**iFlat**

Requirements Specification and Analysis

Version 1.0

Date: 20.11.2016

Alican YIlMAZ

Eren AY

Ö.Meva DEMİRÖZ  
Tolga TANER

Prepared for

SE301 Software Engineering



Table of Contents

[1. Introduction 1](#_Toc431983052)

[1.1. Purpose of the System 1](#_Toc431983053)

[1.2. Scope of the System 1](#_Toc431983054)

[1.3. Objectives and Success Criteria of the Project 2](#_Toc431983055)

[1.4. Definitions, Acronyms, and Abbreviations 2](#_Toc431983056)

[1.5. Overview 3](#_Toc431983057)

[2. Current System 4](#_Toc431983058)

[3. Proposed System 4](#_Toc431983059)

[3.1. Overview 4](#_Toc431983060)

[3.2. Functional Requirements 5](#_Toc431983061)

[3.3. Nonfunctional Requirements 6](#_Toc431983062)

[Usability 6](#_Toc431983063)

[Reliability 6](#_Toc431983064)

[Performance 6](#_Toc431983065)

[Supportability 6](#_Toc431983066)

[Implementation 6](#_Toc431983067)

[Interface 7](#_Toc431983068)

[Packaging 7](#_Toc431983069)

[Legal 7](#_Toc431983070)

[3.4. System Models 8](#_Toc431983071)

[Scenarios 8](#_Toc431983072)

[Use case model 32](#_Toc431983073)

[Object model 33](#_Toc431983074)

[Dynamic model 33](#_Toc431983075)

[User interface—navigational paths and screen mock-ups 37](#_Toc431983076)

[4. Glossary 41](#_Toc431983077)

[5. References 43](#_Toc431983078)

REQUIREMENTS ANALYSIS DOCUMENT[1]

# 1.Introduction

## Purpose of the System

In this system, we designed a rent a flat system so as to ensure a well communication between renter and host also renter can rent own flat. We designed a model that aimed to please both renter and host and tried to provide fast, reliable, efficient service by presenting to both renter and user simple GUI (interface) and design that very easy to use. By developing this system, mobile based and referenced the customer of different rent a flat system. They want something different and simpler. They want to have a faster, lossless and limitless in terms of usability. So we designed our system by taking into consideration of these needs.

## Scope of the System

Our system is mobile system so it can be used everywhere, every different mobile device and access internet site that have Internet access. It is so simple to use; register, select your location, select flat or house, and money payment money with banck account from system! That’s it. Cause our system is mobile compatible, therefore users can rent flats on mobile phone. Our system provides secure and reliable communication renter between and host with association firebase.  
 We use some services for in this sytem.For example: credit card validation that secure payment and T.C identity validations for realibale communication.

## Objectives and Success Criteria of the Project

The success of the system depends on providing the given main set of arguments:

* The design of register, login systems.
* Association between firebase and mobile platform design
* The general design of system in rent to have fast, efficient system.
* The demo of system should ensure good success rate.
* Implementation should be understandable, clear and basic and efficient.

## Definitions, Acronyms, and Abbreviations

*RAD:* Requirements Analysis Document

SysAdmin: System Admin

*Model:* A schematic description of a system that accounts for its known or inferred properties

*System:*  Any interactins performed by the application are considered to

be performed by the system.

*Efficiency:* The properties of an [algorithm](https://en.wikipedia.org/wiki/Algorithm) which relate to the amount of [computational resources](https://en.wikipedia.org/wiki/Computational_resource) used by the algorithm.

*GUI:* Graphical User Iterface

*Service:* servis is a keyword. Purpose of service serve from sytem to customer such as iFlat system, provides system security with credit card validations.

*Firebase:* Firebase is a platform that developed by Google and have knowledge of do changes user transmit information to developer by user [1]

## Overview

This document contains sections; Introduction, Current System, Functional Requirements, Non-Functional Requirements, Scenarios, Use case model, Object model, dynamic model, use interface, Glossary, References. In ıntroduction we’ve described a good explanation of scope prpose, criterias, definitions and other explanations which helps other help people who want to read document. In proposed system, we have described advantages of iFlat and what makes iFlat to unique. In functional requirement, we have described that the functions that is open to end users. Functional requirements states for explaining scope of the system.

In non-functional requirement section, we have described our system’s reilability, performance and others to make the system accesible to end users. In Scenarios section we have descibed use case’s descriptions and explanations. In Use case model we have stated or use case diagram. In Object model, we have described the programming objects which will implemented later on. In Dynamic model, we have described the flow of system in time with operations and function of actors. In glossary section we have described dictionary and unknown words in RAD.

# Current System

Our system is a new system and we designed our system on mobile. So all the inputs information and outputs information are given via mobile device. Platform works synchronous and informs renter and host and system admin very fast. So the communication will never end until rent a flat.

# Proposed System

## Overview

iFlat is an iOS application which is usable within iPhone mobile devices to provide a good solution to hosts who rents thieir home to customer who rents flats. Our system aims to make this people get in touch and communicate easly. With iFlat renters are able to search flats by location or other filters and browse the flats which is satisfies to their mind. Rent flats and contact hosts easily. Renters can use promotion code before making reservation request. Renters also can rate the flats that reserved by themself recently. Also Renters have rate stars that simulate their reliability. Which browsable by hosts before renting the flat to renters they’ve requested reservation. System admins can open promotion codes and maintain issues and complaints. When sys admin closes any issue which created by other actors, notification system notifies the related customer. When renters’s reservation approved by host, renters get notification also. Host can not rent flats but renters can rent flats. Host actor is a user who can create their profile and add flat to their accounts. When hosts add flat to their account, their host can be searched by renters with specified information. Hosts can edit their flats and profile info and can delete their flats. When they have deleted a flat, the flat omited from the system and no longer browsable. Also Renters can cancel their reservation before check-in time. Hosts can add their flat’s properties and other information with photo information to their flats. Hosts can display recent or past reservation related any flat which is owned by related host. iFlat is a efficient and powerful system to make customers happy and create a reliable environment to make renting market wide.

## Functional Requirements

iFlat is an online rent a flat platform for host and renter. On customer’s side, iFlat has several functions.

* Host, renter, customer, sysAdmin, register in the system.
* Host, renter, customer, sysAdmin login in the system
* Host can see own flat and renters who are want to rent his/her flat.
* Host if he/she want to rent an other flat, can change account.
* Host see rate stars and give rate for renters.
* Host and renter can do searching in this system.
* Host add flat in this system.
* Host and renter edit profil.
* Renter can see flat in the system.
* Renter can rent a flat.
* Renter give rate stars for host’s flat
* Renter and host complain to sysAdmin for this system.
* Renter make rezervation flat in this system.
* Renter must be able to cansel reservation.

