B.M.S. COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU

OOMD Mini Project Report

Women Safety Analytics

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

Mechineni Anjutharuni (1BM23CS187)

Megha C H (1BM23CS188)

Meghashree S Karadi (1BM23CS189)

Department of Computer Science and Engineering
B.M.S. College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
2025-26

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, Mechineni Anjutharuni(1BM23CS187), Megha C H(1BM23CS188), Meghashree S Karadi(1BM23CS189) students of 5th Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this OOMD Mini Project entitled "WOMEN SAFETY ANALYTICS" has been carried out in Department of CSE, B.M.S. College of Engineering, Bangalore during the academic semester August 2025- December 2025. I also declare that to the best of our knowledge and belief, the OOMD mini Project report is not from part of any other report by any other students.

Signature of the Candidate

Mechineni Anjutharuni (1BM23CS187)

Megha C H (1BM23CS188)

Meghashree S Karadi (1BM23CS189)

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the OOMD Mini Project titled “**WOMEN SAFETY ANALYTICS”** has been carried out by Mechineni Anjutharuni(1BM23CS187), Megha C H(1BM23CS188), Meghashree S Karadi(1BM23CS189) during the academic year 2025-2026.

Signature of the Faculty in Charge

Table of Contents

Chapter 1: Problem Statement

Problem Statement

In today's society, ensuring the safety of women is a critical concern. Existing safety mechanisms often rely on reactive measures or fragmented systems. The core challenge is the lack of a unified, proactive, data-driven system that can analyze real-time and historical data (crime statistics, infrastructure status, user reports) to identify high-risk areas, predict potential threats, and provide actionable safety recommendations. Specifically, there is an absence of an intelligent platform that can offer timely, localized alerts, guide a user to the nearest safe haven when feeling unsafe, and instantly alert police with precise location data during an emergency.

Chapter 2: Software Requirement Specification

Software Requirements Specification (SRS) Document

Project Title: Women Safety Analytics System

1. Introduction

1.1 Purpose of this Document

The purpose of this SRS document is to define the detailed, specific, and measurable requirements for the Women Safety Analytics System. It outlines the system's objectives, functional and non-functional requirements, interfaces, and constraints. This document serves as the formal blueprint for the development team and the basis for project validation and testing.

1.2 Scope of this Document

The Women Safety Analytics System will focus on data aggregation, machine learning-based risk analysis, geospatial visualization (Safety Heat-map), and real-time intervention. The system will include a mobile application for end-users, offering safe route planning, location-based risk alerts, and a powerful SOS/Emergency feature that automatically recommends the nearest safe haven and communicates the user's location and status directly to local authorities.

1.3 Overview

The Women Safety Analytics System is an intelligent, data-driven platform designed to move safety from reactive response to proactive prevention. It consists of the Data Aggregation Engine, the Risk Assessment Module (the analytical core), the Geospatial Visualization Layer, and the User Safety Module (including route planner, alerts, and SOS).

2. General Description

The system is primarily intended for general public users (primarily women) who need real-time safety information and immediate help, and for city/police administrators who require macro-level safety data for resource allocation and planning. The system will operate 24/7, providing up-to-date risk scores and ensuring rapid emergency response capabilities. The central objective is to leverage urban data to significantly reduce risks to personal safety.

3. Functional Requirements

1. **User Management:** Secure user registration, authentication, and personalized profile management (including emergency contact setup).
2. **Data Ingestion and Pre-processing:** Automated collection, cleaning, and normalization of data from external sources (e.g., crime APIs, transit data, street lighting status).
3. **Risk Assessment and Scoring:** Dynamic calculation of a quantitative Safety Score for every grid cell on the map, updated in real-time based on incident reports, time of day, and environmental factors.

4. **Geospatial Visualization:** Display of an interactive Safety Heat-map to visually represent the real-time risk levels across the covered area.
5. **Route Safety Planner:** Users can input start/end points, and the system will recommend the Safest Route based on the current Safety Scores, prioritizing safety over distance/speed.
6. **Real-Time Risk Alerts:** Send push notifications to the user's device when they enter a pre-defined or dynamically identified high-risk area.
7. **Nearest Safe Haven Recommendation:** On user command (e.g., "I feel unsafe"), the system must identify the absolute nearest official safe zone (e.g., police station, hospital) or pre-vetted busy, public place and provide immediate turn-by-turn navigation guidance.
8. **Emergency (SOS) Activation: A single-click feature must initiate a simultaneous alert to:**
    - Pre-configured private contacts.
    - The appropriate local police/emergency service line.
9. **Automated Police Communication:** Upon SOS activation, the system must automatically and immediately transmit the user's live GPS coordinates, a customizable emergency message (e.g., "SOS: I need help at my current location"), and user identity to the designated police communication channel (e.g., API, secure server, or SMS gateway).
10. **Secure User Reporting:** Allow users to securely report non-emergency safety concerns (e.g., harassment, non-functional streetlights) with location and optional photo evidence.
11. **Admin/Police Dashboard:** Provide authorized personnel with tools for viewing safety trends, aggregating user reports, managing safe zone lists, and tracking active SOS alerts.

4. Interface Requirements

- **User Interface (Mobile):** Intuitive, high-contrast GUI for Android and iOS with a prominent, accessible SOS button and easy-to-read map visualization.
- **Hardware Interface:** Requires GPS, high-speed Internet connectivity, and accelerometer/gyroscope access on the mobile device for accurate location and movement tracking.
- **Software Interface: APIs** for integrating external data sources (e.g., government open data portals, Google Maps/OpenStreetMap), and a defined API for Police/Emergency Services integration (e.g., REST API or established messaging protocol).
- **Database:** A robust, high-performance NoSQL (e.g., MongoDB) or relational (e.g., PostgreSQL) database for handling large volumes of geospatial data.

5. Performance Requirements

1. **Response Time:** The system must identify the safest route or the nearest safe area within 2 seconds of receiving the request.
2. **SOS Latency**: The total time from SOS button press to successful transmission of location and message to the police must not exceed 2 seconds (excluding network transmission time).

