



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

**ΠΡΟΗΓΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΠΛΗΡΟΦΟΡΙΚΗΣ - ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΤΕΧΝΗΤΗΣ
ΝΟΗΜΟΣΥΝΗΣ**

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Διπλωματική Εργασία

Τίτλος Διπλωματικής Εργασίας	(Ελληνικά) Ανάπτυξη εφαρμογής πλοήγησης με προτεινόμενες διαδρομές ανάλογα τις προτιμήσεις του χρήστη (Αγγλικά) Development of a navigation application with recommended routes according to user preferences
Ονοματεπώνυμο Φοιτητή	Νικόλαος Σαρρής
Πατρώνυμο	Ευάγγελος
Αριθμός Μητρώου	ΜΠΣΠ/24043
Επιβλέπουσα	Κωνσταντίνα Χρυσafiάδη, Επίκουρη Καθηγήτρια

Ημερομηνία Παράδοσης Ιούλιος 2026

Copyright

Copyright © Νικόλαος Σαρρής, 2026.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για μη κερδοσκοπικό, εκπαιδευτικό ή ερευνητικό σκοπό, με την προϋπόθεση ότι αναφέρεται η πηγή προέλευσης.

Οι απόψεις και τα συμπεράσματα που περιέχονται στην εργασία εκφράζουν τον συγγραφέα και δεν αντιπροσωπεύουν απαραίτητα τις επίσημες θέσεις του Πανεπιστημίου Πειραιώς.

Τριμελής Εξεταστική Επιτροπή

Χρυσafiάδη Κωνσταντίνα
Επίκουρη Καθηγήτρια

Ευθύμιος Αλέπης
Καθηγητής

Μαρία Βίβρου
Καθηγήτρια

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την επιβλέπουσα καθηγήτριά μου, Κωνσταντίνα Χρυσafiάδη, για την καθοδήγηση, την υποστήριξη και τις παρατηρήσεις της κατά την εκπόνηση της παρούσας διπλωματικής εργασίας.

Επίσης, ευχαριστώ την οικογένειά μου για τη συνεχή στήριξη και ενθάρρυνση σε όλη τη διάρκεια των σπουδών μου.

Περίληψη

Η παρούσα διπλωματική εργασία παρουσιάζει τον σχεδιασμό και την ανάπτυξη της SmartNavix, μιας εφαρμογής πλοήγησης που προτείνει διαδρομές λαμβάνοντας υπόψη τις προτιμήσεις του χρήστη. Η εργασία εξετάζει το πρόβλημα της εξατομικευμένης επιλογής διαδρομής, αναλύει τις απαιτήσεις του συστήματος και παρουσιάζει την αρχιτεκτονική της εφαρμογής, τη βάση δεδομένων και τις βασικές λειτουργίες της.

Ιδιαίτερη έμφαση δίνεται στη χρήση fuzzy logic για την επιλογή της καταλληλότερης προτίμησης, καθώς και στην αξιοποίηση υπηρεσιών χαρτών και τεχνητής νοημοσύνης. Στόχος είναι η δημιουργία μιας λύσης που βελτιώνει την εμπειρία χρήστη και επιτρέπει πιο προσαρμοσμένες προτάσεις διαδρομών.

Επιστημονική περιοχή: Έξυπνες εφαρμογές, συστήματα πλοήγησης και προσαρμοστικά συστήματα.

Λέξεις Κλειδιά: SmartNavix, πλοήγηση, εξατομίκευση, fuzzy logic, εμπειρία χρήστη.

Abstract

This dissertation presents the design and development of SmartNavix, a navigation application that recommends routes according to user preferences. The work examines the problem of personalized route selection, analyzes system requirements, and presents the application's architecture, database design, and core functionality.

Special emphasis is placed on the use of fuzzy logic for selecting the most suitable user preference, as well as on the integration of map services and artificial intelligence APIs. The objective is to provide a solution that improves user experience and supports more adaptive route recommendations.

Scientific area: Smart applications, navigation systems, and adaptive systems.

Key Words: SmartNavix, navigation, personalization, fuzzy logic, user experience.

Πίνακας Περιεχομένων

Copyright.....	i
Ευχαριστίες	ii
Περίληψη.....	iii
Abstract	iii
Πίνακας Περιεχομένων	iv
Κατάλογος Εικόνων	viii
Κατάλογος Πινάκων.....	ix
Κατάλογος Διαγραμμάτων	x
Κεφάλαιο 1: Εισαγωγή.....	1
1.1 Αντικείμενο της διπλωματικής εργασίας.....	1
1.2 Περιγραφή προβλήματος.....	1
1.3 Στόχοι της εργασίας.....	1
1.4 Κίνητρο και σημασία του θέματος	2
1.5 Σύντομη περιγραφή της προτεινόμενης λύσης SmartNavix	2
1.6 Μεθοδολογική προσέγγιση.....	2
1.7 Δομή της εργασίας.....	2
Κεφάλαιο 2: Βιβλιογραφική Ανασκόπηση.....	4
2.1 Ολοκληρωμένα συστήματα λογισμικού	4
2.2 Web-based applications.....	4
2.3 Mobile applications	4
2.4 Smart applications και smart systems.....	5
2.5 Intelligent applications και intelligent systems.....	5
2.6 Artificial Intelligence σε σύγχρονες εφαρμογές	5
2.7 User Experience και ευχρηστία εφαρμογών	6
2.8 Adaptivity και adaptive applications/systems	6
2.9 User-computer interaction	6
2.10 User interfaces και σχεδιασμός διεπαφών	6
2.11 Συστήματα πλοήγησης και αλγόριθμοι διαδρομών	7
2.12 Εξατομίκευση και συστήματα προτάσεων	7
2.13 Fuzzy logic σε συστήματα λήψης αποφάσεων	7
2.14 Χρήση LLMs και AI APIs σε εφαρμογές τοποθεσίας.....	8
2.15 Σύνοψη βιβλιογραφίας και ερευνητικό κενό	8
Κεφάλαιο 3: Παρόμοιες Εφαρμογές και Διαφοροποίηση της SmartNavix.....	9
3.1 Υφιστάμενες εφαρμογές πλοήγησης	9
3.2 Google Maps	9
3.3 Waze.....	9
3.4 Apple Maps και άλλες εφαρμογές χαρτών	10
3.5 Εφαρμογές με προτάσεις σημείων ενδιαφέροντος	10

3.6 Περιορισμοί των υπαρχουσών λύσεων.....	10
3.7 Διαφοροποίηση της SmartNavix	10
3.8 Συγκριτικός πίνακας λειτουργιών.....	11
Κεφάλαιο 4: Ανάλυση Απαιτήσεων	12
4.1 Λειτουργικές απαιτήσεις	12
4.2 Μη λειτουργικές απαιτήσεις.....	13
4.3 Ρόλοι χρηστών: απλός χρήστης, διαχειριστής, contributor	13
4.4 Use cases και σενάρια χρήσης.....	14
4.5 UML διάγραμμα περιπτώσεων χρήσης	14
4.6 Περιορισμοί εγκατάστασης και χρήσης	15
4.7 Απαιτήσεις ασφάλειας και προστασίας δεδομένων.....	15
Κεφάλαιο 5: Σχεδιασμός Συστήματος.....	17
5.1 Συνολική αρχιτεκτονική SmartNavix	17
5.2 Frontend εφαρμογή.....	18
5.3 Backend υπηρεσίες.....	18
5.4 Βάση δεδομένων.....	18
5.5 Εξωτερικά APIs: Google Maps και Gemini	19
5.6 Ροή δεδομένων στο σύστημα	19
5.7 UML / sequence diagrams βασικών λειτουργιών.....	20
5.8 Πολυγλωσσική υποστήριξη.....	20
5.9 Επιλογή τεχνολογιών και λόγοι επιλογής.....	21
Κεφάλαιο 6: Σχεδιασμός Βάσης Δεδομένων.....	22
6.1 Περιγραφή μοντέλου δεδομένων.....	22
6.2 Πίνακες χρηστών και ρόλων	23
6.3 Πίνακες διαδρομών, στάσεων και σημείων ενδιαφέροντος	23
6.4 FilteredPreference και αποθήκευση προτιμήσεων	23
6.5 Presets και εξατομίκευση χρήστη.....	24
6.6 Συσχετίσεις οντοτήτων	24
6.7 ER διάγραμμα.....	25
6.8 Κανόνες ακεραιότητας και απόδοσης	25
Κεφάλαιο 7: Υλοποίηση Βασικών Λειτουργιών	27
7.1 Αναζήτηση διαδρομής.....	27
7.2 Εναλλακτικές διαδρομές.....	27
7.3 Φίλτρα και επιλογές προτιμήσεων	28
7.4 Προτάσεις σημείων ενδιαφέροντος	28
7.5 Presets και custom icons.....	28
7.6 Αξιολόγηση διαδρομής από τον χρήστη.....	29
7.7 Feedback loop και βελτίωση προτάσεων.....	29
7.8 Admin panel και analytics	29
7.9 Διαχείριση σφαλμάτων και περιπτώσεων αποτυχίας	30
Κεφάλαιο 8: Μοντέλο Fuzzy Logic για Επιλογή Προτιμήσεων	31

8.1 Περιγραφή του προβλήματος επιλογής best preference	31
8.2 Είσοδοι του μοντέλου	31
8.3 Frequency, recency, consistency και completeness	32
8.4 Membership functions	32
8.5 Κανόνες IF-THEN	33
8.6 Defuzzification και τελικό score	33
8.7 Παραμετροποίηση μέσω appsettings	33
8.8 Σύγκριση fuzzy προσέγγισης με AI-only προσέγγιση	34
Κεφάλαιο 9: Παραδείγματα Χρήσης με Screenshots	35
9.1 Είσοδος και αρχική οθόνη της εφαρμογής	35
9.2 Αναζήτηση διαδρομής από τον χρήστη	36
9.3 Προβολή βασικής προτεινόμενης διαδρομής	37
9.4 Εμφάνιση εναλλακτικών διαδρομών	38
9.5 Χρήση φίλτρων και προτιμήσεων	38
9.6 Προτάσεις σημείων ενδιαφέροντος	39
9.7 Επιλογή best preference από το σύστημα	39
9.8 Αξιολόγηση διαδρομής και feedback χρήστη	40
9.9 Διαχείριση από admin panel	40
9.10 Παρατηρήσεις από τα παραδείγματα χρήσης	41
Κεφάλαιο 10: Εγχειρίδιο Χρήστη	42
10.1 Σκοπός και κοινό του εγχειριδίου	42
10.2 Σύνδεση στην εφαρμογή	42
10.3 Δημιουργία και βασική διαχείριση λογαριασμού	42
10.4 Χρήση βασικής αναζήτησης διαδρομής	43
10.5 Ρύθμιση φίλτρων και προτιμήσεων	43
10.6 Χρήση γρήγορων presets	43
10.7 Έναρξη, παρακολούθηση και ακύρωση πλοήγησης	43
10.8 Αξιολόγηση διαδρομής και ανατροφοδότηση	44
10.9 Χρήση προφίλ και προσωπικών ρυθμίσεων	44
10.10 Χρήση admin panel και αντιμετώπιση συχνών προβλημάτων	44
Κεφάλαιο 11: Οφέλη της Προτεινόμενης Λύσης	45
11.1 Γενική αποτίμηση της προτεινόμενης λύσης	45
11.2 Οφέλη για τον τελικό χρήστη	45
11.3 Βελτίωση εμπειρίας χρήστη και ευχρηστίας	45
11.4 Οφέλη από την εξατομίκευση και την προσαρμοστικότητα	46
11.5 Υποστήριξη λήψης απόφασης στη διαδρομή	46
11.6 Οφέλη για διαχείριση, εποπτεία και βελτίωση της εφαρμογής	46
11.7 Τεχνικά και εκπαιδευτικά οφέλη της υλοποίησης	47
11.8 Περιορισμοί στην αποτίμηση των οφελών	47
Κεφάλαιο 12: Αξιολόγηση και Ασφάλεια	48
12.1 Σκοπός της αξιολόγησης	48

12.2 Κριτήρια αξιολόγησης.....	48
12.3 Λειτουργική αξιολόγηση βασικών ροών	48
12.4 Αξιολόγηση εμπειρίας χρήστη	49
12.5 Αξιολόγηση προσαρμοστικότητας και προτάσεων	49
12.6 Τεχνική αξιολόγηση και περιορισμοί	49
12.7 Ασφάλεια ταυτοποίησης, ρόλων και πρόσβασης	50
12.8 Προστασία προσωπικών δεδομένων και ιδιωτικότητα.....	50
12.9 Συνολική αποτίμηση.....	50
Κεφάλαιο 13: Συμπεράσματα και Μελλοντικές Επεκτάσεις.....	52
13.1 Σύνοψη της διπλωματικής εργασίας.....	52
13.2 Βασικά συμπεράσματα	52
13.3 Συνεισφορά της SmartNavix	52
13.4 Περιορισμοί της εργασίας	53
13.5 Προβλήματα και δυσκολίες που συναντήθηκαν.....	53
13.6 Μελλοντικές επεκτάσεις.....	53
13.7 Τελική αποτίμηση.....	54
Πίνακας Ορολογίας	55
Συντμήσεις - Αρκτικόλεξα - Ακρωνύμια	56
Βιβλιογραφία.....	57
Παράρτημα Α: API συμβόλαιο request/response	62
Παράρτημα Β: SQL scripts και migration σημειώσεις.....	63
Παράρτημα Γ: Ρυθμίσεις εγκατάστασης και εκτέλεσης.....	64
Παράρτημα Δ: Πρόσθετα UML / ER / sequence diagrams	65
Παράρτημα Ε: Επιπλέον screenshots εφαρμογής.....	66

Κατάλογος Εικόνων

ΕΙΚΟΝΑ 9.1: Οθόνη εισόδου SmartNavix	35
ΕΙΚΟΝΑ 9.2: Αρχική οθόνη πλοήγησης	36
ΕΙΚΟΝΑ 9.3: Εισαγωγή προορισμού από τον χρήστη	37
ΕΙΚΟΝΑ 9.4: Φίλτρα διαδρομής και επιβεβαίωση αποθηκευμένων επιλογών	39
ΕΙΚΟΝΑ 9.5: Απόκριση συστήματος με προτεινόμενη διαδρομή	38
ΕΙΚΟΝΑ 9.6: Αξιολόγηση διαδρομής μετά την έναρξη πλοήγησης	40
ΕΙΚΟΝΑ 9.7: Πίνακας διαχείρισης SmartNavix	41

Κατάλογος Πινάκων

ΠΙΝΑΚΑΣ 3.1: Συγκριτική αποτύπωση παρόμοιων εφαρμογών και SmartNavix	11
ΠΙΝΑΚΑΣ 4.1: Ρόλοι χρηστών και βασικές δυνατότητες	14
ΠΙΝΑΚΑΣ 4.2: Λειτουργικές απαιτήσεις SmartNavix	12
ΠΙΝΑΚΑΣ 4.3: Μη λειτουργικές απαιτήσεις και περιορισμοί	13
ΠΙΝΑΚΑΣ 5.1: Τεχνολογίες υλοποίησης και ρόλος στο σύστημα	21
ΠΙΝΑΚΑΣ 6.1: Κύριες οντότητες βάσης δεδομένων SmartNavix	22
ΠΙΝΑΚΑΣ 6.2: Βασικές συσχετίσεις και κανόνες διαγραφής	24
ΠΙΝΑΚΑΣ 7.1: Υλοποίηση βασικών λειτουργιών SmartNavix	30
ΠΙΝΑΚΑΣ 8.1: Είσοδοι του fuzzy μοντέλου επιλογής best preference	32
ΠΙΝΑΚΑΣ 8.2: Membership categories και υλοποίηση	33
ΠΙΝΑΚΑΣ 8.3: Παράμετροι FilteredPreferenceResolution	34
ΠΙΝΑΚΑΣ O.1: Πίνακας ορολογίας SmartNavix	55
ΠΙΝΑΚΑΣ Σ.1: Συντμήσεις, αρκτικόλεξα και ακρωνύμια	56
ΠΙΝΑΚΑΣ A.1: Ενδεικτικά API endpoints SmartNavix	62
ΠΙΝΑΚΑΣ B.1: Σύνοψη βασικών πινάκων και σημειώσεων βάσης	63
ΠΙΝΑΚΑΣ Γ.1: Βασικά βήματα εγκατάστασης και εκτέλεσης	64
ΠΙΝΑΚΑΣ Δ.1: Σύνοψη διαγραμμάτων της εργασίας	65
ΠΙΝΑΚΑΣ E.1: Ευρετήριο screenshots εφαρμογής	66

Κατάλογος Διαγραμμάτων

ΔΙΑΓΡΑΜΜΑ 4.1: UML διάγραμμα περιπτώσεων χρήσης SmartNavix	15
ΔΙΑΓΡΑΜΜΑ 5.1: Συνολική αρχιτεκτονική SmartNavix	17
ΔΙΑΓΡΑΜΜΑ 5.2: Ροή αναζήτησης διαδρομής SmartNavix	20
ΔΙΑΓΡΑΜΜΑ 6.1: ER διάγραμμα βάσης δεδομένων SmartNavix	25
ΔΙΑΓΡΑΜΜΑ 7.1: Ροή υλοποίησης βασικών λειτουργιών SmartNavix	27
ΔΙΑΓΡΑΜΜΑ 8.1: Ροή fuzzy resolver για best preference	31

Κεφάλαιο 1: Εισαγωγή

Το κεφάλαιο αυτό παρουσιάζει το γενικό πλαίσιο της διπλωματικής εργασίας, το πρόβλημα που επιχειρεί να αντιμετωπίσει η εφαρμογή SmartNavix, τους στόχους της εργασίας και τη διάρθρωση του κειμένου.

1.1 Αντικείμενο της διπλωματικής εργασίας

Η παρούσα διπλωματική εργασία έχει ως αντικείμενο τη μελέτη, τον σχεδιασμό και την ανάπτυξη της εφαρμογής SmartNavix, μιας web-based εφαρμογής πλοήγησης που δεν περιορίζεται στην απλή εύρεση μιας διαδρομής, αλλά επιχειρεί να υποστηρίξει τον χρήστη με προτεινόμενες διαδρομές προσαρμοσμένες στις προτιμήσεις και στο πλαίσιο χρήσης του. Η εργασία εντάσσεται στην περιοχή των έξυπνων εφαρμογών, της αλληλεπίδρασης ανθρώπου-υπολογιστή (human-computer interaction) και των προσαρμοστικών συστημάτων (adaptive systems).

Η βασική ιδέα της SmartNavix είναι ότι η επιλογή διαδρομής δεν αποτελεί πάντοτε καθαρά γεωγραφικό ή χρονικό πρόβλημα. Στην πράξη, διαφορετικοί χρήστες μπορεί να επιλέγουν διαφορετική διαδρομή για την ίδια αφετηρία και τον ίδιο προορισμό, επειδή λαμβάνουν υπόψη παράγοντες όπως η άνεση, τα σημεία ενδιαφέροντος, η εξοικείωση με μια περιοχή, η διάθεση για περπάτημα ή η ανάγκη αποφυγής συγκεκριμένων συνθηκών.

Στο πλαίσιο της εργασίας παρουσιάζονται η ανάλυση απαιτήσεων, η αρχιτεκτονική του συστήματος, η βάση δεδομένων, οι βασικές λειτουργίες της εφαρμογής, η αξιοποίηση εξωτερικών υπηρεσιών χαρτών και τεχνητής νοημοσύνης, καθώς και ο μηχανισμός επιλογής προτιμήσεων με χρήση fuzzy logic.

1.2 Περιγραφή προβλήματος

Οι σύγχρονες εφαρμογές πλοήγησης προσφέρουν στον χρήστη γρήγορη πρόσβαση σε χάρτες, οδηγίες και εναλλακτικές διαδρομές. Παρόλα αυτά, οι περισσότερες λύσεις δίνουν προτεραιότητα σε μετρήσιμα κριτήρια, όπως ο συνολικός χρόνος μετακίνησης, η απόσταση ή η κυκλοφοριακή κατάσταση.

Το πρόβλημα που εξετάζει η παρούσα εργασία είναι η ανάγκη για μια εφαρμογή πλοήγησης που να μπορεί να συνυπολογίζει τις προτιμήσεις του χρήστη και να προτείνει διαδρομές με πιο εξατομικευμένο τρόπο. Ένας χρήστης μπορεί να ενδιαφέρεται για σύντομη διαδρομή, ενώ ένας άλλος μπορεί να προτιμά μια διαδρομή με περισσότερα σημεία ενδιαφέροντος ή με καλύτερη εμπειρία χρήσης.

Επιπλέον, οι προτιμήσεις δεν είναι πάντα απόλυτες ή εύκολα μετρήσιμες. Συχνά εκφράζονται με ασαφή τρόπο, αλλάζουν ανάλογα με τη χρήση και συνδέονται με προηγούμενες επιλογές του χρήστη. Αυτό δημιουργεί την ανάγκη για έναν μηχανισμό που να μπορεί να χειρίζεται αβεβαιότητα και βαθμούς προτίμησης.

1.3 Στόχοι της εργασίας

Κύριος στόχος της εργασίας είναι η ανάπτυξη μιας λειτουργικής εφαρμογής πλοήγησης, η οποία αξιοποιεί δεδομένα διαδρομών, προτιμήσεις χρήστη και έξυπνους μηχανισμούς υποστήριξης αποφάσεων για την παραγωγή πιο κατάλληλων προτάσεων.

Επιμέρους στόχος είναι η ανάλυση των απαιτήσεων του συστήματος, ώστε να προσδιοριστούν οι ρόλοι των χρηστών, οι λειτουργίες που πρέπει να παρέχονται, οι τεχνικοί περιορισμοί και οι απαιτήσεις εγκατάστασης και χρήσης.

Ένας ακόμη στόχος είναι η τεκμηρίωση της διαφοροποίησης της SmartNavix από υφιστάμενες εφαρμογές πλοήγησης. Η εργασία δεν επιχειρεί να αντικαταστήσει ώριμες εμπορικές λύσεις χαρτών, αλλά να διερευνήσει πώς μπορεί να προστεθεί ένα επίπεδο εξατομίκευσης και προσαρμοστικότητας πάνω σε υπάρχουσες υπηρεσίες και τεχνολογίες.

1.4 Κίνητρο και σημασία του θέματος

Το κίνητρο για την επιλογή του θέματος προκύπτει από την καθημερινή χρήση εφαρμογών πλοήγησης και από τη διαπίστωση ότι η έννοια της καλύτερης διαδρομής δεν είναι πάντα αντικειμενική. Η πιο γρήγορη ή πιο σύντομη διαδρομή μπορεί να μην είναι η πιο χρήσιμη για έναν συγκεκριμένο χρήστη.

Η σημασία του θέματος συνδέεται με την αυξανόμενη ανάγκη για έξυπνες εφαρμογές (smart applications) που προσαρμόζονται στον χρήστη και όχι μόνο στο τεχνικό πρόβλημα που καλούνται να λύσουν. Η εμπειρία χρήστη (user experience) αποτελεί πλέον κρίσιμο παράγοντα επιτυχίας για κάθε web ή mobile εφαρμογή.

Παράλληλα, η αξιοποίηση τεχνικών τεχνητής νοημοσύνης και προσαρμοστικών μοντέλων επιτρέπει την ανάπτυξη συστημάτων που μπορούν να λειτουργούν πιο ευέλικτα. Στην περίπτωση της SmartNavix, η χρήση fuzzy logic επιτρέπει την αναπαράσταση προτιμήσεων που δεν είναι απόλυτες.

1.5 Σύντομη περιγραφή της προτεινόμενης λύσης SmartNavix

Η SmartNavix προτείνεται ως εφαρμογή πλοήγησης που συνδυάζει λειτουργίες χάρτη, αναζήτησης διαδρομών, φίλτρων, προτιμήσεων, σημείων ενδιαφέροντος και αξιολόγησης αποτελεσμάτων. Ο χρήστης μπορεί να εισάγει αφετηρία και προορισμό, να επιλέγει ή να αποθηκεύει προτιμήσεις και να λαμβάνει προτάσεις διαδρομών που προσαρμόζονται περισσότερο στο επιθυμητό πλαίσιο μετακίνησης.

Η εφαρμογή οργανώνεται σε διακριτά υποσυστήματα. Το frontend παρέχει το περιβάλλον αλληλεπίδρασης με τον χρήστη, το backend διαχειρίζεται τη λογική της εφαρμογής και τις κλήσεις σε εξωτερικές υπηρεσίες, ενώ η βάση δεδομένων αποθηκεύει χρήστες, προτιμήσεις, presets, διαδρομές, στάσεις και στοιχεία διαχείρισης.

Ιδιαίτερο στοιχείο της λύσης αποτελεί ο μηχανισμός επιλογής προτιμήσεων. Μέσα από παραμέτρους όπως η συχνότητα χρήσης, η πρόσφατη επιλογή, η συνέπεια και η πληρότητα των προτιμήσεων, το σύστημα μπορεί να υπολογίζει ποια προτίμηση εκφράζει καλύτερα τη συμπεριφορά του χρήστη.

1.6 Μεθοδολογική προσέγγιση

Η μεθοδολογική προσέγγιση της εργασίας ακολουθεί διαδοχικά στάδια. Αρχικά, πραγματοποιείται βιβλιογραφική ανασκόπηση σε περιοχές που σχετίζονται με ολοκληρωμένα συστήματα, web και mobile εφαρμογές, smart και intelligent systems, τεχνητή νοημοσύνη, user experience, adaptivity, user interfaces και συστήματα πλοήγησης.

Στη συνέχεια, αναλύονται οι απαιτήσεις της εφαρμογής και σχεδιάζεται η αρχιτεκτονική του συστήματος. Η φάση αυτή περιλαμβάνει τον καθορισμό ρόλων, λειτουργικών απαιτήσεων, μη λειτουργικών απαιτήσεων, περιορισμών, ροών δεδομένων και βασικών οντοτήτων της βάσης.

Τέλος, παρουσιάζεται η υλοποίηση βασικών λειτουργιών, παραδείγματα χρήσης με screenshots, εγχειρίδιο χρήστη και αξιολόγηση της λύσης. Η εργασία ολοκληρώνεται με την αποτίμηση των οφελών, την αναφορά περιορισμών και την πρόταση μελλοντικών επεκτάσεων.

1.7 Δομή της εργασίας

Η εργασία οργανώνεται σε κεφάλαια που ακολουθούν τη λογική πορεία από τη θεωρητική τεκμηρίωση προς τον σχεδιασμό, την υλοποίηση και την αξιολόγηση. Το πρώτο κεφάλαιο εισάγει το αντικείμενο, το πρόβλημα, τους στόχους, το κίνητρο και τη συνολική μεθοδολογική προσέγγιση.

Το δεύτερο κεφάλαιο περιλαμβάνει τη βιβλιογραφική ανασκόπηση και εξετάζει έννοιες όπως ολοκληρωμένα συστήματα, web και mobile εφαρμογές, smart και intelligent systems, artificial intelligence, user experience, adaptivity, user-computer interaction και user interfaces. Στο τρίτο κεφάλαιο παρουσιάζονται παρόμοιες εφαρμογές και αναλύεται η διαφοροποίηση της SmartNavix.

Στα επόμενα κεφάλαια παρουσιάζονται η ανάλυση απαιτήσεων, ο σχεδιασμός του συστήματος, η βάση δεδομένων, η υλοποίηση των βασικών λειτουργιών και το μοντέλο fuzzy logic για την επιλογή

προτιμήσεων. Ακολουθούν παραδείγματα χρήσης, εγχειρίδιο χρήστη, οφέλη, αξιολόγηση, ασφάλεια και τελικά συμπεράσματα.

Κεφάλαιο 2: Βιβλιογραφική Ανασκόπηση

Το κεφάλαιο αυτό παρουσιάζει τη βιβλιογραφική βάση της εργασίας, εστιάζοντας στις περιοχές που στηρίζουν τον σχεδιασμό της SmartNavix: ολοκληρωμένα συστήματα, web και mobile εφαρμογές, smart και intelligent systems, τεχνητή νοημοσύνη, εμπειρία χρήστη, προσαρμοστικότητα, αλληλεπίδραση ανθρώπου-υπολογιστή, διεπαφές, συστήματα πλοήγησης, συστήματα προτάσεων και fuzzy logic.

2.1 Ολοκληρωμένα συστήματα λογισμικού

Τα ολοκληρωμένα συστήματα λογισμικού αποτελούνται από επιμέρους υποσυστήματα που συνεργάζονται για να υποστηρίξουν μια ενιαία ροή εργασίας. Στην περίπτωση εφαρμογών πλοήγησης, η έννοια αυτή περιλαμβάνει διεπαφή χρήστη, υπηρεσίες χαρτών, μηχανισμό δρομολόγησης, αποθήκευση δεδομένων, λογική προτιμήσεων και μηχανισμούς εξωτερικών APIs. Η προσέγγιση της διάχυτης υπολογιστικής έδειξε ότι οι υπηρεσίες λογισμικού μπορούν να ενσωματώνονται στο καθημερινό περιβάλλον του χρήστη και να υποστηρίζουν δράσεις χωρίς να απαιτούν συνεχή τεχνική επίγνωση από αυτόν [1].

Για να είναι ένα τέτοιο σύστημα ουσιαστικά χρήσιμο, δεν αρκεί η τεχνική διασύνδεση των μερών του. Χρειάζεται να αξιοποιεί πληροφορίες πλαισίου, όπως θέση, χρόνο, συσκευή και κατάσταση χρήστη, ώστε η απόκριση να σχετίζεται με το περιβάλλον χρήσης [2], [3]. Η αρχιτεκτονική λογισμικού λειτουργεί ως το επίπεδο που οργανώνει αυτές τις ευθύνες, καθορίζοντας πού ανήκει κάθε λειτουργία και πώς διατηρούνται ποιότητες όπως συντηρησιμότητα, επεκτασιμότητα και αξιοπιστία [4]. Παράλληλα, η τεκμηρίωση απαιτήσεων είναι απαραίτητη ώστε οι λειτουργίες να προκύπτουν από σαφείς ανάγκες και όχι από αποσπασματικές τεχνικές επιλογές [5].

2.2 Web-based applications

Οι web-based εφαρμογές βασίζονται σε αρχιτεκτονική όπου ο χρήστης αλληλεπιδρά μέσω browser ή web frontend, ενώ η επιχειρησιακή λογική και τα δεδομένα οργανώνονται σε απομακρυσμένες υπηρεσίες. Η αξία τους βρίσκεται στην άμεση πρόσβαση, στην ευκολότερη διανομή ενημερώσεων και στη δυνατότητα ενοποίησης με τρίτες υπηρεσίες. Ο σχεδιασμός τους πρέπει να αντιμετωπίζει την εμπειρία χρήστη ως συνολικό αποτέλεσμα δομής, λειτουργικότητας, περιεχομένου και οπτικής οργάνωσης [6].

Η ευχρηστία σε web περιβάλλον δεν περιορίζεται στην αισθητική. Συνδέεται με μάθηση, αποδοτικότητα, αποφυγή σφαλμάτων και ικανοποίηση, στοιχεία που έχουν αναγνωριστεί ως βασικές διαστάσεις αξιολόγησης διαδραστικών συστημάτων [7]. Σε επίπεδο αρχιτεκτονικής, το web ευνοεί διαχωρισμό πόρων, αιτημάτων και αναπαραστάσεων, λογική που επηρεάζει τον τρόπο σχεδιασμού APIs και υπηρεσιών [8]. Για εφαρμογές όπως η SmartNavix, σημαντική είναι και η προσβασιμότητα, επειδή η πλοήγηση και η ανάγνωση οδηγιών πρέπει να παραμένουν λειτουργικές για διαφορετικούς χρήστες και συσκευές [9].

2.3 Mobile applications

Οι mobile εφαρμογές εισάγουν διαφορετικές απαιτήσεις σε σχέση με τις κλασικές web εφαρμογές, επειδή λειτουργούν σε μικρότερες οθόνες, συχνά υπό μεταβαλλόμενες συνθήκες δικτύου και με έντονη εξάρτηση από αισθητήρες, τοποθεσία και ειδοποιήσεις. Η επιλογή ανάμεσα σε native, hybrid και web υλοποίηση επηρεάζει την απόδοση, τη διανομή και την πρόσβαση σε δυνατότητες της συσκευής [10].

Η βιβλιογραφία για την ανάπτυξη mobile εφαρμογών δείχνει ότι θέματα όπως κατακερματισμός πλατφορμών, περιορισμένοι πόροι, συνδεσιμότητα και ασφάλεια πρέπει να λαμβάνονται υπόψη από την αρχή του σχεδιασμού [11]. Η αξιολόγηση ευχρηστίας σε mobile περιβάλλον απαιτεί επίσης προσοχή σε παράγοντες όπως μικρή οθόνη, αφή, κίνηση και περιστασιακή χρήση [12]. Σε εφαρμογές που προτείνουν διαδρομές ή σημεία ενδιαφέροντος, η κινητικότητα του χρήστη συνδέεται άμεσα με εξατομικευμένες συστάσεις και υπηρεσίες βασισμένες στη θέση [13].

2.4 Smart applications και smart systems

Οι smart εφαρμογές και τα smart systems χαρακτηρίζονται από τη δυνατότητα αξιοποίησης δεδομένων περιβάλλοντος, αισθητήρων, κανόνων ή αλγορίθμων για την παραγωγή πιο κατάλληλων αποκρίσεων. Δεν πρόκειται απλώς για εφαρμογές με περισσότερες λειτουργίες, αλλά για συστήματα που προσπαθούν να αντιληφθούν το πλαίσιο χρήσης και να αντιδράσουν ανάλογα. Η σχετική βιβλιογραφία της ubiquitous computing και των smart environments δείχνει ότι η έννοια της «έξυπνης» συμπεριφοράς συνδέεται με το περιβάλλον, τη διαλειτουργικότητα και την αλληλεπίδραση ανθρώπου-συστήματος [14], [15].

Η θέση του χρήστη αποτελεί κρίσιμο δεδομένο για τέτοια συστήματα. Από πρώιμες τεχνολογίες εντοπισμού, όπως το Active Badge, φάνηκε ότι η γνώση της τοποθεσίας μπορεί να χρησιμοποιηθεί για υπηρεσίες που προσαρμόζονται στο πού βρίσκεται ο χρήστης και τι είναι πιθανό να χρειάζεται [16]. Για τη SmartNavix, η έννοια του smart system μεταφράζεται σε συνδυασμό χαρτών, προτιμήσεων, ιστορικού επιλογών και μηχανισμών αξιολόγησης εναλλακτικών διαδρομών.

Συναφής είναι και η πρόσφατη έρευνα σε smart tourism εφαρμογές, όπου η τεχνητή νοημοσύνη και η fuzzy logic αξιοποιούνται για ταξινόμηση ή εξατομίκευση σημείων ενδιαφέροντος. Η χρήση δεδομένων από Wikidata για ερμηνεύσιμη multi-category ταξινόμηση POIs δείχνει πώς μπορούν να συνδεθούν ανοικτά δεδομένα, AI και fuzzy reasoning σε εφαρμογές τουρισμού [17]. Παράλληλα, η fuzzy logic-based personalization σε smart tourism περιβάλλοντα συνδέει άμεσα την εμπειρία χρήστη με την προσαρμογή προτάσεων σε προτιμήσεις και χαρακτηριστικά χρήσης [18].

2.5 Intelligent applications και intelligent systems

Τα intelligent systems προσπαθούν να επιλύσουν προβλήματα που απαιτούν συλλογισμό, αναζήτηση, μάθηση ή λήψη αποφάσεων. Στη γενική βιβλιογραφία της τεχνητής νοημοσύνης, η νοημοσύνη ενός συστήματος συνδέεται με την ικανότητα αντίληψης του περιβάλλοντος, αναπαράστασης γνώσης και επιλογής ενεργειών που εξυπηρετούν έναν στόχο [19], [20].

Σε πρακτικές εφαρμογές, η νοημοσύνη δεν χρειάζεται πάντα να σημαίνει πλήρη αυτονομία. Μπορεί να εκφράζεται μέσα από πράκτορες που συλλέγουν πληροφορίες, αξιολογούν εναλλακτικές και προτείνουν ενέργειες στον χρήστη [21]. Η SmartNavix υιοθετεί αυτή τη μετριοπαθή προσέγγιση: δεν αντικαθιστά την κρίση του χρήστη, αλλά οργανώνει τα δεδομένα διαδρομών και προτιμήσεων ώστε να τον βοηθά να επιλέξει πιο κατάλληλη εναλλακτική.

2.6 Artificial Intelligence σε σύγχρονες εφαρμογές

Η τεχνητή νοημοσύνη στις σύγχρονες εφαρμογές αξιοποιείται σε προβλήματα ταξινόμησης, πρόβλεψης, παραγωγής περιεχομένου, κατανόησης φυσικής γλώσσας και υποστήριξης αποφάσεων. Η βαθιά μάθηση έδωσε ισχυρά εργαλεία για αναπαράσταση πολύπλοκων δεδομένων [22], ενώ η μηχανική μάθηση γενικότερα παραμένει πλαίσιο με σημαντική επίδραση στην επιστήμη και στη μηχανική λογισμικού [23].

