

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σχεδιασμός και υλοποίηση εφαρμογής για online αγορά καρτών στάθμευσης και ανανέωση της ισχύος τους

Πάνος Κωνσταντίνος (**Α.Μ.:** 236332 / 1043805)

Επιβλέπουσα: Εύη Παπαϊωάννου, Επίκουρη Καθηγήτρια

Ιούνιος 2022

Πίνακας περιεχομένων

1 Εισαγωγή 4

1.1 Παρουσίαση προβλήματος και στόχος της διπλωματικής εργασίας 4

1.2 Δομή διπλωματικής εργασίας 5

2 Υπάρχουσες συναφείς εφαρμογές 7

2.1.1 myAthensPass 7

2.1.2 ParkPal 9

2.1.3 ParkAround 11

2.1.4 Flowbird 15

3 Ανάλυση απαιτήσεων και σχεδιασμός εφαρμογής 20

3.1 Ανάλυση απαιτήσεων 20

3.1.1 Οδηγοί 20

3.1.2 Ελεγκτές 21

3.1.3 Διαχειριστές 22

3.1.4 Διαγράμματα περιπτώσεων χρήσης 23

3.2 Σχεδιασμός εφαρμογής 24

3.2.1 Σχεδιασμός βάσης δεδομένων 25

4 Τεχνολογίες και λογισμικό ανάπτυξης 29

4.1 Progressive Web Application – PWA 29

4.1.1 Web App Manifest 30

4.1.2 Service worker 34

4.2 Τεχνολογίες και εργαλεία που χρησιμοποιήθηκαν 36

4.2.1 Node.js 36

4.2.2 Typescript 37

4.2.3 Express.js 37

4.2.4 React.js 38

4.2.5 TailwindCSS και CSS 38

4.2.6 Επιπλέον βιβλιοθήκες που χρησιμοποιήθηκαν 39

4.2.7 Visual Studio Code 40

5 Παρουσίαση εφαρμογής 41

5.1 Εγκατάσταση ενός Progressive Web Application 41

5.2 Παρουσίαση υποσυστημάτων εφαρμογής 44

5.2.1 Υποσύστημα οδηγού 44

5.2.2 Υποσύστημα ελεγκτή 51

5.2.3 Υποσύστημα διαχειριστή 54

6 Παρουσίαση περιπτώσεων χρήσης του συστήματος uPark 58
6.1 Αγορά χρόνου στάθμευσης σε κινητή συσκευή (Οδηγός) 58
6.2 Αγορά χρόνου στάθμευσης σε σταθερό υπολογιστή (Οδηγός) 63
7 Αξιολόγηση ποιότητας της εφαρμογής 69
8 Συμπεράσματα και μελλοντικές βελτιώσεις 73
9 Βιβλιογραφία 74

1 Εισαγωγή

Στο πλαίσιο της παρούσας διπλωματικής εργασίας, αναπτύχθηκε μια διαδικτυακή εφαρμογή αγοράς χρόνου ελεγχόμενης στάθμευσης, με την ονομασία **uPark**. Σε αυτό το εισαγωγικό κεφάλαιο, γίνεται μια γενική περιγραφή του προβλήματος που καλείται να επιλύσει η εφαρμογή αυτή και στη συνέχεια παρουσιάζεται η γενική δομή των κεφαλαίων της εργασίας, καθώς και τί αναλύεται σε κάθε ένα από αυτά.

1.1 Παρουσίαση προβλήματος και στόχος της διπλωματικής εργασίας

Την σημερινή εποχή, ο αριθμός των οχημάτων που κυκλοφορούν στους δρόμους της Ελλάδας αυξάνεται χρόνο με το χρόνο. Το γεγονός αυτό, γίνεται εμφανές αν ελέγξει κανείς τα στοιχεία που δημοσιεύει η Ελληνική Στατιστική Αρχή (ΕΛΣΤΑΤ) με τα τριμηνιαία δημοσιεύματα της. Πιο συγκεκριμένα, σύμφωνα με το δημοσίευμα για το τρίμηνο Ιανουάριος - Μάρτιος 2022 (https://www.statistics.gr/greece-in-figures) το 2019 καταγράφηκαν 5.406.551 επιβατηγά οχήματα, ενώ το 2020 καταγράφηκαν 5.492.176 επιβατηγά οχήματα στους δρόμους της χώρας, παρουσιάζοντας δηλαδή μια αύξηση της τάξης του 1,5%.

Όπως είναι λογικό, ένα μεγάλο μέρος των οχημάτων αυτών συναντάται στα μεγάλα αστικά κέντρα της χώρας, όπου συμβάλλουν σε διάφορα μείζονα προβλήματα όπως το κυκλοφοριακό και βεβαίως το περιβαλλοντικό. Ένα ακόμη πρόβλημα που δημιουργείται από αυτή τη συσσώρευση οχημάτων είναι φυσικά και η εύρεση θέσης στάθμευσης. Αρκετοί δήμοι της χώρας μας, στην προσπάθεια τους να αντιμετωπίσουν το πρόβλημα αυτό εφαρμόζουν κάποιου είδους σύστημα ελεγχόμενης στάθμευσης σε συγκεκριμένα σημεία της πόλης. Η στάθμευση στα σημεία αυτά απαιτεί την αγορά κάποιας κάρτας, συνήθως σε έντυπη μορφή, από συγκεκριμένα φυσικά σημεία πώλησης, όπως περίπτερα ή mini market. Ο συγκεκριμένος τρόπος σε μια εποχή που η τεχνολογία αποτελεί μέρος της καθημερινότητάς μας και η πρόσβαση στο διαδίκτυο είναι ευρέως διαδεδομένη μπορεί να χαρακτηριστεί ως απαρχαιωμένος και μη πρακτικός, καθώς επίσης και περιβαλλοντικά επιβαρυντικός μέσω της άσκοπης χρήσης πρώτων υλών για τη δημιουργία καρτών μιας χρήσεως.

Έχοντας όλα τα παραπάνω υπόψιν και θέτοντας ως στόχο την κάλυψη της ανάγκης για όχι μόνο ευκολότερη εύρεση θέσης στάθμευσης, αλλά και για άμεση αγορά χρόνου στάθμευσης, αναπτύχθηκε η διαδικτυακή εφαρμογή uPark, η οποία προσφέρει τη δυνατότητα στους

οδηγούς που την χρησιμοποιούν, να ενημερώνονται άμεσα για τη διαθεσιμότητα θέσεων και να πραγματοποιούν εύκολα και γρήγορα ηλεκτρονική αγορά χρόνου στάθμευσης για το όχημά τους μέσω της κινητής συσκευής ή του προσωπικού τους υπολογιστή. Η συγκεκριμένη εφαρμογή μπορεί να αποτελέσει μια ολοκληρωμένη λύση για την διαχείριση ενός συστήματος ελεγχόμενης στάθμευσης στο πλαίσιο μίας ολόκληρης πόλης, μιας πανεπιστημιούπολης ή ενός εμπορικού κέντρου. Για το λόγο αυτό, έχουν αναπτυχθεί επίσης υποσυστήματα για χρήση από ελεγκτές, προκειμένου να διαπιστώνεται εάν ένα σταθμευμένο όχημα έχει δικαίωμα στάθμευσης ή όχι, καθώς και για διαχειριστές, για την επίβλεψη και τη διαχείριση του συστήματος.

Τα διαφορετικά υποσυστήματα αναπτύχθηκαν με γνώμονα ότι θα χρησιμοποιηθούν τόσο από χρηστές κινητών συσκευών, όσο και από χρήστες σταθερών προσωπικών ηλεκτρονικών υπολογιστών. Για τον λόγο αυτό έγινε εκτενής χρήση mobile-friendly τεχνικών, δημιουργώντας μια διεπαφή χρήστη η οποία είναι απλή στη χρήση, παρόλα αυτά προσφέρει μια άμεση και διαδραστική εμπειρία, καθώς και την απαραίτητη λειτουργικότητα.

1.2 Δομή διπλωματικής εργασίας

Το κείμενο της διπλωματικής εργασίας ακολουθεί την εξής δομή.

Στο **δεύτερο κεφάλαιο**, θα παρουσιαστούν υπάρχουσες συναφείς εφαρμογές, οι οποίες λειτουργούν στη χώρα μας.

Στο **τρίτο κεφάλαιο**, θα γίνει μια ανάλυση των απαιτήσεων για τα τρία διαφορετικά υποσυστήματα που αναπτύχθηκαν στο πλαίσιο της διπλωματικής εργασίας, καθώς και αναφορά στο σχεδιασμό και την ανάπτυξη της βάσης δεδομένων της εφαρμογής uPark.

Έπειτα, στο **τέταρτο κεφάλαιο**, θα παρουσιαστούν οι βασικές έννοιες που αφορούν την διπλωματική εργασία και στη συνέχεια θα παρουσιαστούν οι διαφορετικές γλώσσες, βιβλιοθήκες και τα εργαλεία ανάπτυξης που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής.

Στο **πέμπτο κεφάλαιο**, αφού πρώτα παρουσιαστεί ο τρόπος εγκατάστασης της εφαρμογής σε κινητές έξυπνες συσκευές, θα γίνει μια παρουσίαση των οθονών των συστημάτων που αναπτύχθηκαν για κάθε έναν από τους τρεις διαφορετικούς τύπους χρηστών.

Συνεχίζοντας, στο **έκτο κεφάλαιο**, θα γίνει λεπτομερής παρουσίαση δύο βασικών περιπτώσεων χρήσης του συστήματος. Συγκεκριμένα θα παρουσιαστεί η διαδικασία αγοράς

χρόνου στάθμευσης από έναν οδηγό σε κινητή συσκευή και η διαδικασία αγοράς χρόνου στάθμευσης από έναν οδηγό όταν χρησιμοποιεί έναν σταθερό υπολογιστή

Στο **έβδομο κεφάλαιο**, παρουσιάζονται τα αποτελέσματα της αξιολόγησης της ποιότητας της εφαρμογής, η οποία πραγματοποιήθηκε με χρήση του εργαλείου ανοιχτού κώδικα Lighthouse, των Chrome Developer Tools.

Τέλος, στο **όγδοο κεφάλαιο**, καταλήγουμε στα συμπεράσματα που προέκυψαν κατά τη διαδικασία ανάπτυξης της εφαρμογής uPark και προτείνουμε μελλοντικές προσθήκες και βελτιώσεις.

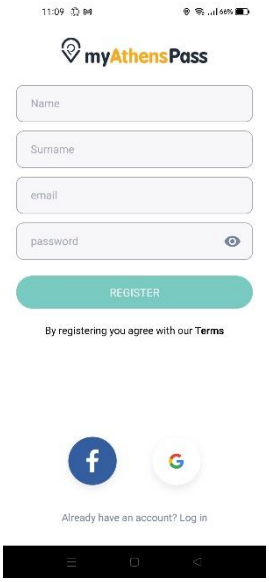
2 Υπάρχουσες συναφείς εφαρμογές

Σε αυτό το κεφάλαιο, γίνεται μια συνοπτική παρουσίαση υπαρχουσών εναλλακτικών εφαρμογών και υπηρεσιών, οι οποίες παρουσιάζουν ορισμένη συνάφεια με την εφαρμογή uPark, που αναπτύχθηκε στο πλαίσιο αυτής της διπλωματικής εργασίας.

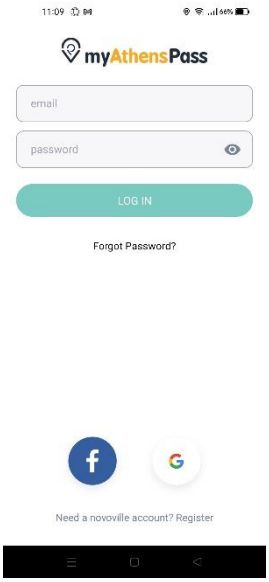
2.1.1 myAthensPass

Το myAthensPass είναι η εφαρμογή για κινητές συσκευές του συστήματος PARKinATHENS (http://parkinathens.gr), η οποία αποτελεί μέρος του Συστήματος Ελεγχόμενης Στάθμευσης (Σ.Ε.Σ.) του Δήμου Αθηναίων. Η εφαρμογή είναι διαθέσιμη για συσκευές Android και iOS και δίνει τη δυνατότητα στους χρήστες της να αγοράζουν και να χρησιμοποιούν χρόνο ελεγχόμενης στάθμευσης με άμεσο τρόπο μέσω της κινητής τους συσκευής, εξαλείφοντας έτσι την ανάγκη για την αγορά ξυστών καρτών στάθμευσης.

Οι βασικότερες υπηρεσίες που προσφέρει η εφαρμογή αυτή, είναι η δυνατότητα ηλεκτρονικής αγοράς χρόνου στάθμευσης με τρόπο άμεσο ή μέσω προπληρωμής για μελλοντική χρήση. Επιπλέον οι οδηγοί ενημερώνονται με push notification 10 λεπτά πριν τη λήξη του χρόνου τους, έχουν όμως τη δυνατότητα επέκτασης του υπολειπόμενου χρόνου.

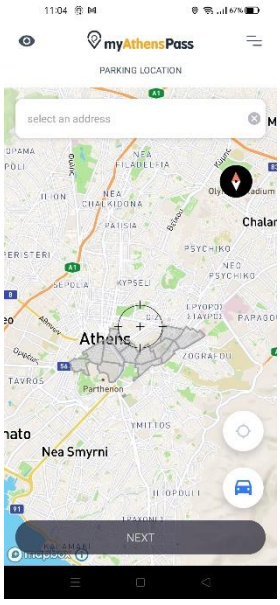


Εικόνα 2.1 Φόρμα εγγραφής νέου χρήστη myAthensPass

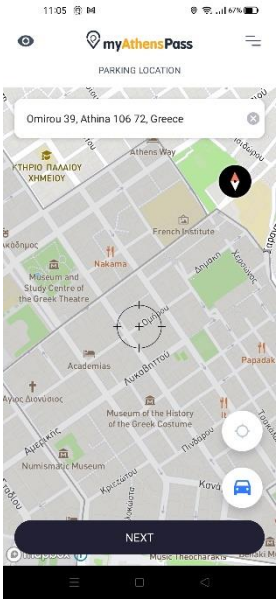


Εικόνα 2.2 Φόρμα εισόδου χρήστη myAthensPass

Κατά το άνοιγμα της εφαρμογής ο χρήστης πραγματοποιεί εγγραφή (εάν δεν διαθέτει λογαριασμό) (Εικόνα 2.1) και είσοδο (Εικόνα 2.2). Αφού ο χρήστης συνδεθεί στο σύστημα, στον χάρτη που του εμφανίζεται (Εικόνα 2.3) βρίσκει την διεύθυνση στην οποία έχει σταθμεύσει (Εικόνα 2.4).



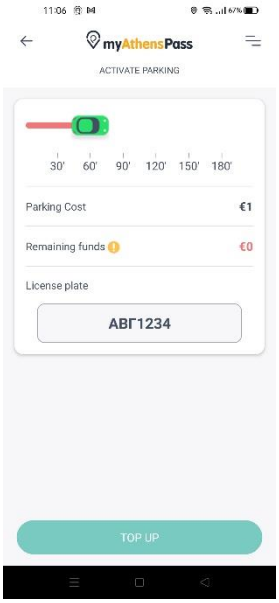
Εικόνα 2.3 Αρχική οθόνη myAthensPass



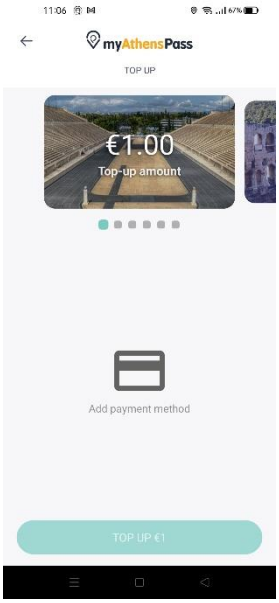
Εικόνα 2.4 Εύρεση διεύθυνσης στάθμευσης οχήματος

Στη συνέχεια (Εικόνα 2.5) επιλέγει για πόση χρονική διάρκεια θα σταθμεύσει και εισάγει τον αριθμό πινακίδας του οχήματός του. Εάν το υπολειπόμενο ποσό που διαθέτει ο χρήστης δεν

επαρκεί, τότε οδηγείται στην οθόνη αγοράς χρόνου (Εικόνα 2.6) στην οποία καλείται να διαλέξει τρόπο πληρωμής και να εισάγει τα στοιχεία που απαιτούνται.



Εικόνα 2.5 Οθόνη εισαγωγής στοιχείων στάθμευσης

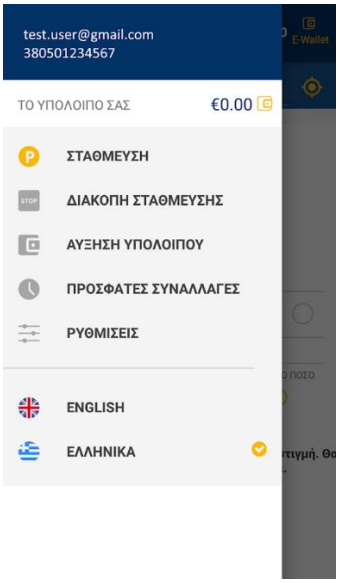


Εικόνα 2.6 Οθόνη αγοράς χρόνου

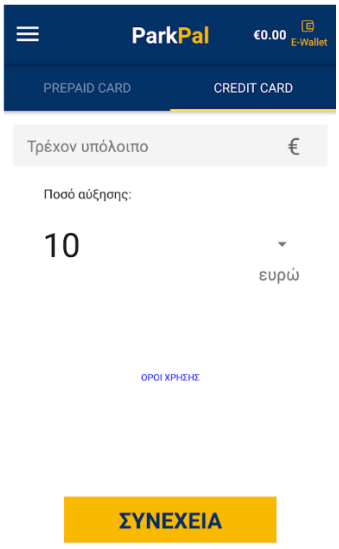
Παρά τις διευκολύνσεις που προσφέρει η εφαρμογή αυτή, κατά τη χρήση της, οι οδηγοί δεν έχουν καμία ενημέρωση ως προς τη διαθεσιμότητα θέσεων στα σημεία στάθμευσης, κάτι που λήφθηκε υπόψιν και αναπτύχθηκε για την εφαρμογή uPark.

2.1.2 ParkPal

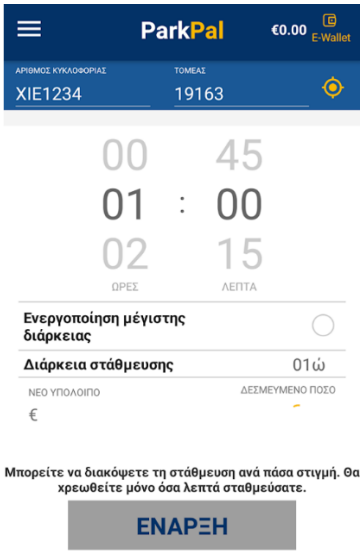
Η εφαρμογή ParkPal αποτελεί μέρος του Συστήματος Ελεγχόμενης Στάθμευσης THESi (https://thesi.gr), το οποίο εφαρμόζεται στη Θεσσαλονίκη και είναι διαθέσιμη για συσκευές Android και iOS. Μέσω της εφαρμογής αυτής, ο χρήστης χρεώνεται για τη στάθμευση του οχήματός του ανά λεπτό με αναλογική χρέωση, μπορεί να ελέγχει το ιστορικό των σταθμεύσεων ενώ ακόμη μπορεί να αγοράσει χρόνο στάθμευσης για πολλαπλά οχήματα. Ο χρήστης μπορεί να ανανεώσει το χρηματικό του υπόλοιπο είτε μέσω προπληρωμένης κάρτας ParkPal από διάφορα σημεία πώλησης είτε μέσω της πιστωτικής ή χρεωστικής του κάρτας.



Εικόνα 2.7 Κεντρικό μενού εφαρμογής Parkpal



Εικόνα 2.8 Οθόνη αύξησης υπολοίπου



Εικόνα 2.9 Οθόνη εισαγωγής στοιχείων στάθμευσης

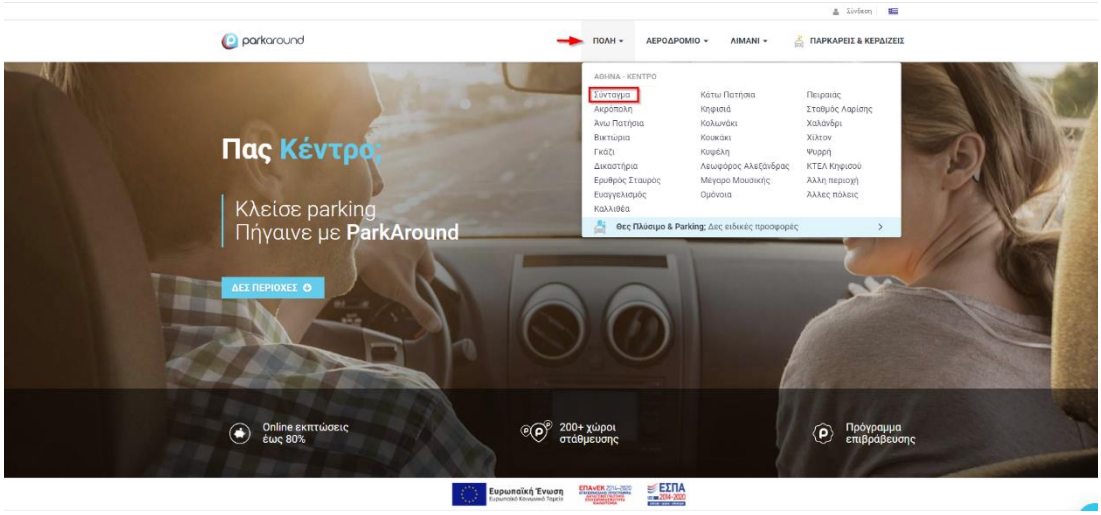
Μέσω του κεντρικού μενού (Εικόνα 2.7) και επιλέγοντας «ΑΥΞΗΣΗ ΥΠΟΛΟΙΠΟΥ» ο χρήστης μπορεί να προσθέσει χρήματα στο λογαριασμό του είτε μέσω προπληρωμένης κάρτας ParkPal είτε μέσω πιστωτικής ή χρεωστικής κάρτας, όπως φαίνεται στην Εικόνα 2.8. Έπειτα ξανά μέσω του κεντρικού μενού και επιλέγοντας «ΣΤΑΘΜΕΥΣΗ» ο χρήστης μεταβαίνει στην οθόνη στάθμευσης (Εικόνα 2.9).

Ένα μειονέκτημα της εφαρμογής ParkPal, είναι η αναγκαστική παροχή του αριθμού τηλεφώνου για την εγγραφή ενός νέου χρήστη, το οποίο αποτελεί μια πληροφορία που

μερικοί χρήστες ενδεχομένως επιθυμούν να παρέχουν σε μια εφαρμογή, λειτουργώντας έτσι αποτρεπτικά στην προσέλκυση μιας μερίδας χρηστών.

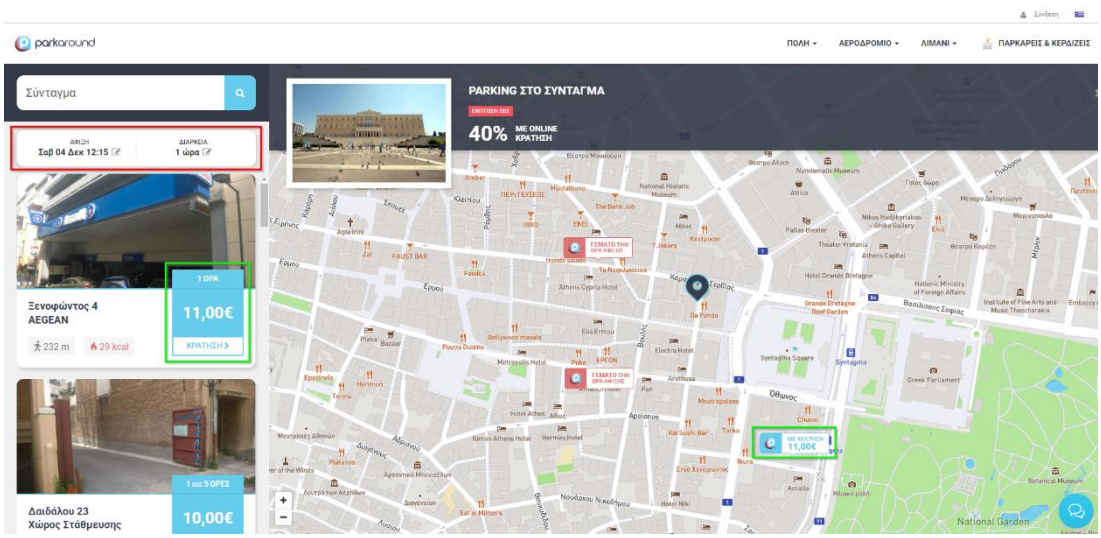
2.1.3 ParkAround

Σε αντίθεση με τις δύο προηγούμενες εφαρμογές, η υπηρεσία ParkAround (https://www.parkaround.gr/), δεν αποτελεί μέρος Συστήματος Ελεγχόμενης Στάθμευσης κάποιου δήμου, καθώς παρέχει τη δυνατότητα για αναζήτηση και κράτηση θέσης στάθμευσης σε ιδιωτικούς χώρους στάθμευσης. Ένας χρήστης μπορεί να έχει πρόσβαση στην υπηρεσία αυτή είτε μέσω φυλλομετρητή, είτε μέσω των διαθέσιμων εφαρμογών για συσκευές Android και iOS. Για την χρήση της υπηρεσίας αυτής δεν απαιτείται ο χρήστης να πραγματοποιήσει κάποια εγγραφή. Ο χρήστης επισκέπτεται από τον φυλλομετρητή του τον ιστότοπο https://www.parkaround.gr/ επιλέγει την περιοχή στην οποία ενδιαφέρεται να σταθμεύσει το όχημά του, όπως φαίνεται στην Εικόνα 2.10, και στη συνέχεια εμφανίζονται σε χάρτη οι διαθέσιμοι χώροι στάθμευσης στην περιοχή αυτή (Εικόνα 2.11).

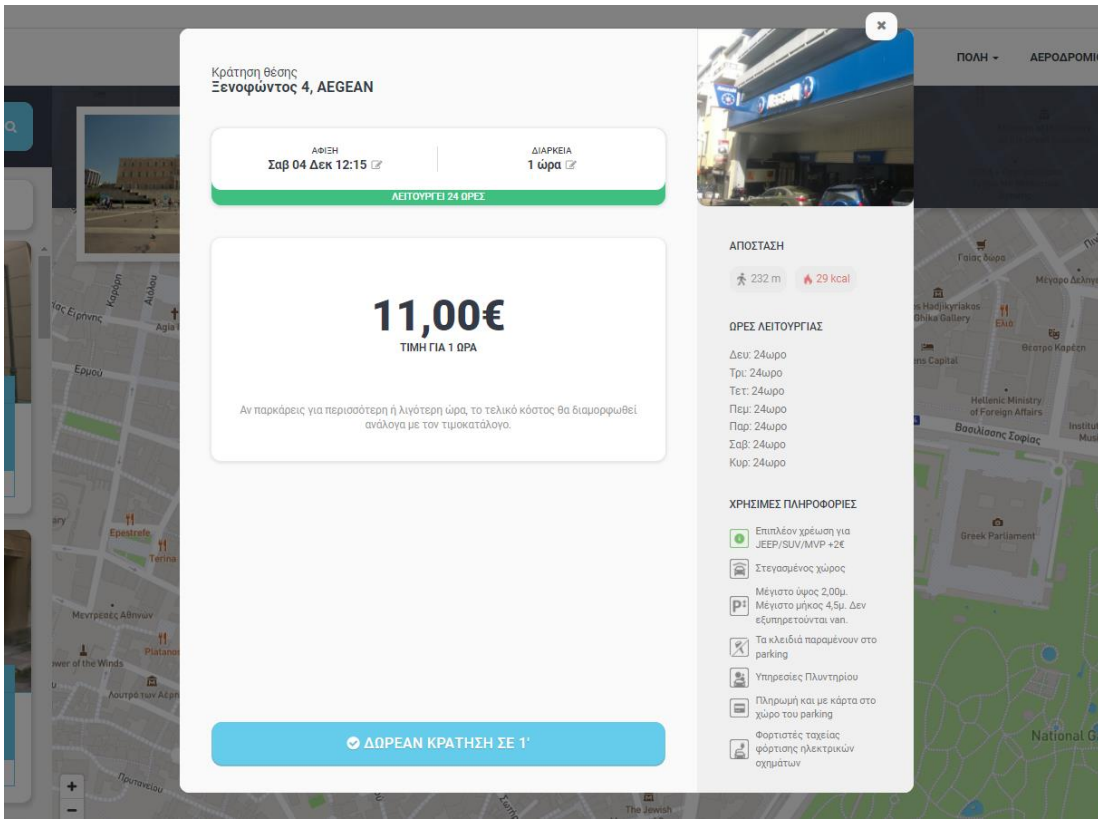


Εικόνα 2.10 Αρχική σελίδα της ιστοσελίδας ParkAround (https://www.parkaround.gr)

Στο σημείο αυτό ο χρήστης μπορεί να επεξεργαστεί τα στοιχεία της στάθμευσης (ώρα-ημερομηνία άφιξης και διάρκεια) και να επιλέξει έναν από τους διαθέσιμους χώρους στάθμευσης, όπως επισημαίνεται με κόκκινο και πράσινο πλαίσιο αντίστοιχα, στην Εικόνα 2.11.



