

Έγγραφο απαιτήσεων λογισμικού (SRS)

Homemade

1. Εισαγωγή

1.1 Εισαγωγή: σκοπός του λογισμικού

Σκοπός του λογισμικού είναι η υλοποίηση ενός παρατηρητηρίου τιμών σχετικά με προσφορές σπιτικού φαγητού.

1.2 Επισκόπηση του λογισμικού

Βλέπε UML διαγράμματα `user_preview.xml` και `provider_preview.xml`

1.3.1 Διεπαφές με εξωτερικά συστήματα και εφαρμογές λογισμικού

Χρήση λειτουργικότητας χαρτών μέσω google maps ή αντίστοιχου api.

Χρήση Mysql database για την βάση δεδομένων της εφαρμογής.

Χρήση node.js για την λειτουργία του back-end της εφαρμογής με javascript.

Χρήση του framework loopback για την αποδοτική υλοποίηση των απαραίτητων δομών και συναρτήσεων του back-end της εφαρμογής.

1.3.2 Διεπαφές με το χρήστη).

Βλέπε διάγραμμα Use-Case VP.xml

2. Αναφορές - πηγές πληροφοριών

Αναφορά σε πηγές πληροφοριών στο μέτρο της αναγκαιότητας για την κατανόηση του συστήματος

ΝΑ ΜΗΝ ΣΥΜΠΛΗΡΩΘΕΙ ΑΝ ΔΕΝ ΑΠΑΙΤΕΙΤΑΙ

3. Προδιαγραφές απαιτήσεων λογισμικού

3.1 Εξωτερικές διεπαφές

Από το google maps api στα πλαίσια της εφαρμογής χρησιμοποιούνται οι λειτουργίες direction και geocoding. Το πρώτο api χρησιμοποιείται για την εύρεση διαδρομής μεταξύ 2 σημείων ενώ το δεύτερο για την μετατροπή μιας διεύθυνσης σε γεωγραφικές συντεταγμένες(γεωγραφικό μήκος -πλάτος).

Direction url : <https://maps.googleapis.com/maps/api/directions/outputFormat?parameters>

Όπου το `outputFormat` καθορίζει τον τύπο αρχείου των δεδομένων εξόδου του `api(json or xml)`.

Οι παράμετροι που θα χρησιμοποιηθούν είναι `origin`, `destination` και `waypoints`. Οι πρώτες 2 παράμετροι καθορίζουν την αρχή και το τέλος της επιθυμητής διαδρομής και η τρίτη, 1 ή περισσότερα ενδιάμεσα σημεία από τα οποία θέλουμε να περνάει η διαδρομή. Επίσης χρειάζεται και μία παράμετρος `key` η οποία είναι μοναδική για κάθε εφαρμογή και χρησιμοποιείται για την χρέωση των υπηρεσιών.

Geocoding url: <https://maps.googleapis.com/maps/api/geocode/outputFormat?parameters>

Η παράμετρος `outputFormat` έχει τον ίδιο ρόλο όπως και παραπάνω ενώ από της διαθέσιμες παραμέτρους θα χρησιμοποιηθεί η παράμετρος `address` για τον καθορισμό διεύθυνσης προς μετατροπή.

3.2 Λειτουργίες: περιπτώσεις χρήσης

3.2.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Login

3.2.1.1 Χρήστες (ρόλοι) που εμπλέκονται

Users(εγγεγραμμένοι χρήστες), Providers(πάροχοι προϊόντων), Admins(διαχειριστές)

3.2.1.2 Προϋποθέσεις εκτέλεσης

Για να λειτουργήσει σωστά η διαδικασία του login πρέπει ο χρήστης που προσπαθεί να συνδεθεί να διαθέτει έγκυρο λογαριασμό καταχωρημένο στη βάση της εφαρμογής και να γνωρίζει τα απαραίτητα στοιχεία εισόδου(`username`, `password`).

3.2.1.3 Περιβάλλον εκτέλεσης

Η διαδικασία login υλοποιείται και στα 3 τμήματα της εφαρμογής (`front-end`, `back-end`, `dbms`).

3.2.1.4 Δεδομένα εισόδου

Δεδομένα εισόδου είναι το `username` και `password` που πληκτολογεί ο χρήστης που προσπαθεί να συνδεθεί. Επειδή το `username` είναι το mail γίνεται έλεγχος ότι αυτό έχει την απαραίτητη μορφή(example@abc.com) πριν γίνει αναζήτηση στο σύστημα για την εγγυρότητα των στοιχείων, διαφορετικά δεν γίνεται δεκτό.

3.2.1.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Βήμα1: Εισαγωγή `username(mail)` και `password`.

