότυπο οθόνης της σελίδας Διαχείρισης Voucher.

8.2.4 Διαχείριση Timeslots

Η σελίδα διαχείρισης timeslots επιτρέπει στον διαχειριστή να οργανώνει τα χρονικά διαστήματα παραγγελιών ή παραλαβών φαγητού, διασφαλίζοντας ομαλή ροή στην υπηρεσία της λέσχης. Κάθε τύπος γεύματος, όπως πρωινό, μεσημεριανό ή βραδινό, έχει τα δικά του ξεχωριστά timeslots. Ο διαχειριστής μπορεί να δημιουργεί νέα timeslots, να επεξεργάζεται τα ήδη υπάρχοντα και να διαγράφει όσα δεν χρειάζονται πλέον. Με αυτόν τον τρόπο διασφαλίζεται ότι τα slots είναι πάντα ενημερωμένα και αντιστοιχούν στη δυναμικότητα της λέσχης για κάθε τύπο γεύματος.

Για λόγους εμφάνισης και ώστε να μην καταλαμβάνει υπερβολικό χώρο, η περιοχή όπου εμφανίζονται οι προβλεπόμενες μερίδες έχει περιοριστεί σε μήκος· η εικόνα έχει γίνει crop ώστε να παραμείνει ευανάγνωστη και κομψή.

ενημερώνονται σωστά για τις μέρες κατά τις οποίες δεν πραγματοποιούνται παραγγελίες ή παραλαβές.



Εικόνα 81: Στιγμιότυπο οθόνης της σελίδας Διαχείρισης Διακοπών.

8.2.6 Προβολή Πρόβλεψης Μερίδων Φαγητού

Η σελίδα προβλέψεων μερίδων επιτρέπει στον χρήστη να επιλέξει μια συγκεκριμένη ημερομηνία και το είδος της ημέρας (π.χ. καθημερινή, αργία, περίοδος εξετάσεων). Βάσει αυτών των δεδομένων, το σύστημα υπολογίζει την προβλεπόμενη ζήτηση για κάθε πιάτο. Στον χρήστη εμφανίζονται οι προβλεπόμενες μερίδες καθώς και οι πληροφορίες καιρού για την επιλεγμένη ημέρα. Δεδομένου ότι η διαδικασία υπολογισμού μπορεί να διαρκέσει λίγο χρόνο, η σελίδα περιλαμβάνει ένα animation αναμονής, ώστε ο χρήστης να γνωρίζει ότι η πρόβλεψη εκτελείται και να παραμένει ενημερωμένος μέχρι την εμφάνιση των αποτελεσμάτων.



Date to Forecast:

09/17/2025

Type of Day:

Exam Period

Weather Details For The Selected Date



Temperature

23.9°C

Temperature



Humidity

66.6%

Humidity Level



Wind Gusts

19.1 km/h

Wind Gust Speed



Wind Speed

21.6 km/h

Wind Speed

Search food by name...

