Emergency Alert Mobile Application

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 02/03/2017 | 1.0 | First version | H.D.Witharana |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 5

1.1 Purpose 5

1.2 Scope 5

1.3 Definitions, Acronyms, and Abbreviations 5

1.4 References 6

1.5 Overview 6

2. Overall Description 6

2.1 Product Perspective 6

2.2 Product Functions 6

2.3 User Characteristics 7

2.4 Constraints 7

3. Specific Requirements 7

3.1 Functionality 7

3.1.1 User sets trusted parties. 7

3.1.2 User rates other users. 8

3.1.3 User calls a trusted party in a critical emergency 8

3.1.4 User informs a critical emergency to nearby users 8

3.1.5 User informs a critical emergency to nearby police 9

3.1.6 User informs a non-critical emergency to helper. 9

3.1.7 User will be notified about nearby dangerous places. 9

3.1.8 Helper registers the organization 9

3.2 Usability 10

3.2.1 User learning should be easy. 10

3.2.2 Simple and understandable GUIs 10

3.2.3 Minimum task time to select nearby users. 10

3.3 Reliability 10

3.3.1 Availability 10

3.3.2 Accuracy 10

3.4 Performance and Security 10

3.4.1 Minimum response time to select nearby users. 10

3.4.2 Minimum response time to inform non-critical emergency 10

3.4.3 Throughput 10

3.4.4 Capacity 11

3.4.5 Security in registering to the system 11

3.5 Supportability 11

3.5.1 Coding standards 11

3.5.2 Naming conventions 11

3.5.3 Class libraries 11

3.5.4 Application testability 11

3.6 Design Constraints 11

3.6.1 Standard development tools 11

3.6.2 Cloud computing approach 11

3.7 On-line User Documentation and Help System Requirements 11

3.8 Purchased Components 12

3.9 Interfaces 12

3.9.1 User Interfaces 12

3.9.2 Hardware Interfaces 12

3.9.3 Software Interfaces 12

3.9.4 Communications Interfaces 12

3.10 Database Requirements 12

3.11 Licensing, Legal, Copyright, and Other Notices 12

3.12 Applicable Standards 13

4. Supporting Information 13

# Introduction

This software requirements specification document captures the complete software requirements for the emergency alert mobile application. This SRS document provide a detailed description about functional and non-functional requirements with no use case modeling. This document is open for both developers and stakeholders.

## Purpose

This SRS fully describe the external behavior of the application. It also describes nonfunctional requirements, design constraints and other factors necessary to provide a complete and comprehensive description of the requirements for the emergency alert system.

## Scope

The emergency alert android application will notify trusted parties and nearby users about critical emergencies by sending the GPS location, time and two photos taken from front and back cameras of the phone. Users get a chance to select necessary help in a non-critical emergency. Helpers can register their company for use in an emergency. Users can rate other users by the service they provide in an emergency. More over system will notify nearby dangerous places to the users.

## Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Term | Description |
| Critical emergency | An emergency where help is needed within few seconds to prevent damages.  E.g. Murder, Robbery, Kidnapping |
| Non critical emergency | An emergency where help is not needed within few seconds.  E.g. Car broke down |
| User | Person who register to application and use it in an emergency. |
| Nearby users | People who have been registered to the system and be near to an emergency and have on GPS. |
| Trusted contact | Contact that is pre entered by a user to inform a critical emergency.  E.g. Mother |
| Helper | Person or organization registered in the system to provide services in a non-critical emergency situation. |

## References

[This subsection provides a complete list of all documents referenced elsewhere in the SRS. Identify each document by title, report number if applicable, date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]

## Overview

The remainder of this document has two chapters and appendixes. The second chapter provides an overall description about the general factors that affect the system functionality. Moreover, it introduces different types of stakeholders and their interaction with the system and it mentions the system constraints and assumptions about the product.

The third chapter provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences.

The fourth chapter deals with the table of contents, index and appendixes. Appendixes in the end of the document include all the results of the requirement prioritization and a Release plan based on them.

# Overall Description

This section of the SRS describes the general factors that affect the product and its requirements. This provides a background for those requirements, which are defined in detail in Section 3.

## Product Perspective

The system will be a mobile application which will be convenient to use to alert emergencies very quickly because most of the population use android phones for their day-to-day work. Moreover, the system will send notifications about nearby dangerous places to users so they can avoid such places.

Further users can rate other users from the service provided. By this information system will automatically blacklist unwanted users.

## Product Functions

* User can register to the system
* User can update information
* User can rate other users
* User can inform emergencies to trusted parties and other nearby users
* User can call nearby police
* Helper can register organization
* System notify dangerous nearby places
* System blacklist unwanted users
* System take photos of the emergency

## User Characteristics

There are two type of users in this system. There are user and helper. Each of them have their own requirements. User register them to seek help in an emergency whereas helper register to the system to help in an emergency.