3. **Scalability:** Must support a minimum of 1,000 concurrent users accessing route planning and visualization features during peak hours.
4. **Data Handling:** The system must efficiently process and store a minimum of 1 million geospatial risk data points daily.

6. Design Constraints

1. **Technology Stack:** The backend must utilize a data science-friendly language (e.g., Python with libraries like Pandas and Scikit-learn for the ML engine). The mobile application should use native or cross-platform frameworks (e.g., Flutter/React Native).
2. **Data Privacy:** All data handling must comply with the strictest applicable data privacy regulations (e.g., GDPR principles), emphasizing user data anonymization and encryption.
3. **Regulatory Compliance:** Integration with emergency services (police) must adhere strictly to local government protocols and communication standards for official emergency messaging.
4. **Availability:** The service architecture must be designed for cloud deployment (e.g., AWS, Azure) to ensure high availability and geo-redundancy.

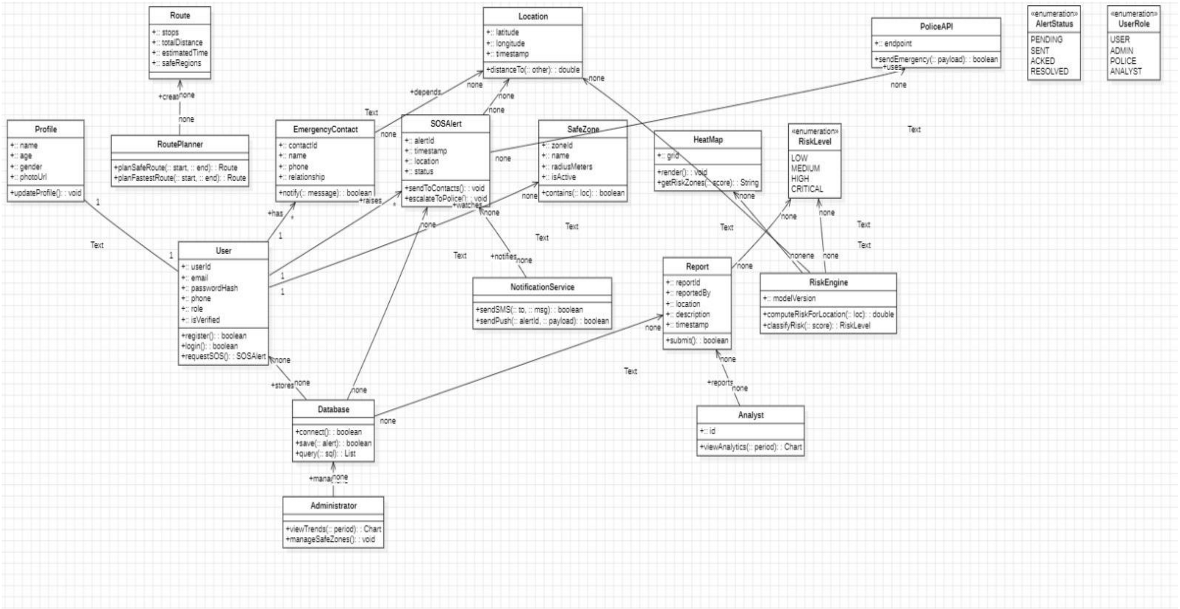
7. Non-Functional Attributes

1. **Security:** Implement OAuth 2.0 for authentication, TLS/SSL for all data transmission, and AES-256 encryption for stored location and personal data.
2. **Reliability:** The core risk engine and SOS component must achieve a minimum 99.99% uptime due to the mission-critical nature of the safety features.
3. **Scalability:** The architecture must employ microservices or decoupled components to allow easy expansion to new geographical regions and integration of new data sources.
4. **Usability:** The mobile UI must be highly intuitive, accessible, and designed for ease of use under stress (e.g., large, contrasting buttons for the SOS feature).
5. **Maintainability:** The entire codebase must be modular, thoroughly documented, and follow industry best practices for version control and testing.

8. Preliminary Schedule and Budget

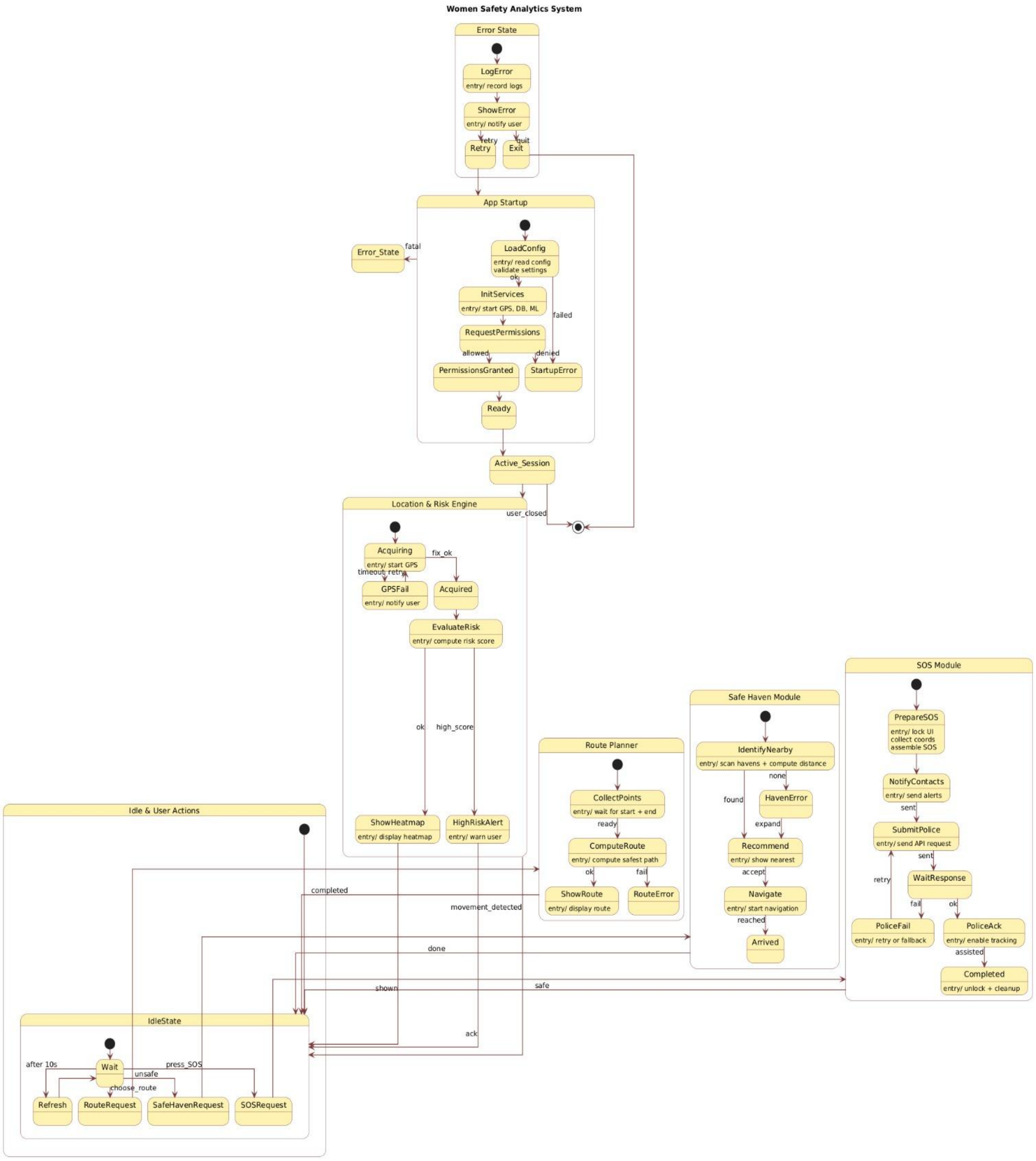
Phase	Duration (Months)
Phase 1: Design & Core ML (Requirements, Architecture, Risk Model Training)	2
Phase 2: Mobile & Backend Development (APIs, UI/UX, Safe Haven Feature)	3
Phase 3: Integration & Testing (Police API Integration, Field Testing, Bug Fixes)	1.5
Phase 4: Deployment & Documentation	0.5
Total	**7 Month**