Penne Arrabbiata

PREDICTED PORTIONS

82

Min Portions

69

Max Portions

93

Model Accuracy Metrics

Metric	Value
MAE	7.21
MAPE	9.2%
RMSE	8.99

Baked Chicken Legs

PREDICTED PORTIONS

79

Min Portions

69

Max Portions

90

Model Accuracy Metrics

Metric	Value
MAE	6.43
MAPE	8.14%
RMSE	8.66

Peas with Potatoes

PREDICTED PORTIONS

85

Min Portions

74

Max Portions

95

Model Accuracy Metrics

Metric	Value
MAE	6.38
MAPE	8.23%
RMSE	8.17

Pesto Penne

PREDICTED PORTIONS

49

Min Portions

41

Max Portions

58

Model Accuracy Metrics

Metric	Value
MAE	5

Rice pilaf

PREDICTED PORTIONS

51

Min Portions

42

Max Portions

60

Model Accuracy Metrics

Metric	Value
MAE	5.44

Stuffed Peppers

PREDICTED PORTIONS

61

Min Portions

38

Max Portions

82

Model Accuracy Metrics

Metric	Value
MAE	14.02

MAPE	10.46%
RMSE	6.42

MAPE	11.22%
RMSE	6.68

MAPE	23.27%
RMSE	15.95

Beetroot salad

PREDICTED PORTIONS **52**

Min Portions	Max Portions
44	60

Model Accuracy Metrics

Metric	Value
MAE	5.2
MAPE	10.95%
RMSE	6.61

Garlic-Roasted Lamb

PREDICTED PORTIONS **76**

Min Portions	Max Portions
66	87

Model Accuracy Metrics

Metric	Value
MAE	6.58
MAPE	8.45%
RMSE	8.3

Baked Meatballs

PREDICTED PORTIONS **86**

Min Portions	Max Portions
75	97

Model Accuracy Metrics

Metric	Value
MAE	6.99
MAPE	9.07%
RMSE	8.71

Baked Sausage

PREDICTED PORTIONS **84**

Min Portions	Max Portions
73	94

Model Accuracy Metrics

Metric	Value
MAE	6.75
MAPE	8.66%
RMSE	7.97

Carbonara

PREDICTED PORTIONS **73**

Min Portions	Max Portions
63	83

Model Accuracy Metrics

Metric	Value
MAE	6.51
MAPE	8.95%
RMSE	8

Chicken Cacciatore

PREDICTED PORTIONS **81**

Min Portions	Max Portions
72	89

Model Accuracy Metrics

Metric	Value
MAE	5.86
MAPE	7.46%
RMSE	7.64

Baked Penne

PREDICTED PORTIONS **82**

Min Portions	Max Portions
72	92

Model Accuracy Metrics

Metric	Value
MAE	6.26
MAPE	8.05%
RMSE	7.88

Meatballs in Tomato Sauce

PREDICTED PORTIONS **80**

Min Portions	Max Portions
69	91

Model Accuracy Metrics

Metric	Value
MAE	6.8
MAPE	8.72%
RMSE	8.18

Moussaka

PREDICTED PORTIONS **65**

Min Portions	Max Portions
45	85

Model Accuracy Metrics

Metric	Value
MAE	11.85
MAPE	22.86%
RMSE	15.99

Chicken with Mustard Sauce

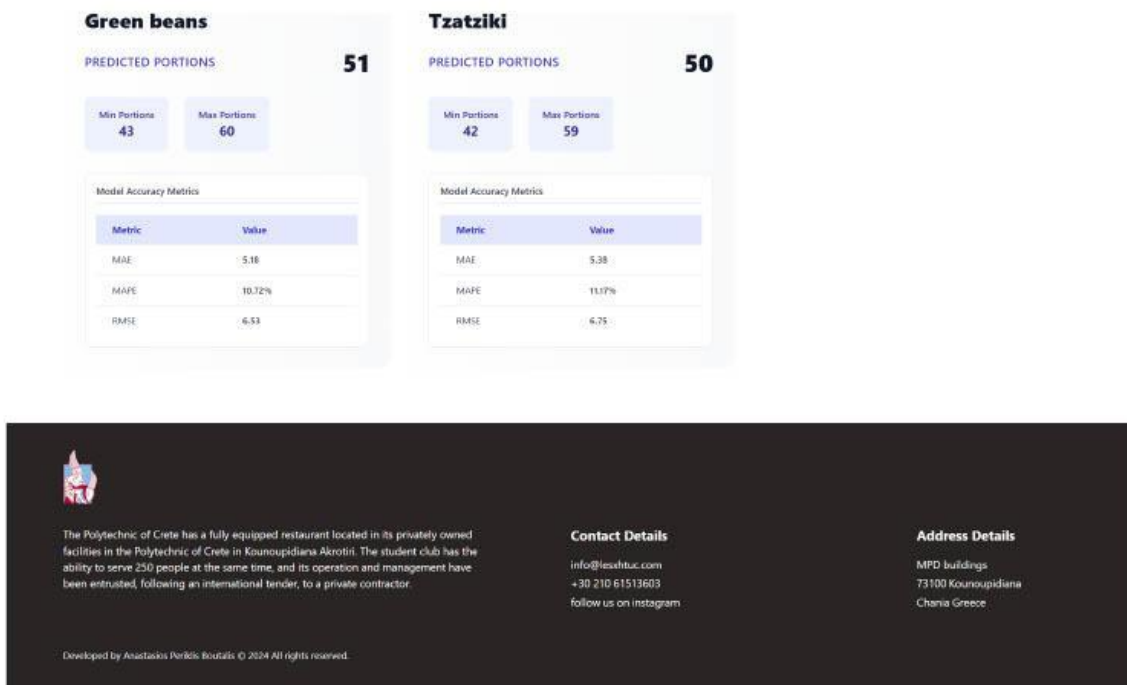
PREDICTED PORTIONS **80**

Four Cheese Penne

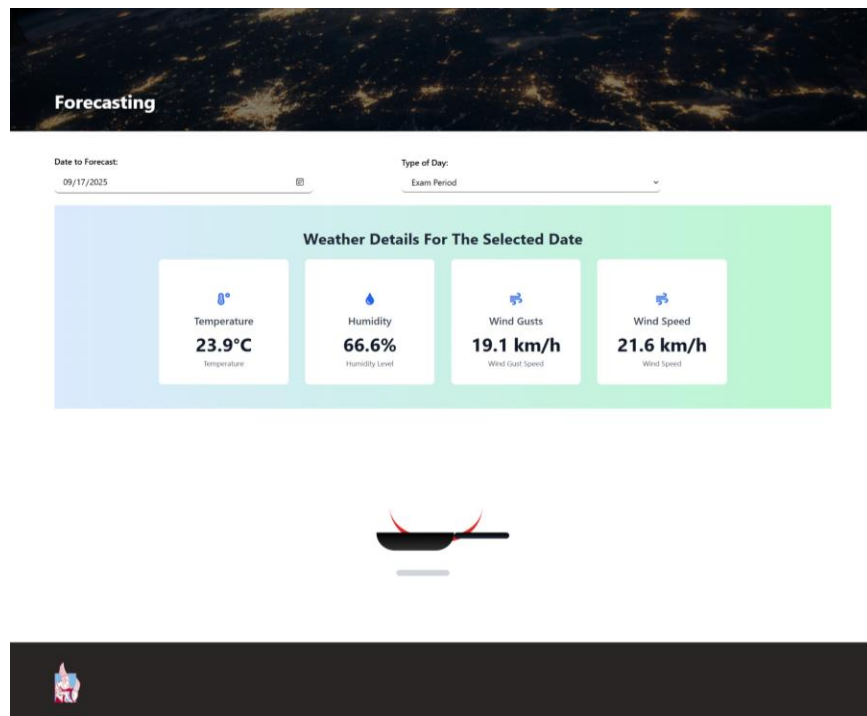
PREDICTED PORTIONS **80**

Roasted Pork Chop

PREDICTED PORTIONS **69**



Εικόνα 82: Στιγμιότυπο που δείχνει τα αποτελέσματα των προβλέψεων μερίδων φαγητού στην αντίστοιχη σελίδα, περιλαμβάνοντας τις τιμές πρόβλεψης, ελάχιστες και μέγιστες.



Εικόνα 83: Στιγμιότυπο οθόνης που παρουσιάζει την ένδειξη αναμονής για την εμφάνιση των αποτελεσμάτων πρόβλεψης μερίδων φαγητού.

9. Αξιολόγηση Λειτουργικότητας, Απόδοσης και Σχεδίασης της Εφαρμογής

Στην ενότητα αυτή παρουσιάζεται η συνολική αξιολόγηση της εφαρμογής τόσο από την πλευρά του σχεδιασμού (UI/UX), όσο και από την τεχνική πλευρά (απόδοση, προσβασιμότητα, βέλτιστες πρακτικές). Στόχος είναι να επιβεβαιωθεί ότι η εφαρμογή πληροί βασικά πρότυπα ποιότητας, είναι φιλική προς τον χρήστη, προσβάσιμη, και αποδοτική σε διάφορες συσκευές.

Για την αξιολόγηση χρησιμοποιήθηκαν τα εργαλεία **BrowserStack**, ώστε να ελεγχθεί η εμφάνιση και λειτουργικότητα της εφαρμογής σε διαφορετικά λειτουργικά συστήματα και διαστάσεις οθόνης, και **Google Lighthouse**, ώστε να μετρηθούν αντικειμενικά δείκτες απόδοσης, προσβασιμότητας και βέλτιστων πρακτικών.

9.1 Εισαγωγή στην Αξιολόγηση της Εφαρμογής

Η αξιολόγηση της εφαρμογής αποτελεί κρίσιμο στάδιο για τη διασφάλιση της ποιότητας και της εμπειρίας χρήσης. Πέρα από την απλή λειτουργικότητα, εξετάστηκαν πτυχές όπως η απόδοση σε διαφορετικές συσκευές και browsers, η προσβασιμότητα, η ασφάλεια και η αντίδραση του συστήματος σε σφάλματα. Η αξιολόγηση συνδυάζει αυτόματους και χειροκίνητους ελέγχους με στόχο μια πλήρη και ρεαλιστική αποτίμηση της εφαρμογής.

9.1.1 Έλεγχος σε Πραγματικές Συσκευές και Browsers (BrowserStack)

Μέσω του εργαλείου **BrowserStack**, πραγματοποιήθηκαν εκτεταμένα tests σε πραγματικά περιβάλλοντα, όπως:

- **Android (Chrome)** και **iOS (Safari)**
- **Windows (Edge, Firefox, Chrome)**
- **macOS (Safari, Chrome)**

Επαληθεύτηκε η **σωστή λειτουργία του responsive design**, καθώς και η συνέπεια στην εμφάνιση των περιεχομένων σε διαφορετικές αναλύσεις και λειτουργικά συστήματα. Επιβεβαιώθηκε ότι η εμπειρία χρήστη παραμένει σταθερή τόσο σε κινητές όσο και σε desktop συσκευές (**Mobile-first προσέγγιση**).

