

# Software Requirements Specification (SRS) Format

For SY B.Tech (IT) – Community Engagement Project

## 1. Title Page

- Project Title : Emergency Service Platform.
- Course: Community Engagement Project (CEP)
- Student Name(s) & Roll No(s)

Names	Roll No.'s
1. Sanika Potdar	SI126
2. Pranav Chavan	SI138
3. Sonal Karche	SI141
4. Harshwardhan Khaire	SI145

- PRN Number(s)
- 

Names	PRN Number's
1. Sanika Potdar	B24IT1026
2. Pranav Chavan	B24IT1038
3. Sonal Karche	B24IT1041
4. Harshwardhan Khaire	B24IT1045

- Class : IT

Division : A

Batch : B

- Faculty Guide Name : Dr. Prakash R. Gadekar
- Date of Submission :

## **2. Abstract:**

The *Emergency Service Platform* is a community-focused solution designed to provide quick, reliable, and accessible emergency assistance. By integrating web and mobile technologies, it ensures that community members can request emergency services such as ambulance, police, or fire department support in real time. The platform reduces response time, improves communication, and ensures transparency in emergency handling,

## **3. Table of Contents:**

1. Title Page
2. Abstract
3. Table of Contents
4. Introduction
  - 4.1 Purpose
  - 4.2 Scope
  - 4.3 Definitions, Acronyms, and Abbreviations
  - 4.4 References
5. Overall Description
  - 5.1 Product Perspective
  - 5.2 Product Functions
  - 5.3 User Classes and Characteristics
  - 5.4 Operating Environment
  - 5.5 Design and Implementation Constraints
  - 5.6 Assumptions and Dependencies
6. Specific Requirements
  - 6.1 Functional Requirements
  - 6.2 Non-Functional Requirements
7. External Interface Requirements
  - 7.1 User Interfaces
  - 7.2 Hardware Interfaces

7.3 Software Interfaces  
7.4 Communication Interfaces

8. Use Case Diagram & Description
9. Activity Diagram / Flow Chart
10. References

## **4. Introduction**

### **4.1 Purpose**

The purpose of the *Emergency Service Platform* is to provide a centralized, digital solution that connects community members with essential emergency services. It addresses the delay and limitations of traditional phone-based emergency systems by offering a faster, user-friendly alternative.

This project benefits the community by saving time during critical situations, ensuring accurate location tracking, and enabling a structured response system.

### **4.2 Scope**

- The system will allow users to request emergency services via web or mobile applications.
- It will integrate with service providers (police, hospitals, fire departments).
- It will provide real-time notifications and track response progress.

#### **Stakeholders:**

- Community members (end users)
- Emergency service providers (ambulance, police, fire department)
- NGOs and volunteers
- Government authorities

#### **Community Benefits:**

- Faster emergency response
- Reduced panic through reliable communication
- Improved safety and trust in technology-driven solutions

#### 4.3 Definitions, Acronyms, and Abbreviations

- **ESP** – Emergency Service Platform
- **NGO** – Non-Governmental Organization
- **UI** – User Interface
- **API** – Application Programming Interface

#### 4.4 References

- Gómez, B., Juiz, C. *Integration of Emergency Web App for Accessing the Emergency Services by Mobile Phones*, IEEE, 2013.
- Community interaction survey reports.
- Official emergency helpline documentation.

## 5. Overall Description

### 5.1 Product Perspective

The system is a **new platform**, not an enhancement of an existing one. It combines mobile/web interfaces with backend server integration for real-time emergency handling.

(Block diagram would go here – showing Users → ESP → Emergency Service Providers)

### 5.2 Product Functions

- User registration & login
- Emergency request submission

- Location tracking
- Notification to emergency services
- Status updates and feedback system

### 5.3 User Classes and Characteristics

- **Community Members:** Limited technical knowledge, need simple UI
- **Service Providers:** Professionals requiring fast notifications
- **NGOs/Volunteers:** Assist with coordination
- **Faculty/Students:** For development & maintenance support

### 5.4 Operating Environment

- **Hardware:** Smartphones, desktops, internet-enabled devices
- **Software:** Web app (HTML/CSS/JS), Mobile app (Android/iOS), Database (MySQL/PostgreSQL), Cloud hosting
- **Network:** Internet/Wi-Fi/4G/5G

### 5.5 Design and Implementation Constraints

- Limited development time (academic project)
- Budget constraints for hosting
- Limited community awareness & digital literacy

### 5.6 Assumptions and Dependencies

- Users have access to smartphones/internet
- Emergency services are willing to cooperate

- Reliable network availability
- 

## 6. Specific Requirements

### 6.1 Functional Requirements

- **FR1:** The system shall allow users to register and log in.
- **FR2:** The system shall enable users to request emergency help.
- **FR3:** The system shall capture and share user location in real time.
- **FR4:** The system shall notify respective emergency departments.
- **FR5:** The system shall allow service providers to update status (on the way, resolved).
- **FR6:** The system shall maintain user and request records.