## Nonfunctional Requirements

### **Usability:** iFlat has an enough user interface and good user experience which requires like 10-15 tap gesture, 3-4 swipe gesture, 2-3 scrolling gesture to use the mobile app. Our interface is powerful which aims to “content first” apporach. İFlat aims to make less tapping gestures to provide whole system functions with full of its software requirements.

### **Reliability:** iFlat must be online functions %95 of the time.

### **Performance**: iFlat uses great No-SQL apporach on persistent data storage so it provides fast responses and must handle like 5000 parallel users at the same time

### **Supportability:** iFlat’s maintenance must be easy wihin a user-friendly portal which only opens to Sys-Admins and Non-End-Users.

### **Implementation:** The system implemented with Xcode 8+ IDE, Swift 3+ native programming language and its great OOP and POP features. iFlat uses in back-end Firebase Real Database NoSQL platform, In Front-End, iFlat uses Apple’s iOS APIs and Libraries and can be accesible in iPhones. And the app can be downloaded from App Store.

### **Interface:** We have designed iFlat from fresh. Therefore, Although the 3rd party APIs and Services used in iFlat which are described to the client, there is no other interfaces used. And there is no legacy project also which provides its services or interfaces.

### 

### **Packaging:** Packaging of the system will be included in the future

### 

### **Legal:** iFlat is licenced by MIT Licence. The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and no infringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

## System Models

Describes the scenarios, use cases, object model, and dynamic models for the system. This section contains the complete functional specification, including mock-ups illustrating the user interface of the system and navigational paths representing the sequence of screens.

***Use Case Scenarios***

|  |  |
| --- | --- |
| Scenario Name | Add Promotion |
| Participant actor instances | Ahmet:SysAdmin |
| Flow of Events | 1. Ahmet enters the system. 2. Ahmet determines promotion's details. 3. Ahmet adds a promotion to the system. |

|  |  |
| --- | --- |
| Scenario Name | Delete Promotion |
| Participant actor instances | Ahmet : SysAdmin |
| Flow of Events | 1. Ahmet enters the system. 2. Ahmet checks out a promotion which he wants delete. 3. Ahmet deletes a promotion from the system. |

|  |  |
| --- | --- |
| Scenario Name | Ban Customer |
| Participant actor instances | Kamil: SysAdmin |
| Flow of Events | 1. Kamil enters the system. 2. Kamil finds out customer to will be banned. 3. Kamil bans a customer. |

|  |  |
| --- | --- |
| Scenario Name | Rate Customer |
| Participant actor instances | Taner: Renter  Tolga:Host |
| Flow of Events | 1. Taner enters the system. 2. Taner searches a flat. 3. Taner rents a Tolga's flat. 4. Rental processes is over. 5. Tolga and Taner gives rating with each other. |

|  |  |
| --- | --- |
| Scenario Name | Report Flat Issue(High Priority) |
| Participant actor instances | Ayşe : Renter  Özge : SysAdmin |
| Flow of Events | 1. Ayşe rents a flat. 2. Ayşe is not pleased about rented flat. 3. Ayşe reports this flat to Ozge. |

|  |  |
| --- | --- |
| Scenario Name | Edit Promotion |
| Participant actor instances | Fatma : SysAdmin |
| Flow of Events | 1. Fatma enters the system. 2. Fatma searches a promotion. 3. Fatma edits a promotion. |

|  |  |
| --- | --- |
| Scenario Name | Display Owner Profile |
| Participant actor instances | Hakan : Customer |
| Flow of Events | 1. Hakan enters the system. 2. Hakan clicks menu button. 3. Hakan displays own profile. |

|  |  |
| --- | --- |
| Scenario Name | Add Flat to Wish List |
| Participant actor instances | Ahmet : Renter |
| Flow of Events | 1. Ahmet logs in the system. 2. Ahmet searches and lists all flat. 3. Ahmet clicks “add wish list” button to add a flat which he likes. |

|  |  |
| --- | --- |
| Scenario Name | View On Map |
| Participant actor instances | Ahmet : Renter |
| Flow of Events | 1. Ahmet searches or list all flats. 2. Ahmet clicks this flat's details. 3. Ahmet sees this flat on map. |

|  |  |
| --- | --- |
| Scenario Name | Report Customer Issue(High Priority) |
| Participant actor instances | Ahmet : Renter  Mehmet : Host  Kemal:SysAdmin |
| Flow of Events | 1. Ahmet or Mehmet comments negative. 2. One of them clicks report customer button. 3. Kemal gets this report. 4. Kemal evaluates this issue. |

|  |  |
| --- | --- |
| Scenario Name | Use Promotion |
| Participant actor instances | Ahmet : Renter |
| Flow of Events | 1. Ahmet finds out a promotion. 2. Ahmet opens payment options. 3. Ahmet enters a promotion code. |

|  |  |
| --- | --- |
| Scenario Name | Close Issue |
| Participant actor instances | Mehmet : SysAdmin |
| Flow of Events | 1. Mehmet searches all issue. 2. Mehmet finds out a issue. 3. Mehmet closes this issue. |

|  |  |
| --- | --- |
| Scenario Name | Apply Filter |
| Participant actor instances | Eren:Host,Renter |
| Flow of Events | 1. Eren logs in the system. 2. Eren searches a flat. 3. Eren searches a flat by money,address,date,location,room numbers. |

|  |  |
| --- | --- |
| Scenario Name | Reserve Flat (Hight Priority) |
| Participant actor instances | Ahmet : Renter  Mehmet:Host |
| Flow of Events | 1. Ahmet logs in the system. 2. Ahmet lists all flat. 3. Ahmet likes a flat. 4. Ahmet makes a reservation request to Mehmet. 5. Mehmet checks out the request. 6. Mehmet accepts the request. 7. Ahmet gets this notification. 8. Ahmet pays money how much Mehmet's flat is. 9. Ahmet reserve Mehmet's flat. |

|  |  |
| --- | --- |
| Scenario Name | Adding a new flat(Hight Priority) |
| Participant actor instances | Ali : Host |
| Flow of Events | 1. Ali logs in the system. 2. Ali opens the menu. 3. Ali chooses add a new flat. |

|  |  |
| --- | --- |
| Scenario Name | Searching Flat(Hight Priority) |
| Participant actor instances | Veli : Renter |
| Flow of Events | 1. Veli enters this system. 2. Veli searches all flat by their feature. |

|  |  |
| --- | --- |
| Scenario Name | Reserve a flat(Hight Priority) |
| Participant actor instances | Ahmet : Renter  Kamil : Host |
| Flow of Events | 1. Ahmet searches a flat. 2. Ahmet finds out Kamil's flat. 3. Ahmet sends a reservation request. 4. Kamil accepts this request. 5. Ahmet chooses a payment option. 6. Ahmet pays flats' money. 7. Ahmet rents Kamil's flat. |

