Έγγραφο απαιτήσεων λογισμικού (SRS)

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

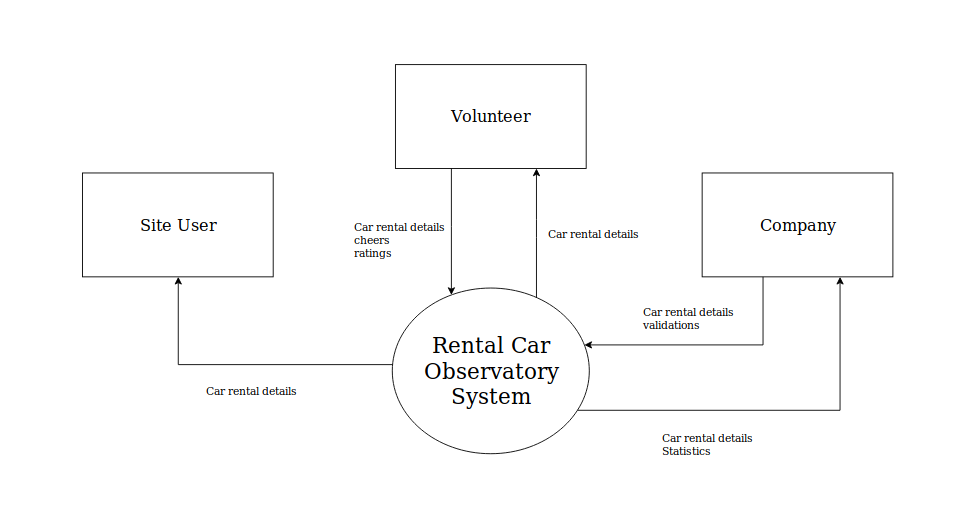
[Rental Car Observatory System]

# Εισαγωγή

## 1.1 Εισαγωγή: σκοπός του λογισμικού

Σκοπός του συστήματος είναι να παρέχει ένα παρατηρητήριο τιμών για ενοικίαση αυτοκινήτων το οποίο τροφοδοτείται από εθελοντές για δημιουργία ανταγωνισμού και εύρεσης χαμηλότερης τιμής από μεγάλους αλλά και μικρούς τοπικούς παρόχους.

