

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ

ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ



Διαδικτυακή πλατφόρμα αναζήτησης ενοικιαζόμενης κατοικίας

Online platform for searching houses for rent

Διπλωματική Εργασία

Καταπίδης Νικόλαος Παντελεήμων

Επιβλέποντες Καθηγητές: Κατσαρός Δημήτριος

Λέκτορας Π.Θ. ΤΗΜΜΥ

Μποζάνης Παναγιώτης

Αναπληρωτής Καθηγητής Π.Θ. ΤΗΜΜΥ

ΒΟΛΟΣ 2014

*“Gather ye rosebuds while ye may,
Old Time is still a-flying;
And this same flower that smiles today,
Tomorrow will be dying.”*

Στον νονό μου

Ευχαριστίες

Με την περάτωση της παρούσης εργασίας, θα ήθελα να ευχαριστήσω τον επιβλέποντα μου κ. Δημήτριο Κατσαρό για την εμπιστοσύνη που έδειξε στο πρόσωπό μου και την καθοδήγησή του σε όλη τη διάρκεια εκπόνησής της.

Ευχαριστώ ακόμη τους φίλους μου τόσο για την υποστήριξή τους, όσο και για τις όμορφες στιγμές που ζήσαμε μαζί κατά τη διάρκεια των σπουδών μας.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου για την αμέριστη υποστήριξη, ενθάρρυνση και κατανόηση που μου έδειξαν σε όλη τη διάρκεια των σπουδών μου, συνεχίζοντας να είναι η κινητήριος δύναμη σε ό,τι κάνω στη ζωή μου.

Καταπίδης Νικόλαος Παντελεήμων

Βόλος, 2014

Περίληψη

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η υλοποίηση μιας διαδικτυακής εφαρμογής καταχώρησης και αναζήτησης οικιών προς ενοικίαση. Αρχικά, αναπτύχθηκε η λειτουργία εγγραφής χρήστη στη εφαρμογή τόσο με τη χρήση λογαριασμού email όσο και μέσω λογαριασμού στο κοινωνικό δίκτυο Facebook. Στη συνέχεια υλοποιήθηκαν οι λειτουργίες καταχώρησης οικίας και αναζήτησης με βάση τα κριτήρια που εισάγει ο χρήστης. Επίσης, υλοποιήθηκε λειτουργικότητα ανταλλαγής μηνυμάτων μεταξύ των χρηστών, καθώς και η καταχώρηση κριτικών-σχολίων για μία οικία. Τέλος, ενσωματώθηκε ένα γραφικό περιβάλλον διαχείρισης της εφαρμογής όπου μπορεί να γίνει επισκόπηση, αναζήτηση και διαγραφή των καταχωρημένων χρηστών καθώς και των σπιτιών που έχουν προστεθεί στην εφαρμογή.

Abstract

The objective of this thesis is the implementation of a web application for registering and searching for houses available for renting. The first functionality that was developed is registering via email or using a Facebook account. The next functions implemented are the registering of a house and searching with criteria set by the user. A user messaging system was also developed, as well as the ability to add reviews-comments to a house. Finally, a graphical interface for administration uses was implemented, where the user with administrative privileges can manage registered users and the houses they have added in the application.

Περιεχόμενα

Ευχαριστίες.....	5
Περίληψη.....	6
Abstract	7
Περιεχόμενα.....	8
Ευρετήριο Εικόνων.....	10
1. Εισαγωγή.....	11
1.1 Πλαίσιο της Διπλωματικής Εργασίας	11
1.2 Αντίστοιχες εφαρμογές	11
1.3 Επισκόπηση	12
2. Τεχνολογική επισκόπηση	13
2.1 Περιβάλλον Εργασίας	13
2.2 Ruby on Rails.....	13
2.2.1 Model-View-Controller (MVC).....	14
2.2.2 Ruby Gems.....	14
3. Δομή της εφαρμογής.....	17
3.1 Models.....	17
3.2 Controllers.....	20
3.2.1 RESTful Αρχιτεκτονική	20
3.2.3 Controllers & Actions.....	21
3.3 Views	23
4. Χαρακτηριστικά της εφαρμογής.....	26
4.1 Λειτουργίες εφαρμογής (User side).....	27
4.1.1 Εγγραφή χρήστη	27
4.1.2 Επεξεργασία προφίλ	28
4.1.3 Αναζήτηση σπιτιού	29
4.1.4 Προβολή καταχωρημένης οικίας	30
4.1.5 Προβολή προφίλ χρήστη	31
4.1.6 Αποστολή μηνύματος.....	32
4.1.7 Προβολή μηνυμάτων χρήστη.....	32
4.1.8 Καταχώρηση οικίας.....	34
4.1.9 Καταχώρηση κριτικής.....	35
4.2 Administration Interface (Active Admin).....	36
4.2.1 Εμφάνιση εγγραφής.....	37

4.2.2 Επεξεργασία εγγραφής	37
.....	38
5. Μελλοντικές Επεκτάσεις	39
Αναφορές.....	40

Ευρετήριο Εικόνων

Εικόνα 1: Δομή του Rails Framework	13
Εικόνα 2: MVC Architecture	14
Εικόνα 3: Database Schema	19
Εικόνα 4: Κεντρική σελίδα.....	27
Εικόνα 5: Sign up via email	28
Εικόνα 6: Επεξεργασία προφίλ (Facebook).....	28
Εικόνα 7: Επεξεργασία προφίλ (email)	29
Εικόνα 8: Αναζήτηση σπιτιού	29
Εικόνα 9: Αποτελέσματα αναζήτησης.....	30
Εικόνα 10: Σελίδα σπιτιού	31
Εικόνα 11: Προφίλ χρήστη	31
Εικόνα 12: Αποστολή προσωπικού μηνύματος χρήστη.....	32
Εικόνα 13: Εισερχόμενα μηνύματα.....	33
Εικόνα 14: Προβολή συνομιλίας.....	33
Εικόνα 15: Trash	34
Εικόνα 16: Καταχώρηση σπιτιού.....	34
Εικόνα 17: Τροποποιημένη κεντρική σελίδα	35
Εικόνα 18: Φόρμα καταχώρησης κριτικής.....	35
Εικόνα 19: Εμφάνιση κριτικής.....	36
Εικόνα 20: Ενδεικτική σελίδα διαχείρισης μοντέλου	36
Εικόνα 21: Προβολή καταχώρησης.....	37
Εικόνα 22: Επεξεργασία καταχώρησης.....	38

1. Εισαγωγή

1.1 Πλαίσιο της Διπλωματικής Εργασίας

Διανύοντας μια εποχή όπου οι διαδικτυακές εφαρμογές γνωρίζουν μεγάλη εξάπλωση, άλλοτε χρονοβόρες διαδικασίες απλουστεύονται, διευκολύνοντας έτσι το χρήστη κι εξοικονομώντας τον πολύτιμο χρόνο του. Παρατηρώντας την ακαδημαϊκή κοινότητα, όπου η εύρεση σπιτιού προς ενοικίαση αποτελεί ανάγκη και συνάμα πρόβλημα σε ετήσια βάση, η χρησιμότητα μιας εφαρμογής που θα διευκολύνει τη διαδικασία αυτή γίνεται άμεσα εμφανής.

Για τους σκοπούς της παρούσας διπλωματικής εργασίας επιλέχθηκε η υλοποίηση μιας εφαρμογής που να διαθέτει λειτουργικότητα καταχώρησης, αναζήτησης και εύρεσης ενοικιαζόμενης κατοικίας. Ακόμη, με τη βοήθεια ενός συστήματος ανταλλαγής προσωπικών μηνυμάτων, οι χρήστες μπορούν να επικοινωνούν μεταξύ τους για επιπλέον πληροφορίες σε σχέση με μια διαθέσιμη κατοικία. Επιπλέον, η δυνατότητα καταγραφής σχολίων και κριτικών διευκολύνει την επιλογή του χρήστη και συμβάλλει στην ανάδειξη των ποιοτικότερων καταχωρήσεων.

Η εφαρμογή μπορεί να χρησιμοποιηθεί από ακαδημαϊκά ιδρύματα ώστε τα μέλη της κοινότητας να μπορούν άμεσα να αναζητήσουν κατοικία, λαμβάνοντας υπόψιν τις γνώμες και τις εντυπώσεις των συναδέλφων τους, ενώ και οι ιδιοκτήτες σπιτιών έχουν τη δυνατότητα να προβάλλουν τις κατοικίες τους απευθυνόμενοι σε ένα ιδιαίτερα ενεργό αγοραστικό κοινό.

1.2 Αντίστοιχες εφαρμογές

Η πλειονότητα των διαθέσιμων εμπορικών εφαρμογών επικεντρώνεται στην ενοικίαση οικίας στα πλαίσια των διακοπών, αλλά και όσες στοχεύουν στην αγορά και ενοικίαση ακινήτων, παρουσιάζουν μεγαλύτερη πολυπλοκότητα χρήσης και δεν παρέχουν αμεσότητα επικοινωνίας μεταξύ των χρηστών. Επίσης, δεδομένης της εμπορικής προσέγγισης των εφαρμογών αυτών, υπάρχουν χρεώσεις για τους ιδιοκτήτες ή και τους ενοικιαστές. Κυριότερες από τις εφαρμογές αυτές για την ελληνική αγορά είναι το spitogatos.gr και το studenthouse.gr, ενώ ιδιαίτερα δημοφιλές αλλά προσανατολισμένο περισσότερο στην ενοικίαση οικίας για διακοπές είναι το airbnb.gr.

Η διαφορά της εφαρμογής που αναπτύχθηκε στα πλαίσια της εργασίας εντοπίζεται σε δύο βασικά στοιχεία, την απλότητα χρήσης και την αμεσότητα επικοινωνίας μεταξύ των χρηστών. Οι προαναφερθείσες πλατφόρμες αποτελούν κατά κανόνα πιο σύνθετα προϊόντα, ενώ η εμπορικός τους προσανατολισμός προσθέτει χρεώσεις τις οποίες έχουμε τη δυνατότητα να αποφύγουμε καθιστώντας την εφαρμογή μια υπηρεσία του ακαδημαϊκού ιδρύματος προς τα μέλη της κοινότητάς του.

1.3 Επισκόπηση

Στο κεφάλαιο 2 γίνεται μια τεχνική επισκόπηση της εφαρμογής, παρουσιάζοντας τις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίησή της.

Στο κεφάλαιο 3 παρουσιάζεται η MVC αρχιτεκτονική πάνω στην οποία κτίστηκε η πλατφόρμα, με επιμέρους ανάλυση των Models, Views και Controllers και των ρόλων που επιτελούν.

Στο κεφάλαιο 4 αναλύονται οι λειτουργίες της εφαρμογής, που χωρίζονται στην πλευρά των απλών χρηστών και στην πλευρά των administrators. Εκεί παρουσιάζονται σενάρια χρήσης συνοδευόμενα από στιγμιότυπα του περιβάλλοντος χρήσης της εφαρμογής.

Το κεφάλαιο 5 περιέχει ιδέες για μελλοντικές επεκτάσεις που μπορούν να εμπλουτίσουν το περιεχόμενο και τη λειτουργικότητα της διαδικτυακής πλατφόρμας.

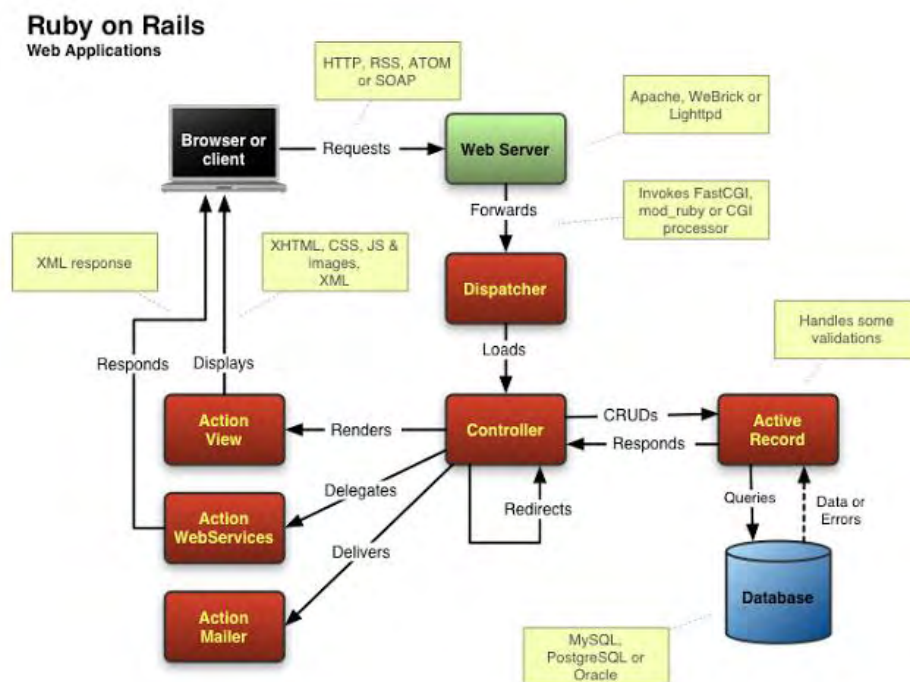
2. Τεχνολογική επισκόπηση

2.1 Περιβάλλον Εργασίας

Η ανάπτυξη της διαδικτυακής πλατφόρμας έγινε σε περιβάλλον Ubuntu 12.04 [1], με τη χρήση της προγραμματιστικής γλώσσας Ruby [2] και του framework Rails [3]. Η βάση δεδομένων που χρησιμοποιήθηκε είναι η sqlite3 [4]. Η συγγραφή του κώδικα έγινε με χρήση του editor Sublime Text [5]. Για την διαχείριση του πηγαίου κώδικα χρησιμοποιήθηκε το σύστημα διαχείρισης εκδόσεων λογισμικού Git [6], ενώ δημιουργήθηκε ένα απομακρυσμένο repository (εναποθετήριο) στο Github [7], όπου μπορεί να εντοπιστεί ο κώδικας της εφαρμογής στο σύνδεσμο: <https://github.com/nikatapi/house-search-app>.

