



CMPE 491 / SENG 491

Senior Design Project I

Project Analysis Report

Burak Güçlü

Gizem Yüksel

İrem Özyurt

Zeynep Sude Bal

Advisor

Emin Kuğu

22.11.2024

TABLE OF CONTENTS

1.	Introduction	4
2.	Proposed System	6
2.1	Overview	6
2.2	Functional Requirements.....	8
2.3	Non-functional Requirements	13
2.4	Pseudo Requirements	16
2.5	System Models	18
2.5.1	Scenarios	18
2.5.2	Use Case Model	25
2.5.3	Object and Class Model	26
2.5.4	Dynamic Models	27
2.5.5	User interface - Navigational Paths and Screen Mock-ups.....	30
3.	Glossary.....	35
4.	References	36

FIGURE LIST

Figure 1 - Emergency Communication Systems (adapted from www.crisis-control.com).....	7
Figure 2 - Use Case Diagram	25
Figure 3 - Class Diagram	26
Figure 4 - Object Diagram	26
Figure 5 - Sequence Diagram I	27
Figure 6 - Sequence Diagram II	27
Figure 7 - Sequence Diagram III.....	27
Figure 8 - State Diagram for Earthquake Alert	28
Figure 9 - State Diagram for Fire Alert.....	28
Figure 10 - State Diagram for Flood Alert	29
Figure 11 - Website Login Page Mock-up	30
Figure 12 - Website Reset Password Page Mock-up	30
Figure 13 - Website Dashboard Mock-up	31
Figure 14 - Navigational Path for Website.....	31
Figure 15 - Mobile App Login Screen Mock-up.....	32
Figure 16 - Mobile App Menu Screen Mock-up.....	32
Figure 17 - Mobile App Communication Screen Mock-up	33
Figure 18 - Mobile App Notifications Screen Mock-up	33
Figure 19 - Mobile App Profile Screen Mock-up	34
Figure 20 - Navigational Path for Mobile App	34

1. Introduction

Today, natural disasters and unexpected emergencies are among the most important risks that threaten human life. Especially, in Turkey which is located in an earthquake zone frequently has to face the difficulties caused by earthquakes. Natural disasters such as earthquakes, floods and fires are events that occur suddenly and have a great impact. Considering the effects of such disasters on society, it is seen that it is of vital importance to access information quickly after the event and to direct emergency response teams when necessary. Determining whether people are safe at the time of a disaster and providing accurate and rapid information to emergency teams is a critical step that can save people's lives. At this point, the opportunities offered by technology play a key role in disaster management and ensuring people's safety.

The main purpose of this project is to detect risk factors through sensors at the time of a disaster and to receive rapid feedback by sending notifications via mobile application to verify people's safety status. Based on the data obtained, status information will be transmitted to both the relatives of the users and the emergency response teams when needed, and rapid action will be taken. In this way, it is aimed to ensure the safety of people, and at the same time, effective communication is provided by collecting the necessary information to quickly direct emergency response teams to the scene.

The working principle of the project is to provide an integrated solution using several different technologies to accelerate emergency response processes at the time of the incident. The system detects dangerous situations by monitoring data from various sensors. For example, different sensors such as temperature sensors, water level sensors and seismic wave sensors automatically send a warning through a central system when they detect a possible disaster situation. These warnings are both sent to a central web application and transmitted as security verification notifications to the mobile devices of individuals in the disaster area. In this way, rapid feedback is provided about the security status of individuals in the area in the event of a disaster.

The mobile application instantly collects security status information by sending a notification to the user containing the question "Are you safe?". The user selects one of the options "I am safe" or "I am not safe" as an answer. If the user marks himself as unsafe or does not respond within a certain period of time, the system automatically records the user's identification information (such as name, surname, location, and phone number) to be forwarded to

emergency response teams. This process not only ensures a faster response to the disaster area, but also ensures that the users' families and loved ones are informed about the situation, thus minimizing possible anxiety and worry.

