****

**Bilkent University**

**CS 491**

**Senior Design Project I**

**Fall 2013**

**On Fire**

***Analysis Report***

November 4th, 2013

Group Members  
Baha Uluğ | Barış Şahin | Bengisu Şahin | Berkin Şansal | Sıla Şirin

Supervisor  
Fazlı Can

Jury Members  
Tuğrul Dayar | Hakan Ferhatosmanoğlu

Innovation Expert  
 Serkan Köse

List of Table

[List of Figures iii](#_Toc371341143)

[1. Introduction 4](#_Toc371341144)

[2. Current System 5](#_Toc371341145)

[3. Proposed System 7](#_Toc371341146)

[3.1. Overview 7](#_Toc371341147)

[3.2. Functional Requirements 7](#_Toc371341148)

[3.3. Non-Functional Requirements 8](#_Toc371341149)

[3.4. Pseudo Requirements 9](#_Toc371341150)

[3.5. System Models 10](#_Toc371341151)

[3.5.1. Scenarios 10](#_Toc371341152)

[Scenario 1: Create a profile 10](#_Toc371341153)

[Scenario 2: Fire up 10](#_Toc371341154)

[Scenario 3: Add friend 10](#_Toc371341155)

[Scenario 4: Become an owner 11](#_Toc371341156)

[3.5.2. Use Case Model 12](#_Toc371341157)

[3.5.3. Object and Class Model 13](#_Toc371341158)

[3.5.4. Dynamic Models 14](#_Toc371341159)

[State Diagrams 15](#_Toc371341160)

[Activity Diagrams 16](#_Toc371341161)

[Sequence Diagrams 17](#_Toc371341162)

[*Scenario 1* 17](#_Toc371341163)

[*Scenario 2* 17](#_Toc371341164)

[*Scenario 3* 18](#_Toc371341165)

[*Scenario 4* 18](#_Toc371341166)

[3.5.5. User Interface – Navigational Paths and Screen Mock-Ups 19](#_Toc371341167)

[Scenario 1 19](#_Toc371341168)

[Scenario 2 20](#_Toc371341169)

[Scenario 3 21](#_Toc371341170)

[4. Glossary 23](#_Toc371341171)

[5. References 24](#_Toc371341172)

# List of Figures

[Figure 1 : Foursquare screenshots. 5](#_Toc371338058)

[Figure 2 : AroundMe screenshots. 6](#_Toc371338059)

[Figure 3: Use case diagram. 12](#_Toc371338060)

[Figure 4 : MVC model. 13](#_Toc371338061)

[Figure 5 : Class diagram. 14](#_Toc371338062)

[Figure 6 : State diagram for possible screens. 15](#_Toc371338063)

[Figure 7 : Activity diagram for adding a video. 16](#_Toc371338064)

[Figure 8 : Activity diagram for searching and adding a friend. 16](#_Toc371338065)

[Figure 9 : Sequence diagram for scenario 1. 17](#_Toc371338066)

[Figure 10 : Sequence diagram for scenario 2. 17](#_Toc371338067)

[Figure 11 : Sequence diagram for scenario 3. 18](#_Toc371338068)

[Figure 12 : Sequence diagram for scenario 4. 18](#_Toc371338069)

# Introduction

By the enhancement of Smart Phones and Internet speed, users can get any information when they need it. However, because of the information pollution, it is very hard to get exact information about something. Especially, in social media there is too much misleading information that leads misconceptions. Our project aims to solve this problem on location-based social media.

As a solution of this problem, our senior project will be a mobile application that allows users to see what is happening in a place in a particular moment by video sharing with social media integration from users and owners of the places. Therefore, videos are best way to give some idea about a place; users could not mislead by other users.

# Current System

* **Foursquare [5]:** Foursquare makes people check in any place. With this app, they can also reach their friends check-in. By this way, they can go a place which got many likes or has a good recommendations.

Our program and Foursquare have mainly similar concept in terms of check-in. In addition to Foursquare, our program is unique on uploading videos at a particular moment. By this way, videos will have a role which will lead a place to come into prominence.



Figure 1 : Foursquare screenshots.

* **Around Me [6]:** This application shows you nearby places and also places according to search results. There are categories to filter the results. You can see nearby places in a list, in a map, or in a camera.

This program shows places on camera by augmented reality. We will have a feature that which will show places and users. By launching the camera in the app, nearby places and people will be appear according to their locations. Location-based augmented reality technology will help us to improve this [4]. Your friends, other crowd, and favorite places will be under your hand with their accurate locations and directions.

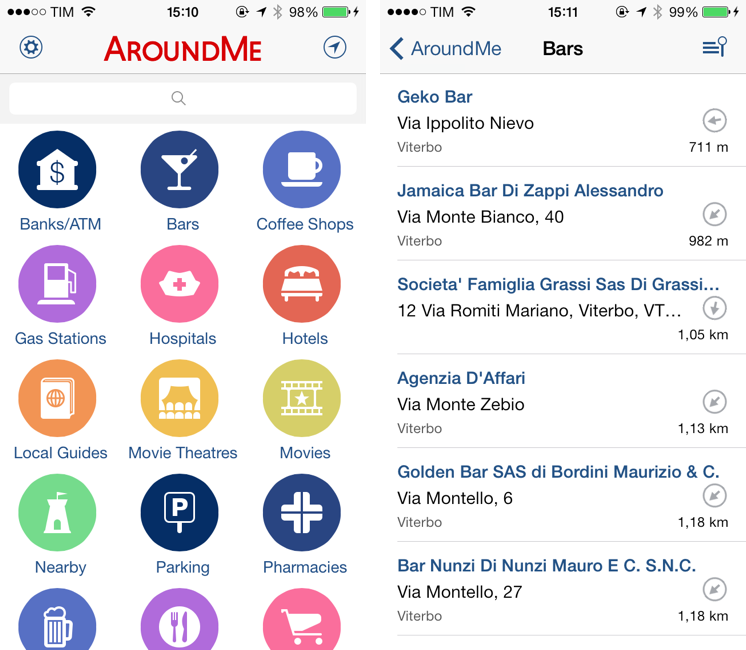


Figure 2 : AroundMe screenshots.

# Proposed System

## Overview

Onfire will be a mobile application for people who like to explore new places by sharing and watching videos about a place at a particular moment. Also, with social media integration and augmented reality, system will help users to share their locations and see their distances from their friends.

Some technical details about Onfire are listed below.

* **Platform:** iOS [7]
* **Language:** Objective-C [7]
* **Architecture:** MVC
* **Design Patterns:** Singleton
* **Database:** Amazon S3 [3]

## Functional Requirements

* **Video Fire-up**

The main purpose of the application will be video fire-up. When user wants to fire-up in a place where he is, he has to share a video of that place which will take between 3 to 15 seconds. This video can be shared either public, so it will be saved under place page, or just for friends.

* **Creating Profile & Facebook Login**

When application runs for first, it will ask for signing in or signing up. User can use their Facebook account to log in easily and directly.

* **Looking Nearby Places & People**

Users can open maps or camera which will be provided inside the application in order to see nearby places and people. Camera will be developed thanks to augmented reality technology [4].

* **Following Places & Making Friends**

User can follow places to see updates and recent videos in that places. In addition, users can add friends to follow their actions.

* **Searching Places**

By searching places, user can reach what’s happening in places which are not even nearby to user’s location.

* **Settings**

By settings, user can find their friends, change their account, privacy and notification settings, and reach information about application and user terms.

## Non-Functional Requirements

* **Performance**

Our application will require low system preferences on mobile. We will try to make codes simple to work on almost all previous generation of mobile phones existing in these days. Because of maps-in-app and camera-in-app, system will require more RAM but we want them to use considerably low memory. Moreover, our application should upload videos fast, so size will be limited to reduce both upload time and requirement of data from user’s network package.

* **Usability**

Young and adult users can use this application without any trouble. In simple “Help” part in “Settings”, the whole information about the app will be given with icons and pictures. In order to be user friendly, there is no unnecessary buttons in the screens. Moreover, most of the mobile phones can meet the application system requirements.

* **Data Management**

Data management will be handled both by storage of telephone and by web server. Amazon web services will be used as server, cloud storage and database [3].

* **Reliability**

Because of low system requirements, our application will not face with complicated situation or any time delay during the process. However network connection will be the most important part of the application to overcome the upload, fire-up, and retrieving information problems, but with network checking in every steps, we will prevent the troubleshoots.

## Pseudo Requirements

* Objective C language with xCode IDE [7] will be used to implementation of the application.
* Amazon S3 cloud storage service [3] will be used for storing comments, likes fires, videos.
* Foursquare’s open source database [2] will be used for locations of places.
* Facebook SDK [8] used for sign-up operations.

## System Models

### Scenarios

#### *Scenario 1: Create a profile*

Barış is a member and downloads OnFire from App Store to his iPhone. He starts using OnFire by signing up to OnFire network with his first name, last name, password and e-mail. After the account creation, he can see the home page screen and start to use OnFire. Additionally, from home page screen, he clicks “Settings” Button in order to arrange and configure his profile. At this page, he sees some categories related to profile settings. After that he clicks “Privacy Settings” button and changes his profile privacy from public to privacy. Then, he completes the configuration process.