|  |  |
| --- | --- |
| Scenario Name | Changing Password |
| Participant actor instances | Mehmet : Customer |
| Flow of Events | 1. Mehmet logs in the system. 2. Mehmet clicks “edit own profile” button. 3. Mehmet clicks change password button. |

|  |  |
| --- | --- |
| Scenario Name | Withdraw Money to Bank Account(Hight Priority) |
| Participant actor instances | Ayşe : Renter  Fatma : Host |
| Flow of Events | 1. Ayşe rents Fatma's flat. 2. Ayşe pays the payment. 3. Ayşe's money goes the system. |

|  |  |
| --- | --- |
| Scenario Name | Withdraw Money to Bank Account |
| Participant actor instances | Ayşe : Renter  Fatma : Host |
| Flow of Events | 1. Ayşe rents Fatma's flat. 2. Ayşe pays the payment. 3. Ayşe's money goes the system. |

|  |  |
| --- | --- |
| Scenario Name | Adding a new payment option |
| Participant actor instances | Zeynep : Renter |
| Flow of Events | 1. Zeynep searches a flat. 2. Zeynep sends a reservation request. 3. Zeynep adds a new payment option. 4. Zeynep rents a flat. |

|  |  |
| --- | --- |
| Scenario Name | Display Flat Profile |
| Participant actor instances | Hakan : Customer |
| Flow of Events | 1. Hakan enters the system. 2. Hakan searches flats. 3. Hakan finds out a flat. 4. Hakan clicks display flat profile. |

|  |  |
| --- | --- |
| Scenario Name | Display Customer Profile |
| Participant actor instances | Ahmet : Customer |
| Flow of Events | 1. Ahmet searchs a customer. 2. Ahmet enters his/her profile. 3. Ahmet displays his/her profile. |

|  |  |
| --- | --- |
| Scenario Name | Edit Own Profile |
| Participant actor instances | Mehmet : Customer |
| Flow of Events | 1. Mehmet logs in the system. 2. Mehmet clicks menu button. 3. Mehmet selects edit Profile button. |

|  |  |
| --- | --- |
| Scenario Name | Edit Flat Profile |
| Participant actor instances | Ali : Host |
| Flow of Events | 1. Ali logs in the system. 2. Ali clicks menu button. 3. Ali clicks edit own profile button. 4. Ali enters edit flat profile. |
|  |  |

|  |  |
| --- | --- |
| Use Case Name | register |
| Participant Actors | Initiated by Customer |
| Flow of Events | 1. Customer opens the Register view by clicking Register button on the menu, if Customer is not registered. 2. Rent a Flat responds by presenting a Register form to Customer. 3. Customer fills out the form by entering TCID, name, surname, address, birthdate, email and phone number. When the form is completed, Customer submits the form. 4. Rent a Flat receives the form, create account, make Customer logged in and redirects Customer to main view. |
| Entry Condition | Customer clicks to Register button. |
| Exit Condition | Customer has completed registration process. |

|  |  |
| --- | --- |
| Use Case Name | rateFlat |
| Participant Actors | Initiated by Customer |
| Flow of Events | 1. Customer clicks Rate button on the Flat Profile, choose start amount out of five and clicks on it. When user clicked on the star, rate request is sent to Rent a Flat. 2. Rent a Flat receives the request and add rating to Flat and responds by message to Customer. |
| Entry Condition | Customer is logged into Rate a Flat.  Customer clicks Account tab on the menu.  Customer chooses Reservations option on the screen.  Customer clicks Rate button at Finished Reservations tab. |
| Exit Condition | Customer has rated the flat. |

|  |  |
| --- | --- |
| Use Case Name | editProfile |
| Participant Actors | Initiated by Customer |
| Flow of Events | 1. Customer clicks menu on the screen.  2. Customer chooses Profile tab.  3. Customer’s profil page open on the screen  4. Click edit profile tab. |
| Entry Condition | Customer is logged into Rent a Flat.  Customer chooses Edit Profile option on the screen. |
| Exit Condition | Customer has edited own profile.  Customer is logout from Rent a Flat. |

|  |  |
| --- | --- |
| Use Case Name | withdrawBalance |
| Participant Actors | Initiated by Customer |
| Flow of Events | 1. Customer click Withdraw button at Account view. 2. Rent a Flat responds by presenting Withdraw view. 3. Customer fills out the form by selecting Bank Account that money will be transferred and enters the amount of money. When the form is completed, Customer submits the form. 4. Rent a flat receives the form, controls balance and bank account, and make transaction from Customer’s balance to Customer’s Bank Account. |
| Entry Condition | Customer is logged into Rent a Flat.  Customer clicks Account tab on the menu.  Customer chooses Withdraw option on the screen. |
| Exit Condition | Customer has completed withdraw.  Customer has no balance. |

|  |  |
| --- | --- |
| Use Case Name | Login |
| Participant Actors | Initiated by Customer (Host, Renter) |
| Flow of Events | 1. The user first enters to Rent a Flat, clicks Login on the menu.  2. Rent a Flat presents the Login Form to the user.  3. The user enters his/her email into Email text field on the screen, also enters his/her password into Password text field on the screen. Lastly, the user sends a request to Rent a Flat by clicking to Login button on the screen to be logged in.  4. Rent a Flat checks the Email and Password and allows the User to be logged in. |
| Entry Condition | Customer enters to login view. |
| Exit Condition | Customer has logged in. OR  Customer has saw pop-up why he/she could not login. |

|  |  |
| --- | --- |
| Use Case Name | editFlatProfile |
| Participant Actors | Initiated by Host |
| Flow of Events | 1. Host clicks menu on the screen.  2. Host’s flat page open on the screen.  3. Host chooses Edit Flat Profile option on the screen. |
| Entry Condition | Host is logged into Rent a Flat.  Host chooses My Flats tab. |
| Exit Condition | Host has edited flat profile.  Host is logout from Rent a Flat. |

|  |  |
| --- | --- |
| Use Case Name | displayFlatProfile |
| Participant Actors | Initiated by Customer |
| Flow of Events | 1. Customer clicks on the Flat Title. 2. Rent a Flat responds by Flat Profile View. |
| Entry Condition | Customer searches a flat.  Customer clicks flat link on the result list. OR  Customer opens Reservations.  Customer clicks flat link on the Reservations. |
| Exit Condition | Customer has displayed flat profile. |

|  |  |
| --- | --- |
| Use Case Name | addFlat |
| Participant Actors | Initiated by Host |
| Flow of Events | 1. The user clicks rent a flat button and become Host. Host opens Add Flat view. 2. Rent a Flat responds by presenting a form to Host. 3. Host fills out the form by entering title, address, price, photos. When the form is completed, Host submits the form. 4. Rent a Flat receives the form and add it to List and displays a message that adding a flat was successful. |
| Entry Condition | Customer is logged into Rent a Flat.  Customer clicks to Rent a Flat button.  Customer become Host.  Host clicks Add Flat button. |
| Exit Condition | Host has added new flat successfully. OR  Host has received an reject. |

