[](http://www.icicte.org/Aegean_LOGO.gif)[AEGEAN - ICSD - eCommerce Technologies](http://www.linkedin.com/groups?home=&gid=3806028&trk=anet_ug_hm)

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

τμημα μηχανικων πληροφοριακων και επικοινωνιακων συστηματων

ΠΡΟΠΤΥΧΙΑΚΟ προγραμμα σπουδων -

ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ

ΕΡΓΑΣΤΗΡΙΟ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ



**Karlovasi Home –** Ενδιάμεσο Παραδοτέο Εργασίας Τεχνολογίας Λογισμικού

Γιώργος Τσιρίδης

321/2013187, icsd13187@aegean.gr

Ιωάννης Σταύρου

321/2014190, icsd14190@aegean.gr

ΦΕΒΡΟΥΑΡΙΟΣ 2019

www.icsd.aegean.gr/is-lab

|  |  |
| --- | --- |
| <http://www.icicte.org/Aegean_LOGO.gif>  © Πανεπιστήμιο Αιγαίου, 2019  Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων  Εργαστήριο Πληροφοριακών Συστημάτων  83200 Καρλόβασι, Σάμος  Εργασία υλοποιημένη στα πλαίσια του μαθήματος:  Τεχνολογία Λογισμικού | |
| Ομάδα Ανάπτυξης: | Γιώργος Τσιρίδης  321/2013187, Email: icsd13187@aegean.gr  Ιωάννης Σταύρου  321/2014190, Email: icsd14190@aegean.gr |
| Επιβλέπων καθηγητής: | Χαραλαμπίδης Ιωάννης, Αναπληρωτής Καθηγητής  [yannisx@aegean.gr](mailto:yannisx@aegean.gr) |

**Περιεχόμενα**

[**Σύνοψη Εφαρμογής** 4](#_Toc304890)

[**Οργάνωση του Έργου** 5](#_Toc304891)

[**Use Cases/Usage Scenarios** 6](#_Toc304892)

[**Ανάλυση Απαιτήσεων** 7](#_Toc304893)

[**Λειτουργικές Απαιτήσεις** 7](#_Toc304894)

[**Μη Λειτουργικές Απαιτήσεις** 8](#_Toc304895)

[**Βασικά Διαγράμματα** 9](#_Toc304896)

[**Κύριες Οθόνες** 11](#_Toc304897)

[**Documentation Κώδικα** 13](#_Toc304898)

[**Testing Documentation** 14](#_Toc304899)

[**Εξωτερικές Πηγές** 14](#_Toc304900)

[**Συμπεράσματα** 14](#_Toc304901)

# **Σύνοψη Εφαρμογής**

Το Karlovasi Home, είναι μία mobile Android εφαρμογή η οποία αναπτύχθηκε με την βοήθεια των τεχνολογιών του **VISUAL STUDIO** **2017** και του **XAMARIN** που ως κύριο στόχο έχει να κάνει την διαδικασία εύρεσης κατοικίας εύκολη και γρήγορη, χρησιμοποιώντας κυρίως την τοποθεσία της συσκευής του χρήστη για την αναζήτηση αυτής.

Παράλληλα, προσφέρει μια απλή λύση στους χρήστες που επιθυμούν να καταχωρήσουν την δική τους ιδιοκτησία.

Πιο συγκεκριμένα, ο χρήστης που αναζητά κάποιο σπίτι, βλέπει τα διαθέσιμα οικήματα που βρίσκονται γύρω του εκείνη τη στιγμή, έχοντας την δυνατότητα αλλαγής των κριτηρίων αναζήτησης του ανά πάσα στιγμή. Έπειτα, μπορεί να επιλέξει κάποιο που τον ενδιαφέρει, να πληροφορηθεί για τα στοιχεία αυτού, να δει τις φωτογραφίες του, να επικοινωνήσει απευθείας με τον ιδιοκτήτη, καθώς και να δει το σημείο της ιδιοκτησίας πάνω σε χάρτη (όπως και την διαδρομή προς αυτό). Επιπρόσθετα, του δίνεται η δυνατότητα να προσθέσει κάποιο σπίτι στην λίστα αγαπημένων του, ώστε να έχει πρόσβαση σε αυτό οποιαδήποτε στιγμή επιθυμεί.

Από την άλλη μεριά, ο χρήστης που καταχωρεί ένα σπίτι προς ενοικίαση, αρχικά εγγράφεται και συνδέεται εύκολα στην εφαρμογή. Στη συνέχεια, καλείται να εισάγει τα στοιχεία της ιδιοκτησίας σε ένα απλό και εύχρηστο περιβάλλον, χωρίς πολύπλοκες και κουραστικές διαδικασίες για τον ίδιο. Τέλος, έχει την δυνατότητα να επεξεργάζεται τις καταχωρήσεις του και να παρακολουθεί την πορεία τους, καθώς και να χαρακτηρίζει σπίτια ως μη διαθέσιμα όταν αυτά έχουν ενοικιαστεί.