#### *Scenario 2: Fire up*

Baha is a member and wants to fire up at Kitchenette. In order to do this action, first of all, he finds “Search Place” section from home page screen and searches “Kitchenette”. After that, he sees result list for Kitchenette. After choosing a place from results of Kitchenette, he clicks selected place “Kitchenette” in order to fire up there. At Kitchenette’s page, he clicks “Fires up Kitchenette” button, uploads Berkin’s Birthday video and finally writes “Party Rocks” as a video name. Then, he clicks “Fire Up” button to share this event with his friends.

#### *Scenario 3: Add friend*

Sıla is a member and wants to add her friend “Berkin Şansal” to her OnFire network to follow their fire activities. Sıla searches name “Berkin Şansal” in OnFire’ search box so as to add him as a friend. After adding Berkin, the system sends a notification message to Berkin.

#### *Scenario 4: Become an owner*

Bengisu is an owner and he desires to own her restaurant “BigChiefs” to OnFire network. On the home page screen, she searches BigChiefs and enters BigChiefs’ page. On this page, he sends a “Request to be owner” button and makes a request to own BigChiefs. Then, the system sends a notification message to system admins. If, there is no problem for her request, they approve Bengüsu’s request and add the system as an owner of BigChiefs

### Use Case Model

The figure below shows use case model of the system.

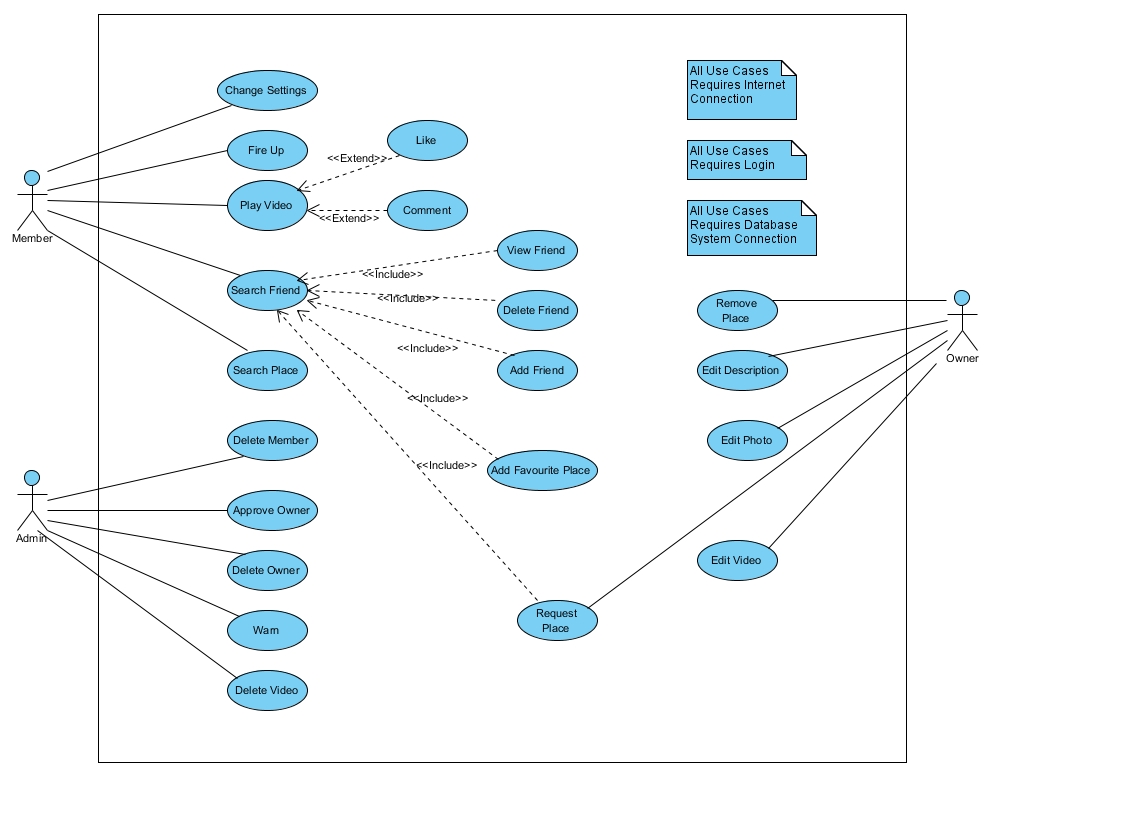


Figure 3 : Use case diagram.

### Object and Class Model

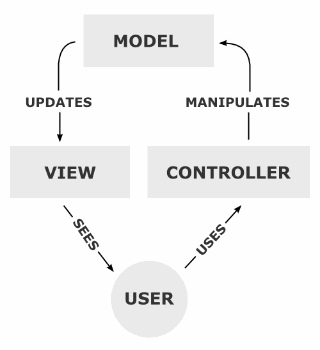


Figure 4 : MVC model.

Our application will provide lots of data by showing videos fires etc. Since there are lots of data flow, Model view controller (MVC) will be used for preventing the mess between user interface and data flow. Also, MVC design pattern is commonly used for iOS applications which are based on social platform.

**Model:** Basic functionalities, main events and database connection will be provided here. Also, this layer is responsible to notify view and controller when a state changed in the program (selecting a different page).

**Controller:** User will be able to interact with this layer. By interacting with controller, user can send commands such as searching places, friends and videos to the Model layer by the help of the controller. This layer consists of AppEngine Class.

**View:** View layer request information from the model and gets an output for showing in a page. This layer consists of UI Manager`s parts which are described in Class Diagram.

Class diagram which contains software representations of database objects is below.

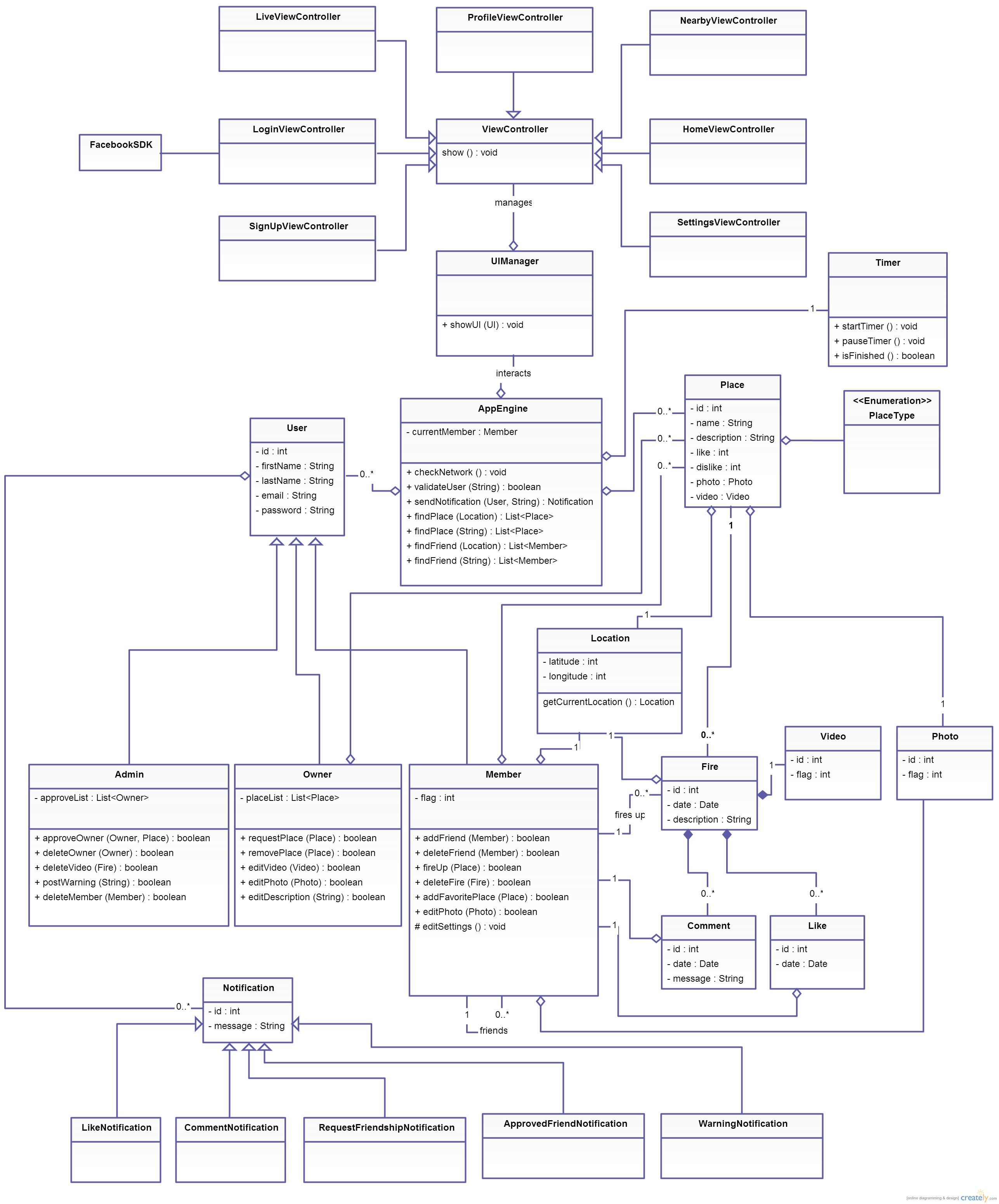


Figure 5 : Class diagram.

### Dynamic Models

#### State Diagrams

The figure below shows the possible states of main screen while using the application.

