



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
«Ανάπτυξη Εφαρμογής Πλοήγησης
Πανεπιστημιούπολης με Διαδραστικό Χάρτη και
Υποστήριξη Προσβασιμότητας»



Του φοιτητή
Ράντζου Βασιλείου
Αρ. Μητρώου: 2020248

Επιβλέπων
Ονοματεπώνυμο Ελβίρα-Μαρία
Αρβανίτου
Βαθμίδα Επίκουρη Καθηγήτρια

Ημερομηνία 10/04/2026

Τίτλος Δ.Ε. **Ανάπτυξη Εφαρμογής Πλοήγησης Πανεπιστημιούπολης με Διαδραστικό Χάρτη και Υποστήριξη Προσβασιμότητας**

Κωδικός Δ.Ε. 25366

Όνοματεπώνυμο φοιτητή **Ράντζος Βασίλειος**

Όνοματεπώνυμο εισηγητή **Ελβίρα-Μαρία Αρβανίτου**

Ημερομηνία ανάληψης Δ.Ε. **13/11/2025**

Ημερομηνία περάτωσης Δ.Ε. **31/05/2026**

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Ράντζου Βασιλείου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιοδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Αφιέρωση»

Σε όλους τους νέους που θέλουν να πραγματοποιήσουν το όνειρό τους.

Πρόλογος

Η επιλογή της παρούσας διπλωματικής εργασίας προέκυψε από το ενδιαφέρον για την ανάπτυξη εφαρμογών που καλύπτουν πραγματικές ανάγκες σε επίπεδο καθημερινής λειτουργίας και εξυπηρέτησης χρηστών. Η πλοήγηση σε μια πανεπιστημιούπολη αποτελεί μια πρακτική ανάγκη, καθώς σχετίζεται με την εξερεύνηση και την εύρεση τμημάτων, κτιρίων και σημείων ενδιαφέροντος για φοιτητές, προσωπικό και επισκέπτες. Έχοντας έρθει σε επαφή σε περιστατικά όπου άτομα άτομα με αναπηρία (ΑμεΑ) αντιμετώπιζαν δυσκολίες μετακίνησης σε πεζοδρόμια και δρόμους, η εφαρμογή UniMap υιοθετεί την αρχή ενός «χάρτη για όλους» με στόχο την ισότιμη πρόσβαση από άκρη σε άκρη εντός της πανεπιστημιούπολης. Προσωπικός μου στόχος, με την ολοκλήρωση της διπλωματικής μου εργασίας είναι να έχω εξελιχθεί στους τομείς που πραγματεύεται, αλλά και να συμβάλλω στη βελτίωση της εμπειρίας χρήστη σε έναν πρωτοετή φοιτητή, σε έναν γονέα ή σε έναν επισκέπτη ώστε να προσανατολιστεί εντός του πανεπιστημιακού χώρου. Η ενασχόλησή μου με τον προγραμματισμό και η επιθυμία να προσφέρω μια χρήσιμη και συμπεριληπτική λύση αποτελούν τους βασικούς λόγους που με οδήγησαν στην επιλογή του συγκεκριμένου θέματος.

Περίληψη

Η παρούσα διπλωματική εργασία αφορά τον σχεδιασμό και της ανάπτυξης της UniMap, μιας εφαρμογής πλοήγησης για κινητές συσκευές Android με στόχο τη διευκόλυνση μετακίνησης εντός πανεπιστημιούπολης. Η εφαρμογή παρέχει διαδραστικό χάρτη, εντοπισμό θέσης χρήστη σε πραγματικό χρόνο και καθοδήγηση προς τμήματα, κτίρια και σημεία ενδιαφέροντος. Κεντρικός άξονας του έργου αποτελεί η προσβασιμότητα, με έμφαση στην υποστήριξη ατόμων με αναπηρία (ΑμεΑ), ώστε η πλοήγηση να είναι ουσιαστικά αξιοποιήσιμη από όλους. Για την υλοποίηση, αξιοποιούνται τεχνολογίες ανάπτυξης mobile εφαρμογών και χαρτογραφικής απεικόνισης, ενώ η δρομολόγηση βασίζεται σε μοντελοποίηση του χώρου ως γράφου διαδρομών και σε αλγοριθμική εύρεση βέλτιστης πορείας μεταξύ σημείων. Η εφαρμογή υποστηρίζει αναζήτηση προορισμών, επιλογή σημείων στον χάρτη και προβολή της προτεινόμενης διαδρομής, επιτρέποντας σαφή προσανατολισμό ακόμη και σε χρήστες που δεν γνωρίζουν τον χώρο. Ως αποτέλεσμα, προέκυψε ένα λειτουργικό πρωτότυπο που μπορεί να χρησιμοποιηθεί για καθημερινή πλοήγηση στην πανεπιστημιούπολη, βελτιώνοντας την εμπειρία μετακίνησης και ενισχύοντας την αυτονομία των χρηστών. Η προσέγγιση «χάρτης για όλους» αναδεικνύει τη σημασία της συμπερίληψης και της προσβασιμότητας στον σχεδιασμό ψηφιακών υπηρεσιών, ενώ παράλληλα το έργο αποτελεί πρακτική εφαρμογή γνώσεων σε mobile ανάπτυξη, χαρτογραφικά δεδομένα και αλγορίθμους δρομολόγησης.

«Development of a Campus Navigation Application with an Interactive Map and Accessibility Support»

«Rantzos Vasileios»

Abstract

The present thesis concerns the design and the development of UniMap, a navigation application for Android mobile devices, aiming to facilitate movement within a university campus. The application provides an interactive map, real-time user location tracking, and guidance to departments, buildings, and points of interest. A central axis of the project is accessibility, with an emphasis on supporting people with disabilities (PwD), so that navigation is truly usable by everyone. For the implementation, technologies for mobile application development and cartographic visualization are utilized, while routing is based on modeling the area as a path graph and on algorithmic computation of the optimal route between points. The application supports destination search, point selection, point selection on the map, and display of the suggested route, enabling clear orientation even for users who are unfamiliar with the area. As a result, a functional prototype was produced that can be used for daily navigation on campus, improving the mobility experience and enhancing users' autonomy. The “map for all” approach highlights the importance of inclusion and accessibility in the design of digital services, while at the same time the project constitutes a practical application of knowledge in mobile development, cartographic data, and routing algorithms.

Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες προς την επιβλέπουσα της διπλωματικής μου εργασίας για την καθοδήγηση, την επιστημονική υποστήριξη και τις χρήσιμες παρατηρήσεις καθ' όλη τη διάρκεια της εκπόνησής της. Η συμβολή της υπήρξε καθοριστική τόσο στην οργάνωση της εργασίας όσο και στην ολοκλήρωση του έργου.

Επιπλέον, ευχαριστώ το Διεθνές Πανεπιστήμιο Ελλάδας και το Τμήμα μου για τις γνώσεις και τα εφόδια που μου προσέφεραν κατά τη διάρκεια των σπουδών μου, τα οποία συνέβαλαν ουσιαστικά στην υλοποίηση της παρούσας εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου και τους ανθρώπους του στενού μου περιβάλλοντος για τη στήριξη, την κατανόηση και την ενθάρρυνση που μου παρείχαν, ιδιαίτερα σε περιόδους αυξημένων απαιτήσεων και δύσκολων συγκυριών. Παρότι δεν τους κατονομάζω, η συμβολή τους υπήρξε σημαντική και καθοριστική για την ολοκλήρωση της προσπάθειας αυτής.

Περιεχόμενα

Πρόλογος.....	iv
Περίληψη.....	v
Abstract	vi
Ευχαριστίες	vii
Περιεχόμενα	viii
Κατάλογος Σχημάτων	x
Συντομογραφίες.....	xiii
Κεφάλαιο 1ο: Εισαγωγή.....	1
1.1 Γενικό Πλαίσιο και Αντικείμενο	1
1.2 Προσδιορισμός του Προβλήματος	2
1.3 Κίνητρο Διπλωματικής Εργασίας	2
1.4 Δομή Διπλωματικής Εργασίας	3
Κεφάλαιο 2ο: Παρόμοιες Εφαρμογές / Ανάλυση Ανταγωνισμού.....	5
2.1 Ανάλυση Υφιστάμενων Εφαρμογών.....	5
2.1.1 Google Maps	5
2.1.2 Apple Maps	5
2.1.3 MazeMap.....	6
2.1.4 WheelMap	6
2.2 Δυνατά και Αδύναμα Σημεία Ανταγωνιστικών Λύσεων	7
2.3 Κενά της Αγοράς και Καινοτομία της Προτεινόμενης Εφαρμογής.....	8
2.3.1 Κενά της Αγοράς.....	8
2.3.2 Καινοτομία της Προτεινόμενης Εφαρμογής (UniMap)	8
Κεφάλαιο 3ο: Μεθοδολογία και Σχεδιαστικές Αποφάσεις	9
3.1 Επιλογή Γλωσσών Προγραμματισμού	9
3.1.1 Γλώσσα Προγραμματισμού TypeScript.....	9
3.2 Επιλογή Εργαλείων και Τεχνολογιών	11
3.2.1 Τεχνολογίες Back-End	11
3.2.2 Τεχνολογίες Front-End.....	11
Κεφάλαιο 4ο: Ανάπτυξη Εφαρμογής και Περιγραφή Λειτουργίας	21
4.1 Ανάπτυξη Back-End.....	21

4.1.1 Οργάνωση Δεδομένων στη Μνήμη	21
4.1.2 Προσδιορισμός Οντοτήτων	25
4.1.3 Αλγόριθμος A* και Δομή Γράφου	30
4.1.4 Σύστημα Προσβασιμότητας Αμαξιδίου	31
4.1.5 API Endpoints και Λογική Δρομολόγησης	32
4.2 Ανάπτυξη Front-End	32
4.2.1 Αρχιτεκτονική – Οντότητες Front-End	32
4.2.2 GPS, Dead-Reckoning και Navigation Camera	44
4.2.3 Οθόνες Διεπαφής Χρηστών – Περιπτώσεις Χρήσης.....	45
Κεφάλαιο 5ο: Συμπεράσματα και Μελλοντική Εργασία	67
5.1 Συμπεράσματα.....	67
5.2 Μελλοντική Εργασία και Βελτιστοποίηση	67
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	69

Κατάλογος Σχημάτων

Σχήμα 2.1: Απεικόνιση Google Maps	6
Σχήμα 2.2: Απεικόνιση Apple Maps	7
Σχήμα 2.3: Απεικόνιση MazeMap	8
Σχήμα 2.4: Απεικόνιση Wheelmap	9
Σχήμα 3.1: Ορισμός των βασικών τύπων δεδομένων και δομών της εφαρμογής.....	13
Σχήμα 3.2: Ορισμός του union type EdgeTag για τον περιορισμό των επιτρεπτών τιμών στις ακμές του γράφου.....	13
Σχήμα 3.3: Λογότυπο Node.js.....	14
Σχήμα 3.4: Λογότυπο framework ExpressJS.	14
Σχήμα 3.5: Αρχικοποίηση του διακομιστή Express και ρύθμιση των βασικών παραμέτρων του back-end	15
Σχήμα 3.6: Υπολογισμός των σημείων της διαδρομής και του συνολικού της μήκους σε μέτρα.	16
Σχήμα 3.7: Λογότυπο πλατφόρμας Render.....	17
Σχήμα 3.8 Ορισμός των βασικών διαδρομών (routes) περιήγησης για τη διεπαφή του χρήστη.....	17
Σχήμα 3.9: Λογότυπο framework Ionic	18
Σχήμα 3.10: Υλοποίηση του HTTP αιτήματος για τη λήψη των δεδομένων δρομολόγησης από τον διακομιστή.....	18
Σχήμα 3.11: Η βιβλιοθήκη Leaflet.js για την ανάπτυξη διαδραστικών χαρτών	19
Σχήμα 3.12: Git με χρήση Command Line	20
Σχήμα 3.13.1: Περιβάλλον GitHub.....	21
Σχήμα 3.13.2: Εκκίνηση της εφαρμογής σε τοπικό περιβάλλον.....	21
Σχήμα 3.14: Δημιουργία build της εφαρμογής και συγχρονισμός με το Android Project	22
Σχήμα 4.1: Συναρτήσεις για τον υπολογισμό του γεωγραφικού πλαισίου οριοθέτησης (Bounding Box) και τον έλεγχο ένταξης ενός σημείου σε αυτό.	24
Σχήμα 4.2: Συνάρτηση για την προσθήκη νέας, μη κατευθυνόμενης ακμής στον γράφο.	24
Σχήμα 4.3: Διαδικασία υποδιαίρεσης μιας μεγάλης ακμής του γράφου μέσω ενδιάμεσων κόμβων (waypoints).....	25
Σχήμα 4.4: Αυτόματη σύνδεση (επούλωση) κόμβων που βρίσκονται σε πολύ κοντινή απόσταση μεταξύ τους.....	25
Σχήμα 4.5: Αλγόριθμος για την εύρεση του μεγαλύτερου συνεχόμενου (συνδεδεμένου) τμήματος του γράφου.....	26
Σχήμα 4.6: Αρχικοποίηση και αποθήκευση του ενοποιημένου γράφου δρομολόγησης κατά την εκκίνηση του διακομιστή.	26
Σχήμα 4.7: Υπολογισμός της πραγματικής απόστασης (σε μέτρα) μεταξύ δύο γεωγραφικών σημείων, λαμβάνοντας υπόψη την καμπυλότητα της Γης.	27
Σχήμα 4.8: Διάγραμμα Οντοτήτων back-end.....	27
Σχήμα 4.9: Ορισμός της βασικής δομής για τις γεωγραφικές συντεταγμένες.	28
Σχήμα 4.10: Ορισμός της μορφής του δικτύου διαδρομών (γράφου).	28
Σχήμα 4.11: Χαρακτηρισμός συγκεκριμένων διαδρομών (π.χ. σκάλες) για τον έλεγχο προσβασιμότητας	28

Σχήμα 4.12: Δημιουργία ενός προσαρμοσμένου χάρτη δρομολόγησης ειδικά για χρήστες αναπηρικών αμαξιδίων.	29
Σχήμα 4.13: Δομή των δεδομένων που επιστρέφονται μετά τον υπολογισμό της διαδρομής.	29
Σχήμα 4.14: Η δομή των δεδομένων που απαιτούνται για την εκκίνηση ενός αιτήματος δρομολόγησης από την εφαρμογή	30
Σχήμα 4.15: Διαδικασία εισαγωγής της θέσης του χρήστη στον χάρτη και εκτέλεση του αλγορίθμου εύρεσης της συντομότερης διαδρομής.	30
Σχήμα 4.16: Διαδικασία εύρεσης του κοντινότερου σημείου αφετηρίας στον χάρτη, βάσει της τρέχουσας τοποθεσίας του χρήστη.....	31
Σχήμα 4.17: Βοηθητική συνάρτηση που προετοιμάζει και καθαρίζει το κείμενο για τη σωστή λειτουργία της αναζήτησης	31
Σχήμα 4.18: Δομή των δεδομένων για τα κτίρια και τους προορισμούς της εφαρμογής.....	32
Σχήμα 4.19: Η υπηρεσία επικοινωνίας (API Service) που στέλνει το αίτημα δρομολόγησης στον server.....	35
Σχήμα 4.20: Οι δομές δεδομένων για την αποστολή των παραμέτρων και τη λήψη του αποτελέσματος δρομολόγησης.	36
Σχήμα 4.21: Διαχείριση και τοπική αποθήκευση των προτιμήσεων του χρήστη.....	37
Σχήμα 4.22: Η υπηρεσία ελέγχου του GPS που ειδοποιεί την εφαρμογή για νέες τοποθεσίες ή σφάλματα σήματος.	37
Σχήμα 4.23: Αλγόριθμος που ελέγχει και απορρίπτει λανθασμένα στίγματα GPS που θα μπερδεύαν την πλοήγηση.....	38
Σχήμα 4.24: Μαθηματική εξομάλυνση της κίνησης του χρήστη στον χάρτη, ώστε ο δείκτης (marker) να μην "τρέμει".	39
Σχήμα 4.25: Υπολογισμός και εξομάλυνση της κατεύθυνσης του χρήστη (πυξίδα) από τους αισθητήρες του κινητού.....	40
Σχήμα 4.26: Δήλωση των γραμμών για την οπτική αναπαράσταση της διαδρομής πάνω στον χάρτη.	40
Σχήμα 4.27: Διαδικασία υπολογισμού της απόστασης του χρήστη από την επιλεγμένη διαδρομή (snapping) πάνω στον χάρτη.	41
Σχήμα 4.28: Ομαλή προσαρμογή (blending) της θέσης του χρήστη πάνω στη γραμμή της διαδρομής, ώστε η κίνηση στον χάρτη να φαίνεται φυσική.	41
Σχήμα 4.29: Βασικές ρυθμίσεις και όρια απόστασης/χρόνου για την ενεργοποίηση του αυτόματου επαναυπολογισμού της διαδρομής (rerouting).	42
Σχήμα 4.30: Λογική ελέγχου άφιξης, η οποία επιβεβαιώνει ότι ο χρήστης έχει προσεγγίσει σταθερά και με ακρίβεια τον τελικό του προορισμό.	42
Σχήμα 4.31: Ορισμός των γραφικών στοιχείων (εικονίδιο κατεύθυνσης και τελεία) για την αναπαράσταση της θέσης του χρήστη στον χάρτη.....	43
Σχήμα 4.32: Μηχανισμός ομαλής κίνησης (animation) του εικονιδίου του χρήστη πάνω στον χάρτη μεταξύ δύο διαδοχικών στιγμάτων GPS.	44
Σχήμα 4.33: Λογική ελέγχου για την αυτόματη περιστροφή του χάρτη σύμφωνα με την κατεύθυνση της πυξίδας της συσκευής.....	44
Σχήμα 4.34: Μαθηματικός υπολογισμός της γωνίας κατεύθυνσης (αζιμούθιο) μεταξύ δύο γεωγραφικών σημείων στον χάρτη.	45
Σχήμα 4.35: Βοηθητικές συναρτήσεις για την ομαλή περιστροφή της πυξίδας της εφαρμογής και τον υπολογισμό της συνολικής απόστασης μιας διαδρομής.....	47
Σχήμα 4.36: Οθόνη αδείας πρόσβασης GPS.....	48

Σχήμα 4.37: Splash Screen εφαρμογής UniMap	49
Σχήμα 4.38: Μηνύματα σφάλματος κατά τη φόρτωση της εφαρμογής	50
Σχήμα 4.39: Κύρια οθόνη χάρτη campus	51
Σχήμα 4.40: Ενεργοποίηση / Απενεργοποίηση Προσβασιμότητας.....	52
Σχήμα 4.41: Προειδοποιητικό μήνυμα επιλογής προορισμού εκτός πανεπιστημιούπολης	53
Σχήμα 4.42: Αναζήτηση προορισμού με drop list.....	54
Σχήμα 4.43: Ιστορικό πρόσφατων αναζητήσεων	55
Σχήμα 4.44: Popup πληροφοριών τμήματος	56
Σχήμα 4.45: Popup πληροφοριών τυχαίου σημείου εντός campus	57
Σχήμα 4.46: Προεπισκόπηση υπολογισμένης διαδρομής με εκτιμώμενη απόσταση και ώρα άφιξης..	58
Σχήμα 4.47: Προειδοποιητικό μήνυμα κατά την πλοήγηση εκτός campus	59
Σχήμα 4.48: Ενεργή πλοήγηση.....	60
Σχήμα 4.49: Λειτουργία heading-up με βήμα-βήμα οδηγίες πλοήγησης.....	61
Σχήμα 4.50: Προειδοποιητικό μηνύμα απώλειας σύνδεσης internet κατά την πλοήγηση	62
Σχήμα 4.51: Προειδοποιητικό μηνύμα απώλειας σήματος GPS κατά την πλοήγηση	63
Σχήμα 4.52: Μήνυμα επιτυχούς άφιξης στον προορισμό μετά τον τερματισμό της πλοήγησης.....	64
Σχήμα 4.53: Καρτέλα Γενικών Ρυθμίσεων και μήνυμα επιτυχούς αποθήκευσης αλλαγών	65
Σχήμα 4.54: Μήνυμα επιτυχούς αποθήκευσης αλλαγών στις Γενικές Ρυθμίσεις.....	66
Σχήμα 4.55: Σύγκριση τύπων χάρτη: (α) CardoDB και (β) MapTiler Basic.....	67
Σχήμα 4.56: Καρτέλα Ρυθμίσεων Χάρτη και παράθυρο φόρτωσης κατά την ανανέωση.....	68
Σχήμα 4.57: Καρτέλα Υποστήριξης με επιλογές επικοινωνίας και πολιτική απορρήτου.	69

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΠΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
LBS	Location-Based Services
OSM	OpenStreetMap
ODbL	Open Database License
UI	User Interface
PWA	Progressive Web App
API	Application Programming Interface
GPS	Global Positioning System
JSON	JavaScript Object Notation
IDE	Integrated Development Environment
I/O	Input/Output
iOS	iPhone Operating System
POI	Point of Interest
OSRM	Open Source Routing Machine
DFS	Depth-First Search

Κεφάλαιο 1ο: Εισαγωγή

Οι χάρτες αποτελούσαν παραδοσιακά το κύριο εργαλείο για την παροχή γεωγραφικών πληροφοριών· ωστόσο, είχαν δημιουργηθεί από χαρτογράφους των οποίων το ενδιαφέρον ήταν περισσότερο καλλιτεχνικό παρά επιστημονικό και, ως εκ τούτου, δεν έδιναν έμφαση στην ακριβή απεικόνιση· αλλά οι άνθρωποι δεν αμφισβητούσαν την αναπαράστασή τους, καθώς δεν είχαν λόγο να αμφισβητήσουν την κοσμοθεωρία τους [5]. Η έλευση της ψηφιακής εποχής έχει αλλάξει ριζικά αυτή την προσέγγιση. Η ανάπτυξη του Διαδικτύου, σε συνδυασμό με τις ραγδαίες βελτιώσεις στα smartphone, μετέτρεψε την πλοήγηση σε μια άμεση, ζωντανή διαδικασία. Χάρη στο προηγμένο υλικό, που περιλαμβάνει μεγάλη ταχύτητα επεξεργασίας, πρόσθετη χωρητικότητα αποθήκευσης και υψηλό ρυθμό μεταφοράς δεδομένων, είναι πλέον δυνατή η παροχή Υπηρεσιών Βασισμένων στην Τοποθεσία (Location-Based Services--LBS), δηλαδή η παροχή πληροφοριών προσαρμοσμένων στον πελάτη ανάλογα με την τοποθεσία του, χρησιμοποιώντας γεωγραφικές πληροφορίες [2]. Η πλοήγηση αποτελεί πλέον μέρος της καθημερινής ζωής εκατομμυρίων ανθρώπων σε όλο τον κόσμο. Ο πυρήνας αυτής της πλοήγησης έγκειται στη χρήση της τεχνολογίας GPS, η οποία χρησιμοποιείται για τον ακριβή προσδιορισμό της θέσης σε εξωτερικούς χώρους μέσω της μέτρησης των συντεταγμένων γεωγραφικού πλάτους και μήκους με τη βοήθεια δορυφορικής επικοινωνίας. Τα συστήματα πλοήγησης GPS έχουν αξιοποιηθεί για τη δημιουργία προηγμένων εμπορικών εφαρμογών πλοήγησης, όπως τα Google Maps¹, το Waze², το Garmin³ και το Magellan⁴, που βοηθούν τους χρήστες να πλοηγούνται στους δρόμους χρησιμοποιώντας πληροφορίες σε πραγματικό χρόνο σχετικά με την κυκλοφορία, τη χάραξη διαδρομών και εικόνες που συλλέγονται από αυτοκίνητα με κάμερες [2][3]. Οι εμπορικές τεχνολογίες GPS έχουν μεταμορφώσει την έννοια της πλοήγησης σε μια εντελώς νέα εποχή.

Ωστόσο, τα συστήματα πλοήγησης GPS έχουν ορισμένους περιορισμούς που περιορίζουν τις εφαρμογές τους σε ορισμένες περιπτώσεις. Το πιο σημαντικό είναι ότι η πλοήγηση με βάση το GPS δεν έχει καμία δυνατότητα για πλοήγηση σε εσωτερικούς χώρους, όπου ο προσδιορισμός της θέσης μέσω δορυφορικής επικοινωνίας γίνεται εξαιρετικά δύσκολος. Ένας τέτοιος περιορισμός υπογραμμίζει ένα σημαντικό ζήτημα όπου οι εμπορικές εφαρμογές GPS αποτυγχάνουν να παρέχουν βοήθεια, το οποίο περιλαμβάνει την ανάγκη για υπηρεσίες μικρο-πλοήγησης σε πολύπλοκους θεσμικούς χώρους, όπως τα πανεπιστήμια. Η μικροπλοήγηση περιλαμβάνει την πλοήγηση ενός ατόμου εντός ενός χώρου, όπως η πανεπιστημιούπολη, μετά την άφιξή του στην κεντρική πύλη, βοηθώντας τον να φτάσει σε ένα συγκεκριμένο εργαστήριο, αμφιθέατρο ή διοικητικό κτίριο. Η παρούσα διπλωματική εργασία καλύπτει αυτό το κενό μέσω του σχεδιασμού και της υλοποίησης της εφαρμογής UniMap App, μιας εξειδικευμένης εφαρμογής πλοήγησης για Android που αναπτύχθηκε για τις εγκαταστάσεις του Διεθνούς Ελληνικού Πανεπιστημίου στη Σίνδο.

1.1 Γενικό Πλαίσιο και Αντικείμενο

Τα προβλήματα που προκύπτουν κατά την περιήγηση σε τόσο μεγάλους πανεπιστημιακούς χώρους δεν περιορίζονται σε κάποιο συγκεκριμένο πανεπιστημιακό συγκρότημα. Υπάρχουν γενικά προβλήματα που αντιμετωπίζουν τα πανεπιστήμια σε όλο τον κόσμο, τα οποία οφείλονται σε παράγοντες όπως η

¹ <https://developers.google.com/maps>

² <https://developers.google.com/waze>

³ <https://www.garmin.com/en-US/p/666925/>

⁴ <https://magellan.manymanuals.com/navigators/smartgps-5390>

χωρική πολυπλοκότητα, η έλλειψη ή η ασυνέπεια της φυσικής σήμανσης, καθώς και οι συνεχώς μεταβαλλόμενες διαρρυθμίσεις που προκύπτουν από τις ανακαινίσεις και τις κατασκευές που πραγματοποιούνται τακτικά [1]. Όλοι αυτοί οι παράγοντες σε συνδυασμό δημιουργούν μια κατάσταση στην οποία ακόμη και οι συχνές χρήστες μπορούν να χαθούν, ενώ οι επισκέπτες – είτε πρόκειται για υποψήφιους φοιτητές, υπαλλήλων ή άτομα που επισκέπτονται το πανεπιστημιακό συγκρότημα για ακαδημαϊκούς σκοπούς – αντιμετωπίζουν ακόμη μεγαλύτερους κινδύνους.

Οι τυπικές υπηρεσίες χαρτογράφησης δεν προσφέρουν μεγάλη βοήθεια στο παρόν σενάριο. Αυτό οφείλεται στο γεγονός ότι χρησιμοποιούν ένα μοντέλο δεδομένων κατάλληλο μόνο για δρόμους και μηχανοκίνητα οχήματα, με αποτέλεσμα να στερούνται των λεπτομερειών που απαιτούνται για την ακριβή χαρτογράφηση εσωτερικών διαδρομών, εισόδων κτιρίων και σχέσεων μεταξύ διαφορετικών εγκαταστάσεων εντός του πανεπιστημιακού χώρου [2]. Ένα άλλο πρόβλημα με αυτές τις υπηρεσίες χαρτογράφησης είναι ότι η συχνότητα ενημέρωσης της βάσης δεδομένων δεν ανταποκρίνεται στον ρυθμό των αλλαγών που συμβαίνουν σε επίπεδο εδάφους σε έναν ενεργό πανεπιστημιακό χώρο [1]. Μια πτυχή αυτής της πρόκλησης που συχνά αγνοείται στα βοηθήματα πλοήγησης είναι το ζήτημα της προσβασιμότητας. Για τα άτομα που αντιμετωπίζουν δυσκολίες κινητικότητας, η έλλειψη πληροφοριών σχετικά με τη διαθεσιμότητα ράμπων, ανελκυστήρων και διαδρομών χωρίς σκαλοπάτια καθιστά την πλοήγησή τους ακόμη πιο δύσκολη. Η ενσωμάτωση της προσβάσιμης πλοήγησης δεν αφορά μόνο τη διαθεσιμότητα ακριβών χωρικών πληροφοριών, αλλά και μια στρατηγική χάραξης διαδρομών που μπορεί να ανταποκρίνεται στις ανάγκες κάθε χρήστη [3].

1.2 Προσδιορισμός του Προβλήματος

Η πανεπιστημιούπολη στη Σίνδο αποτελεί χαρακτηριστικό παράδειγμα των προβλημάτων που περιγράφηκαν παραπάνω. Πρόκειται για μια μεγάλη εγκατάσταση που αποτελείται από πολλά διαφορετικά κτίρια, όπως τμήματα, γραμματείες, εργαστήρια, αμφιθέατρα και βιβλιοθήκη. Η μεγάλη πανεπιστημιούπολη διαθέτει πολλαπλές εισόδους και εξόδους, καθώς και ένα περίπλοκο σύστημα εσωτερικών και εξωτερικών διαδρομών. Ενώ διάφορες εμπορικές εφαρμογές GPS είναι σε θέση να καθοδηγήσουν τους ανθρώπους προς την πανεπιστημιούπολη, δεν προσφέρουν καμία βοήθεια πέρα από αυτό το σημείο, αφήνοντας τη διαδικασία εύρεσης του δρόμου προς ένα συγκεκριμένο σημείο εντός της πανεπιστημιούπολης στην ευχέρεια του κάθε χρήστη [1].

Κατά συνέπεια, αυτό έχει σοβαρές επιπτώσεις για τα άτομα που σχετίζονται με το πανεπιστήμιο. Πράγματι, οι φοιτητές – ειδικά οι πρωτοετείς, τα μέλη του προσωπικού, καθώς και οι εξωτερικοί επισκέπτες – συχνά σπαταλούν πολύ χρόνο και ενέργεια προσπαθώντας να βρουν το δρόμο τους στον πανεπιστημιακό χώρο χωρίς τις κατάλληλες οδηγίες. Τέτοια προβλήματα έχουν επιδεινωθεί από την έλλειψη κατάλληλης σήμανσης μέσα στο ίδιο το κτίριο, καθώς και από τις συχνές εργασίες ανακατασκευής και τις αλλαγές στις διαδρομές που δεν ταιριάζουν πάντα με τους διαθέσιμους χάρτες [1]. Η έλλειψη ενός ολοκληρωμένου, εύκολου στη συντήρηση και επαρκώς λεπτομερούς συστήματος μικρο-πλοήγησης για την πανεπιστημιούπολη μπορεί να θεωρηθεί ως το κύριο πρόβλημα που η παρούσα διατριβή επιδιώκει να αντιμετωπίσει. Το σύστημα αυτό δεν θα πρέπει απλώς να εμφανίζει τον χάρτη, αλλά να απεικονίζει το δίκτυο των διαδρομών της πανεπιστημιούπολης με επαρκή λεπτομέρεια, να επιτρέπει τον υπολογισμό διαδρομών μεταξύ οποιωνδήποτε δύο σημείων της πανεπιστημιούπολης, να πληροί συγκεκριμένες απαιτήσεις προσβασιμότητας για άτομα με περιορισμένη κινητική ικανότητα [3] και να διαθέτει δομή που θα επιτρέπει τη συντήρηση της χωρικής βάσης δεδομένων της πανεπιστημιούπολης. Το ζήτημα αυτό οδήγησε στην ανάπτυξη της εφαρμογής UniMap.