Ιδιαίτερα σημαντική εξέλιξη για τις σύγχρονες εφαρμογές είναι τα μοντέλα μετασχηματιστών και τα μεγάλα γλωσσικά μοντέλα. Η αρχιτεκτονική attention επέτρεψε αποτελεσματική επεξεργασία ακολουθιών [24], ενώ τα μεγάλα γλωσσικά μοντέλα έδειξαν δυνατότητες few-shot προσαρμογής σε διαφορετικά γλωσσικά καθήκοντα [25]. Παράλληλα, η βιβλιογραφία τονίζει ότι τα foundation models προσφέρουν ευκαιρίες αλλά και κινδύνους, όπως αδιαφάνεια, μεροληψίες και ανάγκη ελέγχου αξιοπιστίας [26].

Στη σύγχρονη έρευνα του Τμήματος, η τεχνητή νοημοσύνη συνδέεται επίσης με decision support systems και με απαιτήσεις εκπαιδευτικού λογισμικού. Η μελέτη των AI-empowered decision support systems αναδεικνύει ότι η αξία της AI βρίσκεται κυρίως στην υποστήριξη και τεκμηρίωση αποφάσεων, όχι στην πλήρη αντικατάσταση του ανθρώπου [27]. Αντίστοιχα, η requirements engineering οπτική για generative AI σε εκπαιδευτικό λογισμικό δείχνει ότι οι τεχνικές AI πρέπει να εντάσσονται σε σαφώς ορισμένες ανάγκες, στόχους και περιορισμούς χρήσης [28].

2.7 User Experience και ευχρηστία εφαρμογών

Η εμπειρία χρήστη (user experience - UX) αφορά το πώς ο χρήστης αντιλαμβάνεται, αισθάνεται και αξιολογεί την αλληλεπίδραση με ένα προϊόν ή σύστημα. Δεν ταυτίζεται με την απλή λειτουργικότητα, επειδή περιλαμβάνει αντιλήψεις χρησιμότητας, ευχαρίστησης, εμπιστοσύνης και καταλληλότητας στο συγκεκριμένο πλαίσιο χρήσης [29].

Η κλασική προσέγγιση του ανθρωποκεντρικού σχεδιασμού δείχνει ότι ένα σύστημα πρέπει να γίνεται κατανοητό μέσα από τις ενέργειες που επιτρέπει και την ανατροφοδότηση που παρέχει [30]. Η ευχρηστία ορίζεται μέσα από αποτελεσματικότητα, αποδοτικότητα και ικανοποίηση σε συγκεκριμένο πλαίσιο χρήσης [31], ενώ ο ανθρωποκεντρικός σχεδιασμός οργανώνει επαναληπτικά την κατανόηση χρηστών, τον σχεδιασμό λύσεων και την αξιολόγηση [32]. Για τη SmartNavix, το UX είναι κρίσιμο επειδή η απόφαση διαδρομής συχνά λαμβάνεται γρήγορα και σε πραγματικές συνθήκες μετακίνησης.

Η σχέση τεχνητής νοημοσύνης και εμπειρίας χρήστη έχει εξεταστεί και ως αμοιβαία σχέση: η AI μπορεί να βελτιώνει την εξατομίκευση και τη λήψη απόφασης, αλλά ταυτόχρονα η εμπειρία χρήστη επηρεάζει την αποδοχή, την εμπιστοσύνη και τον τρόπο χρήσης των AI λειτουργιών [33]. Για τη SmartNavix, αυτή η σύνδεση είναι σημαντική, επειδή ο χρήστης πρέπει να κατανοεί γιατί εμφανίζεται μια πρόταση και να αισθάνεται ότι μπορεί να την αποδεχθεί, να την τροποποιήσει ή να την απορρίψει.

2.8 Adaptivity και adaptive applications/systems

Η προσαρμοστικότητα (adaptivity) αναφέρεται στην ικανότητα ενός συστήματος να διαφοροποιεί τη συμπεριφορά ή την παρουσίασή του σύμφωνα με χαρακτηριστικά του χρήστη, του στόχου ή του περιβάλλοντος. Στα adaptive hypermedia, η προσαρμογή οργανώνεται γύρω από μοντέλα χρήστη και τεχνικές που μεταβάλλουν περιεχόμενο, συνδέσμους ή καθοδήγηση [34], [35].

Τα συστήματα μοντελοποίησης χρήστη προσφέρουν μηχανισμούς καταγραφής και αξιοποίησης χαρακτηριστικών όπως γνώσεις, ενδιαφέροντα, προτιμήσεις και ιστορικό αλληλεπίδρασης [36]. Σε επίπεδο διεπαφών, η προσαρμογή μπορεί να αφορά όχι μόνο το περιεχόμενο αλλά και τον τρόπο παρουσίασης ή προτεραιοποίησης επιλογών [37]. Η SmartNavix αξιοποιεί αυτή τη λογική όταν επιχειρεί να μετατρέψει τις προηγούμενες επιλογές και τα φίλτρα του χρήστη σε πιο σχετικές προτάσεις διαδρομών.

Η προσαρμοστικότητα έχει μελετηθεί ιδιαίτερα και σε intelligent educational games, όπου adaptive scenarios και serious games μπορούν να μεταβάλλουν σενάριο, περιβάλλον ή αξιολόγηση δεξιοτήτων με βάση γνωστικά και συμπεριφορικά στοιχεία χρήστη [38], [39]. Η χρήση fuzzy cognitive walkthrough για αποτίμηση soft skills δείχνει επίσης ότι η fuzzy λογική μπορεί να υποστηρίξει αξιολόγηση σύνθετων, μη δυαδικών χαρακτηριστικών σε διαδραστικά περιβάλλοντα [40].

2.9 User-computer interaction

Η αλληλεπίδραση ανθρώπου-υπολογιστή (human-computer interaction - HCI) μελετά τον σχεδιασμό, την υλοποίηση και την αξιολόγηση διαδραστικών συστημάτων. Η περιοχή συνδέει τεχνικές επιλογές με ανθρώπινες δυνατότητες, περιορισμούς, στόχους και συμπεριφορές [41].

Στον σχεδιασμό διαδραστικών συστημάτων, η ορατότητα της κατάστασης, η σαφής ανατροφοδότηση, η συνέπεια και η δυνατότητα ανάκαμψης από σφάλματα αποτελούν βασικές αρχές [42], [43]. Για μια εφαρμογή πλοήγησης, οι αρχές αυτές αποκτούν πρακτική σημασία: ο χρήστης πρέπει να αντιλαμβάνεται ποια διαδρομή επιλέχθηκε, γιατί προτείνεται και πώς μπορεί να αλλάξει φίλτρα ή προτιμήσεις χωρίς σύγχυση.

2.10 User interfaces και σχεδιασμός διεπαφών

Οι διεπαφές χρήστη (user interfaces - UI) είναι το σημείο όπου η λειτουργικότητα του συστήματος γίνεται αντιληπτή και ελέγξιμη από τον χρήστη. Η σχεδίαση διεπαφών δεν περιορίζεται στη μορφή των οθονών, αλλά περιλαμβάνει ροές εργασίας, κατανομή πληροφορίας, χειριστήρια, μηνύματα, καταστάσεις φόρτωσης και αντιμετώπιση σφαλμάτων [44].

Τα σχεδιαστικά patterns βοηθούν στην επαναχρησιμοποίηση δοκιμασμένων λύσεων για συνηθισμένα προβλήματα διεπαφής [45]. Η ιστορία της HCI δείχνει επίσης ότι οι τεχνολογίες διεπαφής εξελίσσονται μαζί με τις ανάγκες αλληλεπίδρασης, από γραφικά περιβάλλοντα μέχρι σύγχρονες πολυτροπικές εφαρμογές [46]. Για τη SmartNavix, η διεπαφή πρέπει να υποστηρίζει γρήγορη εισαγωγή αφετηρίας/προορισμού, σαφή εμφάνιση εναλλακτικών και κατανοητή παρουσίαση του λόγου επιλογής μιας προτεινόμενης διαδρομής.

2.11 Συστήματα πλοήγησης και αλγόριθμοι διαδρομών

Τα συστήματα πλοήγησης βασίζονται σε αλγορίθμους γράφων, όπου κόμβοι και ακμές αναπαριστούν το οδικό ή χωρικό δίκτυο. Ο αλγόριθμος του Dijkstra αποτελεί θεμελιώδη μέθοδο εύρεσης συντομότερων διαδρομών σε γράφους με μη αρνητικά βάρη [47]. Η αναζήτηση A* εισάγει ευρετική καθοδήγηση ώστε η αναζήτηση να γίνεται αποδοτικότερη όταν υπάρχει κατάλληλη εκτίμηση απόστασης προς τον στόχο [48].

Στα πραγματικά δίκτυα μεταφορών, η δρομολόγηση δεν είναι απλώς εφαρμογή ενός κλασικού αλγορίθμου. Απαιτεί προεπεξεργασία, αποδοτικές δομές δεδομένων, ενημέρωση βαρών και χειρισμό πολλών κριτηρίων [49], [50]. Για τη SmartNavix, η βασική δρομολόγηση παρέχεται από εξωτερικές υπηρεσίες χαρτών, όμως η ερευνητική σημασία βρίσκεται στο επιπλέον επίπεδο αξιολόγησης προτιμήσεων πάνω στις εναλλακτικές διαδρομές.

2.12 Εξατομίκευση και συστήματα προτάσεων

Τα συστήματα προτάσεων (recommender systems) στοχεύουν στη μείωση του πληροφοριακού φόρτου, προτείνοντας επιλογές που είναι πιθανό να ταιριάζουν στον χρήστη [51]. Η σχετική βιβλιογραφία αναγνωρίζει τεχνικές όπως collaborative filtering, content-based filtering και υβριδικές προσεγγίσεις, καθώς και ζητήματα όπως cold start, sparsity, εξηγήσεις και αξιολόγηση ποιότητας προτάσεων [52], [53].

Οι υβριδικές προσεγγίσεις είναι ιδιαίτερα χρήσιμες όταν μία μόνο πηγή πληροφορίας δεν επαρκεί [54]. Η αξιολόγηση συστημάτων προτάσεων δεν πρέπει να περιορίζεται σε αριθμητική ακρίβεια, επειδή παράγοντες όπως χρησιμότητα, κάλυψη, novelty και αποδοχή από τον χρήστη επηρεάζουν την πραγματική επιτυχία [55]. Η SmartNavix εντάσσεται σε αυτή τη λογική, καθώς η πρόταση διαδρομής δεν είναι μόνο αποτέλεσμα χάρτη αλλά και αποτέλεσμα προτιμήσεων, φίλτρων και προηγούμενης συμπεριφοράς.

2.13 Fuzzy logic σε συστήματα λήψης αποφάσεων

Η fuzzy logic βασίζεται στην ιδέα ότι πολλές έννοιες του πραγματικού κόσμου δεν είναι απολύτως δυαδικές. Η θεωρία των fuzzy sets εισήγαγε βαθμούς συμμετοχής, επιτρέποντας σε ένα στοιχείο να ανήκει σε ένα σύνολο με τιμή ανάμεσα στο 0 και στο 1 [56]. Η έννοια των γλωσσικών μεταβλητών επέτρεψε την έκφραση κανόνων με όρους όπως «υψηλή προτίμηση», «πρόσφατη χρήση» ή «μέτρια συνέπεια» [57].

Στη λήψη αποφάσεων, τα fuzzy μοντέλα μπορούν να συνδυάζουν ασαφή κριτήρια και να παράγουν υποστηρικτικές εκτιμήσεις όταν τα δεδομένα δεν οδηγούν σε απόλυτη απάντηση [58]. Αυτό ταιριάζει στο πρόβλημα της SmartNavix, επειδή η «καλύτερη» διαδρομή μπορεί να εξαρτάται από βαθμούς προτίμησης και όχι από ένα μοναδικό καθολικό κριτήριο.

Η fuzzy logic έχει ιδιαίτερη σημασία σε intelligent software systems, επειδή επιτρέπει σε ένα σύστημα να χειρίζεται ασαφείς καταστάσεις και ανθρώπινες προτιμήσεις με ερμηνεύσιμο τρόπο [59]. Σχετικές εργασίες αναδεικνύουν τον ρόλο της fuzzy logic σε AI και smart applications [60], καθώς και σε συστήματα όπου πρέπει να παραχθεί ειδοποίηση ή πρόληψη κινδύνου με βάση αβέβαιες ανθρώπινες συνθήκες, όπως emergency alerts και smart home fire prevention [61], [62].

Στο επίπεδο εξατομίκευσης, οι personalized fuzzy logic strategies έχουν εφαρμοστεί και σε intelligent tutoring of programming, δείχνοντας ότι η fuzzy λογική μπορεί να υποστηρίξει προσαρμοστικές εκπαιδευτικές στρατηγικές [63]. Για τη SmartNavix, αυτό ενισχύει την επιλογή ενός fuzzy μηχανισμού για best preference, επειδή το ζητούμενο δεν είναι μόνο να βρεθεί μια αριθμητικά

βέλτιστη τιμή, αλλά να αποτυπωθούν ανθρώπινες προτιμήσεις με τρόπο κατανοητό και παραμετροποιήσιμο.

2.14 Χρήση LLMs και AI APIs σε εφαρμογές τοποθεσίας

Οι σύγχρονες εφαρμογές τοποθεσίας συχνά στηρίζονται σε εξωτερικές πλατφόρμες χαρτών για γεωκωδικοποίηση, αναζήτηση σημείων, υπολογισμό διαδρομών και παρουσίαση χαρτογραφικών δεδομένων [64]. Η χρήση έτοιμων APIs επιτρέπει σε μια εφαρμογή να επικεντρωθεί στη δική της επιχειρησιακή λογική, ενώ αξιοποιεί ώριμες υπηρεσίες για το πολύπλοκο πρόβλημα της δρομολόγησης [65].

Παράλληλα, τα AI APIs και τα μεγάλα γλωσσικά μοντέλα μπορούν να χρησιμοποιηθούν για ερμηνεία φυσικής γλώσσας, παραγωγή επεξηγήσεων ή μετατροπή μη δομημένων αιτημάτων σε πιο δομημένη μορφή [66]. Η χρήση τους όμως πρέπει να γίνεται ελεγχόμενα, επειδή το αποτέλεσμα πρέπει να συνδέεται με επαληθεύσιμα δεδομένα χαρτών και όχι να αντικαθιστά την αξιόπιστη δρομολόγηση.

Η ενσωμάτωση GPT, knowledge graphs και machine learning σε πολιτισμικά e-learning περιβάλλοντα δείχνει ότι τα LLMs μπορούν να συνδεθούν με πιο δομημένη γνώση και κανόνες, αντί να χρησιμοποιούνται μόνο ως γεννήτριες κειμένου [67], [68]. Παράλληλα, όταν AI λειτουργίες χρησιμοποιούνται σε εφαρμογές που αλληλεπιδρούν με χρήστες και δεδομένα, πρέπει να λαμβάνεται υπόψη και το ζήτημα συγκατάθεσης και προστασίας δεδομένων, ιδίως υπό το πλαίσιο του GDPR [69].

2.15 Σύνοψη βιβλιογραφίας και ερευνητικό κενό

Η βιβλιογραφία δείχνει ότι η SmartNavix βρίσκεται στη διασταύρωση πολλών περιοχών: ολοκληρωμένα συστήματα λογισμικού, web και mobile εφαρμογές, context-aware συστήματα, UX/HCI, αλγόριθμοι πλοήγησης, συστήματα προτάσεων και fuzzy decision-making. Κάθε περιοχή παρέχει ένα τμήμα της λύσης, αλλά καμία από μόνη της δεν καλύπτει πλήρως την ανάγκη για εξατομικευμένη επιλογή διαδρομής με συνδυασμό προτιμήσεων και πραγματικών εναλλακτικών [3], [29], [49], [53], [56].

Το ερευνητικό και εφαρμοστικό κενό που προκύπτει αφορά την οργάνωση ενός πρακτικού επιπέδου προσαρμογής πάνω από υπάρχουσες υπηρεσίες χαρτών. Η SmartNavix δεν επιχειρεί να αναπτύξει νέο αλγόριθμο παγκόσμιας δρομολόγησης, αλλά να εξετάσει πώς μπορούν να συνδυαστούν route APIs, προτιμήσεις χρήστη, μοντέλο fuzzy logic και καθαρή διεπαφή ώστε οι προτεινόμενες διαδρομές να είναι πιο κοντά στο προσωπικό πλαίσιο χρήσης [4], [36], [54], [58], [65].

Κεφάλαιο 3: Παρόμοιες Εφαρμογές και Διαφοροποίηση της SmartNavix

Το κεφάλαιο εξετάζει αντιπροσωπευτικές εφαρμογές πλοήγησης και εφαρμογές που σχετίζονται με προτάσεις σημείων ενδιαφέροντος. Η ανάλυση δεν περιορίζεται στην απλή περιγραφή λειτουργιών, αλλά εστιάζει στο ποια ανάγκη καλύπτει κάθε λύση και σε ποιο σημείο τοποθετείται η SmartNavix ως διαφορετική προσέγγιση.

3.1 Υφιστάμενες εφαρμογές πλοήγησης

Η αγορά των εφαρμογών πλοήγησης έχει ωριμάσει σημαντικά, καθώς ο χρήστης μπορεί πλέον να αναζητήσει διαδρομές, να λάβει οδηγίες σε πραγματικό χρόνο, να εντοπίσει σημεία ενδιαφέροντος και να προσαρμόσει ορισμένες επιλογές μετακίνησης. Οι πιο αναγνωρίσιμες λύσεις, όπως Google Maps, Waze και Apple Maps, λειτουργούν ως ολοκληρωμένα οικοσυστήματα χαρτών και μετακίνησης, αξιοποιώντας μεγάλες βάσεις γεωχωρικών δεδομένων, δεδομένα κίνησης και υπηρεσίες αναζήτησης τοποθεσιών [70], [71], [72].

Η ύπαρξη τέτοιων εφαρμογών δεν ακυρώνει την ανάγκη για εξειδικευμένες ή ερευνητικές λύσεις. Αντίθετα, δείχνει ποια λειτουργικότητα θεωρείται πλέον βασική: αναζήτηση προορισμού, εναλλακτικές διαδρομές, πληροφορίες χρόνου, αποφυγή συγκεκριμένων περιορισμών και οπτική αναπαράσταση στον χάρτη. Το ενδιαφέρον της SmartNavix βρίσκεται στο πρόσθετο επίπεδο προσαρμογής πάνω σε αυτές τις δυνατότητες, δηλαδή στον τρόπο με τον οποίο οι προτιμήσεις, τα φίλτρα, τα presets και το ιστορικό χρήσης μπορούν να επηρεάσουν την τελική πρόταση διαδρομής.

3.2 Google Maps

Το Google Maps αποτελεί μία από τις πληρέστερες λύσεις χαρτών και πλοήγησης. Παρέχει αναζήτηση τοποθεσιών, οδηγίες, πληροφορίες για επιχειρήσεις και σημεία ενδιαφέροντος, επιλογές διαδρομής και λειτουργίες που βοηθούν τον χρήστη να εξερευνήσει μια περιοχή. Στην επίσημη περιγραφή του, δίνεται έμφαση τόσο στην πλοήγηση όσο και στην ανακάλυψη τοποθεσιών, με δυνατότητες προτάσεων, λιστών και ερωτήσεων μέσω Gemini σε ορισμένες αγορές [70].

Σε επίπεδο τεχνολογικής αξιοποίησης, η SmartNavix δεν ανταγωνίζεται το Google Maps ως παγκόσμια πλατφόρμα χαρτών. Αντίθετα, αξιοποιεί υπηρεσίες χαρτών και δρομολόγησης ως υποδομή, ώστε να επικεντρωθεί στο επίπεδο της εξατομικευμένης επιλογής. Η διαφορά είναι ότι η SmartNavix οργανώνει ρητά στοιχεία όπως ενδιάμεσες στάσεις, αποφυγή διοδίων ή αυτοκινητοδρόμων, μέγεθος οχήματος, προτιμήσεις, διάθεση χρήστη, EV charging επιλογές και προηγούμενη συμπεριφορά, με σκοπό να παράγει μια πρόταση πιο κοντά στο προφίλ του συγκεκριμένου χρήστη [64], [65], [66].

3.3 Waze

Το Waze διαφοροποιείται κυρίως μέσω της κοινοτικής και crowdsourced λογικής του. Η πλατφόρμα βασίζεται στη συμμετοχή οδηγών, map editors και συνεργατών, ώστε να ενημερώνει τον χάρτη και να προσαρμόζει τη δρομολόγηση σύμφωνα με την κατάσταση του δρόμου [71]. Σύμφωνα με την τεκμηρίωση υποστήριξης, όταν ο χρήστης οδηγεί με ανοιχτό το Waze, η εφαρμογή αξιοποιεί δεδομένα πραγματικού χρόνου για μέση ταχύτητα, σφάλματα χάρτη, διάταξη δρόμων και αναφορές όπως κίνηση, ατυχήματα, μπλόκα, κλειστούς δρόμους και καιρικές συνθήκες [73].

Η προσέγγιση αυτή είναι ιδιαίτερα ισχυρή για οδηγούς που ενδιαφέρονται κυρίως για άμεση προσαρμογή στην κυκλοφορία. Η SmartNavix έχει διαφορετικό κέντρο βάρους. Δεν βασίζεται σε μαζική κοινότητα οδηγών για να ενημερώνει τον χάρτη, αλλά σε ατομικό προφίλ χρήστη και σε αποθηκευμένες επιλογές. Έτσι, το ερώτημα δεν είναι μόνο ποια διαδρομή αποφεύγει την κίνηση, αλλά ποια διαδρομή ταιριάζει καλύτερα στη συγκεκριμένη προτίμηση, στο επιλεγμένο όχημα, στις στάσεις και στο ιστορικό του χρήστη.

3.4 Apple Maps και άλλες εφαρμογές χαρτών

Το Apple Maps δίνει έμφαση στην καθαρή εμπειρία πλοήγησης, στην ενσωμάτωση με το οικοσύστημα Apple και σε λειτουργίες όπως driving, transit, walking, cycling, place cards, guides και CarPlay [72]. Για παράδειγμα, υποστηρίζει πληροφορίες οδήγησης όπως real-time traffic, όρια ταχύτητας, lane guidance και 3D road-level οπτική σε σύνθετες διασταυρώσεις, ενώ σε επιλεγμένα σενάρια προσφέρει στοιχεία για ηλεκτρικά οχήματα και στάσεις φόρτισης [72].

Η διαφοροποίηση της SmartNavix από τέτοιες εφαρμογές δεν αφορά την κλίμακα των χαρτογραφικών δεδομένων ή την ενσωμάτωση με συσκευές. Αυτά αποτελούν πλεονεκτήματα μεγάλων εμπορικών οικοσυστημάτων. Η SmartNavix, ως εφαρμογή στο πλαίσιο διπλωματικής εργασίας, εστιάζει περισσότερο στη διαφανή μοντελοποίηση προτιμήσεων και στην υλοποίηση ενός συστήματος που μπορεί να εξηγηθεί, να επεκταθεί και να αξιολογηθεί σε επίπεδο λογισμικού.

3.5 Εφαρμογές με προτάσεις σημείων ενδιαφέροντος

Πέρα από τις καθαρές εφαρμογές πλοήγησης, υπάρχουν εφαρμογές και APIs που επικεντρώνονται στην αναζήτηση και κατηγοριοποίηση σημείων ενδιαφέροντος (points of interest - POI). Το Foursquare Place Search, για παράδειγμα, προσφέρει αναζήτηση τοποθεσιών με παραμέτρους όπως query, γεωγραφικές συντεταγμένες, ακτίνα, κατηγορίες, τιμή, κατάσταση λειτουργίας και ταξινόμηση βάσει relevance, rating, distance ή popularity [74]. Αντίστοιχα, τα Google Places APIs παρέχουν υποδομή για αναζήτηση και λεπτομέρειες τοποθεσιών μέσα σε εφαρμογές [3].

Η σχετική βιβλιογραφία για mobile recommender systems στον τουρισμό δείχνει ότι οι προτάσεις σημείων ενδιαφέροντος αποκτούν αξία όταν συνδέονται με το πλαίσιο μετακίνησης, τις προτιμήσεις και την κατάσταση του χρήστη [13]. Η SmartNavix αξιοποιεί αυτή τη λογική σε περιορισμένο και πρακτικό επίπεδο: μπορεί να παράγει POI intents από φίλτρα ή διάθεση χρήστη, να αναζητά σχετικές τοποθεσίες κοντά στη διαδρομή και να τις εμφανίζει ως βοηθητικό υλικό, χωρίς να μετατρέπεται σε πλήρη τουριστικό οδηγό.

3.6 Περιορισμοί των υπαρχουσών λύσεων

Οι υπάρχουσες λύσεις προσφέρουν ώριμες δυνατότητες, αλλά συχνά αντιμετωπίζουν την προσαρμογή σε επίπεδο άμεσης επιλογής και όχι ως διακριτό μοντέλο προτίμησης. Ο χρήστης μπορεί να επιλέξει αποφυγή διοδίων, να αναζητήσει σημεία ενδιαφέροντος ή να αποθηκεύσει τοποθεσίες, όμως η εφαρμογή δεν είναι πάντα διαφανής ως προς το πώς συνδυάζει αυτά τα στοιχεία με ιστορικό χρήσης και προηγούμενες αποφάσεις.

Ένας δεύτερος περιορισμός είναι ότι η έννοια της «καλύτερης» διαδρομής παραμένει συχνά συνδεδεμένη με χρόνο, απόσταση ή κυκλοφορία. Η βιβλιογραφία των συστημάτων προτάσεων δείχνει ότι η εξατομίκευση απαιτεί πιο συστηματική αξιοποίηση προτιμήσεων, ιστορικού και αξιολόγησης από τον χρήστη [51], [52], [53]. Για αυτόν τον λόγο, η SmartNavix αντιμετωπίζει τη δρομολόγηση ως απόφαση πολλών κριτηρίων και όχι μόνο ως υπολογισμό συντομότερης ή ταχύτερης πορείας.

3.7 Διαφοροποίηση της SmartNavix

Η SmartNavix διαφοροποιείται επειδή τοποθετεί την εξατομίκευση στο κέντρο της εφαρμογής. Ο χρήστης δεν εισάγει μόνο έναν προορισμό, αλλά μπορεί να ορίσει στάσεις, προτίμηση διαδρομής, διάθεση, μέγεθος οχήματος, αποφυγές, χρονικό πλαίσιο κίνησης και ανάγκη για EV charging. Τα στοιχεία αυτά αποθηκεύονται ως filtered preferences και μπορούν να επαναχρησιμοποιηθούν, ενώ τα presets επιτρέπουν γρήγορη εισαγωγή συχνών τοποθεσιών.

Η δεύτερη διαφοροποίηση αφορά τον μηχανισμό επιλογής της προτίμησης. Η εφαρμογή δεν κρατά απλώς την τελευταία επιλογή του χρήστη, αλλά διαθέτει resolver που εξετάζει συχνότητα, πρόσφατη χρήση, πληρότητα και συνέπεια. Το fuzzy logic μοντέλο επιτρέπει να συνδυαστούν αυτά τα κριτήρια χωρίς να απαιτείται απόλυτος κανόνας τύπου «αν ισχύει A τότε πάντα επέλεξε B» [56], [57], [58]. Με αυτόν τον τρόπο, η SmartNavix προσπαθεί να λειτουργήσει ως μελέτη προσαρμοστικής πλοήγησης πάνω σε πραγματικές υπηρεσίες χαρτών.

3.8 Συγκριτικός πίνακας λειτουργιών

Ο Πίνακας 3.1 συνοψίζει τις βασικές διαφορές ανάμεσα σε αντιπροσωπευτικές υπάρχουσες λύσεις και στη SmartNavix. Η σύγκριση δεν έχει στόχο να δείξει ότι η SmartNavix υπερτερεί σε κλίμακα ή ωριμότητα· οι εμπορικές εφαρμογές διαθέτουν πολύ μεγαλύτερη βάση δεδομένων και υποδομή. Στόχος είναι να αποτυπωθεί το ειδικό πεδίο στο οποίο εστιάζει η εργασία: η οργάνωση προτιμήσεων και η προσαρμοσμένη πρόταση διαδρομής.

ΠΙΝΑΚΑΣ 3.1: Συγκριτική αποτύπωση παρόμοιων εφαρμογών και SmartNavix

Κριτήριο	Google Maps	Waze	Apple Maps	SmartNavix
Κύριος προσανατολισμός	Πλοήγηση, εξερεύνηση και αναζήτηση τοποθεσιών.	Κοινοτική πλοήγηση οδηγών με έμφαση στην κίνηση.	Πλοήγηση με έμφαση στην απλότητα και στο οικοσύστημα Apple.	Προσαρμοσμένη πρόταση διαδρομής με βάση φίλτρα, προτιμήσεις και ιστορικό.
Δεδομένα πραγματικού χρόνου	Υποστηρίζει επιλογές διαδρομής και real-time πληροφορία.	Βασίζεται έντονα σε αναφορές οδηγών και ενεργή κοινότητα.	Παρέχει real-time traffic και οδηγίες ανά μέσο μετακίνησης.	Αξιοποιεί εξωτερικές υπηρεσίες χαρτών και προσθέτει δικό της επίπεδο προτίμησης.
Εξατομίκευση	Προτάσεις, λίστες και αποθηκευμένα σημεία.	Προσαρμογή κυρίως μέσω driving settings και κοινοτικών δεδομένων.	Προσωπική εμπειρία μέσα στο Apple οικοσύστημα.	Filtered preferences, presets, mood, όχημα, στάσεις, αποφυγές και fuzzy επιλογή προτίμησης.
Σημεία ενδιαφέροντος	Ισχυρή αναζήτηση και πληροφορίες τοποθεσιών.	Λιγότερο POI-centric, περισσότερο driver-centric.	Place cards, search και curated guides.	POI intents κοντά στη διαδρομή, όταν προκύπτουν από φίλτρα ή προτίμηση.
Κύριος περιορισμός ως προς το θέμα της εργασίας	Η εσωτερική λογική εξατομίκευσης δεν είναι πλήρως διαφανής στον χρήστη/μελετητή.	Εξαρτάται από ενεργή κοινότητα και εστιάζει κυρίως σε οδηγούς.	Είναι δεμένη με συγκεκριμένο οικοσύστημα συσκευών.	Δεν διαθέτει παγκόσμια χαρτογραφική υποδομή· λειτουργεί ως ερευνητικό/εκπαιδευτικό σύστημα πάνω σε APIs.

Κεφάλαιο 4: Ανάλυση Απαιτήσεων

Το κεφάλαιο αυτό παρουσιάζει τις απαιτήσεις της SmartNavix, δηλαδή τις λειτουργίες που πρέπει να υποστηρίζει το σύστημα, τους ρόλους που αλληλεπιδρούν με αυτό, τα βασικά σενάρια χρήσης και τους τεχνικούς ή οργανωτικούς περιορισμούς που επηρεάζουν την ανάπτυξη. Η ανάλυση λειτουργεί ως γέφυρα ανάμεσα στη βιβλιογραφική και συγκριτική τεκμηρίωση των προηγούμενων κεφαλαίων και στον σχεδιασμό που ακολουθεί.

4.1 Λειτουργικές απαιτήσεις

Η ανάλυση απαιτήσεων αποτελεί το σημείο στο οποίο η γενική ιδέα της SmartNavix μετατρέπεται σε συγκεκριμένες λειτουργίες, ρόλους, δεδομένα και περιορισμούς. Σε ένα σύστημα πλοήγησης που βασίζεται σε προτιμήσεις, οι απαιτήσεις δεν περιορίζονται στην εύρεση διαδρομής. Πρέπει να περιγράφουν με σαφήνεια τι εισάγει ο χρήστης, πώς αξιοποιείται η πληροφορία, ποιες αποφάσεις λαμβάνει η εφαρμογή και ποια στοιχεία αποθηκεύονται για επόμενη χρήση. Η τεκμηρίωση απαιτήσεων βοηθά ώστε το λογισμικό να αναπτυχθεί με ελεγχόμενο τρόπο και να μπορεί αργότερα να αξιολογηθεί ως προς το αν καλύπτει τις ανάγκες που τέθηκαν στην αρχή [5].

Στην περίπτωση της SmartNavix, οι λειτουργικές απαιτήσεις προκύπτουν από τρεις βασικούς άξονες. Ο πρώτος είναι η διαχείριση χρήστη, δηλαδή εγγραφή, σύνδεση, επιβεβαίωση email, ανάκτηση κωδικού και διαχείριση προφίλ. Ο δεύτερος είναι η εξατομικευμένη αναζήτηση διαδρομής, όπου ο χρήστης δηλώνει προορισμό, στάσεις, προτιμήσεις, όχημα, αποφυγές και πρόσθετες επιλογές, ενώ η εφαρμογή παράγει εναλλακτικές διαδρομές και βοηθητικές εξηγήσεις αξιοποιώντας υπηρεσίες χαρτών και τεχνητής νοημοσύνης [64], [65], [66]. Ο τρίτος άξονας αφορά τη διαχείριση δεδομένων, όπως ιστορικό διαδρομών, presets, ρυθμίσεις, αξιολογήσεις και ενέργειες διαχειριστή.

Ο Πίνακας 4.2 παρουσιάζει τις βασικές λειτουργικές απαιτήσεις σε μορφή που μπορεί να χρησιμοποιηθεί και ως βάση για μελλοντικό έλεγχο αποδοχής. Οι απαιτήσεις διατυπώνονται σε επίπεδο συμπεριφοράς συστήματος και όχι σε επίπεδο κώδικα, ώστε να παραμένουν κατανοητές και έξω από το περιβάλλον υλοποίησης.

ΠΙΝΑΚΑΣ 4.2: Λειτουργικές απαιτήσεις SmartNavix

Κωδικός	Απαίτηση	Ρόλος
FR-01	Το σύστημα να υποστηρίζει εγγραφή, σύνδεση και επιβεβαίωση email.	Επισκέπτης / Χρήστης
FR-02	Το σύστημα να υποστηρίζει ανάκτηση και αλλαγή κωδικού πρόσβασης.	Χρήστης
FR-03	Ο χρήστης να μπορεί να αναζητά διαδρομή από τρέχουσα θέση ή αφετηρία προς προορισμό.	Χρήστης
FR-04	Ο χρήστης να μπορεί να ορίζει στάσεις, προτίμηση διαδρομής, διάθεση, όχημα, αποφυγές και ανάγκη EV charging.	Χρήστης
FR-05	Το σύστημα να εμφανίζει εναλλακτικές διαδρομές και βοηθητική εξήγηση της πρότασης.	Χρήστης
FR-06	Το σύστημα να αποθηκεύει διαδρομές που ξεκινά ο χρήστης και να επιτρέπει αξιολόγηση μετά τη χρήση.	Χρήστης
FR-07	Ο χρήστης να μπορεί να δημιουργεί και να επαναχρησιμοποιεί presets.	Χρήστης
FR-08	Ο χρήστης να μπορεί να ενημερώνει προφίλ, ρυθμίσεις εμφάνισης και προτιμήσεις χρήσης.	Χρήστης
FR-09	Ο χρήστης να μπορεί να εξάγει δεδομένα, να διαγράφει ιστορικό ή να ζητά διαγραφή λογαριασμού.	Χρήστης
FR-10	Ο διαχειριστής να μπορεί να βλέπει χρήστες, να αλλάζει ρόλους/κατάσταση επιβεβαίωσης και να διαγράφει λογαριασμούς όπου απαιτείται.	Διαχειριστής
FR-11	Ο διαχειριστής να μπορεί να παρακολουθεί βασικά analytics και audit logs.	Διαχειριστής

4.2 Μη λειτουργικές απαιτήσεις

Οι μη λειτουργικές απαιτήσεις περιγράφουν τις ιδιότητες ποιότητας που πρέπει να έχει η εφαρμογή. Για τη SmartNavix, η ποιότητα δεν αφορά μόνο την τεχνική ορθότητα των αποτελεσμάτων, αλλά και το αν ο χρήστης μπορεί να καταλάβει και να χειριστεί εύκολα τις επιλογές του. Επειδή η εφαρμογή χρησιμοποιείται σε περιβάλλον μετακίνησης, η διεπαφή πρέπει να παραμένει καθαρή, προβλέψιμη και χωρίς περιττή πολυπλοκότητα. Η εμπειρία χρήστη και ο ανθρωποκεντρικός σχεδιασμός λειτουργούν εδώ ως πρακτικά κριτήρια σχεδιασμού και όχι ως γενικές θεωρητικές έννοιες [31], [32].

Ιδιαίτερη σημασία έχει επίσης η προσβασιμότητα. Η ύπαρξη επιλογών όπως μεγάλο κείμενο και υψηλή αντίθεση δεν αρκεί από μόνη της για να θεωρηθεί μια εφαρμογή πλήρως προσβάσιμη, όμως δείχνει ότι η σχεδίαση πρέπει να λαμβάνει υπόψη διαφορετικές ανάγκες ανάγνωσης και χρήσης. Οι οδηγίες WCAG 2.2 χρησιμοποιούνται ως σημείο αναφοράς για την οργάνωση περιεχομένου, την αντιληπτότητα των στοιχείων και τη δυνατότητα χειρισμού της διεπαφής [9].

Παράλληλα, το σύστημα επεξεργάζεται δεδομένα που σχετίζονται με χρήστες, προτιμήσεις και ιστορικό διαδρομών. Επομένως, απαιτούνται σαφείς κανόνες προστασίας δεδομένων, δυνατότητα εξαγωγής ή διαγραφής στοιχείων και προσεκτική διαχείριση πρόσβασης. Ο Γενικός Κανονισμός Προστασίας Δεδομένων δίνει το γενικό πλαίσιο για νόμιμη και διαφανή επεξεργασία προσωπικών δεδομένων [75], ενώ οι οδηγίες OWASP Top 10 προσφέρουν πρακτικό σημείο αναφοράς για συνηθισμένες κατηγορίες κινδύνων σε web εφαρμογές [76].

