

cheapies.gr

Έγγραφο απαιτήσεων Λογισμικού (SRS) Software Requirements Specification

ΒΑΣΙΣΜΕΝΗ ΣΤΟ ΠΡΟΤΥΠΟ ISO/IEC/IEEE 29148:2011

ΓΙΑ ΤΟ ΜΑΘΗΜΑ “ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ”

ΤΗΣ ΣΧΟΛΗΣ ΗΜΜΥ ΕΜΠ

ΟΜΑΔΑ ΑΝΑΠΤΥΞΗΣ

“mycoderocks”

ΜΕΛΗ ΟΜΑΔΑΣ (ΑΛΦΑΒΗΤΙΚΑ)

ΙΩΑΝΝΗΣ ΔΑΡΑΣ (03115018)

ΔΗΜΗΤΡΗΣ ΚΕΛΕΣΗΣ (03115037)

ΜΑΡΙΟΣ ΠΑΠΑΧΡΗΣΤΟΥ (03115101)

ΚΩΝΣΤΑΝΤΙΝΟΣ ΣΤΑΥΡΟΠΟΥΛΟΣ (03115046)

ΙΩΑΝΝΗΣ ΣΙΑΧΟΣ (03115109)

ΔΗΜΗΤΡΗΣ ΧΡΗΣΤΟΥ (03115058)

| | |
|--|-----------------|
| <u>Εισαγωγή</u> | <u>4</u> |
| 1.1 Σκοπός Λογισμικού | 4 |
| 1.2 Επισκόπηση του λογισμικού | 5 |
| 1.2.1 Διεπαφές με εξωτερικά συστήματα και εφαρμογές λογισμικού | 5 |
| 1.2.2 Διεπαφές με το χρήστη | 8 |
| 1.2.3 Διεπαφές με υλικό | 9 |
| 2. Προδιαγραφές απαιτήσεων λογισμικού | 10 |
| 2.1 Εξωτερικές διεπαφές | 10 |
| 2.2 Λειτουργίες: περιπτώσεις χρήσης | 19 |
| 2.2.1 Περίπτωση Χρήσης 1: Καταχώρηση Προϊόντος | 19 |
| 2.2.1.1 Χρήστες (ρόλοι) που εμπλέκονται | 19 |
| 2.2.1.2 Προϋποθέσεις εκτέλεσης | 19 |
| 2.2.1.3 Περιβάλλον εκτέλεσης | 19 |
| 2.2.1.4 Δεδομένα εισόδου | 19 |
| 2.2.1.5 Παράμετροι | 20 |
| 2.2.1.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά | 20 |
| 2.2.2 Περίπτωση Χρήσης 2: Αναζήτηση Προϊόντος | 22 |
| 2.2.2.1 Χρήστες (ρόλοι) που εμπλέκονται | 22 |
| 2.2.2.2 Προϋποθέσεις εκτέλεσης | 22 |
| 2.2.2.3 Περιβάλλον εκτέλεσης | 22 |
| 2.2.2.4 Δεδομένα εισόδου | 23 |
| 2.2.2.5 Παράμετροι | 23 |
| 2.2.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά | 23 |
| 2.2.3 Περίπτωση Χρήσης 3: Διαχείριση Πλατφόρμας | 25 |
| 2.2.3.1 Χρήστες (ρόλοι) που εμπλέκονται | 25 |
| 2.2.3.2 Προϋποθέσεις εκτέλεσης | 25 |

| | |
|--|-----------|
| 2.2.3.3 Περιβάλλον εκτέλεσης | 25 |
| 2.2.3.4 Δεδομένα εισόδου | 25 |
| 2.2.3.5 Παράμετροι | 26 |
| 2.2.3.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά | 26 |
| 2.3 Απαιτήσεις επιδόσεων | 27 |
| 2.4 Απαιτήσεις οργάνωσης δεδομένων | 27 |
| 2.4.1 Περιγραφή Δεδομένων | 27 |
| 2.4.2 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα | 28 |
| 2.4.3 Μοντέλο δεδομένων (UML Class Diagram/ER Diagram) | 29 |
| 2.4.4 Προδιαγραφές ακεραιότητας δεδομένων | 30 |
| 2.4.5 Προδιαγραφές διατήρησης δεδομένων | 31 |
| 2.5 Περιορισμοί σχεδίασης | 31 |
| 2.6 Λοιπές απαιτήσεις | 33 |
| 2.6.1 Απαιτήσεις διαθεσιμότητας λογισμικού | 33 |
| 2.6.2 Απαιτήσεις ασφάλειας | 34 |
| 2.6.3 Απαιτήσεις συντήρησης | 34 |
| 3. Παράρτημα | 35 |
| 3.1 Παραδοχές και εξαρτήσεις | 35 |
| 3.2 Ακρωνύμια και συντομογραφίες | 35 |
| 3.3 Υποστηρικτικά έγγραφα | 35 |

1. Εισαγωγή

1.1 Σκοπός Λογισμικού

Το εγχείρημα της ομάδας *mycoderocks* με την ανάπτυξη του project *cheapies.gr* είναι η δημιουργία μιας διαδικτυακής πλατφόρμας, διαθέσιμης *δωρεάν* σε όλο το ελληνικό κοινό είτε από κινητή συσκευή είτε από ηλεκτρονικό υπολογιστή, στην οποία ο κάθε χρήστης θα μπορεί εύκολα και γρήγορα να ενημερώνεται για κοντινά σε αυτόν προϊόντα καθημερινής χρήσης, έχοντας πρόσβαση σε όλες τις σημαντικές για αυτόν πληροφορίες όπως η τοποθεσία του καταστήματος, το κόστος και η περιγραφή του κάθε προϊόντος.

Ταυτόχρονα, η πλατφόρμα θα λειτουργεί ως ένα διαδικτυακό παρατηρητήριο τιμών όπου ο κάθε χρήστης θα έχει τη δυνατότητα να καταγράφει και να μοιράζεται με άλλους χρήστες ενδιαφέροντα ή ευκαιριακά κατά την άποψή του προϊόντα καθώς και τις πληροφορίες που τα αφορούν (πληθοπορισμός - *crowdsourcing*). Με αυτόν τον τρόπο θα δημιουργηθεί ένα δυναμικό περιβάλλον που είναι πλήρως διαμορφώσιμο από τους ίδιους τους χρήστες που τελικά εξυπηρετεί και κατά συνέπεια ικανό να προσαρμοστεί στις ανάγκες τους.

Στόχος μας είναι η δημιουργία ενός σύγχρονου περιβάλλοντος το οποίο θα χρησιμοποιείται από χιλιάδες χρήστες σε καθημερινή βάση και θα διευκολύνει την καθημερινότητά τους σχετικά με τον τομέα των αγορών. Η εφαρμογή καλών πρακτικών ανάπτυξης λογισμικού, η χρηστικότητα της πλατφόρμας τόσο ως προς την εύρεση όσο και ως προς την καταχώρηση του προϊόντος, η τεκμηρίωση και η συντήρηση του κώδικα και η αποδοτική οργάνωση της βάσης δεδομένων αποτελούν θεμελιώδεις στόχους για εμάς.

Ιδιαίτερη προσοχή έχει δοθεί στη χρηστικότητα και την απλότητα της εφαρμογής ώστε ο κάθε χρήστης να έχει μία ευχάριστη εμπειρία χρήσης. Πέραν των βασικών δυνατοτήτων καταχώρησης προϊόντος, καταστήματος, χρονικού και χωρικού αποτυπώματος και τιμής, το περιβάλλον είναι εφοδιασμένο με δυνατότητες εύκολης και αποδοτικής αναζήτησης (χρήση φίλτρων και συστήματος αγαπημένων), παρακολούθησης και επισκόπησης των δεδομένων (πίνακες, χάρτες, γραφήματα και σύστημα υποστήριξης ερωτοαπαντήσεων).

Η προσεκτική υλοποίηση της πλατφόρμας ώστε να πληρεί τις προδιαγραφές που αναφέραμε θα οδηγήσει στη θετική διαφοροποίηση του λογισμικού μας από άλλα παρεμφερή λογισμικά της αγοράς και συνεπακόλουθα μέσω αυτής θα πετύχουμε τον μεγαλύτερο στόχο μας: να κερδίσουμε την εμπιστοσύνη του χρήστη και να τον πείσουμε ότι το λογισμικό μας είναι το κατάλληλο μέσο με το οποίο μπορεί να οργανώνει τις αγορές του σύμφωνα πάντα με το δικό του συμφέρον και με εύκολο και χρηστικό τρόπο.

1.2 Επισκόπηση του λογισμικού

1.2.1 Διεπαφές με εξωτερικά συστήματα και εφαρμογές λογισμικού

Το σύστημά μας κάνει χρήση άλλων εφαρμογών με στόχο την καλύτερη εξυπηρέτηση και εμπειρία των χρηστών του. Πιο συγκεκριμένα, χρησιμοποιούμε τα εξής API's τρίτων εφαρμογών:

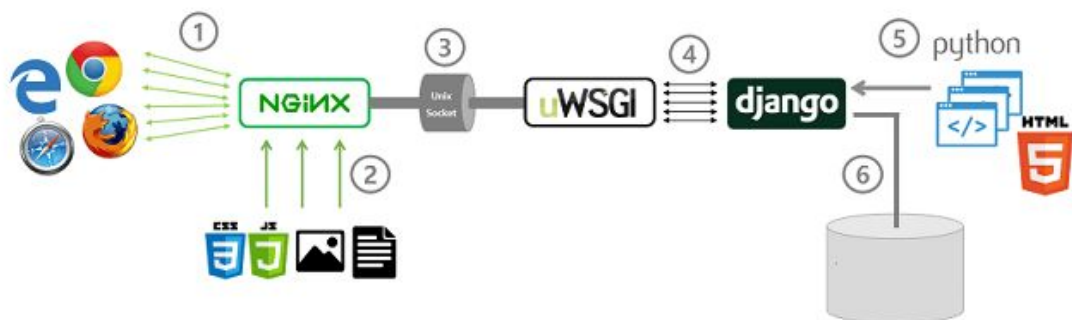
1. **Speech Recognition API (Google):** Πρόκειται για μία εφαρμογή την οποία χρησιμοποιούμε ως έναν εναλλακτικό τρόπο αναζήτησης προϊόντων αντί για την πληκτρολόγηση. Με αυτόν τον τρόπο δίνουμε στο χρήστη τη δυνατότητα να αναζητήσει κάποιο προϊόν καταγράφοντας το αίτημά του με το μικρόφωνο της συσκευής του (εφόσον αυτό υπάρχει). Έτσι παρέχουμε ένα πιο ευέλικτο περιβάλλον στον χρήστη βελτιώνοντας τη συνολική του εμπειρία χρήσης της εφαρμογής μας.
2. **OpenStreetMaps Geocoding API (Nominatim API):** Η εφαρμογή αυτή μας επιτρέπει την αντιστοίχιση τοπωνυμίων σε συντεταγμένες. Με αυτόν τον τρόπο αντιστοιχίζουμε τις συντεταγμένες μιας διεύθυνσης στο όνομά της και κρύβουμε από το χρήστη όλη την πληροφορία που για αυτόν δεν είναι σημαντική, αλλά έχει σημασία για το σύστημα (για παράδειγμα υπολογισμός αποστάσεων, κοντινότερου καταστήματος κ.α).
3. **OpenStreetMaps Maps API:** Πρόκειται για την εφαρμογή που μας δίνει τη δυνατότητα παρουσίασης χαρτών στο χρήστη. Έτσι μπορεί άμεσα από το περιβάλλον της εφαρμογής μας να λάβει όλη την πληροφορία που χρειάζεται για να δει την τοποθεσία ενός καταστήματος που παρέχει ένα ενδιαφέρον για αυτόν προϊόν.

