30.10.2013

CENG 490

SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

Group 10 | The Cereal Killers

Members and Signatures

Member	Student ID	Date	Signature
Yaşar Barış ULU	1746429	30.10.2013	
Kemal Çağın GÜLŞEN	1745991	30.10.2013	
Mert ERGUN	1745942	30.10.2013	
Kerem GÖKHAN	1745983	30.10.2013	

Change History

Date	**Revision**	**Comment**
22.10.2013	0.9	Initial Version
30.10.2013	1.0	Detailed Version
29.12.2013	1.1	Updated Version

Preface

This document contains the system design information of Event-Based Social Network Project. This document is prepared according to the "IEEE Recommended Practice for Software Requirements Specifications - IEEE 830 – 1998" document.

This software requirements specifications document contains an overall description and all specific requirements of Event-Based Social Network Project.

The first section of this document is an introductory section which includes problem definition, purpose, scope, definitions and references of the project.

The second section of this document provides the overall description of the project.

The third section of this document contains specific requirements of Event-Based Social Network Project.

The fourth and the fifth section of this document provides data and behavior model of the project.

The final section explains the planning of the project and the team.

Table of Contents

Table of Figures

1. Introduction

This Software Requirement Specifications document is written by Kemal Çağın Gülşen, Mert Ergun, Yaşar Barış Ulu and Kerem Gökhan for the Event-Based Social Network Project. Firstly, purpose and scope of the project will be described. Secondly overall description of the project will be explained. After these sections, other parts of the document will state specific requirements and models with their descriptions. Finally, the document will end with a conclusion part.

1.1. Problem Definition

Mobile phones, tablet PCs and all other kinds of smart devices are spreading to people's life everyday more and more. With more usage of these devices, people tend to use social networks more. Widely used social networks focus various subjects. Mainly they focus people and people related sharing such as sharing photos, videos, places and texts. None of widely used social networks focus events. There are some social network examples that contain event information, but that concept is out of their focus. People look for events and organizations nearby, but no application gives this information to them. In Event-Based Social Network Project, a different type of networking system will be implemented and hopefully used widely.

1.2. Purpose

Purpose of this Software Requirements Specifications document is to clearly state every requirement, all functions and all features of the project. Furthermore this document will be useful for both developers and customers. It will give more information about the product and consequently people will be able to agree on functions and features of the system.

1.3. Scope

The software that is going to be implemented is a cross-platform mobile application. Users will be able to create, share, promote and join events. Furthermore users will be able to see events nearby, trending events and promoted events. Aim of this project is to provide an application that helps people with all kinds of events. The end product will serve users with various types of events such as academic, entertainment, commercial, technological, leisure activities. This project will be implemented by four people between October 2013 and June 2014.

1.4. Definitions, Acronyms and Abbreviations

PC	Personal Computer
Cross-Platform	Capability of software to run identically on different platforms
WP	Windows Phone
iOS	iPhone Operating System
MB	Megabyte
GB	Gigabyte
RAM	Random Access Memory (Primary Memory)
GPS	Global Positioning System

1.5. References

- Star UML Guide, retrieved from http://staruml.sourceforge.net/docs/user-guide(en)/toc.html
- IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

1.6. Overview

Following sections will give information of:

- Overall Description
- Specific requirements of the project
- Data model and Behavioral model
- Conclusion

2. Overall Description

In this section, the general aspects that influence the product and its requirements will be described. However, they will be defined roughly. The purpose of this section is providing a background information to make the specific requirements, which are going to be given in section 3, easier to understand.

2.1. Product Perspective

Our product is a social network that mainly focuses on events and their locations. It is a stand-alone product that is not depend on a larger system or included in one.

In this subsection of the document, some constraints related to system, hardware, user, software interfaces and some other systems will be given.

2.1.1. System Interfaces

It is important to keep target market wider as much as possible that's why our software will be provided with multiple operation system support, especially smart phone operating systems namely iOS, Android and Windows Phone. It can be used with any of the personal computers as long as they are connected to internet with an up to date browser. Main server will be deployed on a virtual machine which is running on Amazon Web Services. This server not only will contain the web service for all views but also will provide hosting for web page of our application. The virtual machine's operating system is preferably Ubuntu 12.04 LTS since it is more useable with a command prompt.

2.1.2. User Interfaces

There are two main kind of user interfaces in our project. These are namely, web page user interface and mobile application user interface. Mobile applications for each platform, Windows Phone, iOS and Android, will be developed by using Titanium SDK. The Titanium SDK provides community developers with an unmatched ability to create quality native, mobile web or rich hybrid applications to all platforms from a single code base, using JavaScript. In this way the amount of workload for developing mobile applications will be reduced. For web page user interface view, Mojarra 2.2, which is also known as Java Server Faces, and

Primefaces integration will be used. We have decided to use Primefaces because it has lots of built-in basic functionalities for a website. In addition to that, server side and web page view may share some of the code that are written for object classes because both are written in Java.

2.1.3. Hardware Interfaces

Since our application will mainly focuses and emphasizes on maps, smart phones that have GPS capabilities are required. Although, there are other ways to determine the users location, GPS would be the most accurate way to achieve better results. To achieve fully functional and smooth experience on our product at least 1 GB of RAM is required and of course, the device should be connected to the internet. For development phase, the server computer will be a micro instance which is provided without a payment by Amazon. It can fulfill the testing requirements. This instance has only one core of Intel® Xeon® Processor E5-2650 with 590 Megabytes of RAM. When our product is ready, the micro instance will be converted into a better one. Amazon gives the capability of doing that.

2.1.4. Memory Constraints

At least 100 MBs of empty space required for smart phones. There is no such requirement for personal computers as long as there exits enough space for operating system to maintain its functionalities. In addition, smart phones should have at least 1 GB of primary memory and PCs should have at least 2 GBs of primary memory to fulfill our application requirements.

2.1.5. Operations

There is no user initiated task on our application. They only need to create an event and enter detail information or join an already created one. All application data is stored on the server and no backup or recovery operation will be offered to end-user.

2.2. Product Functions

In this section, a summary of the major functions that the software will perform will be mentioned. Our software, which is event based social network, provides two main functionalities to the user namely system functionalities and user functionalities. These functionalities will be explained in later parts of this document in detail. Now, the name of sub titles of main functionalities is given below.

2.2.1. User Functionalities

Download and install application, register manually, register via Facebook, register via Twitter, register via Google+, login, retrieve password, logout, search user, send friend request, show friend requests, accept friend request, reject friend request, remove friend, show user profile, change profile information, change profile picture, create event, cancel event, show event profile, edit event profile, join event, share event, search event, comment on event, rate event, promote event.

2.2.2. System Functionalities

View trending events, view events on map, view route to event, earn badges, view badges, gain experience points.

In Figure 1 below, the use case diagram gives a better and general understanding of the system. This includes just names of sub titles of main functionalities of our software. All of these will be explained later in detail.

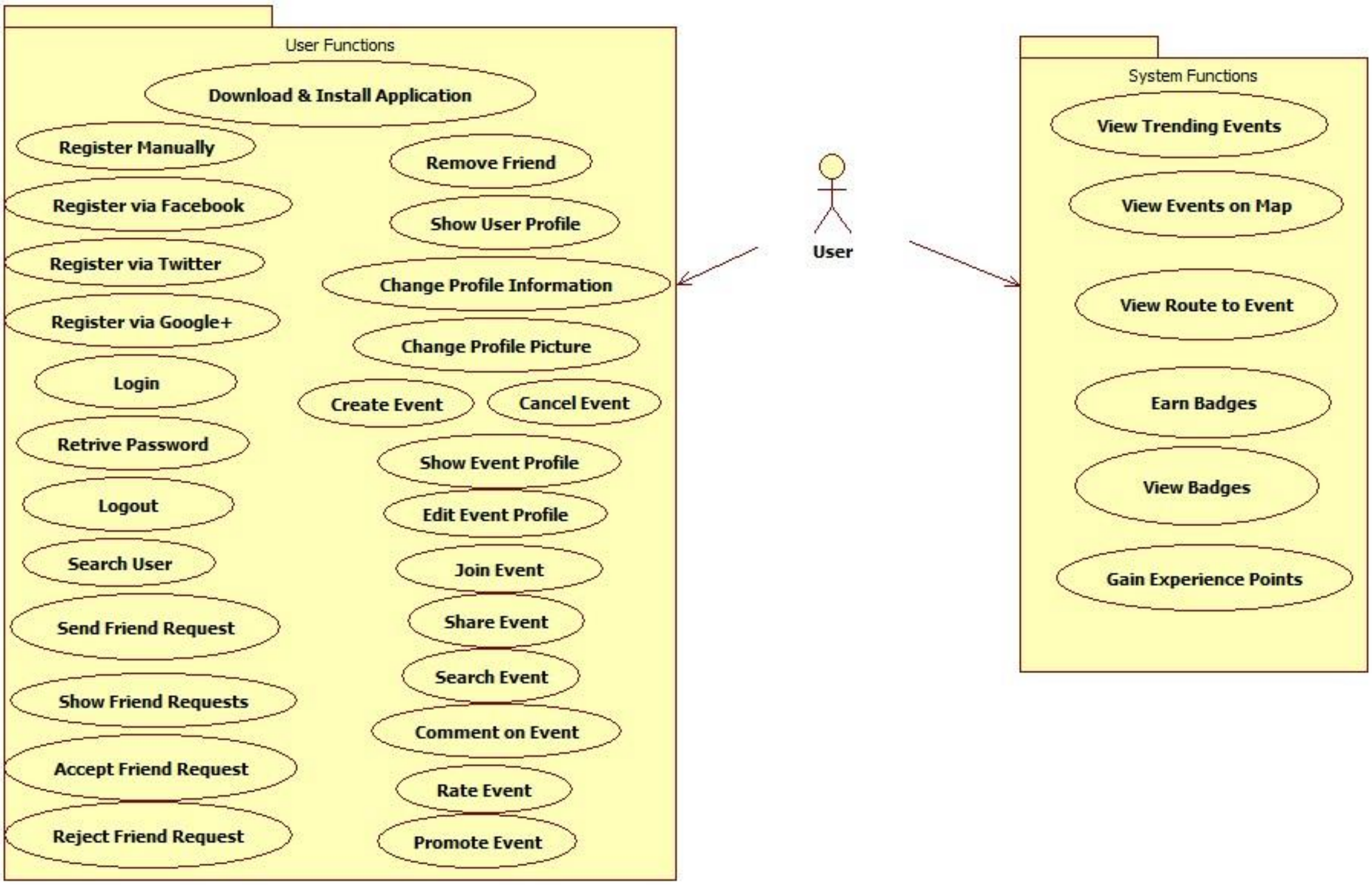


Figure 1: General Use Case Diagram

2.3. User Characteristics

Our software is aimed for use by two kinds of user types. These are regular users which always use the software in daily life and organizers which use the software for creating special events and promoting these events.