The mobile application allows users to create profiles before a disaster and add the relatives they want to receive notifications about in case of safety. Users can edit their profiles, update their relatives' contact information, and prepare emergency notifications quickly through this application if needed. At the same time, they can quickly respond to their loved ones in case of a disaster and report their safety status. The user-friendly and easy-to-use interface of the mobile application facilitates fast and effective feedback in case of a disaster.

The web application functions as a central platform that supports the general operation of the system and facilitates incident management. The web application continuously monitors sensor data and evaluates this data to automatically prepare relevant notifications in case of a possible disaster. At the same time, responses from the mobile application are monitored instantly via this platform and a quick referral is made to the relevant units according to the information obtained. The web application provides a detailed list of registered users in the event of a disaster, allowing users to continuously monitor their security status. This makes it easier for emergency response teams to access information and intervene quickly and effectively. In addition, if users do not respond within a certain period of time or indicate that they do not feel safe, the system automatically notifies the user's relatives to confirm their security status.

This project not only ensures the safety of individuals, but also provides a comprehensive security solution by ensuring that their relatives receive information about the security situation. Accessing fast and accurate information in disaster management is one of the most critical points of the process. The technologies used within the scope of the project contribute to the protection of human life and help society to be more prepared for disasters. The most basic needs of people in times of disaster, security and peace are provided with the integrated solution offered by this project; the advantages offered by technology in disaster management are used most effectively to protect the lives of individuals.

2. Proposed System

2.1 Overview

The system has a function that is activated in the event of a disaster and quickly verifies user safety. First, IoT-based sensors (such as seismic waves, temperature, water level) installed in disaster-prone areas continuously monitor environmental changes. For example, seismic wave sensors are activated in the event of an earthquake, temperature sensors in the event of a fire, or water level sensors in cases such as floods. If the threshold values determined in the detected data are exceeded, the system immediately detects a dangerous situation and sends a warning notification to a central server.

The notifications sent to the server are evaluated via the web application and the accuracy of the disaster scenario is confirmed. When the dangerous scenario is verified, the system immediately sends a notification containing the message “Are you safe?” to users in the region via mobile applications. At this stage, users can quickly respond to the question asked by the system via their mobile devices as “I am safe” or “I am not safe.” This response given by the user is instantly recorded in the central system and processed by the system.

If the user states that they feel safe, no emergency response process is initiated, and the system records the information that this user is safe. However, if the user indicates that they do not feel safe or do not respond within a certain period of time, the system automatically prepares a notification to forward the user's information to the emergency response units. This notification includes details such as the user's name, surname, location information and phone number, and includes the necessary information for emergency teams to intervene quickly and accurately. In this way, accurate information flow is provided quickly in the event of a disaster, ensuring the highest level of people's safety.

The mobile application offers various features for users to prepare before a disaster. Users can create profiles via the mobile application, add close people to be contacted and update the contact information of these people. In addition, users can add contact information of their relatives to the system to ensure their safety in the event of a disaster. According to the security notifications sent in the event of a disaster, users can quickly communicate status information to their relatives and request the help they need.



Figure 1 - Emergency Communication Systems (adapted from www.crisis-control.com)

The web application works as a central platform that supports the general operation of the system and allows emergency response teams to receive information efficiently. The web application continuously monitors the data coming from the sensors and provides information flow to emergency response teams and the user's relatives in case of any danger, according to the security status received from the user via the mobile application. Thus, the data collected during the disaster is displayed on a central screen and the security status of the users is monitored in detail on this screen. If the user does not respond within a certain period of time or indicates that they do not feel safe, the system automatically notifies the user's relatives and continues to provide information flow.

2.2 Functional Requirements

1. Sensor Data Collection and Monitoring

Continuous Data Collection: The system shall continuously capture real-time data from a variety of IoT sensors, including seismic, temperature, and water level sensors, strategically deployed in regions vulnerable to disasters.

Threshold-Based Alerts: Each sensor shall have configurable thresholds; when readings exceed these thresholds, the system will trigger an alert to signal a potential disaster scenario.