Εκτός των παραπάνω (εξωτερικών) διεπαφών, το σύστημά μας λειτουργεί με την υποστήριξη των ακόλουθων εσωτερικών διεπαφών:

1. **MySQL RDBMS Server:** Πρόκειται για το σύστημα με το οποίο διαχειριζόμαστε τη βάση δεδομένων του συστήματός μας. Επιλέξαμε τη MySQL Server που καλύπτει με μεγάλη άνεση τις σχετικά περιορισμένες απαιτήσεις του συστήματος από άποψη βάσεων δεδομένων.
2. **nginx Webserver:** Για την επικοινωνία της εφαρμογής μας με τον έξω κόσμο χρειαζόμαστε έναν webserver. Η ελάχιστη απαίτηση που έχουμε από έναν web server είναι να μπορεί να εξυπηρετεί requests που έρχονται από τον παγκόσμιο ιστό. Επιλέγουμε τον nginx που εκτός από αυτή την ελάχιστη απαίτηση, προσφέρει μια πληθώρα άλλων features όπως: load balancing, caching, handling of static files, autoindexing, κ.α. Παρόλα αυτά, ο webserver δεν μπορεί να επικοινωνήσει απευθείας με τα django applications. Η επικοινωνία του nginx με τις εφαρμογές του django γίνεται έμμεσα, με την ενδιάμεση παρεμβολή του uwsgi.
3. **Django/uWSGI Webserver:** ο uWSGI είναι μια υλοποίηση του WSGI (A Web Server Gateway Interface). Ρόλος του είναι να μεσολαβήσει ανάμεσα στις

εφαρμογές του django και τον nginx webserver. Πιο συγκεκριμένα, ο uwsgi επικοινωνεί με τον nginx μέσω ενός unix socket. Ο uwsgi αναλαμβάνει να κάνει parse τις απαντήσεις του nginx και να μεταδώσει τις κατάλληλες πληροφορίες στην εφαρμογή. Αλλά και αντίστροφα, ο uwsgi αναλαμβάνει να μεταδώσει πληροφορίες στον nginx από την εφαρμογή, όπως ένα html file ή μια πληροφορία της βάσης δεδομένων.

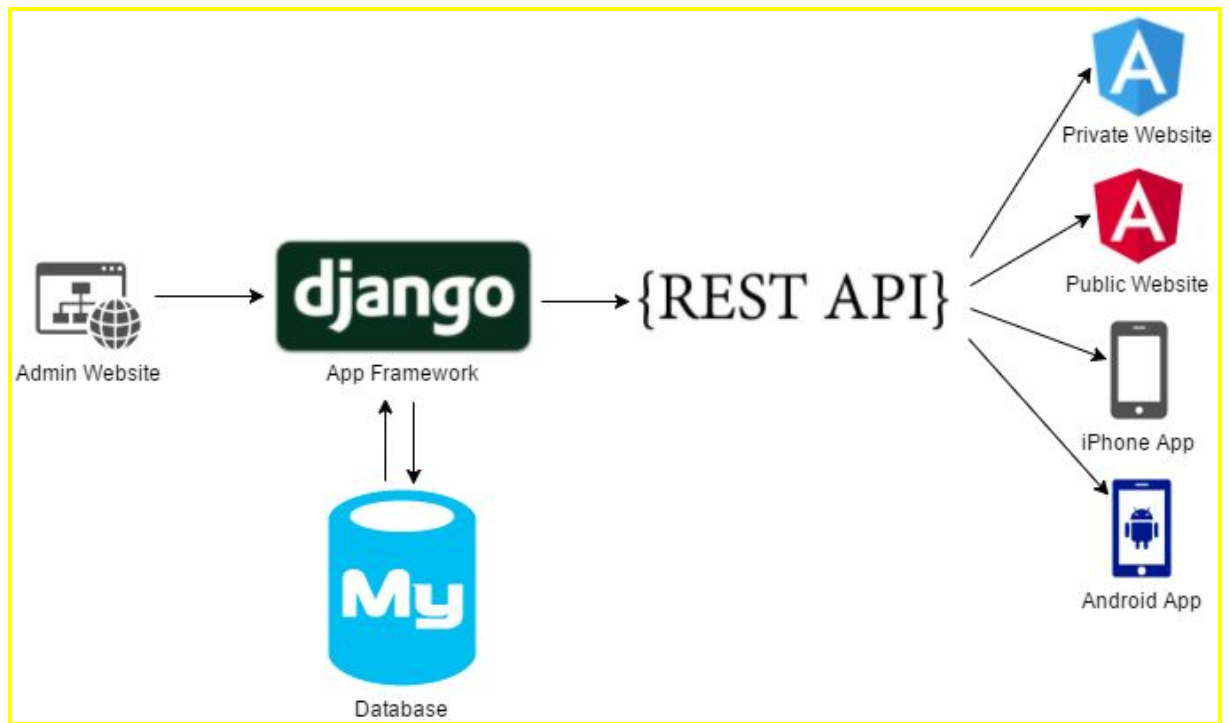
Τα παραπάνω, φαίνονται σχηματικά στην ακόλουθη εικόνα:



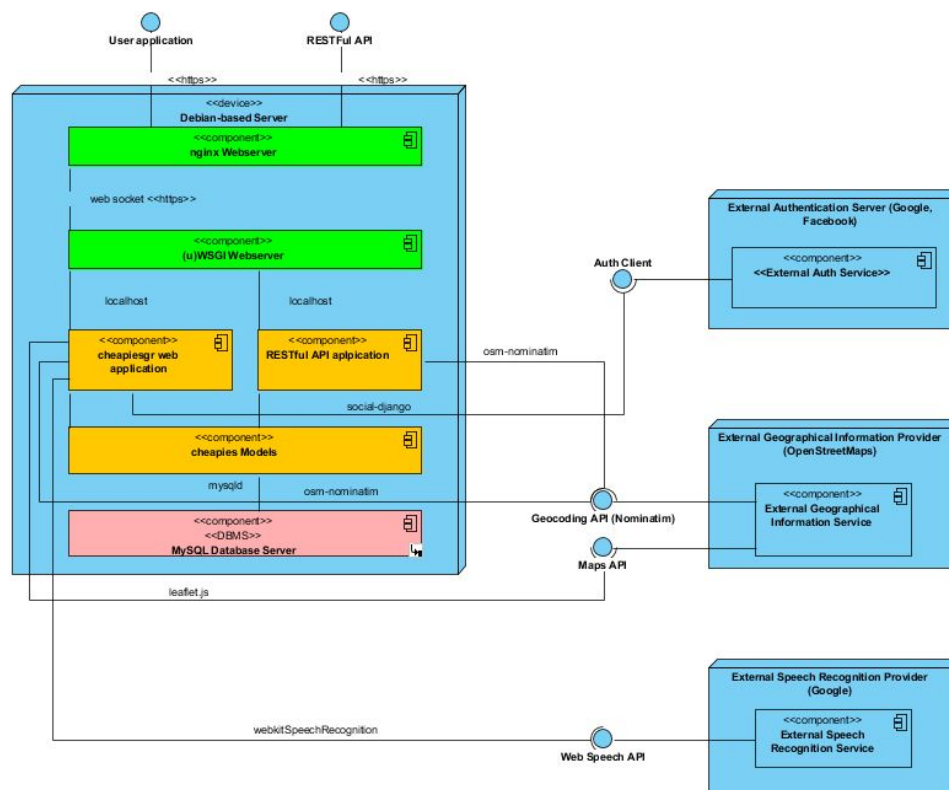
Τέλος, το λογισμικό παρέχεται στους χρήστες με δύο τρόπους-διεπαφές:

1. **Front-End εφαρμογή:** Μέσα από τον ιστότοπο του chearies.gr ο χρήστης θα έχει τη δυνατότητα να εγγραφεί στο σύστημα, να διαμορφώσει το προφίλ του και φυσικά να αναζητήσει και να καταχωρήσει προϊόντα. Πρόκειται για το βασικό κορμό του συστήματός μας και όλες οι δυνατότητες που προσφέρονται είναι αξιοποιήσιμες μέσα από το front-end περιβάλλον της εφαρμογής.
2. **RESTful API:** Η εφαρμογή μας παρέχει ένα endpoint για διαπιστευμένη (authorized), περιορισμένη (restricted) πρόσβαση στη βάση δεδομένων μας, μέσω ενός RESTful API. Το RESTful API βασίζεται στο πρότυπο αρχιτεκτονικής σχεδίασης REST (REpresentational State Transfer). Χρησιμοποιεί HTTP requests (get, put, post, delete, patch) για την επικοινωνία δύο προγραμμάτων. Το RESTful API που αναπτύσσουμε είναι διαθέσιμο μέσω του base url: https://{server_name}:8765/observatory/api

όπου το server_name αντιστοιχεί μέσω DNS στην IP ενός υπολογιστή. Το API υποστηρίζει τον μορφότυπο δεδομένων JSON. Για να διαπιστευθεί ο χρήστης στο API κάνει ένα post request με το όνομα του και τον κωδικό του και περιμένει ως απάντηση ένα authorization token το οποίο χρησιμοποιεί στη συνέχεια για όλες τις υπόλοιπες κλήσεις του στα headers σε περίπτωση που η κλήση θέλει ταυτοποίηση. Το authorization token αυτό γίνεται invalid όταν ο χρήστης κάνει logout ή όταν κάνει expire.



Τα παραπάνω συνοψίζονται με το ακόλουθο component/deployment UML διάγραμμα:

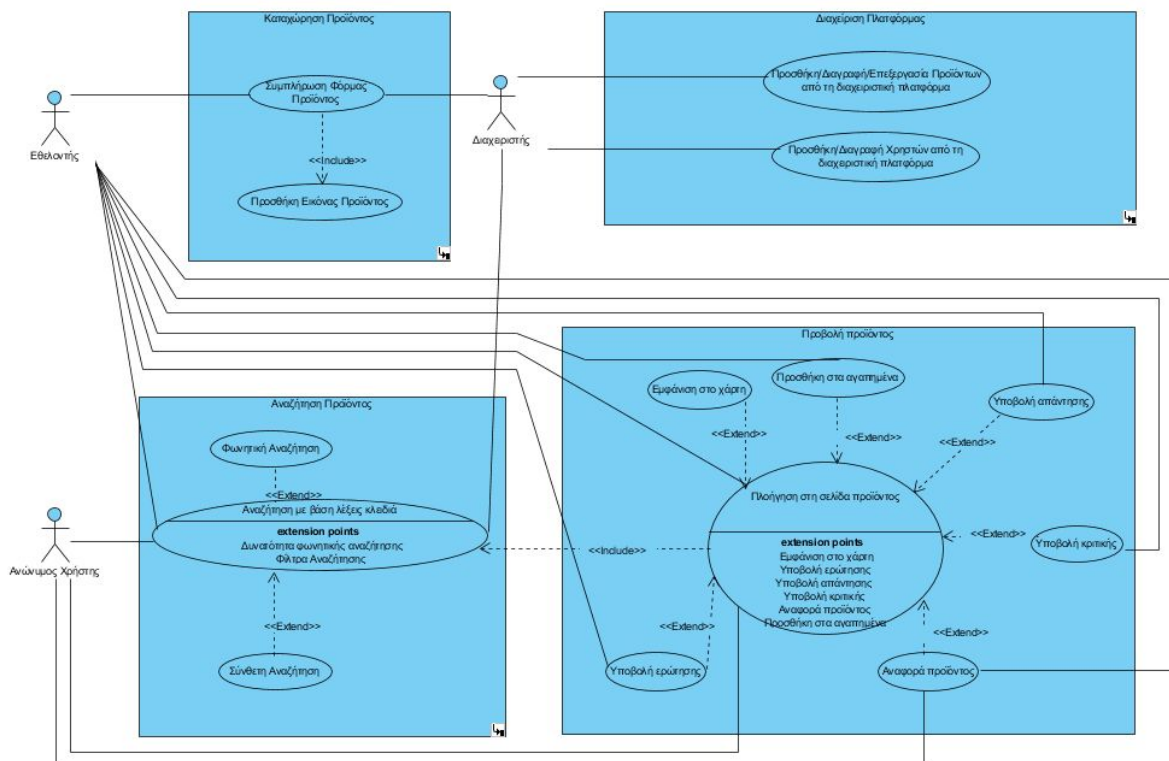


1.2.2 Διεπαφές με το χρήστη

Η πλατφόρμα μας αναγνωρίζει τους εξής τρεις ρόλους χρηστών:

1. **Εγγεγραμμένος χρήστης – Εθελοντής πληθοπορισμού:** Πρόκειται για τον χρήστη του front-end υποσυστήματος ή/και του RESTful API τόσο για την πλοήγηση, αναζήτηση και ανάκτηση των δεδομένων, όσο και για την καταχώρηση και ενημέρωσή τους. Επειδή δίνεται η δυνατότητα σε αυτούς τους χρήστες να μεταβάλλουν το περιεχόμενο της εφαρμογής, απαιτείται πρώτα η εγγραφή τους στο σύστημα ώστε τέτοιες αλλαγές να γίνονται επώνυμα.
2. **Διαχειριστής:** Πρόκειται για τον χρήστη του back-end υποσυστήματος που είναι υπεύθυνος για τη διαχείριση των λογαριασμών χρηστών της πρώτης κατηγορίας. Έχει τη δυνατότητα ανάθεσης ρόλων σε χρήστες, διαγραφής των λογαριασμών τους και τροποποίησης των καταχωρήσεων που βρίσκονται στο σύστημα όταν αυτό κρίνεται απαραίτητο. Επίσης, έχει δικαιώματα εγγεγραμμένου χρήστη.
3. **Αναγνώστης:** Πρόκειται για την πιο απλή κατηγορία χρηστών για τους οποίους υποστηρίζεται ανώνυμη χρήση του front-end υποσυστήματος μέσω web σε desktop ή φορητή συσκευή. Παρέχονται δυνατότητες αναζήτησης προϊόντων και πρόσβασης στις καταχωρήσεις, αλλά όχι προσθήκης νέων καταχωρήσεων.