2.2 Ruby on Rails

Το Ruby on Rails είναι ένα full-stack web application framework [8] ανοικτού κώδικα για την γλώσσα προγραμματισμού Ruby. Δίνει έμφαση στη χρήση software engineering μεθόδων και προτύπων, όπως το “don’t repeat yourself” (DRY) για όσο το δυνατόν καθαρότερο και μη επαναλαμβανόμενο κώδικα, και την αρχιτεκτονική Model-view-controller (MVC). Το framework αποτελείται από μέρη τα οποία αναλαμβάνουν τη διαχείριση διαφορετικών πόρων της εφαρμογής, από τη βάση δεδομένων μέχρι την οπτική αναπαράστασή τους στη μεριά του χρήστη. Παρακάτω δίνεται μια οπτική αναπαράσταση των μερών αυτών και ο τρόπος αλληλεπίδρασής τους:

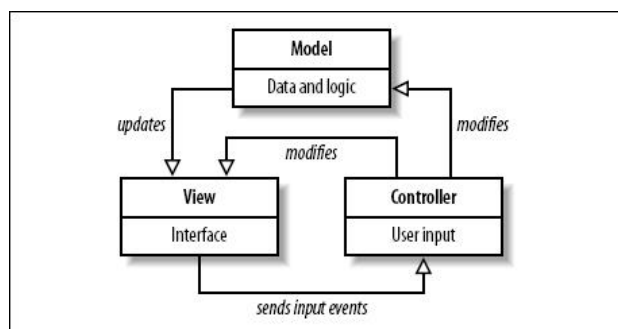


Εικόνα 1: Δομή του Rails Framework

2.2.1 Model-View-Controller (MVC)

Το πρότυπο MVC αποτελεί βασική αρχή του framework Rails, χωρίζοντας την εφαρμογή σε τρία βασικά συστατικά μέρη:

- **Model:** Αντιπροσωπεύει τις οντότητες (δεδομένα) της εφαρμογής, συμπεριλαμβάνοντας και τους κανόνες διαχείρισης τους. Συνήθως κάθε μοντέλο αντιστοιχεί σε έναν πίνακα της βάσης δεδομένων.
- **View:** Το κομμάτι αυτό επικεντρώνεται στην εμφάνιση των δεδομένων στο χρήστη, μέσω του web browser ή οποιουδήποτε άλλου μέσου μπορεί να κάνει αιτήματα για δεδομένα στην εφαρμογή
- **Controller:** Ο controller είναι υπεύθυνος για την αλληλεπίδραση και τη μετάδοση της πληροφορίας μεταξύ του χρήστη, του view και του μοντέλου. Μπορεί να στέλνει εντολές ώστε να ανανεώνει την κατάσταση ενός μοντέλου, ή να ενημερώνει το view ώστε να αλλάζει την εμφανιζόμενη πληροφορία.



Εικόνα 2: MVC Architecture

2.2.2 Ruby Gems

Η γλώσσα Ruby διαθέτει μια ιδιαίτερα ενεργή κοινότητα χρηστών, με πληθώρα βιβλιοθηκών και πακέτων που παρέχουν χρήσιμες λειτουργίες ώστε να επιταχύνεται η διαδικασία ανάπτυξης μιας εφαρμογής. Η διαχείριση αυτών των πακέτων διευκολύνεται με τη βοήθεια του `Rubygems[9]` package manager, που αυτοματοποιεί τη διαδικασία εγκατάστασης. Στο διαδικτυακό χώρο [Rubygems](#) υπάρχει μια μεγάλη βάση από gems για διάφορες λειτουργίες όπου μπορεί κανείς να αναζητήσει κάποιο πακέτο για να καλύψει τις ανάγκες της εφαρμογής του.

Για τους σκοπούς της παρούσας εργασίας, πολλά μέρη της λειτουργικότητας υλοποιήθηκαν με τη βοήθεια gems που απλοποιούν τη διαδικασία ανάπτυξης και παρέχουν δοκιμασμένες λύσεις σε συχνά προβλήματα στο χώρο του web development. Στη συνέχεια παρατίθεται μια λίστα με τα gems και τη λειτουργικότητα που εξυπηρετούν.

- **Rails** - Το framework Rails αποτελεί κι αυτό ένα Ruby gem που εγκαθίσταται με τη βοήθεια του rubygems package manager.
- **Sass-rails** – Παρέχει επίσημη ενσωμάτωση της γλώσσας Sass[10] σε Ruby on Rails εφαρμογές
- **Coffee-Rails** – Προσθέτει υποστήριξη για χρήση της γλώσσας Coffeescript[11] και τη δυνατότητα απάντησης σε Javascript requests (με τη χρήση αρχείων .js.coffee ως views). Παρέχεται εξ’ορισμού κατά τη δημιουργία μιας εφαρμογής από την έκδοση 3.1 του Rails κι έπειτα.
- **jQuery-Rails** – Ενσωματώνει υποστήριξη για jQuery[12] σε εφαρμογές Rails.
- **Turbolinks** – Τα turbolinks επιταχύνουν την μετάβαση μεταξύ τμημάτων της εφαρμογής, κρατώντας “ζωντανή” την τρέχουσα σελίδα κι αντικαθιστώντας μόνο το body και τον τίτλο στο head, χωρίς ο browser να χρειάζεται να μεταγλωττίσει ξανά τον κώδικα Javascript και CSS[13].
- **Devise** – Ένα authentication gem υπεύθυνο για την εγγραφή και σύνδεση των χρηστών στην εφαρμογή. Παρέχει λειτουργίες όπως καταγραφή του αριθμού των συνδέσεων ενός χρήστη, επιβεβαίωση λογαριασμού μέσω email και κλειδωμα λογαριασμού μετά από έναν αριθμό αποτυχημένων προσπαθειών σύνδεσης.
- **Omniauth** – Μια βιβλιοθήκη που διευκολύνει την ταυτοποίηση (authentication) από διαφορετικούς παρόχους (π.χ. Facebook, Google)
- **Omniauth Facebook** – Rack Middleware [14] που σε συνεργασία με το Omniauth δίνει τη δυνατότητα authentication μέσω του API [15] που παρέχει το Facebook
- **Bootstrap-Sass** – Παρέχει μια έκδοση του front-end framework [16] Bootstrap [17] βασισμένη σε Sass, που διευκολύνει την ανάπτυξη του γραφικού περιβάλλοντος της εφαρμογής.
- **Will_paginate & Will_paginate_bootstrap** – Το will_paginate είναι μια βιβλιοθήκη που διευκολύνει τη σελιδοποίηση δεδομένων, ενώ το will_paginate_bootstrap ενσωματώνει τη λειτουργικότητα σελιδοποίησης του Bootstrap με το will_paginate gem.
- **Sprockets** - Βιβλιοθήκη για τη μεταγλώττιση και την παροχή των web assets.
- **Activeadmin** – Ένα πρόσθετο για τη δημιουργία περιβάλλοντος διαχείρισης της εφαρμογής από χρήστες με προνόμια administrator. Συνεργάζεται με το Devise για την ταυτοποίηση των χρηστών.
- **Geocoder** – Βιβλιοθήκη γεωγραφικής κωδικοποίησης/αποκωδικοποίησης. Δίνει τη δυνατότητα μετατροπής διευθύνσεων σε συντεταγμένες και το αντίστροφο καθώς και γεωγραφικών ερωτημάτων προς τη βάση δεδομένων.
- **Gmaps4rails** – Απλοποιεί την ενσωμάτωση χαρτών μέσω της υπηρεσίας Google Maps, παρέχοντας εργαλεία για την παραμετροποίηση του εμφανιζόμενου χάρτη.
- **Carrierwave** – Gem για εύκολο ανέβασμα αρχείων προς την βάση δεδομένων ή προς κάποιο cloud storage solution [18]. Στην παρούσα εφαρμογή χρησιμοποιείται για upload εικόνων, τόσο για το avatar [19] του χρήστη όσο και για τις καταχωρήσεις κατοικιών.
- **Rmagick & MiniMagick** – Βιβλιοθήκες για επεξεργασία εικόνας.
- **Mailboxer** – Χρησιμοποιείται για την λειτουργία ανταλλαγής προσωπικών μηνυμάτων μεταξύ των χρηστών
- **Acts-as-commentable-with-threading** – Ένα gem που δίνει τη δυνατότητα υλοποίησης σχολίων-κριτικών με εύκολο τρόπο.

- **Font-awesome-rails** – Παρέχει γραμματοσειρές και εικονίδια για το γραφικό περιβάλλον της εφαρμογής.

3. Δομή της εφαρμογής

Η αρχιτεκτονική της πλατφόρμας είναι βασισμένη στο MVC πρότυπο, κατά τη σύμβαση που ισχύει κατά κανόνα για εφαρμογές Ruby on Rails. Στη συνέχεια αναλύονται τα μέρη που αποτελούν τα Models, Views και Controllers, καθώς και οι ρόλοι τους οποίους επιτελούν.

3.1 Models

Κατά τη σύμβαση του Rails framework, ένα model συνήθως αντιστοιχεί σε έναν πίνακα της βάσης δεδομένων. Διατηρώντας την αρχή “Convention over Configuration” (CoC) που χαρακτηρίζει το Rails, το όνομα του μοντέλου αντικατοπτρίζεται στο όνομα του πίνακα της database, καθώς και στο όνομα του αρχείου ruby που περιγράφει το μοντέλο. Το αρχείο αυτό επιπλέον περιέχει τις συσχετίσεις με τα υπόλοιπα μοντέλα και κάποιες συναρτήσεις (business logic) που χρησιμοποιούνται ώστε να μην επιβαρύνεται ο αντίστοιχος controller με υπολογισμούς που καταρρίπτουν τη σύμβαση της MVC αρχιτεκτονικής. Η συνολική δομή της βάσης δεδομένων με τα λεπτομερή πεδία κάθε μοντέλου μπορεί να εξεταστεί ενδελεχώς στο αρχείο schema.rb που βρίσκεται στον κατάλογο /db της εφαρμογής. Παρακάτω παρατίθενται τα μοντέλα που υλοποιήθηκαν για την εφαρμογή της παρούσας εργασίας..

User

Ο χρήστης αποτελεί τη βασική οντότητα της εφαρμογής. Το μοντέλο περιέχει πεδία που έχουν καθοριστεί αυτομάτως από το Devise και σχετίζονται με στοιχεία καταγραφής της δραστηριότητας όπως μετρητή για τον αριθμό των συνδέσεων (sign in), την τρέχουσα και τελευταία διεύθυνση IP από την οποία συνδέθηκε ο χρήστης στην εφαρμογή, καθώς και πεδία που καθορίζονται κατά την εγγραφή του χρήστη, όπως το όνομα και το email. Επίσης υπάρχει ένα πεδίο που αποθηκεύει το URL[20] όπου είναι αποθηκευμένο το avatar του χρήστη, εφόσον ο ίδιος επιλέξει να ανεβάσει μια φωτογραφία. Ο χρήστης μπορεί να έχει πολλά σπίτια, μηνύματα και σχόλια-κριτικές που ο ίδιος γράφει σε καταχωρημένα σπίτια.

House

Το μοντέλο που αντιπροσωπεύει την οντότητα ενός σπιτιού, διαθέτει πεδία περιγραφής, χώρας, πόλης, διεύθυνσης, τιμής και γεωγραφικών συντεταγμένων οι οποίες λαμβάνονται μέσω του Geocoder gem. Ανήκει σε ένα χρήστη και μπορεί να διαθέτει πολλές εικόνες και σχόλια-κριτικές.