|  |  |
| --- | --- |
| Use Case Name | Add Promotion |
| Participant Actors | Initiated by SysAdmin,Renter |
| Flow of Events | 1-A renter clicks “rent a flat” button.  2-A renter clicks “apply promotion” if it is exist. |
| Entry Condition | SysAdmin adds a promotion code to somewhere.  Renters find this code to use discount on a flat. |
| Exit Condition | A SysAdmin adds a promotion. |

|  |  |
| --- | --- |
| Use Case Name | Delete Promotion |
| Participant Actors | Initiated by SysAdmin |
| Flow of Events | 1. SysAdmin clicks search all promotions button. 2. SysAdmin displays all promotions. 3. SysAdmin chooses a promotion. 4. SysAdmin delete a promotion. |
| Entry Condition | SysAdmin search all promotions.  SysAdmin selects a promotion in all promotions.  SysAdmin delete a promotion. |
| Exit Condition | A SysAdmin deletes a promotion. |

|  |  |
| --- | --- |
| Use Case Name | banCustomer |
| Participant Actors | Initiated by SysAdmin,Customer |
| Flow of Events | 1.SysAdmin gets a mail about a customer.  2.SysAdmin evaluates a mail.  3.SysAdmin bans customer from the system. |
| Entry Condition | A SysAdmin gets some mail from customers with it's topics.  A SysAdmin examines the mail's content.  A SysAdmin bans this customer by customer's information. |
| Exit Condition | A SysAdmin bans a customer. |

|  |  |
| --- | --- |
| Use Case Name | sendEmail |
| Participant Actors | Initiated by Customer |
| Flow of Events | 1.Customer clicks send mail button to others about their flats.  2.Other one answers the mail what about it is. |
| Entry Condition | Customer is logged into Rent a Flat.  Customer opens flat profile. OR,  Customer opens customer profile.  Customer clicks customer email to send email to target customer. |
| Exit Condition | Customer has sent mail to customer. |

|  |  |
| --- | --- |
| Use Case Name | Display Owner Profile |
| Participant Actors | Initiated by Renter,Host OR Customer |
| Flow of Events | 1.Customer wants display her/his profile to edit or look at it. |
| Entry Condition | A Renter,Host clicks own profile button.  A Renter,Host looks at their own profile. |
| Exit Condition | A renter,host shows own profile. |

|  |  |
| --- | --- |
| Use Case Name | Report Flat Issue |
| Participant Actors | Initiated by Renter,SysAdmin |
| Flow of Events | 1. A renter sends a mail about any host's flat. 2. A SysAdmin gets mail and answers it. 3. If renter is right,SysAdmin bans host from the system. |
| Entry Condition | A renter has a problem about rented flat.  A renter clicks report button in the home page.  A renter opens a issue with it's title and description.  A renter sends this topic to SysAdmin. |
| Exit Condition | A renter reports any bad issue about rented flat. |

|  |  |
| --- | --- |
| Use Case Name | Close Issue |
| Participant Actors | Initiated by SysAdmin |
| Flow of Events | 1. SysAdmin searches all topics. 2. SysAdmin finds the topics. 3. SysAdmin deletes expired topics. |
| Entry Condition | SysAdmin clicks search topics button.  SysAdmin displays all issue.  SysAdmin finds out the issiue. |
| Exit Condition | A SysAdmin closes the issue. |
|  |  |

|  |  |
| --- | --- |
| Use Case Name | Apply Filter |
| Participant Actors | Initiated by Host,Renter or Customer |
| Flow of Events | 1. A Host or renter searches a flat button. 2. A Host or renter applies filter by room number,date,location etc. |
| Entry Condition | A customer clicks a find flat button.  A customer applies filter by room number,date,location etc. |
| Exit Condition | A Customer applies filter. |

|  |  |
| --- | --- |
| Use Case Name | Reservation Flat |
| Participant Actors | Initiated by Renter,Host |
| Flow of Events | 1. A Renter logs in the system. 2. A Renter searches a flat. 3. A Renter finds out a flat to rent it . 4. A Renter sends reservation request to a Host |
| Entry Condition | A Renter finds out Host's flat.  A Renter sends reservation request to a Host in flat's profile. |
| Exit Condition | A Renter reserves a host's flat. |

|  |  |
| --- | --- |
| Use Case Name | Open Issue (Hight Priority) |
| Participant Actors | Initiated by Renter,Host or Customer |
| Flow of Events | 1. A customer wants to report any issiue about technical trouble or any costumer's comment in the system. 2. A customer opens a topic. |
| Entry Condition | A Customer opens customer's menu.  A Customer clicks on open issiue button.  A Customer enters title,body of its issue. |
| Exit Condition | A Customer opens a issue |

|  |  |
| --- | --- |
| Use Case Name | searchPopularHost |
| Participant Actors | Initiated by Renter,Customer |
| Flow of Events | 1. User clicks search field on the main view. 2. The user chooses Search option in the menu. 3. İflat responses by Search View. 4. The user without entering text, clicks Search button. 5. İFlat receives the request and suggest popular host list. 6. The user sees resulted host list. |
| Entry Condition | Customer or renter is logged into Rent a Flat.  User clicks enter. OR,  User clicks search button. |
| Exit Condition | User has searched popular host list.  Renter is logout from Rent a Flat.  If there is no popular host, back to main page |

|  |  |
| --- | --- |
| Use Case Name | searchFlat |
| Participant Actors | Initiated by Customer OR Support |
| Flow of Events | 1. The user chooses Search option in the menu. 2. Rent a Flat responses by Search View. 3. The user enters search text and clicks Search button. 4. Rent a Flat receives the request and finds related flats, list flats and responses by List Flat View. 5. The user sees resulted Flat List. |
| Entry Condition | Customer clicks search field on the main view.  Customer inputs search criteria in search field.  Customer clicks enter. OR,  Customer clicks search button. |
| Exit Condition | Customer has searched flat. |

|  |  |
| --- | --- |
| Use Case Name | rateCustomer |
| Participant Actors | Initiated by Customer |
| Flow of Events | 1.At the end of all rental processes,if customer is a renter,open host's account to mention about their experience.  2.İf Customer is a host, host gives rate about renter's behavior in rental process. |
| Entry Condition | Customer is logged into Rate a Flat.  Customer clicks Account tab on the menu.  Customer chooses Reservations option on the screen.  Customer clicks Rate(Host/Renter) button at Finished Reservations tab. |
| Exit Condition | Customer has rated the customer. |

|  |  |
| --- | --- |
| Use Case Name | Edit Promotion |
| Participant Actors | Initiated by SysAdmin |
| Flow of Events | 1.SysAdmin clicks edit promotion button.  2.SysAdmin edits. attributes of promotion. |
| Entry Condition | SysAdmin edits a promotion. |
| Exit Condition | A SysAdmin edits a promotion |

|  |  |
| --- | --- |
| Use Case Name | Report Customer Issue |
| Participant Actors | Initiated by Customer |
| Flow of Events | 1. A Customer enters the system. 2. A Customer clicks “open issue” button. 3. A Customer report another one. |
| Entry Condition | A Customer clicks login button.  A Customer clicks menu button.  A Customer opens Support. |
| Exit Condition | Customer reports the about each other. |