User register to the system then inform critical emergencies to nearby users and trusted contacts. In non-critical emergencies user can inform helpers and their organization.

Helper register their organization in the system and whenever a user contacts the helper he may provide the service needed.

## Constraints

* Only the registered users can use the system.
* User must on GPS to send the location.
* Nearby users also should on the GPS to identify their location.
* Success rate depends on the number of users.
* Only srilankan users get the benefit from the system since the system only provide srilankan police information.

# Specific Requirements

This section of SRS contains all software requirements with all the details sufficient to developers to implement.

## Functionality

This section describes all the functional requirements of the system of different user types in detail.

**User type: User**

### User sets trusted parties.

|  |  |
| --- | --- |
| Description | User can pre enter trusted parties to the system to use in an emergency |
| Input | * Telephone numbers of trusted party * Contact names * Priority |
| Process | * First the user should fill contact name and telephone number of the trusted party * User should set the priority. |
| Output | A trusted contact is added. |
| Limitations | Only 5 trusted contacts can be added. |

### User rates other users.

|  |  |
| --- | --- |
| Description | User can rate other notified users for their support given in an emergency |
| Input | * Did notified users help? * Did they commit any harm? * Rate for the support |
| Process | * After a critical emergency system will send a form to rate other notified users. * User can rate them according to the support. * Then the system will analyze the ratings. |
| Output | * System will recognize good and cooperating users and mainly inform them about an emergency. * If they had commit any harm those users will be blacklisted and will not inform any emergency. Moreover, system will suggest calling police to the user to inform about the bad behavior of the notified user. |
| Limitations | Users must cooperate and give feedback to make the system useful. |

### User calls a trusted party in a critical emergency

|  |  |
| --- | --- |
| Description | User can press a button in a critical emergency to inform trusted party. |
| Input | * Press the button * GPS should be on |
| Process | * When there is a critical emergency system will first take two pictures from front and back camera. * Then system will find the location using GPS. * Send a message to all the trusted contacts with an alert message, two pictures and location * Then the system will call the first priority contact number of trusted contacts. If not answered, then call the other numbers according to their priority. |
| Output | An emergency messages and call to trusted contacts. |
| Limitations | If the phone does not have enough credit balance to take a call or to send a message this cannot be done. |

### User informs a critical emergency to nearby users

|  |  |
| --- | --- |
| Description | User can press a button in a critical emergency to inform nearby users. |
| Input | * Press the button * GPS should be on |
| Process | * When there is a critical emergency system will first take two pictures from front and back camera. * Then system will find the location using GPS. * Select trustworthy nearby users with their GPS and ratings. * Then the system will send emergency messages with time, two photos and location |
| Output | An emergency messages and call to nearby users. |
| Limitations | * If the phone does not have enough credit balance to take a call or to send a message this cannot be done. * There can be no nearby users at that time available. |

### User informs a critical emergency to nearby police

|  |  |
| --- | --- |
| Description | User can press a button in a critical emergency to inform nearby police. |
| Input | * Press the button * GPS should be on |
| Process | * System will find the location using GPS. * Then system will analyze the police records with the location and give the nearest police station’s number. * User can call to that number and convey the emergency. |
| Output | An emergency call to nearest police. |
| Limitations | If the phone does not have enough credit balance to take a call or to send a message this cannot be done. |

### User informs a non-critical emergency to helper.

|  |  |
| --- | --- |
| Description | User can enter details about the emergency and choose the most convenient helper nearby. |
| Input | * Press the button * GPS should be on |
| Process | * When there is a non-critical emergency user select a category from the given helps. * Then system will find the most appropriate helper by location, ratings and service. * User can contact the helper. |
| Output | An emergency call. |
| Limitations | If the phone does not have enough credit balance to take a call or to send a message this cannot be done. |

### User will be notified about nearby dangerous places.

|  |  |
| --- | --- |
| Description | When user turns on GPS system will notify nearby dangerous places. |
| Input | * GPS should be on |
| Process | * When user turns on GPS system will look through the database to find out any nearby place where a previous emergency occurred. * Then system will notify such places. |
| Output | Locations of nearby dangerous places shown in google maps |

**User type: Helper**

### Helper registers the organization

|  |  |
| --- | --- |
| Description | Helper can register their service in the system so that users can use in an emergency. |
| Input | * Contact name * Contact number * Address * Kind of services provided |
| Process | * Helper first have to register for the system. * If the service, he provides include in the category just select that if not add a new category. |
| Output | A new record |

## Usability

### User learning should be easy.

The required training time of this application for a normal user is about 5 to 8 minutes. Whereas a power user will require 3 minutes of training time to use the application productively.

### Simple and understandable GUIs

GUIs should be very simple to use in an emergency. These elements must be compatible with functional requirements so that users feel free to use the application without any inconvenience.