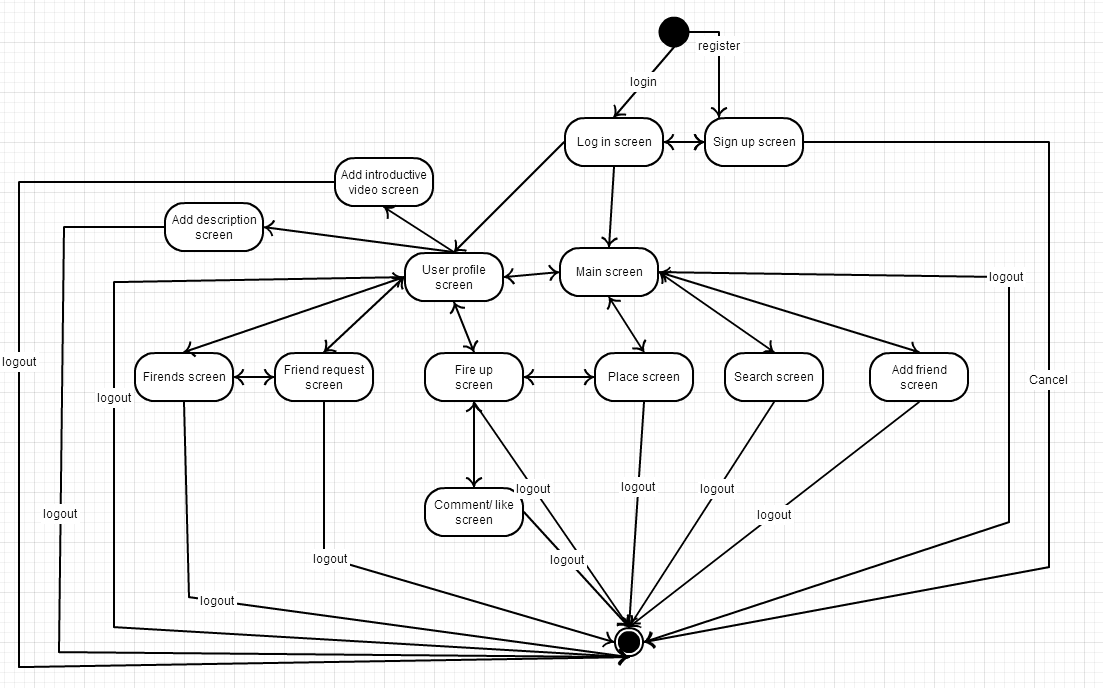


Figure 6 : State diagram for possible screens.

#### Activity Diagrams

**Add video (Fire up)**

The figure below shows the activity diagram for the Fire up option.

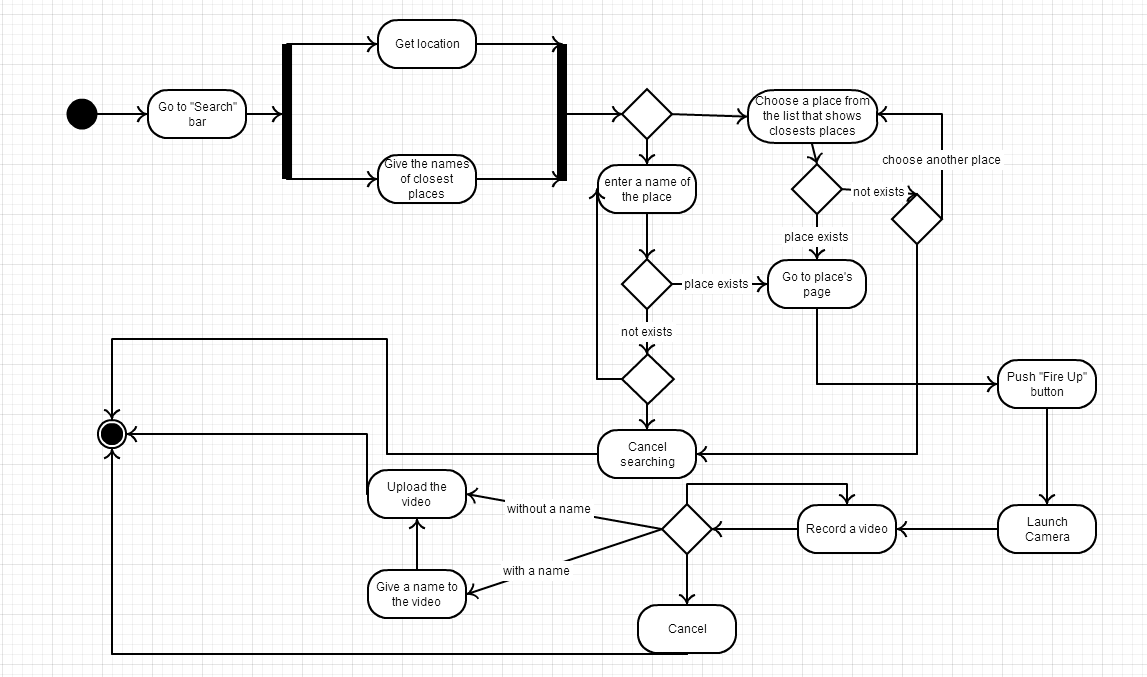


Figure 7 : Activity diagram for adding a video.

**Search and add a friend**

The figure below shows the activity diagram for searching and adding a friend.

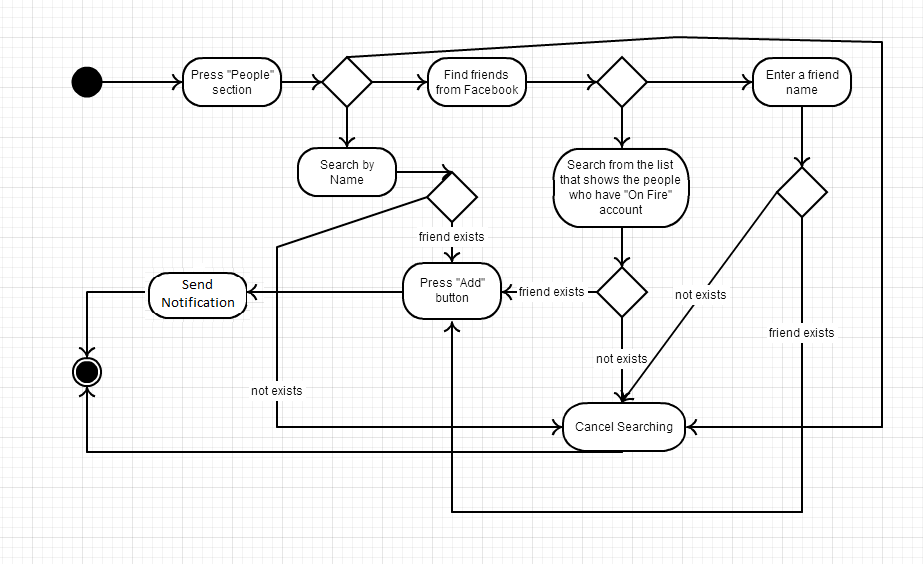


Figure 8 : Activity diagram for searching and adding a friend.

#### Sequence Diagrams

##### Scenario 1

The figure below shows the sequence diagram for scenario 1.

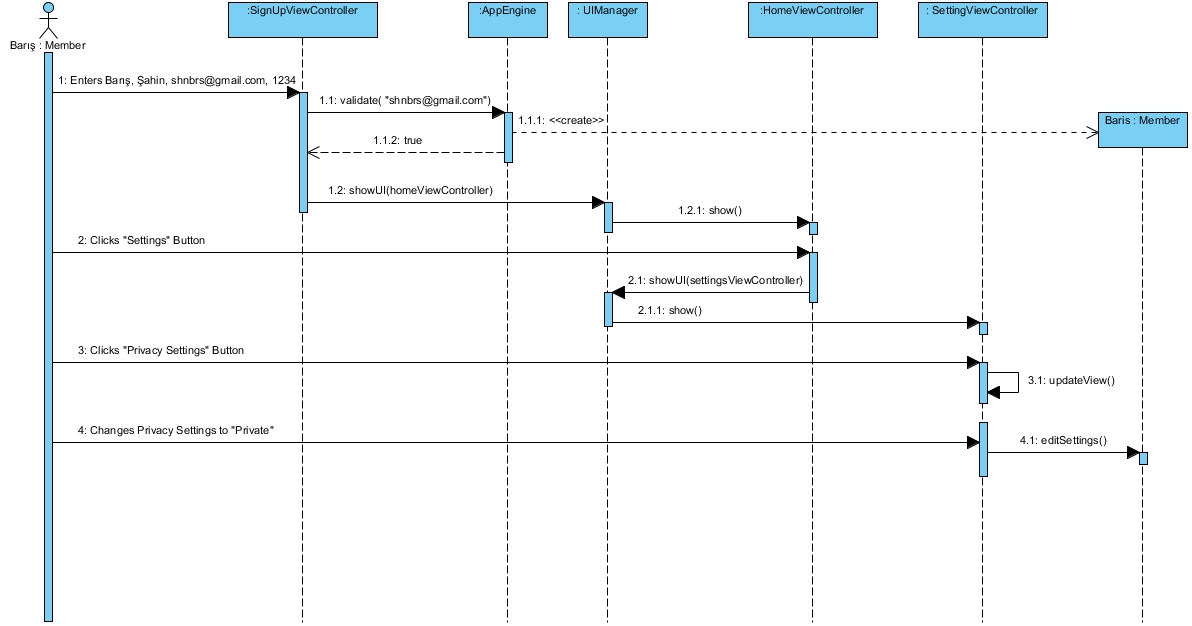
******

Figure 9 : Sequence diagram for scenario 1.