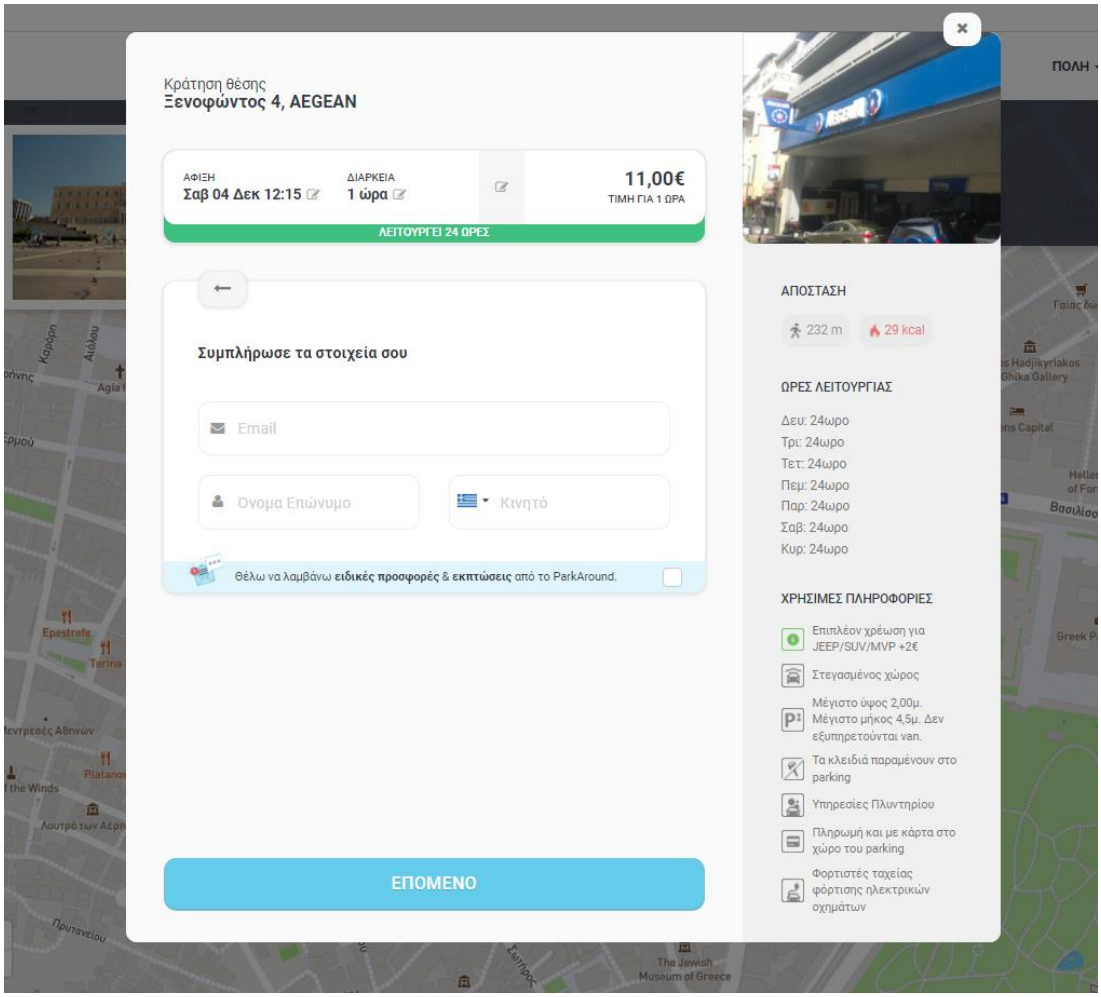
Εικόνα 2.11 Εμφάνιση διαθέσιμων χώρων στάθμευσης



Εικόνα 2.12 Προεπισκόπηση λεπτομερειών στάθμευσης

Επιλέγοντας τον χώρο στάθμευσης της προτίμησής του, εμφανίζεται αναδυόμενο παράθυρο με μια προεπισκόπηση των λεπτομερειών της στάθμευσης (Εικόνα 2.12) και επιλέγοντας να

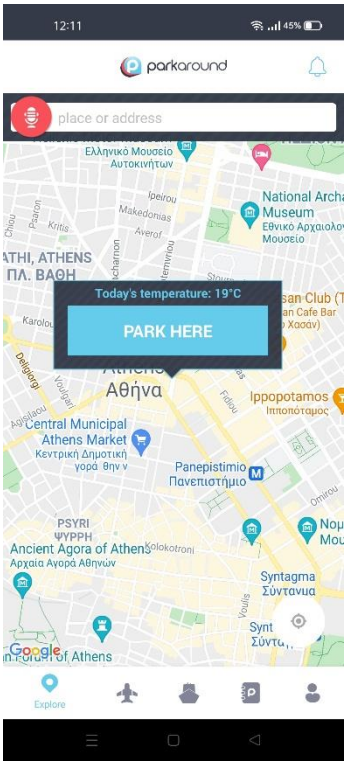
προχωρήσει σε κράτηση θέσης εμφανίζεται ένα νέο παράθυρο για την εισαγωγή των προσωπικών του στοιχείων (Εικόνα 2.13) και η διαδικασία ολοκληρώνεται.



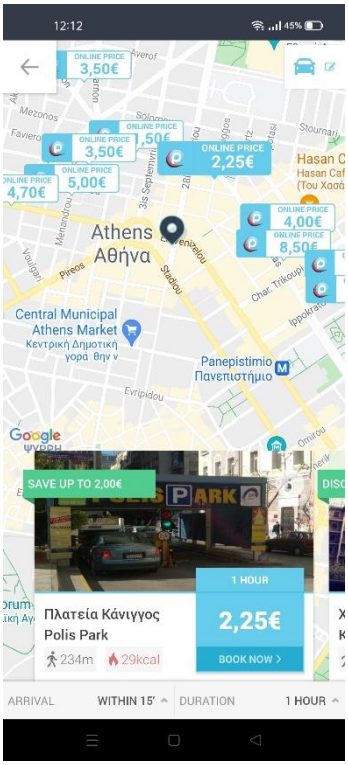
Εικόνα 2.13 Εισαγωγή στοιχείων κράτησης

Στην έκδοση της εφαρμογής για συσκευές Android, στον χρήστη εμφανίζεται αρχικά ένας χάρτης (Εικόνα 2.14). Επιλέγοντας «PARK HERE» η εφαρμογή αναζητά κοντινούς χώρους στάθμευσης, οι τοποθεσίες των οποίων εμφανίζονται στον χάρτη (Εικόνα 2.15).

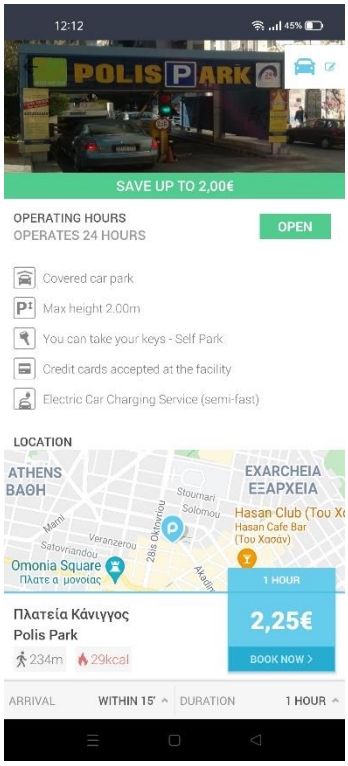
Έπειτα αφού ο χρήστης επιλέξει έναν από τους διαθέσιμους χώρους στάθμευσης, εμφανίζεται ένα παράθυρο με επιπρόσθετες πληροφορίες για τον χώρο αυτό (Εικόνα 2.16). Τέλος επιλέγοντας «BOOK NOW» ο χρήστης καλείται να εισάγει τα προσωπικά του στοιχεία και η διαδικασία κράτησης ολοκληρώνεται.



Εικόνα 2.14 Αρχική οθόνη εφαρμογής ParkAround



Εικόνα 2.15 Αποτελέσματα αναζήτησης



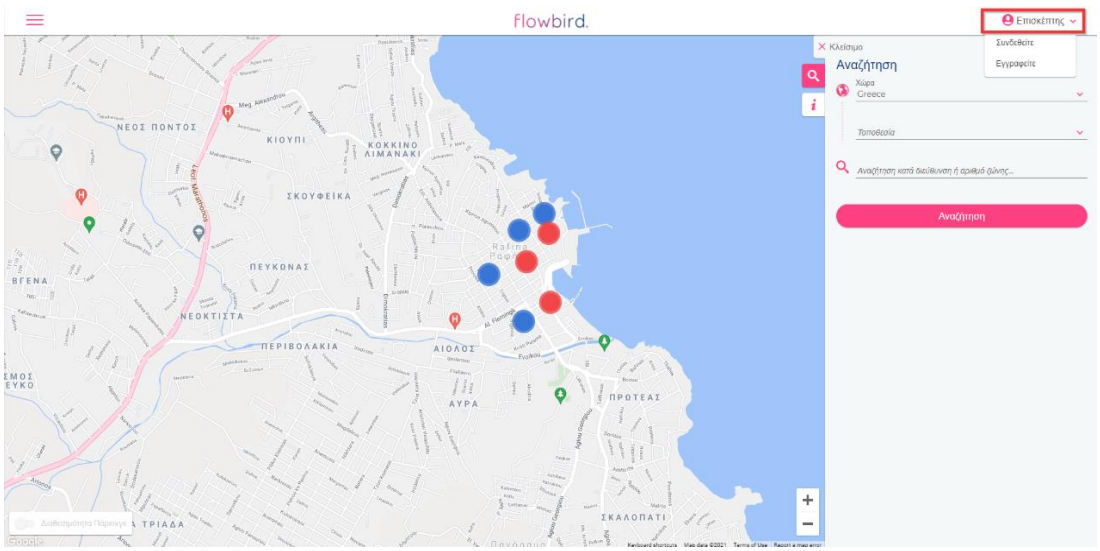
Εικόνα 2.16 Οθόνη επιπρόσθετων πληροφοριών

2.1.4 Flowbird

Η εφαρμογή Flowbird (https://flowbirdapp.com), είναι ακόμη μια υπηρεσία η οποία δίνει τη δυνατότητα στους χρήστες της να πληρώνουν για την στάθμευση των οχημάτων τους. Η υπηρεσία αυτή στην χώρα μας χρησιμοποιείται μόνο στο δήμο Ραφήνας και αποτελεί μέρος του Συστήματος Ελεγχόμενης Στάθμευσής του. Ο χρήστης έχει πρόσβαση σε αυτή την εφαρμογή είτε μέσω της έξυπνης συσκευής του Android ή iOS, είτε μέσω του διαδικτύου από κάποιον φυλλομετρητή.

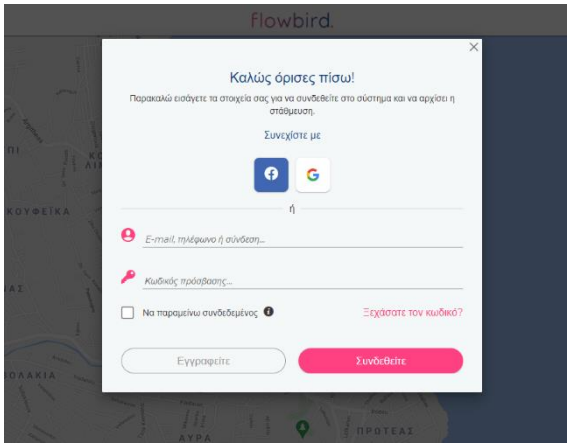
Οι βασικότερες υπηρεσίες που προσφέρει η εφαρμογή Flowbird, είναι η ενημέρωση των χρηστών σχετικά με τη πιθανή διαθεσιμότητα θέσεων στους δρόμους, καθώς και η γρήγορη και ασφαλής αγορά χρόνου στάθμευσης, επίσης οι χρήστες λαμβάνουν ειδοποιήσεις για την λήξη χρόνου της στάθμευσής τους, έχοντας όμως την δυνατότητα για απομακρυσμένη επέκταση του χρόνου αυτού. Πρέπει να σημειωθεί, ότι η χρήση της υπηρεσίας αυτής, αποδείχτηκε η πιο εύχρηστη και φιλική προς το χρήστη, σε σχέση με τις υπόλοιπες εναλλακτικές εφαρμογές που μελετήθηκαν και παρουσιάστηκαν.

Ένας χρήστης όταν επισκέπτεται μέσω του φυλλομετρητή του τη διεύθυνση https://my.flowbirdapp.com οδηγείται στη σελίδα της Εικόνα 2.17.

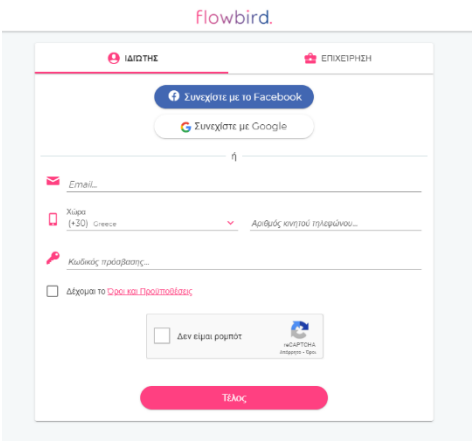


Εικόνα 2.17 Οθόνη καλωσορίσματος της διεύθυνσης https://my.flowbirdapp.com

Καθώς η υπηρεσία είναι διαθέσιμη μόνο για εγγεγραμμένους χρήστες, επιλέγοντας «Επισκέπτης» στο σημείο που υποδεικνύεται με κόκκινο πλαίσιο στην Εικόνα 2.17, ο χρήστης καλείται είτε να συνδεθεί σε υπάρχοντα λογαριασμό (Εικόνα 2.18), είτε να δημιουργήσει ένα νέο λογαριασμό (Εικόνα 2.19).

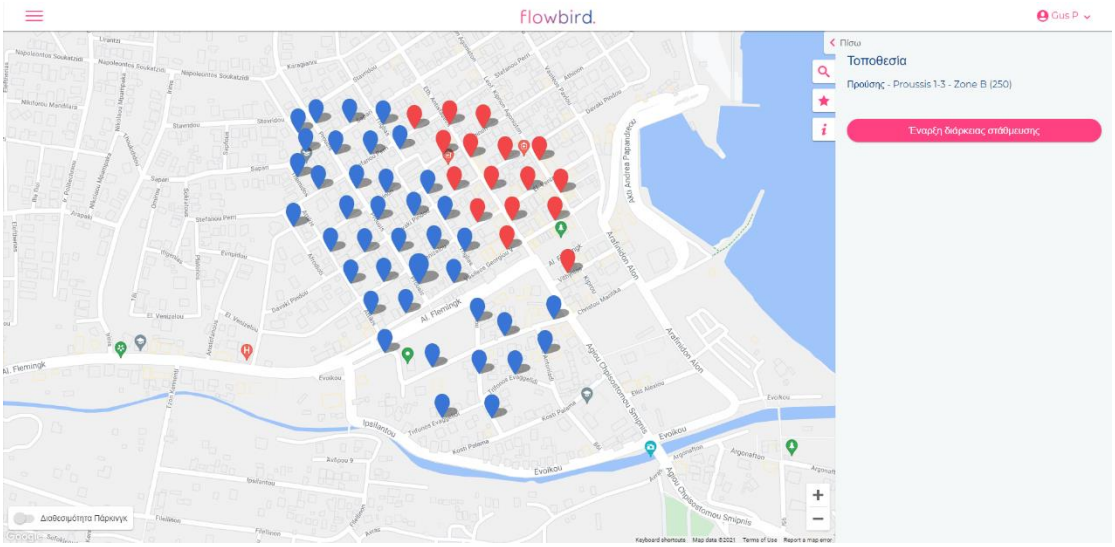


Εικόνα 2.18 Φόρμα σύνδεσης χρήστη



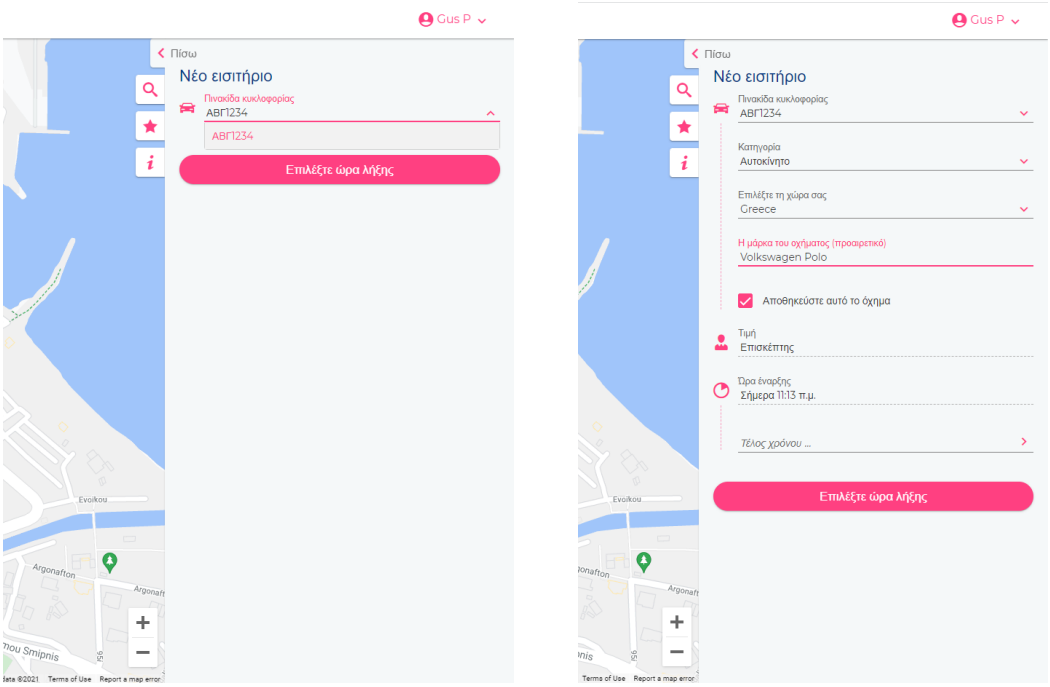
Εικόνα 2.19 Φόρμα εγγραφής νέου χρήστη

Όταν ο χρήστης συνδεθεί στον λογαριασμό του, στην οθόνη που εμφανίζεται(Εικόνα 2.20), επιλέγει την τοποθεσία στάθμευσής του μέσω των διαθέσιμων σημείων στο χάρτη και έπειτα επιλέγει «Έναρξη διάρκειας στάθμευσης».

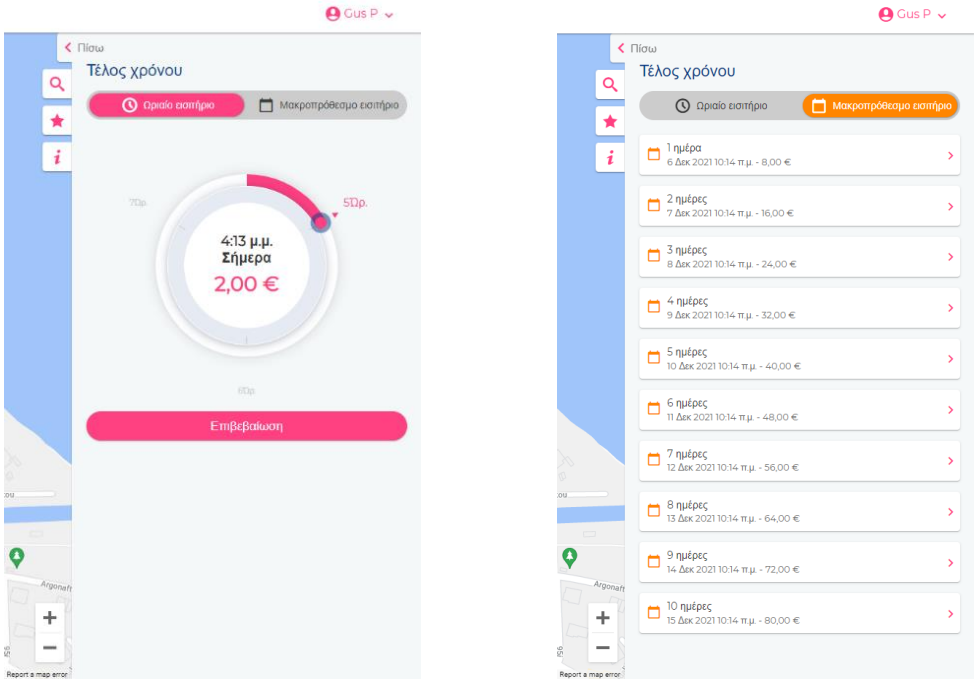


Εικόνα 2.20 Επιλογή σημείου στάθμευσης

Στο σημείο αυτό ο χρήστης καλείται να εισάγει πληροφορίες για το όχημα (Εικόνα 2.21), όπως τον αριθμό πινακίδας, και έπειτα εμφανίζεται «Επιλέξτε ώρα λήξης». Όπως φαίνεται στην Εικόνα 2.22, ο χρήστης μπορεί να επιλέξει είτε ωριαίο είτε μακροπρόθεσμο εισιτήριο για την στάθμευση του οχήματός του και έπειτα καλείται να καταβάλει το απαραίτητο ποσό μέσω ηλεκτρονικής πληρωμής, ολοκληρώνοντας έτσι τη διαδικασία έκδοσης του εισιτηρίου.

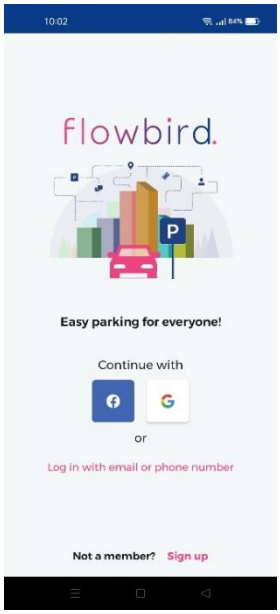


Εικόνα 2.21 Εισαγωγή πληροφοριών οχήματος

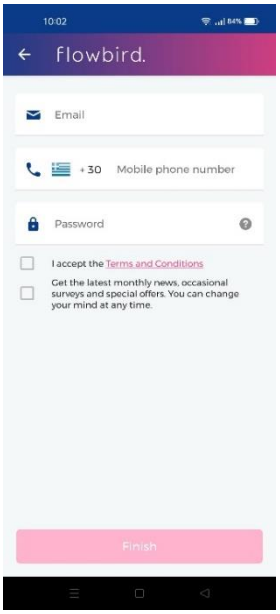


Εικόνα 2.22 Επιλογή τύπου και διάρκειας εισιτηρίου

Στην έκδοση της εφαρμογής για συσκευές Android, ο χρήστης καλείται αρχικά να συνδεθεί στο λογαριασμό του εάν είναι εγγεγραμμένος (Εικόνα 2.23) ή να δημιουργήσει νέο λογαριασμό (Εικόνα 2.24).

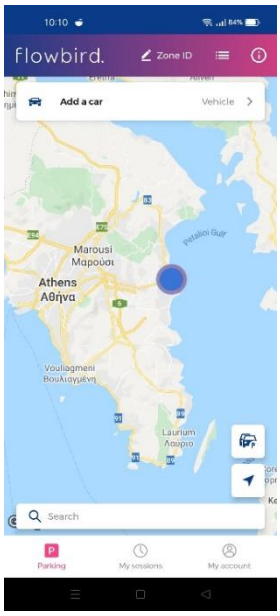


Εικόνα 2.23 Σύνδεση σε υπάρχων λογαριασμό

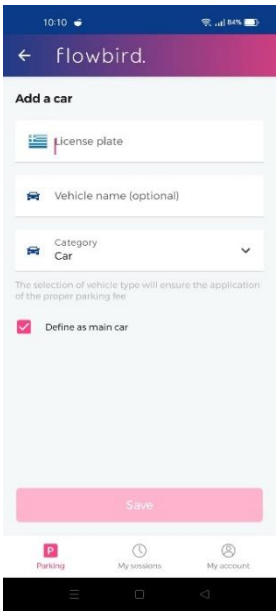


Εικόνα 2.24 Δημιουργία νέου λογαριασμού

Όταν ο χρήστης συνδεθεί στο λογαριασμό του, εμφανίζεται ένας χάρτης με σημεία στα οποία λειτουργεί η υπηρεσία (Εικόνα 2.25). Επιλέγοντας την ένδειξη «Add a car» στο επάνω μέρος της οθόνης ο χρήστης μπορεί να εισάγει πληροφορίες σχετικές με το/τα όχημα/οχήματά του στην φόρμα εισόδου που εμφανίζεται (Εικόνα 2.26).

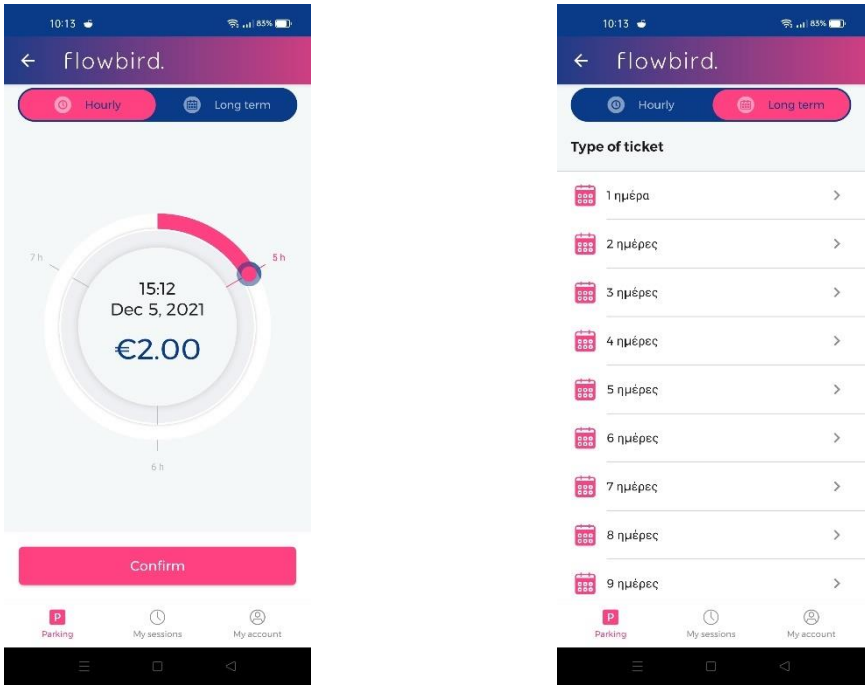


Εικόνα 2.25 Αρχική οθόνη εφαρμογής Flowbird



Εικόνα 2.26 Φόρμα εισόδου πληροφοριών οχήματος

Έπειτα αφού επιλέξει ένα από τα διαθέσιμα σημεία στάθμευσης στο χάρτη, ο χρήστης καλείται όπως και στην περίπτωση του φυλλομετρητή, να επιλέξει είτε ωριαίο είτε μακροπρόθεσμο εισιτήριο για την στάθμευση του οχήματός του (Εικόνα 2.27) και έπειτα να καταβάλει το απαραίτητο ποσό μέσω ηλεκτρονικής πληρωμής, ολοκληρώνοντας έτσι τη διαδικασία έκδοσης του εισιτηρίου.



Εικόνα 2.27Επιλογή τύπου και διάρκειας εισιτηρίου

3 Ανάλυση απαιτήσεων και σχεδιασμός εφαρμογής

Σε αυτό το κεφάλαιο, θα παρουσιαστεί η ανάλυση των απαιτήσεων για τα συστήματα που αναπτύχθηκαν στο πλαίσιο της διπλωματικής εργασίας καθώς και περαιτέρω πληροφορίες σχετικά με την ανάπτυξή τους. Γενικά, στο σύστημα έχουν πρόσβαση τρία διαφορετικά είδη χρηστών: Οδηγοί, Ελεγκτές και Διαχειριστές. Οι χρήστες συνδέονται στην εφαρμογή μέσω της προσωπικής τους συσκευής. Στη συνέχεια τα αιτήματα του πελάτη (client), στέλνονται στον διακομιστή (server), ο οποίος με την σειρά του ανακτά την απαραίτητη πληροφορία από την βάση δεδομένων, προκειμένου να εμφανιστεί η απαραίτητη πληροφορία στον χρήστη.

3.1 Ανάλυση απαιτήσεων

3.1.1 Οδηγοί

Οι οδηγοί αποτελούν τους κεντρικούς χρήστες του συστήματος, και θα πρέπει να έχουν τη δυνατότητα να αγοράζουν κάρτες στάθμευσης και να επεκτείνουν τη χρονική τους ισχύ, από τον προσωπικό υπολογιστή τους ή την κινητή τους συσκευή. Επομένως, θα πρέπει να παρέχεται κατάλληλη και εύχρηστη διεπαφή χρήστη ανάλογα με το μέγεθος της οθόνης που χρησιμοποιεί.

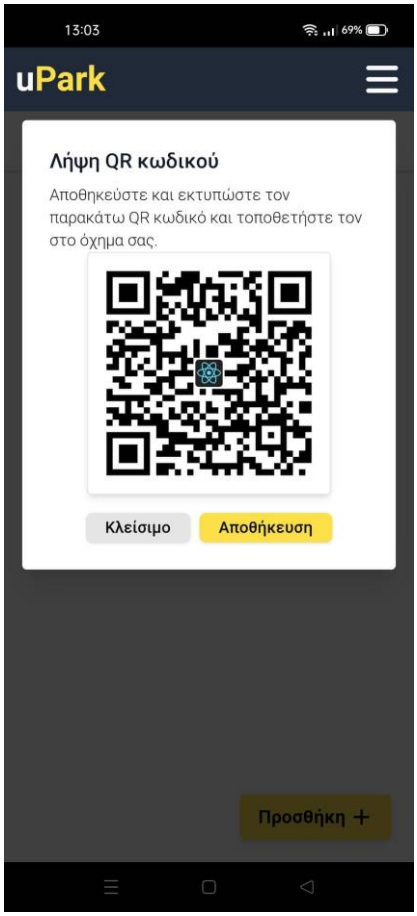
Κάθε οδηγός, προκειμένου να έχει πρόσβαση στην εφαρμογή, θα πρέπει να πραγματοποιήσει εγγραφή χρησιμοποιώντας μια διεύθυνση ηλεκτρονικού ταχυδρομείου, ώστε να δημιουργήσει τον λογαριασμό του, μέσω του οποίου θα μπορεί να πραγματοποιεί αγορά χρόνου στάθμευσης για ένα ή περισσότερα οχήματα. Επομένως, ένας οδηγός που χρησιμοποιεί την εφαρμογή θα πρέπει να μπορεί να πραγματοποιεί ηλεκτρονικές αγορές μέσω αυτής. Για τον λόγο αυτό, χρησιμοποιήθηκε η υπηρεσία ηλεκτρονικών πληρωμών Stripe (https://stripe.com/en-gr).

Αφού ολοκληρωθεί μια αγορά, ο οδηγός θα πρέπει να μπορεί να επεκτείνει τον υπολειπόμενο χρόνο στάθμευσής του ή να τον διακόπτει πρόωρα. Στην περίπτωση αυτή ο υπολειπόμενος χρόνος θα πρέπει να είναι διαθέσιμος για χρήση σε μελλοντική στάθμευση.

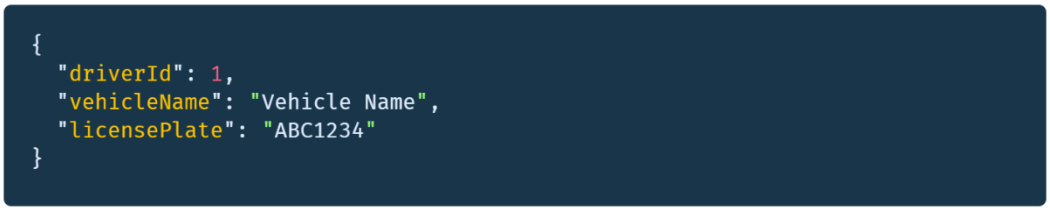
Οι βασικές ενέργειες ενός οδηγού που χρησιμοποιεί την εφαρμογή uPark, παρουσιάζονται συγκεντρωτικά στο διάγραμμα περιπτώσεων χρήσης της Εικόνα 3.3.

