

.Έγγραφο απαιτήσεων λογισμικού (SRS)

ΠΡΟΣΑΡΜΟΓΗ ΤΟΥ ΑΝΤΙΣΤΟΙΧΟΥ ΕΓΓΡΑΦΟΥ ΤΟΥ ΠΡΟΤΥΠΟΥ
ISO/IEC/IEEE 29148:2011

ParkApp

1. Εισαγωγή

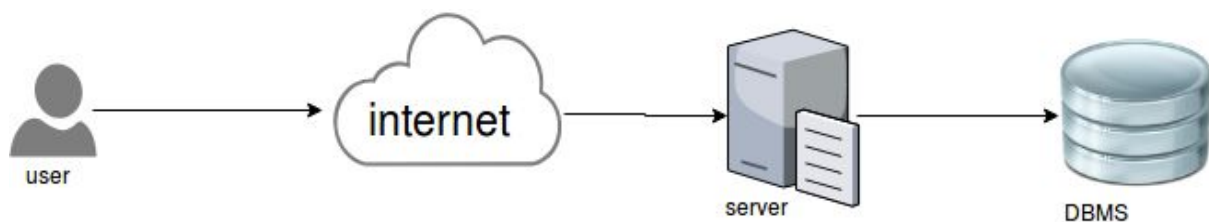
1.1 Εισαγωγή: σκοπός του λογισμικού

Οριοθέτηση του σκοπού του συστήματος

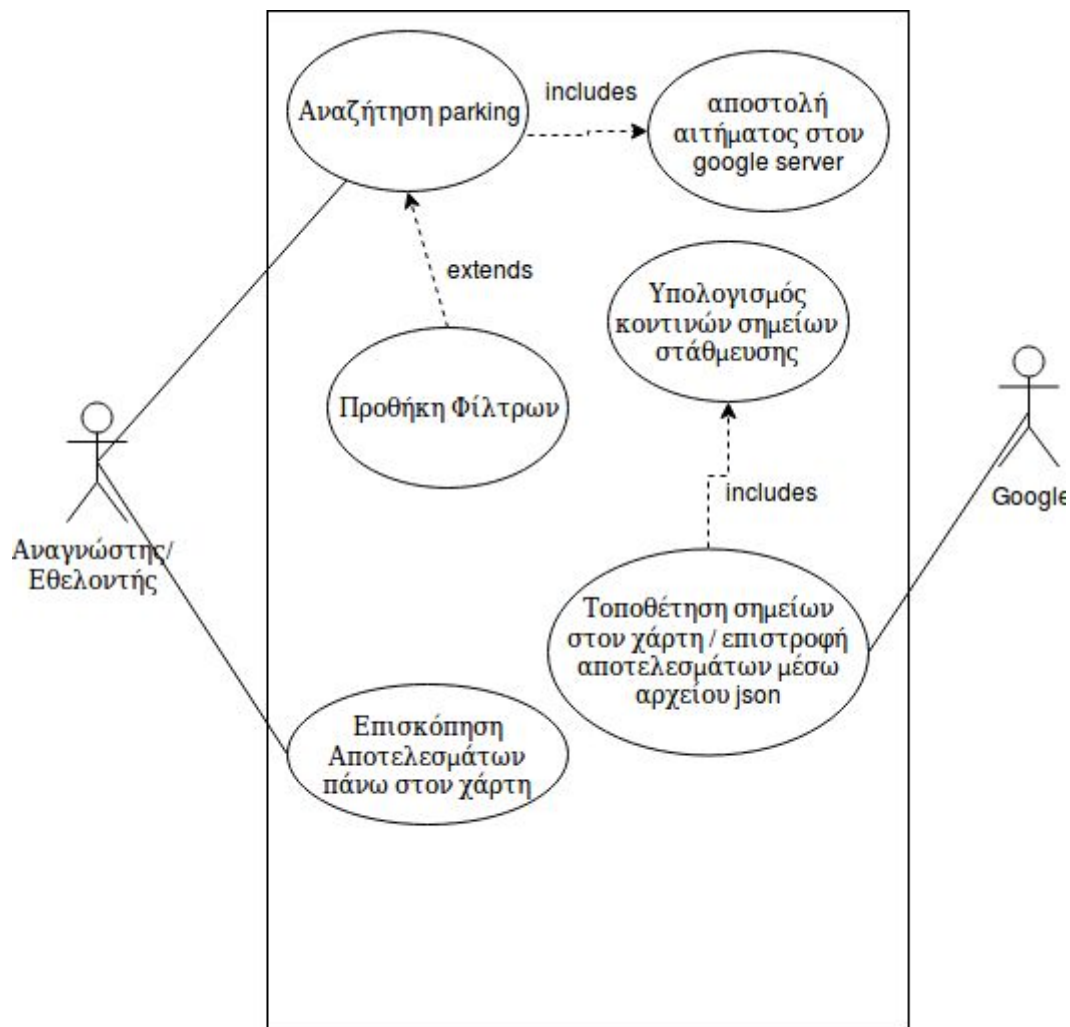
Σκοπός του συστήματος είναι η διευκόλυνση της εύρεσης χώρων στάθμευσης και η ενημέρωση για τις τιμές και την τοποθεσία τους, μέσω της συλλογικής συμβολής των εθελοντών.

1.2 Επισκόπηση του λογισμικού

Γενική περιγραφή με χρήση διαγράμματος UML. Οι διεπαφές αναφέρονται μόνο ως τίτλοι ή/και σε διάγραμμα.