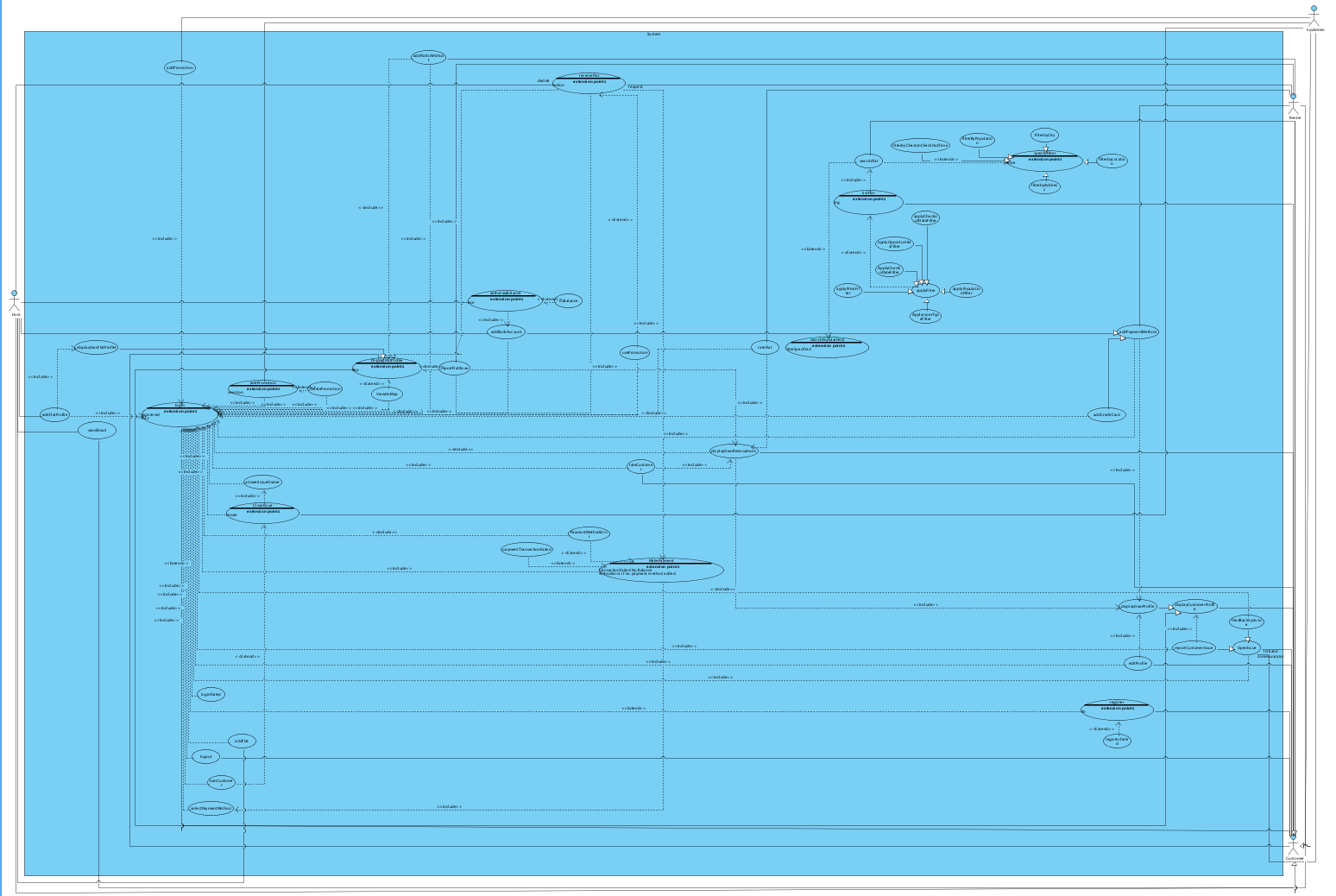
|  |
| --- |
| *Use case name:* MakePayment (High Priority) |
| *Participant actors instances: Michael:Renter*  *Rose:Host* |
| *Flow of events:*   1. Michaels log-in system and wants to rent a Flat. Michael than clicks a Flat that is ok for him and click reserve button. 2. iFlat responses by displaying a screen that contains flat’s info and properties. 3. Michael clicks submit reservation button 4. iFlat responses by displaying paymentation screen and waits to selecting a payment method. 5. Michael than select its payment method than click proceed. 6. iFlat responses by if success or fail of money transaction. 7. Rose receives a notification that contains reservation request of her flat. |
| *Entry Condition:* Michael is logged into iFlat. |
| *Exit Condition:* iFlat says Michaels informations are incorrect. OR,  Michael have no balance for money transaction. |

|  |  |
| --- | --- |
| Use Case Name | listFlat |
| Participant Actors | Initiated by Customer OR Support |
| Flow of Events | A Customer logs in the system.  A Customer clicks search button. |
| Entry Condition | Customer makes search. |
| Exit Condition | Customer clicks on the title of a Flat. |

|  |  |
| --- | --- |
| Use Case Name | Use Promotion |
| Participant Actors | Initiated by Renter |
| Flow of Events | 1. Renter clicks the login button. 2. Renter enters the system. 3. Renters finds out a flat. 4. Renters uses a promotion on a flat. |
| Entry Condition | Renter finds out a promotion.  Renter rents a flat.  Renter uses this promotion on a flat . |
| Exit Condition | A Renter uses a promotion on rented flat. |

|  |
| --- |
| *Use case name:* AddPaymentMethod (High Priority) |
| *Participant actors instances: Michael:Renter* |
| *Flow of events:*  1. Michael enters the iFlat app and click login button and fills the email and password fields than clicks log-in button.  2. iFlat gets the credential info and log Michael to system and reponses by displaying the screen which contains the name and surname of the Michael and the homepage.  3. Michael taps control panel button for displaying his renter functions.  4. iFlat then responses by displaying his control panel functions.  5.Michael clicks add payment method button.  6. iFlat responses by a screen that contains fields of adding payment method like credit or debit card.  7.Michael fills the fields with his credit or debit card info.  8. iFlat responses by success or fail message if it the fields are true. |
| *Entry Condition:* The Renter is logged into iFlat. |
| *Exit Condition:* iFlat says Michaels informations are incorrect. OR,  Michaels clicks logout button before adding methods or seeing succeed info. |