1.3 Κίνητρο Διπλωματικής Εργασίας

Βασικό κίνητρο της παρούσας διπλωματικής αποτελεί η πραγματική δυσκολία προσανατολισμού σε μεγάλες πανεπιστημιακές εγκαταστάσεις., Συχνά οι χρήστες καταφέρνουν να φτάσουν εύκολα “μέχρι την είσοδο” του campus, αλλά δυσκολεύονται να εντοπίσουν το ακριβές κτίριο ή χώρο (αμφιθέατρο, εργαστήριο, γραφείο). Η κατάσταση αυτή επηρεάζει ιδιαίτερα τους πρωτοετείς φοιτητές και τους επισκέπτες, προκαλώντας άγχος, καθυστερήσεις και αρνητική εμπειρία κατά την πρώτη επαφή με το campus.

Εξίσου σημαντικό κίνητρο αποτελεί η ενίσχυση της προσβασιμότητας. Ένα σύγχρονο σύστημα micro-navigation που υποστηρίζει σαφείς οδηγίες για προσβάσιμες διαδρομές (π.χ. ράμπες, αποφυγή σκαλιών, κατάλληλες εισοδοί) μπορεί να ενισχύσει την αυτονομία ατόμων με αναπηρίες και να συμβάλει σε ένα πιο συμπεριληπτικό περιβάλλον μετακίνησης για όλους.

Επιπλέον, η ύπαρξη μιας ψηφιακής λύσης πλοήγησης που μπορεί να ενημερώνεται δυναμικά, να επεκτείνεται και να προσαρμόζεται στις αλλαγές της πανεπιστημιούπολης (ανακαινίσεις, νέα κτίρια, αλλαγές χρήσης χώρων) βελτιώνει τη λειτουργικότητα του χώρου και αναβαθμίζει συνολικά την εικόνα του ιδρύματος. Η εργασία φιλοδοξεί να αποτελέσει ένα πρακτικό βήμα προς μια πιο οργανωμένη, φιλική και αποτελεσματική εμπειρία μετακίνησης στην Πανεπιστημιούπολη της Σίνδου.

Παράλληλα, σημαντικό κίνητρο για την εκπόνηση της εργασίας αποτελεί η αξιοποίηση σύγχρονων τεχνολογιών ανάπτυξης εφαρμογών κινητών συσκευών για την επίλυση ενός πραγματικού προβλήματος καθημερινής χρήσης. Η παρούσα διπλωματική δεν περιορίζεται σε θεωρητικό επίπεδο, αλλά στοχεύει στην ανάπτυξη μιας λειτουργικής εφαρμογής με πρακτική αξία για την ακαδημαϊκή κοινότητα. Με τον τρόπο αυτό, συνδυάζει την τεχνολογική υλοποίηση με την κοινωνική χρησιμότητα και δημιουργεί προοπτικές μελλοντικής επέκτασης και αξιοποίησης και σε άλλους οργανωμένους χώρους

Τέλος, το κίνητρο της εργασίας έχει και έντονη κοινωνική διάσταση, καθώς δίνεται ιδιαίτερη έμφαση στην ισότιμη πρόσβαση και στην ίση ευκαιρία για όλους. Η δυνατότητα μετακίνησης και προσανατολισμού στον χώρο δεν θα πρέπει να αποτελεί εμπόδιο για κανέναν· στόχος είναι κανένα άτομο να μη μένει εκτός λόγω περιορισμών προσβασιμότητας.

1.4 Δομή Διπλωματικής Εργασίας

Στο Κεφάλαιο 1 τίθενται οι βάσεις της έρευνας μέσω της παρουσίασης του γενικού πλαισίου και της ιστορικής εξέλιξης των χαρτογραφικών δεδομένων. Προσδιορίζεται η ανάγκη για εξειδικευμένες υπηρεσίες micro-navigation σε εκτεταμένους χώρους, όπως η πανεπιστημιούπολη του ΔΠΑΕ στη Σίνδο, και αναλύεται το πρόβλημα του προσανατολισμού που αντιμετωπίζουν οι χρήστες. Παράλληλα, αναφέρονται τα κίνητρα της εργασίας, με ιδιαίτερη έμφαση στην ενίσχυση της προσβασιμότητας και της συμπερίληψης της μέσω της τεχνολογίας.

Στο Κεφάλαιο 2 πραγματοποιείται ανασκόπηση του υφιστάμενου ανταγωνισμού και των διαθέσιμων υπηρεσιών πλοήγησης, όπως το Google Maps, το Apple Maps και εξειδικευμένες λύσεις όπως το MazeMap και το WheelMap. Μέσα από μια συγκριτική ανάλυση, αναδεικνύονται τα πλεονεκτήματα και οι περιορισμοί των τρεχουσών λύσεων, επιτρέποντας τον προσδιορισμό των κενών της αγοράς και της καινοτομίας που εισάγει η UniMap για το περιβάλλον του campus.

Στο Κεφάλαιο 3 παρουσιάζεται η μεθοδολογία ανάπτυξης της εφαρμογής και τεκμηριώνονται οι σχεδιαστικές επιλογές σχετικά με τις τεχνολογίες και τα εργαλεία υλοποίησης. Περιγράφονται οι

γλώσσες προγραμματισμού και τα εργαλεία που επιλέχθηκαν, ενώ γίνεται σαφής διαχωρισμός μεταξύ των τεχνολογιών back-end για τη διαχείριση της πληροφορίας και των τεχνολογιών front-end για τη δημιουργία της διεπαφής, με στόχο η εφαρμογή να είναι με τέτοιο τρόπο ώστε να καλύπτει όλα τα κενά που υπάρχουν στις υφιστάμενες εφαρμογές και σε όλες τις διαδικτυακές πλατφόρμες.

Στο Κεφάλαιο 4 περιγράφεται αναλυτικά η τεχνική υλοποίηση και η λειτουργία της εφαρμογής UniMap, χωρισμένη σε δύο βασικούς πυλώνες. Στο τμήμα της ανάπτυξης Back-End, παρουσιάζεται η οργάνωση των δεδομένων στη μνήμη και ο προσδιορισμός των οντοτήτων, ενώ δίνεται ιδιαίτερη έμφαση στην υλοποίηση του αλγορίθμου A* και στη δομή γράφου για τον υπολογισμό των βέλτιστων διαδρομών. Παράλληλα, αναλύεται το εξειδικευμένο σύστημα προσβασιμότητας για αμαξίδια και η λογική των API Endpoints που εξυπηρετούν τη δρομολόγηση. Στο τμήμα της ανάπτυξης Front-End, περιγράφεται η αρχιτεκτονική και οι οντότητες της διεπαφής, καθώς και οι προηγμένες τεχνικές εντοπισμού θέσης και προσανατολισμού, όπως η χρήση του GPS, η μέθοδος Dead-Reckoning και η παραμετροποίηση της Navigation Camera. Το κεφάλαιο ολοκληρώνεται με την παρουσίαση των οθονών διεπαφής (UI) και την ανάλυση των περιπτώσεων χρήσης (Use Cases), δίνοντας την τελική εμπειρία του χρήστη κατά την πλοήγηση στην πανεπιστημιούπολη.

Στο Κεφάλαιο 5 συνοψίζονται τα τελικά συμπεράσματα που προέκυψαν από την ολοκλήρωση της εργασίας και πραγματοποιείται μια συνολική αξιολόγηση της προσέγγισης. Το κεφάλαιο ολοκληρώνεται με την παράθεση προτάσεων για μελλοντική εργασία και περαιτέρω βελτιστοποίηση της εφαρμογής, όπως η επέκταση των λειτουργιών πλοήγησης και η ενσωμάτωση πρόσθετων δυνατοτήτων για τον χρήστη.

Κεφάλαιο 2ο: Παρόμοιες Εφαρμογές / Ανάλυση Ανταγωνισμού

2.1 Ανάλυση Υφιστάμενων Εφαρμογών

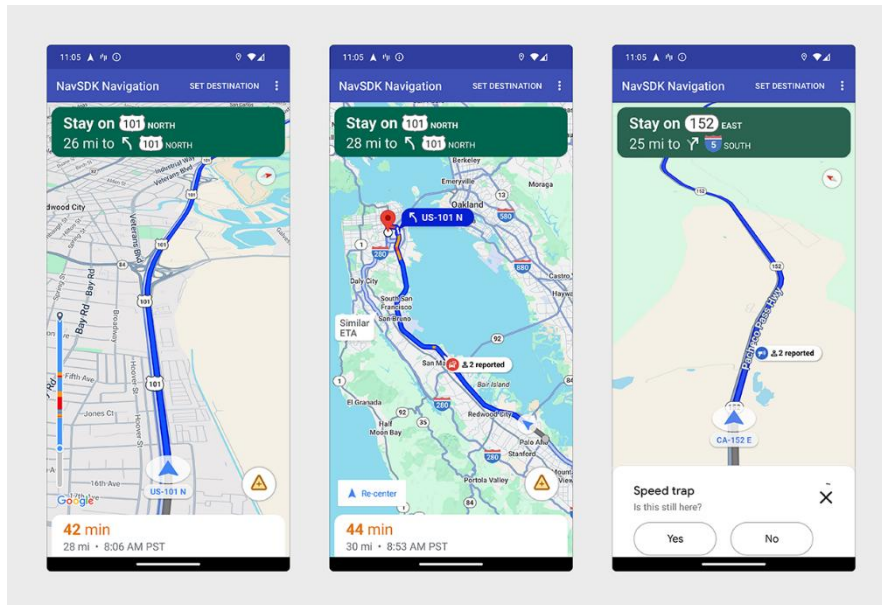
Η ανάλυση των υφιστάμενων εφαρμογών αποτελεί απαραίτητο βήμα για την αξιολόγηση του επιπέδου κάλυψης που προσφέρουν σήμερα οι υπηρεσίες πλοήγησης, ειδικά όταν το ζητούμενο δεν είναι η κλασική μετακίνηση σε οδικά δίκτυα αλλά η καθοδήγηση σε σύνθετους χώρους, όπως οι πανεπιστημιούπολεις. Σνηθισμένα προβλήματα σε τέτοιους χώρους περιλαμβάνουν κτίρια που είναι δύσκολο να βρεθούν λόγω ελλিপών χαρτών, συγκεχυμένων ή λανθασμένων πινακίδων και έλλειψης χρήσιμων χαρακτηριστικών για άτομα με αναπηρίες, όπως σαφείς πληροφορίες για ράμπες για αναπηρικά αμαξίδια ή ανελκυστήρες. Οι πανεπιστημιούπολεις αλλάζουν επίσης συχνά, με νέες κατασκευές ή ανακαινίσεις, αλλά τα τρέχοντα συστήματα σπάνια συμβαδίζουν, αφήνοντας τους ανθρώπους με παλιές ή ανακριβείς πληροφορίες [1]. Στο πλαίσιο αυτό εξετάζονται:

- γενικού σκοπού εφαρμογές όπως τα Google Maps και Apple Maps, που προσφέρουν πλοήγηση υψηλού επιπέδου στην πόλη,
- εξειδικευμένες λύσεις εσωτερικής πλοήγησης όπως το MazeMap, και
- Εφαρμογές με έμφαση στην προσβασιμότητα, όπως το Wheelmap.

2.1.1 Google Maps

Το Google Maps [9], αποτελεί την πλέον διαδεδομένη εφαρμογή χαρτογράφησης και πλοήγησης παγκοσμίως [3]. Αρχικά προγραμματίστηκε στη γλώσσα προγραμματισμού C++ από τους ιδρυτές του, Lars και Jens Eilstrup Rasmussen. Ονομαζόταν «Where 2 Technologies», αλλά το 2004 εξαγοράστηκε από την Google Inc., η οποία μετονόμασε την εφαρμογή σε Google Maps. Αρχικά διέθετε περιορισμένες δυνατότητες που αφορούσαν κυρίως την πλοήγηση, ενώ σήμερα υποστηρίζει προηγμένες λειτουργίες, όπως το Street View και άλλες υπηρεσίες γεωχωρικής απεικόνισης [3]. Το Google Maps χρησιμοποιεί δομή δεδομένων γράφου για τον υπολογισμό της συντομότερης διαδρομής από την πηγή (σημείο A) προς τον προορισμό (σημείο B). Η δομή δεδομένων γραφικών αποτελείται από διάφορους κόμβους και πολλαπλές ακμές που συνδέουν αυτούς τους κόμβους. Οι κόμβοι του γραφήματος συνδέονται με σταθμισμένες ακμές, οι οποίες αντιπροσωπεύουν την απόσταση που πρέπει να διανυθεί για να φτάσει κανείς εκεί. Το Google Maps χειρίζεται τεράστιο όγκο δεδομένων, επομένως ο αριθμός των κόμβων στο γράφημα είναι αμέτρητος [3].

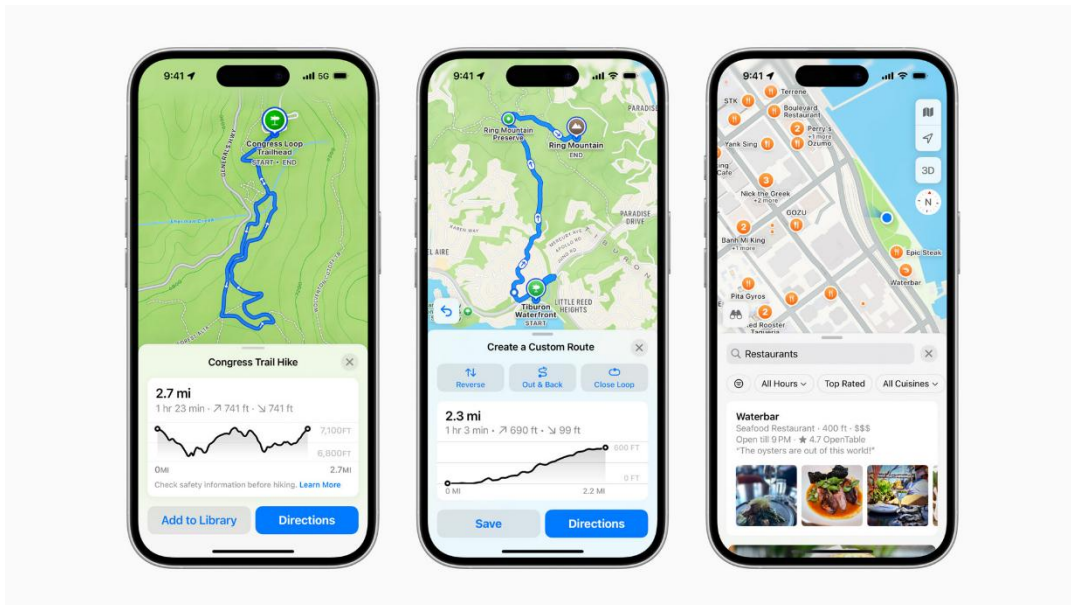
Καθημερινά το χρησιμοποιούν εκατομμύρια χρήστες και έτσι το Google Maps έχει την νούμερο 1 προτίμηση [2]. Το Google Maps χρησιμοποιεί την πολιτική παρακολούθηση GPS και εντοπίζει άτομα και μέρη. Το σύστημα εντοπισμού που χρησιμοποιεί το Παγκόσμιο Σύστημα Εντοπισμού Θέσης είναι το δίκτυο GNSS (Δίκτυο Παγκόσμιου Συστήματος Δορυφορικής Πλοήγησης). Το σύστημα αυτό όχι μόνο αποθηκεύει τη θέση, αλλά διαθέτει και μνήμη για την αποθήκευση της ταχύτητας, της κατεύθυνσης, του χρόνου και άλλων παραμέτρων [3].



Σχήμα 2.1: Απεικόνιση Google Maps [31]

2.1.2 Apple Maps

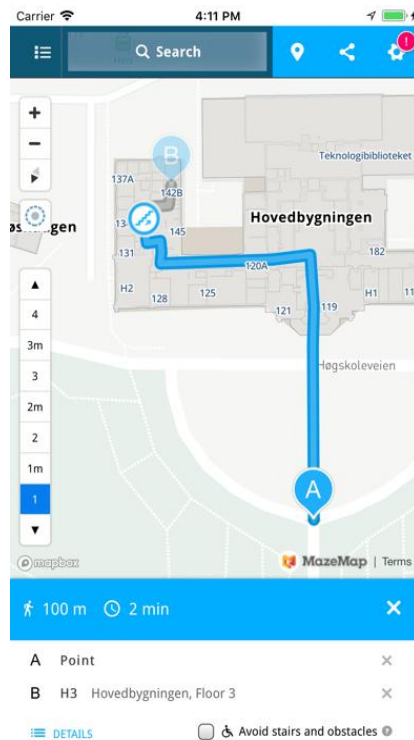
Όταν το iOS 6 κυκλοφόρησε επίσημα τον Σεπτέμβριο του 2012, η αντικατάσταση του Google Maps με το Apple Maps [10] προκάλεσε κριτική εναντίον της επιλογής της Apple, επειδή οι υπηρεσίες του Apple Maps είχαν πολλά ελαττώματα και περιορισμούς σε σύγκριση με το Google Maps (π.χ. σφάλματα χαρτογραφίας, έλλειψη δεδομένων χάρτη, λιγότερες διαθέσιμες επικαλύψεις, κ.α) [4]. Ωστόσο, μετά από τόσα χρόνια, το Apple Maps έχει κάνει σημαντικά βήματα και έχει βελτιώσει κατά πολύ την λειτουργία του. Σημαντική βελτίωση έχει στους δρόμους που συνεχώς προστίθενται και στα νέα μέρη. Σε αυτό το σημείο, αξίζει να σημειώσουμε ότι το Apple Maps δεν μπορεί να «φτάσει» το παγκόσμια διαδεδομένο Google Maps. Δεν μπορούμε να παραλείψουμε ότι το Apple Maps είναι πιο φιλικό στον χρήστη σε πολλά σημεία παρότι έχει λιγότερες λειτουργίες συνολικά. Πιο συγκεκριμένα, οι ελλείψεις του Apple Maps σε σύγκριση με το Google Maps στο iOS περιλαμβάνει τον αριθμό των λεπτομερειών του χάρτη, όπου το Google Maps στο iOS έχει περισσότερα ονόματα οδών, ονόματα κτιρίων κ.λπ. Ουσιαστικά, υπάρχουν λιγότερα σημεία ενδιαφέροντος που δεν είναι καταστήματα ή εστιατόρια στο Apple Maps, κάτι που αποδίδεται στις διαφορετικές πηγές που χρησιμοποιεί το Apple Maps για τα δεδομένα, με μεγαλύτερη επιρροή από το Yelp και το Trip Advisor. αρκετά σημαντικές για να εμφανιστούν όταν είχαμε σμικρύνει τον χάρτη. Ωστόσο, δεν είναι όλα αρνητικά όσον αφορά την εμπειρία του χρήστη, καθώς το Apple Maps χρησιμοποιεί διανυσματικά γραφικά για την απεικόνιση, ενώ το iOS5 Google Maps API χρησιμοποιεί ράστερ γραφικά, πράγμα που σημαίνει πολύ ταχύτερη και ομαλότερη απόδοση για το Apple Maps, καθώς απαιτείται η λήψη λιγότερων δεδομένων [5].



Σχήμα 2.2: Απεικόνιση Apple Maps [32]

2.1.3 MazeMap

Το MazeMap [11], μια υπηρεσία εσωτερικής εντοπισμού θέσης και πλοήγησης, ξεκίνησε ως ένα πρόγραμμα έρευνας και ανάπτυξης. Η πρώτη έκδοση κυκλοφόρησε στις 31 Αυγούστου 2011 με το όνομα CampusGuide. Το MazeMap επιτρέπει στον χρήστη να δει χάρτες των κτιρίων του campus, να εντοπίσει τη θέση του μέσα στο κτίριο, να αναζητήσει όλα τα δωμάτια και διάφορα αντικείμενα (π.χ., τουαλέτες, χώρους στάθμευσης κ.λπ.) και να λάβει αναλυτικές οδηγίες από το σημείο όπου βρίσκεται ο χρήστης μέχρι το σημείο που θέλει να πάει [5]. Είναι η μοναδική εφαρμογή που είναι παγκόσμια και έχει πολλά πανεπιστήμια από όλο τον κόσμο. Το MazeMap είναι μια εφαρμογή που μπορείς να περιηγηθείς στο web μέσω ενός browser αλλά και μέσω του Apple AppStore ή του Google Play Store. Το MazeMap δημιουργήθηκε για να δώσει στους φοιτητές των πανεπιστημίων την αίσθηση ότι μπορούν να πλοηγηθούν γρήγορα και άμεσα. Ακόμη, ολοένα και περισσότεροι φοιτητές κάθε χρόνο πραγματοποιούν μέσω των ευρωπαϊκών προγραμμάτων ανταλλαγής, όπως το Erasmus+ τους δίνεται η δυνατότητα να ταξιδέψουν σε άλλη χώρα, σε άλλο πανεπιστήμιο και πρέπει να υπάρχει μια ενιαία εφαρμογή, γι' αυτό και δημιουργήθηκε το MazeApp. Τέλος, το MazeApp παρέχεται ως λογισμικό ως υπηρεσία, το οποίο επιτρέπει συχνές ενημερώσεις του σχεδιασμού και νέες λειτουργίες χωρίς περιττά σημεία στον χάρτη που θα τον έκαναν δύσκολο [5].



Σχήμα 2.3: Απεικόνιση MazeMap [33]

2.1.4 Wheelmap

Το Wheelmap [12] - ένας χάρτης για χώρους προσβάσιμους σε αναπηρικά αμαξίδια - είναι μια πρωτοβουλία της Sozialhelden, μιας οργάνωσης βάσης από το Βερολίνο της Γερμανίας. Στο Wheelmap όλοι από όλο τον κόσμο μπορούν να βρουν και να προσθέσουν χώρους και να τους αξιολογήσουν χρησιμοποιώντας ένα σύστημα φωτεινού σηματοδότη. Ο χάρτης, ο οποίος είναι διαθέσιμος από το 2010, έχει ως στόχο να βοηθήσει τους χρήστες αναπηρικών αμαξιδίων και τα άτομα με κινητικές αναπηρίες να οργανώσουν την ημέρα τους πιο αποτελεσματικά [7]. Με την παροχή ενός αξιόπιστου και εύχρηστου συστήματος πλοήγησης, το έργο στοχεύει να συμβάλει στη μείωση της απογοήτευσης των χρηστών λόγω της σπατάλης χρόνου κατά την αναζήτηση τοποθεσιών εντός του πανεπιστημίου, στη βελτίωση της προσβασιμότητας για χρήστες με αναπηρίες μέσω της παροχής κατάλληλων διαδρομών για άτομα με αναπηρίες, στην υποστήριξη των προσπαθειών βιωσιμότητας του πανεπιστημίου μέσω της μείωσης των περιττών μετακινήσεων και της βελτιστοποίησης του σχεδιασμού διαδρομών, ενισχύοντας έτσι τη συνολική εμπειρία στο πανεπιστήμιο [1]. Ουσιαστικά η εφαρμογή είναι αποκλειστικά για άτομα με κινητικά προβλήματα αναπηρίας και τους δίνεται η δυνατότητα να πλοηγηθούν με μεγαλύτερη ευκολία. Επίσης, μπορεί ο χρήστης ο οποίος δεν έχει κάποιο πρόβλημα κινητικότητας να βοηθήσει τον χρήστη που έχει κινητικό πρόβλημα να πλοηγηθεί πιο εύκολα με την συμμετοχή του στην χαρτογράφηση μιας εκδήλωσης ή κάποιας διαδρομής.



Σχήμα 2.4: Απεικόνιση Wheelmap [34]

2.2 Δυνατά και Αδύναμα Σημεία Ανταγωνιστικών Λύσεων

Το Google Maps συλλέγει γεωγραφικά και οπτικά δεδομένα μέσω οχημάτων εξοπλισμένων με κάμερες, τα οποία καταγράφουν το οδικό δίκτυο και λαμβάνουν εικόνες από το επίπεδο του εδάφους, επιτρέποντας στην πλατφόρμα να ενημερώνει συνεχώς τους χάρτες της με νέες διαδρομές και πρόσφατη φωτογραφική κάλυψη. Πιο συγκεκριμένα, σήμερα, το Google Maps λαμβάνει υπόψη τις τρέχουσες συνθήκες κυκλοφορίας στην επιλεγμένη διαδρομή, οι οποίες συχνά αποδεικνύονται ότι αυξάνουν τον χρόνο μετακίνησης. Παρέχει στον χρήστη τις τρέχουσες συνθήκες κυκλοφορίας, το οποίο βοηθά τον χρήστη να εκτιμήσει τον χρόνο που απαιτείται για να φτάσει σε οποιονδήποτε συγκεκριμένο προορισμό [3].

Το Apple Maps είναι επίσης μια από τις γνωστές εφαρμογές ωστόσο δεν έχει τόσο άμεση υποστήριξη. Πιο συγκεκριμένα, η διαφορά ποιότητας μεταξύ των δύο υπηρεσιών είναι αρκετά κατανοητή, δεδομένου ότι η Google εφαρμόζει τις υπηρεσίες χαρτογράφησης της εδώ και πολλά χρόνια, συγκεντρώνοντας έναν αριθμό ειδικών, εμπειρογνομοσύνης και γεωγραφικών δεδομένων που δεν έχει αντίπαλο μέχρι σήμερα (αρκεί να σκεφτείτε την κάλυψη με φωτογραφίες σε επίπεδο δρόμου που διατίθεται στη λειτουργία Street View και δεν είναι διαθέσιμη στο Apple Maps ακόμη). Τέλος, το Apple Maps έχει σαφώς πιο όμορφο απεικόνιση στο ευρύ κενό και είναι πιο απλό στην λειτουργία του.

Στον γενικό κανόνα, το Google Maps και το Apple Maps, δίνει στον χρήστη το συναίσθημα ότι δεν θα χαθείς ποτέ και σε βοηθάει με πολλές έξτρα λειτουργίες όπως πλοήγηση με πόδια, αυτοκίνητο ακόμη και με τα Μέσα Μαζικής Μεταφοράς. Ωστόσο, δεν μας δίνει την υποστήριξη που θα θέλαμε σαν πανεπιστημιακή κοινότητά. Πιο συγκεκριμένα, σε αρκετά πανεπιστήμια που βρίσκονται μέσα σε campus δεν υπάρχει πλοήγηση εντός του πανεπιστημίου παρά μόνο εξωτερικά και περιμετρικά. Έχει υποστήριξη σχεδόν πάντα από φωτογραφίες τραβηγμένες από τους χρήστες και από βίντεο. Αυτό από μόνο του δεν αρκεί για την υποστήριξη τόσων πολλών φοιτητών, διδασκόντων και επισκεπτών.

Στην συνέχεια τις αναζήτησης παρόμοιων εφαρμογών παρατηρήσαμε ότι η εφαρμογή MazeMap είναι σε θέση να εντοπίσει τη θέση ενός χρήστη στον πανεπιστημιακό χώρο με ακρίβεια έως 5-10 μέτρων και παρέχει δυνατότητες πλοήγησης σε επίπεδο δωματίου και αναζήτησης αντικειμένων [6]. Η εφαρμογή διαθέτει ιδιαίτερα λεπτομερή αναπαράσταση μιας πανεπιστημιούπολης, υποστηρίζοντας ακριβέστερη πλοήγηση και προσανατολισμό των χρηστών. Στην εφαρμογή MazeMap μας δείχνει κάθε αίθουσα, κάθε γραφείο, κάθε έναν διάδρομο. Είναι πολύ φιλική προς τους φοιτητές, διδάσκοντες και τους επισκέπτες με την δισδιάστατη ή ακόμη και τρισδιάστατη απεικόνιση των αντικειμένων και των

χώρων. Ωστόσο, όπως και σε κάθε εφαρμογή, η απόλυτη υποστήριξη προς όλους δεν είναι εφικτή. Ο φοιτητής για παράδειγμα της Ελλάδας, δεν έχει τέτοια υποστήριξη γιατί σαν εφαρμογή δεν είναι τόσο διαδεδομένη ώστε να μπορεί να χαρτογραφηθεί. Σε ένα πανεπιστήμιο της Κρακοβίας στην Πολωνία είναι λεπτομερώς απεικονισμένο γιατί πιθανόν υπάρχει υποστήριξη. Συνεπώς, ούτε αυτή η εφαρμογή πλοήγησης φοιτητών μπορεί να μας εξυπηρετήσει στο έπακρον. Τέλος, δεν υπάρχει καμία υποστήριξη για τα άτομα με πρόβλημα κίνησης. Κάτι το οποίο είναι σημαντικό για μια εφαρμογή, η ύπαρξη μιας πλήρους απεικόνισης και υποστήριξης σε όλο το κοινό.

2.3 Κενά της Αγοράς και Καινοτομία της Προτεινόμενης Εφαρμογής

Στα αντίστοιχα χρονικά διαστήματα η ανάλυση των υφιστάμενων εφαρμογών πλοήγησης αναδεικνύει ότι, παρότι λύσεις όπως το Google Maps και το Apple Maps είναι εξαιρετικά ώριμες για πλοήγηση σε οδικά δίκτυα και σε περιβάλλον πόλης, δεν καλύπτουν επαρκώς τις ιδιαίτερες ανάγκες πλοήγησης σε χώρους πανεπιστημιούπολης. Η πλοήγηση εντός campus αποτελεί χαρακτηριστικό παράδειγμα “micro-navigation”, όπου ο χρήστης χρειάζεται καθοδήγηση σε μικρές αποστάσεις, σε πεζές διαδρομές, σε διαδρόμους/μονοπάτια, καθώς και σε ακριβή σημεία προορισμού (π.χ. είσοδοι κτιρίων, γραμματείες, συγκεκριμένοι χώροι). Επίσης, το MazeMap δεν είναι πλήρως χαρτογραφημένο και το WheelMap έχει πολλά κενά όσον αναφορά την λειτουργία του.

2.3.1 Κενά της Αγοράς

Η ανάλυση ανέδειξε σαφή κενά στην αγορά, ιδιαίτερα στον τομέα της micro-navigation σε πανεπιστημιακά campus.

Περιορισμένη ακρίβεια “τελικού προορισμού” : Σχεδόν όλες οι εφαρμογές που έχουμε δει και έχουμε μελετήσει οδηγούν σε μια γενική τοποθεσία και όχι στο πρακτικά σημείο που εμείς επιθυμούμε. Για παράδειγμα, μπορεί να αναζητούμε ένα τμήμα ή μια συγκεκριμένη είσοδο και αυτό μας αυξάνει τον χρόνο αναζήτησης και την αβεβαιότητα, ειδικά για τους πρωτοετείς φοιτητές που επισκέπτονται τις πρώτες τους φορές το πανεπιστήμιο.

Ανεπαρκής χαρτογράφηση εσωτερικών και πεζών διαδρομών : Μονοπάτια, ράμπες, είσοδοι-έξοδοι, πεζοδρόμια ή ακόμη και σκαλοπάτια εντός campus, πολλές φορές δεν υπάρχουν καταγεγραμμένα και αν υπάρχουν δεν ενημερώνονται πρόωρα με αποτέλεσμα οι προτεινόμενες διαδρομές να είναι λιγότερο αξιόπιστες σε σχέση με το οδικό δίκτυο μιας πόλης.

Μη ενσωματωμένη προσβασιμότητα στη δρομολόγηση : Αν και υπάρχουν αρκετές προσπάθειες από κάθε έναν δημιουργό εφαρμογής για να υποστηρίξει όλους τους ανθρώπους, δεν είναι πλήρως υλοποιήσιμο. Ακόμη και το WheelMap που είναι ξεκάθαρα για αυτό τον σκοπό, δεν ενσωματώνει πλήρως όλη την χαρτογράφηση (π.χ. σκαλιών, ραμπών κλπ). Έτσι, άτομα με κινητικές δυσκολίες ή χρήστες με καροτσάκι/αμαξίδιο δεν λαμβάνουν πάντα κατάλληλη καθοδήγηση.