Search

Οι αναζητήσεις των χρηστών θεωρούνται ένα μοντέλο, ώστε η αποθήκευσή τους να δίνει τη δυνατότητα δυναμικής ανάλυσης των δεδομένων για μελλοντική βελτίωση της εφαρμογής. Οι αναζητήσεις περιέχουν πεδία χώρας, πόλης, διεύθυνσης, μέγιστης τιμής και μέγιστης

απόστασης από την δοθείσα διεύθυνση αναζήτησης. Τα πεδία αυτά χρησιμοποιούνται ώστε να επιστρέφονται τα κατάλληλα σπίτια μετά από εφαρμογή φίλτρων στον πίνακα `houses` της βάσης δεδομένων μέσω της συνάρτησης `find_houses`.

Comment

Το μοντέλο των σχολίων δημιουργείται κατά την εγκατάσταση του gem `acts_as_commentable_with_threading`. Ένα σχόλιο ανήκει στο σπίτι για το οποίο καταχωρήθηκε, όπως και στο χρήστη ο οποίος το έγραψε. Περιέχει πεδία για το περιεχόμενο του σχολίου, όπως και το `id` του χρήστη που το καταχώρησε.

Picture

Για την αποθήκευση εικόνων, διατηρείται ένας πίνακας με όνομα `pictures` στη βάση δεδομένων. Σε συνδυασμό με τη λειτουργικότητα των `Carrierwave`, `Rmagick` & `MiniMagick` γίνεται αποθήκευση διαφορετικών εκδόσεων μιας εικόνας ανάλογα με το μέγεθος που χρειάζεται σε διαφορετικά `views` της εφαρμογής.

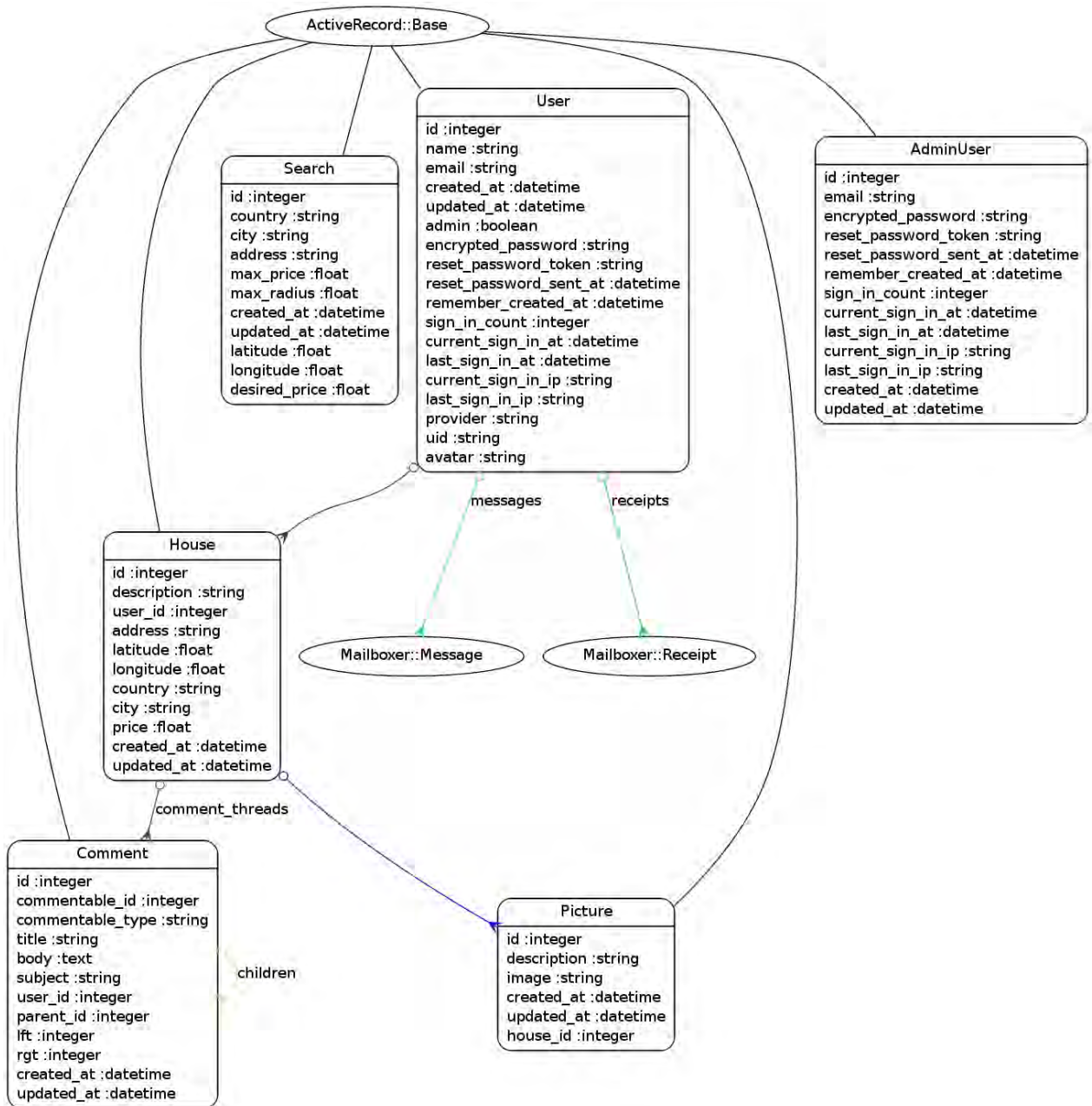
AdminUser

Για την χρήση του `ActiveAdmin` framework υπάρχει το μοντέλο `AdminUser` όπου βρίσκονται καταχωρημένοι οι διαχειριστές της εφαρμογής. Η ύπαρξη του μοντέλου επιπλέον του `User` model υφίσταται λόγω του ότι δεν είναι απαραίτητο ένας διαχειριστής να είναι και χρήστης της εφαρμογής. Το μοντέλο αυτό δημιουργείται αυτόματα κατά την εγκατάσταση του `ActiveAdmin` και συνεργάζεται με το `Devise` για το authentication των χρηστών.

Conversations & Messages

Το `Mailboxer` gem εισάγει μοντέλα για τα μηνύματα, τις ειδοποιήσεις(notifications) και αποδείξεις (receipts) που σχετίζονται με την κατάσταση των μηνυμάτων που έχουν ανταλλαγεί (αναγνωσμένα, διαγεγραμμένα). Τα μηνύματα εξ 'ορισμού ανήκουν σε μια συζήτηση (conversation) που στην εφαρμογή μας περιορίζεται μεταξύ δύο χρηστών.

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



Εικόνα 3: Database Schema

3.2 Controllers

3.2.1 RESTful Αρχιτεκτονική

Η REST (representational state transfer) αποτελεί μια αρχιτεκτονική που αντιστοιχεί κάθε πόρο της εφαρμογής με ένα μοναδικό url, ενώ οι λειτουργίες δημιουργίας, ανάγνωσης, ανανέωσης και διαγραφής γνωστές και ως CRUD (Create, Read, Update, Delete) σε κάθε πόρο γίνονται μέσω των HTTP μεθόδων GET, POST, PUT, DELETE.

Στο framework Rails, ο Rails router αποτελεί το εργαλείο που αναγνωρίζει urls και τα προωθεί στην κατάλληλη ενέργεια (action) του controller που είναι υπεύθυνος για τον πόρο που προσδιορίζεται στο url. Σύμφωνα με τη σύμβαση του Rails, κάθε ενέργεια στον controller αντιστοιχεί σε μια CRUD ενέργεια στη βάση δεδομένων. Οι πόροι καθορίζονται στο αρχείο routes.rb που βρίσκεται στον κατάλογο /config της εφαρμογής, ενώ στο ίδιο αρχείο μπορούν τα προκαθορισμένα urls να αντικατασταθούν με urls που επιθυμεί ο προγραμματιστής.

Για παράδειγμα, ο πόρος house δηλώνεται στο route file ως εξής:

```
resources :houses
```

Η δήλωση αυτή δημιουργεί επτά διαφορετικά routes, το καθένα από τα οποία οδηγείται στον HousesController, όπως φαίνεται στον παρακάτω πίνακα:

HTTP Verb	URL	Controller#Action	Χρήση
GET	/houses	HousesController#index	Εμφανίζει τη λίστα όλων των σπιτιών
GET	/houses/new	HousesController#new	Επιστρέφει μια HTML φόρμα για τη δημιουργία νέου σπιτιού
POST	/houses	HousesController#create	Δημιουργεί ένα νέο σπίτι
GET	/houses/:id	HousesController#show	Εμφανίζει ένα συγκεκριμένο σπίτι
GET	/houses/:id/edit	HousesController#edit	Επιστρέφει μια HTML φόρμα για ενημέρωση των στοιχείων ενός σπιτιού
PATCH/PUT	/houses/:id	HousesController#update	Ανανεώνει τα στοιχεία ενός σπιτιού
DELETE	/houses/:id	HousesController#destroy	Διαγράφει ένα συγκεκριμένο σπίτι

3.2.3 Controllers & Actions

Η εφαρμογή χρησιμοποιεί τους εξής controllers για την διαχείριση των πόρων:

application controller

Ο `application_controller` δημιουργείται αυτόματα και αποτελεί την ιεραρχική ρίζα των controllers της εφαρμογής. Εκεί χρησιμοποιούνται συναρτήσεις και φίλτρα για προστασία από Cross-Site Request Forgery (CSRF)[21] καθώς και για τον ορισμό επιτρεπόμενων παραμέτρων για τη διαδικασία εγγραφής (sign up) και ανανέωσης των στοιχείων του λογαριασμού χρήστη.

static_pages controller

Χρησιμοποιείται για την ενημέρωση της κεντρικής σελίδας της εφαρμογής, η οποία εφόσον ο χρήστης έχει συνδεθεί κι έχει καταχωρήσει δικά του σπίτια, εμφανίζει μια συνοπτική εικόνα των καταχωρήσεών του μαζί με ένα χάρτη όπου επισημαίνονται οι τοποθεσίες των σπιτιών.

users controller

Σε αυτό τον controller ορίζονται οι συναρτήσεις `show`, `edit` και `destroy`. Η `show` ενημερώνει τη μεταβλητή `@hash` που χρησιμοποιείται για την εμφάνιση των γεωγραφικών σημείων όπου εντοπίζονται τα σπίτια του χρήστη, η `edit` ανανεώνει τα στοιχεία του χρήστη και κάνει `redirect` στη σελίδα του εφόσον η καταχώρηση είναι επιτυχής, ενώ η `destroy` διαγράφει το χρήστη από τη βάση δεδομένων. Η χρήση των συναρτήσεων περιορίζεται με τα φίλτρα `:correct_user` (`edit`), `:authenticate_user!` (`edit`), `:admin_user` (`destroy`) ώστε μόνον οι κατάλληλοι χρήστες να μπορούν να εκμεταλλευτούν τη λειτουργικότητά τους. Για παράδειγμα ένας χρήστης χωρίς δικαιώματα διαχειριστή δεν μπορεί να καλέσει το `destroy` action, ή ένας μη συνδεδεμένος χρήστης δεν μπορεί να αλλάξει τα στοιχεία ενός άλλου.

houses controller

Για τη δημιουργία και επεξεργασία των σπιτιών, ο `houses_controller` αναλαμβάνει το συντονισμό μεταξύ `views` και `model`. Η `create` χρησιμοποιεί τη συνάρτηση `build` του Rails για τη δημιουργία του σπιτιού, καθώς αποτελεί πόρο-παιδί του τρέχοντος χρήστη. Επίσης υπάρχει ένας βρόχος για την αποθήκευση πολλαπλών εικόνων εφόσον αυτές έχουν καταχωρηθεί κατά την υποβολή της φόρμας νέου σπιτιού. Η συνάρτηση `show` αναλαμβάνει να δώσει τα σωστά στοιχεία για το `marker` του χάρτη, καθώς και να διαβάσει τα σχόλια που υπάρχουν για το τρέχων σπίτι και να τα αποθηκεύσει στη μεταβλητή `@comments`.

searches controller

Ο controller που είναι υπεύθυνος για τις αναζητήσεις σπιτιών, έχει τις συναρτήσεις new, create και show. Η χρήση των συναρτήσεων ακολουθεί πλήρως τη σύμβαση της ονομασίας τους κι επιπροσθέτως γίνεται ενημέρωση της μεταβλητής @hash για την προβολή των σημείων ενδιαφέροντος στους ανάλογους χάρτες που υπάρχουν στα views.

conversations controller

Το mailboxer gem που χρησιμοποιείται για τη λειτουργικότητα ανταλλαγής προσωπικών μηνυμάτων μεταξύ των χρηστών, ομαδοποιεί τα μηνύματα σε συζητήσεις (conversations). Οι συνήθεις λειτουργίες ενός συστήματος messaging επιβάλλουν τον καθορισμό επιπλέον συναρτήσεων στον conversations_controller. Έτσι, ορίζονται οι: reply (αποστολή απάντησης σε conversation), trashbin (ορίζει το “καλάθι άχρηστων”), trash (μεταφέρει μια συζήτηση στο trashbin), untrash (επαναφορά συζήτησης), empty_trash (άδειασμα του trashbin-οριστική διαγραφή). Επίσης υπάρχουν οι συμβατικές συναρτήσεις index, create για εμφάνιση όλων των συζητήσεων και δημιουργία μιας νέας συζήτησης. Αξίζει να σημειωθεί πως η new δεν υπάρχει αλλά μεταφέρεται στον messages_controller όπου επιστρέφεται η HTML φόρμα αποστολής νέου μηνύματος.

messages controller

Για την δημιουργία ενός νέου μηνύματος και την αποστολή του, ο messages_controller διαθέτει τις συναρτήσεις new και create, ενώ μετά την επιτυχή αποστολή γίνεται επανακατεύθυνση του χρήστη στο index action του conversations_controller μέσω της εντολής `redirect_to :conversations`.

comments controller

Τα σχόλια-κριτικές για ένα σπίτι καταχωρούνται και διαγράφονται με τη βοήθεια του comments_controller, ο οποίος διαθέτει τις ενέργειες create και destroy αντίστοιχα.

