

Software Requirement Specification

for

Emergency Alert System

Prepared by

Abhishek Raymond

Abin Augustine

Visakh Vijay O

Department of Computer Science and Engineering

Rajiv Gandhi Institute of Technology, Kottayam

Contents

1	Introduction	4
1.1	Purpose	4
1.2	Document Conventions	4
1.3	Intended Audience and Reading Suggestions	4
1.4	Project Scope	4
1.5	References	5
2	Overall Description	6
2.1	Product Perspective	6
2.2	Product Features	6
2.3	User Classes and Characteristics	6
2.4	Operating Environment	7
2.5	Design and Implementation Constraints	7
2.6	User Documentation	7
2.7	Assumptions and Dependencies	8
3	System Features	9
3.1	Alert Button	9
3.1.1	Description and Priority	9
3.1.2	Stimulus/Response Sequences	9
3.1.3	Functional Requirements	9
3.2	Push notifications	9
3.2.1	Description and Priority	9
3.2.2	Stimulus/Response Sequences	9
3.2.3	Functional Requirements	9
3.3	User Profile	9
3.3.1	Description and Priority	9
3.3.2	Stimulus/Response Sequences	10
3.3.3	Functional Requirements	10
3.4	Map Integration	10
3.4.1	Description and Priority	10
3.4.2	Stimulus/Response Sequences	10
3.4.3	Functional Requirements	10
3.5	False alarm detection	10
3.5.1	Description and Priority	10
3.5.2	Stimulus/Response Sequences	10
3.5.3	Functional Requirements	10
3.6	Location-Based Notifications	11
3.6.1	Description and Priority	11
3.6.2	Stimulus/Response Sequences	11
3.6.3	Functional Requirements	11
3.7	Emergency contacts	11

3.7.1	Description and Priority	11
3.7.2	Stimulus/Response Sequences	11
3.7.3	Functional Requirements	11
4	External Interface Requirements	11
4.1	User Interfaces	11
4.2	Hardware Interfaces	12
4.3	Software Interfaces	12
4.4	Communications Interfaces	12
5	Other Nonfunctional Requirements	12
5.1	Performance Requirements	12
5.2	Safety Requirements	13
5.3	Security Requirements	13
5.4	Software Quality Attributes	13
6	Appendix A: Glossary	13
7	Appendix B: Issues List	14

1 Introduction

1.1 Purpose

This document outlines the functional and non-functional requirements of the project. It provides a comprehensive overview of the features and functionalities of the app, along with the intended audience, document conventions, project scope, and references.

1.2 Document Conventions

- Fonts: Font used in the document is Computer Modern, entire document follows this font. Font size used is 12pt.
- Section Headings: Section Headings are in larger font size of 17pt, bold, left-aligned and numbered.
- Sub Section Headings: Sub Section headings are slightly smaller at the size of 14pt, also bold, left-aligned and numbered.
- Subsub Section Headings: Sub Sub Section headings are smaller than sub section headings at the size of 12pt, also bold, left-aligned and numbered.
- Highlighting: Section headings and Sub Section Headings are highlighted with bold text.
- Numbering: Section are numbered using decimal numbers. Sub Section Headings are numbered using decimal points after section heading numbers.
- Lists: For ordered lists decimal numbers are used and font is intended.
For unordered lists bullet points are used and font is intended.

1.3 Intended Audience and Reading Suggestions

The Software Requirements Specification (SRS) document is meant for the project evaluation team, project supervisor, and project guide who are responsible for evaluating the team's work. To ensure that all project requirements have been addressed, it is recommended that the evaluation team carefully review this document. The SRS document contains a comprehensive overview of the project's scope, including functional and non-functional requirements, design and implementation constraints.

Technical terms and acronyms used throughout the document are defined in the Glossary (Appendix A). Additionally, the Issues List (Appendix B) provides a dynamic list of unresolved requirements issues, pending decisions, and conflicts that require resolution.

1.4 Project Scope

The software being specified is an emergency alert system app that notifies and alerts app users, emergency contacts, and authorities during emergency situations. The app aims to provide enhanced safety and security through a panic button feature and real-time location sharing. The objectives are to develop a user-friendly app with seamless communication, reliable location tracking, and minimize response time in emergencies. The project scope includes the development of a mobile app for Android platforms with features such as panic button, real-time location sharing, emergency contact management, user profile management, and integration with external APIs such as Google Map. The project does not include development of any external hardware, or softwares for any other device than android smartphones.