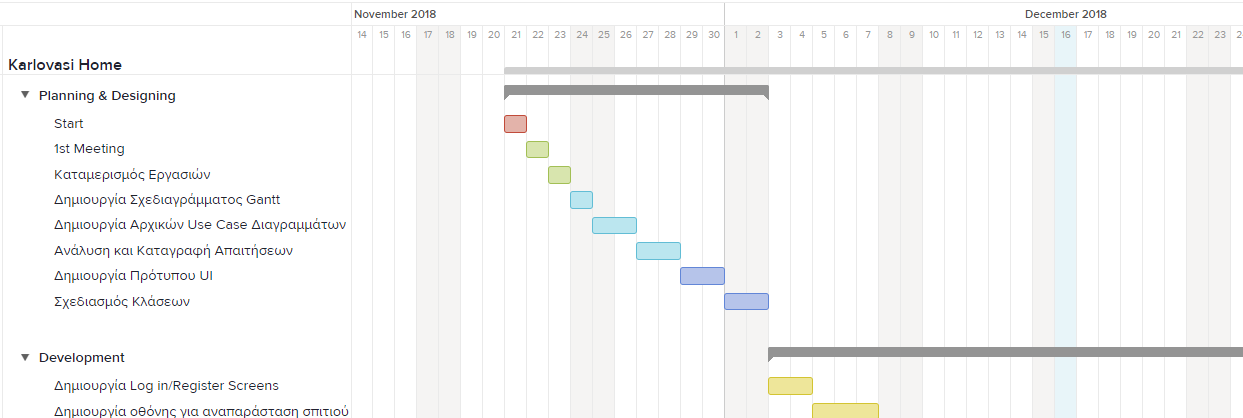
# **Οργάνωση του Έργου**

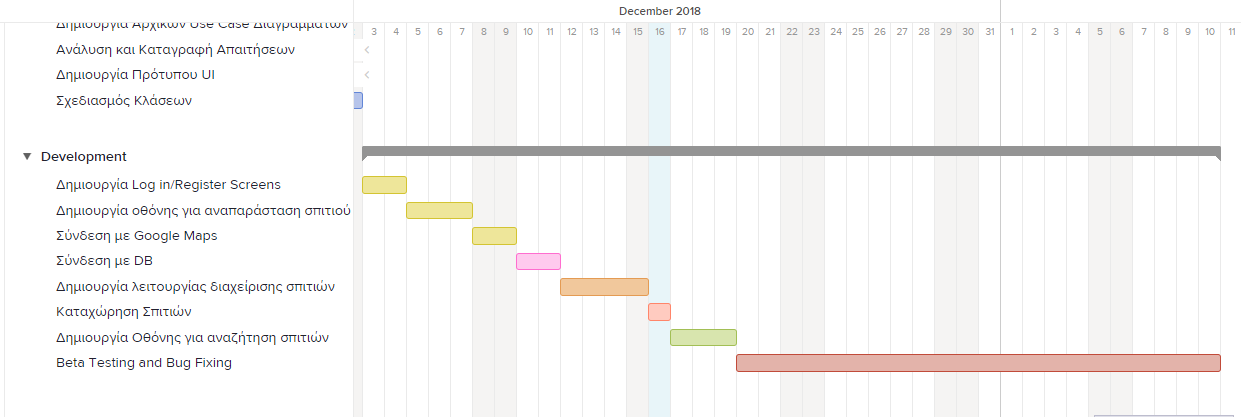
Η υλοποίηση του έργου ξεκίνησε στις 21/11/2018 και ως στόχος για την τελική του παράδοση τέθηκε η 10/1/2019. Χωρίστηκε σε 2 μεγάλες ομάδες, με την πρώτη να είναι αυτή της οργάνωσης και του σχεδιασμού, την οποία ακολούθησε η φάση της ανάπτυξης και του Testing.

Η ανάθεση των ρόλων έγινε ως εξής:

* Γιώργος Τσιρίδης – Project Manager, Planning, Design, Testing
* Ιωάννης Σταύρου – Developer, Database Administrator, Testing

**Gantt Chart**

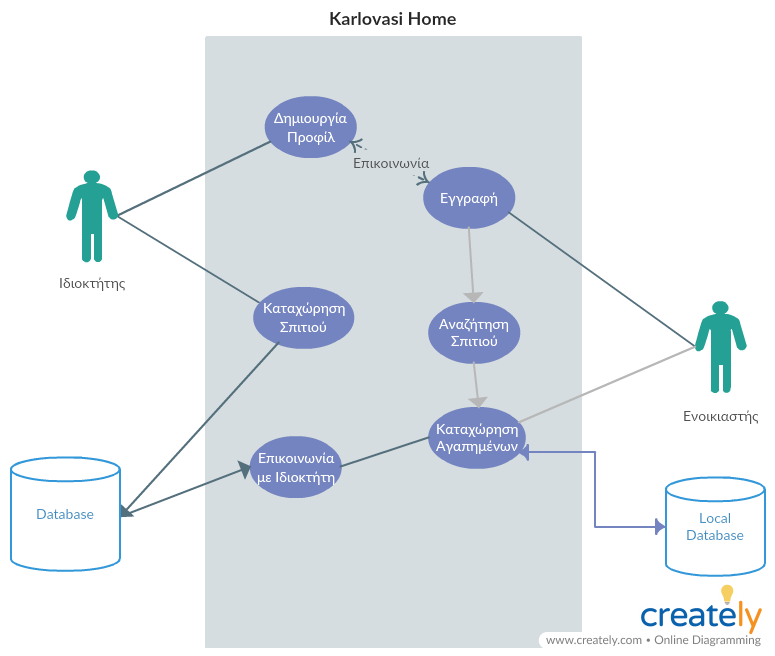




# **Use Cases/Usage Scenarios**

Στο βασικότερο σενάριο χρήσης της εφαρμογής, υπάρχουν χρήστες οι οποίοι αρχικά πρέπει να πραγματοποιήσουν την εγγραφή τους στην πλατφόρμα, είτε ως **ιδιοκτήτες**, είτε ως **ενοικιαστές**.

