



**Πανεπιστήμιο Θεσσαλίας**

Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών και

Μηχανικών Υπολογιστών

**“Σχεδιασμός και Υλοποίηση Διαδικτυακής  
Πλατφόρμας για Ταίριασμα Παρόχων και Αποδεκτών”**

Διπλωματική Εργασία

**Τσολακίδης Γρηγόριος**

Επιβλέπων Καθηγητής:

Κατσαρός Δημήτριος

Επίκουρος Καθηγητής

Πανεπιστήμιο Θεσσαλίας

Βόλος, Ελλάδα

Οκτώβριος 2018





**University of Thessaly**

School of Engineering  
Department of Electrical and  
Computer Engineering

**“Web Platform Design and Implementation for Large  
Purchasing Departments”**

Diploma Thesis Project

**Tsolakidis Grigorios**

Supervisor:

Katsaros Dimitrios  
Assistant Professor  
University of Thessaly

*(Signature)*

.....

Volos, Greece

October 2018



Copyrights © Tsolakidis Grigorios, 2018

Με επιφύλαξη παντός δικαιώματος, All rights reserved. Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσεως, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας ή του αντικειμένου που πραγματεύεται πρέπει να απευθύνονται προς το συγγραφέα.

*“The beautiful thing about  
learning is that no one can  
take it away from you.  
Because what we learn with  
pleasure we never forget”*

## Ευχαριστίες

Με την περάτωση της παρούσας διπλωματικής εργασίας, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κύριο Κατσαρό Δημήτριο για την εμπιστοσύνη που έδειξε στο πρόσωπό μου και την καθοδήγηση που μου προσέφερε τόσο για την εκπόνησή της, όσο και κατά τη διάρκεια των σπουδών μου. Ευχαριστώ επίσης φίλους, συναδέλφους και μη, για την υποστήριξη τους αλλά και για τις όμορφες στιγμές που ζήσαμε αυτά τα χρόνια, και ιδιαίτερα τους προπτυχιακούς φοιτητές του τμήματος Χατζηιωάννου Πέτρο και Γεωργιάδη Δημήτριο με τους οποίους είχα την τιμή να συνεργαστώ άψογα τα τελευταία χρόνια. Επίσης τον πολύ καλό φίλο προπτυχιακό φοιτητή του τμήματος Μηχανολόγων Μηχανικών Κάργα Ευάγγελο, χωρίς τον οποίο τα φοιτητικά χρόνια θα ήταν πολύ διαφορετικά. Πάνω απ' όλα, είμαι ευγνώμων στην οικογένειά μου, στους γονείς μου, Απόστολο και Αικατερίνη, και στον αδερφό μου Γεώργιο, για την αμέριστη ενθάρρυνση, υποστήριξη και κατανόηση από μέρους τους προς το πρόσωπό μου κατά τη διάρκεια των φοιτητικών μου χρόνων.

*Τσολακίδης Α. Γρηγόριος  
Βόλος  
Οκτώβριος 2018*





## Περίληψη

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η υλοποίηση μιας διαδικτυακής πλατφόρμας / εφαρμογής που απευθύνεται σε μικρομεσαίες επιχειρήσεις και παρέχει τη δυνατότητα επικοινωνίας μέσω διαδικτύου για καθαρά επαγγελματικές συναλλαγές. Σκοπός της εφαρμογής δεν είναι να διαχειρίζεται τις συναλλαγές μεταξύ των εταιρών, αλλά να τους φέρει σε επικοινωνία εύκολα και γρήγορα, μέσω ενός εύχρηστου γραφικού περιβάλλοντος. Για τη χρήση της υπηρεσίας απαιτείται η εγγραφή του χρήστη μέσω ενός λογαριασμού email. Στη συνέχεια υλοποιήθηκαν οι λειτουργίες καταχώρησης προϊόντων και αναζήτησής τους με βάση τα χαρακτηριστικά που επιθυμεί ο χρήστης. Τέλος, ο χρήστης μπορεί να αναζητήσει άλλους χρήστες και να παρακολουθεί τη δραστηριότητά τους.



## Abstract

The subject of this diploma thesis is the implementation of a web platform / application addressed to small and medium enterprises that provides them with the possibility of communication through the internet for purely professional transactions. The purpose of the application is not to manage the transactions between the partners but to bring them into communication easily and quickly through a user-friendly graphical environment. Use of the service requires user registration via an email account. The product registration and search function were then implemented, based on the features that the user wants. Finally, the user can search for other users and follow their activity.



## Πίνακας Περιεχομένων

Κεφάλαιο 1 <sup>ο</sup> : Εισαγωγή .....	1
1.1. Γενικό Πλαίσιο .....	1
1.2. Αντίστοιχες εφαρμογές .....	1
1.3. Επισκόπηση .....	2
Κεφάλαιο 2 <sup>ο</sup> : Τεχνολογική Επισκόπηση .....	3
2.1. Περιβάλλον Εργασίας .....	3
2.2. Το πλαίσιο ανάπτυξης Ruby on Rails .....	5
2.2.1. Model-View-Controller .....	6
2.2.2. Ruby Gems .....	7
Κεφάλαιο 3 <sup>ο</sup> : Δομική Επισκόπηση .....	9
3.1. Βασικά Στοιχεία Ruby on Rails .....	9
3.2. Δομή καταλόγου .....	10
3.3. Μοντέλα .....	12
3.4. Όψεις .....	14
3.5. Ελεγκτές .....	17
Κεφάλαιο 4 <sup>ο</sup> : Χαρακτηριστικά της Εφαρμογής .....	21
4.1. Λειτουργίες Εφαρμογής .....	21
4.1.1. Εγγραφή Χρήστη .....	21
4.1.2. Προβολή Προφίλ Χρήστη .....	22
4.1.3. Αναζήτηση και Ακολούθηση Χρηστών .....	23
4.1.4. Καταχώρηση και Αναζήτηση/Σύγκριση Προϊόντων .....	25
Κεφάλαιο 5 <sup>ο</sup> : Επισκόπηση .....	27
5.1. Παραδοχές κατά την υλοποίηση .....	27
5.2. Ιδέες για Μελλοντικές Επεκτάσεις .....	27
Κεφάλαιο 6 <sup>ο</sup> : Αναφορές .....	29

## Ευρετήριο Εικόνων

Εικόνα 1: Το περιβάλλον εργασίας του Cloud9 .....	3
Εικόνα 2. Μοντέλο που απεικονίζει τη συνολική αρχιτεκτονική του πλαισίου (Πηγή) .....	5
Εικόνα 3. Η Αρχιτεκτονική του MVC (Πηγή) .....	6
Εικόνα 4. Το σχήμα της Βάσης Δεδομένων .....	13
Εικόνα 5. Κεντρική σελίδα .....	21
Εικόνα 6. Σελίδα εγγραφής νέου χρήστη .....	22
Εικόνα 7. Προφίλ μετά την ενεργοποίηση του λογαριασμού .....	22
Εικόνα 8. Εισαγωγή τοποθεσίας προφίλ μέσω ρυθμίσεων .....	23
Εικόνα 9. Αναζήτηση Χρηστών .....	24
Εικόνα 10. Προβολή προφίλ διαφορετικού λογαριασμού .....	24
Εικόνα 11. Προβολή χρηστών που ακολουθεί ο χρήστης .....	25
Εικόνα 12. Φόρμα και καταχώρηση νέου προϊόντος .....	25
Εικόνα 13. Αναζήτηση προϊόντων .....	26
Εικόνα 14. Επιλεγμένη σύγκριση προϊόντων μέσα από την αναζήτηση του χρήστη .....	26

# Κεφάλαιο 1<sup>ο</sup>: Εισαγωγή

## 1.1. Γενικό Πλαίσιο

Τα τελευταία χρόνια οι διαδικτυακές εφαρμογές έχουν αναπτυχθεί αρκετά, ώστε να αποτελούν αναπόσπαστο κομμάτι της καθημερινότητάς μας, τόσο σε προσωπικό όσο και σε επαγγελματικό επίπεδο. Παρατηρώντας την αγορά εργασίας, εύκολα συνειδητοποιεί κανείς ότι η ελληνική πραγματικότητα απέχει πολύ από τον υπόλοιπο κόσμο. Οι συναλλαγές μεταξύ επιχειρήσεων συνεχίζουν να πραγματοποιούνται με καθαρά συμβατικά μέσα, αφού οι επιχειρηματίες καλούνται να μεταβούν οι ίδιοι, κατά μεγάλη πλειοψηφία, στην έδρα του προμηθευτή αγαθών ή υπηρεσιών. Ως εκ τούτου, η γεωγραφική απόσταση εξακολουθεί να καθορίζει το εύρος και της δυνατότητας μιας επιχείρησης. Για το λόγο αυτό, η χρησιμότητα μιας πλατφόρμας που θα διευκολύνει τη διαδικασία αυτή γίνεται άμεσα εμφανής.

Για την παρούσα διπλωματική εργασία επιλέχθηκε η υλοποίηση μιας εφαρμογής με δυνατότητες δημιουργίας λεπτομερούς εταιρικού λογαριασμού, καταχώρησης προϊόντων ή/και υπηρεσιών και αναζήτησης των ανωτέρω. Σκοπός της υλοποίησης είναι η εφαρμογή να χρησιμοποιηθεί κυρίως από μικρομεσαίες επιχειρήσεις της ελληνικής αγοράς, οι οποίες δεν μπορούν να διαθέσουν μεγάλα κεφάλαια για την ευρεία αναζήτηση προμηθειών, ούτε και να βρουν απαραίτητα το αγοραστικό τους κοινό σε κοντινή γεωγραφική απόσταση.

## 1.2. Αντίστοιχες εφαρμογές

Μετά από επισταμένη έρευνα, εντοπίστηκαν ορισμένες πλατφόρμες, τόσο εγχώριες όσο και ξένης προέλευσης. Τα μειονεκτήματά τους εντοπίζονται σε δύο κύρια στοιχεία, είτε λόγω των χρεώσεων προς τους χρήστες απευθύνονται σε μεγάλες επιχειρήσεις, είτε περιλαμβάνουν πολύ συγκεκριμένο περιεχόμενο. Επιπλέον, κάποιες από αυτές δεν απευθύνονται αποκλειστικά σε επιχειρήσεις, απλώς διαθέτουν προϊόντα σε χαμηλές τιμές, που πλησιάζουν αυτές της χονδρικής πώλησης. Όμως μια επιχείρηση, όσο μικρή και αν είναι, δεν μπορεί να βασίζεται σε τέτοιου είδους συνεργασίες για τις προμήθειές της. Οι βασικότερες από την ελληνική αγορά είναι το [infodata.gr](http://infodata.gr) το οποίο κυρίως δημιουργεί επαφές στο εξωτερικό, το [www.b2bwave.com](http://www.b2bwave.com) που είναι λογισμικό διαχείρισης παραγγελιών με κόστος που ξεπερνά τα 70 ευρώ μηνιαίως και τέλος μια κατηγορία εξειδικευμένων εφαρμογών, όπως οι [greekcompaniesonline.com](http://greekcompaniesonline.com) , [www.easyexports.eu](http://www.easyexports.eu) , [www.pharmasolutions.gr](http://www.pharmasolutions.gr) και [yourgreekoliveoil.com](http://yourgreekoliveoil.com) . Άξιο αναφοράς είναι και το [www.alibaba.com](http://www.alibaba.com) , ξένης προέλευσης, που δε δημιουργεί σχέσεις μεταξύ επιχειρήσεων, αλλά περιστασιακές συναλλαγές.

Οι βασικές διαφορές της εφαρμογής που υλοποιήθηκε εντοπίζονται στο κόστος χρήσης, στην απλότητα της διάδρασης και στην καθιέρωση διαρκών και αξιόπιστων συνεργασιών. Κατά κανόνα, οι ελληνικές μικρομεσαίες επιχειρήσεις δε μπορούν να διαθέσουν μεγάλα κεφάλαια για την έρευνα αγοράς, ούτε διαθέτουν την τεχνογνωσία για να αξιοποιήσουν σύνθετες λύσεις λογισμικού για τις ανάγκες τους.

### 1.3. Γενική Επισκόπηση

Στο 1<sup>ο</sup> Κεφάλαιο γίνεται μια προσπάθεια εισαγωγής στο γενικό πλαίσιο του αντικειμένου της διπλωματικής εργασίας, όπως οι συνθήκες που οδήγησαν στην υλοποίηση της πλατφόρμας.

Στο 2<sup>ο</sup> Κεφάλαιο πραγματοποιείται μια πιο τεχνική επισκόπηση, με βάση τις τεχνολογίες που αξιοποιήθηκαν για την ανάπτυξη της εφαρμογής.

Στο 3<sup>ο</sup> Κεφάλαιο παρουσιάζεται το μοντέλο της αρχιτεκτονικής της πλατφόρμας (MVC), ενώ γίνεται και εκτενής ανάλυση των επιμέρους στοιχείων του και των ρόλων του καθενός από αυτά.

Στο 4<sup>ο</sup> Κεφάλαιο αναλύονται οι βασικές λειτουργίες της εφαρμογής. Τη λεπτομερή ανάλυση συμπληρώνουν στιγμιότυπα του περιβάλλοντος χρήσης της εφαρμογής.

Στο 5<sup>ο</sup> Κεφάλαιο περιέχονται ιδέες για μελλοντικές επεκτάσεις με σκοπό τον εμπλουτισμό της πλατφόρμας, τόσο στο περιεχόμενο όσο και στη λειτουργικότητα. Επίσης γίνεται μια αναφορά σε κάποιες παραδοχές που έγιναν σχετικά με την υλοποίηση στοιχείων της εφαρμογής.