## 1.2 Επισκόπηση του λογισμικού

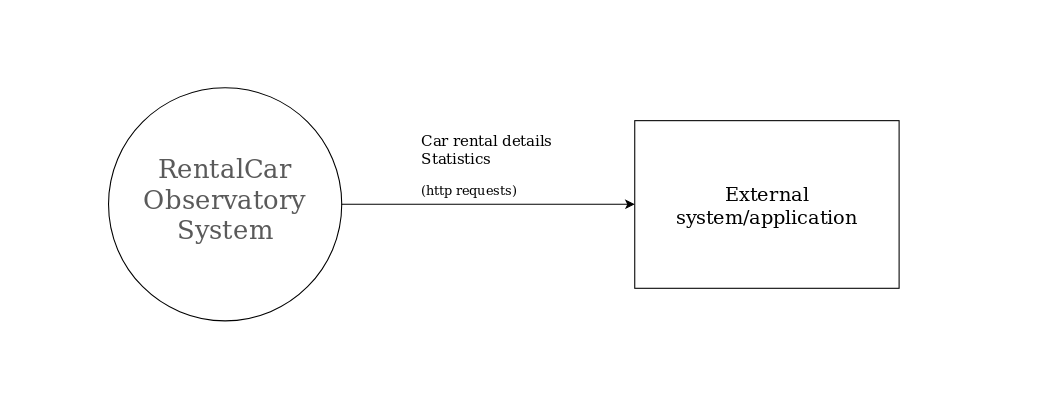


### 1.3.1 Διεπαφές με εξωτερικά συστήματα και εφαρμογές λογισμικού

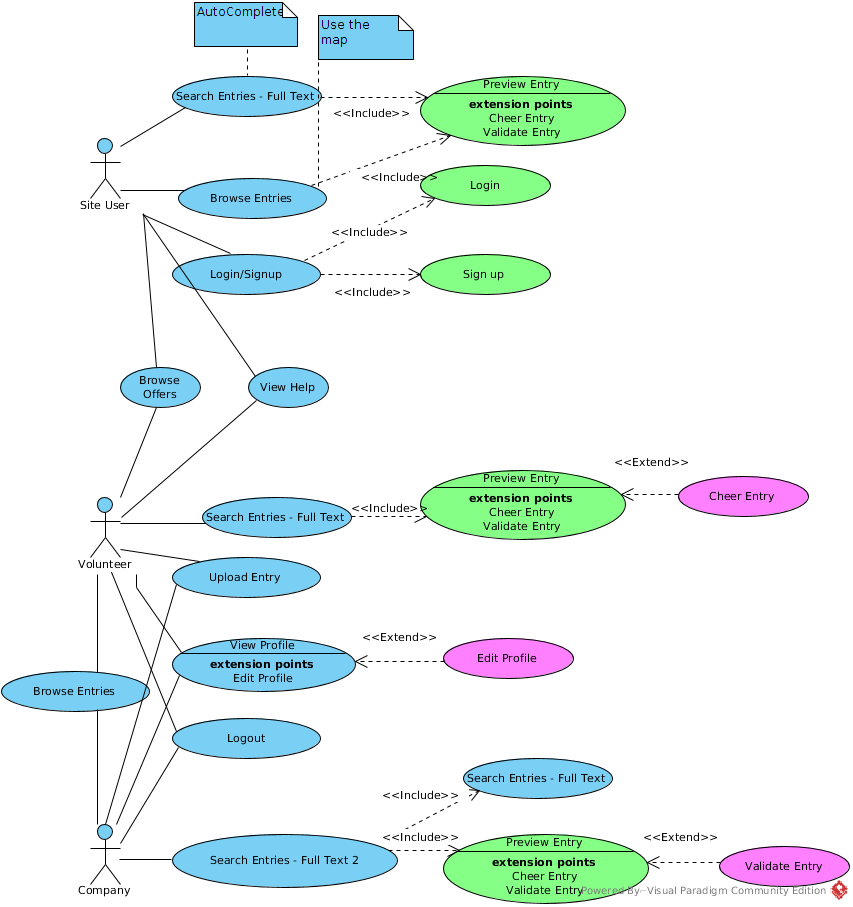
Κάθε εξωτερική ενδιαφερούμενη εφαρμογή/σύστημα για το REST API της εφαρμογής μας θα έχει το δικαίωμα να επικοινωνήσει μαζί μας για να κανονίσουμε κάποια συμφωνία για την παροχή των υπηρεσιών μας σε αυτή.

Οι διάφορες εξωτερικές εφαρμογές/συστήματα θα έχουν διαφορετικά δικαιώματα πρόσβασης ανάλογα με τις συμφωνίες που έχουμε επισυνάψει.

Για την επικοινωνία μαζί μας θα παρέχουμε σχετική ηλεκτρονική διέυθυνση επικοινωνίας.



### 1.3.2 Διεπαφές με το χρήστη



### 1.3.3 Διεπαφές με υλικό

Δεν εφαρμόζεται αφού η εφαρμογή απευθύνεται για χρήστες του web οπότε και χρησιμοποιούμε την τοποθεσία με βάση την πρόσβαση στο ίντερνετ και τη html5.

### 1.3.4 Διεπαφές επικοινωνιών

Δεν εφαρμόζεται.

# Αναφορές - πηγές πληροφοριών

[1] IEEE Standards Association (2011, Last Accessed June 2017) Systems and software

engineering – Life cycle processes –Requirements engineering, Standard. (IEEE 29148-

2011) URL http://standards.ieee.org/findstds/standard/29148-2011.html.

[2] World Wide Web Consortium (2011, Last Accessed June 2017) REST, Web Page. URL

http://www.w3.org/2001/sw/wiki/REST.

# Προδιαγραφές απαιτήσεων λογισμικού

## 3.1 Εξωτερικές διεπαφές

Οι εξωτερικές διεπαφές του λογισμικού μας είναι:

API για την απόκτηση των στατιστικών από την εφαρμογή μας.

API για την εύρεση ενός αυτοκινήτου.

API για την απόκτηση των πληροφοριών ενός αυτοκινήτου.

