

Έγγραφο Απαιτήσεων Λογισμικού *SRS*

Javengers

24 Φεβρουαρίου 2019

Περιεχόμενα

1	Εισαγωγή	2
1.1	Σκοπός του Λογισμικού	2
1.2	Επισκόπηση Λογισμικού	2
1.3	Διεπαφές με Εξωτερικά Συστήματα	5
1.4	Διεπαφές με τον Χρήστη	6
2	Προδιαγραφές απαιτήσεων Λογισμικού	7
2.1	Λειτουργικές Απαιτήσεις	7
2.2	Μη Λειτουργικές Απαιτήσεις	8
3	Λειτουργικές Περιπτώσεις Χρήσης	9
3.1	Εγγραφή ενός Χρήστη	9
3.1.1	Ρόλοι που Εμπλέκονται	9
3.1.2	Προϋποθέσεις Εκτέλεσης	9
3.1.3	Περιβάλλον Εκτέλεσης	9
3.1.4	Δεδομένα Εισόδου	9
3.1.5	Παράμετροι	9
3.1.6	Αλληλουχία Ενεργειών	10
3.1.7	Δεδομένα Εξόδου	11
3.2	Αναζήτηση Προϊόντος	11
3.2.1	Ρόλοι που Εμπλέκονται	11
3.2.2	Προϋποθέσεις Εκτέλεσης	11
3.2.3	Περιβάλλον Εκτέλεσης	11
3.2.4	Δεδομένα Εισόδου	11
3.2.5	Παράμετροι	11
3.2.6	Αλληλουχία Ενεργειών	11
3.2.7	Δεδομένα Εξόδου	12
3.3	Προσθήκη Νέα Καταχώρησης	12
3.3.1	Ρόλοι που Εμπλέκονται	12

3.3.2	Προϋποθέσεις Εκτέλεσης	12
3.3.3	Περιβάλλον Εκτέλεσης	13
3.3.4	Δεδομένα Εισόδου	13
3.3.5	Παράμετροι	13
3.3.6	Αλληλουχία Ενεργειών	14
3.3.7	Δεδομένα Εξόδου	15
3.4	Απαιτήσεις Επιδόσεων	15
3.5	Απαιτήσεις Οργάνωσης Δεδομένων	15
3.5.1	Τεχνική Περιγραφή των Δεδομένων	15
3.5.2	Απαιτήσεις και Περιορισμοί Πρόσβασης στα Δεδομένα	15
3.5.3	Μοντέλο Δεδομένων	16
3.5.4	Προδιαγραφές Ακεραιότητας Δεδομένων	17
3.5.5	Προδιαγραφές Διατήρησης Δεδομένων	17
3.6	Περιορισμοί Σχεδιάσης	17
3.7	Λοιπές Απαιτήσεις	18
3.7.1	Απαιτήσεις Διαθεσιμότητας	18
3.7.2	Απαιτήσεις Ασφάλειας	18
3.7.3	Απαιτήσεις Συντήρησης	18

1 Εισαγωγή

1.1 Σκοπός του Λογισμικού

Σκοπός του παρόντος λογισμικού αποτελεί η πλήρης κατασκευή ενός ηλεκτρονικού παρατηρητήριου τιμών που θα διευκολύνει και θα εμπλουτίζει την εμπειρία των καταναλωτών προσφέροντας τη δυνατότητα απλής και αποτελεσματικής αναζήτησης των τιμών που καταγράφονται για τα επιμέρους προϊόντα στα διάφορα μαγαζιά. Αξίζει να αναφερθεί, ότι η ηλεκτρονική μας πλατφόρμα θα βασίζεται στη μέθοδο του *crowdsourcing*, όπου δηλαδή ένα σύνολο από εγγεγραμμένους εθελοντές θα αποτελεί ενεργητικό κομμάτι της εφαρμογής και των προσφερόμενων υπηρεσιών πραγματοποιώντας καταχωρήσεις για προϊόντα και καταστήματα.

1.2 Επισκόπηση Λογισμικού

Η ιστοσελίδα μας υποστηρίζει τις ακόλουθες τρεις βασικές ομάδες χρηστών, που ενθυλακώνουν και τις επιμέρους λειτουργίες της εφαρμογής:

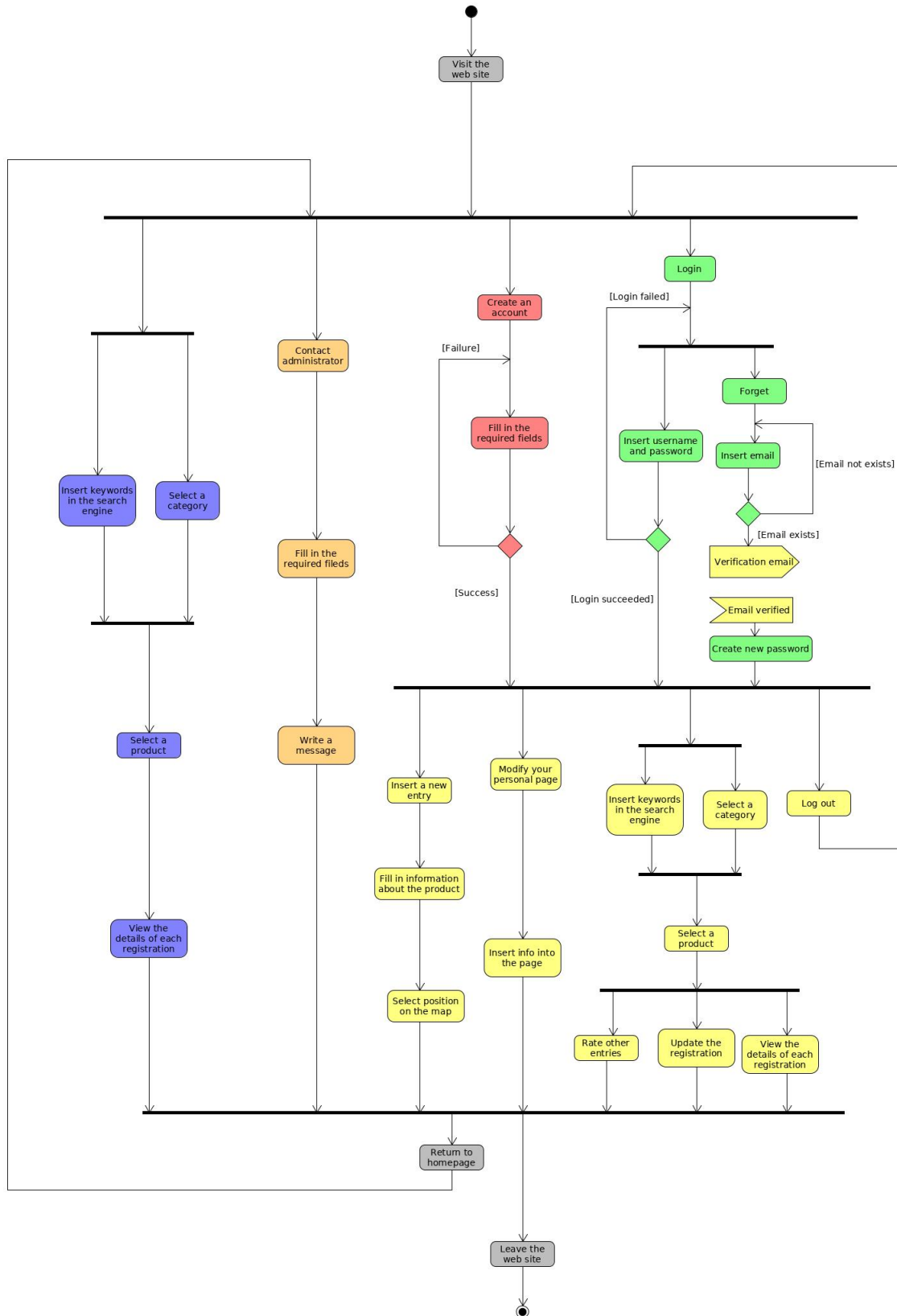
- Ο απλός χρήστης ή παρατηρητής που δεν έχει πραγματοποιήσει εγγραφή και του δίνεται η δυνατότητα να πραγματοποιεί αναζητήσεις στη βάση δεδομένων της εφαρμογής σχετικά με προϊόντα και τις αντίστοιχες καταχωρήσεις τους. Στο πλαίσιο αυτό, του δίνεται επίσης η δυνατότητα να φιλτράρει τα επιμέρους αποτελέσματα που προκύπτουν με βάση τις τιμές των καταχωρήσεων και την τοποθεσία των αντίστοιχων καταστημάτων.