Omniauth_callbacks controller

Για την εγγραφή και σύνδεση μέσω Facebook ο χρήστης επανακατευθύνεται στο Facebook ώστε να εισάγει τα στοιχεία του και στη συνέχεια επαναπροωθείται στην callback μέθοδο της εφαρμογής. Για την υλοποίηση αυτής τη callback μεθόδου δημιουργείται ο Omniauth_callbacks_controller, όπου γίνεται ανάκτηση των στοιχείων του χρήστη από το hash που επιστρέφεται από το Facebook μέσω του OmniAuth και τελικά η σύνδεση ή εγγραφή του μέσω του devise.

registrations controller

Ο controller αυτός αποτελεί μέρος του devise gem και προστίθεται αυτόματα στην εφαρμογή με την εγκατάστασή του. Είναι υπεύθυνος για τις εγγραφές χρηστών καθώς και την ανανέωση των στοιχείων (credentials) τους.

3.3 Views

Τα views αποτελούν την άμεση διεπαφή μεταξύ χρήστη κι εφαρμογής, εμφανίζοντας πληροφορία αλλά και δεχόμενα δεδομένα μέσω των forms. Τα αρχεία εντοπίζονται στον κατάλογο `/app/views` της εφαρμογής, ενώ για κάθε controller υπάρχουν οι αντίστοιχοι φάκελοι στους οποίους περιέχονται αρχεία `.html.erb` που αντιστοιχούν σε κάποιες από τις ενέργειες του controller.

Η προγραμματιστική φιλοσοφία DRY βρίσκει άμεση εφαρμογή στο κομμάτι των views, καθώς δίνεται η δυνατότητα επαναχρησιμοποίησης οπτικών στοιχείων και κατά συνέπεια κώδικα που εμφανίζεται σε διαφορετικά σημεία της πλατφόρμας, μέσω των λεγόμενων partials, τα οποία είναι τμήματα κώδικα html με ενσωματωμένη ruby (`html.erb`). Με τη χρήση της ruby εντολής `render`, μπορούμε να ενσωματώσουμε αυτά τα τμήματα κώδικα σε διαφορετικά view αρχεία, ενώ διευκολύνεται και η διαχείριση του κώδικα καθώς διασπάται σε αυτόνομα μικρότερα τμήματα. Η σύμβαση του Rails framework ορίζει τα ονόματα των partials να ξεκινούν με το σύμβολο `“_”` ακολουθούμενο από το όνομα του, ενώ κατά την κλήση της `render` το σύμβολο αυτό παραλείπεται.

Οι κατάλογοι των views και τα αρχεία που περιέχονται σε αυτούς συνοπτικά έχουν ως εξής:

static_pages

`about`: Μια σελίδα που αναφέρει πληροφορίες για την εφαρμογή

`contact`: Στοιχεία επικοινωνίας με τους διαχειριστές-υπεύθυνους της εφαρμογής

`help`: Βοήθεια σχετικά με τη χρήση της πλατφόρμας

`home`: Κεντρική σελίδα όπου εμφανίζονται πληροφορίες για την εφαρμογή και κουμπιά εγγραφής-σύνδεσης

users

`_user`: Partial με βασικές πληροφορίες για τον χρήστη

`index`: Εμφάνιση όλων των εγγεγραμμένων χρηστών

`show`: Προφίλ χρήστη με πληροφορίες και καταχωρημένα σπίτια

houses

`_house`: Partial που περιέχει τις πληροφορίες ενός σπιτιού

`new`: Φόρμα για την καταχώρηση ενός σπιτιού

`show`: Εμφάνιση πλήρους σελίδας κατοικίας προς ενοικίαση

searches

new: Φόρμα για αναζήτηση σπιτιού

show: Εμφάνιση αποτελεσμάτων μιας αναζήτησης

conversations

_conversation: Partial για την εμφάνιση μιας συνομιλίας

_form: Φόρμα απάντησης σε συνομιλία

index: Εμφάνιση εισερχόμενων, απεσταλμένων και μηνυμάτων που έχουν σταλεί στον κάδο ανακύκλωσης

new: Γίνεται render του _form

show: Εμφάνιση μηνυμάτων μιας συνομιλίας με δυνατότητα απάντησης μέσω υποβολής φόρμας

messages

_form: Φόρμα για την υποβολή νέου μηνύματος προς κάποιο χρήστη

comments

_comment: Partial για την εμφάνιση ενός σχολίου

_form: Φόρμα μέσω της οποίας συλλέγονται τα στοιχεία για ένα νέο σχόλιο-κριτική

shared

Στον κατάλογο αυτό περιέχονται partials που επαναχρησιμοποιούνται σε διάφορες σελίδες της εφαρμογής.

_contact_owner: Φόρμα αποστολής μηνύματος στον προβαλλόμενο χρήστη

_error_messages: Μηνύματα λάθους

_feed: Λίστα με τα σπίτια που έχει καταχωρήσει ο χρήστης

_feed_item: Σπίτι-στοιχείο που ενσωματώνεται στη λίστα του _feed

_house_details: Πληροφορίες κατοικίας

_house_images_slider: Εμφανίζει τις καταχωρημένες εικόνες ενός σπιτιού με τη μορφή carousel

`_house_map`: Script για τον ορισμό χάρτη και των σημείων που εμφανίζονται πάνω του

`_main_jumbotron`: Βασικό οπτικό στοιχείο της κεντρικής σελίδας

`_map_panel`: Bootstrap panel που περιέχει χάρτη

`_message_modal`: αναδύομενο παράθυρο (Bootstrap modal) που περιέχει τη φόρμα νέου μηνύματος

`_reply_to_conversation_modal`: αναδύομενο παράθυρο με φόρμα για απάντηση σε υπάρχουσα συζήτηση

`_user_info`: Προβάλλει βασικές πληροφορίες για έναν χρήστη και κουμπιά για την προβολή προφίλ και αποστολή μηνύματος

`_welcome`: Οπτικό στοιχείο για την κεντρική σελίδα, όπου περιγράφεται η βασική λειτουργικότητα της εφαρμογής

devise

Το Devise κατά την εγκατάστασή του προσθέτει view αρχεία σχετικά με τις λειτουργίες του. Επειδή το Devise αποτελεί ένα engine, όλα τα views είναι πακεταρισμένα στο gem. Ωστόσο, μπορούμε να τα παραμετροποιήσουμε αντιγράφοντάς τα στον κατάλογο `/views` μέσω της εντολής `rails generate devise:views`

Χρησιμοποιώντας το modal component του Bootstrap για αναδύομενα παράθυρα δημιουργήθηκαν τα εξής modals, ως προσθήκη των views που παρέχει το Devise:

`_login_modal`: Για σύνδεση υφιστάμενου χρήστη

`_sign_up_modal`: Χρησιμοποιείται για εγγραφή ενός νέου χρήστη

layouts

Κατάλογος που δημιουργείται αυτόματα από το Rails framework και σχετίζεται με τον `application_controller`.

`application`: πρότυπο (template) που κληρονομείται από όλα τα views και περιέχει scripts για τους χάρτες, τα modals, ενώ επίσης δηλώνει και τα stylesheets και scripts που δηλώνονται στα κατάλληλα αρχεία στο φάκελο `app/assets`

`header`: κεφαλίδα που εμφανίζεται σε όλες τις σελίδες, περιέχει links ανάλογα με το εάν ο χρήστης είναι συνδεδεμένος, όπως για σύνδεση/αποσύνδεση, εμφάνιση μηνυμάτων κι επεξεργασία στοιχείων λογαριασμού.

`footer`: Υποσέλιδο με links στις σελίδες About και Contact

`_shim`: Partial που περιέχει σύνδεσμο σε script για σωστή εμφάνιση της σελίδας σε εκδόσεις του Internet Explorer browser όπου υπάρχουν ζητήματα συμβατότητας.

4. Χαρακτηριστικά της εφαρμογής

Σε αυτό το κεφάλαιο θα αναλυθεί η λειτουργικότητα της εφαρμογής, παρουσιάζοντας με σενάρια χρήσης τις λειτουργίες που έχουν στη διάθεσή τους οι χρήστες. Αρχικά θα εξεταστεί το κομμάτι που απευθύνεται στους απλούς χρήστες και στη συνέχεια το administration backend που προσφέρει το Active Admin gem.

Οι χρήστες χωρίζονται σε δύο κατηγορίες, απλοί και administrators. Οι administrators μπορούν να είναι και απλοί χρήστες της εφαρμογής, αλλά έχουν αποκλειστικά δικαιώματα πρόσβασης στο administration backend του Active Admin.

Οι βασικές λειτουργίες της εφαρμογής για τους απλούς χρήστες είναι:

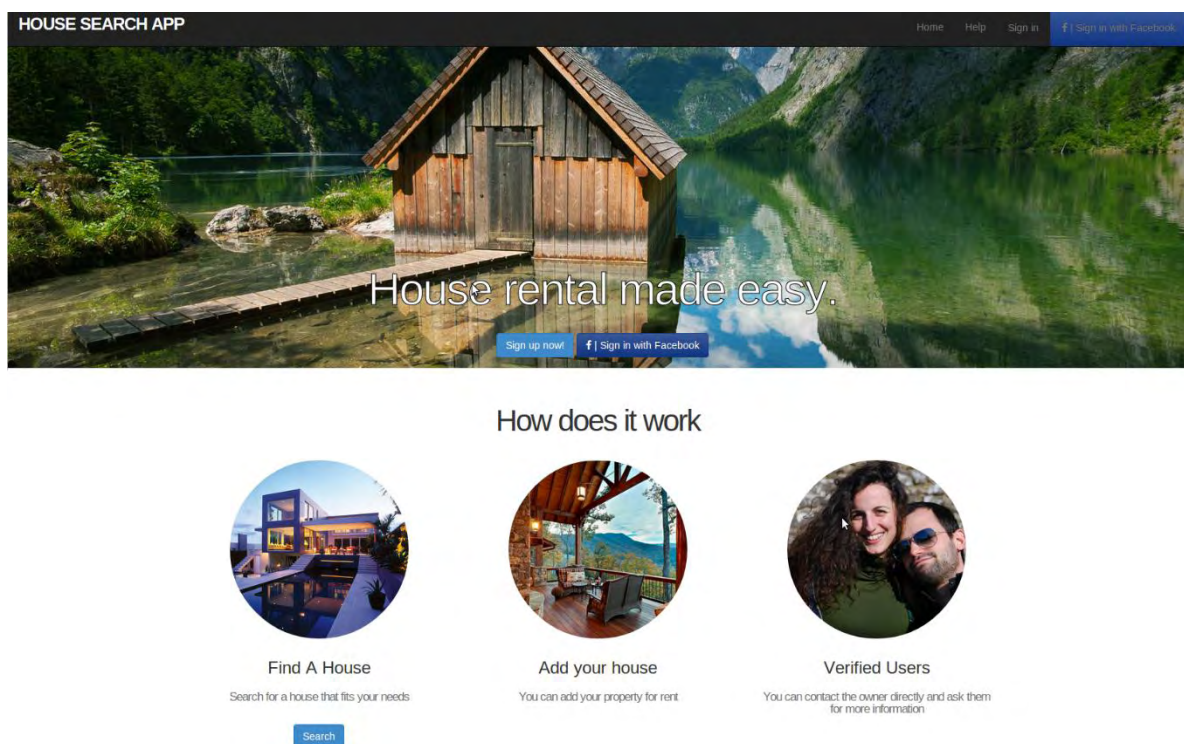
- Αναζήτηση σπιτιού
- Καταχώρηση σπιτιού προς ενοικίαση
- Προβολή σπιτιού με εικόνες, χάρτη τοποθεσίας και πληροφορίες
- Επικοινωνία με τον ιδιοκτήτη μέσω προσωπικών μηνυμάτων
- Διαχείριση συνομιλιών με inbox, sentbox, trash φακέλους

