LazyLoops

# Safety and Security of Single Women Traveler Application

# Software Requirements Specification

## Version <1.0>

Submitted in Partial Fulfilment for the Award of Degree of Bachelor of Technology in Information Technology from Rajasthan Technical University, Kota


**MENTOR:**

**Dr. S.R. Dogiwal**

(Dept. of Information Technology)

**COORDINATOR:**

**Dr. Priyanka Yadav**

(Dept. of Information Technology)

**SUBMITTED BY:**

**Dhruv Sharma (21ESKIT042)**

**Deepesh Choudhary (21ESKIT039)**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**SWAMI KESHWANAND INSTITUTE OF TECHNOLOGY, MANAGEMENT & GRAMOTHAN**

**Ramnagaria (Jagatpura), Jaipur – 302017**

# Table of Contents

# 1. Introduction

The Software Requirements Specification (SRS) document for the **"**Sostify**"** platform serves as a comprehensive guide delineating the functional and non-functional requirements essential for designing, developing, and deploying an innovative safety and security system for single women traveler. This introduction provides an overview of the entire SRS, encompassing its purpose, scope, definitions, acronyms, abbreviations, references, technologies used, and an overview of the SRS contents.

## 1.1 Purpose

The primary objective of this SRS is to establish a clear and unified understanding among stakeholders, including developers, designers, project managers, domain experts in personal safety, and end-users, regarding the functionalities, constraints, and objectives of the **"**Sostify**"** application. By defining and documenting these requirements, this document aims to guide the development team in creating a robust, user-centric application that ensures the safety and empowerment of single women traveler.

## 1.2 Scope

**"Sostify"** is envisioned as a comprehensive, integrated platform offering a suite of functionalities to enhance safety and security during travel. Core features include real-time location tracking, emergency SOS alerts, trusted contact integration, safe route suggestions, community forums for sharing experiences, AI-powered risk assessment, and resource directories for nearby safety points like police stations and hospitals. This SRS delineates the specific requirements for each module, outlining their functionalities, interactions, and constraints.

## 1.3 Definitions, Acronyms and Abbreviations

- **SRS**: Software Requirements Specification
- **API**: Application Programming Interface
- **UI**: User Interface
- **DBMS**: Database Management System
- **IoT**: Internet of Things

## 1.4 References

This SRS references industry standards, research in women's safety technologies, and external documents or specifications used in defining the requirements for **"**Sostify**."**.

### 1.5 Technologies to be used

The "Sostify" platform utilizes cutting-edge technologies and frameworks to deliver its functionalities, including:

- Frontend: React Native for cross-platform mobile application development.
- Authentication: Firebase Authentication for Mobile SMS Verification
- Backend: Spring Boot for the backend server application.
- Database: MongoDB for data storage and retrieval.
- AI Integration: TensorFlow.js for integrating AI models.

### 1.6 Overview

This SRS document is structured systematically into distinct sections, each detailing specific facets of the "Sostify" platform. It includes an executive summary, system overview, detailed descriptions of functional and non-functional requirements, user interfaces, system constraints, assumptions, dependencies, and technical specifications. Additionally, it incorporates diagrams, use cases, and mockups to provide a comprehensive understanding of the system's architecture and functionality.

## 2. Literature Survey

The literature survey conducted for the **"**Sostify**"** SRS involved a comprehensive review of existing research, industry practices, and technological advancements related to women's safety systems, travel security applications, and relevant technologies..

### 2.1 Objective

- **Safety Application:**

  The survey included an in-depth analysis of existing women's safety and security apps such as bSafe, Safetipin, and Circle of 6, evaluating their features, usability, adoption rates, and shortcomings.
- **Travel Safety technologies:**

An examination of safety-focused travel tools and platforms for risk assessment, safe navigation, and community-based reporting systems was conducted to identify trends and gaps..

- **Technological Trends:**

  The survey reviewed advancements in real-time tracking systems, AI-based risk prediction, and wearable devices for personal security to identify emerging technologies that could enhance the **"**Sostify**"** platform.

- **Research Papers and Publications:**

  Relevant publications on women's safety, predictive analytics for crime hotspots, and user behavior in emergency situations were reviewed to gather insights into effective design methodologies and best practices.

- **Standards and Regulations:**

  The survey explored legal and ethical considerations, including data privacy standards like GDPR, and local policies on women's safety and emergency services, ensuring **"**Sostify**"** complies with these requirements.

- **Findings and Insights:**

  The literature survey revealed insightful findings into the realm of women's safety and travel security applications, emphasizing the critical role of leveraging technological advancements to enhance personal safety. Key opportunities were identified for integrating cutting-edge technologies, such as real-time GPS tracking, AI-powered risk prediction, and emergency communication systems, to provide proactive and responsive solutions.

  The survey also highlighted the importance of addressing gaps in accessibility and usability, ensuring that these innovative solutions are intuitive, reliable, and designed to function effectively under high-stress scenarios. Furthermore, emphasis was placed on the significance of community engagement, such as crowd-sourced safety data and peer-driven reporting mechanisms, to create a dynamic and informed ecosystem of safety resources.

Overall, the findings underscore the transformative potential of technology in ensuring women's safety while emphasizing the necessity for robust privacy protections, inclusivity, and collaboration with local authorities and community networks. A concerted effort is required to bridge existing gaps, ensuring equitable access to these advancements for all women traveler, regardless of their geographic or socioeconomic context.

**Conclusion**

In conclusion, Sostify is not just an application but a dedicated effort to empower women with safety and confidence. By fostering collaboration between technology, community networks, and local authorities, it strives to bridge existing gaps and create a world where safety during travel is a right, not a privilege.

# 3.Specific Requirements

## 3.1 Functional Requirement

Functional requirements outline the specific functionalities and features that the Sostify platform must provide to meet the needs and expectations of single women traveler.

1. **Login Module**:
    - Description: Facilitates user authentication through mobile number input with OTP verification. Additionally, it provides a guest login option for users who do not wish to register.
    - Key Features: Secure mobile number authentication, OTP verification, guest login option.
2. **Location Module**:
    - Description: Requests permission to access the device’s location services and fetches latitude and longitude coordinates. Displays the user's current city and state for reference.
    - Key Features: Location permission request, retrieval of latitude and longitude coordinates, display of city and state information.
3. **Safety Check-In Module:**

- Description: Allows users to check in at their current location and set periodic safety check-ins to ensure they remain safe during their journey.
- Key Features: Check-in functionality, automated reminders for periodic safety updates

4. **Add User Info Module**:
    - Description: Collects essential user information such as name, emergency contact numbers, home address, and preferred travel routes to personalize user experience and enable quick response during emergencies.
    - Key Features: User information collection, emergency contact setup, travel personalization.
5. **Emergency Alert Module:**
    - Description: Enables users to send emergency alerts to pre-configured contacts and authorities. Provides options to trigger alerts via button, voice command, or shaking the device.
    - Key Features: SOS alert functionality, multiple trigger mechanisms, real-time GPS location sharing..
6. **Nearby Assistance Module**:
    - Description: Displays nearby safe zones, police stations, hospitals, and trusted accommodations. Provides navigation assistance to these locations.
    - Key Features: Map-based safe zone display, real-time navigation support.
7. **Risk Assessment Module**:
    - Description: Uses AI to evaluate the risk level of the user's surroundings based on real-time data, including crime statistics and user feedback.
    - Key Features: AI-powered risk analysis, safety ratings for locations, contextual alerts.
8. **Ask AI Module**:
    - Description: Offers users the ability to interact with an AI-powered chatbot (LLM model GEMINI Pro 1.5) to ask questions and receive safety tips, travel guidance, and emergency advice.