- Ο χρήστης εθελοντής που είναι σε θέση να εισάγει, να ανανεώνει και να διαγράφει δεδομένα σχετικά με τιμές των προϊόντων, των καταστημάτων και των καταχωρήσεων, συμμετέχοντας έτσι ενεργητικά στη σελίδα μας. Επιπλέον, του δίνεται η δυνατότητα να δημιουργήσει και να εμπλουτίσει ένα προφίλ που θα είναι διαθέσιμο και στους άλλους χρήστες, ενώ όπως γίνεται αντιληπτό είναι σε θέση να εκτελέσει τις ενέργειες ενός απλού μη εγγεγραμμένου χρήστη.
- Ο διαχειριστής της σελίδας, που έχει τον ρόλο του επιβλέποντα, επικοινωνεί και επεξεργάζεται τα αιτήματα των χρηστών ώστε να συντηρείται και να βελτιώνεται διαρκώς η σελίδα, ενώ είναι σε θέση να ελέγξει και να επεξεργαστεί τις καταχωρήσεις της βάσης δεδομένων με μία απλή διεπαφή.

Με βάση λοιπόν τα παραπάνω, μπορούμε να συνοψίσουμε επιγραμματικά τις βασικές λειτουργικότητες που υποστηρίζονται που στη συνέχεια θα περιγράψουμε πιο αναλυτικά:

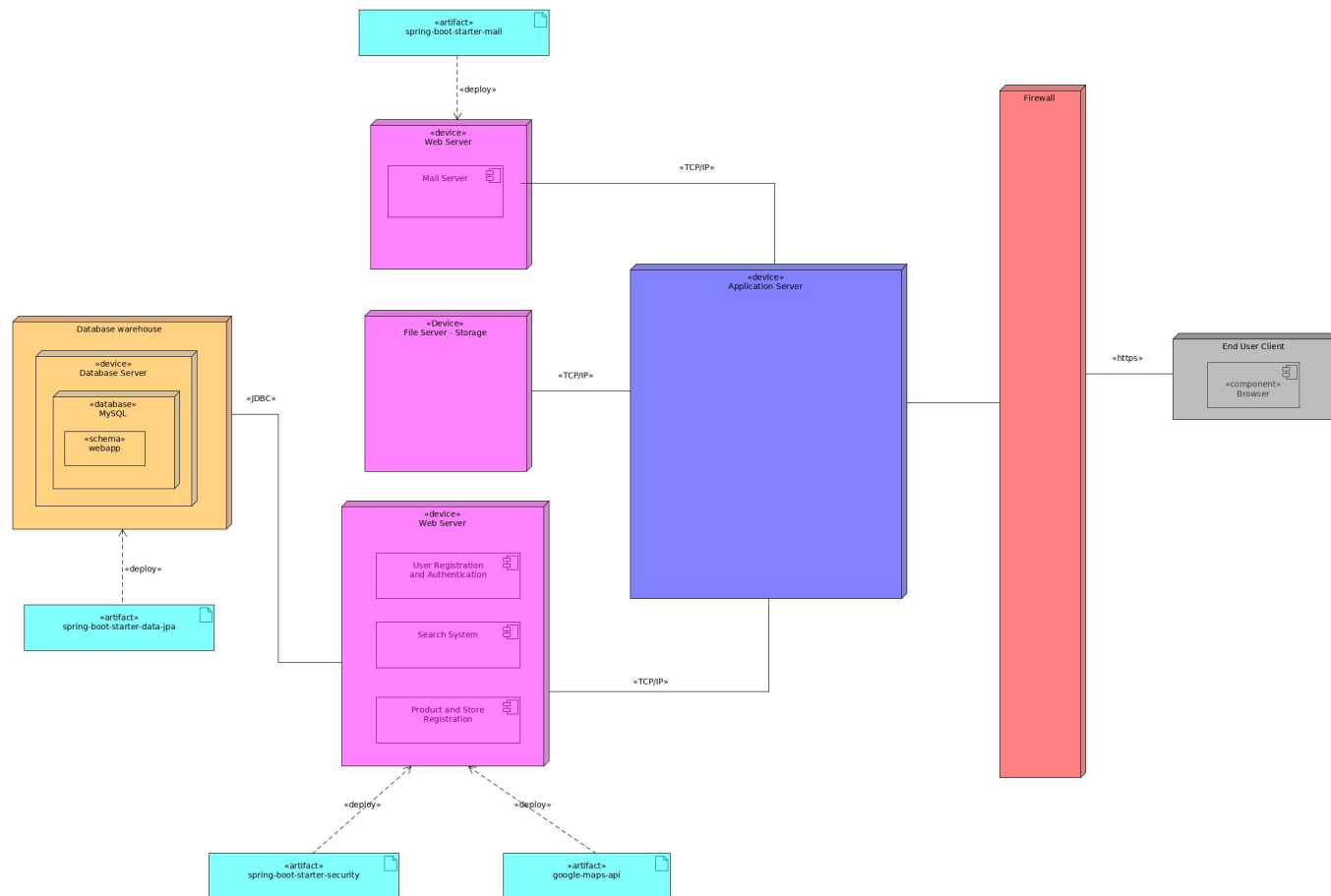
- Πλοήγηση στην αρχική σελίδα της εφαρμογής όπου αναγράφονται βασικές πληροφορίες σχετικά με την εξέλιξη της ιδέας και κατ' επέκταση της ιστοσελίδας
- Αναζήτηση καταχωρήσεων και φιλτράρισμα των αποτελεσμάτων με βάση πολλαπλά κριτήρια
- Επικοινωνία με τον διαχειριστή για να μεταφέρει σχόλια σχετικά με το *UX*
- Δημιουργία Λογαριασμού και Σύνδεση στην Εφαρμογή
- Πλοήγηση στο Δασηβοαρδ του χρήστη και ανανέωση των πληροφοριών του που θα είναι διαθέσιμες και στους υπόλοιπους χρήστες
- Εισαγωγή δεδομένων σχετικά με προϊόντα, καταστήματα και καταχωρήσεις, καθώς και η αντίστοιχη ανανέωσή και διαγραφή τους