Σε επόμενη ενότητα αναλύονται διεξοδικά οι περιπτώσεις χρήσης για την καταχώρηση και την αναζήτηση προϊόντος, καθώς και για τη δυνατότητα διαχείρισης της πλατφόρμας (UML Use-Case diagrams). Στη συνέχεια παρατίθεται το ακόλουθο διάγραμμα που ενθυλακώνει τη βασική αλληλεπίδραση ανάμεσα στο σύστημά μας και τους χρήστες του:



1.2.3 Διεπαφές με υλικό

Δεδομένου ότι ούτε η εφαρμογή για κινητά ούτε η δικτυακή πύλη έχουν κάποιο καθορισμένο υλικό, δεν υπάρχουν κάποιες άμεσες διασυνδέσεις διεπαφών. Ωστόσο, για ορισμένες λειτουργίες του συστήματος είναι αναγκαία η επικοινωνία με συγκεκριμένα κομμάτια υλικού (hardware components). Συγκεκριμένα, για την καταχώρηση ενός προϊόντος πρέπει ο χρήστης να εισάγει μία φωτογραφία με την κάμερά του. Επίσης όπως αναφέραμε και παραπάνω δίνεται η δυνατότητα φωνητικής αναζήτησης. Σε αυτήν την περίπτωση απαιτείται η διασύνδεση με το μικρόφωνο. Τέλος απαιτείται ο προσδιορισμός της τοποθεσίας χρήστη μέσω GPS ή του διαδικτύου ώστε να γίνεται εύκολα η ανάκτηση της τοποθεσίας του και να λειτουργούν τα φίλτρα απόστασης που το σύστημά μας διαθέτει.

2. Προδιαγραφές απαιτήσεων λογισμικού

2.1 Εξωτερικές διεπαφές

Speech Recognition API (Google)

Το Google Cloud Speech-to-Text επιτρέπει στους προγραμματιστές να μετατρέπουν τον ήχο σε κείμενο χρησιμοποιώντας μοντέλα deep neural networks σε ένα εύκολο στη χρήση API. Το API αναγνωρίζει 120 γλώσσες και παραλλαγές για να υποστηρίξει τη φωνητική αναζήτηση σε πραγματικό χρόνο στην αρχική σελίδα της εφαρμογής. Χρησιμοποιείται μέσω JavaScript στην εφαρμογή στην αρχική σελίδα της αναζήτησης. Ο χρήστης πατάει το κουμπί της φωνητικής αναζήτησης και εκφωνεί την περιγραφή του προϊόντος που ψάχνει. Εν συνεχεία, γίνεται κλήση στο SpeechRecognition API και καταχώρηση του κειμένου αναζήτησης στο πεδίο της φόρμας.

OpenStreetMaps Geocoding API (Nominatim API)

Το Nominatim API είναι μια υπηρεσία του OpenStreetMaps που επιτρέπει στη μετάφραση τοπωνυμίων σε γεωγραφικές συντεταγμένες (γεωγραφικό μήκος - γεωγραφικό πλάτος). Χρησιμοποιείται για την εύρεση των διευθύνσεων των καταστημάτων και των προϊόντων. Η επικοινωνία γίνεται μέσω RESTful API που προσφέρεται από το Nominatim API. Κατά την καταχώρηση καταστήματος στην εφαρμογή (στην περίπτωση που θέλει ο χρήστης), οι γεωγραφικές συντεταγμένες του καταστήματος λαμβάνονται αυτόματα μετά από GET request στο Nominatim API.

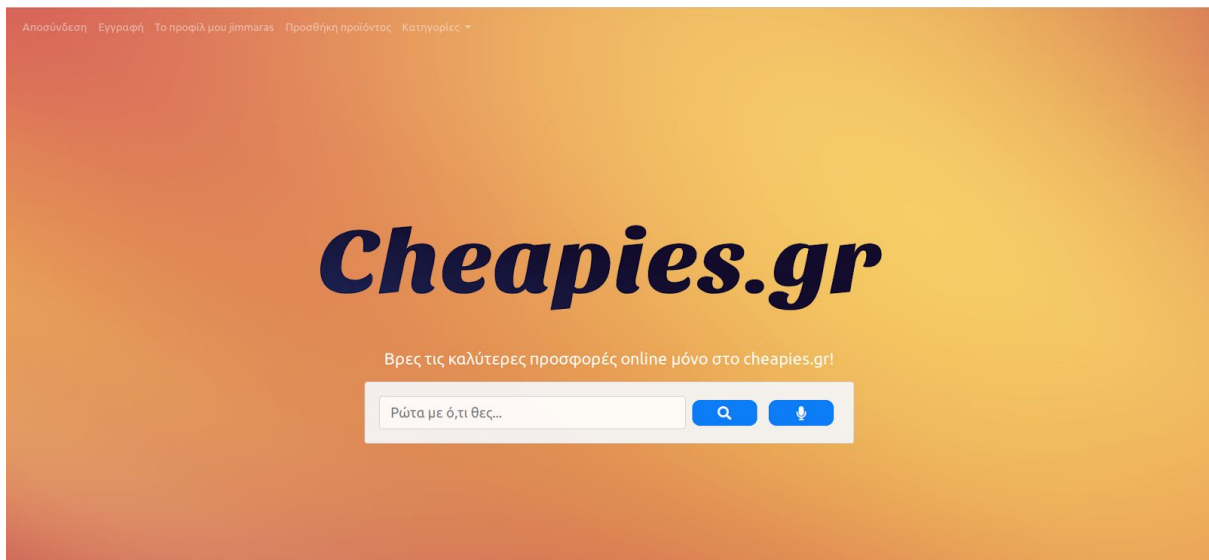
OpenStreetMaps Maps API

Το Maps API προσφέρει πρόσβαση σε χάρτες του OpenStreetMaps. Τα δεδομένα χαρτών γίνονται render στην εφαρμογή με χρήση της βιβλιοθήκης leaflet.js σε interactive widgets. Χρησιμοποιούνται για την προβολή των τοποθεσιών των προϊόντων σε χάρτες. Η επικοινωνία γίνεται μέσω RESTful API που προσφέρεται από το Maps. Ενδεικτικά στην επόμενη υποενότητα φαίνονται στιγμιότυπα του χάρτη.

Σχετικά με τη διεπαφή χρηστών στο front-end application του συστήματος, παρουσιάζονται τα ακόλουθα επεξηγηματικά mockups που παρουσιάζουν τις διάφορες δυνατότητες που παρέχονται στο χρήστη κατά την πλοήγησή του στην εφαρμογή:

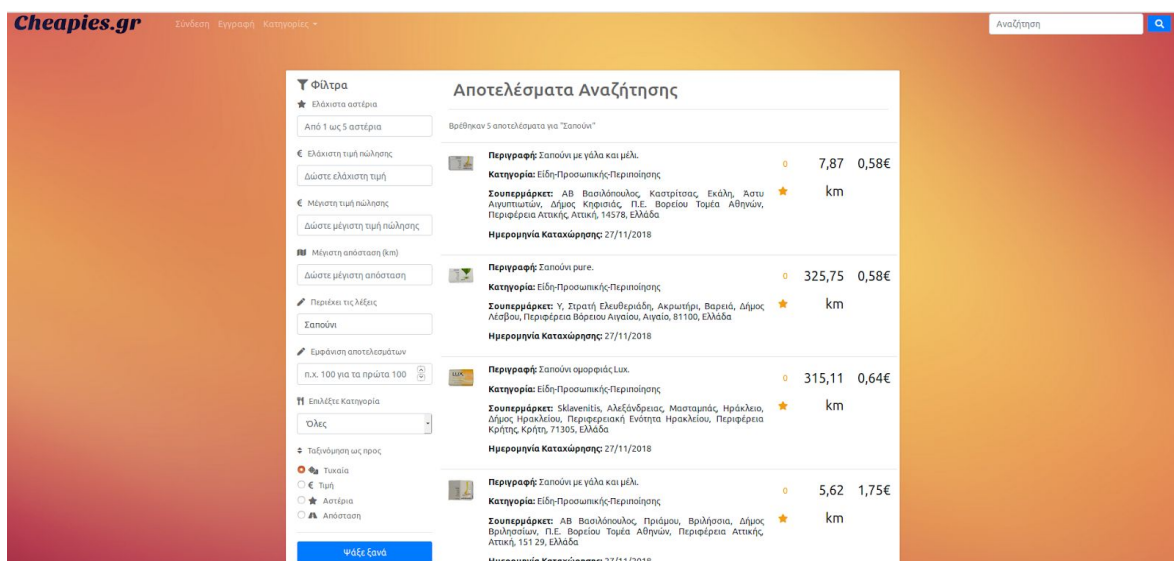
Σελίδα Αναζήτησης

Με την είσοδό του στην εφαρμογή, παρουσιάζουμε στο χρήστη την κεντρική σελίδα με δυνατότητα αναζήτησης και σύνδεσης/έγγραφης.



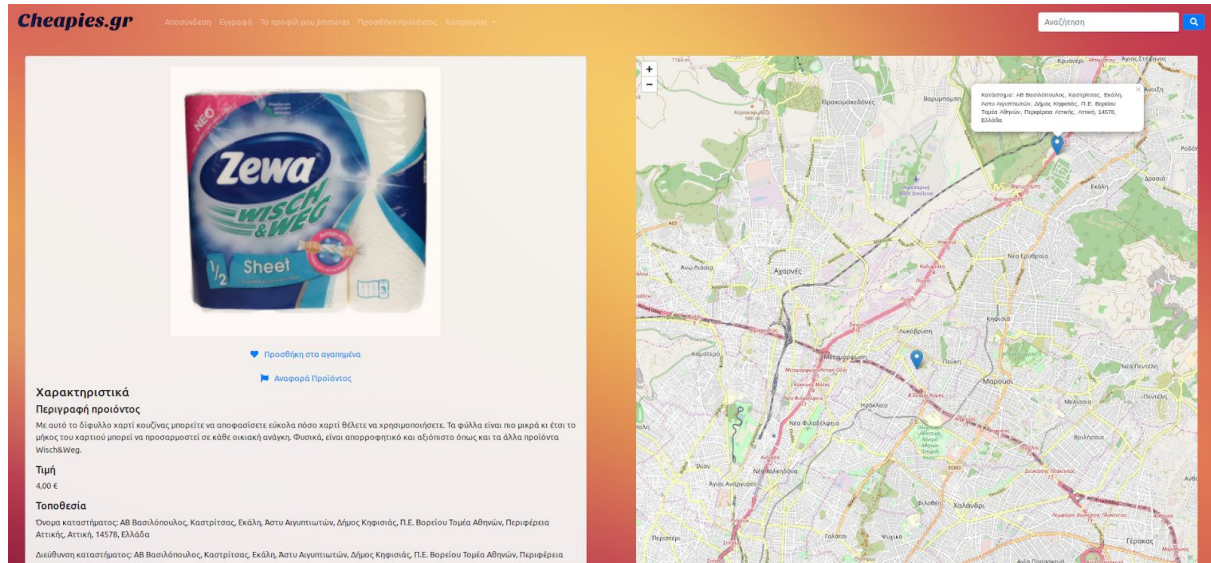
Σελίδα Αποτελεσμάτων Αναζήτησης

Με την αναζήτηση ενός προϊόντος, παρουσιάζονται στο χρήστη τα κατάλληλα αποτελέσματα με δυνατότητες ταξινόμησης και φιλτραρίσματος τους με κριτήριο την απόσταση, την τιμή και τις αξιολογήσεις τους.