- **Regular Users:** This kind of users is the majority of users. They are capable of performing all functionalities mentioned above. Every user can use all of characteristics of software easily. A little experience of using internet and a smart phone application may be helpful for this kind of users.

- **Organizers:** This kind of users is the minority of users. They are capable of performing all functionalities like regular users. In addition to that, organizers can be capable of promoting events in public environment. Same experience for regular users is valid for organizers for usage easily.

2.4. Constraints

It is certain that we will obey the international laws and local laws. Our product has no ethical or legal issue related to patent and copyrights.

There are various mobile operating systems running on many different smart phone brands. GPS Navigation System may work differently on each of them and they may provide different navigation features so, mobile applications of our project are constrained by GPS Navigation System.

Our product is also constrained by the capacity of the database. No events can be created or no new users can join our portal if the capacity runs out. In addition, if it takes too many query requests, the responses can be delayed because it processes them in a sequential way. Since the same database is used for all mobile applications and web site, the database may constraints all parts of the projects during excessive uses.

2.5. Assumptions and Dependencies

It is assumed that all GPS Systems are accurate and works the same way. While implementing the location services the difference between GPS Navigation Systems will not be considered. It is also assumed that our mobile applications will work on devices that fulfill our hardware requirements and performance criteria.

3. Specific Requirements

3.1. Interface Requirements

3.1.1. Login Screen

Login screen will be shown to unregistered users. There will be four button to communicate with the user as shown in Figure 2. “Manual registration” button directs the user to registration screen. If user selects “Facebook”, “Twitter” and “Google+” buttons, registration data will be gain from related social network.

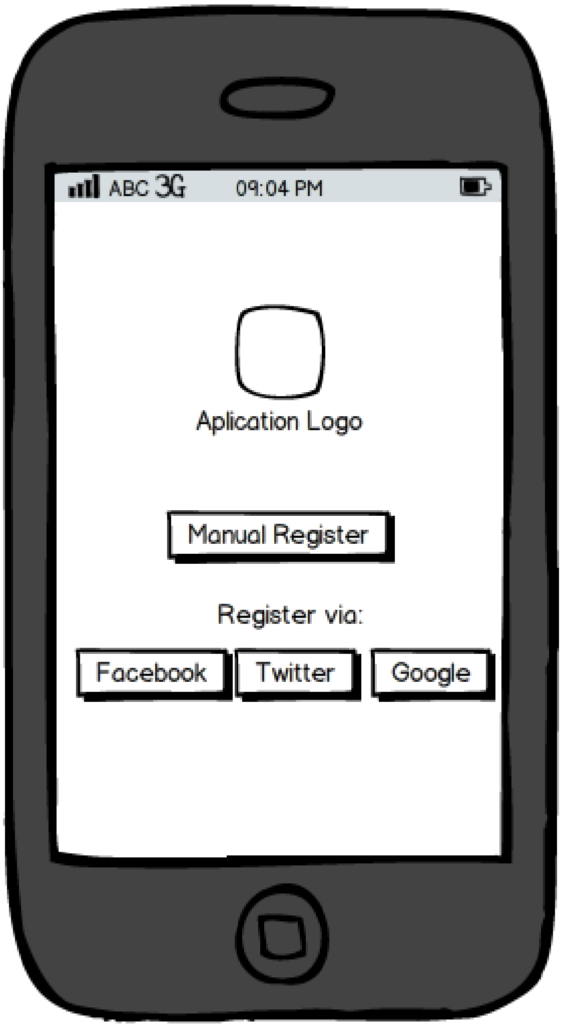
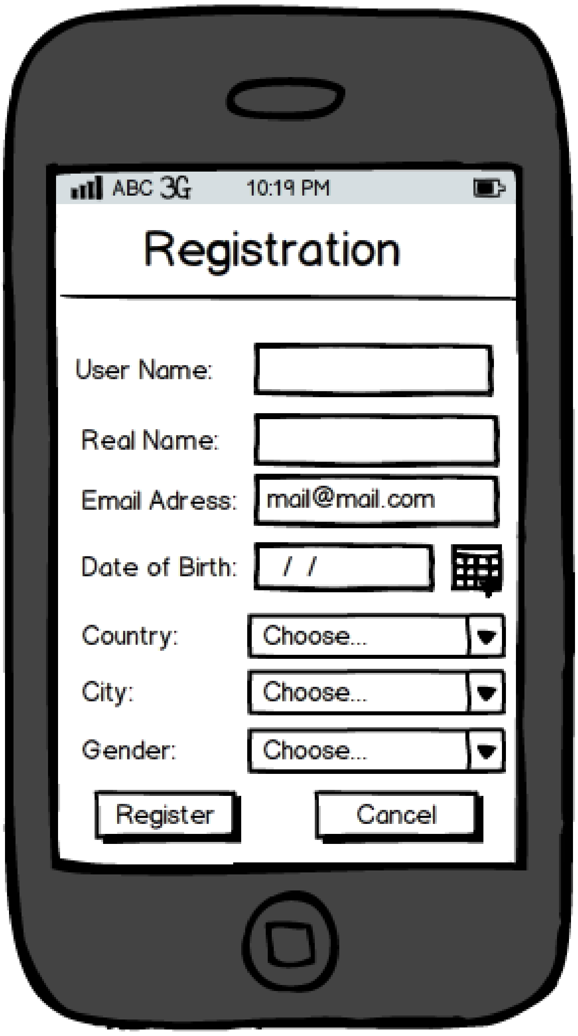


Figure 2: Main Screen

3.1.2. Registration Screen

If the user choose the manual registration, user information will be given in registration screen (Figure 3). User should fill the related fields to give his/her user name, real name, date of birth, gender, country and city.



Registration

User Name:

Real Name:

Email Adress: mail@mail.com

Date of Birth: / /

Country: Choose...

City: Choose...

Gender: Choose...

Figure 3: Registration Screen

3.1.3. Main Screen

The main interface consists of three main tabs, these are home/main screen, user profile and search screen.

In main screen there will be a map, trending events and promoted events as show in Figure 4. There will be icons on the map which show the events on their location. Once the user clicked on an icon, event info will be shown briefly. In addition there are sponsored and trending events lists on the main screen. User can see the events their friends will be attended on the trending events list. There are also featured (sponsored) events list. Once the user click on any event in those lists, it will direct to the event’s profile screen.

There is also a create event button which allows the user to create his/her own events.

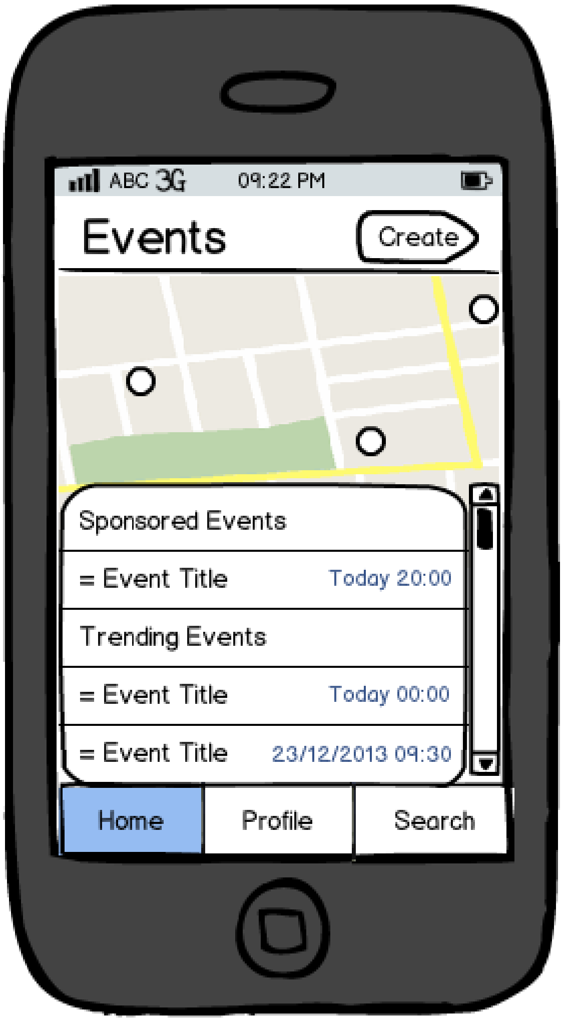


Figure 4: Main Screen

3.1.4. Event Screen

Event screen (Figure 5) shows the information about an event; name of the event, event picture, location, date/time, description of the event, the user's friends attending the event and comments about the event. There are also two buttons, one will be used for adding the event to the users future events list, the other will be used for checking-in to the event if the user's location is close to the event's location after the event is started.