9.1.2 Αξιολόγηση Απόδοσης και Ποιότητας με Google Lighthouse

Για την τεχνική αξιολόγηση της απόδοσης και της ποιότητας του frontend, χρησιμοποιήθηκε το εργαλείο Google Lighthouse, το οποίο παρέχει αναλυτικούς δείκτες για βασικές κατηγορίες όπως απόδοση (Performance), προσβασιμότητα (Accessibility), βέλτιστες πρακτικές (Best Practices) και βελτιστοποίηση για μηχανές αναζήτησης (SEO).

Η αξιολόγηση πραγματοποιήθηκε στις δύο βασικές σελίδες της εφαρμογής:

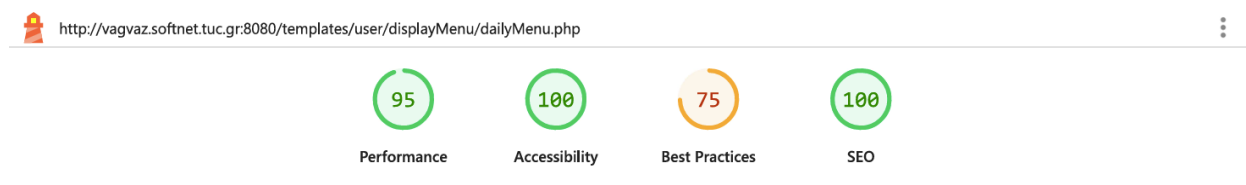
- Σελίδα Παραγγελίας Χρήστη (Daily Menu)
- Σελίδα Προβολής Δηλωμένου Μενού από Διαχειριστή (Declare Menu)

Αναλυτικά Αποτελέσματα Lighthouse:

Metric	Score (Daily Menu)	Score (Declare Menu)
Performance	95	80
Accessibility	100	100
Best Practices	75	75
SEO	100	100

Πίνακας 8: Αναλυτικά αποτελέσματα αξιολόγησης της εφαρμογής με χρήση Google Lighthouse, περιλαμβάνοντας επιδόσεις, προσβασιμότητα, βέλτιστες πρακτικές και SEO.

Ανάλυση Lighthouse για τη Σελίδα “Daily Menu”



Εικόνα 84: Ανάλυση της σελίδας “Daily Menu” με χρήση Google Lighthouse, παρουσιάζοντας επιδόσεις, προσβασιμότητα, βέλτιστες πρακτικές και SEO.

9.1.2.1 Performance (95/100)

Ο δείκτης απόδοσης είναι **ιδιαίτερα υψηλός**, γεγονός που αποδεικνύει ότι οι σελίδες φορτώνουν γρήγορα και είναι βελτιστοποιημένες τόσο σε επίπεδο δικτύου όσο και rendering. Συγκεκριμένα, επιτεύχθηκαν **χαμηλοί χρόνοι First Contentful Paint (FCP)** και **Time to Interactive (TTI)**, γεγονός που ενισχύει τη θετική εμπειρία χρήστη.

9.1.2.2 Accessibility (100/100):

Η εφαρμογή συμμορφώνεται πλήρως με τις **οδηγίες προσβασιμότητας (WCAG)**. Χρησιμοποιούνται κατάλληλα ARIA attributes, semantic HTML tags, καθώς και πλήρης υποστήριξη για πληκτρολόγιο και screen readers.

9.1.2.3 Best Practices (75/100):

Ο μοναδικός λόγος που το σκορ ήταν **χαμηλότερο** οφείλεται στην εκτέλεση των ελέγχων σε **τοπικό περιβάλλον (localhost)**, χωρίς ενεργοποιημένο SSL (HTTPS).

Η αξιολόγηση εντόπισε:

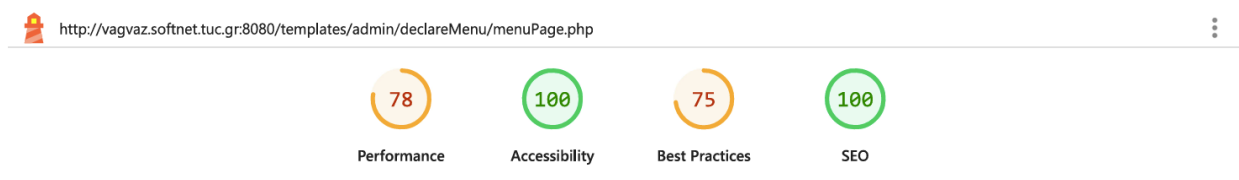
- Απουσία HTTPS σύνδεσης.
- Απουσία ανακατεύθυνσης από HTTP → HTTPS.

Το συγκεκριμένο ζήτημα δεν υφίσταται σε παραγωγικό περιβάλλον, όπου εφαρμόζεται πιστοποιητικό SSL και υποχρεωτική ανακατεύθυνση μέσω .htaccess ή server configuration.

9.1.2.4 SEO (100/100):

Η εφαρμογή πληροί **όλα τα βασικά κριτήρια SEO**, όπως χρήση meta tags, semantic elements, εναλλακτικό κείμενο (alt) σε εικόνες, καθώς και καλή δομή DOM. Αυτό διασφαλίζει την ορατότητα της εφαρμογής σε μηχανές αναζήτησης και υποστηρίζει τη μελλοντική της επέκταση στο διαδίκτυο.

Ανάλυση Lighthouse για τη Σελίδα “Declare Menu”



Εικόνα 85: Ανάλυση της σελίδας “Declare Menu” με χρήση Google Lighthouse, παρουσιάζοντας επιδόσεις, προσβασιμότητα, βέλτιστες πρακτικές και SEO.

Η σελίδα Declare Menu, η οποία προορίζεται για τη διαχείριση του μενού από τον διαχειριστή και περιλαμβάνει λειτουργίες όπως η δήλωση, η επεξεργασία και η καταχώρηση νέων προϊόντων, εμφανίζει χαμηλότερο σκορ απόδοσης (**Performance: 78/100**) σε σχέση με τη σελίδα Daily Menu. Η μείωση αυτή οφείλεται στον αυξημένο αριθμό **αιτημάτων fetch** προς τον διακομιστή και στον εκτεταμένο όγκο **JavaScript** που απαιτείται για την υποστήριξη των δυναμικών λειτουργιών της σελίδας, όπως η δημιουργία και διαχείριση **modal παραθύρων, tabs** και δυναμικών **DOM στοιχείων**. Αυτό το επιπλέον φορτίο καθυστερεί τον χρόνο **First Contentful Paint (FCP)** και **Time to Interactive (TTI)**, μειώνοντας το σκορ Performance, χωρίς να επηρεάζει την τελική εμπειρία χρήστη σε ουσιαστικό βαθμό. Αντίθετα, η σελίδα **Daily Menu**, με λιγότερες δυναμικές λειτουργίες, παρουσιάζει υψηλότερη απόδοση.

