**Emergency Information on Mobile**

Software Requirement Specification

By

Putchakarn Jaikon 542115031

Sawatdiporn Kitirot 542115065

Department of Software Engineering

College of Arts, Media and Technology

Chiang Mai University

Project Advisor

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Aj.Chartchai Doungsa-ard**

**Document History**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Document Name** | **Details** | **Status** | **Date** | **Viewable** | **Reviewer** | **Responsible** |
| **Documents** | | | | | | |
| **EIOM-SRS-V.0.1.docx** | **Chapter 1**      Introduction | Draft | 13/3/2014 | PJ, SK, CD | PJ, SK | PJ, SK |
| **EIOM-SRS-V.0.2.docx** | **Chapter 2**      URS      SRS | Draft | 15/3/2014 | PJ, SK, CD | PJ, SK | PJ, SK |
| **EIOM- SRS -V.0.3.docx** | **Modify Chapter 2**      URS      SRS | Draft | 20/3/2014 | PJ, SK, CD | PJ, SK | PJ, SK |
| **EIOM- SRS -V.0.4.docx** | **Chapter 3**      Use Case Scenario      Use Case Description | Draft | 24/3/2014 | PJ, SK, CD | PJ, SK | PJ, SK |
| **EIOM- SRS -V.0.5.docx** | **Modify Chapter 3**      Use Case Scenario      Use Case Description | Draft | 30/3/2014 | PJ, SK, CD | PJ, SK | PJ, SK |
| **EIOM- SRS -V.0.6.docx** | **Modify Chapter 3**      Add image of Chapter 3 | Draft | 5/4/2014 | PJ, SK, CD | PJ, SK | PJ, SK |
| **EIOM- SRS -V.1.0.docx** | **Review Chapter1- 3**      Add Table of content and cover page | Release | 8/4/2014 | PJ, SK, CD | PJ, SK | PJ, SK |
| **EIOM- SRS -V.1.1.docx** |  Modify chapter 1-3 | Release | 30/7/2014 | PJ, SK, CD | PJ, SK | PJ, SK |
| **EIOM- SRS -V.2.0.docx** |  Add URS-16 – URS-22 | Release | 21/10/2014 | PJ, SK, CD | PJ, SK | PJ, SK |

PJ – Putchakarn Jaikon, SK – Sawatdiporn Kitirot, CD – Chartchai Doungsa-ard

Table of Contents

[Chapter One | Introduction 4](#_Toc394347987)

[1.1 Objective 4](#_Toc394347988)

[1.2 Intended Audience and Reading Suggestions 4](#_Toc394347989)

[1.3 Project Scope 5](#_Toc394347990)

[1.4 User Characteristic 5](#_Toc394347991)

[1.5 Acronyms and Definitions 5](#_Toc394347992)

[Chapter Two | Functional Requirement 7](#_Toc394347993)

[2.1 User Requirement Specification 7](#_Toc394347994)

[2.2 Software Requirement Specification 8](#_Toc394347995)

[Chapter Three | Specific Requirement 10](#_Toc394347996)

[3.1 Use Case Scenarios 10](#_Toc394347997)

[3.2 Use Case Description 11](#_Toc394347998)

# Chapter One | Introduction

## Objective

Software Requirement Specification of Emergency Information on Mobile is the document that describes each function, process, software environment, and constraint. The document is based on the contract and project plan. It is created for software developer and software tester to more understanding in the requirement. The purpose of Emergency Information on Mobile is providing area for presenting information of help place for the users. Admin also can manage information of help place.

## Intended Audience and Reading Suggestions

The Software Requirement Specification was created for everyone that involved with the Emergency Information on Mobile. The document of Software Requirement Specification will make the benefit for people as following:

**1.2.1 Development Team**

• Make strategies and planning process convenient

• Improve the system in right needed and use for prioritize what process become first or what process should be after.

• Reference in testing system because tester will validate if the system is correct and appropriate.

• Verify and specify requirements to ensure the same understanding about requirements. The ensuring can help in working and discuss all detail about requirements for avoid any error in work.

• Control and guarantee qualities of the system to make it right regarding the standard and contract.

• Easier discussion all information about the system because the constant and reliability source of the system.

**1.2.2 Customer**

• Easy for users to understand about quality and limitation of the system.

• Ensure the same understanding about requirement.

## Project Scope

The objective of this software requirement specification is to specify requirements to establish the application that:

• Emergency Information on Mobile is a web application for both computer website and mobile devices.

