Bilkent University
CS 492
Senior Design Project II
Spring 2014

onFire: No Lie, See Live

Low-Level Design Report

February 17th, 2014

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

Contents

List of Figures

1. 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 and true information about the desired search. Especially, in social media there is too much misleading information that leads misconceptions [1]. 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 the best way to give some idea about a place; users could not mislead by other users.

The link to our project website: http://supportonfire.wix.com/onFire

1.1. Object Design Trade-Offs

- **Functionality - Usability:** We gave outmost importance to functionality and usability for our application. It means that a user-friendly interface can provide us to have many and wide range of users. Therefore, our application should be quite simple and clear in terms of usability and functionality. Since we aim to make users' works easier, we prefer usability to functionality. The users will not need to waste much time to understand our system, they can easily use it [4][5].

1.2. Interface Documentation Guidelines

The guidelines for the class interface is below:

- **Class Name:** *ClassName*
- **Description of the Class:** *Description*
- **Attributes**
  - *Attribute1*
  - *Attribute2*
  - *…*
- **Operations**
  - *Operation1*
  - *Operation2*
  - *…*
- **Class Interactions**

- *Class1, Class2*

1.3. Engineering Standards

The Unified Modeling Language (UML) is used for graphic notations in our project [3]. For references, American Psychological Association (APA) style is used [2]. UML and APA are most commonly used techniques in their related areas.

1.4. Definitions, Acronyms, and Abbreviations

The following list is a brief list of acronyms and abbreviations;

- SQL: Structured Query Language
- DB: Database
- UI: User Interface
- PDF: Portable Document Format
- PHP: Hypertext Preprocessor
- FMDB: Flying Meat Database

2. Packages

Packages and their interactions are shown below:

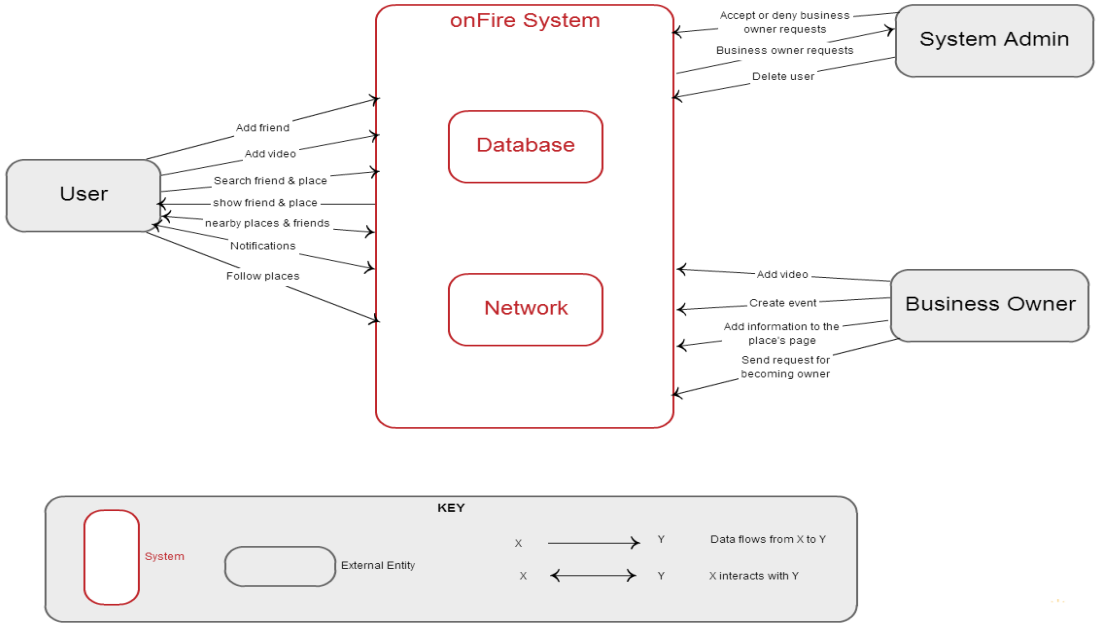

Figure – Context Diagram.