Παρά τη διαφοροποίηση στο Performance, η εφαρμογή διατηρεί υψηλές επιδόσεις σε **Accessibility, Best Practices και SEO**, καθώς χρησιμοποιεί **semantic HTML, ARIA attributes**, κατάλληλα **meta tags** και **εναλλακτικά κείμενα σε εικόνες**, πληρώντας ταυτόχρονα τις βασικές απαιτήσεις προσβασιμότητας και SEO. Η σύντομη αναφορά σε αυτούς τους τομείς επαρκεί για να διαπιστωθεί η συμμόρφωση με τα πρότυπα, χωρίς να απαιτείται περαιτέρω ανάλυση, δεδομένου ότι τα αποτελέσματα είναι πανομοιότυπα για τις δύο σελίδες.

Η χρήση του Google Lighthouse ανέδειξε την **τεχνική πληρότητα** της εφαρμογής, με **πολύ υψηλές επιδόσεις** σε όλους τους κρίσιμους τομείς. Η μοναδική απόκλιση στο σκορ των Best Practices αποδίδεται αποκλειστικά σε παράγοντες του τοπικού περιβάλλοντος και **δεν αφορά την τελική εγκατάσταση**. Η επιτυχία αυτή αποδεικνύει την **εστίαση στην απόδοση, την προσβασιμότητα και τη σωστή δομή κώδικα** ήδη από τη φάση ανάπτυξης, χωρίς την ανάγκη επιπλέον βελτιστοποίησης μετά την ολοκλήρωση του έργου.

9.1.3. Error Handling & Ενημέρωση Χρήστη

Η εφαρμογή διαθέτει **μηχανισμούς εντοπισμού και διαχείρισης σφαλμάτων**, με ξεκάθαρα μηνύματα προς τον χρήστη. Για την οπτική ανατροφοδότηση αξιοποιήθηκαν **ειδοποιήσεις τύπου toast και pop-up μέσω της βιβλιοθήκης SweetAlert2**, διασφαλίζοντας ότι ο χρήστης λαμβάνει άμεσα και σαφή ενημέρωση για την επιτυχία ή αποτυχία μιας ενέργειας (π.χ. υποβολή παραγγελίας, σφάλμα σύνδεσης).

9.1.4. Έλεγχος Συμβατότητας σε Browsers (Cross-browser Testing)

Πραγματοποιήθηκαν δοκιμές συμβατότητας μέσω του **BrowserStack**, σε όλους τους κύριους browsers (Chrome, Firefox, Edge, Safari), χωρίς να παρατηρηθούν σημαντικές ασυμβατότητες. Η διάταξη, οι λειτουργίες και τα animations λειτούργησαν σωστά σε κάθε περιβάλλον.

9.1.5. Βασικός Έλεγχος Ασφαλείας (Security Checks)

Έγινε χρήση τεχνικών **sanitization και validation** σε όλα τα input πεδία, ώστε να αποτραπεί η εισαγωγή κακόβουλων δεδομένων (όπως SQL injection, script injection κ.λπ.). Παράλληλα, οι βάσεις δεδομένων προστατεύονται με την κατάλληλη χρήση προετοιμασμένων ερωτημάτων (prepared statements).

9.2 Αξιολόγηση Email Ειδοποιήσεις με Υψηλή Συμβατότητα και Αξιοπιστία Παράδοσης

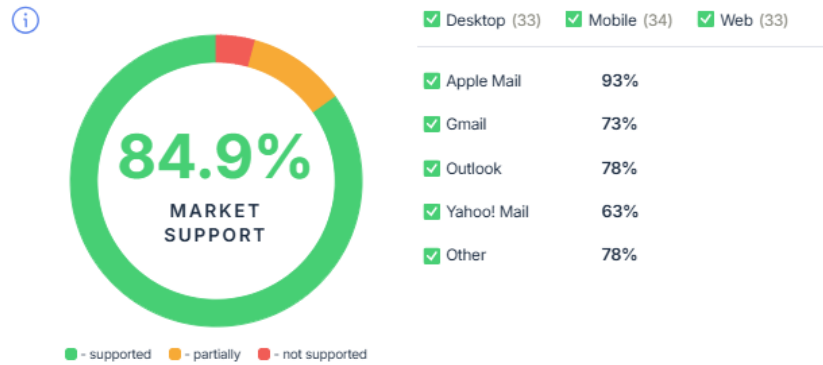
Για την άμεση και αποτελεσματική ενημέρωση των χρηστών σχετικά με την αποδοχή ή απόρριψη παραγγελίας, έχει αναπτυχθεί ένα πλήρως αυτοματοποιημένο σύστημα αποστολής email. Η λειτουργικότητα αυτή ενεργοποιείται είτε αυτόματα (στην απόρριψη από διαχειριστή λόγω έλλειψης), είτε χειροκίνητα (σε ειδικές περιπτώσεις όπως αλλαγή κατάστασης παραγγελίας λίγο πριν την ώρα εκτέλεσης).

Καθώς τα emails αυτά έχουν **λειτουργικό και κρίσιμο χαρακτήρα**, δόθηκε ιδιαίτερη έμφαση στη **συμβατότητα με όλους τους δημοφιλείς email clients**, καθώς και στη **διασφάλιση ότι δεν θα χαρακτηριστούν ως spam**.

Συμβατότητα Email – Στατιστικά Εμβέλειας

Για κάθε τύπο email (αποδοχής ή απόρριψης), πραγματοποιήθηκε ανάλυση «market support», με στόχο να διασφαλιστεί η σωστή εμφάνιση και υποστήριξη σε όλες τις πλατφόρμες. Τα αποτελέσματα συνοψίζονται ως εξής:

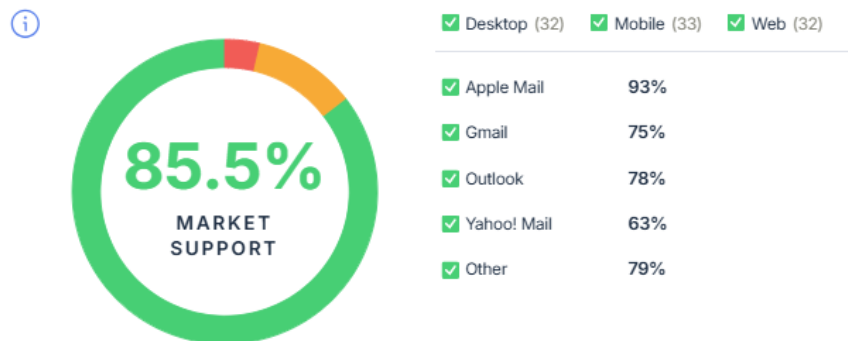
- **Email Αποδοχής Παραγγελίας:**
 - ✓ **Market Support: 84.9%**
 - ✓ Apple Mail: 93%
 - ✓ Gmail: 73%
 - ✓ Outlook: 78%
 - ✓ Yahoo! Mail: 63%
 - ✓ Other: 78%



Εικόνα 86: Στατιστικά συμβατότητας email για την ειδοποίηση αποδοχής παραγγελίας, παρουσιάζοντας την εμβέλεια σε διαφορετικούς πελάτες και πλατφόρμες.