3.1.2 Ελεγκτές

Οι ελεγκτές του συστήματος, θα πρέπει να μπορούν να ελέγχουν μέσω της κινητής τους συσκευής εάν για ένα σταθμευμένο όχημα υπάρχει ενεργή κάρτα ή όχι. Ο έλεγχος αυτός θα μπορεί να γίνει με δύο διαφορετικούς τρόπους. Ο πρώτος τρόπος, απαιτεί να υπάρχει τοποθετημένο στο σταθμευμένο όχημα, από τον οδηγό, ένας ειδικά διαμορφωμένος κωδικός QR, ο οποίος δημιουργείται αυτόματα από την εφαρμογή, κάθε φορά που ένας οδηγός προσθέτει ένα όχημα στο λογαριασμό του. Ένα παράδειγμα τέτοιου κωδικού φαίνεται στο στιγμιότυπο της Εικόνα 3.1, ενώ στην Εικόνα 3.2 φαίνεται ένα παράδειγμα της πληροφορίας που κωδικοποιείται. Ο οδηγός μπορεί να αποθηκεύσει τον κωδικό QR του οχήματός του, τον οποίο αργότερα ένας ελεγκτής θα μπορεί να σαρώσει (scan). Ο δεύτερος τρόπος, χρησιμοποιείται σε περίπτωση που στο όχημα δεν υπάρχει ο αντίστοιχος QR κωδικός του. Ο ελεγκτής τότε, καλείται να πληκτρολογήσει χειροκίνητα τον αριθμό πινακίδας του σταθμευμένου οχήματος σε κατάλληλη φόρμα αναζήτησης, προκειμένου να διαπιστώσει εάν το όχημα είναι νόμιμα σταθμευμένο.



Εικόνα 3.1 Αυτόματα παραγώμενος QR κωδικός οχήματος



Εικόνα 3.2 Παράδειγμα πληροφορίας προς κωδικοποίηση

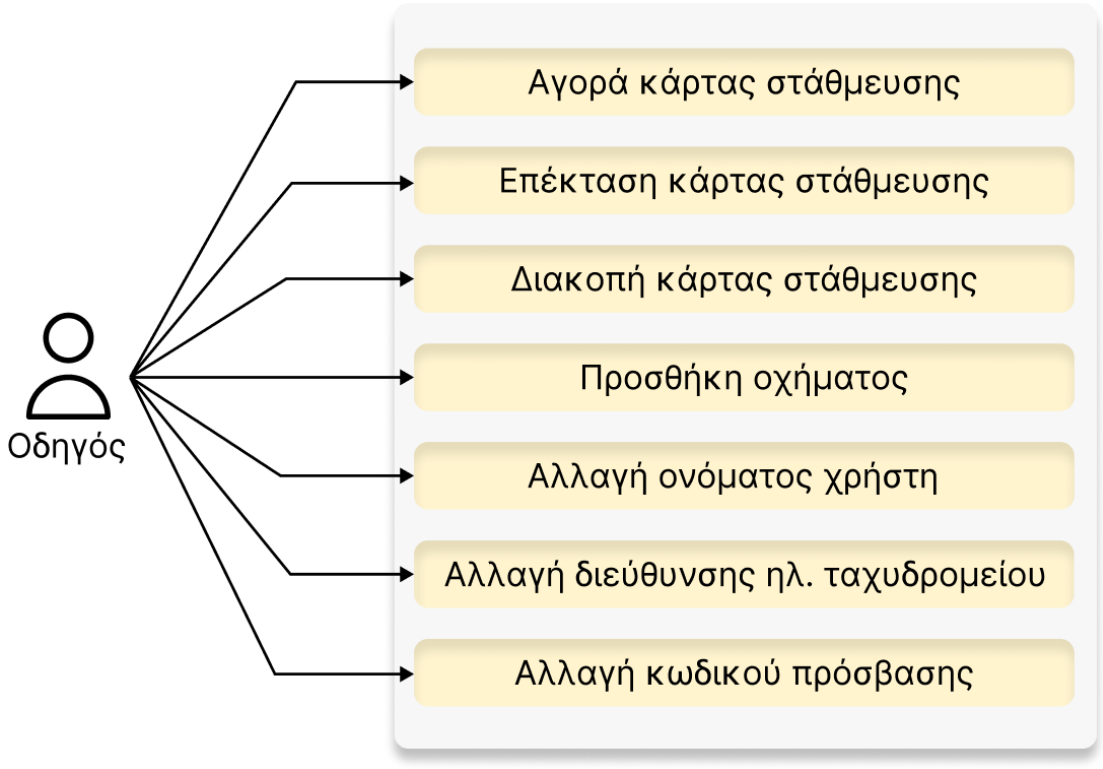
Οι βασικές ενέργειες ενός ελεγκτή του συστήματος, παρουσιάζονται συγκεντρωτικά στο διάγραμμα περιπτώσεων χρήσης της Εικόνα 3.4.

3.1.3 Διαχειριστές

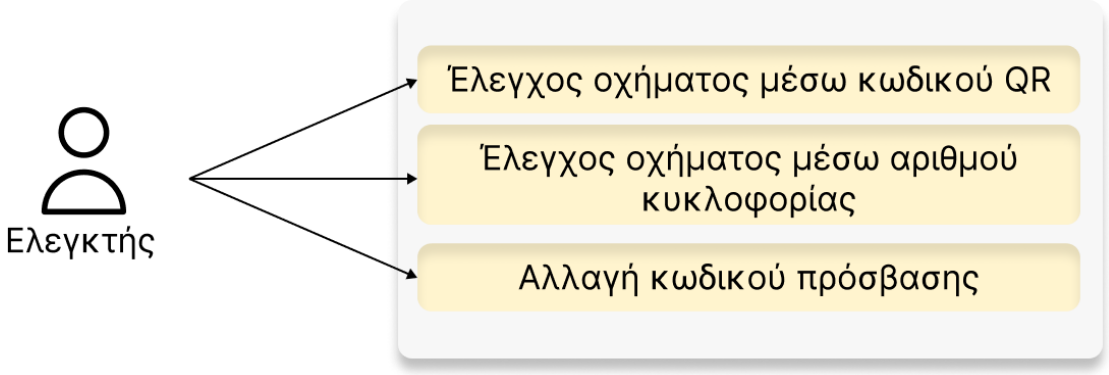
Οι διαχειριστές αποτελούν τον τελευταίο τύπο χρηστών του συστήματος. Έχουν πρόσβαση σε διάφορα δεδομένα ενδιαφέροντος, όπως για παράδειγμα έσοδα από αγορές και πλήθος εγγεγραμμένων χρηστών, καθώς και την πορεία εξέλιξης αυτών με το πέρασμα του χρόνου, τόσο σε μορφή κειμένου όσο και σε μορφή γραφικών αναπαραστάσεων, μια κοινή πρακτική σε πολλά συστήματα διαχείρισης υπηρεσιών και επιχειρήσεων. Επίσης, ένας διαχειριστής του συστήματος θα πρέπει να μπορεί να προβαίνει σε μια σειρά από ενέργειες. Πιο συγκεκριμένα, θα πρέπει να μπορεί να προσθέτει νέα σημεία στάθμευσης και να επεξεργάζεται ή να αφαιρεί τα ήδη υπάρχοντα. Επιπλέον, ένας διαχειριστής θα πρέπει να μπορεί να προσθέτει νέους ελεγκτές στη βάση δεδομένων του συστήματος και να επεξεργάζεται τα στοιχεία τους ή να τους αφαιρεί.

Οι βασικές ενέργειες ενός διαχειριστή του συστήματος, παρουσιάζονται συγκεντρωτικά στο διάγραμμα περιπτώσεων χρήσης της Εικόνα 3.5.

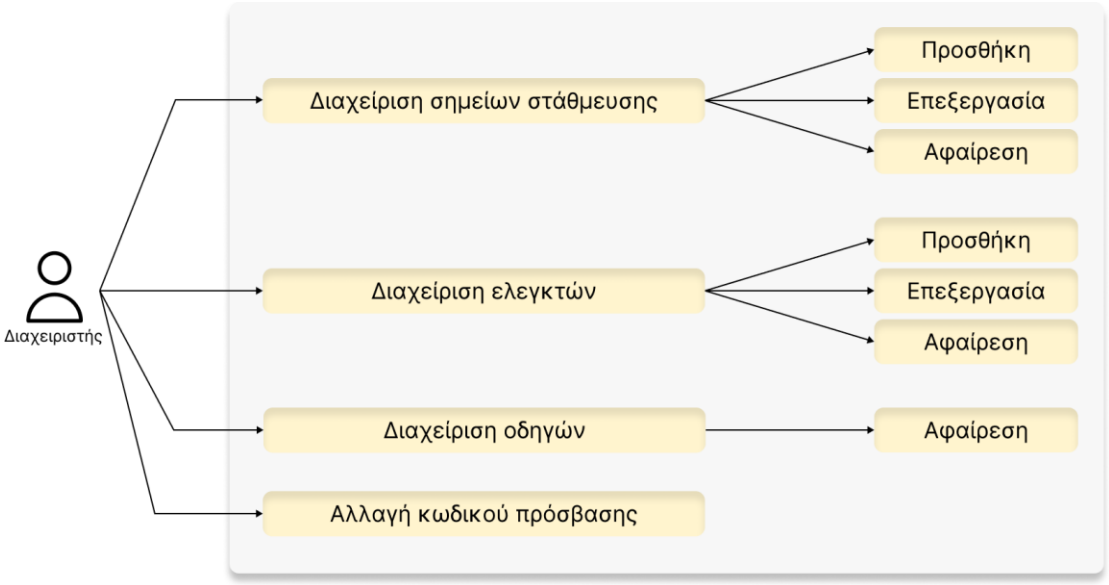
3.1.4 Διαγράμματα περιπτώσεων χρήσης



Εικόνα 3.3 Διάγραμμα περιπτώσεων χρήσης οδηγού



Εικόνα 3.4 Διάγραμμα περιπτώσεων χρήσης ελεγκτή



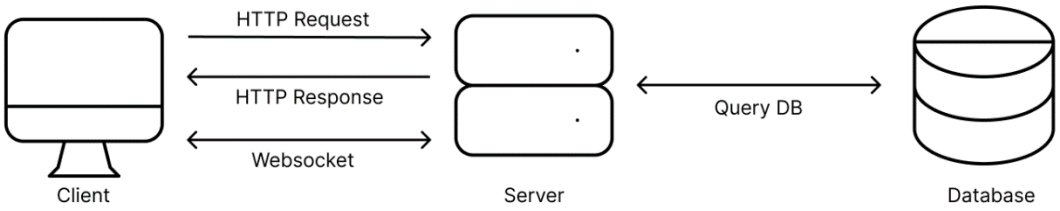
Εικόνα 3.5 Διάγραμμα περιπτώσεων χρήσης διαχειριστή

3.2 Σχεδιασμός εφαρμογής

Τα τρία υποσυστήματα που υλοποιήθηκαν στο πλαίσιο της διπλωματικής εργασίας, για τους τρείς διαφορετικούς τύπους χρηστών που έχουν πρόσβαση σε αυτά (Οδηγοί, Ελεγκτές, Διαχειριστές) ακολουθούν την ίδια φιλοσοφία όσον αφορά τον σχεδιασμό τους. Ειδικότερα, κάθε υποσύστημα εμπεριέχει τρία διαφορετικά μέρη.

Το πρώτο μέρος αφορά το τελικό αποτέλεσμα που εμφανίζεται στις οθόνες των χρηστών όταν αυτοί χρησιμοποιούν την εφαρμογή και μπορούμε να το ονομάσουμε ως το front-end μέρος της εφαρμογής. Για την ανάπτυξή του χρησιμοποιήθηκε η βιβλιοθήκη *React.js* της γλώσσας *Javascript* σε περιβάλλον *Node.* Το μέρος αυτό αποτελεί το σημείο από το οποίο πραγματοποιούνται και λαμβάνονται αιτήματα από και προς τον διακομιστή (server) του συστήματος, αντίστοιχα.

Το δεύτερο μέρος, που είναι ο διακομιστής (server) αποτελεί το back-end μέρος της εφαρμογής. Το μέρος αυτό είναι υπεύθυνο την λήψη, την επεξεργασία και την απάντηση των διαφόρων HTTP αιτημάτων, και πιο συγκεκριμένα αιτημάτων GET και POST, που λαμβάνει από ένα χρήστη της εφαρμογής. Για την ανάπτυξή του χρησιμοποιήθηκε η βιβλιοθήκη *Express.js* της γλώσσας *Javascript* σε περιβάλλον *Node.* Το back-end μπορεί επιπλέον να χαρακτηριστεί ως ο δίαυλος επικοινωνίας μεταξύ του front-end και της βάσης δεδομένων (database) του συστήματος, η οποία αποτελεί το τρίτο μέρος του συστήματος που αναπτύχθηκε.



Εικόνα 3.6 Διαγραμματική αναπαράσταση συστήματος

Στο σχήμα της Εικόνα 3.6, όπου παρουσιάζονται τα μέρη των υποσυστημάτων της εφαρμογής uPark, φαίνεται ότι το front-end δεν βρίσκεται ποτέ σε απευθείας επικοινωνία με τη βάση δεδομένων του συστήματος.

Για την διασύνδεση του περιηγητή με τον διακομιστή χρησιμοποιήσαμε το Websocket, ένα πρωτόκολλο επικοινωνίας δικτύων υπολογιστών το οποίο βασίζεται στο TCP πρωτόκολλο και παρέχει τη δυνατότητα πλήρους αμφίδρομης επικοινωνίας μεταξύ ενός περιηγητή ιστού και ενός διακομιστή. Επειδή στην εφαρμογή uPark και πιο συγκεκριμένα στο υποσύστημα που αφορά τους οδηγούς, πρέπει να εμφανίζεται διαδραστικός χάρτης, ο οποίος απαιτεί συνεχείς ανανεώσεις των δεδομένων του, χρησιμοποιήσαμε το πρωτόκολλο Websocket, καθώς ένα από τα μεγαλύτερα προτερήματά του είναι η άμεση και γρήγορη μεταφορά δεδομένων από ένα διακομιστή σε ένα φυλλομετρητή, σε πραγματικό χρόνο.

3.2.1 Σχεδιασμός βάσης δεδομένων

Για τη βάση δεδομένων του συστήματος uPark, επιλέχθηκε η δωρεάν ανοιχτού κώδικα SQL σχεσιακή βάση δεδομένων, *PostgreSQL* και χρησιμοποιήθηκε σε συνδυασμό με το εργαλείο διαχείρισης *pgAdmin.*

Έπειτα από ανάλυση που πραγματοποιήσαμε σχετικά με τα δεδομένα που θα χρειαστεί να αποθηκευτούν στη βάση δεδομένων του συστήματος, δημιουργήσαμε το Entity Relationship Diagram (ERD) διάγραμμα για τη βάση δεδομένων του συστήματος uPark το οποίο απεικονίζεται στην Εικόνα 3.7.



Εικόνα 3.7 Entity Relationship Diagram βάσης δεδομένων του συστήματος uPark (δημιουργήθηκε μέσω της ιστοσελίδας https://dbdiagram.io/)

Όπως φαίνεται στο διάγραμμα, οι πίνακες **drivers, inspectors** και **admins**, αντικατοπτρίζουν τους τρείς διαφορετικούς τύπους χρηστών που έχουν πρόσβαση στο σύστημα, δηλαδή τους οδηγούς, τους ελεγκτές και τους διαχειριστές αντίστοιχα. Για κάθε έναν από αυτούς τους τύπους χρηστών που είναι εγγεγραμμένοι στο σύστημα, πέρα από τα προσωπικά στοιχεία (όνομα και επώνυμο), αποθηκεύεται στη βάση δεδομένων μια μοναδική διεύθυνση ηλεκτρονικού ταχυδρομείου και ένας κωδικός πρόσβασης τα οποία ανακαλούνται για τον έλεγχο ταυτότητας (authentication) των χρηστών κατά την διαδικασία της σύνδεσης στο σύστημα. Πρέπει να τονιστεί ότι για λόγους ασφαλείας σε περιπτώσεις υποκλοπής

δεδομένων, οι κωδικοί που αποθηκεύονται στη βάση δεδομένων βρίσκονται σε κρυπτογραφημένη μορφή (hash), μια διαδικασία που πραγματοποιείται στο διακομιστή του συστήματος προτού εκτελεστεί το INSERT query στη βάση δεδομένων του συστήματος. Ο τρόπος με τον οποίο αποθηκεύεται ένας κωδικός στη βάση δεδομένων φαίνεται στο στιγμιότυπο της Εικόνα 3.8.

id [PK] integer	first_name character varying (50)	last_name character varying (50)	display_name character varying (100)	email character varying (255)	password text
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
11	Κωνσταντίνος	Πάνος	Κωνσταντίνος Πάνος	kpanos@email.com	$2a$10$r/xXANd4uiUIX5qb0tFt9O3Ja.AIc4HNIV9.ULtM2DiboJWw4SRbK
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

Εικόνα 3.8 Τρόπος αποθήκευσης κωδικού στη βάση δεδομένων

Οι δύο πίνακες με ονομασίες **active_cards** και **inactive_cards**, σχετίζονται με τις κάρτες χρόνου στάθμευσης τις οποίες αγοράζει ένας οδηγός που χρησιμοποιεί την εφαρμογή uPark. Ο παραπάνω διαχωρισμός σε δύο διαφορετικούς πίνακες έγινε για την πιο γρήγορη και αποδοτική εκτέλεση των SQL ερωτημάτων στη βάση δεδομένων. Πιο συγκεκριμένα, στον πίνακα active_cards αποθηκεύονται όλες οι απαραίτητες πληροφορίες, μόνο για τις κάρτες αυτές, για τις οποίες υπάρχει ακόμη ενεργός χρόνος στάθμευσης. Όταν ο χρόνος αυτός λήξει ή όταν μια ενεργή κάρτα ακυρωθεί πρόωρα από έναν οδηγό, τότε αφαιρείται η αντίστοιχη εγγραφή από τον πίνακα active_cards και δημιουργείται κατάλληλη εγγραφή στον πίνακα inactive_cards. Επομένως, ο πίνακας inactive_cards λειτουργεί ουσιαστικά σαν ένα ιστορικό όλων των σταθμεύσεων που έχουν πραγματοποιηθεί και ολοκληρωθεί.

Ο πίνακας με την ονομασία **addresses** χρησιμοποιείται για την αποθήκευση των πληροφοριών για τα σημεία στάθμευσης της εφαρμογής uPark. Κάθε εγγραφή στον συγκεκριμένο πίνακα χρησιμοποιείται για την αποθήκευση ενός συγκεκριμένου σημείου στάθμευσης. Το σημείο αυτό δεν αποτελεί μεμονωμένη θέση, αλλά μια συλλογή θέσεων οι οποίες μπορούν να εντοπιστούν για παράδειγμα σε ένα κομμάτι μιας οδού. Για κάθε σημείο στη βάση δεδομένων αποθηκεύεται μια ονομασία που θα εμφανίζεται στο χρήστη, οι συνολικές διαθέσιμες θέσεις στο σημείο (τμήμα) αυτό, καθώς και ο αριθμός των κατειλημμένων θέσεων, ο οποίος ανανεώνεται αυτόματα ανάλογα με τον αριθμό των ενεργών καρτών στάθμευσης για το σημείο αυτό. Το σημείο αυτό εν συνεχεία, με βάση τις συντεταγμένες που είναι αποθηκευμένες στη στήλη *position*, εμφανίζεται στο διαδραστικό χάρτη που χρησιμοποιούν οι οδηγοί του συστήματος uPark.

Ο πίνακας **vehicles** χρησιμοποιείται για την αποθήκευση των οχημάτων τα οποία εισάγουν οι οδηγοί της εφαρμογής από το λογαριασμό τους. Για κάθε όχημα αποθηκεύεται μια ονομασία που επιλέγει κάποιος οδηγός, καθώς και ο αριθμός της πινακίδας του. Τέλος, ο πίνακας **earnings**, χρησιμοποιείται για την καταγραφή των ηλεκτρονικών αγορών που πραγματοποιούνται από τους οδηγούς που χρησιμοποιούν την εφαρμογή uPark. Δεν γίνεται αποθήκευση προσωπικών στοιχείων, παρά μόνο του ποσού πληρωμής και της ημερομηνίας πραγματοποίησης της ηλεκτρονικής αγοράς. Τα δεδομένα αυτά χρησιμοποιούνται για την εμφάνιση πληροφοριών σχετικών με χρηματικά έσοδα, στους διαχειριστές του συστήματος.

4 Τεχνολογίες και λογισμικό ανάπτυξης

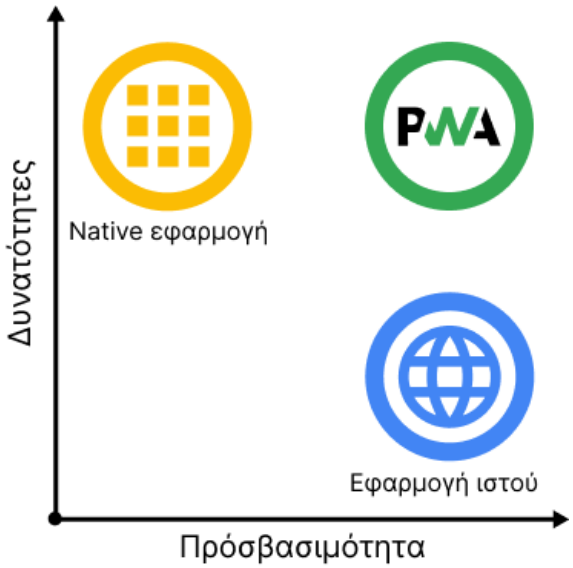
Σε αυτό το κεφάλαιο θα γίνει αναφορά σε ορισμένες βασικές έννοιες που αφορούν την διπλωματική εργασία και στη συνέχεια θα παρουσιαστούν οι διαφορετικές γλώσσες, βιβλιοθήκες και τα εργαλεία ανάπτυξης που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος uPark.

4.1 Progressive Web Application – PWA

Progressive Web Application ή αλλιώς PWA, ονομάζεται μια εφαρμογή διαδικτύου ή μια ιστοσελίδα, η οποία πληροί ορισμένα κριτήρια και χρησιμοποιεί ορισμένες τεχνολογίες που παρέχουν οι σύγχρονοι φυλλομετρητές, οι οποίες της επιτρέπουν να λειτουργεί στην κινητή συσκευή του χρήστη, με την μορφή μιας native mobile εφαρμογής [1].

Στη γενική τους μορφή, η ανάπτυξη των PWAs γίνεται με την χρήση δύο βασικών τεχνολογιών που συναντώνται στην πλειοψηφία των σύγχρονων περιηγητών ιστού, των **web app manifest** αρχείων και των **service workers**, που αναλύονται περαιτέρω στις ενότητες 4.1.1 και 4.1.2, αντίστοιχα. Ένα χαρακτηριστικό στοιχείο των PWA είναι ότι ο προγραμματιστής μπορεί να χρησιμοποιήσει όσες από τις λειτουργιές τους του φαίνονται χρήσιμες και δεν δεσμεύεται σε μια συγκεκριμένη δομή, συμβάλλοντας έτσι στην ευκολία της ανάπτυξης και στην δημιουργία μιας προσιτής αφετηρίας για έναν προγραμματιστή που επιθυμεί να αναπτύξει μια εφαρμογή διαδικτύου μέσω των τεχνολογιών αυτών [2].

Πρέπει να τονιστεί ότι τα PWAs δεν αποτελούν native mobile εφαρμογές αλλά ιστοσελίδες, ανεπτυγμένες με τεχνολογίες ιστού όπως HTML, Javascript και CSS, οι οποίες έχουν επαυξημένες δυνατότητες όπως για παράδειγμα πρόσβαση στη μνήμη και στις κάμερες της κινητής συσκευής. Το γεγονός αυτό μπορεί να γίνει πιο εύκολα αντιληπτό μέσω του διαγράμματος της Εικόνας 4.1.



Εικόνα 4.1 Σύγκριση δυνατοτήτων-προσβασιμότητας
Πηγή: https://web.dev/what-are-pwas

Ένα από τα βασικότερα πλεονεκτήματα που προσφέρει ένα PWA, είναι η δυνατότητα εγκατάστασης μιας εφαρμογής ιστού, ως συντόμευση στην αρχική οθόνη της συσκευής ενός χρήστη, μέσω μηνύματος που εμφανίζεται κατά την επίσκεψη της ιστοσελίδας μέσω ενός περιηγητή ιστού. Όταν ο χρήστης έπειτα χρησιμοποιεί την συγκεκριμένη συντόμευση, έχει πρόσβαση στην εφαρμογή σε πλήρη οθόνη και με μορφή μιας native εφαρμογής, αποκτώντας έτσι βελτιωμένη εμπειρία. Ένα ακόμη σημαντικό πλεονέκτημα των PWA είναι η δυνατότητα χρησιμοποίησης της κρυφής μνήμης (cache), για την αποθήκευση ορισμένων στοιχειών, γεγονός που αυξάνει πολλές φορές τον χρόνο εκκίνησης και εκτέλεσής τους. Ένα PWA μπορεί ακόμη να λαμβάνει και push notifications, δηλαδή μηνύματα από έναν server και να πραγματοποιεί ανανέωση του περιεχομένου της εφαρμογής σε πραγματικό χρόνο, προσφέροντας έτσι στην εφαρμογή διαδραστικότητα, χαρακτηριστικό απαραίτητο για σύγχρονες εφαρμογές [1].

Επομένως, οι εφαρμογές Progressive Web παρέχουν αξιοπιστία, ταχύτητα και διαδραστικότητα χαρακτηριστικά απαραίτητα για κάθε εφαρμογή που θέλει να παρέχει στο χρήστη μια ολοκληρωμένη εμπειρία.

4.1.1 Web App Manifest

Από τα βασικά πλεονεκτήματα μιας εφαρμογής Progressive Web είναι η δυνατότητα εγκατάστασής της στη συσκευή του χρήστη και ένα web app manifest αρχείο είναι αυτό που το καθιστά δυνατό.

Ένα web app manifest αρχείο δεν είναι παρά ένα JSON αρχείο, το οποίο λειτουργεί σαν ένα κεντρικό αποθετήριο μεταδεδομένων που σχετίζονται με την εφαρμογή ιστού [3]. Αυτά τα μεταδεδομένα περιλαμβάνουν μεταξύ άλλων, το όνομα της εφαρμογής ιστού, συνδέσμους προς εικονίδια, καθώς και την προτιμώμενη διεύθυνση URL που θα ανοίξει όταν ένας χρήστης εκκινεί την εφαρμογή ιστού. Το αρχείο αυτό, επιτρέπει επίσης στους προγραμματιστές να δηλώσουν έναν προεπιλεγμένο προσανατολισμό οθόνης για την web εφαρμογή τους, ενώ ακόμη τους δίνει τη δυνατότητα να ορίσουν τον τρόπο εμφάνισης της εφαρμογής (π.χ. σε πλήρη οθόνη) [4]. Ένα τέτοιου είδους αρχείο που αναπτύχθηκε στο πλαίσιο αυτής της διπλωματικής εργασίας φαίνεται στην Εικόνα 4.2.

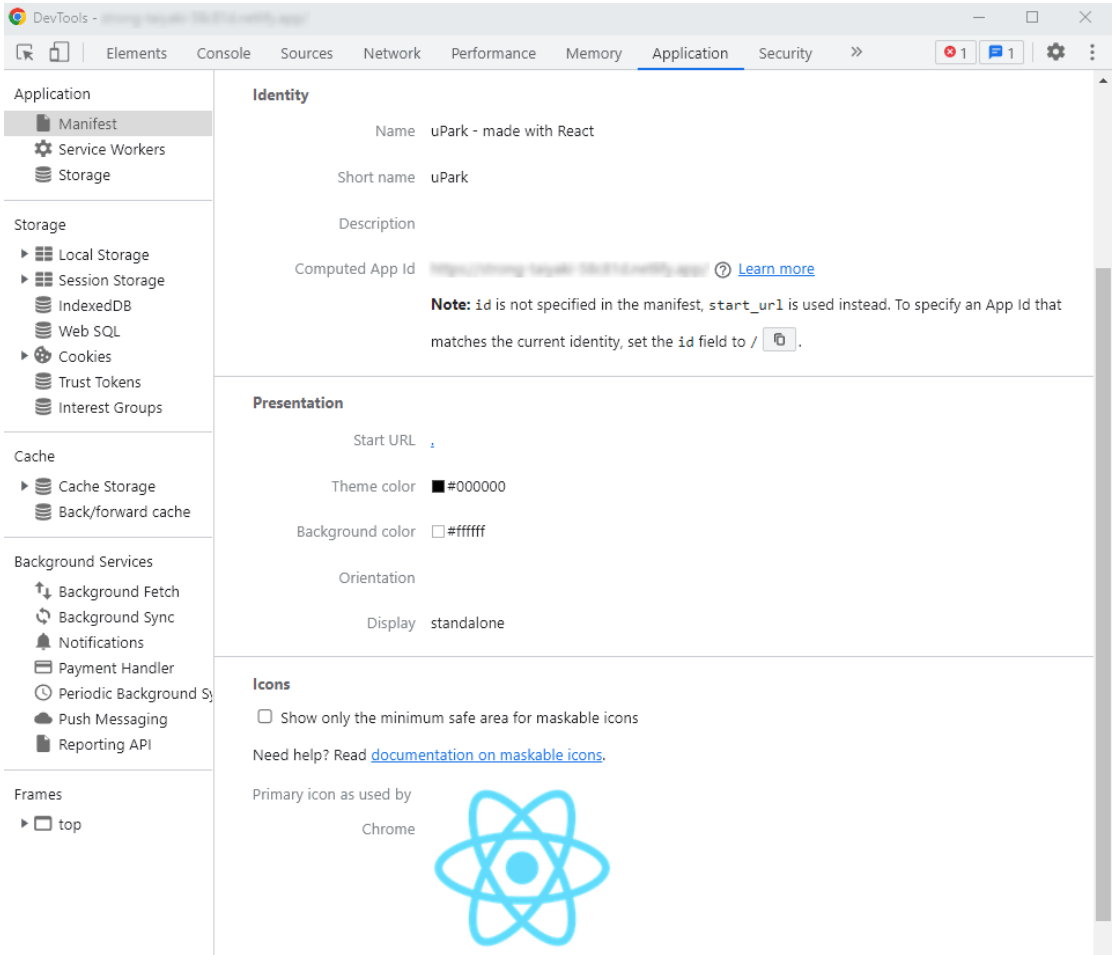
Αφού δημιουργηθεί ένα web app manifest αρχείο, πρέπει να συνδεθεί μέσω μιας HTML <link> ετικέτας, στο κομμάτι <head> κάθε σελίδας της εφαρμογής, με τον τρόπο που φαίνεται στην Εικόνα 4.3. Μπορούμε να βεβαιωθούμε για την επιτυχή σύνδεση του manifest αρχείου χρησιμοποιώντας τα εργαλεία για προγραμματιστές που διαθέτει ο περιηγητής ιστού, που ήταν ο Google Chrome στη δική μας περίπτωση και μεταβαίνοντας στην καρτέλα Εφαρμογή (Application) παρουσιάζεται το παράθυρο της Εικόνα 4.4, το οποίο περιέχει σε ευανάγνωστη μορφή τα μεταδεδομένα που ορίστηκαν στο JSON αρχείο.

```
{
  "short_name": "uPark",
  "name": "uPark - made with React",
  "icons": [
    {
      "src": "favicon.ico",
      "sizes": "64×64 32×32 24×24 16×16",
      "type": "image/x-icon"
    },
    {
      "src": "logo192.png",
      "type": "image/png",
      "sizes": "192×192"
    },
    {
      "src": "logo512.png",
      "type": "image/png",
      "sizes": "512×512"
    }
  ],
  "start_url": ".",
  "display": "standalone",
  "theme_color": "#000000",
  "background_color": "#ffffff"
}
```

Εικόνα 4.2 Παράδειγμα ενός web app manifest αρχείου

```
<head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!--
      manifest.json provides metadata used when your web app is installed on a
      user's mobile device or desktop.
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <title>uPark - made with React</title>
  </head>
  <body>
```

Εικόνα 4.3 Σύνδεση ενός web app manifest αρχείου



Εικόνα 4.4 Επιτυχής σύνδεση web app manifest αρχείου

Στη συνέχεια επεξηγούμε ορισμένα πεδία του web app manifest αρχείου, όπως αυτά ορίζονται από το World Wide Web Consortium (W3C) [4], τα οποία φαίνονται και στο παράδειγμα της Εικόνα 4.2.