Οι ιδιοκτήτες μπορούν να καταχωρήσουν κάποιο σπίτι τους καθώς και να διαχειριστούν όλα τα σπίτια που έχουν ανεβασμένα στην εφαρμογή, και οι ενοικιαστές με την σειρά τους μπορούν να αναζητήσουν σπίτια με βάση συγκεκριμένα κριτήρια που οι ίδιοι επιθυμούν, να αποθηκεύσουν κάποια από αυτά στα αγαπημένα τους και να έρθουν εν τέλη σε επαφή με τους ιδιοκτήτες των εκάστοτε σπιτιών.



# **Ανάλυση Απαιτήσεων**

# **Λειτουργικές Απαιτήσεις**

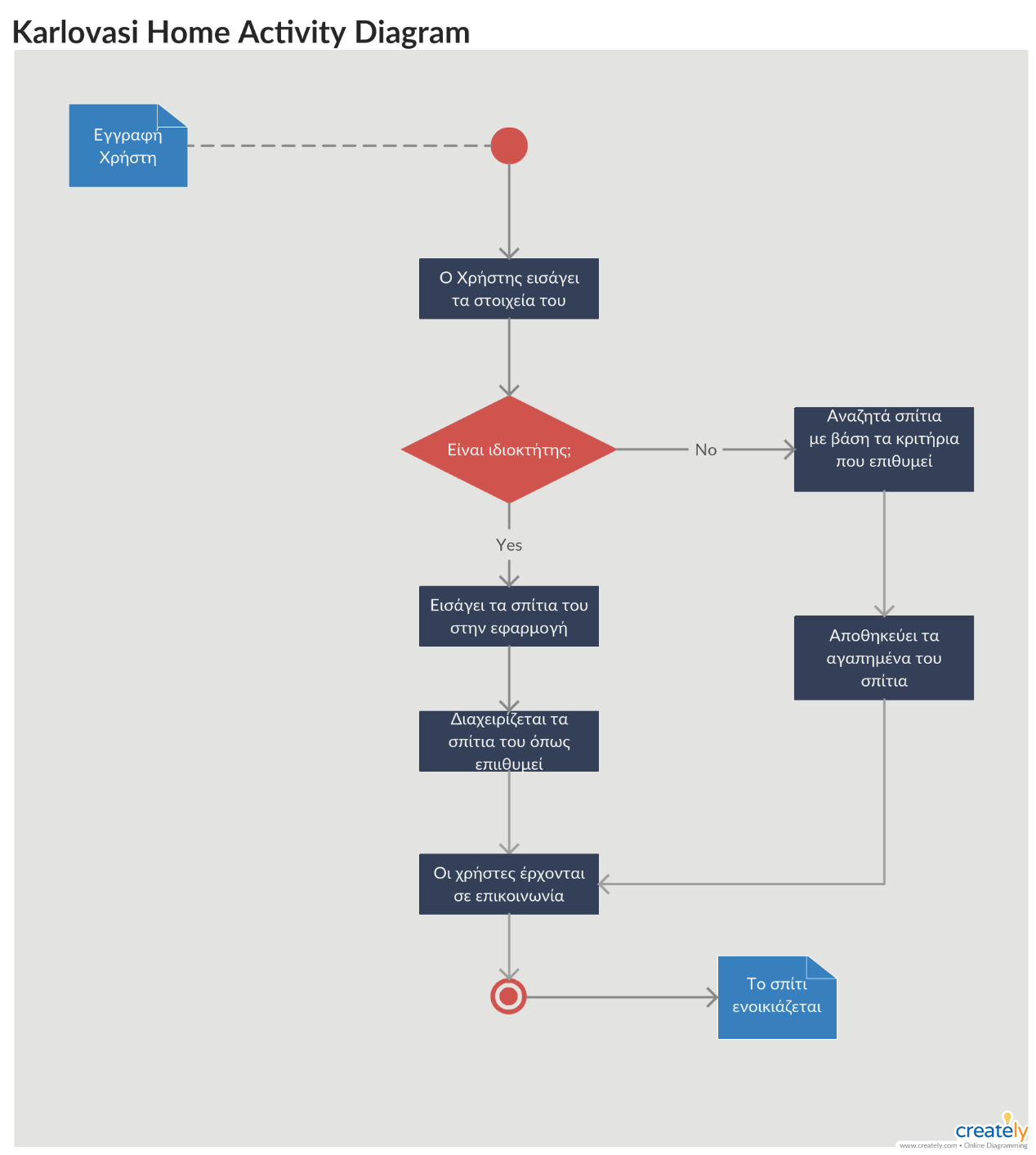
* Ο χρήστης θα πρέπει να έχει πρόσβαση στο διαδίκτυο για να χρησιμοποιήσει την εφαρμογή.
* Ο χρήστης θα πρέπει να έχει ενεργοποιημένη την δυνατότητα εύρεσης τοποθεσίας (GPS) της συσκευής του για να αναζητήσει ή να καταχωρήσει ένα σπίτι.
* Ο χρήστης θα πρέπει να διαθέτει λογαριασμό στην εφαρμογή και να συνδεθεί σε αυτόν για να καταχωρήσει ένα σπίτι.
* Η εφαρμογή θα πρέπει κατά την έναρξή της να εμφανίζει από προεπιλογή τα σπίτια που βρίσκονται κοντά στο χρήστη.
* Η εφαρμογή θα πρέπει να διαθέτει λειτουργία εύρεσης σπιτιών κοντά σε πανεπιστημιακά ιδρύματα.
* Η εφαρμογή θα πρέπει να διαθέτει μπάρα εργαλείων με επιλογές:
  + Καταχώρηση νέας ιδιοκτησίας.
  + Μετάβαση στη λίστα αγαπημένων.
  + Άλλες επιλογές που αφορούν το προφίλ του χρήστη και γενικότερα την εφαρμογή.
* Η εφαρμογή θα πρέπει να παρέχει στο χρήστη τη δυνατότητα αλλαγής κριτηρίων αναζήτησης.
* Η εφαρμογή θα πρέπει να παρέχει στο χρήστη τη δυνατότητα αποθήκευσης των αποτελεσμάτων που τον ενδιαφέρουν (Favorites) και κατ’ επέκταση την εύκολη πρόσβαση του στη λίστα αυτή.
* Ο χρήστης θα πρέπει να μπορεί:
  + Να μεταβεί στην οθόνη προβολής ιδιοκτησίας με tap στο σπίτι που τον ενδιαφέρει από την λίστα αποτελεσμάτων.
  + Να προσθέσει το σπίτι που βλέπει στην λίστα των αγαπημένων του.
  + Να δει το σημείο του σπιτιού πάνω σε χάρτη καθώς και την διαδρομή προς αυτό.
* Η εφαρμογή θα πρέπει να διαθέτει κατάλληλη οθόνη όπου:
  + Θα εμφανίζει τα στοιχεία του χρήστη (προφίλ).
  + Θα παρέχει δυνατότητα επεξεργασίας των στοιχείων του.
  + Θα εμφανίζει σε λίστα τα σπίτια που έχει καταχωρήσει ώστε να μπορεί είτε να τα επεξεργαστεί είτε να τα διαγράψει.