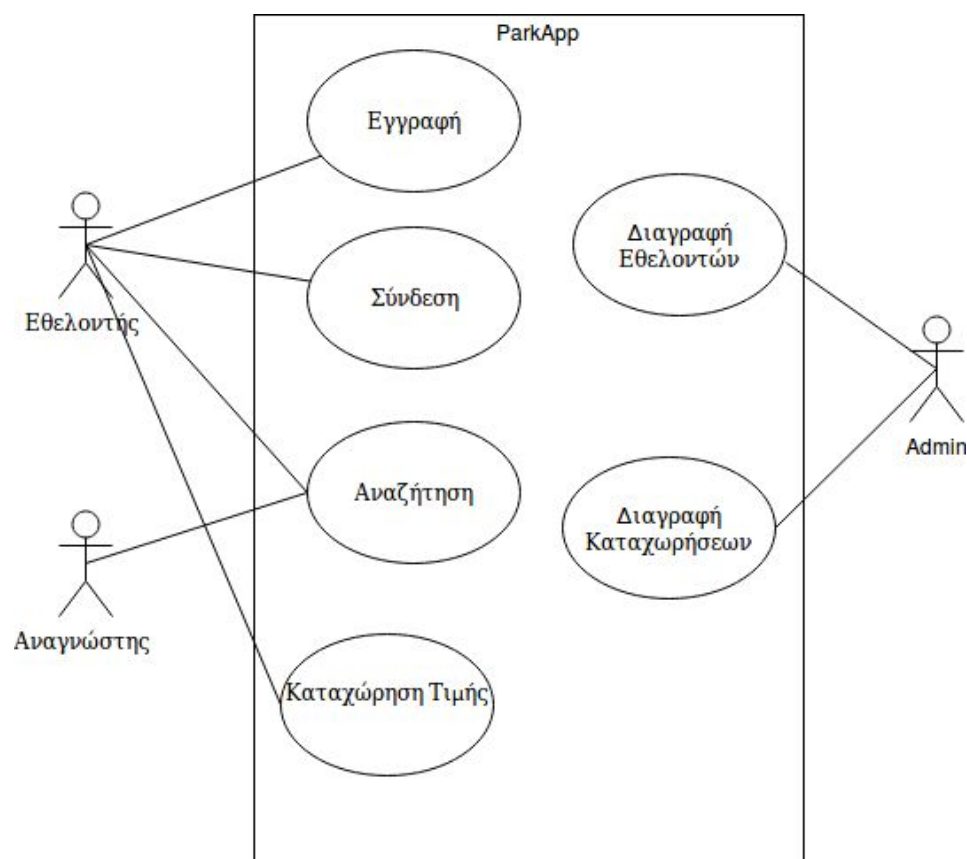


1.3.1 Διεπαφές με εξωτερικά συστήματα και εφαρμογές λογισμικού



Προδιαγραφή διεπαφών με εξωτερικά συστήματα και λογισμικό, με αναφορά σε πρότυπα ανταλλαγής δεδομένων και κλήσης υπηρεσιών. Χρήση διαγραμμάτων UML.

1.3.2 Διεπαφές με το χρήστη



Για να μπορεί κάθε χρήστης να χρησιμοποιήσει την αντίστοιχη διεπαφή θα πρέπει να περάσει από μία επικύρωση αναλογα με την περίπτωση χρήσης, όπως φαίνεται στο ανωτέρω διαγράμμα. Ο αναγνώστης δεν χρειάζεται να περάσει από κάποιον έλεγχο εφόσον κάνει μόνο μια αναζήτηση στο σύστημα, χωρίς να το επηρεάζει.

Προδιαγραφή διεπαφών με το χρήστη. Μοντέλο Use Case (UML).

2. Προδιαγραφές απαιτήσεων λογισμικού

3.1 Εξωτερικές διεπαφές

Στα πλαίσια της εφαρμογής θα χρησιμοποιήσουμε το api της google για χάρτες. Από εκεί θα αντλούμε πληροφορίες σχετικά με την απόσταση 2 σημείων στο χάρτη (με την χρήση αυτοκινήτου) καθώς και για την καταγραφή της τοποθεσίας του εθελοντή την στιγμή της καταχώρησης.

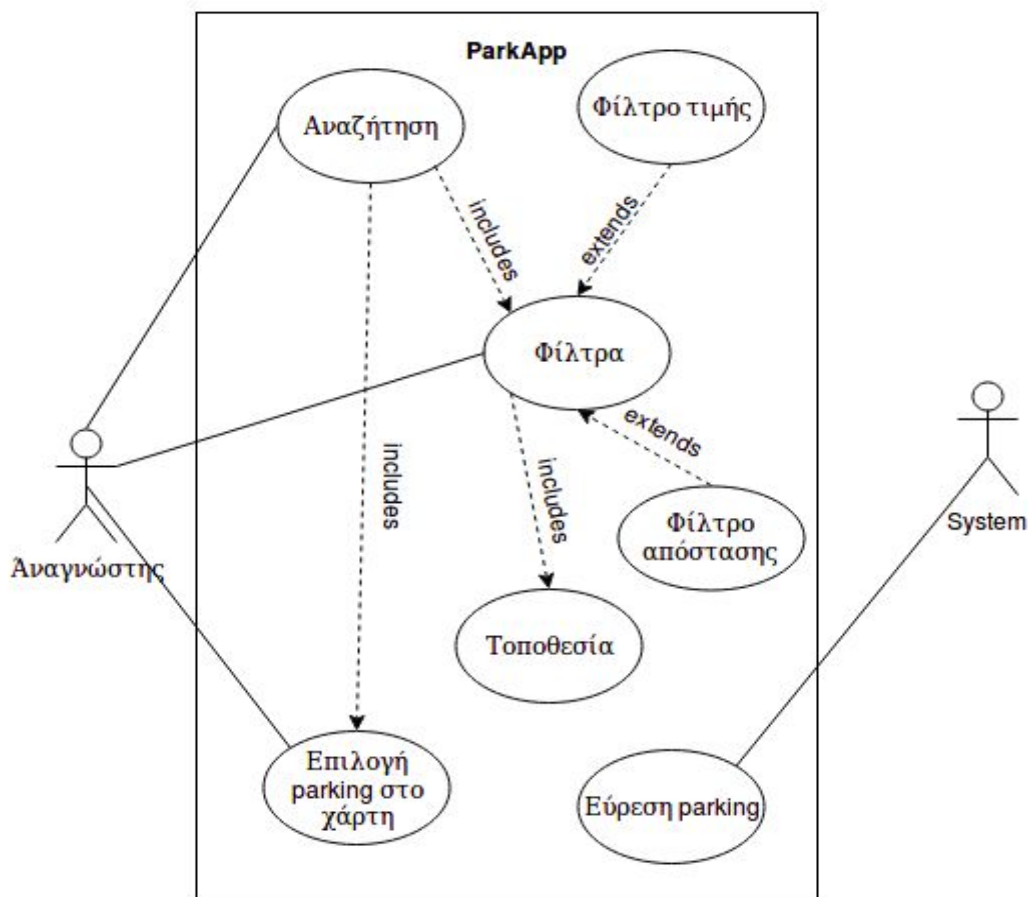
Πιο αναλυτικά όταν κάποιος εθελοντής θα θέλει να καταγράψει ένα νέο κατάστημα θα ανοίγει την εφαρμογή και θα θέλει να δηλώσει την τοποθεσία του, που θα είναι είτε η τωρινή του, είτε κάποια άλλη. Για αυτό θα βοηθάει το api της google. Επίσης θα μπορεί κάποιος να χρησιμοποιήσει το api για να δει ποια parking lots είναι κοντά με βάση την απόστασή με το αυτοκίνητο από το σημείο στο οποίο βρίσκεται.

