SWE599 Project, Spring 2014
Instructor: Fatih ALAGÖZ

SWE599 Project
LRB: LittleRedButton
Requirements Specifications Document

09.03.2014
Revision 1.0

By: Mustafa Göksu GÜRKAŞ
Student Id: 2011719225

Revision History

Revision	Date	Explanation
1.0	09.03.2014	Initial requirements

Table of Contents

1. Introduction

The purpose of this software project is to design and develop a new mobile application on Android devices. The mobile application is to be implemented with JAVA for Android and by using a database system like MySQL. One aims to accomplish the following tasks by using this system.

1. Users shall be able to locate himself/herself by using his smart phone's GPS/Network location services.
2. Users shall be able to add or delete his/her contacts in his/her smart phone as recipients.
3. Users shall be able to display his/her contact list that takes place in his/her smart phone memory.
4. Users shall be able to login to/logout from his/her Facebook account by using this application.
5. Users shall be able to notify his/her friends with his/her current GPS location coordinates as an emergency message via his/her Facebook profile.
6. Users shall be able to login to/logout from his/her Twitter account by using this application.
7. Users shall be able to notify his/her friends with his/her current GPS location coordinates as an emergency message via his/her Twitter profile.
8. Users shall be able to notify his/her selected recipients with his/her current GPS location coordinates as an emergency message via SMS.
9. Users shall be able to notify his/her selected recipients with his/her current GPS location coordinates as an emergency message via e-mail.
10. Users shall be able to let the application track himself/herself by sending emergency messages with his/her current (updated) GPS location coordinates to above channels every 60 seconds.
11. Users shall be able to display his/her current GPS location coordinates on Google Maps.
12. Users shall be able to display nearest police stations and health institutions to his/her current GPS location coordinates on Google Maps.
13. Users shall be able to configure the application's behavior by arranging the settings of the application.

There will be only one actor in this mobile application which is the user himself/herself.

Operations related to the user of the application can be seen in the use case diagram that is provided in Section 2.1 of this document. The functional and non-functional requirements of the mobile application in addition to the UML notations are given in the following sections of this document.

2. Requirements

2.1. Requirement 1. Main User Interface and Functions

Main user interface and functions of LittleRedButton (LRB) application shall be composed of the following items:

LRB - APPLICATION CONFIGURATION OPERATIONS

1. I am a user and I want to login to my Facebook account.
2. I am a user and I want to logout from my Facebook account.
3. I am a user and I want to login to my Twitter account.
4. I am a user and I want to logout from my Twitter account.
5. I am a user and I want to display all contacts that take place on my phone memory.
6. I am a user and I want to display the recipients that I selected.
7. I am a user and I want to select some of my contacts that take place on my phone memory as recipients.
8. I am a user and I want to remove some of the recipients.
9. I am a user and I want to configure application's behavior through application settings.

LRB - SENDING NOTIFICATION OPERATIONS

1. I am a user and I want to notify my current GPS location coordinates to selected contacts through selected channels of the application.

LRB - DISPLAY MAPS OPERATIONS

1. I am a user and I want to display my current GPS location coordinates and nearest police stations/health institutions to that coordinates on Google maps.

All functions above are links to related Android activities.

The use case diagram below depicts the overall functionalities of the LittleRedButton Android application.