ΠΙΝΑΚΑΣ 4.3: Μη λειτουργικές απαιτήσεις και περιορισμοί

Κωδικός	Ποιότητα	Απαίτηση / περιορισμός
NFR-01	Ευχρηστία	Η διεπαφή να παραμένει καθαρή, συνεπής και κατανοητή σε μη τεχνικούς χρήστες.
NFR-02	Προσβασιμότητα	Να υποστηρίζονται επιλογές όπως μεγάλο κείμενο και υψηλή αντίθεση, με αναφορά στις WCAG 2.2.
NFR-03	Ασφάλεια	Να εφαρμόζεται διαχωρισμός ρόλων, ασφαλής χειρισμός κωδικών και έλεγχος πρόσβασης.
NFR-04	Ιδιωτικότητα	Να αποφεύγεται περιττή συλλογή δεδομένων και να υποστηρίζονται εξαγωγή/διαγραφή δεδομένων.
NFR-05	Αξιοπιστία	Να γίνεται χειρισμός αποτυχιών σε εξωτερικά APIs χωρίς κατάρρευση της εφαρμογής.
NFR-06	Απόδοση	Οι απαντήσεις αναζήτησης να παραμένουν πρακτικά χρήσιμες για διαδραστική εφαρμογή πλοήγησης.
NFR-07	Συντηρησιμότητα	Η λογική να παραμένει οργανωμένη σε frontend, backend, services και βάση δεδομένων.
NFR-08	Πολυγλωσσικότητα	Η διεπαφή να μπορεί να υποστηρίζει περισσότερες από μία γλώσσες όπου απαιτείται.

4.3 Ρόλοι χρηστών: απλός χρήστης, διαχειριστής, contributor

Η SmartNavix διακρίνει τους χρήστες με βάση το επίπεδο πρόσβασης και τις ενέργειες που επιτρέπεται να εκτελούν. Ο απλός χρήστης αποτελεί τον κύριο ρόλο της εφαρμογής. Είναι το άτομο που αναζητά διαδρομές, ορίζει προτιμήσεις, χρησιμοποιεί presets, αξιολογεί αποτελέσματα και διαχειρίζεται προσωπικές ρυθμίσεις. Για αυτόν τον ρόλο, η εφαρμογή πρέπει να είναι άμεση και να μην απαιτεί τεχνική γνώση σχετικά με APIs, fuzzy logic ή εσωτερικά μοντέλα επιλογής.

Ο διαχειριστής έχει διαφορετική ευθύνη. Δεν χρησιμοποιεί την εφαρμογή μόνο για να βρει διαδρομή, αλλά για να εποπτεύει χρήστες, ρόλους, κατάσταση λογαριασμών, βασικά analytics και αρχεία ενεργειών. Η ύπαρξη διαχειριστικού ρόλου δημιουργεί πρόσθετες απαιτήσεις ελέγχου πρόσβασης, επειδή οι διαθέσιμες λειτουργίες μπορούν να επηρεάσουν δεδομένα άλλων χρηστών. Για αυτόν τον λόγο, οι ενέργειες διαχείρισης πρέπει να είναι περιορισμένες σε εξουσιοδοτημένους χρήστες και να καταγράφονται όπου είναι σκόπιμο [76].

Ο ρόλος contributor αντιμετωπίζεται στο παρόν στάδιο ως δυνητικός ή επεκτάσιμος ρόλος. Μπορεί να χρησιμοποιηθεί μελλοντικά για χρήστες που θα συμβάλλουν σε επιλεγμένο περιεχόμενο, όπως προτάσεις σημείων ενδιαφέροντος, κατηγοριοποιήσεις ή βοηθητικά δεδομένα, χωρίς να έχουν πλήρη δικαιώματα διαχειριστή. Η διάκριση αυτή είναι χρήσιμη για την εξέλιξη της εφαρμογής, επειδή αποφεύγει την πρακτική όπου κάθε αυξημένη δυνατότητα οδηγεί κατευθείαν σε πλήρη διαχειριστική πρόσβαση.

ΠΙΝΑΚΑΣ 4.1: Ρόλοι χρηστών και βασικές δυνατότητες

Ρόλος	Περιγραφή	Βασικές δυνατότητες
Επισκέπτης	Χρήστης χωρίς ενεργή σύνδεση.	Εγγραφή, επιβεβαίωση email, σύνδεση, ανάκτηση κωδικού.
Απλός χρήστης	Κύριος χρήστης της εφαρμογής πλοήγησης.	Αναζήτηση διαδρομής, φίλτρα, προτιμήσεις, presets, ιστορικό, αξιολόγηση, ρυθμίσεις.
Διαχειριστής	Εξουσιοδοτημένος χρήστης με πρόσβαση στο admin panel.	Διαχείριση χρηστών, ρόλων, επιβεβαίωσης, analytics και audit logs.
Contributor	Μελλοντικός/επεκτάσιμος ρόλος περιορισμένης συνεισφοράς.	Πιθανή συνεισφορά σε POI ή βοηθητικά δεδομένα χωρίς πλήρη admin δικαιώματα.

4.4 Use cases και σενάρια χρήσης

Τα use cases βοηθούν να περιγραφεί η εφαρμογή από την οπτική των ενεργειών που εκτελεί ο χρήστης. Δεν αντικαθιστούν την αναλυτική περιγραφή των απαιτήσεων, αλλά λειτουργούν ως γέφυρα ανάμεσα στις ανάγκες και στον σχεδιασμό. Η UML προσφέρει τυποποιημένο τρόπο αναπαράστασης τέτοιων αλληλεπιδράσεων, ώστε να φαίνονται οι ρόλοι, τα όρια του συστήματος και οι βασικές περιπτώσεις χρήσης [77].

Το βασικό σενάριο ξεκινά όταν ένας εγγεγραμμένος χρήστης συνδέεται στην εφαρμογή και εισάγει προορισμό. Στη συνέχεια μπορεί να επιλέξει ενδιάμεσες στάσεις, όχημα, διάθεση, προτίμηση διαδρομής, αποφυγές και ανάγκη για σταθμούς φόρτισης. Το σύστημα υπολογίζει εναλλακτικές διαδρομές μέσω εξωτερικών υπηρεσιών χαρτών, εμπλουτίζει την πρόταση με επεξηγηματικό ή βοηθητικό περιεχόμενο όπου χρειάζεται και εμφανίζει τις επιλογές στον χρήστη [64], [65], [66]. Αν ο χρήστης ξεκινήσει πλοήγηση, η εφαρμογή μπορεί να αποθηκεύσει τη διαδρομή και αργότερα να δεχθεί αξιολόγηση.

Ένα δεύτερο σενάριο αφορά την επαναχρησιμοποίηση προτιμήσεων. Ο χρήστης δεν χρειάζεται κάθε φορά να ορίζει από την αρχή όλες τις επιλογές. Η εφαρμογή μπορεί να αξιοποιήσει αποθηκευμένες filtered preferences, presets και ιστορικό ώστε να προτείνει την προτίμηση που ταιριάζει καλύτερα στη συνήθη συμπεριφορά του. Η λειτουργία αυτή συνδέει την ανάλυση απαιτήσεων με το μοντέλο fuzzy logic που παρουσιάζεται σε επόμενο κεφάλαιο.

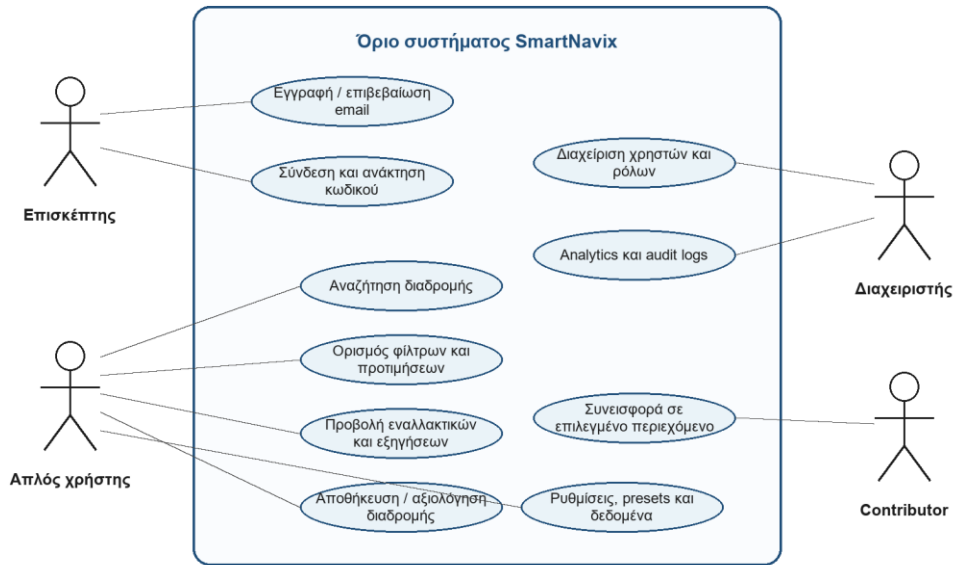
Στο σενάριο διαχείρισης, ο διαχειριστής συνδέεται στο admin panel και ελέγχει συνολικά στοιχεία χρηστών, ρόλους, κατάσταση επιβεβαίωσης και βασικά στατιστικά. Μπορεί επίσης να ενημερώσει στοιχεία λογαριασμών ή να διαγράψει χρήστη όταν αυτό απαιτείται. Οι ενέργειες αυτές πρέπει να αντιμετωπίζονται με προσοχή, επειδή σχετίζονται με προσωπικά δεδομένα και με την ακεραιότητα της εφαρμογής [75], [76].

4.5 UML διάγραμμα περιπτώσεων χρήσης

Το Διάγραμμα 4.1 αποτυπώνει συνοπτικά τις βασικές περιπτώσεις χρήσης της SmartNavix. Στο κέντρο βρίσκεται το όριο του συστήματος, ενώ έξω από αυτό εμφανίζονται οι βασικοί ρόλοι που αλληλεπιδρούν με την εφαρμογή. Ο επισκέπτης μπορεί να εγγραφεί, να επιβεβαιώσει τον λογαριασμό του και να συνδεθεί. Ο εγγεγραμμένος χρήστης εκτελεί τις κύριες λειτουργίες πλοήγησης και εξατομίκευσης, ενώ ο διαχειριστής έχει πρόσβαση σε λειτουργίες εποπτείας και διαχείρισης.

Το διάγραμμα δεν παρουσιάζει όλες τις τεχνικές λεπτομέρειες του backend ή της βάσης δεδομένων. Ο σκοπός του είναι να δείξει σε επίπεδο ανάλυσης ποια είναι τα βασικά σημεία επαφής ανθρώπου-συστήματος. Πιο λεπτομερή sequence, ER ή αρχιτεκτονικά διαγράμματα ανήκουν στα επόμενα κεφάλαια σχεδιασμού και υλοποίησης.

SmartNavix - UML διάγραμμα περιπτώσεων χρήσης



ΔΙΑΓΡΑΜΜΑ 4.1: UML διάγραμμα περιπτώσεων χρήσης SmartNavix

4.6 Περιορισμοί εγκατάστασης και χρήσης

Η εφαρμογή SmartNavix αναπτύσσεται ως web-based σύστημα και, στο παρόν στάδιο, προϋποθέτει περιβάλλον εκτέλεσης που μπορεί να υποστηρίξει frontend εφαρμογή, backend υπηρεσίες και βάση δεδομένων. Η λειτουργία της συνδέεται επίσης με εξωτερικά APIs για χάρτες, δρομολόγηση, τοποθεσίες και τεχνητή νοημοσύνη. Αυτό σημαίνει ότι για πλήρη χρήση απαιτούνται ενεργά API keys, σύνδεση στο διαδίκτυο και σωστή παραμετροποίηση των σχετικών υπηρεσιών [64], [65], [66].

Σε επίπεδο χρήσης, βασικός περιορισμός είναι ότι η ποιότητα της πρότασης εξαρτάται από την ποιότητα των εισόδων. Αν ο χρήστης εισαγάγει ασαφή προορισμό, ελλιπείς προτιμήσεις ή επιλογές που συγκρούονται μεταξύ τους, το σύστημα μπορεί να επιστρέψει λιγότερο κατάλληλες προτάσεις. Επιπλέον, επειδή μέρος της πληροφορίας προέρχεται από εξωτερικές υπηρεσίες, η εφαρμογή πρέπει να χειρίζεται περιπτώσεις καθυστέρησης, αποτυχίας απάντησης ή περιορισμών χρήσης των APIs.

Ένας ακόμη περιορισμός αφορά το ερευνητικό πλαίσιο της εργασίας. Η SmartNavix δεν επιδιώκει να αντικαταστήσει εμπορικές πλατφόρμες χαρτών μεγάλης κλίμακας. Αντίθετα, λειτουργεί ως εφαρμογή που αξιοποιεί υπάρχουσες υπηρεσίες και προσθέτει δικό της επίπεδο προσαρμογής, αποθήκευσης προτιμήσεων και ανάλυσης συμπεριφοράς χρήστη.

4.7 Απαιτήσεις ασφάλειας και προστασίας δεδομένων

Η ασφάλεια της SmartNavix πρέπει να εξεταστεί τόσο από την πλευρά του χρήστη όσο και από την πλευρά της εφαρμογής. Ο χρήστης παρέχει στοιχεία λογαριασμού, προτιμήσεις και ιστορικό χρήσης, ενώ ο διαχειριστής έχει πρόσβαση σε λειτουργίες που μπορούν να επηρεάσουν δεδομένα άλλων χρηστών. Επομένως, απαιτείται διαχωρισμός ρόλων, προσεκτικός έλεγχος πρόσβασης και ασφαλής χειρισμός κωδικών πρόσβασης. Οι σχετικές πρακτικές συνδέονται με γνωστές κατηγορίες κινδύνων web εφαρμογών, όπως προβλήματα ελέγχου πρόσβασης, λανθασμένη παραμετροποίηση και ανεπαρκής προστασία ευαίσθητων δεδομένων [76].

Από την πλευρά της προστασίας δεδομένων, η εφαρμογή πρέπει να αποφεύγει την περιττή συλλογή στοιχείων και να δίνει στον χρήστη δυνατότητα ελέγχου των δεδομένων του. Λειτουργίες όπως εξαγωγή δεδομένων, διαγραφή ιστορικού και διαγραφή λογαριασμού υπηρετούν αυτή τη λογική. Στο πλαίσιο του

GDPR, η διαφάνεια, ο περιορισμός σκοπού και η ελαχιστοποίηση δεδομένων αποτελούν βασικές αρχές που πρέπει να λαμβάνονται υπόψη στον σχεδιασμό [75].

Τέλος, επειδή η SmartNavix χρησιμοποιεί εξωτερικές υπηρεσίες, η ασφάλεια δεν αφορά μόνο τον δικό της κώδικα. Πρέπει να προστατεύονται τα API keys, να περιορίζεται η έκθεση εσωτερικών σφαλμάτων προς τον τελικό χρήστη και να καταγράφονται με προσοχή τα συμβάντα που βοηθούν στη διάγνωση προβλημάτων. Αυτές οι απαιτήσεις δεν είναι πρόσθετες πολυτέλειες· είναι μέρος της βασικής αξιοπιστίας ενός συστήματος που χειρίζεται λογαριασμούς και δεδομένα χρήσης.

Κεφάλαιο 5: Σχεδιασμός Συστήματος

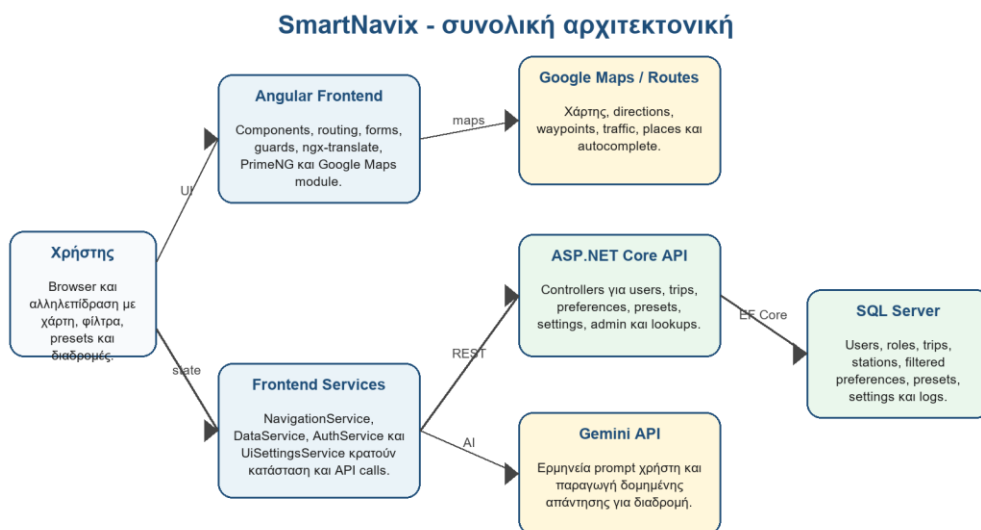
Το κεφάλαιο αυτό παρουσιάζει τον σχεδιασμό της SmartNavix σε επίπεδο αρχιτεκτονικής, τεχνολογιών, δεδομένων και ροής λειτουργιών. Στόχος δεν είναι να επαναληφθούν οι απαιτήσεις του προηγούμενου κεφαλαίου, αλλά να φανεί πώς οι απαιτήσεις μεταφράζονται σε συγκεκριμένη τεχνική οργάνωση.

5.1 Συνολική αρχιτεκτονική SmartNavix

Η SmartNavix σχεδιάζεται ως web-based εφαρμογή με διακριτά επίπεδα παρουσίασης, επιχειρησιακής λογικής και αποθήκευσης δεδομένων. Η επιλογή αυτή εξυπηρετεί τον στόχο της εργασίας, επειδή επιτρέπει στο frontend να εστιάζει στην εμπειρία πλοήγησης και στην αλληλεπίδραση με τον χάρτη, ενώ το backend οργανώνει τα μόνιμα δεδομένα, τους χρήστες, τις προτιμήσεις, τα ταξίδια και τις διαχειριστικές λειτουργίες. Η αρχιτεκτονική δεν αντιμετωπίζει το σύστημα ως μία ενιαία σελίδα με αποσπασματική λογική, αλλά ως σύνολο συνεργαζόμενων υποσυστημάτων.

Στο επίπεδο παρουσίασης χρησιμοποιείται Angular, ένα web framework που παρέχει components, routing, forms, dependency injection και εργαλεία για εφαρμογές που μεγαλώνουν σε μέγεθος και πολυπλοκότητα [78]. Στο επίπεδο υπηρεσιών χρησιμοποιείται ASP.NET Core, το οποίο προσφέρει modular HTTP pipeline, dependency injection, controllers, Swagger/OpenAPI τεκμηρίωση σε development περιβάλλον και δυνατότητα συνεργασίας με client-side frameworks όπως η Angular [79]. Η βάση δεδομένων υλοποιείται με SQL Server, ενώ η πρόσβαση στα δεδομένα γίνεται μέσω Entity Framework Core, το οποίο λειτουργεί ως object-relational mapper και συνδέει τις κλάσεις του domain με τους πίνακες της βάσης [80], [81].

Η συνολική αρχιτεκτονική παρουσιάζεται στο Διάγραμμα 5.1. Η λογική της εφαρμογής στηρίζεται σε τρία κύρια μονοπάτια: ο χρήστης αλληλεπιδρά με το Angular frontend, το frontend καλεί είτε το backend API είτε εξωτερικές υπηρεσίες χαρτών/τεχνητής νοημοσύνης, και το backend αποθηκεύει ή ανακτά δεδομένα από τη βάση. Η διάκριση αυτή κάνει πιο καθαρό το τι ανήκει σε κάθε επίπεδο και βοηθά στη συντήρηση της εφαρμογής.



ΔΙΑΓΡΑΜΜΑ 5.1: Συνολική αρχιτεκτονική SmartNavix

5.2 Frontend εφαρμογή

Το frontend αποτελεί το βασικό σημείο επαφής του χρήστη με τη SmartNavix. Η εφαρμογή οργανώνεται σε routes, components και services. Τα routes διαχωρίζουν τις κύριες οθόνες, όπως σύνδεση, εγγραφή, home, φίλτρα, προφίλ, ρυθμίσεις και admin panel. Η πρόσβαση σε προστατευμένες σελίδες ελέγχεται με auth guard, ενώ το admin panel προστατεύεται επιπλέον με admin guard. Με αυτόν τον τρόπο, η πλοήγηση μέσα στην εφαρμογή δεν είναι απλώς οπτική μετάβαση μεταξύ σελίδων, αλλά συνδέεται με την κατάσταση σύνδεσης και τον ρόλο του χρήστη.

Κεντρικό component είναι το home, όπου εμφανίζεται ο χάρτης, η αναζήτηση προορισμού, οι εναλλακτικές διαδρομές, οι δείκτες σημείων ενδιαφέροντος, τα presets και η προσομοίωση πλοήγησης. Η εφαρμογή χρησιμοποιεί Google Maps μέσω του Angular Google Maps module, ενώ η κατάσταση της αναζήτησης και των φίλτρων διατηρείται σε services με χρήση signals και RxJS. Έτσι, επιλογές που γίνονται στην οθόνη φίλτρων μπορούν να αξιοποιηθούν στη συνέχεια από την οθόνη χάρτη χωρίς να χαθούν κατά τη μετάβαση.

Η οθόνη φίλτρων επιτρέπει στον χρήστη να δηλώσει στάσεις, προτίμηση διαδρομής, διάθεση, μέγεθος οχήματος, αποφυγή διοδίων, αυτοκινητοδρόμων ή ferry, χρονικό πλαίσιο κίνησης και ανάγκη για EV charging. Οι σταθμοί συμπληρώνονται με υποβοήθηση από υπηρεσίες autocomplete και place details, ενώ γίνεται έλεγχος για διπλές στάσεις. Η σχεδίαση αυτή μεταφέρει μέρος της πολυπλοκότητας σε ελεγχόμενες φόρμες, ώστε η τελική αναζήτηση να έχει πιο καθαρά και αξιοποιήσιμα δεδομένα εισόδου.

Σημαντικό στοιχείο του frontend είναι και η πολυγλωσσική υποστήριξη. Η εφαρμογή χρησιμοποιεί translation keys, ώστε τα στατικά κείμενα της διεπαφής, οι επιλογές φίλτρων και οι ετικέτες να μπορούν να αποδοθούν σε διαφορετικές γλώσσες. Αυτό συνδέεται με τις απαιτήσεις ευχρηστίας και προσβασιμότητας, επειδή η πλοήγηση και οι οδηγίες πρέπει να είναι κατανοητές χωρίς να εξαρτώνται από σκληρά κωδικοποιημένο κείμενο.

5.3 Backend υπηρεσίες

Το backend υλοποιείται ως ASP.NET Core Web API. Η εφαρμογή εκκινεί από το Program.cs, όπου δηλώνονται οι controllers, η σύνδεση με τη βάση μέσω ApplicationDbContext, το Swagger για development τεκμηρίωση, η πολιτική CORS για επικοινωνία με το Angular frontend και οι υπηρεσίες που χρησιμοποιούνται με dependency injection. Η επιλογή του ASP.NET Core ταιριάζει με την ανάγκη για οργανωμένα endpoints, διαχωρισμό ευθυνών και δυνατότητα επέκτασης της εφαρμογής [79].

Οι controllers οργανώνουν τη λειτουργικότητα σε επιμέρους περιοχές. Ο UserController χειρίζεται εγγραφή, σύνδεση, επιβεβαίωση email, ανάκτηση και αλλαγή κωδικού. Ο TripController διαχειρίζεται τη δημιουργία, ενημέρωση, αξιολόγηση και ανάκτηση διαδρομών. Ο FilteredPreferenceController αποθηκεύει και ανακτά προτιμήσεις χρήστη, ενώ ο PresetController υποστηρίζει γρήγορες αποθηκευμένες τοποθεσίες. Ο SettingsController επιτρέπει διαχείριση ρυθμίσεων, εξαγωγή δεδομένων, διαγραφή ιστορικού και διαγραφή λογαριασμού. Τέλος, ο AdminController συγκεντρώνει λειτουργίες εποπτείας, όπως χρήστες, ρόλοι, analytics και audit logs.

Η ύπαρξη ξεχωριστών υπηρεσιών, όπως PasswordService και EmailService, περιορίζει την επανάληψη κώδικα και βοηθά ώστε ευαίσθητες λειτουργίες να μην είναι διάσπαρτες στους controllers. Η χρήση BCrypt για κωδικούς πρόσβασης και η αποστολή email για επιβεβαίωση ή reset εντάσσονται στη λογική ότι ο λογαριασμός χρήστη αποτελεί βασικό σημείο ασφάλειας της εφαρμογής. Παρόλα αυτά, στο παρόν στάδιο η εφαρμογή παραμένει εκπαιδευτική/ερευνητική και ορισμένα σημεία, όπως η περαιτέρω σκλήρυνση της αυθεντικοποίησης, μπορούν να αποτελέσουν αντικείμενο μελλοντικής επέκτασης.

5.4 Βάση δεδομένων

Η βάση δεδομένων οργανώνει τις οντότητες που χρειάζεται η SmartNavix για να λειτουργήσει πέρα από μία προσωρινή αναζήτηση χάρτη. Ο ApplicationDbContext περιλαμβάνει χρήστες, avatars, ρόλους, trips, preferences, moods, stations, vehicles, filtered preferences, presets, preset icons, admin action logs και user settings. Η χρήση Entity Framework Core επιτρέπει η πρόσβαση στα δεδομένα να γίνεται μέσα από .NET classes και DbSet συλλογές, ενώ το ίδιο μοντέλο μπορεί να εξελιχθεί με migrations όταν αλλάζει το σχήμα της βάσης [80].

Οι σχέσεις ανάμεσα στις οντότητες έχουν άμεση σημασία για τη λειτουργία της εφαρμογής. Ένα trip μπορεί να συνδέεται με vehicle και με πολλές stations. Τα presets και τα filtered preferences συνδέονται με χρήστη, αλλά χρησιμοποιούνται delete behaviors που αποτρέπουν ανεπιθύμητη απώλεια όλων των σχετικών εγγραφών όταν ένα στοιχείο αλλάξει. Τα user settings έχουν μοναδική σχέση με τον χρήστη, ενώ τα admin action logs κρατούν πληροφορία για ενέργειες διαχείρισης.

Σε επίπεδο βάσης, έχουν οριστεί indexes και περιορισμοί που υπηρετούν τόσο την ακεραιότητα όσο και την απόδοση. Για παράδειγμα, το username είναι μοναδικό, τα vehicles έχουν μοναδικό code, ενώ υπάρχουν indexes για συνδυασμούς όπως UserID και AppliedAt στα filtered preferences. Η ύπαρξη τέτοιων περιορισμών είναι σημαντική, επειδή η εφαρμογή δεν βασίζεται μόνο στον έλεγχο του frontend· η ίδια η βάση πρέπει να βοηθά στη διατήρηση συνεπών δεδομένων [81].

5.5 Εξωτερικά APIs: Google Maps και Gemini

Η SmartNavix δεν υλοποιεί δικό της σύστημα χαρτογράφησης. Αντίθετα, αξιοποιεί εξωτερικές υπηρεσίες ώστε να επικεντρωθεί στο επίπεδο της εξατομίκευσης. Το Google Maps Platform παρέχει τη βάση για χάρτη, γεωκωδικοποίηση, autocomplete, places και υπολογισμό διαδρομών [64]. Η Routes API και οι σχετικές υπηρεσίες δρομολόγησης υποστηρίζουν επιλογές όπως εναλλακτικές διαδρομές, waypoints, traffic options, αποφυγές και επιστροφή στοιχείων διαδρομής [65].

Το Gemini API αξιοποιείται για να μετατρέπει την πιο ελεύθερη διατύπωση του χρήστη σε δομημένη πληροφορία. Στη συγκεκριμένη υλοποίηση, το frontend στέλνει prompt που περιλαμβάνει τρέχουσα θέση, ανάγκη χρήστη και φίλτρα. Η απάντηση αναμένεται σε JSON μορφή με προορισμό, πιθανή αφετηρία, επεξηγηματικό κείμενο και προαιρετικά POI intents. Η δυνατότητα παραγωγής περιεχομένου μέσω API είναι ο λόγος που το Gemini ταιριάζει σε αυτό το σημείο της ροής [66].

Ένα σημείο που πρέπει να αναφερθεί με ειλικρίνεια είναι ότι, στην παρούσα υλοποίηση, μέρος της επικοινωνίας με το Gemini γίνεται από το frontend. Αυτό απλοποιεί την ανάπτυξη στο πλαίσιο της διπλωματικής, αλλά δεν είναι η πιο ασφαλής επιλογή για παραγωγικό περιβάλλον, επειδή τα API keys πρέπει ιδανικά να προστατεύονται σε server-side επίπεδο. Επομένως, η αρχιτεκτονική καταγράφει την τρέχουσα λειτουργική λύση, αλλά αναγνωρίζει ότι η μεταφορά τέτοιων κλήσεων στο backend αποτελεί λογική μελλοντική βελτίωση.

5.6 Ροή δεδομένων στο σύστημα

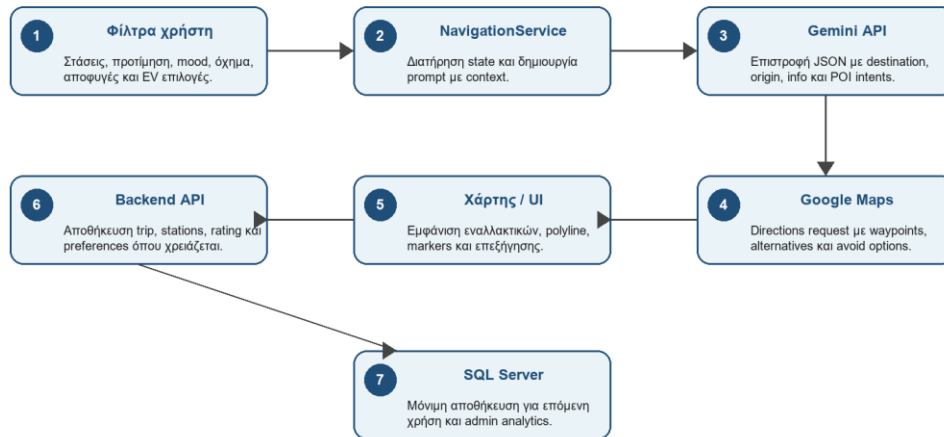
Η βασική ροή δεδομένων ξεκινά από τις επιλογές του χρήστη. Ο χρήστης μπορεί πρώτα να συμπληρώσει φίλτρα, στάσεις και προτιμήσεις στην αντίστοιχη οθόνη. Οι επιλογές αυτές αποθηκεύονται προσωρινά στο NavigationService, ενώ μπορούν επίσης να αποθηκευτούν ως filtered preference στο backend. Όταν ο χρήστης επιστρέψει στην αρχική οθόνη και ζητήσει διαδρομή, οι επιλογές μετατρέπονται σε prompt και συνδυάζονται με τον προορισμό.

Στη συνέχεια, το NavigationService καλεί το Gemini API για να ερμηνεύσει το αίτημα και να επιστρέψει δομημένη πληροφορία. Αν η απάντηση είναι έγκυρη, δημιουργείται DirectionsRequest προς τις υπηρεσίες Google Maps, όπου ενσωματώνονται waypoints, αποφυγές και πιθανές επιλογές κίνησης. Το αποτέλεσμα εμφανίζεται στον χάρτη ως εναλλακτικές διαδρομές, μαζί με συνοπτική εξήγηση, δείκτες και βοηθητικά POI όταν υπάρχουν.

Όταν ο χρήστης ξεκινήσει πλοήγηση, η εφαρμογή δημιουργεί εγγραφή trip στο backend με στοιχεία όπως αφετηρία, προορισμός, απόσταση, ημερομηνία, όχημα και στάσεις. Μετά τη χρήση, ο χρήστης μπορεί να αξιολογήσει τη διαδρομή. Αυτή η ροή είναι κρίσιμη για τη SmartNavix, επειδή τα δεδομένα χρήσης και οι αξιολογήσεις μπορούν να αξιοποιηθούν σε επόμενες φάσεις για βελτίωση των προτάσεων.

Το Διάγραμμα 5.2 παρουσιάζει συνοπτικά τη ροή αναζήτησης διαδρομής, από την επιλογή φίλτρων μέχρι την αποθήκευση του trip. Η αναπαράσταση αυτή βοηθά να φανεί ότι η πρόταση διαδρομής δεν παράγεται από ένα μόνο βήμα, αλλά από αλληλουχία ενεργειών ανάμεσα σε frontend, υπηρεσίες AI, υπηρεσίες χαρτών και backend.

SmartNavix - ροή αναζήτησης διαδρομής



ΔΙΑΓΡΑΜΜΑ 5.2: Ροή αναζήτησης διαδρομής SmartNavix

5.7 UML / sequence diagrams βασικών λειτουργιών

Τα διαγράμματα του κεφαλαίου χρησιμοποιούνται ως εργαλεία κατανόησης της αρχιτεκτονικής. Το Διάγραμμα 5.1 είναι αρχιτεκτονικό και δείχνει τα κύρια υποσυστήματα. Το Διάγραμμα 5.2 είναι πιο κοντά σε sequence/data-flow λογική και δείχνει τη σειρά με την οποία εκτελείται η αναζήτηση διαδρομής. Η χρήση UML και συναφών διαγραμματικών αναπαραστάσεων βοηθά ώστε το σύστημα να τεκμηριώνεται με τρόπο που μπορεί να διαβαστεί και από τεχνικούς αναγνώστες πέρα από τον ίδιο τον υλοποιητή [77].

Στην παρούσα φάση τα διαγράμματα δίνουν τη βασική εικόνα. Σε επόμενα κεφάλαια μπορούν να προστεθούν πιο εξειδικευμένα διαγράμματα, όπως ER diagram για τη βάση δεδομένων ή sequence diagrams για λειτουργίες όπως reset password, αποθήκευση preset και χρήση admin panel. Έτσι, η τεκμηρίωση παραμένει σταδιακή: πρώτα παρουσιάζεται η συνολική αρχιτεκτονική και στη συνέχεια, όπου χρειάζεται, αναλύονται επιμέρους μηχανισμοί.

5.8 Πολυγλωσσική υποστήριξη

Η πολυγλωσσική υποστήριξη δεν αντιμετωπίζεται ως διακοσμητικό χαρακτηριστικό. Σε μια εφαρμογή πλοήγησης, ο χρήστης χρειάζεται να καταλαβαίνει γρήγορα τις επιλογές, τις οδηγίες, τα μηνύματα σφάλματος και τις προειδοποιήσεις. Για αυτόν τον λόγο, το frontend χρησιμοποιεί μηχανισμό μεταφράσεων με translation keys, ώστε τα κείμενα να μπορούν να αλλάζουν χωρίς να τροποποιείται η λογική των components.

Στη SmartNavix η πολυγλωσσικότητα συνδέεται και με δεδομένα που έρχονται από το backend. Για παράδειγμα, preferences, moods, vehicles και preset icons μπορούν να έχουν translation fields, ώστε η ίδια εγγραφή να εμφανίζεται με κατάλληλη ετικέτα στη διεπαφή. Αυτή η επιλογή είναι πιο ευέλικτη από το να αποθηκεύεται μόνο ένα τελικό κείμενο, επειδή επιτρέπει στο frontend να αποφασίζει τη γλώσσα εμφάνισης ανάλογα με τις ρυθμίσεις του χρήστη.

Η σχεδίαση αυτή βοηθά και στη μελλοντική επέκταση. Αν προστεθούν νέες γλώσσες, δεν χρειάζεται να αλλάξει η δομή της εφαρμογής, αλλά να συμπληρωθούν τα αντίστοιχα αρχεία μεταφράσεων και να διατηρηθούν συνεπείς οι κωδικοί μετάφρασης. Έτσι, η πολυγλωσσικότητα παραμένει τεχνικά ελεγχόμενη και όχι διάσπαρτη μέσα στον κώδικα.

5.9 Επιλογή τεχνολογιών και λόγοι επιλογής

Οι τεχνολογίες που χρησιμοποιούνται στη SmartNavix επιλέχθηκαν με κριτήριο τη σαφή διάκριση ευθυνών, την επεκτασιμότητα και τη δυνατότητα γρήγορης ανάπτυξης μιας λειτουργικής εφαρμογής. Η Angular προσφέρει οργανωμένη δομή frontend με components, routing, forms και services [78]. Το ASP.NET Core επιτρέπει την ανάπτυξη web APIs με controllers, dependency injection, configuration και εργαλεία τεκμηρίωσης [79]. Το Entity Framework Core και ο SQL Server δίνουν ένα σταθερό σχήμα για αποθήκευση και ανάκτηση δεδομένων [80], [81].

Η επιλογή Google Maps και Gemini δεν έγινε επειδή η εργασία θέλει να κατασκευάσει από την αρχή χάρτες ή μοντέλο φυσικής γλώσσας. Αντίθετα, η εργασία επικεντρώνεται στο πώς μπορούν να συνδυαστούν υπάρχουσες υπηρεσίες με προτιμήσεις χρήστη, φίλτρα και αποθηκευμένο ιστορικό για να παραχθεί πιο προσαρμοσμένη πρόταση διαδρομής. Ο Πίνακας 5.1 συνοψίζει τις κύριες τεχνολογίες και τον ρόλο τους στο σύστημα.