##### Scenario 2

The figure below shows the sequence diagram for scenario 2.

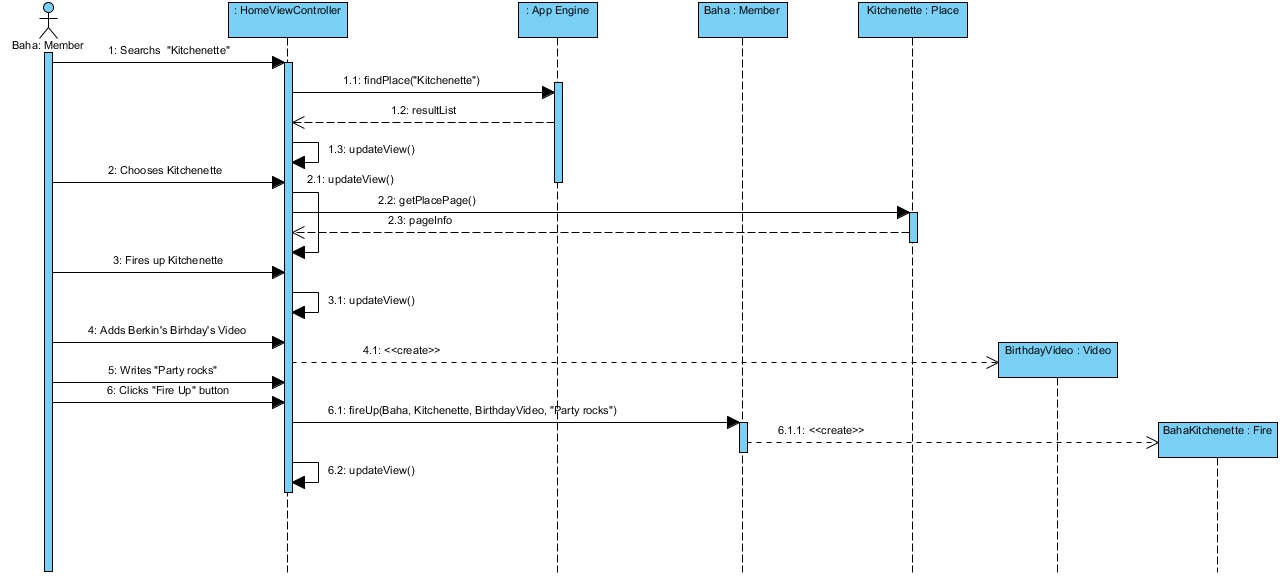
******

Figure 10 : Sequence diagram for scenario 2.

##### Scenario 3

The figure below shows the sequence diagram for scenario 3.

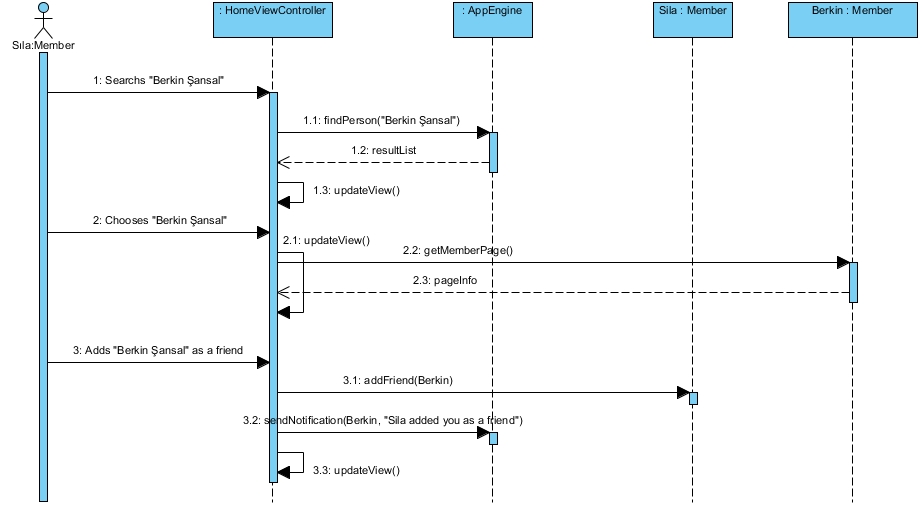
******

Figure 11 : Sequence diagram for scenario 3.

##### Scenario 4

The figure below shows the sequence diagram for scenario 4.

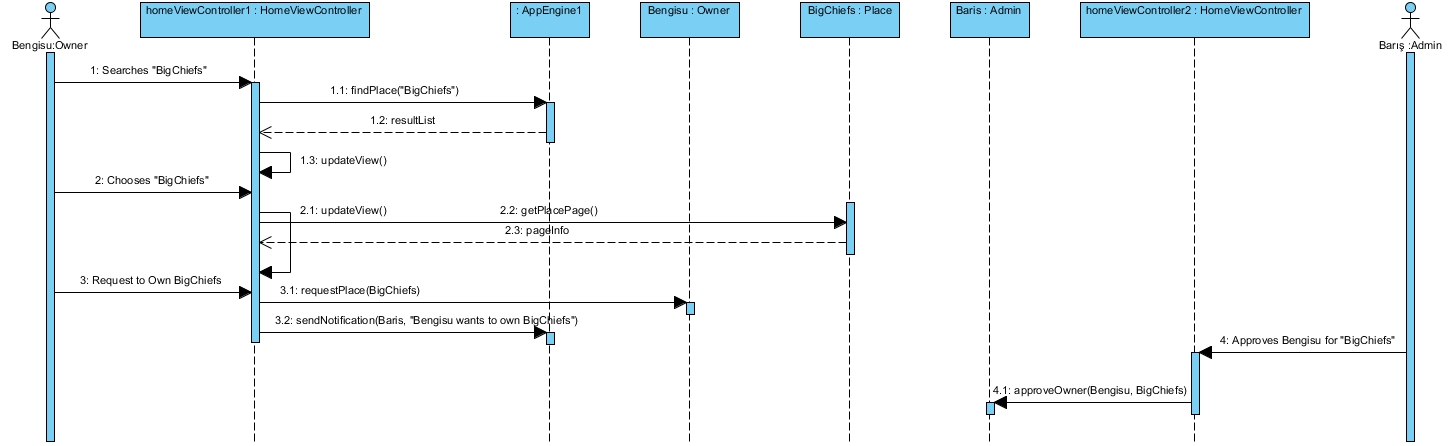
******

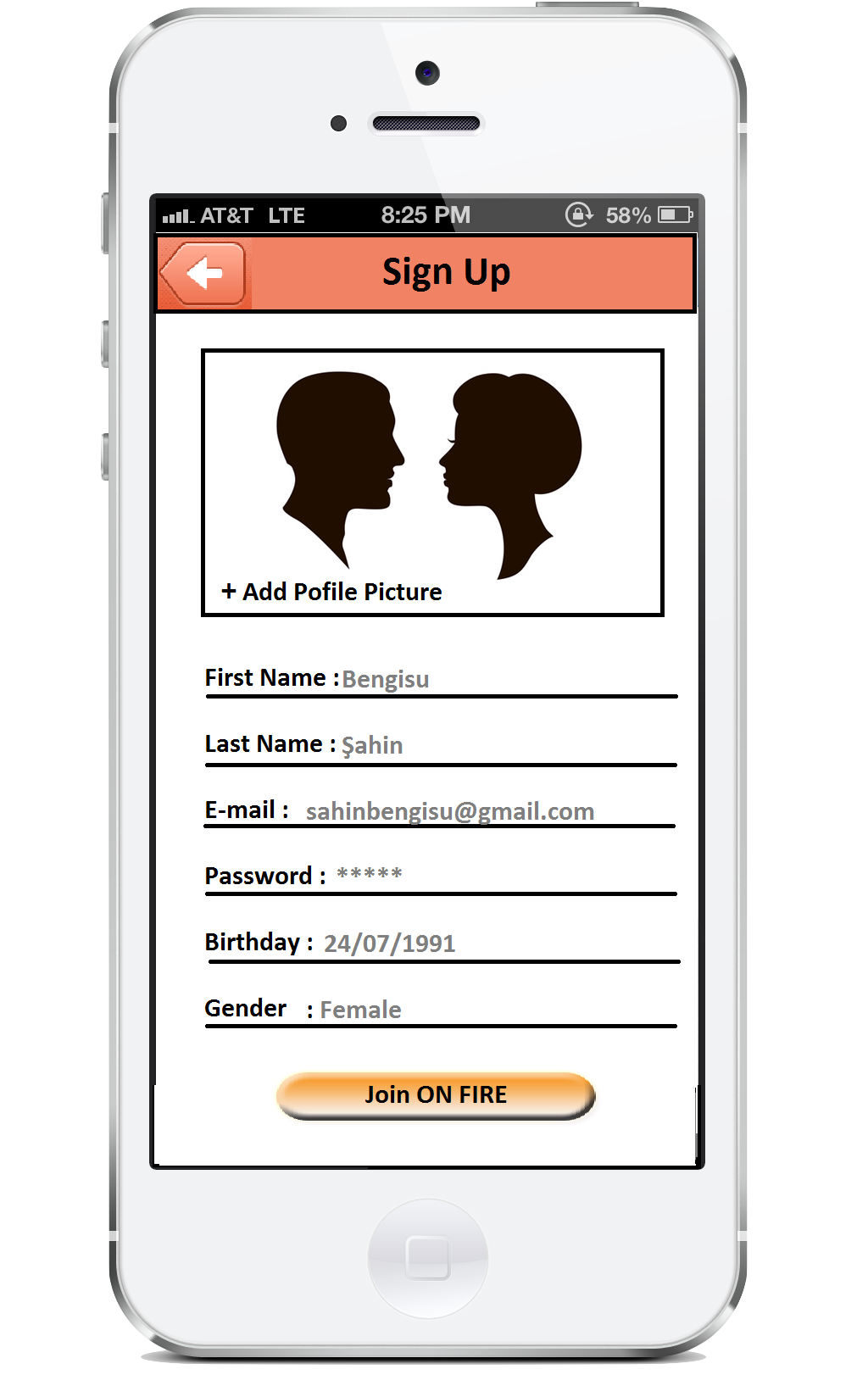
Figure 12 : Sequence diagram for scenario 4.