Οι χρήστες με δικαιώματα διαχειριστή μπορούν επιπλέον των παραπάνω να χρησιμοποιούν το Active Admin backend ώστε να:

- Παρακολουθούν τις εγγραφές χρηστών
- Ενημερώνουν τα στοιχεία των χρηστών
- Διαγράφουν χρήστες
- Παρακολουθούν τις καταχωρήσεις σπιτιών
- Ενημερώνουν τις πληροφορίες των σπιτιών
- Διαγράφουν σπίτια
- Παρακολουθούν/διαγράφουν τις κριτικές-σχόλια που αφήνουν οι χρήστες στη σελίδα κάποιου σπιτιού

4.1 Λειτουργίες εφαρμογής (User side)

Κατά την είσοδο του χρήστη στην εφαρμογή, η κεντρική οθόνη από την οποία είναι δυνατή η εγγραφή/σύνδεση αλλά και αναζήτηση σπιτιών είναι η εξής:



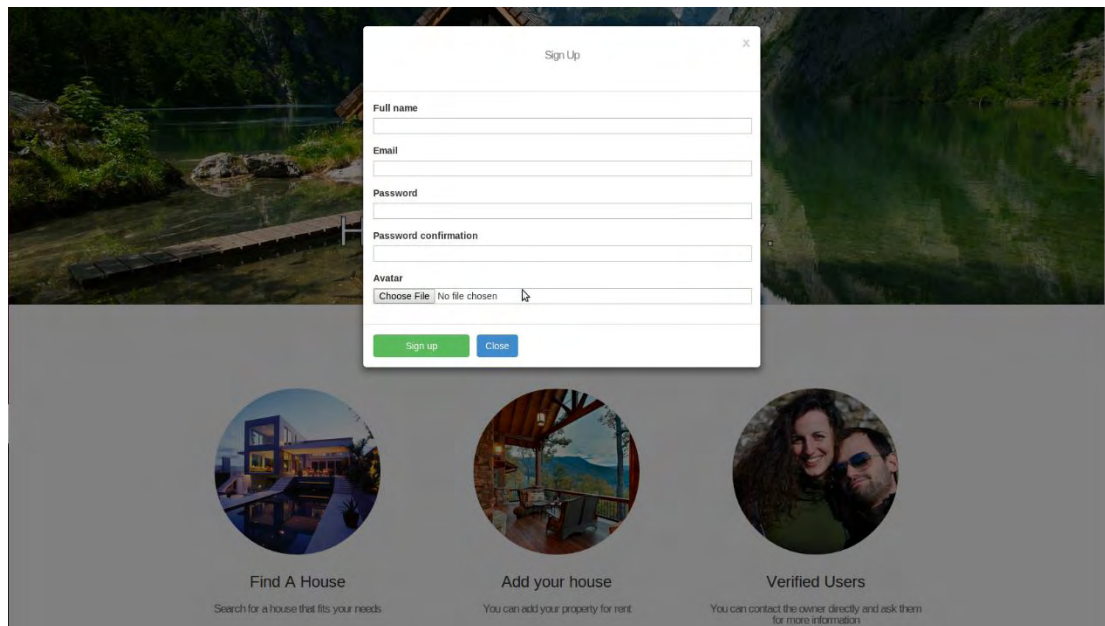
Εικόνα 4: Κεντρική σελίδα

4.1.1 Εγγραφή χρήστη

Για την καταχώρηση σπιτιού και την επικοινωνία με τους ιδιοκτήτες είναι απαραίτητη η ύπαρξη λογαριασμού στην εφαρμογή. Η εγγραφή μπορεί να γίνει είτε μέσω email ή μέσω λογαριασμού στο κοινωνικό δίκτυο Facebook. Όπως φαίνεται παραπάνω στην κεντρική σελίδα υπάρχουν κουμπιά-σύνδεσμοι για την εγγραφή ενός νέου χρήστη.

Εγγραφή με email

Πατώντας το κουμπί “Sign up now!” εμφανίζεται το παρακάτω παράθυρο, στο οποίο ο χρήστης καλείται να καταχωρήσει τα στοιχεία του καθώς και (προαιρετικά) μια εικόνα/avatar:



Εικόνα 5: Sign up via email

Εφόσον η φόρμα συμπληρωθεί χωρίς λάθη, ο χρήστης εγγράφεται επιτυχώς και μπορεί πλέον να χρησιμοποιήσει την εφαρμογή με πλήρη λειτουργικότητα.

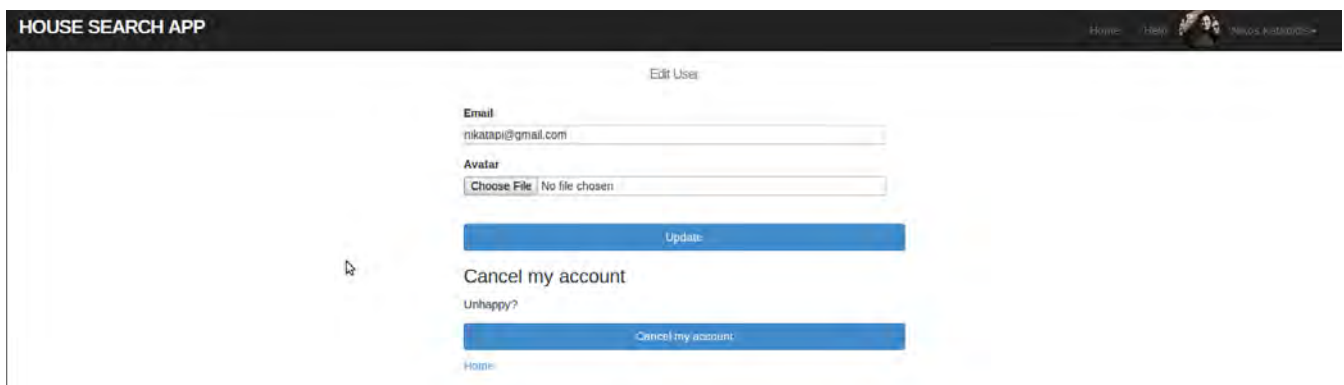
Εγγραφή/Σύνδεση μέσω Facebook

Για τη σύνδεση μέσω Facebook, ο χρήστης ανακατευθύνεται στην ιστοσελίδα του Facebook απ' όπου ζητείται η άδεια του για χρήση των στοιχείων του από την εφαρμογή. Στη συνέχεια ανακατευθύνεται πίσω στην εφαρμογή.

4.1.2 Επεξεργασία προφίλ

Η αλλαγή στοιχείων του χρήστη γίνεται ακολουθώντας το σύνδεσμο που υπάρχει στο dropdown menu του navigation bar που βρίσκεται στο header της εφαρμογής. Ανάλογα με τον τρόπο εγγραφής του χρήστη, η οθόνη επεξεργασίας προφίλ είναι διαφορετική.

Για λογαριασμό μέσω Facebook:



Εικόνα 6: Επεξεργασία προφίλ (Facebook)

Για λογαριασμό μέσω email:

HOUSE SEARCH APP

Home Help Νίκος Φίλιππος

Edit User

Email

Password (leave blank if you don't want to change it)

Password confirmation

Current password (we need your current password to confirm your changes)

Avatar

Choose File No file chosen

Update

Cancel my account

Unhappy?

Cancel my account

home

Εικόνα 7: Επεξεργασία προφίλ (email)

4.1.3 Αναζήτηση σπιτιού

Ακολουθώντας το σύνδεσμο “search” της κεντρικής σελίδας, ο χρήστης οδηγείται στην οθόνη αναζήτησης, όπου εμφανίζεται η εξής φόρμα:

HOUSE SEARCH APP

Home Help Νίκος Καραπίδης

Search

House Locations

Country

Please select a country

City

Address

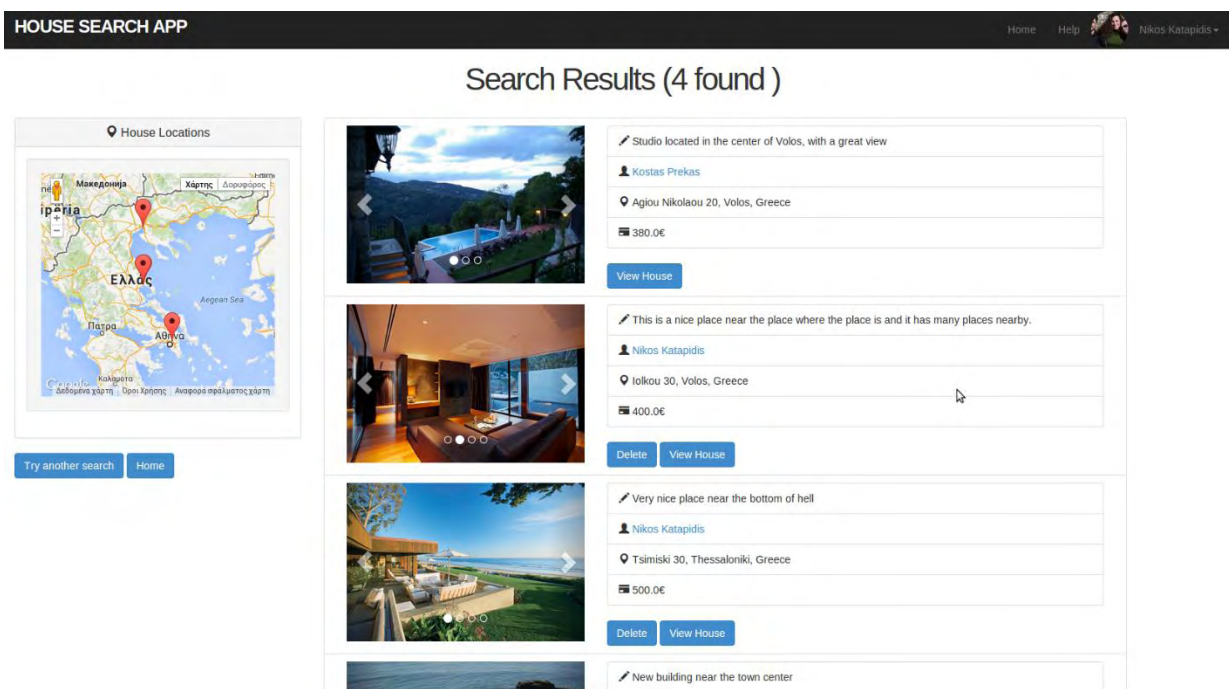
Max price

Max distance from given address


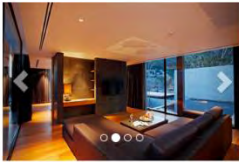


Search!