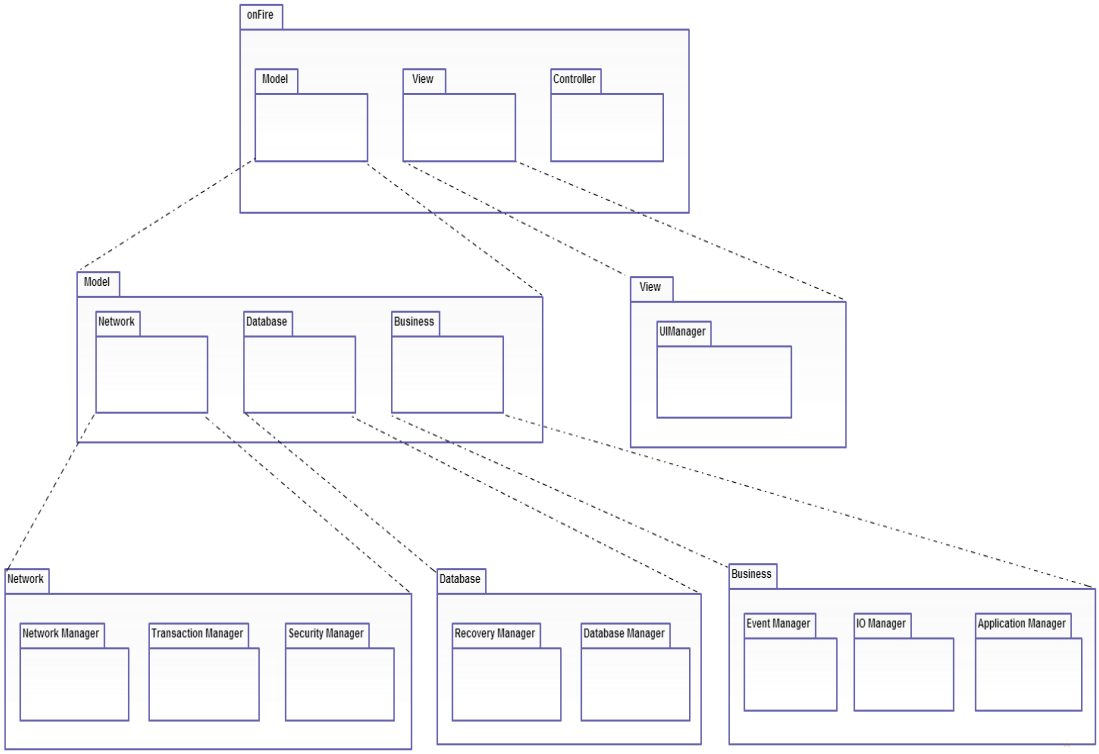

Figure – Packages Diagram.