- **Email Απόρριψης Παραγγελίας:**

- ✓ **Market Support: 85.5%**
- ✓ Apple Mail: 93%
- ✓ Gmail: 75%
- ✓ Outlook: 78%
- ✓ Yahoo! Mail: 63%
- ✓ Other: 79%



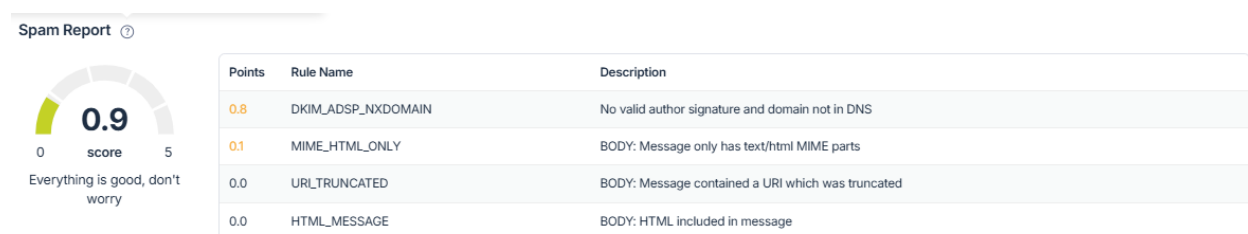
Εικόνα 87: Στατιστικά συμβατότητας email για την ειδοποίηση απόρριψης παραγγελίας, παρουσιάζοντας την εμβέλεια σε διαφορετικούς πελάτες και πλατφόρμες.

Τα ποσοστά αυτά εξασφαλίζουν ότι τα emails παραδίδονται με ορθή απόδοση στο **85% περίπου του συνόλου της αγοράς email clients**, καλύπτοντας όλους τους μεγάλους παρόχους, τόσο σε mobile όσο και σε desktop περιβάλλοντα.

Spam Report – Βαθμολογία Παράδοσης

Και για τους δύο τύπους emails, πραγματοποιήθηκε spam ανάλυση, με σκορ **0.9 / 5**, το οποίο θεωρείται εξαιρετικό και δεν δημιουργεί προβληματισμό:

- **DKIM_ADSP_NXDOMAIN:** 0.8 → Δεν εντοπίζεται DKIM εγγραφή στο DNS (βελτιωτικό, όχι κρίσιμο)
- **MIME_HTML_ONLY:** 0.1 → Χρήση μόνο HTML body (συνηθισμένο σε transactional emails)
- Άλλες τιμές: 0.0



Εικόνα 88: Αναφορά Spam και βαθμολογία παράδοσης για τα email συστήματος, αξιολογώντας την πιθανότητα αποστολής στα εισερχόμενα των χρηστών.

Η τελική εκτίμηση του συστήματος είναι “*Everything is good, don’t worry*”, επιβεβαιώνοντας ότι τα emails παραδίδονται με ασφάλεια, χωρίς να καταλήγουν σε φακέλους ανεπιθύμητης αλληλογραφίας.

9.3 Αξιολόγηση UI/UX της εφαρμογής

Η αξιολόγηση του περιβάλλοντος χρήστη (User Interface – UI) και της συνολικής εμπειρίας χρήσης (User Experience – UX) πραγματοποιήθηκε με ποιοτική ανατροφοδότηση από πραγματικούς χρήστες της εφαρμογής, τόσο από την πλευρά των τελικών χρηστών (φοιτητών που πραγματοποιούν παραγγελίες), όσο και από την πλευρά των διαχειριστών (υπαλλήλων που διαχειρίζονται τα ημερήσια μενού και παρακολουθούν τις μερίδες). Οι συμμετέχοντες κλήθηκαν να χρησιμοποιήσουν την εφαρμογή σε πραγματικά σενάρια χρήσης και να καταγράψουν την εμπειρία τους.

Τα γενικά συμπεράσματα ήταν ιδιαίτερος θετικά, με την εφαρμογή να χαρακτηρίζεται ως **ευχάριστη στη χρήση, καθαρή σχεδιαστικά, και λειτουργική** σε όλες τις συσκευές.

Πιο συγκεκριμένα τα θετικά στοιχεία που αναφέρθηκαν ήταν:

- Η πλοήγηση μέσα στην εφαρμογή ήταν **απλή και ξεκάθαρη**, επιτρέποντας στον χρήστη να ολοκληρώνει τη διαδικασία παραγγελίας χωρίς σύγχυση.
- Η **απόκριση της εφαρμογής** σε κινητές συσκευές (mobile-first προσέγγιση) κρίθηκε πολύ ικανοποιητική.
- Οι **οπτικές ειδοποιήσεις** μέσω της βιβλιοθήκης SweetAlert2 (π.χ. τοστ, pop-up) βοήθησαν στο να παρέχεται άμεση και κατανοητή ενημέρωση στον χρήστη για κάθε του ενέργεια.
- Οι διαχειριστές εκτίμησαν ιδιαίτερα τη **δυνατότητα πρόβλεψης μερίδων**, καθώς τους βοηθάει στον καλύτερο σχεδιασμό του ημερήσιου μενού.
- Τέλος οι διαχειριστές στάθηκαν και στην ευκολία καταχώρησης του μενού μέσω του μηχανισμού drag and drop.

Οι παρατηρήσεις και προτάσεις που συγκεντρώθηκαν από τους χρήστες χωρίζονται στις εξής δύο βασικές κατηγορίες:

- **Προτάσεις από Τελικούς Χρήστες**

1. **Δυνατότητα αλλαγής τύπου γεύματος για την ίδια ημέρα χωρίς επιστροφή στην αρχική οθόνη**
Χρήστες πρότειναν να προστεθεί μηχανισμός (π.χ. dropdown ή tabs) ώστε να μπορούν να μεταβούν από το μεσημεριανό στο βραδινό μενού για την ίδια ημερομηνία **χωρίς να χρειάζεται να γυρίσουν πίσω στην αρχική οθόνη**.
2. **Περισσότερες από μία μερίδες ανά τύπο γεύματος**
Αν και η εφαρμογή συμμορφώνεται με τους κανόνες της πανεπιστημιακής λέσχης (μία μερίδα ανά τύπο γεύματος ανά παραγγελία), υπήρξε ενδιαφέρον για τη δυνατότητα **παραγγελίας περισσότερων μερίδων με επιπλέον κόστος**, κάτι που θα μπορούσε να εφαρμοστεί προαιρετικά στο μέλλον.
3. **Παρακολούθηση κατάστασης παραγγελίας (order tracking)**
Εκφράστηκε η επιθυμία για **ειδοποιήσεις τύπου efood/Wolt** που θα ενημερώνουν τον χρήστη για την πρόοδο της παραγγελίας του (π.χ. «Η παραγγελία παρασκευάζεται», «Είναι έτοιμη», «Βρίσκεται στον δίσκο»).
4. **Αποθήκευση αγαπημένων γευμάτων ή ιστορικού**
Μια επιπλέον ιδέα που προτάθηκε ήταν η δυνατότητα **αποθήκευσης αγαπημένων γευμάτων ή επιλογών**, ώστε ο χρήστης να μπορεί εύκολα να τα επαναλάβει σε μελλοντικές παραγγελίες.