- **short_name:** πεδίο τύπου *string,* το οποίο αντιπροσωπεύει μια σύντομη εκδοχή του ονόματος της εφαρμογής. Προορίζεται για χρήση όπου δεν υπάρχει επαρκής χώρος για την εμφάνιση του πλήρους ονόματος της εφαρμογής.
- **name:** πεδίο τύπου *string*, αντιπροσωπεύει το όνομα της εφαρμογής όπως αυτό εμφανίζεται συνήθως στον χρήστη (π.χ., ανάμεσα σε μια λίστα εφαρμογών ή ως ετικέτα ενός εικονιδίου).
- **icons:** προσδιορίζει αρχεία εικόνων που χρησιμεύουν ως εικονικές αναπαραστάσεις της διαδικτυακής εφαρμογής σε διάφορα περιβάλλοντα. Για παράδειγμα, μπορούν να χρησιμοποιηθούν για την αναπαράσταση της εφαρμογής μεταξύ μιας λίστας άλλων εφαρμογών ή για την απεικόνιση της εφαρμογής στην αρχική οθόνη μιας συσκευής.
- **start_url:** πεδίο τύπου *string*, αντιπροσωπεύει τη βασική διεύθυνση URL της εφαρμογής, η οποία είναι η διεύθυνση URL που ο προγραμματιστής θα προτιμούσε να φορτωθεί όταν ένας χρήστης εκκινεί την εφαρμογή ιστού, όπως για παράδειγμα όταν ο χρήστης κάνει κλικ στο εικονίδιο της εφαρμογής ιστού από τον κατάλογο των εφαρμογών ή στην αρχική οθόνη μιας συσκευής.
- **display:** πεδίο τύπου *string,* με το οποίο ο προγραμματιστής καθορίζει ποια στοιχεία της διεπαφής χρήστη του προγράμματος περιήγησης εμφανίζονται κατά την εκκίνηση της εφαρμογής. Για παράδειγμα, η γραμμή διευθύνσεων του περιηγητή ιστού μπορεί να αποκρυφθεί και να γίνει εκκίνηση της σε πλήρη οθόνη. Οι τιμές που μπορεί να πάρει αυτό το πεδίο είναι οι **fullscreen**, **standalone**, **minimal-ui** και **browser** [5].
  - **fullscreen:** η εφαρμογή ιστού εμφανίζεται με κρυμμένα τα στοιχεία διεπαφής χρήστη του προγράμματος περιήγησης και καταλαμβάνει ολόκληρη τη διαθέσιμη προς εμφάνιση περιοχή.
  - **standalone:** η εφαρμογή ιστού εμφανίζεται ως μια native εφαρμογή. Η εφαρμογή εκκινείται σε δικό της ξεχωριστό παράθυρο, έχει το δικό της εικονίδιο στην λίστα ενεργών εφαρμογών κ.λπ. Σε αυτήν τη λειτουργία, τα στοιχεία διεπαφής χρήστη του προγράμματος περιήγησης, όπως η μπάρα διευθύνσεων αποκρύπτονται. Η γραμμή κατάστασης συστήματος και το κουμπί επιστροφής στην αρχική οθόνη παραμένουν εμφανή.

- **minimal-ui:** αυτή η λειτουργία παρουσιάζει ομοιότητες με την **standalone**, με την διαφορά ότι παρέχει στο χρήστη μερικά στοιχεία διεπαφής όπως για παράδειγμα κουμπιά πλοήγησης ή επαναφόρτωσης της σελίδας και πληροφορίες σχετικά με τη διεύθυνση του ιστότοπου.
- **browser:** η εφαρμογή εκκινείται μέσω του προκαθορισμένου περιηγητή ιστού της συσκευής.

4.1.2 Service worker

Ένας service worker είναι ένα αρχείο κώδικα γραμμένο σε JavaScript, το οποίο εκτελείται ανεξάρτητα από το κύριο νήμα του προγράμματος περιήγησης και μπορεί μεταξύ άλλων να δέχεται και να αποστέλλει αιτήματα δικτύου, να αποθηκεύει προσωρινά ή να ανακτά πόρους από τη μνήμη cache και να στέλνει push notifications [6]. Με την χρήση ενός service worker μια εφαρμογή ιστού μπορεί να επεξεργάζεται αιτήματα δικτύου και να βελτιώνει την απόδοση και τον χρόνο εκτέλεσής της. Ένα ακόμη σημαντικό πλεονέκτημα που μπορεί να προσφέρει η χρήση ενός service worker είναι η εμφάνιση περιεχομένου ακόμη και χωρίς να υπάρχει σύνδεση δικτύου, με χρήση της cache μνήμης της συσκευής [2].

Επιπλέον μέσω ενός service worker, ο προγραμματιστής έχει την δυνατότητα να χρησιμοποιεί ορισμένα σύγχρονα API της γλώσσας Javascript, όπως για παράδειγμα το Push API, το οποίο επιτρέπει στις εφαρμογές Ιστού να λαμβάνουν μηνύματα που προωθούνται σε αυτές από έναν διακομιστή, ανεξάρτητα από το αν η εφαρμογή ιστού είναι ενεργή σε κάποιον περιηγητή. Αυτό επιτρέπει στους προγραμματιστές να στέλνουν ασύγχρονες ειδοποιήσεις και ενημερώσεις σε χρήστες που το επιθυμούν, κάνοντας την εφαρμογή πιο διαδραστική και προσφέροντας στους χρήστες περιεχόμενο ανανεωμένο σε πραγματικό χρόνο [3].

Προτού ένας service worker μπορέσει να λειτουργήσει σε μια εφαρμογή, πρέπει πρώτα να περάσει από τρία διαφορετικά στάδια τα οποία αποτελούν τον λεγόμενο κύκλο ζωής ενός service worker (service worker lifecycle). Τα τρία αυτά στάδια είναι τα εξής [7]:

1. **Registration** – Εγγραφή
2. **Installation** – Εγκατάσταση
3. **Activation** – Ενεργοποίηση

Το στάδιο της εγγραφής, αποτελεί το σημείο έναρξης του κύκλου ζωής ενός service worker. Πρέπει να επισημανθεί ότι ένας service worker μπορεί να λειτουργήσει μόνο σε HTTPS δίκτυα

ή σε localhost δίκτυο κατά την διάρκεια ανάπτυξης της εφαρμογής. Το κομμάτι κώδικα της Εικόνα 4.5 πραγματοποιεί την εγγραφή ενός νέου service worker.

```
<!-- /index.html -->
    <script>
      if ('serviceWorker' in navigator) {
        window.addEventListener('load', () => {
          navigator.serviceWorker
            .register('/sw.js')
            .then((reg) => {
              console.log('Success: ', reg.scope);
            })
            .catch((err) => console.log('Failure: ', err));
        });
      }
    </script>
```

Εικόνα 4.5 Service worker registration

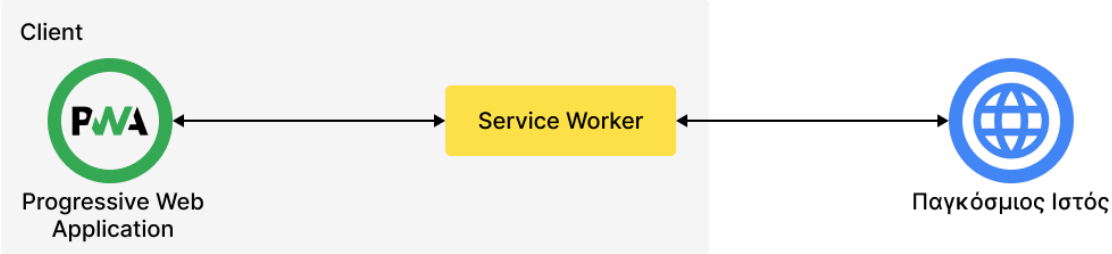
Το στάδιο της εγκατάστασης ξεκινάει έπειτα από την επιτυχημένη εγγραφή ενός service worker και εκτελείται μόνο μια φορά. Στο στάδιο αυτό ο προγραμματιστής μπορεί να προσθέσει ορισμένα στοιχεία στη cache μνήμη για μετέπειτα χρήση. Το κομμάτι κώδικα της Εικόνα 4.6 δημιουργήθηκε στο πλαίσιο της παρούσας διπλωματικής εργασίας και εκκινεί το στάδιο της εγκατάστασης του service worker.

```
// /sw.js
this.addEventListener('install', (e) => {
  e.waitUntil(
    caches.open(CACHE_NAME).then((cache) => {
      console.log('Opened cache.');

      return cache.addAll(urlsToCache);
    })
  );
});
```

Εικόνα 4.6 Service worker installation

Όταν τα δύο πρώτα στάδια ολοκληρωθούν με επιτυχία, τότε ο service worker μεταβαίνει στο στάδιο της ενεργοποίησης. Μετά το πέρας και της ενεργοποίησης ο service worker λαμβάνει την κατάσταση «activated» και ξεκινά την λειτουργία του. Η λειτουργία αυτή είναι ανεξάρτητη από την εφαρμογή, καθώς δεν απαιτείται η εφαρμογή να είναι ενεργή ή ανοιχτή σε κάποια καρτέλα του προγράμματος περιήγησης για να εκτελεστεί όπως φαίνεται και στην Εικόνα 4.7.



Εικόνα 4.7 Λειτουργία ενός service worker

4.2 Τεχνολογίες και εργαλεία που χρησιμοποιήθηκαν

Στη συνέχεια παρουσιάζουμε τα εργαλεία, καθώς και τις τεχνολογίες και γλώσσες προγραμματισμού που χρησιμοποιήσαμε για την ανάπτυξη του συστήματος uPark.

4.2.1 Node.js

Το Node.js είναι ένα δωρεάν και ανοιχτού κώδικα περιβάλλον εκτέλεσης της γλώσσας Javascript και σήμερα αποτελεί μια από τις πλέον διαδεδομένες επιλογές των προγραμματιστών για την ανάπτυξη εφαρμογών ιστού [8]. Ανά τα χρόνια έχουν υπάρξει αρκετές γλώσσες προγραμματισμού που χρησιμοποιούνται για την επεξεργασία αιτημάτων από περιηγητές ιστού σε ένα διακομιστή, όπως για παράδειγμα η PHP, Ruby, Python, ASP και η Java. Αντιθέτως το πεδίο εφαρμογής της Javascript ήταν οι φυλλομετρητές, όπου χρησιμοποιούταν για να παρέχει διαδραστικότητα στις ιστοσελίδες, όπως για παράδειγμα αναπτυσσόμενα μενού.

Το 2009, ο Ryan Dahl, δημιούργησε το Node.js, το οποίο είναι χτισμένο πάνω στη μηχανή V8 του φυλλομετρητή Google Chrome, δίνοντας έτσι τη δυνατότητα εκτέλεσης σε διακομιστές κώδικα γραμμένου σε γλώσσα Javascript. Έτσι, εκτός από τις προαναφερθείσες γλώσσες προγραμματισμού, οι προγραμματιστές είχαν πλέον τη δυνατότητα να επιλέξουν την γλώσσα JavaScript για την ανάπτυξη εφαρμογών τόσο στην πλευρά του διακομιστή (server), όσο και στην πλευρά του περιηγητή (client) [9].

Το Node.js εφαρμόζει το μοντέλο ασύγχρονης, non-blocking διαδικασίας εισόδου/εξόδου μέσω της χρήσης συμβάντων, χρησιμοποιώντας ένα μοναδικό νήμα εκτέλεσης. Έτσι, όταν για παράδειγμα εκτελείται μια διαδικασία εισόδου/εξόδου, όπως για παράδειγμα η ανάγνωση δεδομένων από μια βάση δεδομένων, το Node.js αντί να μπλοκάρει το νήμα εκτέλεσης του κώδικα, συνεχίζει να εκτελείται μέχρις ότου τα δεδομένα να είναι διαθέσιμα [10]. Εφαρμογές ανεπτυγμένες σε Node.js χαρακτηρίζονται από υψηλή ταχύτητα και αποδοτικότητα, γεγονός που καθιστά το Node.js ιδανική επιλογή για εφαρμογές που επιδιώκουν υψηλή διαδραστικότητα και ανανέωση δεδομένων σε πραγματικό χρόνο. Ένα ακόμη σημαντικό

πλεονέκτημα της χρήσης του Node.js, είναι το γεγονός ότι καθιστά δυνατή τη χρήση μιας κοινής γλώσσας τόσο για την ανάπτυξη του front end, όσο και για την ανάπτυξη του back end.

4.2.2 Typescript

Η Typescript (https://www.typescriptlang.org) είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού ανοιχτού κώδικα, η οποία αναπτύχθηκε από την Microsoft. Αποτελεί υπερσύνολο της γλώσσας Javascript, με την έννοια ότι εισάγει την χρήση στατικών τύπων στη γλώσσα, καλύπτοντας έτσι τις ανάγκες που προκύπτουν κατά την ανάπτυξη μεσαίων ή μεγαλύτερων εφαρμογών [11]. Μέσω του TypeScript Compiler (TSC), ο κώδικας που ένα προγραμματιστής γράφει σε TypeScript μεταγλωττίζεται σε κώδικα Javascript, ο οποίος στη συνέχεια μπορεί να εκτελεστεί τόσο σε έναν περιηγητή, όσο και σε έναν διακομιστή [12].

Στα πλεονεκτήματα που προσφέρει η χρήση της γλώσσας Typescript σε σχέση με απλή Javascript, συγκαταλέγεται η χρήση των στατικών τύπων που προσφέρει όταν αυτό είναι επιθυμητό από τον προγραμματιστή, η αυτόματη ανάθεση τύπων σε μεταβλητές χωρίς να απαιτείται κάποια ενέργεια από την μεριά του προγραμματιστή, η χρήση σύγχρονων εκδόσεων της Javascript, καθώς και η δυνατότητα μεταγλώττισης σε παλαιότερες εκδόσεις της, εξασφαλίζοντας συμβατότητα με παλαιότερους περιηγητές ιστού και τέλος η υποστήριξη εργαλείων αυτόματης συμπλήρωσης κώδικα (IntelliSense) [13].

4.2.3 Express.js

Η Express.js (https://expressjs.com) είναι μια δωρεάν και ανοιχτού κώδικα βιβλιοθήκη/framework για το Node.js, που χρησιμοποιείται για την γρήγορη και ευκολότερη ανάπτυξη διακομιστών δικτύου, με στόχο την επεξεργασία αιτημάτων από τους περιηγητές. Η Express.js προσφέρει ένα υψηλότερο επίπεδο αφαίρεσης σε σύγκριση με το Node, γεγονός που επιτρέπει στους προγραμματιστές να χρησιμοποιούν με μεγαλύτερη ευκολία την λειτουργικότητα που προσφέρει, δίνοντάς τους πρόσβαση και σε επιπλέον δυνατότητες.

Ένα από τα σημαντικότερα πλεονεκτήματα που προσφέρει η βιβλιοθήκη Express.js, είναι η χρήση της γλώσσας JavaScript τόσο για την ανάπτυξη του frontend όσο και του backend μέρους μιας εφαρμογής ιστού. Επομένως ο προγραμματιστής δεν χρειάζεται να μάθει ή να χρησιμοποιήσει καμία άλλη γλώσσα. Επίσης, καθώς στηρίζεται στη μηχανή V8 του περιηγητή Google Chrome, η Express.js αποτελεί ιδανική επιλογή για εφαρμογές στις οποίες απαιτείται πολύ γρήγορη απόκριση και επεξεργασία αιτημάτων, καθώς και για εφαρμογές που εκτελούνται σε πραγματικό χρόνο, όπως για παράδειγμα υπηρεσίες live-chat.

4.2.4 React.js

Η React (https://reactjs.org) είναι μια δωρεάν και ανοιχτού κώδικα βιβλιοθήκη της γλώσσας Javascript, η οποία χρησιμοποιείται για την ανάπτυξη διεπαφών χρήστη. Η τελευταία σταθερή της έκδοση κατά τη συγγραφή αυτής της εργασίας είναι η 18.0.0, όμως το σύστημα uPark χρησιμοποιεί την έκδοση 17.0.2. Η React αποτελεί την πιο διαδεδομένη λύση για την ταχεία και αποδοτική ανάπτυξη σύνθετων διεπαφών χρήστη και βρίσκει εφαρμογή τόσο σε εφαρμογές ιστού όσου και κινητών συσκευών.

Εφαρμογές ανεπτυγμένες με τη βιβλιοθήκη React διασπώνται σε επιμέρους τμήματα, που καλούνται components, ανάλογα με τη λειτουργικότητά τους. Για παράδειγμα, η μπάρα πλοήγησης στην κορυφή μιας ιστοσελίδας θα μπορούσε να αποτελέσει ξεχωριστό component. Έτσι το τελικό αποτέλεσμα που προκύπτει είναι ένα μεγάλο και σύνθετο component, μια ιστοσελίδα, η οποία συντίθεται από πολλά μικρότερα σε μέγεθος επιμέρους components [11].

Ένα από τα κυριότερα πλεονεκτήματα της χρήσης της βιβλιοθήκης React είναι η ευκολία που προσφέρει στην ανάπτυξη εφαρμογών και η ταχύτητα των εφαρμογών αυτών. Η React επιτυγχάνει αυτή την ταχύτητα μέσω ενός Virtual DOM (Document Object Model). Το Virtual DOM ιχνηλατεί τα δεδομένα της εφαρμογής, συγκρίνοντας τις προηγούμενες καταστάσεις τους (states) και επικαιροποιεί στο πραγματικό DOM μόνο τα στοιχεία στα οποία ανιχνεύθηκε κάποια αλλαγή, αντί να γίνει πλήρης ανανέωση της σελίδας, όπως συμβαίνει σε συμβατικές εφαρμογές ιστού [11].

4.2.5 TailwindCSS και CSS

Στο πλαίσιο της διπλωματικής εργασίας έγινε εκτενής χρήση του CSS framework TailwindCSS (https://tailwindcss.com/), το οποίο επιτρέπει τη γρήγορη δημιουργία εξατομικευμένων διεπαφών χρήστη και μπορεί να χρησιμοποιηθεί για τη δημιουργία ιστοσελίδων με γρήγορο και απλό τρόπο. Σε σύγκριση με άλλα ευρέως διαδεδομένα frameworks, όπως για παράδειγμα το Bootstrap, το οποίο παρέχει στους προγραμματιστές προκαθορισμένα στοιχεία όπως κουμπιά και μενού, το TailwindCSS αποτελεί ένα utility-first CSS framework, το οποίο προσφέρει μικρότερα δομικά στοιχειά (CSS classes) τα οποία οι προγραμματιστές συνδυάζουν για την δημιουργία μεγαλύτερων και εξειδικευμένων στοιχειών που ανταποκρίνονται καλύτερα στις απαιτήσεις τους, με γρήγορο και αποδοτικό τρόπο. Ένα από τα μεγαλύτερα πλεονεκτήματα που προσφέρει το TailwindCSS είναι η δυνατότητα ανάπτυξης προσαρμοστικών (responsive) διεπαφών χρήστη για διαφορετικά μεγέθη οθονών, ενώ ακόμη προσφέρει και έναν ομοιόμορφο τρόπο ονομασίας για την ονομασία των CSS κλάσεων

μια εφαρμογής [14]. Στο πλαίσιο της ανάπτυξης της εφαρμογής uPark χρειάστηκε σε ορισμένα σημεία και η συγγραφή CSS κλάσεων, με χρήση απλής CSS.

4.2.6 Επιπλέον βιβλιοθήκες που χρησιμοποιήθηκαν

Στη συνέχεια παραθέτουμε περιληπτικά ορισμένες από τις σημαντικότερες βιβλιοθήκες που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος.

- **bcryptjs:** βιβλιοθήκη που χρησιμοποιείται για το hashing των κωδικών ασφαλείας των χρηστών του συστήματος, πριν την αποθήκευσή τους στη βάση δεδομένων.
- **passport:** βιβλιοθήκη που χρησιμοποιείται για την πιστοποίηση της ταυτότητας (authentication) των χρηστών του συστήματος. Παρέχει πρόσβαση σε πολλούς διαφορετικούς τρόπους πιστοποίησης της ταυτότητας χρηστών μέσω επεκτάσεων.
- **passport-local:** επέκταση της βιβλιοθήκης *passport* για την πιστοποίηση της ταυτότητας (authentication) χρηστών με τη χρήση διεύθυνσης ηλεκτρονικού ταχυδρομείου και κωδικού.
- **pg:** διασύνδεση βάσης δεδομένων *PostgreSQL* με διακομιστή υλοποιημένο σε περιβάλλον *Node.js.*
- **socket.io:** βιβλιοθήκη που επιτρέπει την αμφίδρομη επικοινωνία μεταξύ πελάτη (client) και διακομιστή (server) μέσω συμβάντων, υλοποιημένη με βάση το πρωτόκολλο επικοινωνίας *WebSocket*.
- **stripe:** βιβλιοθήκη που παρέχει πρόσβαση στο *Stripe API* σε διακομιστές ανεπτυγμένους σε περιβάλλον *Node.js*, για την πραγματοποίηση ηλεκτρονικών πληρωμών μέσω Stripe.
- **react-router-dom:** βιβλιοθήκη που χρησιμοποιείται για την δρομολόγηση εφαρμογών ιστού υλοποιημένες με *React.js.*
- **leaflet:** βιβλιοθήκη ανοικτού κώδικα για την εύκολη δημιουργία διαδραστικών χαρτών σε εφαρμογές ιστού.
- **axios:** βιβλιοθήκη που χρησιμοποιείται για την πραγματοποίηση HTTP αιτημάτων, βασίζεται στο *Promise API* της γλώσσας Javascript και μπορεί να χρησιμοποιηθεί τόσο στην πλευρά του πελάτη (client-side) όσο και στην πλευρά του διακομιστή (server-side) μιας εφαρμογής ιστού.
- **react-qr-reader:** βιβλιοθήκη που χρησιμοποιείται για την ανάγνωση QR κωδικών από μια διαδικτυακή εφαρμογή μέσω κάμερας.
- **chart.js:** βιβλιοθήκη που χρησιμοποιείται για την απλοποίηση της δημιουργίας διαγραμμάτων και γραφικών αναπαραστάσεων σε εφαρμογές ιστού.

4.2.7 Visual Studio Code

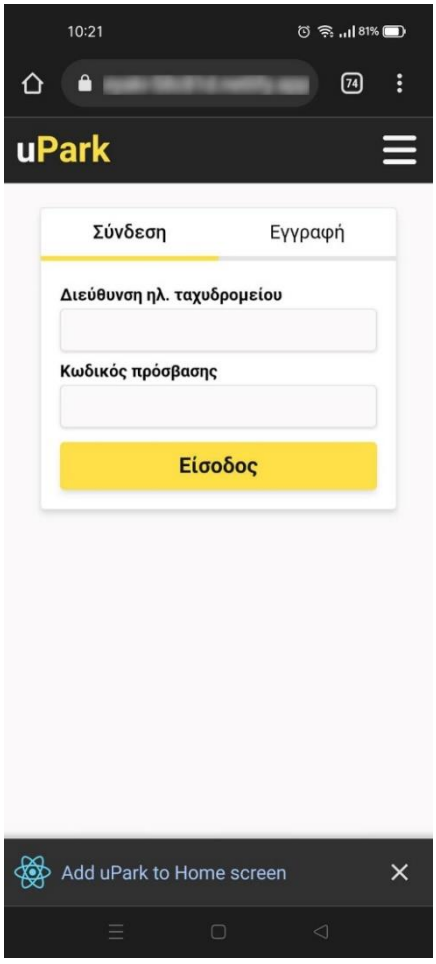
Το Visual Studio Code (https://code.visualstudio.com/docs) αποτέλεσε τον κύριο επεξεργαστή κειμένου για την συγγραφή του κώδικα στο πλαίσιο της διπλωματικής εργασίας. Το Visual Studio Code υποστηρίζει τη συγγραφή για πληθώρα γλωσσών προγραμματισμού συμπεριλαμβανομένης και της Typescript. Επιπλέον, είναι δυνατό να επεκταθεί η λειτουργικότητα που προσφέρει μέσω επεκτάσεων. Σε συνεργασία με τον Typescript Compiler (TSC), η εύρεση πιθανών σφαλμάτων γίνεται κατά τη διάρκεια της συγγραφής του κώδικα, ενώ παράλληλα η αυτόματη συμπλήρωση κώδικα που προσφέρει, συμβάλλει στην αύξηση της παραγωγικότητας του προγραμματιστή.

5 Παρουσίαση εφαρμογής

5.1 Εγκατάσταση ενός Progressive Web Application

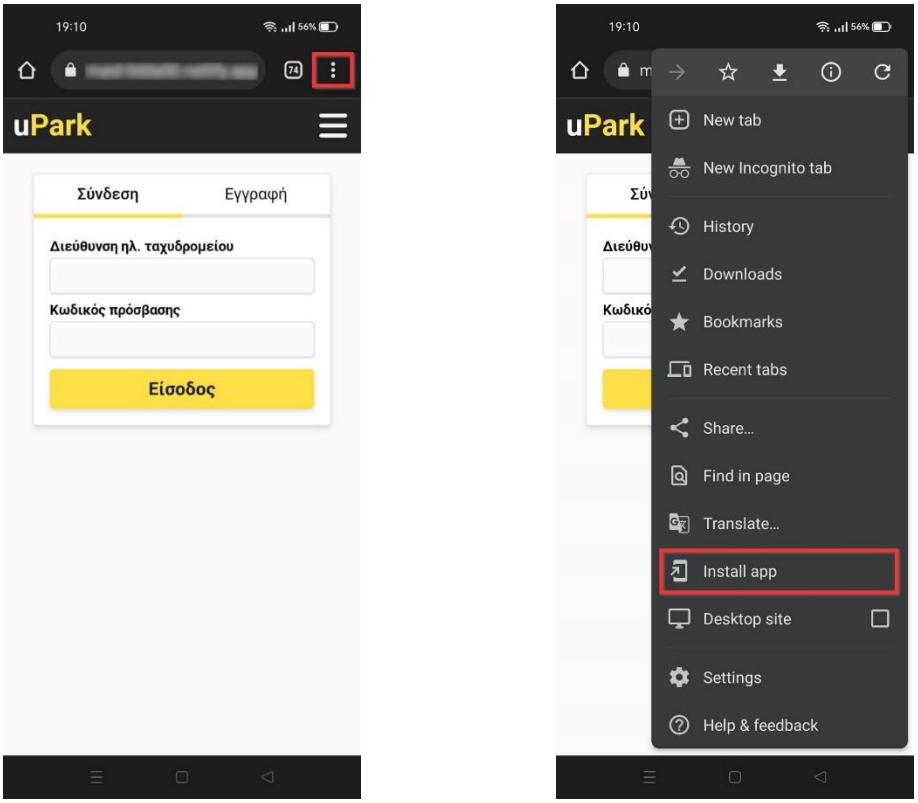
Προτού παρουσιάσουμε το σύστημα uPark που αναπτύχθηκε στο πλαίσιο της παρούσας διπλωματικής εργασίας, θα περιγράψουμε τον τρόπο μπορούμε να εγκαταστήσουμε ένα Progressive Web Application (PWA) στη συσκευή μας, προκειμένου να επωφεληθούμε των προτερημάτων που προσφέρει. Στην παρουσίαση που ακολουθεί, πραγματοποιείται η εγκατάσταση του υποσυστήματος που αφορά τους οδηγούς σε κινητή συσκευή λειτουργικού συστήματος Android έκδοσης 11, μέσω του περιηγητή ιστού Google Chrome.

Αρχικά ένας υποθετικός οδηγός επισκέπτεται την ιστοσελίδα στην οποία φιλοξενείται η εφαρμογή ιστού μέσω του περιηγητή Google Chrome. Στο κάτω μέρος της σελίδας εμφανίζεται αυτόματα αναδυόμενη μπάρα με το μήνυμα «Add uPark to Home screen» (Εικόνα 5.1). Επιλέγοντάς την ο οδηγός ξεκινάει την προσθήκη του Progressive Web Application στην συσκευή του.



Εικόνα 5.1 Μπάρα προσθήκης PWA στην αρχική οθόνη

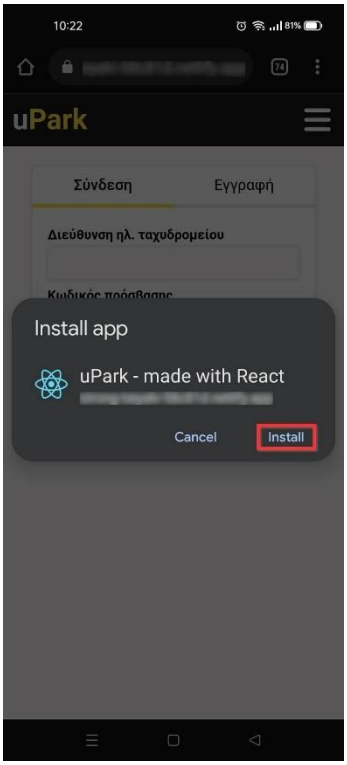
Σε περίπτωση που η μπάρα δεν εμφανιστεί αυτόματα, τότε ο οδηγός θα πρέπει μέσω των επιλογών του περιηγητή Google Chrome (Εικόνα 5.2, αριστερά) να επιλέξει «Install app» (Εικόνα 5.2, δεξιά).