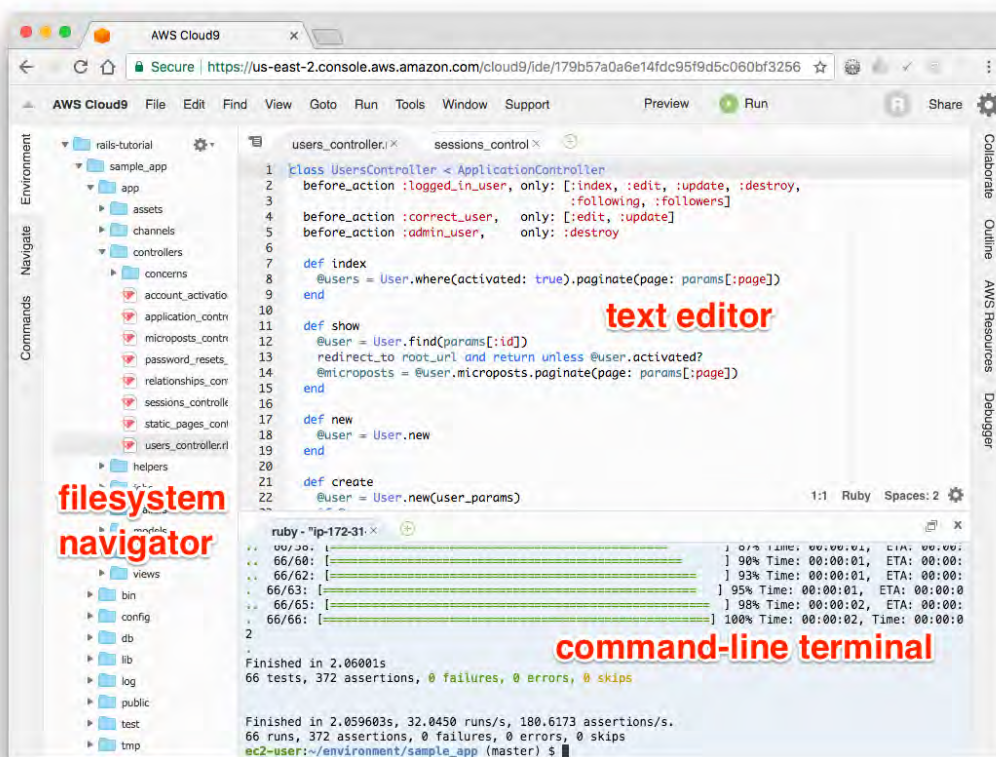
Στο 6<sup>ο</sup> Κεφάλαιο περιλαμβάνονται οι αναφορές της βιβλιογραφίας για την παρούσα εργασία.



## Κεφάλαιο 2<sup>ο</sup>: Τεχνολογική Επισκόπηση

### 2.1. Περιβάλλον Εργασίας

Λαμβάνοντας υπόψη διάφορες ιδιοσυγκρασιακές προσαρμογές, πιθανότατα υπάρχουν τόσα αναπτυξιακά περιβάλλοντα όπως και οι προγραμματιστές. Για να αποφευχθεί αυτή η πολυπλοκότητα, η ανάπτυξη της πλατφόρμας πραγματοποιήθηκε στο εξαιρετικό “cloud” περιβάλλον ανάπτυξης Cloud9 [1], μέρος του Amazon Web Services (AWS). Το περιβάλλον εργασίας είναι προ-διαμορφωμένο με το μεγαλύτερο μέρος του λογισμικού που απαιτείται για ανάπτυξη, ακόμα και σε επαγγελματικό επίπεδο. Περιλαμβάνει μόνο τα τρία βασικά συστατικά που απαιτούνται για την ανάπτυξη εφαρμογών: επεξεργαστή κειμένου, διαχειριστή αρχείων και τερματικό γραμμής εντολών.



Εικόνα 1: Το περιβάλλον εργασίας του Cloud9

Η ανάπτυξη πραγματοποιήθηκε με χρήση του πλαισίου ανάπτυξης λογισμικού ιστού (web application framework) Rails [2] γραμμένο στη γλώσσα προγραμματισμού Ruby [3]. Όλα τα δυναμικά δεδομένα της εφαρμογής αποθηκεύονται σε μια βάση δεδομένων. Για την διαχείριση της βάσης δεδομένων χρησιμοποιήθηκε η SQLite Εκδ. 3 [4] στο στάδιο της ανάπτυξης και η PostgreSQL [5] κατά το στάδιο της παραγωγής. Για τον έλεγχο του πηγαίου κώδικα έγινε χρήση του συστήματος διαχείρισης αναθεωρήσεων λογισμικού Git [6] και δημιουργήθηκε ένα απομακρυσμένο εναποθετήριο (repository) στο Bitbucket [7]. Εξίσου εύκολα θα μπορούσε να χρησιμοποιηθεί και το Github [8], αλλά καθώς περιλαμβάνονται ευαίσθητα δεδομένα, όπως κρυπτογραφικά κλειδιά και κωδικοί προτιμήθηκε το πρώτο.

Φυσικά ο πηγαίος κώδικας μπορεί να γίνει διαθέσιμος σε οποιονδήποτε, μετά από προσωπική επικοινωνία με το δημιουργό κάνοντας χρήση των διαθέσιμων Secure Shell (SSH) κλειδιών. Επιπλέον, αξιοποιήθηκε η διεπαφή προγραμματισμού εφαρμογών (API) που προσφέρει η Google [9] για τους χάρτες και διάφορα εργαλεία παραμετροποίησης όπως η μετατροπή διευθύνσεων σε συντεταγμένες και το αντίστροφο, ενισχύοντας τις δυνατότητες της αναζήτησης στην πλατφόρμα.

Μέχρι και τη στιγμή συγγραφής της παρούσας διπλωματικής εργασίας η πλατφόρμα βρίσκεται αναρτημένη στο διαδίκτυο σε 24ωρη βάση. Για να καταστεί αυτό δυνατό, φιλοξενείται από μία διαδικτυακή υπηρεσία, το Heroku [10]. Η ανάπτυξη Rails εφαρμογών υπήρξε αρκετά δύσκολη, αλλά το οικοσύστημα έχει ωριμάσει γρήγορα τα τελευταία χρόνια και πλέον υπάρχουν αρκετές αξιόπιστες επιλογές. Η επιλογή του Heroku όμως έγινε διότι η πλατφόρμα έχει σχεδιαστεί ειδικά για να φιλοξενεί Rails εφαρμογές. Είναι εξαιρετικά εύκολη η ανάπτυξη τέτοιων εφαρμογών – όσο ο πηγαίος κώδικας βρίσκεται σε κάποιο εναποθετήριο. Επίσης, η ελεύθερη βαθμίδα του Heroku είναι περισσότερο από επαρκής για τους σκοπούς της συγκεκριμένης υλοποίησης.

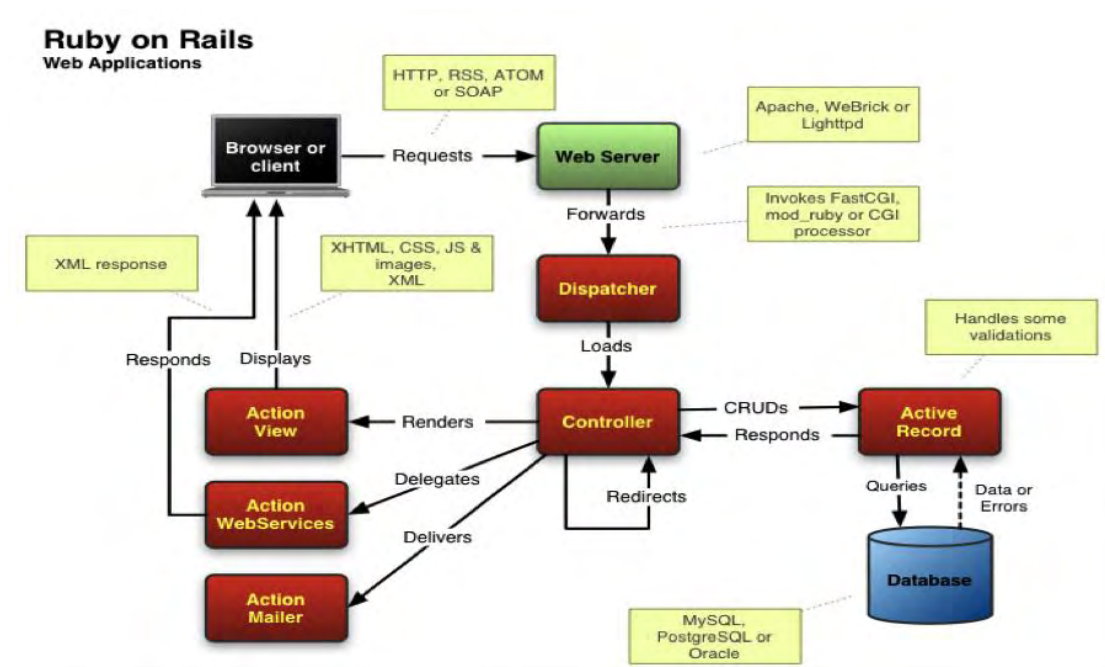
Για την καλύτερη διαχείριση της εφαρμογής, το Heroku παρέχει ένα εργαλείο διαχείρισης δεδομένων για τις PostgreSQL βάσεις της εφαρμογής, το Heroku Data Explorer [11]. Το συγκεκριμένο εργαλείο καθιστά πολύ εύκολη τη διαχείριση των δεδομένων, καθώς αυτή πραγματοποιείται μέσα από τον περιηγητή (browser) σε ένα εύχρηστο περιβάλλον. Τέλος, για την ενεργοποίηση των λογαριασμών από τους χρήστες ή για την αλλαγή του κωδικού πρόσβασης του λογαριασμού τους απαιτείται η μετάβαση μέσω ενός συνδέσμου σε μία προσωρινή σελίδα, ειδικά διαμορφωμένη για αυτό το λόγο. Ο σύνδεσμος αποστέλλεται μέσω ηλεκτρονικής αλληλογραφίας (email) στο προσωπικό ηλεκτρονικό ταχυδρομείο του χρήστη. Αυτή η τεχνική χρησιμοποιείται για να αποτρέψει τις απάτες και να μπλοκάρει τις προσπάθειες δημιουργίας υπερβολικά μεγάλου αριθμού χρηστών από ένα άτομο, και είναι ιδιαίτερα διαδεδομένη σε όλες σχεδόν τις σύγχρονες σελίδες. Για να καταστεί αυτό δυνατό, έγινε χρήση ενός προσθέτου από το Heroku, του SendGrid [12]. Μέσω αυτού, το email με τις αντίστοιχες πληροφορίες αποστέλλεται αυτόματα στο χρήστη, χωρίς να χρειάζεται ανθρώπινη παρέμβαση, διατηρώντας ταυτόχρονα την ασφάλεια στα επιθυμητά επίπεδα.

## 2.2. Το πλαίσιο ανάπτυξης Ruby on Rails

Το Ruby on Rails είναι ένα full-stack web development framework [13] της γλώσσας προγραμματισμού Ruby, γραμμένο υπό την MIT άδεια [14], μια άδεια ελεύθερου λογισμικού που προέρχεται από το Ινστιτούτο Τεχνολογίας της Μασαχουσέτης (MIT). Έχει δημιουργηθεί με βάση την αρχιτεκτονική Model-View-Controller (MVC), η οποία επιβάλλει ένα διαχωρισμό μεταξύ των δεδομένων της εφαρμογής (όπως τα δεδομένα των χρηστών) και του κώδικα που χρησιμοποιείται για την εμφάνισή τους. Παρέχει τη στοιχειώδη δομή για τα βασικά στοιχεία, από τη βάση δεδομένων για τη διαχείριση των πόρων μέχρι την αναπαράστασή τους στην πλευρά του χρήστη. Επίσης ενθαρρύνει και διευκολύνει τη χρήση γνωστών μεθόδων και προτύπων μηχανικής λογισμικού, όπως:

- “Don’t Repeat Yourself” (DRY), που αποτρέπει την επανάληψη ίδιων κομματιών κώδικα, ώστε να γίνεται πιο εύκολα η συντήρηση, η επέκταση και η διόρθωση λαθών.
- “Convention over Configuration” (CoC), ώστε να υπάρχουν συμβάσεις και προεπιλογές για να γίνουν κάποια πράγματα πιο εύκολα, αντί να απαιτείται να οριστούν άπειρες λεπτομέρειες μέσω ατελείωτων αρχείων ρυθμίσεων.
- “Active Record Pattern” (Ενεργό Πρότυπο Καταγραφής), το οποίο αποτελεί μια προσέγγιση για τη σύνδεση των δεδομένων με μία βάση δεδομένων, όπου κάθε αντικείμενο συνδέεται με μία μόνο γραμμή ενός πίνακα, η οποία δημιουργείται, ανανεώνεται και διαγράφεται ανάλογα με την αντίστοιχη χρήση του αντικειμένου.

Τα ανωτέρω περιγραφέντα χαρακτηριστικά καθώς και οι αλληλεπιδράσεις μεταξύ μερικών από αυτά παρουσιάζονται σχηματικά στην παρακάτω εικόνα.