- Key Features: AI-powered chatbot interaction, instant safety-related advice.

9. **Notification Module**:
    - Description: Displays notifications sent by the server, including safety alerts, reminders for check-ins, and informational updates. Offers filtering options by type.
    - Key Features: Notification display, filtering options by alerts, updates, and reminders.
10. **Home Screen Module**:
    - Description: Presents users with key information such as location details, recent safety alerts, check-in status, and navigation buttons for accessing core features.
    - Key Features: Location display, safety updates, navigation buttons.
11. **Dashboard Module**:
    - Description: Provides users with access to personal information, emergency contact details, and account settings. Includes a logout button for account management.
    - Key Features: User information display, access to emergency contact setup, account management.

### 3.2Non-Functional Requirements

Non-functional requirements define the qualities and characteristics that the system must possess, such as performance, security, and usability.

1. Performance
    - The system response time for safety alerts shall be within 2 seconds.
    - The platform shall handle a concurrent user load of at least 1,000 users.
2. Security
    - User passwords and sensitive data shall be securely stored using industry-standard encryption algorithms.

The system shall implement role-based access control to ensure data privacy

3. Usability
    - The user interface shall follow industry best practices for a user-friendly experience.
    - The platform shall support multiple languages for enhanced accessibility

4. Reliability
    - The system shall have a minimum uptime of 99.9%.
    - Data backup and recovery mechanisms shall be in place to prevent data loss.

### 3.3 Hardware Requirements

The hardware requirements outline the necessary infrastructure for deploying and running the Sostify platform.

1. Server Infrastructure:
    - Minimum of dual-core processors with 8 GB RAM for the backend server.
    - Adequate storage capacity for database and system files.
2. Mobile Devices
    - The mobile application shall be compatible with Android devices (version 7.0 and above)

### 3.4 Software Requirements

Software requirements detail the necessary software components and dependencies for the Sostify platform.

1. Backend Technologies:
    - Spring Boot framework for the backend server application. • Express to fetch and use AI Prediction
    - MongoDB database for data storage.
2. Frontend Technologies:
    - React Native for cross-platform mobile application development.
3. Third-Party Integrations:
    - Integration with market analysis APIs.

### 3.5 Agile Methodology

The Sostify project follows the Agile development methodology, prioritizing flexibility, collaboration, and iterative cycles to meet evolving user needs. The process includes:

**1. Project Initiation:**

- Define project objectives, goals, and scope.
- Assemble the project team comprising developers, designers, and domain experts.

**2. Product Backlog Creation:**

- Identify and list all desired features and functionalities.
- Prioritize these features based on their importance and potential impact on Farmers.

**3. Sprint Planning:**

- Break down the prioritized features into smaller tasks for implementation within sprints (short development cycles).
- Estimate the effort required for each task and allocate them to the upcoming sprints.

**4. Sprint Execution:**

- Development teams work on the tasks assigned for the sprint.
- Daily stand-up meetings are held to discuss progress, challenges, and plan adjustments.

**5. Continuous Integration and Testing:**

- Developers continuously integrate their code into the shared repository.
- Automated and manual testing is performed to ensure the developed features meet quality standards.

**6. Sprint Review:**

- At the end of each sprint, a review meeting is held to showcase the completed work to stakeholders.
- Feedback is collected, and adjustments are made to improve the next sprint.

**7. Sprint Retrospective:**

- The team reflects on the sprint's successes and challenges.
- Identify areas for improvement and implement changes to enhance team efficiency and product quality.

**8. Incremental Deployment:**

- Completed and tested features are deployed incrementally to the live environment.
- Farmers can start using and providing feedback on the new functionalities.

**9. Continuous Feedback and Adaptation:**

- Gather feedback from Farmers and stakeholders regularly.
- Adapt the project scope, features, or priorities based on feedback and changing requirements.

**10. Iterative Development:**

- Repeat the cycle, starting from sprint planning, to incorporate new features and improvements.

### 3.6 Business Process Model

1. **Login Module**:
   - Input: User's mobile number and OTP.
   - Process: Verify the mobile number and OTP for authentication.
   - Output: Successful login or error message.
2. **Location Module**:
   - Input: User's permission to access device location.
   - Process: Request permission to access device location services and fetch latitude and longitude coordinates.
   - Output: Display User's current location, city and state information.
3. **Safety Check-in Module**:
   - Input: User's check-in details or periodic reminders.
   - Process: Record check-in data, send reminders.
   - Output: Confirmation of check-in status.
4. **Add User Info Module**:
   - Input: User's personal information such as name, contact number, and address.
   - Process: Collect and store User information in the database.
   - Output: Personalized User experience based on stored information.
5. **Emergency Module**:
   - Input: Store Emergency contacts for emergency.

- Process: Retrieve and display contacts information.
- Output: Contact Emergency contacts while emergency .

6. **Nearby Assistance Module**:
    - Input: Retrieve user's current location.
    - Process: Retrieve nearby data for the selected location.
    - Output: Display nearby safezones.
7. **Risk Assessment Module**:
    - Input: Unusual activities and app opening.
    - Process: Analyse the images using AI algorithms for risk analysis.
    - Output: Based on analysis recommended quick steps .
8. **Ask AI Module**:
    - Input: User's queries related to safety.
    - Process: Process User's queries using AI chatbot.
    - Output: Instant responses and safety advice.
9. **Notification Module**:
    - Input: User's interaction with navigation buttons.
    - Process: Navigate between home screen and dashboard tabs.
    - Output: Seamless access to essential safety information.
10. **Home Screen Module**:
    - Input: Server-generated notifications.
    - Process: Filter notifications by type (offer, alert, info).
    - Output: Display filtered notifications to Farmer.

11. **Dashboard Module**:
    - Input: User's access to personal and safety information.
    - Process: Provide access to User's details, personal info, and account management options.
    - Output: Personalized dashboard with User-specific information

This business process model outlines key processes within the Sostify system, detailing inputs, processes, and resulting outputs. It illustrates how various functionalities interact and contribute to the overall functioning of the platform, ensuring the safety and security of single women traveler. The model highlights how the platform's features seamlessly work together to provide a comprehensive safety solution, empowering users to travel with confidence and peace of mind.

### 3.7 Supplementary Requirements

Supplementary requirements include any additional requirements that are not covered by the previous sections but are essential for the success of the project.

1. Scalability:
    - The system architecture shall support horizontal scaling for an increasing user base.
2. Documentation:
    - Implement multi-factor authentication and data encryption.
3. Security Measures:
    - Data Encryption: Ensure sensitive Farmer data is encrypted to maintain confidentiality.
4. Compatibility and Integration:
    - API Integration: Provide well-documented APIs to allow seamless integration with third-party applications or services.
5. Performance Optimization:
    - Load Balancing: Implement load balancing techniques to ensure optimal performance, especially during high traffic periods.