Chapter 3: Class Modeling



The Report class is the system’s core incident-record entity, created by a user whenever they raise an event. It stores a unique reportId, the reportType, a textual description, a timeStamp, and a composed Location object that captures latitude, longitude, and spatial context. Each report is tightly associated with exactly one user, while a user can generate multiple reports over time. Once created, the report is persisted through the submit() method, which interacts with the Database for storage. A report’s location and details may be used by the RiskEngine to compute local risk scores, influence safe-route planning, and support heatmap generation. The report can also lead to emergency escalation: its information can feed into an SOSAlert, which may call the PoliceAPI for emergency dispatch. In the wider system, reports serve as key input for analysts, dashboards, safety zone monitoring, and long-term data analysis, making them central to risk evaluation, user safety workflows, and incident tracking.

Chapter 4: State Modeling



1. App Startup State Machine

This model describes how the application initializes and prepares itself before becoming usable. When the app is launched, it loads configuration files, initializes essential services such as GPS and database modules, and requests user permissions. If permissions are granted and services are successfully initialized, the app transitions into the Ready state. If any error occurs—such as failed configuration load, permission denial, or hardware issues—the system enters an Error_State, where it displays an error message and prompts the user to retry or exit. Once all conditions are satisfied, the system enters the Active_Session state, which begins the main operational workflow.

2. Location & Risk Engine State Machine

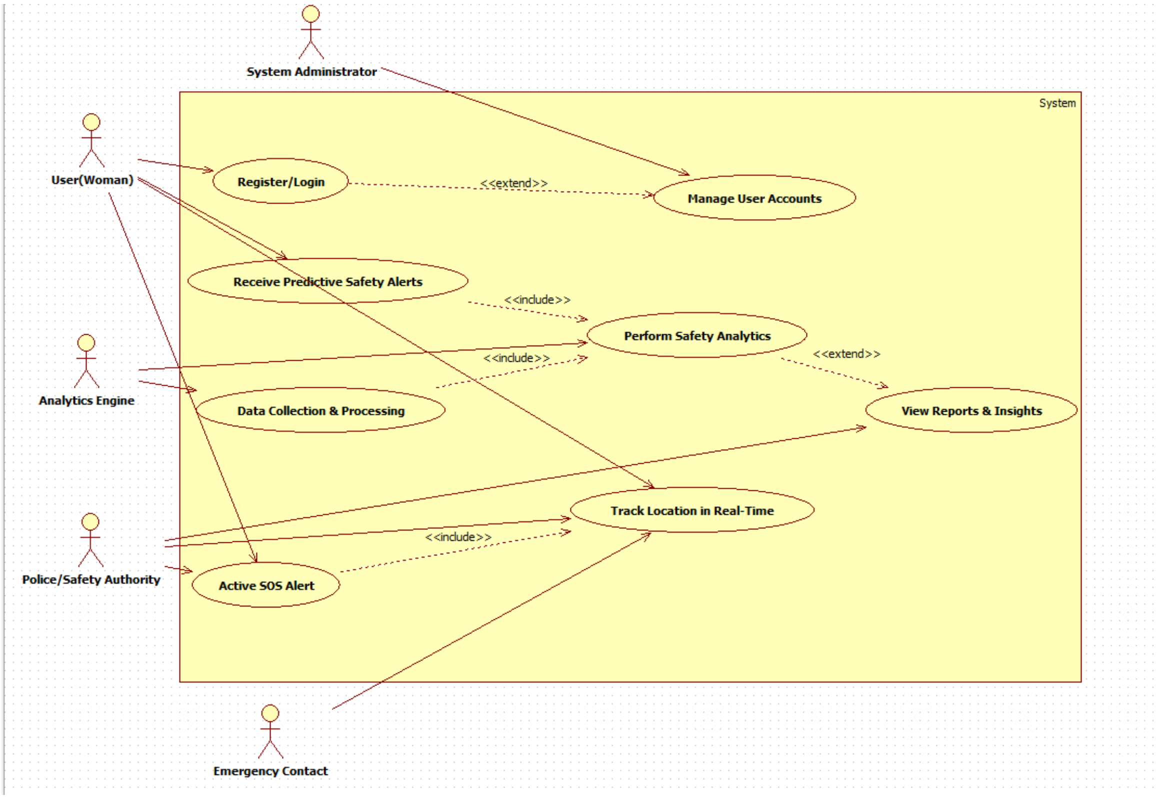
This model controls how the app handles continuous location tracking and safety risk analysis during operation. The system begins in an Acquiring state, where it attempts to obtain GPS signals. If GPS acquisition is successful, it enters the Acquired state and proceeds to the EvaluateRisk state, where it computes a safety or risk score based on crime data and surrounding infrastructure. If the risk is normal, the system simply updates the user interface and continues monitoring. If a High Risk score is detected, it triggers a HighRiskAlert, notifying the user and redirecting them toward safe routes or safe havens. The Location & Risk Engine interacts directly with user actions (movement, input requests) and can revert to idle states when the user pauses movement or closes the session.