# **Μη Λειτουργικές Απαιτήσεις**

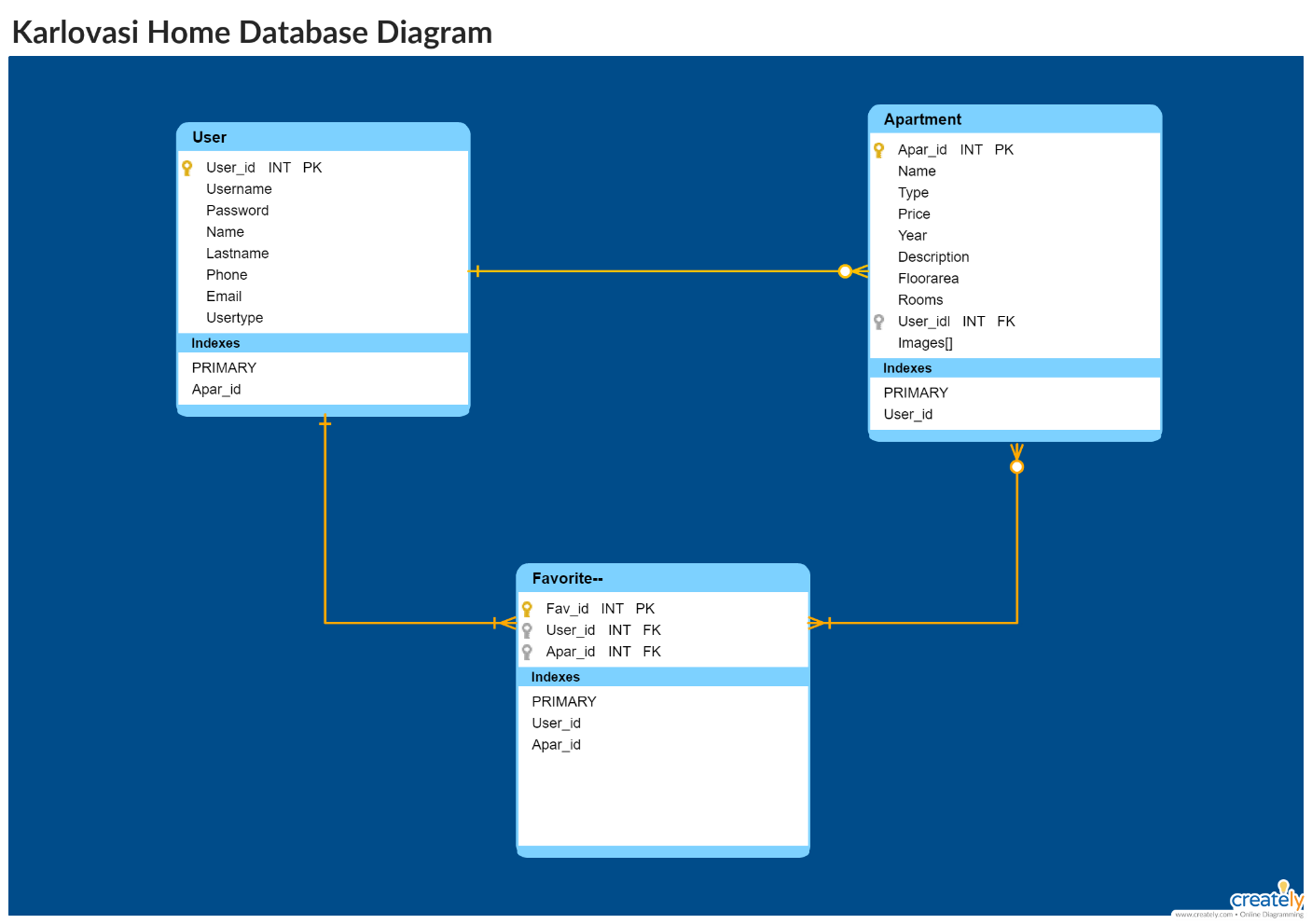
* Η εφαρμογή θα πρέπει να λειτουργεί σε συσκευές με λειτουργικό σύστημα Android ελάχιστης έκδοσης 4.1 (JellyBean, API Level 16) για την καλύτερη δυνατή λειτουργία αυτής και των υπηρεσιών της.
* Το Design του User Interface της εφαρμογής θα πρέπει να είναι μινιμαλιστικό, καλαίσθητο, με αλληλένδετα χρώματα και στοιχεία ώστε να μην κουράζει τον χρήστη.
* Η εφαρμογή θα πρέπει να παρέχει ένα άψογο User Experience στον χρήστη, με ευκολία στην χρήση όλων των λειτουργιών της, χωρίς περιττές διαδρομές και κείμενα.
* Για την δημιουργία, την διαχείριση και την πρόσβαση στη βάση δεδομένων, καθώς και ότι αφορά το backend, θα χρησιμοποιηθεί η πλατφόρμα Xamarin της Microsoft.
* Η εφαρμογή θα πρέπει να ενημερώνει τον χρήστη με κατάλληλο AlertDialog σε περίπτωση που η δυνατότητα εύρεσης τοποθεσίας (GPS) της συσκευής του είναι απενεργοποιημένη και να του δίνει την δυνατότητα να μεταβεί στις ρυθμίσεις αυτής.
* Η εφαρμογή θα πρέπει να εμφανίζει στον χρήστη κατάλληλο ProgressDialog κατά τη διάρκεια φόρτωσης δεδομένων από τον server ή αποστολής δεδομένων προς αυτόν.
* Η εφαρμογή θα πρέπει να ενημερώνει τον χρήστη με κατάλληλο Toast σε περίπτωση που υπήρξε σφάλμα κατά την αναζήτηση ή δεν βρέθηκαν αποτελέσματα με βάση τα κριτήρια που παρείχε.
* Η εφαρμογή θα πρέπει να ενημερώνει τον χρήστη με κατάλληλο Toast για τα αποτελέσματα της αναζήτησης (πλήθος αποτελεσμάτων και ακτίνα αναζήτησης).