ΠΙΝΑΚΑΣ 5.1: Τεχνολογίες υλοποίησης και ρόλος στο σύστημα

Τεχνολογία	Χρήση στη SmartNavix	Λόγος επιλογής
Angular	Frontend, routing, φόρμες, components, services και Google Maps integration.	Οργανωμένη ανάπτυξη διαδραστικής web εφαρμογής.
ASP.NET Core	Backend Web API, controllers, Swagger, dependency injection και CORS.	Καθαρή δομή υπηρεσιών και σύνδεση με .NET οικοσύστημα.
Entity Framework Core	Πρόσβαση σε δεδομένα μέσω ApplicationDbContext και DbSet οντοτήτων.	Σύνδεση αντικειμενοστραφούς κώδικα με σχεσιακή βάση.
SQL Server	Αποθήκευση χρηστών, trips, preferences, presets, settings και logs.	Σχεσιακή αποθήκευση με constraints, indexes και σταθερότητα.
Google Maps / Routes	Χάρτες, διαδρομές, waypoints, traffic options, places και autocomplete.	Ωριμη γεωχωρική υποδομή για web εφαρμογές πλοήγησης.
Gemini API	Ερμηνεία prompt χρήστη και παραγωγή δομημένης απάντησης για διαδρομή/POI intents.	Ενσωμάτωση AI επιπέδου φυσικής γλώσσας χωρίς ανάπτυξη δικού μοντέλου.

Κεφάλαιο 6: Σχεδιασμός Βάσης Δεδομένων

Το κεφάλαιο αυτό παρουσιάζει τον σχεδιασμό της βάσης δεδομένων της SmartNavix, όπως προκύπτει από τα μοντέλα του backend και το AppDbContext. Η ανάλυση εστιάζει στις οντότητες, στις συσχετίσεις, στους κανόνες ακεραιότητας και στον τρόπο με τον οποίο η βάση υποστηρίζει την εξατομίκευση της πλοήγησης.

6.1 Περιγραφή μοντέλου δεδομένων

Η βάση δεδομένων της SmartNavix σχεδιάζεται ως σχεσιακό μοντέλο, επειδή τα βασικά δεδομένα της εφαρμογής έχουν σαφείς οντότητες, σχέσεις και κανόνες ακεραιότητας. Η επιλογή αυτή συμβαδίζει με τη λογική του σχεσιακού μοντέλου, όπου τα δεδομένα οργανώνονται σε πίνακες και οι συσχετίσεις τους ελέγχονται με κλειδιά και περιορισμούς [82]. Παράλληλα, η αποτύπωση των οντοτήτων με ER λογική βοηθά να φανεί πριν από την υλοποίηση ποια πληροφορία ανήκει σε κάθε πίνακα και πώς συνδέεται με τις υπόλοιπες [83].

Στην υλοποίηση, το σχήμα της βάσης ορίζεται μέσα από το Entity Framework Core και εκτελείται σε SQL Server [80], [81]. Το AppDbContext λειτουργεί ως το κεντρικό σημείο αντιστοίχισης ανάμεσα στις κλάσεις του backend και στους πίνακες της βάσης. Έτσι, ο κώδικας δεν αντιμετωπίζει τη βάση ως ανεξάρτητο αποθηκευτικό χώρο, αλλά ως μέρος του ίδιου μοντέλου λειτουργίας της εφαρμογής.

Οι οντότητες μπορούν να χωριστούν σε τέσσερις βασικές ομάδες. Η πρώτη αφορά τον χρήστη και τα στοιχεία προφίλ, όπως ρόλο, avatar, βασική προτίμηση και ρυθμίσεις. Η δεύτερη αφορά τη χρήση της πλοήγησης, δηλαδή τις διαδρομές, το όχημα και τις στάσεις. Η τρίτη αφορά την εξατομίκευση, με filtered preferences και presets. Η τέταρτη αφορά στοιχεία διαχείρισης και παρακολούθησης ενεργειών, όπως τα admin action logs. Ο Πίνακας 6.1 συνοψίζει αυτές τις οντότητες.

ΠΙΝΑΚΑΣ 6.1: Κύριες οντότητες βάσης δεδομένων SmartNavix

Οντότητα	Ρόλος στη βάση	Κύρια πεδία / σημειώσεις
User	Κεντρική οντότητα λογαριασμού χρήστη.	UserName, Password, Name, Surname, Email, Phone, IsVerified, VerificationToken, RoleId, AvatarId, PreferenceId.
LK_Roles	Lookup πίνακας ρόλων και λειτουργικών περιορισμών.	RoleName, Description, MaxDailyRequests, CanVoteRoutes.
LK_Avatars	Διαθέσιμα avatars προφίλ.	AvatarName, AvatarURL, CreatedAt.
LK_Preferences	Βασικές προτιμήσεις πλοήγησης.	Code, Label, Icon, TranslationField, Prompt.
LK_Moods	Διαθέσιμες διαθέσεις/καταστάσεις χρήστη.	Code, Label, Icon, TranslationField, Prompt.
LK_Vehicle	Κατηγορίες οχημάτων.	Code, Label, TranslationField. Το Code είναι μοναδικό.
Trips	Ιστορικό ή εγγραφή χρήσης διαδρομής.	UserId, Destination, Departure, DistanceKM, Score, TripDate, VehicleID.
Station	Ενδιάμεσες στάσεις μιας διαδρομής.	TripID, Street, Number, CityArea, PostalCode, Position.
FilteredPreference	Στιγμιότυπο εφαρμοσμένων φίλτρων/προτιμήσεων.	SelectedPreferenceCode, MoodCode, VehicleSize, avoid flags, traffic fields, StationsJson, AppliedAt.
Preset	Αποθηκευμένες διευθύνσεις/αγαπημένες επιλογές χρήστη.	UserId, Street, Number, CityArea, PostalCode, Position, PresetIconId.
LK_Preset_Icon	Lookup πίνακας εικονιδίων preset.	IconData, TranslationField.
UserSettings	Προσωπικές ρυθμίσεις διεπαφής και AI.	AiAggressiveness, AlwaysShowRouteExplanation, AlternativeRoutesCount, Theme, MapStyle, DistanceUnit, TimeFormat, ChipDensity, LargeText, HighContrast.
AdminActionLog	Καταγραφή ενεργειών διαχείρισης.	AdminUserId, TargetUserId, ActionType, Details, CreatedAt.

6.2 Πίνακες χρηστών και ρόλων

Ο πίνακας User αποτελεί τον πυρήνα του μοντέλου χρηστών. Περιλαμβάνει στοιχεία ταυτοποίησης και επικοινωνίας, όπως username, κωδικό πρόσβασης, όνομα, επώνυμο, email και τηλέφωνο, καθώς και πεδία επαλήθευσης λογαριασμού. Το username έχει μοναδικό index, ώστε να αποφεύγονται διπλοεγγραφές, ενώ το email έχει index για ταχύτερη αναζήτηση. Στον ίδιο πίνακα υπάρχουν ξένα κλειδιά προς Role, Avatar και Preference, επειδή ο χρήστης συνδέεται με δικαιώματα, οπτική αναπαράσταση προφίλ και αρχική προτίμηση πλοήγησης.

Ο πίνακας LK_Roles αποθηκεύει τους ρόλους του συστήματος. Στην τρέχουσα σχεδίαση δεν κρατά μόνο το όνομα και την περιγραφή του ρόλου, αλλά και πεδία όπως MaxDailyRequests και CanVoteRoutes. Με αυτόν τον τρόπο ο ρόλος δεν είναι απλώς ετικέτα πρόσβασης, αλλά μπορεί να επηρεάσει λειτουργικούς περιορισμούς της εφαρμογής. Η σύνδεση Role-User είναι σχέση ένα προς πολλά, αφού ένας ρόλος μπορεί να αντιστοιχεί σε πολλούς χρήστες.

Οι πίνακες LK_Avatars και LK_Preferences λειτουργούν ως βοηθητικοί πίνακες επιλογών. Τα avatars κρατούν διαθέσιμες εικόνες προφίλ, ενώ οι preferences κρατούν κωδικό, ετικέτα, εικονίδιο, πεδίο μετάφρασης και prompt. Επειδή οι πίνακες αυτοί χρησιμοποιούνται από πολλούς χρήστες, η ύπαρξή τους μειώνει την επανάληψη δεδομένων και κάνει πιο εύκολη τη μελλοντική προσθήκη νέων επιλογών. Τα προσωπικά δεδομένα των χρηστών απαιτούν ιδιαίτερη προσοχή, επειδή πρέπει να συνδέονται με αρχές ελαχιστοποίησης, νόμιμης χρήσης και προστασίας δεδομένων όπως προβλέπει ο ΓΚΠΔ [75].

6.3 Πίνακες διαδρομών, στάσεων και σημείων ενδιαφέροντος

Η βασική αποτύπωση μιας πλοήγησης γίνεται στον πίνακα Trips. Κάθε εγγραφή μπορεί να συνδέεται με χρήστη και όχημα και αποθηκεύει προορισμό, αφετηρία, απόσταση, βαθμολογία και ημερομηνία διαδρομής. Η βαθμολογία επιτρέπει στη SmartNavix να κρατά μια πρώτη μορφή feedback, η οποία μπορεί να αξιοποιηθεί αργότερα για ανάλυση ή βελτίωση προτάσεων.

Οι ενδιάμεσες στάσεις αποθηκεύονται στον πίνακα Station. Κάθε στάση ανήκει σε συγκεκριμένο trip και περιγράφεται με στοιχεία διεύθυνσης, όπως οδός, αριθμός, περιοχή, ταχυδρομικός κώδικας και θέση στη σειρά στάσεων. Η σχέση Trip-Station είναι ένα προς πολλά και έχει cascade delete. Αυτό σημαίνει ότι όταν διαγραφεί μια διαδρομή, διαγράφονται και οι στάσεις που δεν έχουν νόημα χωρίς αυτήν.

Ο πίνακας LK_Vehicle κρατά τις διαθέσιμες κατηγορίες οχημάτων, όπως small car, SUV/van, large van, truck και motorcycle. Η συσχέτιση με το trip επιτρέπει στη διαδρομή να συνδέεται με το όχημα που χρησιμοποίησε ο χρήστης, χωρίς να αντιγράφονται οι λεπτομέρειες του οχήματος σε κάθε εγγραφή. Αντίθετα, τα σημεία ενδιαφέροντος δεν αποθηκεύονται στην παρούσα φάση ως ξεχωριστός μόνιμος πίνακας. Παράγονται ή ανακτώνται κατά τη χρήση μέσω των υπηρεσιών χαρτών και προτάσεων, ώστε η βάση να κρατά κυρίως το ιστορικό και τις επιλογές του χρήστη και όχι αντίγραφο δεδομένων εξωτερικών παρόχων [64], [65].

6.4 FilteredPreference και αποθήκευση προτιμήσεων

Η οντότητα FilteredPreference αποθηκεύει το στιγμιότυπο των επιλογών που εφαρμόστηκαν από τον χρήστη σε μια αναζήτηση ή σε ένα σενάριο πλοήγησης. Περιλαμβάνει τον κωδικό της επιλεγμένης προτίμησης, το αντίστοιχο prompt, mood code, μέγεθος οχήματος, επιλογές αποφυγής διωδίων, αυτοκινητοδρόμων και ferry, καθώς και ρυθμίσεις σχετικές με την κυκλοφορία.

Η ύπαρξη των πεδίων TrafficStartDateTime και TrafficEndDateTime δείχνει ότι η εφαρμογή δεν αποθηκεύει μόνο γενικές προτιμήσεις, αλλά και χρονικό πλαίσιο για τη χρήση τους. Το πεδίο IncludeEvChargingStations καλύπτει ειδική ανάγκη για ηλεκτρικά οχήματα, ενώ το StationsJson επιτρέπει την αποθήκευση δομημένης πληροφορίας στάσεων όταν οι στάσεις χρειάζεται να διατηρηθούν μαζί με το φίλτρο. Η επιλογή JSON δεν αντικαθιστά τις κανονικοποιημένες στάσεις του trip, αλλά εξυπηρετεί την αποθήκευση ενός ευέλικτου στιγμιότυπου φίλτρων.

Στο AppDbContext έχει οριστεί index στα πεδία UserID και AppliedAt, ώστε να γίνεται αποδοτικότερη αναζήτηση των πιο πρόσφατων εφαρμοσμένων προτιμήσεων ανά χρήστη. Επιπλέον, η σχέση με τον χρήστη έχει συμπεριφορά SetNull στη διαγραφή, κάτι που δείχνει ότι η εφαρμογή μπορεί

να διατηρεί ιστορικά ή στατιστικά δεδομένα χωρίς να υπάρχει πλέον σύνδεση με συγκεκριμένο ενεργό λογαριασμό, εφόσον αυτό είναι συμβατό με τους κανόνες προστασίας δεδομένων [75].

6.5 Presets και εξατομίκευση χρήστη

Τα presets είναι ένας πρακτικός μηχανισμός εξατομίκευσης. Ο πίνακας Preset αποθηκεύει διευθύνσεις ή αγαπημένα σημεία που ο χρήστης θέλει να επαναχρησιμοποιεί, με στοιχεία όπως οδός, αριθμός, περιοχή, ταχυδρομικός κώδικας και θέση εμφάνισης. Έτσι, η εφαρμογή μπορεί να προσφέρει γρήγορη πρόσβαση σε συχνές επιλογές χωρίς ο χρήστης να τις εισάγει κάθε φορά από την αρχή.

Η σχέση User-Preset είναι ένα προς πολλά, επειδή κάθε χρήστης μπορεί να έχει περισσότερα από ένα presets. Το index στα πεδία UserID και Position βοηθά στην ανάκτηση των presets με τη σωστή σειρά. Σε επίπεδο εμπειρίας χρήστη, αυτό συνδέεται με την ανάγκη η εφαρμογή να θυμάται σταθερές επιλογές και να μειώνει τον χρόνο εισαγωγής δεδομένων.

Ο πίνακας LK_Preset_Icon κρατά τα διαθέσιμα εικονίδια που μπορούν να χρησιμοποιηθούν για την οπτική αναγνώριση ενός preset. Η σχέση PresetIcon-Preset έχει Restrict στη διαγραφή, ώστε να μην αφαιρείται εύκολα ένα εικονίδιο που χρησιμοποιείται ήδη από εγγραφές preset. Αυτή η απόφαση προστατεύει τη συνέπεια των δεδομένων και αποφεύγει σπασμένες αναφορές στη διεπαφή.

6.6 Συσχετίσεις οντοτήτων

Οι συσχετίσεις της βάσης ακολουθούν τη λογική ότι κάθε πληροφορία πρέπει να ανήκει στο σημείο όπου έχει λειτουργικό νόημα. Ο χρήστης συνδέεται με ρόλο, avatar, προτίμηση, ρυθμίσεις, trips, filtered preferences και presets. Το trip συνδέεται με στάσεις και προαιρετικά με όχημα. Τα lookup tables κρατούν επαναχρησιμοποιήσιμες επιλογές, ενώ οι πίνακες ιστορικού κρατούν γεγονότα χρήσης.

Ο Πίνακας 6.2 παρουσιάζει τις βασικές σχέσεις και τις συμπεριφορές διαγραφής που έχουν δηλωθεί ή προκύπτουν από το μοντέλο. Οι κανόνες αυτοί είναι σημαντικοί, επειδή καθορίζουν τι συμβαίνει όταν αφαιρείται μια εγγραφή. Για παράδειγμα, η διαγραφή ενός trip διαγράφει τις στάσεις του, ενώ η διαγραφή ενός χρήστη δεν διαγράφει απαραίτητα όλα τα filtered preferences ή presets, αλλά μπορεί να μηδενίσει τη σύνδεσή τους με τον χρήστη.

Η οργάνωση αυτή επιτρέπει στο μοντέλο να παραμένει σχετικά καθαρό: τα στοιχεία λογαριασμού δεν αναμειγνύονται με τα στοιχεία διαδρομής, τα lookup δεδομένα δεν αντιγράφονται σε κάθε εγγραφή, και τα δεδομένα εξατομίκευσης μπορούν να εξελιχθούν χωρίς να χρειαστεί να αλλάξει όλη η βάση.

ΠΙΝΑΚΑΣ 6.2: Βασικές συσχετίσεις και κανόνες διαγραφής

Σχέση	Πληθικότητα	Κανόνας / παρατήρηση
LK_Roles - User	1:N	Κάθε χρήστης έχει έναν ρόλο, ενώ ένας ρόλος μπορεί να ανήκει σε πολλούς χρήστες.
LK_Avatars - User	1:N	Πολλοί χρήστες μπορούν να χρησιμοποιούν το ίδιο avatar.
LK_Preferences - User	1:N	Η αρχική προτίμηση χρήστη συνδέεται με lookup εγγραφή preference.
User - Trips	1:N	Ένας χρήστης μπορεί να έχει πολλές διαδρομές.
Trips - Station	1:N	Οι στάσεις διαγράφονται με cascade όταν διαγράφεται το trip.
LK_Vehicle - Trips	1:N	Το όχημα συνδέεται προαιρετικά με trip και σε διαγραφή οχήματος εφαρμόζεται SetNull.
User - FilteredPreference	1:N	Οι εφαρμοσμένες προτιμήσεις συνδέονται με χρήστη και σε διαγραφή χρήστη εφαρμόζεται SetNull.
User - Preset	1:N	Τα presets ανήκουν σε χρήστη και σε διαγραφή χρήστη εφαρμόζεται SetNull.
LK_Preset_Icon - Preset	1:N	Το εικονίδιο preset προστατεύεται με Restrict ώστε να μη διαγραφεί αν χρησιμοποιείται.
User - UserSettings	1:1	Κάθε χρήστης έχει μία εγγραφή ρυθμίσεων και η διαγραφή χρήστη

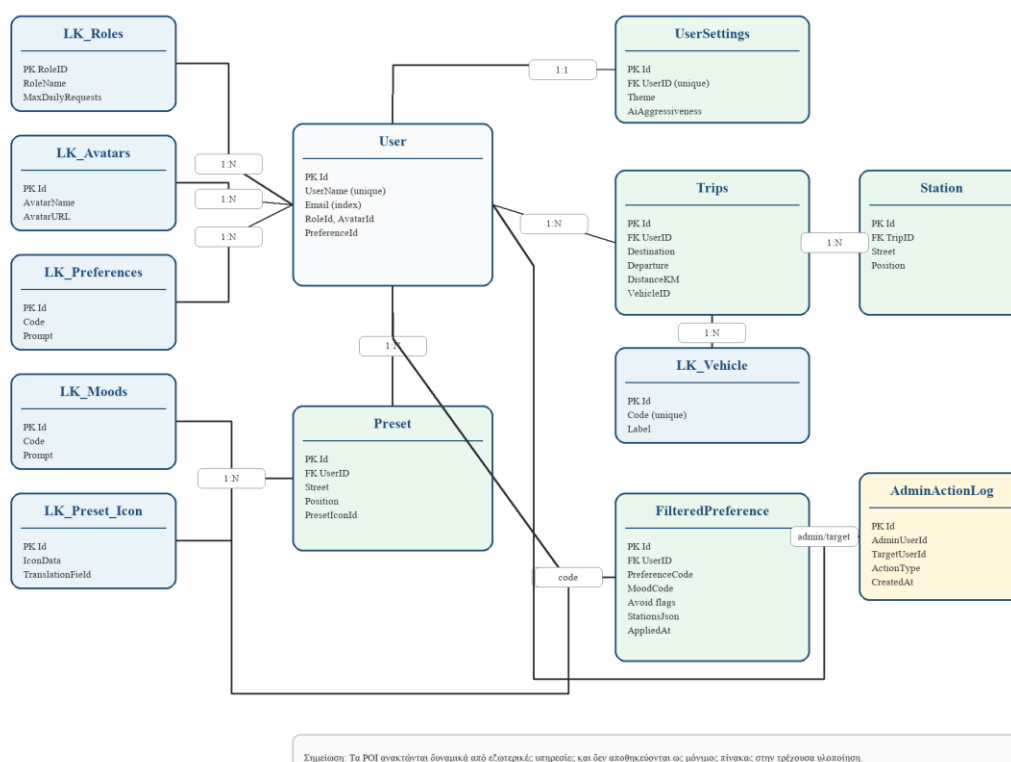
		διαγράφει τις ρυθμίσεις.
--	--	--------------------------

6.7 ER διάγραμμα

Το ER διάγραμμα του Κεφαλαίου 6 αποτυπώνει τις κύριες οντότητες που υλοποιούνται στο backend και τις μεταξύ τους σχέσεις. Δεν προσπαθεί να δείξει κάθε επιμέρους πεδίο, γιατί τότε το διάγραμμα θα γινόταν δυσανάγνωστο. Αντίθετα, εστιάζει στις οντότητες που επηρεάζουν τη λειτουργία της εφαρμογής: χρήστες, ρόλους, προφίλ, διαδρομές, στάσεις, οχήματα, προτιμήσεις, presets, ρυθμίσεις και διαχειριστικές ενέργειες.

Στο διάγραμμα φαίνεται επίσης η διάκριση ανάμεσα σε κανονικές εγγραφές χρήσης, όπως trips και stations, και σε lookup οντότητες, όπως roles, vehicles, preferences, moods και preset icons. Αυτή η διάκριση είναι χρήσιμη, επειδή δείχνει ποια δεδομένα δημιουργούνται από τη χρήση της εφαρμογής και ποια δεδομένα λειτουργούν ως ελεγχόμενες λίστες επιλογών.

SmartNavix - ER διάγραμμα βάσης δεδομένων



ΔΙΑΓΡΑΜΜΑ 6.1: ER διάγραμμα βάσης δεδομένων SmartNavix

6.8 Κανόνες ακεραιότητας και απόδοσης

Η ακεραιότητα της βάσης υποστηρίζεται με πρωτεύοντα κλειδιά, ξένα κλειδιά, μοναδικά indexes, περιορισμούς μήκους και καθορισμένες συμπεριφορές διαγραφής. Το μοναδικό username μειώνει τον κίνδυνο διπλών λογαριασμών, ενώ τα indexes σε email, UserID-AppliedAt, UserID-Position και admin logs υποστηρίζουν συχνές αναζητήσεις. Τα πεδία μήκους, όπως Username, Email, Vehicle Code και ActionType, περιορίζουν την ανεξέλεγκτη εισαγωγή δεδομένων.

Για την απόδοση, η βάση κρατά τα δεδομένα που χρειάζονται συχνά σε κανονικοποιημένους πίνακες και αφήνει δυναμικά δεδομένα, όπως προτεινόμενα σημεία ενδιαφέροντος από εξωτερικές υπηρεσίες, να ανακτώνται κατά τη χρήση. Αυτή η επιλογή μειώνει την ανάγκη συγχρονισμού με τρίτους παρόχους και περιορίζει την αποθήκευση πληροφορίας που μπορεί να αλλάζει γρήγορα.

Σε επίπεδο ασφάλειας, το σχήμα πρέπει να συνοδεύεται από σωστή διαχείριση πρόσβασης, επικύρωση εισόδων και προστασία ευαίσθητων δεδομένων. Η ύπαρξη πινάκων όπως User και AdminActionLog κάνει απαραίτητη την προσεκτική διαχείριση δικαιωμάτων, ενώ οι γενικές οδηγίες ασφαλούς ανάπτυξης web εφαρμογών, όπως αυτές του OWASP, παραμένουν κρίσιμες για την τελική λειτουργία του συστήματος [76].

Κεφάλαιο 7: Υλοποίηση Βασικών Λειτουργιών

Το κεφάλαιο αυτό παρουσιάζει τον τρόπο με τον οποίο οι βασικές λειτουργίες της SmartNavix υλοποιούνται στο frontend, στο backend και στις εξωτερικές υπηρεσίες που αξιοποιεί η εφαρμογή. Η έμφαση δίνεται στη ροή εκτέλεσης και όχι σε πλήρη παράθεση κώδικα, ώστε να γίνει σαφές πώς μετατρέπονται οι απαιτήσεις και ο σχεδιασμός των προηγούμενων κεφαλαίων σε λειτουργικό σύστημα.

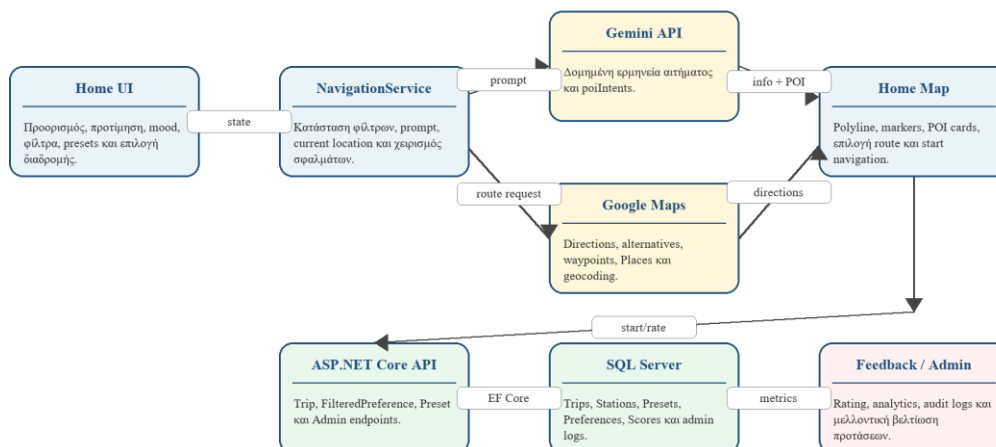
7.1 Αναζήτηση διαδρομής

Η βασική λειτουργία αναζήτησης διαδρομής υλοποιείται στο frontend μέσα από την οθόνη Home και το NavigationService. Ο χρήστης εισάγει προορισμό, ενώ η εφαρμογή συνδυάζει το κείμενο αναζήτησης με την τρέχουσα θέση, την επιλεγμένη προτίμηση, πιθανή διάθεση χρήστη και τα ενεργά φίλτρα. Πριν ξεκινήσει η αναζήτηση γίνεται έλεγχος ότι υπάρχει κείμενο προορισμού, ώστε να μην αποστέλλεται κενό αίτημα προς τις υπηρεσίες πλοήγησης.

Το NavigationService δημιουργεί prompt προς το Gemini API, ζητώντας δομημένη απάντηση με origin, destination, επεξήγηση και πιθανές προθέσεις για σημεία ενδιαφέροντος. Η επιλογή αυτή επιτρέπει στην εφαρμογή να μετατρέψει ένα ελεύθερο αίτημα του χρήστη σε δεδομένα που μπορεί να χρησιμοποιήσει η υπηρεσία χαρτών. Στη συνέχεια, το αποτέλεσμα περνά στη Google Maps Directions υπηρεσία, η οποία επιστρέφει το DirectionsResult με τις διαδρομές που μπορούν να εμφανιστούν στον χάρτη [64], [65], [66].

Η αναζήτηση δεν αποθηκεύει αμέσως trip στη βάση. Πρώτα εμφανίζει τις διαθέσιμες προτάσεις, ώστε ο χρήστης να δει τη διαδρομή, τη διάρκεια και την απόσταση. Η μόνιμη αποθήκευση πραγματοποιείται όταν ο χρήστης ξεκινήσει την πλοήγηση, επειδή τότε η διαδρομή θεωρείται πραγματική επιλογή και όχι απλή δοκιμή.

SmartNavix - ροή υλοποίησης βασικών λειτουργιών



ΔΙΑΓΡΑΜΜΑ 7.1: Ροή υλοποίησης βασικών λειτουργιών SmartNavix

7.2 Εναλλακτικές διαδρομές

Η εφαρμογή ζητά από τη Google Maps Directions υπηρεσία να επιστρέψει εναλλακτικές διαδρομές με την επιλογή provideRouteAlternatives. Στο frontend, η μέθοδος επιλογής διαδρομής περιορίζει τον δείκτη της διαδρομής σε έγκυρο εύρος και δημιουργεί νέο DirectionsResult που περιέχει μόνο την

επιλεγμένη πρόταση. Με αυτόν τον τρόπο ο χάρτης μπορεί να εστιάζει καθαρά στη διαδρομή που έχει διαλέξει ο χρήστης.

Για κάθε διαδρομή υπολογίζονται η συνολική απόσταση και η συνολική διάρκεια από τα legs του DirectionsResult. Οι τιμές αυτές εμφανίζονται στη διεπαφή και ενημερώνονται κάθε φορά που αλλάζει η επιλεγμένη εναλλακτική. Η υλοποίηση αυτή είναι πρακτική, επειδή ο χρήστης δεν βλέπει απλώς μια γραμμή στον χάρτη, αλλά μπορεί να συγκρίνει προτάσεις με βάση μετρήσιμα στοιχεία.

Η επιλεγμένη διαδρομή επηρεάζει και τα υπόλοιπα υποσυστήματα της οθόνης. Όταν αλλάζει route choice, ανανεώνονται οι markers σχετικών χαρακτηριστικών, όπως διόδια, αυτοκινητοδρόμοι, ferry ή EV σημεία, και επανυπολογίζονται τα σημεία ενδιαφέροντος που σχετίζονται με τη συγκεκριμένη πορεία. Έτσι, η διεπαφή παραμένει δεμένη με την τρέχουσα επιλογή του χρήστη και όχι με την πρώτη διαθέσιμη διαδρομή.

7.3 Φίλτρα και επιλογές προτιμήσεων

Τα φίλτρα της SmartNavix υλοποιούνται ως κατάσταση στο NavigationService. Η εφαρμογή κρατά ενδιάμεσες στάσεις, μέγεθος οχήματος, αποφυγές διοδίων, αυτοκινητοδρόμων και ferry, χρονικό πλαίσιο κυκλοφορίας και επιλογή για σταθμούς φόρτισης ηλεκτρικών οχημάτων. Τα δεδομένα αυτά κανονικοποιούνται πριν χρησιμοποιηθούν, ώστε να μη μεταφέρονται κενές ή μη έγκυρες τιμές στα επόμενα βήματα.

Η επιλογή προτίμησης, όπως γρήγορη ή πιο προσαρμοσμένη διαδρομή, συνδέεται με prompt από lookup δεδομένα του backend. Αν ο χρήστης έχει επιλέξει και mood, το prompt της διάθεσης συνδυάζεται με το prompt της βασικής προτίμησης. Έτσι, η αναζήτηση δεν εξαρτάται μόνο από το τελικό κείμενο του προορισμού, αλλά και από το πλαίσιο που έχει διαμορφώσει ο χρήστης μέσα στην εφαρμογή.

Για τη μόνιμη αποθήκευση των φίλτρων χρησιμοποιείται ο FilteredPreferenceController. Το backend δέχεται τα πεδία φίλτρων, ελέγχει ότι υπάρχει έγκυρος χρήστης, κανονικοποιεί τις τιμές και αποθηκεύει την εγγραφή με χρόνο εφαρμογής. Η ύπαρξη endpoint για την τελευταία προτίμηση χρήστη επιτρέπει στην εφαρμογή να επαναφέρει ή να προτείνει προηγούμενες επιλογές, αντί να ξεκινά πάντα από κενή κατάσταση [80], [81].

7.4 Προτάσεις σημείων ενδιαφέροντος

Οι προτάσεις σημείων ενδιαφέροντος δεν αντιμετωπίζονται ως στατικός πίνακας της βάσης. Στην παρούσα υλοποίηση προκύπτουν δυναμικά από το αποτέλεσμα του Gemini και από τις υπηρεσίες Places/Maps. Το Gemini επιστρέφει poiIntents, δηλαδή κατηγορίες και λέξεις αναζήτησης που σχετίζονται με το αίτημα ή τα φίλτρα του χρήστη. Στη συνέχεια, το frontend χρησιμοποιεί τη διαδρομή στον χάρτη για να αναζητήσει κατάλληλα σημεία κοντά στη διαδρομή.

Το NavigationService καθαρίζει τα poiIntents πριν τα χρησιμοποιήσει. Απορρίπτει γενικούς όρους δρόμων, λεωφόρων ή αυτοκινητοδρόμων, επειδή αυτά δεν είναι πραγματικά POI. Αυτός ο έλεγχος είναι σημαντικός, γιατί η τεχνητή νοημοσύνη μπορεί να επιστρέφει λέξεις που ακούγονται σχετικές με την πλοήγηση αλλά δεν αντιστοιχούν σε επισκέψιμο σημείο.

Στη διεπαφή, τα POI εμφανίζονται με markers και κάρτες πληροφορίας. Όταν αλλάζει η επιλεγμένη διαδρομή ή εκκινείται νέα αναζήτηση, οι παλιές προτάσεις καθαρίζονται και δημιουργούνται νέες. Έτσι αποφεύγεται η σύγχυση ανάμεσα σε POI που ανήκαν σε προηγούμενο αίτημα και σε αυτά που αφορούν την τρέχουσα πρόταση διαδρομής.

7.5 Presets και custom icons

Τα presets υλοποιούνται ως γρήγορη αποθήκευση συχνών διευθύνσεων. Ο χρήστης ανοίγει το σχετικό modal, εισάγει στοιχεία διεύθυνσης και επιλέγει εικονίδιο από τον πίνακα διαθέσιμων preset icons. Η φόρμα ελέγχει ότι υπάρχει έγκυρος χρήστης, έγκυρο εικονίδιο και τουλάχιστον ένα πεδίο διεύθυνσης πριν στείλει το αίτημα στο backend.

Στο backend, ο `PresetController` ελέγχει την ύπαρξη χρήστη και εικονιδίου, υπολογίζει την επόμενη θέση εμφάνισης και αποθηκεύει το preset. Η ανάκτηση γίνεται ταξινομημένα με βάση το `Position` και το `Id`, ώστε η διεπαφή να διατηρεί σταθερή σειρά. Η χρήση lookup πίνακα για τα εικονίδια επιτρέπει κοινά εικονίδια για πολλούς χρήστες χωρίς αντιγραφή του ίδιου SVG σε κάθε λογική επιλογή.

Μετά την αποθήκευση, το Home component φορτώνει ξανά τα presets του χρήστη και επιχειρεί να τα μετατρέψει σε markers πάνω στον χάρτη. Για τον εντοπισμό θέσης χρησιμοποιείται geocoding και, όταν χρειάζεται, fallback με Places search. Με αυτόν τον τρόπο το preset δεν μένει απλώς ως κείμενο στη βάση, αλλά γίνεται ορατό αντικείμενο στην εμπειρία πλοήγησης [64].

7.6 Αξιολόγηση διαδρομής από τον χρήστη

Η αξιολόγηση διαδρομής ενεργοποιείται αφού ο χρήστης ξεκινήσει πλοήγηση και έχει δημιουργηθεί ή ανακτηθεί ενεργό trip. Το frontend κρατά το `activeTripId`, την τιμή αξιολόγησης, την κατάσταση αποθήκευσης και το αν η αξιολόγηση έχει ήδη υποβληθεί. Η επιλογή γίνεται με αστέρια από 1 έως 5, ώστε ο χρήστης να δίνει γρήγορη ποσοτική ανατροφοδότηση.

Το endpoint `Trip/Rate` δέχεται `UserId`, προαιρετικό `TripId` και `Score`. Αν υπάρχει `TripId`, αναζητά τη συγκεκριμένη διαδρομή του χρήστη. Αν δεν υπάρχει, χρησιμοποιεί την πιο πρόσφατη διαδρομή του χρήστη ως fallback. Το backend απορρίπτει τιμές εκτός του εύρους 1-5 και επιστρέφει σφάλμα όταν δεν υπάρχει διαδρομή. Έτσι, η αξιολόγηση παραμένει συνδεδεμένη με πραγματική εγγραφή και δεν αποθηκεύεται ως ανεξάρτητο, ασαφές γεγονός.

Η αποθήκευση της αξιολόγησης στο πεδίο `Score` δημιουργεί βάση για μελλοντική αξιοποίηση. Στην τρέχουσα φάση χρησιμοποιείται κυρίως για προβολή και admin analytics. Σε επόμενο στάδιο μπορεί να συνδεθεί πιο άμεσα με μηχανισμό βελτίωσης προτάσεων, ώστε η εφαρμογή να προσαρμόζεται καλύτερα στις επιλογές που ο χρήστης θεωρεί επιτυχημένες.

7.7 Feedback loop και βελτίωση προτάσεων

Το feedback loop της SmartNavix δεν βασίζεται σε ένα μόνο σήμα. Συνδυάζει ιστορικό `filtered preferences`, επιλεγμένες διαδρομές, στάσεις, όχημα και αξιολογήσεις. Η λογική αυτή είναι συνεπής με τον χαρακτήρα της εφαρμογής: ο χρήστης δεν εκφράζει την προτίμησή του μόνο όταν πατά μια βαθμολογία, αλλά και μέσα από τις επιλογές που επαναλαμβάνει.

Στον `FilteredPreferenceController` υπάρχει μηχανισμός επιλογής καλύτερης πρόσφατης προτίμησης. Η υλοποίηση υπολογίζει συχνότητα, πρόσφατη χρήση, πληρότητα φίλτρων και συνέπεια, και στη συνέχεια τα συνδυάζει με fuzzy λογική και παραμετροποιήσιμα βάρη από το configuration. Η λύση αυτή δεν παρουσιάζεται ως πλήρες σύστημα μηχανικής μάθησης, αλλά ως ελεγχόμενος κανόνας προσαρμογής που μπορεί να εξηγηθεί και να τροποποιηθεί.