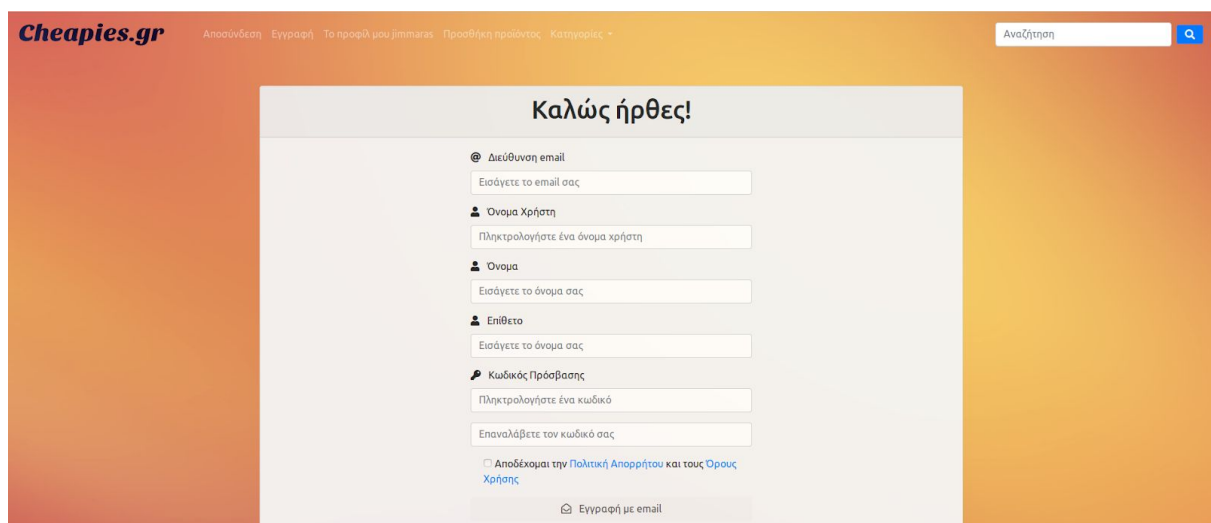
Σελίδα Προϊόντος

Η σελίδα του κάθε προϊόντος περιλαμβάνει όλες τις πληροφορίες σχετικά με την περιγραφή την τιμή και την τοποθεσία του. Δίνεται επίσης η δυνατότητα πλοήγησης στο τμήμα ερωτήσεων και απαντήσεων που αφορούν το προϊόν, η προσθήκη ερώτησης ή απάντησης εφόσον ο χρήστης είναι εγγεγραμμένος και τέλος η καταχώρηση του προϊόντος στα αγαπημένα του χρήστη ώστε να έχει εύκολη πρόσβαση σε αυτό μελλοντικά.



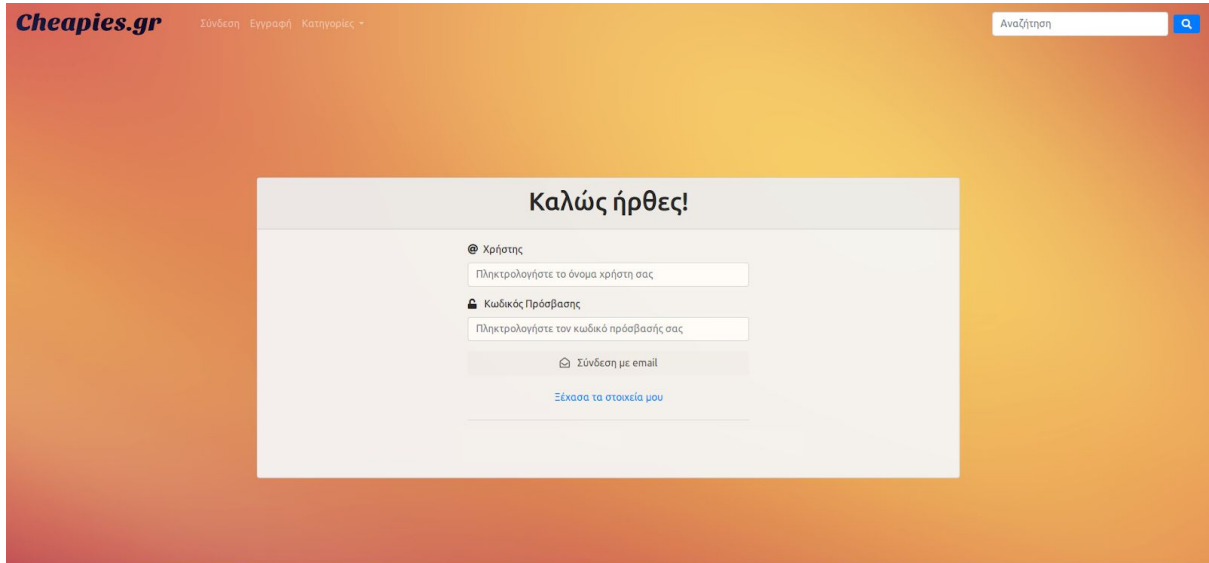
Σελίδα Εγγραφής

Εάν ένας ανώνυμος χρήστης επιθυμεί να εγγραφεί στο σύστημα, τότε κατευθύνεται στην αντίστοιχη σελίδα όπου συμπληρώνει τις απαραίτητες πληροφορίες.



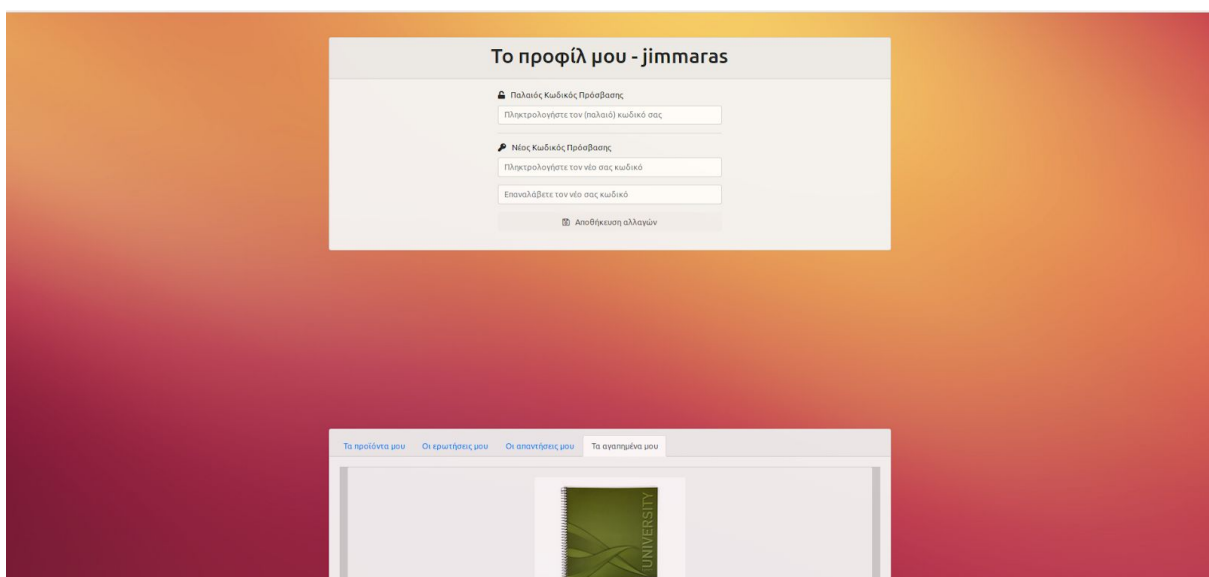
Σελίδα Σύνδεσης

Ένας εγγεγραμμένος χρήστης μπορεί να συνδεθεί στο σύστημα ταυτοποιώντας τα στοιχεία του. Στη συνέχεια τα στοιχεία του διατηρούνται κατά τη διάρκεια της συνεδρίας και του παρέχονται όλες οι επιπρόσθετες δυνατότητες.



Σελίδα Προφίλ

Ένας εγγεγραμμένος χρήστης διαθέτει μια σελίδα προφίλ όπου να μπορεί να επεξεργαστεί τη διεύθυνση ηλεκτρονικού ταχυδρομείου και τον κωδικό πρόσβασης. Επίσης στη σελίδα του χρήστη υπάρχει πρόσβαση στις ερωτήσεις, τις απαντήσεις του και τα προϊόντα που έχει στα αγαπημένα του.



Σελίδα Προσθήκης Προϊόντος

Κατά τη διαδικασία προσθήκης προϊόντος (από έναν εγγεγραμμένο χρήστη) ο χρήστης καλείται να εισάγει τις πληροφορίες του προϊόντος του καθώς και μία φωτογραφία του.

The screenshot shows the 'Καταχώρηση Προϊόντος' (Product Registration) form on the Cheapies.gr website. The form is titled 'Καταχώρηση Προϊόντος' and contains the following fields and options:

- Περιγραφή Προϊόντος:** A text area for the product description.
- Τιμή:** A text input field for the price.
- Τοποθεσία:** A dropdown menu with options 'Άλλα' and 'Νέο κατάστημα'.
- Κατηγορία:** A dropdown menu with options 'Είδος Προσωπικής Περιποίησης' and 'Άλλα'.
- Φωτογραφία:** A section for uploading a photo, with a 'Browse...' button and a 'No file selected.' message.
- Υποβολή Καταχώρησης:** A button to submit the registration.

Σελίδες Διαχείρισης

Οι διαχειριστές (administrators) της εφαρμογής έχουν το δικαίωμα να:

1. Κλειδώνουν χρήστες στους οποίους έχει γίνει αναφορά επανειλημμένα
2. Αναθέτουν και να ανακαλούν ρόλους
3. Να διαχειρίζονται τα προϊόντα και τις κατηγορίες

Στο project μας υπάρχει ξεχωριστή σελίδα διαχείρισης για τους διαχειριστές. Προσπέλαση στη σελίδα διαπίστευσης ενός διαχειριστή γίνεται μέσω του url: https://{server_name}/admin

Η δημιουργία ενός διαχειριστή γίνεται μόνο από το terminal του host μηχανήματος του site, στο οποίο συνδεόμαστε με ssh.

Ένας διαχειριστής δημιουργείται με χρήση της εντολής: ``python manage.py createsuperuser --username {username}``.

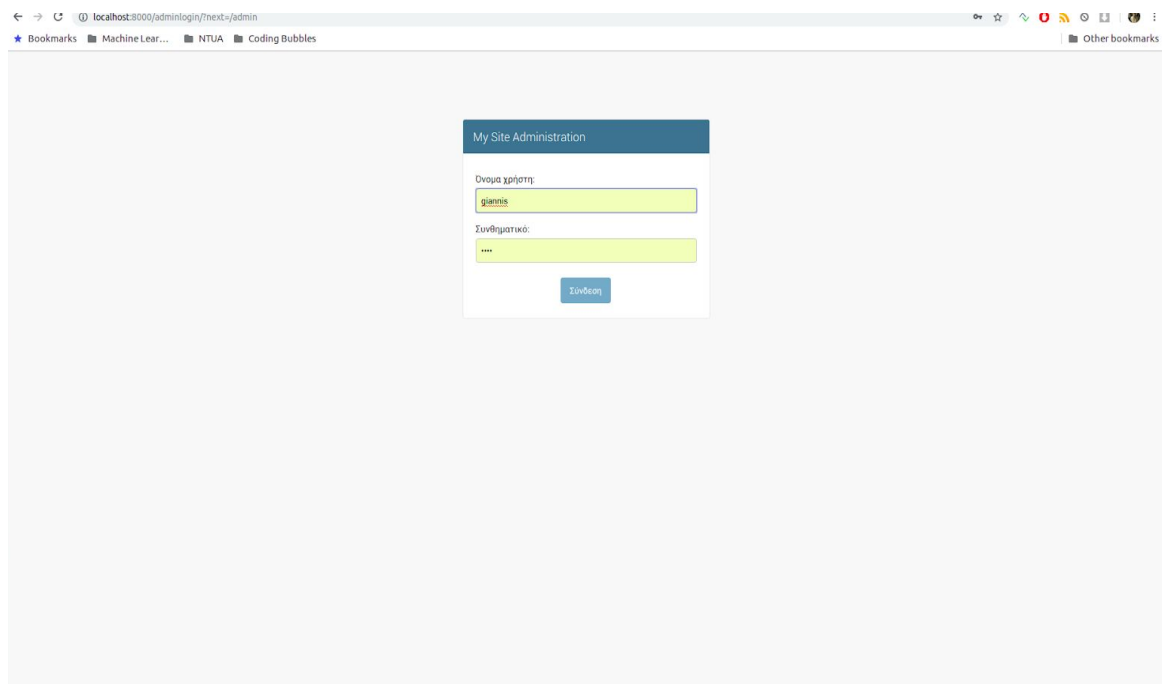
Είναι προφανές ότι δικαίωμα δημιουργίας νέων διαχειριστών έχουν μόνο όσοι έχουν ssh πρόσβαση στο host μηχανήμα, το οποίο σημαίνει πρακτικά η ομάδα ανάπτυξης. Αυτό είναι

θεμιτό, καθώς διαχειριστές προστίθενται σπάνια και πρέπει να υπάρχει έλεγχος για την ταυτότητα τους κατά τη δημιουργία.