* Η εφαρμογή θα πρέπει να διακόπτει την λειτουργία εύρεσης τοποθεσίας του χρήστη όταν δεν χρειάζεται για εξοικονόμηση μπαταρίας και μνήμης.
* Η ακτίνα αναζήτησης θα πρέπει να περιορίζεται απαραίτητα μεταξύ 1 και 10 χιλιομέτρων.
* Ο αριθμός των αποτελεσμάτων της αναζήτησης θα πρέπει να περιορίζεται στα 100.
* Ο αριθμός των αποθηκευμένων (αγαπημένων) σπιτιών του χρήστη στη αντίστοιχη λίστα θα πρέπει να περιορίζεται στα 20.
* Η λίστα αγαπημένων θα πρέπει να αποθηκεύεται τοπικά στη μνήμη της συσκευής του χρήστη και όχι στη βάση δεδομένων της εφαρμογής.
* Ο χάρτης απεικόνισης της τοποθεσίας ενός σπιτιού θα πρέπει να υλοποιηθεί με την χρήση του Google Maps API v2.
* Η εφαρμογή δεν θα πρέπει να περιορίζει γεωγραφικά τον χρήστη (μπορεί να καταχωρεί σπίτι από οποιοδήποτε σημείο του κόσμου).
* Η εφαρμογή θα πρέπει να ελέγχει την πληρότητα και την ακρίβεια των δεδομένων που εισάγει ο χρήστης σε κάθε πεδίο πριν τα μεταφέρει στον server και να τον ενημερώνει για τυχόν λανθασμένες τιμές.
* Η αρχιτεκτονική της εφαρμογής θα πρέπει να διαχειρίζεται άψογα τους πόρους του συστήματος ώστε να είναι γρήγορη κατά τη χρήση της και να μην επιβαρύνει την συσκευή του χρήστη.
* Οι χρονοβόρες διαδικασίες λήψης ή αποστολής δεδομένων θα πρέπει να γίνονται ασύγχρονα με χρήση AsyncTask στο background προς αποφυγή επιβάρυνσης του κύριου UI thread.