Η πρακτική αξία του feedback loop είναι ότι η εφαρμογή μπορεί να επανέρχεται σε επιλογές που μοιάζουν αντιπροσωπευτικές για τον χρήστη. Αν για παράδειγμα ένας χρήστης επιλέγει συχνά συγκεκριμένο όχημα, αποφυγές ή στάσεις, η εφαρμογή μπορεί να χρησιμοποιήσει αυτά τα δεδομένα ως αφετηρία επόμενης πρότασης. Η τελική βελτίωση των προτάσεων εξαρτάται από το πόσο σταθερά και επαρκή είναι τα δεδομένα χρήσης.

7.8 Admin panel και analytics

Το admin panel υλοποιείται ως ξεχωριστή οθόνη Angular και ως σύνολο endpoints στον `AdminController`. Πριν εκτελεστεί κάθε διαχειριστική λειτουργία, το backend καλεί `EnsureAdminAsync`, ελέγχει ότι υπάρχει χρήστης και ότι ο ρόλος του περιλαμβάνει admin ή αντίστοιχη διαχειριστική ένδειξη. Αν ο έλεγχος αποτύχει, η λειτουργία επιστρέφει σφάλμα δικαιωμάτων.

Οι βασικές λειτουργίες διαχείρισης περιλαμβάνουν προβολή χρηστών, αναζήτηση, αλλαγή ρόλου, αλλαγή κατάστασης επαλήθευσης, μαζική εφαρμογή αλλαγών και διαγραφή λογαριασμού. Για ευαίσθητες ενέργειες, όπως αλλαγή ρόλου ή διαγραφή, το backend αποτρέπει ενέργειες του admin πάνω στον ίδιο τον λογαριασμό. Παράλληλα, επιχειρεί να γράψει admin action log, ώστε να υπάρχει ίχνος διαχειριστικής ενέργειας.

Το κομμάτι analytics συγκεντρώνει στοιχεία από trips, vehicles και stations. Το dashboard εμφανίζει συνολικούς χρήστες, επαληθευμένους χρήστες, πλήθος trips, presets, filtered preferences, trips τελευταίων επτά ημερών και μέσο score. Η οθόνη admin δημιουργεί γραφήματα χρήσης οχημάτων, συνολικής απόστασης ανά όχημα και πλήθους στάσεων ανά trip. Έτσι, ο διαχειριστής μπορεί να δει όχι μόνο ποιοι χρήστες υπάρχουν, αλλά και πώς χρησιμοποιείται η εφαρμογή.

7.9 Διαχείριση σφαλμάτων και περιπτώσεων αποτυχίας

Η εφαρμογή χειρίζεται σφάλματα σε δύο επίπεδα: στο frontend και στο backend. Στο frontend, το NavigationService μετατρέπει αποτυχίες του Gemini, του Directions API ή της γεωεντοπιστικής άδειας σε μηνύματα που μπορούν να εμφανιστούν στον χρήστη. Για παράδειγμα, αν δεν δοθεί πρόσβαση τοποθεσίας, ενημερώνεται το errorMessage και η εφαρμογή χρησιμοποιεί fallback όπου είναι δυνατό.

Στο backend, τα endpoints επιστρέφουν συγκεκριμένα BadRequest, NotFound ή 403/500 ανάλογα με την περίπτωση. Το Trip/Create ελέγχει UserID και έγκυρο όχημα, αποφεύγει διπλές εγγραφές με ανοχή απόστασης και χρονικό παράθυρο, και δημιουργεί στάσεις μόνο όταν υπάρχουν χρήσιμα δεδομένα. Το Trip/Rate απορρίπτει score εκτός ορίων, ενώ το Preset/Create απαιτεί χρήστη, εικονίδιο και τουλάχιστον ένα πεδίο διεύθυνσης.

Η γενική υπηρεσία DataService έχει κοινό χειρισμό HTTP σφαλμάτων. Σε περίπτωση 403 αποσυνδέει τον χρήστη και φορτώνει ξανά την εφαρμογή, ενώ σε περίπτωση 405 εμφανίζει μήνυμα ανεπαρκών δικαιωμάτων. Αυτή η κεντροποίηση δεν λύνει όλα τα πιθανά προβλήματα ασφαλείας, αλλά δίνει ενιαία συμπεριφορά στα βασικά σφάλματα πρόσβασης. Για παραγωγική χρήση, η ίδια λογική πρέπει να συνοδεύεται από αυστηρό έλεγχο εισόδων και προστασία API keys σύμφωνα με πρακτικές ασφαλούς web ανάπτυξης [76].

ΠΙΝΑΚΑΣ 7.1: Υλοποίηση βασικών λειτουργιών SmartNavix

Λειτουργία	Frontend υλοποίηση	Backend / υπηρεσία
Αναζήτηση διαδρομής	Home.findPath και NavigationService.getSmartRoute.	Gemini API για δομημένη ερμηνεία και Google Directions για διαδρομές.
Εναλλακτικές διαδρομές	selectRouteChoice, routeChoiceOptions και selectedRouteDirections.	Google DirectionsResult με provideRouteAlternatives.
Φίλτρα προτιμήσεων	NavigationService signals για στάσεις, όχημα, αποφυγές, traffic και EV.	FilteredPreferenceController για αποθήκευση και ανάκτηση ιστορικού.
POI προτάσεις	poiIntents, poiMarkers και poiPlaces στην οθόνη Home.	Google Places/Maps σε runtime, χωρίς μόνιμο POI table.
Preset	Modal γρήγορων presets, autocomplete και markers στον χάρτη.	PresetController και LK_Preset_Icon.
Αξιολόγηση διαδρομής	routeRatingValue, routeRatingPanelVisible και submitRouteRating.	TripController.Rate αποθηκεύει score 1-5.
Admin analytics	Admin component με πίνακες και γραφήματα.	AdminController.Dashboard, Users, AuditLogs και AnalyticsByUser.
Σφάλματα	errorMessage\$, toast messages και κοινό DataService.handleError.	BadRequest, NotFound, 403 και transaction rollback όπου χρειάζεται.

Κεφάλαιο 8: Μοντέλο Fuzzy Logic για Επιλογή Προτιμήσεων

Το κεφάλαιο αυτό παρουσιάζει το μοντέλο fuzzy logic που χρησιμοποιείται στη SmartNavix για την επιλογή της καταλληλότερης αποθηκευμένης προτίμησης χρήστη. Η ανάλυση βασίζεται στην υλοποίηση του FilteredPreferenceController και εξηγεί τις εισόδους, τις membership functions, τους κανόνες, το defuzzification και την παραμετροποίηση του resolver.

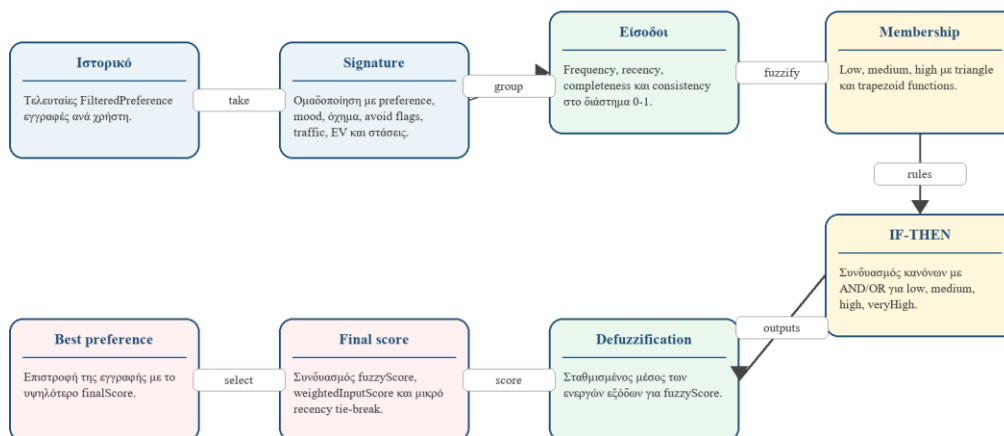
8.1 Περιγραφή του προβλήματος επιλογής best preference

Η SmartNavix δεν χρειάζεται μόνο να αποθηκεύει τις επιλογές που κάνει ο χρήστης. Χρειάζεται επίσης να μπορεί να αναγνωρίζει ποια από αυτές τις επιλογές είναι πιο αντιπροσωπευτική όταν ο χρήστης επιστρέφει στην εφαρμογή ή όταν πρέπει να προταθεί μια αρχική ρύθμιση. Το πρόβλημα αυτό δεν είναι απόλυτα δυαδικό. Μια προτίμηση δεν είναι απλώς σωστή ή λάθος, αλλά μπορεί να είναι περισσότερο ή λιγότερο κατάλληλη με βάση το ιστορικό, την πρόσφατη χρήση, την πληρότητα και τη συνέπεια.

Για αυτόν τον λόγο χρησιμοποιείται fuzzy προσέγγιση. Η fuzzy λογική είναι κατάλληλη για περιπτώσεις όπου οι κατηγορίες δεν έχουν αυστηρά όρια, αλλά εκφράζουν βαθμούς συμμετοχής σε έννοιες όπως χαμηλή, μέτρια ή υψηλή καταλληλότητα [56]. Στη SmartNavix, οι έννοιες αυτές μεταφέρονται στο πρόβλημα επιλογής προτίμησης: μια επιλογή μπορεί να είναι αρκετά πρόσφατη, αρκετά συχνή ή αρκετά πλήρης, χωρίς να χρειάζεται να περάσει ένα σκληρό κατώφλι.

Το μοντέλο υλοποιείται στον FilteredPreferenceController, στη μέθοδο ResolveBestPreference. Η μέθοδος παίρνει ένα περιορισμένο ιστορικό filtered preferences του χρήστη, υπολογίζει επιμέρους δείκτες, εφαρμόζει fuzzy inference και επιστρέφει την εγγραφή με το υψηλότερο τελικό score. Η επιλογή αυτή λειτουργεί ως πρακτικός μηχανισμός προσαρμογής της εφαρμογής, χωρίς να απαιτείται εκπαίδευση μοντέλου μηχανικής μάθησης.

SmartNavix - fuzzy resolver best preference



ΔΙΑΓΡΑΜΜΑ 8.1: Ποή fuzzy resolver για best preference

8.2 Είσοδοι του μοντέλου

Οι εισοδοι του fuzzy resolver προέρχονται από τις εγγραφές FilteredPreference. Κάθε εγγραφή περιλαμβάνει κωδικό προτίμησης, prompt, mood, μέγεθος οχήματος, αποφυγές, χρονικές επιλογές κυκλοφορίας, επιλογή για EV charging και προαιρετικές στάσεις. Αυτά τα στοιχεία δεν

χρησιμοποιούνται αυτούσια ως τελικό αποτέλεσμα, αλλά μετατρέπονται σε δείκτες που μπορούν να συγκριθούν μεταξύ διαφορετικών εγγραφών.

Το μοντέλο εξετάζει τέσσερις βασικές εισόδους: frequency, recency, completeness και consistency. Η frequency δείχνει πόσο συχνά εμφανίζεται μια ίδια υπογραφή προτίμησης στο ιστορικό. Η recency δείχνει πόσο πρόσφατη είναι η εγγραφή. Η completeness δείχνει πόσο πλούσια είναι η επιλογή σε φίλτρα και στάσεις. Η consistency δείχνει πόσο σταθερά επαναλαμβάνεται η ίδια υπογραφή μέσα στο διαθέσιμο ιστορικό.

Ο Πίνακας 8.1 συνοψίζει τις εισόδους του μοντέλου. Η επιλογή αυτών των τεσσάρων δεικτών έγινε επειδή συνδέουν τη συμπεριφορά χρήστη με μετρήσιμα στοιχεία. Έτσι, η εφαρμογή δεν στηρίζεται μόνο στην τελευταία επιλογή ούτε μόνο στην πιο συχνή, αλλά προσπαθεί να ισορροπήσει πρόσφατη πρόθεση και επαναλαμβανόμενη συνήθεια.

ΠΙΝΑΚΑΣ 8.1: Είσοδοι του fuzzy μοντέλου επιλογής best preference

Είσοδος	Τρόπος υπολογισμού	Ρόλος στην επιλογή
Frequency	Πλήθος εγγραφών με ίδια preference signature σε σχέση με το μέγιστο πλήθος ομάδας.	Ενισχύει επιλογές που επαναλαμβάνονται συχνά.
Recency	Συνάρτηση με βάση τις ημέρες από το AppliedAt και το RecencyHalfLifeDays.	Δίνει μεγαλύτερη σημασία σε πρόσφατες επιλογές.
Completeness	Συνδυασμός ύπαρξης στάσεων και προηγμένων φίλτρων, όπως mood, όχημα, avoid flags, traffic και EV.	Ενισχύει πιο πλούσιες και συγκεκριμένες επιλογές.
Consistency	Πλήθος ίδιας υπογραφής σε σχέση με το συνολικό ιστορικό που εξετάζεται.	Εκφράζει σταθερότητα συμπεριφοράς χρήστη.

8.3 Frequency, recency, consistency και completeness

Η frequency υπολογίζεται με βάση το πλήθος των εγγραφών που έχουν την ίδια υπογραφή προτίμησης. Η υπογραφή δημιουργείται από τα βασικά χαρακτηριστικά της επιλογής: selected preference, mood, όχημα, αποφυγές, traffic mode, ώρες κυκλοφορίας, EV επιλογή και fingerprint στάσεων. Με αυτόν τον τρόπο δύο εγγραφές θεωρούνται ίδιες όταν αντιπροσωπεύουν ουσιαστικά την ίδια λογική επιλογής.

Η recency υπολογίζεται από την απόσταση της εγγραφής από την τρέχουσα χρονική στιγμή. Η υλοποίηση χρησιμοποιεί συνάρτηση με half-life ημερών, ώστε μια πρόσφατη επιλογή να παίρνει υψηλότερη τιμή αλλά μια παλαιότερη επιλογή να μην μηδενίζεται απότομα. Αυτό είναι σημαντικό, γιατί οι προτιμήσεις χρήστη μπορεί να έχουν διάρκεια, αλλά η τελευταία συμπεριφορά εξακολουθεί να έχει ιδιαίτερη σημασία.

Η completeness συνδέεται με το αν ο χρήστης έχει δώσει πλούσιο σύνολο φίλτρων. Η ύπαρξη στάσεων συνεισφέρει με ξεχωριστό βάρος, ενώ τα υπόλοιπα advanced filters, όπως mood, όχημα, αποφυγές, traffic και EV επιλογή, συνθέτουν το υπόλοιπο μέρος του score. Η consistency, τέλος, εκφράζει πόσο μεγάλο μέρος του ιστορικού αντιστοιχεί στην ίδια υπογραφή. Οι τέσσερις δείκτες κανονικοποιούνται στο διάστημα 0-1 ώστε να μπορούν να εισαχθούν στις membership functions.

8.4 Membership functions

Οι membership functions μετατρέπουν κάθε κανονικοποιημένο δείκτη σε βαθμούς συμμετοχής σε γλωσσικές κατηγορίες. Για παράδειγμα, η frequency μπορεί να συμμετέχει με κάποιον βαθμό στην κατηγορία low, με άλλον βαθμό στην κατηγορία medium και με άλλον στην κατηγορία high. Αυτή η λογική συνδέεται με την έννοια της linguistic variable, όπου ποσοτικές τιμές εκφράζονται με γλωσσικούς όρους [57].

Στην υλοποίηση χρησιμοποιούνται τριγωνικές και τραπεζοειδείς συναρτήσεις. Η Triangle χρησιμοποιείται κυρίως για μεσαίες περιοχές, ενώ η Trapezoid χρησιμοποιείται για χαμηλές και υψηλές περιοχές όπου το score μπορεί να παραμένει πλήρως ενεργό σε ένα εύρος τιμών. Για παράδειγμα, η frequencyHigh γίνεται ισχυρή όταν η frequency βρίσκεται σε υψηλό εύρος, ενώ η frequencyLow ενεργοποιείται για χαμηλές επαναλήψεις.

Η επιλογή αυτών των απλών membership functions κάνει το μοντέλο διαφανές. Δεν χρειάζεται να εξηγηθεί με περίπλοκους συντελεστές ή κρυφές παραμέτρους, αλλά μπορεί να παρουσιαστεί με όρια και περιοχές τιμών. Ο Πίνακας 8.2 δείχνει συνοπτικά τις κατηγορίες που χρησιμοποιούνται στον resolver.

ΠΙΝΑΚΑΣ 8.2: Membership categories και υλοποίηση

Μεταβλητή	Membership categories	Υλοποίηση
Frequency	low, medium, high	Trapezoid για low/high και Triangle για medium.
Recency	low, medium, high	Trapezoid για χαμηλή/υψηλή πρόσφατη χρήση και Triangle για ενδιάμεση.
Consistency	low, medium, high	Αντίστοιχη λογική με frequency, αλλά πάνω στη σταθερότητα υπογραφής.
Completeness	medium, high	Triangle για medium και Trapezoid για high, επειδή η πληρότητα λειτουργεί κυρίως ως ενισχυτικός παράγοντας.
Έξοδος	low, medium, high, veryHigh	Κέντρα 0.30, 0.56, 0.78 και 0.92 για defuzzification.

8.5 Κανόνες IF-THEN

Μετά τον υπολογισμό των membership values, ο resolver εφαρμόζει κανόνες IF-THEN. Οι κανόνες αυτοί δεν είναι αποθηκευμένοι ως ξεχωριστό αρχείο, αλλά υλοποιούνται στον κώδικα με συνδυασμούς Math.Min και Math.Max. Η χρήση Math.Min λειτουργεί ως λογικό AND ανάμεσα σε δύο συνθήκες, ενώ το Math.Max λειτουργεί ως OR ανάμεσα σε εναλλακτικούς κανόνες.

Οι βασικές κατηγορίες εξόδου είναι low, medium, high και veryHigh. Για παράδειγμα, αν η frequency είναι high και η recency είναι high, η προτίμηση ενισχύεται προς veryHigh. Αν η frequency είναι medium αλλά η recency είναι high, η έξοδος μπορεί να ανέβει προς high. Αντίθετα, όταν frequency και recency είναι χαμηλές, το αποτέλεσμα τείνει προς low. Η προσέγγιση αυτή μοιάζει περισσότερο με ανθρώπινο συλλογισμό παρά με αυστηρό if-else κατώφλι.

Οι κανόνες δεν προσπαθούν να αντικαταστήσουν την κρίση του χρήστη. Λειτουργούν ως μηχανισμός επιλογής αρχικής προτίμησης από το ιστορικό. Αν ο χρήστης αλλάξει φίλτρα, το νέο αποτέλεσμα αποθηκεύεται και μπορεί αργότερα να επηρεάσει τον resolver. Έτσι, το σύστημα διατηρεί έναν μικρό κύκλο ανατροφοδότησης, αλλά παραμένει ελεγχόμενο και αναγνώσιμο.

8.6 Defuzzification και τελικό score

Το defuzzification μετατρέπει τα fuzzy αποτελέσματα σε έναν αριθμό που μπορεί να συγκριθεί ανάμεσα σε διαφορετικές εγγραφές. Στη SmartNavix, κάθε κατηγορία εξόδου έχει κέντρο: low, medium, high και veryHigh αντιστοιχούν σε αριθμητικές τιμές. Ο resolver υπολογίζει σταθμισμένο μέσο όρο των ενεργών κατηγοριών και επιστρέφει fuzzyScore στο διάστημα 0-1.

Το fuzzyScore δεν χρησιμοποιείται μόνο του. Συνδυάζεται με ένα weightedInputScore, το οποίο προκύπτει απευθείας από frequency, recency, completeness και consistency με βάρη από το configuration. Η τελική τιμή υπολογίζεται ως συνδυασμός fuzzyScore και weightedInputScore, με το FuzzyInferenceWeight να καθορίζει πόσο ισχυρά επηρεάζει η fuzzy inference την τελική επιλογή.

Στο τέλος προστίθεται μικρό tie-break bonus από τη recency. Αυτό βοηθά όταν δύο εγγραφές έχουν σχεδόν ίδιο score: προτιμάται η πιο πρόσφατη, χωρίς όμως να ακυρώνεται ο συνολικός υπολογισμός. Η εγγραφή με το μεγαλύτερο finalScore γίνεται η best preference που επιστρέφεται στο frontend.

8.7 Παραμετροποίηση μέσω appsettings

Οι βασικές παράμετροι του resolver δεν είναι σκληρά κλειδωμένες μόνο μέσα στον αλγόριθμο. Διαβάζονται από την ενότητα FilteredPreferenceResolution του appsettings. Αυτό επιτρέπει αλλαγή της

συμπεριφοράς χωρίς αλλαγή του κύριου κώδικα, κάτι που είναι χρήσιμο στη φάση δοκιμών της διπλωματικής αλλά και σε πιθανή μελλοντική εξέλιξη.

Η παράμετρος HistoryLimit καθορίζει πόσες πρόσφατες εγγραφές εξετάζονται. Το RecencyHalfLifeDays ρυθμίζει πόσο γρήγορα μειώνεται η σημασία παλαιότερων επιλογών. Τα FrequencyWeight, RecencyWeight, CompletenessWeight και ConsistencyWeight καθορίζουν τη βαρύτητα των εισόδων στον weightedInputScore. Το FuzzyInferenceWeight καθορίζει πόσο επηρεάζει το fuzzy κομμάτι σε σχέση με τον απλό σταθμισμένο υπολογισμό.

Ο Πίνακας 8.3 παρουσιάζει τις παραμέτρους που χρησιμοποιούνται στην τρέχουσα υλοποίηση. Οι τιμές αυτές δεν πρέπει να θεωρηθούν οριστικές για παραγωγικό σύστημα. Αποτελούν ρεαλιστική αρχική παραμετροποίηση που μπορεί να αναπροσαρμοστεί όταν υπάρξουν περισσότερα δεδομένα χρήσης.

ΠΙΝΑΚΑΣ 8.3: Παράμετροι FilteredPreferenceResolution

Παράμετρος	Τρέχουσα τιμή	Σημασία
HistoryLimit	150	Πλήθος πρόσφατων filtered preferences που εξετάζονται.
RecencyHalfLifeDays	14	Ρυθμίζει τη μείωση βαρύτητας παλαιότερων επιλογών.
FrequencyWeight	0.35	Βάρος της συχνότητας στο weightedInputScore.
RecencyWeight	0.35	Βάρος της πρόσφατης χρήσης στο weightedInputScore.
CompletenessWeight	0.20	Βάρος της πληρότητας φίλτρων.
ConsistencyWeight	0.10	Βάρος της συνέπειας της ίδιας υπογραφής.
FuzzyInferenceWeight	0.70	Ποσοστό συμμετοχής του fuzzyScore στο finalScore.
RecentTieBreakerWeight	0.02	Μικρή ενίσχυση της πιο πρόσφατης επιλογής σε ισοβαθμίες.
StationPresenceWeight	0.20	Βάρος ύπαρξης στάσεων στο completeness.
AdvancedFilterPresenceWeight	0.80	Βάρος λοιπών προηγμένων φίλτρων στο completeness.

8.8 Σύγκριση fuzzy προσέγγισης με AI-only προσέγγιση

Μια εναλλακτική λύση θα ήταν η εφαρμογή να βασίζεται αποκλειστικά στο Gemini για να αποφασίζει ποια προτίμηση ταιριάζει καλύτερα στον χρήστη. Αυτή η λύση θα είχε το πλεονέκτημα της φυσικής γλώσσας και της ευελιξίας, αλλά θα δυσκόλευε την τεκμηρίωση, την επαναληψιμότητα και τον έλεγχο της απόφασης. Για μια διπλωματική εργασία, όπου πρέπει να εξηγείται καθαρά ο μηχανισμός, η αποκλειστικά AI προσέγγιση θα ήταν λιγότερο διαφανής.

Η fuzzy προσέγγιση έχει το πλεονέκτημα ότι είναι ερμηνεύσιμη. Μπορεί να περιγραφεί με εισόδους, membership functions, κανόνες και τελικό score. Παράλληλα, συνδέεται με τη θεωρία λήψης αποφάσεων σε περιβάλλον αβεβαιότητας, όπου οι εναλλακτικές λύσεις δεν αξιολογούνται πάντα με απόλυτες τιμές [58]. Στη SmartNavix αυτό ταιριάζει, γιατί η καλύτερη προτίμηση δεν είναι αντικειμενικό γεγονός αλλά εκτίμηση με βάση προηγούμενη συμπεριφορά.

Η επιλογή της SmartNavix δεν απορρίπτει την τεχνητή νοημοσύνη. Αντίθετα, χωρίζει τους ρόλους: το Gemini χρησιμοποιείται για ερμηνεία φυσικής γλώσσας και εμπλουτισμό του route request, ενώ το fuzzy μοντέλο χρησιμοποιείται για επαναλήψιμη επιλογή stored preference. Αυτός ο διαχωρισμός κάνει το σύστημα πιο ισορροπημένο, επειδή συνδυάζει ευελιξία AI με ελεγχόμενη λογική προσαρμογής.

Κεφάλαιο 9: Παραδείγματα Χρήσης με Screenshots

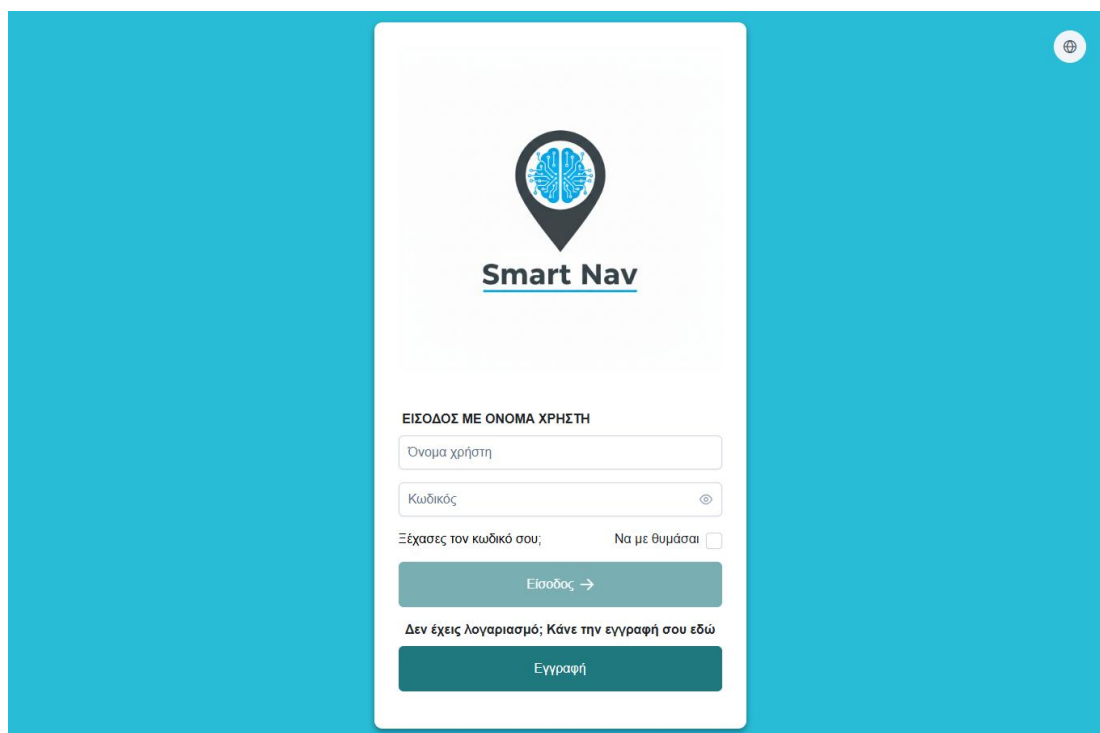
Το κεφάλαιο παρουσιάζει τη χρήση της SmartNavix μέσα από screenshots που δημιουργήθηκαν από την τοπική εκτέλεση της εφαρμογής. Τα παραδείγματα δείχνουν τι εισάγει ο χρήστης, πώς οργανώνεται η διεπαφή και ποια απόκριση εμφανίζει το σύστημα σε βασικά σημεία της ροής.

9.1 Είσοδος και αρχική οθόνη της εφαρμογής

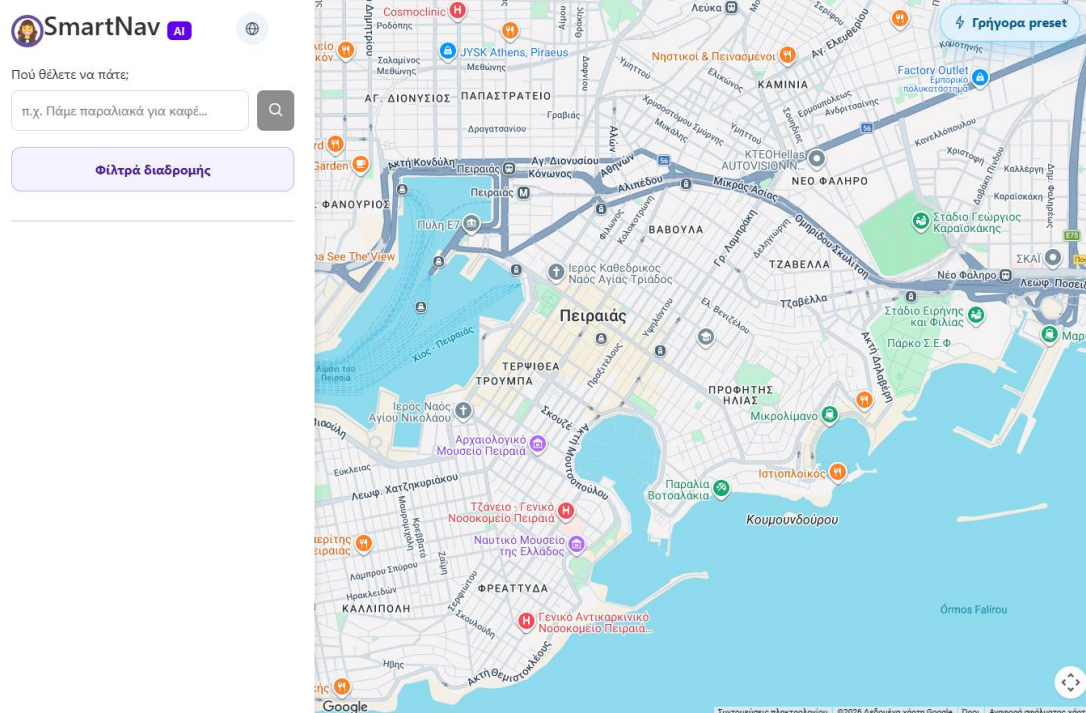
Το κεφάλαιο αυτό παρουσιάζει ενδεικτικά σενάρια χρήσης της SmartNavix μέσα από πραγματικά στιγμιότυπα της εφαρμογής. Ο στόχος δεν είναι να επαναληφθεί η τεχνική ανάλυση των προηγούμενων κεφαλαίων, αλλά να φανεί πώς ο χρήστης αλληλεπιδρά με το σύστημα και πώς η εφαρμογή αποκρίνεται σε βασικές ενέργειες. Η προσέγγιση αυτή συνδέεται με την έννοια της ευχρηστίας, όπου η αξιολόγηση ενός συστήματος εξετάζεται μέσα σε συγκεκριμένο πλαίσιο χρήσης και όχι μόνο ως τεχνική λειτουργία [31].

Η πρώτη οθόνη που συναντά ο χρήστης είναι η οθόνη εισόδου. Περιλαμβάνει πεδία για όνομα χρήστη και κωδικό πρόσβασης, επιλογή απομνημόνευσης, δυνατότητα ανάκτησης κωδικού και μετάβαση στην εγγραφή νέου χρήστη. Η διάταξη είναι απλή και συγκεντρώνει την προσοχή στην ταυτοποίηση, ώστε ο χρήστης να μην χρειάζεται να αναζητήσει την κύρια ενέργεια.

Μετά την επιτυχή είσοδο, εμφανίζεται η βασική οθόνη πλοήγησης. Στα αριστερά υπάρχει η περιοχή αναζήτησης και πρόσβασης στα φίλτρα, ενώ το μεγαλύτερο μέρος της οθόνης καλύπτεται από τον χάρτη. Η επιλογή αυτή είναι συμβατή με την πρακτική των εφαρμογών πλοήγησης, όπου ο χάρτης πρέπει να παραμένει το κύριο οπτικό σημείο αναφοράς.



ΕΙΚΟΝΑ 9.1: Οθόνη εισόδου SmartNavix



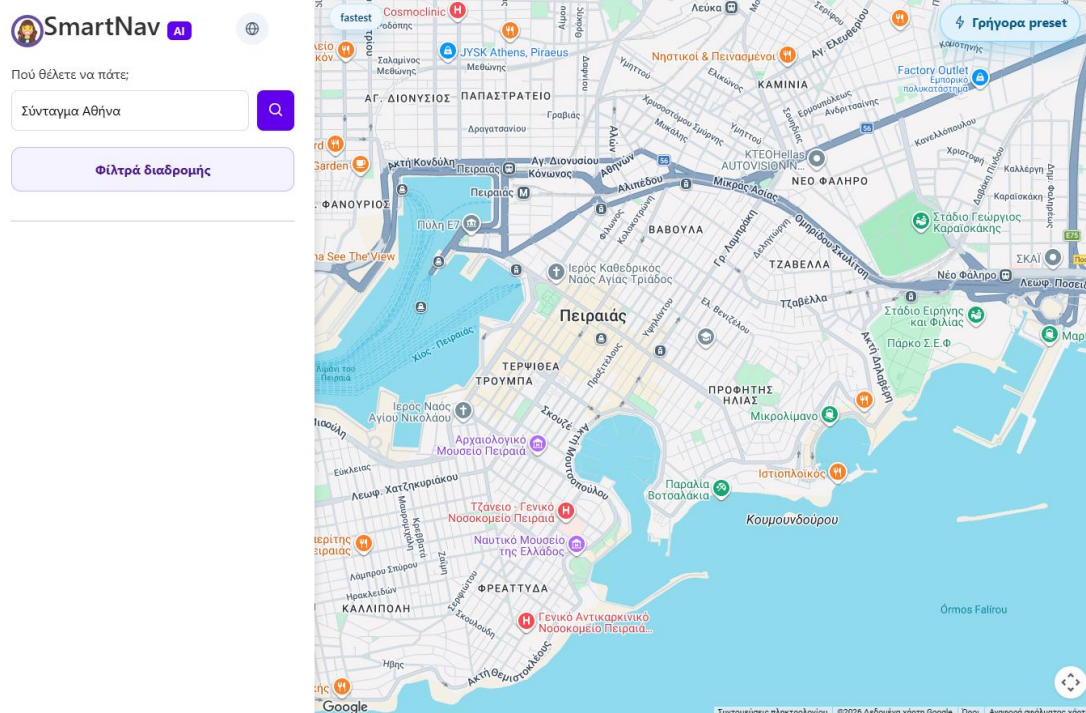
ΕΙΚΟΝΑ 9.2: Αρχική οθόνη πλοήγησης

9.2 Αναζήτηση διαδρομής από τον χρήστη

Η βασική ενέργεια του χρήστη είναι η εισαγωγή προορισμού στο πεδίο αναζήτησης. Στο παράδειγμα χρησιμοποιείται ο προορισμός «Σύνταγμα Αθήνα». Η εφαρμογή εμφανίζει το query πριν την εκτέλεση, ώστε ο χρήστης να μπορεί να ελέγξει τι έχει εισαγάγει και να διορθώσει το κείμενο πριν πατήσει το κουμπί αναζήτησης.

Η αναζήτηση δεν λειτουργεί ως απλό πεδίο διεύθυνσης. Συνδέεται με τις ήδη αποθηκευμένες ή επιλεγμένες προτιμήσεις του χρήστη και με την τρέχουσα θέση που παρέχεται από τον browser. Στην πράξη, ο χρήστης δίνει έναν προορισμό, ενώ το σύστημα συνθέτει το πλήρες αίτημα λαμβάνοντας υπόψη το πλαίσιο χρήσης.

Η οθόνη κρατά ορατά τα φίλτρα διαδρομής και την αναζήτηση μαζί με τον χάρτη. Έτσι ο χρήστης μπορεί να αντιληφθεί ότι η αναζήτηση και η εξατομίκευση δεν είναι δύο απομονωμένες λειτουργίες, αλλά δύο μέρη της ίδιας ροής.