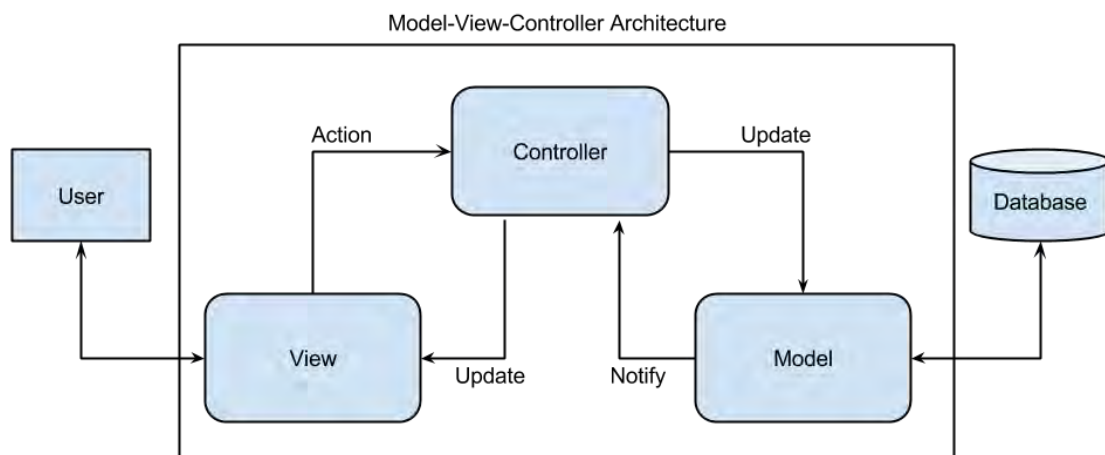


Εικόνα 2. Μοντέλο που απεικονίζει τη συνολική αρχιτεκτονική του πλαισίου (Πηγή)

### 2.2.1. Model-View-Controller (MVC)

Όπως αναφέρθηκε και προηγουμένως, το Rails framework έχει δημιουργηθεί με βάση το αρχιτεκτονικό μοτίβο MVC που χρησιμοποιείται ευρέως για την ανάπτυξη διεπαφών χρήστη (user interfaces) και χωρίζει την εφαρμογή σε τρία διασυνδεδεμένα μέρη. Αυτό γίνεται για να διαχωριστούν οι εσωτερικές αναπαραστάσεις πληροφοριών από τους τρόπους με τους οποίους οι πληροφορίες παρουσιάζονται στο χρήστη. Το πρότυπο MVC αποσυνδέει αυτά τα κύρια στοιχεία επιτρέποντας την αποτελεσματική επαναχρησιμοποίηση και την παράλληλη ανάπτυξη κώδικα. Τα τρία αυτά βασικά μέρη είναι:

- **Model:** Το μοντέλο ορίζει ποια δεδομένα περιλαμβάνει η εφαρμογή. Είναι η δυναμική δομή δεδομένων της εφαρμογής, ανεξάρτητα από το περιβάλλον χρήστη. Διαχειρίζεται άμεσα τα δεδομένα, τη λογική και τους κανόνες της εφαρμογής και λαμβάνει είσοδο από τον ελεγκτή.
- **View:** Η όψη ορίζει πως εμφανίζονται τα δεδομένα της εφαρμογής. Παρέχει τη διάδραση του χρήστη με την εφαρμογή, συνήθως μέσω μιας ιστοσελίδας. Τα στοιχεία της όψης παρέχουν δεδομένα στο χρήστη και αποστέλλουν τις ενέργειες στον ελεγκτή για τη διαχείριση των δεδομένων.
- **Controller:** Ο ελεγκτής βασικά παρέχει τη διεπαφή μεταξύ της όψης και του μοντέλου. Περιλαμβάνει τη λογική με βάση την οποία ανανεώνεται το μοντέλο ή/και η όψη με βάση την είσοδο που παρέχει ο χρήστης.



Εικόνα 3. Η Αρχιτεκτονική του MVC (Πηγή)

### 2.2.2. Ruby Gems

Για τους προγραμματιστές που χρησιμοποιούν τη γλώσσα προγραμματισμού Ruby, μια ιδιαίτερα ενεργή και αναπτυσσόμενη κοινότητα χρηστών, υπάρχει μια πληθώρα πακέτων και βιβλιοθηκών που παρέχουν τη λειτουργικότητα για την αυτοματοποίηση της ανάπτυξης της εφαρμογής. Αυτά τα πακέτα ονομάζονται gems και η διαδικασία λήψης, εγκατάστασης και ανανέωσής τους πραγματοποιείται με τη χρήση του RubyGems [15] package manager framework. Στη σελίδα φιλοξενούνται σχεδόν 10.000 διαφορετικά gems για κάθε σκοπό, από συστήματα διαχείρισης περιεχομένου και λύσεις για «ανέβασμα» (upload) αρχείων μέχρι API's και μηχανές εσωτερικής αναζήτησης με περισσότερα από 26.000.000.000 «κατεβάσματα» (downloads). Στο αρχείο Gemfile στη ρίζα του καταλόγου βρίσκονται οι εκδόσεις και οι εξαρτήσεις των gems. Τα gems που χρησιμοποιήθηκαν για την ανάπτυξη της παρούσας εργασίας παρατίθενται παρακάτω, μαζί με τη λειτουργικότητα που εξυπηρετούν.

- **Rails:** Το ίδιο το Rails framework αποτελεί και αυτό ένα gem.
- **Font-awesome-rails:** Παρέχει γραμματοσειρές, εικονίδια και γενικά κανόνες για το στυλ της εφαρμογής.
- **Sass-rails:** Παρέχει την επίσημη ενσωμάτωση της γλώσσας Sass [16] για εφαρμογές σε Ruby on Rails.
- **Bootstrap-sass:** Παρέχει μια έκδοση του Bootstrap [17], μιας συλλογής εργαλείων ανοιχτού κώδικα για τη δημιουργία διαδικτυακών εφαρμογών, που υποστηρίζει τη γλώσσα Sass.
- **Inline\_svg:** Παρέχει μια βοηθητική μέθοδο Rails για την αναγνώριση και διαμόρφωση εγγράφων SVG με χρήση της γλώσσας CSS [18].
- **jQuery:** Παρέχει μια έκδοση της jQuery [19], την πλέον διαδεδομένη βιβλιοθήκη της γλώσσας προγραμματισμού JavaScript [20], για τη διαχείριση της διεπαφής με το χρήστη.
- **jQuery-rails:** Παρέχει τους αυτοματισμούς για τη χρήση της jQuery σε εφαρμογές Ruby on Rails.
- **Will\_paginate:** Παρέχει ένα απλό API για τη σελιδοποίηση δεδομένων της εφαρμογής.
- **Bootstrap-will\_paginate:** Επεκτείνει τις δυνατότητες του will\_paginate για να ταιριάζει με το στυλ του Bootstrap.
- **Ransack:** Επιτρέπει τόσο τη δημιουργία απλών και εξελιγμένων φορμών αναζήτησης, όσο και την επεξεργασία, μορφοποίηση και διαχείριση των δεδομένων της αναζήτησης για προβολή.
- **Puma:** Είναι ένας παράλληλος διακομιστής (server) HTTP [21] για εφαρμογές σε Ruby on Rails. Το Puma προορίζεται για χρήση τόσο σε περιβάλλον ανάπτυξης όσο και σε περιβάλλον παραγωγής.
- **Uglifier:** Ελαχιστοποιεί τα JavaScript αρχεία, κάνοντας το εργαλείο συμπίεσης UglifyJS [22] διαθέσιμο σε Ruby.
- **Coffee-rails:** Παρέχει μια έκδοση της γλώσσας προγραμματισμού CoffeeScript [23]. Η CoffeeScript μετατρέπεται σε JavaScript πριν εκτελεστεί (μετατρέποντας τα .coffee αρχεία σε .js) και επιπλέον παρέχει τη δυνατότητα απάντησης σε αιτήσεις JavaScript με τη χρήση αρχείων .js.coffee ως όψεις).
- **Turbolinks:** Επιταχύνει την απόδοση της εμπειρίας πλοήγησης στην εφαρμογή. Χρησιμοποιεί τη λογική των εφαρμογών μιας σελίδας (Single-Page Application) [24], κρατώντας μία σελίδα «ενεργή» και αλλάζοντας κάθε φορά το σώμα ( <body> ) και τον

τίτλο στην κεφαλή ( `<head>` ), χωρίς να επιβαρύνει τη διαδικασία με το κόστος πλήρους φόρτωσης της σελίδας από την αρχή.

- **JBuilder:** Παρέχει μια απλή εκδοχή μιας Domain-Specific Language (DSL) [25] για τη δημιουργία δομών JSON (JavaScript Object Notation) [26] για την καταπολέμηση των γιγαντιαίων κατακερματισμών.
- **Figaro:** Διευκολύνει τη διαμόρφωση Rails εφαρμογών για ασφαλή χρήση των υπηρεσιών που παρέχει το Heroku.
- **Stub:** Επιτρέπει τη χρήση stubs («ψεύτικες» μέθοδοι που επιστρέφουν προκαθορισμένες τιμές για μερικώς ορισμένα αντικείμενα) για τη δοκιμή μεθόδων, χωρίς ούτε να κληθούν στην πραγματικότητα ούτε να αποθηκευτεί οτιδήποτε στη βάση δεδομένων.
- **Faker:** Χρησιμοποιείται για την εύκολη παραγωγή μη υπαρκτών δεδομένων όπως ονόματα, διευθύνσεις, τηλέφωνα, κτλ., για τη δοκιμή και τον έλεγχο της εφαρμογής.
- **Mini\_magick:** Χρησιμοποιείται για τη διαχείριση και επεξεργασία εικόνων με την ελάχιστη δυνατή χρήση μνήμης.
- **Carrierwave:** Διευκολύνει το ανέβασμα αρχείων στη βάση δεδομένων ή σε κάποιο cloud storage [27], όπως το Amazon S3 Bucket που χρησιμοποιείται στην παρούσα εφαρμογή. Για τους σκοπούς της εργασίας, χειριζόμαστε μόνο εικόνες για τα προφίλ των χρηστών και τα προϊόντα.
- **Bootsnap:** Αποτελεί μια βιβλιοθήκη που συνδέεται με τη Ruby για βελτιστοποίηση και προσωρινή αποθήκευση δαπανηρών υπολογισμών.
- **Geocoder:** Παρέχει δυνατότητες γεωγραφικής κωδικοποίησης ή/και αποκωδικοποίησης αντικειμένων, δηλαδή μετατροπή διευθύνσεων σε γεωγραφικές συντεταγμένες (γεωγραφικό μήκος και πλάτος) και το αντίστροφο, καθώς και γεωγραφικών ερωτημάτων.
- **Gmaps4rails:** Απλοποιεί τη δυνατότητα ενσωμάτωσης χαρτών μέσω του αντίστοιχου API του Google Maps, ενώ ταυτόχρονα παρέχει και διάφορα εργαλεία για την παραμετροποίηση των χαρτών.
- **Tzinfo-data:** Παρέχει χρονολογικά δεδομένα από τη Βάση IANA Time Zone [28] σε μορφή που είναι εύκολα διαχειρίσιμη σε γλώσσα Ruby. Χρησιμοποιείται για την καταγραφή ώρας και ημερομηνίας σε διάφορα μέρη της εφαρμογής.

## Κεφάλαιο 3<sup>ο</sup>: Δομική Επισκόπηση

### 3.1. Βασικά Στοιχεία Ruby on Rails

Το Rails framework παρέχει μια πλήρη σειρά βασικών στοιχείων για τη δημιουργία εφαρμογών ιστού. Οι βασικές αυτές κλάσεις είναι:

- **Action Controller** (`ActionController::Base`) : Ο Ελεγκτής Δράσης είναι το στοιχείο που διαχειρίζεται τους ελεγκτές σε μία εφαρμογή Rails. Το framework αυτό επεξεργάζεται τα εισερχόμενα αιτήματα, εξάγει παραμέτρους και τις αποστέλλει στην προβλεπόμενη ενέργεια. Ως ενέργεια ορίζεται μια δημόσια μέθοδος στον ελεγκτή, η οποία θα γίνει αυτόματα προσβάσιμη στον web-server μέσω του Δρομολογητή (Router). Ο Rails Router αναγνωρίζει τις διευθύνσεις URL [29] και τις αποστέλλει στην ενέργεια του αντίστοιχου ελεγκτή. Τα δύο βασικά πρότυπα δράσης [`render` και `redirect_to`] που χρησιμοποιούνται στους Ελεγκτές Δράσης είναι το `get-and-show` (πάρε και δείξε) και το `do-and-redirect` (κάνε και ανακατεύθυνε). Οι υπόλοιπες ενέργειες είναι παραλλαγές αυτών.
- **Action View** (`ActionView::Base`) : Η Όψη Δράσης διαχειρίζεται τις όψεις μια εφαρμογής Rails. Δημιουργεί τις εξόδους είτε σε κάποια από τις προεπιλεγμένες μορφές (HTML [30], XML [31]) είτε σε κάποιο πρότυπο, πχ συνδυασμό κώδικα `embedded Ruby (ERb)` [32] και κώδικα HTML.
- **Active Record** (`ActiveRecord::Base`) : Η Ενεργή Εγγραφή είναι το επίπεδο αντικειμενοστραφούς χαρτογράφησης (ORM) [33] που παρέχει για Rails εφαρμογές. Τα συστήματα διαχείρισης βάσεων δεδομένων (DBMS) [34] μπορούν να αποθηκεύουν και να χειρίζονται μόνο διαβαθμισμένες τιμές όπως ακέραιοι αριθμοί και συμβολοσειρές που είναι οργανωμένοι σε πίνακες. Η αντικειμενοστραφής χαρτογράφηση μετατρέπει τις τιμές των αντικειμένων σε ομάδες απλούστερων τιμών για αποθήκευση στη βάση δεδομένων (και τις μετατρέπει ξανά κατά την ανάκτηση). Η Ενεργή Εγγραφή ακολουθεί στενά το πρότυπο ORM, μετατρέποντας πίνακες σε κλάσεις/μοντέλα, σειρές πινάκων σε αντικείμενα και στήλες πινάκων σε χαρακτηριστικά αντικειμένων. Παρέχει ανεξαρτησία από το είδος της βάσης, βασική λειτουργικότητα CRUD (`create, read, update and delete`) [35] προηγμένες δυνατότητες εύρεσης και δυνατότητα αλληλεπίδρασης μεταξύ μοντέλων και άλλων υπηρεσιών. Τα αντικείμενα Ενεργής Εγγραφής δεν καθορίζουν άμεσα τα χαρακτηριστικά τους, αλλά αυτά συμπεραίνονται από τον ορισμό του πίνακα με τον οποίο συνδέονται. Η προσθήκη, η αφαίρεση και η αλλαγή χαρακτηριστικών γίνονται απευθείας στη βάση δεδομένων. Οποιαδήποτε αλλαγή αντικατοπτρίζεται άμεσα στα αντικείμενα της Εγγραφής.
- **Action Mailer** (`ActionMailer::Base`) : Η Ενέργεια Ταχυδρομείου είναι ένα framework για την κατασκευή υπηρεσιών ηλεκτρονικού ταχυδρομείου. Μπορεί να χρησιμοποιηθεί για την αποστολή μηνυμάτων βάσει ευέλικτων προτύπων ή για τη λήψη και επεξεργασία τους.