Συνολικά, μπορούμε να αναπαραστήσουμε και τις λειτουργίες της εφαρμογής μας στο ακόλουθο διάγραμμα *Activity UML*:



1.3 Διεπαφές με Εξωτερικά Συστήματα

Όπως κάθε σύγχρονο έργο λογισμικού, η εφαρμογή μας επικοινωνεί, αλληλεπιδρά και αξιοποιεί μια πληθώρα από εξωτερικά λογισμικά, πόρους και βιβλιοθήκες για την αποδοτική και αποτελεσματική διεκπεραίωση των στόχων του λογισμικού. Για να γίνει καλύτερα κατανοητή η εξάρτηση αυτή, κρίνεται εύλογο να παρουσιάσουμε ένα διάγραμμα *UML – Deployment*, όπου αναπαριστώνται οι επιμέρους εξαρτήσεις της εφαρμογής μας με εξωτερικά λογισμικά καθώς και η δομή της σε επίπεδο κόμβων υλικού:



Το παραπάνω μοντέλο περιέχει και ορισμένους κόμβους που ενδεχομένως λόγω έλλειψης πόρων δεν θα είναι δυνατό να υλοποιηθούν, αλλά αποτελεί ένα πιο γενικευμένο καταναεμημένο μοντέλο αρχιτεκτονικής για έργα λογισμικού μεγαλύτερης κλίμακας. Στα πλαίσια της δικής μας εφαρμογής, έχουμε χρησιμοποιήσει μία εικονική μηχανή που φιλοξενεί τη βάση δεδομένων μας, ώστε τα μέλη της ομάδας μας να έχουν κοινή αναφορά στα δεδομένα της εφαρμογής. Επιπλέον, από τις παραπάνω εξαρτήσεις αξίζει να επιμείνουμε στο *API* της *Google* που μας πρόσφερε μία διεπαφή για την αποτελεσματική εισαγωγή και αναζήτηση καταστημάτων σε χάρτες, δυνατότητα ιδιαίτερης σημασίας για την εφαρμογή μας.

Στο σημείο αυτό, κρίνεται εύλογο να αναφέρουμε ότι για την υλοποίηση της εφαρμογής χρησιμοποιήσαμε το *framework Spring Boot* καθώς και το εργαλείο αυτοματισμού και χειρισμού εξαρτήσεων *Maven* που μας επιτρέπει να διαχειριζόμαστε εύκολα και αποδοτικά τις εξαρτήσεις μεταξύ επιμέρους τμημάτων λογισμικού.

1.4 Διεπαφές με τον Χρήστη

Για να οπτικοποιήσουμε καλύτερα τις διεπαφές που προσφέρει η εφαρμογή μας με τους διάφορους χρήστες και έχουμε ήδη αναφέρει παραθέτουμε το ακόλουθα διάγραμμα *UML Use Case*:



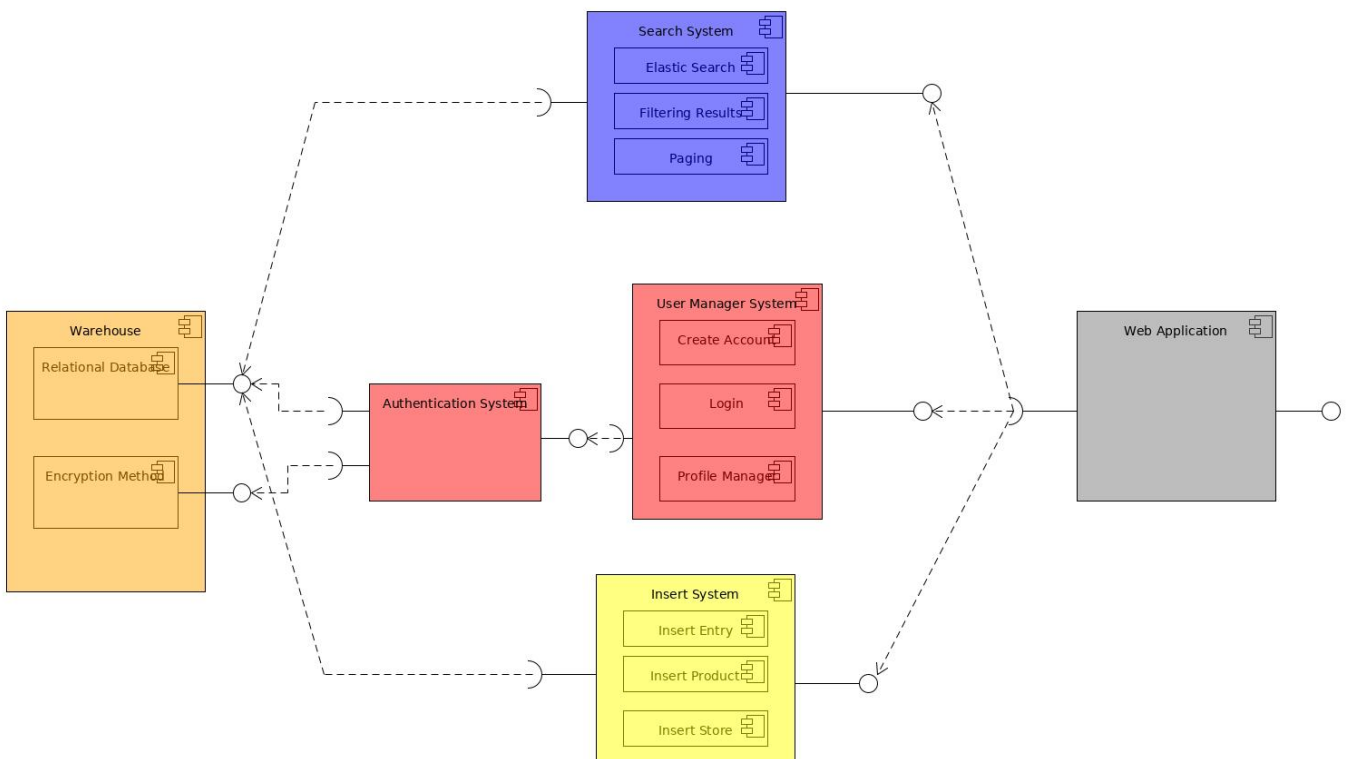
2 Προδιαγραφές απαιτήσεων Λογισμικού

2.1 Λειτουργικές Απαιτήσεις

Το παρών λογισμικό αποτελεί ένα ηλεκτρονικό παρατηρητήριο τιμών το οποίο χαρακτηρίζεται συνοπτικά από τις ακόλουθες βασικές λειτουργικότητες:

- Υποστήριξη διεπαφών χρήστη για αποτελεσματική αναζήτηση προϊόντων, των τιμών τους καθώς και του χωρικού αποτυπώματός τους, ενώ δίνεται ιδιαίτερη σημασία στην δυνατότητα παροχής ενός εύληπτου μηχανισμού σύγκρισης των επιμέρους καταχωρήσεων
- Εγγραφή χρηστών στην υπηρεσία
- Εισαγωγή, Ανανέωση και Διαγραφή των επιμέρους καταχωρήσεων από τους εγγεγραμμένους χρήστες

Στο πλαίσιο αυτό, παρουσιάζουμε το ακόλουθο διάγραμμα *UML Components* που απεικονίζονται σε υψηλό επίπεδο τα επίμερους τμήματα που υποστηρίζουν τις βασικές λειτουργικές απαιτήσεις.