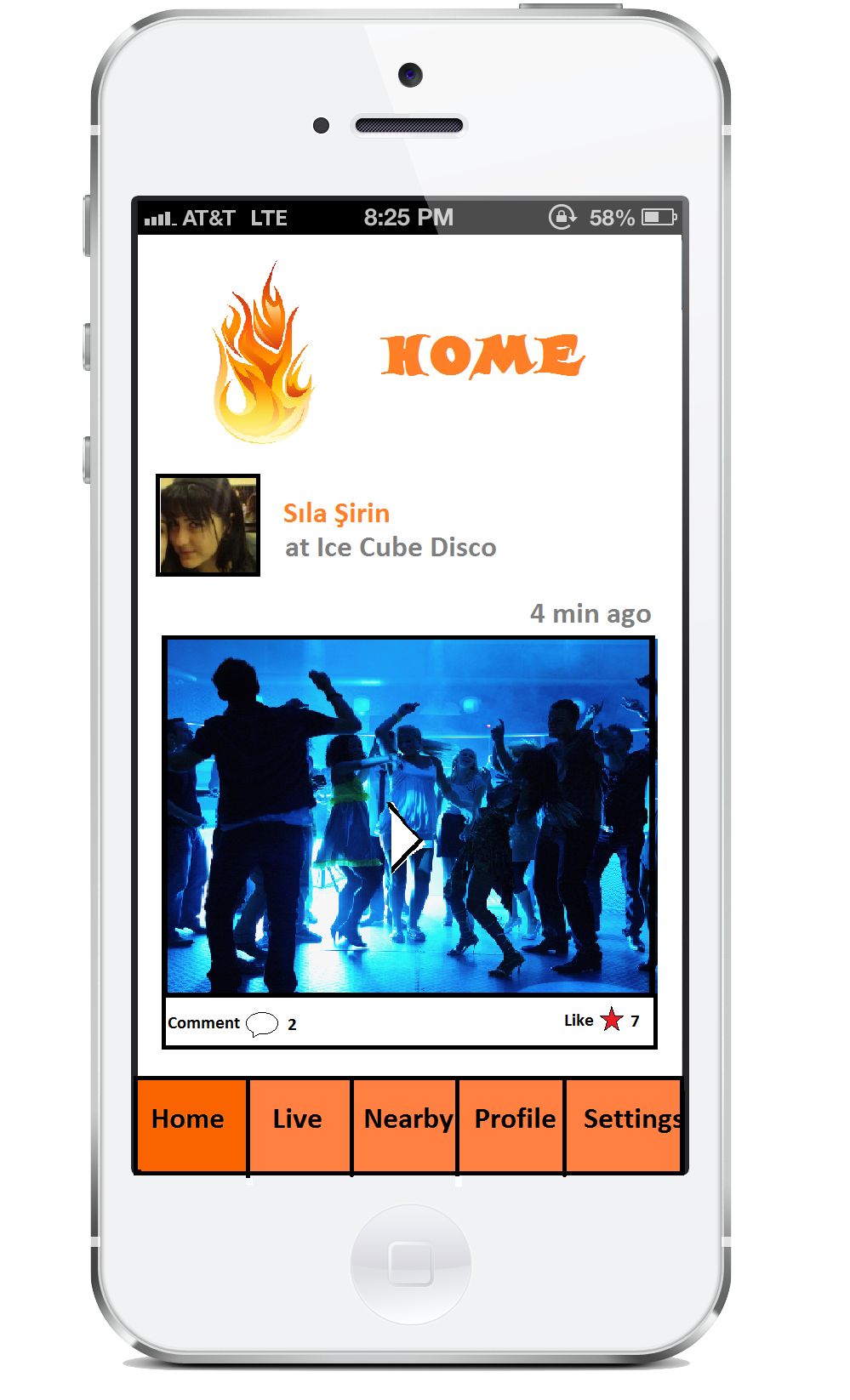
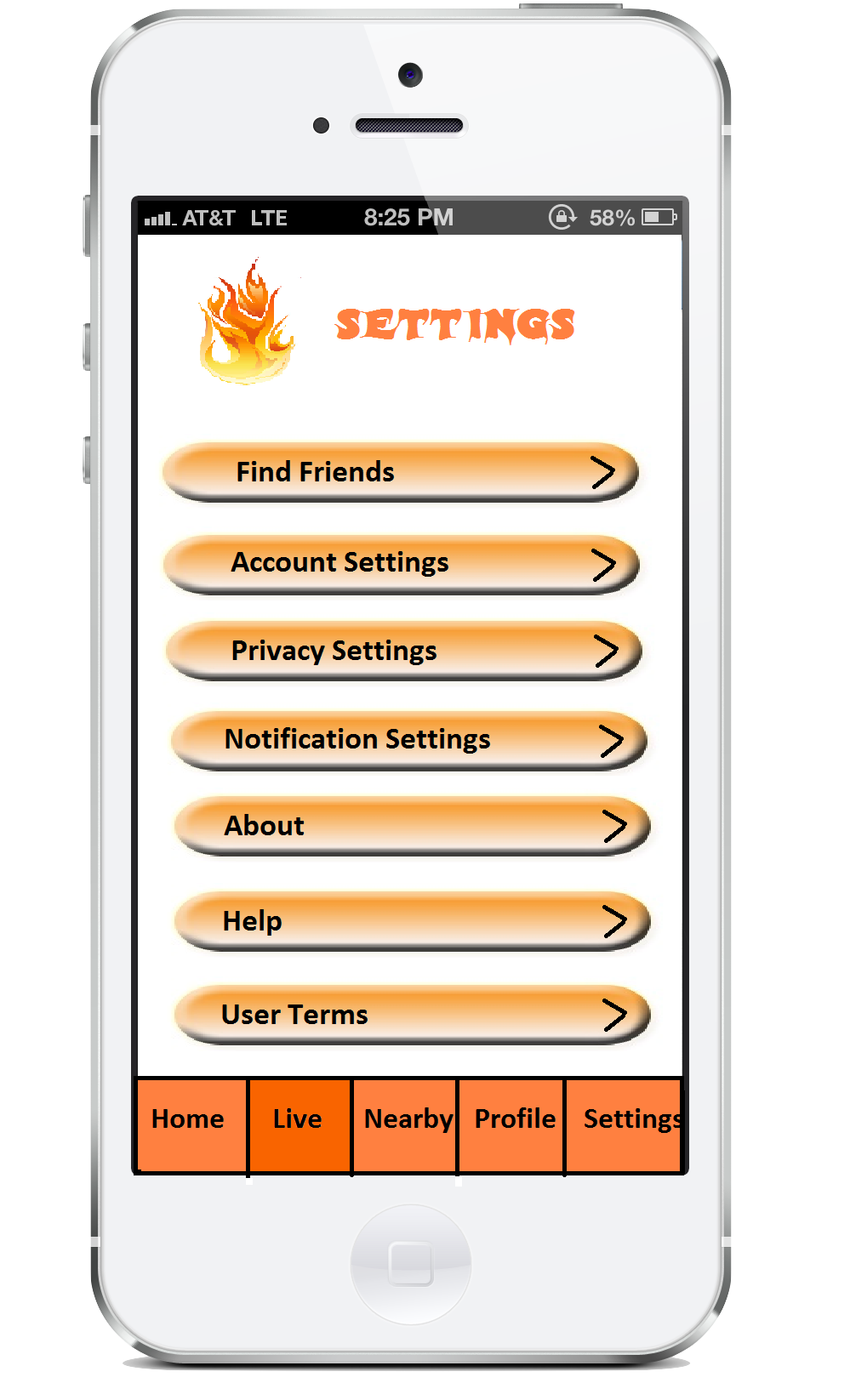
### User Interface – Navigational Paths and Screen Mock-Ups