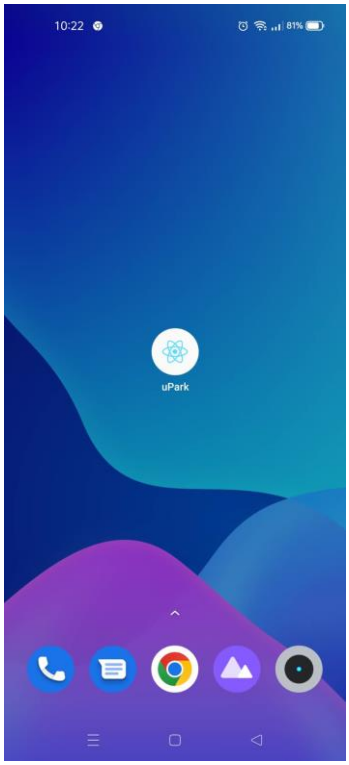
Εικόνα 5.2 Εναλλακτικός τρόπος προσθήκης PWA στην αρχική οθόνη

Στη συνέχεια, στο αναδυόμενο παράθυρο που εμφανίζεται (Εικόνα 5.3), ο οδηγός επιλέγει «Install» και ξεκινάει η προσθήκη του PWA στην συσκευή του. Όταν η εγκατάσταση ολοκληρωθεί, τότε ο οδηγός μπορεί να εντοπίσει στην αρχική οθόνη της συσκευής του το εικονίδιο της εφαρμογής uPark (Εικόνα 5.4), μαζί με τις υπόλοιπες εφαρμογές του.

Κάνοντας εκκίνηση της εφαρμογής μέσω της συντόμευσης αυτής ο οδηγός καλωσορίζεται αρχικά από ένα splash screen (Εικόνα 5.5) και στη συνέχεια η εφαρμογή εκτελείται σε πλήρη οθόνη ακριβώς όπως θα γινόταν και με μια native Android εφαρμογή (Εικόνα 5.6). Στο σημείο αυτό η διαδικασία της εγκατάστασης του Progressive Web Application στη κινητή συσκευή του οδηγού έχει ολοκληρωθεί με επιτυχία.



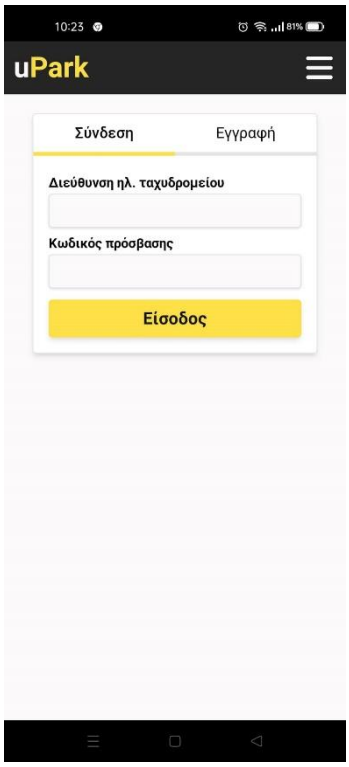
Εικόνα 5.3 Εγκατάσταση του PWA



Εικόνα 5.4 Προσθήκη συντόμευσης στην αρχική οθόνη



Εικόνα 5.5 Οθόνη splash screen



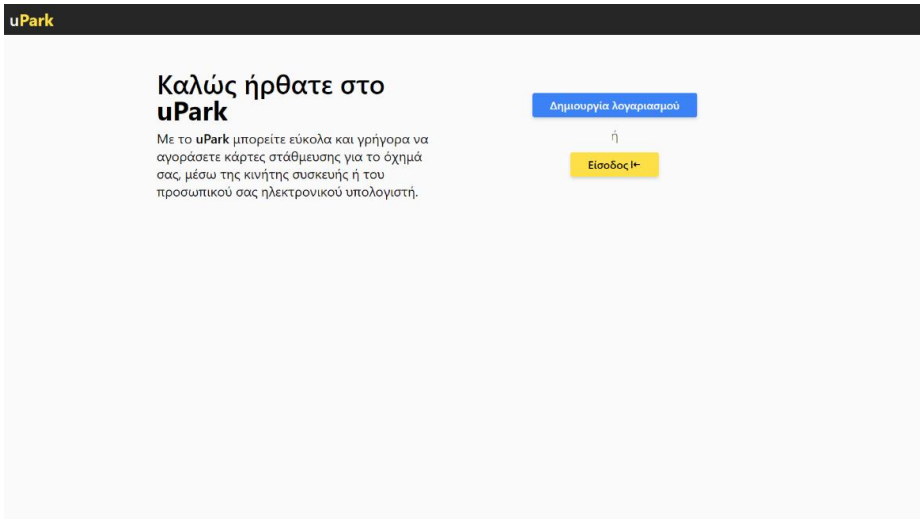
Εικόνα 5.6 Εκτέλεση σε πλήρη οθόνη

5.2 Παρουσίαση υποσυστημάτων εφαρμογής

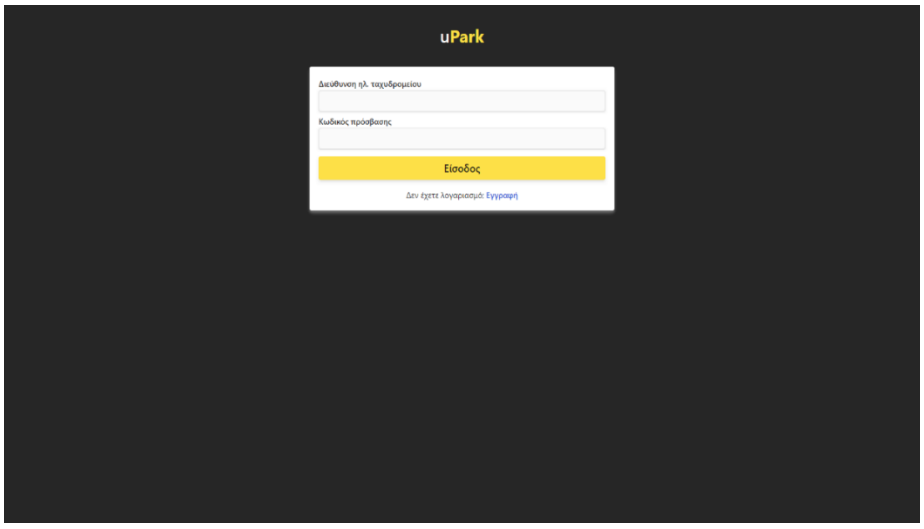
Στο σημείο αυτό θα προβληθούν και θα περιγραφούν οι οθόνες του συστήματος uPark που αναπτύχθηκε στο πλαίσιο αυτής της διπλωματικής εργασίας. Αρχικά θα παρουσιάσουμε το υποσύστημα που αφορά τους οδηγούς, στη συνέχεια το υποσύστημα που αφορά τους ελεγκτές και τέλος, το υποσύστημα που αφορά τους διαχειριστές του συστήματος.

5.2.1 Υποσύστημα οδηγού

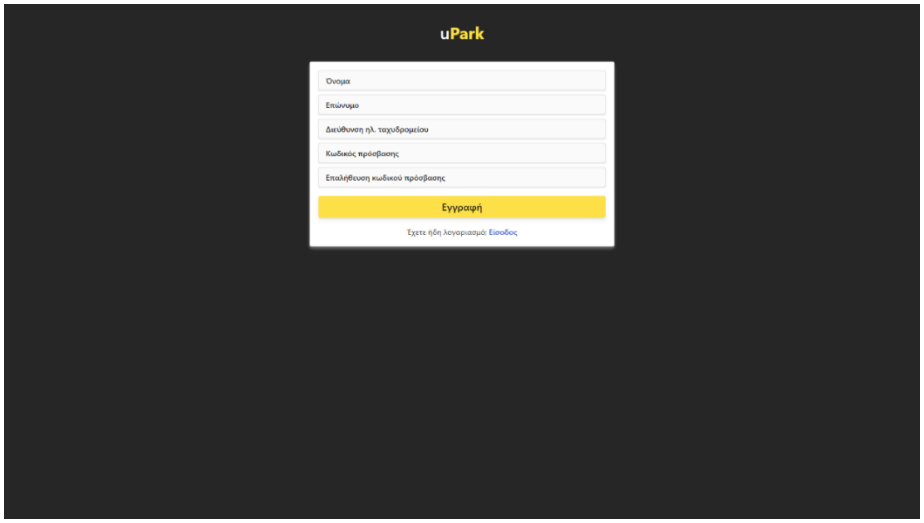
Ένας οδηγός ο οποίος επισκέπτεται την εφαρμογή, ανακατευθύνεται στην αρχική οθόνη του συστήματος, στιγμιότυπο της οποίας παρουσιάζεται στην οθόνη της Εικόνα 5.7 και της Εικόνα 5.10 για χρήστες σταθερών και κινητών συσκευών, αντίστοιχα.



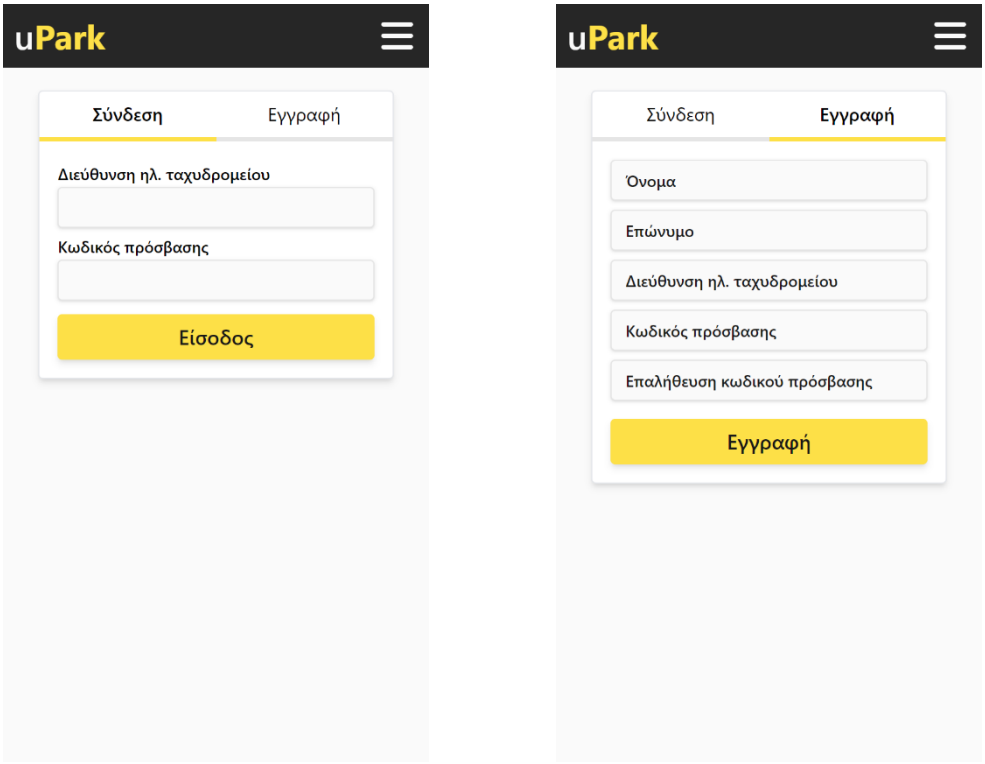
Εικόνα 5.7 Αρχική οθόνη οδηγού (desktop)



Εικόνα 5.8 Οθόνη εισόδου οδηγού (desktop)



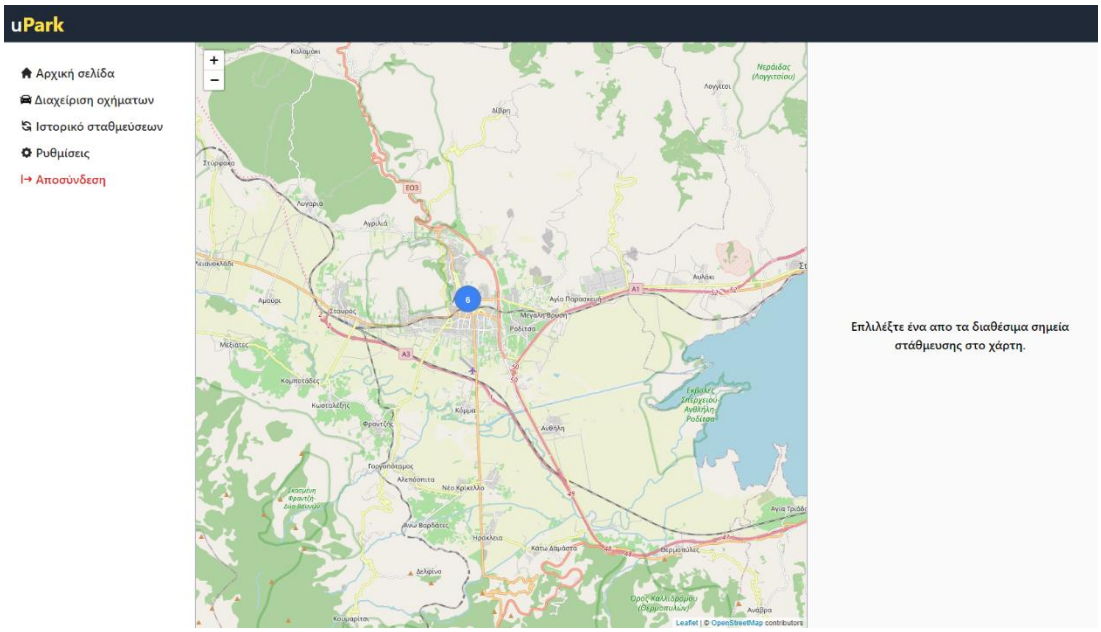
Εικόνα 5.9 Οθόνη εγγραφής οδηγού (desktop)



Εικόνα 5.10 Αρχική οθόνη οδηγού (mobile)

Μέσω της αρχικής οθόνης του συστήματος ένας οδηγός έχει την επιλογή να συνδεθεί σε αυτό εάν διαθέτει ήδη λογαριασμό ή να δημιουργήσει ένα νέο λογαριασμό μέσω των φορμών που παρουσιάζονται στις οθόνες των εικόνων 5.8, 5.9 και 5.10.

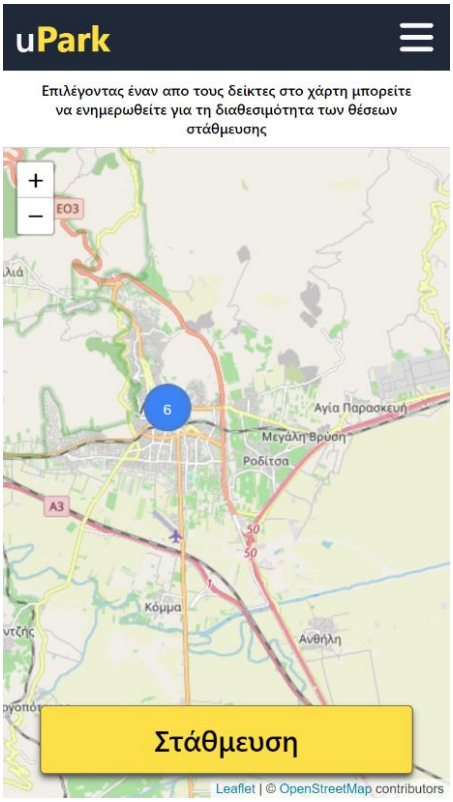
Εάν η σύνδεση ενός οδηγού στο σύστημα πραγματοποιηθεί με επιτυχία, τότε μέσα από το περιβάλλον του λογαριασμού του, του παρουσιάζεται η οθόνη της Εικόνα 5.11 όταν συνδέεται από τον προσωπικό του υπολογιστή, και της Εικόνα 5.12 όταν συνδέεται μέσω της κινητής του συσκευής.



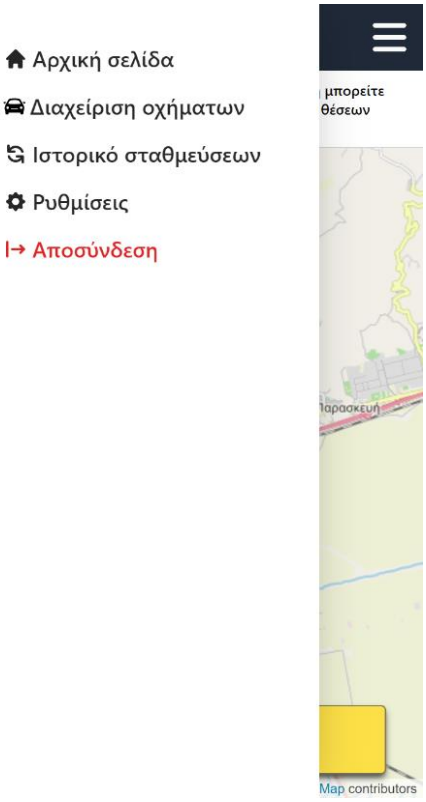
Εικόνα 5.11 Αρχική οθόνη συνδεδεμένου οδηγού (desktop)

Στην Εικόνα 5.13 απεικονίζεται το αναδυόμενο μενού πλοήγησης για του χρήστες κινητών συσκευών.

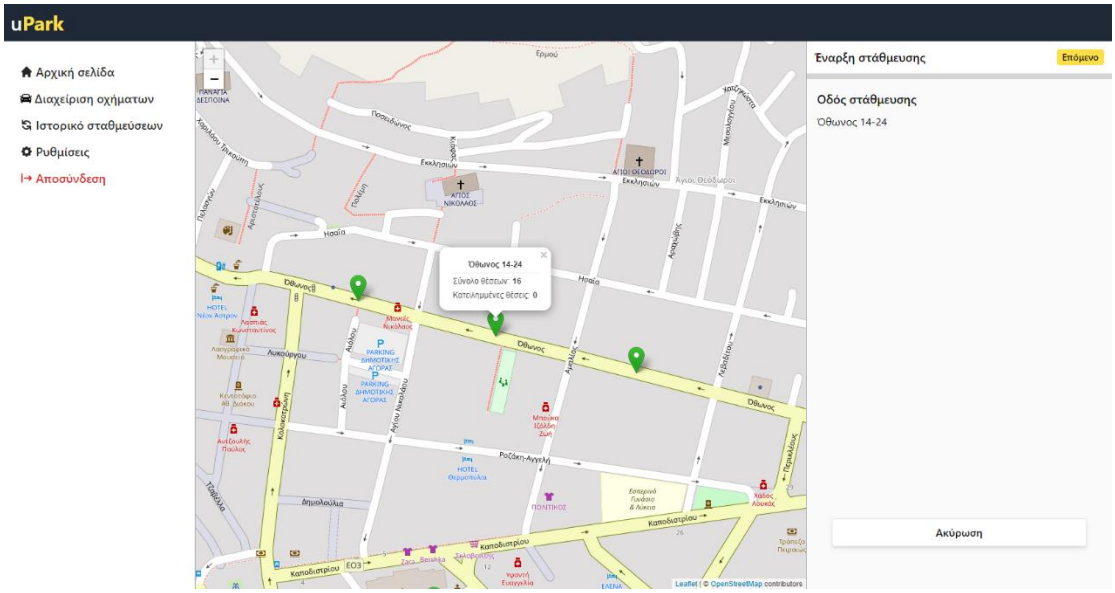
Σε αυτό το σημείο ο οδηγός μπορεί να ενημερωθεί μέσω του διαδραστικού χάρτη, επιλέγοντας ένα από τα διαθέσιμα σημεία, για την διαθεσιμότητα των θέσεων στάθμευσης στα προκαθορισμένα σημεία της πόλης και εν συνεχεία να εκκινήσει την διαδικασία αγοράς χρόνου στάθμευσης.



Εικόνα 5.12 Αρχική οθόνη συνδεδεμένου οδηγού (mobile)



Εικόνα 5.13 Αναδυόμενο μενού

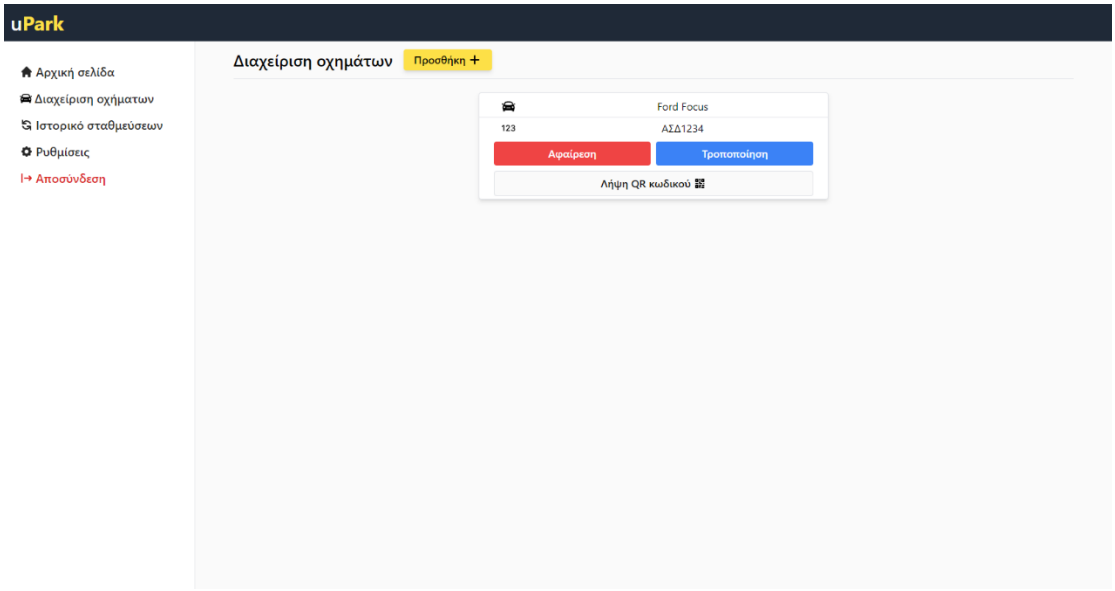


Εικόνα 5.14 Πληροφορίες σημείου στάθμευσης (desktop)

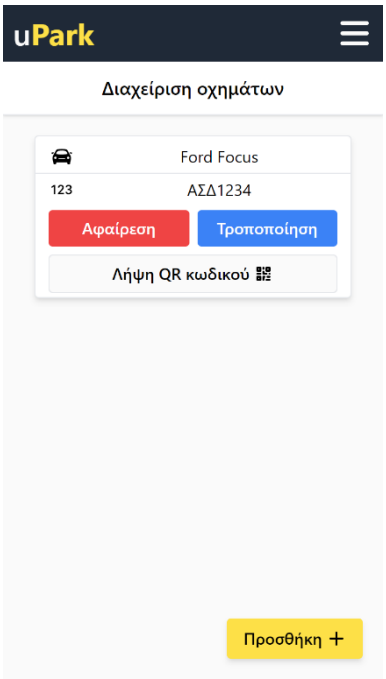


Εικόνα 5.15 Πληροφορίες σημείου στάθμευσης (mobile)

Στους οδηγούς δίνεται επιπλέον η δυνατότητα να αποθηκεύουν πληροφορίες σχετικά τα οχήματά τους, έτσι ώστε να επιταχύνεται η διαδικασία της αγοράς χρόνου στάθμευσης, αφού τους δίνεται η δυνατότητα για άμεση επιλογή οχήματος από τη λίστα αποθηκευμένων οχημάτων, αντί να χρειάζεται να εισάγουν χειροκίνητα τις συγκεκριμένες πληροφορίες κάθε φορά που πραγματοποιούν μια αγορά. Στην Εικόνα 5.16, απεικονίζεται η οθόνη διαχείρισης των αποθηκευμένων οχημάτων ενός οδηγού. Πιο συγκεκριμένα, ο οδηγός μπορεί να προσθέσει ένα νέο όχημα στη λίστα, να τροποποιήσει και να αφαιρέσει ένα όχημα. Επιπλέον για κάθε όχημα δημιουργείται ένας μοναδικός QR κωδικός, ο οποίος μπορεί προαιρετικά να τοποθετηθεί στο όχημα του οδηγού, έτσι ώστε μελλοντικά οι ελεγκτές να μπορούν να ελέγχουν πιο εύκολα και γρήγορα τα σταθμευμένα οχήματα.

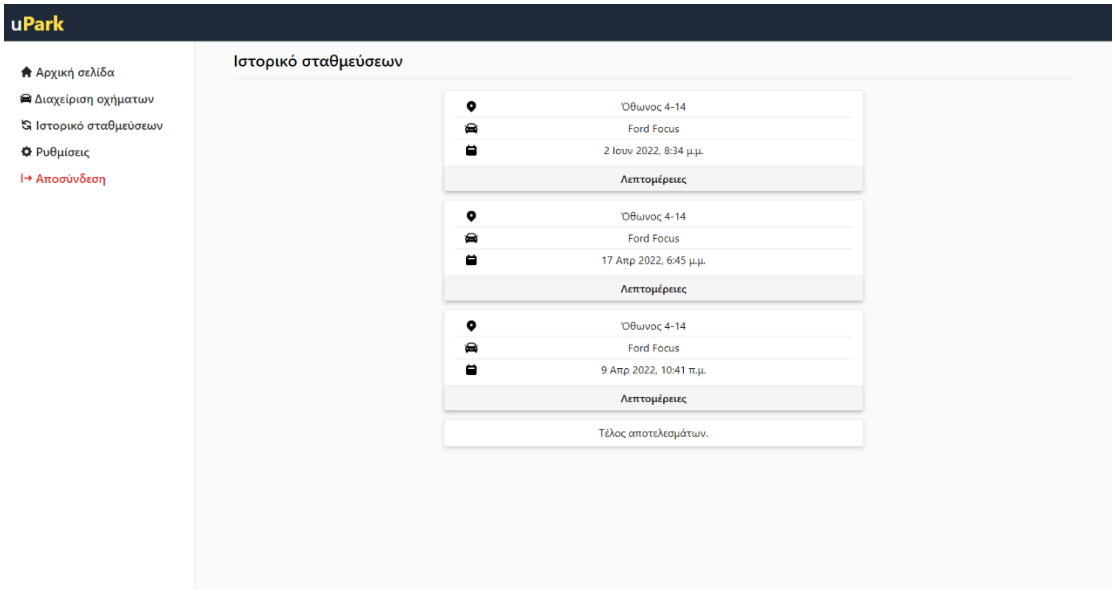


Εικόνα 5.16 Διαχείριση οχημάτων (desktop)

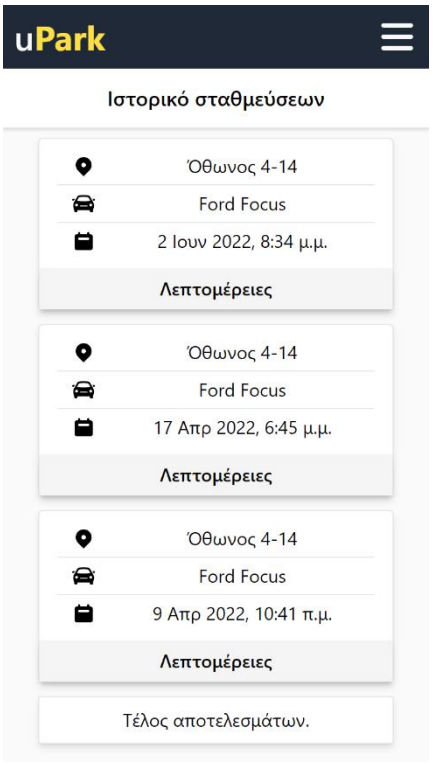


Εικόνα 5.17 Διαχείριση οχημάτων (mobile)

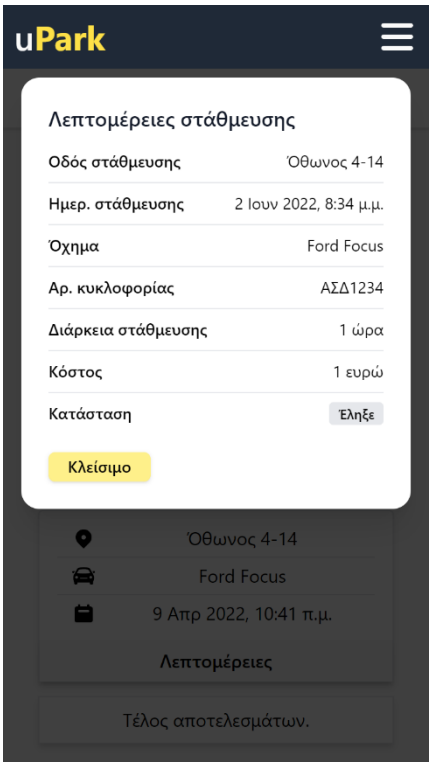
Ένας οδηγός έχει επίσης πρόσβαση σε ένα ιστορικό των παλαιότερων του σταθμεύσεων, το οποίο παρουσιάζεται στις εικόνες 5.18 και 5.19, για χρήστες σταθερών και κινητών συσκευών, αντίστοιχα. Αρχικά, στον χρήστη εμφανίζεται μια επισκόπηση των παλαιότερων του σταθμεύσεων σε χρονολογική φθίνουσα σειρά, αλλά δίνεται επίσης η δυνατότητα εμφάνισης λεπτομερούς περιγραφής μιας συγκεκριμένης εγγραφής από τη λίστα (Εικόνα 5.20).



Εικόνα 5.18 Ιστορικό σταθμεύσεων (desktop)



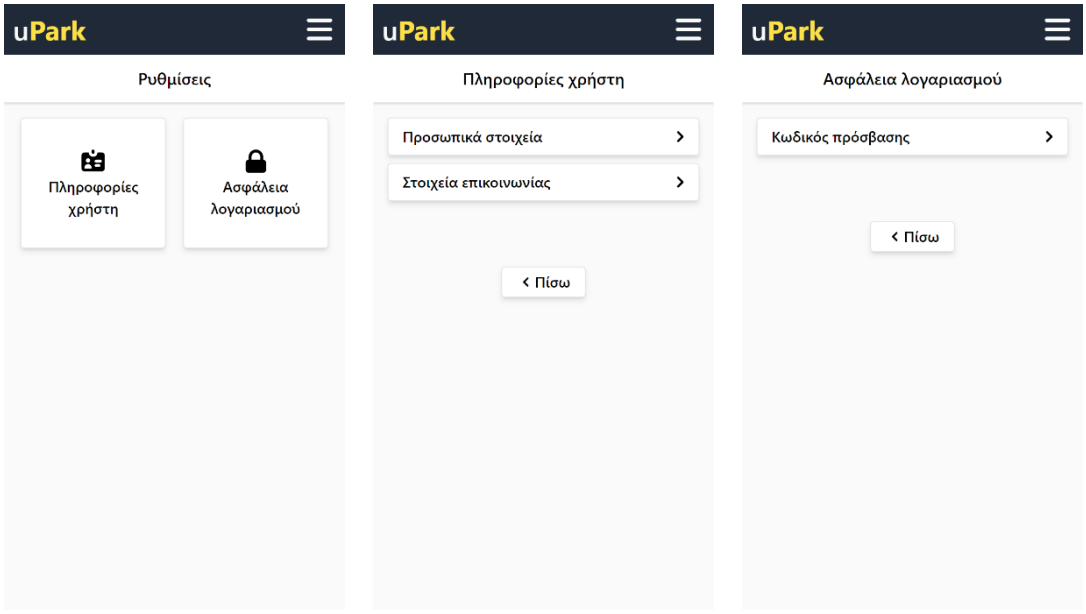
Εικόνα 5.19 Ιστορικό σταθμεύσεων (mobile)



Εικόνα 5.20 Λεπτομέρειες παλαιότερης στάθμευσης

Τέλος στους οδηγούς δίνεται η δυνατότητα επεξεργασίας μερικών στοιχείων του λογαριασμού τους μέσω της οθόνης των ρυθμίσεων, η οποία απεικονίζεται στην Εικόνα 5.21. Οι οδηγοί μπορούν να επεξεργαστούν τα προσωπικά τους στοιχεία (όνομα, επώνυμο), τα

στοιχεία επικοινωνίας τους (διεύθυνση ηλεκτρονικού ταχυδρομείου) καθώς και τον κωδικό πρόσβασης στο λογαριασμό τους.