### 6.2 Non-Functional Requirements

- **Performance:** Requests processed within 2 seconds.
  - **Security:** Data encryption, authentication required.
  - **Usability:** Multilingual support, simple UI.
  - **Reliability:** 99% uptime expected during emergencies.
- 

## 7. External Interface Requirements

### 7.1 User Interfaces

- Mobile app interface with emergency buttons (Ambulance, Police, Fire).
- Web dashboard for service providers.

## 7.2 Hardware Interfaces

- GPS-enabled smartphones
- Desktop browsers

## 7.3 Software Interfaces

- Google Maps API (for location tracking)
- SMS/Email API (for notifications)

## 7.4 Communication Interfaces

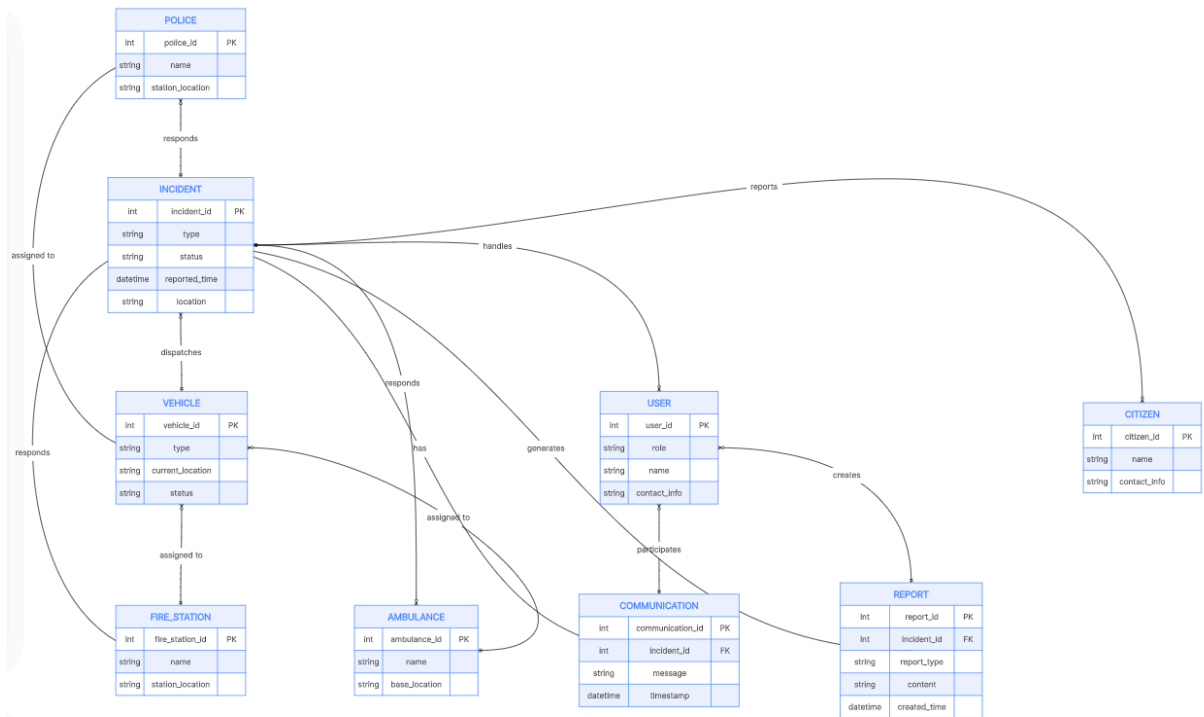
- Internet (Wi-Fi, 4G/5G)
  - SMS fallback in case of no internet
- 

# 8. Use Case Diagram & Description

(UML diagram – Users request services, System processes, Service providers respond)