2.2 Μη Λειτουργικές Απαιτήσεις

Επιπρόσθετα με τις παραπάνω απαιτήσεις, η επιτυχία της εφαρμογής μας εξαρτάται άμεσα και από ορισμένες προδιαγραφές που είναι ανεξάρτητες από τη καθαρή λειτουργικότητα της εφαρμογής. Για το λόγο αυτό, δόθηκε ιδιαίτερη έμφαση τόσο κατά τον σχεδιασμό όσο και κατά την υλοποίηση στις παρακάτω προδιαγραφές:

- **Απόδοση:** Η ελαχιστοποίηση του χρόνου απόκρισης για τις αναζητήσεις που πραγματοποιεί ο χρήστης αλλά και συνολικά για τις επιμέρους ενέργειες που υποστηρίζονται από την πλατφόρμα. Για να το εφασφαλίσουμε αυτό έγινε χρήση κατάλληλων *indexes* στη βάση δεδομένων για τη γρήγορη πραγματοποίηση *queries*, ενώ επίσης δόθηκε έμφαση στον σεβασμό των βασικών σχεδιαστικών αρχών που εξασφαλίζουν μεγαλύτερη απόδοση στην εφαρμογή
- **Ασφάλεια:** Η προστασία των προσωπικών δεδομένων του χρήστη χρησιμοποιώντας μεθόδους κρυπτογράφησης, αλλά και η χρήση πρωτοκόλλων ασφαλούς μεταφοράς δεδομένων (*https*). Συγκεκριμένα, χρησιμοποιήθηκε η μέθοδος *SaltPepper*, ενώ φροντίσαμε οι ευαίσθητες πληροφορίες των χρηστών, όπως είναι ο αριθμός τηλεφώνου, να μην είναι διαθέσιμες σε άλλους χρήστες
- **Διαθεσιμότητα:** Η εξασφάλιση της διαρκούς εύρυθμης λειτουργίας της ιστοσελίδας ανεξάρτητα από την ενδεχόμενη αποτυχία του δικτύου χρησιμοποιώντας κατακευκτωμένους *servers*. Στην περίπτωσή μας, χρησιμοποιήσαμε εικονικές μηχανές, ώστε στην περίπτωση αποτυχίας να υπάρχει *back – up* εναλλακτική
- **Επεκτασιμότητα:** Η οργάνωση και το στήσιμο της εφαρμογής με τέτοιο τρόπο που οι αλλαγές και οι επεκτάσεις των λειτουργικοτήτων να πραγματοποιούνται εύκολα και αποδοτικά
- **Απλότητα:** Το σύνολο των διεπαφών που προσφέρει η εφαρμογή να ακολουθούν τις βασικές σχεδιαστικές αρχές της αλληλεπίδρασης ανθρώπου μηχανής
- **Κομψότητα:** Η αισθητική πλευρά των διεπαφών του χρήστη

3 Λειτουργικές Περιπτώσεις Χρήσης

3.1 Εγγραφή ενός Χρήστη

3.1.1 Ρόλοι που Εμπλέκονται

Εμπλέκεται ο χρήστης που πραγματοποιεί την εγγραφή

3.1.2 Προϋποθέσεις Εκτέλεσης

Ο χρήστης θα πρέπει βρίσκεται στη σελίδα που πραγματοποιείται η εγγραφή και να συμπληρώσει τα απαιτούμενα πεδία με έγκυρες τιμές ανάλογα με το εκάστοτε πεδίο. Επίσης, στα πεδία *username* , *email* και *phone number* ο χρήστης απαιτείται να εισάγει μοναδικές τιμές που δεν υπάρχουν ήδη στη βάση.

3.1.3 Περιβάλλον Εκτέλεσης

Τα τμήματα που εμπλέκονται στη συγκεκριμένη περίπτωση χρήσης είναι η διαδικτυακή διεπαφή της εφαρμογής μας για εγγραφή, καθώς και το σύστημα διαχείρισης βάσης που καλείται να πραγματοποιήσει και να ελέγξει την εγκυρότητα των εισαγομένων πεδίων.

3.1.4 Δεδομένα Εισόδου

- Ονοματεπώνυμο
- *Username*
- *Email*
- Αριθμός Τηλεφώνου
- Κωδικός Πρόσβασης

3.1.5 Παράμετροι

- Ονοματεπώνυμο: Αποτελείται αποκλειστικά από αλφαβητικούς χαρακτήρες
- *Username*: Πρέπει να είναι μοναδικό για κάθε χρήστη, αφού ουσιαστικά αποτελεί το μοναδικό προσδιοριστικό για το *avatar* του χρήστη στη σελίδα
- *Email*: Πρέπει επίσης να είναι μοναδικό ώστε κάθε διεύθυνση να αντιστοιχεί σε ένα χρήστη, ενώ ακολουθεί συγκεκριμένη δομή που ελέγχεται κατά την εισαγωγή του
- Αριθμός τηλεφώνου: Οφείλει να είναι μοναδικό, ενώ είναι προαιρετικό πεδίο και αποτελείται από ακριβώς 10 ψηφία
- Κωδικό πρόσβασης: Πρέπει να αποτελείται από τουλάχιστον 10 χαρακτήρες