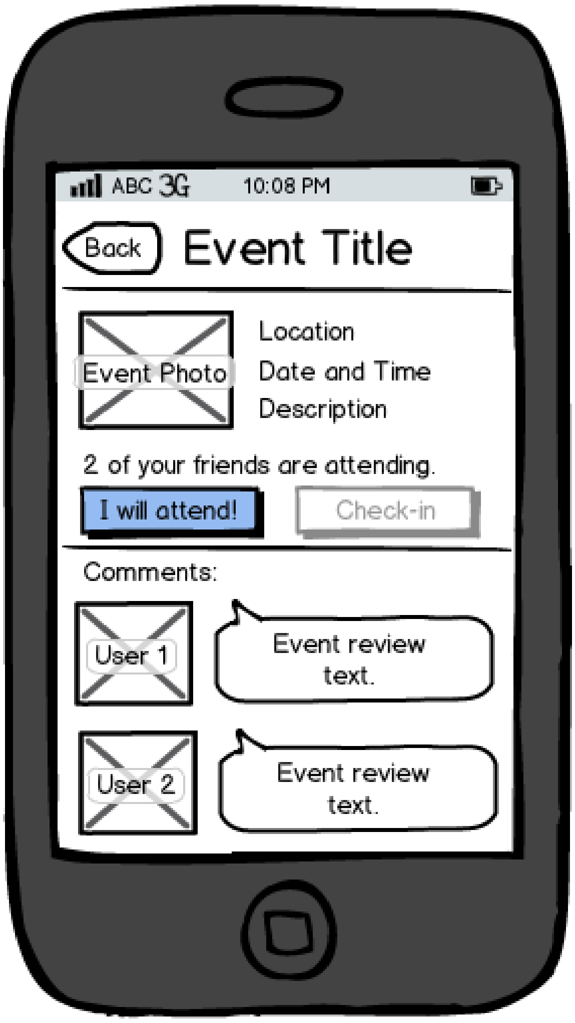
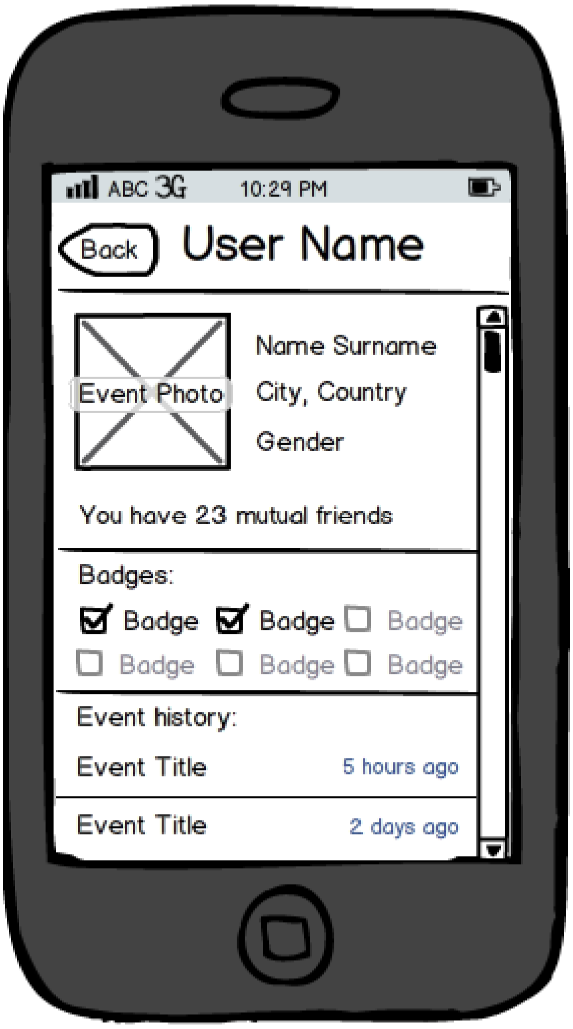


Figure 5: Event Screen

3.1.5. User Profile Screen

User profile screen (Figure 6) shows the information about a user; name of the user, city, country, gender, number of mutual friends, badges, event history and future events.



User Name

Name Surname
City, Country
Gender

You have 23 mutual friends

Badges:

☑ Badge ☑ Badge □ Badge
□ Badge □ Badge □ Badge

Event history:

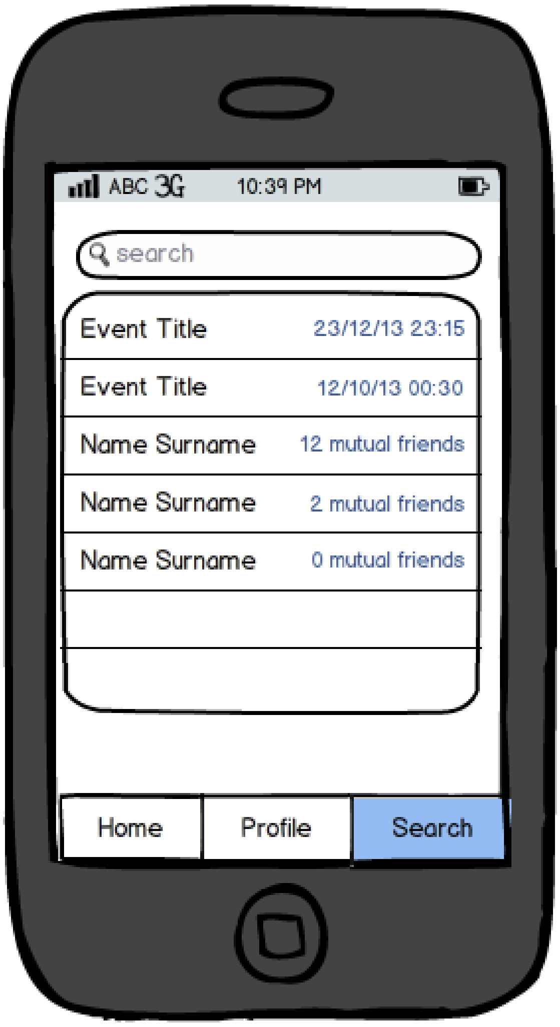
Event Title 5 hours ago

Event Title 2 days ago

Figure 6: User Profile Screen

3.1.6. Search Screen

Search screen will be used for searching any events or users. Search results will appear on the screen (Figure 7). If user clicks on any of the results, he/she will be directed to related user or event profile screen.



Event Title 23/12/13 23:15

Event Title 12/10/13 00:30

Name Surname 12 mutual friends

Name Surname 2 mutual friends

Name Surname 0 mutual friends

Figure 7: Search Screen

3.1.7. Event Creation Screen

In event creation screen (Figure 8), information about the event which will be created is given by the user. These information are event name, address, privacy (public, private), date and event picture.

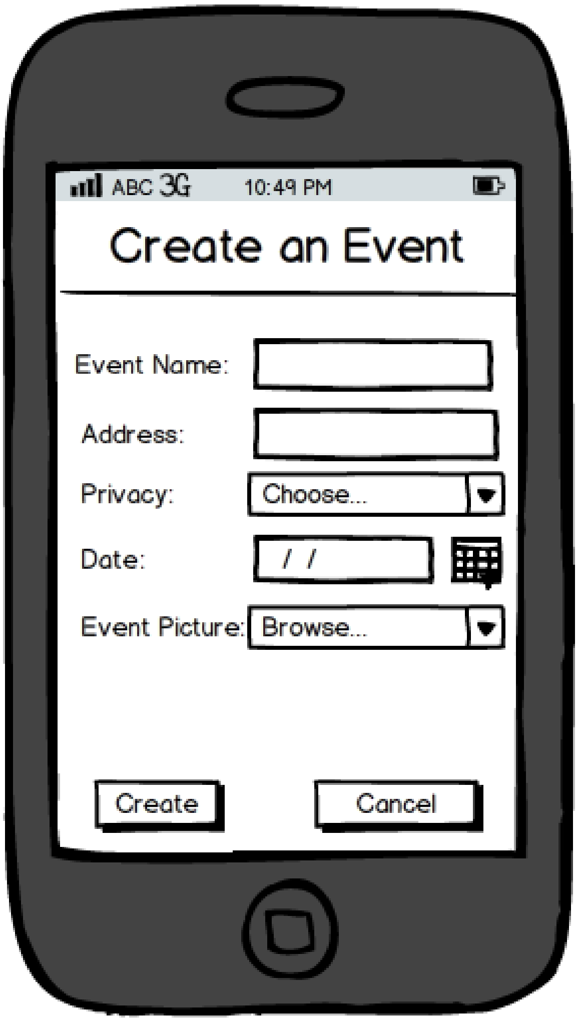


Figure 8: Event Creation Screen

3.2. Functional Requirements

This part explains all the functions of the system in details with two main parts. These functions are explained in use-case diagram part and description part.

3.2.1. User Functions

3.2.1.1. Download & Install Application

3.2.1.1.1. Use-Case Diagram

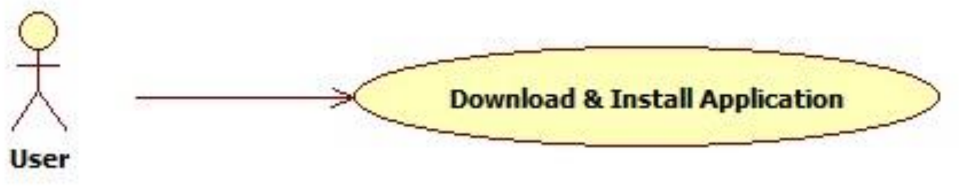


Figure 9: Download & Install Application

3.2.1.1.2. Description

Our application will need to be downloaded and installed via internet in order to be used by users. The application will be totally free to download for every user. It will be downloaded and installed in individual smartphones in every platform.

3.2.1.2. Register Manually

3.2.1.2.1. Use-Case Diagram

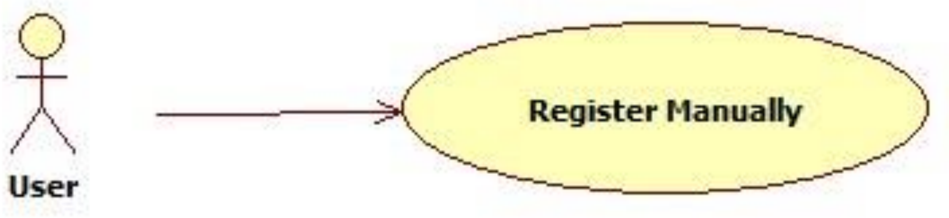


Figure 10: Register Manually

3.2.1.2.2. Description

User will register to system in order to benefits opportunities of the application. First type of registration, user will be able to register by manually. Name, surname, age, e-mail and password will be required in order to complete registration. User should fill these necessary fields on own.

3.2.1.3. Register via Facebook

3.2.1.3.1. Use-Case Diagram

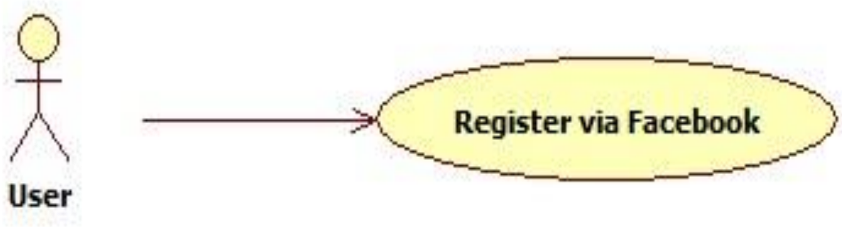


Figure 11: Register via Facebook

3.2.1.3.2. Description

User will be able to register himself/herself for registration via Facebook. There will be “Facebook” button on main screen of our application. They will be able to press and register automatically to our application through Facebook.