### **Use case model [5]**

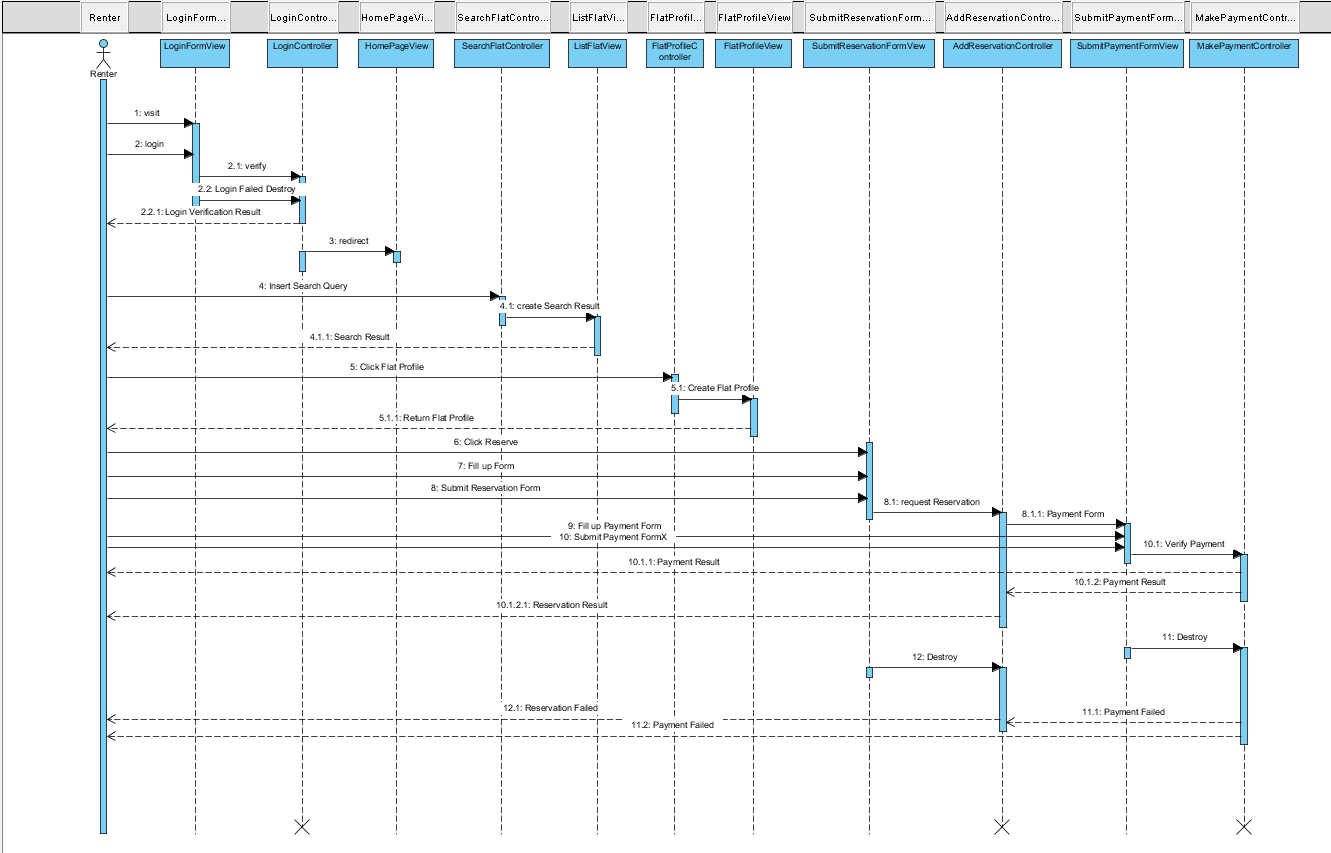


### **Object model [3]**

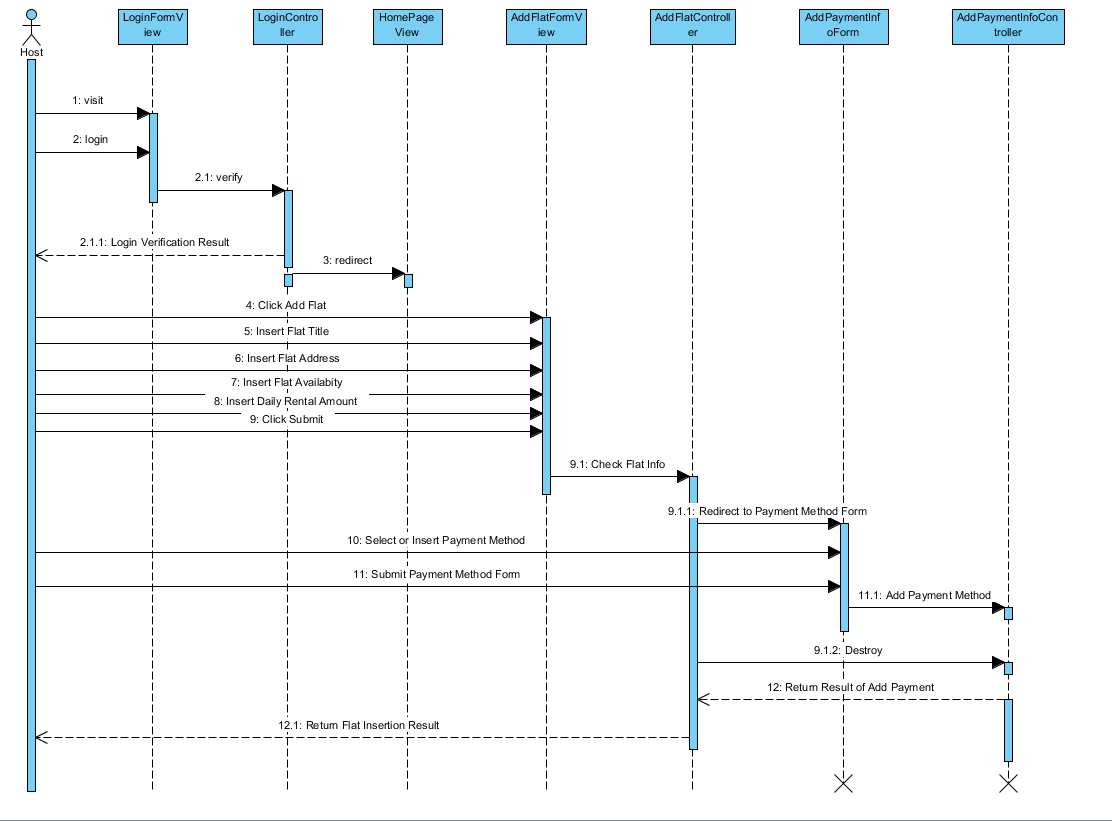
# ../Downloads/IMG_20112016_032034.png

### **Dynamic model [4]**

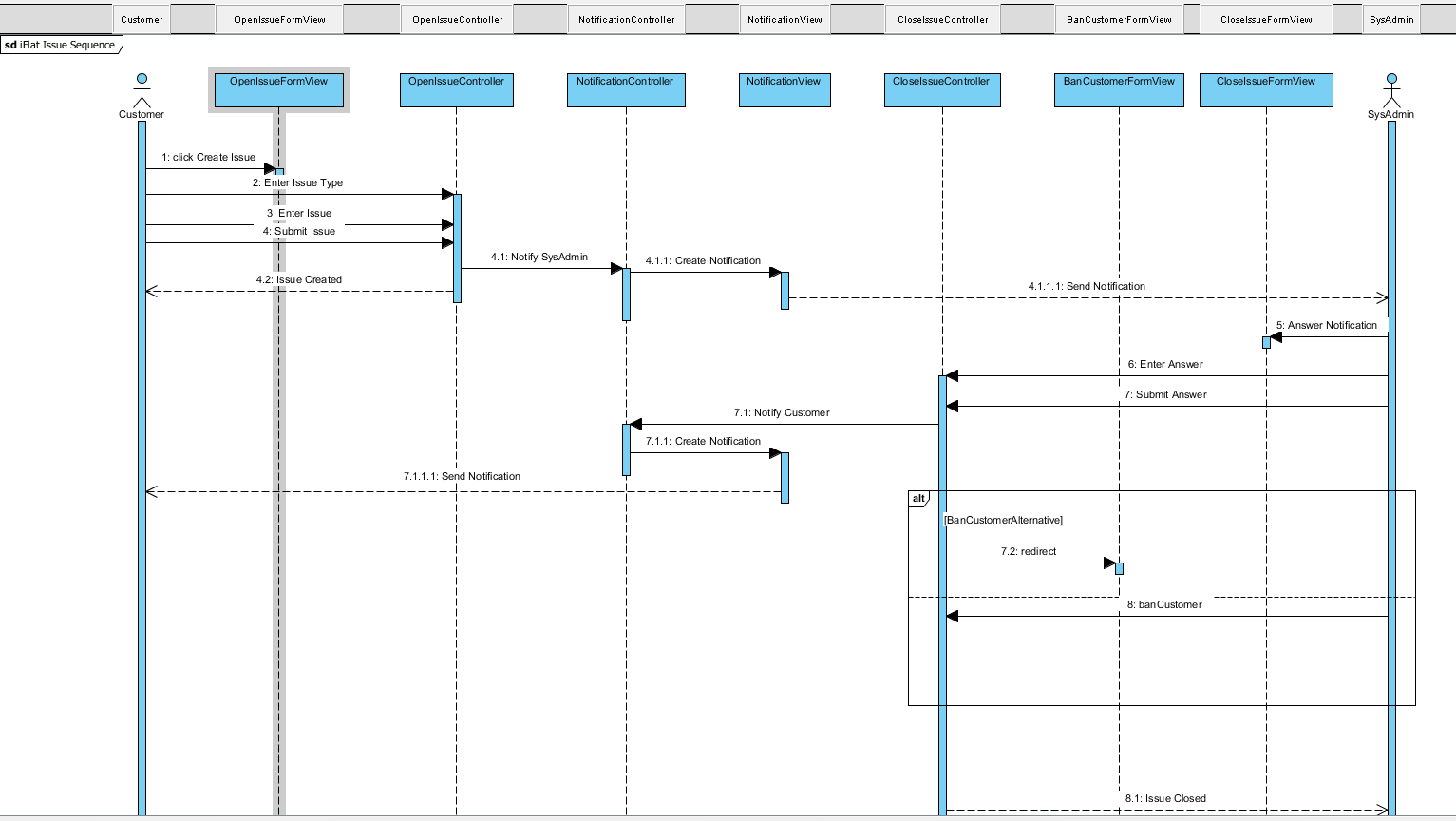
RentFlat



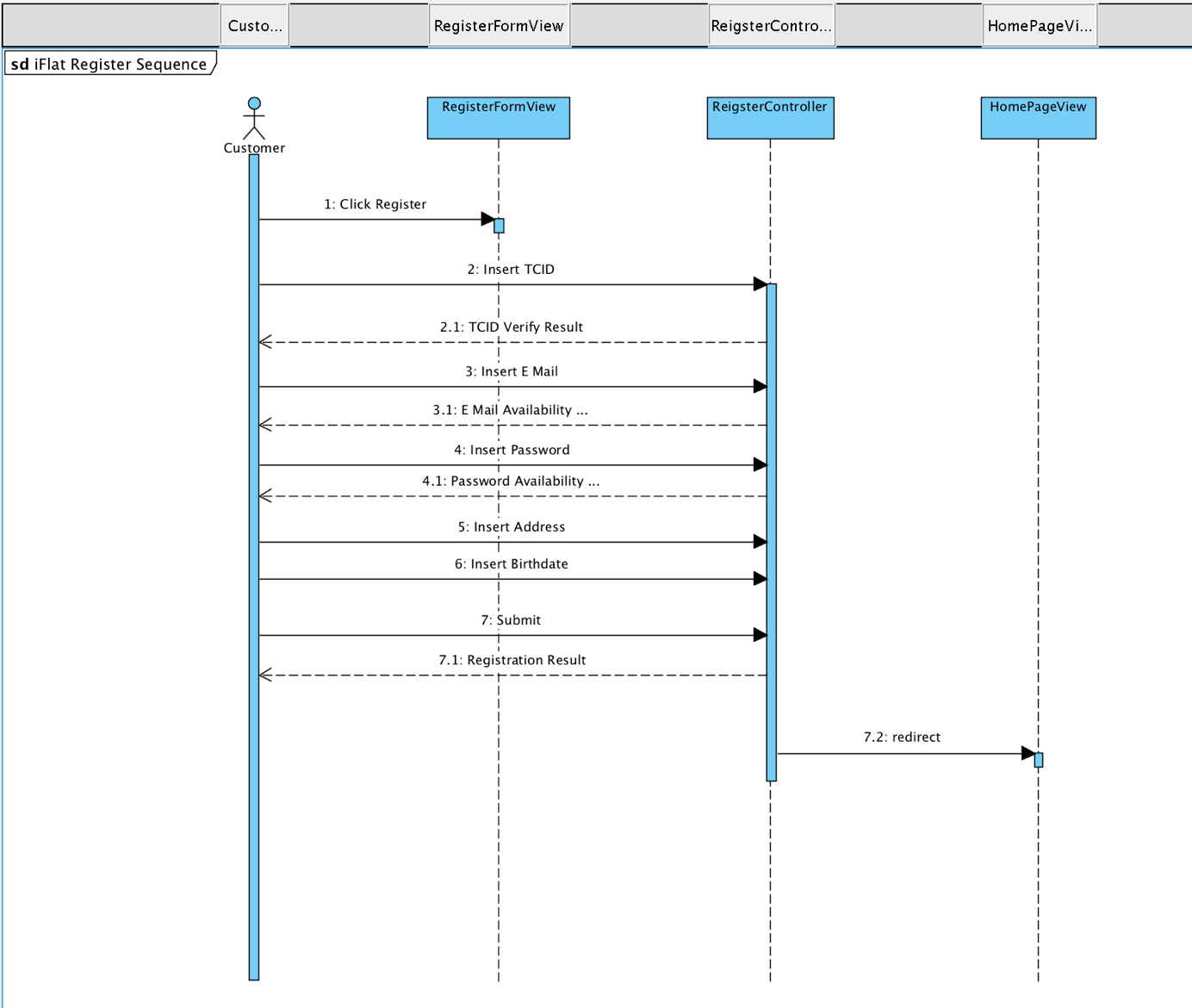
AddFlat

****

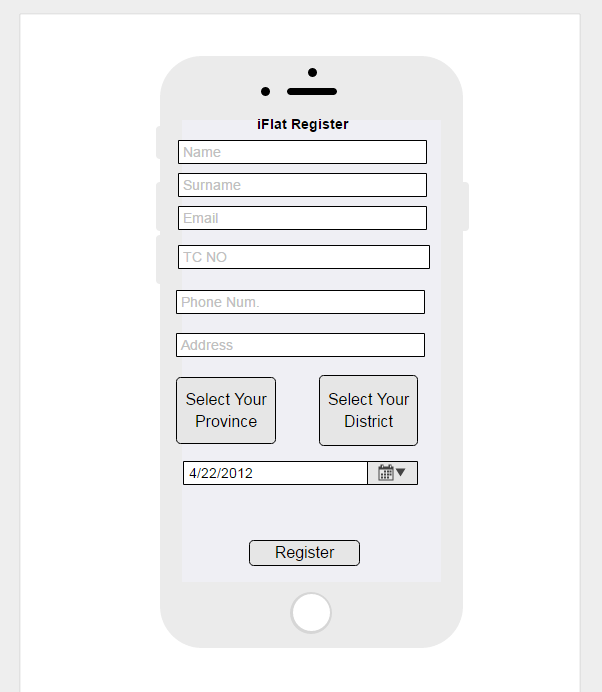
Issue

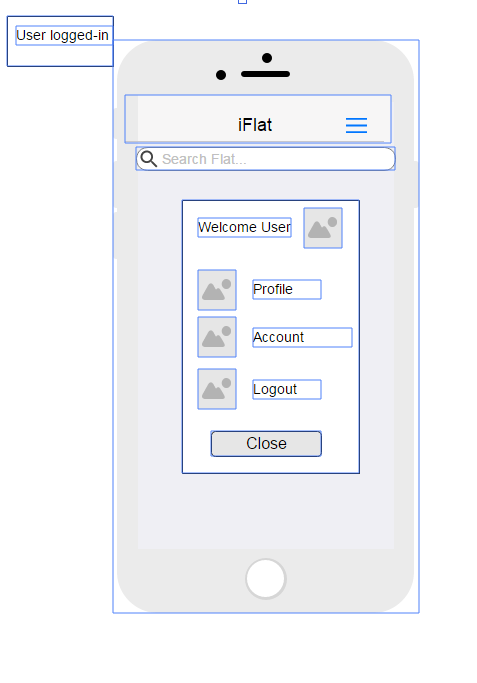
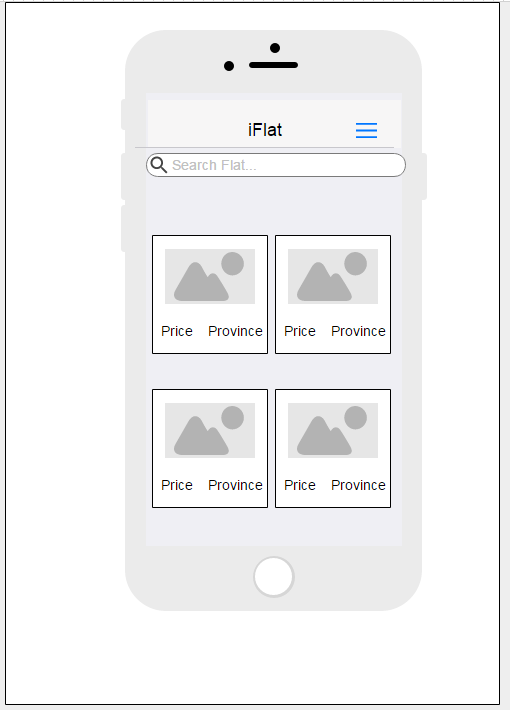
****

Register



**User Interface**

****

****

# Glossary

**Customer**: A customer can be a host or a renter according to her/his choice.

**Host**: A host adds own flat to the system.

**Renter**: A renter rents host’s flat which is in the system.

**SysAdmin**: A sysAdmin solves technical problems, bans a customer and check all case of promotion.

**Use Promotion:** It is an advantagewhich is given by sysAdmin to renter. A renter has a promotion pays less money for a flat.