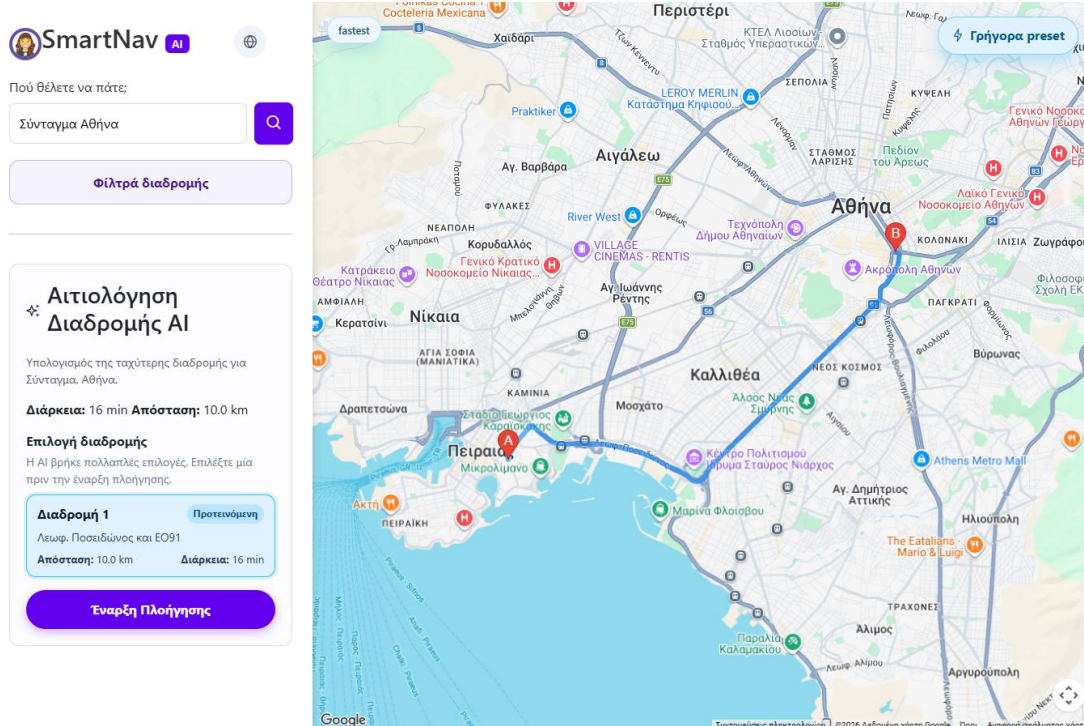
ΕΙΚΟΝΑ 9.3: Εισαγωγή προορισμού από τον χρήστη

9.3 Προβολή βασικής προτεινόμενης διαδρομής

Μετά την εκτέλεση της αναζήτησης, η SmartNavix εμφανίζει τη διαδρομή πάνω στον χάρτη και παράλληλα παρουσιάζει συνοπτική αιτιολόγηση στο πλαϊνό panel. Στο συγκεκριμένο παράδειγμα η εφαρμογή υπολογίζει διαδρομή προς το Σύνταγμα, εμφανίζει διάρκεια και απόσταση και σχεδιάζει τη διαδρομή από την περιοχή εκκίνησης προς τον προορισμό.

Η αιτιολόγηση της διαδρομής είναι σημαντική γιατί δεν αρκεί το σύστημα να επιστρέφει μια γραμμή στον χάρτη. Ο χρήστης χρειάζεται να βλέπει, με απλό τρόπο, γιατί προτείνεται η συγκεκριμένη επιλογή. Η ύπαρξη σύντομης περιγραφής, απόστασης και χρόνου ενισχύει την αναγνωσιμότητα της απόκρισης και μειώνει την ανάγκη δοκιμής πολλών επιλογών χωρίς εξήγηση.

Το στιγμιότυπο δείχνει επίσης ότι η εφαρμογή παραμένει σε κατάσταση λήψης απόφασης πριν από την έναρξη πλοήγησης. Ο χρήστης μπορεί να δει το αποτέλεσμα, να το συγκρίνει με τις ανάγκες του και έπειτα να πατήσει «Έναρξη Πλοήγησης».



ΕΙΚΟΝΑ 9.5: Απόκριση συστήματος με προτεινόμενη διαδρομή

9.4 Εμφάνιση εναλλακτικών διαδρομών

Η περιοχή «Επιλογή διαδρομής» χρησιμοποιείται για την παρουσίαση των διαθέσιμων επιλογών που επιστρέφει το σύστημα. Όταν υπάρχουν περισσότερες διαδρομές, εμφανίζονται ως ξεχωριστές κάρτες με σύνοψη, απόσταση και διάρκεια. Στο συγκεκριμένο στιγμιότυπο εμφανίζεται μία προτεινόμενη επιλογή, η οποία χαρακτηρίζεται ως «Προτεινόμενη».

Η ίδια δομή υποστηρίζει την ύπαρξη πολλών εναλλακτικών χωρίς να αλλάζει η λογική της διεπαφής. Ο χρήστης βλέπει τις επιλογές στην ίδια περιοχή και μπορεί να διαλέξει πριν ξεκινήσει την πλοήγηση. Αυτό είναι σημαντικό για εφαρμογές πλοήγησης, επειδή η «καλύτερη» διαδρομή μπορεί να εξαρτάται από προσωπικές προτιμήσεις και όχι μόνο από τον ελάχιστο χρόνο.

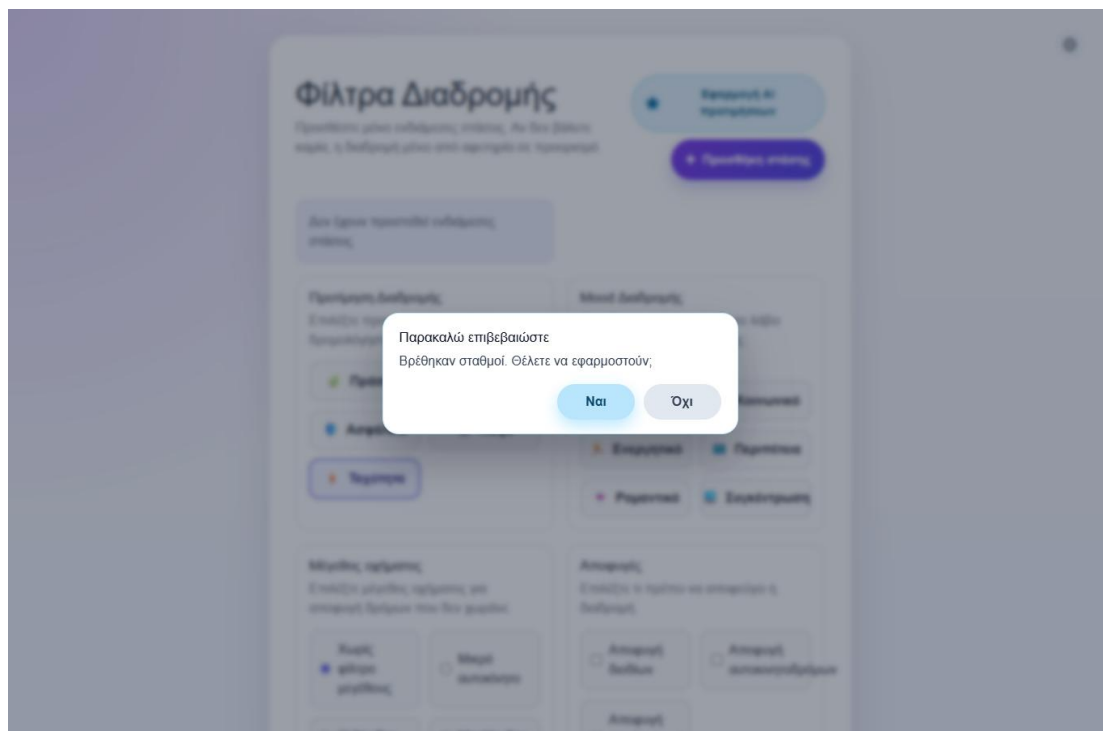
Η εμφάνιση της προτεινόμενης κάρτας λειτουργεί και ως οπτική επιβεβαίωση του decision layer της SmartNavix. Το σύστημα δεν επιστρέφει μόνο γεωμετρία διαδρομής, αλλά οργανώνει το αποτέλεσμα με τρόπο που μπορεί να αξιολογηθεί από τον χρήστη.

9.5 Χρήση φίλτρων και προτιμήσεων

Η οθόνη φίλτρων επιτρέπει στον χρήστη να επηρεάσει τη λογική της διαδρομής. Περιλαμβάνει επιλογές για προτίμηση διαδρομής, mood διαδρομής, μέγεθος οχήματος, αποφυγές, χρονικές ρυθμίσεις κυκλοφορίας και EV charging. Με αυτόν τον τρόπο η εφαρμογή δεν αντιμετωπίζει όλους τους χρήστες ως ίδιους, αλλά δίνει χώρο σε διαφορετικές ανάγκες μετακίνησης.

Στο στιγμιότυπο εμφανίζεται και το confirmation dialog που ενημερώνει τον χρήστη ότι έχουν βρεθεί αποθηκευμένοι σταθμοί ή φίλτρα και ζητά επιβεβαίωση πριν εφαρμοστούν. Η συμπεριφορά αυτή είναι χρήσιμη, γιατί δεν εφαρμόζει σιωπηρά παλιές επιλογές χωρίς ενημέρωση. Ο χρήστης διατηρεί τον έλεγχο της απόφασης.

Η ύπαρξη τέτοιας επιβεβαίωσης είναι σύμφωνη με βασικές αρχές σχεδιασμού διεπαφών, όπου οι ενέργειες του συστήματος πρέπει να είναι κατανοητές και να παρέχουν ανατροφοδότηση στον χρήστη [42]. Στη SmartNavix, η ανατροφοδότηση αυτή είναι κρίσιμη επειδή τα φίλτρα μπορούν να αλλάξουν ουσιαστικά το αποτέλεσμα της διαδρομής.



ΕΙΚΟΝΑ 9.4: Φίλτρα διαδρομής και επιβεβαίωση αποθηκευμένων επιλογών

9.6 Προτάσεις σημείων ενδιαφέροντος

Η SmartNavix αξιοποιεί τον χάρτη όχι μόνο για τη γραμμή της διαδρομής, αλλά και για την εμφάνιση σημείων ενδιαφέροντος γύρω από την περιοχή μετακίνησης. Στα στιγμιότυπα της διαδρομής εμφανίζονται σημεία όπως χώροι εστίασης, υπηρεσίες, μέσα μεταφοράς και άλλα οπτικά markers του χάρτη. Αυτά βοηθούν τον χρήστη να κατανοήσει καλύτερα το περιβάλλον γύρω από τη διαδρομή.

Η λειτουργία των σημείων ενδιαφέροντος συνδέεται με την εμπειρία χρήστη, επειδή μια διαδρομή δεν αξιολογείται πάντα μόνο με βάση χρόνο και απόσταση. Σε πραγματικές συνθήκες, ο χρήστης μπορεί να ενδιαφέρεται για σημεία στάσης, εξυπηρέτησης ή προσανατολισμού. Η παρουσία τους στον χάρτη δίνει περισσότερα συμφοραζόμενα στην απόφαση.

Στην τρέχουσα ροή του παραδείγματος, τα σημεία ενδιαφέροντος εμφανίζονται κυρίως ως markers πάνω στον χάρτη και όχι ως ξεχωριστή λίστα στο πλαϊνό panel. Αυτό δείχνει ότι η λειτουργία υπάρχει στο οπτικό επίπεδο της πλοήγησης, ενώ η πλήρης αξιοποίηση της ως ανεξάρτητης λίστας μπορεί να αποτελέσει αντικείμενο μελλοντικής βελτίωσης.

9.7 Επιλογή best preference από το σύστημα

Η επιλογή best preference γίνεται ορατή στον χρήστη μέσα από την εφαρμογή φίλτρων και την προτεινόμενη διαδρομή. Όταν υπάρχουν αποθηκευμένες προτιμήσεις ή σταθμοί, η εφαρμογή ζητά επιβεβαίωση για την εφαρμογή τους. Στη συνέχεια, η προτεινόμενη επιλογή εμφανίζεται στην περιοχή επιλογής διαδρομής με σχετική ένδειξη.

Η λειτουργία αυτή συνδέει το fuzzy μοντέλο του Κεφαλαίου 8 με την πραγματική διεπαφή. Ο χρήστης δεν βλέπει τους εσωτερικούς υπολογισμούς frequency, recency ή completeness, αλλά βλέπει την πρακτική τους συνέπεια: μια αρχική πρόταση που βασίζεται στο ιστορικό και στις ρυθμίσεις του.

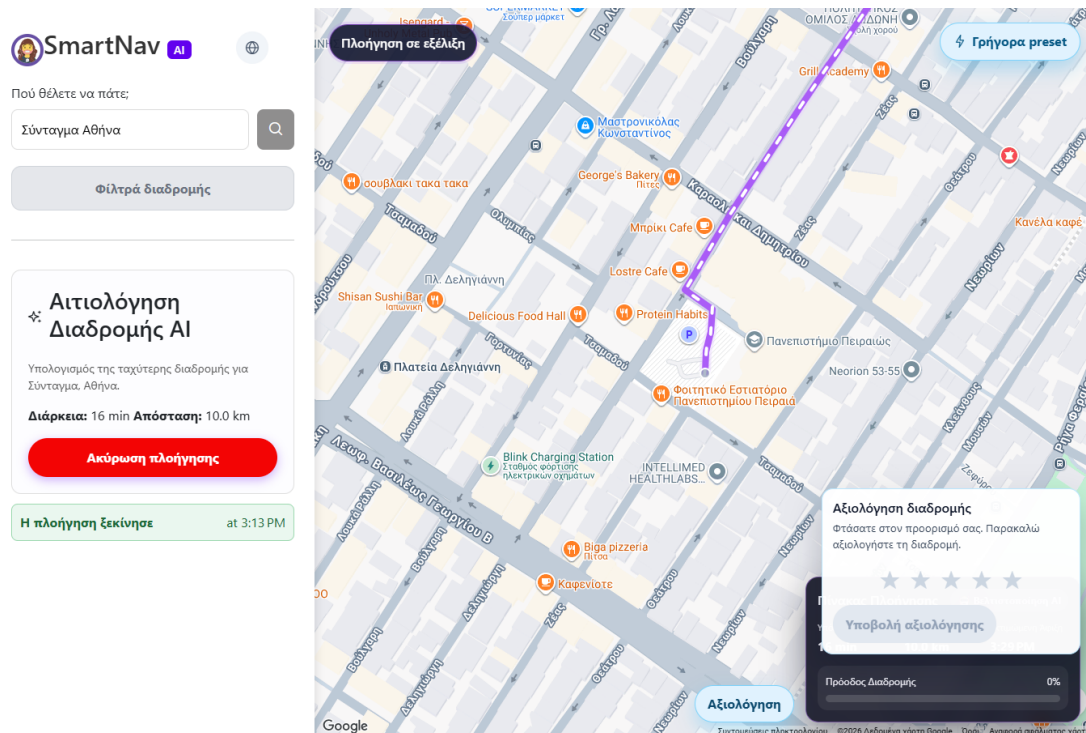
Είναι σημαντικό ότι η εφαρμογή δεν κλειδώνει τον χρήστη στην προτεινόμενη επιλογή. Η πρόταση λειτουργεί ως βοήθεια και όχι ως υποχρεωτική απόφαση. Έτσι η προσαρμοστικότητα παραμένει διαφανής και ελεγχόμενη από τον τελικό χρήστη.

9.8 Αξιολόγηση διαδρομής και feedback χρήση

Μετά την έναρξη πλοήγησης, η εφαρμογή εμφανίζει την κατάσταση «Η πλοήγηση ξεκίνησε» και ενεργοποιεί την περιοχή αξιολόγησης. Ο χρήστης μπορεί να βαθμολογήσει τη διαδρομή με αστέρια και να υποβάλει την αξιολόγηση. Η λειτουργία αυτή μετατρέπει την εμπειρία χρήσης σε δεδομένο που μπορεί να αξιοποιηθεί από το σύστημα.

Το feedback είναι ιδιαίτερα χρήσιμο σε μια εφαρμογή που επιχειρεί να προσαρμόζεται στις προτιμήσεις του χρήστη. Αν μια διαδρομή θεωρηθεί καλή, η πληροφορία αυτή μπορεί να ενισχύσει μελλοντικές επιλογές. Αντίστοιχα, χαμηλή αξιολόγηση μπορεί να δείξει ότι η διαδρομή δεν ταίριαζε στις πραγματικές ανάγκες του χρήστη, ακόμη και αν ήταν τεχνικά σωστή.

Η σχεδίαση της φόρμας αξιολόγησης είναι ελαφριά και εμφανίζεται πάνω στον χάρτη χωρίς να απομακρύνει τον χρήστη από την πλοήγηση. Αυτό μειώνει το κόστος αλληλεπίδρασης και αυξάνει την πιθανότητα ο χρήστης να δώσει ανατροφοδότηση.



ΕΙΚΟΝΑ 9.6: Αξιολόγηση διαδρομής μετά την έναρξη πλοήγησης

9.9 Διαχείριση από admin panel

Η εφαρμογή περιλαμβάνει admin panel για χρήστες με ρόλο διαχειριστή. Στο σχετικό στιγμιότυπο εμφανίζονται συνοπτικοί δείκτες, όπως σύνολο χρηστών, επιβεβαιωμένοι χρήστες, συνολικές διαδρομές και μέσο score. Ακολουθεί πίνακας διαχείρισης χρηστών με ρόλους, κατάσταση επιβεβαίωσης και διαθέσιμες ενέργειες.

Το admin panel είναι σημαντικό γιατί δείχνει ότι η SmartNavix δεν περιορίζεται στην εμπειρία του απλού χρήστη. Υποστηρίζει και λειτουργίες εποπτείας, οι οποίες είναι απαραίτητες όταν το σύστημα έχει ρόλους, εγγραφές χρηστών, αξιολογήσεις και ιστορικό ενεργειών. Ο διαχειριστής μπορεί να ελέγχει χρήστες και να παρακολουθεί βασικά στοιχεία λειτουργίας.

Στην ίδια οθόνη εμφανίζεται και ιστορικό ενεργειών διαχείρισης. Η ύπαρξη audit πληροφορίας βοηθά στη διαφάνεια των διοικητικών ενεργειών και διευκολύνει τον εντοπισμό αλλαγών που έχουν γίνει σε χρήστες ή ρόλους.

Κεφάλαιο 10: Εγχειρίδιο Χρήστη

Το κεφάλαιο αυτό περιγράφει τη χρήση της SmartNavix από την πλευρά του τελικού χρήστη. Οι οδηγίες οργανώνονται σύμφωνα με τις βασικές εργασίες που εκτελούνται στην εφαρμογή, ώστε να μπορεί να χρησιμοποιηθεί ως πρακτικός οδηγός κατά την παρουσίαση ή τη δοκιμή του συστήματος.

10.1 Σκοπός και κοινό του εγχειριδίου

Το παρόν κεφάλαιο λειτουργεί ως εγχειρίδιο χρήστη για τη SmartNavix. Σε αντίθεση με τα προηγούμενα κεφάλαια, τα οποία περιγράφουν την ανάλυση, τον σχεδιασμό και την υλοποίηση του συστήματος, εδώ δίνεται έμφαση στις ενέργειες που εκτελεί ο τελικός χρήστης μέσα στην εφαρμογή. Το εγχειρίδιο απευθύνεται σε απλούς χρήστες, contributors και διαχειριστές, ανάλογα με τα δικαιώματα που διαθέτει κάθε λογαριασμός.

Η παρουσίαση ακολουθεί τη φυσική ροή χρήσης: είσοδος στην εφαρμογή, βασική αναζήτηση, ρύθμιση φίλτρων, έναρξη πλοήγησης, αξιολόγηση διαδρομής, διαχείριση προφίλ και χρήση του admin panel. Η διάκριση αυτή βοηθά τον αναγνώστη να αντιστοιχίσει κάθε λειτουργία με συγκεκριμένη ενέργεια και όχι μόνο με τεχνικό χαρακτηριστικό του συστήματος.

Η λογική του εγχειριδίου συνδέεται με την ευχρηστία, επειδή ένα σύστημα δεν αρκεί να διαθέτει λειτουργίες· πρέπει ο χρήστης να μπορεί να τις κατανοήσει και να τις εκτελέσει μέσα στο πραγματικό πλαίσιο χρήσης [31]. Για τον λόγο αυτό, οι οδηγίες διατυπώνονται με πρακτικό τρόπο, χωρίς να εισάγουν εσωτερικές λεπτομέρειες υλοποίησης που δεν χρειάζεται να γνωρίζει ο τελικός χρήστης.

10.2 Σύνδεση στην εφαρμογή

Για να χρησιμοποιήσει τη SmartNavix, ο χρήστης μεταβαίνει στην οθόνη εισόδου της εφαρμογής. Στο πεδίο «Όνομα χρήστη» πληκτρολογεί το username του λογαριασμού και στο πεδίο «Κωδικός» τον αντίστοιχο κωδικό πρόσβασης. Αν επιθυμεί να παραμείνει συνδεδεμένος στη συσκευή, μπορεί να ενεργοποιήσει την επιλογή «Να με θυμάσαι». Στη συνέχεια πατά το κουμπί «Είσοδος».

Αν τα στοιχεία είναι σωστά, η εφαρμογή μεταφέρει τον χρήστη στην κεντρική οθόνη πλοήγησης. Αν τα στοιχεία δεν είναι σωστά ή ο λογαριασμός δεν είναι διαθέσιμος, η σύνδεση δεν ολοκληρώνεται και ο χρήστης πρέπει να ελέγξει τα δεδομένα που εισήγαγε. Η εφαρμογή δεν εμφανίζει περιττές τεχνικές πληροφορίες σε αυτή τη φάση, επειδή ο σκοπός είναι να δοθεί κατανοητή ανατροφοδότηση και όχι λεπτομέρειες εσωτερικού σφάλματος.

Σε περίπτωση που ο χρήστης δεν θυμάται τον κωδικό του, μπορεί να χρησιμοποιήσει τη λειτουργία ανάκτησης κωδικού από τον σύνδεσμο «Ξέχασες τον κωδικό σου;». Αν δεν διαθέτει λογαριασμό, επιλέγει «Εγγραφή» για να μεταφερθεί στη φόρμα δημιουργίας νέου λογαριασμού.

10.3 Δημιουργία και βασική διαχείριση λογαριασμού

Η δημιουργία λογαριασμού πραγματοποιείται από την οθόνη εγγραφής. Ο χρήστης συμπληρώνει τα απαιτούμενα στοιχεία, όπως username, email και κωδικό πρόσβασης, και υποβάλλει τη φόρμα. Η εφαρμογή ελέγχει την εγκυρότητα των στοιχείων πριν επιτρέψει την ολοκλήρωση της εγγραφής, ώστε να μειωθούν λάθη όπως κενά πεδία ή μη αποδεκτές τιμές.

Μετά τη σύνδεση, ο χρήστης μπορεί να μεταβεί στο προφίλ του από το εικονίδιο avatar που βρίσκεται στην κεντρική οθόνη. Εκεί εμφανίζονται βασικές πληροφορίες λογαριασμού, ο ρόλος του χρήστη και στοιχεία ιστορικού, όπως οι διαδρομές που έχουν αποθηκευτεί. Η οθόνη προφίλ λειτουργεί ως σημείο προσωπικής επισκόπησης και όχι ως τεχνικός πίνακας ρυθμίσεων.

Από το ίδιο περιβάλλον ο χρήστης μπορεί να αλλάξει κωδικό, να επιστρέψει στην αρχική οθόνη, να μεταβεί στις ρυθμίσεις ή να αποσυνδεθεί. Αν ο λογαριασμός έχει ρόλο διαχειριστή, εμφανίζεται επιπλέον επιλογή μετάβασης στο admin panel.

10.4 Χρήση βασικής αναζήτησης διαδρομής

Η βασική αναζήτηση εκτελείται από την κεντρική οθόνη της εφαρμογής. Ο χρήστης πληκτρολογεί τον προορισμό στο πεδίο αναζήτησης και πατά το κουμπί με το εικονίδιο αναζήτησης. Για να λειτουργήσει σωστά η αναζήτηση, ο browser πρέπει να επιτρέπει πρόσβαση στην τρέχουσα τοποθεσία, επειδή η εφαρμογή χρειάζεται σημείο εκκίνησης για να υπολογίσει τη διαδρομή.

Μετά την υποβολή, η SmartNavix επεξεργάζεται το αίτημα και εμφανίζει τη διαδρομή στον χάρτη. Παράλληλα, στο πλαϊνό panel εμφανίζονται βασικά στοιχεία όπως διάρκεια, απόσταση και σύντομη αιτιολόγηση της διαδρομής. Ο χρήστης μπορεί να ελέγξει το αποτέλεσμα πριν επιλέξει να ξεκινήσει την πλοήγηση.

Αν η εφαρμογή δεν μπορεί να δημιουργήσει διαδρομή, ο χρήστης πρέπει πρώτα να ελέγξει αν έχει δοθεί άδεια τοποθεσίας, αν ο προορισμός είναι επαρκώς σαφής και αν υπάρχει σύνδεση με τις εξωτερικές υπηρεσίες χαρτών. Η αναζήτηση μπορεί να επαναληφθεί με πιο συγκεκριμένο προορισμό, όπως περιοχή, οδός ή γνωστό σημείο ενδιαφέροντος.

10.5 Ρύθμιση φίλτρων και προτιμήσεων

Ο χρήστης μπορεί να πατήσει το κουμπί «Φίλτρα διαδρομής» για να ανοίξει την οθόνη προτιμήσεων. Εκεί μπορεί να ορίσει στάσεις, προτίμηση διαδρομής, mood, μέγεθος οχήματος, αποφυγές, χρονική πληροφορία κυκλοφορίας και επιλογή για σταθμούς φόρτισης ηλεκτρικών οχημάτων. Οι επιλογές αυτές επηρεάζουν τον τρόπο με τον οποίο συντίθεται το αίτημα διαδρομής.

Αν υπάρχουν αποθηκευμένες προτιμήσεις ή σταθμοί, η εφαρμογή εμφανίζει μήνυμα επιβεβαίωσης πριν τις εφαρμοδίσει. Ο χρήστης μπορεί να επιλέξει «Ναι» αν θέλει να χρησιμοποιήσει τις αποθηκευμένες επιλογές ή «Όχι» αν θέλει να ρυθμίσει τη διαδρομή από την αρχή. Η επιβεβαίωση αυτή είναι σημαντική, επειδή αποφεύγεται η ακούσια εφαρμογή παλαιότερων ρυθμίσεων.

Μετά την επιλογή φίλτρων, ο χρήστης πατά «Εφαρμογή». Η εφαρμογή επιστρέφει στην κεντρική οθόνη και εμφανίζει τα εφαρμοσμένα φίλτρα ως οπτικά στοιχεία πάνω από τον χάρτη. Με αυτόν τον τρόπο ο χρήστης έχει άμεση ένδειξη ότι οι επιλογές του έχουν ληφθεί υπόψη. Η σαφής οπτική ανατροφοδότηση αποτελεί βασικό στοιχείο αποτελεσματικής διεπαφής χρήστη [42].

10.6 Χρήση γρήγορων presets

Η λειτουργία γρήγορων presets επιτρέπει στον χρήστη να αποθηκεύει συχνά χρησιμοποιούμενα σημεία, όπως σπίτι, εργασία ή άλλο επαναλαμβανόμενο προορισμό. Από το κουμπί «Γρήγορα preset» στην κεντρική οθόνη, ο χρήστης ανοίγει τη σχετική φόρμα και εισάγει στοιχεία διεύθυνσης, όπως οδό, αριθμό, πόλη ή ταχυδρομικό κώδικα.

Κατά τη συμπλήρωση της διεύθυνσης, η εφαρμογή μπορεί να εμφανίσει προτάσεις μέσω autocomplete. Ο χρήστης επιλέγει την καταλληλότερη πρόταση και στη συνέχεια διαλέγει εικονίδιο για το preset. Το εικονίδιο βοηθά στην αναγνώριση του σημείου στον χάρτη και κάνει τη λειτουργία πιο άμεση σε επαναλαμβανόμενη χρήση.

Αφού συμπληρωθούν τα απαραίτητα στοιχεία, ο χρήστης αποθηκεύει το preset. Στη συνέχεια το σημείο μπορεί να εμφανίζεται στον χάρτη ή να χρησιμοποιείται μέσα από τις επιλογές φίλτρων. Η λειτουργία αυτή μειώνει την ανάγκη επαναλαμβανόμενης πληκτρολόγησης και υποστηρίζει πιο γρήγορη χρήση της εφαρμογής.

10.7 Έναρξη, παρακολούθηση και ακύρωση πλοήγησης

Όταν η εφαρμογή εμφανίσει προτεινόμενη διαδρομή, ο χρήστης μπορεί να πατήσει «Έναρξη Πλοήγησης». Με την ενέργεια αυτή η διαδρομή περνά σε κατάσταση ενεργής πλοήγησης και ο χάρτης εμφανίζει την πορεία με διαφορετικό οπτικό τρόπο. Το πλαϊνό panel ενημερώνεται ώστε να δείχνει ότι η πλοήγηση έχει ξεκινήσει.

Κατά τη διάρκεια της πλοήγησης, η εφαρμογή εμφανίζει στοιχεία όπως πρόοδος διαδρομής, εκτιμώμενος χρόνος και απόσταση. Ο χρήστης μπορεί να παρακολουθεί την πορεία χωρίς να χρειάζεται

να μεταβεί σε άλλη οθόνη. Η οργάνωση αυτή κρατά το βασικό περιεχόμενο στο ίδιο περιβάλλον, κάτι που μειώνει τη γνωστική επιβάρυνση του χρήστη.

Αν ο χρήστης δεν θέλει να συνεχίσει, μπορεί να πατήσει «Ακύρωση πλοήγησης». Με την ακύρωση, η εφαρμογή σταματά την ενεργή πλοήγηση και επιστρέφει σε κατάσταση όπου ο χρήστης μπορεί να αναζητήσει νέα διαδρομή ή να αλλάξει φίλτρα.

10.8 Αξιολόγηση διαδρομής και ανατροφοδότηση

Μετά την έναρξη πλοήγησης, η εφαρμογή δίνει τη δυνατότητα αξιολόγησης της διαδρομής. Ο χρήστης ανοίγει την περιοχή αξιολόγησης και επιλέγει βαθμολογία με αστέρια. Στη συνέχεια πατά «Υποβολή αξιολόγησης» για να αποθηκεύσει την ανατροφοδότηση.

Η αξιολόγηση δεν είναι απλώς διακοσμητική λειτουργία. Συνδέεται με την προσπάθεια της SmartNavix να κατανοεί καλύτερα αν οι διαδρομές που προτείνονται είναι χρήσιμες για τον συγκεκριμένο χρήστη. Η βαθμολογία μπορεί να αξιοποιηθεί ως ένδειξη ικανοποίησης και ως μέρος του ιστορικού χρήσης.

Ο χρήστης πρέπει να αξιολογεί τη διαδρομή με βάση την πραγματική εμπειρία μετακίνησης. Για παράδειγμα, μια διαδρομή μπορεί να είναι σύντομη αλλά όχι άνετη ή μπορεί να αποφεύγει δρόμους που ο χρήστης δεν επιθυμεί. Αυτή η διάκριση είναι σημαντική, επειδή η εμπειρία χρήστη δεν περιορίζεται μόνο σε αντικειμενικά μετρήσιμα χαρακτηριστικά, αλλά περιλαμβάνει και την αντιληπτή καταλληλότητα της λύσης [29], [31].

10.9 Χρήση προφίλ και προσωπικών ρυθμίσεων

Το προφίλ χρήστη συγκεντρώνει βασικά στοιχεία του λογαριασμού και επιτρέπει γρήγορη πρόσβαση σε προσωπικές λειτουργίες. Ο χρήστης μπορεί να δει το username, το email, τον ρόλο και στατιστικά χρήσης, όπως αριθμό διαδρομών και συνολική απόσταση. Αν υπάρχουν αποθηκευμένες διαδρομές, εμφανίζονται σε πίνακα ιστορικού.

Από το προφίλ ο χρήστης μπορεί να αλλάξει κωδικό, να επεξεργαστεί βασικά στοιχεία όπου επιτρέπεται, να μεταβεί στις ρυθμίσεις ή να αποσυνδεθεί. Οι ρυθμίσεις μπορεί να περιλαμβάνουν επιλογές που επηρεάζουν τη συμπεριφορά της εφαρμογής, όπως αριθμό εναλλακτικών διαδρομών που προτιμά να εμφανίζονται.

Η οργάνωση του προφίλ βοηθά στη διάκριση ανάμεσα στις λειτουργίες πλοήγησης και στις λειτουργίες λογαριασμού. Έτσι, ο χρήστης δεν χρειάζεται να ψάχνει προσωπικές ρυθμίσεις μέσα στην οθόνη του χάρτη, ενώ η κύρια ροή πλοήγησης παραμένει καθαρή.

10.10 Χρήση admin panel και αντιμετώπιση συχνών προβλημάτων

Οι χρήστες με ρόλο διαχειριστή έχουν πρόσβαση στο admin panel. Από εκεί μπορούν να δουν συνοπτικά στατιστικά, λίστα χρηστών, ρόλους, κατάσταση επιβεβαίωσης και ιστορικό διοικητικών ενεργειών. Ο διαχειριστής μπορεί να αναζητήσει χρήστη, να αλλάξει ρόλο, να τροποποιήσει κατάσταση επιβεβαίωσης ή να διαγράψει χρήστη όπου η εφαρμογή το επιτρέπει.

Κατά τη χρήση του admin panel πρέπει να αποφεύγονται άσκοπες αλλαγές ρόλων, επειδή οι ρόλοι επηρεάζουν την πρόσβαση σε λειτουργίες της εφαρμογής. Αν ο διαχειριστής αλλάξει στοιχεία χρηστών, πρέπει να εφαρμόσει τις αλλαγές από το αντίστοιχο κουμπί, ώστε οι τροποποιήσεις να αποθηκευτούν στο backend.

Συχνά προβλήματα που μπορεί να συναντήσει ο χρήστης είναι η αποτυχία σύνδεσης, η μη εμφάνιση διαδρομής, η άρνηση πρόσβασης τοποθεσίας και η μη εμφάνιση επιλογών διαχειριστή. Σε αποτυχία σύνδεσης ελέγχονται username και password. Σε πρόβλημα διαδρομής ελέγχεται η άδεια τοποθεσίας και η σαφήνεια του προορισμού. Αν δεν εμφανίζεται admin panel, ο λογαριασμός πιθανότατα δεν έχει ρόλο διαχειριστή. Με αυτόν τον τρόπο, η αντιμετώπιση ξεκινά από απλά σημεία ελέγχου πριν θεωρηθεί ότι υπάρχει τεχνικό σφάλμα.

Κεφάλαιο 11: Οφέλη της Προτεινόμενης Λύσης

Το κεφάλαιο αυτό συνοψίζει τα οφέλη που αναμένονται από τη λύση SmartNavix. Η αποτίμηση βασίζεται στις λειτουργίες που υλοποιήθηκαν, στη σχεδίαση του συστήματος και στη σύνδεση της εφαρμογής με έννοιες όπως ευχρηστία, εξατομίκευση, προσαρμοστικότητα και υποστήριξη απόφασης.

11.1 Γενική αποτίμηση της προτεινόμενης λύσης

Η SmartNavix προτείνεται ως μια εφαρμογή πλοήγησης που δεν περιορίζεται στον υπολογισμό μιας διαδρομής, αλλά επιχειρεί να εντάξει τις προτιμήσεις του χρήστη στη διαδικασία επιλογής. Το βασικό αναμενόμενο όφελος είναι ότι η διαδρομή αντιμετωπίζεται ως απόφαση που εξαρτάται από το πλαίσιο χρήσης και όχι μόνο ως γεωγραφικό πρόβλημα ελάχιστης απόστασης ή χρόνου.

Η λύση συνδυάζει λειτουργίες χάρτη, φίλτρα, ιστορικό επιλογών, fuzzy logic και υποστήριξη τεχνητής νοημοσύνης. Αυτός ο συνδυασμός δημιουργεί ένα πιο πλούσιο περιβάλλον αλληλεπίδρασης, επειδή ο χρήστης μπορεί να εκφράσει ανάγκες που συχνά δεν αποτυπώνονται πλήρως σε μια απλή αναζήτηση προορισμού. Η προσέγγιση συνδέεται με τη λογική των προσαρμοστικών συστημάτων, όπου το σύστημα μεταβάλλει τη συμπεριφορά του με βάση στοιχεία χρήστη ή χρήσης [34], [35].

Τα οφέλη που παρουσιάζονται στο κεφάλαιο αυτό πρέπει να διαβαστούν ως αναμενόμενα οφέλη της προτεινόμενης λύσης. Δεν αποτελούν αποτελέσματα μεγάλης κλίμακας παραγωγικής αξιολόγησης, αλλά τεκμηριωμένη εκτίμηση με βάση τις λειτουργίες που υλοποιήθηκαν, τις απαιτήσεις του συστήματος και τη σχετική βιβλιογραφία.

11.2 Οφέλη για τον τελικό χρήστη

Για τον τελικό χρήστη, το σημαντικότερο όφελος είναι η δυνατότητα πιο προσωπικής αναζήτησης διαδρομής. Ο χρήστης δεν χρειάζεται να αποδέχεται μόνο την ταχύτερη ή συντομότερη επιλογή, αλλά μπορεί να δηλώσει προτιμήσεις, στάσεις, αποφυγές, mood διαδρομής και ανάγκες όπως η ύπαρξη σταθμών φόρτισης. Έτσι, η εφαρμογή μπορεί να υποστηρίξει διαφορετικές μετακινήσεις και διαφορετικούς τύπους χρηστών.

Ένα δεύτερο όφελος είναι η μείωση επαναλαμβανόμενων ενεργειών. Μέσα από presets και αποθηκευμένες προτιμήσεις, ο χρήστης μπορεί να επαναχρησιμοποιεί επιλογές που εμφανίζονται συχνά στην καθημερινή του χρήση. Αυτό είναι πρακτικό σε εφαρμογές πλοήγησης, επειδή οι χρήστες συχνά επιστρέφουν σε γνώριμους προορισμούς ή επαναλαμβάνουν συγκεκριμένα μοτίβα μετακίνησης.

Η εφαρμογή προσφέρει επίσης σαφέστερη ανατροφοδότηση. Η προτεινόμενη διαδρομή συνοδεύεται από απόσταση, διάρκεια, αιτιολόγηση και δυνατότητα αξιολόγησης. Η ανατροφοδότηση αυτή βοηθά τον χρήστη να κατανοήσει τι επιστρέφει το σύστημα και να αποφασίσει αν θα ξεκινήσει την πλοήγηση ή αν θα αλλάξει φίλτρα.