3.2.1.4. Register via Twitter

3.2.1.4.1. Use-Case Diagram

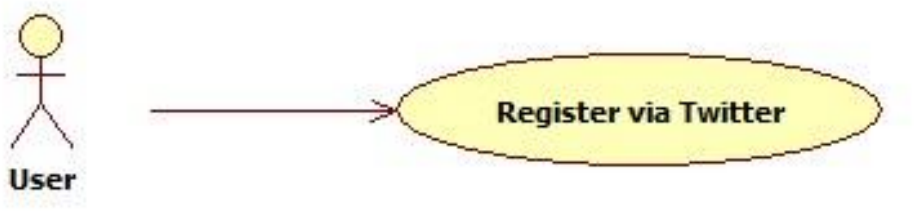


Figure 12: Register via Twitter

3.2.1.4.2. Description

User will be able to register himself/herself for registration via Twitter. There will be "Twitter" button on main screen of our application. They will be able to press and register automatically to our application through Twitter.

3.2.1.5. Register via Google+

3.2.1.5.1. Use-Case Diagram

User
Register via Google+

Figure 13: Register via Google+

3.2.1.5.2. Description

User will be able to register himself/herself for registration via Google+. There will be "Google+" button on main screen of our application. They will be able to press and register automatically to our application through Google+.

3.2.1.6. Login

3.2.1.6.1. Use-Case Diagram

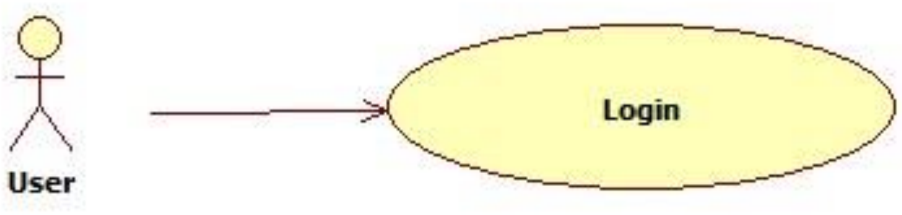


Figure 14: Login

3.2.1.6.2. Description

User will be capable of login our application after completing registration successfully. The login information will be stored on the phone and in the future user should be logged in automatically.

3.2.1.7. Retrieve Password

3.2.1.7.1. Use-Case Diagram

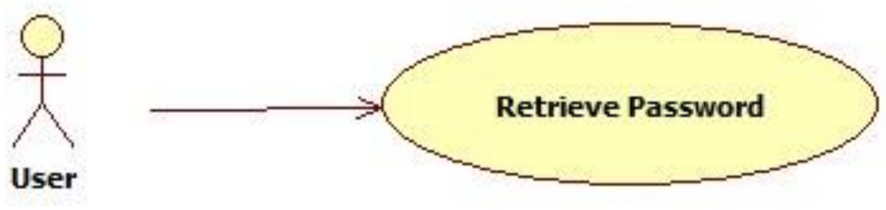


Figure 15: Retrieve Password

3.2.1.7.2. Description

Users will be able to request passwords when they will forget their own passwords. They will be able to retrieve passwords by their registered e-mails. After receiving e-mail from application, they will be capable of login our application with the password through e-mail.

3.2.1.8. Logout

3.2.1.8.1. Use-Case Diagram

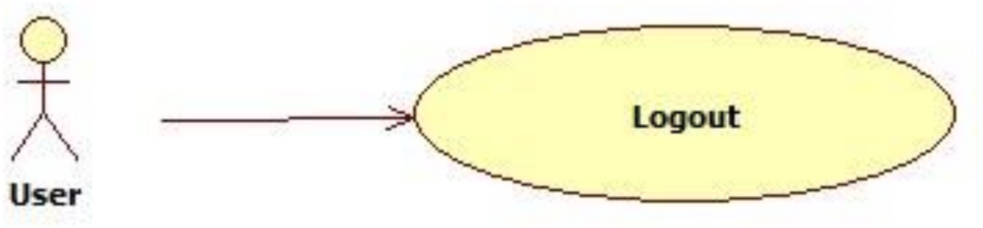


Figure 16: Logout

3.2.1.8.2. Description

Users will be able to logout from application safely after they will login application successfully. After logout operation users will be on the main screen of our application and again they will be able to login our system.

3.2.1.9. Search User

3.2.1.9.1. Use-Case Diagram

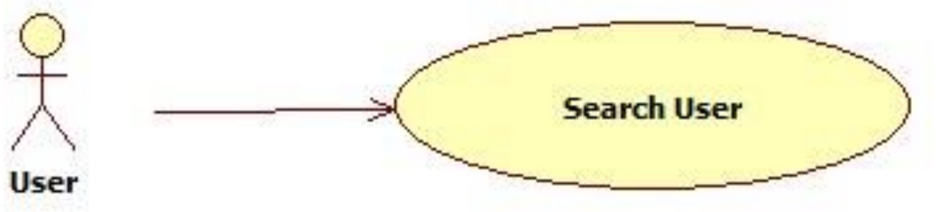


Figure 17: Search User

3.2.1.9.2. Description

Users will be able to search among users registered in database of our application when they will be already logged in application. Name, username and e-mail will be used to start search operation for users. According to desired criterion users will reach user list from search operation.

3.2.1.10. Send Friend Request

3.2.1.10.1. Use-Case Diagram

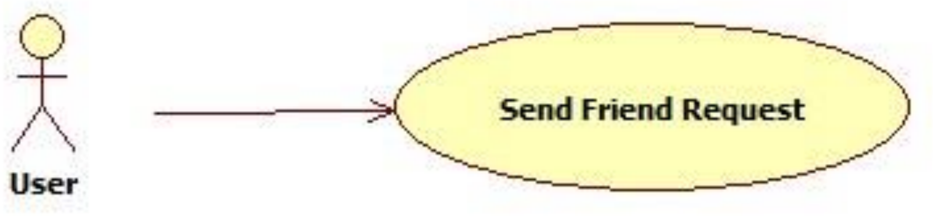


Figure 18: Send Friend Request

3.2.1.10.2. Description

Users will be able to send friend request to other users inside our system. Important thing is that this operation is just sending a request. It doesn't mean that after sending friend request there will be a friendship between those users.

3.2.1.11. Show Friend Requests

3.2.1.11.1. Use-Case Diagram

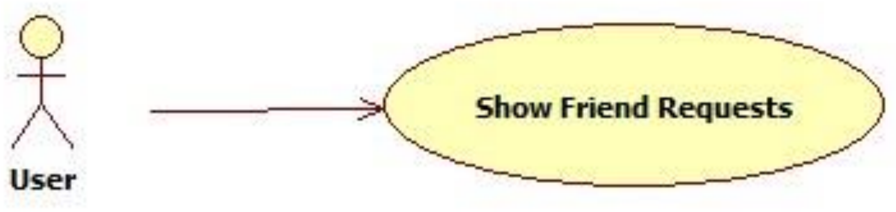


Figure 19: Show Friend Requests

3.2.1.11.2. Description

Users will be able to show friend requests which will be already sent to them in a list. On the other hand, if there will not exist any request then users will get empty list.

3.2.1.12. Accept Friend Request

3.2.1.12.1. Use-Case Diagram

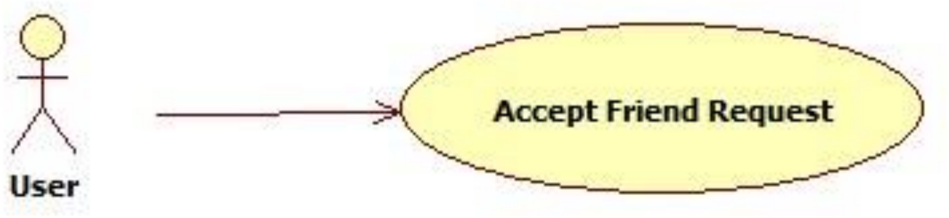


Figure 20: Accept Friend Request

3.2.1.12.2. Description

Users will be to accept friend requests which will be already sent to them.

3.2.1.13. Reject Friend Request

3.2.1.13.1. Use-Case Diagram

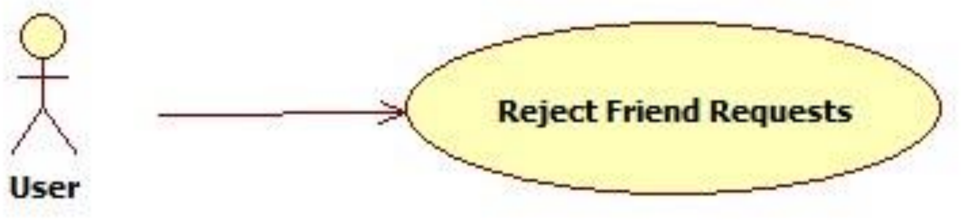


Figure 21: Reject Friend Requests

3.2.1.13.2. Description

Users will be to accept friend requests which will be already sent to them.

3.2.1.14. Remove Friend

3.2.1.14.1. Use-Case diagram

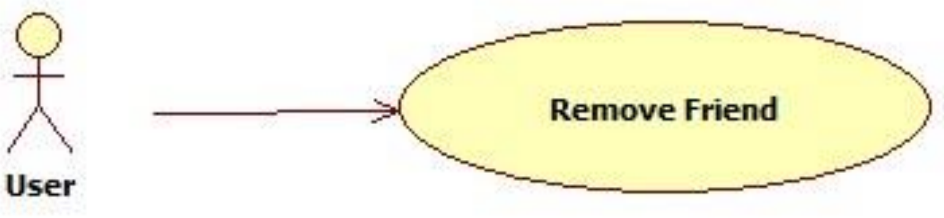


Figure 22: Remove Friend

3.2.1.14.2. Description

Users will able to remove any friend/friends from their friend list.

3.2.1.15. Show User Profile

3.2.1.15.1. Use-Case Diagram

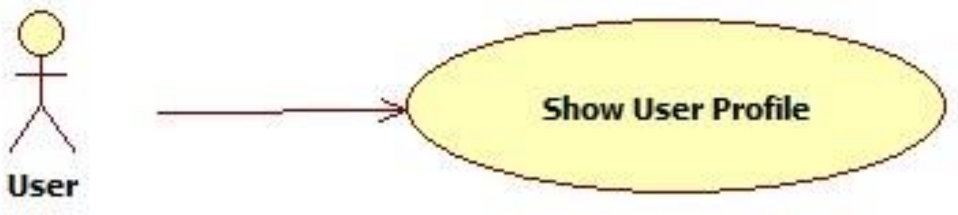


Figure 23: Show User Profile

3.2.1.15.2. Description

Users will be able to show a user profile in the application. They will be capable of view own profiles, friends' profiles and stranger profiles.