Ανομοιογένεια υπηρεσιών μεταξύ ιδρυμάτων : Παρόλο που και γι’ αυτό έχουν συμβάλει πολλές εφαρμογές (ειδικά το WazeMap), όπου μπορούν να προσφέρουν πολύ λεπτομερή πλοήγηση στο campus ενός πανεπιστημίου σε κτήρια και τμήματα. Όμως η κάλυψη δεν είναι δεδομένη για κάθε πανεπιστήμιο και απαιτεί οργανωμένη χαρτογράφηση/υποστήριξη. Αυτό οδηγεί σε άνισο επίπεδο υπηρεσίας από ίδρυμα σε ίδρυμα.

2.3.2 Καινοτομία της προτεινόμενης εφαρμογής (UniMap)

Με βάση τα παραπάνω, προκύπτει ότι οι υπάρχουσες λύσεις είτε είναι γενικού σκοπού είτε δεν εξασφαλίζουν καθολική κάλυψη και προσβασιμότητα σε κάθε campus. Αυτό δημιουργεί ένα σαφές κενό, το οποίο επιχειρεί να καλύψει η προτεινόμενη εφαρμογή UniMap.

Campus-specific δρομολόγηση με γράφο διαδρομών : Η εφαρμογή υιοθετεί προσέγγιση βασισμένη σε γράφο (κόμβοι-ακμές) που αναπαριστά τα πραγματικά πεζά μονοπάτια και τις συνδέσεις μεταξύ σημείων ενδιαφέροντος μέσα στον χώρο. Με αυτόν τον τρόπο η δρομολόγηση γίνεται πιο ελεγχόμενη και προσαρμοσμένη στις ανάγκες της πανεπιστημιούπολης, σε αντίθεση με τη γενική λογική πλοήγησης σε αστικό οδικό δίκτυο.

Σαφής ορισμός προορισμών και “πραγματικού” σημείου άφιξης : Τα τμήματα, οι αίθουσες και γενικά οι προορισμοί ορίζονται με τρόπο που εξυπηρετεί την πραγματική μετακίνηση του χρήστη στο χώρο της πανεπιστημιούπολης. Έτσι, μειώνεται δραματικά το φαινόμενο «είμαι απ έξω, δεν το βλέπω-δεν το βρίσκω» όπου είναι συνηθισμένο είτε βρίσκεται κάποιος εντός του πανεπιστημίου είτε οπουδήποτε.

Ενσωμάτωση προσβασιμότητας στη λήψη απόφασης διαδρομής : Στην εφαρμογή UniMap δίνεται μεγάλη έμφαση στην ίση προσβασιμότητα όλων των ανθρώπων που επισκέπτονται το πανεπιστήμιο. Η εφαρμογή προβλέπει δυνατότητα επιλογής διαδρομών με κριτήρια προσβασιμότητας (π.χ. αποφυγή σκαλοπατιών, προτίμηση ραμπών. Είναι από τις πρώτες προτεραιότητες που έχει η συγκεκριμένη εφαρμογή.

Βελτιωμένη εμπειρία micro-navigation : Το UniMap στοχεύει να καθοδηγεί τον χρήστη σε μικρές αποστάσεις με απλό και άμεσο τρόπο, παρέχοντας σαφείς πληροφορίες για την πορεία και υποστηρικτικές ενέργειες (π.χ. προβολή πληροφοριών προορισμού, μέτρηση βημάτων για την απόσταση, κοινοποίηση τοποθεσίας, χρήσιμα στοιχεία του τμήματος).

Συνολικά, η προτεινόμενη λύση της εφαρμογής UniMap δεν επιχειρεί να αντικαταστήσει τις γενικές εφαρμογές χαρτών - καθώς δεν είναι αυτός ο σκοπός, αλλά να καλύψει το εξειδικευμένο κενό της πλοήγησης εντός μιας μικρής ή μεγάλης πανεπιστημιούπολης, με έμφαση στη micro-navigation, στην αξιοπιστία των πεζών διαδρομών και στην προσβασιμότητα των ανθρώπων με κινητικά προβλήματα.

Κεφάλαιο 3ο: Μεθοδολογία και Σχεδιαστικές Αποφάσεις

Στο συγκεκριμένο κεφάλαιο θα παρουσιαστούν οι τεχνολογίες και τα εργαλεία που επιλέχθηκαν για την υλοποίηση της εφαρμογής UniMap. Μετά τον προσδιορισμό του σκοπού και των λειτουργικών απαιτήσεων το επόμενο στάδιο στον κύκλο ζωής ανάπτυξης της εφαρμογής αφορά στην ανάλυση των τεχνολογικών μέσων που θα χρησιμοποιηθούν. Τα κριτήρια επιλογής των τεχνολογιών που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής περιλάμβαναν διάφορους παράγοντες. Πρώτα απ' όλα, δόθηκε μεγαλύτερη προσοχή στις τεχνολογίες ανοιχτού κώδικα κατά την επιλογή των τεχνολογιών που θα χρησιμοποιηθούν στην ανάπτυξη της εφαρμογής, προκειμένου να αποφευχθούν τυχόν κόστη και έξοδα που σχετίζονται με τις άδειες χρήσης των τεχνολογιών που θα χρησιμοποιηθούν στην ανάπτυξη της εφαρμογής. Η εφαρμογή των τεχνολογιών σε σύγχρονες εφαρμογές ήταν επίσης ένας παράγοντας που λήφθηκε υπόψη κατά την επιλογή των τεχνολογιών που θα χρησιμοποιηθούν στην ανάπτυξη της εφαρμογής, καθώς εξασφάλιζε ότι οι τεχνολογίες ήταν αρκετά σταθερές για να εφαρμοστούν στην ανάπτυξη της εφαρμογής.

Στο παρόν κεφάλαιο αναλύονται οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής UniMap. Αυτές μπορούν να χωριστούν σε δύο μεγάλες κατηγορίες:

- **Front-End Development:** Τεχνολογίες και εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της διεπαφής χρήστη και της λογικής πλοήγησης στην πλευρά του client.
- **Back-End Development:** Τεχνολογίες και εργαλεία που χρησιμοποιήθηκαν για τη δημιουργία του server, τον υπολογισμό διαδρομών και την οργάνωση των γεωγραφικών δεδομένων.

3.1 Επιλογή Γλωσσών Προγραμματισμού

Όσον αφορά τις γλώσσες προγραμματισμού για την ανάπτυξη τόσο του front-end όσο και του back-end της εφαρμογής UniMap, επιλέχθηκε η TypeScript. Είναι σημαντικό να σημειωθεί ότι η επιλογή μιας μόνο γλώσσας προγραμματισμού και για τα δύο επίπεδα της εφαρμογής δεν ήταν τυχαία, αλλά είχε πολλά πλεονεκτήματα, όπως αναλύεται παρακάτω λεπτομερώς.

3.1.1 Γλώσσα Προγραμματισμού TypeScript

Η TypeScript [13], είναι μια στατικά τυποποιημένη αντικειμενοστραφής γλώσσα προγραμματισμού που αναπτύχθηκε από τη Microsoft και λειτουργεί ως υπερσύνολο της JavaScript. Αυτό σημαίνει ότι όλος ο έγκυρος κώδικας JavaScript μπορεί να θεωρηθεί έγκυρος κώδικας TypeScript χωρίς να απαιτούνται αλλαγές.

Τα κύρια οφέλη που προσφέρει το TypeScript σε αυτό το έργο είναι:

- **Ασφάλεια τύπου:** Η δυνατότητα ορισμού διεπαφών και τύπων για κρίσιμες δομές δεδομένων όπως LatLng, RouteResult και CampusRouteRequest εξασφαλίζει ότι εάν υπάρξουν κάποια προγραμματιστικά λάθη εντοπίζονται κατά τη διάρκεια της ανάπτυξης και όχι κατά την εκτέλεση.
- **Ομοιομορφία κώδικα:** Η δυνατότητα χρήσης της ίδιας γλώσσας προγραμματισμού τόσο για το front-end (Angular) όσο και για το back-end (Node.js/Express.js) εξασφαλίζει ότι δεν υπάρχουν προβλήματα ασυμβατότητας δεδομένων μεταξύ των δύο επιπέδων και φαίνεται πιο καθαρό και αναγνώσιμο το κάθε αρχείο κώδικα.
- **Συντηρησιμότητα κώδικα:** Η στατική τυποποίηση εξασφαλίζει ότι ο κώδικας είναι αυτονόητος και κατανοητός, ειδικά λαμβάνοντας υπόψη την πολυπλοκότητα που συνεπάγεται η διαχείριση διαδρομών.

- **Υποστήριξη κώδικα:** Η TypeScript διαθέτει εξαιρετική τεκμηρίωση, υποστήριξη και ενσωμάτωση με τα περισσότερα σύγχρονα ολοκληρωμένα περιβάλλοντα ανάπτυξης (IDE).

Στο σχήμα 3.1 φαίνεται ένα χαρακτηριστικό παράδειγμα ορισμού τύπων από το βασικό αρχείο του back-end (αρχείο types.ts), όπου ορίζονται οι βασικές δομές δεδομένων της εφαρμογής.

```
export interface LatLng {
  lat: number;
  lng: number;
}

export type Adjacency = Record<string, Record<string, number>>;

export type EdgeTag = 'ALL' | 'STAIRS' | 'RAMP';

export interface RouteResult {
  path: LatLng[];
  lengthM: number;
}

export interface CampusRouteRequest {
  fromLat: number;
  fromLng: number;
  destinationName?: string;
  destLat?: number; // fallback αν δεν βρεθεί με όνομα
  destLng?: number;
  wheelchair?: boolean;
}
```

Σχήμα 3.1: Ορισμός των βασικών τύπων δεδομένων και δομών της εφαρμογής

Η χρήση του τύπου EdgeTag ως union type ('ALL' | 'STAIRS') αποτελεί χαρακτηριστικό παράδειγμα όπως παρουσιάζεται στο Σχήμα 3.2, πώς η TypeScript εξασφαλίζει ότι κάθε ακμή του γράφου μπορεί να πάρει μόνο μία από τις προκαθορισμένες τιμές (ALL, STAIRS, RAMP), αποτρέποντας λανθασμένες τιμές κατά τη δρομολόγηση για χρήστες με αναπηρία κινητικότητας που στην δικιά μας εφαρμογή είναι σημαντικό.

```
export type EdgeTag = 'ALL' | 'STAIRS' | 'RAMP';
```

Σχήμα 3.2: Ορισμός του union type EdgeTag για τον περιορισμό των επιτρεπτών τιμών στις ακμές του γράφου.

3.2 Επιλογή Εργαλείων και Τεχνολογιών

Πέρα από τη γλώσσα προγραμματισμού TypeScript, η ανάπτυξη του UniMap βασίστηκε σε μια σειρά από εξειδικευμένα frameworks, βιβλιοθήκες και εργαλεία, τόσο για το front-end όσο και για το back-end. Οι επιλογές αυτές αναλύονται στις επόμενες ενότητες.

3.2.1 Τεχνολογίες Back-End

Το back-end της εφαρμογής UniMap λειτουργεί ως server υπολογισμού διαδρομών και παροχής γεωγραφικών δεδομένων. Οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη του αναλύονται παρακάτω.

3.2.1.1 Node.js

Το Node.js [14] είναι ένα περιβάλλον εκτέλεσης JavaScript/TypeScript εκτός περιηγητή (runtime environment), βασισμένο στη μηχανή V8 της Google. Επιτρέπει την ανάπτυξη server-side εφαρμογών χρησιμοποιώντας JavaScript ή TypeScript, ενώ το μοντέλο μη-αποκλεισμού (non-blocking I/O) το καθιστά ιδιαίτερα αποδοτικό για εφαρμογές που διαχειρίζονται πολλαπλά ταυτόχρονα αιτήματα.

Για το UniMap, το Node.js επιλέχθηκε διότι επιτρέπει τη χρήση TypeScript απευθείας στο server, διασφαλίζοντας την ομοιομορφία κώδικα με το front-end. Επιπλέον, η ελαφριά φύση του Node.js το καθιστά κατάλληλο για server που αναλαμβάνει κυρίως υπολογισμούς σε μνήμη (in-memory graph traversal), χωρίς βαριά I/O λειτουργία.



Σχήμα 3.3: Λογότυπο Node.js

3.2.1.2 Express.js

Το Express.js [15] είναι ένα minimalist web framework για Node.js, το οποίο παρέχει τα απαραίτητα εργαλεία για τη δημιουργία RESTful API με απλό και κατανοητό τρόπο. Χαρακτηρίζεται από ευελιξία, ελαφριά αρχιτεκτονική και τεράστια κοινότητα χρηστών.



Σχήμα 3.4 Λογότυπο framework ExpressJS.

Στο UniMap, το Express χρησιμοποιείται για τον ορισμό τριών ομάδων endpoints, οργανωμένων σε ξεχωριστά Router modules:

- `/api/destinations` — Επιστρέφει τη λίστα ή τα στοιχεία ενός συγκεκριμένου προορισμού.
- `/api/route/campus` — Υπολογίζει και επιστρέφει τη βέλτιστη διαδρομή εντός campus με τον αλγόριθμο A*.
- `/api/route/outdoor` — Διαχειρίζεται αιτήματα δρομολόγησης για εξωτερικές διαδρομές.

Παρακάτω στο Σχήμα 3.5, φαίνεται η αρχικοποίηση του Express server και η οργάνωση των routes αρχείο `app.ts`:

```
import express from 'express';
import cors from 'cors';
import apiRouter from './routes';

const app = express();

app.use(cors());
app.use(express.json());

app.get('/health', (_req, res) => {
  res.json({ status: 'ok' });
});

app.use('/api', apiRouter);
```

Σχήμα 3.5: Αρχικοποίηση του διακομιστή Express και ρύθμιση των βασικών παραμέτρων του backend

3.2.1.3 Αλγόριθμος A*

Ο αλγόριθμος A* [29] παρουσιάστηκε από τους Peter Hart, Nils Nilsson και Bertram Raphael το 1968 και βασίζεται στη χρήση μιας ευριστικής μεθόδου. Ο αλγόριθμος A* είναι ένας αλγόριθμος αναζήτησης που λαμβάνει δεδομένα εισόδου, αξιολογεί ορισμένες πιθανές διαδρομές και επιστρέφει τη λύση. Ο A* χρησιμοποιείται ευρέως στην εύρεση διαδρομών και στη διάσχιση γραφημάτων, δηλαδή στη διαδικασία χάραξης μιας διαδρομής που μπορεί να διανυθεί αποτελεσματικά μεταξύ σημείων. Ο A* είναι ένας αλγόριθμος αναζήτησης «best-first» (πρώτα το καλύτερο) που τροποποιεί τη ευρετική συνάρτηση. Αυτός ο αλγόριθμος ελαχιστοποιεί το συνολικό κόστος της διαδρομής και, υπό τις σωστές συνθήκες, παρέχει την καλύτερη λύση σε βέλτιστο χρόνο [8]. Στην αρχή της ανάπτυξης της εφαρμογής χρησιμοποιήθηκε ο αλγόριθμος Dijkstra αλλά μιας που η εφαρμογή χειρίζεται αρκετούς κόμβους, η δραστηκή αύξηση του μεγέθους του γραφήματος περιορίζει την αποτελεσματικότητα του αλγορίθμου. Έτσι, αλγόριθμοι γραφημάτων όπως ο αλγόριθμος του Dijkstra ενδέχεται να αποτύχουν λόγω της αυξημένης χρονικής και χωρικής πολυπλοκότητας. Ο αλγόριθμος A* είναι παρόμοιος με τον αλγόριθμο Dijkstra, ο οποίος χρησιμοποιεί μια ευριστική συνάρτηση για να πλοηγηθεί σε μια καλύτερη και πιο αποδοτική διαδρομή. Ο αλγόριθμος A* αποδίδει υψηλότερη προτεραιότητα στους κόμβους που θεωρούνται καλύτεροι (ελέγχει παραμέτρους όπως απαιτήσεις χρόνου, απόσταση και άλλες παρόμοιες παραμέτρους) από τους άλλους, ενώ ο Dijkstra εξερευνά όλους τους κόμβους [3].

Η συνάρτηση που φαίνεται στο Σχήμα 3.6, αναπαρίσταται η `calculatePathWithLength` αποτελεί την κεντρική συνάρτηση δρομολόγησης:

```
export function calculatePathWithLength(
  startNodeId: string,
  endNodeId: string,
  opts?: { wheelchair?: boolean }
): RouteResult | null {
  const points = calculatePath(startNodeId, endNodeId, opts);
  if (!points || points.length < 2) return null;

  let len = 0;
  for (let i = 1; i < points.length; i++)
    len += distanceTo(points[i - 1], points[i]);

  return { path: points, lengthM: Math.round(len) };
}
```

Σχήμα 3.6: Υπολογισμός των σημείων της διαδρομής και του συνολικού της μήκους σε μέτρα.

3.2.1.4 OpenStreetMap

Το Open-StreetMap [21] είναι ένας ανοιχτός, επεξεργάσιμος χάρτης του ψηφιακού χάρτη ανοιχτού κώδικα του κόσμου. Όλοι μπορούν να αναζητήσουν τοποθεσίες και, εφόσον έχουν επισημανθεί, να λάβουν πληροφορίες σχετικά με το πόσο εύκολα προσβάσιμες είναι αυτές οι τοποθεσίες. Όσοι εγγράφονται ως χρήστες μπορούν να προσθέτουν και να αξιολογούν νέες τοποθεσίες [7]. Παρέχει ελεύθερα γεωγραφικά δεδομένα υπό ανοιχτή άδεια χρήσης (Open Database Library--ODbL), καθιστώντας το ιδανική επιλογή για εφαρμογές που απαιτούν χαρτογραφικά δεδομένα χωρίς κόστος αδειοδότησης.

Στο UniMap, τα δεδομένα OSM χρησιμοποιούνται σε δύο επίπεδα: πρώτον, τα δεδομένα του γράφου δρόμων (κόμβοι και ακμές) εξάγονται και αποθηκεύονται τοπικά στο back-end (αρχείο `osm-nodes.ts`), σχηματίζοντας τη βάση του δικτύου μονοπατιών του campus. Δεύτερον, οι χαρτογραφικοί tiles εμφανίζονται στη διεπαφή του χρήστη μέσω της πλατφόρμας MapTiler, η οποία βασίζεται επίσης σε δεδομένα OSM.

3.2.1.5 Render

Το Render [23] είναι μια πλατφόρμα cloud hosting που επιτρέπει την ανάπτυξη εφαρμογών server-side με ελάχιστη διαμόρφωση. Υποστηρίζει άμεση ανάπτυξη από ένα αποθετήριο GitHub, αναγνωρίζει αυτόματα το περιβάλλον (Node.js σε αυτή την περίπτωση) και προσφέρει αυτόματο HTTPS, δημόσια διεύθυνση URL και υποστήριξη μεταβλητών περιβάλλοντος. Το Render επιλέχθηκε για να φιλοξενήσει τον back-end server για την εφαρμογή UniMap για τους ακόλουθους λόγους:

Αρχικά, προσφέρει ένα δωρεάν πακέτο φιλοξενίας που είναι επαρκές για το μέγεθος της εφαρμογής χωρίς να χρειάζεται κάποια δυσκολία στη διαχείριση υποδομών. Επιπλέον ενσωματώνεται άψογα με το αποθετήριο GitHub, επιτρέποντας την αυτόματη δημοσίευση της νέας έκδοσης της εφαρμογής κάθε φορά που νέος κώδικας εισάγεται στο αποθετήριο, περιορίζοντας την ανάγκη στη διαδικασία ανάπτυξης και συντήρησης. Τέλος το Render προσφέρει μια δημόσια διεύθυνση URL μέσω HTTPS για τον διακομιστή, η οποία χρησιμοποιείται από το front-end για τη δρομολόγηση αιτημάτων όταν η εφαρμογή διατίθεται σε ένα εγγενές περιβάλλον χρησιμοποιώντας Capacitor.



Σχήμα 3.7: Λογότυπο πλατφόρμας Render

3.2.2 Τεχνολογίες Front-End

Το front-end της εφαρμογής UniMap αναπτύχθηκε ως Progressive Web App (PWA), δηλαδή μια εφαρμογή ιστού που συμπεριφέρεται ως native εφαρμογή σε κινητές συσκευές. Οι τεχνολογίες που χρησιμοποιήθηκαν αναλύονται στο παρακάτω Σχήμα(3.8).

```
const routes: Routes = [
  {
    path: 'home',
    loadComponent: () =>
      import('./home/home.page').then(m => m.HomePage)
  },
  {
    path: '',
    redirectTo: 'splash',
    pathMatch: 'full'
  },
  {
    path: 'splash',
    loadChildren: () =>
      import('./splash/splash.module').then(m => m.SplashPageModule)
  },
];
```

Σχήμα 3.8: Ορισμός των βασικών διαδρομών (routes) περιήγησης για τη διεπαφή του χρήστη.

3.2.2.1 Angular

Το Angular framework [16] είναι ένα ολοκληρωμένο και ώριμο πλαίσιο ανοιχτού κώδικα για front-end, το οποίο συντηρείται και υποστηρίζεται από την Google. Διακρίνεται για την αρχιτεκτονική του που βασίζεται σε στοιχεία, τον μηχανισμό έγχυσης εξαρτήσεων και την υποστήριξη για TypeScript, καθιστώντας το κατάλληλο για την ανάπτυξη σύνθετων εφαρμογών με καθαρή και διαχωρισμένη αρχιτεκτονική. Στο πλαίσιο του έργου UniMap, το Angular χρησιμοποιείται για τη δομή της εφαρμογής σε Υπηρεσίες (MapService, RouteService, GpsService, ApiService) και Σελίδες/Στοιχεία (HomePage, SplashPage, SearchBarComponent, DepartmentPopurComponent, κ.λπ.). Ο μηχανισμός δρομολόγησης χρησιμοποιείται για την πλοήγηση στην οθόνη, όπως φαίνεται στο αρχείο app-routing.module.ts:

3.2.2.2 Ionic Framework

Το Ionic [17] είναι ένα δωρεάν και ανοιχτού κώδικα framework ανάπτυξης διεπαφής χρήστη ανοιχτού κώδικα που επιτρέπει τη δημιουργία εφαρμογών για πολλαπλές πλατφόρμες κινητών συσκευών. Προσφέρει μια σειρά από έτοιμα στοιχεία διεπαφής χρήστη, όπως modals, ειδοποιήσεις, κουμπιά και εικονίδια, τα οποία είναι κατάλληλα για κινητές συσκευές και εξασφαλίζουν μια σωστή εμφάνιση και αίσθηση τόσο σε συσκευές iOS όσο και Android. Είναι πλήρως συμβατό με το Angular και επιτρέπει την ανάπτυξη της εφαρμογής είτε ως PWA είτε ως native app μέσω του Capacitor.

Στην περίπτωση της εφαρμογής UniMap, το Ionic χρησιμοποιείται για τα ακόλουθα περιβάλλοντα εργασίας χρήστη: το modal πληροφοριών τμήματος, το modal ρυθμίσεων, το modal πολιτικής απορρήτου και η σειρά μηνυμάτων toast και ειδοποιήσεων που εμφανίζονται κατά τη λειτουργία.



Σχήμα 3.9: Λογότυπο framework Ionic.

3.2.2.3 Capacitor

Το Capacitor [18] είναι ένα cross-platform native runtime που επιτρέπει σε διαδικτυακές εφαρμογές να αποκτήσουν πρόσβαση σε native APIs κινητών συσκευών, όπως το GPS, η κάμερα και οι ειδοποιήσεις.

Στο UniMap, το Capacitor χρησιμοποιείται για δύο κρίσιμες λειτουργίες: αρχικά, για την πρόσβαση στο GPS της συσκευής μέσω του plugin @capacitor/geolocation, που επιτρέπει την παρακολούθηση της θέσης του χρήστη σε πραγματικό χρόνο. Επιπλέον, για τη χρήση του CapacitorHttp αντί του κανονικού HttpClient [30] της Angular σε native περιβάλλον, αποφεύγοντας περιορισμούς CORS. Παρακάτω στο Σχήμα 3.10, φαίνεται η υλοποίηση στο ApiService:

```
getCampusRoute(params: CampusRouteParams): Promise<CampusRouteResponse> {
  if (Capacitor.isNativePlatform()) {
    return CapacitorHttp.post({
      url: `${this.baseUrl}/api/route/campus`,
      headers: { 'Content-Type': 'application/json' },
      data: params,
    }).then(r => r.data as CampusRouteResponse);
  }
  return firstValueFrom(
    this.http.post<CampusRouteResponse>(
      `${this.baseUrl}/api/route/campus`, params
    )
  );
}
```

Σχήμα 3.10: Υλοποίηση του HTTP αιτήματος για τη λήψη των δεδομένων δρομολόγησης από τον διακομιστή.

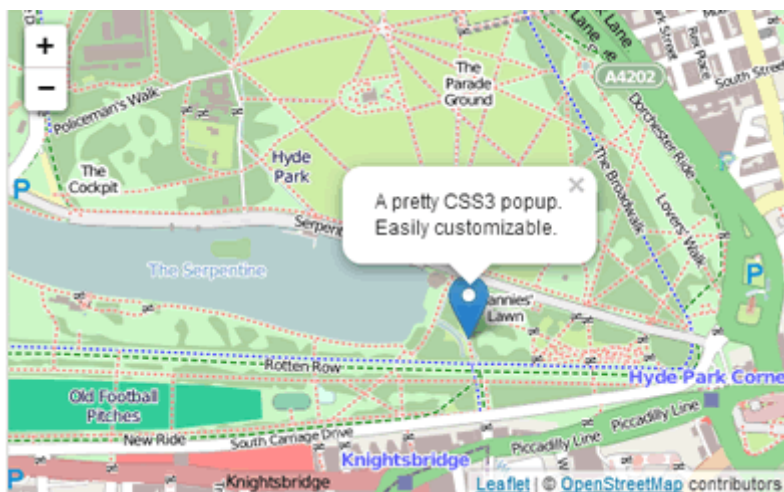
3.2.2.4 Leaflet.js

Η Leaflet.js [19] είναι από τις δημοφιλέστερες βιβλιοθήκες της JavaScript για διαδραστικούς χάρτες, η οποία είναι ανοιχτού κώδικα χωρίς χρέωση χρήσης. Διακρίνεται για το μικρό της μέγεθος (~42KB), την υψηλή απόδοση σε κινητές συσκευές και το εκτεταμένο API για τη διαχείριση δεικτών (markers), των polylines, των tile layers και map events.

Η απόφαση να χρησιμοποιηθεί η Leaflet.js αντί για άλλες επιλογές όπως το Google Maps ή το Mapbox GL βασίστηκε στους ακόλουθους λόγους:

- Η Leaflet.js δεν έχει χρέωση χρήσης και είναι πλήρως ανοιχτού κώδικα.
- Παρέχει πλήρη έλεγχο της απόδοσης.
- Επιτρέπει τη χρήση οποιουδήποτε tile layer.

Η ευελιξία της Leaflet.js χρησιμοποιήθηκε για την ενσωμάτωση λειτουργιών όπως η λειτουργία heading-up κατά την οποία ο χρήστης μπορεί να προσανατολιστεί με βάση την κίνηση του, η οποία επιτυγχάνεται με μετασχηματισμό SCSS, και η ομαλή κίνηση του marker του χρήστη με requestAnimationFrame ώστε η ενημέρωση της θέσης του χρήστη να εμφανίζεται πιο φυσική και να είναι συνεχής.



Σχήμα 3.11: Η βιβλιοθήκη Leaflet.js για την ανάπτυξη διαδραστικών χαρτών

3.2.2.5 MapTiler

Το MapTiler [20] είναι μια πλατφόρμα παροχής map tiles, βασισμένη σε δεδομένα OpenStreetMap. Παρέχει tiles υψηλής ποιότητας με διάφορα στυλ εμφάνισης (OpenStreetMap, Outdoor, Dark, Positron) μέσω ενός απλού API key.

Στο πλαίσιο της εφαρμογής UniMap, τα map tiles που παρέχονται από την υπηρεσία MapTiler χρησιμοποιούνται ως προεπιλεγμένη επιλογή, προσφέροντας τη δυνατότητα αλλαγής του στυλ των tiles μέσω των ρυθμίσεων χρήστη. Η απόφαση να βασιστεί η υπηρεσία στα tiles που παρέχει η MapTiler αντί για τον διακομιστή tiles του OpenStreetMap οφείλεται στην ποιότητα των tiles που παρέχει η MapTiler, καθώς και στη δυνατότητα υποστήριξης επιπέδων ζουμ έως 22, σε αντίθεση με τα tiles του OpenStreetMap που δεν μπορούν να ξεπεράσουν το επίπεδο 19, επιτρέποντας έτσι στον χρήστη να έχει μια λεπτομερή εικόνα των δρόμων, των σημείων και άλλων χαρακτηριστικών που εμφανίζονται στον χάρτη.

3.2.2.6 GeoJSON

Το GeoJSON [22] είναι ένα ανοικτό πρότυπο μορφοποίησης γεωχωρικών δεδομένων που βασίζεται στη μορφή αρχείου JSON για τον ορισμό γεωγραφικών χαρακτηριστικών. Το GeoJSON υποστηρίζει τα ακόλουθα γεωμετρικά χαρακτηριστικά: σημεία (Point), γραμμές (LineString) και πολύγωνα (Polygon).

Στο πλαίσιο της εφαρμογής UniMap, τα αρχεία GeoJSON χρησιμοποιούνται για τον ορισμό των ορίων των περιοχών του πανεπιστημίου (campus.geojson, campus_clean.geojson, components.geojson) καθώς και των σημείων ενδιαφέροντος (campus_points.json). Αυτά τα αρχεία φορτώνονται από το front-end της εφαρμογής για να ορίσουν την περιοχή στην οποία μπορεί να πλοηγηθεί ο χρήστης.

3.2.2.7 Git

Το Git [24] είναι ένα σύστημα ελέγχου εκδόσεων ανοιχτού κώδικα που μπορεί να χρησιμοποιηθεί για την παρακολούθηση αλλαγών στον κώδικα, τη διαχείριση παράλληλων κλάδων ανάπτυξης και την επαναφορά του κώδικα σε προηγούμενη έκδοση. Επίσης, το Git είναι ένα πολύ γνωστό εργαλείο διότι μπορεί να οργανώσει τις εκδόσεις του project και σε περίπτωση που πάνω από ένας προγραμματιστής πρέπει ταυτόχρονα να συνεργάζεται στο ίδιο project.

Αρχικά το Git [25] χρησιμοποιήθηκε για την παρακολούθηση των αλλαγών του κώδικα καθόλη την διάρκεια τη διάρκεια ανάπτυξης της εφαρμογής. Το αποθετήριο του project ανέβηκε στο GitHub ώστε να υπάρχει online αποθήκευση και να είναι πάντα ασφαλής η διαχείριση και του project αλλά και των εκδόσεων. Η ανάπτυξη οργανώθηκε με το βασικό main brunch, στον οποίο υπήρχε η πιο σταθερή έκδοση της εφαρμογής. Μιας και στο project υπήρχε ένας και μοναδικός προγραμματιστής δεν δημιουργήθηκαν άλλα branches και δεν ελέγχονταν ο κώδικας καθώς ήταν περιττό. Οι αλλαγές αποθηκεύονταν σταδιακά με commits, έτσι ώστε κάθε βήμα της ανάπτυξης να είναι καταγεγραμμένο και ανακτήσιμο. Κεφάλαιο Κάθε commit συνοδεύεται από περιγραφικό μήνυμα, το οποίο δήλωνε με σαφήνεια το είδος της αλλαγής, όπως προσθήκη νέας λειτουργίας, διόρθωση σφάλματος ή βελτίωση διεπαφής. Μέσω του ιστορικού των commits ήταν δυνατή η εύκολη παρακολούθηση της εξέλιξης του έργου και η επιστροφή σε προηγούμενη κατάσταση όταν αυτό χρειαζόταν.