11.3 Βελτίωση εμπειρίας χρήστη και ευχρηστίας

Η εμπειρία χρήστη δεν εξαρτάται μόνο από το αν η εφαρμογή εκτελεί σωστά μια λειτουργία, αλλά και από το πώς ο χρήστης αντιλαμβάνεται τη χρησιμότητα, τη σαφήνεια και την καταλληλότητα της αλληλεπίδρασης [29]. Στη SmartNavix, η διεπαφή οργανώνεται γύρω από έναν χάρτη, ένα απλό πεδίο αναζήτησης και πρόσβαση σε φίλτρα, ώστε η βασική ροή να παραμένει αναγνωρίσιμη.

Η ευχρηστία σχετίζεται με την αποτελεσματικότητα, την αποδοτικότητα και την ικανοποίηση μέσα σε συγκεκριμένο πλαίσιο χρήσης [31]. Η SmartNavix επιχειρεί να βελτιώσει αυτά τα στοιχεία με τρεις τρόπους: μειώνει την ανάγκη επαναεισαγωγής προτιμήσεων, κάνει ορατές τις επιλογές που επηρεάζουν τη διαδρομή και δίνει στον χρήστη δυνατότητα να αξιολογήσει το αποτέλεσμα.

Παράλληλα, ο ανθρωποκεντρικός σχεδιασμός δίνει έμφαση στην κατανόηση χρηστών, εργασιών και περιβάλλοντος χρήσης [32]. Η SmartNavix ακολουθεί αυτή τη λογική στο επίπεδο της λειτουργικής σχεδίασης, επειδή δεν θεωρεί ότι όλοι οι χρήστες έχουν την ίδια αντίληψη για την κατάλληλη διαδρομή.

Η ίδια μετακίνηση μπορεί να αξιολογηθεί διαφορετικά ανάλογα με τον σκοπό, το όχημα, τη διάθεση ή την ανάγκη αποφυγής συγκεκριμένων συνθηκών.

11.4 Οφέλη από την εξατομίκευση και την προσαρμοστικότητα

Η εξατομίκευση αποτελεί κεντρικό όφελος της προτεινόμενης λύσης. Μέσα από την αποθήκευση φίλτρων και την επιλογή best preference, η εφαρμογή μπορεί να αξιοποιεί προηγούμενη συμπεριφορά για να υποστηρίζει πιο κατάλληλες μελλοντικές επιλογές. Αυτό συνδέεται με τη λογική των συστημάτων προτάσεων, τα οποία οργανώνουν πληροφορία ώστε να βοηθούν τον χρήστη να επιλέξει ανάμεσα σε εναλλακτικές [51], [52].

Στη SmartNavix, η εξατομίκευση δεν βασίζεται μόνο σε μια κρυφή απόφαση τεχνητής νοημοσύνης. Το fuzzy μοντέλο προσφέρει έναν ερμηνεύσιμο μηχανισμό επιλογής, όπου συνδυάζονται συχνότητα, πρόσφατη χρήση, πληρότητα και συνέπεια. Η χρήση fuzzy λογικής είναι κατάλληλη σε προβλήματα όπου οι έννοιες δεν έχουν αυστηρά όρια, αλλά εκφράζονται με βαθμούς συμμετοχής [56].

Το πρακτικό όφελος είναι ότι η εφαρμογή μπορεί να προτείνει μια αρχική προτίμηση χωρίς να ζητά κάθε φορά από τον χρήστη να ξεκινήσει από μηδενική ρύθμιση. Ταυτόχρονα, ο χρήστης διατηρεί τον έλεγχο, επειδή μπορεί να αποδεχθεί ή να απορρίψει αποθηκευμένες επιλογές και να αλλάξει τα φίλτρα πριν από την αναζήτηση.

11.5 Υποστήριξη λήψης απόφασης στη διαδρομή

Η επιλογή διαδρομής είναι συχνά απόφαση με περισσότερα από ένα κριτήρια. Ο συνολικός χρόνος μετακίνησης είναι σημαντικός, αλλά δεν είναι πάντα το μόνο κριτήριο. Ο χρήστης μπορεί να ενδιαφέρεται για στάσεις, σημεία ενδιαφέροντος, αποφυγή διοδίων ή αυτοκινητοδρόμων, φόρτιση ηλεκτρικού οχήματος ή μια διαδρομή που ταιριάζει καλύτερα στη διάθεση και στον σκοπό της μετακίνησης.

Η SmartNavix υποστηρίζει αυτή τη λήψη απόφασης επειδή εμφανίζει διαδρομή, συνοπτική αιτιολόγηση, διάρκεια, απόσταση και προτεινόμενη επιλογή. Η απόφαση δεν παρουσιάζεται ως αδιαφανές αποτέλεσμα, αλλά ως αποτέλεσμα που μπορεί να διαβαστεί και να αξιολογηθεί από τον χρήστη. Αυτό είναι σημαντικό σε συστήματα που χρησιμοποιούν προτάσεις, επειδή η εμπιστοσύνη του χρήστη ενισχύεται όταν το αποτέλεσμα είναι κατανοητό.

Η fuzzy προσέγγιση προσφέρει επιπλέον όφελος σε αυτό το σημείο, επειδή ταιριάζει σε περιβάλλοντα όπου η απόφαση δεν είναι πλήρως βέβαιη. Η επιλογή μιας προτίμησης μπορεί να θεωρηθεί πρόβλημα λήψης απόφασης σε ασαφές περιβάλλον, όπου οι εναλλακτικές δεν αξιολογούνται μόνο με απόλυτους αριθμούς [58].

11.6 Οφέλη για διαχείριση, εποπτεία και βελτίωση της εφαρμογής

Η ύπαρξη admin panel προσθέτει οφέλη σε επίπεδο διαχείρισης. Ο διαχειριστής μπορεί να βλέπει χρήστες, ρόλους, επιβεβαιώσεις, ιστορικό ενεργειών και βασικά στατιστικά. Αυτό βοηθά στην εποπτεία της εφαρμογής και στη διαχείριση πρόσβασης, ειδικά όταν υπάρχουν διαφορετικοί ρόλοι όπως user, contributor και admin.

Τα στοιχεία αξιολόγησης διαδρομών και τα συνοπτικά analytics μπορούν να αξιοποιηθούν για μελλοντική βελτίωση. Για παράδειγμα, αν συγκεκριμένοι τύποι διαδρομής λαμβάνουν συχνά χαμηλές αξιολογήσεις, αυτό μπορεί να υποδείξει ανάγκη αλλαγής φίλτρων, βαρών ή τρόπου παρουσίασης των επιλογών. Έτσι, η εφαρμογή αποκτά μηχανισμό ανατροφοδότησης και δεν παραμένει στατική.

Η καταγραφή διοικητικών ενεργειών ενισχύει και τη διαφάνεια της διαχείρισης. Όταν οι αλλαγές σε ρόλους ή χρήστες καταγράφονται, είναι πιο εύκολο να εντοπιστεί τι έγινε και από ποιον. Αυτό δεν αντικαθιστά μια πλήρη πολιτική ασφάλειας, αλλά αποτελεί σημαντικό στοιχείο υπεύθυνης λειτουργίας.

11.7 Τεχνικά και εκπαιδευτικά οφέλη της υλοποίησης

Σε τεχνικό επίπεδο, η SmartNavix λειτουργεί ως ολοκληρωμένο παράδειγμα web εφαρμογής που συνδυάζει frontend, backend, βάση δεδομένων, εξωτερικά APIs, μηχανισμό προτιμήσεων και διαχείριση χρηστών. Η αξία της λύσης δεν βρίσκεται μόνο στην τελική οθόνη του χάρτη, αλλά και στον τρόπο με τον οποίο συνδέονται τα επιμέρους υποσυστήματα.

Η υλοποίηση αναδεικνύει επίσης πώς μπορεί να ενσωματωθεί τεχνητή νοημοσύνη σε μια εφαρμογή χωρίς να αναλάβει όλο το βάρος της λογικής απόφασης. Το Gemini χρησιμοποιείται για ερμηνεία και εμπλουτισμό αιτημάτων, ενώ το fuzzy μοντέλο παραμένει ελεγχόμενος μηχανισμός επιλογής αποθηκευμένης προτίμησης. Αυτός ο διαχωρισμός ρόλων κάνει την αρχιτεκτονική πιο κατανοητή και πιο εύκολη στην τεκμηρίωση.

Για τη διπλωματική εργασία, η εφαρμογή έχει και εκπαιδευτικό όφελος. Επιτρέπει την παρουσίαση πολλών γνωστικών περιοχών σε ένα ενιαίο παράδειγμα: requirements engineering, UX, HCI, συστήματα προτάσεων, fuzzy decision-making, APIs χαρτών, web development και σχεδιασμό βάσης δεδομένων. Με αυτόν τον τρόπο, η λύση λειτουργεί όχι μόνο ως εφαρμογή, αλλά και ως τεκμηριωμένο έργο ανάπτυξης λογισμικού.

11.8 Περιορισμοί στην αποτίμηση των οφελών

Παρότι τα αναμενόμενα οφέλη είναι σαφή, πρέπει να επισημανθούν και οι περιορισμοί τους. Η εργασία δεν περιλαμβάνει εκτεταμένη αξιολόγηση με μεγάλο αριθμό πραγματικών χρηστών, ούτε συγκριτική μέτρηση απέναντι σε εμπορικές εφαρμογές πλοήγησης σε παραγωγικές συνθήκες. Επομένως, τα οφέλη τεκμηριώνονται κυρίως από τη σχεδίαση, την υλοποίηση και τη βιβλιογραφική σύνδεση.

Επιπλέον, μέρος της εμπειρίας εξαρτάται από εξωτερικές υπηρεσίες, όπως APIs χαρτών, υπηρεσίες τοποθεσίας browser και υπηρεσίες τεχνητής νοημοσύνης. Αν κάποια από αυτές τις υπηρεσίες δεν αποκρίνεται ή έχει περιορισμούς, η συνολική εμπειρία του χρήστη μπορεί να επηρεαστεί. Αυτό δεν ακυρώνει την προτεινόμενη λύση, αλλά δείχνει ότι η αξιοπιστία της εξαρτάται και από υποδομές εκτός της ίδιας εφαρμογής.

Τέλος, η πραγματική αξία της εξατομίκευσης θα μπορούσε να εκτιμηθεί καλύτερα με μακροχρόνια χρήση. Όσο περισσότερο ιστορικό διαθέτει η εφαρμογή, τόσο καλύτερα μπορεί να φανεί αν οι προτεινόμενες προτιμήσεις ανταποκρίνονται στις ανάγκες του χρήστη. Αυτό αποτελεί φυσικό σημείο συνέχειας για την αξιολόγηση και τις μελλοντικές επεκτάσεις της εργασίας.

Κεφάλαιο 12: Αξιολόγηση και Ασφάλεια

Το κεφάλαιο αυτό αξιολογεί τη SmartNavix ως ολοκληρωμένο πρωτότυπο και εξετάζει τα βασικά ζητήματα ασφάλειας και προστασίας δεδομένων. Η αξιολόγηση στηρίζεται στις απαιτήσεις, στις υλοποιημένες ροές, στη δομή της εφαρμογής και στη σχετική βιβλιογραφία, χωρίς να παρουσιάζει ανύπαρκτες μετρήσεις μεγάλης κλίμακας.

12.1 Σκοπός της αξιολόγησης

Η αξιολόγηση της SmartNavix έχει ως στόχο να εξετάσει αν η εφαρμογή ανταποκρίνεται στον σκοπό για τον οποίο σχεδιάστηκε: να υποστηρίξει την επιλογή διαδρομής με βάση τις προτιμήσεις του χρήστη και όχι μόνο με βάση ένα απλό γεωγραφικό κριτήριο. Επειδή η εργασία αφορά εκπαιδευτική/ερευνητική υλοποίηση, η αξιολόγηση δεν παρουσιάζεται ως παραγωγική μέτρηση μεγάλης κλίμακας, αλλά ως τεκμηριωμένος έλεγχος λειτουργιών, ροών, εμπειρίας χρήστη και βασικών ζητημάτων ασφάλειας.

Η προσέγγιση συνδέεται με την ανάλυση απαιτήσεων που προηγήθηκε. Οι απαιτήσεις δεν έχουν αξία μόνο ως αρχική καταγραφή, αλλά λειτουργούν και ως σημείο ελέγχου για το αν το σύστημα υλοποιεί όσα υποσχέθηκε [5]. Στην περίπτωση της SmartNavix, αυτό σημαίνει ότι η αξιολόγηση πρέπει να καλύπτει τον απλό χρήστη, τον contributor, τον διαχειριστή, τις λειτουργίες πλοήγησης, τα φίλτρα, τα presets, την αποθήκευση ιστορικού και τη διαχείριση ρόλων.

Παράλληλα, η αξιολόγηση πρέπει να είναι ειλικρινής ως προς τα όρια της εφαρμογής. Η SmartNavix δοκιμάστηκε ως ολοκληρωμένο πρωτότυπο, όμως δεν αντικαθιστά μια εμπορική εφαρμογή πλοήγησης ούτε έχει δοκιμαστεί σε μεγάλο αριθμό πραγματικών μετακινήσεων. Επομένως, το κεφάλαιο αυτό εστιάζει στο αν η υλοποίηση είναι συνεπής, κατανοητή, λειτουργική και επεκτάσιμη.

12.2 Κριτήρια αξιολόγησης

Τα κριτήρια αξιολόγησης οργανώνονται γύρω από τέσσερις άξονες: λειτουργικότητα, εμπειρία χρήστη, τεχνική αξιοπιστία και ασφάλεια. Η λειτουργικότητα αφορά το αν οι βασικές εργασίες εκτελούνται σωστά, όπως σύνδεση, αναζήτηση διαδρομής, εφαρμογή φίλτρων, δημιουργία preset και διαχείριση λογαριασμού. Η εμπειρία χρήστη αφορά το αν οι ενέργειες αυτές είναι κατανοητές και αν ο χρήστης μπορεί να προβλέψει τι θα συμβεί μετά από κάθε επιλογή.

Η ευχρηστία συνδέεται με την αποτελεσματικότητα, την αποδοτικότητα και την ικανοποίηση του χρήστη στο συγκεκριμένο πλαίσιο χρήσης [31]. Στη SmartNavix, αυτό μεταφράζεται σε απλή πρόσβαση στον χάρτη, σαφή φίλτρα, ορατή ανατροφοδότηση, περιορισμό περιττών βημάτων και δυνατότητα αλλαγής επιλογών χωρίς να χάνεται ο έλεγχος της ροής. Η αξιολόγηση δεν πρέπει να κοιτά μόνο αν ένα κουμπί λειτουργεί, αλλά αν η συνολική διαδικασία παραμένει λογική για τον χρήστη.

Ο ανθρωποκεντρικός σχεδιασμός απαιτεί κατανόηση χρηστών, εργασιών και περιβάλλοντος χρήσης [32]. Για αυτόν τον λόγο, η αξιολόγηση της SmartNavix εξετάζει αν το σύστημα υποστηρίζει διαφορετικά επίπεδα πρόσβασης, διαφορετικές ανάγκες διαδρομής και διαφορετικούς τρόπους διαχείρισης προτιμήσεων. Έτσι, το αποτέλεσμα δεν περιορίζεται σε έναν τεχνικό έλεγχο, αλλά συνδέεται με το πώς η εφαρμογή μπορεί να χρησιμοποιηθεί στην πράξη.

12.3 Λειτουργική αξιολόγηση βασικών ροών

Η πρώτη βασική ροή είναι η πρόσβαση στην εφαρμογή. Ο χρήστης πρέπει να μπορεί να συνδεθεί, να δει την αρχική οθόνη και να χρησιμοποιήσει τις βασικές λειτουργίες χάρτη. Η ύπαρξη ρόλων επιτρέπει διαφορετική εμπειρία ανάλογα με το αν ο λογαριασμός είναι user, contributor ή admin. Αυτό είναι σημαντικό, επειδή η εφαρμογή δεν έχει μόνο μία κατηγορία αλληλεπίδρασης, αλλά συνδυάζει τελικό χρήστη και διαχειριστικές δυνατότητες.

Η δεύτερη ροή αφορά την επιλογή διαδρομής. Ο χρήστης εισάγει προορισμό, δηλώνει ή εφαρμόζει προτιμήσεις και λαμβάνει αποτέλεσμα που περιλαμβάνει χάρτη, απόσταση, χρόνο και αιτιολόγηση. Η

ροή είναι συμβατή με τον κεντρικό στόχο της εργασίας, επειδή δεν παρουσιάζει τη διαδρομή ως απλή απάντηση χάρτη, αλλά ως απόφαση που επηρεάζεται από τα δεδομένα προτίμησης.

Η τρίτη ροή αφορά την αποθήκευση και επαναχρησιμοποίηση επιλογών. Τα presets, οι αποθηκευμένες προτιμήσεις και το ιστορικό βοηθούν τον χρήστη να μη ρυθμίζει κάθε φορά από την αρχή το ίδιο πλαίσιο διαδρομής. Στο σημείο αυτό η εφαρμογή συνδέεται με τη λογική των συστημάτων προτάσεων, όπου η πληροφορία οργανώνεται ώστε να υποστηρίζεται καλύτερη επιλογή ανάμεσα σε εναλλακτικές [51], [52].

12.4 Αξιολόγηση εμπειρίας χρήστη

Η εμπειρία χρήστη της SmartNavix στηρίζεται σε γνώριμα στοιχεία: χάρτης, πεδίο αναζήτησης, φίλτρα, ρυθμίσεις και προφίλ. Αυτή η επιλογή μειώνει το γνωστικό φορτίο, επειδή ο χρήστης δεν χρειάζεται να μάθει από την αρχή έναν εντελώς νέο τρόπο πλοήγησης. Η διαφοροποίηση εμφανίζεται κυρίως στο επίπεδο της εξατομίκευσης, όπου οι προτιμήσεις αποκτούν μεγαλύτερο ρόλο στη διαμόρφωση του αποτελέσματος.

Θετικό στοιχείο είναι ότι η εφαρμογή προσφέρει άμεση ανατροφοδότηση. Ο χρήστης βλέπει την προτεινόμενη διαδρομή και μπορεί να αξιολογήσει το αποτέλεσμα. Η δυνατότητα αυτή βοηθά ώστε η χρήση να μην είναι μονόδρομη: το σύστημα δεν παράγει απλώς μια απάντηση, αλλά δέχεται σήμα από τον χρήστη για το αν η απάντηση ήταν χρήσιμη. Μελλοντικά, η ανατροφοδότηση αυτή μπορεί να αξιοποιηθεί πιο συστηματικά για βελτίωση των προτάσεων.

Υπάρχουν όμως και σημεία που χρειάζονται προσοχή. Όσο αυξάνονται τα φίλτρα και οι δυνατότητες, υπάρχει κίνδυνος η διεπαφή να γίνει πιο σύνθετη από όσο χρειάζεται. Για τον λόγο αυτό, η SmartNavix πρέπει να διατηρεί καθαρή ιεράρχηση επιλογών και να αποφεύγει την υπερφόρτωση της βασικής οθόνης. Η καλή εμπειρία χρήστη δεν προκύπτει από την ποσότητα λειτουργιών, αλλά από το αν αυτές εμφανίζονται την κατάλληλη στιγμή και με σαφή τρόπο [29].

12.5 Αξιολόγηση προσαρμοστικότητας και προτάσεων

Η προσαρμοστικότητα της SmartNavix εκφράζεται κυρίως μέσα από την αποθήκευση προτιμήσεων, την επαναχρησιμοποίηση presets και την επιλογή best preference. Το σύστημα επιχειρεί να αναγνωρίσει ποια αποθηκευμένη προτίμηση είναι πιθανότερο να ταιριάζει στον χρήστη, χωρίς να αφαιρεί από τον ίδιο τη δυνατότητα αλλαγής. Αυτή η ισορροπία είναι σημαντική, επειδή μια προσαρμοστική εφαρμογή πρέπει να βοηθά χωρίς να επιβάλλει.

Η χρήση fuzzy logic προσφέρει πλεονέκτημα στην αξιολόγηση, διότι η επιλογή κατάλληλης προτίμησης δεν είναι πάντα δυαδικό πρόβλημα. Μια προτίμηση μπορεί να είναι σχετικά πρόσφατη, σχετικά πλήρης ή σχετικά συχνή. Η ασαφής λογική επιτρέπει τέτοιες ενδιάμεσες καταστάσεις, επειδή βασίζεται σε βαθμούς συμμετοχής και όχι μόνο σε απόλυτες κατηγορίες [56].

Παρόλα αυτά, η ποιότητα της πρότασης εξαρτάται από την ποιότητα των δεδομένων. Αν ο χρήστης έχει ελάχιστο ιστορικό ή αν οι αποθηκευμένες προτιμήσεις του δεν αποτυπώνουν πραγματικές ανάγκες, το αποτέλεσμα μπορεί να είναι περιορισμένο. Επομένως, η αξιολόγηση της προσαρμοστικότητας δείχνει ότι η αρχιτεκτονική είναι κατάλληλη, αλλά η πραγματική αποτελεσματικότητα χρειάζεται περισσότερη χρήση και δεδομένα.

12.6 Τεχνική αξιολόγηση και περιορισμοί

Τεχνικά, η SmartNavix συνδυάζει Angular frontend, .NET backend, SQL Server βάση δεδομένων και εξωτερικές υπηρεσίες για χάρτες και τεχνητή νοημοσύνη. Η αρχιτεκτονική αυτή είναι επαρκής για την ανάπτυξη ενός ολοκληρωμένου πρωτοτύπου, επειδή διαχωρίζει τη διεπαφή από τη λογική του server και από την αποθήκευση δεδομένων. Ο διαχωρισμός αυτός βοηθά στη συντήρηση, στη δοκιμή και στη μελλοντική επέκταση.

Οι βασικοί περιορισμοί προκύπτουν από την εξάρτηση από εξωτερικά APIs. Η απόκριση της εφαρμογής επηρεάζεται από την ποιότητα των υπηρεσιών χαρτών, τη διαθεσιμότητα του browser geolocation, την ακρίβεια των δεδομένων τοποθεσίας και τυχόν περιορισμούς χρήσης. Για αυτόν τον

λόγο, η εφαρμογή πρέπει να χειρίζεται αποτυχίες με κατανοητά μηνύματα και να μην αφήνει τον χρήστη χωρίς ανατροφοδότηση.

Ένας ακόμη περιορισμός είναι ότι η υλοποίηση δεν έχει υποβληθεί σε φορτίο παραγωγικής κλίμακας. Δεν υπάρχουν μετρήσεις για μεγάλο αριθμό ταυτόχρονων χρηστών, μακροχρόνια καταγραφή επιδόσεων ή δοκιμές σε πολλές διαφορετικές συσκευές. Αυτό δεν μειώνει την αξία της διπλωματικής υλοποίησης, αλλά ορίζει καθαρά το στάδιο ωριμότητας του συστήματος.

12.7 Ασφάλεια ταυτοποίησης, ρόλων και πρόσβασης

Η ασφάλεια της SmartNavix ξεκινά από την ταυτοποίηση χρήστη και τον διαχωρισμό ρόλων. Η εφαρμογή διαθέτει διαφορετικές δυνατότητες για user, contributor και admin, ενώ η πρόσβαση σε προστατευμένες οθόνες ελέγχεται από guards στο frontend και αντίστοιχους ελέγχους στο backend. Η ύπαρξη αυτών των μηχανισμών είναι απαραίτητη, επειδή η αλλαγή ρόλων, η διαχείριση χρηστών και η πρόσβαση σε στατιστικά δεν πρέπει να είναι διαθέσιμες σε όλους.

Σύμφωνα με τις συνηθισμένες κατηγορίες κινδύνων σε web εφαρμογές, ο ανεπαρκής έλεγχος πρόσβασης αποτελεί από τα πιο κρίσιμα σημεία [76]. Για τη SmartNavix, αυτό σημαίνει ότι κάθε διαχειριστική λειτουργία πρέπει να ελέγχεται από την πλευρά του server και όχι μόνο από το αν εμφανίζεται ή κρύβεται ένα κουμπί στη διεπαφή. Η διεπαφή βοηθά την εμπειρία, αλλά η πραγματική ασφάλεια πρέπει να επιβεβαιώνεται στο backend.

Η χρήση ασφαλούς χειρισμού κωδικών πρόσβασης και η ύπαρξη λειτουργιών επιβεβαίωσης ή επαναφοράς λογαριασμού ενισχύουν την αξιοπιστία του συστήματος. Ωστόσο, για παραγωγικό περιβάλλον θα έπρεπε να εξεταστούν πρόσθετα μέτρα, όπως αυστηρότερη πολιτική κωδικών, περιορισμός προσπαθειών σύνδεσης, ασφαλέστερη διαχείριση tokens και πιο πλήρης καταγραφή ύποπτων ενεργειών.

12.8 Προστασία προσωπικών δεδομένων και ιδιωτικότητα

Η SmartNavix χειρίζεται δεδομένα που μπορούν να συνδεθούν με τον χρήστη, όπως λογαριασμός, προτιμήσεις, αποθηκευμένες επιλογές και ιστορικό διαδρομών. Για αυτόν τον λόγο, η προστασία προσωπικών δεδομένων δεν είναι δευτερεύον θέμα, αλλά μέρος της συνολικής ποιότητας της λύσης. Ο Γενικός Κανονισμός Προστασίας Δεδομένων θέτει αρχές όπως νομιμότητα, διαφάνεια, περιορισμό σκοπού και ελαχιστοποίηση δεδομένων [75].

Στο πλαίσιο της εφαρμογής, η ελαχιστοποίηση σημαίνει ότι πρέπει να συλλέγονται μόνο τα δεδομένα που είναι αναγκαία για τη λειτουργία της πλοήγησης και της εξατομίκευσης. Η ύπαρξη λειτουργιών όπως εξαγωγή ή διαγραφή δεδομένων κινείται προς αυτή την κατεύθυνση, επειδή δίνει στον χρήστη μεγαλύτερο έλεγχο πάνω στο περιεχόμενο που τον αφορά. Η διαφάνεια ενισχύεται όταν ο χρήστης καταλαβαίνει γιατί αποθηκεύεται μια προτίμηση και πώς επηρεάζει την επόμενη πρόταση.

Ιδιαίτερη προσοχή χρειάζεται και στις εξωτερικές υπηρεσίες. Όταν μια εφαρμογή στέλνει στοιχεία σε APIs χαρτών ή τεχνητής νοημοσύνης, πρέπει να είναι σαφές ποια δεδομένα μεταφέρονται και για ποιον σκοπό. Σε μελλοντική παραγωγική έκδοση θα ήταν σκόπιμο να υπάρχει πλήρης πολιτική απορρήτου, μηχανισμός συγκατάθεσης όπου απαιτείται και τεχνικός περιορισμός των δεδομένων που αποστέλλονται σε τρίτες υπηρεσίες.

12.9 Συνολική αποτίμηση

Η συνολική αποτίμηση δείχνει ότι η SmartNavix ανταποκρίνεται στον βασικό στόχο της διπλωματικής εργασίας: παρουσιάζει μια εφαρμογή πλοήγησης που ενσωματώνει προτιμήσεις, εξατομίκευση, προτάσεις και διαχείριση χρηστών σε ένα ενιαίο περιβάλλον. Η λύση είναι συνεπής με τις απαιτήσεις που καταγράφηκαν και διαθέτει λειτουργίες που υποστηρίζουν τόσο τον τελικό χρήστη όσο και τον διαχειριστή.

Η αξιολόγηση αναδεικνύει επίσης ότι η εφαρμογή βρίσκεται σε στάδιο λειτουργικού πρωτοτύπου. Υπάρχει καθαρή αρχιτεκτονική και ολοκληρωμένη βασική ροή, αλλά χρειάζονται επιπλέον δοκιμές για

παραγωγική χρήση: μεγαλύτερο δείγμα χρηστών, συστηματικές μετρήσεις ευχρηστίας, δοκιμές επιδόσεων, έλεγχος ασφάλειας και πιο αυστηρή διαχείριση εξωτερικών APIs.

Με αυτή την έννοια, το κεφάλαιο δεν παρουσιάζει την SmartNavix ως οριστικά ολοκληρωμένο εμπορικό προϊόν. Την παρουσιάζει ως τεκμηριωμένη λύση που υλοποιεί τον ερευνητικό και τεχνικό στόχο της εργασίας, ενώ ταυτόχρονα δείχνει καθαρά ποια σημεία πρέπει να ενισχυθούν σε επόμενη φάση ανάπτυξης.

Κεφάλαιο 13: Συμπεράσματα και Μελλοντικές Επεκτάσεις

Το κεφάλαιο αυτό συνοψίζει τα βασικά συμπεράσματα της διπλωματικής εργασίας, αποτιμά τη συνεισφορά της SmartNavix, καταγράφει τους περιορισμούς και παρουσιάζει πιθανές μελλοντικές επεκτάσεις. Η αποτίμηση γίνεται με βάση τη βιβλιογραφική θεμελίωση, τις απαιτήσεις, την υλοποίηση και την αξιολόγηση που παρουσιάστηκαν στα προηγούμενα κεφάλαια.

13.1 Σύνοψη της διπλωματικής εργασίας

Η παρούσα διπλωματική εργασία είχε ως αντικείμενο τη μελέτη, τον σχεδιασμό και την ανάπτυξη της SmartNavix, μιας web-based εφαρμογής πλοήγησης που αξιοποιεί προτιμήσεις χρήστη, εξωτερικές υπηρεσίες χαρτών, τεχνητή νοημοσύνη και fuzzy logic για να υποστηρίξει πιο εξατομικευμένη επιλογή διαδρομής. Η εργασία ξεκίνησε από το πρόβλημα ότι οι συνηθισμένες εφαρμογές πλοήγησης συχνά προτεραιοποιούν χρόνο, απόσταση ή κυκλοφοριακά δεδομένα, ενώ οι πραγματικές ανάγκες του χρήστη μπορεί να είναι πιο σύνθετες.

Στα πρώτα κεφάλαια παρουσιάστηκε το θεωρητικό και τεχνολογικό πλαίσιο της εργασίας. Εξετάστηκαν ολοκληρωμένα συστήματα, web και mobile εφαρμογές, smart και intelligent systems, τεχνητή νοημοσύνη, user experience, προσαρμοστικότητα, αλληλεπίδραση ανθρώπου-υπολογιστή, διεπαφές χρήστη, συστήματα πλοήγησης, recommender systems και fuzzy logic. Η βιβλιογραφική ανασκόπηση δεν λειτούργησε ως απλή παράθεση πηγών, αλλά ως βάση για να εξηγηθεί γιατί μια εφαρμογή όπως η SmartNavix έχει νόημα ως τεχνική και ερευνητική πρόταση.

Στη συνέχεια αναλύθηκαν οι απαιτήσεις, η αρχιτεκτονική, η βάση δεδομένων, οι βασικές λειτουργίες, το fuzzy μοντέλο, τα παραδείγματα χρήσης, το εγχειρίδιο χρήστη, τα αναμενόμενα οφέλη και η αξιολόγηση της λύσης. Με αυτόν τον τρόπο, η εργασία κάλυψε ολόκληρη τη διαδρομή από την αρχική ιδέα μέχρι την υλοποιημένη εφαρμογή και την κριτική αποτίμησή της.

13.2 Βασικά συμπεράσματα

Το πρώτο συμπέρασμα είναι ότι η επιλογή διαδρομής μπορεί να αντιμετωπιστεί ως πρόβλημα απόφασης με περισσότερα από ένα κριτήρια. Η SmartNavix δείχνει ότι ο χρήστης δεν ενδιαφέρεται πάντοτε μόνο για τη γρηγορότερη διαδρομή. Μπορεί να λαμβάνει υπόψη σημεία ενδιαφέροντος, στάσεις, προσωπικές συνήθειες, αποφυγές, διάθεση, μεταφορικό μέσο ή άλλους παράγοντες που δύσκολα χωρούν σε ένα μόνο αριθμητικό κριτήριο.

Το δεύτερο συμπέρασμα είναι ότι η εξατομίκευση έχει πραγματική αξία όταν παραμένει κατανοητή και ελεγχόμενη από τον χρήστη. Η SmartNavix δεν στηρίζεται αποκλειστικά σε μια αδιαφανή απόφαση τεχνητής νοημοσύνης, αλλά συνδυάζει αποθηκευμένες προτιμήσεις, presets, ιστορικό και fuzzy score. Αυτή η προσέγγιση είναι πιο ερμηνεύσιμη, επειδή επιτρέπει να περιγραφεί πώς επηρεάζεται η τελική επιλογή.

Το τρίτο συμπέρασμα αφορά την ίδια την ανάπτυξη λογισμικού. Μια εφαρμογή σαν τη SmartNavix δεν είναι μόνο χάρτης και αναζήτηση. Χρειάζεται σωστή ανάλυση απαιτήσεων, ρόλους χρηστών, βάση δεδομένων, ελέγχους πρόσβασης, διαχείριση εξωτερικών υπηρεσιών, χειρισμό σφαλμάτων και σαφή διεπαφή. Η συστηματική ανάλυση απαιτήσεων λειτουργεί εδώ ως πρακτικό εργαλείο οργάνωσης της υλοποίησης [5].

13.3 Συνεισφορά της SmartNavix

Η βασική συνεισφορά της SmartNavix είναι ότι προτείνει μια πιο ευέλικτη προσέγγιση στην πλοήγηση. Αντί να θεωρεί τη διαδρομή ως στατικό αποτέλεσμα, τη συνδέει με το προφίλ, τις επιλογές και το ιστορικό του χρήστη. Έτσι, η εφαρμογή κινείται προς τη λογική των adaptive systems, όπου η συμπεριφορά του συστήματος μπορεί να μεταβάλλεται με βάση τον χρήστη και το πλαίσιο χρήσης [34], [35].

Σε επίπεδο σχεδιασμού, η SmartNavix συνδυάζει διαφορετικές τεχνολογίες σε ένα συνεκτικό σύστημα: Angular frontend, .NET backend, SQL Server, Google Maps API, Gemini API, μηχανισμό προτιμήσεων και admin panel. Η συνεισφορά δεν βρίσκεται μόνο στο ότι αυτές οι τεχνολογίες συνυπάρχουν, αλλά στο ότι οργανώνονται γύρω από μια συγκεκριμένη λειτουργική ιδέα: την προτεινόμενη διαδρομή με βάση τις προτιμήσεις.

Σε ερευνητικό επίπεδο, η εργασία δείχνει πώς έννοιες από UX, HCI, recommender systems και fuzzy decision-making μπορούν να συνδεθούν σε μια εφαρμογή καθημερινής χρήσης. Η SmartNavix λειτουργεί έτσι ως παράδειγμα εφαρμοσμένης ανάπτυξης λογισμικού, όπου η θεωρία δεν μένει αποκομμένη από την υλοποίηση, αλλά επηρεάζει άμεσα τον τρόπο σχεδιασμού του συστήματος.

13.4 Περιορισμοί της εργασίας

Παρότι η SmartNavix καλύπτει τον βασικό στόχο της διπλωματικής, υπάρχουν περιορισμοί που πρέπει να αναφερθούν με σαφήνεια. Πρώτον, η εφαρμογή αποτελεί λειτουργικό πρωτότυπο και όχι παραγωγικό εμπορικό προϊόν. Αυτό σημαίνει ότι ορισμένα σημεία, όπως η πλήρης σκλήρυνση ασφάλειας, η διαχείριση μεγάλου φόρτου και η μακροχρόνια παρακολούθηση επιδόσεων, απαιτούν επιπλέον εργασία.

Δεύτερον, η αξιολόγηση δεν έγινε με μεγάλο αριθμό χρηστών ούτε με μακροχρόνια χρήση σε πραγματικές καθημερινές μετακινήσεις. Η αξία της εξατομίκευσης και του fuzzy μοντέλου μπορεί να εκτιμηθεί καλύτερα όταν υπάρχουν περισσότερα δεδομένα χρήσης, περισσότερες αξιολογήσεις διαδρομών και μεγαλύτερη ποικιλία σε πραγματικές προτιμήσεις. Χωρίς αυτά, η αποτίμηση παραμένει τεκμηριωμένη αλλά περιορισμένη.

Τρίτον, η εφαρμογή εξαρτάται από εξωτερικές υπηρεσίες. Χάρτες, τοποθεσία browser και υπηρεσίες τεχνητής νοημοσύνης μπορούν να επηρεάσουν τη διαθεσιμότητα, την ακρίβεια και την εμπειρία χρήστη. Η εξάρτηση αυτή είναι φυσιολογική για σύγχρονες web εφαρμογές, αλλά πρέπει να λαμβάνεται υπόψη στον σχεδιασμό αξιοπιστίας, κόστους και προστασίας δεδομένων.

13.5 Προβλήματα και δυσκολίες που συναντήθηκαν

Μία βασική δυσκολία ήταν ο συνδυασμός πολλών διαφορετικών λειτουργιών σε μια ενιαία εφαρμογή. Η SmartNavix δεν περιορίζεται σε απλή εμφάνιση χάρτη, αλλά περιλαμβάνει login, ρόλους, προφίλ, ρυθμίσεις, προτιμήσεις, presets, admin panel, analytics, εξωτερικά APIs και μηχανισμό επιλογής best preference. Η ενοποίηση αυτών των στοιχείων απαιτούσε προσεκτικό σχεδιασμό ώστε η εφαρμογή να μην γίνει ασύνδετη ή δυσνόητη.