VOLUNTEER

CAR

(GET /car/{carID}) get car info by id

(GET /car/{carID}/comments) get comments of car

(GET /car/rating) get cars ordered by biggest rating first

(GET /car/cheer) get cars ordered by most cheers first

(GET /car/popular) get cars ordered by popularity

(GET /car/popular) get cars ordered by popularity

(POST /car/{carID}) give car details

(POST /car/{carID}/comment) comment on car

(POST /car/{carID}/cheer) cheer a car by id

(DELETE /car/{carID}/cheer) uncheer a car by id

COMPANY

(GET /company/{companyID}) get company info by id

(GET /company/{companyID}/ \_\_ ) get company \_\_\_\_\_ by id

(GET /carCompany/x/y) get car company by coordinates.

SEARCH

(GET /search/x/y/zoom) get cars by coordinates and zoom level

(GET /search/**filter1=value1,filter2=value2,...**/x/y/zoom) get filtered results of cars by coordinates and zoom level

LISTS

(GET /categories/car/list) get list of all the categories available

(GET /brand/car/list) get list of all the brands available

(GET /model/brand/car/list) get list of all the models of the specific brand available

(GET /company/list) get list of all the companies available

OFFERS

(GET /offer/{offerID}) get offer id information

COMPANY

CAR

(POST /car/{carID}/validate) validate a car by id

OFFERS

(POST /offer/{offerID})

COMPANY

(POST /company/{companyID}/link) give link of the company by id

(POST /company/{companyID}/rating) rate a company by id

## 3.2 Λειτουργίες: περιπτώσεις χρήσης

### 3.2.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: (Upload Entry)

3.2.1.1 Χρήστες (ρόλοι) που εμπλέκονται

Οι ρόλοι που εμπλέκονται στην συγκεκριμένη περίπτωση χρήσης είναι:

* Εθελοντές
* Εμπλεκόμενες Εταιρίες

Οι εθελοντές προσθέτουν νέα καταχώρηση για να εμπλουτίσουν στην εμπειρία των άλλων χρηστών ενώ οι Εταιρίες προσθέτουν με σκοπό να διαφημιστούν. Στην πρώτη περίπτωση πρέπει να είναι ελεγχόμενη λόγο της αξιοπιστίας των δεδομένων ενώ στη δεύτερη πρέπει να είναι ελεγχόμενη προκειμένου να διατηρηθεί αντικειμενικός χαρακτήρας σε όλη την έκταση της εφαρμογής. Αυτό σημαίνει ότι δεν θα μπορεί μια εταιρία να ανεβάσει πολλές καταχωρήσεις σε πολύ μικρό χρονικό διάστημα αφού έτσι θα επηρεάσει το ποσοστό των αυτοκινήτων που έχει η ίδια σε σχέση με το συνολικό. Αντίστοιχα και ένας εθελοντής θα κρίνεται μη αξιόπιστος αν κάνει πολλές καταχωρήσεις για τον ίδιο πάροχο/εταιρία.

#### 3.2.1.2 Προϋποθέσεις εκτέλεσης

Αρχικά απαιτείται και από τον εθελοντή και από την εκάστοτε εταιρία-πάροχο να είναι συνδεδεμένοι ως εθελοντής/εταιρία αντίστοιχα προκειμένου να παρουσίαζεται η επιλογή της προσθήκης νέας καταχώρησης.

Επιπλέον απαιτείται συμπλήρωση όλων των απαραίτητων στοιχείων της καταχώρησης.

#### 3.2.1.3 Περιβάλλον εκτέλεσης

Το περιβάλλον εκτέλεσης είναι διαδικτυακή διεπαφή χρήστη.

#### 3.2.1.4 Δεδομένα εισόδου

Τα δεδομένα εισόδου είναι τα ακόλουθα:

1. Επιλογή κατηγορίας.
2. Επιλογή μάρκας. (Πρέπει να υπάρχουν αυτοκίνητα της μάρκας στην επιλεγμένη κατηγορία και ανάστροφα: πρέπει να υπάρχουν αυτοκίνητα της κατηγορίας για τη επιλεγμένη μάρκα)
3. Επιλογή μοντέλου. (Πρέπει να υπάρχει το μοντέλο από την επιλεγμένη μάρκα και ανάστροφα)
4. Επιλογή τιμής που ενοικιάζεται. (Είναι η τιμή που ο εθελοντής γνωρίζει ότι παρέχεται το αυτοκίνητο ή εναλλακτικά η τιμή που η Εταιρία το διαθέτει).
5. Επιλογή θέσης του παρόχου-Εταιρία του αυτοκινήτου. (Επιλογή στον χάρτη ή με την χρήση της τοποθεσίας της συσκευής. Και στις δύο περιπτώσεις το όνομα του παρόχου/εταιρίας είναι προαιρετικό)
6. Επιλογή επιπλέον παροχών του αυτοκινήτου. (πχ Gps, bluetooth, κα. Η επιλογή γίνεται ανάμεσα σε καθορισμένη λίστα επιλογών. Περαιτέρω χαρακτηριστικά του αυτοκίνητου θεωρούνται περιττές πληροφορίες και δεν καταγράφονται.)
7. Η ημερομηνία της καταγραφής τοποθετείται αυτόματα χωρίς την απαίτηση από χρήστη.