Για την εκτέλεση των βασικών ενεργειών χρησιμοποιήθηκαν εντολές όπως git add, git commit, git push, καθώς και μηχανισμοί για αναίρεση του push ή για αλλαγή του commit. Επίσης η χρήση του GitHub [26] λειτούργησε επίσης ως μέσο backup και ασφάλειας, αφού το έργο παρέμενε αποθηκευμένο και διαθέσιμο online. Η συνολική χρήση Git και GitHub συνέβαλε στη συστηματική οργάνωση της ανάπτυξης, στη συντήρηση του κώδικα και στη σταδιακή εξέλιξη της εφαρμογής με ελεγχόμενο τρόπο.

```
D:\UniMap\UniMap>git add src/app/services/route.service.ts src/app/services/map.service.ts src/app/home/home.page.ts src
/app/home/home.page.html src/app/components/department-popup/department-popup.component.ts src/app/components/department
-popup/department-popup.component.html src/app/components/department-popup/department-popup.component.scss
warning: in the working copy of 'src/app/services/map.service.ts', LF will be replaced by CRLF the next time Git touches
it

D:\UniMap\UniMap>git commit -m "display walked distance UI and persistent polyline trail"
[main 3715b53] display walked distance UI and persistent polyline trail
 7 files changed, 50 insertions(+)

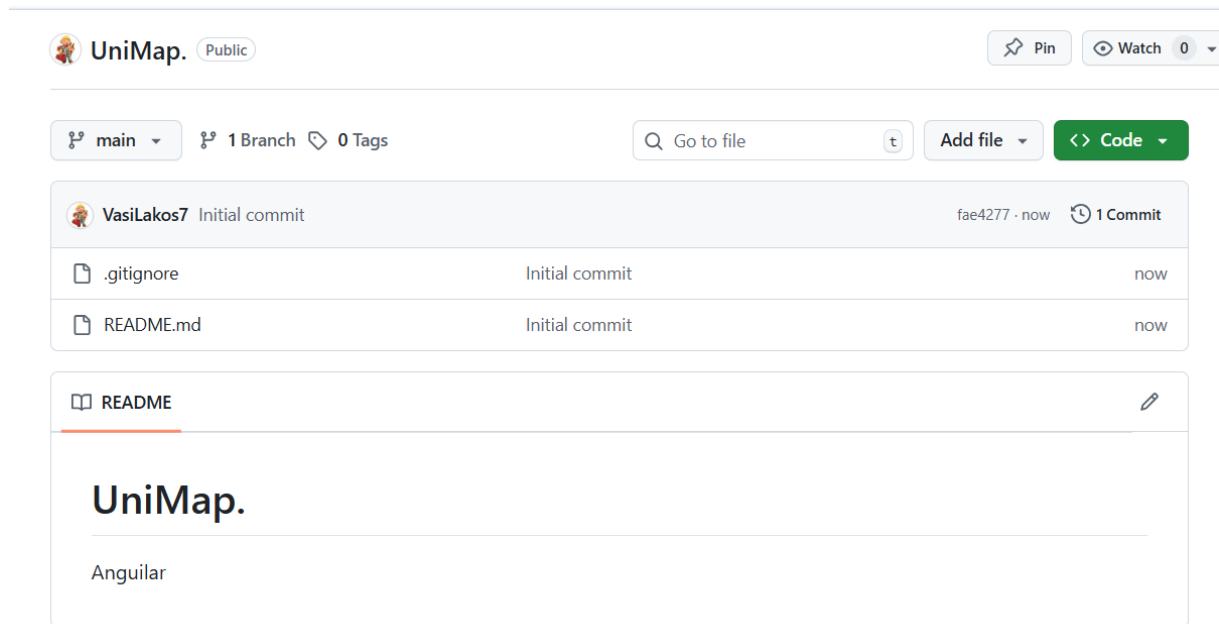
D:\UniMap\UniMap>git push
Enumerating objects: 29, done.
Counting objects: 100% (29/29), done.
Delta compression using up to 8 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (15/15), 1.90 KiB | 650.00 KiB/s, done.
Total 15 (delta 14), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (14/14), completed with 14 local objects.
To https://github.com/Vasilakos7/UniMap.git
 0275689..3715b53  main -> main

D:\UniMap\UniMap>
```

Σχήμα 3.12: Git με χρήση Command Line

Στο Σχήμα 3.13.1 παρουσιάζεται η αρχική εικόνα του repository της εφαρμογής στο GitHub. Μέσα από τη σελίδα αυτή είναι δυνατή η συνολική επισκόπηση του έργου, καθώς εμφανίζονται τα αρχεία του project, η δομή του αποθετηρίου, καθώς και βασικές πληροφορίες που σχετίζονται με την ανάπτυξή του. Στο κάτω μέρος της αρχικής σελίδας βρίσκεται το αρχείο README, το οποίο λειτουργεί ως συνοδευτικό πληροφοριακό έγγραφο του repository. Στο αρχείο αυτό περιγράφονται συνοπτικά ο

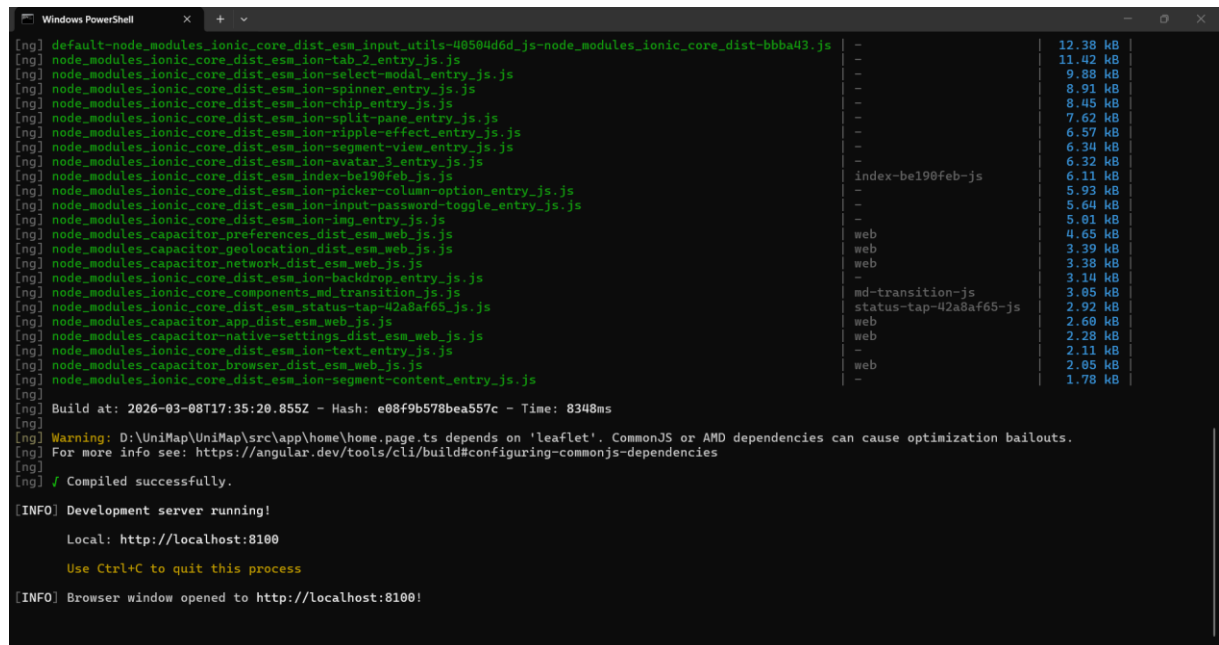
σκοπός του έργου, ο τρόπος εκτέλεσής του, τυχόν βασικές οδηγίες εγκατάστασης και χρήσης, αλλά και οποιαδήποτε επιπλέον στοιχεία θεωρούνται χρήσιμα για την κατανόηση και αξιοποίηση του κώδικα.



Σχήμα 3.13.1: Περιβάλλον GitHub

3.2.2.8 Εκτέλεση και συγχρονισμός εφαρμογής σε Android μέσω Capacitor

Για την αρχική δοκιμή της εφαρμογής κατά την ανάπτυξη χρησιμοποιήθηκε η εντολή `ionic serve`. Μέσω αυτής η εφαρμογή εκτελούνταν τοπικά σε περιβάλλον browser, επιτρέποντας γρήγορο έλεγχο της διεπαφής και των αλλαγών στον κώδικα.



Σχήμα 3.13.2: Εκκίνηση της εφαρμογής σε τοπικό περιβάλλον.

Για την προετοιμασία της εφαρμογής για εκτέλεση σε περιβάλλον Android χρησιμοποιήθηκαν διαδοχικά οι εντολές `ionic build` και `npx cap sync android`. Η πρώτη εντολή δημιουργεί το ενημερωμένο production build της εφαρμογής, παράγοντας τα απαραίτητα στατικά αρχεία στον φάκελο `www`. Η δεύτερη εντολή αναλαμβάνει τον συγχρονισμό αυτών των αρχείων με το αντίστοιχο Android project του Capacitor, ενημερώνοντας παράλληλα τυχόν αλλαγές σε plugins και εξαρτήσεις.

Στη συνέχεια, το project ανοίχτηκε στο Android Studio [27] μέσω της εντολής `npx cap open android`, όπου έγινε η τελική μεταγλώττιση και εκτέλεση της εφαρμογής. Το Android Studio χρησιμοποιήθηκε για την εκτέλεση της εφαρμογής είτε σε εικονική συσκευή (Android Emulator) είτε σε φυσική συσκευή μέσω USB. Η διαδικασία αυτή επέτρεψε τον έλεγχο της συμπεριφοράς της εφαρμογής σε πραγματικό Android περιβάλλον, εντοπίζοντας τυχόν προβλήματα που δεν εμφανίζονται κατά την εκτέλεση στον browser μέσω `ionic serve`.

```

C:\WINDOWS\system32\cmd. x + v
2836.132fb0bf60a168a.js | index-be190feb-js | 1.66 kB | 763 bytes
2924.7c4a83fb14b792cb.js | web | 1.57 kB | 578 bytes
7240.0d54ae3b07df46b2.js | - | 1.43 kB | 586 bytes
3780.bd8da0556bbf53cc.js | web | 1.22 kB | 520 bytes
1577.b898b4ac5cc38f2c.js | - | 1.15 kB | 633 bytes
2920.14e5749cdf9e99c8.js | web | 1.13 kB | 445 bytes
7179.535754b9a1ce7fff.js | md-transition-js | 1.07 kB | 488 bytes
9303.81cc21c0f087ae30.js | web | 875 bytes | 354 bytes
9072.1c25d3b340462d71.js | - | 577 bytes | 343 bytes
7825.b5ae021d76bb8406.js | status-tap-42a8af65-js | 537 bytes | 344 bytes
4786.9281ab15ab0f2929.js | web | 526 bytes | 285 bytes
1709.75d3ab2a8cce1705.js | web | 511 bytes | 215 bytes

Build at: 2026-03-11T15:44:36.361Z - Hash: c5b89dab2b8868d9 - Time: 14459ms

Warning: D:\UniMap\UniMap\src\app\plash\plash.page.scss exceeded maximum budget. Budget 6.00 kB was not met by 706 bytes with a total of 6.71 kB.

Warning: D:\UniMap\UniMap\src\app\home\home.page.ts depends on 'leaflet'. CommonJS or AMD dependencies can cause optimization bailouts.
For more info see: https://angular.dev/tools/cli/build#configuring-commonjs-dependencies

D:\UniMap\UniMap>npx cap sync android
/ Copying web assets from www to android\app\src\main\assets\public in 687.39ms
/ Creating capacitor.config.json in android\app\src\main\assets in 1.77ms
/ copy android in 732.75ms
/ Updating Android plugins in 13.18ms
[info] Found 9 Capacitor plugins for android:
  @capacitor/app@7.0.1
  @capacitor/browser@7.0.3
  @capacitor/geolocation@7.1.7
  @capacitor/haptics@7.0.1
  @capacitor/keyboard@7.0.1
  @capacitor/network@7.0.3
  @capacitor/preferences@7.0.3
  @capacitor/status-bar@7.0.1
  capacitor-native-settings@7.0.2
/ update android in 208.49ms
[info] Sync finished in 1s

D:\UniMap\UniMap>

```

Σχήμα 3.14: Δημιουργία build της εφαρμογής και συγχρονισμός με το Android Project

Κεφάλαιο 4ο: Ανάπτυξη Εφαρμογής και Περιγραφή Λειτουργίας

Σύμφωνα με τις αποφάσεις σχεδιασμού και ανάπτυξης που αναλύθηκαν στο προηγούμενο κεφάλαιο, αυτό το κεφάλαιο παρουσιάζει την υλοποίηση στην εφαρμογή UniMap στο σύνολό της. Το κεφάλαιο θα ξεκινήσει με το σύστημα back-end, εξηγώντας τον τρόπο οργάνωσης των δεδομένων στη μνήμη και τις οντότητες που συνθέτουν το σύστημα (4.1). Στη συνέχεια, θα αναλυθεί η ανάπτυξη του front-end, με βάση το Ionic Framework και το TypeScript, με ιδιαίτερη έμφαση στην αρχιτεκτονική που χρησιμοποιείται και στον τρόπο με τον οποίο αντανακλάται στο σύνολο των υπηρεσιών, των στοιχείων και των μοντέλων που έχουν αναπτυχθεί (4.2). Τέλος, θα παρουσιαστούν οι οθόνες και οι περιπτώσεις της εφαρμογής, με έμφαση στη λειτουργικότητα της εφαρμογής (4.3).

4.1 Ανάπτυξη Back-End

Το back-end της εφαρμογής έχει δημιουργηθεί με χρήση Node.js και Express.js, με TypeScript ως γλώσσα προγραμματισμού. Η βασική σχεδιαστική απόφαση που κάνει αυτή την εφαρμογή να ξεχωρίζει από άλλες εφαρμογές είναι η αποθήκευση όλων των δεδομένων στη μνήμη, δηλαδή η αποθήκευση στη μνήμη, και η απουσία εξωτερικής βάσης δεδομένων για αποθήκευση. Αυτή η απόφαση είναι κατάλληλη για την εφαρμογή αυτή, καθώς τα δεδομένα για την πλοήγηση στον πανεπιστημιακό χώρο είναι στατικά, μικρού μεγέθους και δεν απαιτούν ταυτόχρονες ενημερώσεις. Τα δεδομένα φορτώνονται στη μνήμη κατά την εκκίνηση του διακομιστή και είναι διαθέσιμα καθ' όλη τη διάρκεια της λειτουργίας του διακομιστή, αποφεύγοντας έτσι τις συνεχείς κλήσεις I/O στη βάση δεδομένων.

4.1.1 Οργάνωση Δεδομένων στη Μνήμη

Τα δεδομένα του back-end οργανώνονται σε τέσσερα modules TypeScript, τα οποία εισάγονται και συγχωνεύονται κατά την εκκίνηση. Αυτοί οι τέσσερις λογικοί τύποι δεδομένων είναι: `osm-nodes`, που περιέχει τους κόμβους και τις ακμές του γραφήματος OSM τα οποία εξάγονται από το OSM, `manual-nodes`, που περιέχει χειροκίνητα καθορισμένους ενδιάμεσους κόμβους για τη συμπλήρωση του συνόλου δεδομένων OSM οι οποίοι δημιουργήθηκαν μεταγενέστερα διότι οι κόμβοι και οι ακμές δεν επαρκούσαν, `poi-nodes` που περιέχει τον κόμβο εισόδου για κάθε σημείο ενδιαφέροντος για το κάθε ένα τμήμα που υπάρχει εντός της πανεπιστημιούπολης και `destinations` που περιέχει τα πλήρη λεπτομερή στοιχεία του κάθε τμήματος, συμπεριλαμβανομένων εικόνων, αριθμών τηλεφώνου και ορίων κτιρίων.

Κατά την εκκίνηση του διακομιστή, η συνάρτηση `buildMergedGraph()` στο `campus-graph.ts` όπου φαίνεται στο Σχήμα 4.1, εισάγει και συγχωνεύει όλους τους παραπάνω τύπους δεδομένων σε ένα ενιαίο γράφημα πλοήγησης. Η διαδικασία συγχώνευσης αποτελείται από διάφορα βήματα, καθένα από τα οποία υλοποιείται ως ξεχωριστή συνάρτηση χρησιμότητας. Το πρώτο βήμα είναι η οριοθέτηση του campus: υπολογίζεται ένα `bounding box` βάσει των POI και των χειροκίνητων κόμβων, και κρατούνται μόνο οι OSM κόμβοι που βρίσκονται εντός αυτού. Οι υπόλοιποι, που αντιστοιχούν σε περιοχές εκτός πανεπιστημιούπολης, απορρίπτονται.

```

function computeBBox(points: LatLng[]) {
  let minLat = Infinity, maxLat = -Infinity,
      minLng = Infinity, maxLng = -Infinity;
  for (const p of points) {
    minLat = Math.min(minLat, p.lat);
    maxLat = Math.max(maxLat, p.lat);
    minLng = Math.min(minLng, p.lng);
    maxLng = Math.max(maxLng, p.lng);
  }
  return { minLat, maxLat, minLng, maxLng };
}

function inBBox(p: LatLng, bb): boolean {
  return p.lat >= bb.minLat && p.lat <= bb.maxLat
    && p.lng >= bb.minLng && p.lng <= bb.maxLng;
}

```

Σχήμα 4.1: Συναρτήσεις για τον υπολογισμό του γεωγραφικού πλαισίου οριοθέτησης (Bounding Box) και τον έλεγχο ένταξης ενός σημείου σε αυτό.

Μετά το φιλτράρισμα, ο γράφος κατασκευάζεται με βάση τις έτοιμες ακμές OSM και τις χειροκίνητες ακμές. Οι ακμές προστίθενται παράλληλα χρησιμοποιώντας τη μέθοδο `addUndirectedEdge()`, όπου το βάρος είναι η απόσταση Haversine σε μέτρα μεταξύ των συντεταγμένων των δύο ακραίων σημείων, στρογγυλοποιημένη στον πλησιέστερο ακέραιο αριθμό και με ελάχιστο βάρος 1. Στο Σχήμα 4.2 βλέπουμε την συγκεκριμένη υλοποίηση.

```

function addUndirectedEdge(
  g: Adjacency,
  coords: Record<string, LatLng>,
  u: string, v: string
): void {
  const a = coords[u];
  const b = coords[v];
  if (!a || !b) return;

  const w = Math.max(1, Math.round(distanceTo(a, b)));
  if (!g[u]) g[u] = {};
  if (!g[v]) g[v] = {};
  // Keep the minimum weight if the edge already exists
  g[u][v] = Math.min(g[u][v] ?? Infinity, w);
  g[v][u] = Math.min(g[v][u] ?? Infinity, w);
}

```

Σχήμα 4.2: Συνάρτηση για την προσθήκη νέας, μη κατευθυνόμενης ακμής στον γράφο.

Ένα σημαντικό μέρος της διαδικασίας δημιουργίας γραφημάτων είναι η μέθοδος `splitEdgeWithChain()`, η οποία χρησιμοποιείται για να χωρίσει υπάρχουσες ακμές OSM προσθέτοντας μια σειρά απο ενδιάμεσους χειροκίνητους κόμβους. Αυτό χρησιμοποιείται σε περιπτώσεις όπου τα δεδομένα OSM έχουν μία μόνο ακμή μεταξύ δύο κόμβων, αλλά στην πραγματικότητα η διαδρομή περνάει από μια σειρά ενδιάμεσων σημείων, τα οποία δεν απεικονίζονται στα δεδομένα OSM. Αυτό επιτυγχάνεται με

την αφαίρεση της ακμής και την αντικατάστασή της με μια σειρά ακμών μέσω της αλυσίδας στο Σχήμα 4.3.

```
function splitEdgeWithChain(
  g: Adjacency,
  coords: Record<string, LatLng>,
  a: string, b: string,
  chain: string[]
): void {
  removeUndirectedEdge(g, a, b); // remove original long edge
  let prev = a;
  for (const mid of chain) {
    addUndirectedEdge(g, coords, prev, mid);
    prev = mid;
  }
  addUndirectedEdge(g, coords, prev, b);
}

// Example usage – subdivide a path through intermediate waypoints:
splitEdgeWithChain(g, ALL, 'N0068', 'N0060',
  ['M_68_TO_BOTTOM_1', 'M_BOTTOM_MID', 'M_BOTTOM_TO_60_1']);
```

Σχήμα 4.3: Διαδικασία υποδιαίρεσης μιας μεγάλης ακμής του γράφου μέσω ενδιάμεσων κόμβων (waypoints).

Στο Σχήμα 4.4 απεικονίζεται η αντιμετώπιση καταστάσεων στις οποίες δύο κόμβοι βρίσκονται φυσικά κοντά αλλά δεν συνδέονται απαραίτητα με μια ακμή, όπως ένας κόμβος OSM και ένας κόμβος που έχει προστεθεί χειροκίνητα σε μια διασταύρωση, η συνάρτηση `healCloseNodes()` συνδέει όλα τα ζεύγη κόμβων που βρίσκονται σε απόσταση μικρότερη από 1,5 μέτρα:

```
function healCloseNodes(
  ids: string[], coords: Record<string, LatLng>,
  g: Adjacency, maxDistM: number
): void {
  for (let i = 0; i < ids.length; i++) {
    for (let j = i + 1; j < ids.length; j++) {
      const a = coords[ids[i]];
      const b = coords[ids[j]];
      if (!a || !b) continue;
      if (distanceTo(a, b) <= maxDistM)
        addUndirectedEdge(g, coords, ids[i], ids[j]);
    }
  }
}
```

Σχήμα 4.4: Αυτόματη σύνδεση (επούλωση) κόμβων που βρίσκονται σε πολύ κοντινή απόσταση μεταξύ τους.

Στη συνέχεια στο Σχήμα 4.5, προσδιορίζεται η μέγιστη συνεκτική συνιστώσα μέσω μιας αναδρομικής (ή ακόμη και επαναληπτικής) αναζήτησης κατά βάθος (DFS). Αυτό διασφαλίζει ότι μόνο οι αμοιβαία προσβάσιμοι κόμβοι χρησιμοποιούνται ως σημεία αναφοράς για την πλοήγηση, εμποδίζοντας τον αλγόριθμο δρομολόγησης από το να προσπαθήσει να προσεγγίσει απομονωμένους κόμβους:

```
function getLargestComponent(
  ids: string[], g: Adjacency
): Set<string> {
  const seen = new Set<string>();
  let best = new Set<string>();

  for (const start of ids) {
    if (seen.has(start)) continue;
    const comp = new Set<string>();
    const stack = [start];
    seen.add(start);
    while (stack.length) {
      const u = stack.pop()!;
      comp.add(u);
      for (const v of Object.keys(g[u] ?? {})) {
        if (!seen.has(v)) { seen.add(v); stack.push(v); }
      }
    }
    if (comp.size > best.size) best = comp;
  }
  return best;
}
```

Σχήμα 4.5: Αλγόριθμος για την εύρεση του μεγαλύτερου συνεχόμενου (συνδεδεμένου) τμήματος του γράφου.

Τέλος, στο Σχήμα 4.6 αναπαρίσταται, ο κόμβος εισόδου κάθε POI, ο οποίος συνδέεται με τον πλησιέστερο κόμβο στο μεγαλύτερο στοιχείο, προσθέτοντας μία μόνο άκρη μεταξύ των δύο. Εάν το POI απέχει περισσότερο από 60 μέτρα από τον πλησιέστερο κόμβο κατά τον μηχανισμό επαλήθευσης της ορθότητας των γεωγραφικών δεδομένων εμφανίζεται ένα προειδοποιητικό μήνυμα. Η όλη διαδικασία δημιουργίας εκτελείται μία φορά κατά την εκκίνηση και προκύπτει ο γράφος που αποθηκεύεται ως singleton:

```
// Singleton – built once at server startup
registerAccessibilityEdges();
const MERGED = buildMergedGraph();
```

Σχήμα 4.6: Αρχικοποίηση και αποθήκευση του ενοποιημένου γράφου δρομολόγησης κατά την εκκίνηση του διακομιστή.

Το module geo.ts (Σχήμα 4.7) διαθέτει μια συνάρτηση distanceTo(), η οποία χρησιμοποιεί τον τύπο Haversine για να υπολογίσει την απόσταση κατά μήκος ενός μεγάλου κύκλου μεταξύ δύο γεωγραφικών θέσεων. Ο τύπος αυτός λαμβάνει υπόψη την καμπυλότητα της Γης και δίνει την απόσταση σε μέτρα:

```

const EARTH_RADIUS_M = 6371000;

export function distanceTo(a: LatLng, b: LatLng): number {
  const dLat = ((b.lat - a.lat) * Math.PI) / 180;
  const dLng = ((b.lng - a.lng) * Math.PI) / 180;
  const lat1 = (a.lat * Math.PI) / 180;
  const lat2 = (b.lat * Math.PI) / 180;

  const x =
    Math.sin(dLat/2) * Math.sin(dLat/2) +
    Math.sin(dLng/2) * Math.sin(dLng/2) *
    Math.cos(lat1) * Math.cos(lat2);

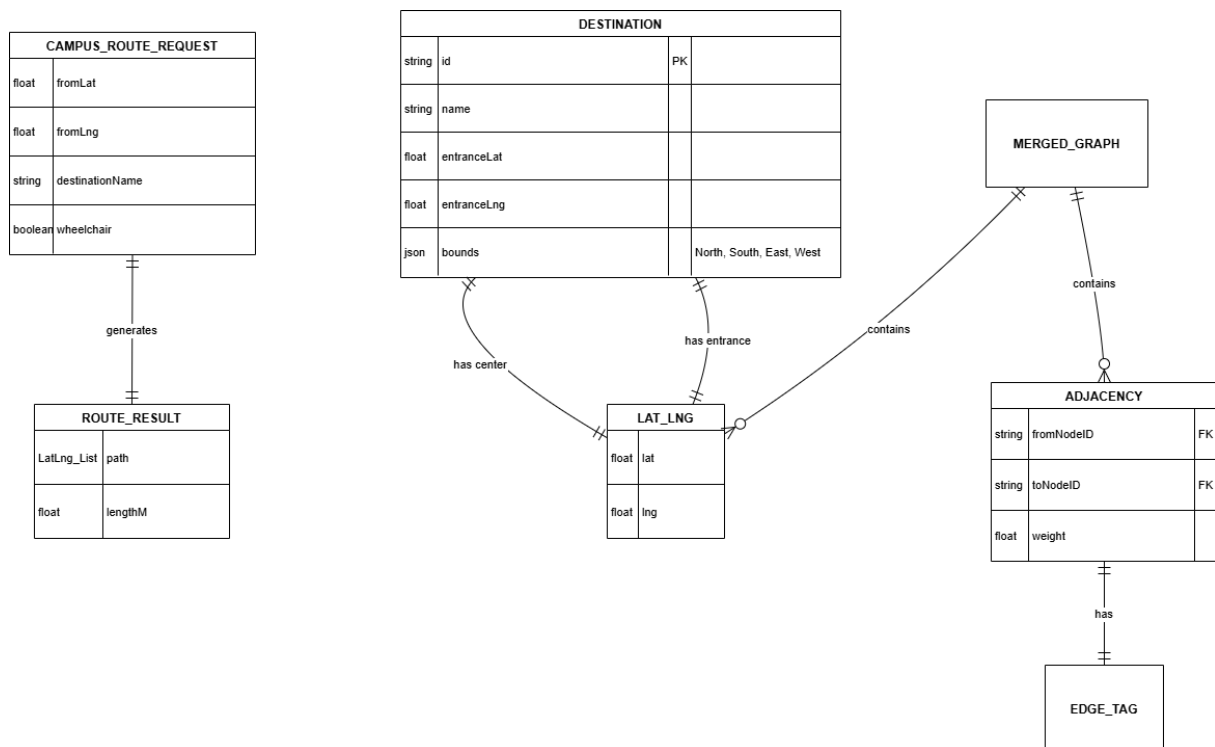
  return EARTH_RADIUS_M * 2 * Math.atan2(
    Math.sqrt(x), Math.sqrt(1 - x)
  );
}

```

Σχήμα 4.7: Υπολογισμός της πραγματικής απόστασης (σε μέτρα) μεταξύ δύο γεωγραφικών σημείων, λαμβάνοντας υπόψη την καμπυλότητα της Γης.

4.1.2 Προσδιορισμός Οντοτήτων

Λαμβάνοντας υπόψη τη ρύθμιση της μνήμης και τις ανάγκες της εφαρμογής, έχουν σχεδιαστεί οι οντότητες δεδομένων του back-end. Το Σχήμα 4.1 δείχνει τις οντότητες του back-end.



Σχήμα 4.8: Διάγραμμα Οντοτήτων back-end

4.1.2.1 LatLng

Η οντότητα LatLng που φαίνεται στο Σχήμα 4.9, αντιπροσωπεύει το κύριο σύστημα γεωγραφικών συντεταγμένων που χρησιμοποιείται σε όλη την εφαρμογή. Αυτή η οντότητα χρησιμοποιείται σε ολόκληρη την εφαρμογή, από τις θέσεις που αντιπροσωπεύονται από τους κόμβους έως τη θέση του χρήστη:

```
export interface LatLng {
  lat: number;
  lng: number;
}
```

Σχήμα 4.9: Ορισμός της βασικής δομής για τις γεωγραφικές συντεταγμένες.

4.1.2.2 Adjacency

Η οντότητα Adjacency αντιπροσωπεύει τον πίνακα του γραφήματος πλοήγησης (Σχήμα 4.10). Πρόκειται για μια διδιάστατη δομή, όπου το πρώτο κλειδί είναι το αναγνωριστικό του κόμβου, το δεύτερο κλειδί είναι το αναγνωριστικό του γειτονικού κόμβου και η τιμή είναι το βάρος της ακμής ή η απόσταση σε μέτρα. Κάθε άκρη αντιγράφεται στον πίνακα Adjacency. Με άλλα λόγια, εάν ο κόμβος A συνδέεται με τον κόμβο B με συγκεκριμένο βάρος ή απόσταση, τότε ο κόμβος B συνδέεται με τον κόμβο A με το ίδιο βάρος ή απόσταση.

```
export type Adjacency = Record<string, Record<string, number>>;
```

Σχήμα 4.10: Ορισμός της μορφής του δικτύου διαδρομών (γράφου).

4.1.2.3 EdgeTag

Ο τύπος EdgeTag του τύπου ένωσης EdgeTag αντιπροσωπεύει τον τρόπο με τον οποίο τα άκρα ταξινομούνται σε σχέση με την προσβασιμότητα. Στον Σχήμα 4.11, η συνάρτηση registerAccessibilityEdges() στο αρχείο accessibility.ts επισημαίνει τα άκρα κατά τη διάρκεια της κατασκευής, με τα τμήματα σκαλοπατιών να επισημαίνονται έτσι όπως πρέπει. Στη συνάρτηση δρομολόγησης για τη δρομολόγηση αναπηρικών αμαξιδίων, τα άκρα με την ετικέτα STAIRS αγνοούνται:

```
export type EdgeTag = 'ALL' | 'STAIRS' | 'RAMP';

// accessibility.ts
export function registerAccessibilityEdges(): void {
  setEdgeTag('M_DOWN_1', 'M_CENTRAL_TO_DOWN_1', 'STAIRS');
  setEdgeTag('M_DOWN_1', 'M_BOTTOM_MID', 'STAIRS');
}

export function edgeAllowedForWheelchair(tag: EdgeTag): boolean {
  return tag !== 'STAIRS';
}
```

Σχήμα 4.11: Χαρακτηρισμός συγκεκριμένων διαδρομών (π.χ. σκάλες) για τον έλεγχο προσβασιμότητας.