1.5 References

1. **GeoReach Article:**
<https://hypersense-software.com/projects/georeach-emergency-app>
2. **VithU Article:**
<https://www.novelwebcreation.com/blog/for-women-safety-v-channel-launched-vithu-mobile-app>
3. **BSafe Documentation:**
<http://ijcem.in/wp-content/uploads/2014/10/BSafe-BSecure.pdf>

2 Overall Description

2.1 Product Perspective

The emergency alert application is a new, self-contained product that is designed to replace existing emergency alert systems. The application uses technology to quickly notify nearby users, emergency contacts, and emergency services in real-time during an emergency. It sends notifications with location information and other relevant details using a panic button, making it easier for people to request for help they need when they need it most.

2.2 Product Features

The major features of the emergency alert application include:

- Personal profile creation with emergency information like contacts, blood type, and other pertinent medical details.
- Easily accessible and reliable panic button.
- Real-time location sharing to alert nearby users, emergency contacts, and emergency services.
- Google maps integration to view the current location of the person in emergency.
- Facilities to report false alarms and prevent the sending of notifications that may cause panic and unnecessary emergency responses.
- Security features to protect user's personal data and prevent unauthorized entry.
- Intuitive and user-friendly interface.
- Strong user authentication mechanism to ensure that only authorized users can access the system and send emergency notifications.
- Rate limiting mechanisms to restrict the number of requests that can be made from a single IP address in a given period of time.

2.3 User Classes and Characteristics

The various user classes that we anticipate using the emergency alert application include:

- General public: Anyone in the area who needs emergency assistance can use the system to send an alert and request help.
- Nearby Users: Individuals who are nearby the location of an personal emergency will receive an emergency notification.
- Family and friends of the person in distress: Emergency contacts can receive alerts from the system to notify them of an emergency
- Emergency responders: Police, fire, and medical personnel can use the system to receive alerts and coordinate their response to emergencies.
- Healthcare providers: Medical professionals, including hospitals and clinics, can use the system to receive medical information about patients in emergency situations.

2.4 Operating Environment

The emergency alert application operates on Android mobile devices with version 10.0 and above, utilizing an internet connection and GPS location services to function optimally. The application is designed to provide a reliable user experience, taking advantage of the latest Android OS features while ensuring the protection of user data with security measures.

2.5 Design and Implementation Constraints

1. Hardware Constraints:

- Software should be designed for smartphones with android 10.0 and above Operating System

2. Security Considerations:

- Data Privacy Regulations Compliance: The emergency alert application must comply with all relevant data privacy regulations to ensure the protection of user data and maintain the confidentiality of sensitive information.
- Reliability in Emergency Situations: The emergency alert application must be reliable and able to function in emergency situations, providing accurate and timely information to nearby users, emergency contacts and emergency services in times of crisis.

3. Technical Constraints:

- Front end of the application should be made for android application with flutter framework.
- Back end of the application should be made with node js with express.js framework.
- MongoDB will be used as database.
- The system should use RESTful API for client server communication.

4. Design constraints:

- Intuitive and User-Friendly Interface: The emergency alert application must provide an intuitive and user-friendly interface that is easy to navigate and understand, even for users with limited technical knowledge or experience.
- The software should follow android material UI.

2.6 User Documentation

User documentation components that will be delivered with the emergency alert application include:

- Application Demonstration: A detailed demonstration of the application's functionality will be provided to users. This demonstration will encompass all the essential features and functionalities of the application, including how to navigate through the application and how to configure and set up the emergency alert system.
- Frequently Asked Questions (FAQs): A section dedicated to addressing common questions and concerns that users may have about the application will be included. This section will cover a range of topics, including troubleshooting, system requirements, and how to use the application effectively.

- **User Manual:** A comprehensive user manual that will provide detailed information on the application's features and functionalities, as well as best practices for optimizing its performance.