3.2.1.16. Change Profile Information

3.2.1.16.1. Use-Case Diagram

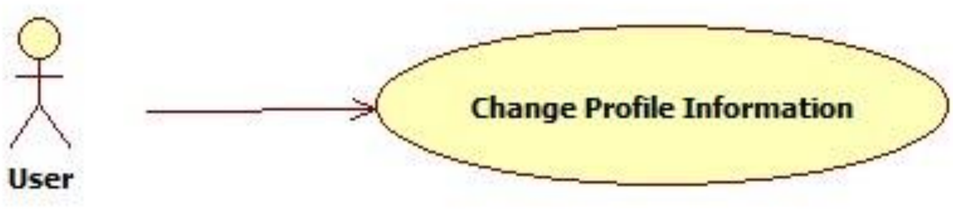


Figure 24: Change Profile Information

3.2.1.16.2. Description

Users will be able to change profile information from their profile. They will be capable of changing name, surname, username, password and location.

3.2.1.17. Change Profile Picture

3.2.1.17.1. Use-Case Diagram

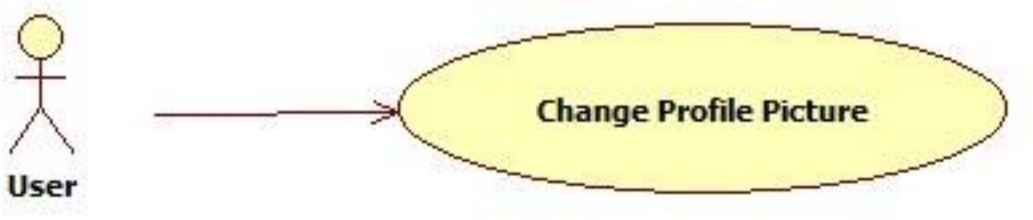


Figure 25: Change Profile Picture

3.2.1.17.2. Description

Users will be able to change their old profile pictures with new ones inside the profile page.

3.2.1.18. Create Event

3.2.1.18.1. Use-Case Diagram

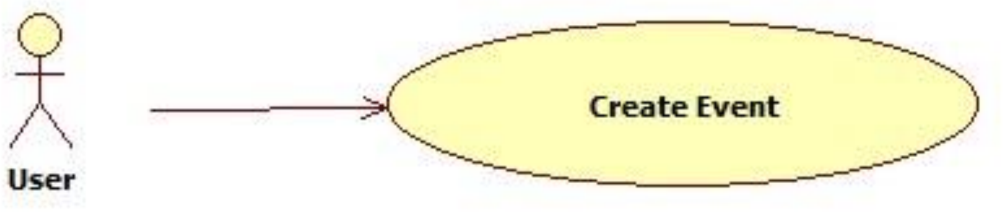


Figure 26: Create Event

3.2.1.18.2. Description

Users will be able to create their own private or public events. While creating an event, users will take an opportunity on setting properties of events which are time, location, category, privacy, picture and description.

3.2.1.19. Cancel Event

3.2.1.19.1. Use-Case Diagram

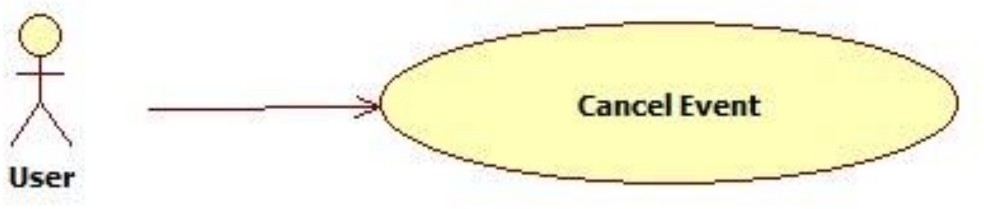


Figure 27: Cancel Event

3.2.1.19.2. Description

Users will be able to cancel events which will have been created by them before.

3.2.1.20. Show Event Profile

3.2.1.20.1. Use-Case Diagram

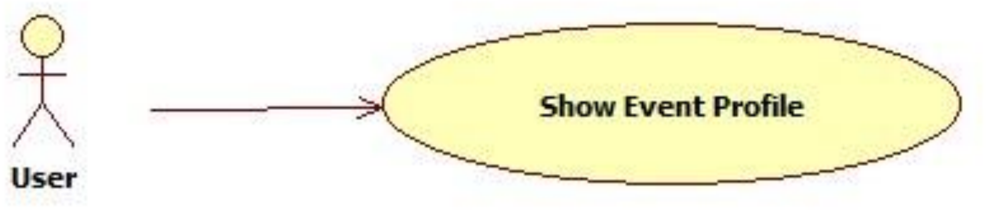


Figure 28: Show Event Profile

3.2.1.20.2. Description

Users will be able to show event profile which will belong to them or belong to another person provided that privacy is public. Event profiles will consist of some properties like time, location, category, privacy, picture, description and participants. Besides, users will be able to show location of an event on the map.

3.2.1.21. Edit Event Profile

3.2.1.21.1. Use-Case Diagram

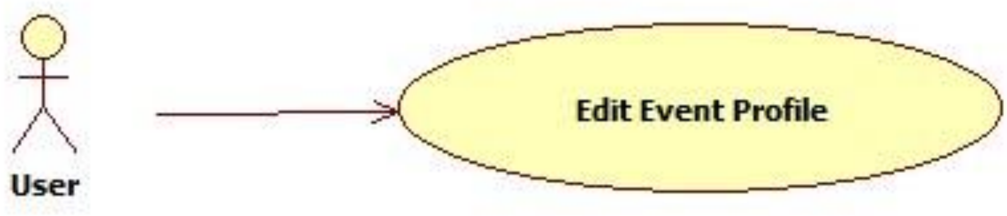


Figure 29: Edit Event Profile

3.2.1.21.2. Description

Users will be able to edit their event profiles. They will be capable of editing event properties which are time, location, category, privacy, picture and description.

3.2.1.22. Join Event

3.2.1.22.1. Use-Case Diagram

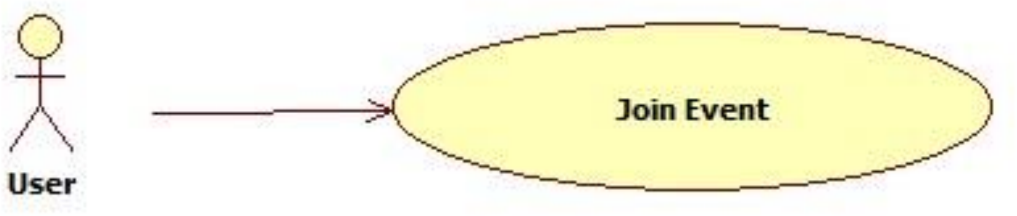


Figure 30: Join Event

3.2.1.22.2. Description

Users will be able to join an event which will be already created by them or by others provided that privacy is public.

3.2.1.23. Check-in

3.2.1.23.1. Use-Case Diagram

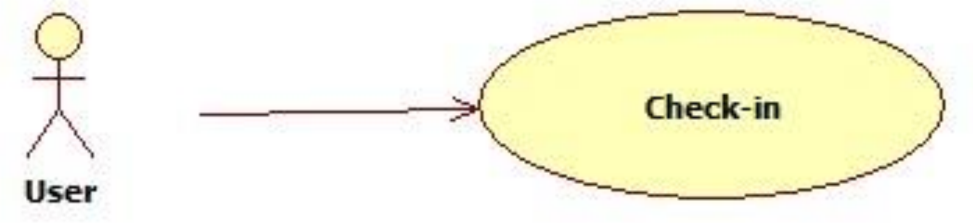


Figure 31: Check-in

3.2.1.23.2. Description

Users will be able to check-in when they will be in the event. Their locations will be checked according to event locations.

3.2.1.24. Share Event

3.2.1.24.1. Use-Case Diagram

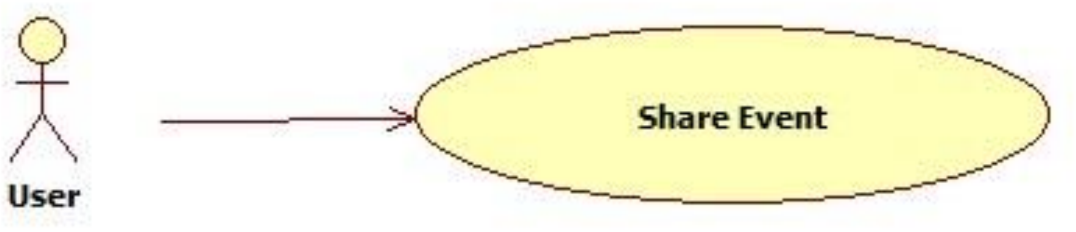


Figure 32: Share Event

3.2.1.24.2. Description

Users will be able to share their events or any other events.

3.2.1.25. Search Event

3.2.1.25.1. Use-Case Diagram

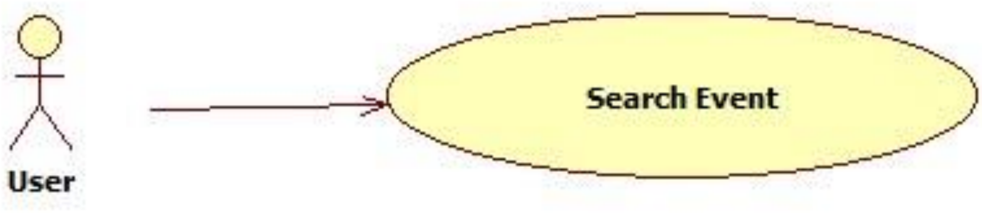


Figure 33: Search Event

3.2.1.25.2. Description

Users will be able to search events with some parameters like location, time, category or popularity.