Disaster Scenario Validation: Flagged alerts shall be routed to the central server, where the system will verify if conditions signify a genuine disaster risk.

2. Disaster Detection and Notification

User Alerts for Affected Areas: Upon validation, users in impacted zones shall receive a notification on their mobile app, inquiring, “Are you safe?” message.

Critical Alert Delivery: Disaster notifications shall bypass device “Do Not Disturb” settings, ensuring urgent delivery to users.

3. User Response Collection and Analysis

Response Options: Users shall respond to notifications by selecting “I am safe” or “I am not safe.” which will help emergency response teams to identify the needs of the area.

Response Logging: All user responses shall be logged in real-time to a central database.

Non-Response Management: Users who fail to respond within a designated timeframe will be automatically flagged as potentially unsafe, prompting additional actions.

4. Emergency Contact and Response Team Notification

Automatic Notifications to Emergency Contacts: If users report being unsafe or do not respond, the system will notify designated emergency contacts with relevant details (e.g., username, location).

Response Team Alerts: Emergency response teams shall receive real-time notifications with user information to facilitate swift action.

5. User Profile and Emergency Contact Management

Profile Creation: Users should create profiles with basic information (e.g., name, address) and set emergency contacts within the application.

Contact Management: Users shall have the ability to add, modify, or remove emergency contacts.

Data Synchronization: Profile and contact information shall be updated and synchronized in the central system.

6. Real-Time Monitoring and Dashboard Display

Centralized Dashboard: A web-based dashboard shall display live sensor data and user statuses, offering emergency responders a comprehensive view of the situation.

Geolocation Tracking: The dashboard shall include a map interface to visually indicate user locations and disaster-affected areas.

7. Automated Safety Escalation and Follow-Up

Escalation for Non-Responding Users: If users fail to respond within a set period, they will be flagged as unresponsive, and their emergency contacts will be notified.

Automated Re-Notification: The system may periodically resend “Are you safe?” prompts to unresponsive users when necessary.

8. Disaster Simulation and System Testing

Scenario Simulation: Administrators shall be able to simulate disaster events to assess notification and response features.

Performance Metrics: During simulations, the system shall log metrics (e.g., notification delivery speed, response times) to facilitate optimization.

9. Location-Based Alerts and Geofencing

Geofencing for Targeted Alerts: The system shall use geolocation to identify users in affected areas, targeting alerts accordingly.

Location Verification: Accurate location tracking shall be employed to enhance alert relevance and precision.

10. Incident History and Reporting

Event Logging: The system shall log disaster events, notifications sent, and user responses for future reference.

User Access to Incident History: Users shall have access to a history of incidents and responses within their app profiles.

Administrative Reporting: System administrators shall generate reports on performance, user responses, and incident summaries.

11. Offline Functionality and Data Synchronization

Local Data Storage: Essential user data (e.g., profile, emergency contacts) shall be stored locally on mobile devices to support low-connectivity scenarios.

Offline Safety Updates: Users shall be able to update their safety status offline, with responses syncing to the central server when connectivity is restored.

12. User Notifications and Feedback

Status Confirmation: Users shall receive confirmations upon submitting their safety responses.

Periodic Check-Ins: For prolonged disaster events, the system may periodically prompt users to reaffirm their status.

13. Security and Data Privacy

Data Encryption: All user data, including profiles and location information, shall be encrypted during transmission and while stored.

Access Control: Sensitive data shall only be accessible to authorized users and emergency responders.

Audit Trails: The system shall log all data access and modifications for accountability and auditing.

14. Accessibility and Device Compatibility

Cross-Platform Support: The mobile app shall be available for both Android and iOS to ensure broad accessibility.

Accessibility Options: Both mobile and web applications shall incorporate accessibility features, such as screen readers and high-contrast modes, for visually impaired users.

Battery Optimization: The app shall be designed to minimize battery usage, particularly during extended disaster scenarios.

15. Predictive Analysis and Early Warnings

Real-Time Analysis: The system shall analyze sensor data continuously to identify patterns indicative of escalating disaster scenarios.