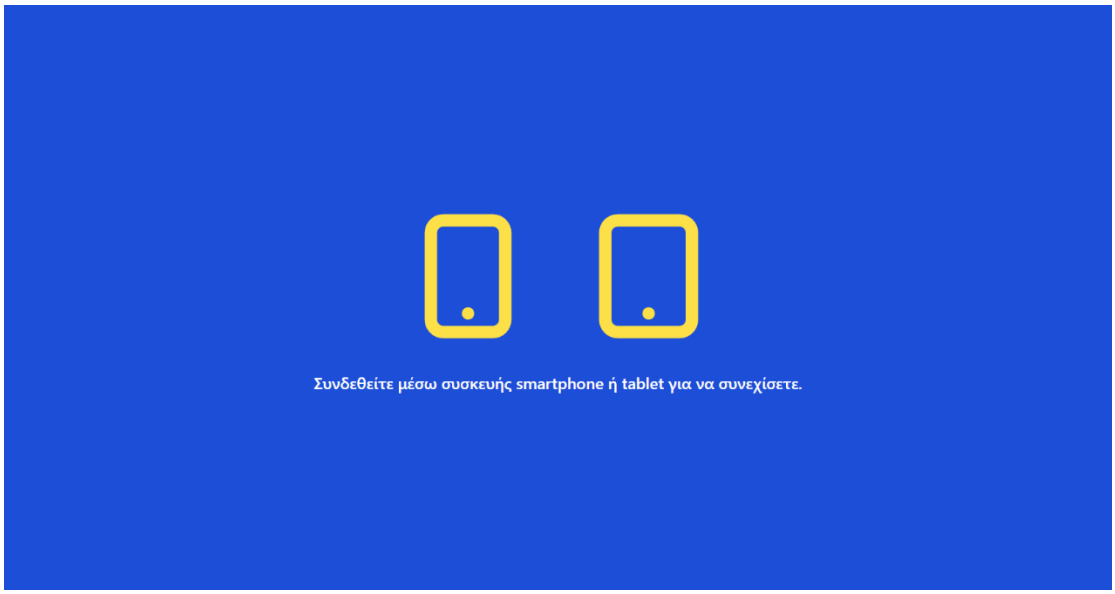


Ρυθμίσεις *Ρυθμίσεις > Πληροφορίες χρήστη* *Ρυθμίσεις > Ασφάλεια λογαριασμού*

Εικόνα 5.21 Ρυθμίσεις λογαριασμού οδηγού (mobile)

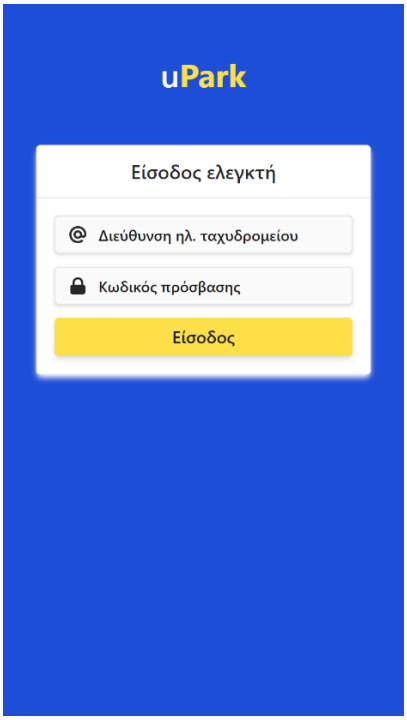
5.2.2 Υποσύστημα ελεγκτή

Όταν ένας ελεγκτής του συστήματος προσπαθήσει να συνδεθεί μέσω ενός σταθερού υπολογιστή, τότε του εμφανίζεται η οθόνη της Εικόνα 5.22.



Εικόνα 5.22 Ενημέρωση ελεγκτή

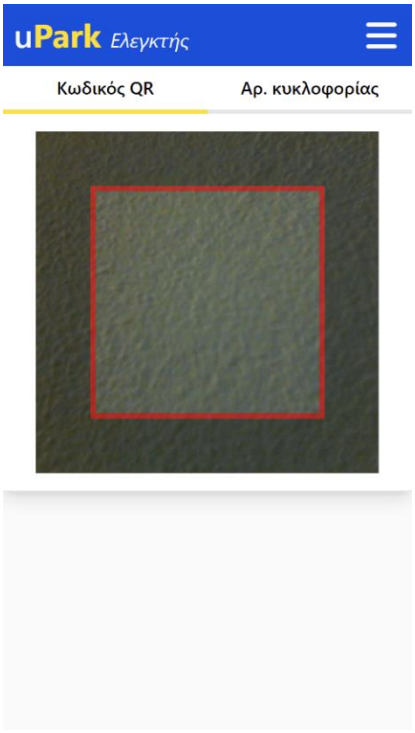
Όταν ένας ελεγκτής επισκέπτεται την εφαρμογή, ανακατευθύνεται στην αρχική οθόνη του συστήματος η οποία αποτελείται από μια φόρμα εισόδου, στιγμιότυπο της οποίας παρουσιάζεται στην οθόνη της Εικόνα 5.23.



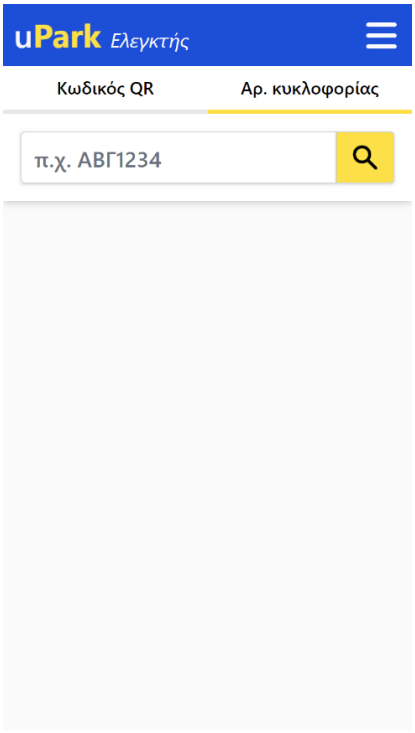
Εικόνα 5.23 Οθόνη εισόδου ελεγκτή

Εάν η σύνδεση ενός ελεγκτή πραγματοποιηθεί με επιτυχία, τότε μεταφέρεται στην οθόνη της Εικόνα 5.24, η οποία αποτελείται από δύο καρτέλες, που αντιστοιχούν στους δύο τρόπους με τους οποίους ο ελεγκτής μπορεί να ελέγξει εάν υπάρχει ενεργή κάρτα χρόνου στάθμευσης για ένα όχημα σταθμευμένο σε θέση ελεγχόμενης στάθμευσης. Ο πρώτος τρόπος απαιτεί τη χρήση ενός QR αναγνώστη και θέτει ως προϋπόθεση ένας οδηγός να έχει τοποθετημένο στο όχημα του τον QR κωδικό που σχετίζεται με το συγκεκριμένο όχημα και δημιουργείται αυτόματα κατά την προσθήκη ενός οχήματος από έναν οδηγό. Παράδειγμα ενός τέτοιου QR κωδικού φαίνεται στην Εικόνα 3.1. Ο δεύτερος τρόπος ελέγχου, της δεύτερης καρτέλας (Εικόνα 5.25), απαιτεί από έναν ελεγκτή να πληκτρολογήσει τα στοιχεία της πινακίδας (δηλ., τον αριθμό κυκλοφορίας) του σταθμευμένου οχήματος.

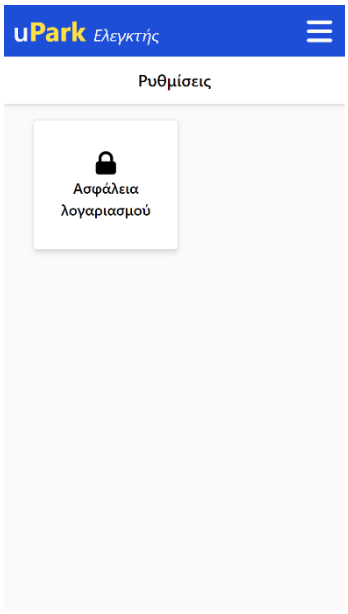
Στους ελεγκτές δίνεται επίσης η δυνατότητα επεξεργασίας του κωδικού ασφαλείας του λογαριασμού τους μέσω της οθόνης των ρυθμίσεων, η οποία απεικονίζεται στην Εικόνα 5.26.



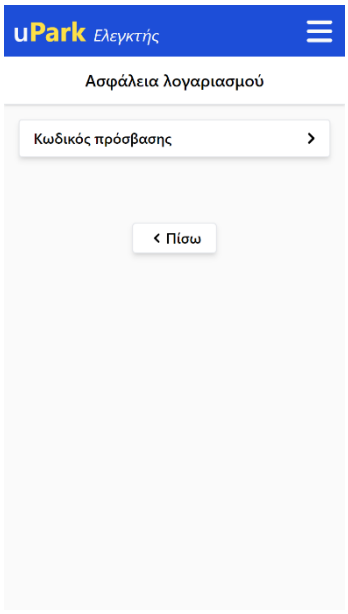
Εικόνα 5.24 Αρχική οθόνη (καρτέλα QR αναγνώστη)



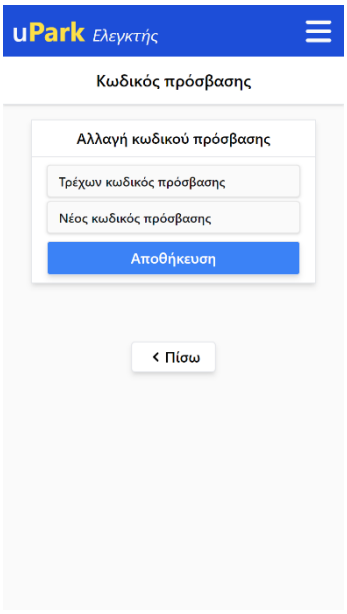
Εικόνα 5.25 Αρχική οθόνη (καρτέλα εισόδου αριθμού κυκλοφορίας)



Ρυθμίσεις



Ρυθμίσεις > Ασφάλεια λογαριασμού

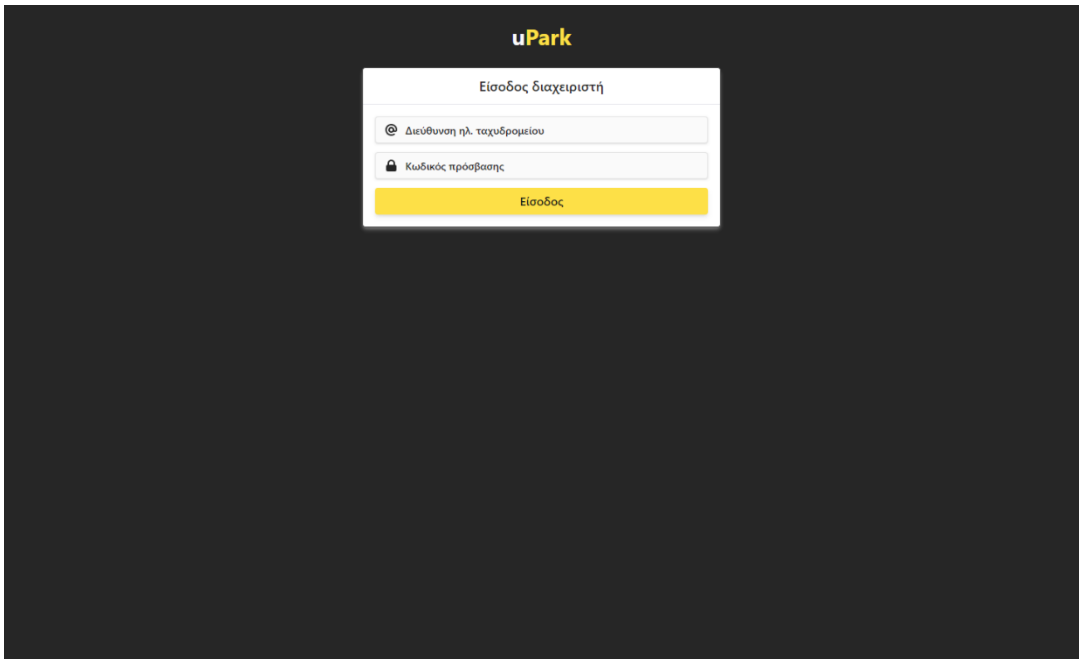


Ρυθμίσεις > Ασφάλεια λογαριασμού > Κωδικός πρόσβασης

Εικόνα 5.26 Ρυθμίσεις ελεγκτή

5.2.3 Υποσύστημα διαχειριστή

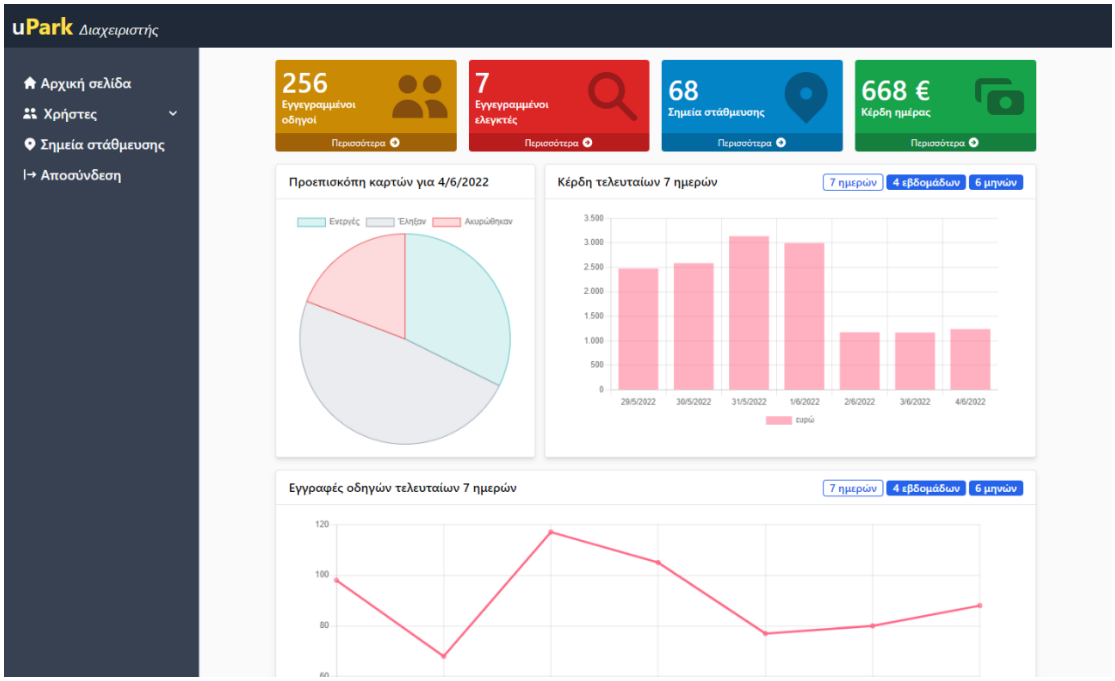
Όταν ένας διαχειριστής προσπαθήσει να συνδεθεί στο σύστημα διαχείρισης της εφαρμογής uPark, τότε του εμφανίζεται η οθόνη εισόδου που απεικονίζεται στην Εικόνα 5.27. Εισάγοντας τα κατάλληλα στοιχεία στη φόρμα εισόδου, γίνεται ταυτοποίηση του διαχειριστή και ανακατευθύνεται στην αρχική οθόνη της Εικόνα 5.28. Στην αρχική οθόνη, υπάρχουν διαθέσιμες διάφορες πληροφορίες σχετικά με τους χρήστες του συστήματος, τα έσοδα κ.ά, τόσο σε μορφή απλού κειμένου, όσο και σε διαγραμματικές αναπαραστάσεις, έτσι ώστε να παρέχεται στο διαχειριστή μια ολοκληρωμένη εικόνα με βάση την οποία να μπορεί να προβεί σε μελλοντικές ενέργειες.



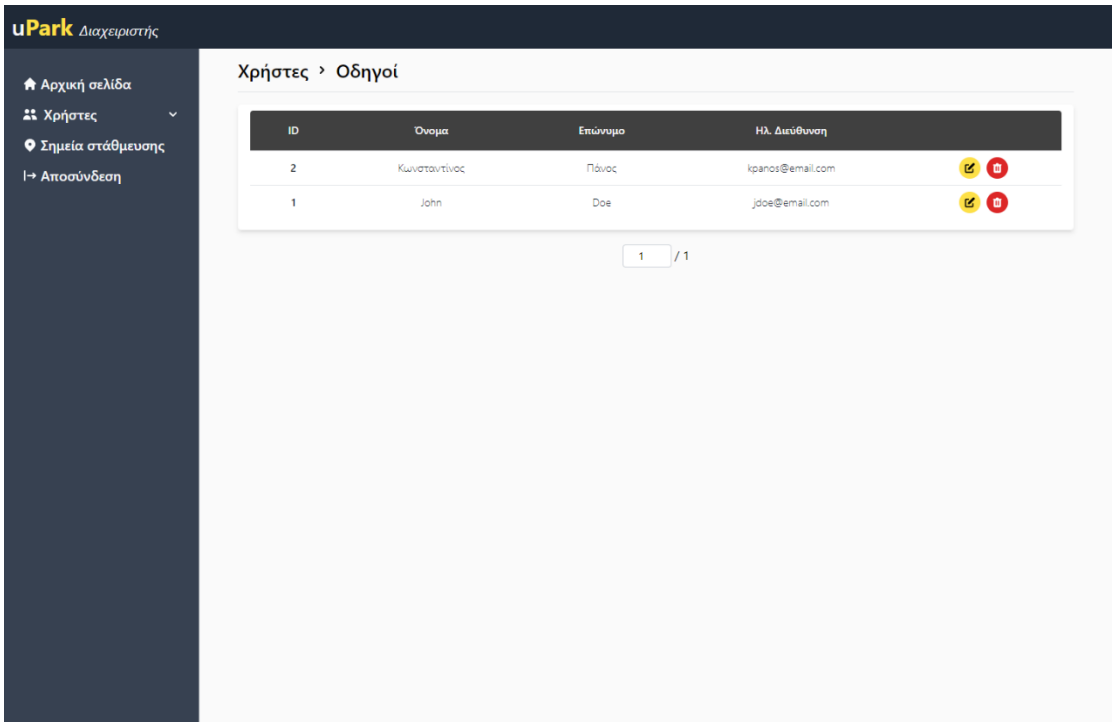
Εικόνα 5.27 Οθόνη εισόδου ελεγκτή

Ένας διαχειριστής, μπορεί να επεξεργάζεται όλους του χρήστες που είναι εγγεγραμμένοι στο σύστημα (οδηγούς και ελεγκτές). Πιο συγκεκριμένα, όπως φαίνεται στην Εικόνα 5.29, ο διαχειριστής έχει πρόσβαση σε μια λίστα με όλους τους εγγεγραμμένους οδηγούς, ενώ επιπλέον έχει τη δυνατότητα αφαίρεσης κάποιου οδηγού από το σύστημα. Ο διαχειριστής μπορεί επίσης να επεξεργάζεται και τους εγγεγραμμένους ελεγκτές. Στην οθόνη διαχείρισης ελεγκτών (Εικόνα 5.30), παρέχεται στο διαχειριστή μια λίστα με όλους τους εγγεγραμμένους ελεγκτές. Για ένα συγκεκριμένο ελεγκτή, ένας διαχειριστής μπορεί να επεξεργαστεί τις πληροφορίες του (όνομα, ηλεκτρονική διεύθυνση, επαναφορά κωδικού πρόσβασης) ή να προβεί σε αφαίρεσή του από το σύστημα.

Επιπλέον, ο διαχειριστής μπορεί να πραγματοποιήσει προσθήκη ενός νέου ελεγκτή.

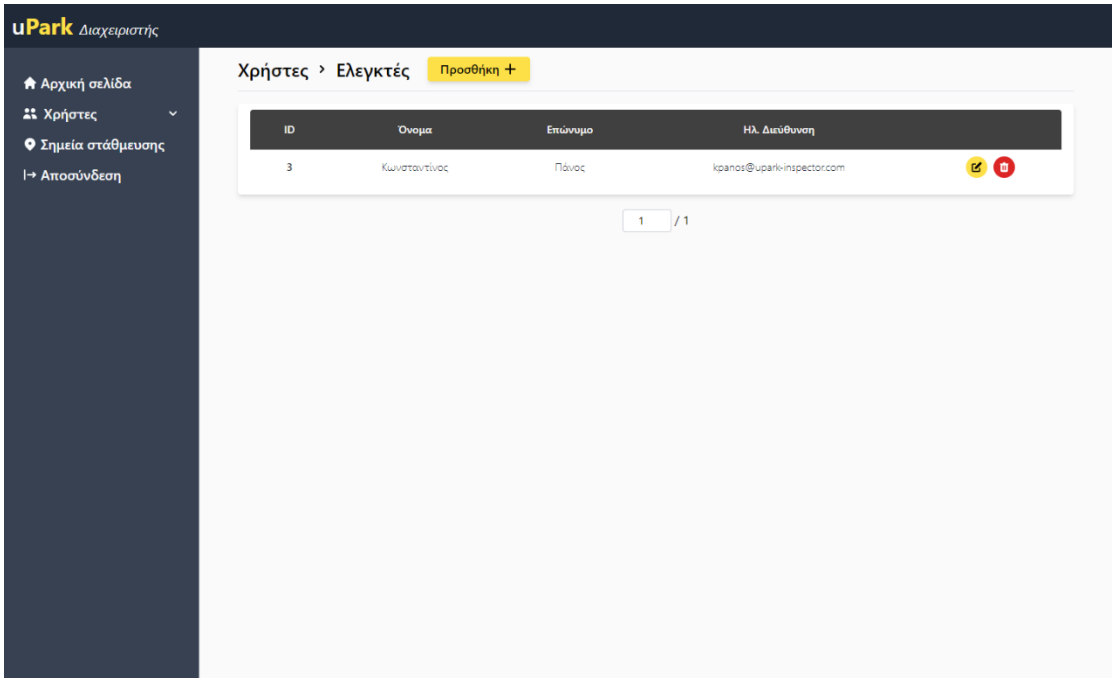


Εικόνα 5.28 Αρχική οθόνη διαχειριστή

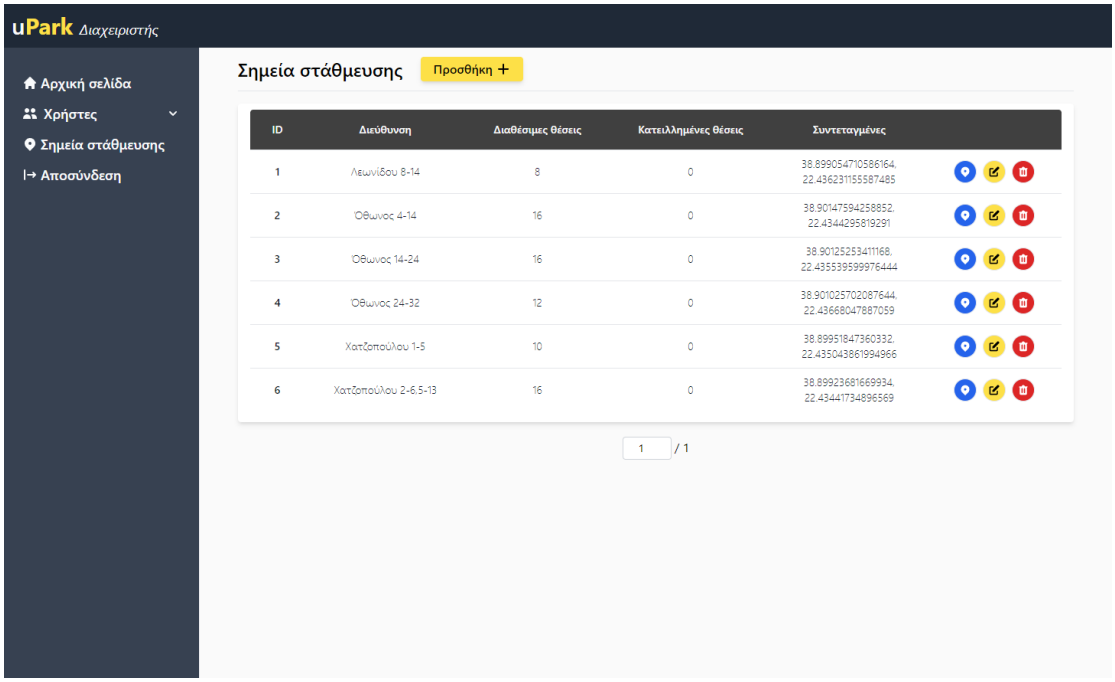


Εικόνα 5.29 Οθόνη διαχείρισης οδηγών

Ο διαχειριστής, είναι υπεύθυνος για την προσθήκη, αφαίρεση ή επεξεργασία των σημείων στάθμευσης που είναι διαθέσιμα στους οδηγούς. Όλες οι απαιτούμενες σχετικές λειτουργίες είναι διαθέσιμες στην οθόνη που απεικονίζεται στην Εικόνα 5.31.



Εικόνα 5.30 Οθόνη διαχείρισης ελεγκτών



Εικόνα 5.31 Οθόνη διαχείρισης σημείων στάθμευσης

Στην οθόνη διαχείρισης σημείων στάθμευσης (Εικόνα 5.31), ο διαχειριστής συναντά μια λίστα με τα ήδη υπάρχοντα σημεία στάθμευσης. Για κάθε στοιχείο της λίστας, ο διαχειριστής μπορεί να δει μια προεπισκόπηση του σημείου στάθμευσης σε χάρτη, να επεξεργαστεί τις πληροφορίες για το σημείο αυτό και να το αφαιρέσει από τη βάση δεδομένων. Ο διαχειριστής μπορεί επίσης να προσθέσει νέα σημεία στάθμευσης στη βάση δεδομένων του συστήματος.

6 Παρουσίαση περιπτώσεων χρήσης του συστήματος uPark

Στο κεφάλαιο αυτό, θα παρουσιάσουμε ενδεικτικά σενάρια χρήσης του συστήματος, τα οποία ένας χρήστης συχνά θα πραγματοποιεί κατά τη χρήση του uPark.

6.1 Αγορά χρόνου στάθμευσης σε κινητή συσκευή (Οδηγός)

Στη συνέχεια παρουσιάζουμε τη διαδικασία αγοράς χρόνου στάθμευσης από έναν οδηγό ο οποίος συνδέεται στην εφαρμογή μέσω της κινητής του συσκευής. Για την παρουσίαση αυτής της διαδικασίας, γίνονται οι εξής δύο υποθέσεις:

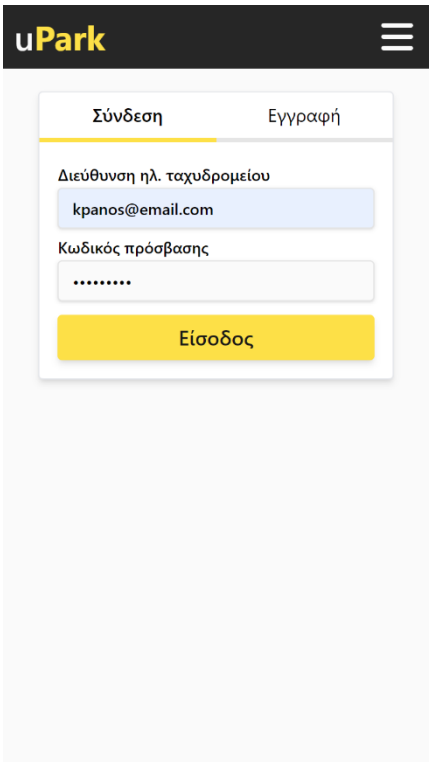
1. Η οπίσθια κάμερα της κινητής συσκευής του οδηγού είναι λειτουργική
2. Στο σημείο στάθμευσης του οχήματος του οδηγού, υπάρχει τοποθετημένη από κατάλληλο φορέα πινακίδα σήμανσης σαν αυτή της Εικόνα 6.1, η οποία φέρει τον QR κωδικό τον οποίο ο οδηγός στη συνέχεια καλείται να σκανάρει μέσω της εφαρμογής του.



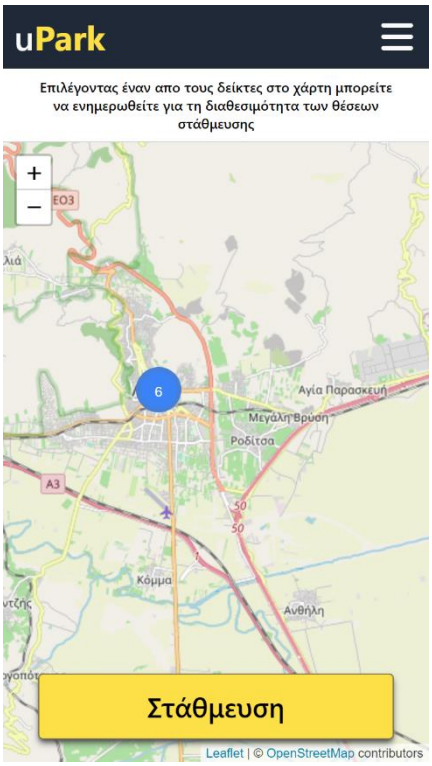
Εικόνα 6.1 Ενδεικτική πινακίδα ελεγχόμενης στάθμευσης με τον QR κωδικό του σημείου στάθμευσης

Αρχικά ο οδηγός συνδέεται στο λογαριασμό του εισάγοντας τα στοιχεία σύνδεσής του στη φόρμα εισόδου της Εικόνα 6.2, και ανακατευθύνεται στην αρχική οθόνη της εφαρμογής (Εικόνα 6.3), όπου επιλέγοντας «Στάθμευση» στο κάτω μέρος της οθόνης, εκκινεί τη διαδικασία αγοράς χρόνου στάθμευσης για το όχημα του. Στη συνέχεια ο οδηγός καλείται να σκανάρει τον κωδικό QR του σημείου στάθμευσής του, μέσω του QR αναγνώστη που του παρέχεται από την εφαρμογή (Εικόνα 6.4). Εναλλακτικά, σε περίπτωση αδυναμίας

ανάγνωσης του QR κωδικού, για παράδειγμα λόγω χαμηλού φωτισμού, ένας οδηγός μπορεί να πληκτρολογήσει τον κωδικό του σημείου στάθμευσής του, μέσω της φόρμας της Εικόνα 6.5.