3.2.1.26. Comment on Event

3.2.1.26.1. Use-Case Diagram

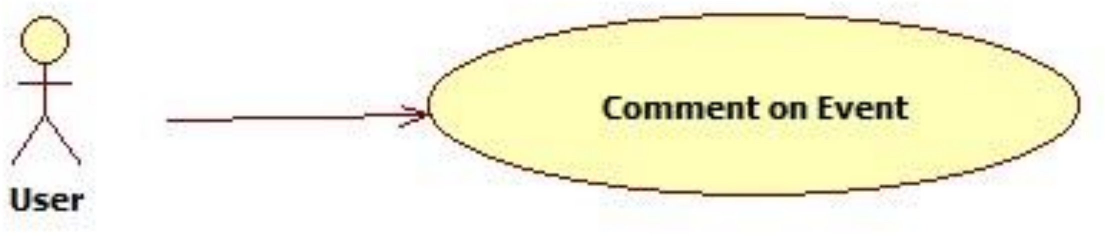


Figure 34: Comment on Event

3.2.1.26.2. Description

Users will able to comment on an event which belongs to them or not. Also, they will be capable of commenting on any event which they will have joined or not.

3.2.1.27. Rate Event

3.2.1.27.1. Use Case Diagram

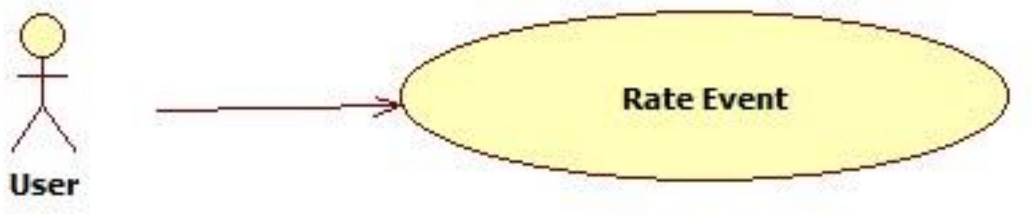


Figure 35: Rate Event

3.2.1.27.2. Description

Users will be able to rate an event which they will have already attended.

3.2.1.28. Promote Event

3.2.1.28.1. Use-Case Diagram

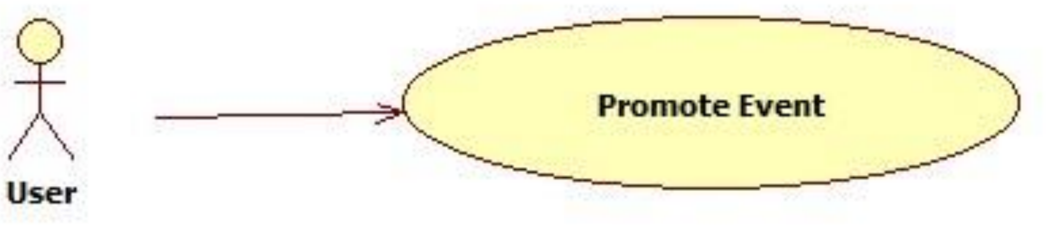


Figure 36: Promote Event

3.2.1.28.2. Description

Users will be able to promote an event for any organization.

3.2.2. System Functions

3.2.2.1. View Trending Events

3.2.2.1.1. Use-Case Diagram

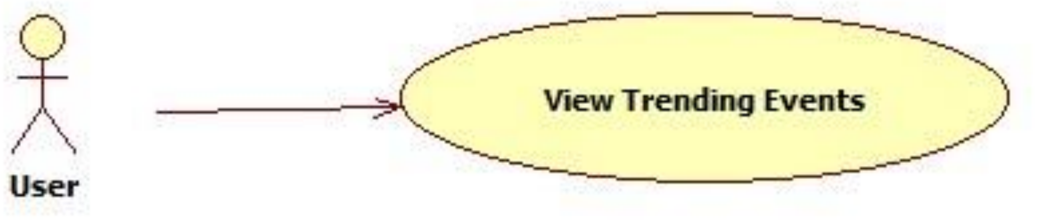


Figure 37: View Trending Events

3.2.2.1.2. Description

Users will be able to view trending events on home page of our application. Top trending events will be listed according to their popularity.

3.2.2.2. View Events on Map

3.2.2.2.1. Use-Case Diagram

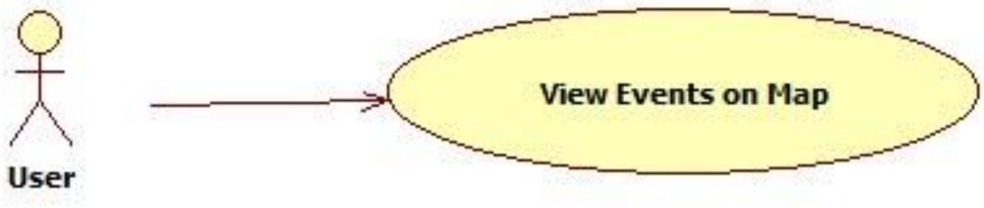


Figure 38: View Events on Map

3.2.2.2.2. Description

Users will be able to view events on map home page of our application through Google Maps via GPS or Internet.

3.2.2.3. View Route to Event

3.2.2.3.1. Use-Case Diagram

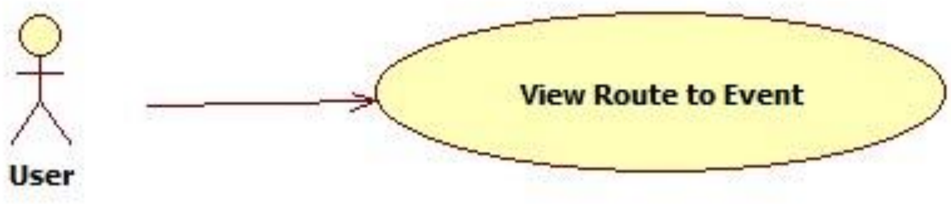


Figure 39: View Route to Event

3.2.2.3.2. Description

Users will be able to view the route to an event through Google Maps via GPS or Internet.

3.2.2.4. Earn Badges

3.2.2.4.1. Use-Case Diagram

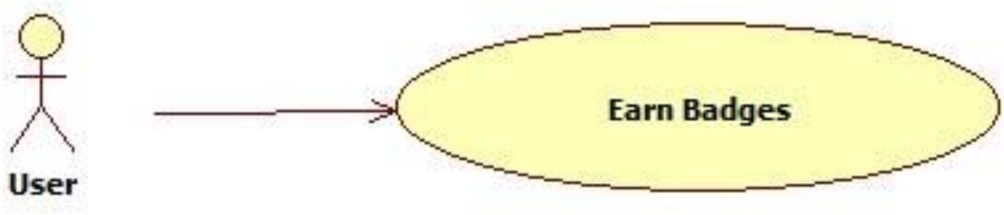


Figure 40: Earn Badges

3.2.2.4.2. Description

Users will be able to earn badges from our application according to their activities in system.

3.2.2.5. View Badges

3.2.2.5.1. Use-Case Diagram

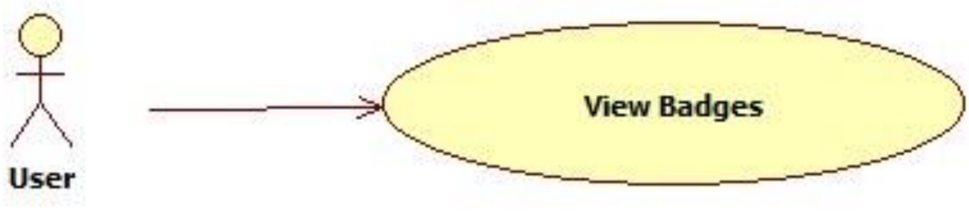


Figure 41: View Badges

3.2.2.5.2. Description

Users will be able to view badges which will have earned by them so far. Besides, they will able to look for badges of friends.

3.2.2.6. Gain Experience Points

3.2.2.6.1. Use-Case Diagram

User
Gain Experience Points

Figure 42: Gain Experience Points

3.2.2.6.2. Description

Users will be able to gain experience points with respect to their activity in events such as creating events, attending events, promoting events.

3.3. Non-functional Requirements

3.3.1. Performance Requirements

It is important to present users a smooth application interface. All the views, pages and applications need to be operated immediately when users want to. It should not take more than six seconds. Users shall be able to get a valid response to their searches in eight seconds. 95% of transactions should be processed in less than eight seconds. These requirements are need to be fulfilled as long as the internet connection of users does not constraint the system performance.

The application must support 500 online users. It should work at full performance within this limit. However, scaling up the project to reach more users may be considered according to the demand of the product. In other words, the product should be scalable.

3.3.2. Logical Database Requirements

Database is the most essential part of our project. All the user and event data such as friendship relationships, comments, friend requests, event participant lists and other crucial data is stored in the database. Only the privileged users are able to execute insertion and deletion queries that means username and password are required.

Main data entities are listed below:

- User
- Event
- Location
- Level
- Badge

3.3.3. Design Constraints

The product will be supported by three major mobile operating systems. Three different applications created by HTML5 technology. Therefore, this technology constraints the mobile applications on each operating system differently. Server-side code will be written in JAVA. For the diagrams used in reports, UML standards are considered and for all reports IEEE standards are obeyed.

4. Data Model and Description

This section provides information of classes that will contain data and their relationships. However, these classes, their attribute will be updated on later stages.

4.1. Data Description

This part explains data classes and their functions which are related with data variables.

4.1.1. Event Class

The Event class will handle event related data operations of the system and contain all event information with location, owners and attendees information. This class will be updated at the later stages of the project.

Class Diagram:

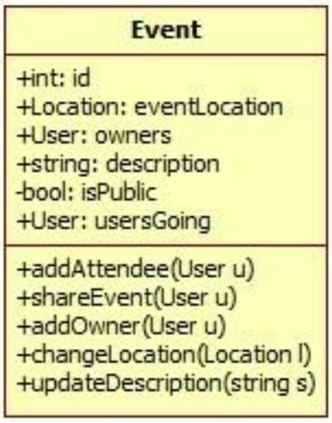


Figure 43: Event Class

4.1.2. User Class

The User class will handle user related data operations of the system and contain all user information with badge and level information. This class will be updated at the later stages of the project.

Class Diagram:

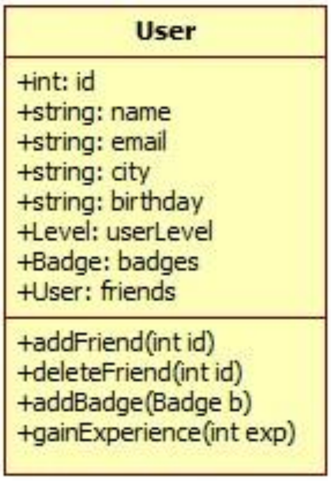


Figure 44: User Class

4.1.3. Location Class

The Location class will handle location related data operations of the system and contain all location information. This class will be updated at the later stages of the project.