- **Προτάσεις από Διαχειριστές**

1. **Ενσωμάτωση πρόβλεψης μερίδων στη σελίδα δημιουργίας μενού**
Οι διαχειριστές πρότειναν η πρόβλεψη μερίδων να εμφανίζεται **αυτόματα** μέσα

στη σελίδα δήλωσης του ημερήσιου μενού (drag & drop interface), ώστε κατά την επιλογή κάθε φαγητού να φαίνεται και ο προβλεπόμενος αριθμός μερίδων.

2. Αυτόματη δημιουργία μενού βάσει προβλέψεων

Προτάθηκε η ύπαρξη μηχανισμού που να δημιουργεί **προτεινόμενο μενού** για κάθε ημέρα, βάσει των ιστορικών δεδομένων ζήτησης και των προβλεπόμενων μερίδων. Αυτό θα μπορούσε να λειτουργήσει ως οδηγός για την τελική επιλογή των φαγητών από τον διαχειριστή.

3. Ειδοποίηση κατά τη δήλωση μενού σε περίπτωση προβλεπόμενης χαμηλής ή υψηλής ζήτησης

Ζητήθηκε, επιπλέον, η δυνατότητα να εμφανίζεται **προειδοποίηση στον διαχειριστή** (μέσω toast ή alert), όταν δηλώνεται κάποιο φαγητό με **χαμηλή προβλεπόμενη κατανάλωση**, ώστε να ληφθεί υπόψη πριν την τελική υποβολή.

Συμπερασματικά βάσει των παραπάνω κατανοούμε ότι η ανατροφοδότηση από πραγματικούς χρήστες της εφαρμογής αποτέλεσε **πολύτιμο εργαλείο** για την αποτίμηση του επιπέδου χρηστικότητας και αποτελεσματικότητας του UI/UX. Παρά τη συνολική ικανοποίηση από τη λειτουργικότητα και την ευχρηστία, οι παραπάνω παρατηρήσεις προσφέρουν μια **σαφή κατεύθυνση για μελλοντική εξέλιξη της εφαρμογής** με γνώμονα την εμπειρία του χρήστη και την καλύτερη υποστήριξη των διαχειριστικών διαδικασιών.

10. Συμπεράσματα και Μελλοντική Εργασία

10.1 Σύνοψη Συμβολής της Διπλωματικής

Το παρόν έργο αποσκοπούσε στην ανάπτυξη ενός ολοκληρωμένου πληροφοριακού συστήματος διαχείρισης φοιτητικής λέσχης, το οποίο να ανταποκρίνεται στις ανάγκες τόσο των τελικών χρηστών (φοιτητών) όσο και των διαχειριστών της μονάδας σίτισης. Με αφετηρία την παρατήρηση υπαρχουσών αναποτελεσματικών διαδικασιών και την απουσία τεχνολογικής υποστήριξης σε πολλές φοιτητικές λέσχες, σχεδιάστηκε ένα σύστημα που επιτρέπει τη διαχείριση κάθε σταδίου της λειτουργίας της λέσχης μέσα από μία ενιαία, αποδοτική πλατφόρμα.

Η συμβολή του έργου είναι ουσιαστική, καθώς προσφέρει ένα λειτουργικό και ευέλικτο περιβάλλον στο οποίο ο φοιτητής μπορεί να πραγματοποιεί παραγγελίες, να ελέγχει το ιστορικό του, να διαχειρίζεται voucher, και να αλληλεπιδρά με το σύστημα με ευκολία και ασφάλεια. Από την άλλη πλευρά, ο διαχειριστής έχει στη διάθεσή του πλήρες φάσμα εργαλείων που του επιτρέπουν να οργανώσει τη λέσχη με τρόπο που δεν ήταν εφικτός μέχρι σήμερα: από τον καθορισμό μενού και ωραρίων, μέχρι τον έλεγχο διαθεσιμότητας, την πρόβλεψη μερίδων και τον έλεγχο voucher.

Η υλοποίηση ενός μηχανισμού πρόβλεψης μερίδων που βασίζεται σε ιστορικά δεδομένα ενδυναμώνει περαιτέρω τη συμβολή του έργου, καθώς εισάγει στοιχείο προγνωστικής ανάλυσης σε έναν χώρο όπου παραδοσιακά κυριαρχεί η εμπειρική εκτίμηση. Έτσι, η εφαρμογή λειτουργεί όχι μόνο ως εργαλείο καθημερινής χρήσης, αλλά και ως υποδομή για τον ψηφιακό μετασχηματισμό των φοιτητικών μονάδων σίτισης.

10.2 Δυνατά Σημεία και Καινοτομία

Το σύστημα που αναπτύχθηκε περιλαμβάνει πληθώρα καινοτόμων στοιχείων, τα οποία ενισχύουν τη λειτουργικότητα, την αποδοτικότητα και την εμπειρία χρήσης. Πρώτα απ' όλα, η πρόβλεψη μερίδων βάσει ιστορικών δεδομένων αποτελεί μια σημαντική προσθήκη που μετατρέπει την παρασκευή φαγητού από στατική διαδικασία σε δυναμική και προσαρμοστική. Μέσα από τον υπολογισμό αναμενόμενης ζήτησης για κάθε είδος φαγητού, ο διαχειριστής μπορεί πλέον να λαμβάνει πιο τεκμηριωμένες αποφάσεις, αποφεύγοντας τη σπατάλη ή την έλλειψη.

Ένα ακόμη σημείο καινοτομίας είναι η ενσωμάτωση ψηφιακού πορτοφολιού (voucher), το οποίο επιτρέπει διαφάνεια και ακρίβεια στις συναλλαγές, καθώς και η λειτουργία αυτοματοποιημένων ειδοποιήσεων μέσω email για την ενημέρωση των φοιτητών αναφορικά με την κατάσταση των παραγγελιών τους.

Η αρχιτεκτονική του συστήματος σχεδιάστηκε με modular προσέγγιση και έμφαση στην ασφάλεια. Η χρήση tokens, μηχανισμών refresh και έλεγχος πρόσβασης στα API ενισχύει την προστασία των δεδομένων και εξασφαλίζει τη σταθερή λειτουργία του συστήματος. Ταυτόχρονα, η χρήση UML διαγραμμάτων για την απεικόνιση των ροών μεταξύ frontend και backend καθιστά το σύστημα απόλυτα τεκμηριωμένο και εύκολα επεκτάσιμο ή συντηρήσιμο.