3. SOS & Safety Assistance State Machine

The third model represents the emergency-handling workflow. When the user activates SOS, the system enters the PrepareSOS state, where it collects the current location and prepares emergency data. It then transitions to the NotifyContacts state, sending alerts to pre-configured emergency contacts. Simultaneously or subsequently, it moves to SubmitPolice, where it sends the SOS message and location details to the nearest police server. The WaitResponse state monitors for acknowledgment from the authorities. If a response is received, the system enters the PolicelAck state and provides confirmation to the user. If not, it enters a fallback mode, retriggering SOS or attempting alternative communication methods. After completion, the system transitions to the Completed state, where cleanup tasks are performed before returning to the main session.

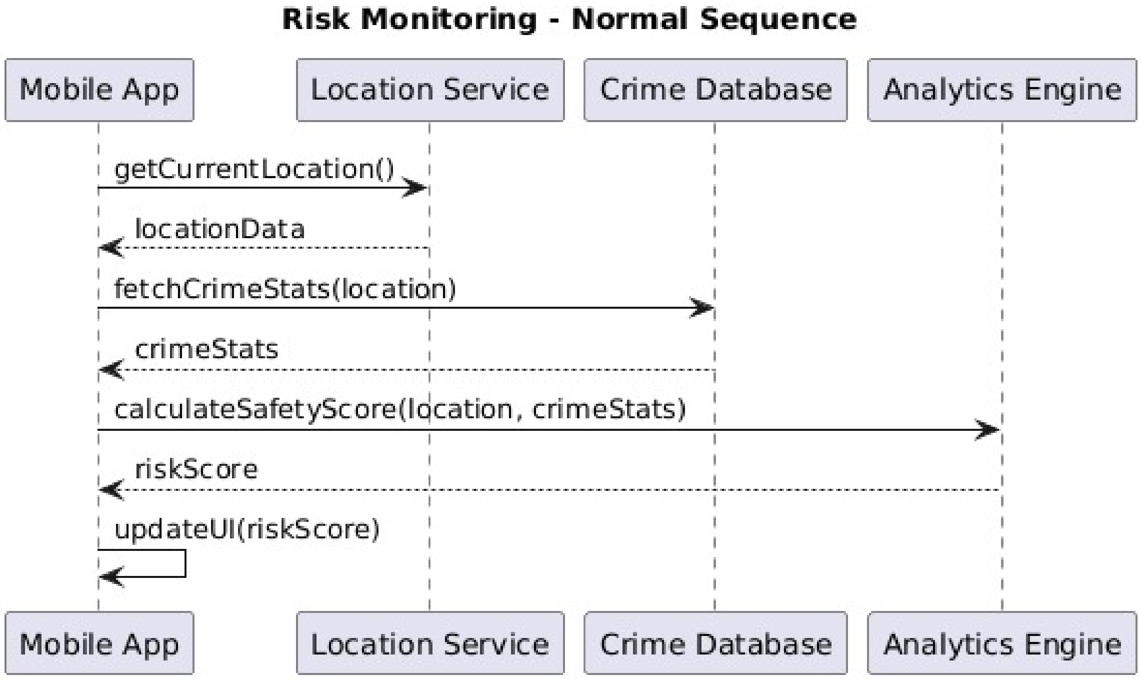
Chapter 5: Interaction Modeling

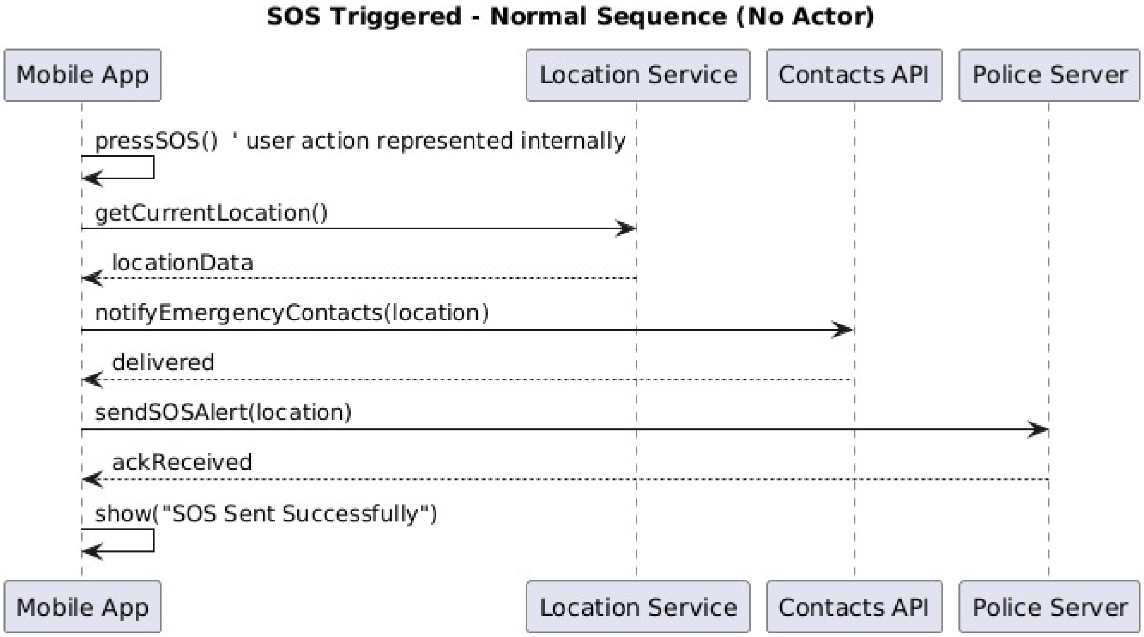
Use Case Model



The use case diagram illustrates the major interactions within the Women Safety Analytics System by highlighting the roles of different actors and their associated functionalities. The primary user, a woman, can register or log in to the system, receive predictive safety alerts, and have her location tracked in real time for enhanced safety. The system administrator manages user accounts and oversees system operations. The analytics engine interacts with the system through the *Data Collection & Processing* and *Perform Safety Analytics* use cases, which feed safety insights and predictive alerts back