- **Active Resource** (ActiveResource::Base) : Τα αντικείμενα Ενεργής Πηγής αντιπροσωπεύουν τους REST [36] πόρους που μπορεί να διαχειριστεί η Ruby ως αντικείμενα. Για την αντιστοίχιση αυτή, η Ενεργή Πηγή χρειάζεται μόνο ένα όνομα κλάσης που αντιστοιχεί στο όνομα του πόρου (πχ, η κλάση User αντιστοιχεί στους πόρους των χρηστών, παρόμοια με την Ενεργή Εγγραφή) και μια τιμή τοποθεσίας που περιέχει το URL των πόρων αυτών.
- **Active Support** (ActiveSupport::Base) : Η Ενεργή Υποστήριξη είναι μια εκτεταμένη συλλογή χρήσιμων κλάσεων και τυπικών επεκτάσεων βιβλιοθηκών της γλώσσας Ruby που χρησιμοποιεί το framework Rails, τόσο στον κώδικα του πυρήνα όσο και στον κώδικα της εφαρμογής.

### 3.2. Δομή Καταλόγου

Το framework Rails προϋποθέτει μια συγκεκριμένη διάταξη του καταλόγου των αρχείων για την εκτέλεση μιας εφαρμογής. Παρακάτω παρατίθεται μια σύνοψη των καταλόγων ανώτατου επιπέδου που παράγονται κατά τη δημιουργία ενός Rails project.

Αρχείο/Κατάλογος	Σκοπός
<b>app/</b>	Περιέχει το βασικό κώδικα της εφαρμογής, περιλαμβανομένων των μοντέλων, όψεων, ελεγκτών και βοηθητικών κλάσεων.
<b>app/assets</b>	Περιλαμβάνει τα εργαλεία της εφαρμογής όπως τα αρχεία CSS, JavaScript και τις εικόνες.
<b>app/controllers</b>	Ο υπο-κατάλογος των ελεγκτών, όπου υπάρχουν οι κλάσεις τους.
<b>app/helpers</b>	Ο υπο-κατάλογος των βοηθητικών κλάσεων, οι οποίες δημιουργούνται για να βοηθούν τις κλάσεις των μοντέλων, όψεων, ελεγκτών.
<b>app/models</b>	Ο υπο-κατάλογος των μοντέλων, όπου υπάρχουν οι κλάσεις που μοντελοποιούν τα δεδομένα της βάσης δεδομένων της εφαρμογής.
<b>app/views</b>	Ο υπο-κατάλογος των όψεων, στον οποίο διατηρούνται τα πρότυπα προβολής που συμπληρώνονται με τα δεδομένα από την εφαρμογή, μετατρέπονται σε HTML και επιστρέφουν στο πρόγραμμα περιήγησης του χρήστη.
<b>bin/</b>	Διαδικά εκτελέσιμα αρχεία.



<b>config/</b>	Αυτός ο κατάλογος περιέχει τη μικρή ποσότητα κώδικα διαμόρφωσης που χρειάζεται μια εφαρμογή Rails, συμπεριλαμβανομένης της διαμόρφωσης της βάσης δεδομένων (στο αρχείο <code>database.yml</code> ), τη δόμηση του περιβάλλοντος (στο <code>environment.rb</code> ) και τη δρομολόγηση των εισερχόμενων αιτημάτων ιστού (στο <code>routes.rb</code> ).
<b>db/</b>	Αυτός ο κατάλογος περιέχει τα αρχεία της βάσης δεδομένων, όπως τα αρχεία συγχώνευσης της βάσης [37] (στον υπο-κατάλογο <code>db/migrate</code> ) και το σχήμα της βάσης [38] (στο <code>schema.rb</code> ).
<b>doc/</b>	Εδώ βρίσκονται τα αρχεία τεκμηρίωσης της εφαρμογής.
<b>lib/</b>	Αυτός ο κατάλογος περιέχει συγκεκριμένες βιβλιοθήκες εφαρμογών και κάθε είδους προσαρμοσμένο κώδικα που δεν ανήκει στους ελεγκτές, τα μοντέλα ή τις βοηθητικές κλάσεις.
<b>log/</b>	Τα αρχεία καταγραφής σφαλμάτων βρίσκονται εδώ. Το framework Rails δημιουργεί σενάρια ( <code>scripts</code> ) που βοηθούν στη διαχείριση διάφορων σφαλμάτων. Σε αυτό τον κατάλογο υπάρχουν τα ξεχωριστά αρχεία καταγραφής για το διακομιστή ( <code>server.log</code> ) και για κάθε ένα από τα τρία περιβάλλοντα Rails για τη δοκιμή ( <code>test.log</code> ), την ανάπτυξη ( <code>development.log</code> ) και την παραγωγή ( <code>production.log</code> ).
<b>public/</b>	Εδώ υπάρχουν δημόσια διαθέσιμα δεδομένα (πχ, μέσω του διακομιστή), όπως οι σελίδες σφάλματος.
<b>test/</b>	Τα αρχεία δοκιμής του δημιουργού και αυτά που δημιουργήθηκαν αυτόματα από το framework πηγαίνουν εδώ.
<b>tmp/</b>	Το Rails χρησιμοποιεί αυτόν τον κατάλογο για τη διατήρηση προσωρινών αρχείων για ενδιάμεση επεξεργασία.
<b>vendor/</b>	Οι βιβλιοθήκες που παρέχονται από τρίτους προμηθευτές (όπως βιβλιοθήκες ασφαλείας, βοηθητικά προγράμματα βάσης δεδομένων ή πρόσθετα [39] πέρα από αυτά της βασικής διανομής Rails) βρίσκονται εδώ.
<b>README.md</b>	Αυτό το αρχείο περιέχει μια σύντομη περιγραφή της εφαρμογής.
<b>Rakefile</b>	Αυτό το αρχείο είναι παρόμοιο με το <code>Makefile</code> [40] των Unix και βοηθά στην κατασκευή και δοκιμή του κώδικα Rails. Θα χρησιμοποιηθεί από το βοηθητικό πρόγραμμα <code>rake</code> [41] που παρέχεται με την εγκατάσταση της Ruby.

### 3.3. Μοντέλα

Το κάθε Μοντέλο είναι υπεύθυνο για τη διαχείριση των δεδομένων της εφαρμογής που το αφορούν. Αυτό, στο Rails framework, μεταφράζεται ως η αντιστοίχιση ενός Μοντέλου σε έναν πίνακα της Βάσης Δεδομένων. Με βάση την αρχή “Convention over Configuration” (CoC) που, όπως αναφέραμε παραπάνω, διέπει το framework, το Μοντέλο, ο πίνακας στη Βάσης και το αρχείο Ruby (File.rb) που περιγράφει το Μοντέλο, πρέπει να έχουν το ίδιο όνομα, το οποίο θα ξεκινά με κεφαλαίο γράμμα. Το αρχείο αυτό περιλαμβάνει επίσης τις συσχετίσεις με τα υπόλοιπα μοντέλα και κάποιες βοηθητικές συναρτήσεις, ώστε να μειωθεί ο όγκος των υπολογισμών του αντίστοιχου ελεγκτή. Όλα αυτά βρίσκονται αναλυτικά στο αρχείο /db/schema.rb της εφαρμογής, όμως παρακάτω παρατίθενται τα μοντέλα αυτά με κάποιες βασικές πληροφορίες:

- **User:**

Ο χρήστης αποτελεί τη βασική οντότητα ολόκληρης της εφαρμογής. Το μοντέλο αυτό περιλαμβάνει όλες τις λειτουργίες (business logic [42]) για την ταυτοποίηση του χρήστη με σκοπό είτε τη δημιουργία καινούριου, είτε την είσοδο ήδη υπάρχοντα, την αλλαγή των στοιχείων του ή την αναζήτηση άλλων χρηστών. Κάθε ένας μπορεί να ακολουθεί άλλους χρήστες για να παρακολουθεί τη δραστηριότητά τους.

- **Product:**

Αποτελεί το μοντέλο που αντιπροσωπεύει την οντότητα ενός προϊόντος. Διαθέτει λίγα και απλά πεδία για να διευκολύνει τη δημιουργία. Αυτά είναι τα: όνομα, περιεχόμενο και προαιρετική φωτογραφία. Κάθε χρήστης μπορεί να διαθέτει κανένα, ένα ή περισσότερα προϊόντα προς διαφήμιση.

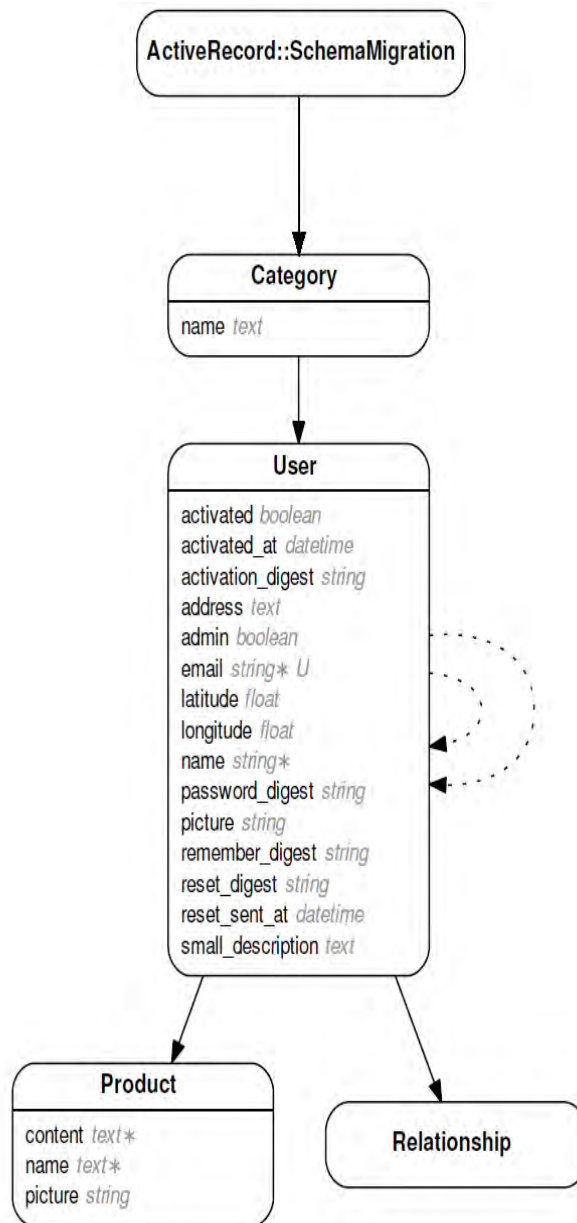
- **Category:**

Οι χρήστες έχουν διαχωριστεί σε κατηγορίες με βάση το αντικείμενο του καθενός. Για την εύκολη διαχείριση των κατηγοριών αυτών, έχει δημιουργηθεί ένα επιπλέον μοντέλο που να τις περιλαμβάνει.

- **Relationship:**

Για την ενέργεια της ακολούθησης ενός χρήστη από έναν άλλο δημιουργήθηκε και το μοντέλο των σχέσεων. Περιλαμβάνει τις ταυτότητες (αύξων αριθμός χρήστη – id) του χρήστη που ακολουθεί και αυτού που ακολουθείται για την καθιέρωση της μεταξύ τους σχέσης.

Στην επόμενη σελίδα παρουσιάζονται διαγραμματικά τα μοντέλα που περιγράφηκαν παραπάνω και οι μεταξύ τους συσχετίσεις.



Εικόνα 4. Το σχήμα της Βάσης Δεδομένων

### 3.4. Όψεις

Οι Όψεις αποτελούν το πιο απλό κομμάτι αλλά και ταυτόχρονα την πιο άμεση διεπαφή μεταξύ του χρήστη και της εφαρμογής. Εμφανίζουν πληροφορία στο χρήστη, αλλά και δέχονται από αυτόν δεδομένα. Η τεχνολογία που χρησιμοποιεί το Rails για αυτή τη διαδικασία είναι τα πρότυπα RHTML, δηλαδή ένα κράμα ERb και HTML για τη δημιουργία ιστοσελίδων. Σε γενικές γραμμές, η εφαρμογή χρησιμοποιεί ERb για την παρουσίαση δυναμικού περιεχομένου, όπως δεδομένα από τη βάση, φόρμες, πίνακες, κ.ά. και HTML για την παρουσίαση στατικού περιεχομένου. Κάθε όψη συνδέεται με έναν ελεγκτή και εμφανίζεται αμέσως μετά την εκτέλεση κάποιας ενέργειάς του. Για το λόγο αυτό και τα αρχεία των όψεων (με κατάληξη `.html.erb`) βρίσκονται στους αντίστοιχους υπο-καταλόγους των ελεγκτών.

Η προγραμματιστική φιλοσοφία DRY που αναφέραμε προηγουμένως, βρίσκει άμεση εφαρμογή κατά τη δημιουργία των όψεων στην τεχνολογία των `partials`. Τα `partials` είναι αυτόνομα αρχεία όψεων, τα οποία εκτελούν μια συγκεκριμένη εργασία και μπορούν να επαναχρησιμοποιηθούν σε πολλαπλές όψεις, χωρίς να ξαναγράφεται ο ίδιος κώδικας. Έτσι, το σύνολο του εμφανιζόμενου κώδικα μικραίνει, ώστε να διευκολύνεται η διαχείρισή του μέσω της οργάνωσης σε μικρότερα τμήματα. Τα ονόματα αυτών των αρχείων συνήθως ξεκινούν με το σύμβολο “\_” (κάτω παύλα) και έχουν κατάληξη `.html.erb`, όπως αυτά των όψεων. Η διαδικασία κλήσης αρχείων `partials` ονομάζεται απόδοση (`rendering`) και για το λόγο αυτό η εντολή κλήσης τους είναι η “`render`”.