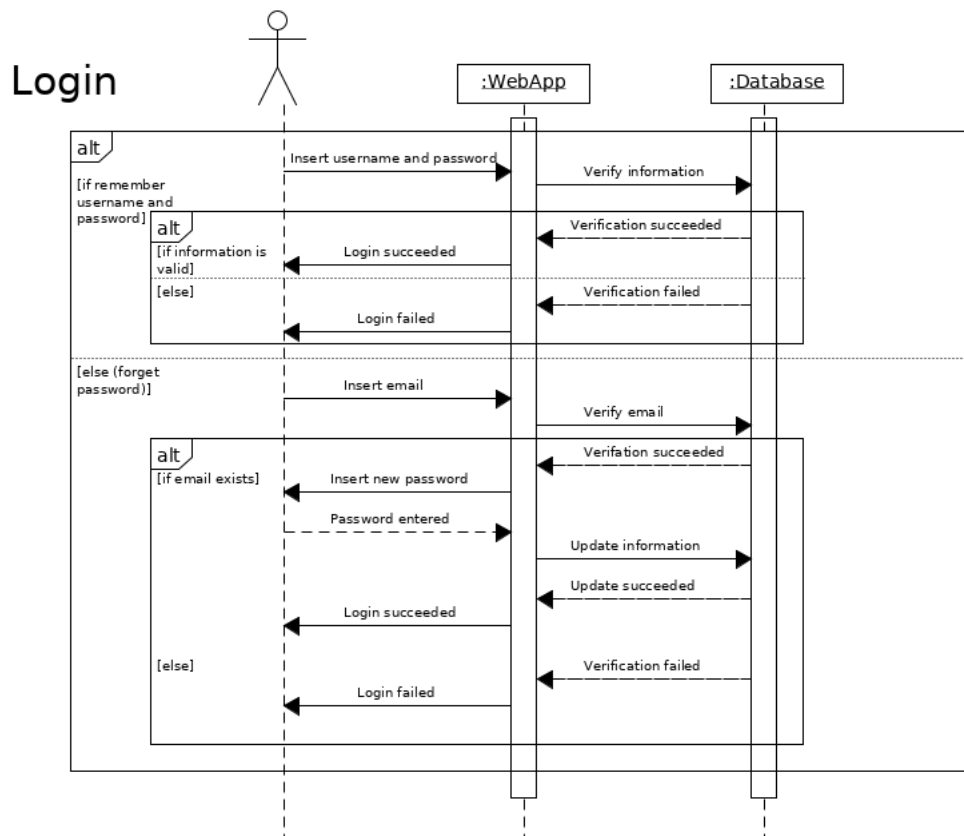
Μάλιστα, για τον κωδικό πρόσβασης έχουμε συμπεριλάβει ένα πεδίο που γίνεται η επιβεβαίωση του κωδικού πρόσβασης για λόγους μεγαλύτερης ασφάλειας καθώς κατά τη συγγραφή του κωδικού οι χαρακτήρες κωδικοποιούνται. Αξίζει να αναφερθεί επίσης, ότι όλα τα παραπάνω πεδία έχουν ένα εύλογο περιορισμένο μέγεθος που μπορεί να εισάγει ο χρήστης, ώστε να αποφύγουμε τη σπατάλη χώρου στη βάση δεδομένων.

3.1.6 Αλληλουχία Ενεργειών

Η αλληλουχία των ενεργειών είναι η εξής:

1. Συμπλήρωση όλων των απαιτούμενων πεδίων
2. Επιλογή κουμπιού *submit*
3. Σε περίπτωση που δεν συμπληρώθηκαν κάποια πεδία σύμφωνα με τους παραπάνω περιορισμούς ο χρήστης πρέπει να συμπληρώσει ξανά σωστά τα πεδία για να ολοκληρωθεί η εγγραφή

Πιο αναλυτικά και οπτικά, η παραπάνω διαδικασία παρουσιάζεται στο ακόλουθο *sequence diagram*:



3.1.7 Δεδομένα Εξόδου

Σε περίπτωση επιτυχημένης εγγραφής δημιουργείται ένας νέος χρήστης με τα αντίστοιχα χαρακτηριστικά και κατ' επέκταση μία νέα καταχώρηση στη βάση δεδομένων ενώ εμφανίζεται αντίστοιχο μήνυμα επιτυχίας στην οθόνη του χρήστη. Σε αντίθετη περίπτωση, εμφανίζεται μήνυμα αποτυχίας και παραμένει στην ίδια οθόνη για να συμπληρώσει τελικά τα στοιχεία σύμφωνα με τους περιορισμούς που έχουν επισημανθεί.

3.2 Αναζήτηση Προϊόντος

3.2.1 Ρόλοι που Εμπλέκονται

Εμπλέκεται και πάλι ένας χρήστης που είτε είναι εγγεγραμμένος-εθελοντής, είτε είναι απλός επισκέπτης της σελίδας.

3.2.2 Προϋποθέσεις Εκτέλεσης

Ο χρήστης θα πρέπει να βρίσκεται στην αρχική σελίδα της εφαρμογής και είτε να συμπληρώσει ένα *keyword* στη μηχανή αναζήτησης είτε να επιλέξει μία από τις καταγραφές.

3.2.3 Περιβάλλον Εκτέλεσης

Τα τμήματα που εμπλέκονται στη συγκεκριμένη περίπτωση χρήσης, είναι η διαδικτυακή διεπαφή της εφαρμογής μας για αναζήτηση, καθώς και το σύστημα διαχείρισης βάσης που καλείται να πραγματοποιήσει την εύρεση των προϊόντων.

3.2.4 Δεδομένα Εισόδου

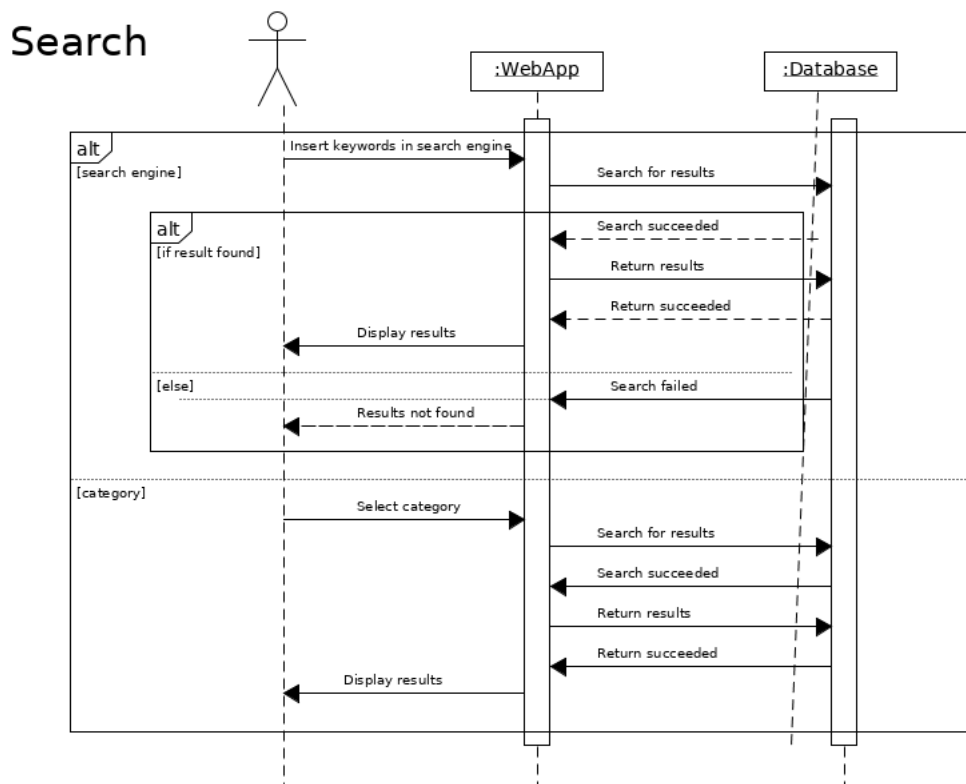
Εισαγωγή κειμένου σχετικού με τις ετικέτες (*tags*) του προϊόντος που αναζητάμε ή επιλογή με τον κέρσορα της επιθυμητής κατηγορίας μέσα από τη διεπαφή που μας προσφέρει η σελίδα.