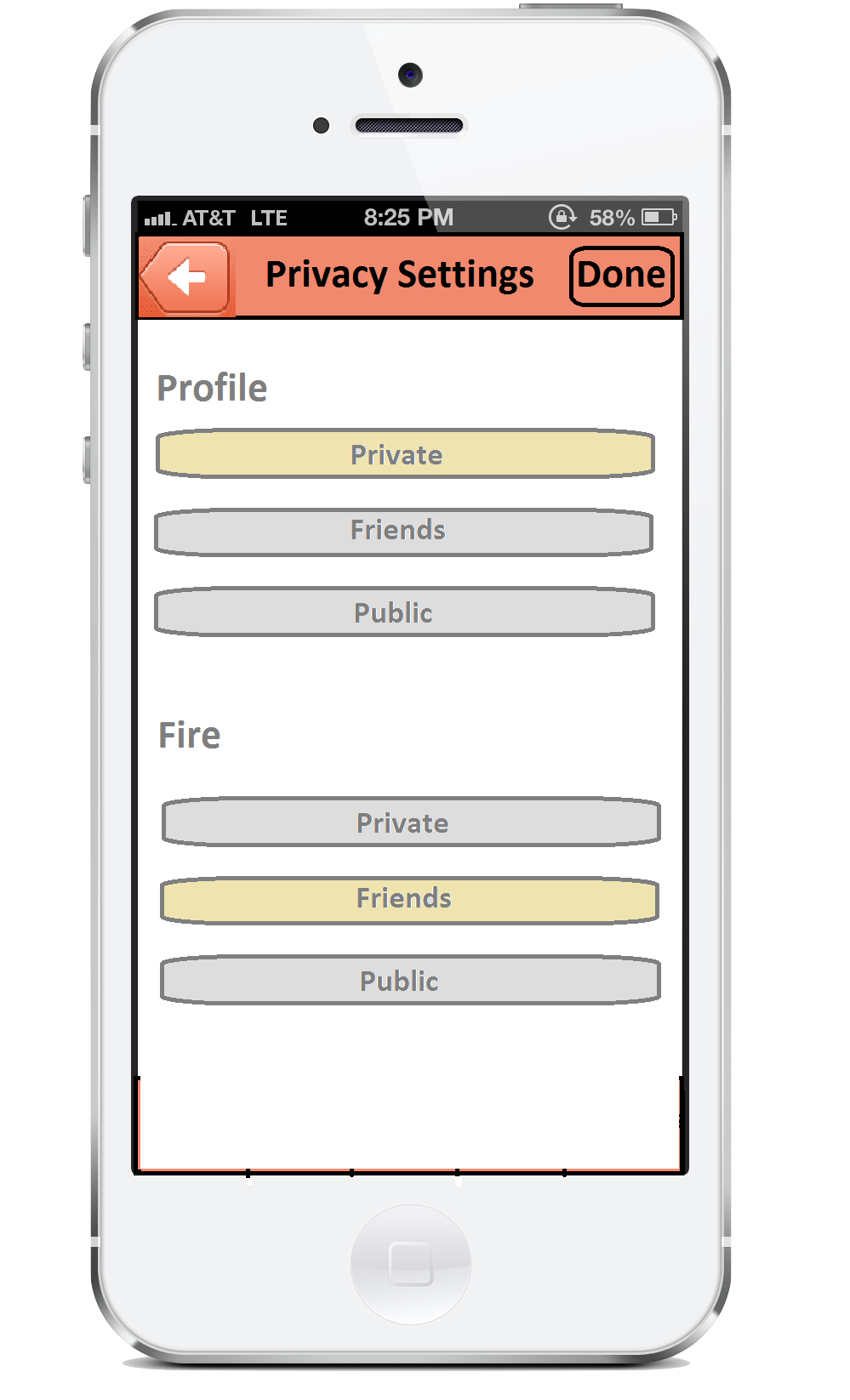
\* iPhone image used in mockups was taken from a website [1].

#### Scenario 1

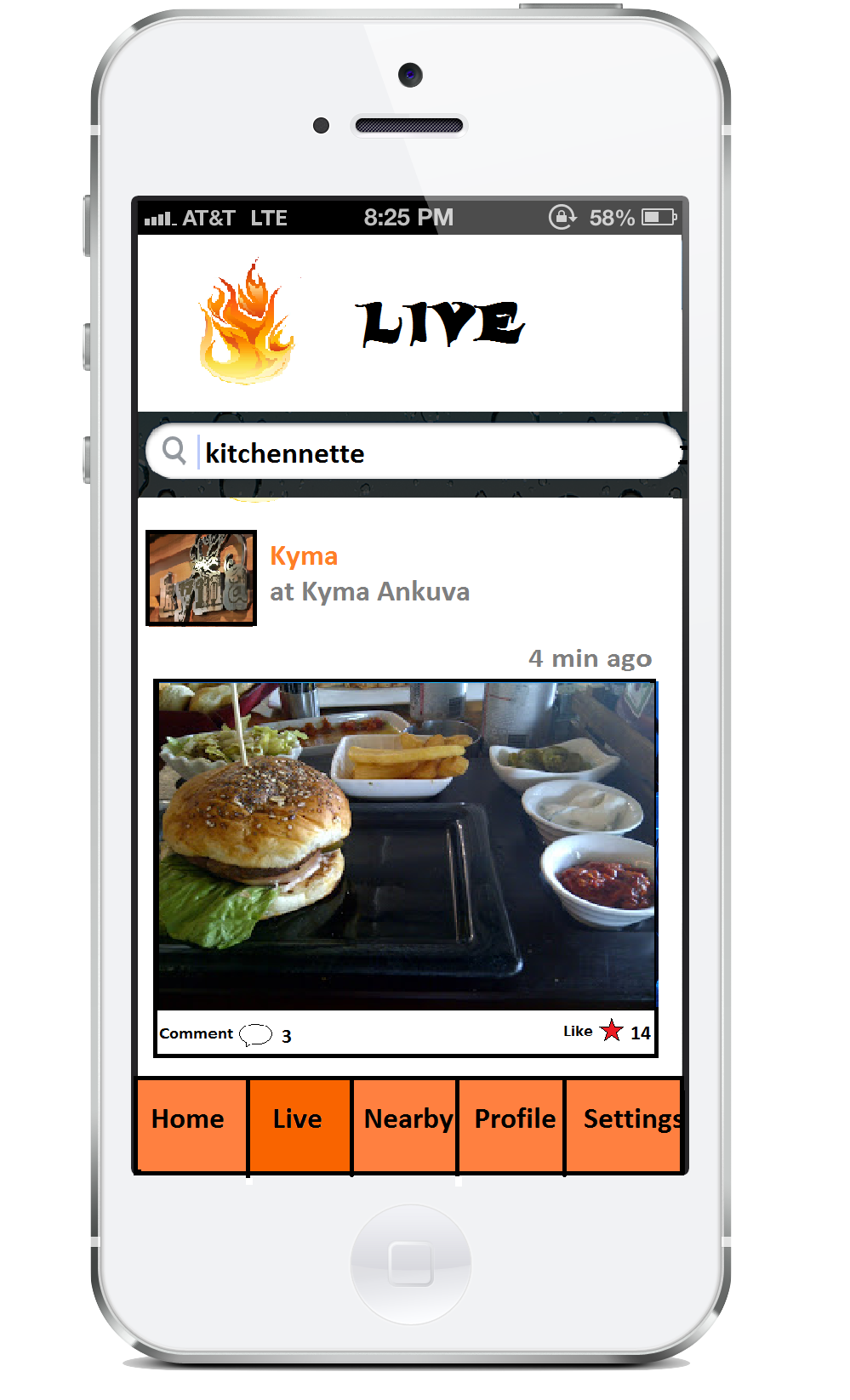
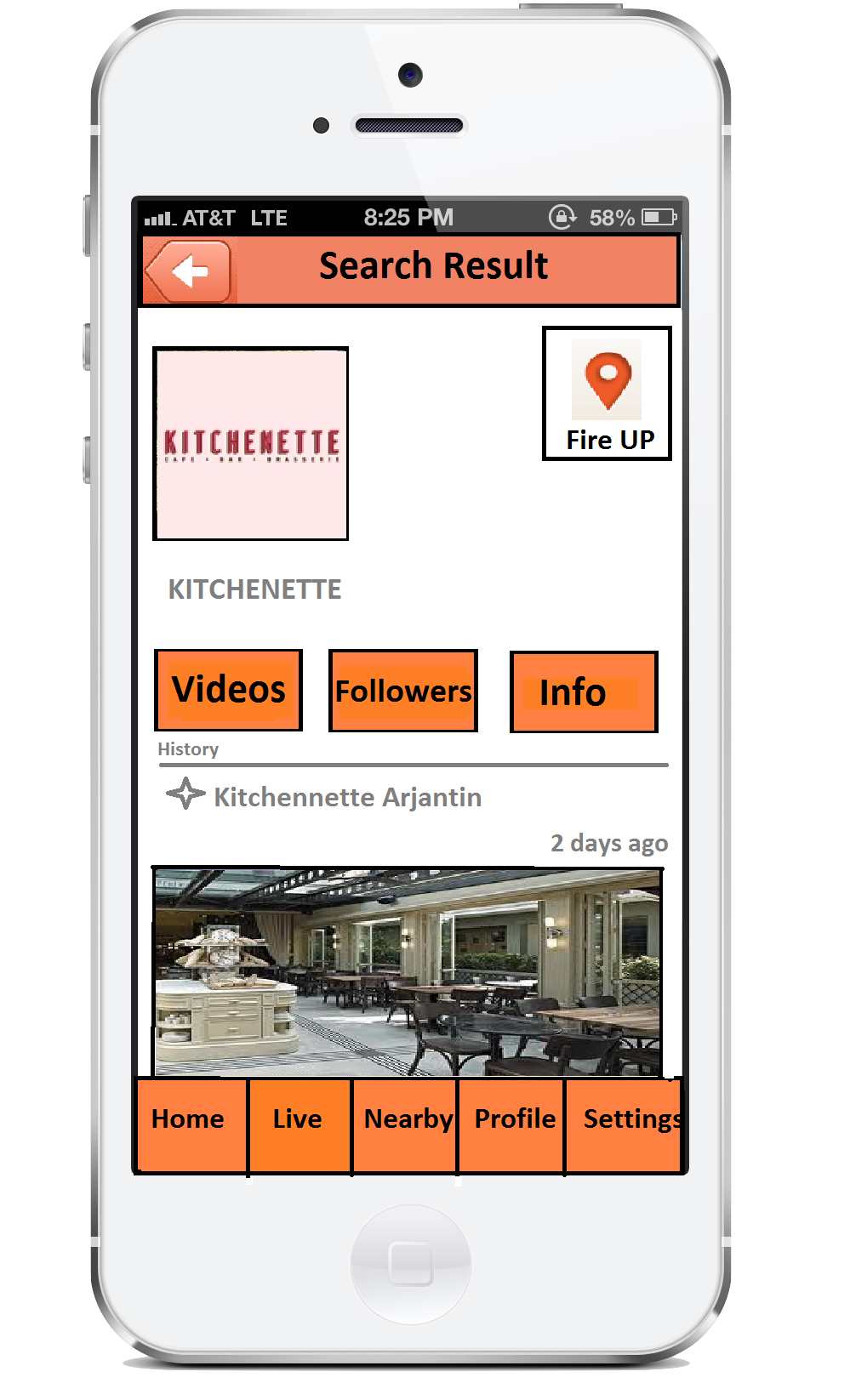
******