Οι υπο-κατάλογοι των όψεων και τα αρχεία που περιέχονται σε αυτούς συνοπτικά έχουν ως εξής:

- **Categories**
  - `_form.html.erb`: φόρμα για τη δημιουργία νέας κατηγορίας για τους χρήστες
  - `edit.html.erb`: επεξεργασία κατηγορίας
  - `index.html.erb`: εμφάνιση όλων των κατηγοριών
  - `new.html.erb`: δημιουργία νέας κατηγορίας
  - `show.html.erb`: εμφάνιση όλων των στοιχείων μιας κατηγορίας
- **Password\_resets**
  - `edit.html.erb`: αλλαγή κωδικού πρόσβασης χρήστη εφόσον είναι συνδεδεμένος
  - `new.html.erb`: αίτημα για αλλαγή κωδικού πρόσβασης όταν ο χρήστης δεν μπορεί να αποκτήσει πρόσβαση στο λογαριασμό του

- **Products**
  - `_product.html.erb`: partial για την εμφάνιση των στοιχείων ενός προϊόντος
  - `compare.html.erb`: σύγκριση επιλεγμένων προϊόντων από την αναζήτηση
  - `index.html.erb`: αναζήτηση προϊόντων
- **Static\_pages**
  - `about.html.erb`: πληροφορίες σχετικά με την εφαρμογή
  - `help.html.erb`: βοήθεια σχετικά με τη χρήση της πλατφόρμας
  - `home.html.erb`: κεντρική σελίδα που περιέχει τα κουμπιά σύνδεσης και εγγραφής
- **User\_mailer**
  - `account_activation.html.erb`: πρότυπο για το email ενεργοποίησης λογαριασμού σε HTML
  - `account_activation.text.erb`: πρότυπο για το email ενεργοποίησης λογαριασμού σε μορφή απλού κειμένου
  - `password_reset.html.erb`: πρότυπο για το email αλλαγής κωδικού πρόσβασης σε HTML
  - `password_reset.text.erb`: πρότυπο για το email αλλαγής κωδικού πρόσβασης μορφή απλού κειμένου
- **Users**
  - `edit.html.erb`: επεξεργασία στοιχείων χρήστη
  - `index.html.erb`: αναζήτηση χρηστών
  - `new.html.erb`: δημιουργία νέου λογαριασμού
  - `show.html.erb`: εμφάνιση προφίλ χρήστη
  - `show_follow.html.erb`: εμφάνιση χρηστών που ακολουθεί ο χρήστης και αυτούς που τον ακολουθούν
- **Shared**

Αποτελεί μια σύμβαση του Rails να χρησιμοποιείται ένας ειδικός κοινόχρηστος κατάλογος για `partials` που αναμένεται να χρησιμοποιηθούν σε όψεις για πολλούς ελεγκτές.

  - `_error_messages.html.erb`: μηνύματα σφάλματος
  - `_product_form.html.erb`: δημιουργία καινούριου προϊόντος
  - `_user_info.html.erb`: συνοπτικές πληροφορίες χρήστη

- **Sessions**

Το HTML είναι ένα stateless protocol [43], που σημαίνει ότι αντιμετωπίζει κάθε αίτημα ως ανεξάρτητη συναλλαγή που δεν μπορεί να χρησιμοποιήσει πληροφορίες από προηγούμενα αιτήματα. Αυτό σημαίνει ότι δεν υπάρχει κανένας τρόπος να θυμάται την ταυτότητα ενός χρήστη από σελίδα σε σελίδα. Αντίθετα, οι εφαρμογές ιστού που απαιτούν σύνδεση χρήστη (login) πρέπει να χρησιμοποιούν μια περίοδο σύνδεσης (session), η οποία είναι μια ημι-μόνιμη σύνδεση μεταξύ δύο υπολογιστών (όπως ένας υπολογιστής-πελάτης που εκτελεί ένα πρόγραμμα περιήγησης ιστού και ένας διακομιστής που χρησιμοποιεί Rails). Ο πιο κοινός τρόπος υλοποίησης περιόδων σύνδεσης σε Rails είναι με τη χρήση cookies [44], τα οποία είναι μικρά κομμάτια κειμένου που τοποθετούνται στο πρόγραμμα περιήγησης του χρήστη και διατηρούνται από τη μία σελίδα στην επόμενη, διατηρώντας τις απαραίτητες πληροφορίες.

- new.html.erb: δημιουργία νέας περιόδου σύνδεσης κατά τη σύνδεση του χρήστη

- **Layouts**

Αυτός ο κατάλογος δημιουργείται αυτόματα από το Rails για τον γενικό ελεγκτή της εφαρμογής. Τα αρχεία που βρίσκονται εδώ ορίζουν το περιβάλλον μιας σελίδας HTML, δηλαδή μια κοινή εμφάνιση και αίσθηση της τελικής εφαρμογής. Υπάρχουν αρχεία που χρησιμοποιούνται από προεπιλογή και δημιουργούνται αυτόματα, αλλά μπορούμε να κατασκευάσουμε και άλλα πιο εξειδικευμένα, αρκεί να γνωρίζει ο αντίστοιχος ελεγκτής ότι θέλουμε να τα χρησιμοποιήσουμε.

- \_footer.html.erb: υποσέλιδο με κουμπιά προς χρήσιμες σελίδες εντός και εκτός εφαρμογής
- \_header.html.erb: κεφαλίδα με κουμπιά προς όλες τις σελίδες εντός της εφαρμογής, ανάλογα με το αν ο χρήστης είναι συνδεδεμένος
- \_shim.html.erb: αρχείο με μετα-δεδομένα για τις σελίδες της εφαρμογής
- new.html.erb: πρότυπο για τη σελίδα της εγγραφής χρήστη
- user.html.erb: πρότυπο για τις σελίδες που εμπλέκονται δεδομένα χρηστών

### 3.5. Ελεγκτές

Το επίπεδο των ελεγκτών στο MVC μοντέλο χρησιμοποιείται για τη λογική ελέγχου της εφαρμογής. Λειτουργεί τόσο με το επίπεδο των μοντέλων όσο και με το επίπεδο των όψεων. Για το μεν πρώτο, ο ελεγκτής θα χρησιμοποιήσει τα αντικείμενα του μοντέλου για να εκτελέσει τις διαδικασίες της business logic. Για το δε δεύτερο, είναι υπεύθυνος για την επιλογή της επόμενης όψης που θα προβληθεί στο χρήστη. Η εφαρμογή χρησιμοποιεί το αρχιτεκτονικό πρότυπο REST, όπου κάθε πόρος αντιστοιχεί σε ένα μοναδικό URL. Επίσης, οι λειτουργίες δημιουργίας, ανάγνωσης, ανανέωσης και διαγραφής δεδομένων (CRUD) υλοποιούνται μέσω των HTTP μεθόδων GET, POST, PUT και DELETE.

Στη γενική περίπτωση, η εφαρμογή δέχεται ένα αίτημα από ένα πρόγραμμα περιήγησης, το επεξεργάζεται, και στέλνει μια απόκριση. Στο framework Rails, ο ελεγκτής λειτουργεί ως ενιαία διεπαφή για όλες τις εισερχόμενες HTTP αιτήσεις. Οι πληροφορίες κωδικοποιούνται στη διεύθυνση URL του αιτήματος, και ένα υποσύστημα που ονομάζεται "δρομολόγηση" χρησιμοποιείται για να καθορίσει τι πρέπει να γίνει με αυτό το αίτημα. Η εφαρμογή καθορίζει το όνομα του ελεγκτή που χειρίζεται αυτό το συγκεκριμένο αίτημα, μαζί με μια λίστα από τις υπόλοιπες παραμέτρους του. Τυπικά, μία από αυτές τις πρόσθετες παραμέτρους προσδιορίζει την ενέργεια που πρέπει να κληθεί από τον ελεγκτή. Μόλις αναγνωριστεί ο ελεγκτής, οι παράμετροι της αίτησης περνούν στην αντίστοιχη μέθοδο με σκοπό τη δημιουργία του αντικειμένου της απόκρισης. Στη συνέχεια, ο ελεγκτής καλεί τη μέθοδο, η οποία εκτελεί την επεξεργασία της business logic, ενημερώνει τα αντίστοιχα αντικείμενα και αποφασίζει ποια όψη θα επιστραφεί για προβολή στον περιηγητή.

Το αρχείο `config/routes.rb` περιέχει όλες τις πληροφορίες δρομολόγησης της εφαρμογής. Πρακτικά, σχεδιάζει ένα χάρτη που συνδέει τις εξωτερικές URL με τα εσωτερικά στοιχεία της εφαρμογής. Η επεξεργασία του αρχείου γίνεται από πάνω προς τα κάτω όταν εισέρχεται μια νέα αίτηση, την οποία αποστέλλει στην πρώτη διαδρομή που αντιστοιχεί. Εάν δε βρεθεί διαδρομή που να αντιστοιχεί, επιστρέφεται ένα HTTP μήνυμα λάθους 404 – Not found.

Στον πίνακα που βρίσκεται παρακάτω υπάρχουν οι διαδρομές που χρησιμοποιήθηκαν για την εφαρμογή μαζί με τον αντίστοιχο ελεγκτή και την ενέργεια που εκτελείται. Οι χρήσεις των ενεργειών περιγράφηκαν παραπάνω, στις αντίστοιχες όψεις. Στο αρχείο `routes.rb`, υπάρχουν ρητές δηλώσεις δρομολόγησης, με τη μορφή που παρουσιάζονται στον πίνακα. Για πόρους, όμως, που περιλαμβάνουν και τις τέσσερις HTTP μεθόδους, και μάλιστα για πολλές εμφανίσεις (instances), όπως οι χρήστες, χρησιμοποιούμε την εντολή "resources", που δημιουργεί αυτόματα τις δρομολογήσεις. Η εντολή αυτή (πχ "**resources :users**") δεν προσθέτει απλά αυτόματα την αντίστοιχη διεύθυνση URL, αλλά παρέχει στην εφαρμογή όλες τις ενέργειες που απαιτούνται για να γίνει ο πόρος (πχ Users) RESTful, μαζί με έναν μεγάλο αριθμό ονοματικών δρομολογήσεων (named routes) για τη δημιουργία των αντίστοιχων URL για τους χρήστες. Αυτές οι ονοματικές δρομολογήσεις δημιουργούνται ως εξής: "**action\_controller\_path**".

HTTP Μέθοδος	URL	Controller#Action
GET	password_resets/new	password_resets#new
GET	password_resets/edit	password_resets#edit
GET	sessions/new	sessions#new
GET	/help	static_pages#home
GET	/about	static_pages#about
GET	/search	users#index
GET	/products	products#index
GET	/signup	users#new
GET	/login	sessions#new
POST	/signup	users#create
POST	/login	sessions#create
DELETE	/logout	sessions#destroy
POST	/compare	products#compare
GET	/users/new	users#new
GET	/users/:id	users#show
GET	/users/:id/edit	users#edit
PATCH/PUT	/users/:id	users#update
DELETE	/users/:id	users#destroy
GET	/users/:id/following	users#following
GET	/users/:id/followers	users#followers
POST	/products	products#create
GET	/products/:id	products#show
DELETE	/products/:id	products#destroy
GET	/account_activations/:id	account_activations#edit
POST	/relationships	relationships#create
DELETE	/relationships/:id	relationships#destroy

Πίνακας 1. Πίνακας Δρομολόγησης της Εφαρμογής



Να σημειωθεί εδώ ότι έχουν εσκεμμένα διαγραφεί οι δρομολογήσεις για τον πόρων των Κατηγοριών, ώστε να μην έχουν πρόσβαση οι χρήστες σε αυτές. Κατά την υλοποίηση της εφαρμογής δημιουργήθηκαν οι παρακάτω ελεγκτές:

- **account\_activations\_controller**

Ο ελεγκτής αυτός είναι υπεύθυνος για τη διαδικασία ενεργοποίησης καινούριων λογαριασμών για τους χρήστες της εφαρμογής. Πραγματοποιεί αναζήτηση στη βάση δεδομένων να εντοπίσει το χρήστη και τον ανακατευθύνει στην αντίστοιχη σελίδα ανάλογα με το αν έχει ενεργοποιηθεί ο λογαριασμός του ή όχι.

- **application\_controller**

Ο ελεγκτής της εφαρμογής δημιουργείται αυτόματα κατά τη δημιουργία της εφαρμογής. Είναι υπεύθυνος για την προστασία από κακόβουλες επιθέσεις όπως η Cross-Site Request Forgery (CSRF) [45] καθώς και για την επιβεβαίωση ότι ο χρήστης που προσπαθεί να πραγματοποιήσει μια ενέργεια έχει συνδεθεί στην εφαρμογή.

- **categories\_controller**

Με αυτό τον ελεγκτή μπορούμε να δημιουργήσουμε, να προβάλλουμε και να διαγράψουμε τις κατηγορίες στις οποίες εντάσσονται οι χρήστες. Όπως αναφέρθηκε και προηγουμένως όμως, μετά την αρχική δημιουργία των βασικών κατηγοριών, απενεργοποιήθηκε η δρομολόγηση προς τις αντίστοιχες όψεις. Πλέον, όλες αυτές οι λειτουργίες πραγματοποιούνται μέσω των εργαλείων από τη διαχειριστή.