#### 3.2.1.5 Παράμετροι

**Δεν υπάρχουν παράμετροι.**

#### 3.2.1.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

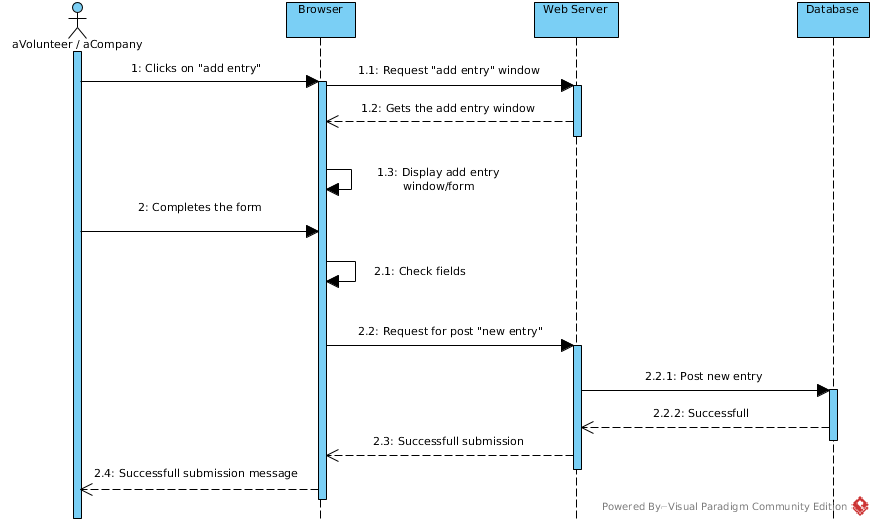
Βήμα 1: Εμφάνιση της φόρμας ειδόδου.

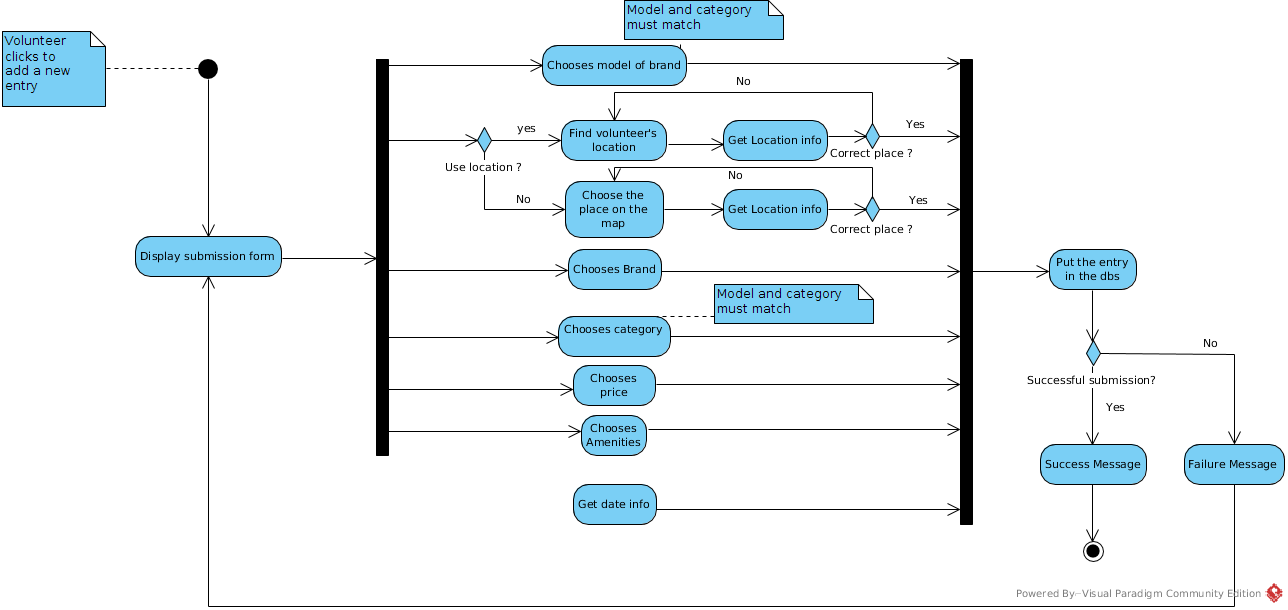
Βήμα 2: Συμπλήρωση των απαραίτητων πεδίων της φόρμας.