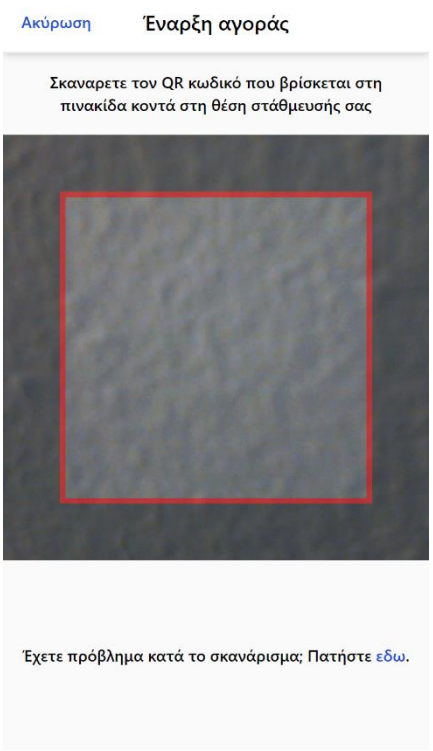


Εικόνα 6.2 Είσοδος οδηγού στην εφαρμογή

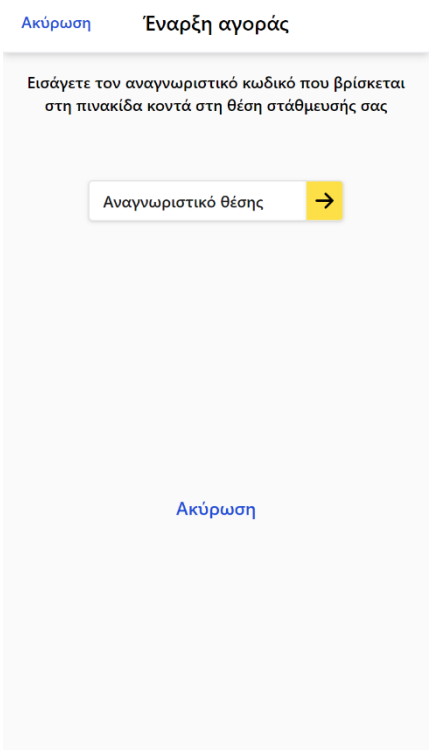


Εικόνα 6.3 Αρχική οθόνη εφαρμογής

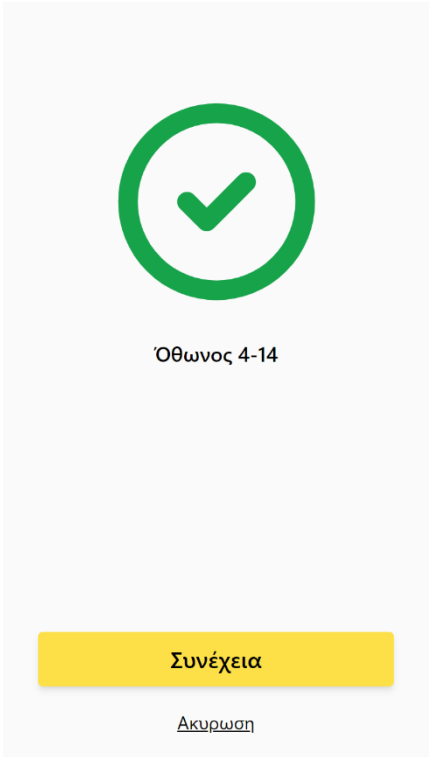
Μετά την επιτυχή ανάγνωση του QR κωδικού (ή εναλλακτικά της πληκτρολόγησης έγκυρου κωδικού σημείου στάθμευσης) εμφανίζεται μήνυμα επιτυχούς ανεύρεσης της οδού, το οποίο φαίνεται στην Εικόνα 6.6.



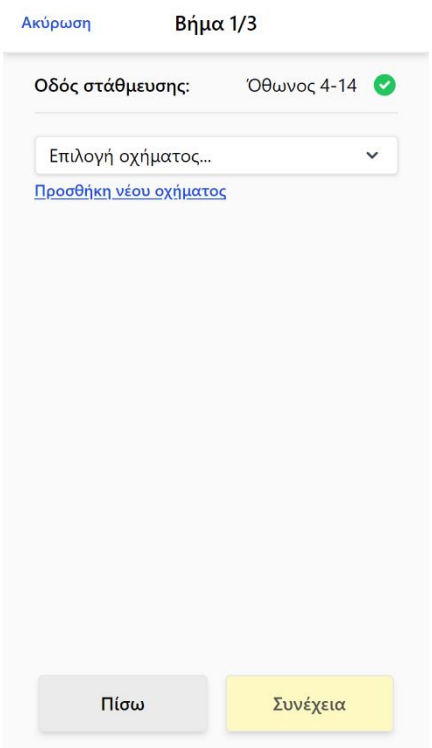
Εικόνα 6.4 Αναγνώστης κωδικού QR



Εικόνα 6.5 Φόρμα εισόδου κωδικού σημείου στάθμευσης

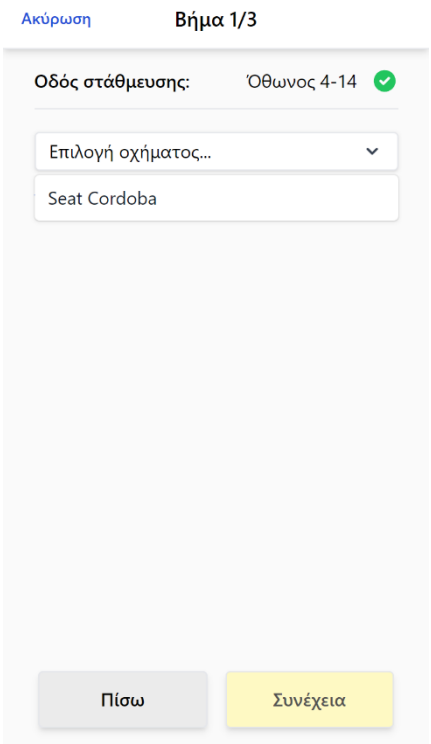


Εικόνα 6.6 Οθόνη επιτυχούς ανεύρεσης σημείου στάθμευσης

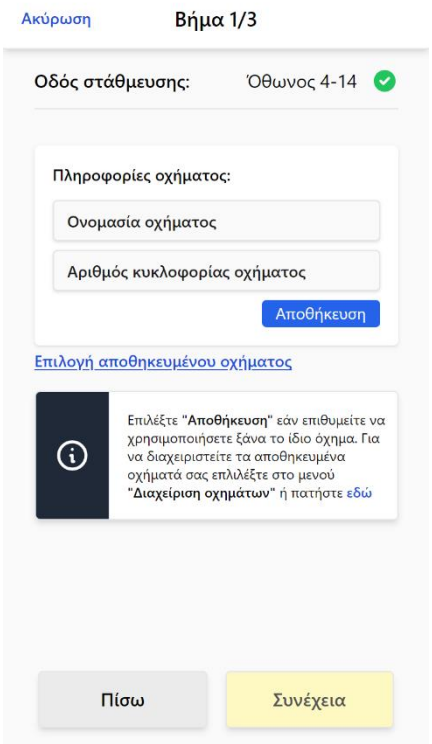


Εικόνα 6.7 Οθόνη επιλογής οχήματος

Στο επόμενο βήμα της διαδικασίας ζητείται από τον οδηγό να δηλώσει το όχημα το οποίο σταθμεύει. Αυτό επιτυγχάνεται είτε μέσω της επιλογής από λίστα, ενός από τα ήδη αποθηκευμένα οχήματα ενός οδηγού (Εικόνα 6.8), είτε μέσω της προσθήκης ενός διαφορετικού οχήματος μέσω της αντίστοιχης φόρμας εισόδου (Εικόνα 6.9). Έπειτα, ο οδηγός πρέπει να δηλώσει τη χρονική διάρκεια της στάθμευσής του. Η διαθέσιμη χρονική διάρκεια εκτείνεται από 30 λεπτά μέχρι 5 ώρες, με ενδιάμεσο βήμα αύξησης τα 30 λεπτά. Στη συνέχεια, γίνεται μια σύνοψη των πληροφοριών που έχουν εισαχθεί (Εικόνα 6.11) και επιλέγοντας «Συνέχεια», ο οδηγός μεταφέρεται στην οθόνη πληρωμής (Εικόνα 6.12). Ο οδηγός έχει δύο επιλογές. Μπορεί είτε να εισάγει τα στοιχεία της τραπεζικής του κάρτας και να κάνει πληρωμή, είτε να κάνει χρήση του υπολειπόμενου χρόνου του από προηγούμενες ακυρωμένες κάρτες στάθμευσης, εάν αυτός ισούται με διάρκεια μεγαλύτερη του επιθυμητού χρόνου στάθμευσης.

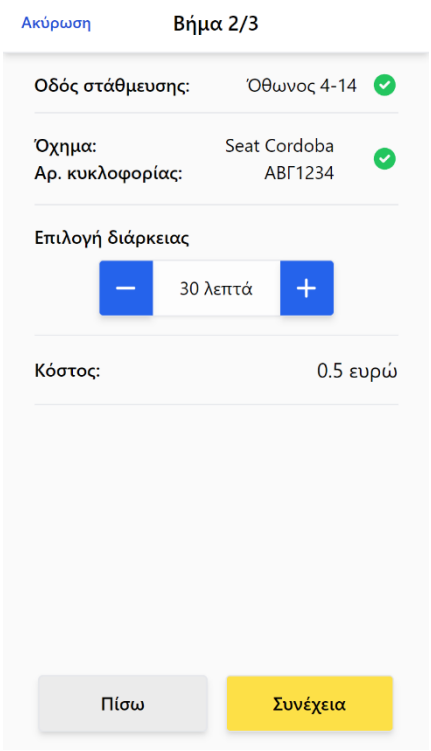


Εικόνα 6.8 Επιλογή οχήματος από τη λίστα αποθηκευμένων οχημάτων

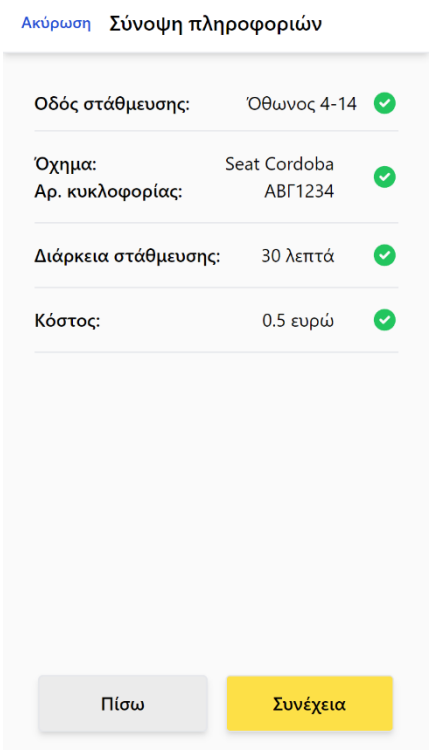


Εικόνα 6.9 Προσθήκη νέου οχήματος

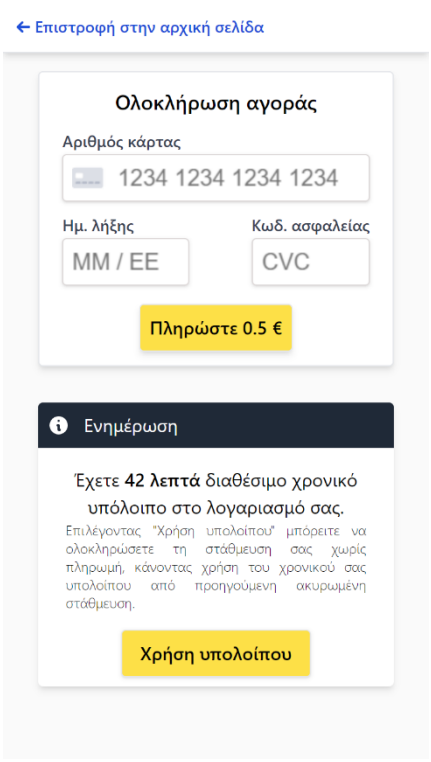
Επιλέγοντας έναν από του δύο τρόπους, η διαδικασία αγοράς ολοκληρώνεται επιτυχώς και στον οδηγό εμφανίζεται ένα μήνυμα επιτυχούς αγοράς (Εικόνα 6.13).



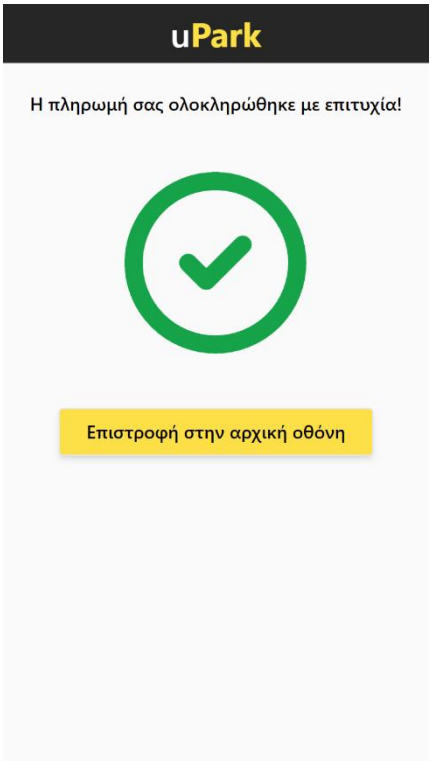
Εικόνα 6.10 Επιλογή χρονικής διάρκειας στάθμευσης



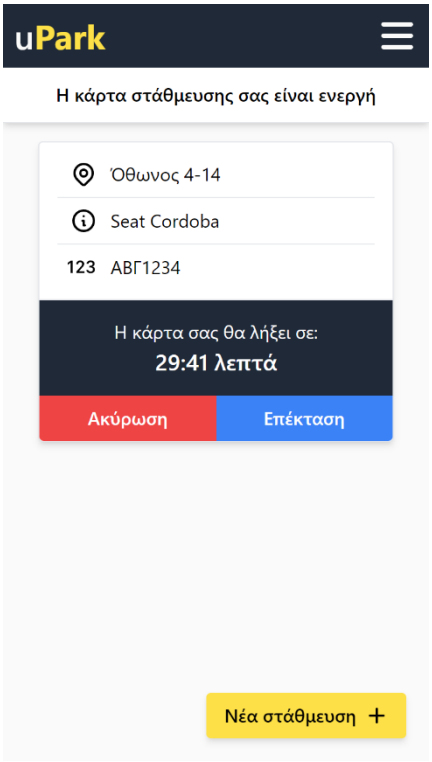
Εικόνα 6.11 Οθόνη σύνοψης πληροφοριών κάρτας στάθμευσης



Εικόνα 6.12 Οθόνη ολοκλήρωσης αγοράς



Εικόνα 6.13 Μήνυμα επιτυχούς αγοράς



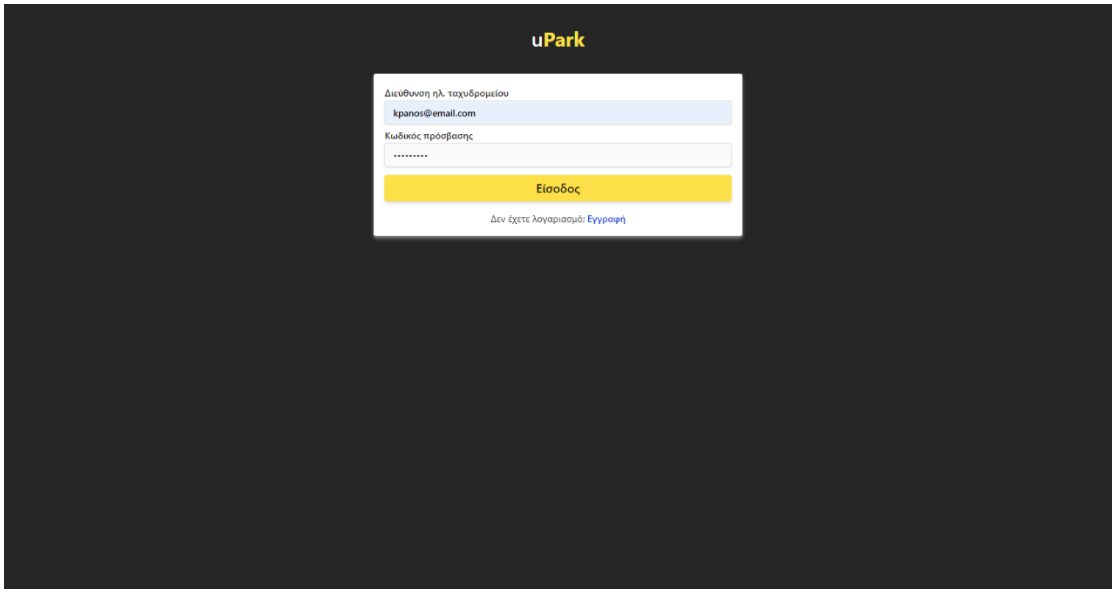
Εικόνα 6.14 Προεπισκόπηση ενεργής κάρτας στάθμευσης στην αρχική σελίδα

Επιστρέφοντας στην αρχική οθόνη της εφαρμογής, ο οδηγός μπορεί να δει πληροφορίες για την «κάρτα» χρόνου στάθμευσης που μόλις αγόρασε (Εικόνα 6.14), ενώ ακόμη του δίνεται τόσο η δυνατότητα επέκτασης του υπολειπόμενου χρόνου, όσο και η δυνατότητα πρόωρης λήξης της κάρτας, μεταφέροντας τον υπολειπόμενο χρόνο της για μελλοντική χρήση.

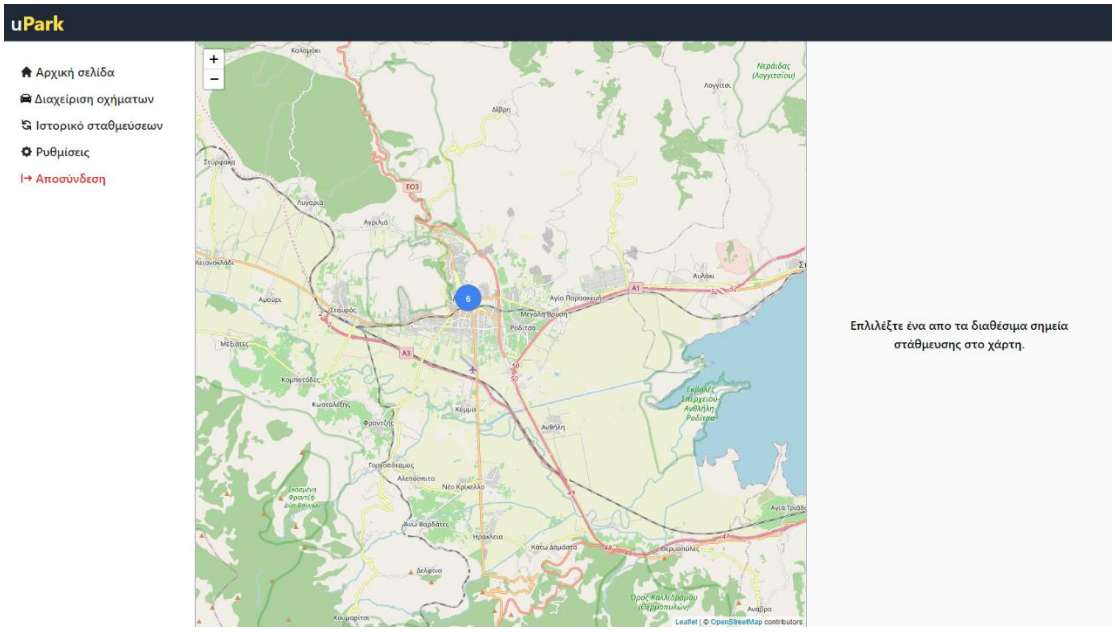
6.2 Αγορά χρόνου στάθμευσης σε σταθερό υπολογιστή (Οδηγός)

Στη συνέχεια θα παρουσιάσουμε τη διαδικασία αγοράς χρόνου στάθμευσης από έναν οδηγό ο οποίος συνδέεται στην εφαρμογή μέσω ενός σταθερού υπολογιστή, μια διαδικασία που εκτός από μερικές μικρές διαφορές είναι όμοια με αυτήν της αγοράς μέσω κινητών συσκευών.

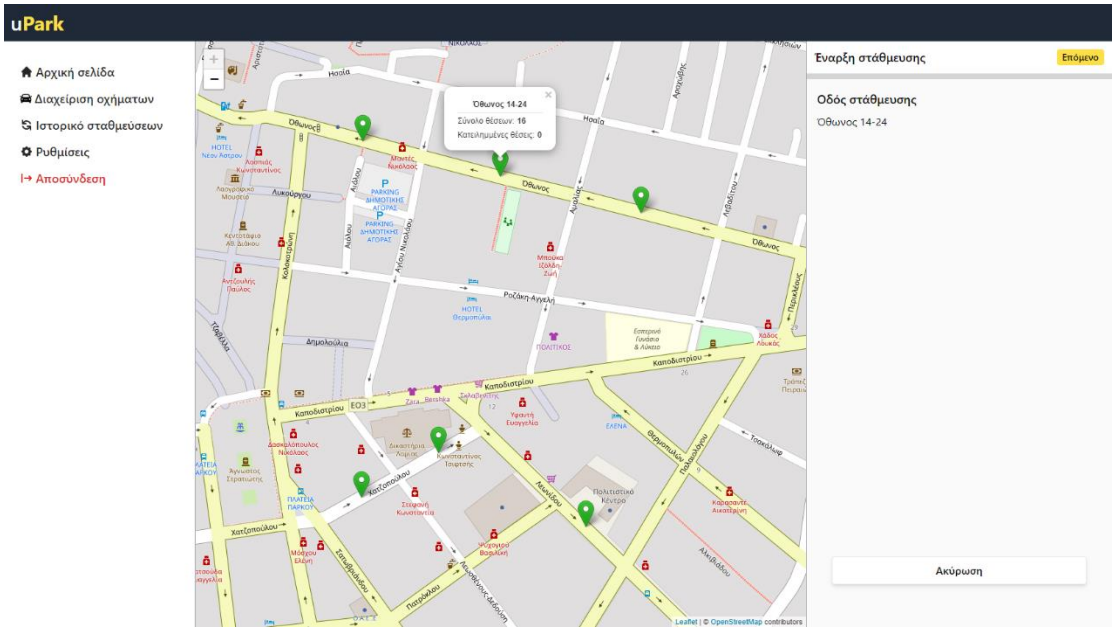
Αρχικά ο οδηγός συνδέεται στο λογαριασμό του εισάγοντας τα στοιχεία σύνδεσής του στη φόρμα εισόδου της Εικόνα 6.15, και ανακατευθύνεται στην αρχική οθόνη της εφαρμογής (Εικόνα 6.16). Για να ξεκινήσει η διαδικασία αγοράς της κάρτας χρόνου στάθμευσης, ένας οδηγός πρέπει να επιλέξει ένα από τα διαθέσιμα σημεία στάθμευσης που εμφανίζονται στον χάρτη της αρχικής οθόνης, κάνοντας κλικ πάνω σε αυτό. Τότε, η δεξιά στήλη της αρχικής οθόνης ανανεώνεται, αναδεικνύοντας το επιλεγμένο σημείο στάθμευσης και προσφέροντας επιλογές για συνέχιση ή ακύρωση της διαδικασίας αγοράς.



Εικόνα 6.15 Είσοδος οδηγού στην εφαρμογή

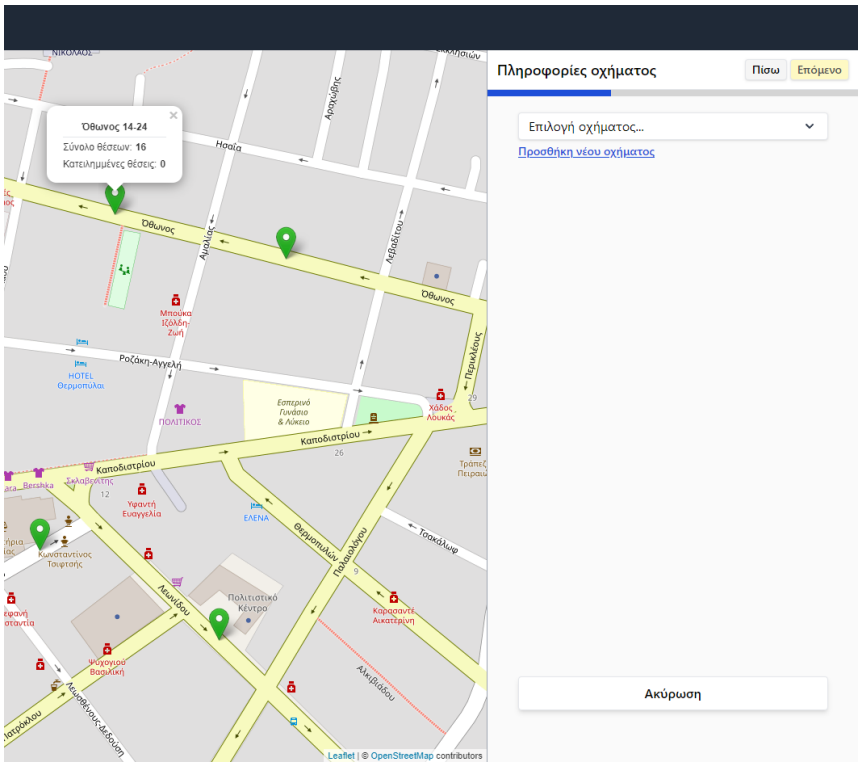


Εικόνα 6.16 Αρχική οθόνη εφαρμογής

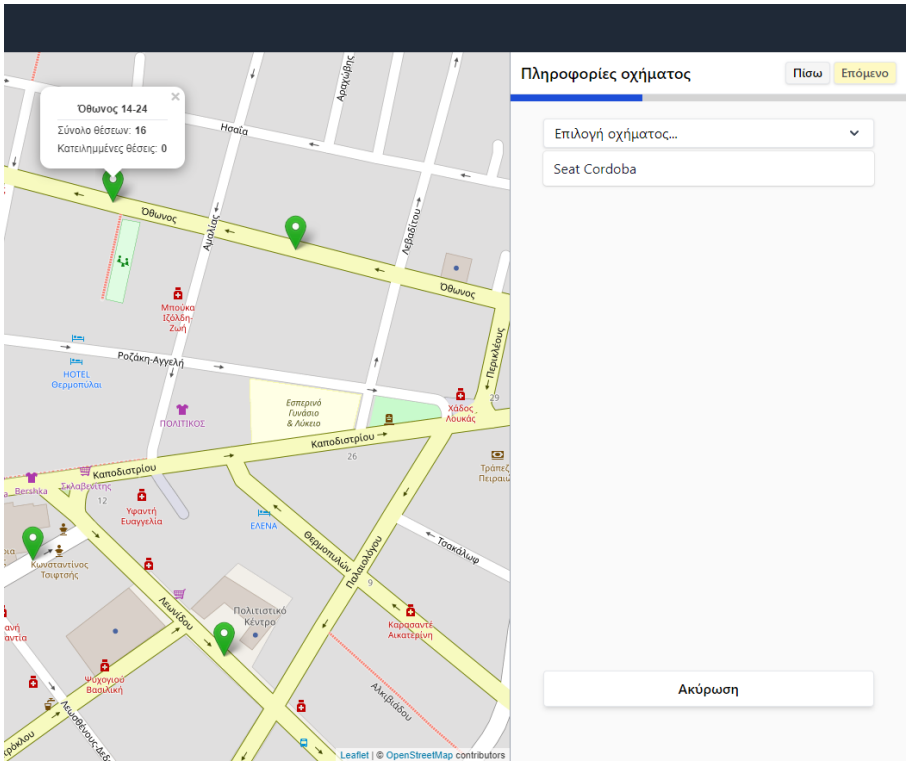


Εικόνα 6.17 Επιλογή σημείου στάθμευσης

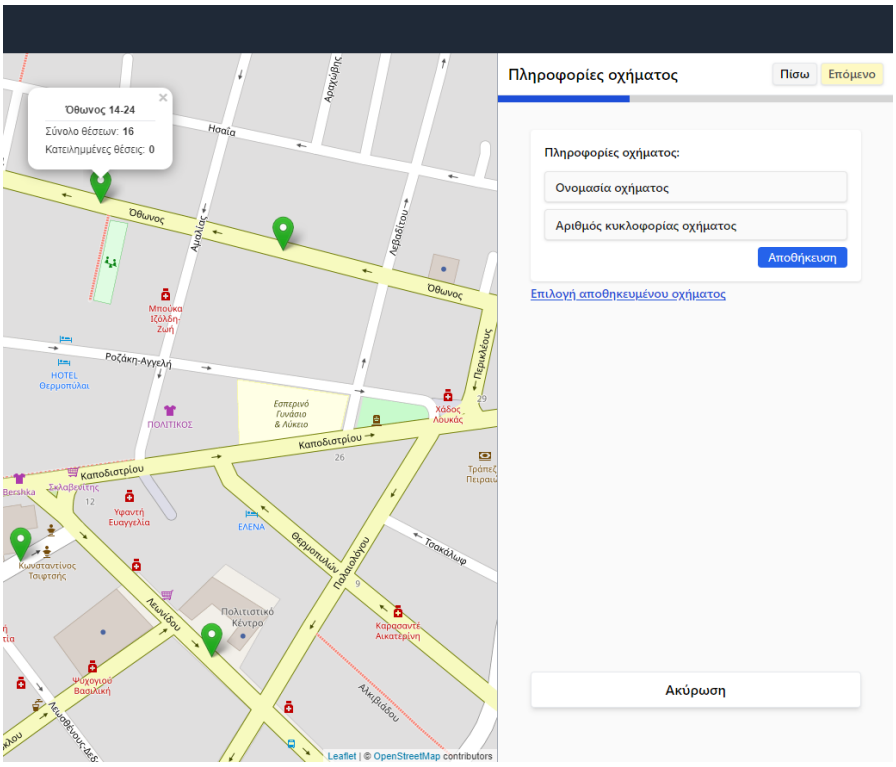
Μετά την επιλογή σημείου στάθμευσης, ο οδηγός επιλέγοντας «Επόμενο» στο επάνω δεξιά μέρος της Εικόνα 6.17, μεταφέρεται στο επόμενο βήμα της διαδικασίας, αυτό της εισαγωγής των πληροφοριών του οχήματος που σταθμεύει (Εικόνα 6.18), η οποία πραγματοποιείται είτε μέσω επιλογής από λίστα για ήδη αποθηκευμένο όχημα (Εικόνα 6.19), είτε μέσω πληκτρολόγησης των πληροφοριών στη φόρμα της Εικόνα 6.20.