to the user. The police or safety authority actor participates in the *Active SOS Alert* use case, enabling rapid response during emergencies. Additionally, emergency contacts are notified during SOS events. The system also supports extended features such as viewing reports and insights generated from analytical processes. Together, these use cases show how data, analytics, tracking, and emergency response work collaboratively to provide a comprehensive safety solution.

Sequence Diagrams





Risk Monitoring

Normal Scenario

The mobile app gets the current location, fetches nearby crime statistics, sends both to the analytics engine, receives a risk score, and updates the UI.

<<include>> Use Cases

- **Fetch Crime Stats**
- **Calculate Safety Score**

<<extend>> Exception Scenarios

- **Location Error:** GPS unavailable → show error / retry.
- **Crime Data Fetch Failure:** Server unreachable → show message / use cached data.
- **Analytics Error:** Score calculation fails → use fallback method.

SOS Triggered

Normal Scenario

When the user presses SOS, the app gets the location, sends it to emergency contacts, sends an alert to police, receives acknowledgment, and shows that the SOS was sent successfully.

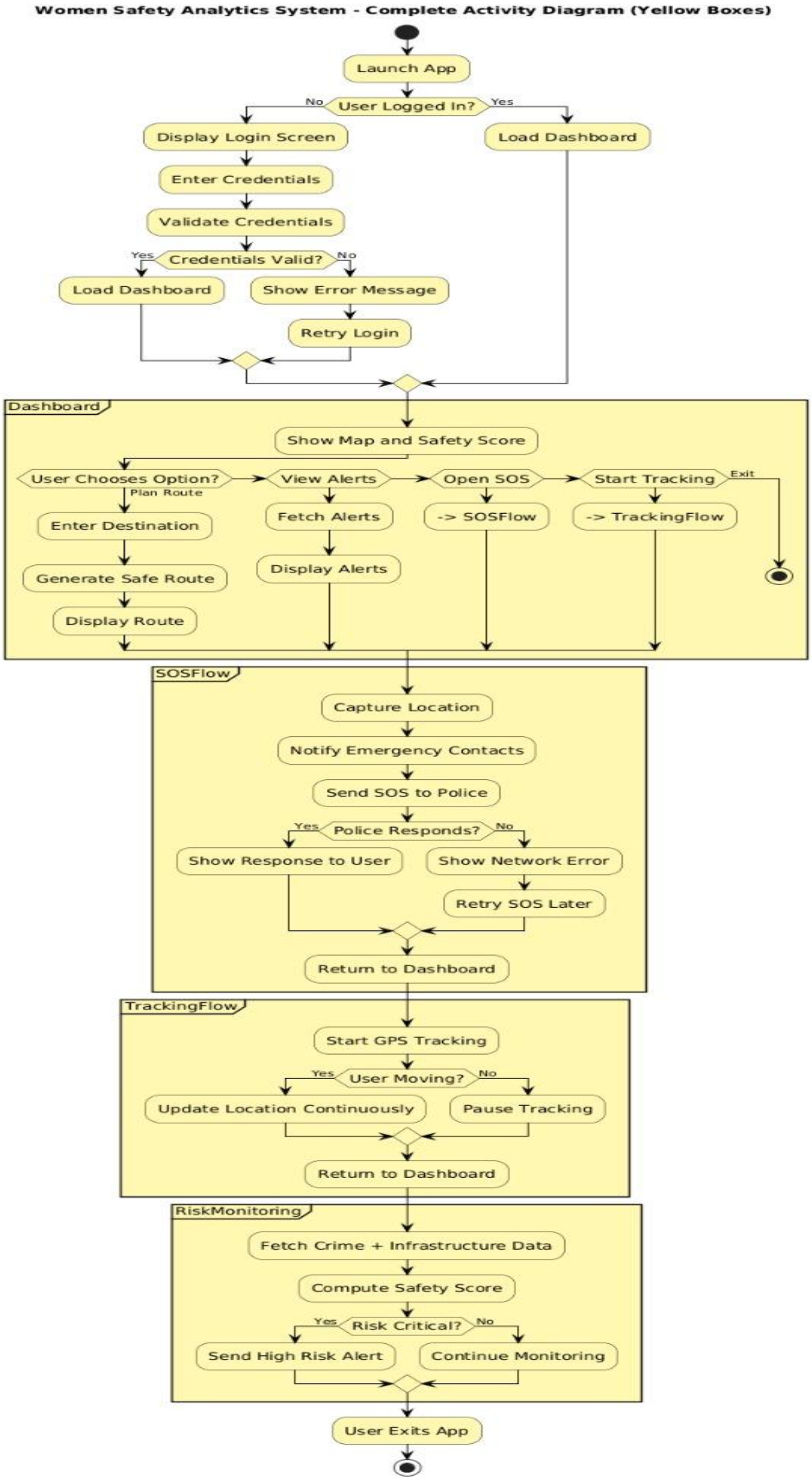
<<include>> Use Cases

- **Get Current Location**
- **Notify Emergency Contacts**
- **Send SOS Alert to Police**

<<extend>> Exception Scenarios

- **GPS Failure:** Use last known location.
- **Contacts Delivery Failure:** Retry sending alert.
- **Police Server Not Responding:** Show network error → retry later.
- **No Internet:** Send SMS only, queue police alert.