Ο γράφος που φιλτράρεται από το αναπηρικό αμαξίδιο υπολογίζεται με καθυστέρηση την πρώτη φορά που ζητείται το αναπηρικό αμαξίδιο (για λόγους εξοικονόμησης μνήμης), μετά την οποία αποθηκεύεται ως μεταβλητή σε επίπεδο ενότητας (`wheelchairAdj`). Αυτή η προσέγγιση αποφεύγει το κόστος του φιλτραρίσματος κάθε φορά που ζητείται το αναπηρικό αμαξίδιο, αλλά εξακολουθεί να επιτρέπει τον υπολογισμό κατά ζήτηση όπως φαίνεται στο Σχήμα 4.12.

```
let wheelchairAdj: Adjacency | undefined;

function getAdjacency(wheelchair: boolean): Adjacency {
  if (!wheelchair) return MERGED.adjacency;
  if (wheelchairAdj) return wheelchairAdj;

  const filtered: Adjacency = {};
  for (const [u, nbrs] of Object.entries(MERGED.adjacency)) {
    for (const [v, w] of Object.entries(nbrs)) {
      if (!edgeAllowedForWheelchair(getEdgeTag(u, v))) continue;
      (filtered[u] ??= {})[v] = w;
    }
  }
  wheelchairAdj = filtered;
  return filtered;
}
```

Σχήμα 4.12: Δημιουργία ενός προσαρμοσμένου χάρτη δρομολόγησης ειδικά για χρήστες αναπηρικών αμαξιδίων.

4.1.2.4 RouteResult

Η οντότητα `RouteResult` που απεικονίζεται στο Σχήμα 4.13, χρησιμοποιείται ως απάντηση backend σε ένα αίτημα δρομολόγησης, το οποίο περιέχει την ακολουθία των σημείων `LatLng` και το συνολικό μήκος της διαδρομής σε μέτρα:

```
export interface RouteResult {
  path: LatLng[];
  lengthM: number;
}
```

Σχήμα 4.13: Δομή των δεδομένων που επιστρέφονται μετά τον υπολογισμό της διαδρομής.

4.1.2.5 CampusRouteRequest

Η κλάση `CampusRouteRequest` που απεικονίζεται στο Σχήμα 4.14, ορίζει τη δομή του σώματος JSON που το frontend στέλνει στο τελικό σημείο `/api/route/campus`. Παρέχει δύο τρόπους για τον προσδιορισμό του προορισμού: μέσω του ονόματος του προορισμού (`destinationName`) ή μέσω των συντεταγμένων fallback (`destLat`, `destLng`). Το τελικό σημείο εκτελεί πρώτα την αναζήτηση ονόματος:

```

export interface CampusRouteRequest {
  fromLat: number;
  fromLng: number;
  destinationName?: string;
  destLat?: number;
  destLng?: number;
  wheelchair?: boolean;
}

```

Σχήμα 4.14: Η δομή των δεδομένων που απαιτούνται για την εκκίνηση ενός αιτήματος δρομολόγησης από την εφαρμογή

Ο χειριστής διαδρομών `campus.ts` που φαίνεται στο Σχήμα 4.15, χρησιμοποιεί έναν μηχανισμό επίλυσης τριών βημάτων. Το πρώτο βήμα είναι ο υπολογισμός του αναγνωριστικού του κόμβου προορισμού, μαζί με τον υπολογισμό του καλύτερου κόμβου εκκίνησης κοντά στην τοποθεσία GPS του χρήστη και τέλος τον υπολογισμό της συντομότερης διαδρομής χρησιμοποιώντας τον αλγόριθμο A*.

```

// Εικονικός κόμβος __USER__ στη θέση GPS
const adj: Adjacency = { ...baseAdj, [VIRTUAL_START]: {} };
for (const c of activeTop) {
  adj[VIRTUAL_START][c.id] = Math.round(c.distM) || 1;
}

const nodePath = aStarPath(adj, allCoords, VIRTUAL_START, endNodeId);
if (!nodePath) return null;

return { path: points, lengthM: Math.round(len) };
}

```

Σχήμα 4.15: ιαδικασία εισαγωγής της θέσης του χρήστη στον χάρτη και εκτέλεση του αλγορίθμου εύρεσης της συντομότερης διαδρομής.

Η συνάρτηση `calculateRouteFromPosition()`, που φαίνεται στο Σχήμα 4.16, δεν επιλέγει απλώς τον πλησιέστερο κόμβο, αλλά αξιολογεί όλους τους υποψήφιους κόμβους εντός ακτίνας 80 μέτρων (`MAX_START_RADIUS`). Για κάθε υποψήφιο κόμβο υπολογίζεται ένα σύνθετο κόστος που αποτελείται από την απόστασή του από τον χρήστη συν ποινή 25 μέτρων για κάθε ακμή του γράφου που τέμνει η ευθεία γραμμή χρήστη→κόμβος (crossing penalty). Με αυτόν τον τρόπο αποφεύγονται κόμβοι που βρίσκονται «πίσω από» κτίρια ή εμπόδια, εξασφαλίζοντας ότι η διαδρομή ξεκινά από ένα πραγματικά προσβάσιμο σημείο.

```
export function calculateRouteFromPosition(
  lat: number, lng: number,
  endNodeId: string,
  opts?: { wheelchair?: boolean }
): RouteResult | null {
  const here: LatLng = { lat, lng };
  const snapCands = getSnapCandidates(!opts?.wheelchair);

  for (const id of snapCands) {
    const nodeLL = MERGED.coords[id];
    const distM = distanceTo(here, nodeLL);
    if (distM > MAX_START_RADIUS) continue; // 80m
    const crossings = countApproachCrossings(here, nodeLL, ...);
    const score = distM + crossings * CROSSING_PENALTY_M; // ποινή 25m
    candidates.push({ id, distM, score });
  }
}
```

Σχήμα 4.16: Διαδικασία εύρεσης του κοντινότερου σημείου αφετηρίας στον χάρτη, βάσει της τρέχουσας τοποθεσίας του χρήστη.

Η επεξεργασία των ονομάτων των τμημάτων/προορισμών υλοποιείται μέσω της συνάρτησης κανονικοποίησης `norm()`. Στο Σχήμα 4.17, φαίνεται η συνάρτηση αυτή αφαιρεί τους τόνους, μετατρέπει το κείμενο σε κεφαλαία, αφαιρεί κοινά προθέματα (όπως "ΤΜΗΜΑ" ή "ΣΧΟΛΗ") και διορθώνει τα κενά διαστήματα. Με αυτόν τον τρόπο, η αναζήτηση γίνεται πιο ευέλικτη και αναγνωρίζει τα ονόματα ακόμα και αν υπάρχουν μικρές ορθογραφικές αποκλίσεις από το front-end.

```
function norm(s: string): string {
  return s
    .toUpperCase()
    .normalize('NFD')
    .replace(/[\u0300-\u036f]/g, '')
    .replace(/&/g, 'KAI')
    .replace(/ΤΜΗΜΑ /g, '')
    .replace(/\\s+/g, ' ')
    .trim();
}
```

Σχήμα 4.17: Βοηθητική συνάρτηση που προετοιμάζει και καθαρίζει το κείμενο για τη σωστή λειτουργία της αναζήτησης.

4.1.2.6 Προορισμός

Η οντότητα «Προορισμός» είναι ο κεντρικός κόμβος δεδομένων για όλους τους πανεπιστημιακούς χώρους και περιέχει όλα τα δεδομένα και πληροφορίες για ένα τμήμα ή σημείο ενδιαφέροντος, συμπεριλαμβανομένων των συντεταγμένων της εισόδου οι οποίες διαφέρουν από το κέντρο του κτιρίου. Αυτό γίνεται για να διασφαλιστεί ότι ο χρήστης θα οδηγηθεί στο σωστό σημείο εισόδου και όχι στο κέντρο βάρους του κτιρίου που διαφέρει (Σχήμα 4.18).

```
export interface Destination {
  id: string;
  name: string;
  lat: number;           // building center
  lng: number;
  entranceLat?: number; // entrance point (used for routing)
  entranceLng?: number;
  image?: string;
  description?: string;
  phone?: string;
  website?: string;
  bounds?: {
    north: number; south: number;
    east: number;  west: number;
  };
}
```

Σχήμα 4.18: Δομή των δεδομένων για τα κτίρια και τους προορισμούς της εφαρμογής.

Το πεδίο των ορίων(bounds) ορίζει το ορθογώνιο πλαίσιο που καθορίζει το περίγραμμα ενός κτιρίου. Αυτή η περιοχή επιτρέπει στο front-end να ανιχνεύει πότε ένας χρήστης πατά οπουδήποτε εντός του αποτυπώματος ενός κτιρίου στον χάρτη, έτσι ώστε να εμφανίζεται ένα αναδυόμενο παράθυρο χωρίς να πατάει σε ένα ακριβές σημείο. Για να προσδιορίσει εάν μια θέση πατήματος βρίσκεται εντός ενός κτιρίου, το front-end ελέγχει τη θέση πατήματος σε σχέση με όλα τα όρια προορισμού και χρησιμοποιεί την πρώτη αντιστοιχία. Αυτό γίνεται και για λόγους σωστής πλοήγησης καθώς η εσφαλμένη ή άκυρη διαδρομή προς κάποιο τυχαίο σημείο του δρόμου ή ενός κενού χώρου μπορεί να οδηγήσει σε «κακή» πλοήγηση.

4.1.3 Αλγόριθμος A* και Δομή Γράφου

Ο αλγόριθμος εύρεσης της συντομότερης διαδρομής που έχει εφαρμοστεί στην εφαρμογή UniMap βασίζεται στον παραδοσιακό αλγόριθμο A*. Ο αλγόριθμος έχει εφαρμοστεί σε έναν σταθμισμένο, μη κατευθυνόμενο γράφο που αντιπροσωπεύει το δίκτυο διαδρομών του πανεπιστημίου. Ο γράφος κατασκευάζεται από τρεις κατηγορίες κόμβων: κόμβους που προέρχονται από τα έτοιμα δεδομένα από το OpenStreetMap, κόμβους που δημιουργήθηκαν χειροκίνητα για την κάλυψη περιοχών με ελλιπή χαρτογράφηση και κόμβους σημείων ενδιαφέροντος που αντιστοιχούν στις εισόδους των κτιρίων.

Κάθε κόμβος στο γράφημα αντιπροσωπεύει ένα γεωγραφικό σημείο με τις σχετικές συντεταγμένες γεωγραφικού πλάτους και μήκους (LatLng) και κάθε άκρη αντιπροσωπεύει ένα τμήμα διαδρομής μεταξύ δύο κόμβων. Οι βαρύτητες κάθε άκρης στο γράφημα έχουν υπολογιστεί με βάση την ευκλείδεια

απόσταση μεταξύ κάθε ζεύγους κόμβων. Η ευκλείδεια απόσταση έχει προσεγγιστεί χρησιμοποιώντας τον τύπο Haversine για να εξασφαλιστεί ο ακριβής υπολογισμός της απόστασης σε μια σφαιρική επιφάνεια. Ο τύπος λαμβάνει υπόψη την καμπυλότητα της Γης, έχοντας έτσι σωστούς υπολογισμούς απόστασης για μια εφαρμογή εξωτερικού και εσωτερικού χώρου.

Όταν δημιουργείται ο γράφος, καλείται η συνάρτηση `addUndirectedEdge`, η οποία προσθέτει μια άκρη και στις δύο κατευθύνσεις ($u \rightarrow v$ και $v \rightarrow u$), διατηρώντας την ελάχιστη τιμή βάρους σε περίπτωση που υπάρχουν πολλαπλές διαδρομές μεταξύ των ίδιων κόμβων.

Η συνάρτηση `splitEdgeWithChain` είναι ιδιαίτερα σημαντική, καθώς επιτρέπει τη δυναμική εισαγωγή ενδιάμεσων κόμβων σε μια υπάρχουσα άκρη. Αυτό απαιτείται για την ακριβή σύνδεση κτιρίων και σημείων ενδιαφέροντος στο ήδη υπάρχον δίκτυο χωρίς να χρειάζεται η ανακατασκευή του γραφήματος.

Ο αλγόριθμος A^* λαμβάνει ως είσοδο ολόκληρο το γράφημα γειτνίασης (adjacency graph), τον κόμβο εκκίνησης και τον κόμβο προορισμού. Για να βρει τον πλησιέστερο κόμβο για την εκκίνηση, η εφαρμογή υπολογίζει την απόσταση από την τρέχουσα θέση GPS του χρήστη προς όλους τους διαθέσιμους κόμβους και επιλέγει τον κόμβο που βρίσκεται πλησιέστερα εντός του οριοθετικού πλαισίου της περιοχής.

Η χρήση του οριοθετικού πλαισίου (`computeBBBox / inBBBox`) είναι ιδιαίτερα σημαντική για τη βελτιστοποίηση της αναζήτησης και την αποφυγή του υπολογισμού περιττών κόμβων εκτός της περιοχής της πανεπιστημιούπολης.

Το αποτέλεσμα του αλγορίθμου είναι ένα αντικείμενο `RouteResult` που περιέχει τη διαδρομή ως λίστα συντεταγμένων `LatLng` και το συνολικό μήκος σε μέτρα. Αυτή η λίστα συντεταγμένων στέλνεται στο `front-end`, όπου χρησιμοποιείται για να σχεδιάσει τη διαδρομή ως πολύγραμμη στον χάρτη `Leaflet` με μπλε χρώμα καθοδηγώντας έτσι τον χρήστη.

4.1.4 Σύστημα Προσβασιμότητας Αμαξιδίου

Η εφαρμογή `UniMap` έχει ένα σύστημα προσβασιμότητας που επιτρέπει στους χρήστες με κινητική αναπηρία να λαμβάνουν διαδρομές χωρίς σκαλιά και με την επιλογή ειδικών ραμπών. Το σύστημα αυτό βασίζεται στην έννοια των `EdgeTag`, δηλαδή ετικετών που χαρακτηρίζουν κάθε ακμή του γράφου ανάλογα με το είδος της διαδρομής που αναπαριστά. Ορίζονται δύο τύποι ετικετών: `ALL` για γενικά προσβάσιμες διαδρομές και `STAIRS` για σκαλιά που είναι αδύνατα για αμαξίδια.

Κάθε ακμή του γράφου αποθηκεύεται στον κεντρικό χάρτη `EDGE_TAGS` με κλειδί που προκύπτει από τα ονόματα των δύο κόμβων σε αλφαβητική σειρά, εξασφαλίζοντας έτσι μοναδικότητα ανεξαρτήτως κατεύθυνσης. Κατά την κατασκευή της εφαρμογής, οι ακμές που αντιστοιχούν σε σκαλιά εντοπίστηκαν έπειτα από διεύρυνσή σε δεύτερο χρόνο στο `campus` και καταγράφηκαν χειροκίνητα μέσω της συνάρτησης `registerAccessibilityEdges`.

Όταν ο χρήστης ενεργοποιεί την επιλογή αναπηρικού αμαξιδίου μέσω του κουμπιού που υπάρχει στο πάνω μέρος της εφαρμογής, το αίτημα δρομολόγησης αποστέλλεται στο `back-end` με την παράμετρο `wheelchair: true`. Κατά την κατασκευή του γράφου για αυτό το αίτημα, όλες οι ακμές με ετικέτα `STAIRS` αφαιρούνται αυτόματα μέσω της συνάρτησης `edgeAllowedForWheelchair`, η οποία επιστρέφει `false` μόνο για ακμές τύπου `STAIRS`. Ο αλγόριθμος A^* εκτελείται στον φιλτραρισμένο γράφο, επιστρέφοντας τη βέλτιστη διαδρομή που είναι προσβάσιμη με αμαξίδιο, ακόμα κι αν αυτή είναι μεγαλύτερη σε απόσταση από τη συντομότερη διαδρομή χωρίς φιλτράρισμα.

4.1.5 API Endpoints και Λογική Δρομολόγησης

Στο back-end αναπτύσσεται ένα RESTful API οργανωμένο σε τρεις ομάδες endpoints. Η πρώτη ομάδα αφορά την πλοήγηση εντός campus (campus routes) και διαχειρίζεται αιτήματα δρομολόγησης που χρησιμοποιούν τον εσωτερικό αλγόριθμο A*. Η δεύτερη ομάδα αφορά την πλοήγηση εξωτερικού χώρου (outdoor routes) και βασίζεται στο εξωτερικό OSRM routing API. Η τρίτη ομάδα παρέχει πληροφορίες για τους διαθέσιμους προορισμούς (destinations), επιστρέφοντας τη λίστα κτιρίων και τμημάτων με τα δεδομένα τους.

Το κεντρικό endpoint δρομολόγησης campus δέχεται αίτημα τύπου CampusRouteRequest, το οποίο περιλαμβάνει τις συντεταγμένες αφετηρίας (fromLat, fromLng), το όνομα του προορισμού (destinationName) ή εναλλακτικά τις συντεταγμένες προορισμού (destLat, destLng), και τη boolean παράμετρο wheelchair για φιλτράρισμα ακμών.

Το back-end ακολουθεί τη σειρά:

- Εντοπισμός κόμβου αφετηρίας — αναζήτηση πλησιέστερου κόμβου στη θέση GPS,
- Ανάλυση ονόματος προορισμού μέσω ομαλοποίησης (normalization) και αντιστοίχισης με alias
- Εκτέλεση αλγορίθμου A*
- Επιστροφή διαδρομής ως λίστα LatLng με συνολικό μήκος.

Στο front-end, η επικοινωνία με το back-end γίνεται μέσω του ApiService, ο οποίος χρησιμοποιεί το Angular HttpClient. Για τη δρομολόγηση εξωτερικού χώρου, το RoutingService καλεί το endpoint GET /api/route/outdoor του back-end, το οποίο λειτουργεί ως proxy και προωθεί το αίτημα στο δημόσιο OSRM API [28] με παράμετρο profile "foot" (πεζοπορία), λαμβάνοντας τη διαδρομή σε μορφή GeoJSON. Σε περίπτωση αποτυχίας του OSRM (απουσία σύνδεσης ή μη διαθεσιμότητα), ο RouteService εφαρμόζει αυτόματα fallback λογική: σχεδιάζει μια διακεκομμένη ευθεία γραμμή μεταξύ αφετηρίας και προορισμού, ενημερώνοντας τον χρήστη ότι η διαδρομή είναι κατά προσέγγιση.

4.2 Ανάπτυξη Front-End

Το front-end αναπτύχθηκε χρησιμοποιώντας το πλαίσιο Ionic σε συνδυασμό με Angular και TypeScript. Η χρήση του πλαισίου Ionic κατέστησε δυνατή την ανάπτυξη της εφαρμογής σε όλες τις πλατφόρμες, από Android και iOS έως το διαδίκτυο, χρησιμοποιώντας μία βάση κώδικα, με το Capacitor να αποτελεί τον δίαυλο επικοινωνίας. Η εφαρμογή ακολουθεί μια modular αρχιτεκτονική. Συγκεκριμένα, το HomeModule υλοποιείται με lazy loading για βελτιστοποίηση της απόδοσης, ενώ το SplashModule διαχειρίζεται την αρχική οθόνη. Η επιχειρησιακή λογική (business logic) είναι πλήρως απομονωμένη σε injectable services, εξασφαλίζοντας ότι μπορεί να χρησιμοποιηθεί ξανά η δομή τους.

4.2.1 Αρχιτεκτονική – Οντότητες Front-End

Η δομή του κώδικα βασίζεται στις αρχιτεκτονικές ιδέες της Angular. Υπάρχει σαφής διαχωρισμός μεταξύ υπηρεσιών, Components, μοντέλων και βοηθητικές συναρτήσεις. Επιπλέον, κάθε υπηρεσία χρησιμοποιεί @Injectable({ providedIn: "root" }), καθιστώντας κάθε υπηρεσία ένα singleton στο οποίο μπορεί να έχει πρόσβαση ολόκληρη η εφαρμογή. Αυτό επιτρέπει στις υπηρεσίες να επικοινωνούν μεταξύ τους απευθείας, χωρίς να απαιτείται event bus ή κεντρικό State Management για διαμοιρασμό του state. Για καλύτερη κατανόηση, παρέχονται παραδείγματα για κάθε οντότητα.

4.2.1.1 Αρχιτεκτονική Εφαρμογής

Οι πέντε βασικές υπηρεσίες σχηματίζουν μια αλυσίδα εξαρτήσεων (dependency chain). Η GpsService βασίζεται στην RouteService για τη λήψη snapshot υπολογισμών και της κατάστασης άφιξης. Αντίστοιχα, η MapService χρησιμοποιεί την GpsService για τα events εντοπισμού θέσης (locationFound) και την RouteService για τη λήψη των waypoints της διαδρομής. Τη διαχείριση αυτή αναλαμβάνει ο Dependency Injector της Angular, ο οποίος συνδέει αυτόματα τις εξαρτήσεις κατά το runtime. Τέλος, η ApiService και η SettingsService λειτουργούν αυτόνομα, χωρίς εξαρτήσεις από τις υπόλοιπες υπηρεσίες.

Η επικοινωνία μεταξύ των υπηρεσιών γίνεται μέσω του EventEmitter του Angular, το οποίο είναι ένα σύστημα ειδοποιήσεων μονής κατεύθυνσης. Για παράδειγμα, το στοιχείο της αρχικής σελίδας εγγράφεται στο συμβάν locationFound του GpsService και, όταν αυτό ενεργοποιείται, δίνει εντολή στο MapService να ανανεώσει τη θέση του χρήστη στον χάρτη. Σε ένα άλλο παράδειγμα, η αρχική σελίδα εγγράφεται στο συμβάν routeProgress του RouteService για να ανανεώσει τη γραμμή προόδου του UI. Αυτή η χαλαρή σύζευξη σημαίνει ότι οι υπηρεσίες μας δεν έχουν άμεσες αναφορές στα στοιχεία του UI μας.

4.2.1.2 ApiService

Το ApiService είναι υπεύθυνο για όλες τις επικοινωνίες HTTP με το back end. Ελέγχει την πλατφόρμα χρησιμοποιώντας το Capacitor.isNativePlatform() αναγνωρίζει την πλατφόρμα και επιλέγει τον κατάλληλο HTTP client. Σε περιβάλλον Native Android και Ios, χρησιμοποιεί το CapacitorHttp, το οποίο εκτελεί το αίτημα HTTP απευθείας εκτός του WebView και παρακάμπτει τα προβλήματα CORS. Σε περιβάλλον Web (Browser), χρησιμοποιεί το HttpClient της Angular και μετατρέπει το Observable σε Promise χρησιμοποιώντας τη συνάρτηση firstValueFrom() όπως απεικονίζεται στο Σχήμα 4.19.

```
@Injectable({ providedIn: 'root' })
export class ApiService {
  private readonly baseUrl = Capacitor.isNativePlatform()
    ? environment.apiUrl
    : 'http://localhost:3000';

  constructor(private http: HttpClient) {}

  getCampusRoute(
    params: CampusRouteParams
  ): Promise<CampusRouteResponse> {
    if (Capacitor.isNativePlatform()) {
      return CapacitorHttp.post({
        url: `${this.baseUrl}/api/route/campus`,
        headers: { 'Content-Type': 'application/json' },
        data: params,
      }).then(r => r.data as CampusRouteResponse);
    }
    return firstValueFrom(
      this.http.post<CampusRouteResponse>(
        `${this.baseUrl}/api/route/campus`, params
      )
    );
  }
}
```

Σχήμα 4.19: Η υπηρεσία επικοινωνίας (API Service) που στέλνει το αίτημα δρομολόγησης στον server.

Οι διεπαφές (interfaces) `CampusRouteParams` και `CampusRouteResponse` αντιστοιχούν στους τύπους του backend `CampusRouteRequest` και `RouteResult`, οι οποίοι παρέχουν ασφάλεια τύπου σε όλο το εύρος του HTTP (Σχήμα 4.20).

```
export interface CampusRouteParams {
  fromLat: number; fromLng: number;
  destinationName?: string;
  destLat?: number; destLng?: number;
  wheelchair?: boolean;
}

export interface CampusRouteResponse {
  path: LatLngPoint[];
  lengthM: number;
}
```

Σχήμα 4.20: Οι δομές δεδομένων για την αποστολή των παραμέτρων και τη λήψη του αποτελέσματος δρομολόγησης.

4.2.1.3 SettingsService

Το `SettingsService` είναι υπεύθυνο για τη διαχείριση των μόνιμων ρυθμίσεων της εφαρμογής χρησιμοποιώντας το `Capacitor Preferences API`. Το API αυτό αποτελεί ένα `abstraction layer` πάνω από τα `SharedPreferences` (Android), `NSUserDefaults` (iOS) και το `localStorage` (Web). Όλες οι ρυθμίσεις αποθηκεύονται ως μία συμβολοσειρά JSON με το κλειδί «`unimap_settings_v2`». Όλες οι ρυθμίσεις αποθηκεύονται ως μία JSON συμβολοσειρά (string) με το κλειδί `unimap_settings_v2`. Κατά τη φόρτωση, οι αποθηκευμένες τιμές συγχωνεύονται (merge) με τις προκαθορισμένες τιμές (DEFAULTS). Αυτό διασφαλίζει τη συμβατότητα με μελλοντικές εκδόσεις, σε περίπτωση που προστεθούν νέα πεδία ρυθμίσεων όπως απεικονίζεται στο Σχήμα 4.21.

```

export type AppLanguage = 'el' | 'en';
export type UnitsMode = 'm' | 'km';
export type BaseLayerMode = 'osm' | 'maptiler';

export interface AppSettings {
  language: AppLanguage;
  units: UnitsMode;
  northLock: boolean;
  baseLayer: BaseLayerMode;
}

const KEY = 'unimap_settings_v2';
const DEFAULTS: AppSettings = {
  language: 'el', units: 'm',
  northLock: false, baseLayer: 'maptiler',
};

async load(): Promise<AppSettings> {
  const { value } = await Preferences.get({ key: KEY });
  if (!value) return { ...DEFAULTS };
  try {
    const parsed = JSON.parse(value);
    // Spread DEFAULTS first – ensures new fields are present
    return { ...DEFAULTS, ...parsed } as AppSettings;
  } catch {
    return { ...DEFAULTS };
  }
}

```

Σχήμα 4.21: Διαχείριση και τοπική αποθήκευση των προτιμήσεων του χρήστη

4.2.1.4 GpsService

Στο Σχήμα 4.22, φαίνεται η συνάρτηση GpsService αποτελεί το πλέον πιο αναπόσπαστο και σύνθετο κομμάτι της εφαρμογής, καθώς είναι υπεύθυνη για τον προσδιορισμό της θέσης (GPS), την κατεύθυνση της πυξίδας(προσανατολισμού) και τη μετάδοση ακριβών location events στο υπόλοιπο σύστημα. Η υπηρεσία αξιοποιεί τόσο το Native Capacitor Geolocation για Android και iOS, όσο και το Web Geolocation API (navigator.geolocation) του browser. Για τη διασύνδεση με το UI, η υπηρεσία εκθέτει τρία EventEmitters.

```

@Injectable({ providedIn: 'root' })
export class GpsService {
  // Emits processed position on every valid GPS fix
  public locationFound = new EventEmitter<{
    lat: number; lng: number; accuracy?: number
}>();
  // Emits when GPS fails or permission is denied
  public locationError = new EventEmitter<void>();
  // Emits true when no fix received for > 15 seconds
  public gpsStale = new EventEmitter<boolean>();
}

```

Σχήμα 4.22: Η υπηρεσία ελέγχου του GPS που ειδοποιεί την εφαρμογή για νέες τοποθεσίες ή σφάλματα σήματος.

Ο πυρήνας της ροής επεξεργασίας είναι η private συνάρτηση `handleFix()` που επεξεργάζεται κάθε εισερχόμενη ακατέργαστη (raw) ενημέρωση GPS. Η συνάρτηση αυτή εφαρμόζει μια σειρά από φίλτρα και παραμέτρους. Πρώτα είναι το φίλτρο ανίχνευσης απότομης μετακίνησης (jump detection filter), το οποίο εξαλείφει τις διορθώσεις GPS που το οποίο απορρίπτει τις μετρήσεις GPS που θα προκαλούσαν ένα μη ρεαλιστικό "άλμα" στη θέση του χρήστη. Αυτό γίνεται με τον υπολογισμό της ταχύτητας με βάση την απόσταση που διανύθηκε και τον χρόνο που μεσολάβησε μεταξύ δύο διαδοχικών μετρήσεων. Εάν η υπολογισμένη ταχύτητα υπερβαίνει τα 7,0 m/s (ταχύτητα τρεξίματος) και τυχόν μετρήσεις που δεν δικαιολογούνται να υποστηρίζουν μια τέτοια μεταβολή στην ταχύτητα ή την ακρίβεια, αυτή η διόρθωση απορρίπτεται. Όλη αυτή η διαδικασία απεικονίζεται στο Σχήμα 4.23.

```

private readonly JUMP_MIN_DT_S      = 1.1;
private readonly JUMP_MAX_DIST_M    = 35;
private readonly JUMP_MAX_WALK_MPS  = 7.0;

private isLikelyJump(
    rawNow: L.LatLng, nowMs: number,
    accM: number, speedMps: number
): boolean {
    const dtS      = Math.max(0.001,
        (nowMs - this.rawFixAt) / 1000);
    const distM    = this.rawFixLL!.distanceTo(rawNow);

    if (dtS >= 6) return false; // stale - trust the fix

    if (dtS >= this.JUMP_MIN_DT_S &&
        distM > this.JUMP_MAX_DIST_M) {
        const impliedMps = distM / dtS;
        const speedFast  = speedMps >= 3.5;
        const accBad     = accM >= 18;
        if (!speedFast &&
            (impliedMps > this.JUMP_MAX_WALK_MPS || accBad))
            return true;
    }
    return false;
}

```

Σχήμα 4.23: Αλγόριθμος που ελέγχει και απορρίπτει λανθασμένα στίγματα GPS που θα μπερδεύαν την πλοήγηση.

Μετά την ανίχνευση άλματος, και εφόσον η πλοήγηση είναι ενεργή, η τρέχουσα θέση μεταβιβάζεται στη μέθοδο `RouteService.computeSnappedPosition()` (Σχήμα 4.24), η οποία αναλαμβάνει την προσαρμογή της (snapping) στην πολυγραμμή (polyline) της διαδρομής. Στη συνέχεια, η προσαρμοσμένη θέση εξομαλύνεται μέσω ενός εκθετικού κινητού μέσου όρου (Exponential Moving Average). Η εξομάλυνση EMA εφαρμόζεται μόνο όταν ο χρήστης είναι ακίνητος, για την καταστολή της ταλάντωσης GPS ± 3 έως 8m. Όταν ο χρήστης κινείται, η EMA παρακάμπτεται εντελώς και χρησιμοποιείται απευθείας η snapped θέση. Το alpha είναι σταθερό: 0,20 όταν η ακρίβεια GPS είναι χαμηλή (>35m) και 0,28 διαφορετικά.

```

private getSmoothAlpha(): number {
    const spd = this.lastSpeedMps ?? 0;
    const acc = this.lastAccM ?? 9999;

    if (this.routeSvc.isMapMatchEnabled() && spd >= 0.8)
        return 0.55; // high alpha – trust snapped position
    if (spd >= 1.2) return 0.42;
    if (spd >= 0.6) return 0.32;
    if (acc > 35) return 0.18; // poor accuracy – smooth heavily
    return 0.25; // default
}

private smoothLatLng(next: L.LatLng): L.LatLng {
    if (!this.smoothLL) { this.smoothLL = next; return next; }
    const a = this.getSmoothAlpha();
    this.smoothLL = L.LatLng(
        this.smoothLL.lat + (next.lat - this.smoothLL.lat) * a,
        this.smoothLL.lng + (next.lng - this.smoothLL.lng) * a
    );
    return this.smoothLL;
}

```

Σχήμα 4.24: Μαθηματική εξομάλυνση της κίνησης του χρήστη στον χάρτη, ώστε ο δείκτης (marker) να μην "τρέμει".