3. Class Interfaces

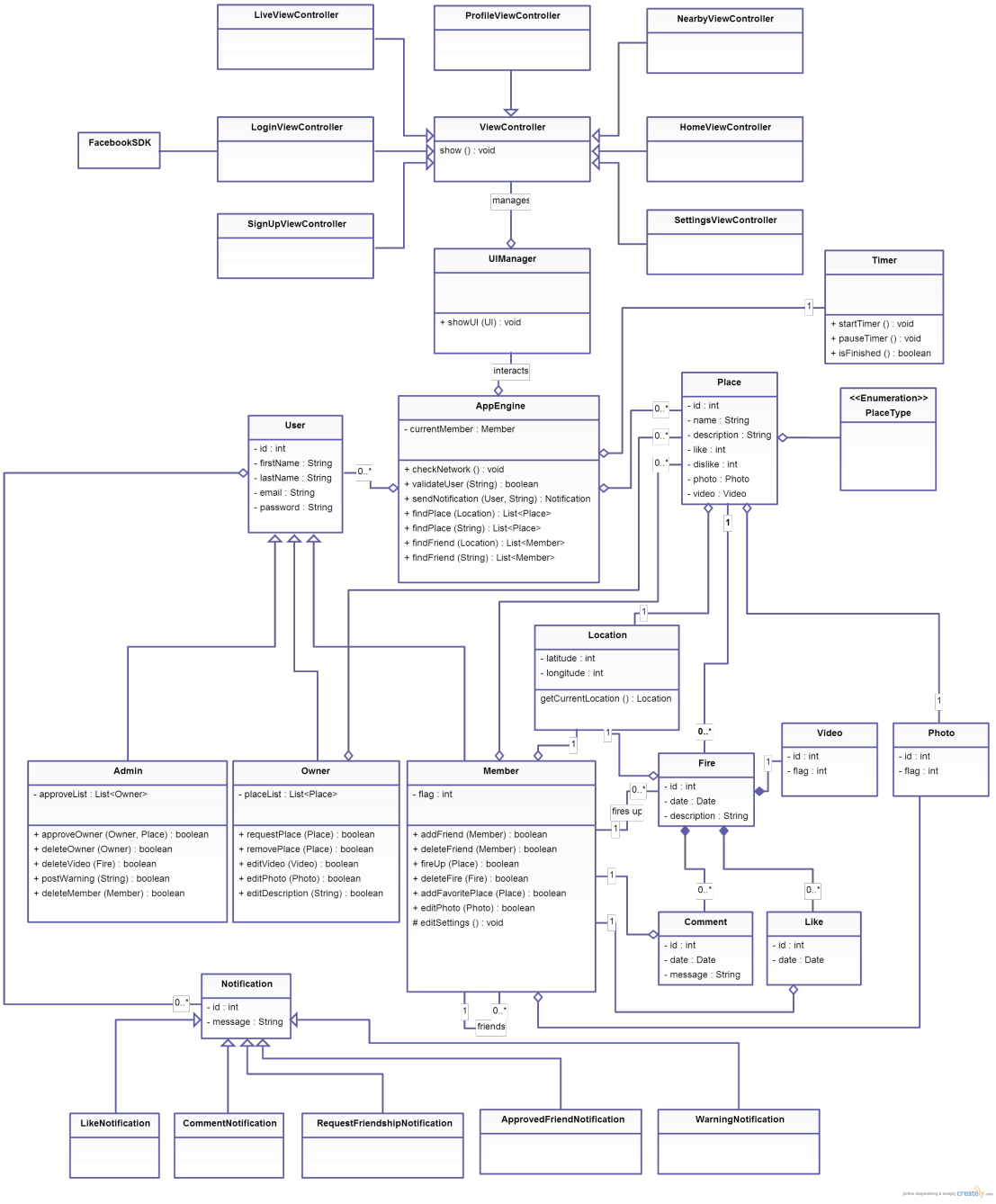

Figure – Class Diagram.

Class Name: AppEngine

Description of the Class: This class is the brain of the program and it manages the data of users and places.

Attributes

- currentMember: Member

Operations

+ checkNetwork(): void

+ validateUser(String, String): Boolean

+ sendNotification(User, String): Notification

+ findPlace(String): List<Place>

+ findPlace(Location): List<Place>

+ findFriend(String): List<Member>

+ findPlace(Location): List<Member>

Class Interactions

User, UIManager, Timer, Place

Class Name: Place

Description of the Class: This class holds the place information of id, name and description of place, popularity, related videos and photos and so on.

Attributes

- id: int
- name: String
- description: String
- like: int
- dislike: int
- photo: Photo
- video: Video

Class Interactions

AppEngine, Member, Owner, Location, Fire, Photo, PlaceType

Class Name: Location

Description of the Class: It holds the location of the user for check-in.

Attributes

- latitude: int
- longtitude: int

Operations

- getCurrentLocation(): Location

Class Interactions

Member, Fire, Place

Class Name: Fire

Description of the Class: It holds fire information such as id, date and description of fire.

Attributes

- id: int
- date: Date
- decription: String

Class Interactions

Member, Location, Place, Video, Comment, Like, Comment

Class Name: Comment

Description of the Class: It holds the content of the user comments.

Attributes

- id: int
- date: Date
- message: String

Class Interactions

Member, Fire

Class Name: Like

Description of the Class: It holds how much like user gets from other users for a fire.