3.2 Λειτουργίες: περιπτώσεις χρήσης

Λεπτομερής προδιαγραφή των λειτουργιών του λογισμικού σε επόμενο περιπτώσεων χρήσης.

1. Αναγνώστης- Αναζήτηση χώρου στάθμευσης
2. Εθελοντής- Σύνδεση στο σύστημα (login)

3.2.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Αναγνώστης



3.2.1.1 Χρήστες(ρόλοι) που εμπλέκονται

Οι χρήστες που εμπλέκονται στην πρώτη περίπτωση χρήσης είναι οι αναγνώστες οι οποίοι χρησιμοποιούν την εφαρμογή για προσωπική χρήση, δηλαδή για εύρεση parking που πληροί τις απαιτήσεις τους καθώς και το σύστημα της εφαρμογής για την ικανοποίηση των αιτημάτων τους.

Αναφορά στους ρόλους που αφορά η περίπτωση χρήσης

3.2.1.2 Προϋποθέσεις εκτέλεσης

Για να μπορεί ένας χρήστης να βρεθεί στη περίπτωση χρήσης του αναγνώστη θα πρέπει απλώς να εισέλθει στο site της εφαρμογής. Είναι η πιο απλή περίπτωση χρήσης της εφαρμογής και για αυτό δεν χρειάζεται η ικανοποίηση κάποιας ειδικής συνθήκης.

Καταγραφή των συνθηκών που πρέπει να ισχύουν ώστε να μπορεί να εκτελεστεί η περίπτωση χρήσης

3.2.1.3 Περιβάλλον εκτέλεσης

Στη περίπτωση χρήσης του αναγνώστη, χρησιμοποιείται η διαδικτυακή διεπαφή χρήστη ώστε να προστεθούν τα δεδομένα για να γίνει η αναζήτηση. Στη συνέχεια γίνεται αλληλεπίδραση με τη βάση δεδομένων μέσω της εφαρμογής μας ώστε να βρεθούν να κατάλληλα αποτελέσματα (αν υπάρχουν) και να παρουσιαστούν στον αναγνώστη.

Αναφορά στο περιβάλλον στο οποίο εκτελείται η περίπτωση χρήσης. Πχ "διαδικτυακή διεπαφή χρήστη", "DBMS" κλπ

3.2.1.4 Δεδομένα εισόδου

Στην συγκεκριμένη περίπτωση χρήσης τα δεδομένα εισόδου αποτελούνται από τις προδιαγραφές που θέλει ο αναγνώστης να πληροί ο χώρος στάθμευσης που αναζητεί (περιοχή, τιμή, απόσταση από την επιθυμητή περιοχή κλπ). Έτσι δεδομένα ως είσοδο μία περιοχή και κάποια (προαιρετικά) επιπλέον φίλτρα αναζήτησης, αναζητούμε στο σύστημα και εμφανίζουμε τα κατάλληλα αποτελέσματα. Σε περίπτωση που δεν βρεθούν αποτελέσματα που να ταιριάζουν στις απαιτήσεις του αναγνώστη, εμφανίζεται μήνυμα no results found.

Καταγραφή δεδομένων εισόδου και εξόδου και συνθηκών εγκυρότητας αυτών.

3.2.1.5 Παράμετροι

Οι παράμετροι στην αναζήτηση κάποιου χώρου στάθμευσης είναι, όπως αναφέρθηκε, η περιοχή που θέλουμε να σταθμεύσουμε, ένα εύρος τιμής, ακτίνα γύρω από την επιθυμητή περιοχή που είμαστε διατεθειμένοι να σταθμεύσουμε. Το σύστημα θα ελέγχει αν η περιοχή είναι έγκυρη καθώς και η ακτίνα και η τιμή πρέπει να είναι θετικοί αριθμοί με δύο δεκαδικά ψηφία.

Καταγραφή παραμέτρων και συνθηκών εγκυρότητας αυτών

3.2.1.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Βήμα 1: Άνοιγμα Browser

Βήμα 2: Είσοδος στο site

Βήμα 3: Κλικ στην αναζήτηση χώρου στάθμευσης

Βήμα 4: Προσθήκη περιοχής αναζήτησης

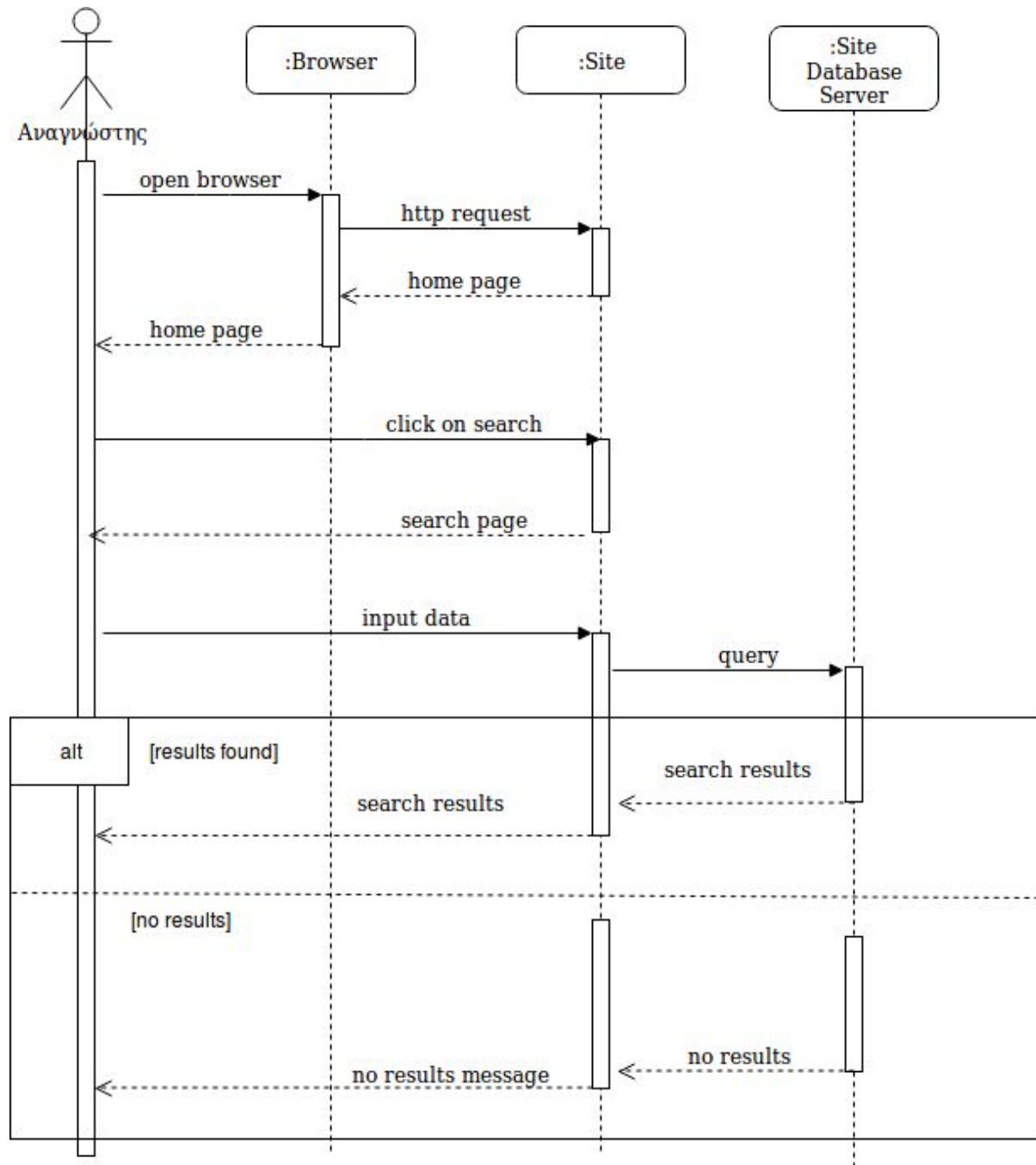
Βήμα 5: Προσθήκη επιπρόσθετων φίλτρων (τιμής, ακτίνας)