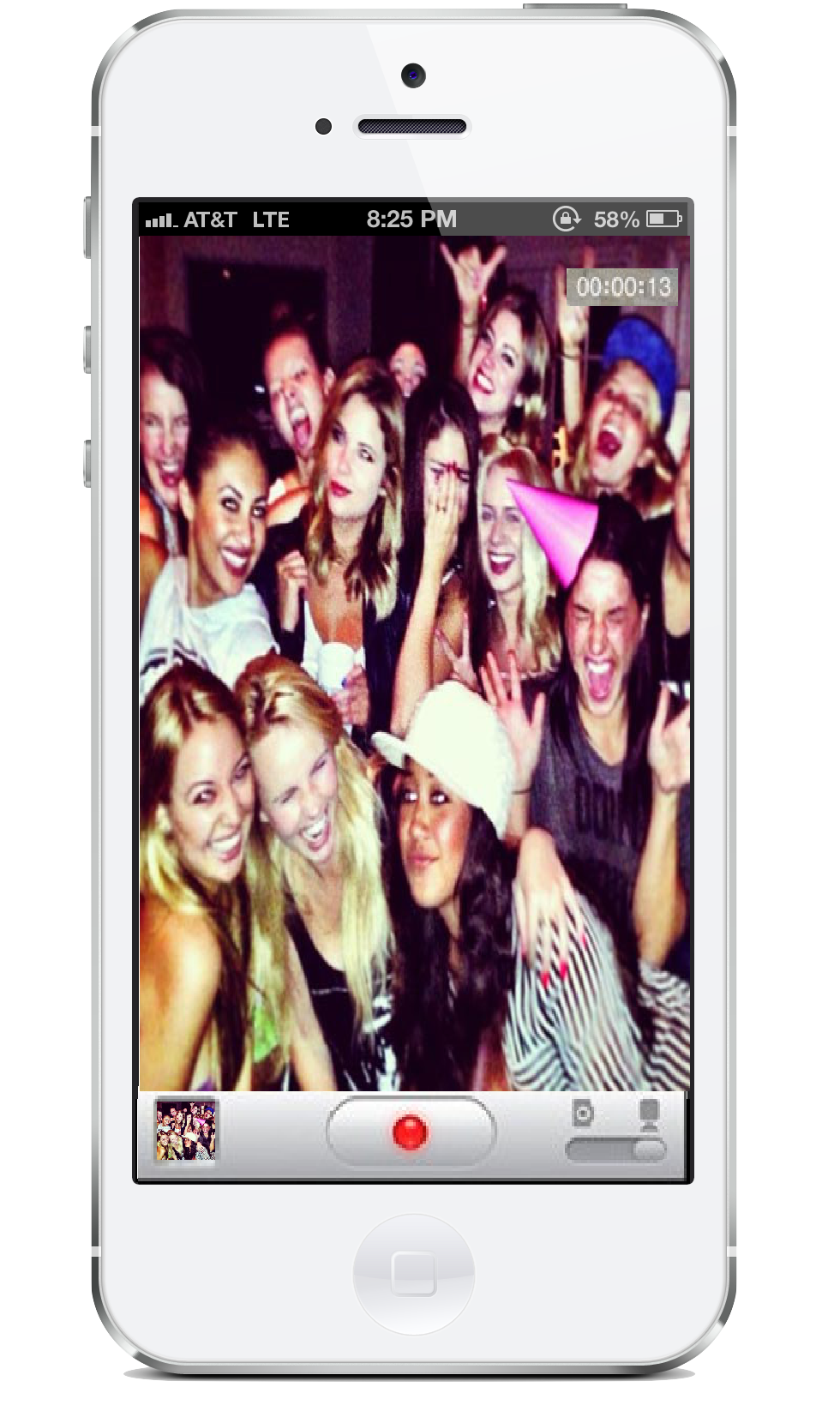
******

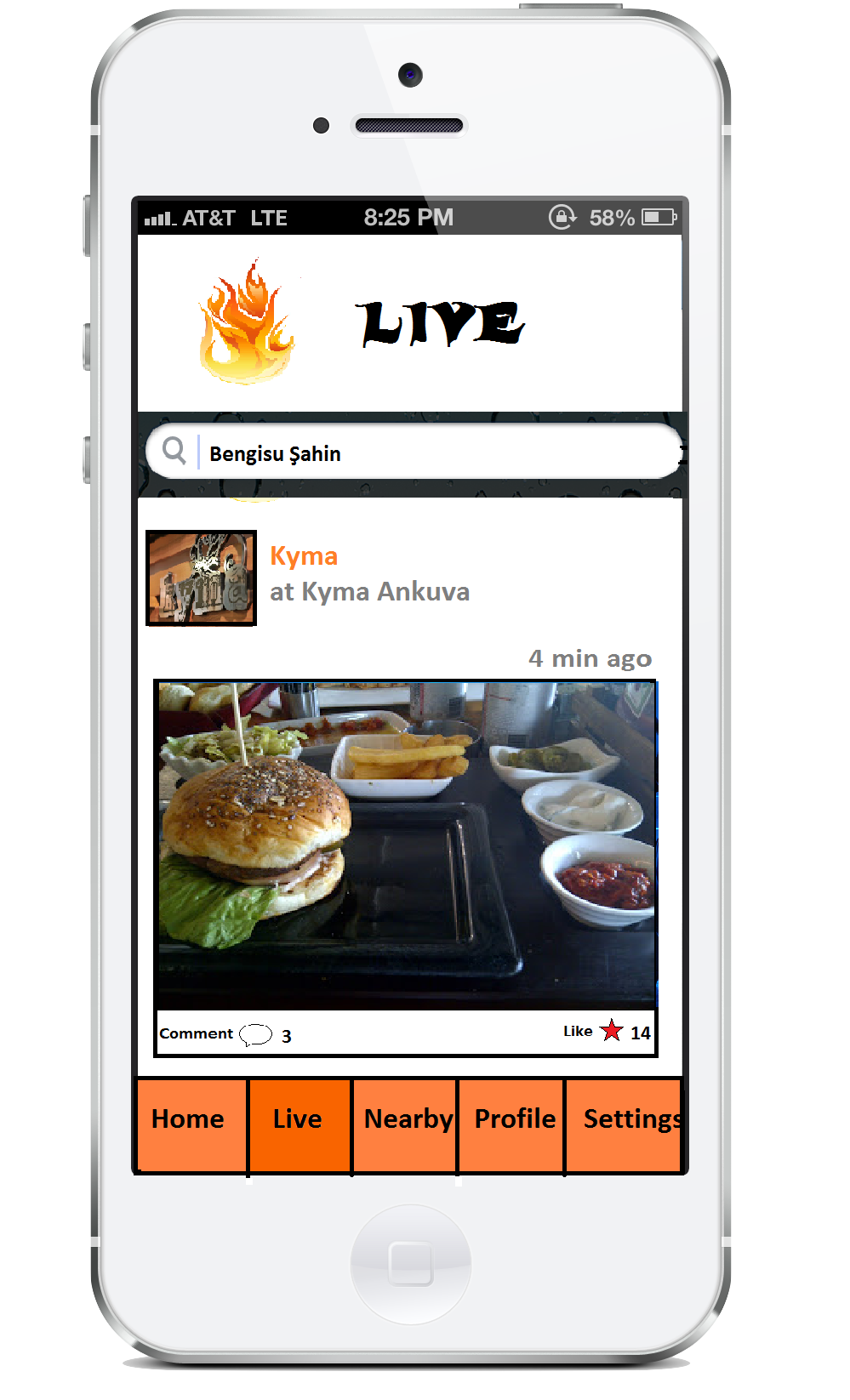
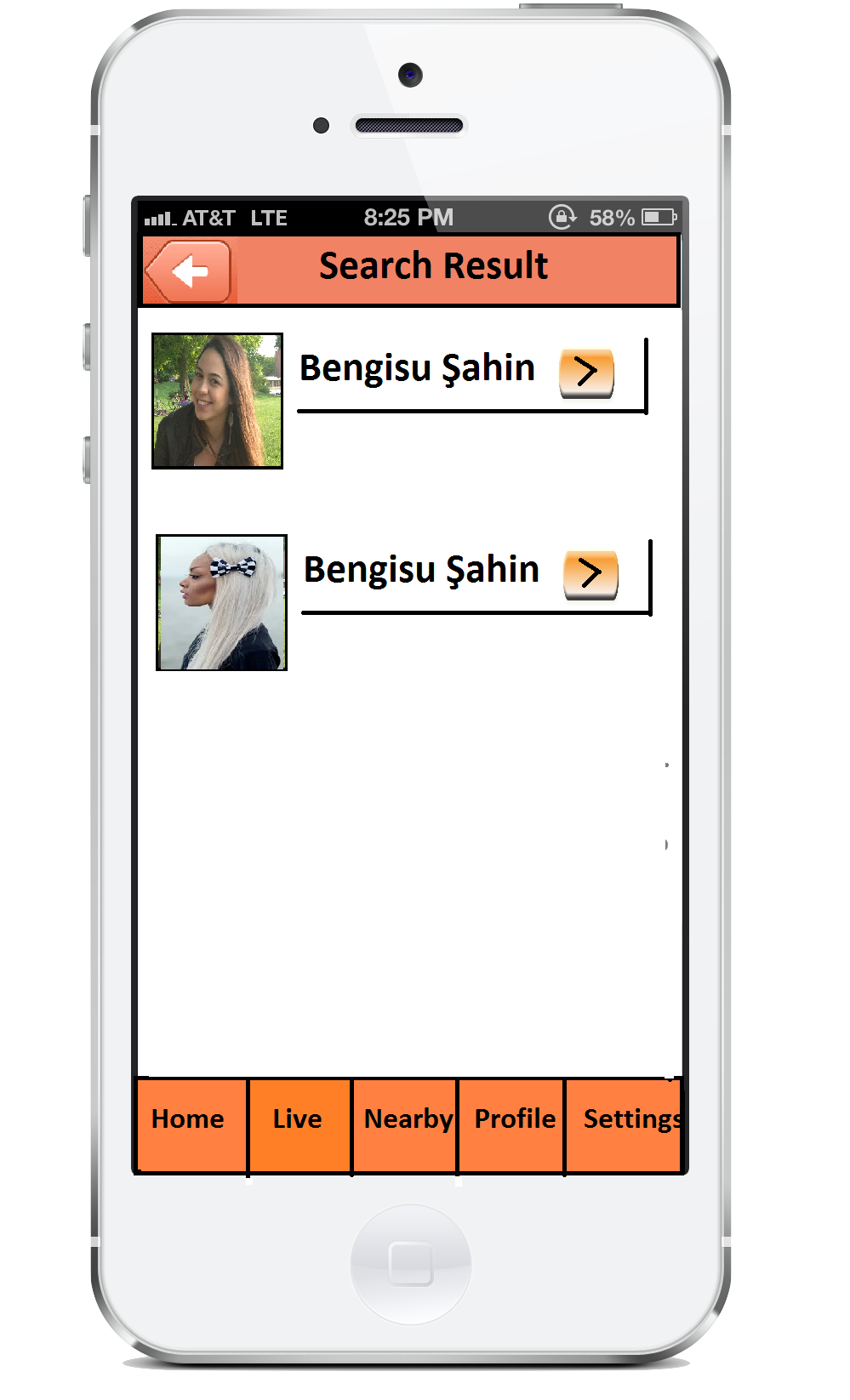