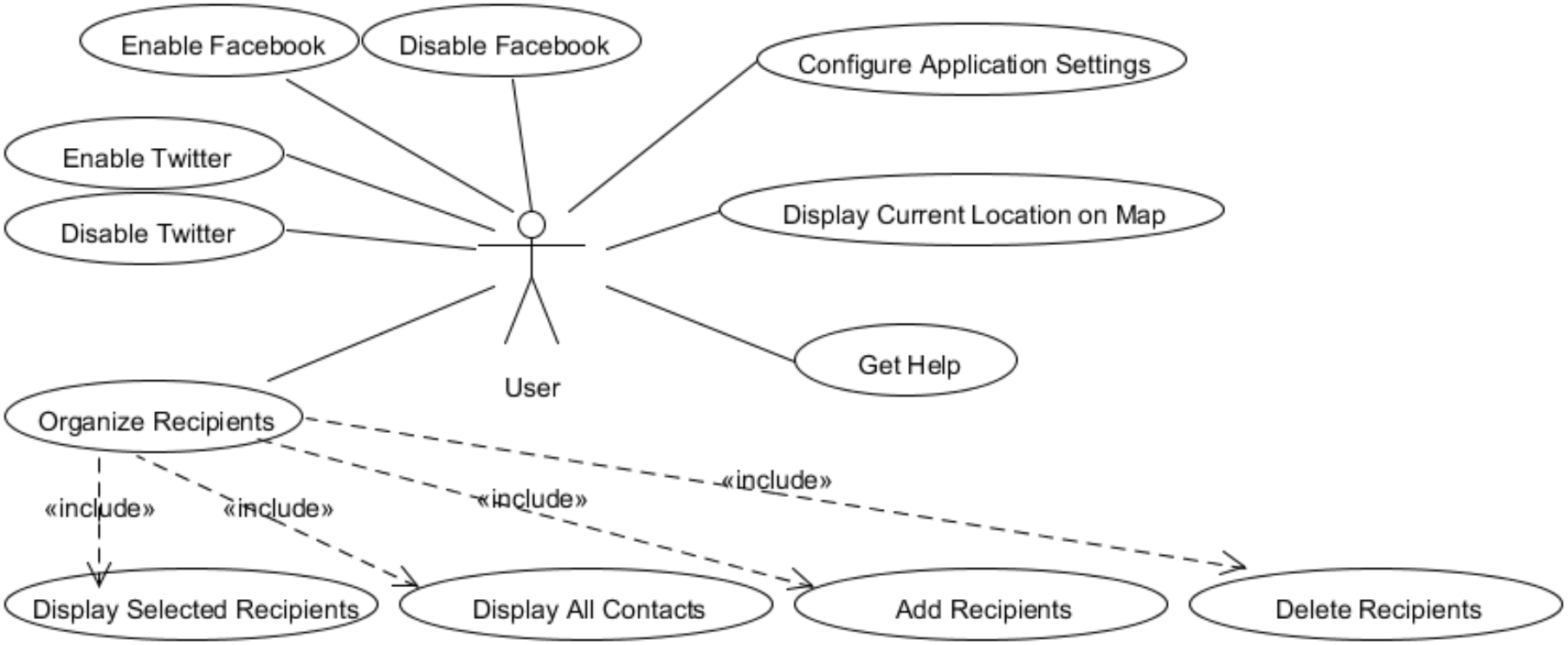

Figure 1: Use Case Diagram

2.2. Requirement 2. Function 1: Configure Facebook

2.2.1 Use Case Name: Enable Facebook

Brief Description:
User shall be able to enable his/her Facebook account via the application in order to let the application to post his current GPS location coordinates to his/her Facebook wall when he/she declares an emergency by hitting on the "Get Help" button.

Actors:

- User

Triggers:
User indicates that he/she wants to login to his/her Facebook account through LittleRedButton.

Pre-conditions:

1. User has to be logged out from Facebook if he wants to login to Facebook.

Post-conditions:
User will be logged into his/her Facebook account.

Use Case Flow:

1. User indicates that he/she wants to enable his/her Facebook account.
2. User clicks on the "Enable Facebook" button that takes place on the main screen.
3. System checks whether the application has the rights to access Facebook API.
4. System directs the user to Facebook Login Page.
5. User enters his/her Facebook username to "Email or Phone" field.
6. User enters his/her Facebook password to "Password" field.
7. User hits on "Log In" button.
8. System checks whether the provided credentials are valid or not.
9. System asks the user whether he/she accepts the permissions on Facebook in order to send posts automatically on behalf of the user.
10. User clicks on "OK" button in order to give necessary permissions to the system.
11. System updates the "Enable Facebook" button as "Disable Facebook" button.
12. System directs the user back to the application.

2.2.2 Use Case Name: Disable Facebook

Brief Description:
User shall be able to disable his/her Facebook account via the application in order to avoid the application to post his current GPS location coordinates to his/her Facebook wall when he/she declares an emergency by hitting on the "Get Help" button.

Actors:

- User

Triggers:
User indicates that he/she wants to logout from his/her Facebook account through LittleRedButton.