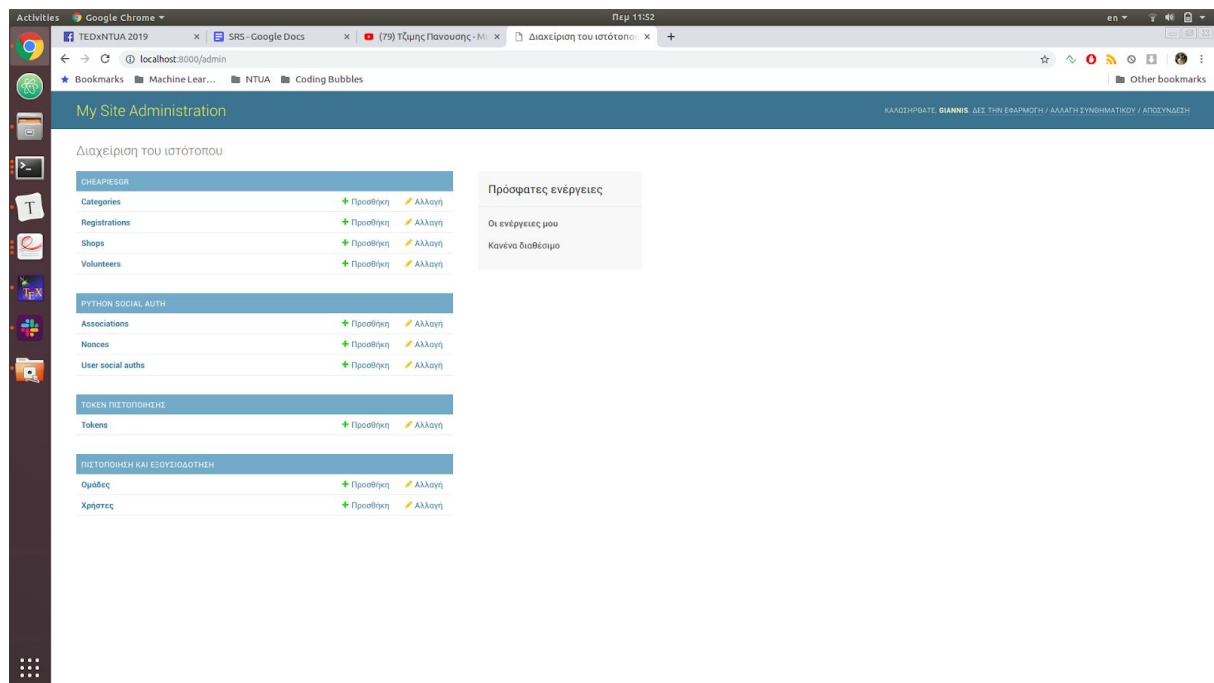
Το site για τη διαχείριση του περιεχομένου αποτελεί μια επέκταση του admin system που είναι integrated στο Django. Αυτό σημαίνει, ότι υπάρχει άμεσα η δυνατότητα τροποποίησης οποιουδήποτε πίνακα από το διαχειριστή της βάσης δεδομένων, μέσα από μια χρηστική διεπαφή (GUI).

Παρακάτω, παραθέτουμε ορισμένα στιγμιότυπα από πιθανές ενέργειες ενός διαχειριστή στο σύστημα διαχείρισης.

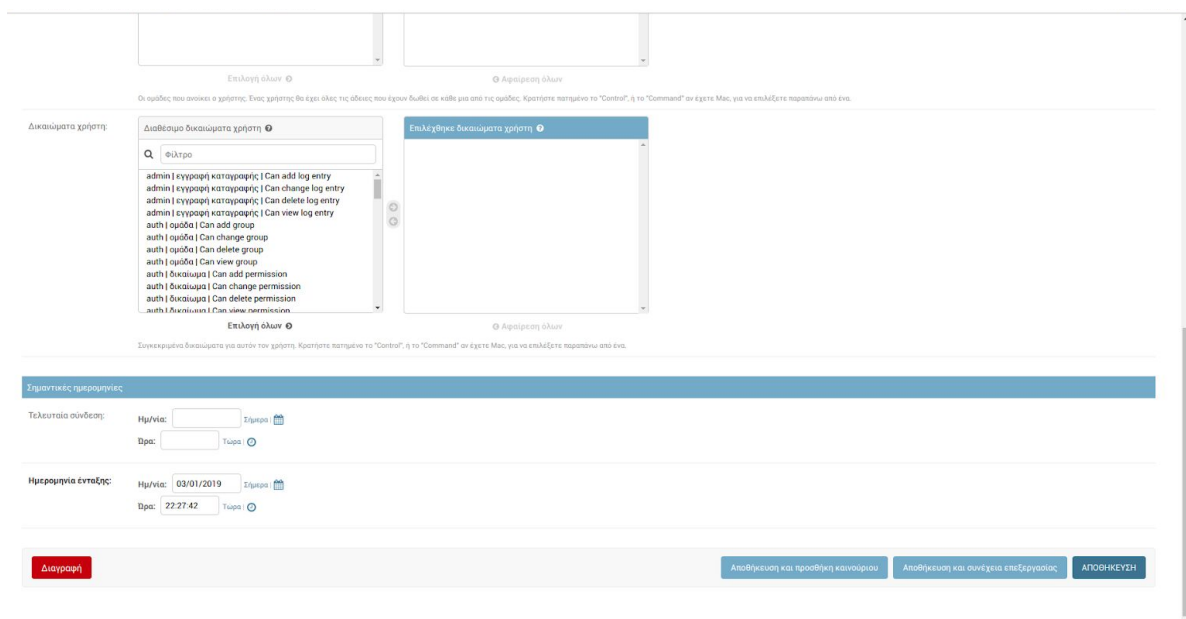
Σύνδεση διαχειριστή:



Κεντρική σελίδα διαχείρισης:



Σελίδες διαχείρισης χρηστών:



Σελίδα προσθήκης καταστήματος:

Activities Google Chrome Περ. 11:53

TEDxNTUA 2019 SRS - Google Docs (79) Λεωνίδας Μπαλά... Προσθήκη shop | My Site

My Site Administration ΚΑΛΩΣΗΡΘΑΤΕ, GIANNIS ΔΕΣ ΤΗΝ ΕΒΑΡΜΟΤΗ / ΑΔΑΜΗ ΣΥΝΘΗΜΑΤΟΥ / ΑΠΟΣΥΝΔΕΣΗ

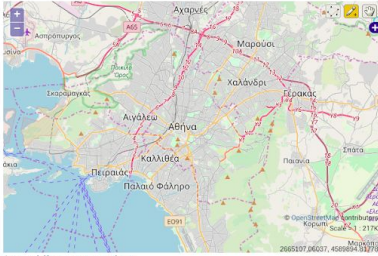
Αρχική Cheapsiesgr Shops Προσθήκη shop

Προσθήκη shop

Όνομα:

Address:

City:

Location: 

Διαγραφή όλων των αντικειμένων

Tags:

☐ Withdrawn

Αποθήκευση και προσθήκη κανόνισου Αποθήκευση και συνέχεια επεξεργασίας ΑΠΟΘΗΚΕΥΣΗ

Σελίδα αποσύνδεσης διαχειριστή:

My Site Administration

Αρχική

Έγινε αποσύνδεση

Ευχαριστούμε που διαθέσατε κάποιο ποστικό χρόνο στον ιστότοπο σήμερα.

Επανα σύνδεση

2.2 Λειτουργίες: περιπτώσεις χρήσης

2.2.1 Περίπτωση Χρήσης 1: Καταχώρηση Προϊόντος

2.2.1.1 Χρήστες (ρόλοι) που εμπλέκονται

Οι δύο ρόλοι που εμπλέκονται είναι ο Εθελοντής και ο Διαχειριστής. Συγκεκριμένα, θεωρούμε ότι ο διαχειριστής είναι κι αυτός εγγεγραμμένος χρήστης, που διαθέτει, ωστόσο, περισσότερες δυνατότητες. Έτσι, έχει, γενικότερα όλες τις δυνατότητες που έχουν οι εθελοντές.

2.2.1.2 Προϋποθέσεις εκτέλεσης

- Για να μπορέσει να πραγματοποιηθεί η καταχώρηση ενός προϊόντος, θα πρέπει ο εγγεγραμμένος χρήστης που επιθυμεί να την πραγματοποιήσει (εθελοντής ή διαχειριστής) να έχει ταυτοποιηθεί από την υπηρεσία.
- Θα πρέπει, παράλληλα, να παρασχεθεί μία φωτογραφία του προϊόντος.
- Τέλος, ο εθελοντής θα πρέπει να εισάγει την τοποθεσία του προϊόντος μέσω του καταστήματος στο οποίο το βρήκε. Μπορεί είτε να επιλέξει κάποιο από τα καταστήματα που υπάρχουν ήδη στη βάση, είτε να προσθέσει ένα νέο κατάστημα.

2.2.1.3 Περιβάλλον εκτέλεσης

- Web application
- REST API (POST product/{id})
- Σελίδα διαχείρισης (χωρίς τη χρήση φόρμας)

2.2.1.4 Δεδομένα εισόδου

| Πεδίο | Τύπος Δεδομένων |
|--------------------|-------------------------|
| Description | Char Field |
| Category | Foreign Key (Category) |
| DateOfRegistration | Date Field |
| Shop | Foreign Key (Shop) |
| Image url | Char Field |
| Withdrawn | BooleanField |
| Tags | Char Field List |
| Volunteer | Foreign Key (Volunteer) |
| Price | Decimal Field |