Βήμα2: Έλεγχος στοιχείων εισόδου

Βήμα3: Επιτυχής αντιστοιχία mail – password

Βήμα4: Είσοδος του χρήστη στην εφαρμογή

Αλλαγή κωδικού πεδίου

Μορφοποιήθηκε: Ελληνικά

Εναλλακτικές περιπτώσεις:

Κενά πεδία: Μήνυμα λάθους και ανακατεύθυνση στη σελίδα login

Μορφοποιήθηκε: Ελληνικά

Λάθος αντιστοιχία username-password: Μήνυμα λάθους και ανακατεύθυνση στη σελίδα login

Μορφοποιήθηκε: Ελληνικά

Βλέπε UML login_activity.xml, login_sequence.xml

3.2.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Εισαγωγή νέου προϊόντος

3.2.2.1 Χρήστες (ρόλοι) που εμπλέκονται

Provider

Μορφοποιήθηκε: Ελληνικά

3.2.2.2 Προϋποθέσεις εκτέλεσης

Πρέπει ο χρήστης να διαθέτει ενεργό λογαριασμό παρόχου στην εφαρμογή και να έχει συνδεθεί με τον λογαριασμό αυτό επιτυχώς στην εφαρμογή.

3.2.2.3 Περιβάλλον εκτέλεσης

Η εκτέλεση πραγματοποιείται και στα 3 τμήματα της εφαρμογής.

3.2.2.4 Δεδομένα εισόδου

Απαιτούνται τίτλος προϊόντος, τιμή και μερίδες. Επιπρόσθετα μπορούν να δίνονται κατηγορία και επιπρόσθετα tags και διάρκεια που είναι διαθέσιμο το προϊόν. Γίνεται έλεγχος πως ο τίτλος αποτελείται μόνο από χαρακτήρες, ότι η τιμή είναι αριθμός και οι μερίδες θετικός ακέραιος.

3.2.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Βήμα1: Επιλογή εισαγωγής νέου προϊόντος

Βήμα2: Συμπλήρωση στοιχείων εισαγωγής νέου προϊόντος

Βήμα3: Προσθήκη καταχώρησης στην βάση δεδομένων

Βήμα4: Προβολή μηνύματος επιτυχίας

Εναλλακτικές περιπτώσεις:

Συμπλήρωση πεδίων με μη έγκυες πληροφορίες ή παράληψη συμπλήρωσης κάποιων εκ των υποχρεωτικών πεδίων: Προβολή μηνύματος λάθους και ανακατεύθυνση στην σελίδα εισαγωγής νέου προϊόντος.

Βλέπε διάγραμμα provider-newpost_activity.xml

3.2.2.7 Δεδομένα εξόδου

Δημιουργία μίας καταχώρησης στον πίνακα product και μίας στον πίνακα price.

3.2.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Αναζήτηση προϊόντος

Μορφοποιήθηκε: Ελληνικά

3.2.3.1 Χρήστες (ρόλοι) που εμπλέκονται

Users(εγγεγραμμένοι χρήστες)

3.2.1.2 Προϋποθέσεις εκτέλεσης

Πρέπει ο χρήστης να έχει συνδεθεί επιτυχώς στην εφαρμογή.

3.2.1.3 Περιβάλλον εκτέλεσης

Η εκτέλεση της λειτουργίας υλοποιείται και στα 3 τμήματα της εφαρμογής.

3.2.1.4 Δεδομένα εισόδου

Εισαγωγή διεύθυνσης στην οποία βρίσκεται ο χρήστης και προεραϊτική επιλογή επιπλέον φίλτρων προβολής αποτελεσμάτων. Για την διεύθυνση δεν γίνεται περαιτέρω έλεγχος αλλά αν δεν είναι έγκυρη τότε το αρί που χρησιμοποιείται για το geocoding θα επιστρέψει μήνυμα λάθους.

3.2.1.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Βήμα1: Επιλογή αναζήτησης προϊόντος

Βήμα2:Εισαγωγή στοιχείων διεύθυνσης χρήστη

Βήμα3:Αναζήτηση διαθέσιμων προϊόντων με βάση περιορισμούς γεωγραφικής απόστασης.

Βήμα4: Εφαρμογή επιθυμητών φίλτρων επί των αποτελεσμάτων

Βήμα5: Επανάληψη διαδικασίας ή επιστροφή στην αρχική

Εναλλακτικές περιπτώσεις:

Μη έγκυρη διεύθυνση: Μήνυμα λάθους και επιστροφή στην σελίδα αναζήτησης

3.3 Απαιτήσεις επιδόσεων

Μέτρηση χρόνου απόκρισης του συστήματος αναζήτησης

Μέτρηση της αύξησης του χρόνου απόκρισης συγκριτικά με την άυξηση των διαθέσιμων προϊόντων στην βάση.

Μέτρηση της καθυστέρησης του συστήματος συγκριτικά με το πλήθος των ταυτόχρονα συνδεδεμένων χρηστών.

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.