Predictive Alerts: If emerging data suggests a high probability of escalation, the system shall issue early warnings to users and response teams.

16. Redundancy and Failover Support

Data Backup and Recovery: Critical data shall be backed up regularly to prevent loss during outages.

Failover Servers: The system shall switch to backup servers during downtimes to maintain operational continuity.

Sensor Network Redundancy: In case of sensor failure, the system shall rely on adjacent sensors to sustain monitoring.

17. Integration with External Emergency Systems

API Access: The system shall provide APIs allowing external entities (e.g., government and emergency services) to access relevant data.

Automated Data Sharing: Upon disaster confirmation, data shall be automatically shared with external agencies to enhance coordination.

2.3 Non-functional Requirements

1. Performance

Response Time: The system shall process an alert from a sensor-earthquake tremor, water level rise, or fire detection-and notify within 5 seconds. This ensures that in case of any emergency, the users will be communicated on time.

Scalability: The system shall support concurrent users during any disaster scenario without any degradation in performance. This covers real-time notifications of a large number of users based on sensor triggers.

Handling Loads: The backend has to bear enormous loads, especially with respect to SMS requests, sensor inputs, and emergency notifications during peak times of disaster incidents.

2. Reliability

System Availability: It should guarantee the system's availability during the time of an emergency and ensure that backup systems are provided against failure. This aspect is highly critical since emergency communication must be continuously available during times of disaster.

Fault tolerance: This system should function in such a way where its operations would not be jeopardized by partial failures of its components. For instance, when the main notification mechanism fails, then it must not be the case that notifications cease; rather, it falls to an alternative means of communication, such as SMS, to reach out to users and emergency services.

Data Consistency: The information of the users regarding their emergency contact and safety status should always be consistent in case of any system failure or recovery.

3. Security

Data Privacy: Users' information, especially regarding emergency contacts and the status of health, needs to be kept secure. It needs to be encrypted in transit and also at rest to avoid unauthorized access.

Access Control: Access to the administrative functions should be strictly provided only to the authorized personnel of the system. In other words, while the status of the user can, for instance, be viewed by emergency services, such services should not alter the information about users.

This would involve authentication and authorization, where the users and emergency personnel would authenticate to the system using secure mechanisms such as two-factor or multi-factor authentication in order to make sure sensitive data is kept secure.

4. Usability

Usability: The UI of the mobile application needs to be intuitive and easy to use; respond quickly to the requests of safety status, and in high-stress situations, send alerts to emergency services.

Accessibility: The application should be made accessible for people with disabilities. An example of this could be voice commands or enlargement of the font to support visually impaired users.

Multilingual Support: The system shall be able to support various languages to assist a massive amount of people in disaster situations, enabling them to read and act accordingly in native languages.

5. Maintainability

Modularity: This means the system design should be such that it is modular. That way, when perhaps one component, say a sensor, notification service, or user management, will need updating or replacement, it would not have to affect the whole system.

Code Quality: The codebase should follow best practices to ensure readability, maintainability, and scalability in order to easily extend or modify it in the future when other disaster scenarios, like tornadoes and tsunamis, will be added.

Emergency Response Regulations: In relation to notification and coordination between the public and emergency services, the system shall be in line with the relevant government and disaster management regulations.

6. Environmental Requirements

Energy Efficiency: All sensors and mobile applications are to be designed in such a way that they are power-effective; specifically, at times of emergency that lasts for a duration period their operation should continue without the need for frequent recharging.

Resilience under Harsh Conditions: These sensors should be weatherproof; they should function at very high temperatures or humidity and even after calamities like fires or floods.