Βήμα 3: Αίτηση για προσθήκη νέας καταχώρησης στην βάση δεδομένων.

Βήμα 4: Τοποθέτηση των στοιχείων στην βάση.

Βήμα 5: Εμφάνιση μηνύματος επιτυχημένης καταχώρησης στον χρήστη.





#### 3.2.1.7 Δεδομένα εξόδου

1. Επιλογή κατηγορίας αυτοκινήτου.
2. Επιλογή μάρκας.
3. Επιλογή μοντέλου.
4. Επιλογή τιμής που ενοικιάζεται.
5. Επιλογή θέσης του παρόχου-Εταιρία του αυτοκινήτου
6. Επιλογή επιπλέον παροχών του αυτοκινήτου.
7. Η ημερομηνία της καταγραφής.

### 3.2.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: (Browse Entries)

#### 3.2.2.1 Χρήστες (ρόλοι) που εμπλέκονται

* Χρήστες εφαρμογής
* Εθελοντές
* Εμπλεκόμενες Εταιρίες

Οι χρήστες/εθελοντές αναζητούν καταχωρήσεις με σκοπό να βρούν κάποιο αυτοκίνητο που τους ενδιαφέρει να ενοικιάσουν. Οι Εταιρίες αναζητούν με σκοπό να βρούν τιμές αυτοκινήτων τους ή ανταγωνιστικές τιμές.

#### 3.2.2.2 Προϋποθέσεις εκτέλεσης

Δεν υπάρχουν απαιτήσεις εκτέλεσης. Κάθε χρήστης/εθελοντής/εταιρία μπορεί να χειριστεί τον χάρτη και να δεί τα ενοικιαζόμενα αυτοκίνητα που είναι διαθέσιμα.

#### 3.2.2.3 Περιβάλλον εκτέλεσης

Το περιβάλλον εκτέλεσης είναι διαδικτυακή διεπαφή χρήστη.

#### 3.2.2.4 Δεδομένα εισόδου

**Δεν** υπάρχουν δεδομένα εισόδου. Ο χρήστης απλά χειρίζεται τον χάρτη. Η επιλογή ενός σημείου-καταχώρησης στον χάρτη δείχνει τις λεπτομέρειες της καταχώρησης.

Στην περίπτωση του **εθελοντή** δίνεται δυνατότητα “cheer” ή αλλίως “Like” της καταχώρησης.

Στην περίπτωση της **εταιρίας** δίνεται δυνατότητα “validate” , δηλαδή επικύρωση της καταχώρησης.

#### 3.2.2.5 Παράμετροι

**Δεν υπάρχουν παράμετροι**

#### 3.2.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Βήμα 1: Διάβασε τη λίστα των φίλτρων

Βήμα 2: Καθάρισε τον χάρτη

Βήμα 3: Είναι η λίστα άδεια?

1. Αν ναι, τότε φέρε τις 30 πιο πρόσφατες καταχωρήσεις
2. Αν όχι, τότε φέρε τις 30 πιο πρόσφατες καταχωρήσεις που ικανοποιούν τα φίλτρα

Βήμα 4: Εμφάνισε τις καταχωρήσεις στον χάρτη

Βήμα 5: Περίμενε για συμβάν

Βήμα 6: Ανάλογα με το συμβάν:

1. Αν αλλαχτούν τα φίλτρα τότε πήγαινε στο βήμα 1.
2. Αν ο χρήστης πατήσει στον χάρτη, εμφάνισε το κουμπί “Προσθήκη αυτοκινήτου εδώ?”

Αν ο χρήστης πατήσει το κουμπί τότε :

Βήμα 7: Πήγαινε στο πρώτο βήμα της διαδικασίας: Upload Entry

**Τέλος**.

1. Αν ο χρήστης πατήσει στο κουμπί “Βρες με” τότε

Βήμα 7: Εύρεση της τοποθεσίας του χρήστη.