- **password\_resets\_controller**

Αυτός ο ελεγκτής είναι υπεύθυνος για τις αιτήσεις αλλαγής κωδικού πρόσβασης των χρηστών σε δύο περιπτώσεις. Είτε όταν ο χρήστης είναι συνδεδεμένος και επιθυμεί να αλλάξει τον κωδικό του, είτε όταν επιθυμεί αυτή την αλλαγή διότι δε μπορεί να συνδεθεί. Ο ελεγκτής δημιουργεί προσωρινά link τα οποία αποστέλλει στο χρήστη μέσω email, τα οποία οδηγούν στην αλλαγή αυτή. Όταν ο χρήστης ακολουθήσει ένα τέτοιο link, ο ελεγκτής επιβεβαιώνει τα στοιχεία και ολοκληρώνει τη διαδικασία.

- **products\_controller**

Τη δημιουργία, προβολή και διαγραφή προϊόντων από τους χρήστες αναλαμβάνει ο ελεγκτής. Για να πραγματοποιηθούν οι διαδικασίες της δημιουργίας και της διαγραφής πρέπει να επιβεβαιώσει ότι ο χρήστης έχει τα αντίστοιχα δικαιώματα. Επίσης αναλαμβάνει και την αναζήτηση προϊόντων κάνοντας χρήση του gem ransack.

- **relationships\_controller**

Εδώ επιβεβαιώνονται και αποθηκεύονται οι διαδικασίες για την ακολούθηση ενός χρήστη από έναν άλλο. Αφού γίνει αναζήτηση του χρήστη, προστίθενται τα αντίστοιχα δεδομένα στη βάση. Επίσης ένας χρήστης μπορεί να σταματήσει να ακολουθεί κάποιον, οπότε η διαδικασία περνά ξανά από εδώ.

- **sessions\_controller**

Κατά την εγγραφή ή την είσοδο του χρήστη στην εφαρμογή, ο ελεγκτής δημιουργεί μία καινούρια περίοδο σύνδεσης, όπως περιγράφηκε παραπάνω. Πρέπει να αναζητήσει το χρήστη με βάση τις παραμέτρους και να κάνει τους απαραίτητους ελέγχους για την ομαλή χρήση της εφαρμογής.

- **static\_pages\_controller**

Ο ελεγκτής αυτός δεν έχει κάποια ιδιαίτερη ενέργεια να εκτελέσει, εκτός από την ανακατεύθυνση προς τις σελίδες **“home”**, **“help”** και **“about”**.

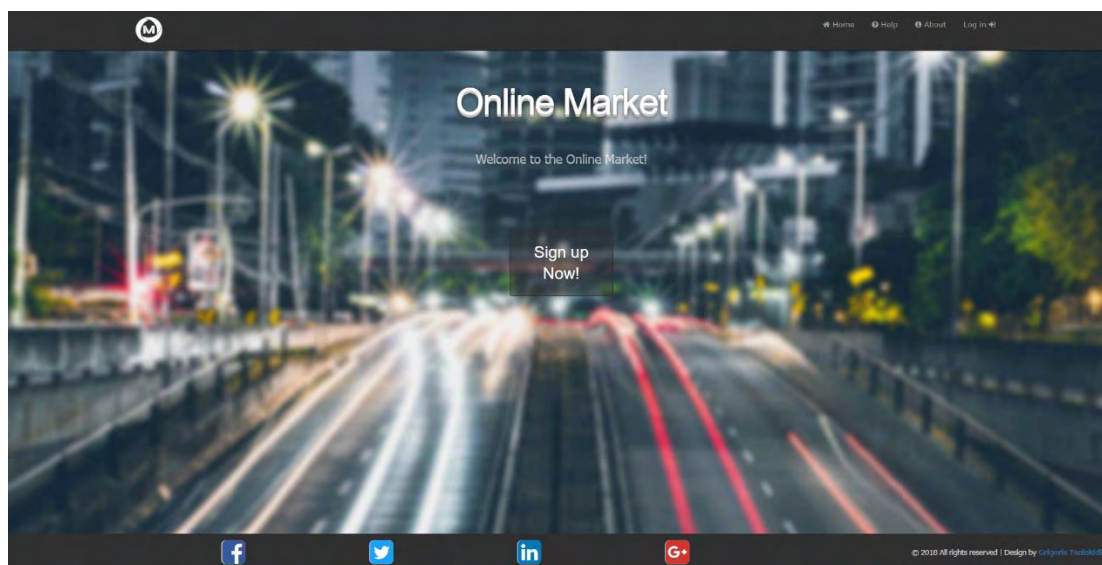
- **users\_controller**

Για τη συγκεκριμένη εφαρμογή αυτός είναι ο πιο σημαντικός ελεγκτής, με την έννοια ότι διαχειρίζεται το μεγαλύτερο μέρος των πόρων και ενεργειών. Με βάση αυτόν, ο χρήστης μπορεί να δημιουργεί, να επεξεργάζεται, να προβάλλει και να διαγράφει το ίδιο το προφίλ του ή λεπτομέρειες αυτού. Επίσης διαχειρίζεται την αναζήτηση των χρηστών και την εμφάνιση των στοιχείων και λεπτομερειών που χρειάζεται.

## Κεφάλαιο 4<sup>ο</sup>: Χαρακτηριστικά της εφαρμογής

### 4.1. Λειτουργίες Εφαρμογής

Κατά την είσοδο του χρήστη στην εφαρμογή εμφανίζεται η κεντρική οθόνη, από την οποία μπορεί να πραγματοποιήσει τη σύνδεση ή εγγραφή και είναι:

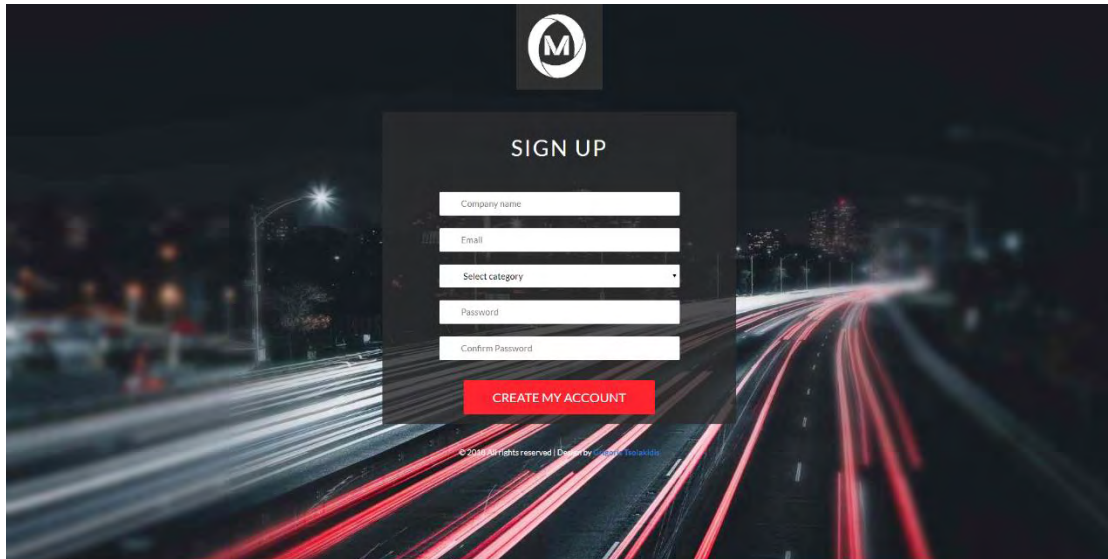


Εικόνα 5. Κεντρική σελίδα

#### 4.1.1. Εγγραφή Χρήστη

Για την καταχώρηση προϊόντων και την επικοινωνία με τους ενδιαφερόμενους είναι απαραίτητη η ύπαρξη λογαριασμού στην εφαρμογή. Για την εγγραφή απαιτείται μόλις ένας λογαριασμός email. Όπως φαίνεται παραπάνω στην κεντρική σελίδα, το βασικό κουμπί αποτελεί σύνδεσμος προς τη σελίδα της εγγραφής ενός νέου χρήστη.

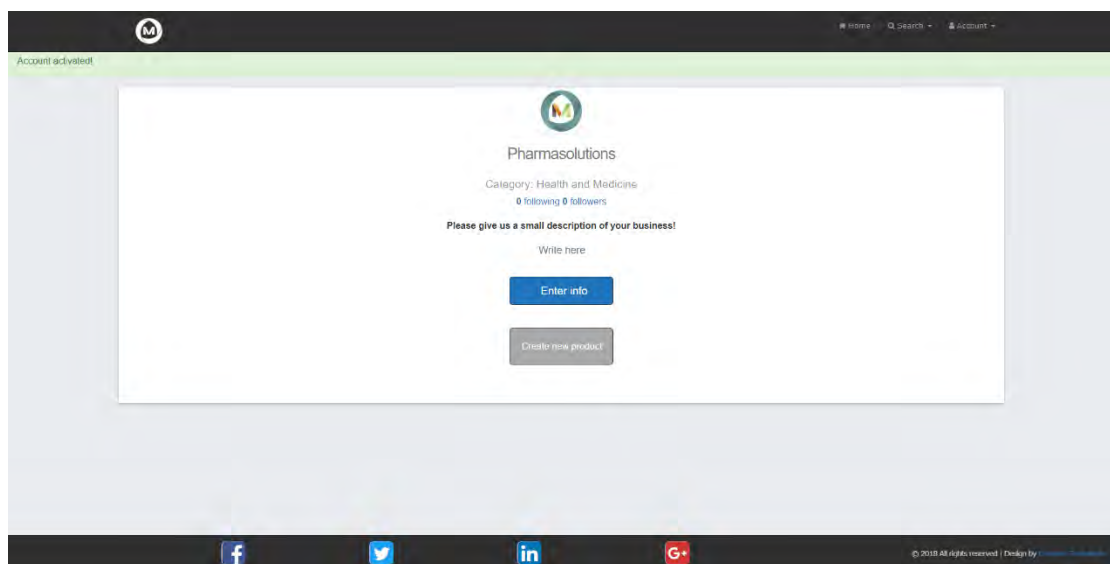
Πατώντας το κουμπί "Sign up Now!", ο χρήστης μεταβαίνει στη σελίδα εγγραφής, στην οποία καλείται να καταχωρήσει κάποια βασικά στοιχεία, όπως το όνομα της εταιρίας, ένα λογαριασμό email, έναν κωδικό πρόσβασης και προαιρετικά μια κατηγορία στην οποία πιστεύει πως εντάσσεται η επιχείρησή του:



Εικόνα 6. Σελίδα εγγραφής νέου χρήστη

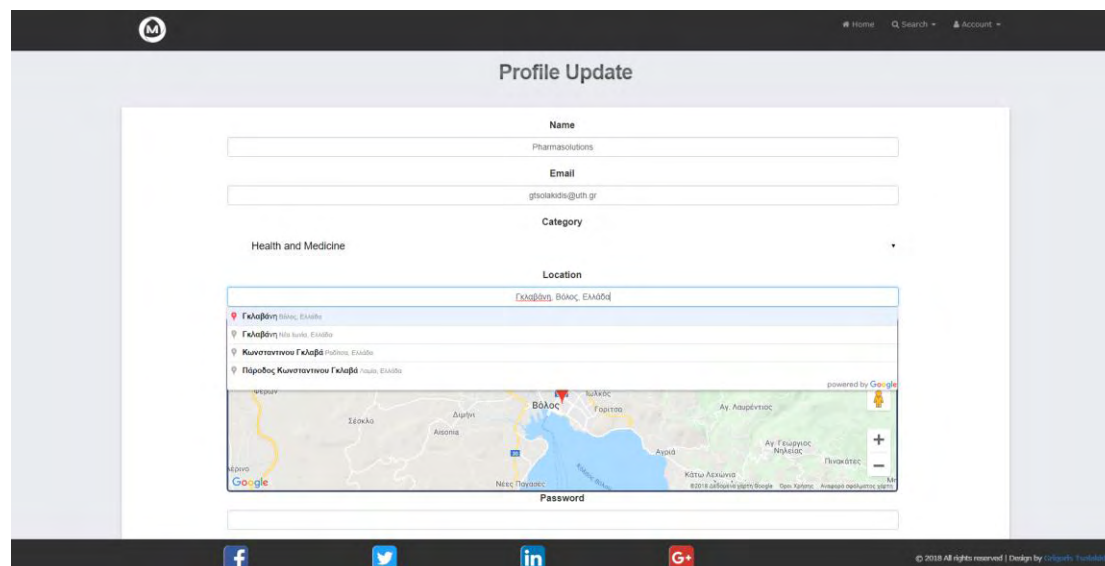
#### 4.1.2. Προβολή Προφίλ Χρήστη

Μετά την επιτυχή εγγραφή του χρήστη στην εφαρμογή και τη σύνδεσή του με τα στοιχεία του, ανακατευθύνεται στην προσωπική σελίδα του προφίλ του. Εκεί, τον περιμένει ένα ειδοποιητικό μήνυμα ότι ο λογαριασμός του έχει ενεργοποιηθεί.



Εικόνα 7. Προφίλ μετά την ενεργοποίηση του λογαριασμού

Εάν ακολουθήσει ο χρήστης τις διακριτικές οδηγίες που υπάρχουν στο προφίλ του, μπορεί να συμπληρώσει πληροφορίες, όπως η περιγραφή της επιχείρησης, η διεύθυνσή της, αλλά και να αλλάξει άλλες όπως το όνομα, η κατηγορία, κτλ. Κάποια από αυτά μπορεί να τα πραγματοποιήσει στη σελίδα του προφίλ, ενώ τα υπόλοιπα μέσω της σελίδας των ρυθμίσεων του λογαριασμού του. Ενδεικτικά παρατίθεται ένα παράδειγμα αλλαγής τοποθεσίας, όπου φαίνεται η αυτόματη συμπλήρωση και ο χάρτης από το Google Maps API.