Βήμα 6: Αναζήτηση αποτελεσμάτων στη βάση

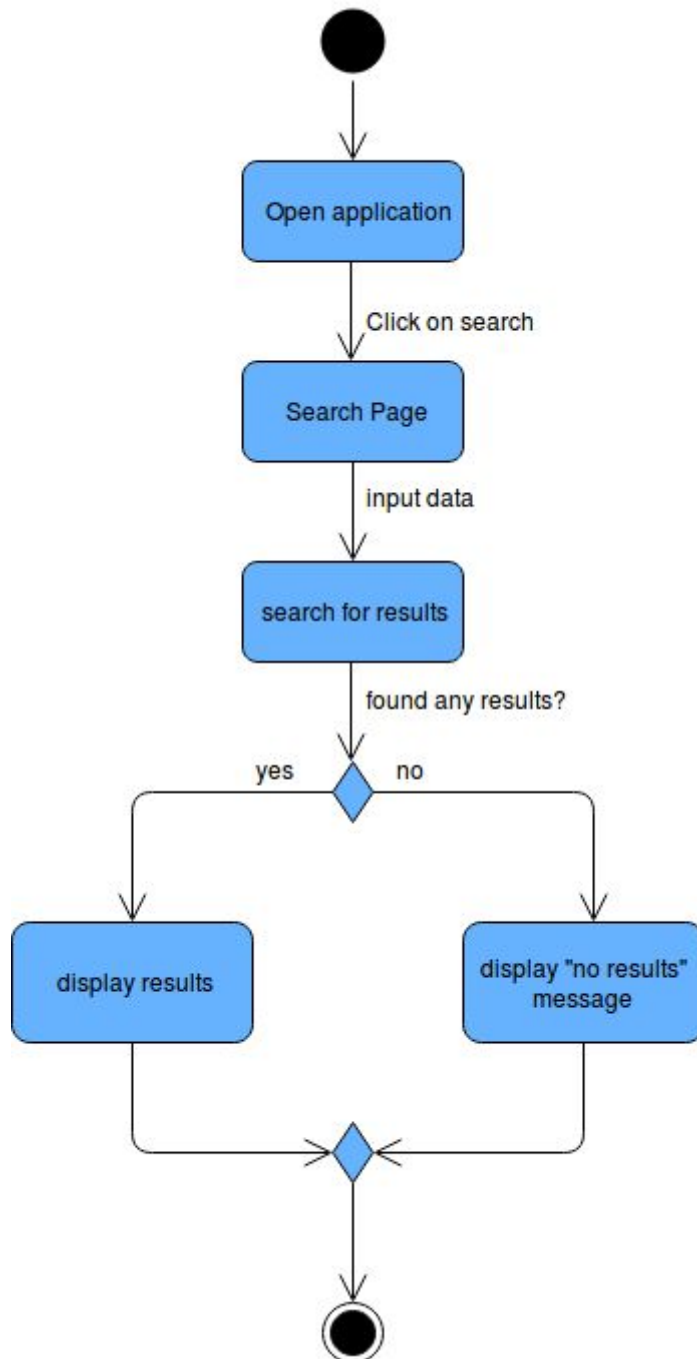
Βήμα 7: Εμφάνιση αποτελεσμάτων (αν υπάρχουν - αλλιώς εμφανίζει μήνυμα ότι δεν βρέθηκαν αποτελέσματα)

Βήμα 8: Κλικ σε κάποιο αποτέλεσμα και επισκόπηση των στοιχείων του

Sequence Diagram:

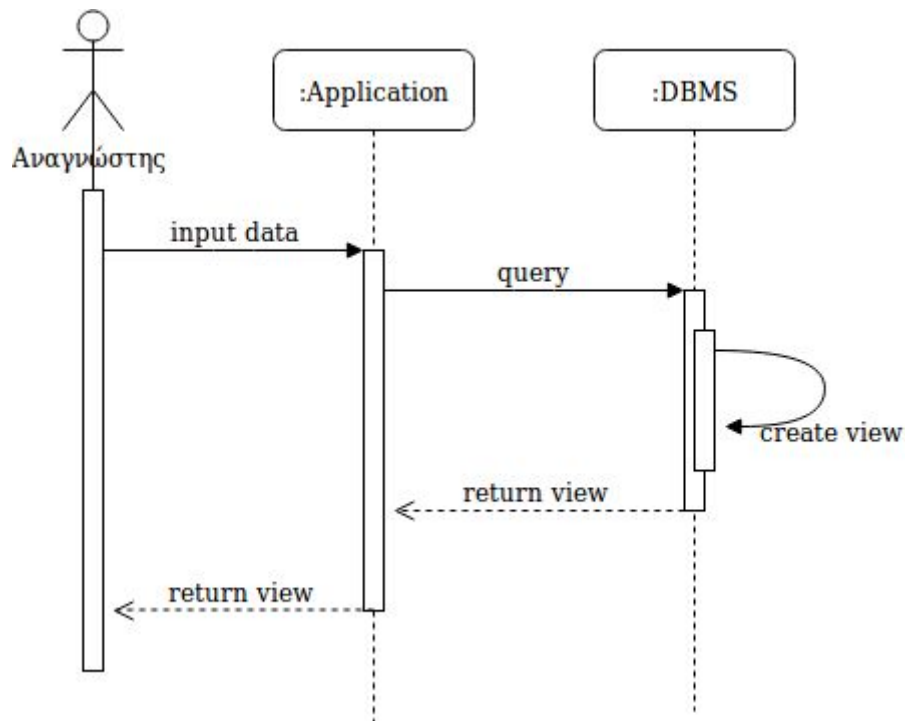


Activity Diagram:



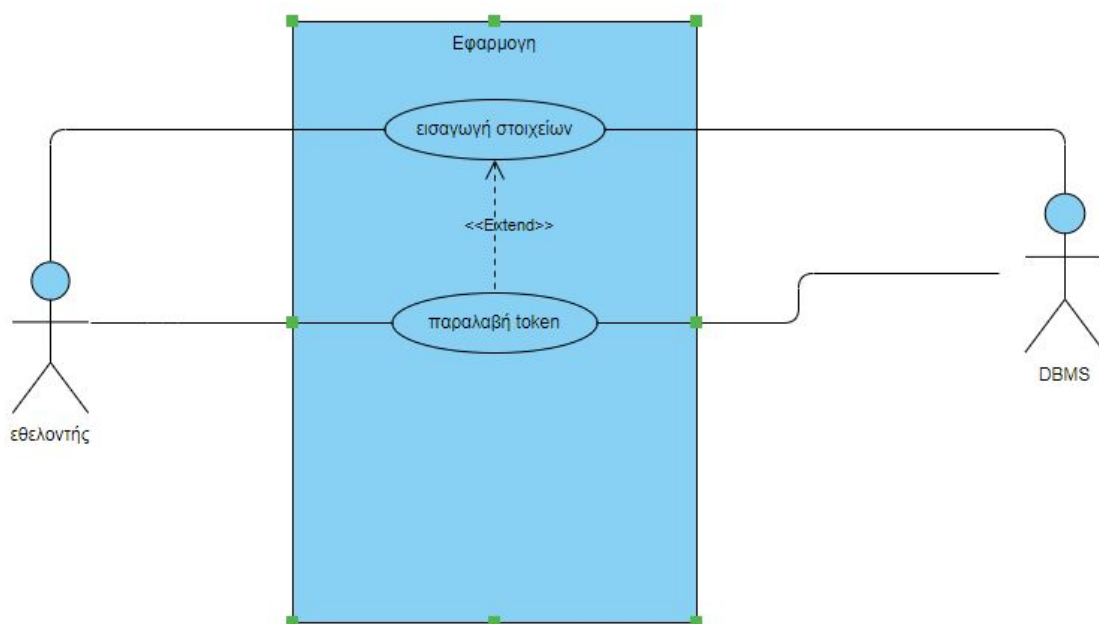
Περιγράψιμε κείμενο (Βήμα 1, Βήμα 2 κλπ) και διαγράμματα UML αλληλουχίας (Sequence) και δραστηριοτήτων (Activity). Περιλαμβάνεται η συμπεριφορά σε απρόβλεπτες καταστάσεις και σφάλματα (εναλλακτικές ροές).

3.2.1.7 Δεδομένα εξόδου



Διαγράμματα UML αλληλουχίας για την παραγωγή δεδομένων εξόδου. Ως δεδομένα εξόδου νοούνται όλα τα δεδομένα του συστήματος τα οποία δημιουργούνται ή μεταβάλλονται κατά την εκτέλεση (αν υπάρχουν τέτοια)

3.2.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Εθελοντής



3.2.2.1 Χρήστες(ρόλοι) που εμπλέκονται

Σε αυτή τη περίπτωση χρήσης εμπλέκονται οι εθελοντές, οι οποίοι θέλουν να πιστοποιήσουν ότι είναι μέλη.

Αναφορά στους ρόλους που αφορά η περίπτωση χρήσης

3.2.2.2 Προϋποθέσεις εκτέλεσης

Προϋπόθεση για να γίνει το παραπάνω σενάριο είναι ο εθελοντής να έχει πρόσβαση στην εφαρμογή μέσω του διαδικτύου και να έχει λογαριασμό.

Καταγραφή των συνθηκών που πρέπει να ισχύουν ώστε να μπορεί να εκτελεστεί η περίπτωση χρήσης

3.2.2.3 Περιβάλλον εκτέλεσης

Το παραπάνω εκτελείται στην διαδικτυακή επαφή του εθελοντή. Η αλληλεπίδραση του χρήστη γίνεται με το σύστημα το οποίο επιφέρει αλλαγές στο DBMS.

Αναφορά στο περιβάλλον στο οποίο εκτελείται η περίπτωση χρήσης. Πχ "διαδικτυακή διεπαφή χρήστη", "DBMS" κλπ

3.2.2.4 Δεδομένα εισόδου

Ο χρήστης σαν δεδομένο εισόδου στο login περνάει το username και το password του και το αποτέλεσμα της εξόδου είναι το authentication token.

Καταγραφή δεδομένων εισόδου και εξόδου και συνθηκών εγκυρότητας αυτών.