3.2.5 Παράμετροι

Οποιαδήποτε μορφή κειμένου δίνεται από το πληκτρολόγιο, χωρίς να ξεπεράσει τους 50 χαρακτήρες

3.2.6 Αλληλουχία Ενεργειών

Η αλληλουχία των ενεργειών περιλαμβάνει είτε την είσοδο δεδομένων από τη μηχανή αναζήτησης είτε την επιλογή κάποιας κατηγορίας και παρουσιάζεται παρακάτω πιο αναπαραστατικά στο ακόλουθο *sequence diagram*:



3.2.7 Δεδομένα Εξόδου

Αφου πραγματοποιηθεί η αναζήτηση ο χρήστης μεταφέρεται σε μία νέα σελίδα όπου απεικονίζεται ένα πλέγμα με τα προϊόντα, ανάλογα με την αναζήτηση που πραγματοποιήθηκε. Σε περίπτωση που δεν υπάρχουν αποτελέσματα αναζήτησης εμφανίζεται σχετικό μήνυμα και ο χρήστης παραμένει στην αρχική οθόνη.

3.3 Προσθήκη Νέα Καταχώρησης

3.3.1 Ρόλοι που Εμπλέκονται

Η προσθήκη νέου προϊόντος μπορεί να πραγματοποιηθεί μόνο από εθελοντή, δηλαδή από έναν χρήστη που έχει πραγματοποιήσει εγγραφή στην υπηρεσία.

3.3.2 Προϋποθέσεις Εκτέλεσης

Ο χρήστης θα πρέπει να συμπληρώσει τα απαιτούμενα πεδία με έγκυρες τιμές ανάλογα με το εκάστοτε πεδίο.

3.3.3 Περιβάλλον Εκτέλεσης

Τα τμήματα που εμπλέκονται στη συγκεκριμένη περίπτωση χρήσης, είναι η διαδικτυακή διεπαφή της εφαρμογής μας για προσθήκη νέας καταχώρησης , καθώς και το σύστημα διαχείρισης βάσης που καλείται να πραγματοποιήσει την νέα καταχώρηση ελέγχοντας αν τα δεδομένα εισόδου(προϊόν ή κατάστημα) υπάρχουν ήδη στη βάση.

3.3.4 Δεδομένα Εισόδου

Ο χρήστης καλείται να εισάγει τις ακόλουθες πληροφορίες:

- *Barcode*
- Ετικέτες
- Αξιολόγηση
- Ετικέτες
- Όνομα καταστήματος
- Τοποθεσία καταστήματος
- Τιμή προϊόντος

Αν το *barcode* δεν υπάρχει στη βάση δεδομένων, ζητείται από την εθελοντή να κάνει την εισαγωγή του νέου προϊόντος εισάγοντας και τα εξής χαρακτηριστικά:

- Όνομα προϊόντος
- Κατασκευαστής
- Κατηγορία
- Εικόνα προϊόντος
- Περιγραφή προϊόντος

3.3.5 Παράμετροι

Τα παραπάνω δεδομένα εισόδου χαρακτηρίζονται από τους εξής περιορισμούς:

- Τιμή Προϊόντος: Είναι αριθμός διπλής ακρίβειας με περιορισμό δυο δεκαδικών ψηφίων
- *Barcode*: Περιέχει το πολύ 20 χαρακτήρες
- Όνομα καταστήματος: Αποτελείται από αλφαριθμητικούς χαρακτήρες

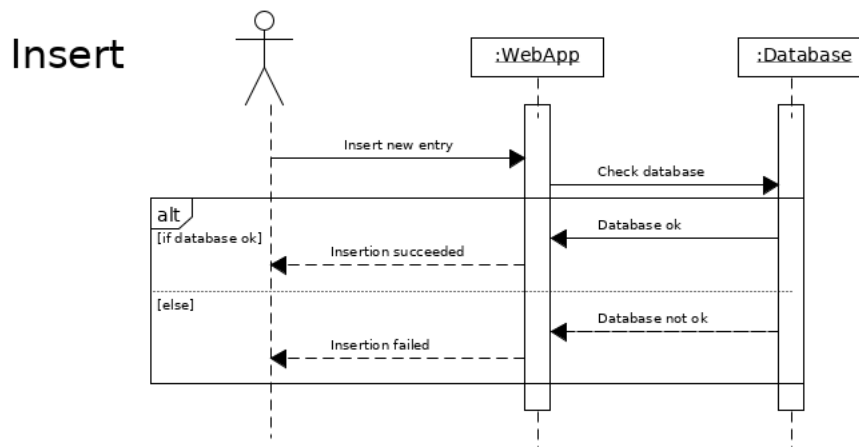
- Τοποθεσία καταστήματος: Η εισαγωγή της θα γίνει εισάγοντας τη διεύθυνση ή επιλέγοντας την τοποθεσία του στο χάρτη
- Ετικέτες: Αποτελούνται από χαρακτήρες χωρίς κενά διαστήματα
- Αξιολόγηση: Αριθμός από 0.5 μέχρι 5.0 με βήμα 0.5
- Όνομα προϊόντος: Αποτελείται από αλφαριθμητικούς χαρακτήρες
- Κατασκευαστής: Αποτελείται από αλφαριθμητικούς χαρακτήρες
- Κατηγορία: Αποτελείται από αλφαριθμητικούς χαρακτήρες
- Εικόνα Προϊόντος: Έγκυρο *URL* εικόνας
- Περιγραφή προϊόντος: Οποιαδήποτε μορφή κειμένου κρίνει ο χρήστης ότι απαιτείται για να περιγράψει το προϊόν

3.3.6 Αλληλουχία Ενεργειών

Η αλληλουχία των ενεργειών περιλαμβάνει τις ακόλουθες ενέργειες:

- Εισαγωγή *barcode*, ενώ αν δεν υπάρχει ήδη στη βάση καλείται να συμπληρώσει τα πεδία που σχετίζονται με το προϊόν
- Εισαγωγή αξιολόγησης και ετικετών για το προϊόν
- Εισαγωγή τοποθεσίας καταστήματος, ονόματος καταστήματος καθώς και τιμής του προϊόντος

Η παραπάνω αλληλουχία γίνεται καλύτερα κατανοητή μέσα από το παρακάτω *sequence diagram*:



3.3.7 Δεδομένα Εξόδου

Εμφανίζεται κατάλληλο μήνυμα που επιβεβαιώνει ότι η καταχώρηση ήταν επιτυχής, ενώ ο χρήστης μεταφέρεται εκ νέου στη σελίδα προφίλ.

3.4 Απαιτήσεις Επιδόσεων

Για να θεωρείται επιτυχημένη η εφαρμογή μας και η διεπαφή μας ενδιαφέρει ο χρόνος που απαιτείται για την πραγματοποίηση μιας καταχώρησης, ο αριθμός των λαθών που γίνονται από τη μεριά του χρήστη καθώς και η ικανοποίηση που λαμβάνει ο χρήστης από τη χρησιμοποίηση της διεπαφής. Συνολικά, η επίδοση της εφαρμογής και του λογισμικού μας θα καθοριστεί από το πλήθος των εγγεγραμμένων χρηστών, από το ποσοστό των εθελοντών που θα έχουν ενεργητικό ρόλο στην εφαρμογή καθώς και από την ποιότητα της βάσης δεδομένων που θα προκύψει τελικά μέσω από τις εθελοντικές καταχωρήσεις των χρηστών. Στο σημείο αυτό, μπορούμε να αναδείξουμε ένα βασικό κομμάτι της φιλοσοφίας μας ως σχεδιαστές Λογισμικού, που είναι η εμπιστοσύνη των εθελοντών για τη δημιουργία μίας αμερόληπτης και πλήρης βάσης δεδομένων, έναντι της συνεργασίας με μεγάλες αλυσίδες καταστημάτων, επιλογή που θα αδικούσε τα μικρά μαγαζιά και επιχειρήσεις.