2.7 Assumptions and Dependencies

Assumptions and dependencies for the emergency alert application project include:

1. Assumptions:

- **The availability of reliable internet and GPS location services:** The emergency alert application relies on the availability of internet and GPS location services to accurately and quickly send and receive notifications. Therefore, it is assumed that users have access to reliable internet and GPS location services in the area where the emergency alert application will be used.
- **The availability of mobile devices with compatible operating systems:** The emergency alert application is designed to be used on mobile devices, such as smartphones and tablets. Therefore, it is assumed that users have access to mobile devices with compatible android operating systems.
- **The cooperation and support of emergency responders and healthcare professionals:** The emergency alert application is intended to support emergency responders and healthcare professionals in providing timely and accurate assistance to users in emergency situations. Therefore, it is assumed that these professionals will be willing and able to use the application to access relevant information about the user in need of help.
- **The user's basic understanding of technology:** While the emergency alert application is designed to be intuitive and user-friendly, it is still assumed that users have a basic understanding of technology and can navigate the app with minimal guidance.
- **User's ability to give accurate information:** The emergency alert application relies on users to provide accurate information about their medical condition, location, and emergency contacts. Therefore, it is assumed that users will be able to provide this information to the best of their ability.
- **Availability of emergency services in the user's location:** The effectiveness of the emergency alert application is dependent on the availability of emergency services in the user's location. Therefore, it is assumed that users are in a location where emergency services are available and can respond to requests for assistance.
- **Compatibility with third-party apps and services:** The emergency alert application need to integrate with other third-party apps and services including google maps, it is assumed that the necessary integrations and compatibility issues will be addressed during the development process.

2. Dependencies:

- **Google Maps API:** The emergency response system is dependent on the Google Maps API to provide accurate location information for users and emergency responders.
- **SMS Services:** The system require integration with a third-party SMS service provider to ensure reliable delivery of emergency alerts via SMS.
- **Android API:** The system require integration with Android APIs to enable real-time location tracking and to receive push notifications.

3 System Features

3.1 Alert Button

3.1.1 Description and Priority

A big, easy-to-access button on the app's home screen that sends an emergency alert to nearby users, authorities, and emergency contacts when pressed.

- High Priority

3.1.2 Stimulus/Response Sequences

User pressing the alert button from notification bar or from inside the application triggers the emergency alert

3.1.3 Functional Requirements

- The Alert Button should be prominently displayed on the home screen of the app.
- The Alert Button should be easily accessible with a single tap.
- The Alert Button should trigger the emergency alert 5 seconds after being pressed.
- 5 second time is given to cancel the emergency alert
- alert button should also be shown as a permanent notification on lock screen for easy access.

3.2 Push notifications

3.2.1 Description and Priority

The app should use push notifications to notify users of emergency situations even when the app is not open.

- High Priority

3.2.2 Stimulus/Response Sequences

User pressing the alert button application triggers push notification in other users

3.2.3 Functional Requirements

- Push notifications should be shown in all situations, including in lockscreen.
- Push notification should have an alert sound.
- Notification should be easily noticeable and clear to view.

3.3 User Profile

3.3.1 Description and Priority

user profile includes emergency contact information, medical information, location

- High priority

3.3.2 Stimulus/Response Sequences

User will be creating an emergency profile upon registration

3.3.3 Functional Requirements

- User profile should contain emergency contact, emergency contacts will be notified when the user is pressing alert button.
- User profile should contain relevant medical information which may be useful in case of an emergency.
- User profile should be created at registration time and should be able to update/edit details after registration.
- Details in user profile should be sent upon pressing the emergency button.

3.4 Map Integration

3.4.1 Description and Priority

Google Map integration to view current location of the person in emergency

- Moderate priority

3.4.2 Stimulus/Response Sequences

Real time location will be sent along with emergency alerts.

3.4.3 Functional Requirements

- The app should use GPS and other location technologies to track the user's location.
- Location should be accurate as possible.
- Location should be realtime and should be updated in specified interval of time.

3.5 False alarm detection

3.5.1 Description and Priority

Report false alarms and prevent the sending of notifications that may cause panic and unnecessary emergency responses.

- Moderate priority

3.5.2 Stimulus/Response Sequences

False alarms can be flagged inside the application

3.5.3 Functional Requirements

- There should be an option to flag false alarms inside the application
- False alarms should be removed from alerts list
- False should not be allowed to cause panic

3.6 Location-Based Notifications

3.6.1 Description and Priority

The app should use the user's current location to send notifications to nearby users in the event of an emergency

- Moderate priority

3.6.2 Stimulus/Response Sequences

Notification is to be sent when the alert button is pressed

3.6.3 Functional Requirements

- Notification should contain realtime location of the user
- Relevant medical and user information should be in the notification
- Notification should not be delayed

3.7 Emergency contacts

3.7.1 Description and Priority

Integrate emergency contact information, such as phone number to receive notifications in the event of an emergency.