Attributes

- id: int
- date: Date

Class Interactions

Member, Fire

Class Name: Video

Description of the Class: It holds the video id and also flag that checks whether video is undesirable or not.

Attributes

- id: int
- flag: int

Class Interactions

Fire

Class Name: Photo

Description of the Class: It holds the photo id and also flag that checks whether photo is undesirable or not.

Attributes

- id: int
- flag: int

Class Interactions

Member, Place

Class Name: User

Description of the Class: It holds the information about users such as id, email, first and last names.

Attributes

- id: int
- firstName: String
- lastName: String
- email: String
- password: String

Class Interactions

Notification, Admin, Owner, Member, AppEngine

Class Name: Admin

Description of the Class: It has some additional rights like approving/disapproving owners' requests, posting warnings, deleting members and so on.

Attributes

- approveList: List<Owner>

Operations

+ approveOwner(Owner, Place): Boolean

+ deleteOwner(Owner): Boolean

+ deleteVideo(Fire): Boolean

+ postWarning (String): Boolean

+ deleteMember (Member): Boolean

Class Interactions

User

Class Name: Owner

Description of the Class: It holds the information about owners of places and it gives some options to add, delete videos to places' pages.

Attributes

- placeList: List<Place>

Operations

+ requestPlace(Place): Boolean

+ removePlace(Place): Boolean

+ editVideo(Video): Boolean

+ editPhoto (Photo): Boolean

+ editDescription (String): Boolean

Class Interactions

User, Place

Class Name: Member

Description of the Class: It holds the information about members and their actions.

Attributes

- flag: int

Operations

+ addFriend(Member): Boolean

+ deleteFriend (Member): Boolean

+ fireUp(Place): Boolean

+ deleteFire (Fire): Boolean

+ addFavoritePlace (Place): Boolean

+ editPhoto (Photo): Boolean

editSettings (): void

Class Interactions

User, Place, Location

Class Name: Notification

Description of the Class: It holds the information about notifications.

Attributes

- id: int
- message: String

Operations

- getNotification (): Notification

Class Interactions

User, Fire, Place

Inheritance classes of Notification class

1. Class Name: LikeNotification

   Description of the Class: It holds the notifications about liked videos, photos and locations

2. Class Name: CommentNotification

   Description of the Class: It holds the notifications about made comments by other users.

3. Class Name: RequestFriendshipNotification

   Description of the Class: It holds the notifications about friendship request sended by other users.

4. Class Name: ApprovedFriendNotification

Description of the Class: It holds the notifications about the approved request by other users

5. Class Name: WarningNotification

   Description of the Class: It holds the notifications about unexpected errors.

4. Conclusion

As a result, our project is a social network application, which enables 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. In this report, we prepared documentation for class interface guidelines and mentioned about each class in the format of class interface.

5. 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

6. Schedule

- Project Specifications - Monday, Oct. 7 2013, **Done**
- Analysis Report - Monday, Nov. 4, 2013, **Done**
- High-Level Design Report - Friday, Dec. 31, 2013, **Done**
- Low-Level Design Report - Monday, Feb. 17, 2014, **This Report**
- Final Report - Thursday, Apr. 30, 2014, **Will be done**
- Presentations & Demonstrations - May 5 - 9, 2014, **Will be done**

7. References

[1] M. J. Dollinger. (2013). *Business Horizons* [Journal]

[2] . (2010). *General APA Guidelines* [Online]. Available: https://owl.english.purdue.edu/owl/resource/560/01/

[3] . (2013). *Unified Modeling Language (UML)* [Online]. Available: http://www.uml.org/

[4] J.R. van der Hoeven. (2007). *Object Design Document (ODD)* [Online]. Available: http://dioscuri.sourceforge.net/docs/ODD_Dioscuri_KBNA_v1_1_en.pdf

[5] . (2007). *ODD Object Design Trade-Offs* [Online]. Available: http://dev.campuseventfinder.com/index.php?title=ODD_Object_Design_Trade-Offs