Pre-conditions:

1. User has to be logged into Facebook if he wants to logout from Facebook.

Post-conditions:
User will be logged out from his/her Facebook account.

Use Case Flow:

1. User indicates that he/she wants to disable his/her Facebook account.
2. User clicks on the "Disable Facebook" button that takes place on the main screen.
3. System clears all the user credentials that are stored during Facebook login.
4. System updates the "Disable Facebook" button as "Enable Facebook" button.

2.3. Requirement 3. Function 2: Configure Twitter

2.3.1 Use Case Name: Enable Twitter

Brief Description:
User shall be able to enable his/her Twitter account via the application in order to let the application to send his current GPS location coordinates to his/her Twitter status when he/she declares an emergency by hitting on the "Get Help" button.

Actors:

- User

Triggers:
User indicates that he/she wants to login to his/her Twitter account through LittleRedButton.

Pre-conditions:

1. User has to be logged out from Twitter if he wants to login to Twitter.

Post-conditions:

User will be logged into his/her Twitter account.

Use Case Flow:

1. User indicates that he/she wants to enable his/her Twitter account.
2. User clicks on the "Enable Twitter" button that takes place on the main screen.
3. System checks whether the application has the rights to access Twitter API.
4. System directs the user to Twitter Login Page.
5. System asks the user whether he/she accepts the permissions on Twitter in order to send tweets automatically on behalf of the user.
6. User enters his/her Twitter username to "Username or email" field.
7. User enters his/her Twitter password to "Password" field.
8. User hits on "Sign In" button in order to login and give necessary permissions to the system.
9. System checks whether the provided credentials are valid or not.
10. System updates the "Enable Twitter" button as "Disable Twitter" button.
11. System directs the user back to the application.

2.3.2 Use Case Name: Disable Twitter

Brief Description:

User shall be able to disable his/her Twitter account via the application in order to avoid the application to send his current GPS location coordinates to his/her Twitter status when he/she declares an emergency by hitting on the "Get Help" button.

Actors:

- User

Triggers:

User indicates that he/she wants to logout from his/her Twitter account through LittleRedButton.

Pre-conditions:

1. User has to be logged into Twitter if he wants to logout from Twitter.

Post-conditions:

User will be logged out from his/her Twitter account.

Use Case Flow:

1. User indicates that he/she wants to disable his/her Twitter account.
2. User clicks on the "Disable Twitter" button that takes place on the main screen.
3. System clears all the user credentials that are stored during Twitter login.
4. System updates the "Disable Twitter" button as "Enable Twitter" button.

2.4. Requirement 4. Function 3: Organize Recipients

2.4.1 Use Case Name: Organize Recipients

Brief Description:

User shall be able to display/add/delete some recipients among the contacts from the phone memory who shall receive notifications from the user through email or SMS when he/she declares an emergency by hitting on the "Get Help" button.

Actors:

- User

Triggers:

User indicates that he/she wants to organize the recipients that will receive notifications from the user in an emergency.

Pre-conditions:

None.

Post-conditions:

Recipients will be added or deleted to/from recipients list.

Use Case Flow:

1. User indicates that he/she wants to organize his/her recipients list.
2. User clicks on the "Organize e-mail & SMS Recipients" button that takes place on the main screen.
3. System displays the list of the current recipients **(include "Display Selected Recipients")**.
4. User wants to delete a recipient that takes place on the recipients list **(include "Delete Recipients")**.
5. User wants to add a recipient to the recipients list **(include "Add Recipients")**.
6. User wants to display all contacts that take place on the phone memory **(include "Display All Contacts")**.
7. User hits the "OK" button.
8. System directs the user to the main screen of the application.

2.4.2 Use Case Name: Display Selected Recipients

Brief Description:
User shall be able to display the current recipients list.
Actors:
- User

Triggers:
User indicates that he/she wants to organize the recipient list.
Pre-conditions:
None.
Post-conditions:
None.
Use Case Flow:
1. System queries the recipients list database to fetch all the recipients previously added by the user.
2. For each recipient that is present in the recipients list, system displays the name of the recipient and a "Delete" button next to his/her name in order to let the user delete this recipient later on in a formatted table view.