Class Diagram:

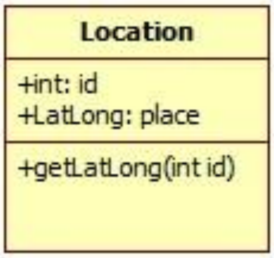


Figure 45: Location Class

4.1.4. Level Class

The Level class will handle level related data operations of the system and contain all level information. This class will be updated at the later stages of the project.

Class Diagram:

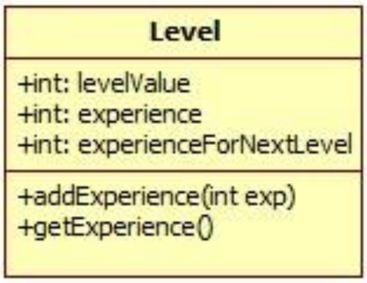


Figure 46: Level Class

4.1.5. Badge Class

The Badge class will handle level related data operations of the system and contain all badge information. This class will be updated at the later stages of the project.

Class Diagram:

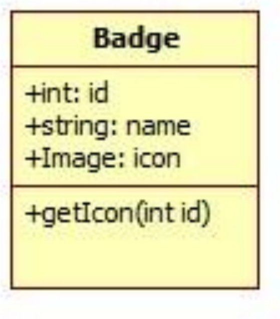


Figure 47: Badge Class

4.2. Data Relations

The relationship between classes:

- The Event class has elements from User and Location classes
- The User class has elements from Badge and Level classes

Relationships are shown at Class Diagram below:

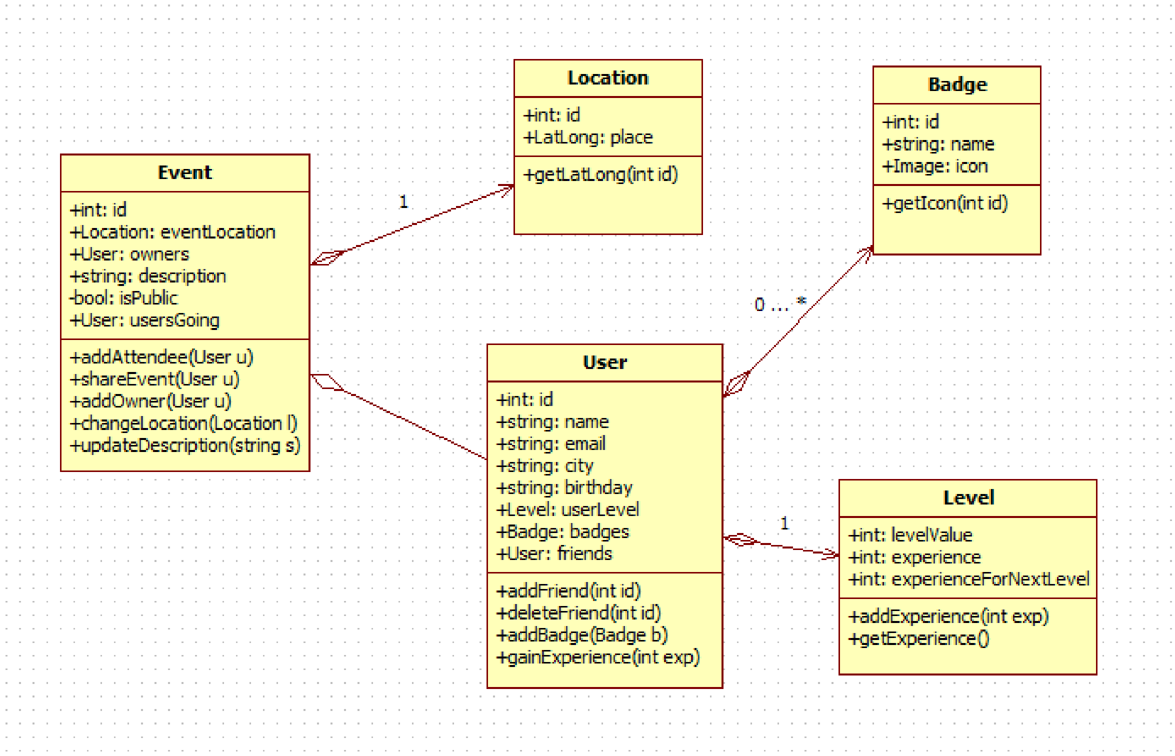


Figure 48: UML Diagram

5. Behavioral Model and Description

This section describes main events of the software and its behavior related to them.

5.1. Description for software behavior

Our software focuses user actions and cases. As a consequence, there are no state diagrams for this section because our application does not have different states. To explain software’s behavioral model, main events and system’s reaction will be given.

Firstly to use the system, users require install application to their mobile device.

Login Event: After installation, the user can start the software. In order to use system properly, the user needs a unique account. He/she can register a new account with other social network accounts such as his/her “Facebook” account. If user picks this option, system will automatically connect to Facebook server and get required account information with the help of Facebook API. This process can be done also for other social networks. If the other option is selected, which is registering with e-mail account, system will communicate with its own database and create an account.

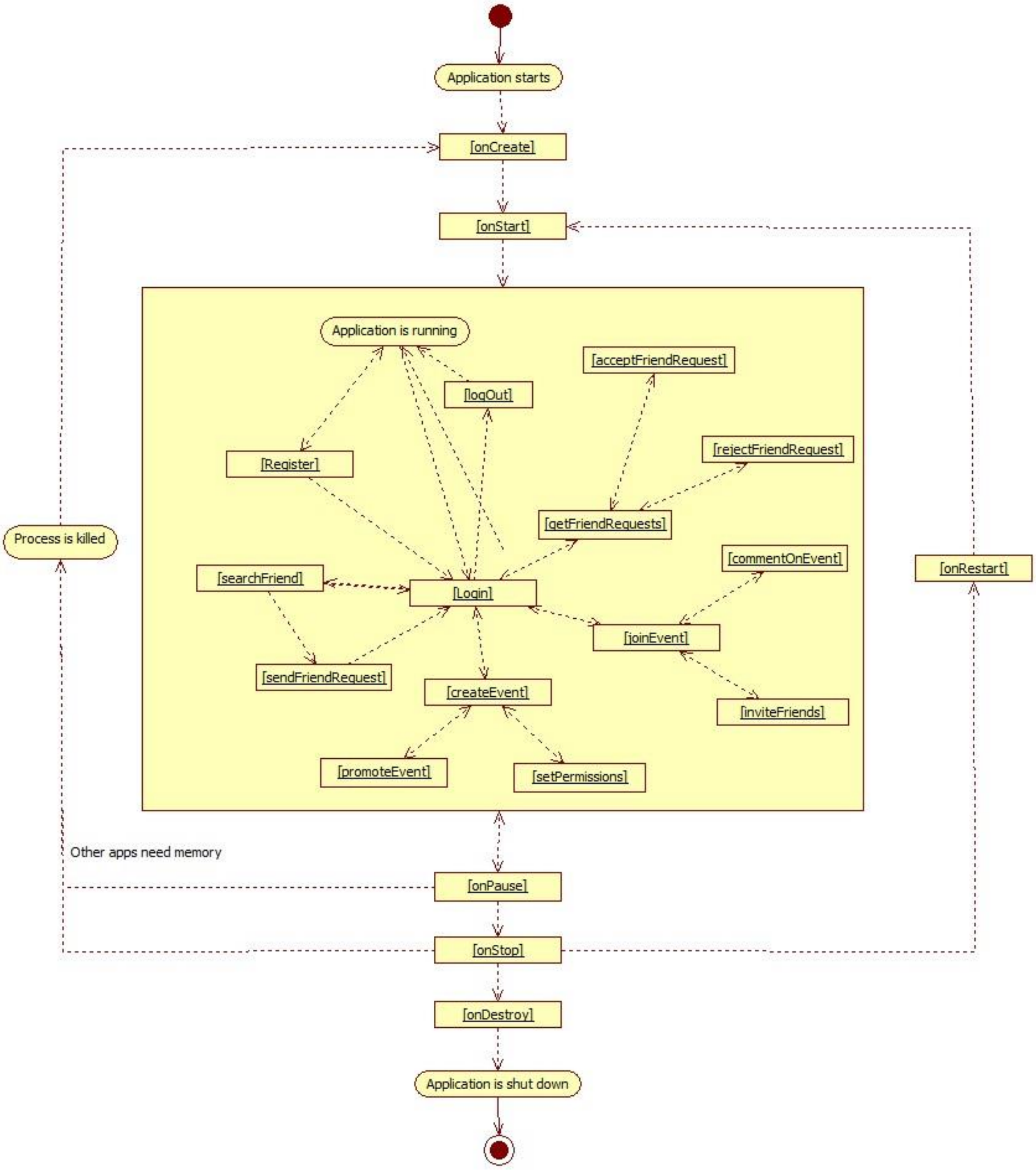
View trending events nearby: To be able to see trending events nearby, user has to open Wi-Fi or GPS. Then system will connect to Google Maps API and load the map to the screen. System will automatically get users location with device's operating system and show at the map. With the given location information, server will search events that are close to user's location at its database and send the data to application. Finally, application will show given data at the map.

Get route directions to event location: To be able to get directions the event location, user has to pick event. After deciding the event to join, user will request the route information from the application. System will force user to open device's Wi-Fi or GPS for location information. System will collect user's location and event's location data. Then it will send the data to Google Directions Service with using API. Google Directions Service will give route information and application will process this information to the map. Finally, each route step will be appeared at map.

Profile, Friend and Event actions: All functionalities related to users' profile (edit profile, change profile picture, etc.), friends (add friend, delete friend, etc.) and events (add events, share events, join events, etc.) that does not require any other API (Facebook, Google) will be handled with systems own database. User will interact with interface and requested functionality will be handled. Then system will connect to server. After that, server will handle request and run proper database operations. Afterwards, data will be given back to user's device and application will handle data. Finally, application will show result of the action of user at its GUI.