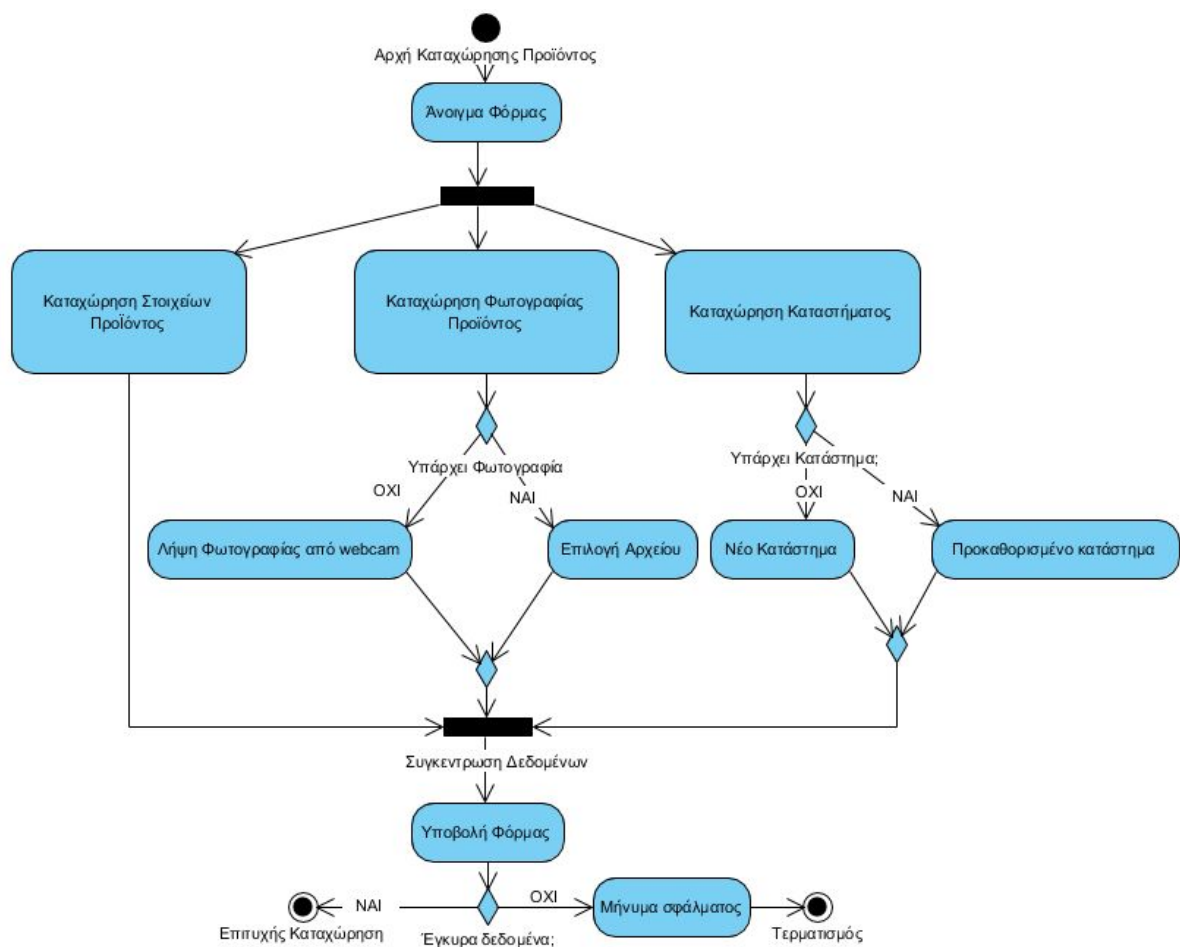
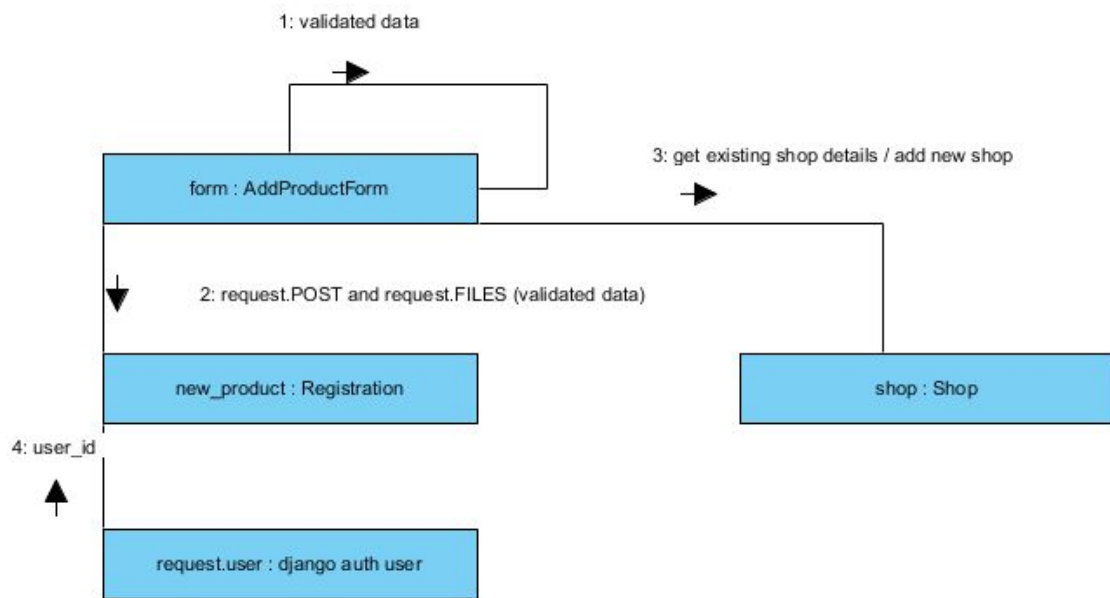
2.2.1.5 Παράμετροι

| Πεδίο | Τύπος Δεδομένων | Περιορισμοί |
|--------------------|-------------------------|------------------|
| Description | Char Field | <1000 Characters |
| Category | Foreign Key (Category) | NOT NULL |
| DateOfRegistration | Date Field | >= TODAY |
| Shop | Foreign Key (Shop) | NOT NULL |
| Image url | Char Field | <500 Characters |
| Withdrawn | BooleanField | |
| Tags | Char Field List | |
| Volunteer | Foreign Key (Volunteer) | NOT NULL |
| Price | Decimal Field | > 0.0 |

2.2.1.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

1. Ο χρήστης ταυτοποιείται, ώστε να επιβεβαιωθεί ότι είναι ένας εγγεγραμμένος χρήστης.
2. Ανοίγει η φόρμα που περιλαμβάνει τα προς συμπλήρωση πεδία που αφορούν πληροφορίες για το νέο προϊόν. Συγκεκριμένα, πρέπει να καταχωρηθεί η τιμή του, μία περιγραφή, η κατηγορία στην οποία ανήκει, καθώς και το κατάστημα στο οποίο το βρήκε ο χρήστης, το οποίο αν δεν υπάρχει ήδη στη βάση δεδομένων θα πρέπει να προσδιοριστεί γεωγραφικά από το χρήστη.
3. Στη συνέχεια, τα δεδομένα συλλέγονται και ελέγχονται ώστε να διασφαλιστεί η συνέπεια της βάσης δεδομένων μετά την εισαγωγή τους και αν είναι έγκυρα, η καταχώρηση πραγματοποιείται και η διαδικασία τερματίζεται. Αν δεν είναι έγκυρα, η διαδικασία τερματίζεται με κατάλληλο μήνυμα σφάλματος.

Στα ακόλουθα διαγράμματα, communication και activity, περιγράφεται σχηματικά η λειτουργία αυτή:



2.2.2 Περίπτωση Χρήσης 2: Αναζήτηση Προϊόντος

2.2.2.1 Χρήστες (ρόλοι) που εμπλέκονται

Στην λειτουργία αυτή έχει πρόσβαση οποιοσδήποτε χρήστης της εφαρμογής, εγγεγραμμένος ή μη.

2.2.2.2 Προϋποθέσεις εκτέλεσης

Αναζήτηση ενός προϊόντος (καταχώρησης) στο παρατηρητήριο τιμών.

2.2.2.3 Περιβάλλον εκτέλεσης

- Web application
- REST API (GET product/{id})
- Σελίδα διαχείρισης

2.2.2.4 Δεδομένα εισόδου

Φίλτρα Αναζήτησης:

| Πεδίο | Τύπος Δεδομένων |
|-------------------------|-----------------|
| Αναγνωριστικό Προϊόντος | Integer Field |
| Αξιολόγηση Προϊόντος | Integer Field |
| Απόσταση | Decimal Field |
| Κατηγορία Προϊόντος | Category |
| Περιγραφή Προϊόντος | Char Field |
| Γεωγραφικό Μήκος/Πλάτος | Decimal Field |

Επιλογές ταξινόμησης:

1. Με βάση την ημερομηνία εγγραφής.
2. Με βάση την τιμή.
3. Με βάση τη βαθμολογία της εγγραφής.
4. Με βάση την απόσταση από τη γεωγραφική θέση του χρήστη.

2.2.2.5 Παράμετροι

Φίλτρα Αναζήτησης

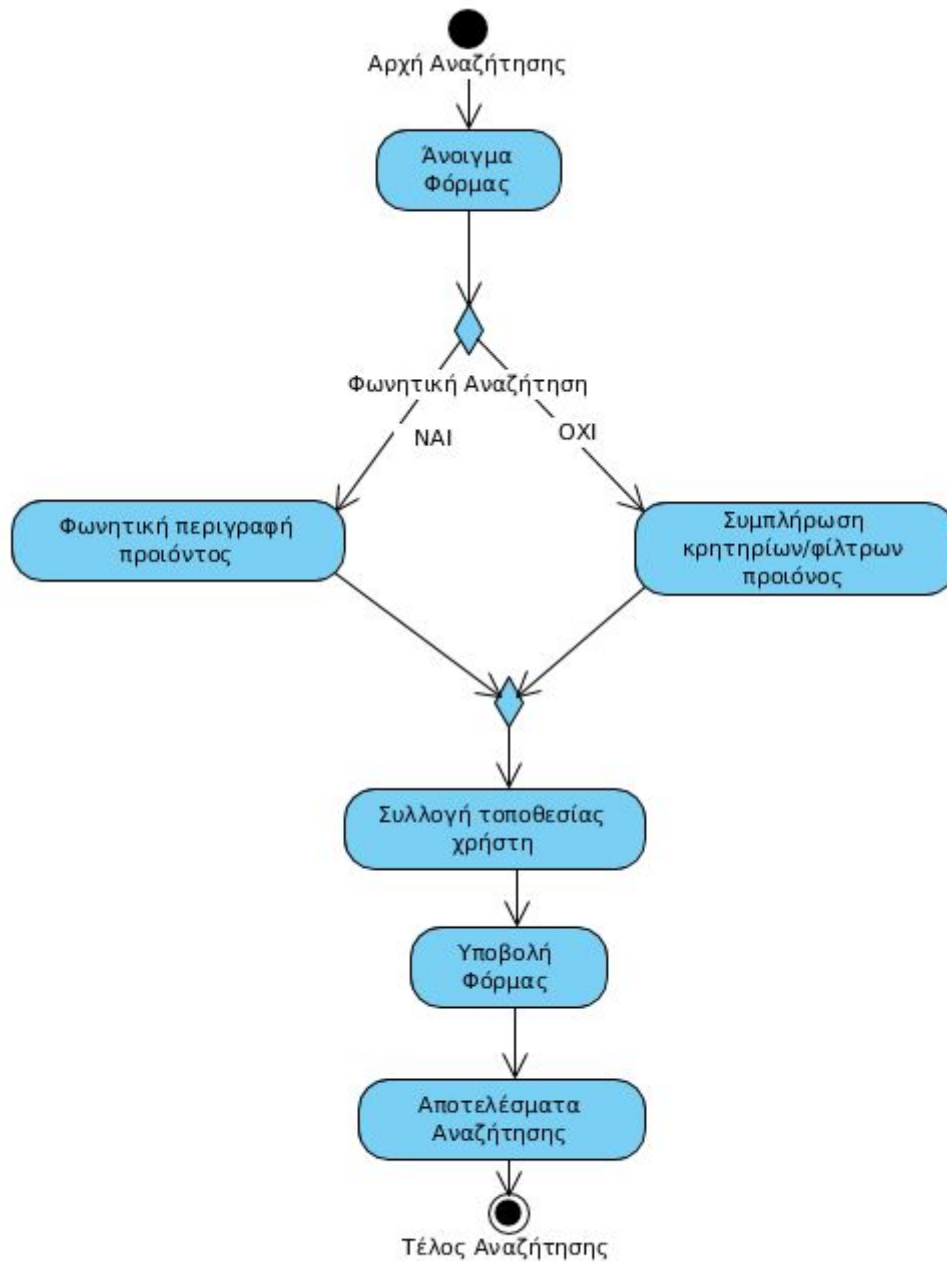
| Πεδίο | Τύπος Δεδομένων | Περιορισμοί |
|-------------------------|-----------------|-----------------------------------|
| Αναγνωριστικό Προϊόντος | Integer Field | ≥ 0 |
| Αξιολόγηση Προϊόντος | Integer Field | Μεταξύ 1 ως 5 |
| Απόσταση | Decimal Field | ≥ 0 |
| Κατηγορία Προϊόντος | Category | Έγκυρη Κατηγορία |
| Περιγραφή Προϊόντος | Char Field | ≤ 10000 characters |
| Γεωγραφικό Μήκος/Πλάτος | Decimal Field | Έγκυρες Γεωγραφικές Συντεταγμένες |

2.2.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

1. Αρχικά, ο χρήστης εισάγει τιμές για τα φίλτρα αναζήτησης και επιλέγει το επιθυμητό κριτήριο ταξινόμησης. Εναλλακτικά, μπορεί να χρησιμοποιήσει την φωνητική αναζήτηση.
2. Συλλέγεται η τοποθεσία του χρήστη.
3. Η φόρμα υποβάλλεται και επιστρέφουν ταξινομημένα με βάση το κριτήριο που δόθηκε τα προϊόντα που ικανοποιούν τα δεδομένα αναζήτησης. Παράλληλα με αυτά, επιστρέφει και η απόσταση του χρήστη από το κατάστημα στο οποίο βρίσκεται η εγγραφή αυτή.