- Moderate priority

3.7.2 Stimulus/Response Sequences

Notification is sent to emergency contacts when the alert button is pressed

3.7.3 Functional Requirements

- User should be able to create emergency contacts
- Emergency contacts should have phone number and other details to help user

4 External Interface Requirements

4.1 User Interfaces

- Interface is in android material Design
- User Interface include:
 - Registration page: first time users will be able to register
 - Login page: registered users can login from login page
 - User profile page: after registration user will be able to create user profile with emergency contacts and relevant medical informations
 - Alert Button: Alert button inside the app will be used to send emergency alerts
 - Notification Button: Emergency alert button in lockscreen notification bar for easy access

4.2 Hardware Interfaces

- Supported Hardware: Android 10+
- Hardware with GPS and Wi-Fi support
- Wi-Fi and cellular information for location services
- Access the device's GPS for location services
- Cellular data access for SMS services and network connection

4.3 Software Interfaces

- Node.js 18.15.0 : As Backend
- npm 9.5.0
- MongoDB 6.0 : As Database
- Flutter 3.7.11
- RESTful API: For data transfer between client and server

4.4 Communications Interfaces

- Internet connection is required for client to server communication
- Data transfer between client and server will through HTTPS
- Use secure communication protocols, such as SSL or TLS, to protect user data and ensure privacy.
- SMS will be used to notify nearby users, authorities, and emergency contacts.
- The app will use the HTTP protocol for communication between the front end and back end components.
- RESTful API will be used for communication between the app and external APIs.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

1. Scalability: The emergency response system should be able to handle a large number of emergency alerts and users simultaneously without significant performance degradation.
2. Reliability: The emergency response system should be able to handle different types of network conditions, such as low bandwidth or intermittent connectivity. *in case of network unavailability, the system should be able to send notification whenever the device reconnects to the network*
3. Response Time: The emergency response system should be able to send emergency alerts within a specified time-frame to ensure timely response to emergencies.
4. Concurrency: System should be able to manage multiple alerts concurrently
5. Real-time updates: Location information should be updated in real-time

5.2 Safety Requirements

1. False Alerts: False alerts should be flagged to avoid unnecessary panic
2. Data Privacy: The app must abide by all applicable data privacy laws and rules, including the California Consumer Privacy Act (CCPA) and the General Data Protection Regulation (GDPR). User information must be safely kept in the database and transmitted using encryption.

5.3 Security Requirements

1. The emergency response system should use encryption to protect sensitive data, such as medical information and emergency contact information.
2. The emergency response system should be able to authenticate users and ensure that only authorized users can send emergency alerts.
3. The emergency response system should use secure communication protocols, such as SSL or TLS, to protect user data and ensure privacy.
4. Rate limiting mechanisms to restrict the number of requests that can be made from a single IP address in a given period of time, to prevent Denial of Service attacks

5.4 Software Quality Attributes

1. Scalability: This refers to the ability of the emergency response system to handle an increasing number of users, emergency alerts, and system components without significant degradation in performance. It includes aspects such as load balancing, horizontal and vertical scaling, and fault tolerance.
2. Portability: This refers to the ability of the emergency response system to run on different hardware devices without significant modifications.
3. Reliability: Should perform its intended functions accurately, consistently, and without errors or failures. It includes aspects such as fault tolerance, error recovery, and redundancy.
4. Security: Protect user data, prevent unauthorized access, and ensure the confidentiality, integrity, and availability of the system. It includes aspects such as encryption, secure communication protocols, access control, and auditing.
5. Testability: Should be able to test to ensure its correctness, reliability, and performance. It includes aspects such as test automation, testability of software components, and ease of setting up test environments.

6 Appendix A: Glossary

- User: End user of the application, user will be able to register and create emergency profile, and sent emergency notifications
- Client: Client software will be an android applications, that will be used by the end user to send notifications
- Server: Server will be the backend of the system, It will handle emergency alerts coming from clients and will send it to other users

- Panic Button: Panic button is an emergency alert button, that will be used to send notifications
- SSL(Secure Docket Layer): SSL will be used to transfer data securely
- API: Application Programming Interface
- HTTPS: Hyper Text Transfer Protocol (Secure)

7 Appendix B: Issues List

- Number of users who can concurrently able to use the application is not specified.
- Using android gestures to trigger the emergency alert is not implementing.