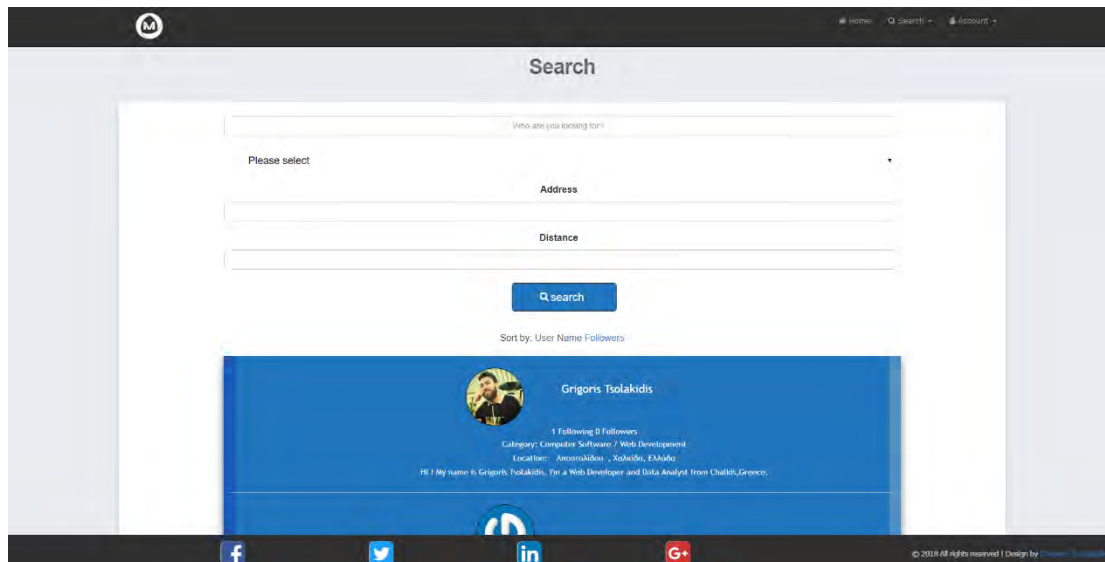
The image shows a 'Profile Update' form with the following fields and content:

- Name:** Pharnasolutions
- Email:** gtsiakidis@uth.gr
- Category:** Health and Medicine
- Location:** Ελευθέριος Βόλος, Ελλάδα. Below this, a Google Maps interface is shown with a list of suggestions:
  - Γελοβόνη Βόλος, Ελλάδα
  - Γελοβόνη Νέα Ιωνία, Ελλάδα
  - Κωνσταντίνου Γελοβόνη Βόλος, Ελλάδα
  - Πύραρος Κωνσταντίνου Γελοβόνη Νέα Ιωνία, ΕλλάδαThe map below shows the area around Βόλος, with a red pin indicating the selected location.
- Password:** (Empty field)

Εικόνα 8. Εισαγωγή τοποθεσίας προφίλ μέσω ρυθμίσεων

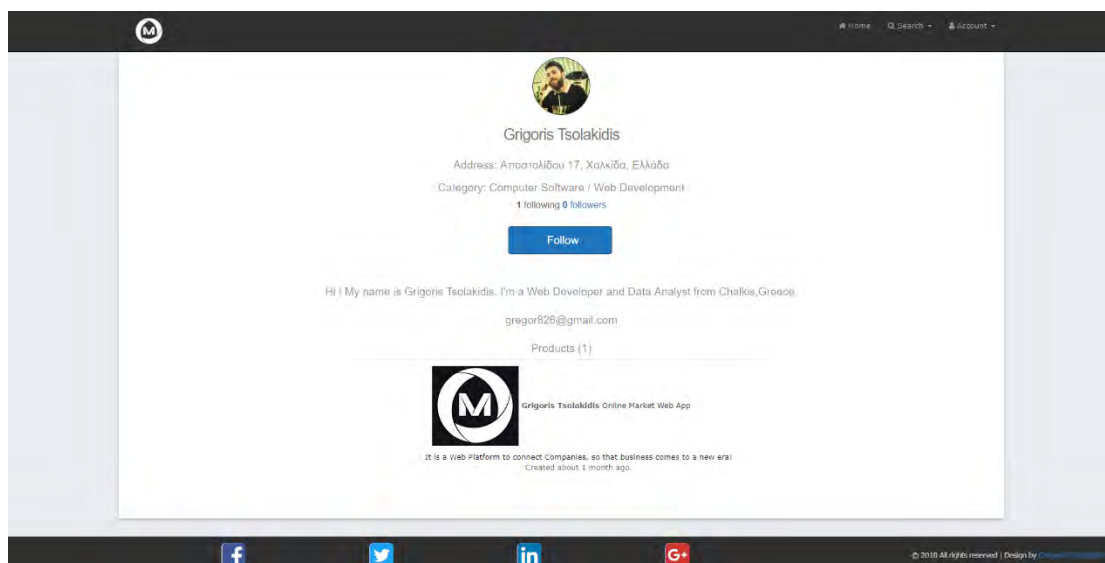
#### 4.1.3. Αναζήτηση και Ακολούθηση Χρηστών

Στην κεφαλίδα της σελίδας υπάρχει πλέον η Ενότητα της Αναζήτησης. Σε αυτό το σημείο θα ασχοληθούμε με την αναζήτηση χρηστών. Κατά τη μετάβαση του χρήστη στη σελίδα, εμφανίζεται η φόρμα της αναζήτησης καθώς και όλοι οι χρήστες της εφαρμογής από προεπιλογή. Η αναζήτηση μπορεί να γίνει με βάση το όνομα μιας επιχείρησης, την κατηγορία στην οποία ανήκει αλλά και την απόσταση από μια συγκεκριμένη διεύθυνση. Εάν δεν δοθεί διεύθυνση από τον χρήστη, επιλέγεται αυτόματα αυτή που έχει ορίσει για την επιχείρησή του. Εάν δεν έχει οριστεί, η προεπιλογή διεύθυνσης είναι προς το παρόν η Αθήνα. Στο σημείο που εμφανίζονται όλοι οι χρήστες, θα βρίσκονται τα αποτελέσματα της αναζήτησης.



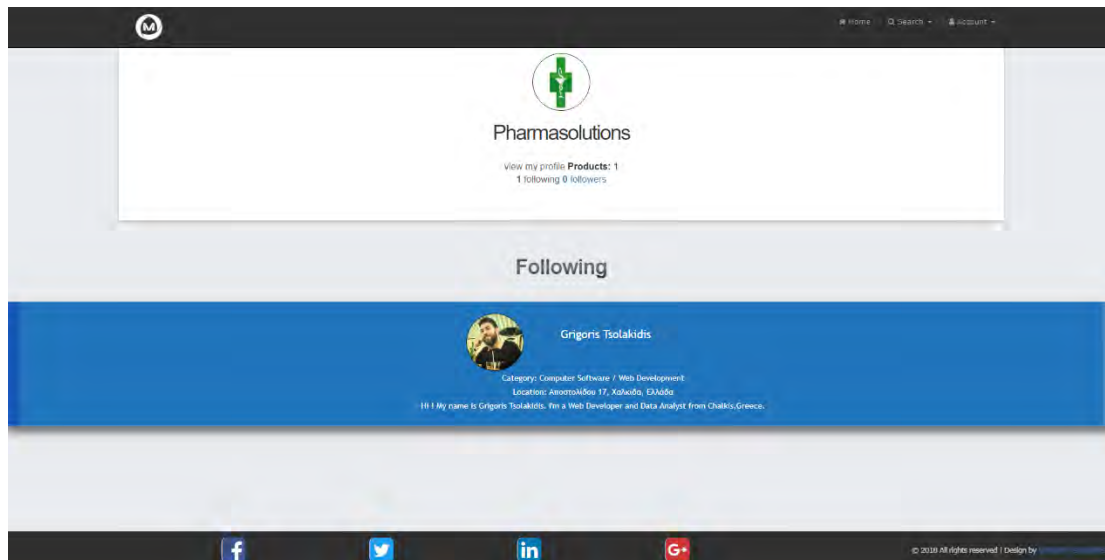
Εικόνα 9. Αναζήτηση Χρηστών

Όλοι οι χρήστες έχουν τη δυνατότητα να προβάλλουν τα προφίλ όλων των υπολοίπων, όπως το δικό τους. Κατά τη μετάβαση σε ξένο προφίλ εμφανίζεται το κουμπί της ακολούθησης, για τη δημιουργία της σχέσης μεταξύ των χρηστών.



Εικόνα 10. Προβολή προφίλ διαφορετικού λογαριασμού

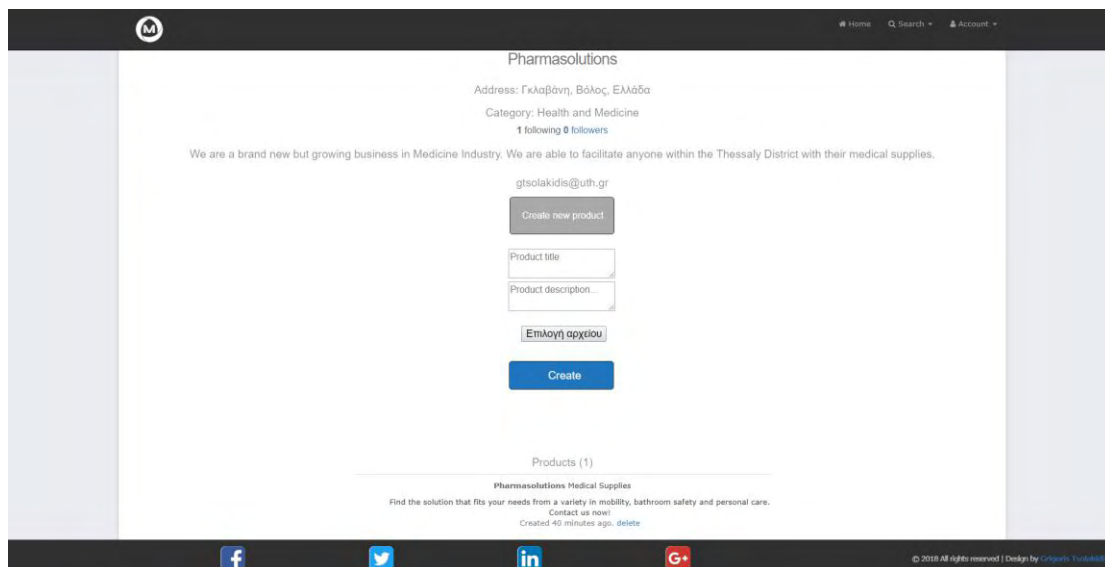
Μετά τη διαδικασία αυτή, ενημερώνονται οι σελίδες και πλέον μπορεί ο κάθε χρήστης να δει ποιους ακολουθεί και ποιοι τον ακολουθούν, μέσα από το προφίλ του.



Εικόνα 11. Προβολή χρηστών που ακολουθεί ο χρήστης

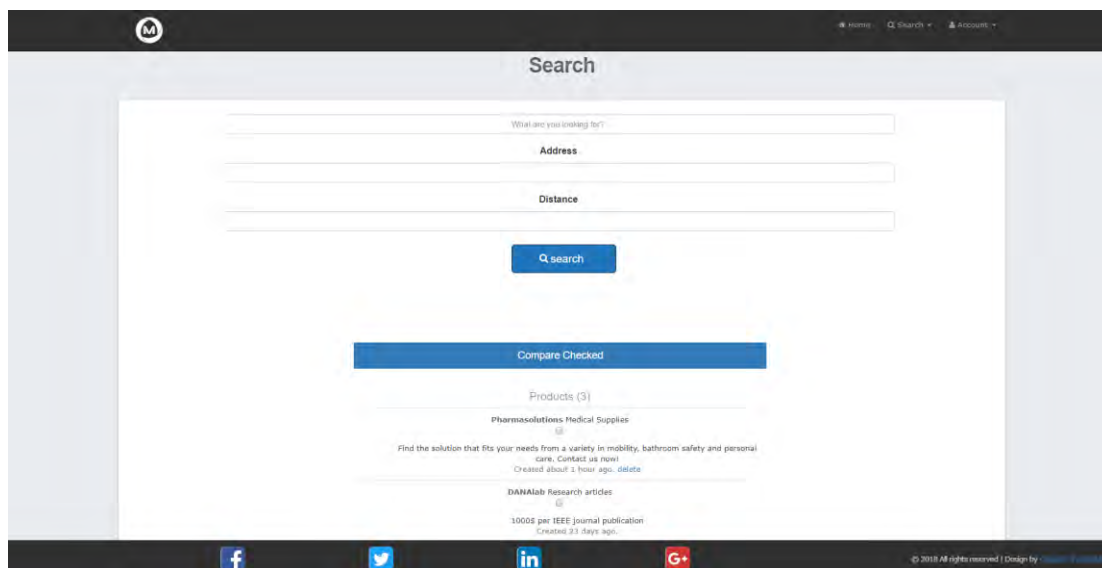
#### 4.1.4. Καταχώρηση και Αναζήτηση/Σύγκριση Προϊόντων

Κάτω από τις πληροφορίες του χρήστη, εμφανίζεται μια λίστα με τα προϊόντα που έχει καταχωρήσει κάθε χρήστης. Όταν ο χρήστης μεταβαίνει στο προσωπικό του προφίλ, εμφανίζεται ένα κουμπί για τη δημιουργία νέας καταχώρησης προϊόντος. Η καταχώρηση περιλαμβάνει υποχρεωτικό έναν τίτλο και μία σύντομη περιγραφή και προαιρετικά μία φωτογραφία.