Η κατεύθυνση της πυξίδας λαμβάνεται από το DeviceOrientationEvent API. Η υπηρεσία αξιοποιεί το συμβάν deviceorientationabsolute (Σχήμα 4.25), το οποίο παρέχει άμεσα την πραγματική κατεύθυνση του βορρά (True North) καθώς και στο deviceorientation, το οποίο παρέχει μια σχετική κατεύθυνση που είναι λιγότερο αξιόπιστη. Η ακατέργαστη τιμή alpha του συμβάντος μετατρέπεται σε κατεύθυνση πυξίδας και προσαρμόζεται με βάση το screen orientation (γωνία οθόνης). Για τη βελτιστοποίηση των δεδομένων εφαρμόζονται τρία φίλτρα: ένα φίλτρο νεκρής ζώνης (deadzone) 4,0° για την εξάλειψη του θορύβου του αισθητήρα, ένα φίλτρο αιχμής (spike filter) 150° που απορρίπτει τις απότομες κινήσεις εάν η συσκευή είναι ακίνητη και ένα φίλτρο μέγιστης ταχύτητας περιστροφής (COMPASS_MAX_TURN_DPS = 360°/s) που περιορίζει την ταχύτητα περιστροφής κατά τη διάρκεια των animations.

```

private readonly COMPASS_DEADBAND_DEG      = 4.0;
private readonly COMPASS_REJECT_SPIKE_DEG = 150;
private readonly COMPASS_MAX_TURN_DPS     = 360;
private readonly COMPASS_SMOOTH_ALPHA_ABS = 0.30;
private readonly COMPASS_SMOOTH_ALPHA_REL = 0.24;

const computeHeading = (ev: DeviceOrientationEvent) => {
  const alpha = (ev as any).alpha;
  if (typeof alpha !== "number") return null;

  let heading = (360 - alpha) % 360;
  // Correct for screen rotation (portrait/landscape)
  const angle = screen.orientation?.angle ?? 0;
  heading = (heading + angle + 360) % 360;

  const prev = this.compassHeadingDeg ?? this.lastHeadingDeg;
  if (prev !== null) {
    const diff = angleDiffDeg(prev, heading);
    if (Math.abs(diff) < this.COMPASS_DEADBAND_DEG)
      return null; // ignore noise
    if (slow && Math.abs(diff) >= this.COMPASS_REJECT_SPIKE_DEG)
      return null; // reject spike when stationary
    const dtS = (Date.now() - this.compassHeadingAt) / 1000;
    const maxStep = this.COMPASS_MAX_TURN_DPS * dtS;
    heading = applyMaxStep(prev, heading, maxStep);
  }
  const isAbs = (ev as any).absolute === true;
  const a = isAbs
    ? this.COMPASS_SMOOTH_ALPHA_ABS
    : this.COMPASS_SMOOTH_ALPHA_REL;
  return smoothAngle(prev, heading, a);
};

```

Σχήμα 4.25: Υπολογισμός και εξομάλυνση της κατεύθυνσης του χρήστη (πυξίδα) από τους αισθητήρες του κινητού.

4.2.1.5 RouteService

Η RouteService διαχειρίζεται πλήρως τον κύκλο ζωής μιας διαδρομής πλοήγησης. Αυτό περιλαμβάνει την υποβολή αιτήματος για τη διαδρομή στο backend, την απεικόνισή της στον χάρτη, την παρακολούθηση της προόδου του χρήστη και το snapping (προσαρμογή) της θέσης του πάνω στην πολύγραμμή (polyline). Επιπλέον, η υπηρεσία αναγνωρίζει την άφιξη στον προορισμό/τμήμα και ενεργοποιεί αυτόματα τη διαδικασία αναδρομολόγησης (rerouting) σε περίπτωση παρέκκλισης από την πορεία. Για τη βελτιστοποίηση της απόδοσης, η RouteService διατηρεί στη μνήμη έως και τέσσερα επίπεδα Leaflet polylines ταυτόχρονα (Σχήμα 4.26)

```

// Solid blue main route
private currentPolyline: L.Polyline | null = null;
// Grey translucent already-walked segment
private passedPolyline: L.Polyline | null = null;
// Dashed grey line from user position to route start
private approachPolyline: L.Polyline | null = null;
// Dashed blue line for final approach to destination
private endApproachPolyline: L.Polyline | null = null;

```

Σχήμα 4.26: Δήλωση των γραμμών για την οπτική αναπαράσταση της διαδρομής πάνω στον χάρτη.

Για να προσαρμόσουμε τη θέση GPS του χρήστη στη διαδρομή, πρώτα μετατρέπουμε τις συντεταγμένες στον προβολικό χώρο Web Mercator (EPSG 3857). Χρησιμοποιώντας τον προβολικό χώρο Σχήμα 4.27), μπορούμε να υπολογίσουμε την απόσταση μονάδα χάρτη, η οποία είναι πιο ομοιόμορφη από τον χώρο lat/lon. Αυτό μας βοηθά να υπολογίσουμε με μεγαλύτερη ακρίβεια την κάθετη απόσταση από τα ευθύγραμμα τμήματα (segments). Η προσαρμογή της θέσης GPS του χρήστη στη διαδρομή βασίζεται στην εύρεση του πλησιέστερου σημείου σε οποιοδήποτε τμήμα ης τρέχουσας πολύγραμμής (polyline).

```

private snapToCurrentRoute(
  raw: L.LatLng
): { snapped: L.LatLng; distM: number } | null {
  const pts = this.currentRoutePoints;
  if (!pts || pts.length < 2) return null;

  const P = L.CRS.EPSG3857.project(raw);
  let bestD = Infinity;
  let bestPt: L.Point | null = null;

  for (let i = 0; i < pts.length - 1; i++) {
    const A = L.CRS.EPSG3857.project(pts[i]);
    const B = L.CRS.EPSG3857.project(pts[i + 1]);
    const res = this.closestPointOnSegment(P, A, B);
    if (res.dist < bestD) {
      bestD = res.dist;
      bestPt = res.pt;
    }
  }

  const snappedLL = L.CRS.EPSG3857.unproject(bestPt!);
  return { snapped: snappedLL, distM: bestD };
}

```

Σχήμα 4.27: Διαδικασία υπολογισμού της απόστασης του χρήστη από την επιλεγμένη διαδρομή (snapping) πάνω στον χάρτη.

Ο μηχανισμός snapping λειτουργεί με αργοπορία: ενεργοποιείται όταν ο χρήστης βρίσκεται σε απόσταση μικρότερη των 12m (SNAP_ENTER_M) της διαδρομής και θα απενεργοποιηθεί μόνο αφού απομακρυνθεί ο χρήστης σε απόσταση μεγαλύτερη των 18m (SNAP_EXIT_M). Η τελική θέση που αναφέρθηκε είναι ένας σταθμισμένος συνδυασμός της ακατέργαστης θέσης GPS και της θέσης snap, με το βάρος να κυμαίνεται από 0,60 έως 0,97 και να μετακινείται προς τη θέση GPS καθώς απομακρύνεστε από τη διαδρομή (Σχήμα 4.28).

```

private readonly SNAP_ENTER_M = 12;
private readonly SNAP_EXIT_M = 18;
private readonly SNAP_FULL_M = 5; // full snap weight
private readonly SNAP_BLEND_M = 30; // start blending

const w =
  d <= this.SNAP_FULL_M ? 0.97 :
  d >= this.SNAP_BLEND_M ? 0.80 :
  0.97 - (d - this.SNAP_FULL_M) *
  (0.17 / (this.SNAP_BLEND_M - this.SNAP_FULL_M));

const chosen = L.latLng(
  rawNow.lat + (target.lat - rawNow.lat) * w,
  rawNow.lng + (target.lng - rawNow.lng) * w
);

```

Σχήμα 4.28: Ομαλή προσαρμογή (blending) της θέσης του χρήστη πάνω στη γραμμή της διαδρομής, ώστε η κίνηση στον χάρτη να φαίνεται φυσική.

Η συνάρτηση `MaybeReroute()` όπως φαίνεται στο Σχήμα 4.29, εκτελείται με κάθε νέο εντοπισμό GPS για να χειριστεί την αλλαγή διαδρομής (ανακατεύθυνσης). Αν δεν υπάρξουν πέντε συνεχόμενες ενδείξεις εκτός διαδρομής, η αναδρομολόγηση δεν ενεργοποιείται ενώ μετά από κάθε αλλαγή διαδρομής εφαρμόζεται αναμονή 8 δευτερολέπτων. Αν το API έχει δύο συνεχόμενες αποτυχίες, ενεργοποιείται ένας μηχανισμός προστασίας (circuit breaker) και θα υπάρξει καθυστέρηση 30 δευτερολέπτων σε οποιαδήποτε αλλαγή διαδρομής. Τέλος, εάν ο χρήστης βρίσκεται σε απόσταση μικρότερη των 45m από τον προορισμό, η αναδρομολόγηση παραλείπεται (skip).

```
private readonly REROUTE_COOLDOWN_MS      = 8000;
private readonly REROUTE_CONFIRM_FIXES    = 5;
private readonly REROUTE_OFF_M           = 25;
private readonly REROUTE_ON_M            = 10;
private readonly REROUTE_MAX_ACC_M       = 22;
private readonly REROUTE_MIN_SPEED_MPS   = 0.35;
private readonly REROUTE_SKIP_NEAR_DEST_M = 45;
private readonly REROUTE_BLOCK_MS        = 30_000;
```

Σχήμα 4.29: Βασικές ρυθμίσεις και όρια απόστασης/χρόνου για την ενεργοποίηση του αυτόματου επαναυπολογισμού της διαδρομής (rerouting).

Η ανίχνευση άφιξης χρησιμοποιεί έλεγχο σε δύο στάδια (Σχήμα 4.30). Ο χρήστης πρέπει να βρίσκεται σε απόσταση μικρότερη ή ίση των 8 μέτρων από την είσοδο και η ακρίβεια του GPS πρέπει να είναι μικρότερη ή ίση από 15 μέτρα, κάτι που πρέπει να έχει επιβεβαιωθεί σε δύο συνεχόμενες τοποθεσίες. Αυτό διασφαλίζει ότι ο χρήστης δεν θα αναγνωριστεί λανθασμένα ότι έφτασε λόγω απόκλισης του GPS.

```
private readonly ARRIVE_PIN_DIST_M      = 8;
private readonly ARRIVE_PIN_MAX_ACC_M   = 15;
private readonly ARRIVE_PIN_STREAK      = 2;

checkArrival(rawNow: L.LatLng, acc: number): void {
    if (this.arrivedNearPinTriggered) return;
    const accOk = isFinite(acc) && acc <= this.ARRIVE_PIN_MAX_ACC_M;
    if (!accOk) { this.arriveStreak = 0; return; }

    const d = rawNow.distanceTo(this.activeEndPoint!);
    if (isFinite(d) && d <= this.ARRIVE_PIN_DIST_M) {
        this.arriveStreak++;
        if (this.arriveStreak >= this.ARRIVE_PIN_STREAK) {
            this.arrivedNearPinTriggered = true;
            this.arrivedNearPin.emit();
        }
    } else {
        this.arriveStreak = 0;
    }
}
```

Σχήμα 4.30: Λογική ελέγχου άφιξης, η οποία επιβεβαιώνει ότι ο χρήστης έχει προσεγγίσει σταθερά και με ακρίβεια τον τελικό του προορισμό.

4.2.1.6 MapService

Η συνάρτηση MapService που απεικονίζεται στο Σχήμα 4.31, διαχειρίζεται το αντικείμενο χάρτη Leaflet και όλα όσα εμφανίζονται στον χάρτη, συμπεριλαμβανομένου του layer tiles, του δείκτη θέσης χρήστη, της παρακολούθησης από την κάμερα και της περιστροφής του χάρτη. Το MapService περιέχει δύο στυλ για το εικονίδιο δείκτη χρήστη, ένα για το βέλος που υποδεικνύει τη λειτουργία πλοήγησης και ένα για την κουκκίδα για την κατάσταση αναμονής, τα οποία εναλλάσσονται κατά την έναρξη και το τέλος της πλοήγησης.

```
private userArrowIcon = L.divIcon({
  className: 'user-marker',
  html: ``,
  iconSize: [28, 28], iconAnchor: [14, 14],
});

private userDotIcon = L.divIcon({
  className: 'user-dot',
  html: `<div class="cone"></div>
    <span class="dot"></span>
    <span class="halo"></span>`,
  iconSize: [28, 28], iconAnchor: [14, 14],
});
```

Σχήμα 4.31: Ορισμός των γραφικών στοιχείων (εικονίδιο κατεύθυνσης και τελεία) για την αναπαράσταση της θέσης του χρήστη στον χάρτη.

Η κίνηση του δείκτη πραγματοποιείται με την τεχνική του Dead Reckoning στο Σχήμα 4.32. Μόλις ληφθεί ένα νέο στίγμα GPS, ο δείκτης μετακινείται ομαλά από την τρέχουσα στη νέα θέση μέσω της μεθόδου requestAnimationFrame(). Στο μεσοδιάστημα των ενημερώσεων, ο δείκτης συνεχίζει να κινείται βάσει της τελευταίας γνωστής ταχύτητας (σε μοίρες ανά ms), επιτρέποντας την προέκταση (extrapolation) της θέσης στον χρόνο για μέγιστο διάστημα 2 δευτερολέπτων (EXTRAP_MAX_MS).

```

private vellat = 0; // degrees / ms
private vellng = 0;
private readonly EXTRAP_MAX_MS = 2000;

// Animation loop – runs every frame via requestAnimationFrame
private animateUserMarker(): void {
  const now = performance.now();
  const dt = now - this.animStart;
  const total = this.lastFixAt
    ? Math.max(1, this.lastFixAt - this.animStart)
    : 500;
  const t = Math.min(1, dt / total);

  // Lerp from animFrom to animTo
  const lat = this.animFrom!.lat +
    (this.animTo!.lat - this.animFrom!.lat) * t;
  const lng = this.animFrom!.lng +
    (this.animTo!.lng - this.animFrom!.lng) * t;

  this.userMarker.setLatLng([lat, lng]);
  if (t < 1)
    this.animReq = requestAnimationFrame(
      () => this.animateUserMarker()
    );
}

```

Σχήμα 4.32: Μηχανισμός ομαλής κίνησης (animation) του εικονιδίου του χρήστη πάνω στον χάρτη μεταξύ δύο διαδοχικών στιγμάτων GPS.

Η περιστροφή της κατεύθυνσης επιτυγχάνεται μέσω της κλήσης μιας μεθόδου `setBearing()` που παρέχεται από το πρόσθετο Leaflet. Προκειμένου να αποφευχθεί η υπερβολική κλήση της διαδικασίας απόδοσης, η περιστροφή επιτυγχάνεται μόνο εάν η απόκλιση στην κατεύθυνση σε σχέση με την τρέχουσα κατεύθυνση του χάρτη είναι μεγαλύτερη από 40° (`AUTO_ROTATE_DEG`). Επιπλέον, εφαρμόζεται ένα χειροκίνητο κλείδωμα περιστροφής (`MANUAL_LOCK_MS = 5` δευτερόλεπτα) για την αποφυγή της αυτόματης επαναφοράς, αφού ο χρήστης περιστρέψει τον χάρτη με χειρονομία (gesture). Αυτή η διαδικασία απεικονίζεται στο Σχήμα 4.33.

```

private readonly MANUAL_LOCK_MS = 5000;
private readonly AUTO_ROTATE_DEG = 40;

applyHeading(deg: number): void {
  this.gpsHeadingDeg_ = deg;
  const manualLock =
    Date.now() - this.lastManualRotateAt < this.MANUAL_LOCK_MS;
  if (manualLock) return;

  const diff = Math.abs(angleDiffDeg(this.mapBearingDeg, deg));
  if (diff < this.AUTO_ROTATE_DEG) return;

  this.rotateTo(deg);
}

```

Σχήμα 4.33: Λογική ελέγχου για την αυτόματη περιστροφή του χάρτη σύμφωνα με την κατεύθυνση της πυξίδας της συσκευής.

4.2.1.7 Models

Τα μοντέλα αποτελούν αποκλειστικά TypeScript interfaces χωρίς επιχειρησιακή λογική. Το κεντρικό μοντέλο είναι το Destination, το οποίο είναι πανομοιότυπο σε frontend και backend. Επιπλέον, το frontend ορίζει τα CampusRouteParams, CampusRouteResponse και LatLngPoint στην ApiService, καθώς και το AppSettings στη SettingsService. Κάθε αρχείο εξάγει τόσο τους τύπους όσο και τα στατικά δεδομένα. Για παράδειγμα, το destination.model.ts εξάγει το interface Destination μαζί με τον σταθερό πίνακα destinationList. Αυτό επιτρέπει την άμεση πρόσβαση στη λίστα προορισμών χωρίς την ανάγκη κλήσης κάποιου API, καθώς τα δεδομένα είναι hardcoded στην εφαρμογή.

4.2.1.8 map-geo.utils

Η συνάρτηση module map-geo.utils που φαίνεται στο Σχήμα 4.32, περιλαμβάνει τέσσερις pure βοηθητικές συναρτήσεις που χρησιμοποιούνται από πολλαπλές υπηρεσίες. Κάθε συνάρτηση είναι stateless (χωρίς κατάσταση) και δεν προκαλεί παρενέργειες (side effects). Η συνάρτηση bearingDeg() υπολογίζει το αζιμούθιο (γωνία πυξίδας) μεταξύ δύο σημείων σε μορφή LatLng, εφαρμόζοντας τον σφαιρικό νόμο των ημιτόνων:

```
export function bearingDeg(
  from: L.LatLng, to: L.LatLng
): number {
  const lat1 = (from.lat * Math.PI) / 180;
  const lat2 = (to.lat * Math.PI) / 180;
  const dLng = ((to.lng - from.lng) * Math.PI) / 180;

  const y = Math.sin(dLng) * Math.cos(lat2);
  const x = Math.cos(lat1) * Math.sin(lat2)
    - Math.sin(lat1) * Math.cos(lat2) * Math.cos(dLng);

  return ((Math.atan2(y, x) * 180 / Math.PI) + 360) % 360;
}
```

Σχήμα 4.34: Μαθηματικός υπολογισμός της γωνίας κατεύθυνσης (αζιμούθιο) μεταξύ δύο γεωγραφικών σημείων στον χάρτη.

Η συνάρτηση smoothAngle() (Σχλημα 4.35) υπολογίζει τον εκθετικό κινητό μέσο όρο της γωνίας, ενώ χειρίζεται σωστά τη διακοπή των 0/360 μοιρών, χρησιμοποιώντας τη διαφορά της γωνίας και όχι την ίδια τη γωνία. Η συνάρτηση angleDiffDeg() επιστρέφει τη συνυπογεγραμμένη μικρότερη διαφορά μεταξύ των δύο γωνιών, εντός του διαστήματος (-180, +180]. Η συνάρτηση applyMaxStep() περιορίζει το ρυθμό αλλαγής της γωνίας, αποτρέποντας την πυξίδα από το να περιστρέφεται ανεξέλεγκτα.

```

// Signed shortest angular difference: result in (-180, +180]
export function angleDiffDeg(a: number, b: number): number {
  return ((b - a + 540) % 360) - 180;
}

// EMA on angles – handles 0/360 wraparound
export function smoothAngle(
  prev: number | null, next: number, alpha = 0.25
): number {
  if (prev == null) return next;
  const diff = angleDiffDeg(prev, next);
  return (prev + diff * alpha + 360) % 360;
}

// Clamp angular step to prevent too-fast rotation
export function applyMaxStep(
  prev: number, next: number, maxStep: number
): number {
  const d = angleDiffDeg(prev, next);
  if (Math.abs(d) <= maxStep) return next;
  return (prev + Math.sign(d) * maxStep + 360) % 360;
}

// Total length of a polyline in meters
export function sumDistanceMeters(pts: L.LatLng[]): number {
  if (!pts || pts.length < 2) return 0;
  let total = 0;
  for (let i = 1; i < pts.length; i++)
    total += pts[i-1].distanceTo(pts[i]);
  return total;
}

```

Σχήμα 4.35: Βοηθητικές συναρτήσεις για την ομαλή περιστροφή της πυξίδας της εφαρμογής και τον υπολογισμό της συνολικής απόστασης μιας διαδρομής.

4.2.2 GPS, Dead-Reckoning και Navigation Camera

Η εφαρμογή UniMap υλοποιεί ένα προηγμένο σύστημα εντοπισμού θέσης και παρακολούθησης χρήστη, που συνδυάζει GPS με τεχνική dead-reckoning για ομαλή κινούμενη εμπειρία χρήστη. Το GPS παρέχει νέες συντεταγμένες σε ακανόνιστα χρονικά διαστήματα (συνήθως ανά 1-5 δευτερόλεπτα), τα οποία από μόνα τους θα οδηγούσαν σε τρεμουλιαστή και μη ρεαλιστική κίνηση του marker του χρήστη στον χάρτη. Για την αντιμετώπιση αυτού του προβλήματος, υλοποιήθηκε μηχανισμός παρεμβολής κίνησης (dead-reckoning) που εκτιμά τη θέση του χρήστη ανάμεσα στις λήψεις GPS.

Κάθε φορά που λαμβάνεται νέα θέση GPS, υπολογίζεται η ταχύτητα μετατόπισης σε μοίρες ανά χιλιστό του δευτερολέπτου, τόσο για το γεωγραφικό πλάτος (velLat) όσο και για το γεωγραφικό μήκος (velLng). Ανάμεσα στα GPS fixes, ένας animation loop (requestAnimationFrame) προβλέπει τη θέση χρησιμοποιώντας αυτές τις ταχύτητες, δημιουργώντας ρεαλιστική κινούμενη εικόνα του χρήστη που κινείται πάνω στον χάρτη.

Επιπρόσθετα, η εφαρμογή υλοποιεί λειτουργία πλοήγησης με προσανατολισμένο χάρτη (heading-up navigation camera). Στη λειτουργία αυτή, ο χάρτης περιστρέφεται αυτόματα ώστε η κατεύθυνση

κίνησης του χρήστη να δείχνει πάντα προς την κορυφή της οθόνης, διευκολύνοντας τον προσανατολισμό. Ο χρήστης μπορεί να περιστρέψει χειροκίνητα τον χάρτη, οπότε ενεργοποιείται κλειδώμα χειροκίνητης περιστροφής (manual rotation lock) για 5 δευτερόλεπτα (MANUAL_LOCK_MS), μετά το οποίο επανέρχεται αυτόματα η αυτόματη περιστροφή. Η αυτόματη διόρθωση ενεργοποιείται όταν η απόκλιση μεταξύ της πραγματικής κατεύθυνσης και της οθόνης υπερβαίνει 40 μοίρες (AUTO_ROTATE_DEG). Η κάμερα ακολουθεί τον χρήστη κεντράροντας τον χάρτη στην τρέχουσα θέση του σε κάθε frame animation (setView), χωρίς επιπλέον μετατόπιση, αφήνοντας περισσότερο χώρο μπροστά για να βλέπει τη διαδρομή που ακολουθεί.

4.2.3 Απαιτήσεις Εφαρμογής και Περιπτώσεις Χρήσης (Use Cases)

Βασικός σκοπός της εφαρμογής UniMap είναι η διευκόλυνση του χρήστη κατά την περιήγησή του στην πανεπιστημιούπολη. Πριν την αναλυτική παρουσίαση των οθονών της διεπαφής, κρίνεται απαραίτητος ο καθορισμός των βασικών λειτουργικών απαιτήσεων του συστήματος, οι οποίες εκφράζονται μέσα από τις παρακάτω κύριες Περιπτώσεις Χρήσης (Use Cases):

PX-1 (Εκκίνηση και Εντοπισμός): Ο χρήστης εκκινεί την εφαρμογή και παραχωρεί άδεια GPS για τον αυτόματο εντοπισμό της θέσης του στον χάρτη του campus.

PX-2 (Αναζήτηση και Επιλογή Προορισμού): Ο χρήστης αναζητά ένα συγκεκριμένο κτίριο, τμήμα ή σημείο ενδιαφέροντος (είτε μέσω της μπάρας αναζήτησης είτε επιλέγοντας σημείο στον χάρτη) και βλέπει τις πληροφορίες του.

PX-3 (Πλοήγηση και Παρακολούθηση Διαδρομής): Ο χρήστης λαμβάνει οδηγίες βήμα-βήμα από το σημείο που βρίσκεται προς τον επιλεγμένο προορισμό, παρακολουθώντας την πορεία του σε πραγματικό χρόνο με αυτόματη περιστροφή χάρτη (heading-up).

PX-4 (Πλοήγηση Προσβασιμότητας - ΑμεΑ): Ο χρήστης ενεργοποιεί την ειδική λειτουργία πλοήγησης, ώστε η εφαρμογή να σχεδιάσει διαδρομή χωρίς εμπόδια ή σκαλοπάτια για την εξυπηρέτηση ατόμων με μαξιτίδιο.

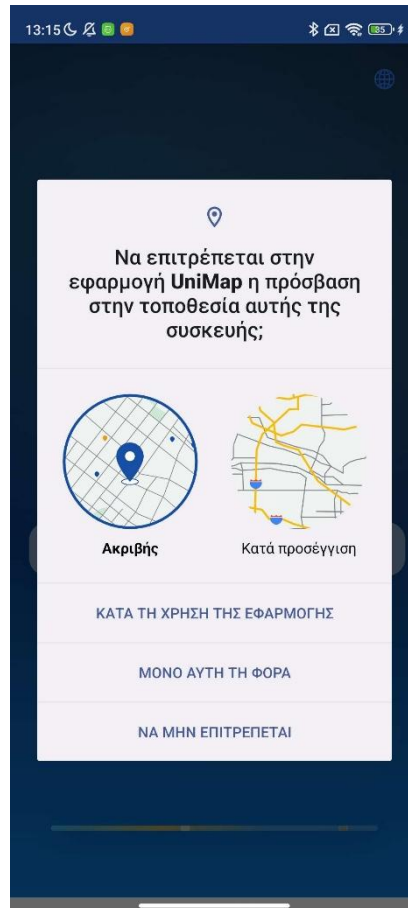
PX-5 (Προσαρμογή Ρυθμίσεων): Ο χρήστης τροποποιεί την εμφάνιση του χάρτη, αλλάζει τη μονάδα μέτρησης αποστάσεων και εναλλάσσει τη γλώσσα της εφαρμογής (Ελληνικά/Αγγλικά).

4.2.4 Οθόνες Διεπαφής Χρηστών (UI)

Στην ενότητα αυτή παρουσιάζονται αναλυτικά οι κύριες οθόνες της εφαρμογής UniMap. Για κάθε οθόνη παρατίθεται η οπτική αναπαράσταση, η λειτουργική της περιγραφή, καθώς και η αντίστοιχη περίπτωση χρήσης (use case) από τις παραπάνω, την οποία έρχεται να καλύψει. Η εφαρμογή UniMap είναι δίγλωσση (Ελληνικά και Αγγλικά), οπότε για λόγους απλοποίησης, θα παρουσιαστεί μόνο η έκδοση στα Ελληνικά, η οποία είναι και η βασική για την εφαρμογή. Σε κάθε περίπτωση θα δείξουμε αναλυτικά και την αλλαγή γλώσσας εντός της εφαρμογής.

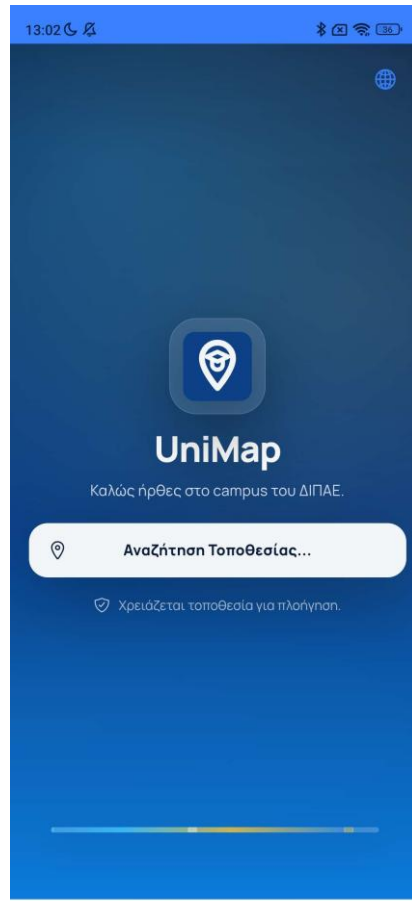
4.2.4.1 Οθόνη 1: Splash Screen — Εκκίνηση και Αίτηση GPS

Με την πρώτη εκκίνηση της εφαρμογής, το UniMap ζητά από τον χρήστη την άδεια πρόσβασης στην τοποθεσία της συσκευής του μέσω του συστήματος GPS όπως βλέπουμε στο Σχήμα 4.36. Η ενέργεια αυτή είναι σημαντικής σημασίας, καθώς η τοποθεσία αποτελεί τον πυρήνα της λειτουργικότητας της εφαρμογής. Χωρίς αυτήν, δεν είναι δυνατή η παροχή οδηγιών πλοήγησης, ο εντοπισμός του χρήστη στον χάρτη ή η προβολή τμημάτων και γενικά της πανεπιστημιούπολης που βρίσκονται σε κοντινή απόσταση.



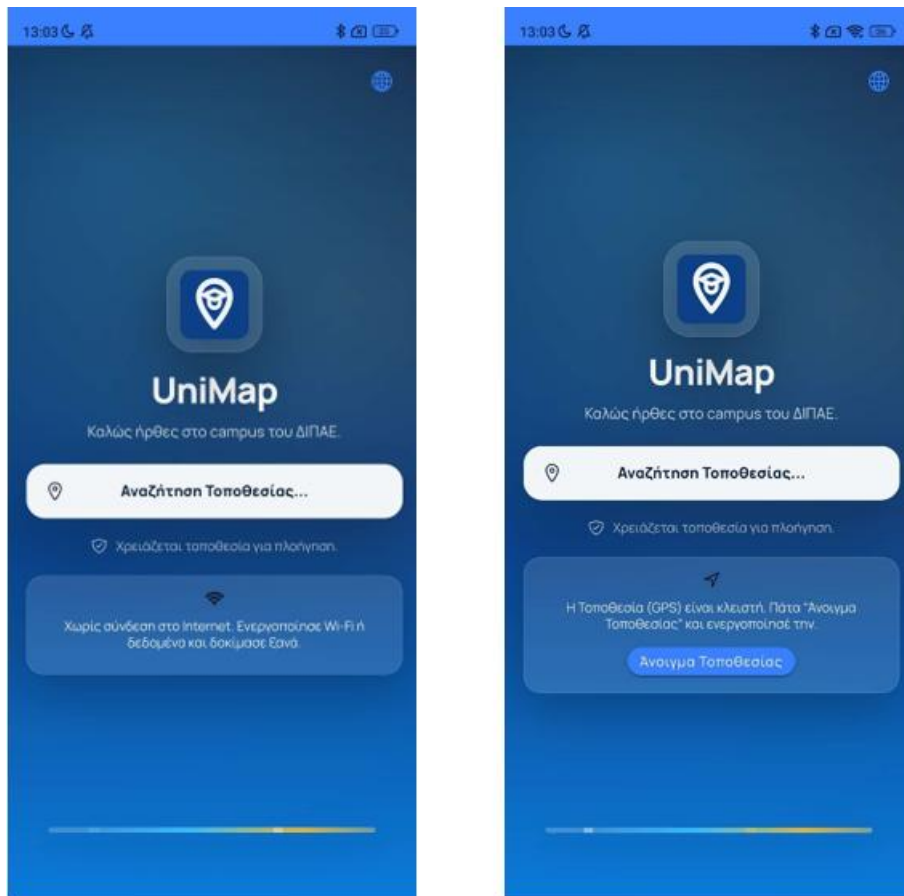
Σχήμα 4.36: Οθόνη αδειάς πρόσβασης GPS

Στην περίπτωση που ο χρήστης αποδεχτεί την τοποθεσία του, βλέπει την οθόνη εκκίνησης (splash screen), στην οποία εμφανίζονται κάποια βασικά αντικείμενα (Σχήμα 4.37). Για αρχή, στο κέντρο βλέπουμε το logo της εφαρμογής, στην γωνία πάνω δεξιά παρατηρούμε ένα κουμπί το οποίο αλλάζει την γλώσσα σε Αγγλικά, ένα μήνυμα καλωσορίσματος, στην συνέχεια το βασικό κουμπί Αναζήτηση Τοποθεσίας το οποίο δεν είναι απλό κουμπί αλλά όλη η ουσία αυτής της πρώτης επαφής με την εφαρμογή όπως αναφέραμε. Κατά το πάτημα του συγκεκριμένου κουμπιού, το σύστημα εκτελεί δύο κρίσιμες λειτουργίες αρχικοποίησης: γίνεται αίτηση από τη συσκευή στην άδεια πρόσβασης στο GPS και φορτώνει τα απαραίτητα γεωγραφικά δεδομένα (campus tiles).