3.5 Απαιτήσεις Οργάνωσης Δεδομένων

3.5.1 Τεχνική Περιγραφή των Δεδομένων

Για την αποδοτική αποθήκευση, διαχείριση και ανάκτηση των δεδομένων που εισάγονται από τη εφαρμογή έχει χρησιμοποιηθεί μία Σχεσιακή βάση δεδομένων και συγκεκριμένα το σύστημα διαχείρισης *MySQL*. Στο πλαίσιο αυτό, οι περιορισμοί που έχουμε ήδη αναφέρει και ελέγχονται στο *front – end* μέσα από τις διάφορες φόρμες πρέπει να είναι συνεπείς με τους περιορισμούς της βάσης δεδομένων, καθώς ένας καθόβουλος χρήστης θα μπορούσε να πραγματοποιήσει εισαγωγές παρακάπτοντας τις φόρμες, δημιουργώντας με αυτόν τον τρόπο ασυνέπειες στη βάση μας. Δηλαδή, κάθε εισαγωγή ή ανανέωση είναι ευθυγραμμισμένη με τους περιορισμούς που έχουμε ήδη αναφέρει. Όπως γίνεται αντιληπτό, σχετικά με τα πρότυπα των δεδομένων έχουν χρησιμοποιηθεί τα καθιερωμένα που σχετίζονται με σχεσιακές βάσεις δεδομένων και συγκεκριμένα με την *MySQL*. Έχουμε αναφέρει τους βασικούς περιορισμούς που υπόκεινται τα δεδομένα, ενώ στη συνέχεια παρουσιάζουμε μία πλήρη εικόνα σχετικά με το μοντέλο της εφαρμογής μέσω κατάλληλων διαγραμμάτων.

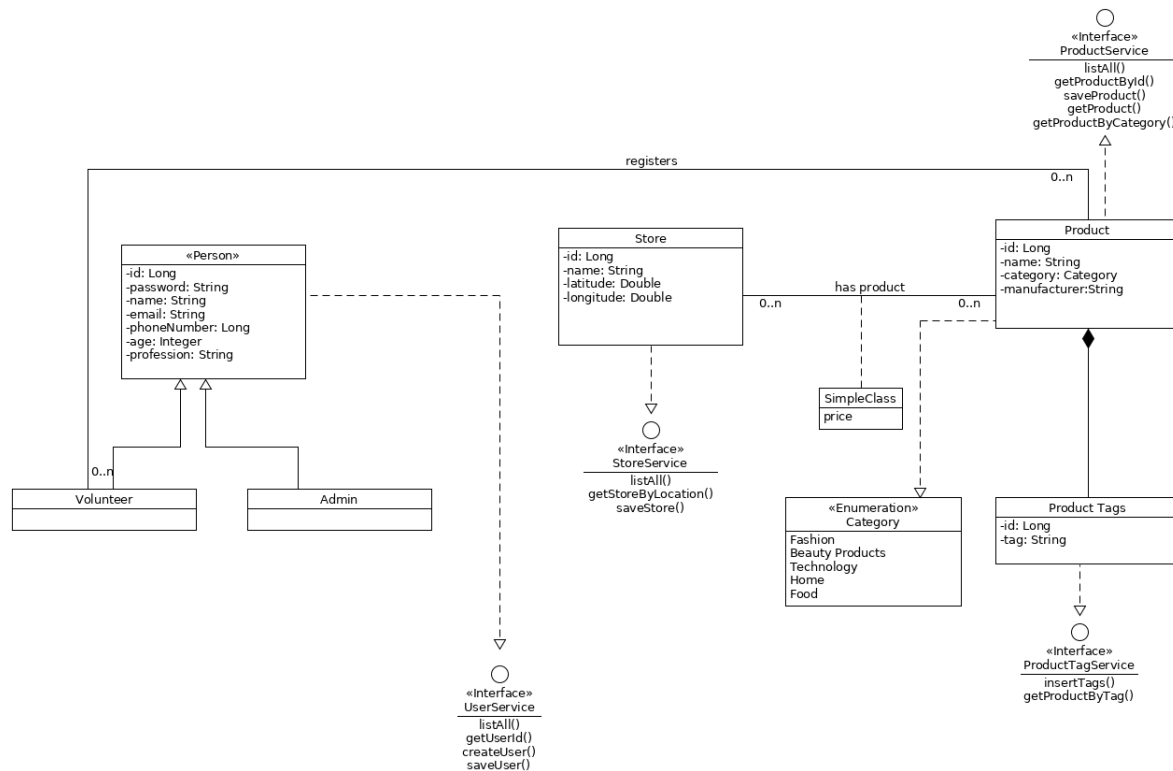
3.5.2 Απαιτήσεις και Περιορισμοί Πρόσβασης στα Δεδομένα

Κάθε χρήστης μπορεί να δει τα προσωπικά δεδομένα που έχει δημοσιεύσει ένας άλλος εγγεγραμμένος χρήστης στο προφίλ του, εκτός από ευαίσθητες πληροφορίες όπως ο αριθμός τηλεφώνων και φυσικά ο κωδικός πρόσβασης. Από την άλλη, οι πληροφορίες που σχετίζονται με τις καταχωρήσεις προϊόντων και καταστημάτων είναι διαθέσιμες χωρίς περιορισμούς, ενώ πρέπει να αναφερθεί ότι αν και ο διαχειριστής του συστήματος έχει πρόσβαση στην ολότητα της πληρο-