# 4.Overall Description

### 4.1 Use-Case Model Survey

**Actors:**

**User**: Primary user who interacts with the system.

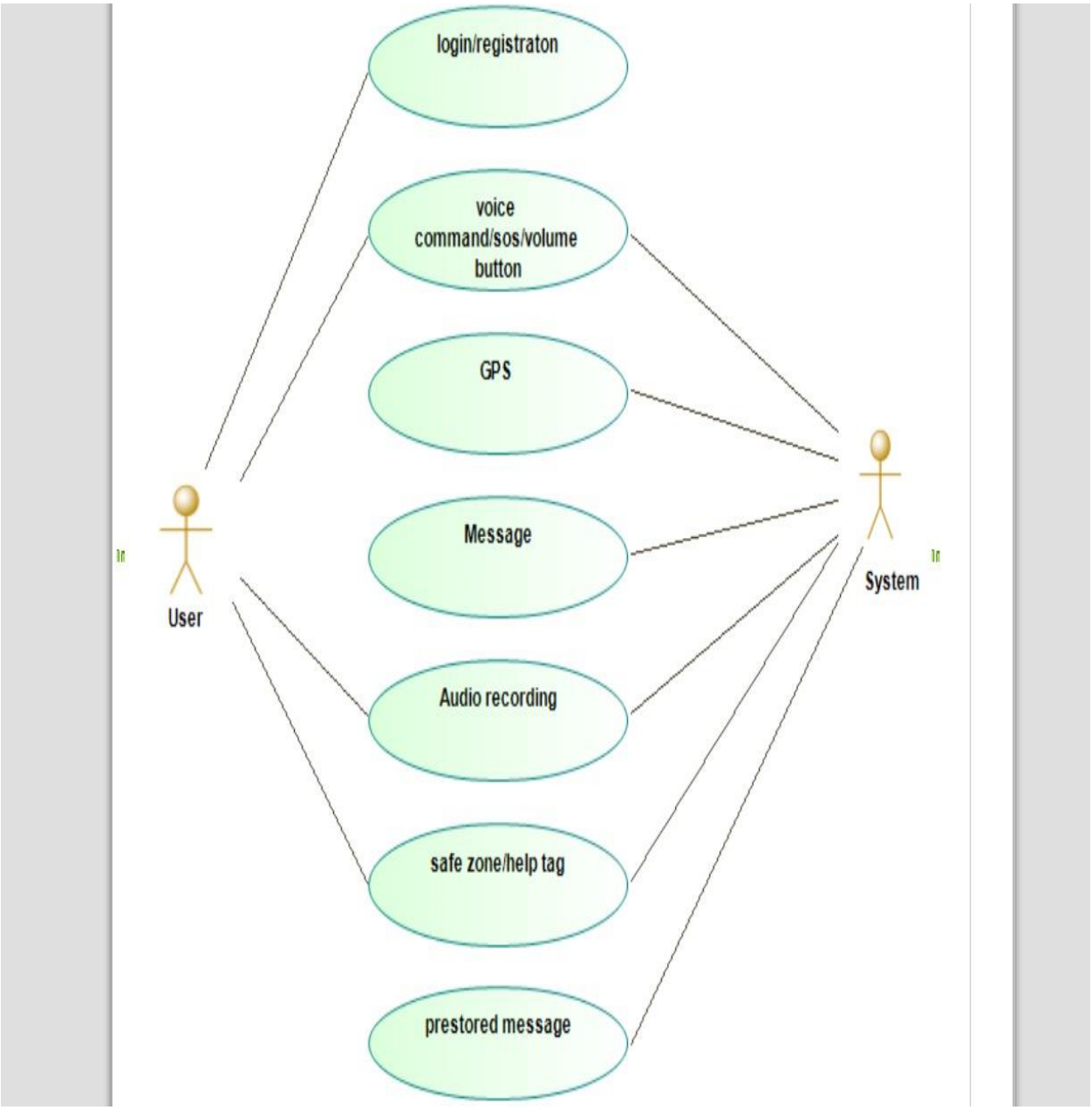


*Figure 4.1:Sostify Platform Use Case Diagram*

## 4.2 Behaviors Diagrams

• Activity Diagram

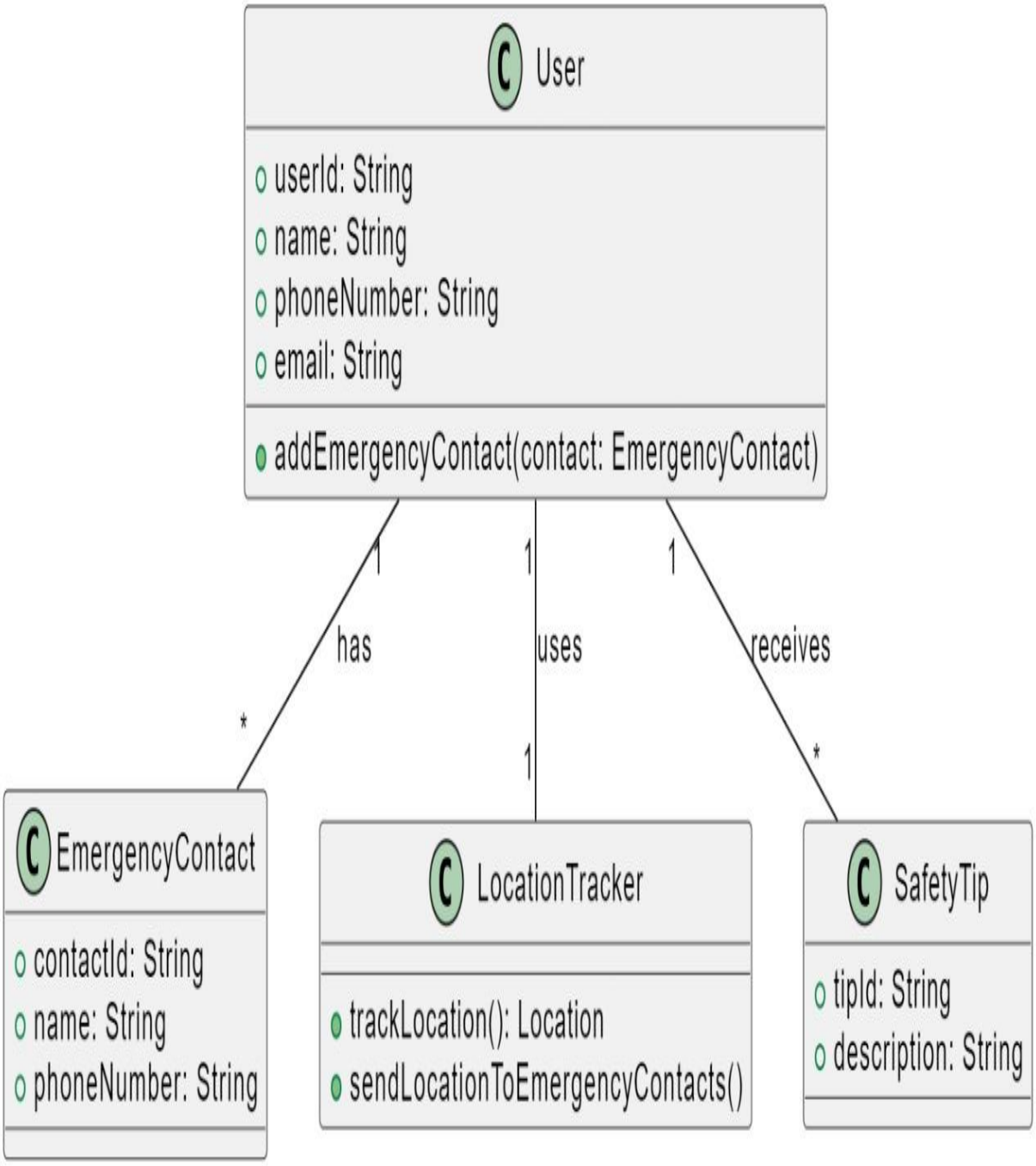


Figure 4.2:Sostify Activity Diagram

### 4.3 Sequence Diagram

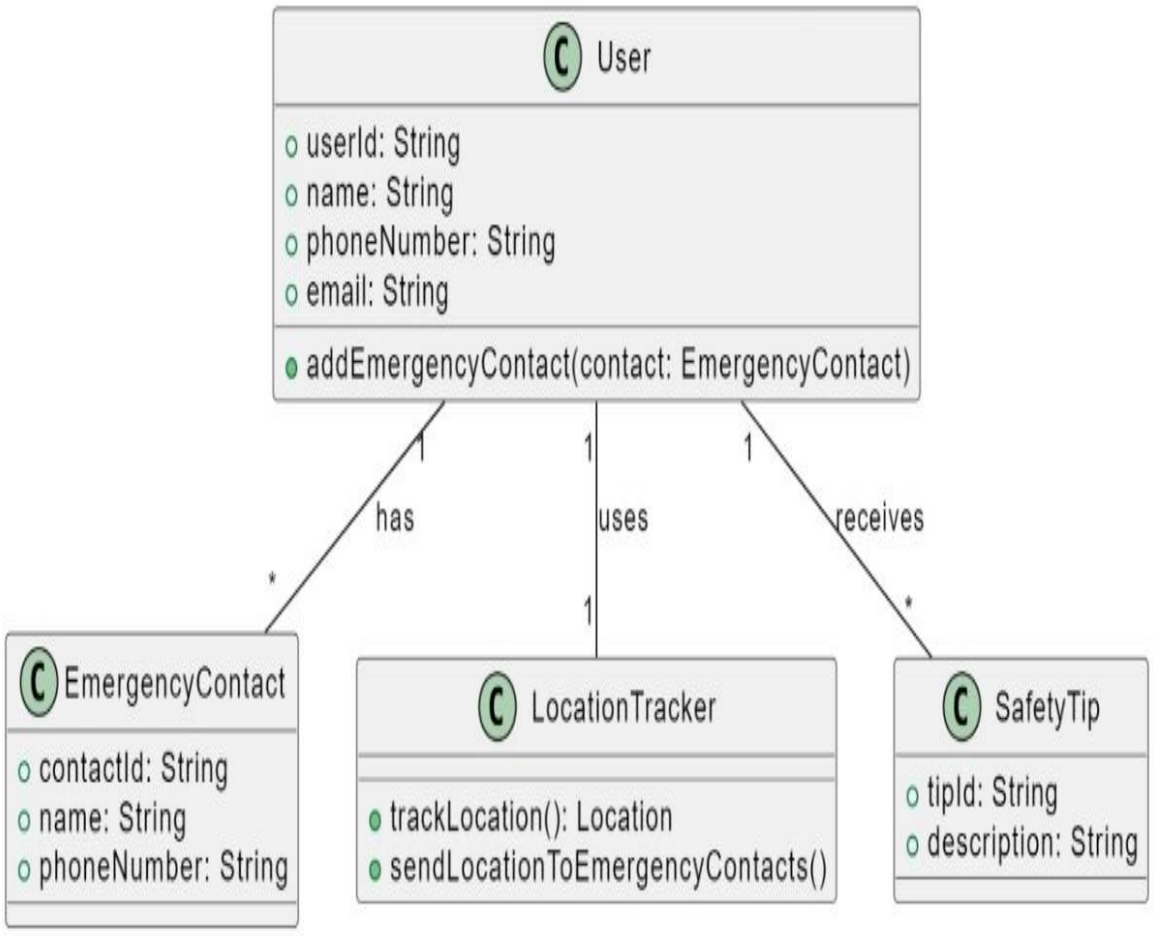


*Figure 4.3: Sequence Diagram for Users all Modules*

- Data Flow Diagram – Level 0

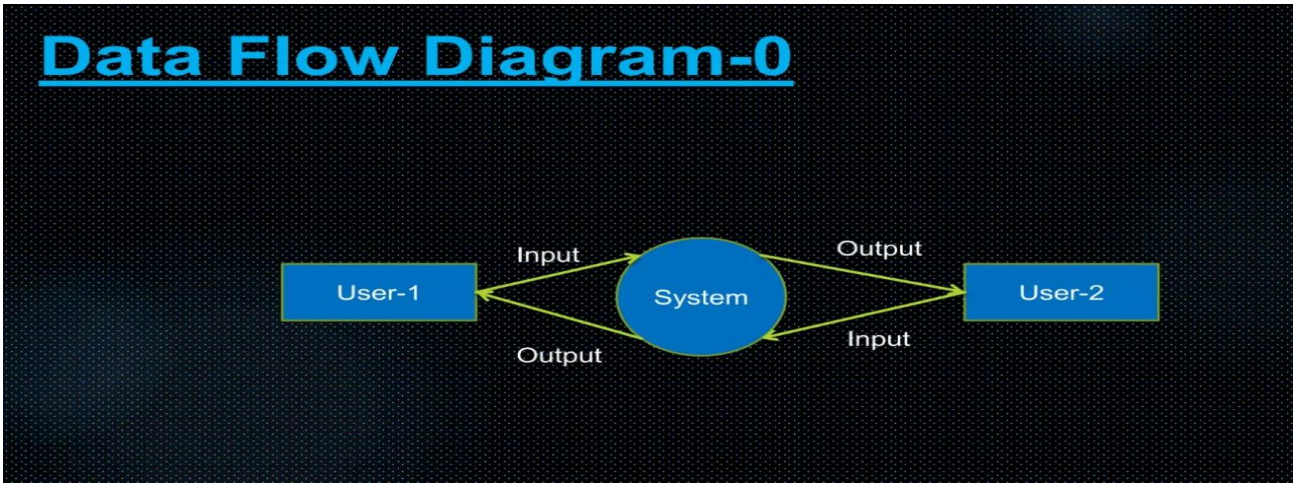


*Figure 4.4: DFD Level 0*

- Data Flow Diagram – Level 1

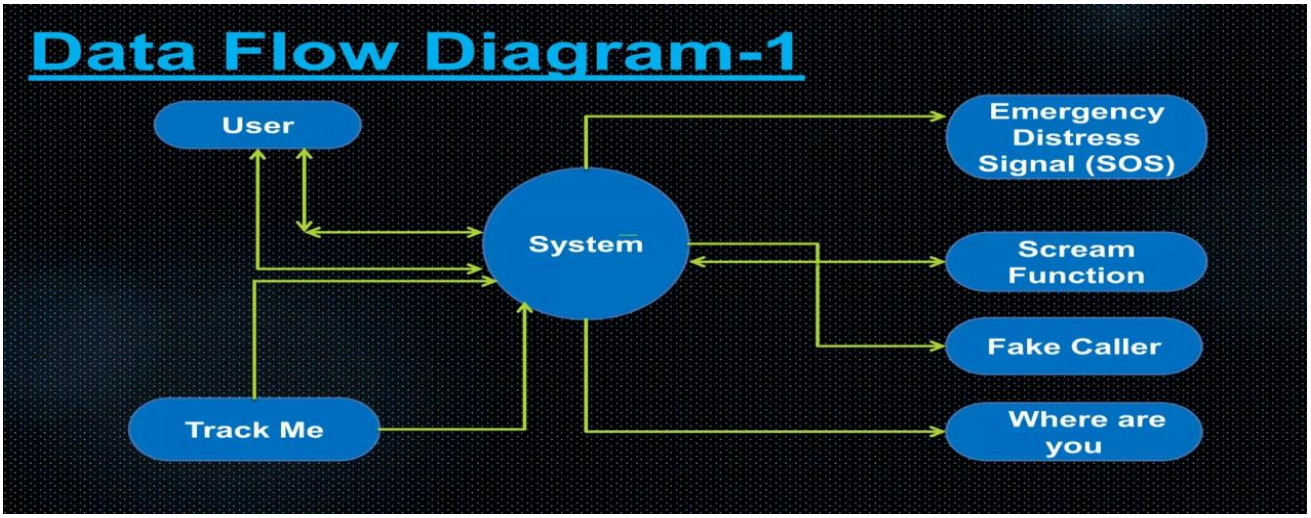


*Figure 4.5:DFD Level 1*

- Data Flow Diagram – Level 2

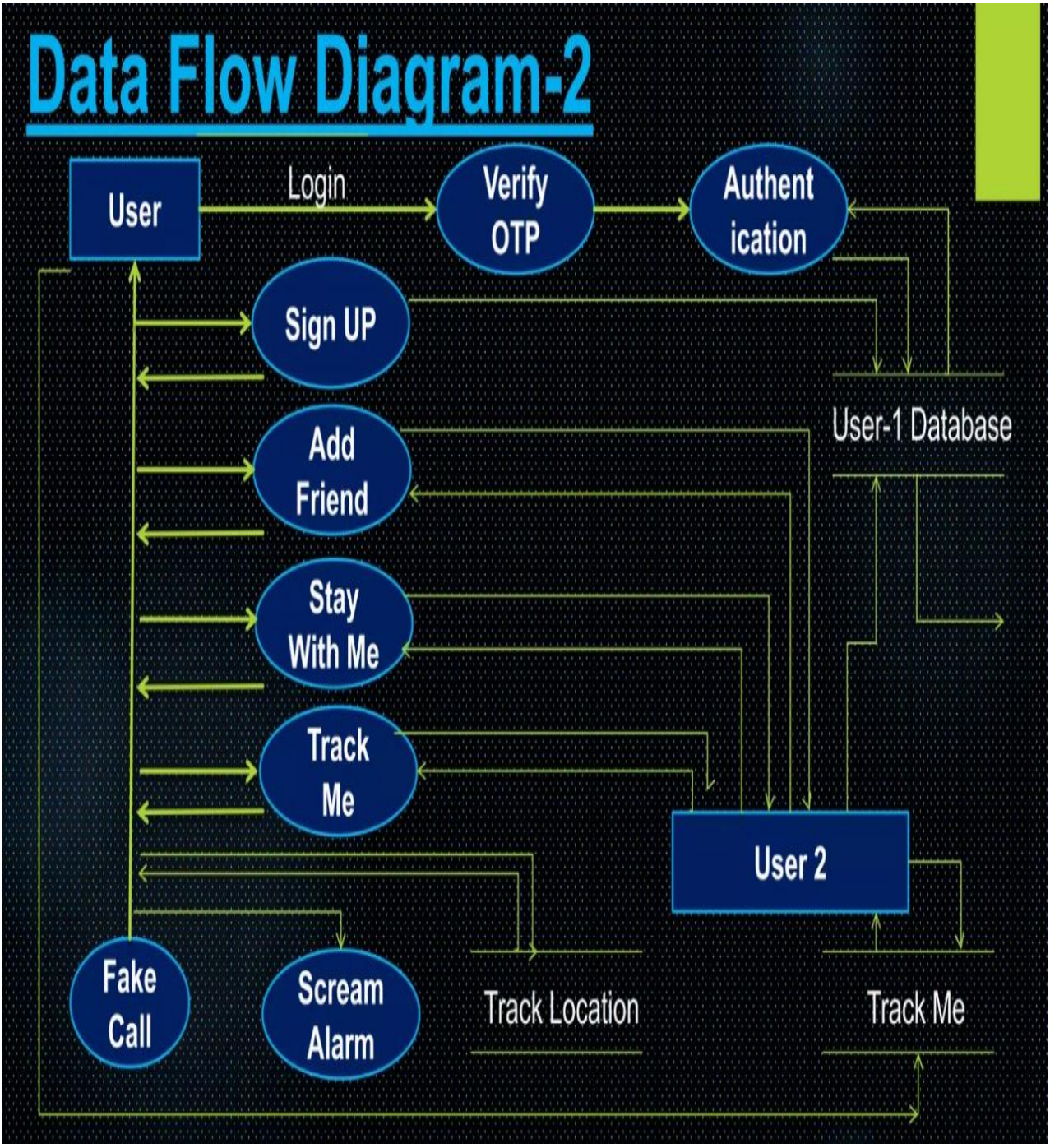


*Figure 4.6: DFD Level 2*

- Communication Diagram

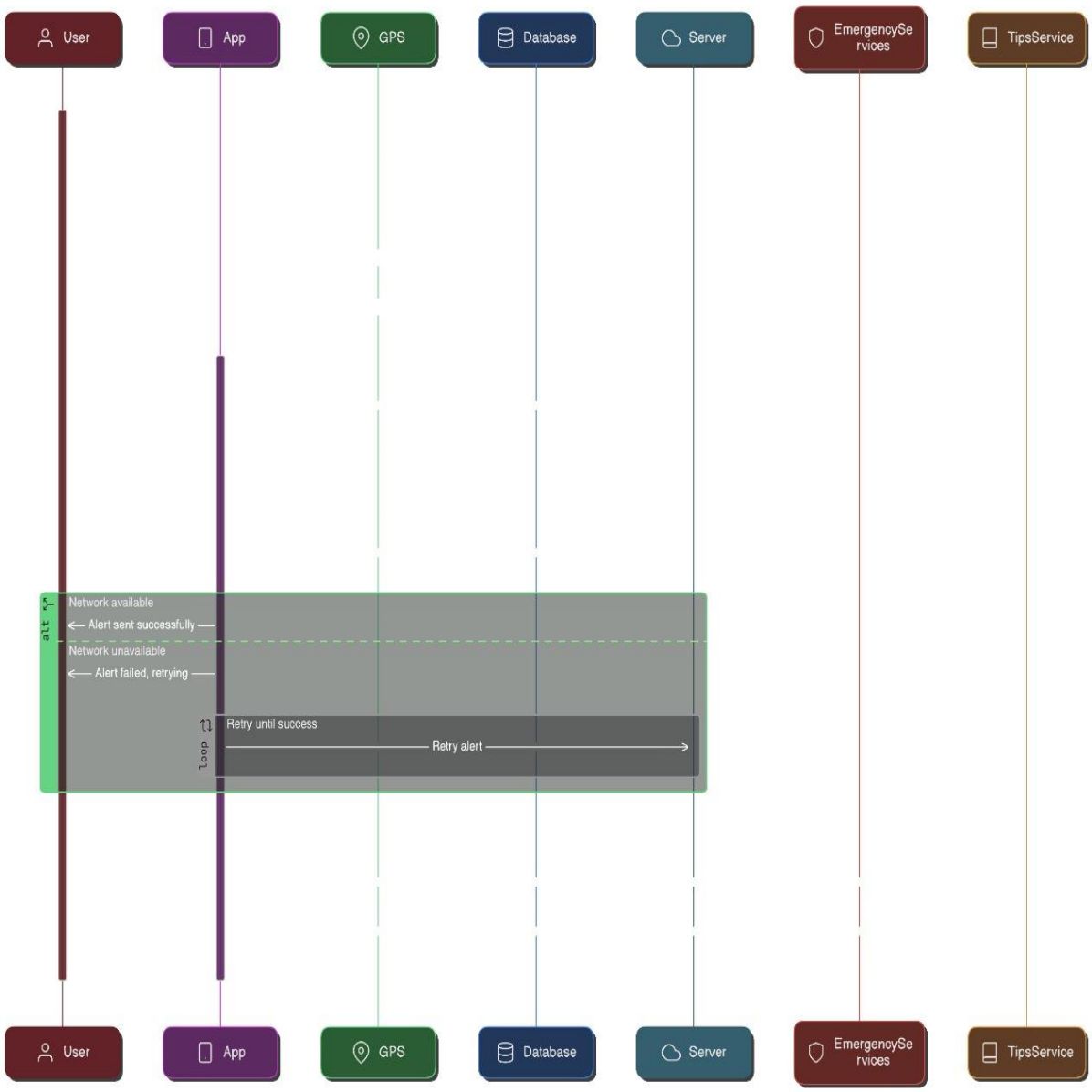


*Figure 4.7: Sostify Communication Diagram*

### 4.3 Structural Diagrams

• Deployment Diagram

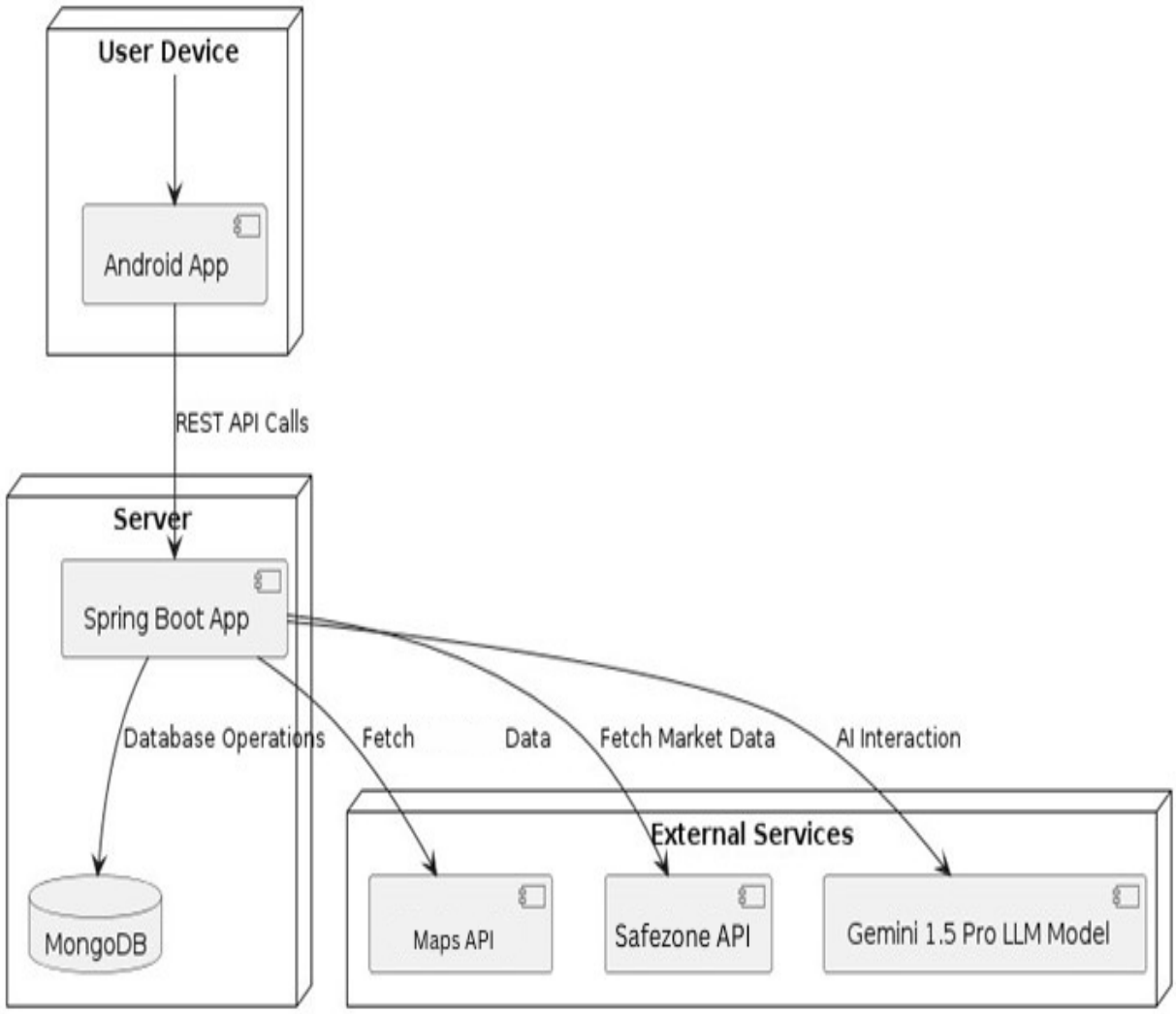


*Figure 4.8: Deployment Diagram for Project*

- Component Diagram

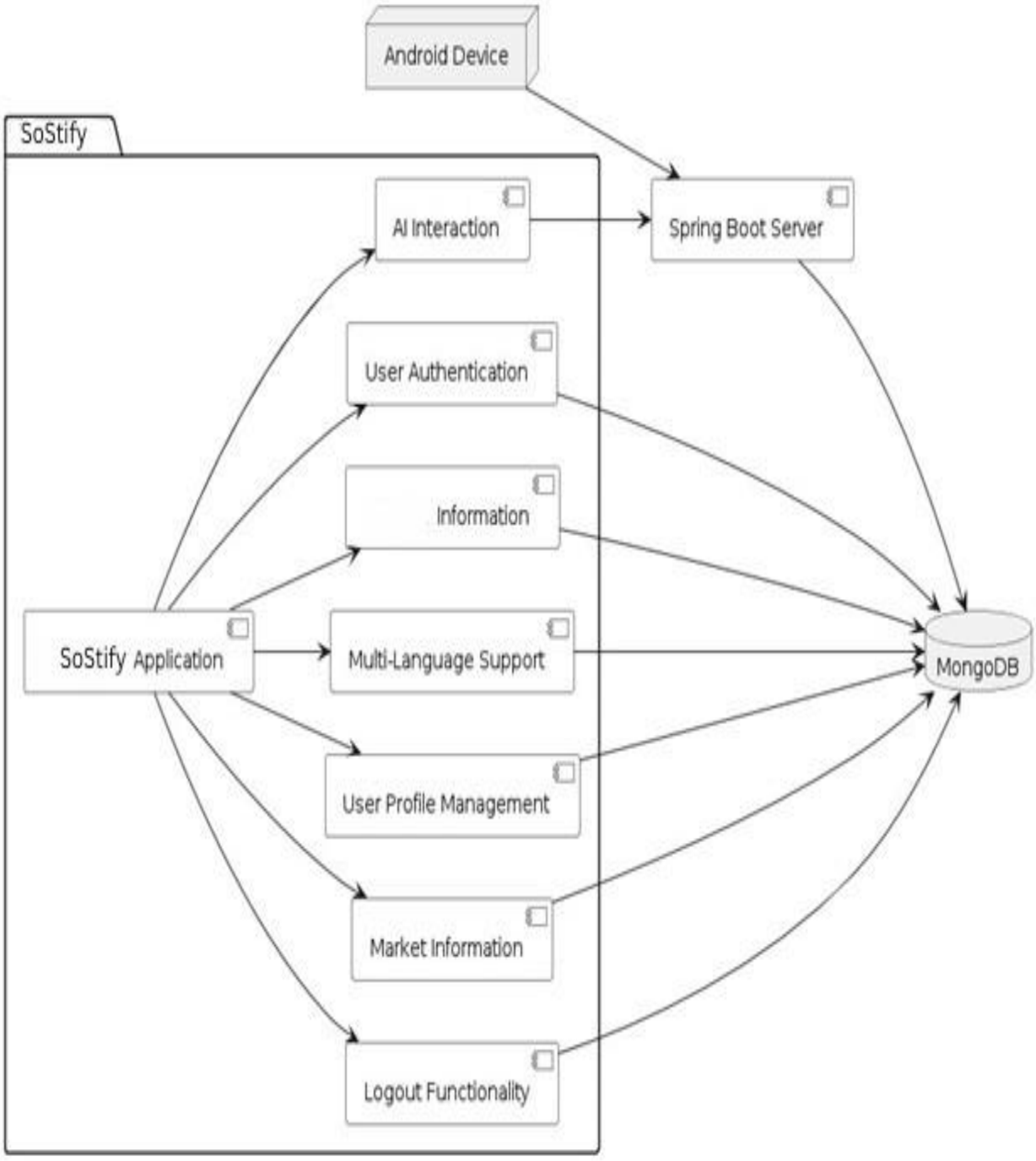


*Figure 4.9: Component Diagram*

## 4.4 Database Diagram

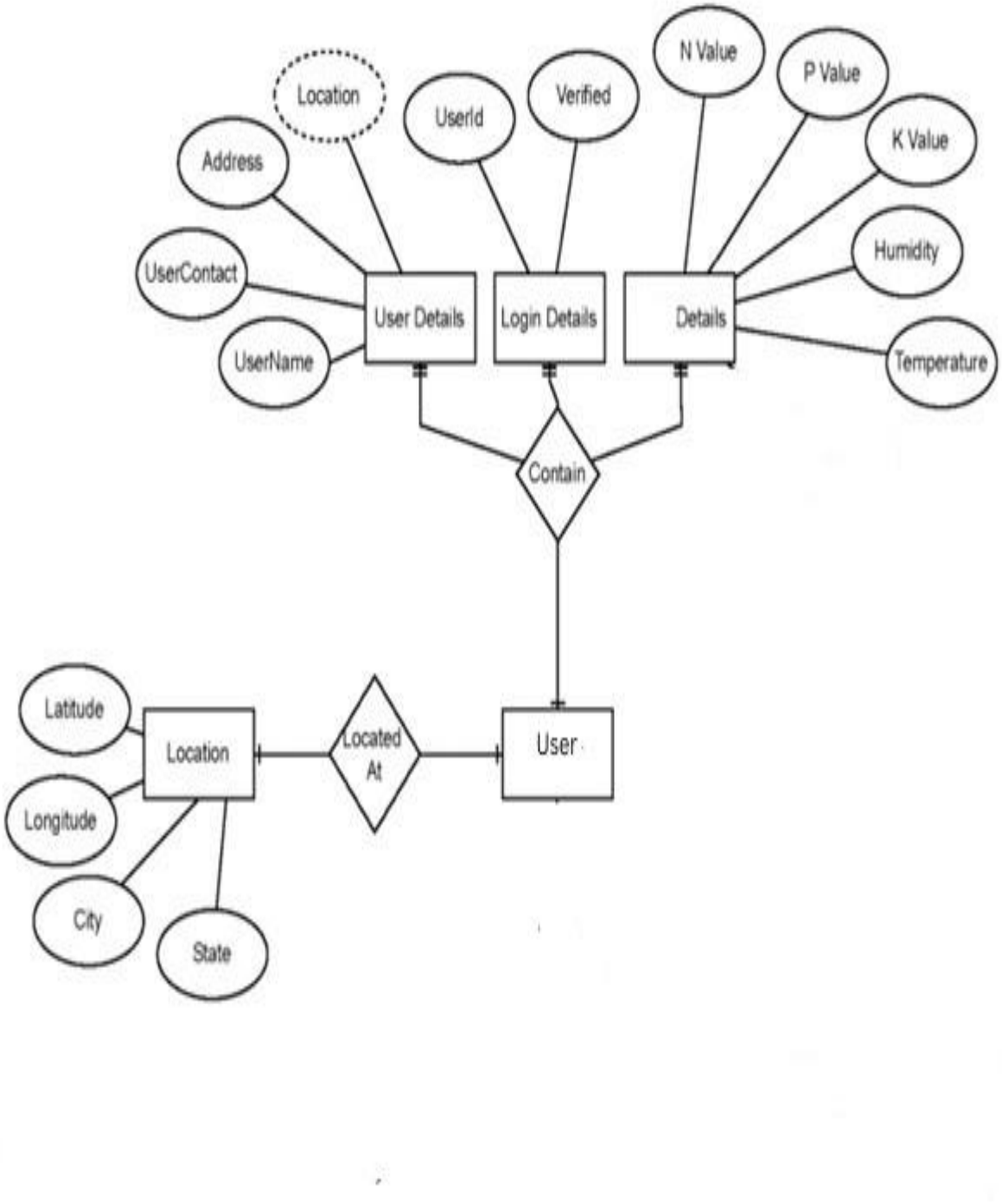


*Figure 4.10: Entity Relationship Diagram*

## 4.5 Assumptions and Dependencies

1. **Technical Feasibility Assumptions:**