### Minimum task time to select nearby users.

When the user needs to convey the emergency to the other nearby users, it should be done as fast as possible in order to gain better prevention. Hence time to search nearby users is very critical and should be done in less than 2 seconds.

## Reliability

### Availability

100% availability should be provided to the system. But informing nearby users depend on whether users are using the application at that moment. This cannot be controlled by the system. Informing trusted contacts can be done in any time.

### Accuracy

This application is highly time critical. System should provide services in such a way accuracy is preserved even in a time critical domain.

## Performance and Security

### Minimum response time to select nearby users.

When the user needs to convey the emergency to the other nearby users, it should be done as fast as possible in order to gain better prevention. Hence time to search nearby users is very critical and should be done in less than 1 seconds.

### Minimum response time to inform non-critical emergency

When the user needs to convey non-critical emergency to helpers, it is not essential to be as fast as possible but must be accurate. Hence time to search nearby helpers should done in less than 1 minute.

### Throughput

Any number of users can use the application at same time. So, transactions per second should be high. System should try to handle all the needed transactions at a given time. Mostly 20 to 30 transactions in a critical emergency.

### Capacity

Any number of users can be added to the database. The success of the system mainly depends on the number of users.

### Security in registering to the system

Users registering to the system should give their own phone number. So a verification code should be sent to the number they provide in order to make sure the security.

## Supportability

### Coding standards

Entire coding should be done in a standard way so that it helps extendibility and maintainability. Always comments should be used to describe what the code does.

### Naming conventions

Naming conventions should be done in a meaningful way where the developers can easily understand the variables and methods which will improve the maintainability.

### Class libraries

Before using any class library all the limitations should be understood so that after sometime of using the library it won’t affect the systems functionality.

### Application testability

Test environment should test all the functions of the system in order to make sure the requirements are correctly implemented.

## Design Constraints

### Standard development tools

Mobile application should be built in a way that all sizes of phones and tablets must be able to handle the GUIs very conveniently.

### Cloud computing approach

Since the database cannot be saved inside a phone there must be a server to connect to the database. To perform this cloud architecture is used. In this case user, should first switch on data to connect to the server and not very applicable for time critical. Hence cloud computing is partially use for this system with slight modifications which will preserve the time critical functionality.

## On-line User Documentation and Help System Requirements

After registering to the system there should be a description about the functionalities of the components and where to use and what will be the output. This is essential to make the system user friendly and understandable.

## Purchased Components

Not applicable

## Interfaces

[This section defines the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, and the like, so that the software can be developed and verified against the interface requirements.]

**You do not have to include the screen shots of user interfaces itself ( as at this point those things are not implemented yet).**

**You may include (not all):**

**1. Explain the things that should be display in / consider for the interfaces**

**2. Describe**

**- For user interfaces : functionalities and the required menu items/ panels/ text boxes/ option buttons/ drop down lists that should be in the interfaces (eg. login page/ data entry page/ view pages/ analysis pages)**

**- For software interfaces: required interfaces to connect with the server, interfaces for access web services/ plugins**

**- For hardware interfaces: required client side pre-requisites( Disk space/ RAM/ Processor)**

**- For communication interfaces: eg. Asynchronous HTTP protocol requests over internet/ FTP file transfer**

**3. A draft diagram (block diagram) showing some main interfaces required by the user**

### User Interfaces

[Describe the user interfaces that are to be implemented by the software.]

### Hardware Interfaces

[This section defines any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, expected behavior, and so on.]

### Software Interfaces

* DB server – MongoDB and nodeJS

[This section describes software interfaces to other components of the software system. These may be purchased components, components reused from another application or components being developed for subsystems outside of the scope of this SRS but with which this software application must interact.]

### Communications Interfaces

[Describe any communications interfaces to other systems or devices such as local area networks, remote serial devices, and so forth.]

## Database Requirements

[Defines any database requirements for the system.]

## Licensing, Legal, Copyright, and Other Notices

[Defines any licensing enforcement requirements or other usage restriction requirements that are to be exhibited by the software.]

[This section describes any necessary legal disclaimers, warranties, copyright notices, patent notices, wordmark, trademark, or logo compliance issues for the software.]

## Applicable Standards

[This section describes by reference any applicable standard and the specific sections of any such standards which apply to the system being described. For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, and so forth.]

# Supporting Information

[The supporting information makes the SRS easier to use. It includes:

* Table of contents
* Index
* Appendices

These may include use-case storyboards or user-interface prototypes. When appendices are included, the SRS should explicitly state whether or not the appendices are to be considered part of the requirements.]

**Refer any data/ information in a standard format (eg. IEEE referencing style)**

**For different algorithms/ techniques/ theories you can refer text books.**

**For tools you can refer web pages.**

**For similar work you can refer research paper articles that describe the work.**

**You may include white paper articles for the description of technologies; web URL for the tool references. When you refer such a web page, you have to indicate the (Accessed on <<date>>)**