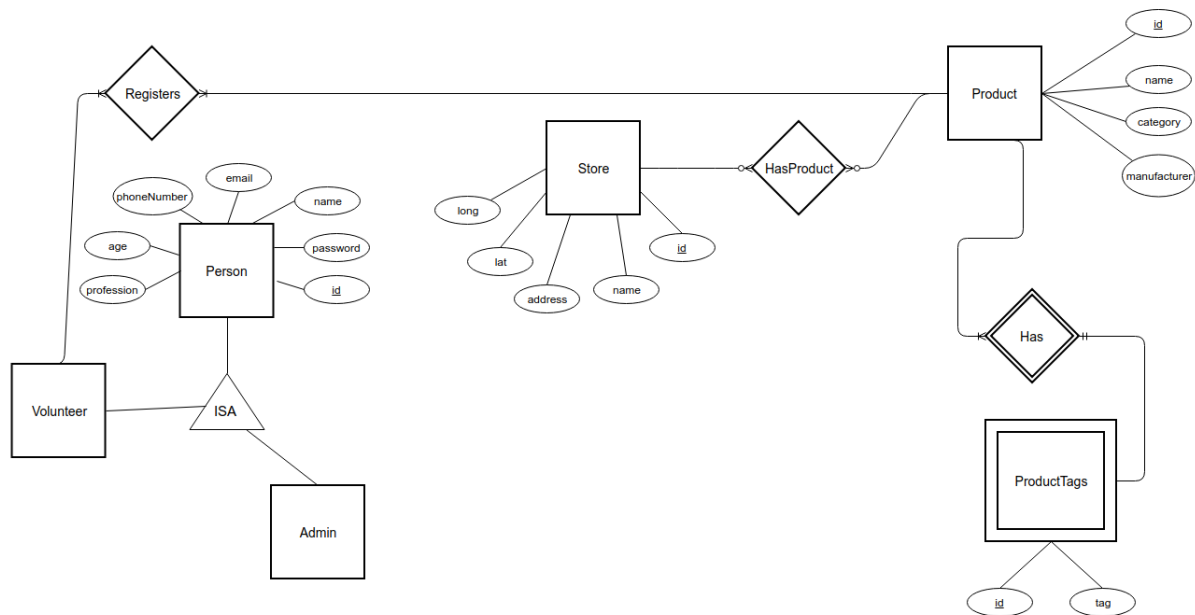
φορίας στη βάση δεδομένων, δεν έχει πρόσβαση στον κωδικό του εκάστοτε χρήστη καθώς είναι κρυπτογραφημένος.

3.5.3 Μοντέλο Δεδομένων

Για να δείξουμε τη δομή στην οποία βασίζεται το μοντέλο τη εφαρμογής παραθέτουμε το ακόλουθο διάγραμμα *UML*:



Το παραπάνω διάγραμμα υπ' άλλο πρίσμα, παρουσιάζεται στο ακόλουθο διάγραμμα *ER*:



3.5.4 Προδιαγραφές Ακεραιότητας Δεδομένων

Στη βάση δεδομένων αρχικά για να εξασφαλίσουμε την ακεραιότητα της κάθε οντότητας χρησιμοποιούμε ένα πρωτεύον κλειδί, ενώ επιπλέον έχουν χρησιμοποιηθεί οι κανόνες ακεραιότητας των δεδομένων που σχετίζονται με τις εξαρτήσεις των επιμέρους οντοτήτων, δηλαδή οι *foreign key constraints*. Συγκεκριμένα, η προδιαγραφή αυτή επιτυγχάνεται με τη χρήση *foreign – keys* και με ενέργειες όπως το *on – delete – cascade*, ώστε τα δεδομένα της βάσης μας να είναι συνεπή. Επιπλέον, οι προδιαγραφές ακεραιότητας επιτυγχάνονται χρησιμοποιώντας κατάλληλα *triggers* ώστε να ικανοποιούνται οι περιορισμοί που αναφέρθηκαν αλλά και να ικανοποιηθεί η απαίτηση για *ACID – Transactions*.

3.5.5 Προδιαγραφές Διατήρησης Δεδομένων

Για την διατήρηση δεδομένων μπορούμε να αναφέρουμε μεθόδους όπως η *RAID* για την ασφάλιση των δεδομένων με χρήση πλεονασμού της πληροφορίας στην περίπτωση που αποτύχει ένας δίσκος καθώς και την πολλαπλή αντιγραφή των κρίσιμων δεδομένων της εφαρμογής σε διαφορετικά φυσικά σχήματα.

3.6 Περιορισμοί Σχεδίασης

Σχετικά με τους περιορισμούς σχεδίασης ακολουθήσαμε το τυπικό πρότυπο που ακολουθείται κατά τον προγραμματισμό σε *Java*. Η συμμόρφωση στα πρότυπα αυτά αν και δεν είναι είναι υποχρεωτική μας βοηθάει να διατηρούμε ευανάγνωστο τον κώδικα για όλα τα μέλη της ομάδας, καθώς και για άλλα εξωτερικά άτομα. Οι βασικοί κανόνες είναι οι ακόλουθοι:

- Τα ονόματα Κλάσεων ξεκινούν με κεφαλαία γράμματα και ακολουθούν *camel case*
- Το ονόματα πεδίων σε Κλάσεις ξεκινούν με πεζό γράμμα, ενώ ακολουθούν επίσης *camel case*.
- Τα ονόματα των *packages* ξεκινούν με μικρό γράμμα, ενώ σε περίπτωση που αποτελούνται από πολλές λέξεις χωρίζονται από τελείες.

Συνολικά, κατά το σχεδιασμό του λογισμικού δόθηκε ιδιαίτερη έμφαση στη δομημένη και τμηματική ανάπτυξη του κώδικα, αποφεύγοντας τις επιακλύψεις και την περιττή δουλειά, κρατώντας την υλοποίηση όσο γίνεται απλή, κατανοητή και επεκτάσιμη.

3.7 Λοιπές Απαιτήσεις

3.7.1 Απαιτήσεις Διαθεσιμότητας

Όπως έχει ήδη αναφερθεί, για να εξασφαλίσουμε τη διαθεσιμότητα θα πρέπει το σύστημα να επικοινωνεί με πολλαπλούς *database servers*, ώστε να υπάρχουν πολλαπλά αντίγραφα για τη βάση δεδομένων σε διαφορετικούς παρόχους. Επίσης, θα πρέπει να συνεργαζόμαστε με παρόχους που μα εξασφαλίζουν 99.99% διαθεσιμότητα διαδικτυακής σύνδεσης μέσω των υποδομών τους.

3.7.2 Απαιτήσεις Ασφάλειας

Οι απαιτήσεις ασφάλειας συνοψίζονται στα ακόλουθα σημεία:

- Το σύστημα θα χρησιμοποιεί ασφαλή *sockets* σε όλες τις ανταλλαγές ευαίσθητων δεδομένων του χρήστη
- Το σύστημα θα αποσυνδέει αυτόματα τους χρήστες μετά από μια περίοδο αδράνειας
- Το σύστημα δεν θα προβάλλει ποτέ τον κωδικό του χρήστη κατά την πληκτρολόγηση, αλλά θα χρησιμοποιεί αντ' αυτού ειδικούς κωδικοποιημένους χαρακτήρες.
- Οι κωδικοί των χρηστών θα αποθηκεύονται στη βάση δεδομένων κρυπτογραφημένοι ώστε ακόμα και σε περίπτωση που ένας κακόβουλος χρήστης αποκτήσει πρόσβαση στη βάση μας, δε θα είναι σε θέση να ανιχνεύσει τον πραγματικό κωδικό.
- Ο *server* του συστήματος θα είναι προσβάσιμος μονάχα από επικυρωμένους διαχειριστές.

3.7.3 Απαιτήσεις Συντήρησης

Ο πηγαίος κώδικος που αντιστοιχεί στην εφαρμογή θα πρέπει να περνάει από τακτικούς ελέγχους και να αναθεωρείται με βάση και τις απαιτήσεις και την αλληλεπίδραση των χρηστών.

3.8 Παρατηρήσεις

Το παρών έγγραφο ενδέχεται να υποστεί αλλαγές ανάλογα με την πορεία της υλοποίησης, καθώς όπως γνωρίζουμε ένα έργο λογισμικού είναι δυναμικό και οι προδιαγραφές εξελίσσονται.