    - Hardware Infrastructure: The assumption is that the required hardware infrastructure, including servers, network components, and IoT devices, will be available and capable of supporting the system's functionalities.
    - Software Dependencies: The project assumes access to necessary software components like Node.js, Spring Boot, React Native, MySQL, etc., with compatible versions and configurations to run the system smoothly.
    - IoT Device Integration: Assumes the successful integration and functionality of IoT devices (e.g., soil moisture sensors, temperature sensors) within the system architecture.
2. **Subsystems or Component Availability:**
    - Availability of APIs: The project relies on third-party APIs for weather forecasts, market data, etc., assuming their consistent availability and access for the system's functionality.
    - Database System: Assumes uninterrupted access to the MySQL database system for data storage and retrieval.
3. **Project-Related Assumptions:**
    - User Adoption: Assumes an adoption rate among the target user base (farmers and buyers) for the mobile and web platforms, leading to active system usage.
    - Data Accuracy: Assumes the accuracy and reliability of data received from IoT devices, external APIs, and user inputs.
4. **Dependencies on External Factors:**
    - Internet Connectivity: Dependencies exist on stable internet connectivity for Farmer to access the system's online features and real-time data.
    - Regulatory Compliance: The project assumes adherence to agricultural regulations and standards governing farming practices and data privacy laws.

# 5. System Architecture

## 5.1 Client-Server Architecture

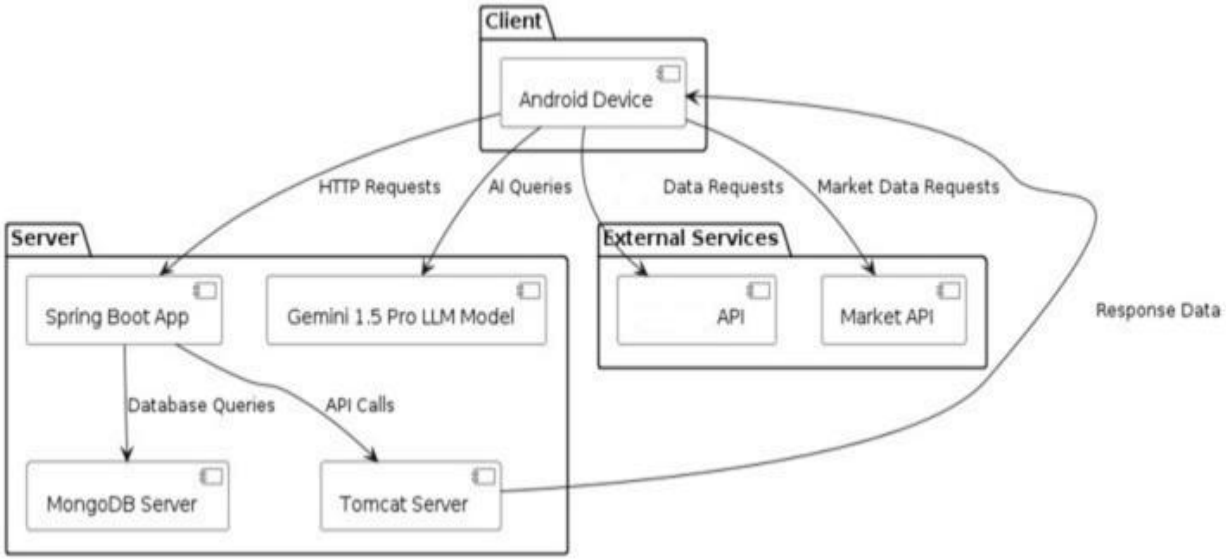


*Figure 5.1: Client Server Architecture*

## 5.2 Communication Interface

Communication interfaces refer to the channels or mechanisms through which various system components, devices, or modules interact and exchange data. Here's an outline of communication interfaces in the project:

**API Endpoints:**

- RESTful APIs: Utilized for communication between the client-side applications (React Native mobile app and web interface) and the server-side (Express and Spring Boot).
- Endpoints for Data Exchange: Different API endpoints are defined for functionalities like User's authentication, crop management, weather data retrieval, forum interaction, financial services, and IoT device integration.