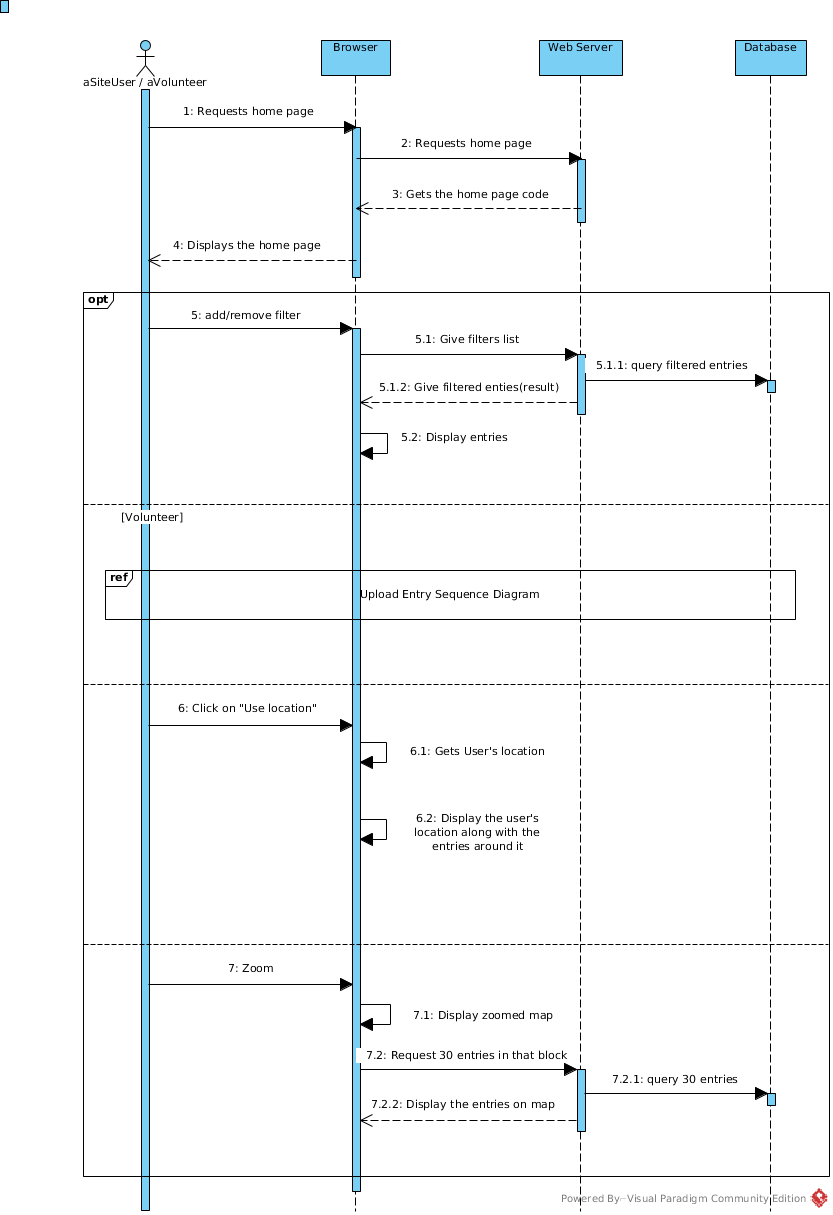
Βήμα 8: Εκκαθάρριση χάρτη

Βήμα 9: Τοποθέτηση της θέσης του χάρτη με κέντρο την θέση του χρήστη και κάνε το απαραίτητο ζουμ.

Βήμα 10: Τοποθέτηση των 30 πιο πρόσφατων καταχωρήσεων στην περιοχή που είναι ορατή.

**!**Αν αποτύχει το βήμα 9 βγάλε σχετικό μήνυμα, καθάρισε το χάρτη και το ζουμ επίπεδο.

1. Αν ο χρήστης κάνει ζουμ μέσα/έξω τότε επανάλαβε τη διαδικασία.