• Emergency Information on Mobile for mobile devices supports android operating system.

• Emergency Information on Mobile provides offline map and last downloaded information of help places.

• Emergency Information on Mobile supports English language.

• Emergency Information on Mobile provides admin to manage the information of help place, such as add, edit and remove help place.

## User Characteristic

**1.4.1 User**

• The group of person who already to use online map and offline map. The user of the application that will receive the information of help place to call and get an address.

**1.4.2 Administrator**

• The group of person who manage the information of the help place.

## 1.5 Acronyms and Definitions

**1.5.1 Acronyms**

EIOM Emergency Information on Mobile

SRS Software Requirement Specification

URS User Requirement Specification

UC Use Case

UI User Interface

**1.5.2 Definitions**

|  |  |
| --- | --- |
| Feature | Transformation of input parameters to output parameters based on a specified algorithm. It describes the functionality of a product. Used for requirements analysis, design, coding, testing or maintenance. [IEEE90] |
| IEEE | Institute for Electrical and Electronics Engineers. Biggest global interest group for engineers of different branches and for computer scientists. [IEEE90] |
| Requirement | (1) A condition or capability needed by a user to solve a problem or achieve an objective. (2) A condition or capability that must be met or processed by system or system component to satisfy a contract, standard, specification, or other formally imposed document. (3) A documented representation of a condition or capability as in definition (1) or (2). [IEEE90] |
| Specification | Precise description of an activity or work product which serves as basis or input for further activities or work product. A specification can comprise requirements to a product and how they will be solved. Different parts of a specification (e.g. what is to be done, how it will be done) must not be mixed. [IEEE90] |
| Design | The period of time in the software life cycle during which the designs for architecture, software component, interfaces and data are created, documented, and verified to satisfy requirements. [IEEE90] |
| UML | Unified Modeling Language. Standardized notation for Modeling design descriptions, architecture or scenarios. Not depending on a specific method. Issued and maintained by the Object Management Group (OMG). [IEEE90] |
| Use Case | (1) Concept to describe a system based on usage of system resource by its environment. Characterized by an objective set of interactions within and at the borders a scenario (Usage approach, operational scenario) from the perspective of this user. [IEEE90] |