5.2. State Transition Diagrams

Since our software will focus on mobile part, we will give mobile state transition diagram. Web page state transition diagram is very similar to mobile state transition diagram, only difference is mobile part of the software will have more states since mobile software works differently comparing to browsers. State transition diagram of our project can be seen below:



Briefly:

onCreate State : Software will be ready to start.

onStart State : Software will be started and put into front as main application that is working on mobile device.

onPause State : Software will be taken down from 'main application' to 'waiting to work application'. Software will be able to work again at the same state because application is not fully closed. Mobile operating systems can kill the application manually if device needs more memory. If this happens, software needs to be created again at onCreate state.

onStop State : Software will stop working at this state. In order to open the software again, operating system will restart the application. Again, operating system can decide to kill software's process and if this happens, application has to enter onCreate state to start again.

onRestart State : Software will be ready to be started again.

onDestroy State : Software will be fully stopped and closed. Application will have no presence in memory.

Application Running State: Software is active and there are two states that user can pick: Register and login states.

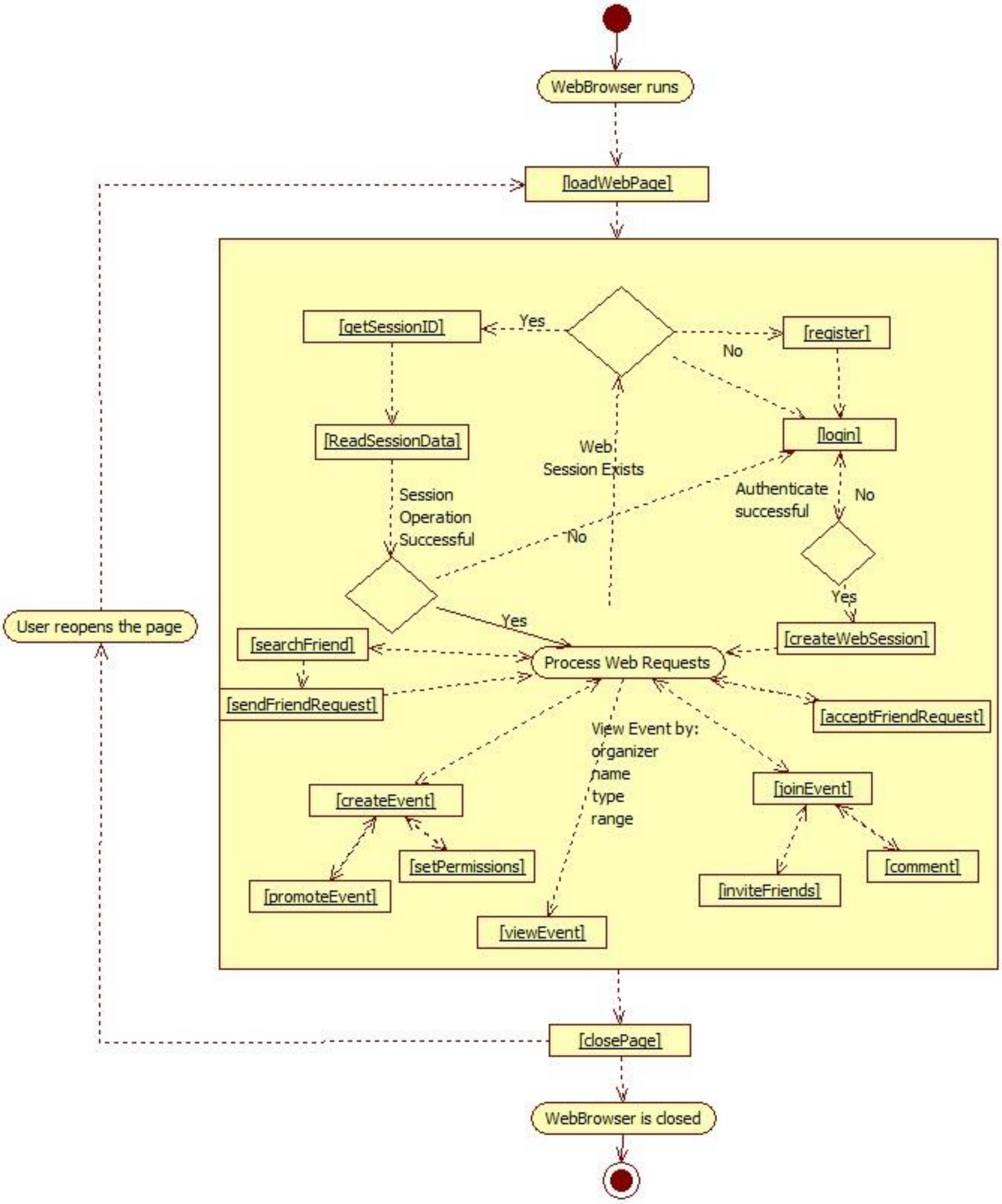
Register State: User will be able to register himself/herself to the system and use the software with login state later on.

Login State: Software will authenticate the user and user will be ready to use system's main functionalities except 'register'.

After Login State: User will be able to system's functionalities and system will handle these web requests. User will be able to use functionalities such as Friendship functionalities, Event operations, Level and badge process and the other functionalities. These functionalities are all defined in Functional Requirements part (3.2) of this document. Software will return to Login State when user logouts.

Application Running State, Register State, Login State and After Login State can be transferred to onPause State and to other states after onPause State as one can see from the State Transition Diagram.

As stated before, web state transitions are similar to mobile state transitions. There are no intermediate operating system states such as onCreate, onPause, onRestart, onStop, onDestroy. When browser opens the web part of our software, it will follow same states only difference is there can be an existing session since user can open this web page before and then close it. If session exists, software will be at Process Web Request State; else, software will be at Login or Register states. Here is the web state diagram of the project:


WebBrowser runs
[loadWebPage]
[getSessionID]
Yes
[register]
No
[ReadSessionData]
Web Session Exists
[login]
Session Operation Successful
Authenticate successful
No
No
User reopens the page
Yes
Yes
[searchFriend]
Process Web Requests
[createWebSession]
[sendFriendRequest]
[acceptFriendRequest]
View Event by:
organizer
name
type
range
[createEvent]
[joinEvent]
[setPermissions]
[comment]
[promoteEvent]
[inviteFriends]
[viewEvent]
[closePage]
WebBrowser is closed

6. Planning

6.1. Team Structure

Attila Özgit – Advisor

Buğra Özkan – Advisor

Mert Ergun – Researcher, Developer, Windows Phone Developer, Server Side Coder

Yaşar Barış Ulu – Researcher, Developer, Database Management, Server Side Coder

Kemal Çağın Gülşen – Researcher, Developer, Android Developer, Database Management

Kerem Gökhan – Researcher, Developer, iOS Developer, Server Side Coder

6.2. Estimation

Week - 1 (October 28th - November 3rd)

1. Deciding of tools and frameworks that are going to be used for our project.

Week - 2 (November 4th - November 10th)

1. Design of Database (ER Diagram, Relational Schema Design)
2. Design of Model Classes (Initial version of UML Diagram)

Week - 3 (November 11th - November 17th)

1. Design of database (Relational Database Implementation)
2. Generating Model Classes

Week - 4 (November 18th - November 24th)

1. Integration of both model classes and database with frameworks.
2. Design Report

Week - 5 (November 25th - December 1st)

1. Implementation of methods that specified in UML Diagram (Server Side)

Week - 6 (December 2nd - December 8th)

1. Implementation of methods that specified in UML Diagram (Server Side)

Week - 7 (December 9th - December 15th)

1. Implementation of methods that specified in UML Diagram (Server Side)

Week - 8 (December 16th - December 22nd)

1. Implementation of methods that specified in UML Diagram (Server Side)
2. Constructing web service
3. (End of Iteration 1) , Updated Reports

Week - 9 (December 23rd - December 29th)

1. Constructing web service

Week - 10 (December 30th - January 5th)

1. Connection between web service & mobile application

Week - 11 (January 6th - January 12th)

1. Connection between web service & mobile application

Week - 12 (January 13th - January 19th)

1. Connection between web service & mobile application
2. Design of first prototype of mobile application interface

Week - 13 (January 20th - January 26th)

1. Design of first prototype of mobile application interface

Week - 14(January 27th - ...)

1. Design of first prototype of mobile application interface
2. Prototype Demonstration

6.3. Process Model

Project members are going to use Agile Method, which is shown below, for the project. Firstly, the requirements of the functionality will be implemented for server-side. Then, that functionality will be implemented for client-side applications. After developing enough functionalities, they will be integrated to the system and tested. Later on a new version of the system will be released. With the clients' help, changes will be made and errors will be fixed. Finally, the system will be tested again and this process will end. After all, next iteration will start for adding new functionalities.

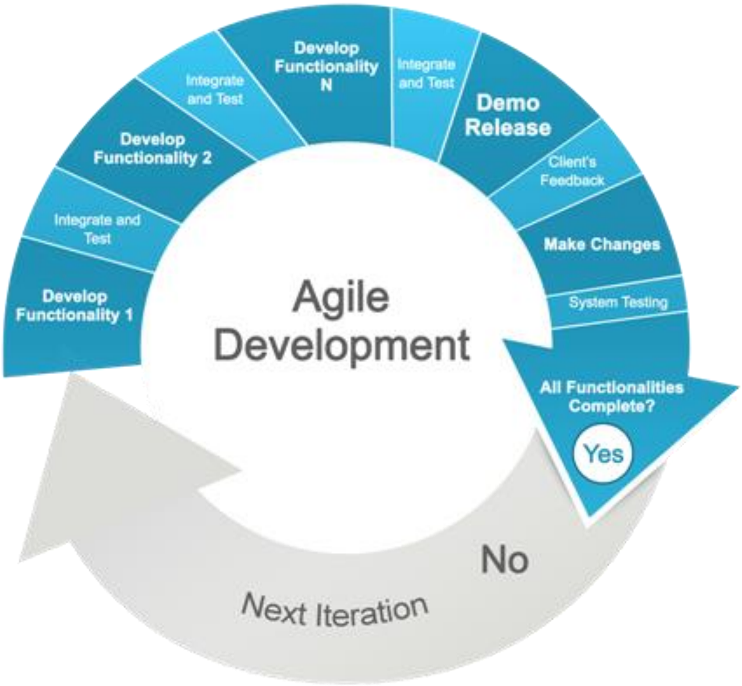


Figure 49: Agile Development

Image taken from: http://blogs.salesforce.com/.a/6a00e54ee3905b8833017c31e54390970b-pi

7. Conclusion

This Software Requirements Specifications document is prepared for giving detailed requirement information of Event-Based Social Network Project. First, purpose and scope of the project is explained. Secondly, the project is described with more details. Then all specific requirements of the project are stated. Moreover data model and behavior model is explained. At the following steps of the project, this Software Requirements Specifications document will be used as a basis.