3.2.2.5 Παράμετροι

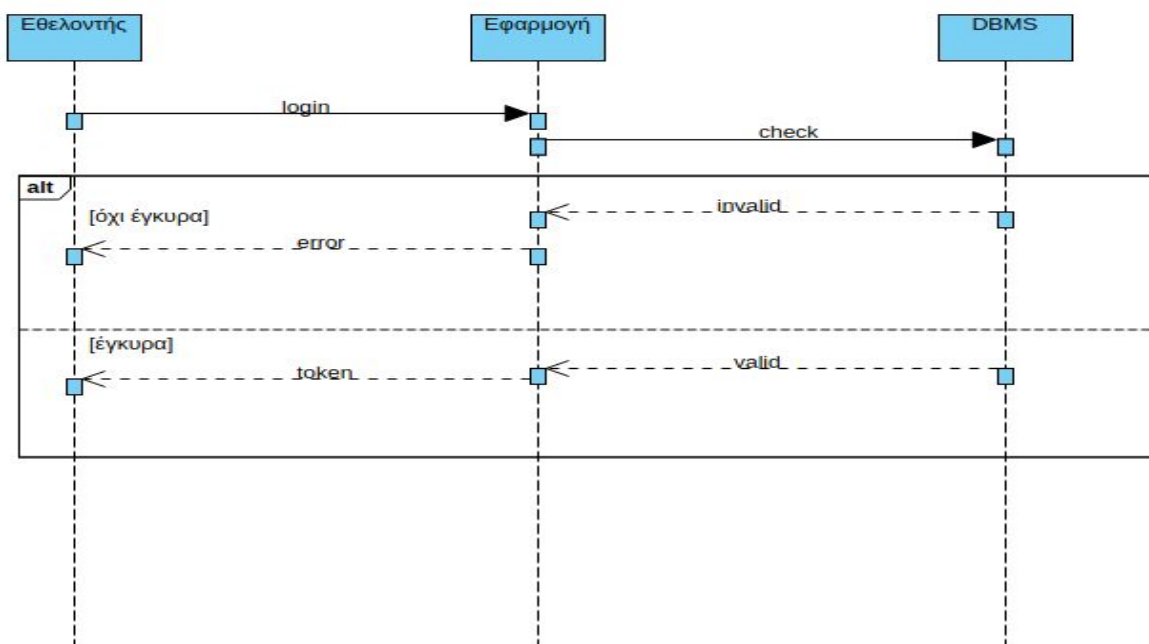
Οι παράμετροι είναι το username και το password του χρήστη.

3.2.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

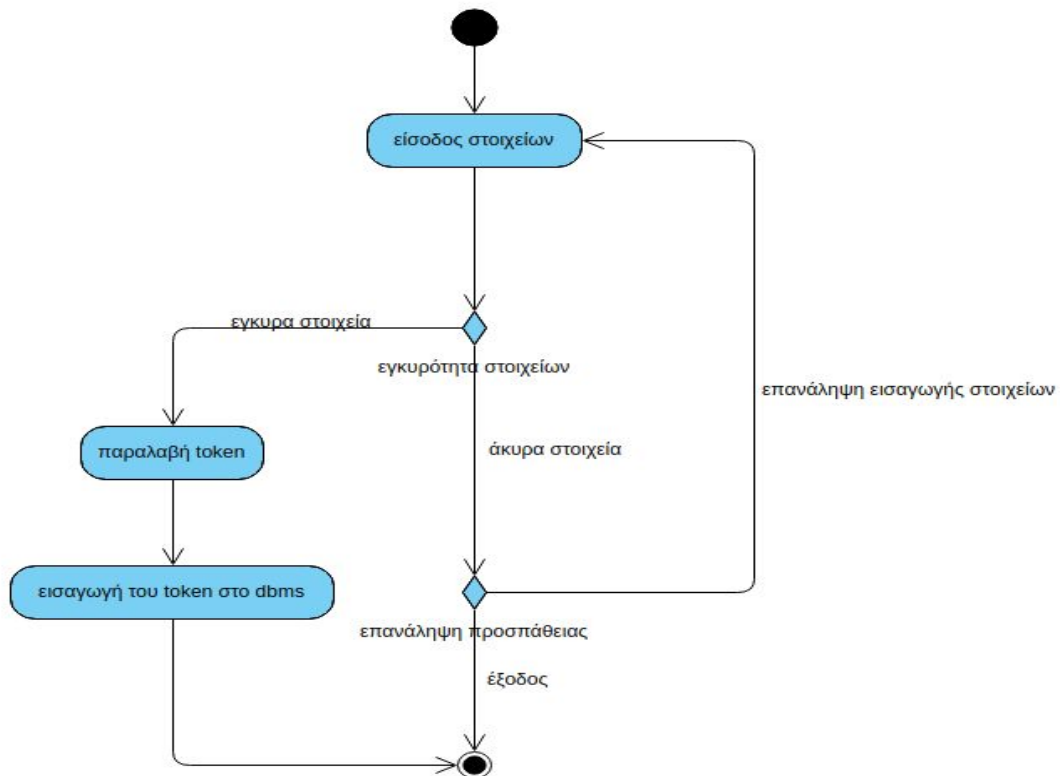
Βήματα:

1. Εισαγωγή username και password
2. Η εφαρμογή παίρνει αυτά τα δεδομένα και κοιτάει αν υπάρχουν στο DBMS
3. Αν δεν υπάρχουν επιστρέφει αντίστοιχο μήνυμα
4. Αν υπάρχουν τότε δημιουργεί ένα token
5. Το token το βάζει στο dbms και το δίνει και στον εθελοντή για να συνεχίσει

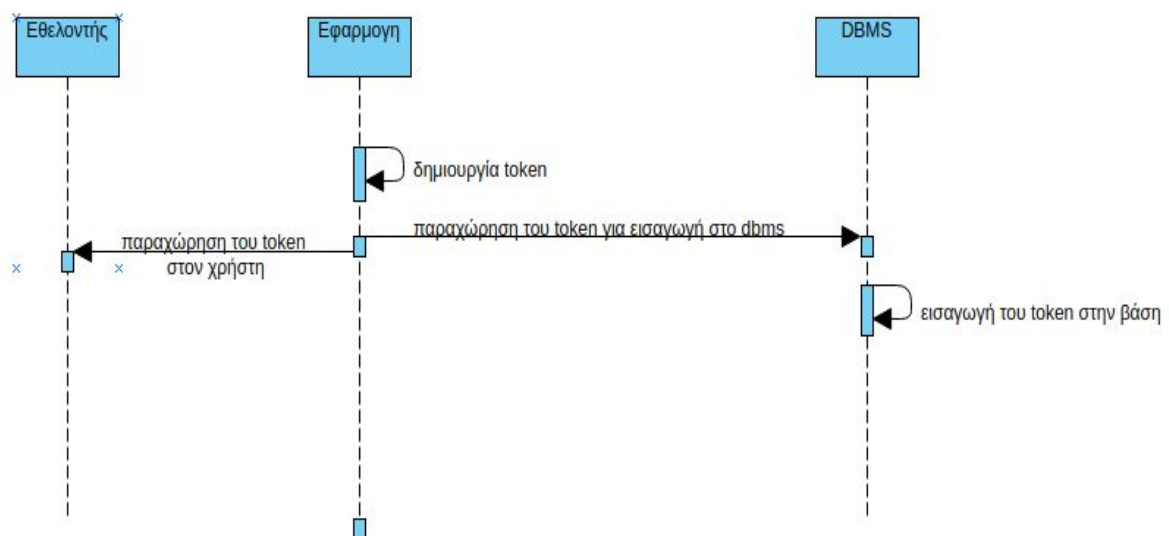
Sequence Diagram:



Activity Diagram:



3.2.2.7 Δεδομένα εξόδου



Διαγράμματα UML αλληλουχίας για την παραγωγή δεδομένων εξόδου. Ως δεδομένα εξόδου νοούνται όλα τα δεδομένα του συστήματος τα οποία δημιουργούνται ή μεταβάλλονται κατά την εκτέλεση (αν υπάρχουν τέτοια)

3.3 Απαιτήσεις επιδόσεων

Η επίδοση του συστήματος μας καθορίζεται από το ποσοστό των ενεργών χρηστών (εγγεγραμμένοι χρήστες που χρησιμοποιούν ουσιαστικά την εφαρμογή-αναζήτηση/ενημέρωση/προσθήκη προϊόντων) σε σχέση με τους συνολικούς χρήστες. Επίσης η επίδοση του συστήματος επηρεάζεται και από την ατομική συμπεριφορά των χρηστών, οι οποίοι παρουσιάζουν κάποια συχνότητα χρήσης του λογισμικού, η οποία χαρακτηρίζεται από μικρές εναλλαγές. Έτσι σε περιόδους με αυξημένα ποσοστά ενεργών χρηστών, οι υπολογιστικοί πόροι του συστήματος πρέπει να είναι αυξημένοι ανάλογα, ώστε να εξυπηρετούν το αντίστοιχο φορτίο.

Ποσοτική εκμείωση μέτρων και κριτηρίων επιθυμητών επιδόσεων με αναφορά στα ποσοτικά χαρακτηριστικά εισόδων και φορτίου του λογισμικού.

3.4 Απαιτήσεις οργάνωσης δεδομένων

3.4.1 Τεχνική περιγραφή των δεδομένων που διαχειρίζεται το λογισμικό και των σχετικών μετρικών φορτίου δεδομένων εισόδου, επεξεργασίας κ.λπ.

Τα δεδομένα που διαχειρίζομαστε χωρίζονται σε μόνιμα δεδομένα και προσωρινά. Τα μόνιμα δεδομένα απαρτίζονται από τα δεδομένα χρηστών/προϊόντων/καταστημάτων και τα προσωρινά αφορούν τους χρήστες που έχουν κάνει login στην βάση.

Πιο συγκεκριμένα τα δεδομένα χρηστών/προϊόντων/καταστημάτων δημιουργούνται όταν κάποιος αποφασίζει να κάνει signup ή να καταχωρήσει ένα καινούριο κατάστημα ή ένα καινούριο προϊόν. Τα δεδομένα που καταχωρεί και εμείς αποθηκεύουμε είναι τα δεδομένα που φαίνονται παρακάτω με τους αντίστοιχους περιορισμούς. Πρέπει να τονιστεί ότι ο όγκος των δεδομένων αυξάνεται με την προσθήκη νέων καταστημάτων ή χρηστών ή προϊόντων και όχι με την επεξεργασία τους.

Η επεξεργασία γίνεται με σκοπό την επικαιρότητα των δεδομένων και πραγματοποιείται από τους εθελοντές. Αυτή περιλαμβάνει ενημέρωση τιμών η ονομάτων αλλά δεν επιβαρύνει την εφαρμογή αναφορικά με την χωρητικότητα. Επίσης το table logged δεν μπορεί να έχει μέγεθος μεγαλύτερο από αυτό των users, αφού

δεν μπορεί να είναι περισσότεροι logged in από ό,τι ο αριθμός των χρηστών.

Σαν μετρική θα μπορούσαμε να χρησιμοποιήσουμε το πόσα δεδομένα μας προκαλούν πρόβλημα το οποίο για να λυθεί απαιτεί αύξηση του αποθηκευτικού χώρου. Στην αρχή που θα είναι σχεδόν κενή η βάση δεν θα αντιμετωπίζουμε τέτοια προβλήματα. Ούτε επίσης όταν θα έχουν καταγραφεί τα περισσότερα προϊόντα και οι ενέργειες που θα γίνονται κατά κύριο λόγο θα είναι η επεξεργασία των δεδομένων και η προσθήκη νέων χρηστών που δεν επιβαρύνουν το σύστημα.

Αναλυτική αναφορά στα δεδομένα εισόδου, τα σχετικά πρότυπα δεδομένων και υπηρεσιών, καθώς και σε μετρικές που σχετίζονται με τα δεδομένα (storage capacity planning).

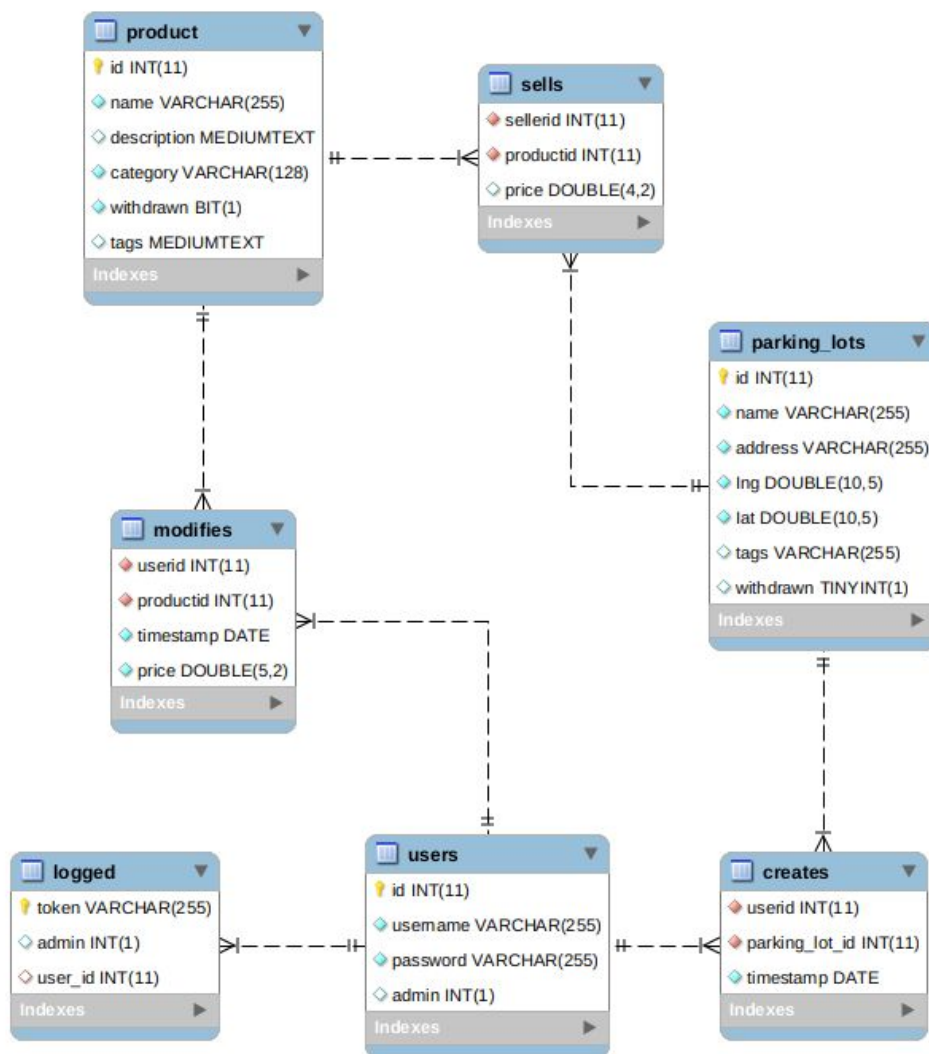
3.4.2 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