Εικόνα 8: Αναζήτηση σπιτιού

Το πεδίο Country είναι το μόνο υποχρεωτικό κατά την αναζήτηση, ενώ συμπληρώνοντας τα υπόλοιπα η αναζήτηση γίνεται πιο ακριβής με βάση τα εισαγόμενα πεδία. Ο χάρτης στην αριστερή πλευρά της σελίδας δείχνει τις περιοχές όπου έχουν καταχωρηθεί σπίτια προς ενοικίαση. Μια επιτυχής αναζήτηση οδηγεί στην οθόνη εμφάνισης αποτελεσμάτων όπου αναγράφονται τα στοιχεία κάθε αποτελέσματος και σύνδεσμοι επίσκεψης της σελίδας εμφάνισής τους, όπως φαίνεται παρακάτω:



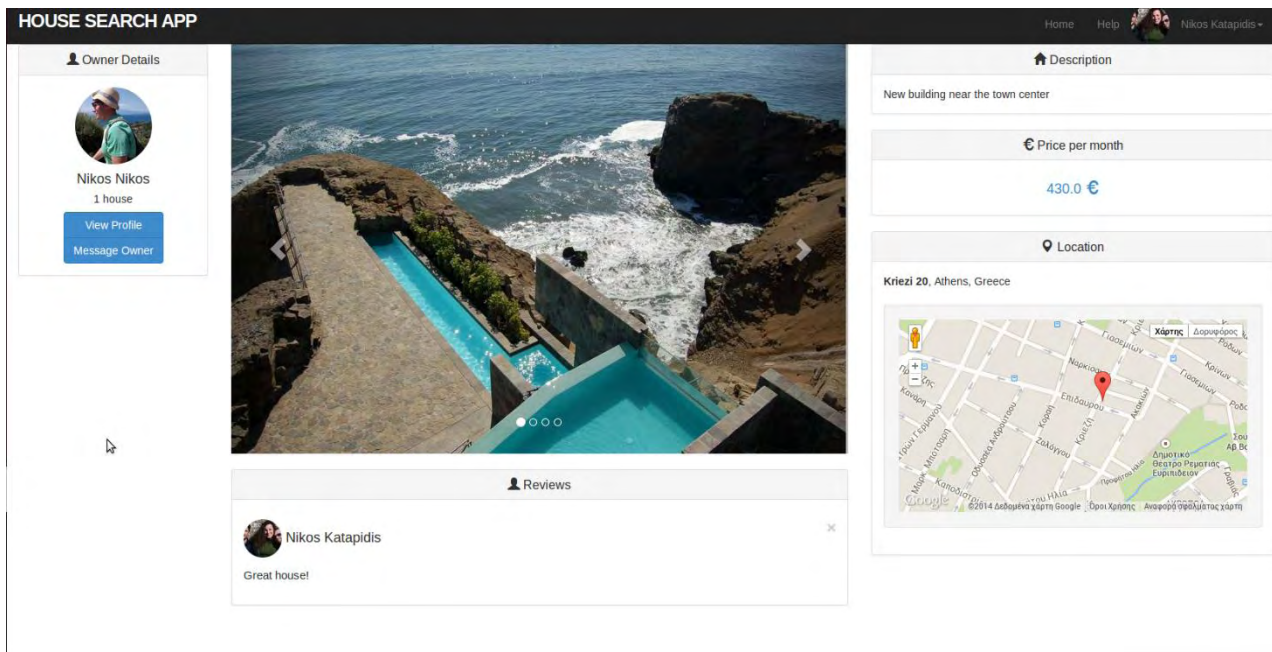
The screenshot displays the 'HOUSE SEARCH APP' interface. At the top, there is a navigation bar with 'Home', 'Help', and the user's name 'Nikos Katapidis'. The main heading is 'Search Results (4 found)'. On the left, a map titled 'House Locations' shows the search area in Greece, with red pins indicating property locations. Below the map are buttons for 'Try another search' and 'Home'. The search results are listed in a grid. Each result includes a thumbnail image, a description, the user's name, the address, the price, and buttons for 'View House' or 'Delete'.

Image	Description	User	Address	Price	Buttons
	Studio located in the center of Volos, with a great view	Kostas Prekas	Agίου Nikolaou 20, Volos, Greece	380.0€	View House
	This is a nice place near the place where the place is and it has many places nearby.	Nikos Katapidis	Iolkou 30, Volos, Greece	400.0€	Delete View House
	Very nice place near the bottom of the hell	Nikos Katapidis	Tsimiski 30, Thessaloniki, Greece	500.0€	Delete View House
	New building near the town center				

Εικόνα 9: Αποτελέσματα αναζήτησης

4.1.4 Προβολή καταχωρημένης οικίας

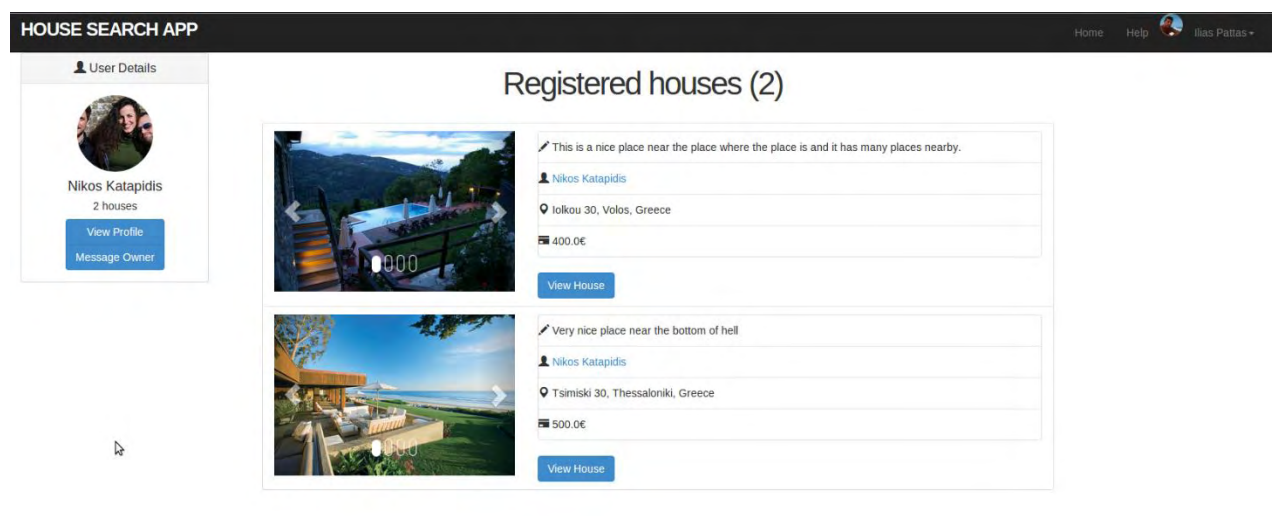
Ο χρήστης μπορεί να επισκεφθεί τη σελίδα ενός σπιτιού είτε μετά από μια επιτυχημένη αναζήτηση είτε μέσα από το προφίλ του ιδιοκτήτη. Στο αριστερό μέρος υπάρχουν πληροφορίες για τον ιδιοκτήτη, σύνδεσμος για επίσκεψη το προφίλ του καθώς και σύνδεσμος για αποστολή μηνύματος. Στο κέντρο εμφανίζονται οι εικόνες που έχει καταχωρήσει ο ιδιοκτήτης, όπως και κριτικές χρηστών. Στο αριστερό μέρος εμφανίζονται πληροφορίες σχετικά με το σπίτι όπως περιγραφή, τιμή και τοποθεσία. Η σελίδα προβολής ενός σπιτιού είναι η εξής:



Εικόνα 10: Σελίδα σπιτιού

4.1.5 Προβολή προφίλ χρήστη

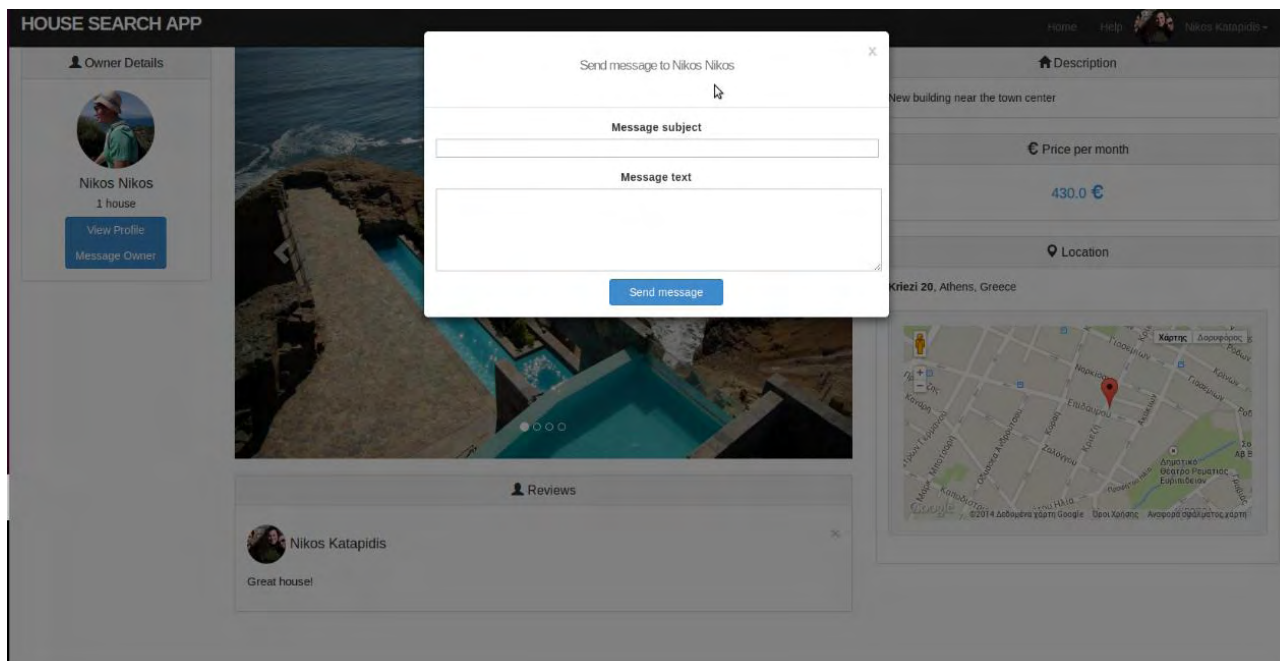
Από τη σελίδα εμφάνισης των αποτελεσμάτων μιας αναζήτησης ή τη σελίδα προβολής καταχωρημένης οικίας, υπάρχουν σύνδεσμοι προς το προφίλ του ιδιοκτήτη όπου προβάλλονται πληροφορίες και δίνεται η δυνατότητα επικοινωνίας μαζί του μέσω προσωπικών μηνυμάτων.



Εικόνα 11: Προφίλ χρήστη

4.1.6 Αποστολή μηνύματος

Η αποστολή μηνύματος προς ένα χρήστη είναι δυνατή είτε από τη σελίδα εμφάνισης του προφίλ του, είτε από τη σελίδα προβολής κάποιας οικίας που έχει καταχωρήσει. Επιλέγοντας το σύνδεσμο “Message Owner” εμφανίζεται ένα αναδυόμενο παράθυρο το οποίο αποτελεί φόρμα υποβολής μηνύματος, όπως φαίνεται στο παρακάτω στιγμιότυπο:

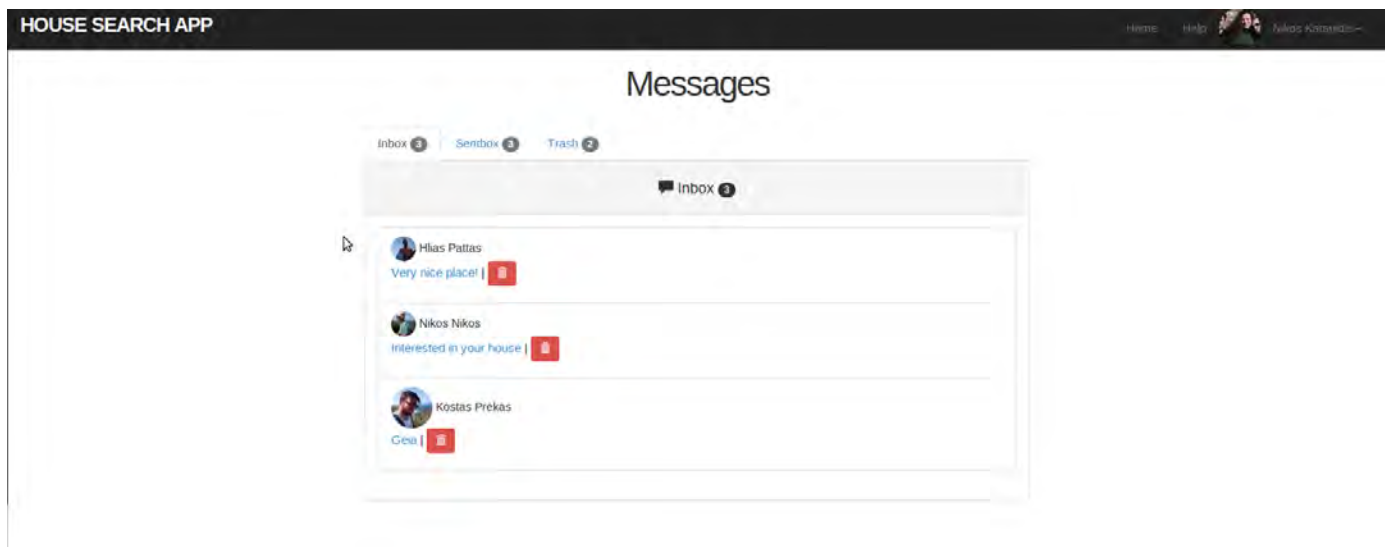


Εικόνα 12: Αποστολή προσωπικού μηνύματος χρήστη

Μετά την υποβολή του μηνύματος, ο χρήστης ανακατευθύνεται στη σελίδα εμφάνισης των προσωπικών μηνυμάτων του.