**Database Interaction:**

- Database Connectivity: Interfaces responsible for connecting the backend server (Node.js and Spring Boot) to the MySQL database for data storage and retrieval.
- Query Interfaces: Mechanisms facilitating the execution of database queries to manage and retrieve information from the database.

**Networking Interfaces:**

- Internet Connectivity: Ensures connectivity between the client-side applications, serverside components, and external services via the internet.
- Secure Communication Protocols: Integration of secure communication protocols (HTTPS, SSL/TLS) for encrypted data transmission between the clients and the server.

These communication interfaces define the pathways and methods through which different system components and external entities interact, ensuring effective data exchange and system functionality within the "Kisan Vikas" agricultural platform.

# 6. Supporting Information

### 6.1 List of Diagrams



# 7. Conclusion & Future scope

### 7.1 Conclusion

Summary of Achievements:

- Accomplishments: Summarize the achievements and successful implementations within the "Sostify" project.

- Key Objectives: Highlight how the project has addressed its primary objectives, such as providing safety support, and freedom, etc.

- Challenges Overcome: Discuss any challenges faced during the project and how they were mitigated.

Impact:

- Empowered women travelers with tools to enhance personal safety, reducing travel-related anxiety.
- Simplified emergency assistance with one-tap alerts and seamless connectivity to trusted contacts and authorities.

## 7.2 Future Scope

**Potential Enhancements:**

- Advanced Analytics: Introduce AI-driven predictive safety alerts based on real-time trends and historical data.
- Enhanced Community Engagement: Build forums or groups within the app for users to share tips, experiences, and feedback.
- Expanded Market Reach: Incorporate location-specific safety insights for more regions and cultural contexts.

**Research and Development:**

- AI Integration: Develop advanced algorithms for personalized safety suggestions based on travel history and user behavior.
- IoT Integration: Integrate IoT wearables (e.g., smart bracelets) for discreet SOS triggering and location sharing.
- Mobile App Enhancement: Optimize for cross-platform compatibility, ensuring seamless performance on diverse devices and operating systems.

**Community Engagement and Partnerships:**

- Collaborations: Partner with global travel organizations, women's advocacy groups, and local law enforcement to broaden safety coverage.
- Community Growth**:** Foster community trust by highlighting user success stories and featuring safety tips contributed by active users.

**Conclusion of Future Scope:**

- Vision and Direction: Sostify aims to evolve into a comprehensive, global safety platform for women, integrating cutting-edge technology, community-driven insights, and partnerships to ensure seamless, confident, and secure travel experiences. The project envisions sustained innovation, adaptability, and a proactive approach to addressing the evolving needs of women travelers.

# 8. Concerns / Queries / Doubts if any:

Project-related Queries:

- How can the app integrate with local emergency services across multiple countries?
- What strategies can ensure seamless multi-language support to cater to a global user base?

Technological Queries:

- What challenges might arise in implementing voice-activated safety features using advanced NLP tools?
- How can React Native be optimized to handle complex UI components for emergency and real-time safety alerts?
- How can we ensure reliable connectivity with IoT devices for SOS and location tracking functionalities?
- What are the best practices for employing Agile methodology to accelerate feature delivery without compromising quality?