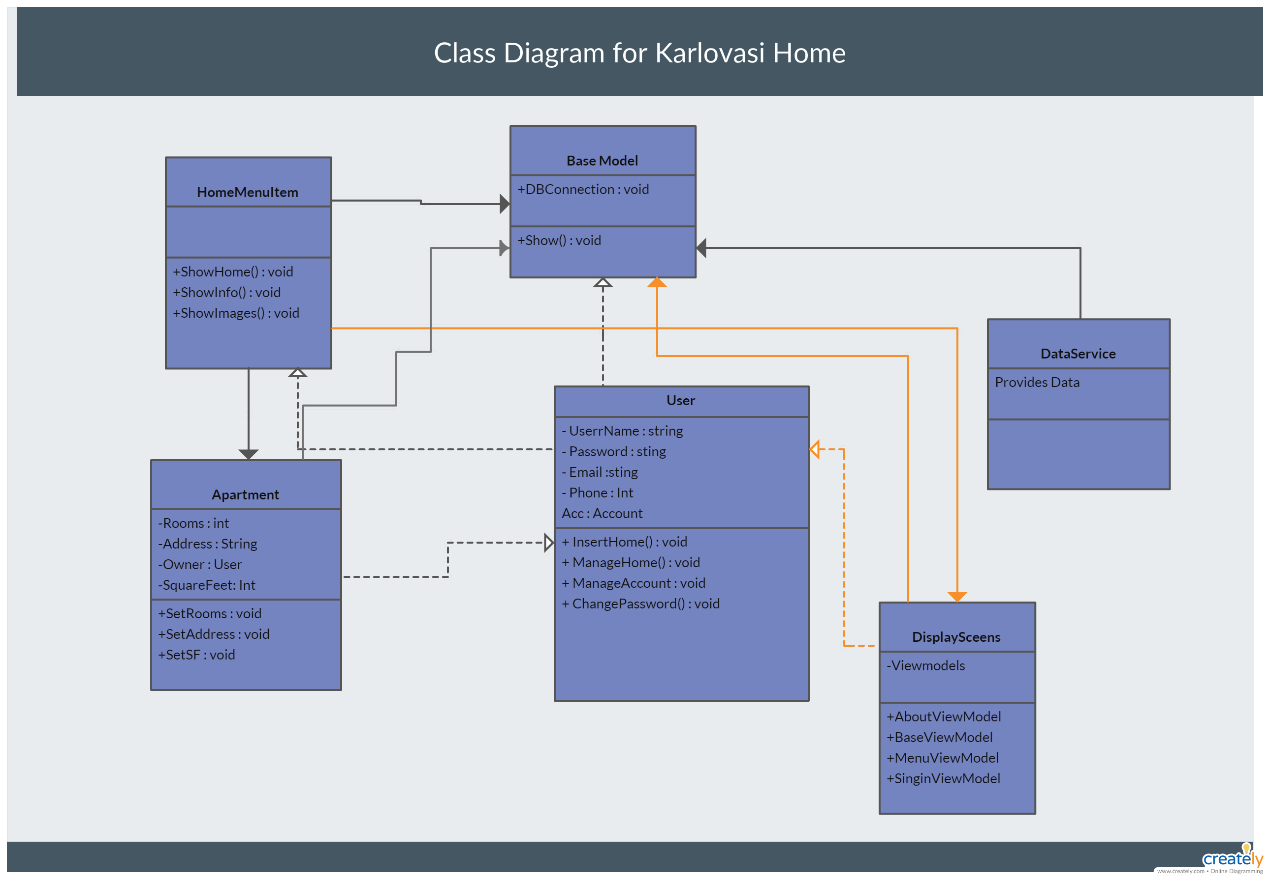
# **Βασικά Διαγράμματα**

Το Activity Diagram της εφαρμογής δείχνει την βασική ακολουθία διαδικασιών για την εισαγωγή ενός σπιτιού από τον ιδιοκτήτη και την τελική ενοικίαση από τον χρήστη.