2.4 Pseudo Requirements

General System Requirements

- **Sensor Data Management:** Collect data from **seismic** wave sensor, **temperature**, and **water level** sensors through Arduino with Wi-Fi module, and send this data to the central system using Firebase.
- **Notification and Emergency Management:** In case of an emergency detected by sensors, a notification will be sent to mobile users, asking if they are safe. If a user indicates they are not safe, their information will be relayed to relevant emergency services.
- **User Management:** Users can register, add emergency contacts, and update their profiles through both web and mobile platforms.
- **Timeout Response:** If a user does not respond within a specific timeout period or reports as “Not Safe,” the system will automatically notify the user’s emergency contacts.

Mobile Application Requirements

1. User Authentication

- a. **Login:** Users should be able to log into the app using email and password.
- b. **Register:** New users can create an account with their name, email, and phone number.

2. Emergency Contact Management

- a. **Add Contacts:** Users should be able to add emergency contacts with name, phone number, and email.
- b. **Remove Contacts:** Users should be able to delete previously added contacts.

3. Notifications

- a. **Receive Emergency Notification:** In case of an emergency, users will receive a “Are you safe?” notification.
- b. **Send Response:** Users can mark themselves as “Safe” or “Not Safe,” which will then update their status on the system.

4. Profile Management

- a. **Edit Profile:** Users should be able to update their profile details, including contacts, phone number, and email.

Web Application Requirements

1. User Authentication

- a. **Login:** Users should be able to log into the web application using their email and password.
- b. **Register:** New users should be able to sign up through the web portal.

2. Sensor Data Evaluation

- a. **Track Sensor Notifications:** Notifications sent by sensors (through Arduino and Wi-Fi module) will be displayed on the system and reviewed by operators.
- b. **Emergency Alert Creation:** If sensor data exceeds threshold limits, it should trigger an emergency alert in the system.

3. Monitoring Mobile Responses

- a. **Track Safety Status:** Responses from users (safe/not safe) will be displayed on a monitor screen.
- b. **Unsafe User Details:** For users marked as “Not Safe,” display details like name, location, and phone number on the monitor screen.

4. Timeout and No-Response Handling

- a. **Emergency Contact Notification:** If no response is received within the timeout period, or if a user is marked as “Not Safe,” an automated alert will be sent to the user’s emergency contacts.

5. User Management

- a. **List Registered Users:** The web application should allow viewing and managing registered users.

Technology Requirements

1. **React:** Front-end for the web application.
2. **React Native:** Front-end for the mobile application.
3. **Node.js:** Back-end development, API, and database management.
4. **JavaScript:** For core functionality and interactions across web and mobile applications.
5. **PostgreSQL:** Relational database for storing user and notification data.
6. **Firebase:** Database and real-time data syncing for managing sensor information and user responses.
7. **Arduino + Wi-Fi Module:** Collect sensor data and transmit it to Firebase.
8. **Domain:** For hosting the web application, accessible publicly.

2.5 System Models

2.5.1 Scenarios

Scenario 1- Seismic Wave Sensor Detects Earthquake (User Safe)

Situation	The seismic wave sensor detects an earthquake.
System	Earthquake notification is sent to the center and mobile applications.
User Response	Reports being safe.
Result	No action is taken; the user is marked as safe.

Scenario 2 - Seismic Wave Sensor Detects Earthquake (User Not Safe)

Situation	The seismic wave sensor detects an earthquake.
System	Earthquake notification is sent to the center and mobile applications.
User Response	Reports not being safe.
Result	The user's location and contact information are sent to emergency response teams.

Scenario 3 - Fire Sensor Detects Hazard

Situation	The fire sensor detects a hazard.
System	Fire notification is sent to the center and mobile applications.
User Response	Does not respond.
Result	The user is marked as unsafe; information is sent to relatives and emergency response teams.

Scenario 4 - Water Level Sensor Reaches Overflow Level (User Safe)

Situation	The water level sensor reaches overflow level.
System	Flood notification is sent to the center and mobile applications.
User Response	Reports being safe.
Result	No action is taken; the user is marked as safe.

Scenario 5 - Water Level Sensor Reaches Overflow Level (User Not Safe)

Situation	The water level sensor reaches overflow level.
System	Flood notification is sent to the center and mobile applications.
User Response	Reports not being safe.
Result	The user's information is sent to emergency response teams; relatives are also notified.