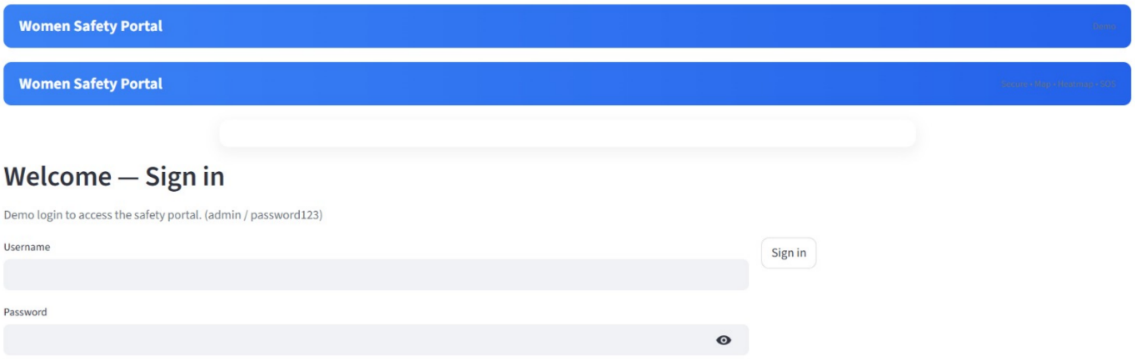
Activity Diagram



The activity diagram illustrates the complete workflow of the Women Safety Analytics System, beginning from the moment the user launches the mobile application. If the user is not logged in, the system displays the login screen, validates credentials, and loads the dashboard upon success. Once the dashboard is loaded, the user can view the safety map and choose between several options such as planning a safe route, viewing alerts, activating SOS service, or starting GPS tracking. When planning a route, the user enters a destination, and the system generates and displays the safest possible path. The SOS flow captures the user's location, notifies emergency contacts, and sends an alert to the police; it also handles cases where the police fail to respond by showing a network error and retrying later. The tracking flow continuously updates the user's GPS location when movement is detected, otherwise it pauses tracking. Additionally, the system performs background risk monitoring by fetching crime and infrastructure data, computing a safety score, and issuing high-risk alerts if danger is detected. Finally, the process ends when the user exits the application. Overall, this activity diagram represents the complete operational behavior of the safety system, covering login, dashboard interactions, routing, SOS handling, tracking, and real-time risk evaluation.

Chapter 6: UI Design with Screenshots

Login Page:



Women Safety Portal Demo

Women Safety Portal Secure • Map • Heatmap • SOS

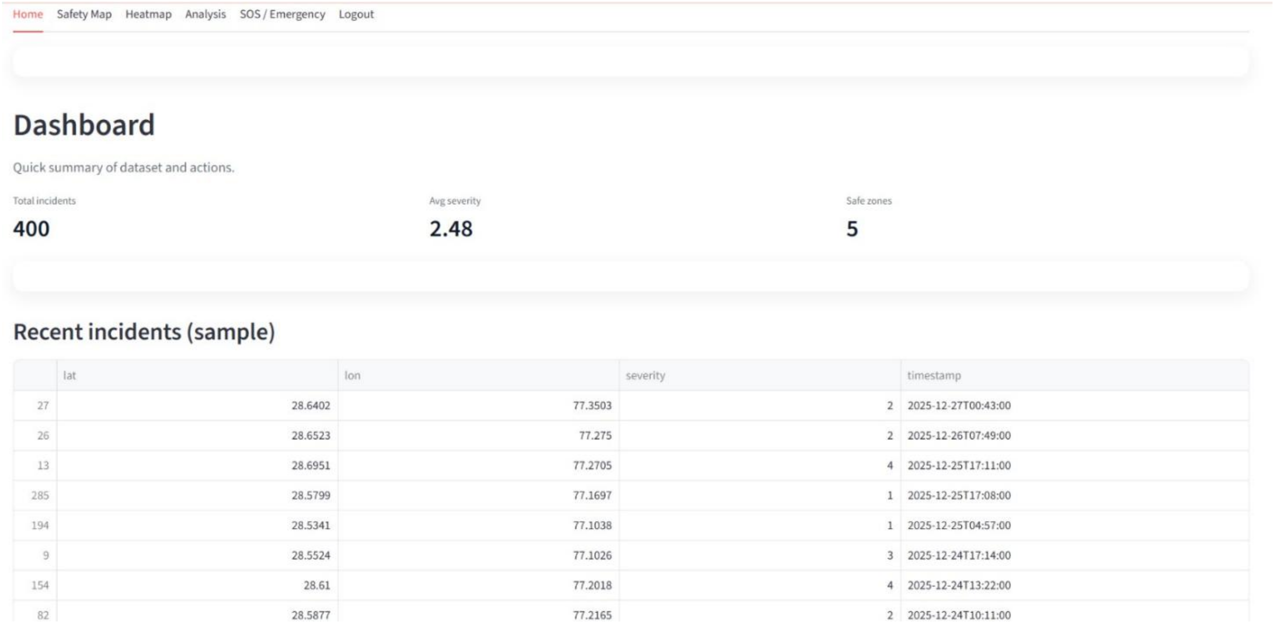
Welcome — Sign in

Demo login to access the safety portal. (admin / password123)

Username

Password

Home page



Dashboard

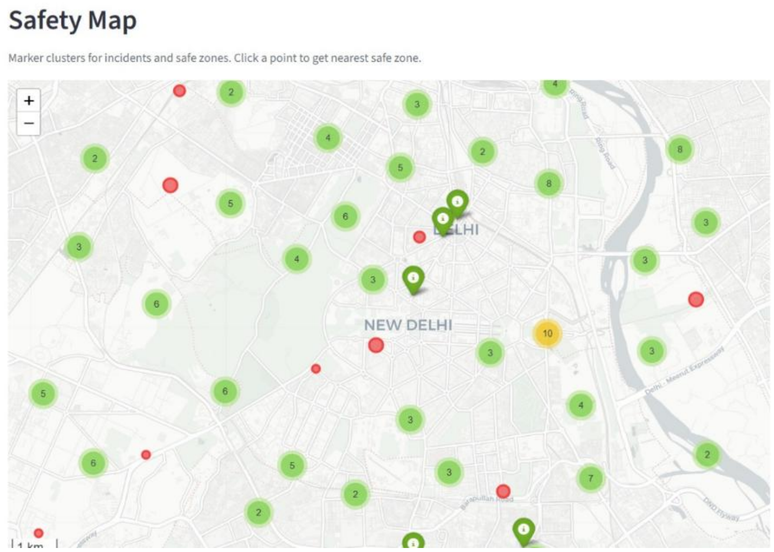
Quick summary of dataset and actions.