2.4.3 Use Case Name: Display All Contacts

Brief Description:
User shall be able to display all contacts from the phone memory.
Actors:
- User

Triggers:
User indicates that he/she wants to display all contacts.
Pre-conditions:
None.
Post-conditions:
None.
Use Case Flow:
1. User clicks on the "Explore Contacts" button.
2. System queries the contacts that take place on the memory phone and fetch his/her name, phone number and e-mail address.
3. For each contact on the phone memory, system displays the name, phone number and e-mail address in a formatted table view.
4. User hits on the back button to return to previous screen.

2.4.4 Use Case Name: Add Recipients

Brief Description:
User shall be able to add a contact from his/her phone memory as a recipient (A recipient is a contact who will receive notifications from the user when he/she declares an emergency by hitting on the "Get Help" button).
Actors:
- User

Triggers:
User indicates that he/she wants add one of his/her phone contacts as a recipient.
Pre-conditions:
None.
Post-conditions:
A new recipient will be added to recipients list.
Use Case Flow:
1. User clicks on the "Add Recipient" button.
2. System displays the list of all contacts from the phone memory.
3. User explores the contacts list and finds the one that he/she wants to add.
4. User clicks on the name of the contact.
5. System copies the name, phone number and e-mail address of that selected contact to the application and adds it to the recipients list.
6. System adds the name of the selected recipient to recipients list and displays the name of all recipients and adds a "Delete" button next to his/her name in order to let the user delete this recipient later on.

2.4.5 Use Case Name: Delete Recipients

Brief Description:
User shall be able to delete a recipient from the recipients list.
Actors:
- User

Triggers:
User indicates that he/she wants delete one of the recipients.
Pre-conditions:
None.
Post-conditions:
The selected recipient will be deleted from the recipients list.
Use Case Flow:
1. User clicks on the "Delete" button that is next to the name of the recipient that he/she wants to delete.
2. System suppresses the name of the recipient from the recipients list screen.
3. System deletes the record of the selected recipient from the recipients list database.

2.5. Requirement 5. Function 4: Display Current Location on Map

2.5.1 Use Case Name: Display Current Location on Map

Brief Description:
User shall be able to display his/her current GPS location coordinates on Google Maps. User shall also be able to display the nearest police stations and health institutions (hospitals, medical centers etc.)on Google Maps, next to his current location marker.
Actors:
- User

Triggers:
User indicates that he/she wants to display his/her current location coordinates and the nearest help centers on Google Maps.
Pre-conditions:
None.
Post-conditions:
User location and nearest help locations will be marked on Google Maps.
Use Case Flow:
1. User indicates that he/she wants to display his/her current coordinates on Google Maps.
2. User clicks on "Where am I?" button that takes place on the main screen.
3. System gets the user's current GPS location coordinates.
4. System sends the current GPS location coordinates and some decisive parameters (what to look as a help point (police and hospital points), in which range should Google Maps search for those points etc.) to Google Maps servers.
5. Google Maps sends the responsive data to the system.
6. System parses that map data sorts out what the user needs (name, address etc.).
7. System displays the current GPS location of the user with a green marker on Google Maps.
8. System displays the nearest police stations to the location of the user with a blue marker on Google Maps.
9. System displays the nearest health institutions to the location of the user with a red marker on Google Maps.
10. User clicks on one of the markers that are pointed on the map.
11. System displays the name and the address of the place that the user selected.
12. User zooms in/out the map as he/she needs to explore the area much in detail.

2.6. Requirement 6. Function 5: Configure Application Settings

2.6.1 Use Case Name: Configure Application Settings

Brief Description:
User shall be able to configure application settings which includes setting the Message Text (a plain text that is added to emergency message), Display Name (a label which is used to identify the user on the messages that he sends through e-mail and SMS), Gmail Settings (optional – which is composed of Gmail username and password pair, if he/she wants to send e-mail by his/her own Gmail account. If the user leaves that information empty, system shall send the e-mails by using its default Gmail account) and tracking functionality (Tracking shall be used to update user's current GPS location every 60 seconds to the selected notification channels).

Actors:

- User

Triggers:

User indicates that he/she wants to configure application's settings.

Pre-conditions:

None.