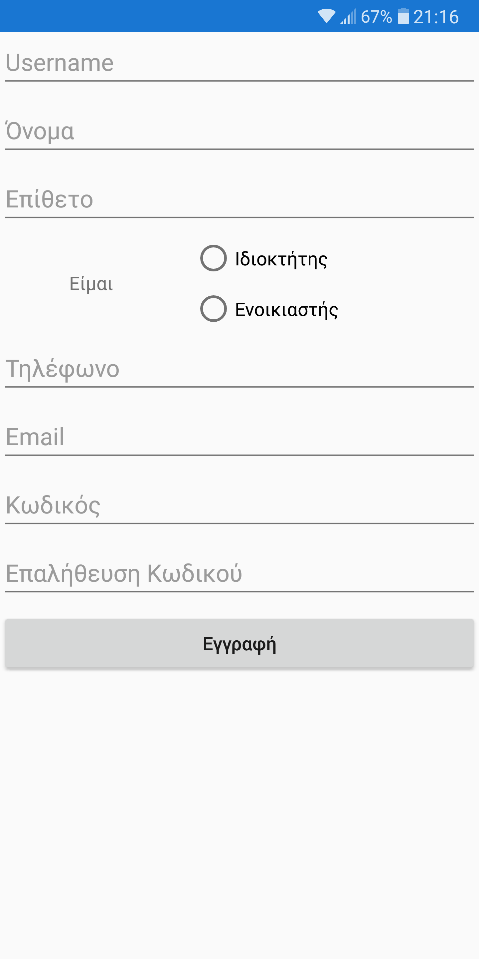
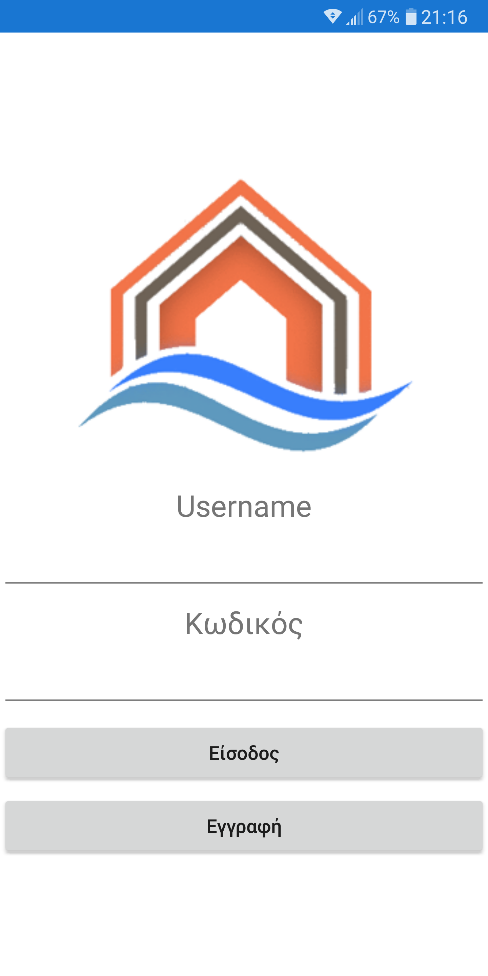
Η υλοποίηση της Βάσης Δεδομένων μας είναι σχετικά απλή καθώς έχουμε έναν πίνακα για όλους τους χρήστες, έναν πίνακα για τα σπίτια ο οποίος συνδέεται με το Foreign Key των ιδιοκτητών και έναν πίνακα για την αποθήκευση των αγαπημένων σπιτιών των χρηστών ο οποίος συνδέεται με τους 2 προηγούμενους πίνακες με τα Foreign Keys τους.



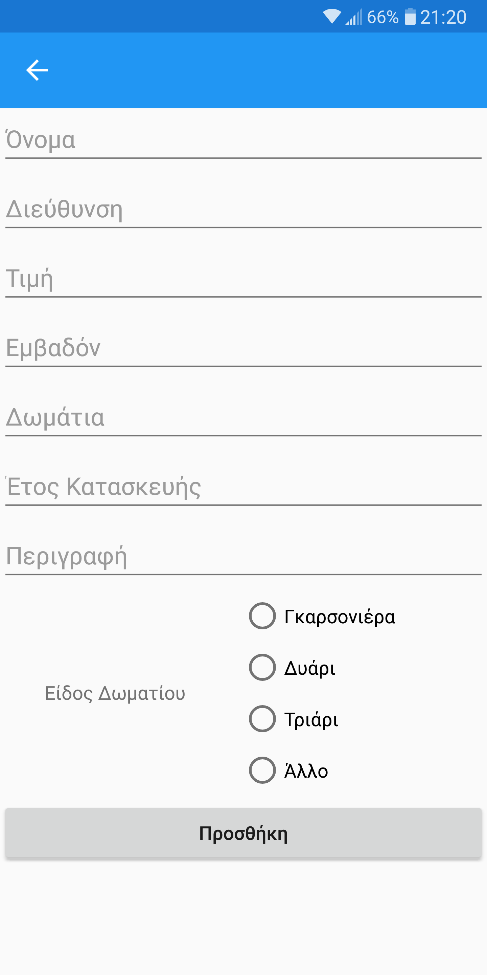


# **Κύριες Οθόνες**

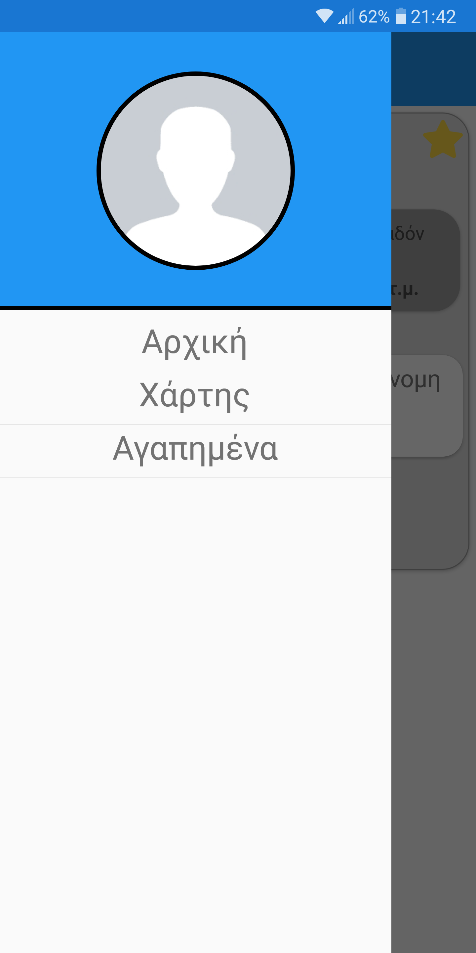
**Log In Register**

****

 **Αναζήτηση Σπιτιού** **Καταχώρηση Σπιτιού**



**Διαχείριση Σπιτιών Επιλογές Χρήστη**

****

**Χάρτης Σπιτιών** **Αγαπημένα Σπίτια**



# **Documentation Κώδικα**

Η αρχιτεκτονική που χρησιμοποιήσαμε για την ανάπτυξη της εφαρμογής μας είναι η “Model, View, View Model” (MVVM) και την επιλέξαμε επειδή προσφέρει μεγαλύτερη αποδοτικότητα και μικρότερη πολυπλοκότητα στον κώδικα συνδέοντας το User Interface με τον Backend Κώδικα και προσαρμόζοντας τα μοντέλα για τη σωστή εμφάνιση τους στο User Interface.