# Chapter Two | Functional Requirement

## 2.1 User Requirement Specification

* **Feature5: Manage information system**

**URS-16:** The mobile application can get list of all help places in the database.

**URS-17:** The mobile application can get the nearest help place by the selected category.

**URS-18:** The mobile application can get list of help places where locate in the setting scope.

* **Feature2: Search information system**

**URS-19:** The user can search help place’s name by keyword in online map.

**URS-20:** The user can find the nearest help place by selection the category in online map.

* **Feature4: Automatic collecting data system**

**URS-21:** The user can set the scope for downloading data.

**URS-22:** The mobile application can collect help place information automatically.

## 2.2 Software Requirement Specification

**URS-16: The mobile application can get list of all help places in the database.**

**SRS-25:** The system shall retrieve all help places from system database.

**SRS-64:** The system shall show list of all help places in form of JSON.

**URS-17:** **The mobile application can get the nearest help place by the selected category.**

**SRS-65:** The system shall find the nearest help place by the selected category.

**SRS-66:** The system shall show the nearest help place in form of JSON.

**URS-18: The mobile application can get list of help places where locate in the setting scope.**

**SRS-67:** The system shall retrieve list of help places where locate in the setting scope from the database.

**SRS-68:** The system shall show list of help places where locate in setting scope in form of JSON.

**URS-19: The user can search help place’s name by keyword in online map.**

**SRS-69:** The system shall provide search button UI.

**SRS-70:** The system shall provide text field UI.

**SRS-71:** The system shall receive all help places from server.

**SRS-72:** The system shall search help places by keyword from user inputting.

**SRS-73:** The system shall matching keyword with help places.

**SRS-74:** The system shall display help places which matching with keyword.

**SRS-75:** The system shall provide map with help place that user selection from searching by keyword.

**URS-20: The user can find the nearest help place by selection the category in online map.**

**SRS-76:** The system shall provide category button.

**SRS-77:** The system shall receive the current location of user and category’s id.

**SRS-78:** The system shall send the current location of user and category’s id to the server.

**SRS-79:** The system shall receive help place object from the server.

**SRS-80:** The system shall change color of marker to show the position of nearest help place by searching.

**SRS-81:** The system shall display the nearest help place of each category from user selection.

**URS-21: The user can set the scope for downloading data.**

**SRS-82:** The system shall provide menu setting UI.

**SRS-83:** The system shall provide number for setting scope with radio button UI.

**SRS-84:** The system shall define a default value of scope.

**URS-22: The mobile application can collect help place information automatically.**

**SRS-85:** The system shall check distance between original coordinates and new coordinates position automatically.

**SRS-86:** The system shall send latitude and longitude of user to the server.

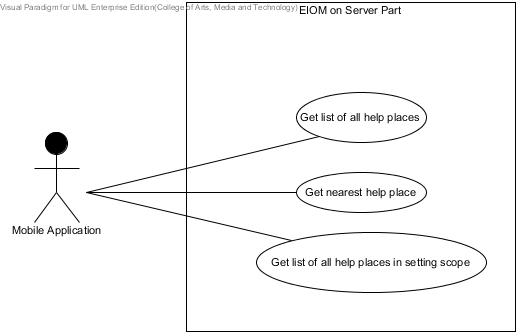
**SRS-87:** The system shall receive help places from server.

**SRS-88:** The system shall delete the latest data of help places from the database.

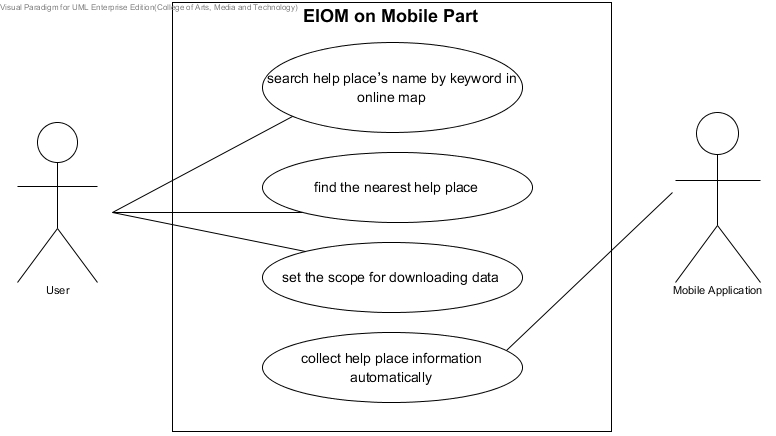
**SRS-89:** The system shall add new data of help places into database.

# Chapter Three | Specific Requirement

## 3.1 Use Case Scenarios

**• Emergency Information Server**

**Figure 1 Show use case of “Emergency Information Server”**

**• Emergency Information on Mobile**

**Figure 2 Show use case of “Emergency Information on Mobile”**

## 3.2 Use Case Description

**UC-16:** Get list of all help places

The mobile application can get list of all help places in the database.

**Actor**

Mobile application

**Prerequisite**

The mobile application requests to get list of all help places.

**Input**

-

**Output**

System shows list of all help places in database in JSON form.

**Flow of Execution**

1. The mobile application requests to get list of all help places.
2. The system retrieves list of all help places from database.
3. The system shows list of all help places in form of JSON.

**Relate SRS:**

**SRS-25:** The system shall retrieve all help places from system database.

**SRS-64:** The system shall show list of all help places in form of JSON.

**UC-17:** Get nearest help place

The mobile application can get the nearest help place by the selected category.

**Actor**

Mobile application

**Prerequisite**

The mobile application requests to get nearest help place.

**Input**

The identity number of selected category.

Latitude of the mobile application’s current location.

Longitude of the mobile application’s current location.

**Output**

System shows nearest help place in JSON form.

**Flow of Execution**

1. The mobile application requests to get nearest help place.
2. The system finds the nearest help place by the selected category.
3. The system shows the nearest help place in form of JSON.

**Relate SRS:**

**SRS-65:** The system shall find the nearest help place by the selected category.

**SRS-66:** The system shall show the nearest help place in form of JSON.

**UC-18:** Get list of all help places in setting scope

The mobile application can get list of help places where locate in the setting scope.

**Actor**

Mobile application

**Prerequisite**

The mobile application requests to get list of help places where locate in the setting scope.

**Input**

The meter number of setting scope.

Latitude of the mobile application’s current location.

Longitude of the mobile application’s current location.

**Output**

System shows list of help places where locate in setting scope in JSON form.