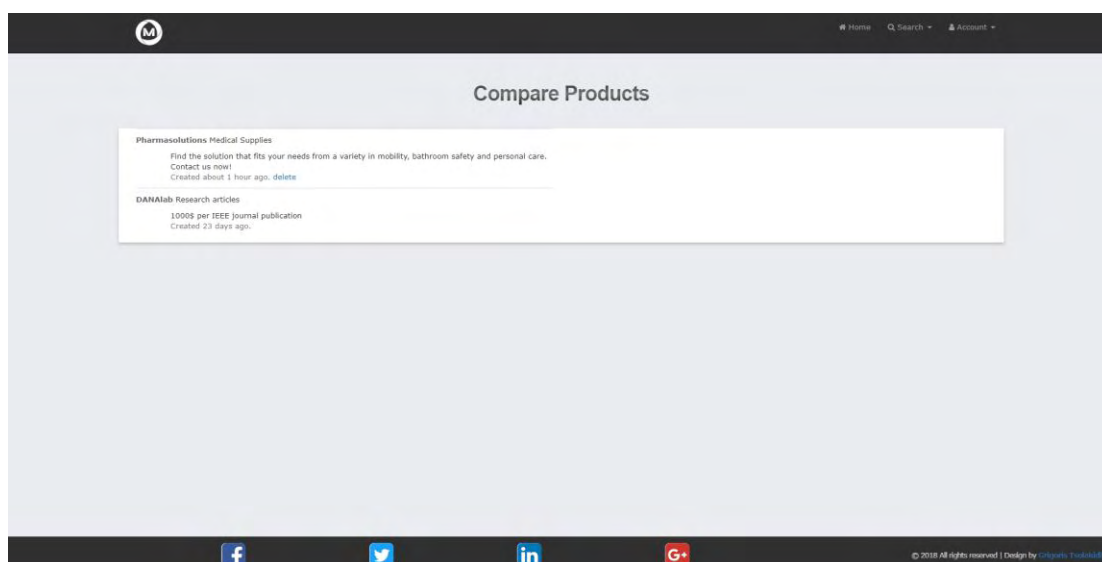
Εικόνα 12. Φόρμα και καταχώρηση νέου προϊόντος

Όπως και για τους χρήστες, έτσι και για τα προϊόντα, υπάρχει η ανάλογη ενότητα αναζήτησης στην κεφαλίδα. Η αναζήτηση προϊόντων λειτουργεί όπως ακριβώς η αντίστοιχη για τους χρήστες, και αναζητά λέξεις-κλειδιά τόσο στον τίτλο όσο και στην περιγραφή.



Εικόνα 13. Αναζήτηση προϊόντων

Εδώ βέβαια δίνεται επιπλέον η δυνατότητα να επιλέξει ο χρήστης όσα προϊόντα επιθυμεί και να τα εμφανίσει σε νέα σελίδα για ανάλυση και σύγκριση. Με τον τρόπο αυτό πετυχαίνουμε μεγαλύτερη εξατομίκευση στην αναζήτηση, ώστε να βοηθήσουμε το χρήστη να επιλέξει αυτό που πραγματικά επιθυμεί.



Εικόνα 14. Επιλεγμένη σύγκριση προϊόντων μέσα από την αναζήτηση του χρήστη



## Κεφάλαιο 5<sup>ο</sup>: Επισκόπηση

### 5.1. Παραδοχές κατά την υλοποίηση

Κατά τη μελέτη και έρευνα πριν την έναρξη της υλοποίησης, αλλά και κατά τη διάρκεια της ανάπτυξης της εφαρμογής έγιναν κάποιες παραδοχές οι οποίες οδήγησαν στην υλοποίηση ή μη κάποιων χαρακτηριστικών. Το πρώτο από αυτά ήταν η δυνατότητα σύνδεσης του χρήστη μέσω social media (πχ facebook, google plus, κτλ). Αποφασίστηκε να μην υλοποιηθεί διότι η εφαρμογή απευθύνεται σε επιχειρήσεις, οι οποίες σε μεγάλο βαθμό δε διατηρούν εταιρικά προφίλ σε social media, οπότε δεν προσφέρει κάποια επιπλέον χρησιμότητα. Αντίθετα, εάν η εφαρμογή μπορούσε να συλλέξει δεδομένα και κάποιος δημιουργούσε λογαριασμό χρησιμοποιώντας προσωπικό προφίλ από μέσα κοινωνικής δικτύωσης, τα δεδομένα αυτά θα ήταν εσφαλμένα και θα δυσκόλευαν περισσότερο την κατάσταση. Επίσης, δεν προσφέρεται στους χρήστες πεδίο τιμής στην καταχώρηση των προϊόντων. Αυτό συνέβη διότι το αντικείμενο της εφαρμογής δεν είναι να διαχειρίζεται τις συναλλαγές, αλλά στοχεύει στην επικοινωνία μεταξύ επιχειρήσεων, μέσα από την οποία θα διαπραγματεύεται η τιμή του εκάστοτε προϊόντος ή υπηρεσίας. Επίσης, εάν υπήρχε τιμολόγηση των προϊόντων, αυτό ίσως λειτουργούσε ως αποτρεπτικός παράγοντας για ορισμένες επιχειρήσεις από το να τα καταχωρήσουν ή και ακόμα από το να χρησιμοποιήσουν την ίδια την εφαρμογή. Τέλος, επιλέχθηκε να μην υπάρχουν χρήστες με δικαιώματα διαχείρισης σε αυτό το στάδιο, καθώς η διαχείριση ολόκληρης της εφαρμογής γίνεται από το δημιουργό με τη χρήση πολλών εξειδικευμένων εργαλείων λογισμικού.

### 5.2. Ιδέες για Μελλοντικές Επεκτάσεις

Η εφαρμογή διαθέτει μια βασική λειτουργικότητα για να αποτελέσει ένα χρήσιμο εργαλείο αναζήτησης επαγγελματικών επαφών, κατά κύριο λόγο στον κλάδο των μικρομεσαίων επιχειρήσεων. Η ανάπτυξη της βρίσκεται σε πρώιμο στάδιο και η επιλογή του frameworks Rails διευκολύνει την περαιτέρω υλοποίηση επεκτάσεων που θα την έκαναν ακόμα περισσότερο παραμετροποιήσιμη και χρηστική.

Στη συνέχεια παρουσιάζονται ορισμένες ιδέες για μελλοντικές επεκτάσεις που θα μπορούσαν να διευρύνουν τις λειτουργίες της εφαρμογής.

- Δημιουργία και διαχείριση συζητήσεων-μηνυμάτων με άλλους χρήστες, με σκοπό την καλύτερη και αμεσότερη επικοινωνία μεταξύ τους.
- Δυνατότητα προσθήκης εξειδικευμένων κατηγοριών από τους ίδιους τους χρήστες, με σκοπό την πιο εξειδικευμένη περιγραφή των προϊόντων και υπηρεσιών τους.
- Δυνατότητα ελέγχου και επιβεβαίωσης λογαριασμών χρηστών μέσω στοιχείων, όπως το ΑΦΜ (Αριθμός Φορολογικού Μητρώου) της επιχείρησης.
- Αποθήκευση πληροφοριών χρήσης της πλατφόρμας για την ανάλυση και την προσφορά εξατομικευμένης εμπειρίας προς τους χρήστες.



## Κεφάλαιο 6<sup>ο</sup>: Αναφορές

- [1] «AWS Cloud9 Amazon Web Services,» Amazon Web Services, Inc., [Ηλεκτρονικό]. Available: <https://aws.amazon.com/cloud9/>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [2] «Ruby on Rails,» Ruby on Rails, [Ηλεκτρονικό]. Available: <https://rubyonrails.org/>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [3] «Ruby Programming Language,» Ruby-lang.org, [Ηλεκτρονικό]. Available: <https://www.ruby-lang.org/>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [4] «SQLite Home Page,» Sqlite.org, [Ηλεκτρονικό]. Available: <https://www.sqlite.org/index.html>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [5] «PostgreSQL: The world's most advanced open source database,» Postgresql.org, [Ηλεκτρονικό]. Available: <https://www.postgresql.org/>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [6] «Git,» Git-scm.com, [Ηλεκτρονικό]. Available: <https://git-scm.com/>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [7] «Bitbucket | The Git solution for professional teams,» Bitbucket, [Ηλεκτρονικό]. Available: <https://bitbucket.org/>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [8] «Build software better, together,» GitHub, [Ηλεκτρονικό]. Available: <https://github.com>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [9] «Google Maps Platform - Geo-location APIs | Google Maps Platform | Google Cloud,» Google Cloud, [Ηλεκτρονικό]. Available: <https://cloud.google.com/maps-platform/>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [10] «Cloud Application Platform | Heroku,» Heroku.com, [Ηλεκτρονικό]. Available: <https://www.heroku.com/>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [11] Datazenit.com, «Heroku Data Explorer – Datazenit,» [Ηλεκτρονικό]. Available: <https://datazenit.com/heroku-data-explorer.html>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [12] «App.sendgrid.com,» SendGrid, [Ηλεκτρονικό]. Available: <https://app.sendgrid.com/>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [13] «Web framework,» En.wikipedia.org, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Web\\_framework](https://en.wikipedia.org/wiki/Web_framework). [Πρόσβαση 25 Σεπτέμβριος 2018].
- [14] «MIT License,» En.wikipedia.org, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/MIT\\_License](https://en.wikipedia.org/wiki/MIT_License). [Πρόσβαση 25 Σεπτέμβριος 2018].
- [15] «RubyGems.org | your community gem host,» Rubygems.org, [Ηλεκτρονικό]. Available: <https://rubygems.org/>. [Πρόσβαση 25 Σεπτέμβριος 2018].

- [16] «Sass: Syntactically Awesome Style Sheets,» Sass-lang.com, [Ηλεκτρονικό]. Available: <https://sass-lang.com/>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [17] «Bootstrap,» Bootstrap and contributors Mark Otto, [Ηλεκτρονικό]. Available: <http://getbootstrap.com/>. [Πρόσβαση 2018 Σεπτέμβριος 2018].
- [18] «Cascading Style Sheets,» En.wikipedia.org, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets). [Πρόσβαση 25 Σεπτέμβριος 2018].
- [19] «jQuery,» JS js.foundation, [Ηλεκτρονικό]. Available: <https://jquery.com/>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [20] «Free JavaScript training, resources and examples for the community,» Javascript.com, [Ηλεκτρονικό]. Available: <https://www.javascript.com/>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [21] «Web server,» En.wikipedia.org, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Web\\_server](https://en.wikipedia.org/wiki/Web_server). [Πρόσβαση 25 Σεπτέμβριος 2018].
- [22] mishoo, «UglifyJS2,» GitHub, [Ηλεκτρονικό]. Available: <https://github.com/mishoo/UglifyJS2>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [23] «CoffeeScript,» Coffeescript.org, [Ηλεκτρονικό]. Available: Coffeescript.org. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [24] «Single-page application,» En.wikipedia.org, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Single-page\\_application](https://en.wikipedia.org/wiki/Single-page_application). [Πρόσβαση 25 Σεπτέμβριος 2018].
- [25] «Domain-specific language,» En.wikipedia.org, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Domain-specific\\_language](https://en.wikipedia.org/wiki/Domain-specific_language). [Πρόσβαση 25 Σεπτέμβριος 2018].
- [26] «JSON,» Json.org, [Ηλεκτρονικό]. Available: <http://www.json.org/>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [27] «Cloud storage,» En.wikipedia.org, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Cloud\\_storage](https://en.wikipedia.org/wiki/Cloud_storage). [Πρόσβαση 25 Σεπτέμβριος 2018].
- [28] «IANA — Time Zone Database,» Iana.org, [Ηλεκτρονικό]. Available: <https://www.iana.org/time-zones>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [29] «URL,» En.wikipedia.org, [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/URL>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [30] «HTML,» En.wikipedia.org, [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/HTML>. [Πρόσβαση 25 Σεπτέμβριος 2018].

- [31] «XML,» En.wikipedia.org, [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/XML>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [32] «eRuby,» En.wikipedia.org, [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/ERuby>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [33] «Object-relational mapping,» En.wikipedia.org, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Object-relational\\_mapping](https://en.wikipedia.org/wiki/Object-relational_mapping). [Πρόσβαση 25 Σεπτέμβριος 2018].
- [34] «Database,» En.wikipedia.org, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Database#Database\\_management\\_system](https://en.wikipedia.org/wiki/Database#Database_management_system). [Πρόσβαση 25 Σεπτέμβριος 2018].
- [35] «Create, read, update and delete,» En.wikipedia.org, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Create,\\_read,\\_update\\_and\\_delete](https://en.wikipedia.org/wiki/Create,_read,_update_and_delete). [Πρόσβαση 25 Σεπτέμβριος 2018].
- [36] «Representational state transfer,» En.wikipedia.org, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer). [Πρόσβαση 25 Σεπτέμβριος 2018].
- [37] «Schema migration,» En.wikipedia.org, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Schema\\_migration](https://en.wikipedia.org/wiki/Schema_migration). [Πρόσβαση 25 Σεπτέμβριος 2018].
- [38] «Database schema,» En.wikipedia.org, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Database\\_schema](https://en.wikipedia.org/wiki/Database_schema). [Πρόσβαση 25 Σεπτέμβριος 2018].
- [39] «Plug-in (computing),» En.wikipedia.org, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Plug-in\\_\(computing\)](https://en.wikipedia.org/wiki/Plug-in_(computing)). [Πρόσβαση 25 Σεπτέμβριος 2018].
- [40] «Makefile,» En.wikipedia.org, [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/Makefile>. [Πρόσβαση 25 Σεπτέμβριος 2018].
- [41] «Rake (software),» En.wikipedia.org, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Rake\\_\(software\)](https://en.wikipedia.org/wiki/Rake_(software)). [Πρόσβαση 25 Σεπτέμβριος 2018].
- [42] «Business logic,» En.wikipedia.org, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Business\\_logic](https://en.wikipedia.org/wiki/Business_logic). [Πρόσβαση 25 Σεπτέμβριος 2018].
- [43] «Stateless protocol,» En.wikipedia.org, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Stateless\\_protocol](https://en.wikipedia.org/wiki/Stateless_protocol). [Πρόσβαση 25 Σεπτέμβριος 2018].
- [44] «HTTP cookie,» En.wikipedia.org, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/HTTP\\_cookie](https://en.wikipedia.org/wiki/HTTP_cookie). [Πρόσβαση 25 Σεπτέμβριος 2018].
- [45] «Cross-site request forgery,» En.wikipedia.org, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Cross-site\\_request\\_forgery](https://en.wikipedia.org/wiki/Cross-site_request_forgery). [Πρόσβαση 25 Σεπτέμβριος 2018].