Η συγκεκριμένη εφαρμογή όταν ξεκινάει, εκκινεί μια κλάση που κάνει τις απαραίτητες συνδέσεις με τη βάση δεδομένων και κατόπιν εμφανίζεται η πρώτη σελίδα της εφαρμογής. Παρατηρούμε ότι σχεδόν κάθε View συνδέεται με ένα View Model το οποίο είναι σε άμεση επικοινωνία με τη βάση δεδομένων και κάνει αιτήματα σε αυτή για κάθε ενέργεια του χρήστη. Για παράδειγμα, όταν εμφανίζεται η αρχική σελίδα, συνδέεται με το View Model και όταν ο χρήστης εισάγει τα στοιχεία του για να εισέλθει στην εφαρμογή, αφού πίσω από τη View γίνουν κάποιοι έλεγχοι (Πχ συμπλήρωση όλων των πεδίων απαραίτητα), καλείται το View Model για να στείλει ένα αίτημα στη βάση και να ελέγξει αν ο χρήστης υπάρχει σε αυτή.

Αφού ανοίξουμε το φάκελο της εφαρμογής παρατηρούμε 3 διαφορετικά project που την συνθέτουν και είναι απαραίτητα για την υλοποίηση και λειτουργία της.

1. Το project KarlovasiHome είναι το Portable project στο οποίο είναι γραμμένος όλος ο κώδικας
2. Το project KarlovasiHome.Android είναι το Android project στο οποίο μεταφράζεται ο κώδικας του Portable και στην ουσία είναι η Android εφαρμογή μας
3. Το project KarlovasiHomeService είναι η υλοποίηση του Web Service μέσω του οποίου γίνονται τα αιτήματα στη βάση δεδομένων

Στη συνέχεια παρατηρούμε στο project τους διάφορους υποφακέλους που το αποτελούν.

1. Ο πιο γνωστός φάκελος είναι ο Views, ο οποίος περιέχει τα διάφορα User Interface της εφαρμογής μαζί με τον Backend κώδικα τους
2. φάκελος Models περιέχει τα μοντέλα/οντότητες που χρησιμοποιούμε στην εφαρμογή
3. Ο φάκελος Services περιέχει το μοναδικό service που χρησιμοποιούμε για την επικοινωνία με τη βάση δεδομένων
4. Ο φάκελος Converters περιέχει κάποιες κλάσεις οι οποίες δέχονται μεταβλητές και ανάλογα με την τιμή τους επιστρέφουν κάποια άλλη τιμή και χρησιμοποιούνται για την αποδοτικότερη και πιο ξεκάθαρη προβολή των μοντέλων. Για παράδειγμα, αν ένα διαμέρισμα δεν είναι διαθέσιμο, η τιμή αυτή πηγαίνει στον Converter και επιστρέφει σαν τιμή το χρώμα κόκκινο για την προβολή του διαμερίσματος
5. Τέλος, ο φάκελος ViewModels περιέχει τα View Models που συνδέονται με τα Backend των προβολών της εφαρμογής με σκοπό την υλοποίηση πολύπλοκων διαδικασιών καθώς και για πιο καθαρό και αναγνώσιμο κώδικα

# **Testing Documentation**

Το Testing της εφαρμογής πραγματοποιήθηκε από εμάς τους ίδιους κατά την διάρκεια ολοκλήρωσης της εφαρμογής. Ακολουθήσαμε διάφορα σενάρια όπως η δημιουργία χρηστών, εισαγωγή σπιτιών, αναζήτηση σπιτιών κλπ, με τα οποία ελέγξαμε τις διάφορες λειτουργίες της εφαρμογής και κάναμε τις απαραίτητες διορθώσεις σε bugs και δυσλειτουργίες που συναντήσαμε, οι οποίες διορθώθηκαν επιτυχώς στην τελική παραδοτέα έκδοση της εφαρμογής.

# **Εξωτερικές Πηγές**

* Βάση Δεδομένων: <https://azure.microsoft.com/en-us/>
* Δημιουργία Διαγραμμάτων: <https://creately.com/>

<https://www.teamgantt.com/>

* IDE: <https://visualstudio.microsoft.com/>

# **Συμπεράσματα**

Η υλοποίηση της εφαρμογής ακολούθησε τον αρχικό προγραμματισμό και δεν συναντήσαμε ιδιαίτερες δυσκολίες και προβλήματα. Παρόλα αυτά, υπάρχει κάποιο περιθώριο βελτιώσεων σε θέματα όπως το Design της εφαρμογής καθώς και σε κάποιες λειτουργίες όπως η αναζήτηση σπιτιών με κάποια συγκεκριμένα φίλτρα ή ακόμη και η άμεση επικοινωνία του χρήστη με τον ιδιοκτήτη του σπιτιού, τα οποία θα αναβαθμιστούν σε επόμενες εκδόσεις της εφαρμογής.