### Use Cases:

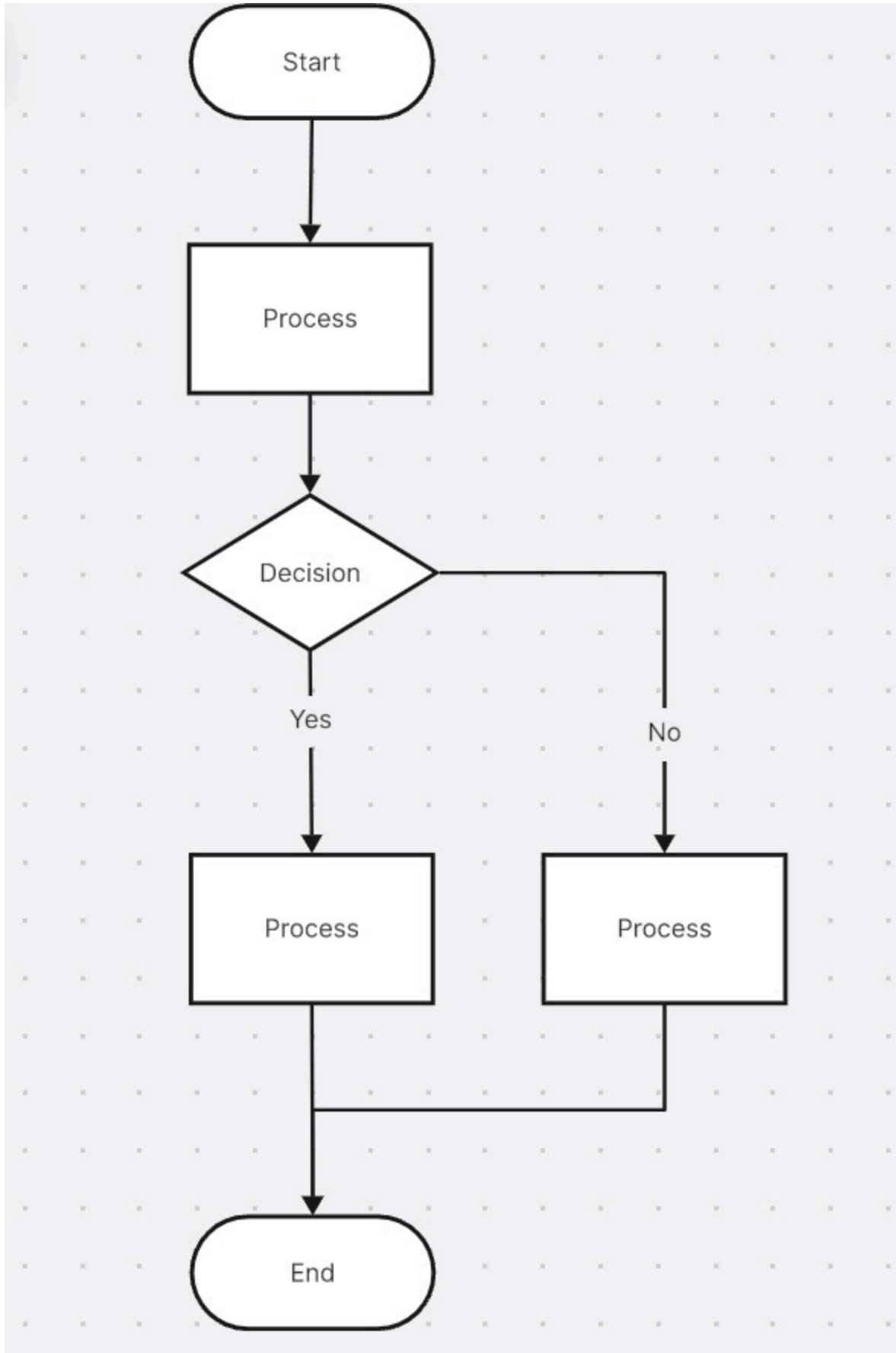
- UC1: User raises an emergency request
- UC2: System sends location and details to provider
- UC3: Provider updates response status
- UC4: User receives confirmation



## 9. Activity Diagram / Flow Chart

- Step 1: User opens app →
- Step 2: Selects emergency type →
- Step 3: System captures location →
- Step 4: Sends alert to provider →
- Step 5: Provider responds →
- Step 6: User receives updates





---

## 10. References

- IEEE paper: *Integration of Emergency Web App for Accessing the Emergency Services by Mobile Phones* (2013).
- Community engagement survey results.
- Discussions with NGOs and local authorities.
- Photos & reports from field visits.