Δεύτερη δυσκολία ήταν η ισορροπία ανάμεσα στην τεχνητή νοημοσύνη και σε ελεγχόμενη λογική απόφασης. Η χρήση ΑΙ μπορεί να προσφέρει ευελιξία, αλλά δεν είναι πάντα εύκολο να εξηγηθεί ή να προβλεφθεί. Για αυτόν τον λόγο, η εργασία διατήρησε το fuzzy μοντέλο ως πιο ερμηνεύσιμο μηχανισμό επιλογής προτίμησης, ενώ η τεχνητή νοημοσύνη αξιοποιείται συμπληρωματικά.

Τρίτη δυσκολία ήταν η τεκμηρίωση της εφαρμογής με τρόπο που να είναι ταυτόχρονα τεχνικός και κατανοητός. Η εργασία έπρεπε να περιγράψει κώδικα, βάση δεδομένων, ροές, screenshots, απαιτήσεις και θεωρητικό υπόβαθρο χωρίς να καταλιξεί σε απλή λίστα λειτουργιών. Αυτό απαιτήσε συνεχή σύνδεση ανάμεσα στο τι υλοποιήθηκε και στο γιατί υλοποιήθηκε με αυτόν τον τρόπο.

13.6 Μελλοντικές επεκτάσεις

Μια πρώτη μελλοντική επέκταση είναι η συστηματική αξιολόγηση με πραγματικούς χρήστες. Θα μπορούσε να σχεδιαστεί μελέτη ευχρηστίας με συγκεκριμένα σενάρια, χρόνο ολοκλήρωσης εργασιών, ποσοστό επιτυχίας, ερωτηματολόγια ικανοποίησης και ποιοτικά σχόλια. Η ευχρηστία πρέπει να εξετάζεται στο πραγματικό πλαίσιο χρήσης και όχι μόνο ως γενική αίσθηση της διεπαφής [31], [32].

Μια δεύτερη επέκταση αφορά την ενίσχυση του μηχανισμού προτάσεων. Το fuzzy μοντέλο μπορεί να εμπλουτιστεί με περισσότερα κριτήρια, δυναμική προσαρμογή βαρών και αξιοποίηση feedback από προηγούμενες διαδρομές. Επιπλέον, θα μπορούσε να συνδυαστεί με τεχνικές recommendation, ώστε η εφαρμογή να μαθαίνει καλύτερα από τη συμπεριφορά του χρήστη χωρίς να χάνει την ερμηνευσιμότητά της [51], [52].

Μια τρίτη επέκταση είναι η μεταφορά περισσότερων εξωτερικών κλήσεων στο backend, ώστε να προστατεύονται καλύτερα τα API keys και να υπάρχει κεντρικότερος έλεγχος ασφάλειας. Παράλληλα, θα μπορούσαν να προστεθούν μηχανισμοί rate limiting, πληρέστερο logging, καλύτερη διαχείριση tokens και πολιτικές προστασίας προσωπικών δεδομένων σύμφωνα με τις αρχές του GDPR [75] και τις πρακτικές ασφαλούς ανάπτυξης web εφαρμογών [76].

13.7 Τελική αποτίμηση

Συνολικά, η SmartNavix δείχνει ότι η πλοήγηση μπορεί να εμπλουτιστεί με εξατομίκευση χωρίς να χαθεί η απλότητα της βασικής εμπειρίας. Ο χρήστης εξακολουθεί να έχει έναν χάρτη, έναν προορισμό και μια διαδρομή, αλλά πίσω από αυτή τη φαινομενικά απλή ροή υπάρχει ένα σύστημα που λαμβάνει υπόψη προτιμήσεις, ιστορικό, ρόλους, δεδομένα και κανόνες απόφασης.

Η εργασία επιβεβαιώνει επίσης ότι η ανάπτυξη μιας τέτοιας εφαρμογής απαιτεί συνδυασμό θεωρίας και πράξης. Οι έννοιες της προσαρμοστικότητας, της ευχρηστίας, της αλληλεπίδρασης ανθρώπου-υπολογιστή και των συστημάτων προτάσεων αποκτούν πρακτικό νόημα μόνο όταν μεταφράζονται σε συγκεκριμένες οθόνες, ροές, πίνακες βάσης δεδομένων και ελέγχους πρόσβασης.

Η SmartNavix δεν παρουσιάζεται ως τελικό προϊόν χωρίς αδυναμίες, αλλά ως ολοκληρωμένη πρόταση που μπορεί να επεκταθεί. Η αξία της βρίσκεται στο ότι συνδέει ένα καθημερινό πρόβλημα, την επιλογή διαδρομής, με σύγχρονες τεχνολογίες και με μια πιο ανθρώπινη αντίληψη της πλοήγησης: όχι μόνο πώς φτάνει κάποιος στον προορισμό του, αλλά με ποιον τρόπο θέλει να φτάσει εκεί.

Πίνακας Ορολογίας

Ο πίνακας ορολογίας συγκεντρώνει βασικούς όρους που χρησιμοποιούνται στην εργασία. Η καταγραφή τους βοηθά ώστε η ανάγνωση να παραμένει συνεπής, ειδικά επειδή η SmartNavix συνδυάζει έννοιες από ανάπτυξη λογισμικού, εμπειρία χρήστη, συστήματα προτάσεων και fuzzy logic.

ΠΙΝΑΚΑΣ Ο.1: Πίνακας ορολογίας SmartNavix

Όρος	Αγγλικός όρος	Επεξήγηση
Προσαρμοστικότητα	Adaptivity	Ικανότητα μιας εφαρμογής να μεταβάλλει τη συμπεριφορά ή την παρουσίαση των επιλογών με βάση στοιχεία χρήστη, ιστορικού ή περιβάλλοντος.
Προσαρμοστικό σύστημα	Adaptive system	Σύστημα που αξιοποιεί πληροφορίες χρήσης ή προφίλ ώστε να προσφέρει πιο κατάλληλη απόκριση σε διαφορετικούς χρήστες.
Εξατομίκευση	Personalization	Προσαρμογή λειτουργιών, περιεχομένου ή προτάσεων με βάση τις ανάγκες και τις προτιμήσεις συγκεκριμένου χρήστη.
Σύστημα προτάσεων	Recommender system	Σύστημα που οργανώνει πληροφορία και προτείνει επιλογές οι οποίες εκτιμάται ότι ταιριάζουν στον χρήστη.
Ασαφής λογική	Fuzzy logic	Μέθοδος μοντελοποίησης που επιτρέπει ενδιάμεσους βαθμούς συμμετοχής αντί για αυστηρά δυαδικές αποφάσεις.
Συνάρτηση συμμετοχής	Membership function	Συνάρτηση που αποδίδει σε μια τιμή βαθμό συμμετοχής σε ασαφή κατηγορία, όπως low, medium ή high.
Αποασαφοποίηση	Defuzzification	Διαδικασία μετατροπής ασαφών αποτελεσμάτων σε τελική αριθμητική τιμή ή απόφαση.
Εμπειρία χρήστη	User experience	Συνολική αντίληψη του χρήστη για τη χρησιμότητα, σαφήνεια, ευχρηστία και καταλληλότητα μιας εφαρμογής.
Ευχρηστία	Usability	Βαθμός στον οποίο ένας χρήστης μπορεί να πετύχει τον στόχο του αποτελεσματικά, αποδοτικά και με ικανοποίηση.
Αλληλεπίδραση ανθρώπου-υπολογιστή	Human-computer interaction	Πεδίο που εξετάζει τον σχεδιασμό και την αξιολόγηση της αλληλεπίδρασης ανάμεσα σε χρήστες και υπολογιστικά συστήματα.
Διεπαφή χρήστη	User interface	Το οπτικό και λειτουργικό επίπεδο μέσω του οποίου ο χρήστης αλληλεπιδρά με την εφαρμογή.
Προτίμηση χρήστη	User preference	Επιλογή ή χαρακτηριστικό που εκφράζει το τι θεωρεί κατάλληλο ο χρήστης για μια διαδρομή.
Preset	Preset	Αποθηκευμένο σύνολο ρυθμίσεων που μπορεί να επαναχρησιμοποιηθεί από τον χρήστη.
Σημείο ενδιαφέροντος	Point of interest	Τοποθεσία που μπορεί να έχει αξία για τον χρήστη, όπως στάση, υπηρεσία, κατάστημα ή σημείο φόρτισης.
Ρόλος χρήστη	User role	Κατηγορία πρόσβασης που καθορίζει ποιες λειτουργίες μπορεί να εκτελέσει ένας χρήστης.
Πρωτότυπο	Prototype	Λειτουργική υλοποίηση που αποδεικνύει την ιδέα και τις βασικές ροές, χωρίς να αποτελεί απαραίτητα τελικό παραγωγικό προϊόν.

Συντμήσεις - Αρκτικόλεξα - Ακρωνύμια

Ο πίνακας αυτός παρουσιάζει τις βασικές συντμήσεις και τα ακρωνύμια που εμφανίζονται στην εργασία. Όπου χρειάζεται, η επεξήγηση αποδίδεται στα ελληνικά ώστε να είναι σαφής ο ρόλος κάθε τεχνικού όρου μέσα στο κείμενο.

ΠΙΝΑΚΑΣ Σ.1: Συντμήσεις, αρκτικόλεξα και ακρωνύμια

Σύντμηση	Πλήρης μορφή	Επεξήγηση
AI	Artificial Intelligence	Τεχνητή νοημοσύνη, δηλαδή τεχνικές που επιτρέπουν σε συστήματα να εκτελούν εργασίες που σχετίζονται με αντίληψη, απόφαση ή παραγωγή απάντησης.
API	Application Programming Interface	Διεπαφή μέσω της οποίας διαφορετικά συστήματα ή υπηρεσίες ανταλλάσσουν δεδομένα και λειτουργίες.
CRUD	Create, Read, Update, Delete	Βασικό σύνολο λειτουργιών δημιουργίας, ανάγνωσης, ενημέρωσης και διαγραφής δεδομένων.
DB	Database	Βάση δεδομένων στην οποία αποθηκεύονται οι οντότητες της εφαρμογής.
DTO	Data Transfer Object	Αντικείμενο που μεταφέρει δεδομένα ανάμεσα σε επίπεδα ή υπηρεσίες της εφαρμογής.
ER	Entity Relationship	Μοντέλο ή διάγραμμα οντοτήτων και συσχετίσεων βάσης δεδομένων.
GDPR	General Data Protection Regulation	Γενικός Κανονισμός Προστασίας Δεδομένων της Ευρωπαϊκής Ένωσης.
HCI	Human-Computer Interaction	Αλληλεπίδραση ανθρώπου-υπολογιστή.
HTTP	HyperText Transfer Protocol	Πρωτόκολλο επικοινωνίας που χρησιμοποιείται στον Παγκόσμιο Ιστό.
JSON	JavaScript Object Notation	Μορφή ανταλλαγής δεδομένων που χρησιμοποιείται συχνά σε web APIs.
LLM	Large Language Model	Μεγάλο γλωσσικό μοντέλο που μπορεί να επεξεργάζεται και να παράγει φυσική γλώσσα.
POI	Point of Interest	Σημείο ενδιαφέροντος.
REST	Representational State Transfer	Αρχιτεκτονικό ύφος για web υπηρεσίες που οργανώνονται γύρω από πόρους και HTTP αιτήματα.
SQL	Structured Query Language	Γλώσσα διαχείρισης και αναζήτησης δεδομένων σε σχεσιακές βάσεις.
TOC	Table of Contents	Πίνακας περιεχομένων.
UML	Unified Modeling Language	Γλώσσα μοντελοποίησης για την οπτική περιγραφή συστημάτων λογισμικού.
UX	User Experience	Εμπειρία χρήστη.

Βιβλιογραφία

- [1] Weiser, M. (1991). The computer for the 21st century. *Scientific American*, 265(3), 94-104. Διαθέσιμο στο: <https://doi.org/10.1038/scientificamerican0991-94>
- [2] Schilit, B. N., Adams, N., & Want, R. (1994). Context-aware computing applications. 1994 First Workshop on Mobile Computing Systems and Applications, 85-90. Διαθέσιμο στο: <https://doi.org/10.1109/MCSA.1994.512740>
- [3] Dey, A. K. (2001). Understanding and using context. *Personal and Ubiquitous Computing*, 5, 4-7. Διαθέσιμο στο: <https://doi.org/10.1007/s007790170019>
- [4] Bass, L., Clements, P., & Kazman, R. (2021). *Software architecture in practice* (4th ed.). Addison-Wesley. Διαθέσιμο στο: <https://www.informit.com/store/software-architecture-in-practice-9780136886099>
- [5] ISO/IEC/IEEE. (2018). ISO/IEC/IEEE 29148:2018 Systems and software engineering - Life cycle processes - Requirements engineering. Διαθέσιμο στο: <https://www.iso.org/standard/72089.html>
- [6] Garrett, J. J. (2011). *The elements of user experience: User-centered design for the web and beyond* (2nd ed.). New Riders. Διαθέσιμο στο: <https://www.pearson.com/en-us/subject-catalog/p/elements-of-user-experience-the-user-centered-design-for-the-web-and-beyond/P200000009548>
- [7] Nielsen, J. (1993). *Usability engineering*. Morgan Kaufmann. Διαθέσιμο στο: <https://doi.org/10.1016/C2009-0-21512-1>
- [8] Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures* (Doctoral dissertation, University of California, Irvine). Διαθέσιμο στο: <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- [9] World Wide Web Consortium. (2023). *Web Content Accessibility Guidelines (WCAG) 2.2*. Διαθέσιμο στο: <https://www.w3.org/TR/WCAG22/>
- [10] Charland, A., & Leroux, B. (2011). Mobile application development: Web vs. native. *Communications of the ACM*, 54(5), 49-53. Διαθέσιμο στο: <https://doi.org/10.1145/1941487.1941504>
- [11] Wasserman, A. I. (2010). Software engineering issues for mobile application development. *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*, 397-400. Διαθέσιμο στο: <https://doi.org/10.1145/1882362.1882443>
- [12] Harrison, R., Flood, D., & Duce, D. (2013). Usability of mobile applications: Literature review and rationale for a new usability model. *Journal of Interaction Science*, 1, Article 1. Διαθέσιμο στο: <https://doi.org/10.1186/2194-0827-1-1>
- [13] Gavalas, D., Konstantopoulos, C., Mastakas, K., & Pantziou, G. (2014). Mobile recommender systems in tourism. *Journal of Network and Computer Applications*, 39, 319-333. Διαθέσιμο στο: <https://doi.org/10.1016/j.jnca.2013.04.006>
- [14] Poslad, S. (2009). *Ubiquitous computing: Smart devices, environments and interactions*. Wiley. Διαθέσιμο στο: <https://doi.org/10.1002/9780470779446>
- [15] Cook, D. J., & Das, S. K. (2007). How smart are our environments? An updated look at the state of the art. *Pervasive and Mobile Computing*, 3(2), 53-73. Διαθέσιμο στο: <https://doi.org/10.1016/j.pmcj.2006.12.001>
- [16] Want, R., Hopper, A., Falcao, V., & Gibbons, J. (1992). The active badge location system. *ACM Transactions on Information Systems*, 10(1), 91-102. Διαθέσιμο στο: <https://doi.org/10.1145/128756.128759>
- [17] Kontogianni, A., Chrysafiadi, K., Virvou, M., & Alepis, E. (2026). From Wikidata to smart tourism: A reproducible pipeline based on AI and fuzzy logic for interpretable multi-category classification of points of interest. *Mathematics*, 14(12), 2227.

- [18] Chrysafiadi, K., Kontogianni, A., Virvou, M., & Alepis, E. (2025). Enhancing user experience in smart tourism via fuzzy logic-based personalization. *Mathematics*, 13(5), 846.
- [19] Russell, S., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th ed.). Pearson. Διαθέσιμο στο: <https://aima.cs.berkeley.edu/>
- [20] Poole, D. L., & Mackworth, A. K. (2017). *Artificial intelligence: Foundations of computational agents* (2nd ed.). Cambridge University Press. Διαθέσιμο στο: <https://artint.info/>
- [21] Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2), 115-152. Διαθέσιμο στο: <https://doi.org/10.1017/S0269888900008122>
- [22] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. Διαθέσιμο στο: <https://www.deeplearningbook.org/>
- [23] Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255-260. Διαθέσιμο στο: <https://doi.org/10.1126/science.aaa8415>
- [24] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30. Διαθέσιμο στο: <https://arxiv.org/abs/1706.03762>
- [25] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877-1901. Διαθέσιμο στο: <https://arxiv.org/abs/2005.14165>
- [26] Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., ... Liang, P. (2021). On the opportunities and risks of foundation models. *arXiv*. Διαθέσιμο στο: <https://arxiv.org/abs/2108.07258>
- [27] Tsihrintzis, G. A., Virvou, M., Doukas, H., & Jain, L. C. (2024). *Advances in artificial intelligence-empowered decision support systems*. Springer.
- [28] Virvou, M., Tsihrintzis, G. A., Chrysafiadi, K., Sakkopoulos, E., & Tsihrintzi, E. A. (2025, July). Bi-directional 21st century skills for generative AI in educational software: A requirements engineering perspective. In *2025 16th International Conference on Information, Intelligence, Systems & Applications (IISA)* (pp. 1-10). IEEE.
- [29] Hassenzahl, M., & Tractinsky, N. (2006). User experience - A research agenda. *Behaviour & Information Technology*, 25(2), 91-97. Διαθέσιμο στο: <https://doi.org/10.1080/01449290500330331>
- [30] Norman, D. A. (2013). *The design of everyday things: Revised and expanded edition*. Basic Books. Διαθέσιμο στο: <https://www.basicbooks.com/titles/don-norman/the-design-of-everyday-things/9780465050659/>
- [31] ISO. (2018). ISO 9241-11:2018 Ergonomics of human-system interaction - Part 11: Usability: Definitions and concepts. Διαθέσιμο στο: <https://www.iso.org/standard/63500.html>
- [32] ISO. (2019). ISO 9241-210:2019 Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems. Διαθέσιμο στο: <https://www.iso.org/standard/77520.html>
- [33] Virvou, M. (2023). Artificial intelligence and user experience in reciprocity: Contributions and state of the art. *Intelligent Decision Technologies*, 17(1), 73-125.
- [34] Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6, 87-129. Διαθέσιμο στο: <https://doi.org/10.1007/BF00143964>
- [35] Brusilovsky, P. (2001). Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11, 87-110. Διαθέσιμο στο: <https://doi.org/10.1023/A:1011143116306>
- [36] Kobsa, A. (2001). Generic user modeling systems. *User Modeling and User-Adapted Interaction*, 11, 49-63. Διαθέσιμο στο: <https://doi.org/10.1023/A:1011187500863>

- [37] Jameson, A. (2008). Adaptive interfaces and agents. In J. A. Jacko & A. Sears (Eds.), *The human-computer interaction handbook* (2nd ed., pp. 433-458). CRC Press. Διαθέσιμο στο: <https://doi.org/10.1201/9781410615862>
- [38] Chrysafiadi, K., Papadimitriou, S., & Virvou, M. (2022). Cognitive-based adaptive scenarios in educational games using fuzzy reasoning. *Knowledge-Based Systems*, 250, 109111.
- [39] Chrysafiadi, K., Papadimitriou, S., Kamitsios, M., & Virvou, M. (2024, August). User experience in serious games: Adaptivity in the plot and environment of the game. In *International Conference on Artificial Intelligence-Empowered Software Engineering* (pp. 215-230). Cham: Springer Nature Switzerland.
- [40] Chrysafiadi, K., Papadimitriou, S., & Virvou, M. (2025, November). A fuzzy-logic based cognitive walkthrough to assess the degrees of soft skills targeted in an intelligent educational adventure game. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence* (pp. 347-361). Cham: Springer Nature Switzerland.
- [41] Dix, A., Finlay, J., Abowd, G. D., & Beale, R. (2004). *Human-computer interaction* (3rd ed.). Pearson. Διαθέσιμο στο: <https://www.pearson.com/en-gb/subject-catalog/p/human-computer-interaction/P200000003157>
- [42] Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). *Designing the user interface: Strategies for effective human-computer interaction* (6th ed.). Pearson. Διαθέσιμο στο: <https://www.pearson.com/en-us/subject-catalog/p/designing-the-user-interface-strategies-for-effective-human-computer-interaction/P200000003334>
- [43] Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 249-256. Διαθέσιμο στο: <https://doi.org/10.1145/97243.97281>
- [44] Preece, J., Rogers, Y., & Sharp, H. (2015). *Interaction design: Beyond human-computer interaction* (4th ed.). Wiley. Διαθέσιμο στο: <https://www.wiley.com/en-us/Interaction+Design%3A+Beyond+Human+Computer+Interaction%2C+4th+Edition-p-9781119020752>
- [45] Tidwell, J. (2010). *Designing interfaces* (2nd ed.). O'Reilly Media. Διαθέσιμο στο: <https://www.oreilly.com/library/view/designing-interfaces-2nd/9781449379711/>
- [46] Myers, B. A. (1998). A brief history of human-computer interaction technology. *Interactions*, 5(2), 44-54. Διαθέσιμο στο: <https://doi.org/10.1145/274430.274436>
- [47] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269-271. Διαθέσιμο στο: <https://doi.org/10.1007/BF01386390>
- [48] Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 100-107. Διαθέσιμο στο: <https://doi.org/10.1109/TSSC.1968.300136>
- [49] Bast, H., Delling, D., Goldberg, A., Muller-Hannemann, M., Pajor, T., Sanders, P., Wagner, D., & Werneck, R. F. (2016). Route planning in transportation networks. In L. Kliemann & P. Sanders (Eds.), *Algorithm engineering* (pp. 19-80). Springer. Διαθέσιμο στο: https://doi.org/10.1007/978-3-319-49487-6_2
- [50] Delling, D., Sanders, P., Schultes, D., & Wagner, D. (2009). Engineering route planning algorithms. In J. Lerner, D. Wagner, & K. A. Zweig (Eds.), *Algorithmics of large and complex networks* (pp. 117-139). Springer. Διαθέσιμο στο: https://doi.org/10.1007/978-3-642-02094-0_7
- [51] Resnick, P., & Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, 40(3), 56-58. Διαθέσιμο στο: <https://doi.org/10.1145/245108.245121>
- [52] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734-749. Διαθέσιμο στο: <https://doi.org/10.1109/TKDE.2005.99>
- [53] Ricci, F., Rokach, L., & Shapira, B. (Eds.). (2015). *Recommender systems handbook* (2nd ed.). Springer. Διαθέσιμο στο: <https://doi.org/10.1007/978-1-4899-7637-6>

- [54] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12, 331-370. Διαθέσιμο στο: <https://doi.org/10.1023/A:1021240730564>
- [55] Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1), 5-53. Διαθέσιμο στο: <https://doi.org/10.1145/963770.963772>
- [56] Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3), 338-353. Διαθέσιμο στο: [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)
- [57] Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning-I. *Information Sciences*, 8(3), 199-249. Διαθέσιμο στο: [https://doi.org/10.1016/0020-0255\(75\)90036-5](https://doi.org/10.1016/0020-0255(75)90036-5)
- [58] Bellman, R. E., & Zadeh, L. A. (1970). Decision-making in a fuzzy environment. *Management Science*, 17(4), B141-B164. Διαθέσιμο στο: <https://doi.org/10.1287/mnsc.17.4.B141>
- [59] Chrysafiadi, K. (2024, August). Incorporating fuzzy logic in intelligent software systems. In *International Conference on Artificial Intelligence-Empowered Software Engineering* (pp. 470-475). Cham: Springer Nature Switzerland.
- [60] Chrysafiadi, K. (2023). The role of fuzzy logic in artificial intelligence and smart applications. In *Fuzzy logic-based software systems* (pp. 25-29). Cham: Springer International Publishing.
- [61] Chrysafiadi, K., & Tsichrintzi, E. A. (2021). An intelligent fuzzy-based emergency alert generation for persons with episodic memory decline problems. *Intelligent Decision Technologies*, 15(4), 681-690.
- [62] Chrysafiadi, K., Tsichrintzi, E. A., & Sakkopoulos, E. (2024). A smart home fuzzy logic-based cooking fire prevention system addressing human errors due to distraction or minor medical memory lapses. *Procedia Computer Science*, 246, 3390-3399.
- [63] Chrysafiadi, K., & Virvou, M. (2024). PerFuSIT: Personalized fuzzy logic strategies for intelligent tutoring of programming. *Electronics*, 13(23), 4827.
- [64] Google. (n.d.). Google Maps Platform documentation. Retrieved June 15, 2026, from Διαθέσιμο στο: <https://developers.google.com/maps/documentation>
- [65] Google. (n.d.). Routes API documentation. Retrieved June 15, 2026, from Διαθέσιμο στο: <https://developers.google.com/maps/documentation/routes>
- [66] Google. (n.d.). Gemini API documentation. Retrieved June 15, 2026, from Διαθέσιμο στο: <https://ai.google.dev/gemini-api/docs>
- [67] Panagoulas, D. P., Sotiropoulos, D., Chrysafiadi, K., Tsichrintzi, E. A., Sakkopoulos, E., Tsichrintzis, G. A., & Virvou, M. (2024, July). LYRICEL: A rule-augmented artificial intelligence-empowered cultural e-learning with GPT and machine learning for song lyrics. In *2024 15th International Conference on Information, Intelligence, Systems & Applications (IISA)* (pp. 1-5). IEEE.
- [68] Panagoulas, D. P., Tsichrintzi, E. A., Sotiropoulos, D., Chrysafiadi, K., Sakkopoulos, E., Tsichrintzis, G. A., & Virvou, M. (2025). LYRICEL: Knowledge graphs combined with large language models and machine learning for cross-cultural analysis of lyrics: The case of Greek songs. *IEEE Access*.
- [69] Mougiakou, E., Papadimitriou, S., Chrysafiadi, K., & Virvou, M. (2025). Artificial intelligence in educational games and consent under general data protection regulation. *Intelligent Decision Technologies*, 19(2), 670-686.
- [70] Google. (n.d.). Google Maps: Your guide to anywhere. Retrieved June 15, 2026, from Διαθέσιμο στο: <https://www.google.com/maps/about/>
- [71] Waze. (n.d.). About Waze. Retrieved June 15, 2026, from Διαθέσιμο στο: <https://www.waze.com/about>
- [72] Apple. (n.d.). Maps. Retrieved June 15, 2026, from Διαθέσιμο στο: <https://www.apple.com/maps/>
- [73] Google Help. (n.d.). About Waze. Retrieved June 15, 2026, from Διαθέσιμο στο: <https://support.google.com/waze/answer/6071177>

- [74] Foursquare. (n.d.). Place Search. Retrieved June 15, 2026, from Διαθέσιμο στο: <https://docs.foursquare.com/developer/reference/place-search>
- [75] European Parliament and Council of the European Union. (2016). Regulation (EU) 2016/679, General Data Protection Regulation. Διαθέσιμο στο: <https://eur-lex.europa.eu/eli/reg/2016/679/oj>
- [76] OWASP Foundation. (2021). OWASP Top 10:2021. Διαθέσιμο στο: <https://owasp.org/Top10/>
- [77] Object Management Group. (2017). Unified Modeling Language, version 2.5.1. Διαθέσιμο στο: <https://www.omg.org/spec/UML/>
- [78] Angular. (2026). What is Angular? Retrieved June 17, 2026, from Διαθέσιμο στο: <https://angular.dev/overview>
- [79] Microsoft. (2025). Overview of ASP.NET Core. Retrieved June 17, 2026, from Διαθέσιμο στο: <https://learn.microsoft.com/en-us/aspnet/core/overview>
- [80] Microsoft. (2024). Entity Framework Core. Retrieved June 17, 2026, from Διαθέσιμο στο: <https://learn.microsoft.com/en-us/ef/core/>
- [81] Microsoft. (2026). SQL Server technical documentation. Retrieved June 17, 2026, from Διαθέσιμο στο: <https://learn.microsoft.com/en-us/sql/sql-server/>
- [82] Codd, E. F. (1970). A relational model of data for large shared data banks. Communications of the ACM, 13(6), 377-387. Διαθέσιμο στο: <https://doi.org/10.1145/362384.362685>
- [83] Chen, P. P. (1976). The entity-relationship model: Toward a unified view of data. ACM Transactions on Database Systems, 1(1), 9-36. Διαθέσιμο στο: <https://doi.org/10.1145/320434.320440>

Παράρτημα Α: API συμβόλαιο request/response

Το παράρτημα αυτό συνοψίζει ενδεικτικά API endpoints που υποστηρίζουν τις βασικές λειτουργίες της SmartNavix. Η καταγραφή δεν αντικαθιστά τον πηγαίο κώδικα, αλλά βοηθά τον αναγνώστη να συνδέσει τις λειτουργίες της εφαρμογής με το backend.

Τα παραδείγματα παρουσιάζονται σε επίπεδο λειτουργίας και δεν περιλαμβάνουν μυστικά, κωδικούς ή API keys.

ΠΙΝΑΚΑΣ Α.1: Ενδεικτικά API endpoints SmartNavix

Περιοχή	Ενδεικτικό endpoint / λειτουργία	Σκοπός
Χρήστες	POST api/User/CreateUser	Δημιουργία νέου λογαριασμού με βασικό έλεγχο στοιχείων, ρόλου, avatar και προτίμησης.
Χρήστες	POST api/User/Login	Σύνδεση χρήστη και ανάκτηση βασικών στοιχείων πρόσβασης.
Χρήστες	POST api/User/Update	Ενημέρωση στοιχείων προφίλ χρήστη.
Διαδρομές	POST api/Trip/Create	Αποθήκευση διαδρομής, απόστασης, προορισμού, βαθμολογίας και στάσεων.
Διαδρομές	POST api/Trip/GetUserTripDetails	Ανάκτηση πρόσφατων διαδρομών και συνοπτικών στατιστικών χρήστη.
Διαδρομές	POST api/Trip/Rate	Καταχώριση αξιολόγησης διαδρομής από τον χρήστη.
Προτιμήσεις	POST api/FilteredPreference/Create	Αποθήκευση εφαρμοσμένων φίλτρων και προτιμήσεων χρήστη.
Προτιμήσεις	POST api/FilteredPreference/GetLatestByUser	Ανάκτηση της καταλληλότερης πρόσφατης προτίμησης με χρήση resolver.
Presets	POST api/Preset/Create	Δημιουργία αποθηκευμένου preset προτιμήσεων.
Διαχείριση	POST api/Admin/...	Λειτουργίες admin panel για χρήστες, ρόλους, στατιστικά και ιστορικό διοικητικών ενεργειών.

Παράρτημα Β: SQL scripts και migration σημειώσεις

Το παράρτημα αυτό συγκεντρώνει συνοπτικές σημειώσεις για τη βάση δεδομένων της εφαρμογής. Η πλήρης τεχνική λεπτομέρεια βρίσκεται στον πηγαίο κώδικα και στα migration στοιχεία του έργου, ενώ εδώ παρουσιάζεται η λειτουργική εικόνα που χρειάζεται για την κατανόηση της εργασίας.

ΠΙΝΑΚΑΣ Β.1: Σύνοψη βασικών πινάκων και σημειώσεων βάσης

Περιοχή βάσης	Βασικές οντότητες	Σημείωση υλοποίησης
Χρήστες και ρόλοι	User, Role, Avatar, Preference	Κρατούν λογαριασμό, πρόσβαση, avatar και βασική προτίμηση χρήστη.
Διαδρομές	Trip, Station, Vehicle	Αποθηκεύουν προορισμό, αφετηρία, απόσταση, βαθμολογία, όχημα και ενδιάμεσες στάσεις.
Φίλτραρισμένες προτιμήσεις	FilteredPreference	Κρατά τις επιλογές που εφαρμόζει ο χρήστης σε συγκεκριμένο πλαίσιο αναζήτησης.
Presets	Preset, PresetIcon	Επιτρέπουν επαναχρησιμοποίηση αποθηκευμένων συνόλων ρυθμίσεων και εικονιδίων.
Ρυθμίσεις χρήστη	UserSettings	Κρατούν ρυθμίσεις διεπαφής και προτιμήσεις εμφάνισης.
Διοικητικό ιστορικό	AdminActionLog	Καταγράφει ενέργειες διαχείρισης για λόγους διαφάνειας και ελέγχου.
Ακεραιότητα	Foreign keys και κανόνες διαγραφής	Οι σχέσεις προστατεύουν τη συνέπεια των δεδομένων και μειώνουν σπασμένες αναφορές.
Απόδοση	Index σε UserID και AppliedAt	Υποστηρίζει γρηγορότερη ανάκτηση πρόσφατων προτιμήσεων ανά χρήστη.

Παράρτημα Γ: Ρυθμίσεις εγκατάστασης και εκτέλεσης

Το παράρτημα αυτό περιγράφει συνοπτικά τα βήματα που απαιτούνται για τοπική εκτέλεση της εφαρμογής. Η περιγραφή είναι σκόπιμα γενική, επειδή οι πραγματικές τιμές ρυθμίσεων εξαρτώνται από το περιβάλλον του χρήστη και δεν πρέπει να δημοσιεύονται μαζί με την εργασία.

ΠΙΝΑΚΑΣ Γ.1: Βασικά βήματα εγκατάστασης και εκτέλεσης

Βήμα	Περιγραφή
1. Προαπαιτούμενα	Εγκατάσταση .NET SDK, Node.js, Angular tooling και διαθέσιμου SQL Server ή συμβατού περιβάλλοντος βάσης.
2. Ρυθμίσεις backend	Ορισμός connection string και απαραίτητων ρυθμίσεων σε ασφαλές τοπικό αρχείο ή περιβάλλον εκτέλεσης.
3. Εκκίνηση backend	Εκτέλεση του .NET project ώστε να είναι διαθέσιμα τα REST endpoints στο localhost.
4. Ρυθμίσεις frontend	Έλεγχος environment αρχείου της Angular εφαρμογής ώστε να δείχνει στη σωστή backend διεύθυνση.
5. Εκκίνηση frontend	Εκτέλεση της Angular εφαρμογής και πρόσβαση από browser στο αντίστοιχο localhost port.
6. Έλεγχος ροής	Σύνδεση με δοκιμαστικό χρήστη, αναζήτηση διαδρομής, εφαρμογή φίλτρων και επιβεβαίωση ότι επιστρέφεται αποτέλεσμα.
7. Ασφάλεια ρυθμίσεων	Τα API keys και οι κωδικοί δεν πρέπει να ενσωματώνονται στο κείμενο της εργασίας ή σε δημόσιο αποθετήριο.

Παράρτημα Δ: Πρόσθετα UML / ER / sequence diagrams

Στο κυρίως σώμα της εργασίας έχουν ενσωματωθεί τα βασικά διαγράμματα που χρειάζονται για την κατανόηση της SmartNavix. Το παρόν παράρτημα λειτουργεί ως συγκεντρωτικός οδηγός των διαγραμμάτων και του ρόλου τους.

ΠΙΝΑΚΑΣ Δ.1: Σύνοψη διαγραμμάτων της εργασίας

Διάγραμμα	Θέση στο κείμενο	Ρόλος
UML use case	Κεφάλαιο 4	Περιγράφει τους ρόλους και τις βασικές περιπτώσεις χρήσης της SmartNavix.
Αρχιτεκτονική συστήματος	Κεφάλαιο 5	Δείχνει τη σύνδεση frontend, backend, βάσης δεδομένων και εξωτερικών APIs.
Ροή αναζήτησης διαδρομής	Κεφάλαιο 5	Αποτυπώνει τα βασικά βήματα από την είσοδο χρήστη μέχρι την απόκριση του συστήματος.
ER διάγραμμα	Κεφάλαιο 6	Παρουσιάζει τις βασικές οντότητες και σχέσεις της βάσης δεδομένων.
Ροή υλοποίησης λειτουργιών	Κεφάλαιο 7	Συνδέει τις βασικές λειτουργίες με frontend, backend και αποθήκευση.
Ροή fuzzy resolver	Κεφάλαιο 8	Δείχνει πώς επιλέγεται η best preference μέσα από ιστορικό και κανόνες βαθμολόγησης.

Παράρτημα Ε: Επιπλέον screenshots εφαρμογής

Τα βασικά screenshots παρουσιάζονται στο Κεφάλαιο 9, κοντά στις αντίστοιχες ροές χρήσης. Το παρόν παράρτημα λειτουργεί ως ευρετήριο των οθονών, ώστε ο αναγνώστης να μπορεί να εντοπίσει γρήγορα ποια λειτουργία τεκμηριώνει κάθε εικόνα.

ΠΙΝΑΚΑΣ Ε.1: Ευρετήριο screenshots εφαρμογής

Screenshot	Περιγραφή	Σχετική ενότητα
EIKONA 9.1	Οθόνη εισόδου SmartNavix	Σύνδεση χρήστη
EIKONA 9.2	Αρχική οθόνη πλοήγησης	Βασική διεπαφή
EIKONA 9.3	Εισαγωγή προορισμού από τον χρήστη	Αναζήτηση διαδρομής
EIKONA 9.4	Φίλτρα διαδρομής και αποθηκευμένες επιλογές	Προτιμήσεις χρήστη
EIKONA 9.5	Απόκριση συστήματος με προτεινόμενη διαδρομή	Αποτέλεσμα πλοήγησης
EIKONA 9.6	Αξιολόγηση διαδρομής μετά την έναρξη πλοήγησης	Feedback loop
EIKONA 9.7	Πίνακας διαχείρισης SmartNavix	Admin panel