Εικόνα 6.18 Εισαγωγή πληροφοριών οχήματος

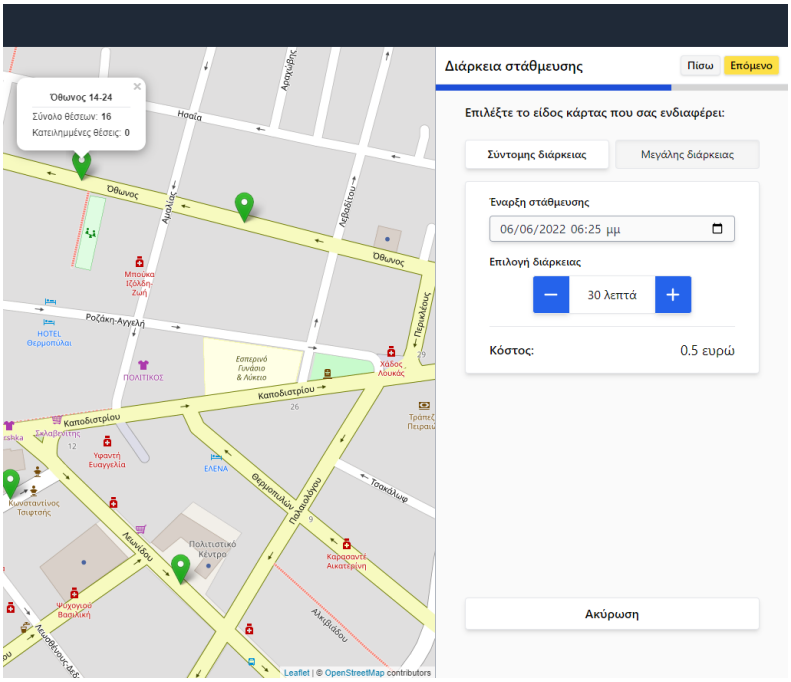


Εικόνα 6.19 Επιλογή οχήματος από τη λίστα αποθηκευμένων οχημάτων



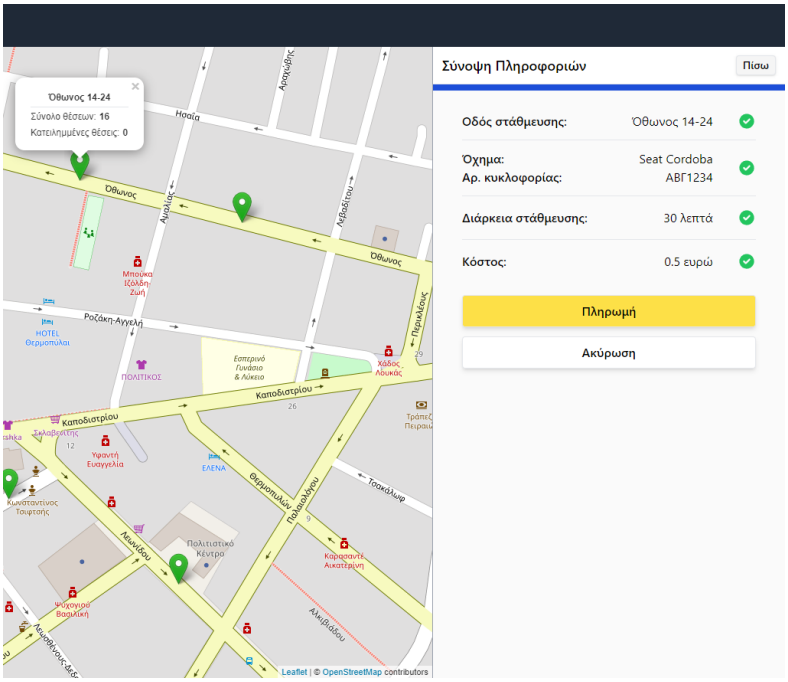
Εικόνα 6.20 Προσθήκη νέου οχήματος

Έπειτα, ο οδηγός πρέπει να δηλώσει τη χρονική διάρκεια της στάθμευσης του καθώς και την χρονική στιγμή ενεργοποίησής της (Εικόνα 6.21).

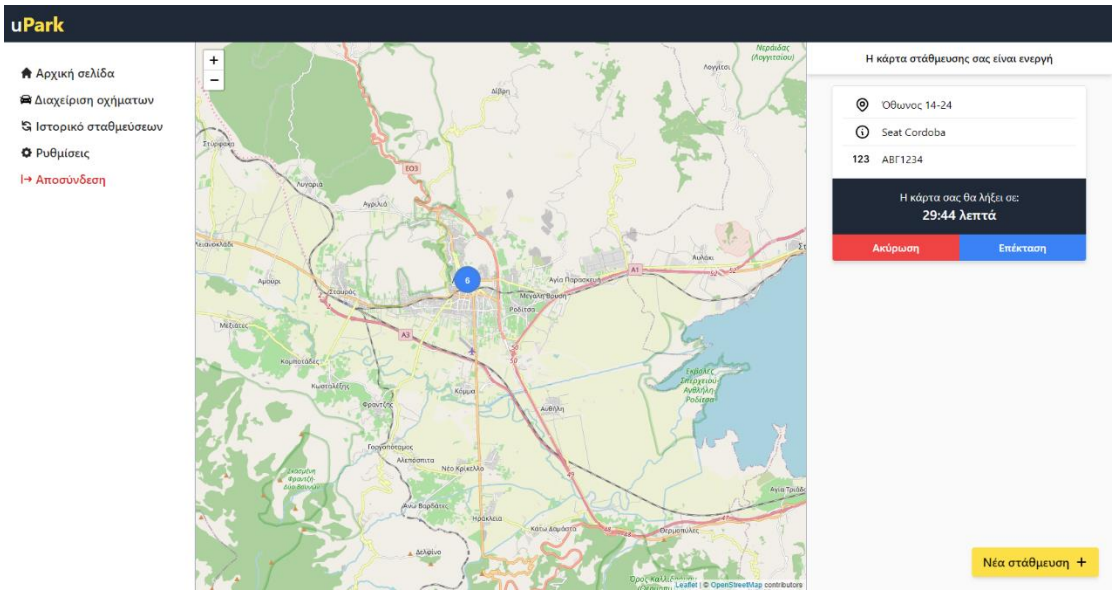


Εικόνα 6.21 Επιλογή χρονικής διάρκειας στάθμευσης

Στη συνέχεια, γίνεται μια σύνοψη των πληροφοριών που έχουν εισαχθεί (Εικόνα 6.22) και επιλέγοντας «Πληρωμή», ο οδηγός μεταφέρεται στην οθόνη πληρωμής (Εικόνα 6.12). Η διαδικασία πληρωμής είναι ίδια με αυτή που περιγράφτηκε για την περίπτωση σύνδεσης οδηγού στο uPark από κινητή συσκευή.



Εικόνα 6.22 Σύνοψη πληροφοριών κάρτας στάθμευσης



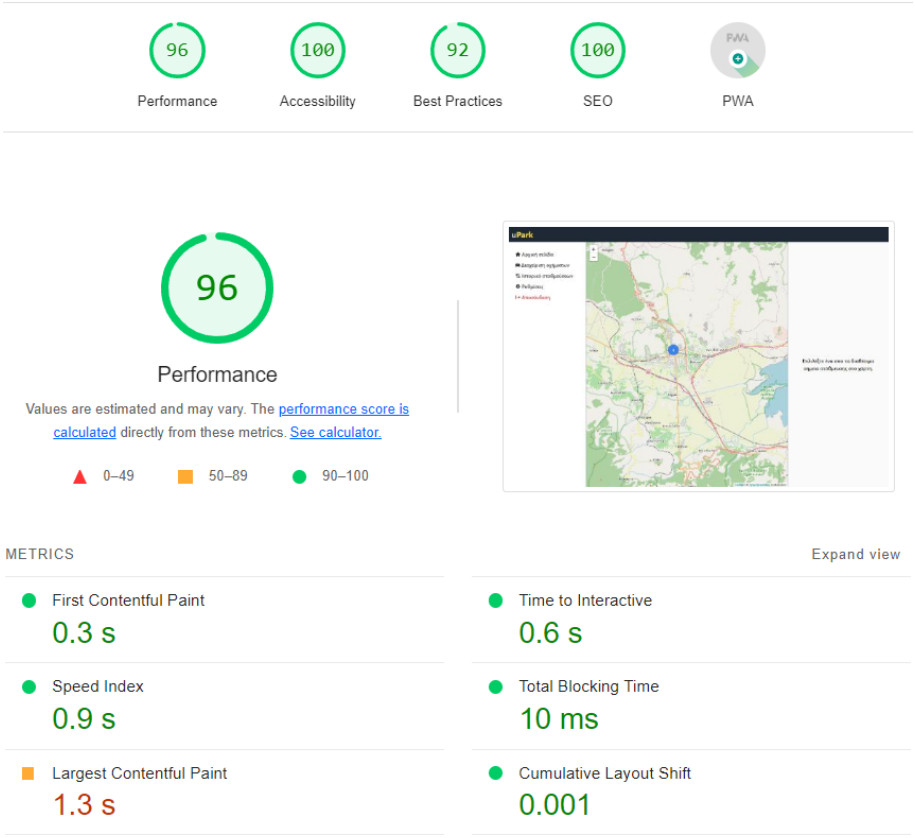
Εικόνα 6.23 Προεπισκόπηση ενεργής κάρτας στάθμευσης στην αρχική σελίδα

Επιστρέφοντας στην αρχική οθόνη, ο οδηγός μπορεί να δει πληροφορίες για την «κάρτα» χρόνου στάθμευσης που μόλις αγόρασε (Εικόνα 6.23), ενώ ακόμη του δίνεται τόσο η δυνατότητα επέκτασης του υπολειπόμενου χρόνου, όσο και η δυνατότητα πρόωρης λήξης της κάρτας, μεταφέροντας τον υπολειπόμενο χρόνο της για μελλοντική χρήση.

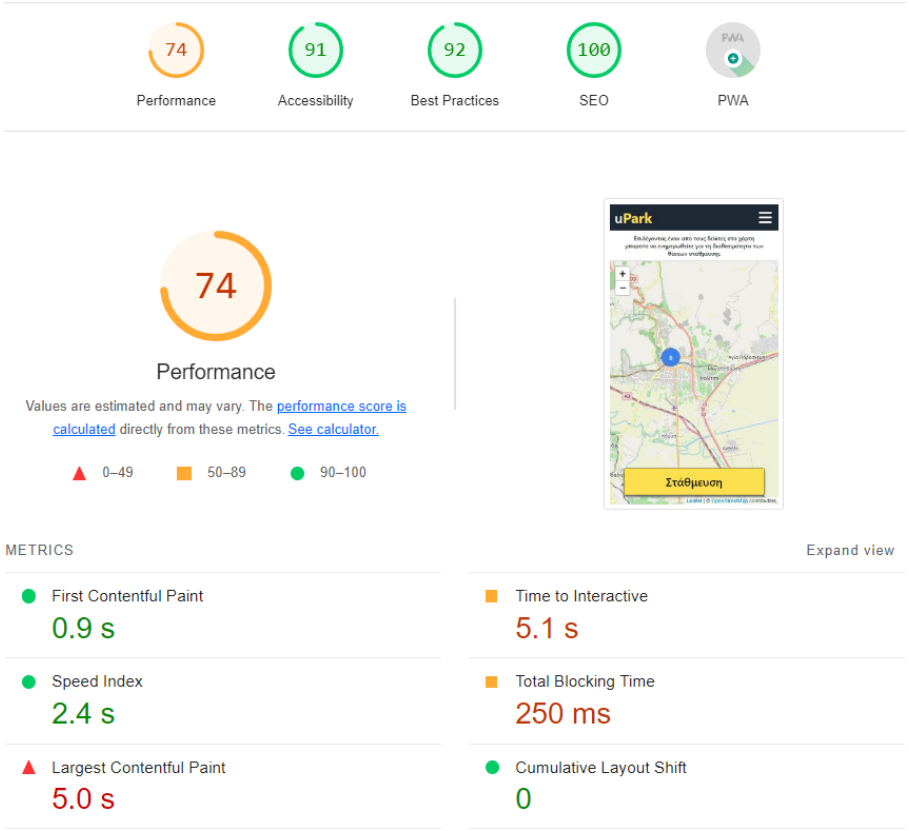
7 Αξιολόγηση ποιότητας της εφαρμογής

Σε αυτό το κεφάλαιο θα παρουσιαστεί μια πρώτη αξιολόγηση της ποιότητας της εφαρμογής. Η αξιολόγηση αυτή πραγματοποιήθηκε με τη χρήση του εργαλείου Lighthouse (https://developer.chrome.com/docs/lighthouse/). Το Lighthouse αποτελεί ένα αυτοματοποιημένο εργαλείο ανοιχτού κώδικα το οποίο συμβάλλει στη βελτίωση της ποιότητας των ιστοσελίδων. Μπορεί να λειτουργήσει για οποιαδήποτε ιστοσελίδα, είτε αυτή είναι δημόσια είτε απαιτεί έλεγχο ταυτότητας. Διαθέτει ελέγχους οι οποίοι προσομοιώνουν την λειτουργία μιας ιστοσελίδας τόσο σε κινητές συσκευές, όσο και σε σταθερούς Η/Υ και παρέχουν πληροφορίες σχετικά με την επίδοση και την ταχύτητα μια ιστοσελίδας, την προσβασιμότητα, πληροφορίες που αφορούν Progressive Web Applications και SEO. Παρακάτω ακολουθούν τα αποτελέσματα των ελέγχων αυτών για τα τρία υποσυστήματα (οδηγού, ελεγκτή, διαχειριστή) που αναπτύχθηκαν στο πλαίσιο αυτής της εργασίας.

1. Οδηγός

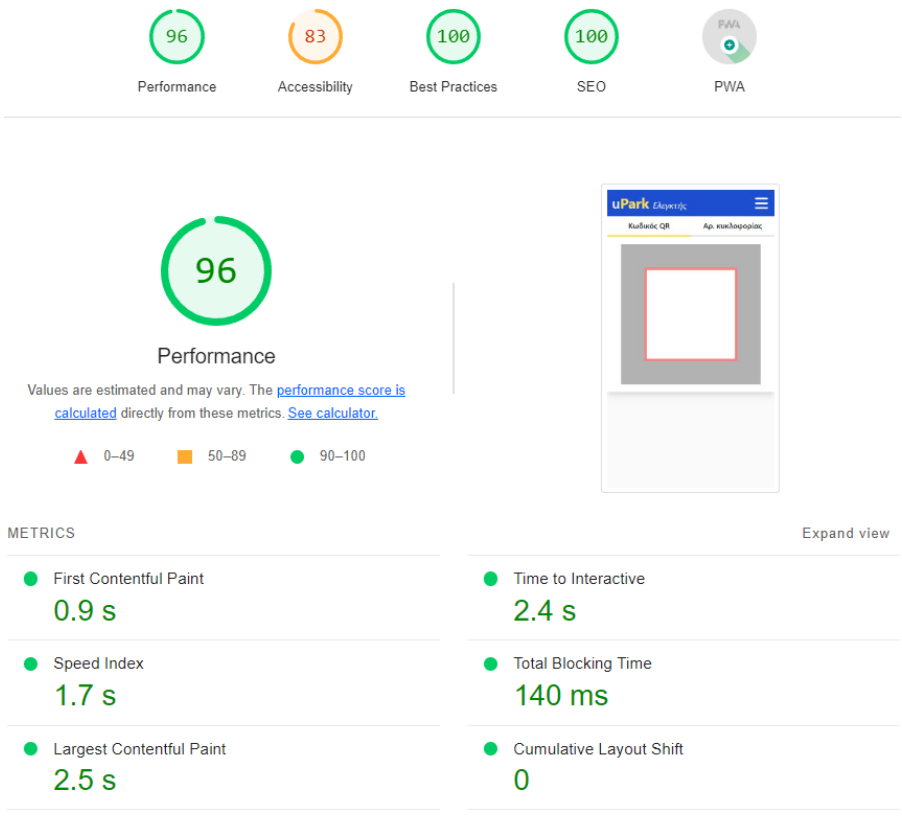


Εικόνα 7.1 Αποτελέσματα ελέγχου desktop εκδοχής



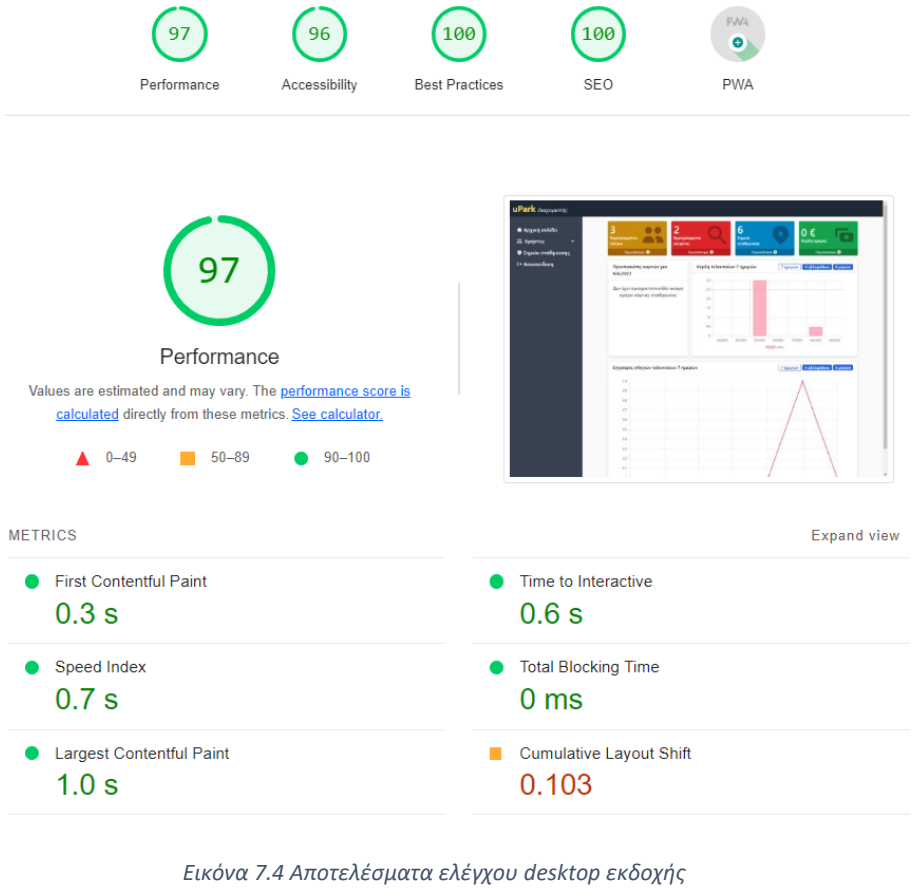
Εικόνα 7.2 Αποτελέσματα ελέγχου mobile εκδοχής

2. Ελεγκτής

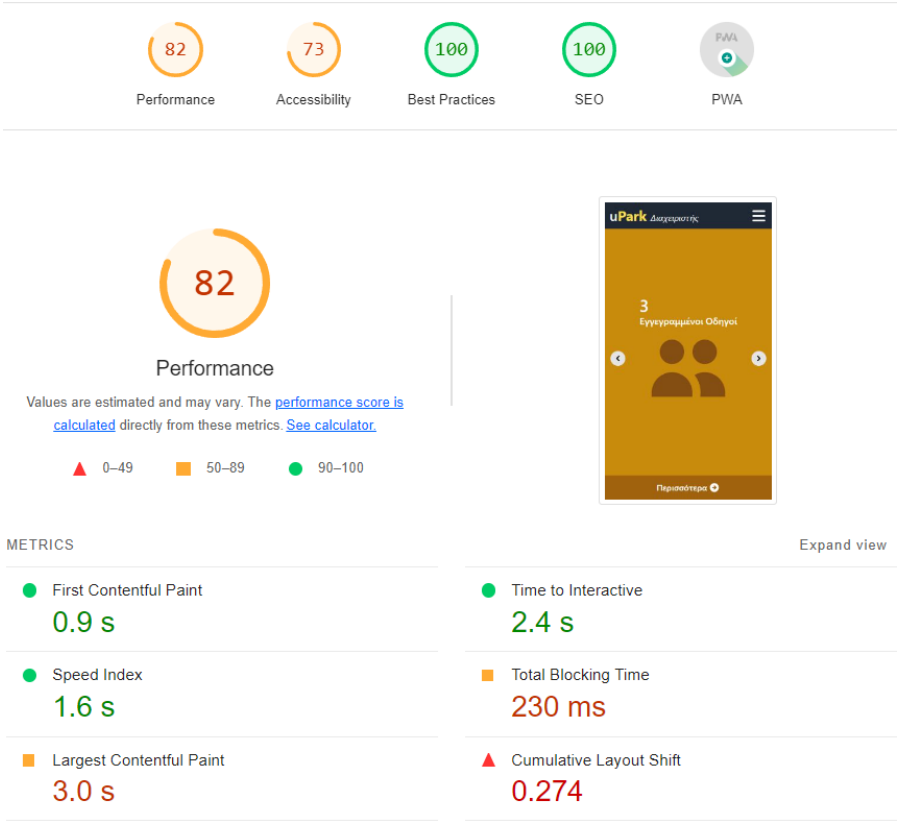


Εικόνα 7.3 Αποτελέσματα ελέγχου mobile εκδοχής

3. Διαχειριστής



Εικόνα 7.4 Αποτελέσματα ελέγχου desktop εκδοχής



Εικόνα 7.5 Αποτελέσματα ελέγχου mobile εκδοχής

Αναφορικά με τα αποτελέσματα που παρουσιάζονται στις Εικόνες 7.1 έως 7.5 παρατηρούμε αρχικά, από την ένδειξη στο δεξιότερο επάνω μέρος κάθε εικόνας, ότι και οι τρείς ιστοσελίδες αναγνωρίζονται ως Progressive Web Applications, ενώ ακόμη έχουμε μια σύνοψη σχετικά με τον δείκτη απόδοσης της ιστοσελίδας και κατά πόσο εφαρμόζει τεχνικές προσβασιμότητας, σωστές πρακτικές και SEO. Στη συνέχεια παρουσιάζεται μια πιο λεπτομερής ανάλυση σχετικά με την επίδοση και την ταχύτητα της ιστοσελίδας, καθώς και οι χρόνοι απόκρισής της στους διάφορους ελέγχους που εφαρμόστηκαν. Ακολουθεί μια περιγραφή σχετικά με το τί αντιπροσωπεύει κάθε μια από αυτές τις μετρήσεις.

- **First Contentful Paint:** η μέτρηση αυτή, αναπαριστά πόσο χρόνο χρειάζεται ένας περιηγητής ιστού για να εμφανίσει το πρώτο DOM στοιχείο όταν κάποιος χρήστης επισκέπτεται την ιστοσελίδα. Οι εικόνες, τα μη λευκά στοιχεία <canvas> και τα SVG στοιχεία της σελίδας συγκαταλέγονται στα στοιχεία DOM, ενώ οτιδήποτε μέσα σε ένα <iframe> στοιχείο δεν περιλαμβάνεται.
- **Time to Interactive:** η μέτρηση αυτή, αναπαριστά τον χρόνο που απαιτείται για να γίνει πλήρως διαδραστική μια ιστοσελίδα, δηλαδή να εμφανίσει χρήσιμο περιεχόμενο, το οποίο υπολογίζει η First Contentful Paint μέτρηση.
- **Speed Index:** η μέτρηση αυτή, μετρά πόσο γρήγορα εμφανίζεται στο χρήστη το περιεχόμενο κατά τη φόρτωση της σελίδας.
- **Total Blocking Time:** η μέτρηση αυτή, υπολογίζει το συνολικό χρονικό διάστημα κατά το οποίο μια ιστοσελίδα καθυστερεί να αποκριθεί σε εισόδους από τον χρήστη, όπως για παράδειγμα είσοδο από το ποντίκι, το πληκτρολόγια ή μια οθόνη αφής.
- **Largest Contentful Paint:** η μέτρηση αυτή μετρά το χρόνο που απαιτείται για την εμφάνιση του μεγαλύτερου στοιχείου της ιστοσελίδας στην οθόνη.

Όσον αφορά τις μετρήσεις Performance των τριών υποσυστημάτων, τιμές που κυμαίνονται μεταξύ 90 και 100 υποδεικνύουν μια εφαρμογή που προσφέρει μια καλή εμπειρία χρήστη, τιμές που κυμαίνονται μεταξύ 50 και 89 υποδεικνύουν μια εφαρμογή που προσφέρει μια αποδεκτή εμπειρία χρήστη, η οποία όμως έχει περιθώρια βελτίωσης, ενώ τιμές που κυμαίνονται μεταξύ 0 και 49 υποδεικνύουν μια εφαρμογή που προσφέρει μια «φτωχή» εμπειρία χρήστη. Γίνεται επομένως αντιληπτό, ότι η mobile εκδοχές των υποσυστημάτων για οδηγούς και διαχειριστές χρήζουν βελτίωσης.

8 Συμπεράσματα και μελλοντικές βελτιώσεις

Εν κατακλείδι, στο πλαίσιο αυτής της διπλωματικής εργασίας αναπτύχθηκε ένα Progressive Web Application με την ονομασία uPark, που είναι μια εφαρμογή ιστού η οποία μπορεί να αποτελέσει μια εναλλακτική λύση στο μείζον πρόβλημα της εύρεσης θέσεων στάθμευσης σε αστικές περιοχές (κέντρα πόλεων, χώρους στάθμευσης αγορών, πανεπιστημιουπόλεις, κτλ) και της αγοράς άυλων καρτών χρόνου στάθμευσης για τα οχήματα των οδηγών που δραστηριοποιούνται εντός των περιοχών αυτών. Το uPark μπορεί να αποτελέσει βασική εφαρμογή και να υιοθετηθεί από έναν φορέα, είτε δημόσιο είτε ιδιωτικό (όπως για παράδειγμα ένα δήμο, μια πανεπιστημιούπολη ή ένα εμπορικό κέντρο), ο οποίος επιθυμεί να εφαρμόσει ένα σύστημα ελεγχόμενης στάθμευσης. Με άλλα λόγια η εφαρμογή uPark μπορεί να αποτελέσει μια ολοκληρωμένη λύση καθώς εκτός από την πλατφόρμα για τη σύνδεση των οδηγών, παρέχει επίσης πλατφόρμα για ελεγκτές, καθώς και για διαχειριστές για τη συνολική επίβλεψη του συστήματος. Ως Progressive Web Application, το uPark είναι εύκολα προσβάσιμο τόσο από κινητές συσκευές όσο και από σταθερούς Η/Υ, ενώ η επιλογή και η χρήση σύγχρονων και διαδεδομένων τεχνολογιών εφαρμογών ιστού του προσδίδουν την απαραίτητη ταχύτητα, λειτουργικότητα και εμφάνιση που πρέπει να έχει μια σύγχρονη εφαρμογή.

Παρά την εκτενή λειτουργικότητα που ήδη προσφέρει [15], υπάρχουν δυνατότητες για περαιτέρω εμπλουτισμό του uPark. Μια σημαντική προσθήκη θα ήταν η υλοποίηση ενός συστήματος push notifications, που το uPark, ως ένα PWA, μπορεί να το υποστηρίξει. Επίσης, σε συνεργασία με πιθανούς φορείς που ενδιαφέρονται, θα ήταν χρήσιμη η συμπερίληψη θέσεων ΑΜΕΑ καθώς και η δημιουργία αγγλικής εκδοχής της εφαρμογής, κάτι που θα καταστούσε εφικτή τη χρήση της εφαρμογής από μεγαλύτερη μερίδα πολιτών. Τέλος, η δημιουργία ενός ηλεκτρονικού πορτοφολιού για τους οδηγούς, εντός της εφαρμογής, έτσι ώστε να μη χρειάζεται η εισαγωγή των τραπεζικών τους στοιχείων σε κάθε αγορά χρόνου στάθμευσης, θα μπορούσε να συμβάλει στη βελτίωση της εμπειρίας των χρηστών της εφαρμογής uPark. Αυτές αποτελούν μόνο μερικές από τις πιθανές βελτιώσεις και επεκτάσεις, καθώς η συνεχείς κοινωνικές και τεχνολογικές αλλαγές παίζουν σημαντικό ρόλο στην εξέλιξη και στη μελλοντική πορεία της εφαρμογής uPark.

9 Βιβλιογραφία

[1] M. J. Wargo, Learning Progressive Web Apps, Addison-Wesley Professional, 2020.

[2] D. Sheppard, Beginning Progressive Web App Development: Creating a Native App Experience on the Web, Apress, 2017.

[3] T. Ater, Building Progressive Web Apps: Bringing the Power of Native to the Browser, O'Reilly, 2017.

[4] Web Application Manifest, https://w3c.github.io/manifest

[5] Media Queries Level 5, https://www.w3.org/TR/mediaqueries-5/

[6] Service workers, https://developers.google.com/web/ilt/pwa/introduction-to-service-worker

[7] A service worker's life - Chrome Developers, https://developer.chrome.com/docs/workbox/service-worker-lifecycle/#the-lifecycle-of-a-new-service-worker

[8] Introduction to Node.js, https://nodejs.dev/learn

[9] G. Lim, Beginning Node.js, Express & MongoDB Development, Independently published, 2019.

[10] M. Casciaro και L. Mammino, Node.js Design Patterns: Design and implement production-grade Node.js applications using proven patterns and techniques, Packt Publishing, 2020.

[11] D. Choi, Full-Stack React, TypeScript, and Node: Build cloud-ready web applications using React 17 with Hooks and GraphQL, Packt Publishing, 2020.

[12] B. Cherny, Programming TypeScript: Making Your JavaScript Applications Scale, O'Reilly , 2019.

[13] Benefits of Typescript, https://ionicframework.com/docs/v3/developer-resources/typescript/

[14] Introduction to Tailwind CSS - GeeksforGeeks, https://www.geeksforgeeks.org/introduction-to-tailwind-css/

[15] K. Panos, E. Papaioannou, C. Kaklamanis. uPark: a progressive web parking application. Technium: Romanian Journal of Applied Sciences and Technology, vol. 4, no 5, pp. 26-38, 2022.