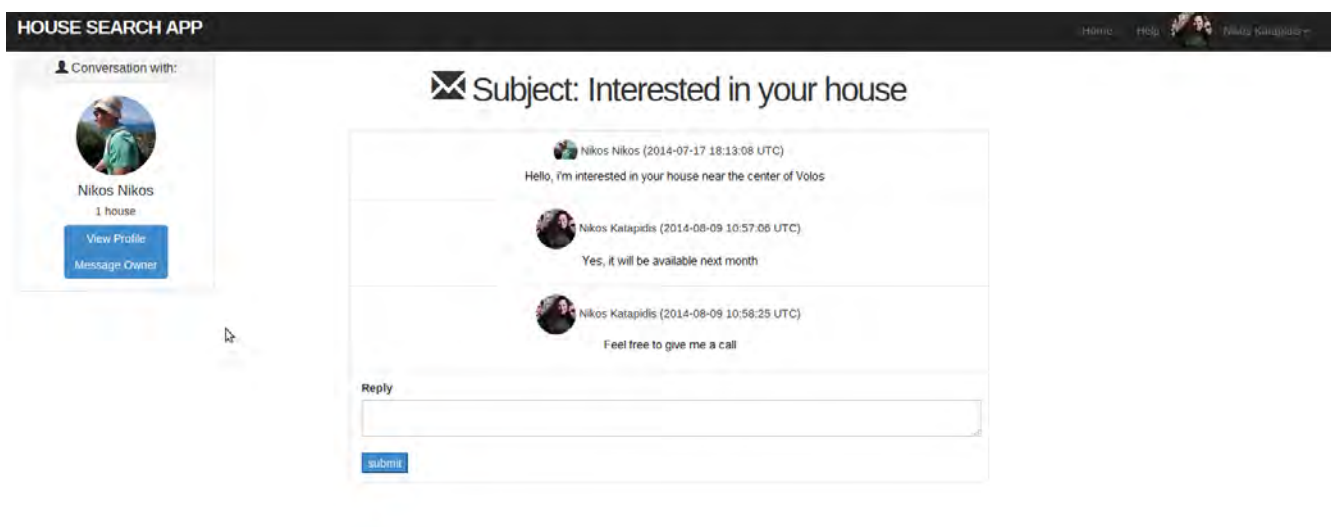
4.1.7 Προβολή μηνυμάτων χρήστη

Για την προβολή και διαχείριση των συζητήσεων-μηνυμάτων με άλλους χρήστες, έχει δημιουργηθεί μια κεντρική οθόνη που περιέχει τρεις καρτέλες (Inbox, Sentbox, Trash). Από εκεί, υπάρχει η δυνατότητα ανάγνωσης των συνομιλιών, αποστολής μηνυμάτων στο Trash καθώς και οριστικής διαγραφής.



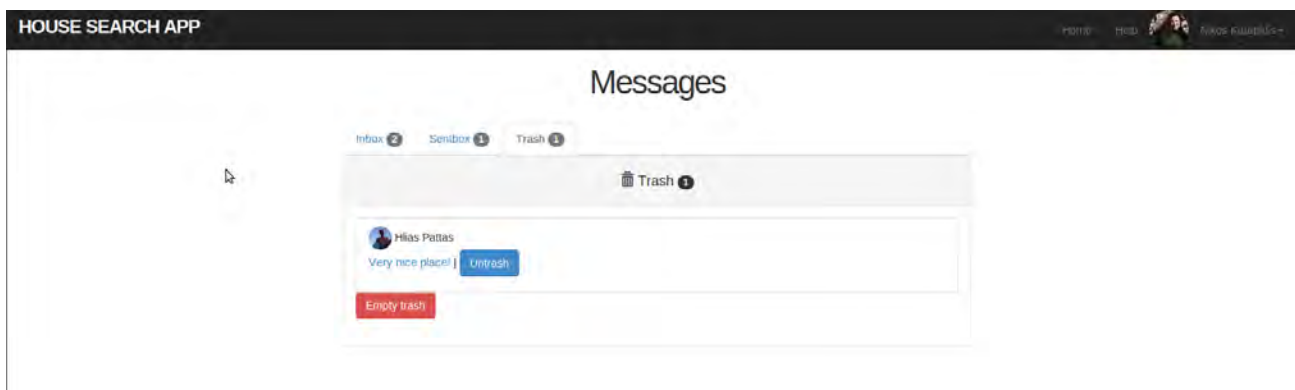
Εικόνα 13: Εισερχόμενα μηνύματα

Επιλέγοντας μια ενεργή συνομιλία, οδηγούμαστε στην οθόνη ανάγνωσης συζήτησης, απ' όπου μπορούμε να διαβάσουμε τα μηνύματα και να απαντήσουμε στη συνομιλία.



Εικόνα 14: Προβολή συνομιλίας

Τα μηνύματα που στέλνονται στο trash μπορούν να επανέλθουν στην προηγούμενη κατάσταση (inbox, sentbox) ή να διαγραφούν οριστικά, όπως φαίνεται παρακάτω:



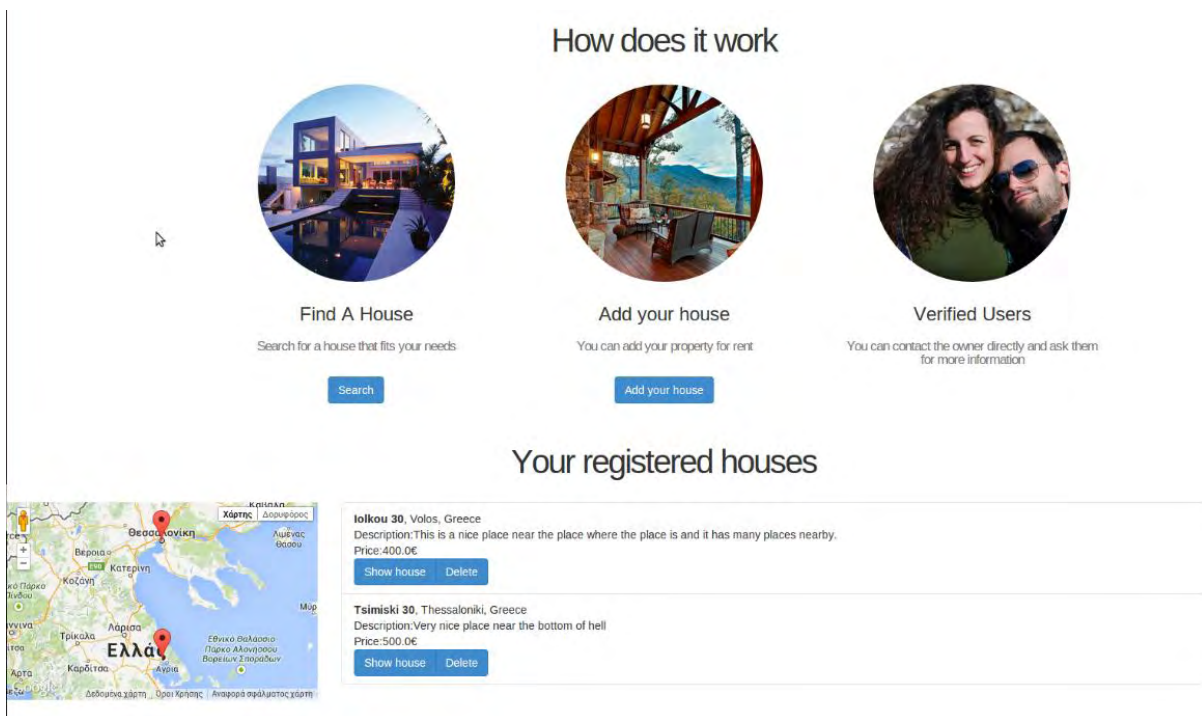
Εικόνα 15: Trash

4.1.8 Καταχώρηση οικίας

Μετά την επιτυχή εγγραφή του ο χρήστης μπορεί να προσθέσει σπίτια προς ενοικίαση, συμπληρώνοντας την παρακάτω φόρμα:

Εικόνα 16: Καταχώρηση σπιτιού

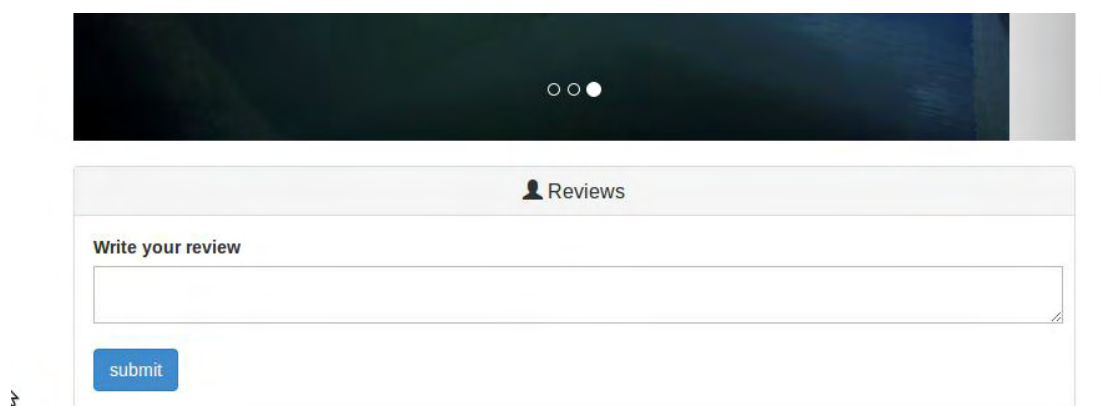
Μετά την καταχώρηση σπιτιών από το χρήστη, η κεντρική σελίδα της εφαρμογής ανανεώνεται προσθέτοντας όσα σπίτια του ανήκουν, όπως φαίνεται στην επόμενη εικόνα:



Εικόνα 17: Τροποποιημένη κεντρική σελίδα

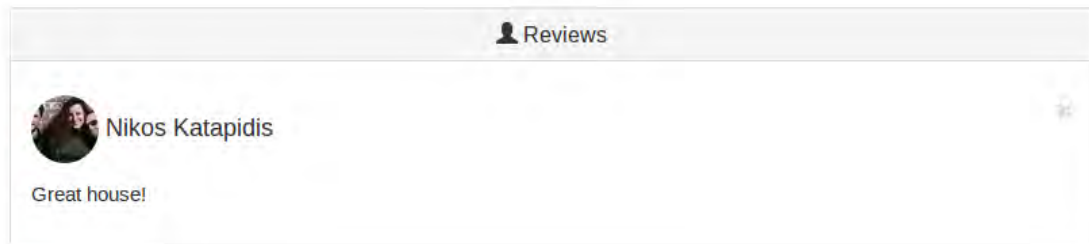
4.1.9 Καταχώρηση κριτικής

Στη σελίδα προβολής μιας οικίας, είναι δυνατή η προσθήκη κριτικής, η οποία γίνεται μέσω Ajax [22] φόρμας και τροποποίηση της σελίδας μέσω κώδικα coffeescript ώστε να μην χρειάζεται ανανέωση της σελίδας. Επίσης είναι δυνατή η αφαίρεση της κριτικής του χρήστη με το πάτημα ενός κουμπιού.



Εικόνα 18: Φόρμα καταχώρησης κριτικής

Μετά την καταχώρηση της κριτικής, το παραπάνω τμήμα της σελίδας τροποποιείται με την εξής προσθήκη:



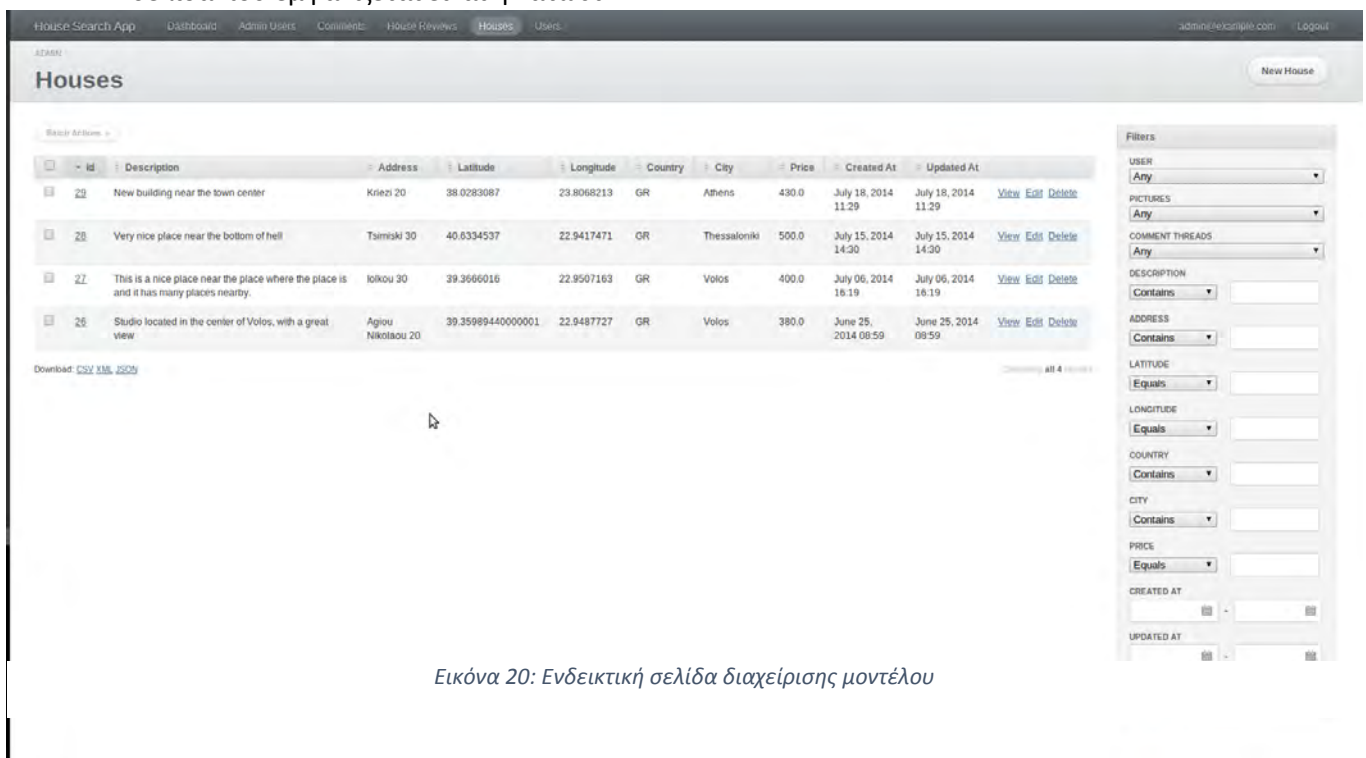
Εικόνα 19: Εμφάνιση κριτικής

4.2 Administration Interface (Active Admin)

Το Active Admin gem παρέχει ένα ολοκληρωμένο περιβάλλον διαχείρισης της εφαρμογής, όπου ο administrator μπορεί να επιβλέπει τα resources που καταχωρούνται από τους χρήστες, να τροποποιεί τα στοιχεία τους και να διαγράφει ό,τι είναι απαραίτητο για την ορθή λειτουργία (για παράδειγμα ψεύτικες καταχωρήσεις σπιτιών). Κάθε μοντέλο μπορεί να καταχωρηθεί ως διαχειρίσιμος πόρος, με τη βοήθεια της εντολής:

```
rails generate active_admin:resource [ModelName]
```

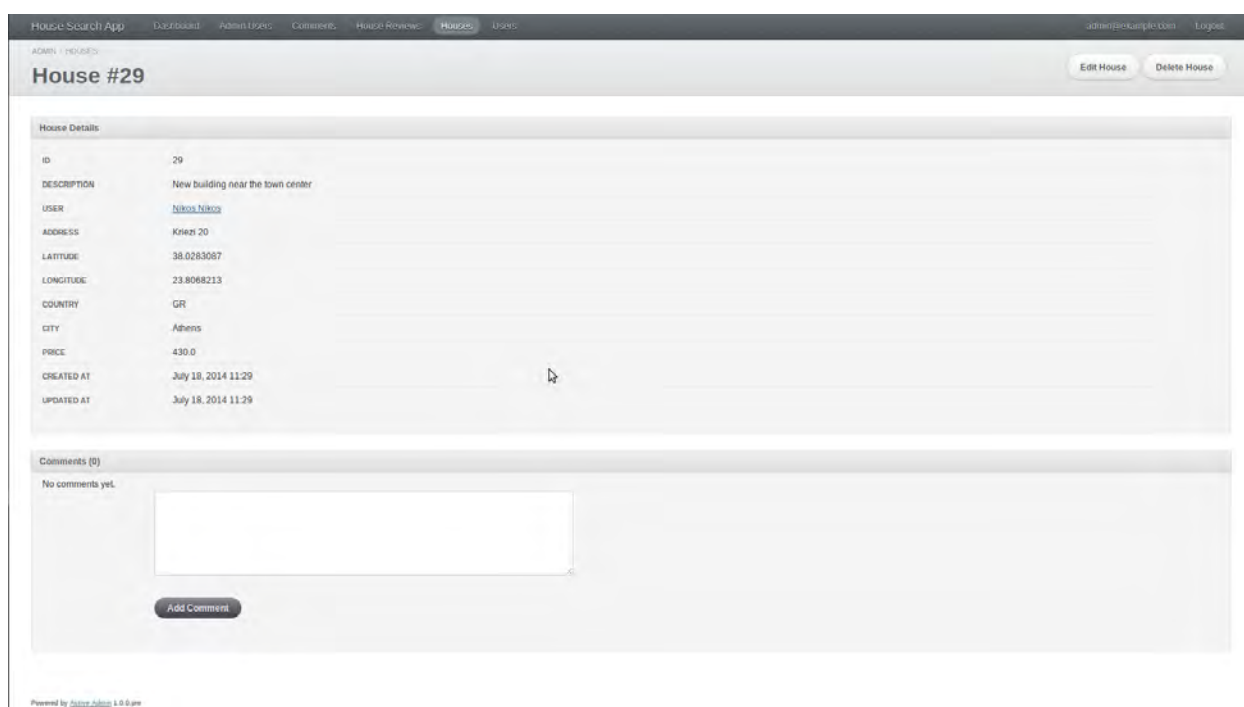
Για κάθε μοντέλο που προστίθεται δημιουργείται αυτόματα μια καρτέλα από την οποία ο administrator μπορεί να δει μια λίστα με τις καταχωρήσεις, ενώ μπορεί να φιλτράρει τα αποτελέσματα με βάση κάποιο attribute του μοντέλου. Για το μοντέλο House η αντίστοιχη σελίδα που εμφανίζεται είναι η κάτωθι:



Εικόνα 20: Ενδεικτική σελίδα διαχείρισης μοντέλου

4.2.1 Εμφάνιση εγγραφής

Από την οθόνη εμφάνισης της λίστας όλων των καταχωρήσεων ενός μοντέλου, μπορεί να επιλεγεί μια συγκεκριμένη καταχώρηση, πράξη που οδηγεί στην οθόνη εμφάνισης λεπτομερειών με αναλυτική καταγραφή των attributes, με δυνατότητα προσθήκης σχολίων, επεξεργασίας των attributes και διαγραφής της καταχώρησης.



Εικόνα 21: Προβολή καταχώρησης

4.2.2 Επεξεργασία εγγραφής

Ο διαχειριστής δύναται να επεξεργαστεί τα στοιχεία μιας καταχώρησης εάν αυτό καθίσταται απαραίτητο για την ομαλή λειτουργία της εφαρμογής. Το Active Admin παρέχει την κατάλληλη φόρμα μέσω της οποίας μπορεί να γίνει η αλλαγή των attributes για μια εγγραφή, όπως φαίνεται παρακάτω:

House Search App Dashboard Admin Users Comments House Reviews Houses Users admin@example.com Logout

ADMIN > HOUSES > (HOUSE 678)

Edit House

User	<input type="text" value="Nikos Nikos"/>
Description*	<input type="text" value="New building near the town center"/>
Address*	<input type="text" value="Kriez 20"/>
Latitude	<input type="text" value="38.0283067"/>
Longitude	<input type="text" value="23.8068213"/>
Country*	<input type="text" value="Greece"/>
City*	<input type="text" value="Athens"/>
Price*	<input type="text" value="430.0"/>

Powered by [Admin Editor](#) 1.0.0.0

Εικόνα 22: Επεξεργασία καταχώρησης

5. Μελλοντικές Επεκτάσεις

Η εφαρμογή διαθέτει μια βασική λειτουργικότητα για να αποτελέσει ένα χρήσιμο εργαλείο εύρεσης σπιτιού προς ενοικίαση στα πλαίσια της ακαδημαϊκής κοινότητας. Η ανάπτυξη της βρίσκεται σε αρχικό στάδιο και η επιλογή του framework Rails διευκολύνει την περαιτέρω υλοποίηση επεκτάσεων που θα την έκαναν ακόμη πιο παραμετροποιήσιμη και χρηστική.

Παρακάτω παρουσιάζονται μερικές μελλοντικές επεκτάσεις που θα μπορούσαν να διευρύνουν τις λειτουργίες της εφαρμογής:

- *Σύστημα online κρατήσεων:* Οι χρήστες θα μπορούν να κλείνουν μέσω διαδικτύου ένα διαθέσιμο σπίτι, βλέποντας τις διαθέσιμες ημερομηνίες κι επιλέγοντας αυτές που επιθυμούν.
- *Online πληρωμή προκαταβολής:* Για την ενοικίαση ενός σπιτιού η καταβολή προκαταβολής θα μπορεί να γίνεται μέσω ηλεκτρονικής πληρωμής με τη χρήση online banking.
- *Προσθήκη κατηγοριών για την περιγραφή των σπιτιών:* Με τη βοήθεια ενός tagging system τα σπίτια θα χωρίζονται σε κατηγορίες που θα διευκολύνουν την αναζήτηση.
- *Δυνατότητα αναφοράς κακόβουλων χρηστών/καταχωρήσεων:* Οι χρήστες θα μπορούν να αναφέρουν τυχόν κακόβουλη συμπεριφορά ή καταχωρήσεις που δεν ανταποκρίνονται στην πραγματικότητα, ώστε να αφαιρούνται από τους διαχειριστές.
- *Ενσωμάτωση καταχωρήσεων από άλλες πλατφόρμες:* Με τη χρήση web crawlers [24] είναι δυνατή η συλλογή καταχωρήσεων σπιτιών από άλλες διαδικτυακές εφαρμογές κι ενσωμάτωσής τους στη βάση δεδομένων της εφαρμογής.
- *Στατιστική ανάλυση χρήσης:* Τα δεδομένα των αναζητήσεων των χρηστών μπορούν να χρησιμοποιηθούν για βελτίωση των αποτελεσμάτων αναζήτησης και επιπλέον παραμετροποίηση της εφαρμογής στις ανάγκες του κάθε χρήστη.

Αναφορές

- [1] Ubuntu operating system - <http://www.ubuntu.com/>
- [2] Ruby programming language - <https://www.ruby-lang.org/en/>
- [3] Ruby on Rails - <http://rubyonrails.org/>
- [4] SQLite - <http://www.sqlite.org/>
- [5] Sublime Text - <http://www.sublimetext.com/>
- [6] Git version control system - <http://git-scm.com/>
- [7] Github - <https://github.com/>
- [8] Web application framework - http://en.wikipedia.org/wiki/Web_application_framework
- [9] RubyGems - <http://rubygems.org/>
- [10] Sass - <http://sass-lang.com/>
- [11] Coffeescript - <http://coffeescript.org/>
- [12] JQuery - <http://jquery.com/>
- [13] CSS - http://en.wikipedia.org/wiki/Cascading_Style_Sheets
- [14] Rack - [http://en.wikipedia.org/wiki/Rack_\(web_server_interface\)](http://en.wikipedia.org/wiki/Rack_(web_server_interface))
- [15] API - http://en.wikipedia.org/wiki/Application_programming_interface
- [16] Front End - <http://whatis.techtarget.com/definition/front-end>
- [17] Bootstrap - <http://getbootstrap.com/>
- [18] Cloud Storage - <http://searchcloudstorage.techtarget.com/definition/cloud-storage>
- [19] Avatar - [http://en.wikipedia.org/wiki/Avatar_\(computing\)](http://en.wikipedia.org/wiki/Avatar_(computing))
- [20] URL - http://en.wikipedia.org/wiki/Uniform_resource_locator
- [21] Cross-site request forgery - http://en.wikipedia.org/wiki/Cross-site_request_forgery
- [22] Ajax - [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))
- [23] Sass-Rails - <https://github.com/rails/sass-rails>

- [24] Devise - <https://github.com/plataformatec/devise>
- [25] Omniauth - <https://github.com/intridea/omniauth>
- [26] Omniauth-Facebook - <https://github.com/mkdynamic/omniauth-facebook>
- [27] Bootstrap-Sass - <https://github.com/twbs/bootstrap-sass>
- [28] Will_paginate - https://github.com/mislav/will_paginate
- [29] Will_paginate_bootstrap - https://github.com/bootstrap-ruby/will_paginate-bootstrap
- [30] Active Admin - <http://activeadmin.info/>
- [31] Carrierwave - <https://github.com/carrierwaveuploader/carrierwave>
- [32] Google Maps for Rails - <https://github.com/apneadiving/Google-Maps-for-Rails>
- [33] Ruby Geocoder - <https://github.com/alexreisner/geocoder>
- [34] mini_magick - https://github.com/probablycorey/mini_magick
- [35] rmagick - <https://github.com/rmagick/rmagick>
- [36] Mailboxer - <https://github.com/mailboxer/mailboxer>
- [37]acts_as_commentable_with_threading-
https://github.com/elight/acts_as_commentable_with_threading
- [38] Rails Tutorial - <http://www.railstutorial.org/>
- [39] Ajax comments with Rails - <http://twocentstudios.com/blog/2012/11/15/simple-ajax-comments-with-rails/>
- [40] Customizing the mailboxer gem - <http://jamestansley.com/2014/02/22/customizing-the-mailboxer-ruby-gem-2/>