Total incidents	Avg severity	Safe zones
400	2.48	5

Recent incidents (sample)

	lat	lon	severity	timestamp
27	28.6402	77.3503	2	2025-12-27T00:43:00
26	28.6523	77.275	2	2025-12-26T07:49:00
13	28.6951	77.2705	4	2025-12-25T17:11:00
285	28.5799	77.1697	1	2025-12-25T17:08:00
194	28.5341	77.1038	1	2025-12-25T04:57:00
9	28.5524	77.1026	3	2025-12-24T17:14:00
154	28.61	77.2018	4	2025-12-24T13:22:00
82	28.5877	77.2165	2	2025-12-24T10:11:00

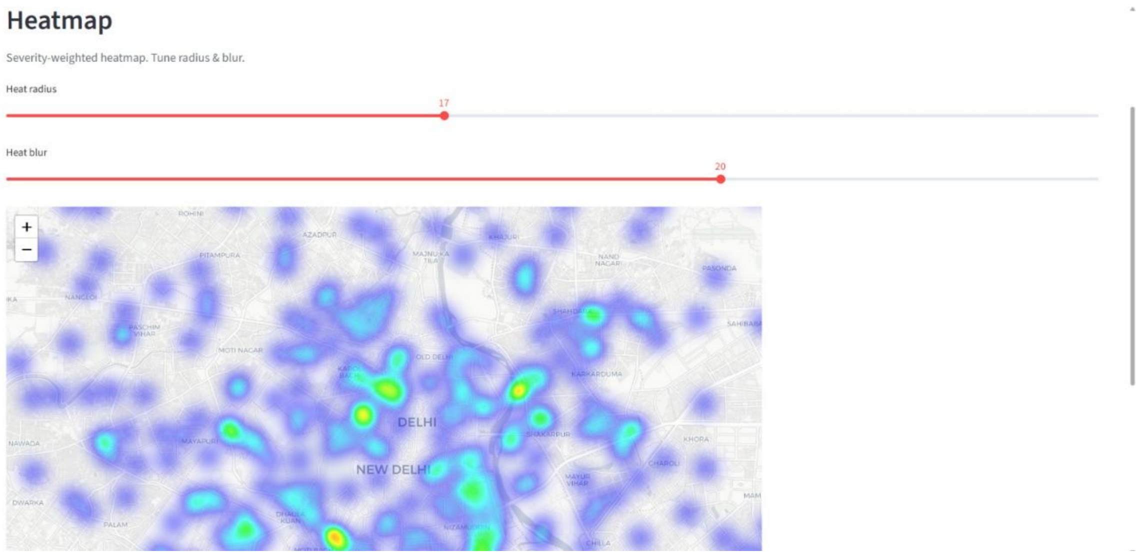
Safety Map



Safety Map

Marker clusters for incidents and safe zones. Click a point to get nearest safe zone.

Heatmap



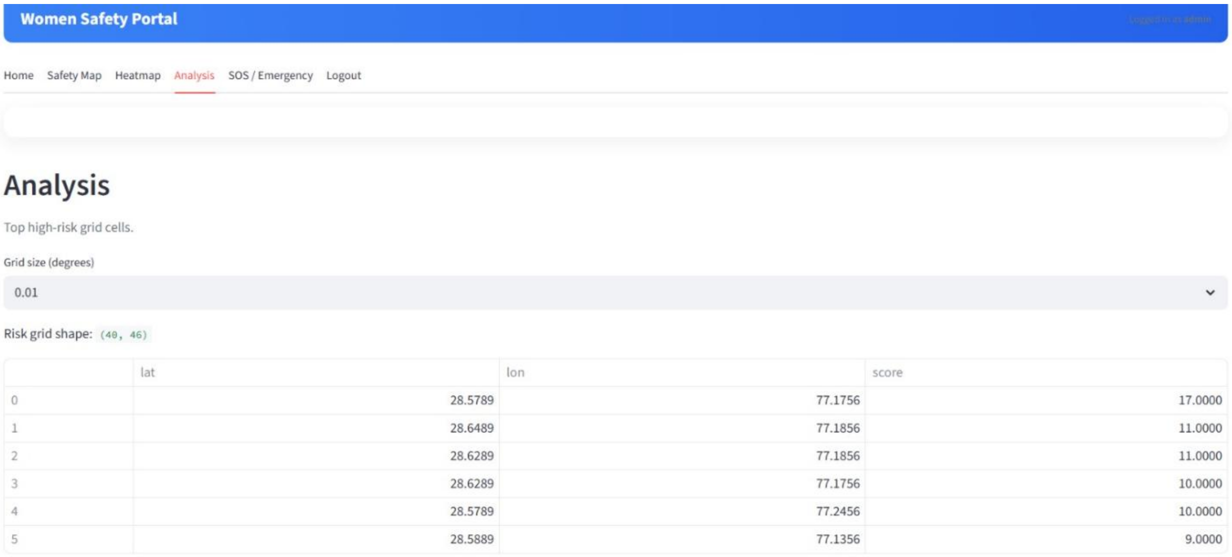
Heatmap

Severity-weighted heatmap. Tune radius & blur.

Heat radius: 17

Heat blur: 20

Analysis



Analysis

Top high-risk grid cells.