Στα ακόλουθα διαγράμματα communication και activity, παρατίθεται μία γραφική αναπαράσταση της διαδικασίας:





2.2.3 Περίπτωση Χρήσης 3: Διαχείριση Πλατφόρμας

2.2.3.1 Χρήστες (ρόλοι) που εμπλέκονται

- Διαχειριστής

2.2.3.2 Προϋποθέσεις εκτέλεσης

Διαχειριστικές Λειτουργίες Πλατφόρμας

- Προσθήκη/Αφαίρεση/Επεξεργασία Προϊόντων
- Προσθήκη/Διαγραφή Χρηστών

2.2.3.3 Περιβάλλον εκτέλεσης

- Web Application (Admin Page)

2.2.3.4 Δεδομένα εισόδου

- Μοντέλο προϊόντων όπως περιγράφεται και στις προηγούμενες περιπτώσεις χρήσης.
- Μοντέλο χρήστη

| Πεδίο | Τύπος Δεδομένων |
|----------|-----------------|
| Username | Char Field |
| Email | Char Field |

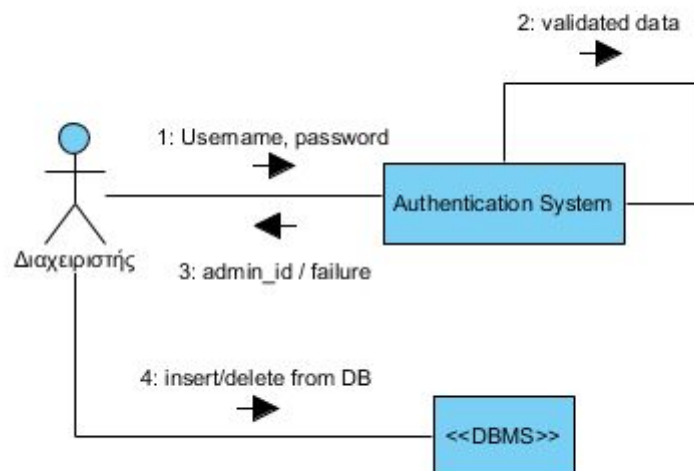
2.2.3.5 Παράμετροι

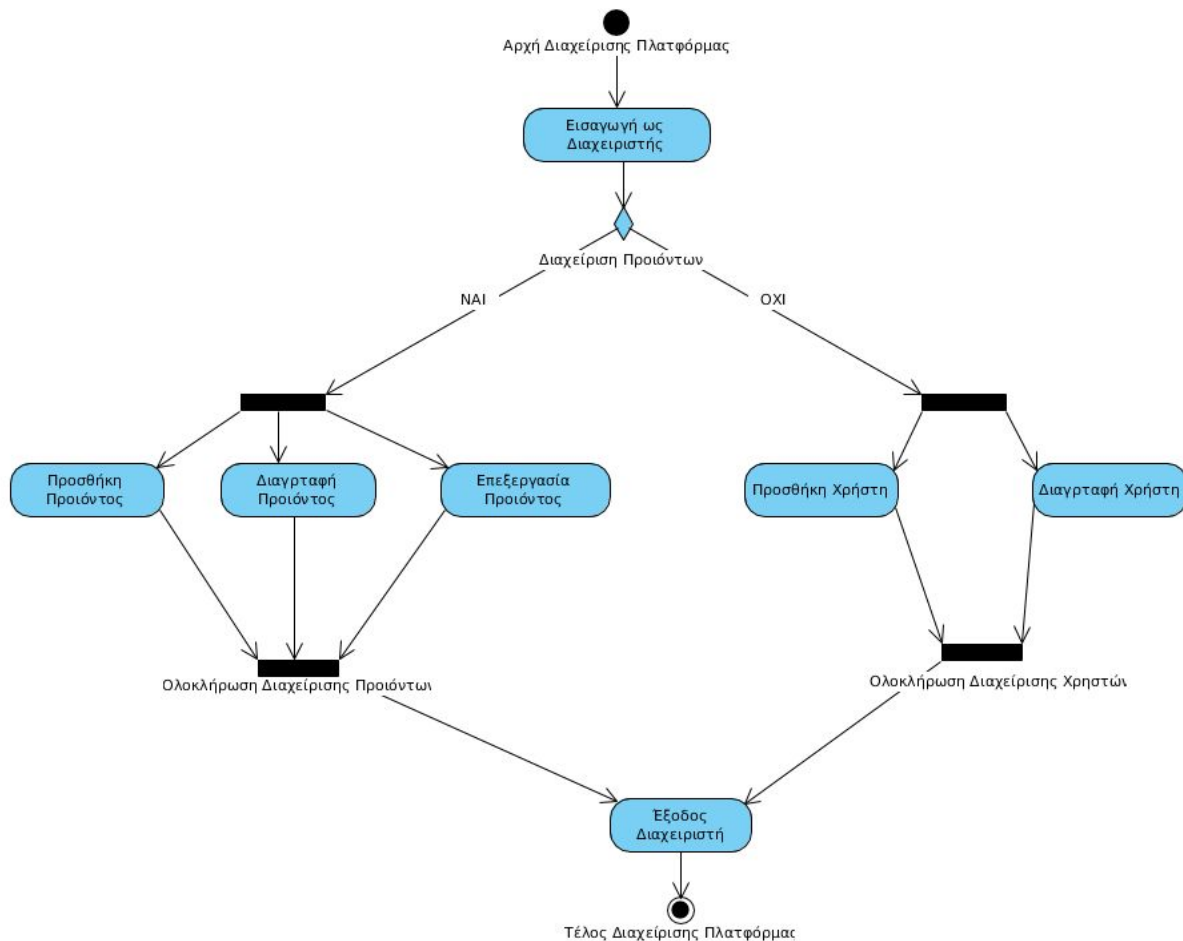
- Για τα προϊόντα ισχύουν οι ίδιες παράμετροι με την 1η περίπτωση χρήσης
- Για τους χρήστες

| Πεδίο | Τύπος Δεδομένων | Περιορισμοί |
|----------|-----------------|--|
| Username | Char Field | ([A-Z]*[a-z]*[0-9]*)+ |
| Email | Char Field | r"^[a-zA-Z0-9_+-.]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-]+\$)" |

2.2.3.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

1. Ταυτοποίηση Διαχειριστή.
2. Επεξεργασία βάσης δεδομένων από τον διαχειριστή.
3. Έξοδος.





2.3 Απαιτήσεις επιδόσεων

Το λογισμικό μας θα είναι συνεχώς διαθέσιμο σε αυτούς που επιθυμούν να το χρησιμοποιήσουν. Αναμένουμε σε ώρες αιχμής φορτίο 500 ταυτόχρονα συνδεδεμένων χρηστών τους οποίους και μπορούμε να διαχειριστούμε. Τέλος, το REST API μας μπορεί να καλύψει 100 requests το δευτερόλεπτο προς όλα τα resources.

2.4 Απαιτήσεις οργάνωσης δεδομένων

2.4.1 Περιγραφή Δεδομένων

Τα δεδομένα που διαχειρίζεται το λογισμικό αποθηκεύονται μέσω κατάλληλων μοντέλων στη βάση δεδομένων. Η βάση δεδομένων που χρησιμοποιείται είναι η MySQL. Αναλυτική περιγραφή των μοντέλων γίνεται στα class και ER diagrams. Τα μοντέλα περιέχουν επίσης και γεωγραφικά δεδομένα τα οποία διαχειρίζονται από σύστημα GIS μέσω της βάσης

δεδομένων και του web framework (Geodjango). Τέλος, οι καταχωρήσεις περιέχουν και στατικά αρχεία, όπως εικόνες, που αποθηκεύονται στο media directory της εφαρμογής.

Η καταχώρηση των δεδομένων μπορεί να γίνει μέσω φορμών από τη web εφαρμογή ή μέσω post request στο REST API. Η αποθήκευση των δεδομένων γίνεται στο σκληρό δίσκο του web server.

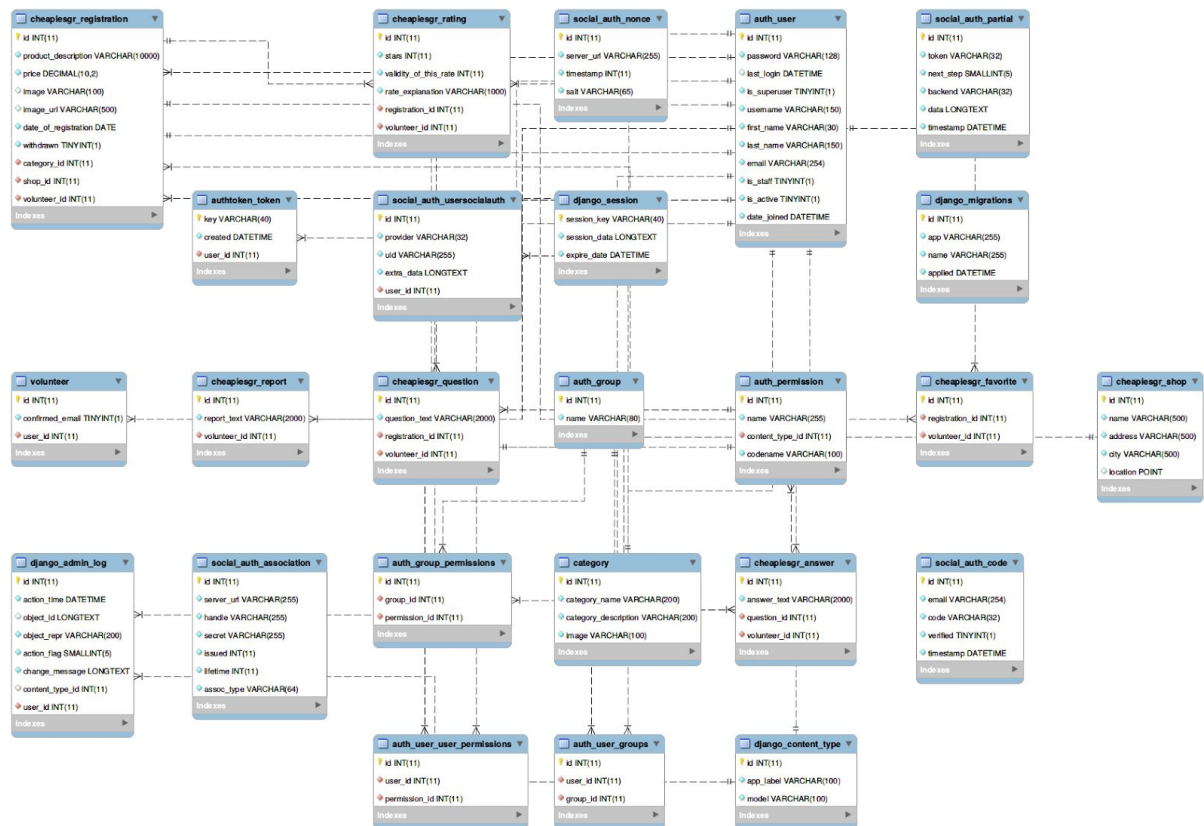
2.4.2 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

Στον ακόλουθο πίνακα παρουσιάζονται τα διάφορα μοντέλα δεδομένων που χρησιμοποιούμε καθώς και η προσβασιμότητά τους από την κάθε κατηγορία χρηστών. Όπου R=Read (δυνατότητα προσπέλασης) και όπου W=Write (δυνατότητα τροποποίησης).

| Μοντέλο | Διαχειριστής | Χρήστης | Ανώνυμος Χρήστης |
|--------------|--------------|---------|------------------|
| Registration | RW | RW | R |
| Answers | RW | RW | R |
| Questions | RW | RW | R |
| Favorite | RW | RW | R |
| Category | RW | RW | R |
| Shop | RW | RW | R |
| Rating | RW | RW | R |
| Report | RW | RW | R |
| Volunteer | RW | RW | R |