Η αναζήτηση και η θέαση των καταχωρήσεων που υπάρχουν στο σύστημα είναι ελεύθερες για όλους. Στην περίπτωση όμως που κάποιος επιθυμεί να κάνει μία νέα καταχώρηση ή να τροποποιήσει μία ήδη υπάρχουσα, θα πρέπει είτε να συνδεθεί στο σύστημα αν έχει λογαριασμό, είτε να δημιουργήσει έναν. Μετά την επιτυχή εγγραφή/σύνδεση του στο σύστημα ο εθελοντής έχει τη δυνατότητα να καταχωρήσει μία δική του παρατήρηση.

Απαιτήσεις πρόσβασης και περιορισμοί.

3.4.3 Μοντέλο δεδομένων (μοντέλο κλάσεων UML ή/και μοντέλο ER)

Διάγραμμα ER:



Μοντέλα δεδομένων UML ή/και ER

3.4.4 Προδιαγραφές ακεραιότητας δεδομένων

Αρχικά οι τιμές που καταχωρεί ένας εθελοντής πρέπει να είναι θετικοί αριθμοί με δύο δεκαδικά ψηφία καθώς επίσης θα λέγεται η εγκυρότητα της τοποθεσίας ενός νέου χώρου στάθμευσης. Οι administrators έχουν την δυνατότητα να ελέγχουν τις αναρτήσεις τιμών και τους εθελοντές, διαγράφοντας ακυρες αναρτήσεις και κακόβουλους εθελοντές που έχουν επανειλημμένα αναρτήσει ακυρες τιμές ή/και χώρους στάθμευσης.

Κανόνες ακεραιότητας και εγκυρότητας δεδομένων

3.4.5 Προδιαγραφές διατήρησης δεδομένων

1. Όταν ένα προϊόν αποσύρεται τότε θεωρείται withdrawn αλλά δεν διαγράφεται από την βάση εκτός αν το διαγράψει κάποιος admin

2. Όταν κάποιος εθελοντής διαγράφει το λογαριασμό του διαγράφεται ο λογαριασμός του αλλά παραμένουν οι καταχωρήσεις του

Απαιτήσεις διατήρησης δεδομένων σε βάθος χρόνου.

3.5 Περιορισμοί σχεδίασης

Οι περιορισμοί στην σχεδίαση του συστήματος οφείλονται στην δομή της βάσης δεδομένων που έχουμε φτιάξει. Πιο αναλυτικά ένας χρήστης δεν θα είναι σε θέση να διαγράψει ένα χώρο στάθμευσης αν δεν είναι administrator. Αν και εφόσον είναι θα έχει το δικαίωμα να αφαιρέσει μια καταχώρηση από την βάση είτε αυτή πρόκειται για account κάποιου εθελοντή, είτε πρόκειται για έναν χώρο στάθμευσης. Εν αντιθέσει ένας εθελοντής η δικαιοδοσία που θα έχει του επιτρέπει απλώς να θέσει ένα χώρο στάθμευσης ως "withdrawn". Επιπλέον τα πεδία που αποθηκεύονται στην βάση δεδομένων έχουν περιορισμούς ως προς την τιμή. Επιγραμματικά τα πεδία που αποθηκεύονται οι τιμές θα μπορεί να είναι μόνο αριθμητική κάμε 2 δεκαδικά ψηφία ενώ τα πεδία συντεταγμένων θα μπορεί να είναι μόνο αριθμητική κάμε 5 δεκαδικά ψηφία.

Λεπτομερής τεχνική τεκμηρίωση των περιορισμών σχεδίασης οι οποίοι επιβάλλονται από απαιτήσεις συμμόρφωσης σε πρότυπα, κανονισμούς, ή άλλους περιορισμούς του έργου. Περιλαμβάνεται η πολιτική ονοματολογίας οντοτήτων δεδομένων και πεδίων. Τέτοιοι περιορισμοί μπορεί να επιβάλλονται από τη χρήση βιβλιοθηκών, frameworks, περιβαλλόντων ανάπτυξης κλπ

3.6 Λοιπές απαιτήσεις

3.6.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Για να είναι το λογισμικό της εφαρμογής διαθέσιμο απαιτείται η σύνδεση στο internet, αφού είναι μια web εφαρμογή, καθώς και η σύνδεση με το API της Google για το embedding των χαρτών στην εφαρμογή.

Τεκμηρίωση απαιτήσεων διαθεσιμότητας

3.6.2 Απαιτήσεις ασφάλειας

Η αποθήκευση των στοιχείων εισόδου είτε του εθελοντή είτε του administrator θα πρέπει να είναι ασφαλής οπότε θα πρέπει να τηρηθούν τα απαραίτητα μέτρα προστασίας, όπως η αποθήκευση κωδικών πρόσβασης με hash function (και όχι σε plaintext) στην βάση δεδομένων, η προστασία έναντι σε sql injection κ.τ.λ.

Τεκμηρίωση απαιτήσεων ασφαλείας

3.6.3 Απαιτήσεις συντήρησης

Οι administrators θα έχουν την δυνατότητα να διαγράφουν κάποια καταχώρηση αν πιστεύουν ότι δεν είναι έγκυρη ή έχει ανανεωθεί. Επιπλέον θα μπορούν να διαγράφουν έναν εθελοντή αν κάνει λανθασμένες ή ακύρες αναρτήσεις. Για το maintenance του web app θα υπάρχει μια ομάδα προγραμματιστών που θα ασχολείται με αυτό προσφέροντας έτι ένα συνεχώς ανανεωμένο περιεχόμενο.

Τεκμηρίωση απαιτήσεων συντήρησης