Σχήμα 4.37: Splash Screen εφαρμογής UniMap

Στην συνέχεια και αφού έχει πατηθεί το κουμπί για εντοπισμό τοποθεσίας έχουμε κάποιους μηχανισμούς πίσω από όλο αυτό που βοηθάνε στην ομαλή μετάβαση. Όπως βλέπουμε στο Σχήμα 4.38, όταν ο χρήστης ανοίγει την εφαρμογή και δεν έχει internet (Wi-Fi ή δεδομένα), τότε η εφαρμογή δεν μπορεί να ξεκινήσει να φορτώσει τα δεδομένα του χάρτη και εμφανίζεται ενημερωτικό μήνυμα (Σχήμα 4.38a). Στην περίπτωση, που η τοποθεσία δεν είναι ανοιχτή βγάζει επίσης ενημερωτικό μήνυμα το οποίο είναι βοηθητικό και ανοίγει τις ρυθμίσεις τοποθεσίας της εφαρμογής (Σχήμα 4.38b).



(a) (b)
Σχήμα 4.38: Μηνύματα σφάλματος κατά τη φόρτωση της εφαρμογής

4.2.4.2 Οθόνη 2: Home Page (Κύρια οθόνη) – Χάρτης Campus

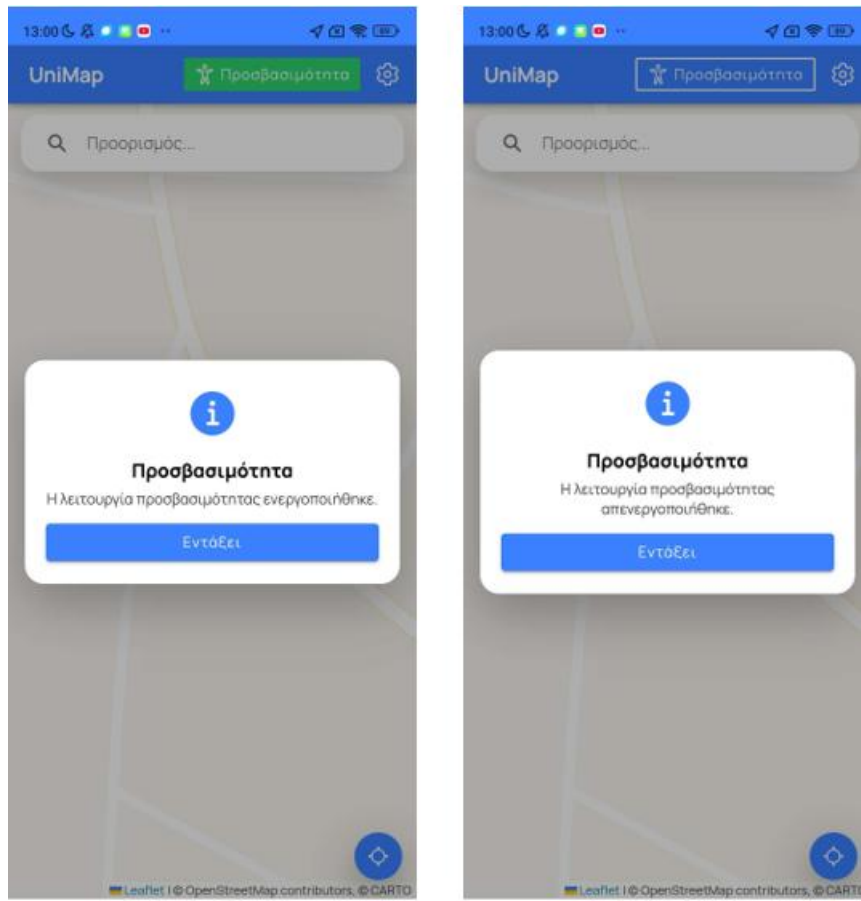
Η κύρια οθόνη της εφαρμογής UniMap (Σχήμα 4.39) αποτελεί το βασικό στοιχείο όλης της εφαρμογής καθώς σε αυτή την οθόνη γίνονται όλες οι αλληλεπιδράσεις με τον χρήστη. Αφού φορτωθεί ο διαδραστικός χάρτης φορτωμένος με τα tiles από το MapTiler, εμφανίζεται η τοποθεσία του χρήστη στην οποία βρίσκεται την εκάστοτε στιγμή. Ο χρήστης μπορεί την δεδομένη στιγμή να βρίσκεται εκτός campus, θα βγάλει την τοποθεσία του χρήστη στην πραγματική τοποθεσία του αλλά η αλληλεπίδραση δεν θα μπορεί να προχωρήσει. Για κάθε μια περίπτωση θα αναλυθεί ξεχωριστά.

Στην κύρια οθόνη, ο χρήστης εντοπίζει την τοποθεσία του στον διαδραστικό χάρτη, μαζί με το λογότυπο, τις Ρυθμίσεις, το κουμπί Προσβασιμότητας και το κεντρικό πεδίο αναζήτησης προορισμών. Στο κάτω δεξιά τμήμα, το κουμπί Επανεστίασης επιτρέπει την άμεση επιστροφή στην τρέχουσα θέση του χρήστη.



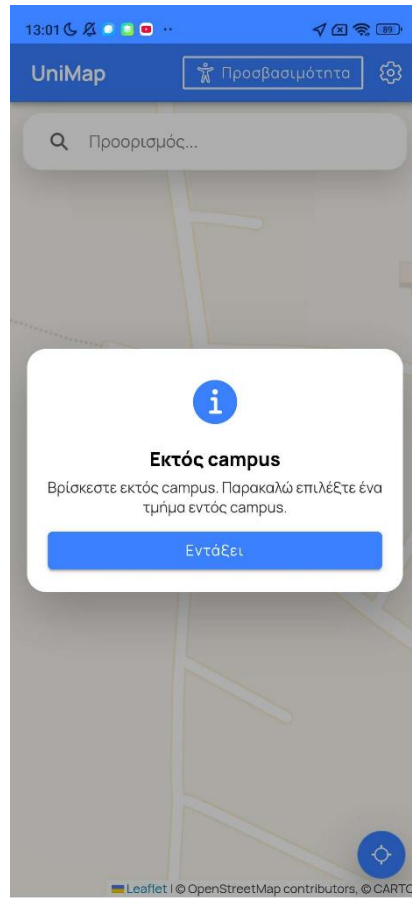
Σχήμα 4.39: Κύρια οθόνη χάρτη campus.

Το κουμπί Προσβασιμότητα βρίσκεται στο toolbar της κύριας οθόνης και επιτρέπει στον χρήστη να ενεργοποιήσει την λειτουργία πλοήγησης για άτομα με κινητικά προβλήματα. Καθώς ανοίγει η εφαρμογή, το κουμπί αυτό είναι ανενεργό, ενώ όταν ο χρήστης το επιλέξει, ενεργοποιείται και αλλάζει σε πράσινο χρώμα, ώστε να ξέρουμε ότι έχει αλλάξει η κατάσταση. Κατά Ενεργοποίηση / Απενεργοποίηση της Προσβασιμότητας, βγάζει κατάλληλο μήνυμα (Σχήμα 4.40), προειδοποιώντας τον χρήστη ότι έχει ενεργοποιηθεί, βοηθώντας τον χρήστη να καταλάβει καθώς μπορεί να πατήθηκε ακόμη καταλάθος.



Σχήμα 4.40: Ενεργοποίηση / Απενεργοποίηση Προσβασιμότητας

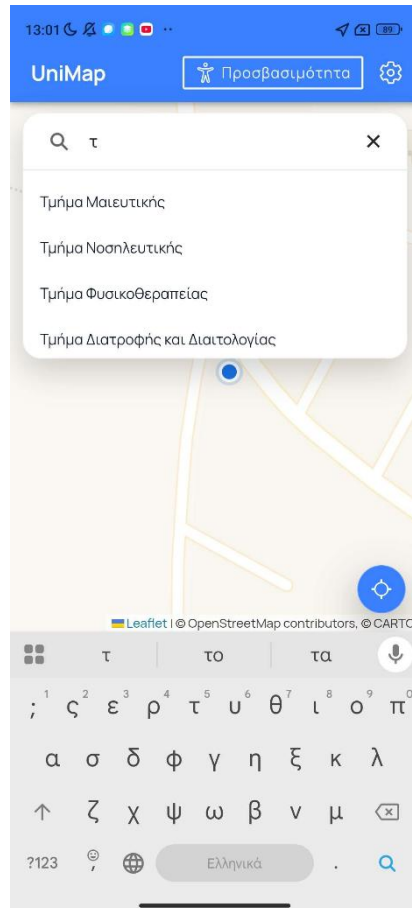
Τέλος, κατά την πλοήγηση στην αρχική-κύρια οθόνη, θα πρέπει ο χρήστης να γνωρίζει ότι η πλοήγηση γίνεται ΜΟΝΟ εντός πανεπιστημιούπολης. Αυτό σημαίνει ότι αν ο χρήστης επιλέξει κάποιο άλλο σημείο/προορισμό εκτός πανεπιστημιούπολης, θα βγάλει κατάλληλο μήνυμα για προειδοποίηση (Σχήμα 4.41). Η λεπτομερέστατη ανάλυση θα γίνεται σε παρακάτω υποκεφαλαία.



Σχήμα 4.41: Προειδοποιητικό μήνυμα επιλογής προορισμού εκτός πανεπιστημιούπολης

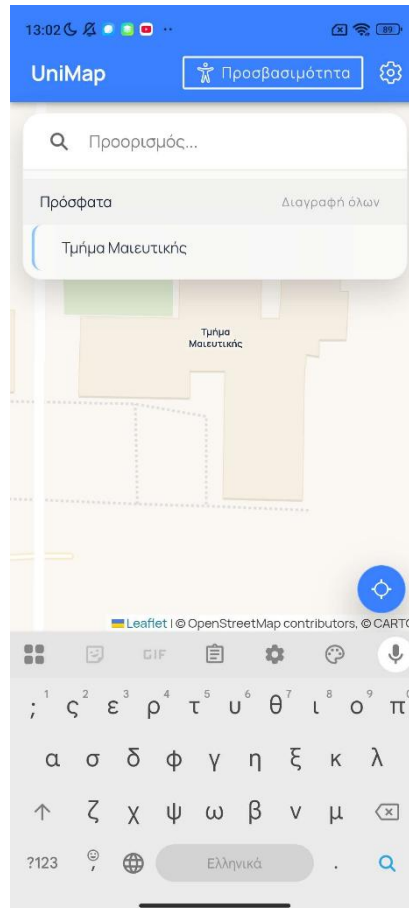
4.2.4.2.1 Αναζήτηση Προορισμού

Πατώντας ο χρήστης πάνω στην μπάρα αναζήτησης, μπορεί να αναζητήσει ένα από τα τμήματα που θέλει να περιηγηθεί. Η πληκτρολόγηση είναι ρυθμισμένη έτσι ώστε οτιδήποτε πληκτρολογήσει ο χρήστης να εμφανίζεται σαν drop-down menu και να μπορεί να επιλέξει αυτό που ο χρήστης επιθυμεί. Επίσης, και χωρίς τονισμό το αποτέλεσμα δεν διαφοροποιείται. Αν ο χρήστης αλλάξει γνώμη, τότε πατάει το κουμπί X και σβήνεται ότι έχει πληκτρολογήσει. Όλα τα παραπάνω απεικονίζονται στο Σχήμα 4.42.



Σχήμα 4.42: Αναζήτηση προορισμού με drop list

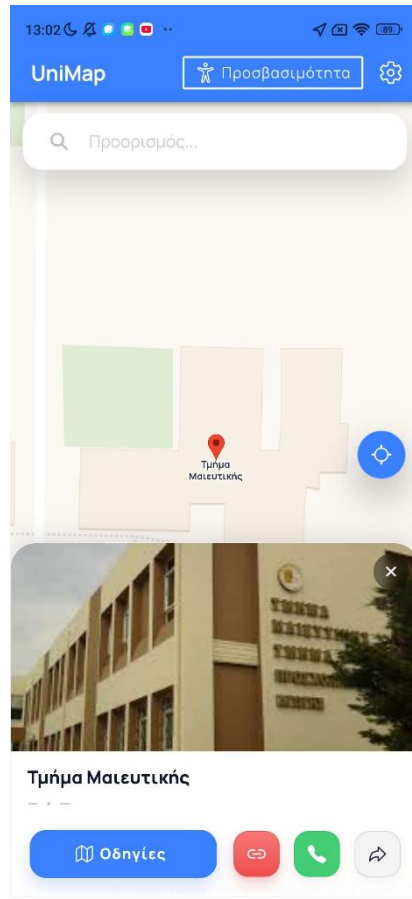
Επιπλέον, η εφαρμογή διατηρεί ιστορικό πρόσφατων αναζητήσεων. Στην περίπτωση που ο χρήστης επιλέξει ένα τμήμα, αυτό αποθηκεύεται τοπικά στην συσκευή με μέγιστο αριθμό των 6 τμημάτων. Την επόμενη φορά που ο χρήστης θα ανοίξει την μπάρα αναζήτησης θα βρει τις πρόσφατες αναζητήσεις του με την δυνατότητα επιλογής τους χωρίς επαναπληκτρολόγηση καθώς και θα μπορεί να τις διαγράψει μαζικά με το κουμπί Διαγραφή όλων (Σχήμα 4.43).



Σχήμα 4.43: Ιστορικό πρόσφατων αναζητήσεων

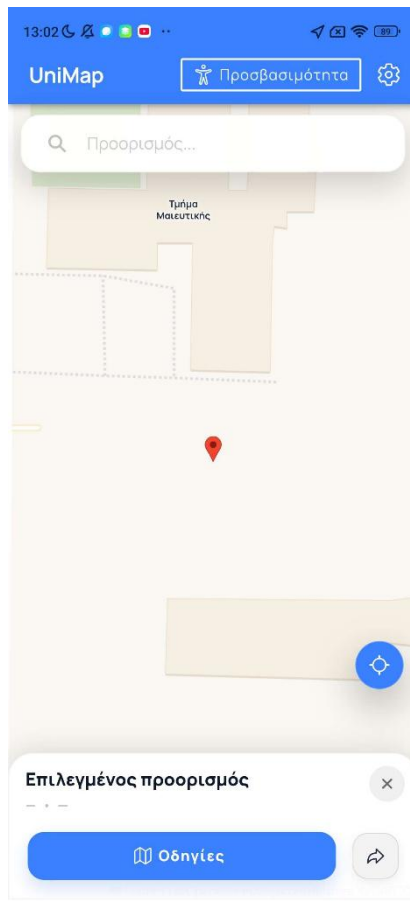
4.2.4.2.2 Ρομπ Πληροφοριών Τμήματος

Όταν ο χρήστης επιλέξει έναν προορισμό από την μπάρα αναζήτησης, είτε με το πάτημα απευθείας πάνω στον χάρτη, τότε εμφανίζεται αυτόματα το αναδυόμενο παράθυρο πληροφοριών. Το συγκεκριμένο αναδυόμενο παράθυρο ανοίγει στο κάτω μέρος της οθόνης και εμφανίζει κάποια στοιχεία του τμήματος, όπως φαίνεται στο Σχήμα 4.44, εμφανίζεται η εικόνα του τμήματος, το όνομα του τμήματος, το κουμπί οδηγίες, το οποίο είναι το πιο σημαντικό για την διαδικασία της έναρξης της πλοήγησης εντός της πανεπιστημιούπολης, τα μέτρα και τα λεπτά της απόστασης (τα οποία είναι disable αλλά γίνονται ενεργά μόλις υπολογιστεί η διαδρομή), το κόκκινο κουμπί το οποίο σε παραπέμπει στην επίσημη ιστοσελίδα του τμήματος, το πράσινο κουμπί με το τηλέφωνο της Γραμματείας και τέλος το τελευταίο κουμπί σε αυτό το αναδυόμενο παράθυρο που είναι η κοινοποίηση του τμήματος που δίνονται οι συντεταγμένες του τμήματος/προορισμού μέσω του Google Maps. Εάν ο χρήστης θέλει να ακυρώσει την επιλογή του τμήματος, μπορεί να την ακυρώσει μέσω του κουμπιού X που υπάρχει σε αυτό το παράθυρο.



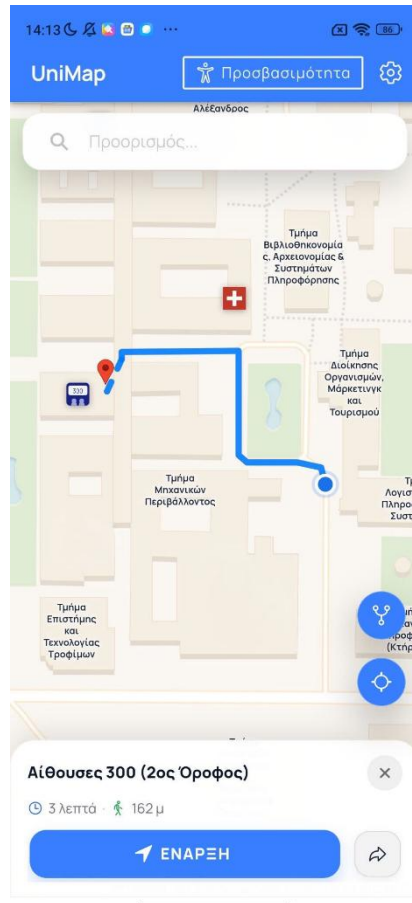
Σχήμα 4.44: Ρομπ πληροφοριών τμήματος

Στην περίπτωση που δεν υπάρχει κάποιο συγκεκριμένο τμήμα να περιηγηθεί ο χρήστης και έχει επιλεγεί ένα τυχαίο εντός της πανεπιστημιούπολης, τότε εμφανίζεται μόνο ο συγκεκριμένος X προορισμός με την ονομασία «Επιλεγμένος προορισμός» (Σχήμα 4.45).



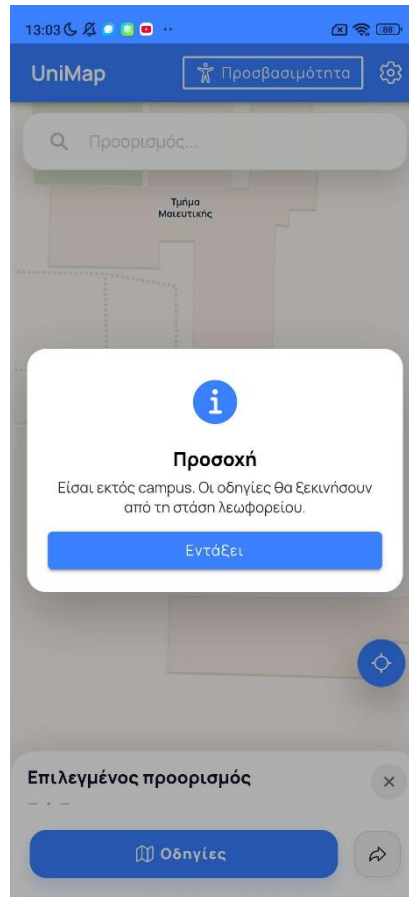
Σχήμα 4.45: Popup πληροφοριών τυχαίου σημείου εντός campus

Αφού ο χρήστης επιλέξει ένα τμήμα ή κάποιον προορισμό εντός της πανεπιστημιούπολης και πατήσει το κουμπί Οδηγίες, η εφαρμογή υπολογίζει την πιο σύντομη διαδρομή και το αναδυόμενο παράθυρο αλλάζει και μπαίνει σε κατάσταση πλοήγησης (Σχήμα 4.46). Σε αυτή την κατάσταση εμφανίζονται, η εκτιμώμενη ώρα άφιξης καθώς και τα μέτρα που πρέπει να διανύσει ο χρήστης. Επίσης, η διαδρομή εμφανίζεται στον χάρτη με μπλε γραμμή. Επιπλέον, όπως μπορεί να παρατηρήσει ο χρήστης, το κουμπί με την αποτύπωση της διαδρομής φαίνεται στα δεξιά (fit route) και μπορεί να το πατήσει αν θέλει να δει ξανά την διαδρομή σε περίπτωση περιήγησης στο campus. Ο χρήστης για να ακυρώσει την προεπισκόπηση της διαδρομής μπορεί να το κάνει πατώντας το X στο αναδυόμενο παράθυρο. Όταν επιλεγτεί ο κατάλληλος προορισμός, ο χρήστης μπορεί να πατήσει Έναρξη για να ξεκινήσει η πλοήγηση.



Σχήμα 4.46: Προεπισκόπηση υπολογισμένης διαδρομής με εκτιμώμενη απόσταση και ώρα άφιξης

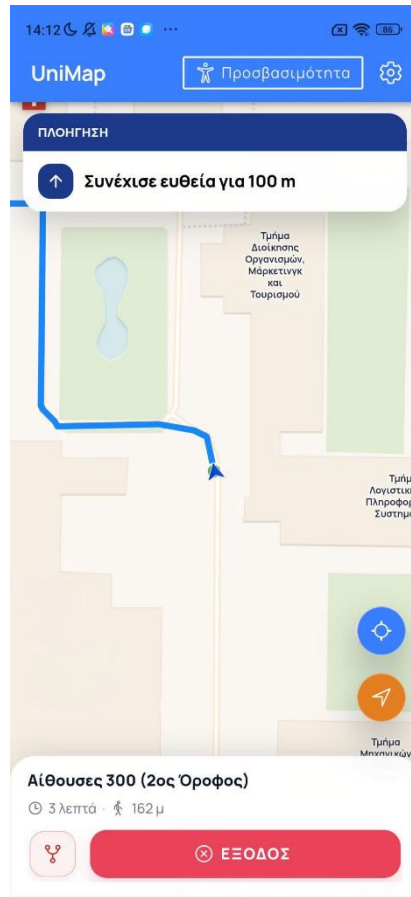
Ωστόσο, ο χρήστης μπορεί να «ανοίξει» την εφαρμογή και εκτός πανεπιστημιούπολης. Στην περίπτωση αυτή, εάν επιλέξει έναν προορισμό και πατήσει το κουμπί οδηγίες, εφαρμογή βγάζει προειδοποιητικό μήνυμα προς τον χρήστη, ότι οι οδηγίες θα εμφανιστούν από την στάση του λεωφορείου. Παρόλο που η διαδρομή μπορεί να σχεδιαστεί μέχρι τον προορισμό και να απεικονιστεί στην εφαρμογή, το κουμπί Έναρξη είναι απενεργοποιημένο και ο χρήστης δεν μπορεί πλέον να αλληλεπιδράσει και θα πρέπει να κλείσει την πλοήγηση με το κουμπί X που υπάρχει στο αναδυόμενο παράθυρο (Σχήμα 4.47).



Σχήμα 4.47: Προειδοποιητικό μήνυμα κατά την πλοήγηση εκτός campus

4.2.4.2.3 Ενεργή πλοήγηση

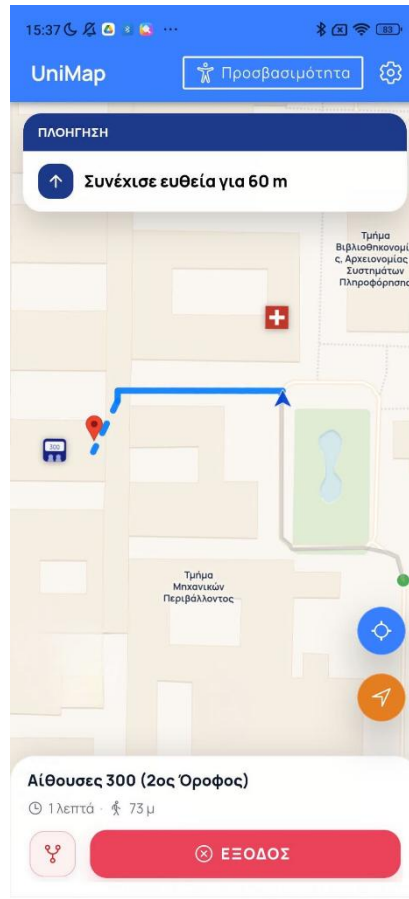
Αφού ο χρήστης πατήσει το κουμπί Έναρξη, η εφαρμογή μπαίνει σε κατάσταση ενεργής πλοήγησης (όπως φαίνεται στο Σχήμα 4.48). Το αναδυόμενο παράθυρο στο κάτω μέρος της εφαρμογής μπορεί να μην αλλάξει και να παραμένει το ίδιο ωστόσο, εμφανίζεται το κουμπί Έξοδος, που βοηθάει να ακυρωθεί η διαδρομή στην οποία έχει ξεκινήσει ο χρήστης να περιηγείται. Επίσης βλέπουμε ότι το κουμπί για να δει ο χρήστης την διαδρομή έχει προστεθεί στο αναδυόμενο παράθυρο. Η εναπομένουσα απόσταση και ο χρόνος, μειώνονται καθώς ο χρήστης κινείται εντός του χώρου της πανεπιστημιούπολης. Επιπλέον, για τον χρήστη έχουν δημιουργηθεί 2 κουμπιά τα οποία τον βοηθάνε στην πλοήγηση. Πιο συγκεκριμένα, έχουμε το κουμπί της επιστροφής στην θέση που βρίσκεται την εκάστοτε στιγμή (μπλε κουμπί) και ένα πορτοκαλί κουμπί. Το πορτοκαλί κουμπί δείχνει στον χρήστη ότι όσο προχωράει, «αφήνει πίσω» τον χάρτη και θα πρέπει μόνος του να σύρει για να πάει μπροστά. Σε περίπτωση που το πατήσει και γίνει πράσινο, η «κάμερα» ακολουθεί τον χρήστη και προχωράει μαζί με τον χρήστη. Η επιστροφή σε πορτοκαλί κουμπί (χειροκίνητης μεταφοράς) γίνεται μόλις ο χρήστης σύρει το δάχτυλό του στην οθόνη και κουνήσει την «κάμερα».



Σχήμα 4.48: Ενεργή πλοήγηση

Κατά την ενεργή πλοήγηση, ο χρήστης μπαίνει αυτόματα σε λειτουργία heading-up, δηλαδή περιστρέφεται με βάση την κατεύθυνση που κινείται ο χρήστης, έτσι ώστε αυτό που βρίσκεται μπροστά του να εμφανίζεται πάντα στο πάνω μέρος της οθόνης. Ο marker του χρήστη μετατρέπεται από κυκλικός σε βελάκι που δείχνει με βάση την κατεύθυνση του χρήστη που έχει. Παράλληλα, ο χάρτης κάνει αυτόματο zoom στην περιοχή πλοήγησης και η μετάβαση γίνεται ομαλά μέσω animation, εξασφαλίζοντας μια φυσική και άνετη εμπειρία για τον χρήστη. Επίσης, μπορεί να πλοηγηθεί μπροστά πίσω στην οθόνη ώστε να δει την εναπομένοντα διαδρομή που πρέπει να διανύσει.

Επιπλέον, στο πάνω μέρος της οθόνης η μπάρα αναζήτησης αντικαθίσταται από ένα παράθυρο πλοήγησης, το οποίο παρέχει στον χρήστη λεπτομερώς οδηγίες. Οι οδηγίες αυτές ενημερώνονται βήμα-βήμα και αλλάζουν καθώς προχωράει στην διαδρομή όπως απεικονίζονται στο Σχήμα 4.49.



Σχήμα 4.49: Λειτουργία heading-up με βήμα-βήμα οδηγίες πλοήγησης

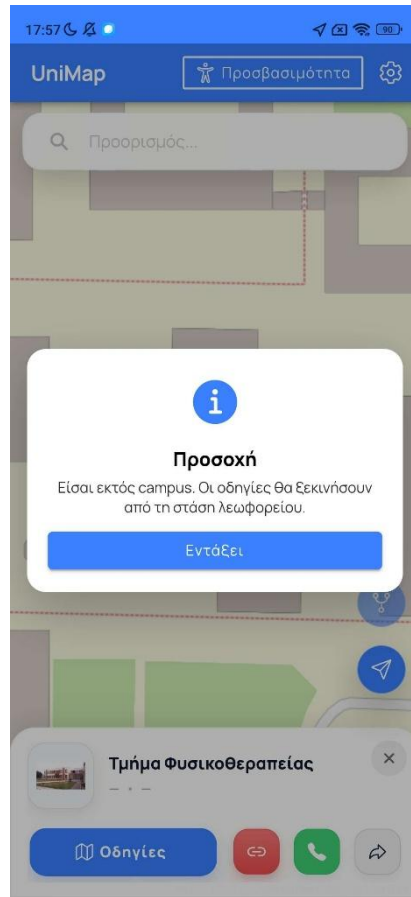
Κατά την διάρκεια της πλοήγησης, η διαδρομή εμφανίζεται στον χρήστη μέσω του χάρτη σε τρεις φάσεις:

- Με μπλε συνεχή γραμμή για το υπολειπόμενο τμήμα που πρέπει να διανύσει ο χρήστης
- Με γκρι γραμμή για το τμήμα που έχει ήδη διανυθεί από την στιγμή που ξεκίνησε
- Με μπλε διακεκομμένη γραμμή για το τελευταίο τμήμα πριν φτάσει ο χρήστης στην είσοδο του κτηρίου

Επίσης, εμφανίζεται στον χρήστη μια πράσινη βούλα, η οποία βοηθάει τον χρήστη να γνωρίζει από που ξεκίνησε.

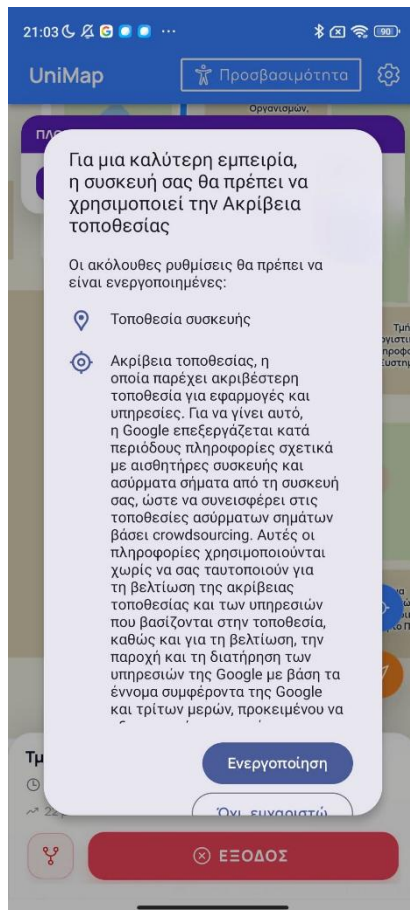
Σε περίπτωση που ο χρήστης αποφασίσει για οποιονδήποτε λόγο να αλλάξει πορεία και να πάει από κάποιο άλλο σημείο, η εφαρμογή υπολογίζει αυτόματα την νέα διαδρομή από το σημείο που βρίσκεται την εκάστοτε στιγμή.