Η φιλικότητα προς τον χρήστη, η καθαρή διασύνδεση λειτουργιών και η δυνατότητα εύκολης διαχείρισης παραμέτρων από τον διαχειριστή συνιστούν επιπλέον στοιχεία που ενισχύουν την αξιοπιστία και αποδοχή της εφαρμογής.

10.3 Περιορισμοί του Συστήματος

Παρόλο που το σύστημα καλύπτει ένα ευρύ φάσμα λειτουργιών, υπάρχουν ορισμένοι περιορισμοί που αξίζει να αναφερθούν και να αξιολογηθούν. Πρωτίστως, ο μηχανισμός πρόβλεψης βασίζεται αποκλειστικά σε εσωτερικά ιστορικά δεδομένα χωρίς να λαμβάνει υπόψη εξωτερικούς παράγοντες που ενδέχεται να επηρεάζουν τη ζήτηση. Επομένως, σε περιπτώσεις που παρουσιάζονται εξωγενείς αποκλίσεις, η ακρίβεια της πρόβλεψης μπορεί να μειωθεί.

Επιπλέον, η εφαρμογή έχει σχεδιαστεί με επίκεντρο τη χρήση μέσω υπολογιστή και όχι κινητής συσκευής, κάτι που περιορίζει την προσβασιμότητα από το ευρύ κοινό που χρησιμοποιεί smartphones για καθημερινές συναλλαγές. Το περιβάλλον είναι responsive αλλά δεν καλύπτει εξ ολοκλήρου τις δυνατότητες ενός native mobile περιβάλλοντος.

Ένας ακόμα περιορισμός αφορά τη μονοκαναλική ειδοποίηση των χρηστών, η οποία περιορίζεται στα email. Παρότι αυτή η επιλογή είναι αξιόπιστη, σε περιπτώσεις όπως λανθασμένες διευθύνσεις email ή περιορισμένη πρόσβαση των χρηστών στην αλληλογραφία τους, ενδέχεται να προκύψουν προβλήματα ενημέρωσης.

Τέλος, απουσιάζει ένα δομημένο σύστημα αξιολόγησης της απόδοσης και ποιότητας των παρεχόμενων υπηρεσιών, π.χ. μέσα από feedback χρηστών ή στατιστική αποτύπωση συμπεριφορών, κάτι που θα μπορούσε να συνεισφέρει στη μελλοντική αναβάθμιση της εφαρμογής.

10.4 Προτάσεις για Μελλοντική Βελτίωση

Με βάση τους παραπάνω περιορισμούς, μπορούν να διατυπωθούν συγκεκριμένες προτάσεις για τη βελτίωση και επέκταση του συστήματος. Μία από τις πιο σημαντικές είναι η ενίσχυση του μηχανισμού πρόβλεψης με χρήση εξωτερικών παραμέτρων, όπως δεδομένα καιρού, ακαδημαϊκό

ημερολόγιο, ή ακόμα και API που σχετίζονται με φοιτητικές δραστηριότητες.

Παράλληλα, θα ήταν χρήσιμη η δημιουργία δυναμικού πίνακα ελέγχου (dashboard) για τον διαχειριστή, ο οποίος θα περιλαμβάνει στατιστικά, δείκτες απόδοσης και συγκεντρωτικά δεδομένα χρήσης που θα του επιτρέπουν να αξιολογεί και να προβλέπει με μεγαλύτερη ακρίβεια. Επίσης, η ενσωμάτωση ενός μηχανισμού αξιολόγησης από τους φοιτητές (feedback για την ποιότητα του φαγητού, την ακρίβεια της πρόβλεψης κ.λπ.) θα προσέφερε πολύτιμες πληροφορίες για τη βελτίωση της εμπειρίας χρήσης.

Άλλες σημαντικές προτάσεις περιλαμβάνουν την υποστήριξη και άλλων καναλιών ειδοποιήσεων, όπως push notifications ή SMS, και τη δημιουργία ενιαίας πολιτικής ενημέρωσης που να καλύπτει όλες τις πλατφόρμες. Η βελτιστοποίηση του UI για χρήση από κινητές συσκευές κρίνεται επίσης απαραίτητη για την κάλυψη των σύγχρονων αναγκών.

10.5 Δυνατότητες Επέκτασης (π.χ. mobile app, AI βελτιστοποίηση, chatbot)

Η αρχιτεκτονική του συστήματος είναι κατάλληλη για μελλοντική επεκτασιμότητα και ενσωμάτωση νέων τεχνολογιών. Μία από τις πιο προφανείς δυνατότητες είναι η ανάπτυξη **native mobile εφαρμογής**, η οποία θα επιτρέπει στους φοιτητές να έχουν άμεση πρόσβαση στην πλατφόρμα, με δυνατότητα offline mode και in-app ειδοποιήσεις.

Επιπλέον, η ενσωμάτωση μηχανισμών **τεχνητής νοημοσύνης** για την αυτόματη πρόβλεψη ζήτησης, τη βελτιστοποίηση του μενού, ή ακόμα και την προσαρμογή προτάσεων φαγητού ανάλογα με τις διατροφικές προτιμήσεις, θα μπορούσε να εκτοξεύσει τη λειτουργικότητα της εφαρμογής σε νέο επίπεδο.

Μία ακόμα ενδιαφέρουσα προοπτική είναι η ανάπτυξη **chatbot υποστήριξης**, που θα επιτρέπει στους φοιτητές να λαμβάνουν απαντήσεις σε ερωτήσεις, να καθοδηγούνται στη διαδικασία παραγγελίας και να εξοικειώνονται πιο εύκολα με τη χρήση της πλατφόρμας. Παράλληλα, το σύστημα μπορεί να διασυνδεθεί με **πλατφόρμες διοίκησης φοιτητικών ιδρυμάτων**, προκειμένου να αυτοματοποιείται η ενημέρωση στοιχείων όπως το δικαίωμα σίτισης ή η φοιτητική ιδιότητα.

Τέλος, υπάρχει δυνατότητα διαχωρισμού ρόλων διαχείρισης — για παράδειγμα, δημιουργία υπολογαριασμών για προσωπικό κουζίνας, οικονομικό τμήμα ή διοίκηση — με διαφορετικά επίπεδα πρόσβασης, ώστε το σύστημα να προσαρμόζεται στις οργανωτικές ανάγκες μεγαλύτερων μονάδων σίτισης ή πανεπιστημιακών ιδρυμάτων.