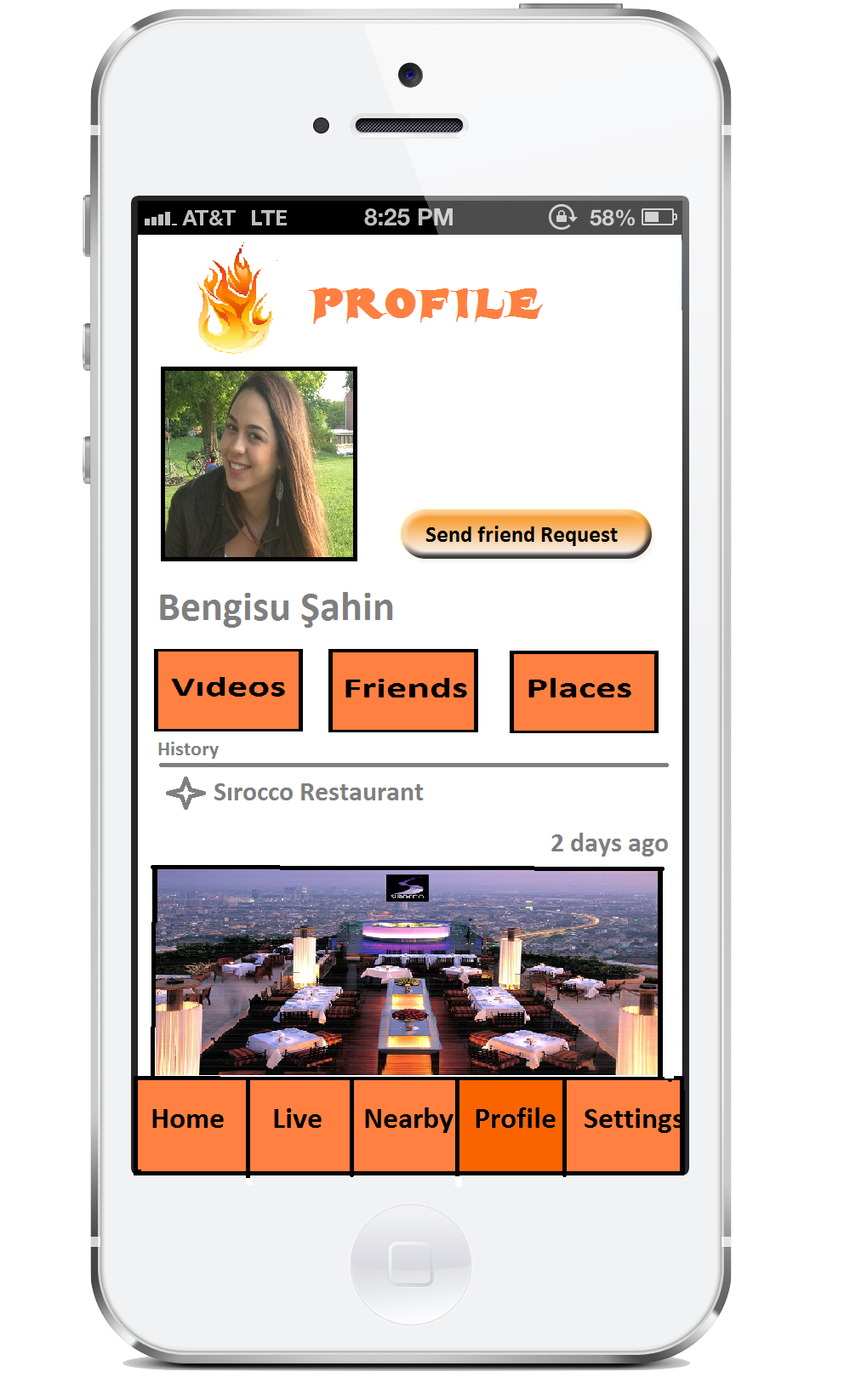
#### Scenario 2

#### Scenario 3

****** ******

******

# **Glossary**

|  |  |
| --- | --- |
| **Keyword** | **Definition** |
| Fire | Name for the activity which made in a place |
| Fire up | The verb form of fire |
| Video | Video is used in each fire |
| Description | Explanation either for places or for fires |
| Comment | Users’ idea either for places or for fires |
| Like | Positive feedback either for places or for fires |
| Dislike | Negative feedback for places |
| Nearby Places | Places around a specific location |
| Favorite | Adding a place to users special list |
| Member | Who can fire up and use app with its main purposes |
| Owner | Who can add a place to his place list |
| Admin | Who controls and manages content by users and owners |

# References

[1] . (2011). *Apple iPhone 4S* [Online]. Available:  
<http://www.geekandhype.com/apple-iphone-4s-8830/iphone-4s-white/>

[2] . (2013). *Foursquare API* [Online]. Available:   
<https://developer.foursquare.com/>

[3] . (2013). *Web, Mobile, and Social Apps on AWS* [Online]. Available: <http://aws.amazon.com/web-mobile-social/>

[4] . (2013). *Augmented Reality iOS Tutorial: Location Based* [Online]. Available: <http://www.raywenderlich.com/42266/augmented-reality-ios-tutorial-location-based>

[5] . (2013). *Foursquare* [Online]. Available:  
<https://itunes.apple.com/tr/app/foursquare-find-restaurants/id306934924?mt=8>

[6] . (2013). *Around Me* [Online]. Available: <https://itunes.apple.com/us/app/aroundme/id290051590?mt=8>

[7] . (2013). *Xcode* [Online]. Available:   
<https://developer.apple.com/xcode/>

[8] . (2013). *Facebook SDK for iOS* [Online]. Available: <https://developers.facebook.com/docs/ios/>