Επιπλέον, η εφαρμογή έχει κάποιες βοηθητικές προειδοποιήσεις ώστε ο χρήστης να ενημερώνεται έγκαιρα για κάθε μια από τις περιπτώσεις που τον αφορούν. Πιο συγκεκριμένα, κατά την διάρκεια της πλοήγησης, όταν ο χρήστης κλείσει το internet (Wi-Fi ή δεδομένα) ή χάσει το σήμα της κινητής του τηλεφωνίας τότε εμφανίζεται το κατάλληλο μήνυμα στην μπάρα που δείχνει τις οδηγίες. Οι οδηγίες που ήδη έχουν δρομολογηθεί δεν σταματάνε αλλά σε περίπτωση που αλλάξει διαδρομή τότε δεν μπορεί να προχωρήσει. Όταν ξανά έχει σήμα, τότε αυτό το προειδοποιητικό μήνυμα εξαφανίζεται (Σχήμα 4.50).



Σχήμα 4.50: Προειδοποιητικό μήνυμα απώλειας σύνδεσης κατά την πλοήγηση

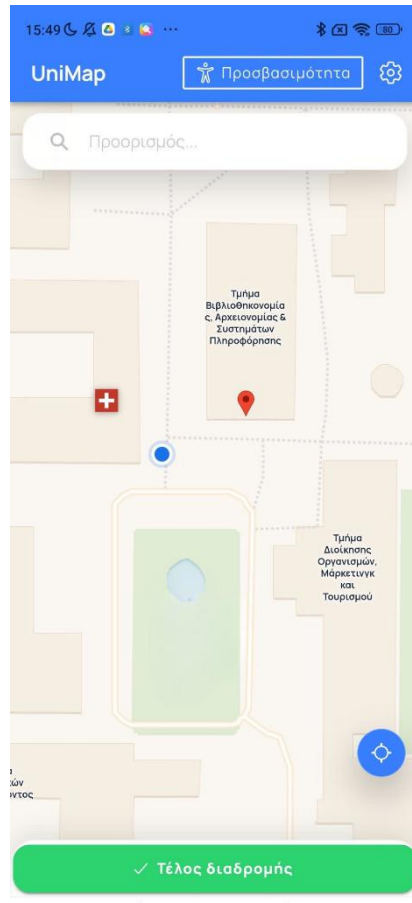
Σε άλλη περίπτωση που το GPS κλείσει απρόσμενα κατά την διάρκεια της πλοήγησης, τότε εμφανίζεται κατάλληλο προειδοποιητικό μήνυμα το οποίο δεν μπορεί ο χρήστης να προχωρήσει χωρίς τοποθεσία. Πιο συγκεκριμένα, μπορεί να επιλέξει είτε ενεργοποίηση τοποθεσίας ώστε να μπορεί να προχωρήσει (αυτό θα ξεκινήσει την διαδικασία φόρτωσης και αναζήτησης τοποθεσίας από την αρχή), είτε την επιλογή όχι ευχαριστώ που θα σε στείλει στο άνοιγμα της εφαρμογής για να αναζητήσει ξανά τοποθεσία.



Σχήμα 4.51: Προειδοποιητικό μήνυμα απώλεια σήματος GPS κατά την πλοήγηση

4.2.4.2.4 Άφιξη στον προορισμό

Η εφαρμογή παρακολουθεί συνεχώς την απόσταση του χρήστη από την είσοδο του κτηρίου-προορισμού κατά την διάρκεια της πλοήγησης. Όταν ο χρήστης φτάσει κοντά και σε απόσταση μικρότερη των 8 μέτρων, τότε η πλοήγηση σταματάει. Με τον τερματισμό της πλοήγησης, το αναδιδόμενο παράθυρο, από τα μέτρα, την ώρα άφιξης και το κουμπί της Εξόδου γίνεται ένα ενιαίο και μας εμφανίζει ότι ο χρήστης έχει φτάσει επιτυχώς στον προορισμό του. Όπως παρατηρούμε, στο Σχήμα 4.52, το παράθυρο πλοήγησης μας εμφανίζει και αυτό το κατάλληλο μήνυμα(κουμπί) ότι έχουμε ολοκληρώσει την διαδρομή και ότι έχουμε φτάσει στον προορισμό μας.



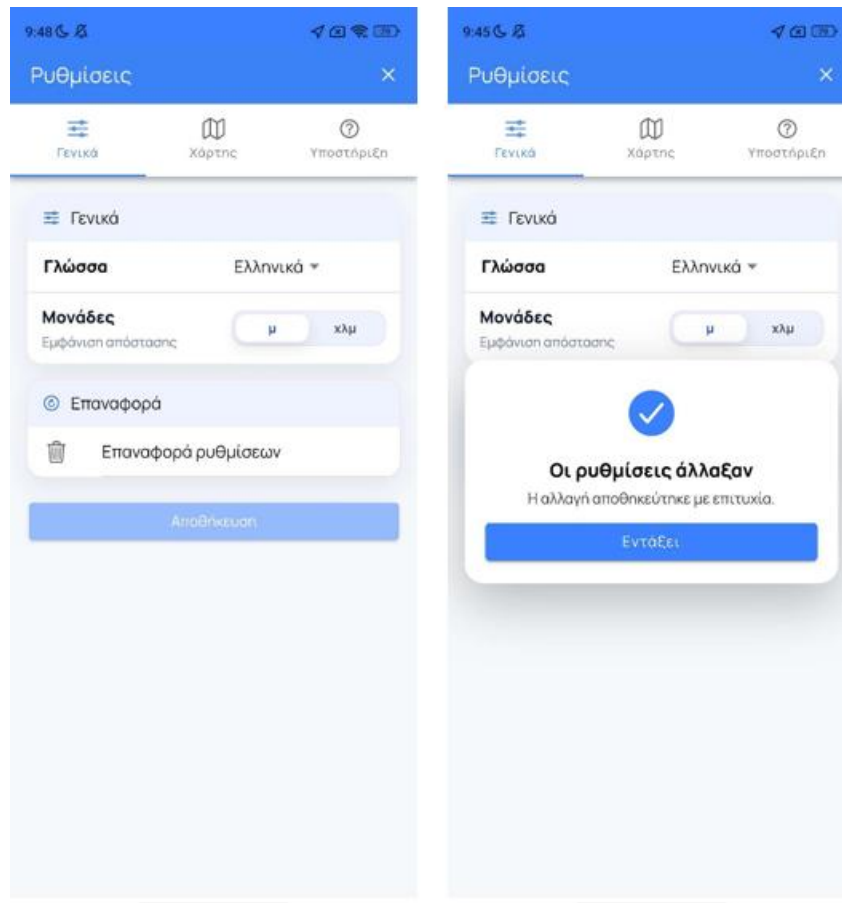
Σχήμα 4.52: Μήνυμα επιτυχούς άφιξης στον προορισμό μετά την ολοκλήρωση της πλοήγησης

4.2.4.3 Οθόνη 3: Ρυθμίσεις

Στις ρυθμίσεις μπορεί να μεταβεί ο χρήστης με το πάτημα του γραναζιού που βρίσκεται στην κύρια οθόνη. Οι ρυθμίσεις επιτρέπουν στον χρήστη να προσαρμόσει την εφαρμογή βάσει των προτιμήσεων. Οι ρυθμίσεις οργανώνεται σε τρεις καρτέλες: Γενικά, Χάρτης και Υποστήριξη οι οποίες αναλύονται παρακάτω λεπτομερώς. Για το κλείσιμο των ρυθμίσεων υπάρχει το κουμπί X πάνω δεξιά ώστε ο χρήστης να επιστρέψει στην προηγούμενη οθόνη, αυτή της κύριας οθόνης

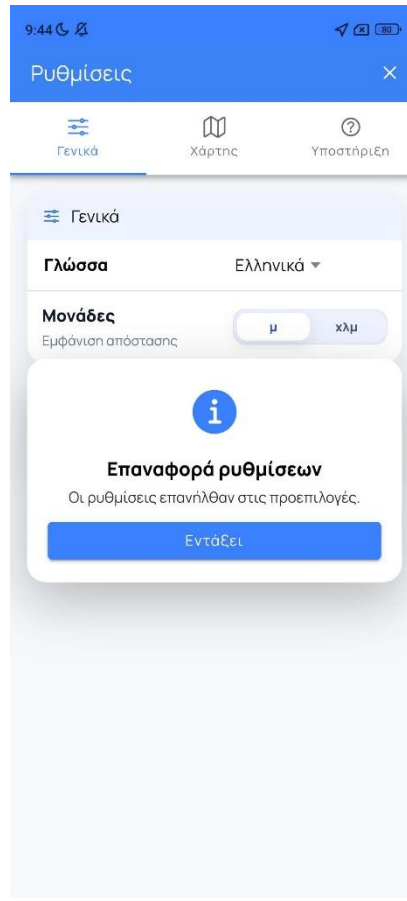
4.2.4.3.1 Καρτέλα Γενικά

Όταν ο χρήστης επιλέξει Ρυθμίσεις, ανοίγει η καρτέλα Γενικά. Σε αυτή την καρτέλα υπάρχουν γενικές πληροφορίες για που επιθυμεί ο χρήστης. Η εφαρμογή είναι δίγλωσση όπως έχει αναφερθεί παραπάνω, οπότε ο χρήστης μπορεί να αλλάξει την γλώσσα από αυτό το σημείο. Οι επιλογές είναι Ελληνικά (προεπιλογή) και Αγγλικά. Στην συνέχεια, υπάρχει η επιλογή το πως θα εμφανίζεται η απόσταση, την μονάδα μέτρησης. Οι επιλογές και εδώ είναι δυο, των μέτρων (μ) και την χιλιομέτρων (χλμ). Επιπλέον, μπορεί ο χρήστης να επιλέξει την επαναφορά όλων των ρυθμίσεων. Κατά την επαναφορά των ρυθμίσεων εμφανίζεται επίσης κατάλληλο μήνυμα και επανέρχεται η οθόνη στην κύρια (Σχήμα 4.53).



Σχήμα 4.53: Καρτέλα Γενικών Ρυθμίσεων και μήνυμα επιτυχούς αποθήκευσης αλλαγών

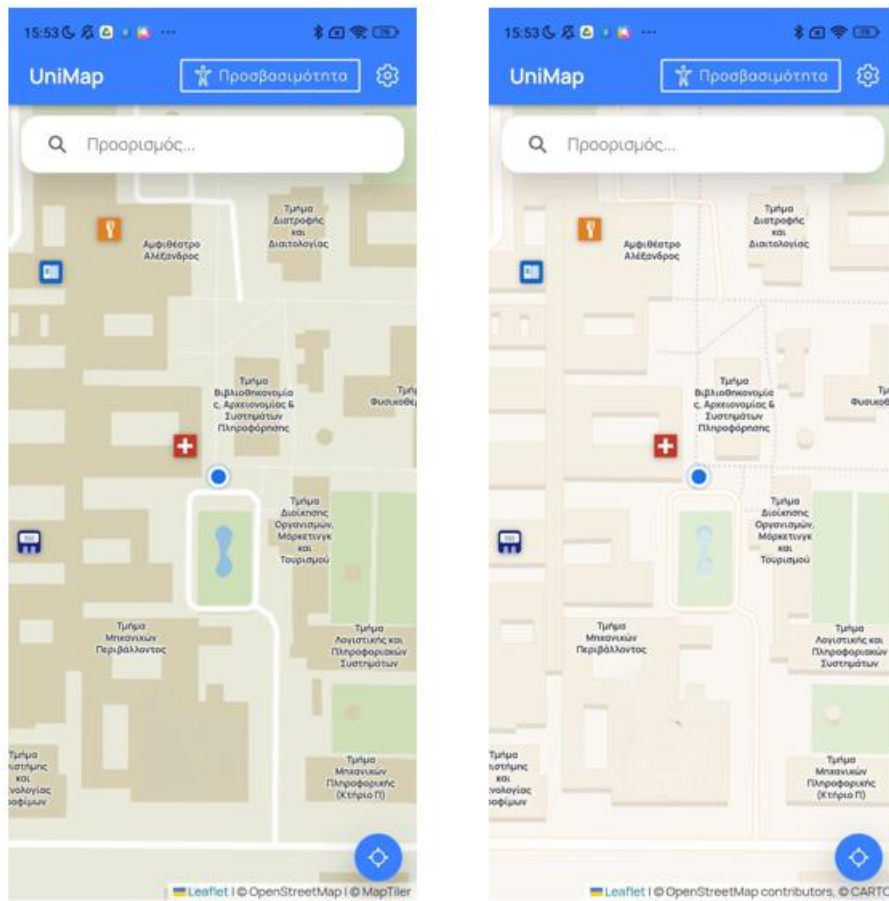
Στο τέλος της καρτέλας υπάρχει το κουμπί Αποθήκευση, που κάνει αποθήκευση των δεδομένων που άλλαξε ο χρήστης. Στην συγκεκριμένη περίπτωση, εμφανίζεται κατάλληλο μήνυμα με την αλλαγή που έγινε (όπως φαίνεται στο Σχήμα 4.54) και στην συνέχεια πραγματοποιούνται οι αλλαγές του χρήστη.



Σχήμα 4.54: Μήνυμα επιτυχούς αποθήκευσης αλλαγών στις Γενικές Ρυθμίσεις

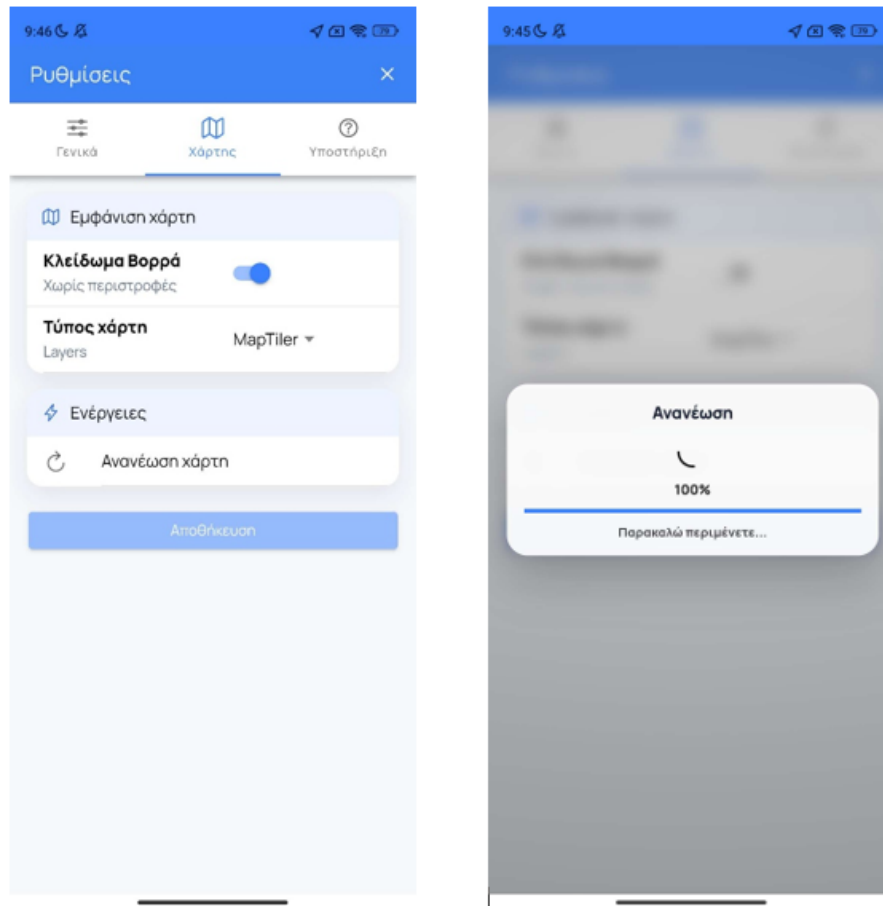
4.2.4.3.2 Καρτέλα Χάρτης

Στην συγκεκριμένη καρτέλα, είναι η δεύτερη κατά σειρά και υπάρχουν διάφορες λειτουργίες για την εμφάνιση του χάρτη. Για αρχή υπάρχει η επιλογή Κλείδωμα Βορρά που ουσιαστικά κρατάει τον χάρτη σταθερό με τον βορρά πάντα στο πάνω μέρος της οθόνης, ανεξάρτητα από την κατεύθυνση που κινείται ο χρήστης, δηλαδή απενεργοποιεί το heading-up mode. Στην συνέχεια μπορεί ο χρήστης να επιλέξει τον τύπο του χάρτη, Οι επιλογές σε αυτή την περίπτωση είναι δύο: (α) CardoDB και (β) MapTiler Basic (Σχήμα 4.55).



Σχήμα 4.55: Σύγκριση τύπων χάρτη: (α) CardoDB και (β) MapTiler Basic

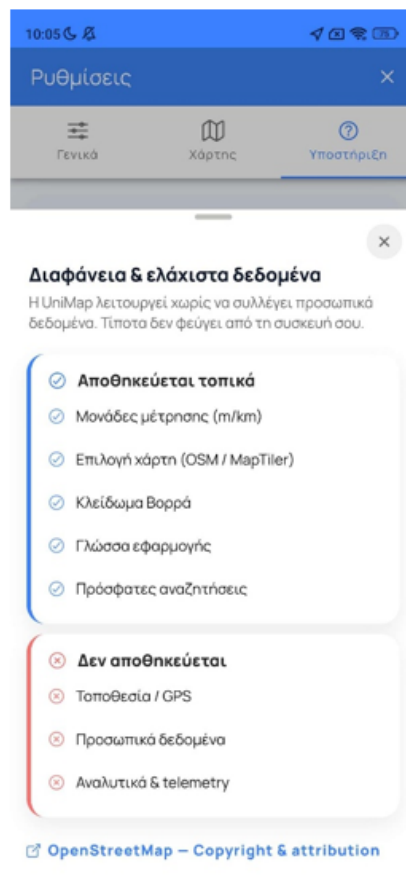
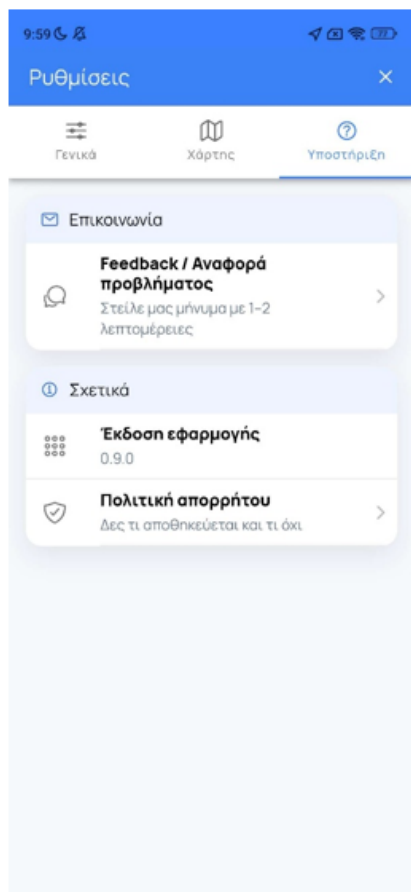
Στο κεντρικό μενού των ρυθμίσεων περιλαμβάνεται επίσης η δυνατότητα Κλειδώματος του Βορρά, η οποία διασφαλίζει ότι ο χάρτης θα παραμένει σταθερά προσανατολισμένος προς τον Βορρά, ανεξάρτητα από την κίνηση της συσκευής. Η επιλογή αυτή, σε συνδυασμό με τη λειτουργία Ανανέωσης (Σχήμα 4.56) για την επίλυση τυχόν σφαλμάτων απεικόνισης, προσφέρει στον χρήστη τον πλήρη έλεγχο της πλοήγησης. Τέλος, το κουμπί Αποθήκευση στο κάτω μέρος της καρτέλας οριστικοποιεί όλες τις αλλαγές, επιστρέφοντας τον χρήστη στην κύρια οθόνη με το περιβάλλον πλήρως ενημερωμένο



Σχήμα 4.56: Καρτέλα Ρυθμίσεων Χάρτη και παράθυρο φόρτωσης κατά την ανανέωση.

4.2.4.3.3 Καρτέλα Υποστήριξη

Η τελευταία καρτέλα του μενού προσφέρει στον χρήστη τη δυνατότητα άμεσης επικοινωνίας με τον δημιουργό, ανοίγοντας αυτόματα μια προκαθορισμένη φόρμα μέσω email. Στην ίδια ενότητα (Σχήμα 4.57), αναγράφεται η τρέχουσα έκδοση της εφαρμογής, ενώ παρέχεται και σύνδεσμος προς την πολιτική απορρήτου, διασφαλίζοντας τη διαφάνεια σχετικά με τη διαχείριση των δεδομένων



Σχήμα 4.57: Καρτέλα Υποστήριξης με επιλογές επικοινωνίας και πολιτική απορρήτου.

Κεφάλαιο 5ο: Συμπεράσματα και Μελλοντική Εργασία

5.1 Συμπεράσματα

Η παρούσα διπλωματική εργασία είχε ως στόχο τον σχεδιασμό και την υλοποίηση μιας εφαρμογής micro-navigation για το campus του ΔΙΠΑΕ στη Σίνδο, η οποία να καλύπτει ένα συγκεκριμένο κενό που δεν αντιμετωπίζουν οι γενικού σκοπού εφαρμογές πλοήγησης, όπως την ακριβή καθοδήγηση εντός πανεπιστημιακού χώρου με σεβασμό στις ανάγκες όλων των χρηστών συμπεριλαμβανομένων των ατόμων με κινητική αναπηρία. Η εφαρμογή UniMap υλοποιήθηκε επιτυχώς και λειτουργεί πλήρως στο campus του ΔΙΠΑΕ. Ο χρήστης μπορεί να αναζητήσει οποιοδήποτε τμήμα ή κτίριο, να λάβει τη βέλτιστη διαδρομή για να πλοηγηθεί με βάση την τρέχουσα θέση GPS του και να παρακολουθεί την πορεία του σε πραγματικό χρόνο πάνω στον χάρτη.

Ιδιαίτερα σημαντική είναι η υλοποίηση του συστήματος προσβασιμότητας για χρήστες αναπηρικού αμαξιδίου και γενικά στο ευρύτερο κοινό που έχει δυσλειτουργίες στην κίνηση. Η εφαρμογή είναι σε θέση να ψάξει, να βρει και να δώσει εναλλακτικές διαδρομές που αποφεύγουν πλήρως τα σκαλοπάτια, προσφέροντας μια απλή και έξυπνη λειτουργικότητα που απουσιάζει από τις εμπορικές εφαρμογές πλοήγησης για τον συγκεκριμένο χώρο. Η προσθήκη του κουμπιού της προσβασιμότητας δείχνει στην πράξη πώς η τεχνολογία μπορεί να γίνει ένας αληθινός σύμμαχος για όλους. Η εφαρμογή UniMap δεν είναι απλώς ένας ψηφιακός χάρτης, αλλά μια εφαρμογή που βοηθάει στην αυτονομία των φοιτητών και επισκεπτών με ή χωρίς κινητικές δυσκολίες, χωρίς τις δυσάρεστες εκπλήξεις που συναντά κανείς στον δρόμο του όπως σκαλοπάτια ή στενά μονοπάτια.

5.2 Μελλοντική Εργασία και Βελτιστοποίηση

Παρόλο που η εφαρμογή UniMap στην παρούσα της μορφή αποτελεί ένα ολοκληρωμένο και πλήρως λειτουργικό σύστημα πλοήγησης, η φύση του πανεπιστημιακού περιβάλλοντος προσφέρει σημαντικά περιθώρια για την ένταξη επιπρόσθετων λειτουργιών που θα επαυξήσουν την προστιθέμενη αξία της.

Πιο συγκεκριμένα, η εφαρμογή UniMap διαθέτει σημαντικές δυνατότητες περαιτέρω ανάπτυξης και βελτιστοποίησης. Μια από τις πιο σημαντικές επεκτάσεις που θα μπορούσε να γίνει θα ήταν η προσθήκη της φωνητικής καθοδήγησης, η οποία θα έδινε στον χρήστη την ευελιξία να προχωράει εντός της πανεπιστημιούπολης χωρίς να κοιτάει την οθόνη του βελτιώνοντας παράλληλα την προσβασιμότητα για άτομα με πρόβλημα όρασης. Ταυτόχρονα και στο ίδιο μήκος κύματος θα μπορούσε να επεκταθεί αυτή η λειτουργικότητα και για άτομα με χαμηλή όραση. Αυτό θα επιτευχθεί με την ενσωμάτωση των screen reader στο οποίο είναι ένα λογισμικό που «διαβάζει» τι συμβαίνει στην οθόνη και το μεταφέρει στον χρήστη μέσω συνθετικής ομιλίας.

Επιπλέον, η ενσωμάτωση πλήρους indoor navigation μέσω τεχνολογιών Bluetooth beacons ή Wi-Fi fingerprinting θα επέκτεινε την υπάρχουσα τεχνολογία καλύπτοντας μεγάλο όγκο χώρων όπως γραφεία, φοιτητικά στέκια για φοιτητές, συνελύσεις τμημάτων που η τωρινή εφαρμογή δεν μπορεί να τα προσφέρει. Σε επίπεδο περιεχομένου, η προσθήκη ωραρίων λειτουργίας για κάθε τμήμα, για κάθε υπηρεσία του πανεπιστημίου θα έκανε την εφαρμογή πιο πλήρη και πιο χρήσιμη στην καθημερινότητα των φοιτητών και των επισκεπτών.

Παράλληλα, μια κρίσιμη μελλοντική αναβάθμιση αφορά την εισαγωγή Push Notifications για την υποστήριξη πλοήγησης turn-by-turn. Με τη λειτουργία αυτή, ο χρήστης θα λαμβάνει οδηγίες κατεύθυνσης σε πραγματικό χρόνο απευθείας στην οθόνη κλειδώματος, γεγονός που καθιστά την περιήγηση στο campus πιο ασφαλή και άνεση, αφού δεν απαιτείται η συνεχής προσήλωση στην οθόνη

της εφαρμογής. Στο ίδιο πλαίσιο, η εισαγωγή δυνατοτήτων Crowdsourcing θα επιτρέψει τη δυναμική ενημέρωση του συστήματος από τους ίδιους τους χρήστες. Οι φοιτητές και το προσωπικό θα μπορούν να αναφέρουν σε πραγματικό χρόνο τυχόν έκτακτα εμπόδια, όπως έργα οδοποιίας ή προσωρινά μη λειτουργικές ράμπες, επιτρέποντας στον αλγόριθμο να αναδρομολογεί αυτόματα τους υπόλοιπους χρήστες από εναλλακτικές, προσβάσιμες διαδρομές.

Τέλος, ο μελλοντικός σχεδιασμός περιλαμβάνει την πλήρη επέκταση της εφαρμογής σε περιβάλλον iOS. Η διάθεση του UniMap στο οικοσύστημα της Apple θα εξασφαλίσει την καθολική πρόσβαση όλων των μελών της πανεπιστημιακής κοινότητας στις υπηρεσίες της, αξιοποιώντας παράλληλα εξειδικευμένα εργαλεία προσδιορισμού θέσης για ακόμα μεγαλύτερη ακρίβεια.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] JIIT Campus Navigator. (2025, August 7). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/document/11290263/references#references>
- [2] An event-driven university campus navigation system on android platform. (2014, February 12). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/document/7083850>
- [3] Mehta, H., Kanani, P., & Lande, P. (2019). Google Maps. International Journal of Computer Applications, 178(8), 41–46. <https://doi.org/10.5120/ijca2019918791>
- [4] Andreucci, G. (2013). Introduction to Apple Maps and the Map Kit Framework. In Apress eBooks (pp. 237–260). https://doi.org/10.1007/978-1-4302-4705-0_9
- [5] Samet, H., Fruin, B. C., & Nutanong, S. (2012). Diking it out at the smartphone mobile app mapping API corral. Proceedings of the First ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems, 41–48. <https://doi.org/10.1145/2442810.2442818>
- [6] Navigating MazeMap: Indoor human mobility, spatio-logical ties and future potential. (2014, March 1). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/abstract/document/6815215/>
- [7] Mobasheri, A., Deister, J., & Dieterich, H. (2017b). Wheelmap: the wheelchair accessibility crowdsourcing platform. Open Geospatial Data Software and Standards, 2(1). <https://doi.org/10.1186/s40965-017-0040-5>
- [8] Dijkstra’s and A-Star in Finding the Shortest Path: a Tutorial. (2020, July 1). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/abstract/document/9190342>
- [9] Google Maps [Online]. Available: <https://play.google.com/store/apps/details?id=com.google.android.apps.maps>
- [10] Apple Maps [Online]. Available: <https://apps.apple.com/us/app/apple-maps/id915056765>
- [11] MazeMap [Online]. Available: <https://play.google.com/store/apps/details?id=no.trt.ivecto.androidclient.mazemap&hl=el>
- [12] WheelMap [Online]. Available: <https://play.google.com/store/apps/details?id=org.wheelmap.android.online>
- [13] TypeScript Documentation [Online]. Available: <https://www.typescriptlang.org/docs/>
- [14] Node.js Documentation [Online]. Available: <https://nodejs.org/en/docs/>
- [15] Express Node.js Framework [Online]. Available: <https://expressjs.com/>
- [16] Angular Documentation [Online]. Available: <https://angular.io/docs>
- [17] Ionic Framework Documentation [Online]. Available: <https://ionicframework.com/docs>
- [18] Capacitor Documentation [Online]. Available: <https://capacitorjs.com/docs>
- [19] Leaflet - a JavaScript library for interactive maps [Online]. Available: <https://leafletjs.com/>
- [20] MapTiler Documentation [Online]. Available: <https://docs.maptiler.com/>

- [21] OpenStreetMap [Online]. Available: <https://www.openstreetmap.org/>
- [22] GeoJSON Specification [Online]. Available: <https://geojson.org/>
- [23] Render Documentation [Online]. Available: <https://render.com/docs>
- [24] Git [Online]. Available: <https://git-scm.com>
- [25] Git Documentation – Reference Manual [Online]. Available: <https://git-scm.com/docs>
- [26] GitHub [Online]. Available: <https://github.com>
- [27] Android Developers – Android Studio [Online]. Available: <https://developer.android.com/studio>
- [28] Open Source Routing Machine (OSRM) API [Online]. Available: <http://project-osrm.org/docs/v5.24.0/api/>
- [29] npm – A* [Online]. Available: <https://www.npmjs.com/package/a-star>
- [30] Angular – HttpClient [Online]. Available: <https://angular.dev/guide/http>
- [31] *Google Maps Platform Documentation*. (n.d.). Google for Developers. <https://developers.google.com/maps/documentation/navigation/android-sdk>
- [32] Apple. (2026, May 13). Apple Maps introduces new ways to explore the world. *Apple Newsroom*. <https://www.apple.com/newsroom/2024/09/apple-maps-introduces-new-ways-to-explore-the-world/>
- [33] Store, A. (2025, August 8). MazeMap. *App Store*. <https://appstore.no/apps/mazemap.html>
- [34] Wikipedia contributors. (2026, April 28). *Wheelmap.org*. Wikipedia. https://en.wikipedia.org/wiki/Wheelmap.org#/media/File:Screenshot_Wheelmap.org.png