#### 

#### 3.2.2.7 Δεδομένα εξόδου

1. Η θέση του κέντρου του χάρτη. Καθώς ο χρήστης της εφαρμογής χειρίζεται τον χάρτη, η θέση του κέντρου καθορίζει το ποια δεδομένα θα παρουσιάζονται. Επομένως, οι συντεταγμένες του κέντρου του χάρτη στέλνονται συνεχώς στον server και μετά στην βάση δεδομένων με σκοπό να επιστρέφονται τα αυτοκίνητα που αντιστοιχούν στο συγκεκριμένο ορατό κομμάτι του χάρτη.
2. Το ζουμ. Κάθε φορά που ο χρήστης αλλάζει την κατάσταση του ζουμ στέλνεται μια κλήση προς τον web server και την βάση δεδομένω ομοίως με πριν.

## 3.3 Απαιτήσεις επιδόσεων

Ο χρόνος εκτέλεσης των queries θα είναι μικρός και δε θα αποτελεί αντιληπτή καθυστέρηση (κάτω του 1s). Αν απαιτηθεί θα μπορούσαμε να κανονικοποιήσουμε την βάση . Με τη χρήση indexes σε πίνακες με συχνά queries επιτυγχάνουμε αυξημένη επίδοση. Θα υπάρξει και μια πιθανή μέριμνα συγχρονισμού επερωτήσεων κατά την ταυτόχρονη χρήση της εφαρμογής από διαφορετικούς χρήστες.

## 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 εισόδου είναι το προφίλ των εγγεγραμένων χρηστών και των εταιριών, τα δεδομένα μιας καταχώρησης που θα κάνει είτε ένας εθελοντής είτε μια εταιρία και τα σχόλια. Αυτά είναι το μοντέλο, η κατηγορία, η μάρκα, η τοποθεσία,οι φωτογραφίες,η ημερομηνία,η τιμή ενός αυτοκινήτου και μια σύντομη περιγραφή της εμπειρίας (μόνο στο Observation).Τα πεδία κάθε πίνακα θα είναι είτε ακέραιοι αριθμοί , είτε strings μικρού μήκους (της τάξης του 100 και κάτω). Θα φροντίσουμε για το σωστό και γρήγορο χειρισμό 10^4 γραμμών ανά πίνακα ή και λιγότερων ενδεχομένως στα αρχικά στάδια κυκλοφορίας της εφαρμογής. Όλες οι εικόνες και τα avatar θα είναι αποθηκευμένα στον υπολογιστή μας.

### 3.4.2 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

Για να γίνει εισαγωγή στη βάση , πρέπει είτε ο χρήστης είτε η εταιρία να έχουν κάνει σύνδεση για να μπορέσουν να πραγματοποιήσουν κάποια καταχώρηση.  
Στα στατιστικά της επιχείρησής μας, πρόσβαση έχουμε μόνο εμείς ως διαχειριστές (δηλαδή δεν είναι ανοιχτά στο κοινό). Mετά από συμφωνία με άλλες εφαρμογές μπορούμε να έχουμε πιο ανοιχτό API προς αυτές.

Υπάρχουν ωστόσο και κοινά-ανοιχτά στατιστικά και δεδομένα που τα κάνουμε display στους χρήστες της εφαρμογής (πχ αναζήτηση με φίλτρα).

Για προσωπικά δεδομένα (κωδικοί, email) υπάρχει αυξημένη μέριμνα για ασφάλεια. Για τους σκοπούς της ασφάλειας θα χρησιμοποιήσουμε κρυπτογραφία και αποτροπή SQL injection.

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

Μοντέλα δεδομένων UML ή/και ER

### 3.4.4 Προδιαγραφές ακεραιότητας δεδομένων

Για την εγκυρότητα θα πραγματοποιείται έλεγχος εισόδου σε επίπεδο HTML και σε JAVA καθώς επίσης θα μπορούσαν να υπάρχουν triggers σε επίπεδο SQL. Οι τύποι του κάθε πεδίου είναι ρητά ορισμένοι στους πίνακες SQL και κατά την εισαγωγή θα ελέγχεται αν τα δεδομένα έχουν το σωστό τύπο και ικανοποιούν τυχόν περιορισμούς που εμείς θέτουμε στον χρήστη (πχ μέγεθος σχολίου ,format ημερομηνίας κτλ). Θα υπάρχει και σε αυτό το σημείο μέριμνα για αποφυγή SQL injection. Για να αλλάξουν τα δεδομένα της βάσης , πρέπει είτε ο χρήστης είτε η εταιρία να έχουν κάνει σύνδεση για να μπορέσουν να πραγματοποιήσουν κάποιο update.