**Rating**: There are two ratings type in this system; given by host or renter about their ideas each other, given for a flat. (For example: 5 stars (excellent),4 stars (very good),3 stars(good), 2 stars (not good), 1 star(bad)).

**Reserve Flat:** This action is realizing by renter. This is also a request which must accept or decline from host within a certain time**.**

**Make Payment:** This action is realizing by renter agaisnt host’s flat to use various type of payment such as credit card, debit card.

**Withdraw Balance:** It is transferred money from a renter to a host.

**Search Flat:**  Without a registeration or a login a customer which enters this system searches flat by a city, population, location, address, date.

**List Flat:** Without a registeration or a login a customer which enters this system list flat by price, roomtype, location, address, date etc.

**Open Issue:** An issue which contains a complain with an explaination to the sysAdmin.The complain may contains about a flat or a customer.

**Apply Filter:** Without a registeration or a login a customer may apply search a flat by some criteria such as; price, roomtype etc.

**Report Customer Issue:** An negative topics or comment is entered by customer. It solves by sysAdmin.

**Select Payment Method:** Choose payment method such as credit card, debit card.

**Ban Customer:** May system admin ban customers who complain another use in issue report within closing the issue.

**Display Flat Profil:** Showing Flat’s information page.

**Specify Filter:** The customer to search according to the required criteria such as filter by city, filter check in check out time, filter by population, filter by address, filter by location.

**Edit Flat Profile:** Flat profile can change Flat's infor such as flat address.

# References

1. <https://firebase.google.com/features/>
2. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.
3. <http://www.uml-diagrams.org/class-diagrams-examples.html>
4. <http://univera-ng.blogspot.com.tr/2010/03/uml-ve-modelleme-bolum-7-sequence.html>

5. https://www.airbnb.com.tr