Παράρτημα Α – Ερωτηματολόγιο Χρηστών

A.1 Γενικές Πληροφορίες

- Ηλικία χρήστη
- Έχετε χρησιμοποιήσει παρόμοιες εφαρμογές; Αν ναι, ποια ήταν η εμπειρία σας;
- Πόσο συχνά χρησιμοποιείτε κινητό για παραγγελίες;
- Ποιες είναι οι αγαπημένες σας λειτουργίες και ποιες οι απογοητεύσεις σας από αντίστοιχες εφαρμογές;

A.2 Πριν την Παρουσίαση του Wireframe

- Τι περιμένετε να μπορείτε να κάνετε με την εφαρμογή;
- Πώς φαντάζεστε την εμφάνιση και λειτουργικότητά της;
- Ποιες λειτουργίες θεωρείτε απολύτως απαραίτητες;
- Ποιες πληροφορίες θα θέλατε να είναι άμεσα προσβάσιμες;
- Ποιες ανάγκες σας θα μπορούσε να καλύψει;

A.3 Κατά την Παρουσίαση του Wireframe

- Ποια είναι η πρώτη σας εντύπωση από τη διάταξη της σελίδας;
- Σας φαίνεται ελκυστική;
- Καταλαβαίνετε με την πρώτη ματιά τη λειτουργία των στοιχείων;
- Είναι εύχρηστη και κατανοητή;
- Υπάρχει κάποιο στοιχείο που σας φαίνεται περιττό ή ασαφές;
- Ποια λειτουργικότητα ξεχωρίζει;
- Αν έπρεπε να περιγράψετε την εμπειρία σας με μια λέξη, ποια θα ήταν;
- Πώς θα βελτιώσετε την εμφάνιση ή τη λειτουργικότητα;

A.4 Μετά τη Χρήση του Wireframe

- Ήταν εύκολη η χρήση της εφαρμογής;
- Είστε ικανοποιημένοι από τη συνολική εμπειρία;
- Αν είχατε ένα μαγικό ραβδί, τι θα αλλάζατε;
- Πόσο πιθανό είναι να χρησιμοποιούσατε την εφαρμογή στο μέλλον;
- Τι άλλο θα θέλατε να περιλαμβάνει;
- Θα τη συστήνατε σε φίλους;

A.5 Tasks Εκτελέσιμα Κατά τη Δοκιμή

- Επιλογή ημερομηνίας και τύπου γεύματος
- Προσθήκη φαγητών στο καλάθι
- Φιλτράρισμα φαγητών ανά κατηγορία
- Σύνοψη παραγγελίας και τροποποιήσεις
- Επιλογή τρόπου παραγγελίας, timeslot και μεθόδου πληρωμής

- Προσθήκη σχολίων στην παραγγελία
- Αποστολή παραγγελίας
- Διαχείριση ειδοποιήσεων (επιτυχία, αποτυχία)
- Επίσκεψη στο προφίλ και προβολή ιστορικού
- Εμφάνιση αγαπημένων φαγητών
- Διαγραφή λογαριασμού

A.6 Μετά την Εκτέλεση των Tasks

- Ολοκληρώθηκε το task με επιτυχία;
- Αν όχι, τι δυσκόλεψε την υλοποίησή του;
- Πόσο εύκολο ήταν να εντοπίσετε την απαιτούμενη λειτουργία;
- Ήταν λογικό το layout και η ροή της εφαρμογής;
- Υπήρχαν ασάφειες σε κείμενα ή κουμπιά;
- Η εναλλαγή των οθονών ήταν κατανοητή και σε σωστή σειρά;

Αναφορές

- [1] Πολυτεχνείο Κρήτης, *Προτεινόμενο Εβδομαδιαίο Πρόγραμμα Σίτισης*, διαθέσιμο στο: <https://www.tuc.gr/fileadmin/various/menu/TUC-restaurant-menu.pdf> (πρόσβαση: 29/09/2025).
- [2] *Wireframes*, διαθέσιμο στο: <https://www.interaction-design.org/literature/topics/wireframe> (πρόσβαση: 02/10/2025).
- [3] *Balsamiq Wireframes*, διαθέσιμο στο: <https://balsamiq.com/> (πρόσβαση: 02/10/2025).
- [4] *Figma*, διαθέσιμο στο: <https://www.figma.com/> (πρόσβαση: 02/10/2025).
- [5] Tailwind CSS, διαθέσιμο στο: <https://tailwindcss.com/> (πρόσβαση: 02/10/2025).
- [6] SweetAlert2, διαθέσιμο στο: <https://sweetalert2.github.io/> (πρόσβαση: 02/10/2025).
- [7] Swiper, διαθέσιμο στο: <https://swiperjs.com/> (πρόσβαση: 02/10/2025).
- [8] Tabulator, διαθέσιμο στο: <https://tabulator.info/> (πρόσβαση: 02/10/2025).
- [9] PHP, διαθέσιμο στο: <https://www.php.net/> (πρόσβαση: 02/10/2025).
- [10] JSON Web Token (JWT), διαθέσιμο στο: <https://jwt.io/> (πρόσβαση: 02/10/2025).
- [11] REST API, διαθέσιμο στο: <https://restfulapi.net/> (πρόσβαση: 02/10/2025).
- [12] Postman, διαθέσιμο στο: <https://www.postman.com/> (πρόσβαση: 02/10/2025).
- [13] MySQL, διαθέσιμο στο: <https://www.mysql.com/> (πρόσβαση: 02/10/2025).
- [14] ER (Entity-Relationship) διαγράμματα, διαθέσιμο στο: https://en.wikipedia.org/wiki/Entity–relationship_model (πρόσβαση: 02/10/2025).
- [15] Python, διαθέσιμο στο: <https://www.python.org/> (πρόσβαση: 02/10/2025).
- [16] Facebook Prophet, διαθέσιμο στο: <https://facebook.github.io/prophet/> (πρόσβαση: 02/10/2025).
- [17] Git, διαθέσιμο στο: <https://git-scm.com/> (πρόσβαση: 02/10/2025).
- [18] GitHub, διαθέσιμο στο: <https://github.com/> (πρόσβαση: 02/10/2025).
- [19] Visual Studio Code (VS Code), διαθέσιμο στο: <https://code.visualstudio.com/> (πρόσβαση: 02/10/2025).
- [20] Mailtrap, διαθέσιμο στο: <https://mailtrap.io/> (πρόσβαση: 02/10/2025).

[21] Miro, διαθέσιμο στο: <https://miro.com/> (πρόσβαση: 02/10/2025).

[22] DrawSQL, διαθέσιμο στο: <https://drawsql.app/> (πρόσβαση: 02/10/2025).

[23] Human-Centered Design - HCD, διαθέσιμο στη διεύθυνση:
<https://cdn.standards.iteh.ai/samples/77520/8cac787a9e1549e1a7ffa0171dfa33e0/ISO-9241-210-2019.pdf>
(πρόσβαση: 02/10/2025).

[24] Συμπληρωματικό Υλικό Διπλωματικής Εργασίας – Εικόνες Υψηλής Ανάλυσης, διαθέσιμο στο: <https://github.com/aboutalis/thesis-figures>, 2025.