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

Οι λογαριασμοί εγγεγραμμένων χρηστών έχουν δυνατότητα τροποποίησης μόνο των προσωπικών τους δεδομένων και διαθέτουν read only πρόσβαση στις καταχωρήσεις προϊόντων. Οι λογαριασμοί παρόχων έχουν δυνατότητα τροποποίησης των προσωπικών τους στοιχείων και των καταχωρήσεων προϊόντων που έχουν προέλθει από το λογαριασμό τους ενώ έχουν read only πρόσβαση στις υπόλοιπες καταχωρήσεις προϊόντων. Τέλος οι λογαριασμοί διαχειριστών έχουν δυνατότητα τροποποίησης των προσωπικών τους δεδομένων καθώς και των δικαιωμάτων των λογαριασμών των παρόχων και χρηστών.

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

Βλέπε ER.xml

Μορφοποιήθηκε: Ελληνικά

3.4.4 Προδιαγραφές ακεραιότητας δεδομένων

Ο έλεγχος της ακεραιότητας των δεδομένων γίνεται σε 2 στάδια. Πρώτον στο front end με χρήση patterns ώστε να αποτρέπεται ο χρήστης να εισάγει μη έγκυρα δεδομένα και να μην φτάνουν καν αυτά στη βάση άρα να αποφεύγονται άσκοπες προσπελάσεις σε αυτή. Δεύτερον η ίδια η υλοποίηση της βάσης έχει κανόνες ακεραιότητας των δεδομένων όσων αφορά τον τύπο τους ώστε αν τυχόν κάποια μη έγκυρα δεδομένα φτάσουν στη βάση, να μην γίνει δημιουργία εγγραφής.

3.4.5 Προδιαγραφές διατήρησης δεδομένων

Η διατήρηση των δεδομένων σε βάθος χρόνου παρουσιάζει 2 τεχνικές δυσκολίες. Η πρώτη είναι η διόγκωση του χώρου αποθήκευσης δεδομένων και η δεύτερη είναι η επιβάρυνση του χρόνου αναζητήσεων και ενημερώσεων όσο αυξάνεται ο όγκος πληροφορίας. Η πρώτη δυσκολία μπορεί να αντιμετωπιστεί με την απόκτηση μεγαλύτερου αποθηκευτικού χώρου μέσω διαφόρων διαθέσιμων μεθόδων (cloud, φυσικοί δίσκοι κλπ). Για την δεύτερη περίπτωση υπάρχουν 2 πολιτικές αντιμετώπισης. Είτε γίνεται διαγραφή δεδομένων όταν κάποιος χρήστης πραγματοποιεί διαγραφή λογαριασμού είτε δημιουργούνται πίνακες ιστορικού στη βάση στους οποίους μεταφέρονται οι αντίστοιχες καταχωρήσεις όταν οι χρήστες πραγματοποιούν διαγραφή λογαριασμών.

3.5 Περιορισμοί σχεδίασης

Από θέμα περιορισμών σχεδίασης οι μόνοι που παρουσιάστηκαν αφορούν το λογικό μοντέλο λειτουργίας της εφαρμογής ως προς τον τρόπο που καταχωρούνται τα δικαιώματα πρόσβασης στα δεδομένα στους χρήστες. Αυτός οφείλεται στο framework που χρησιμοποιήθηκε το οποίο από την στιγμή που ορίζει τις αρχικές παραμέτρους της

εφαρμογής ορίζει τους πίνακες role , rolemapping και acl. Ο πρώτος ορίζει τον ρόλο κάθε χρήστη στην εφαρμογή, ο δεύτερος αντιστοιχίζει κάθε χρήστη σε 1 ρόλο και ο τρίτος περιέχει τα δικαιώματα κάθε χρήστη ως προς την πρόσβαση και τροποποίηση των δεδομένων.

3.6 Λοιπές απαιτήσεις

3.6.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Η χρήση των google maps api απαιτεί την καταβολή χρηματικού ποσού ανάλογα με τη χρήση επομένως πιθανότητα θα οδηγήσει στην αναζήτηση νέων δωρεάν εναλλακτικών.

Όλα τα υπόλοιπα τμήματα λογισμικού είναι δωρεάν και διαθέσιμα.

3.6.2 Απαιτήσεις ασφάλειας

Τα ευαίσθητα δεδομένα πρέπει να αποθηκεύονται κρυπτογραφημένα στη βάση καθώς και η ανταλλαγή όλων των δεδομένων μεταξύ client server πρέπει να γίνεται με τη χρήση κάποιας μεθόδου κρυπτογραφίας. Επίσης πρέπει να χρησιμοποιείται το πρωτόκολλο HTTPS.

4. Παράρτημα

4.3 Υποστηρικτικά έγγραφα, πρότυπα κ.λπ.

Τα απαραίτητα σχεδιαγράμματα UML επισυνάπτονται στον ίδιο φάκελο του github μαζί με το παρόν αρχείο.