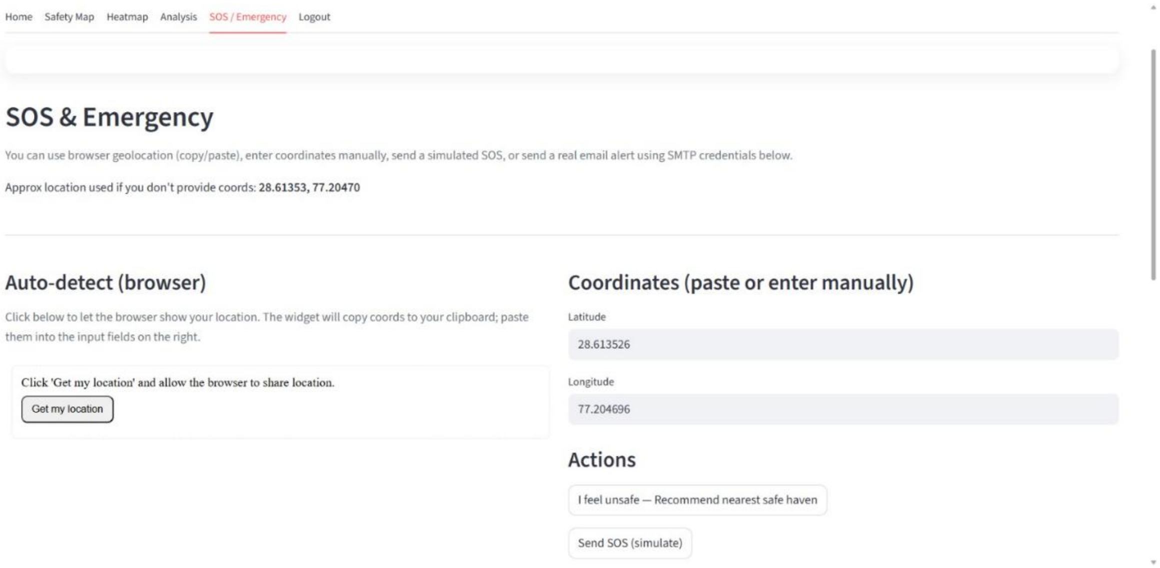
Grid size (degrees)

0.01

Risk grid shape: `(40, 46)`

	lat	lon	score
0	28.5789	77.1756	17.0000
1	28.6489	77.1856	11.0000
2	28.6289	77.1856	11.0000
3	28.6289	77.1756	10.0000
4	28.5789	77.2456	10.0000
5	28.5889	77.1356	9.0000

SOS & Emergency



SOS & Emergency

You can use browser geolocation (copy/paste), enter coordinates manually, send a simulated SOS, or send a real email alert using SMTP credentials below.

Approx location used if you don't provide coords: **28.61353, 77.20470**

Auto-detect (browser)

Click below to let the browser show your location. The widget will copy coords to your clipboard; paste them into the input fields on the right.

Click 'Get my location' and allow the browser to share location.

Coordinates (paste or enter manually)

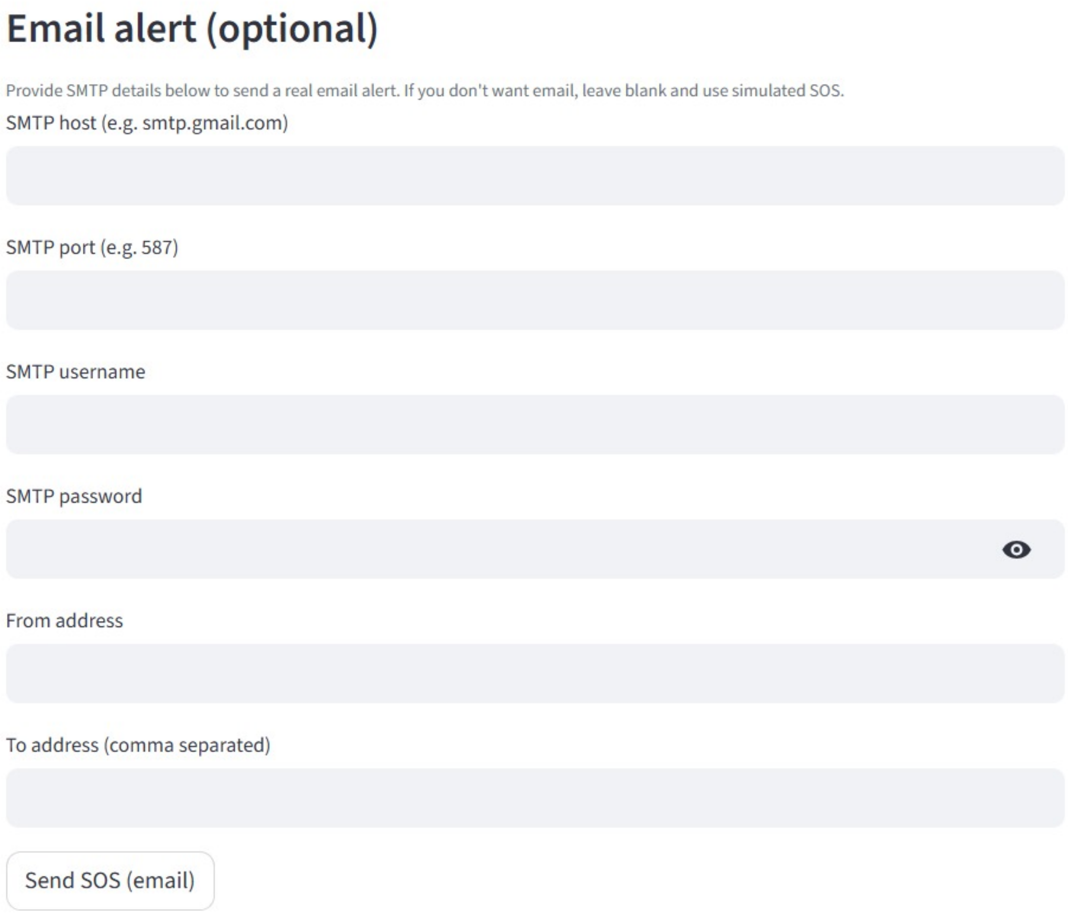
Latitude

28.613526

Longitude

77.204696

Actions



Email alert (optional)

Provide SMTP details below to send a real email alert. If you don't want email, leave blank and use simulated SOS.

SMTP host (e.g. smtp.gmail.com)

SMTP port (e.g. 587)

SMTP username

SMTP password

From address

To address (comma separated)