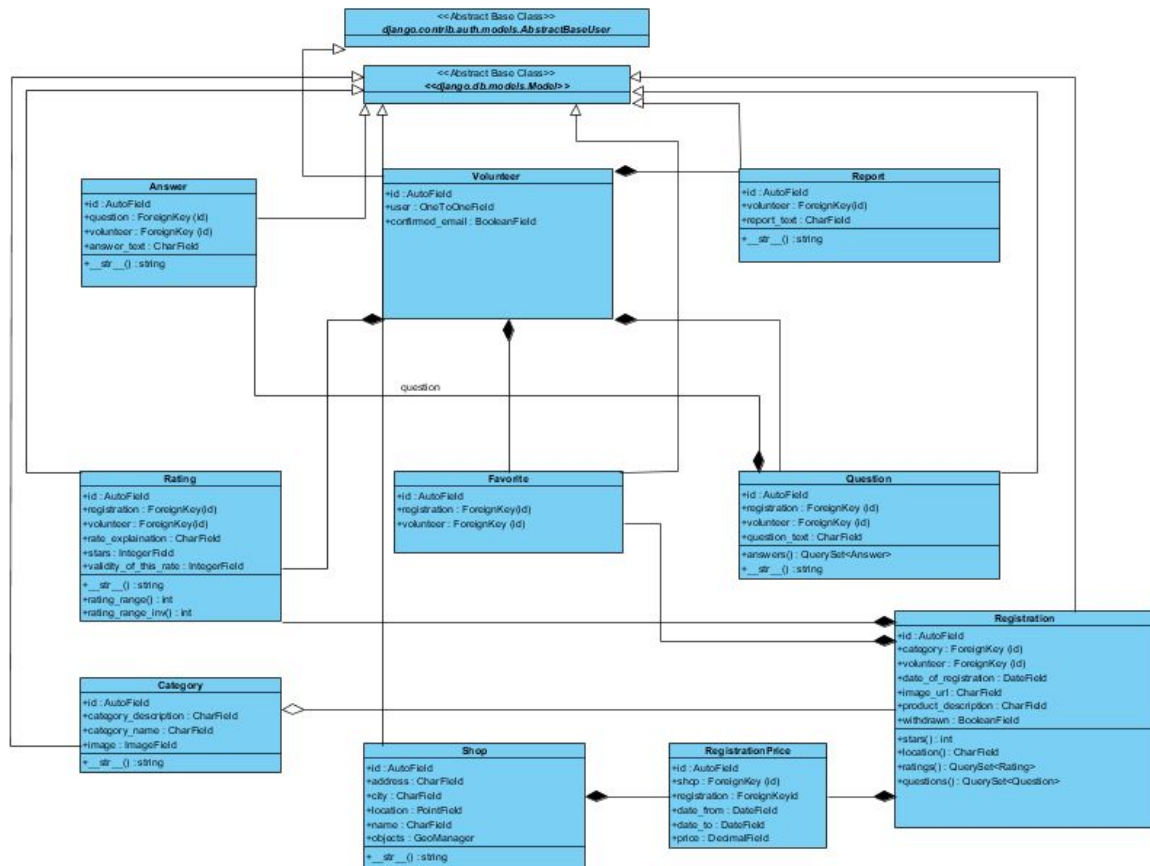
2.4.3 Μοντέλο δεδομένων (UML Class Diagram/ER Diagram)

Διάγραμμα ER



*)Το διάγραμμα έχει παραχθεί από το MySQL Workbench.

Διάγραμμα Κλάσεων (UML Class Diagram)



2.4.4 Προδιαγραφές ακεραιότητας δεδομένων

Physical Integrity

Τα δεδομένα αποθηκεύονται στο virtual machine που είναι υπεύθυνο για να σερβιρεί την εφαρμογή.

Integrity Constraints

Οι περιορισμοί ακεραιότητας περιγράφονται ενδελεχώς στο διάγραμμα ER.

2.4.5 Προδιαγραφές διατήρησης δεδομένων

Για τα δεδομένα που εισάγονται στο σύστημά μας λαμβάνουμε τις εξής αποφάσεις:

1. Τα προϊόντα του ΠΤ θα πρέπει να ανανεώνονται σε τακτική βάση. Ο μέγιστος χρόνος που ένα προϊόν μπορεί να θεωρείται ενεργό, τίθεται στους 24 μήνες.
2. Οι λογαριασμοί των χρηστών που δεν έχουν χρησιμοποιηθεί πάνω από ένα χρόνο διαγράφονται μετά από σχετική ενημέρωση των χρηστών
3. Expired token ώστε σε περίπτωση υποκλοπής σε σύντομο χρονικό διάστημα να απαιτείται ανανέωση, περιορίζοντας τον χρόνο που θα έχει στη διάθεση του ένας κακόβουλος χρήστης.
4. Refresh token που απαιτεί την πιστοποίηση της εγκυρότητας του πελάτη για μεγαλύτερη ασφάλεια.
5. Αποθήκευση στον Session Manager που παρέχει η εφαρμογή των Access και Refresh tokens στο αντίστοιχο session id. Όταν απαιτείται μία σελίδα από τον χρήστη και αυτή χρειάζεται πρόσβαση στους πόρους του συστήματος τότε χρησιμοποιούμε το Access token και αν αυτό έχει λήξει τότε χρησιμοποιούμε το Refresh token για να αποκτήσουμε καινούριο.
6. Για την διάρκεια ζωής των παραπάνω, στις περιπτώσεις που απαιτείται η ασφάλεια των προσωπικών δεδομένων θα πρέπει να υπάρχει μικρός χρόνος μέχρι την λήξη του access token. Αντίθετα όταν τα δεδομένα που μεταφέρονται δεν είναι ευαίσθητα δίνεται μεγαλύτερο βάρος στην διευκόλυνση του χρήστη επιτρέποντας μεγαλύτερο χρόνο ζωής του token.
7. Ο κάθε χρήστης παραμένει συνδεδεμένος μετά την πρώτη σύνδεση και αν δεν έχει αλληλεπιδράσει με το σύστημα για δύο εβδομάδες τότε έχουμε session expiration και το σύστημα τον αποσυνδέει αυτόματα.

2.5 Περιορισμοί σχεδίασης

Στη συνέχεια αναφέρονται τα **τεχνολογικά πρότυπα** που χρησιμοποιήθηκαν κατά την ανάπτυξη της εφαρμογής:

- Back-end
 - Python v3.5
 - Django v2.1
 - MySQL v14.14
 - Django REST Framework
 - OpenStreetMaps API

- Front-end
 - Bootstrap v4.1.3
 - Tachyons v4.1
 - wow.js
 - Leaflet.js
 - jQuery
 - slick.js
- Deployment
 - Gunicorn
 - nginx
 - Docker
- Continuous Integration & Unit Testing
 - Travis CI
 - Selenium (Browser-based testing)
 - Locust.io (RESTful API Testing)
- Build Automation
 - GNU Make

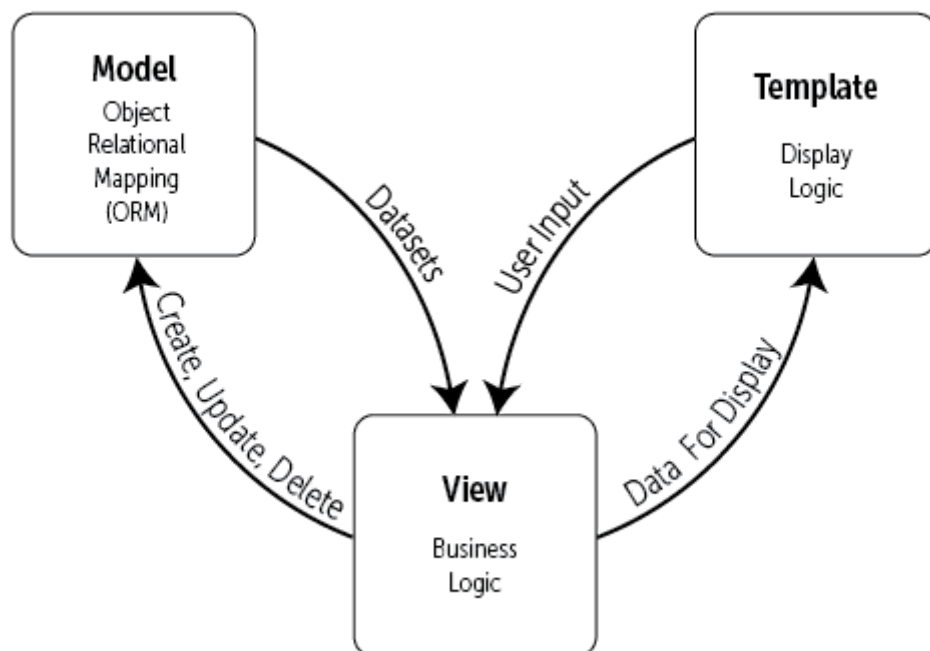
Τα **πρότυπα συγγραφής κώδικα** που χρησιμοποιήθηκαν από όλη την ομάδα κατά την ανάπτυξη της εφαρμογής είναι:

- Back-end: [PEP8](#) και [Django Coding Style](#)
- Front-end: [Bootstrap CSS Coding Standards](#)

Τα **πρότυπα συνεισφοράς** με τα οποία επιτεύχθηκε ο συντονισμός και η συνεργατικότητα στα πλαίσια της ομάδας είναι:

- Development με branches (master/develop workflow)
- Kanban Scheduling με project boards
- GitHub Issues

Τέλος, το **πρότυπο σχεδίασης** γύρω από το οποίο έγινε η ανάπτυξη της εφαρμογής ήταν το MVT (Model-View-Template).



2.6 Λοιπές απαιτήσεις

2.6.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Στον ακόλουθο πίνακα αναγράφονται τα δικαιώματα που παραχωρούνται στις τρεις κατηγορίες χρηστών κατά τη χρήση της εφαρμογής:

| Σενάριο | Χρήστης | Αναγνώστης | Διαχειριστής |
|--|---------|------------|--------------|
| Χρήση front-end και πλοήγηση στις σελίδες αναζήτησης προϊόντων | X | X | X |
| Ανάκτηση και ενημέρωση προϊόντων | X | | X |
| Διαχείριση Προφίλ | X | | X |
| Διαχείριση Λογαριασμών | | | X |
| Χρήση RESTFul API (GET/POST/DELETE) | X | | X |

2.6.2 Απαιτήσεις ασφάλειας

Ιδιαίτερη προσοχή έχει δοθεί στον τομέα της ασφάλειας των προσωπικών στοιχείων κάθε χρήστη. Σε αντίθεση με τις υπόλοιπες οντότητες δεδομένων, για τα username και τους κωδικούς χρησιμοποιείται ειδική κλάση αντικειμένων του django framework που παρέχει **hashing** για τους κωδικούς πρόσβασης των χρηστών ώστε να μην είναι εύκολα προσβάσιμοι μέσα στη βάση.

Επίσης, έπειτα από την ταυτοποίησή του ο χρήστης συνεχίζει να παραμένει συνδεδεμένος στο σύστημα με τη χρήση session cookies και δεν ζητείται ποτέ ξανά η επιβεβαίωση των στοιχείων του κατά τη διάρκεια της συνεδρίας του.

Παρόμοιοι περιορισμοί έχουν τεθεί και σε ορισμένες λειτουργίες του RESTful API οι οποίες απαιτούν user authentication για να εκτελεστούν.

2.6.3 Απαιτήσεις συντήρησης

- Το λογισμικό πρόκειται να φιλοξενηθεί σε διακομιστή (server) στο okeanos.
- Τα SSL Certificates πρέπει να ανανεώνονται τακτικά. Η ύπαρξη τους διαβεβαιώνει ότι οι ευαίσθητες πληροφορίες (username, password, διευθύνσεις, μέθοδοι και στοιχεία πληρωμών) που ανταλλάσσονται μεταξύ του χρήστη και του συστήματος μεταφέρονται μέσω ασφαλούς δικτύου, κρυπτογραφημένα και προστατευμένα από κακόβουλες επιθέσεις. Απαιτείται κόστος συντήρησης και ανανέωσης καθώς και αύξηση της πολυπλοκότητας για την υλοποίησή τους. Με χρήση ενός SSL προστατεύονται όλα οι επιμέρους τομείς της εφαρμογής αποφεύγοντας έτσι την εγκατάσταση ξεχωριστού SSL σε κάθε τομέα.

3. Παράρτημα

3.1 Παραδοχές και εξαρτήσεις

Δεχόμαστε ότι οι χρήστες δεν ενδιαφέρονται να βρίσκουν μονάχα τις πιο πρόσφατες και εγκυρότερες καταχωρήσεις. Δεδομένου ότι η σελίδα βασίζεται στο crowdsourcing, μπορούν κάποια προϊόντα να καταχωρούνται πολλές φορές ακόμη και για το ίδιο κατάστημα. Ωστόσο, κάποιες εγγραφές, με τις υψηλότερες βαθμολογίες, θα είναι και αυτές που τελικά οι χρήστες θα εμπιστεύονται. Οι εγγραφές δεν αντιστοιχούν σε προϊόντα, αλλά σε παρατηρήσεις των καταναλωτών για τις τιμές των προϊόντων και έχουν συμβουλευτικό χαρακτήρα.

3.2 Ακρωνύμια και συντομογραφίες

SRS: Software Requirements Specification

ΠΤ: Πρατήριο Τιμών

ER: Entity-Relation

SQL: Structured Query Language

UML: Universal Modelling Language

API: Application Programming Interface

SSL: Secure Sockets Layer

3.3 Υποστηρικτικά έγγραφα

1. ΠΡΟΤΥΠΟ ISO/IEC/IEEE 29148:2011
2. Πηγαίος Κώδικας Λογισμικού cheapies.gr