**Flow of Execution**

1. The mobile application requests to get list of help places where locate in the setting scope.
2. The system retrieves list of help places where locate in setting scope from the database.
3. The system shows list of help places where locate in setting scope in form of JSON.

**Relate SRS:**

**SRS-67:** The system shall retrieve list of help places where locate in the setting scope from the database.

**SRS-68:** The system shall show list of help places where locate in setting scope in form of JSON

**UC-19:** Search help place’s name by keyword in online map

The user can search the help places by input a keyword.

**Actor**

User

**Prerequisite**

The user has to connect with the internet.

The user enters to the online map page, which shows the menu for searching.

**Input**

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Example** |
| keyword | a part of word use to input for searching help place | "a" , "1", "hospital", "Maharaj" |

**Output**

System shows list of result.

**Flow of Execution**

1. The user choose search menu.
2. The system shall provide search UI with text field.
3. The user inputs keyword into text field.
4. The system shall receive all help places from server.
5. The system shall search help places by keyword from user inputting.
6. The system shall matching keyword with help places.
7. The system shall display help places which matching with keyword.
8. The user selects help places list to view in the online map.
9. The system shall provide map with help place that user selection from searching by keyword.

**Relate SRS:**

**SRS-69:** The system shall provide search button UI.

**SRS-70:** The system shall provide text field UI.

**SRS-71:** The system shall receive all help places from server.

**SRS-72:** The system shall search help places by keyword from user inputting.

**SRS-73:** The system shall matching keyword with help places.

**SRS-74:** The system shall display help places which matching with keyword.

**SRS-75:** The system shall provide map with help place that user selection from searching by keyword.

**UC-20:** Find the nearest help place

The user can find the nearest help places by choosing the help place’s category.

**Actor**

User

**Prerequisite**

The user has to connect with the internet.

The user enters to the online map page, which shows the menu for finding nearest help place.

**Input**

Category’s id of help place.

**Output**

Position of the nearest help place on a map.

**Flow of Execution**

1. The system shall provide category button.
2. The user selects the category of help place such as, hospital, police station, highway police and garage.
3. The system shall receive the current location of user and category’s id.
4. The system shall send the current location of user and category’s id to the server.
5. The system shall receive help place object from the server.
6. The system shall change color of marker to show the position of nearest help place by searching.
7. The system shall display the position of help places on the online map.

**Relate SRS:**

**SRS-76:** The system shall provide category button.

**SRS-77:** The system shall receive the current location of user and category’s id.

**SRS-78:** The system shall send the current location of user and category’s id to the server.

**SRS-79:** The system shall receive help place object from the server.

**SRS-80:** The system shall change color of marker to show the position of nearest help place by searching.

**SRS-81:** The system shall display the nearest help place of each category from user selection

**UC-21:** Set the scope for downloading data

The user can define the scope and duration of storage.

**Actor**

User

**Prerequisite**

The user has to connect with the internet.

The user enters to the application.

**Input**

Number of scope in meter.

**Output**

Number of scope changed.

**Flow of Execution**

1. The system shall provide menu setting UI.
2. The user selects menu for setting scope of the storage.
3. The system shall provide number for setting scope with radio button UI.
4. The system shall define a default value of scope.
5. The user chooses the number of scope.
6. The system shall receive the scope of user’s setting.

**Relate SRS:**

**SRS-82:** The system shall provide menu setting UI.

**SRS-83:** The system shall provide number for setting scope with radio button UI.

**SRS-84:** The system shall define a default value of scope.

**UC-22:** Collect help place information automatically

The mobile application can collect help place information automatically when they move in length of setting scope from their old location

**Actor**

Mobile application

**Prerequisite**

The mobile application has to connect with the internet.

**Input**

List of help places from server.

**Output**

Help places that saved show in offline map

**Flow of Execution**

1. The mobile application has changed position more than length of setting scope
2. The system shall receive latitude and longitude when the user changes position.
3. The system shall check distance between original coordinates and new coordinates position automatically
4. The system shall send latitude, longitude and setting scope of user to the server.
5. The system shall receive help places from server.
6. The system shall delete the original data of help places from the database.
7. The system shall add new data of help places into database.

**Relate SRS:**

**SRS-85:** The system shall check distance between original coordinates and new coordinates position automatically.

**SRS-86:** The system shall send latitude and longitude of user to the server.

**SRS-87:** The system shall receive help places from server.

**SRS-88:** The system shall delete the latest data of help places from the database.

**SRS-89:** The system shall add new data of help places into database