### 3.4.5 Προδιαγραφές διατήρησης δεδομένων

Θα ακολουθήσουμε μια λογική ανανέωσης των παλιών δίσκων και ενδεχομένως μια λογική αποθήκευσης αντιγράφων ασφαλείας, πχ RISK5. Καθώς επεκτείνεται η βάση με την πάροδο του χρόνου θα προσθέτουμε δίσκους στο σύστημα διατήρησης των δεδομένων (αύξηση του όγκου των δεδομέων με την πάροδο του χρόνου). Με την πάροδο του χρόνου τα παλιά δεδομένα θεωρούνται outdated και μειώνουμε την πληροφορία που κρατάμε από αυτά. Κρατάμε πχ ένα γενικό στατιστικό δείκτη, όπως <<top car του προηγόυμενου έτους>>. Αντιθέτως για τις πρόσφατες ημερομηνίες κρατάμε πιο πυκνές πληροφορίες.

3.5 Περιορισμοί σχεδίασης

Χρήση openlayers

Χρήση angular, typescript

Χρήση restApi

Χρήση visual studio code

Χρήση gradle, spring

Χρήση mysql

Χρήση git

Η ανάπτυξη στα παραπάνω περιβάλλοντα με τα παραπάνω εργαλεία και ο προγραμματισμός στις παραπάνω γλώσσες συνεπάγεται συγκεκριμένους περιορισμούς, πχ ονόματα μεταβλητών με λατινικούς χαρακτήρες.

Νομικά θέματα: Θα πρέπει να δεχόμαστε reports και να διαγράφουμε τα προσβλητικά post αν δεχτούμε report. Θα πρέπει να τηρούμε τους όρους χρήσης της εφαρμογής.

## 3.6 Λοιπές απαιτήσεις

### 3.6.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Η εφαρμογή θα είναι ανεβασμένη σε δωρεάν έκδοση σε κάποια ιστοσελίδα ή και στο playstore. Σε περίπτωση που κάνουμε ενημέρωση του λογισμικού, θα φροντίσουμε ώστε να μπορούν κατά το δυνατό να τρέχουν και οι παλιότερες εκδόσεις. Οι παλιότερες εκδόσεις δε θα είναι διαθέσιμες.

### 3.6.2 Απαιτήσεις ασφάλειας

Όροι χρήσης της εφαρμογής. Οι χρήστες θα πρέπει να δηλώνουν ότι τους αποδέχοντα. Διασφάλιση προσωπικών δεδομένων (mail, κωδικοί). Ούτε εμείς οι ίδιοι δε θα πρέπει να μπορούμε να δούμε κωδικούς χρηστών. Επιστράτευση κρυπτογραφίας, αποτροπή sql injection κτλ. Ανώνυμη πλοήγηση αν δεν είναι συνδεδεμένος χρήστης, μόνο τα «clicks» αποθηκεύουμε και αυτό χωρίς να αποθηκεύσουμε τίποτα για αυτόν που τα έκανε. Τα στατιστικά αναζητήσεων παραμένουν κλειστά στο κοινό και μπορεί να δωθούν επιλεγμένες πληροφορίες και στατιστικά για αυτά μέσω του API μας σε εταιρείες με τις οποίες συνάπτουμε συμφωνίες.

*3.6.3 Απαιτήσεις συντήρησης*

Θα δεχόμαστε σχόλια για την εφαρμογή μας από τους χρήστες, με τη βοήθεια των οποίων θα εντοπίζουμε bugs και θα προσπαθούμε να τα διορθώσουμε. Επίσης θα επεξεργαζόμαστε προτάσεις για βελτίωση της εφαρμογής. Αυτοματοποιημένα τεστ στα οποία θα υποβάλλουμε το λογισμικό καθόλα τα στάδια ανάπτυξης και ζωής. Επέκταση της βάσης δεδομένων κατά τον τρόπο που ορίστηκε στην ενότητα «προδιαγραφές διατήρησης δεδομένων».

# Παράρτημα

## 4.1 Παραδοχές και εξαρτήσεις

Εξάρτηση από API άλλων εφαρμογών (χάρτες). Συμφωνίες με εταιρείες

(ανταλλαγή rewards). Η χρήση της εφαρμογής απαιτεί internet.

4.2 Ακρωνύμια και συντομογραφίες

## 4.3 Υποστηρικτικά έγγραφα, πρότυπα κ.λπ.