Post-conditions:

Application settings will be updated upon user needs.

Use Case Flow:

1. User indicates that he/she wants to change application's settings.
2. User clicks on the "Menu" button of his/her Android device.
3. User clicks on the "Settings" button from the inflated menu.
4. System displays the settings list to the user.
5. User hits the "Message Text" button.
6. System asks the user to enter a plain text that will be added to emergency messages.
7. User enters his/her emergency message to be sent to recipients and clicks on "OK" button.
8. System stores the message text to be sent.
9. User hits the "Display Name" button.
10. System asks the user to enter a name which will be used as the sender of messages that would be sent on emergency occasions.
11. User enters his/her name to the "Display Name" text field and hits on "OK" button.
12. System stores the value of that text field as display name.
13. User hits on "Gmail Settings" button from the inflated menu.
14. System displays "User Name" and "Password" options to the user.
15. Users clicks on either "User Name" or "Password" button in order to provide his/her credentials in order to send e-mails by his/her own Gmail account.
16. System asks the user to enter his user name or password to the related field.
17. User enters his/her user name and password and clicks ok "OK" button.
18. System stores the user's Gmail credentials.
19. User checks/unchecks the "Track me" check box if he/she wants/doesn't want to be tracked by the application.
20. System stores the user's demand on tracking facility of the application.

2.7. Requirement 7. Function 6: Get Help

2.7.1 Use Case Name: Get Help

Brief Description:

User shall be able to send his/her current GPS location coordinates to his/her Facebook wall, Twitter account and to his/her selected recipients as e-mail and SMS messages.

Actors:

- User

Triggers:

User indicates that he/she wants to notify his/her recipients on his current situation.

Pre-conditions:

1. User has to be logged in to his/her Facebook account if he/she wants to post the emergency message to his/her Facebook wall.
2. User has to be logged in to his/her Twitter account if he/she wants to update his/her Twitter status.
3. User has to select some recipients among the phone contacts with phone number/e-mail addresses if he/she wants to send the emergency message to his/her recipients.

Post-conditions:

An emergency message will be sent through selected notification channels.

Use Case Flow:

1. User indicates that he/she wants to declare his/her emergency to his friends.
2. User clicks on the "Get Help" button that resides on the main screen.
3. System shoots a picture of the moment immediately.
4. System gets the current GPS location coordinates of the user's phone.
5. System checks whether the user selected Facebook as a notification channel.
6. System posts the photo and the message with a Google Maps link (pointing user's current GPS location coordinates) to user's Facebook wall.
7. System checks whether the user selected Twitter as a notification channel.
8. System updates the photo and the message with a Google Maps link (pointing user's current GPS location coordinates) to user's Twitter status.

9. For each recipient in the recipients list, system sends an SMS that is composed of the emergency message and GPS location coordinates as a Google Maps link.
10. For each recipient in the recipients list, system sends an e-mail that is composed of the emergency message and GPS location coordinates as a Google Maps link.
11. If the user has enabled "Track Me" option from the application settings, system waits for 60 seconds and returns to Step 3 of this use case and continues.
12. If the user has disabled "Track Me" option from the application settings, use case ends.

2.8. Requirement 8. Development Environment

The software will be developed in an object oriented programming language such as JAVA. The database of LRB shall be MySQL. The development and operation environment shall be Windows.

2.9. Requirement 9. Security

The actions in both mobile applications will not violate rules and regulations for the applications being distributed through application market or store. Unlike iOS applications, as Android application gives permission to the developer for sending e-mail and SMS without user intervention, those messages shall be sent directly to the recipients.

Preferences Requirement

All user's preferences that are set by the user by following Requirement 6 of this document shall be stored in Shred Preferences data structure of Android device.

Recipients Requirement

All the contacts that are saved by the user as recipients of the application shall be stored in MySQL database tables.

Client/Server Connection Requirement

All important client-server operations through the Internet will be through SSL to provide security.

2.10. Requirement 10. Reconfigurability

Not applicable.

2.11. Requirement 11. Logging

None.

3. Testing

The test document must cover the details of testing the correct operation and boundary conditions of each requirement given in Section 2. The details of the testing document will be produced during the design. No acceptance test is specified as part of the requirements specification.