Scenario 6 - Earthquake Notification and Waiting for User Response

Situation	Earthquake notification is sent.
System	Wait for a response from the user within a set time.
User Response	Does not respond.
Result	The user is marked as unsafe; relatives and emergency teams are notified.

Scenario 7 - Fire Hazard Detected (Multiple Family Users)

Situation	Fire hazard is detected.
System	Fire notification is sent to three users from the same family.
User Response	Two report being safe, one reports not being safe.
Result	The information of the user not being safe is shared with teams, and the other two users are marked as safe.

Scenario 8 - Water Level Sensor Reaches Overflow Level (Child User Safe)

Situation	The water level sensor reaches overflow level.
System	A notification is sent to a child user.
User Response	Reports being safe.
Result	No action is taken; the user is marked as safe.

Scenario 9 - Earthquake Detected (Disabled User Does Not Respond)

Situation	Earthquake is detected.
System	Earthquake notification is sent to a disabled user.
User Response	Does not respond.
Result	The user is marked as unsafe; relatives and teams are notified.

Scenario 10 - Flood Risk but User Logged Out

Situation	There is a flood risk, but the user has logged out of the system.
System	Disaster notification cannot be sent.
User Response	-
Result	The user's relatives are informed of the situation.

Scenario 11 - Disaster Over and All Users Safe

Situation	The disaster is over.
System	All users in the area are notified that the situation has returned to normal.
User Response	Reports being safe after the disaster.
Result	The system updates the status; all users are marked as safe.

Scenario 12 - Earthquake and Dangerous Water Level Aftershock (User Safe)

Situation	After the earthquake, the water level reaches a dangerous level.
System	Separate notifications for both disasters are sent to the user.
User Response	Reports being safe for both disasters.
Result	The system updates the status; the user is marked as safe.

Scenario 13 - User Learns a Relative Is Not Safe During Disaster

Situation	During the disaster, the user learns that one of their relatives is not safe.
System	The information is presented to the user.
User Response	-
Result	The system sends information to relatives and initiates emergency intervention.

Scenario 14 - Earthquake Notification (User Safe and Updates Location)

Situation	Earthquake notification is sent.
System	Wait for a response from the user.
User Response	Reports being safe and updates their location.
Result	The current location is recorded; the user is marked as safe.

Scenario 15 - Flood Risk Notification (User Does Not Respond)

Situation	Flood risk is notified.
System	The user does not respond.
User Response	-
Result	The user is marked as unsafe; a notification is sent to relatives.

Scenario 16 - Seismic Wave Sensor Detects Earthquake (User Safe)

Situation	The seismic wave sensor detects an earthquake.
System	Earthquake notification is sent to the center and mobile applications.
User Response	Reports being safe.
Result	No action is taken; the user is marked as safe.

Scenario 17 - Water Level Sensor Reaches Overflow Level (User Not Safe)

Situation	The water level sensor reaches overflow level.
System	Flood notification is sent to the center and mobile applications.
User Response	Reports not being safe.
Result	The user's information is sent to emergency response teams; relatives are also notified.

Scenario 18 - Fire Sensor Detects Hazard (User Does Not Respond)

Situation	The fire sensor detects a hazard.
System	Fire notification is sent to the center and mobile applications.
User Response	Does not respond.
Result	The user is marked as unsafe; information is sent to relatives and emergency response teams.

Scenario 19 - Earthquake Notification (No User Response)

Situation	Earthquake notification is sent.
System	Wait for a response from the user within a set time.
User Response	Does not respond.
Result	The user is marked as unsafe; relatives and emergency teams are notified.

Scenario 20 - Fire Hazard Detected (Multiple Family Users, One Not Safe)

Situation	Fire hazard is detected.
System	Fire notification is sent to three users from the same family.
User Response	Two users report being safe, one reports not being safe.
Result	The information of the user not being safe is shared with teams, and the other two users are marked as safe.

2.5.2 Use Case Model

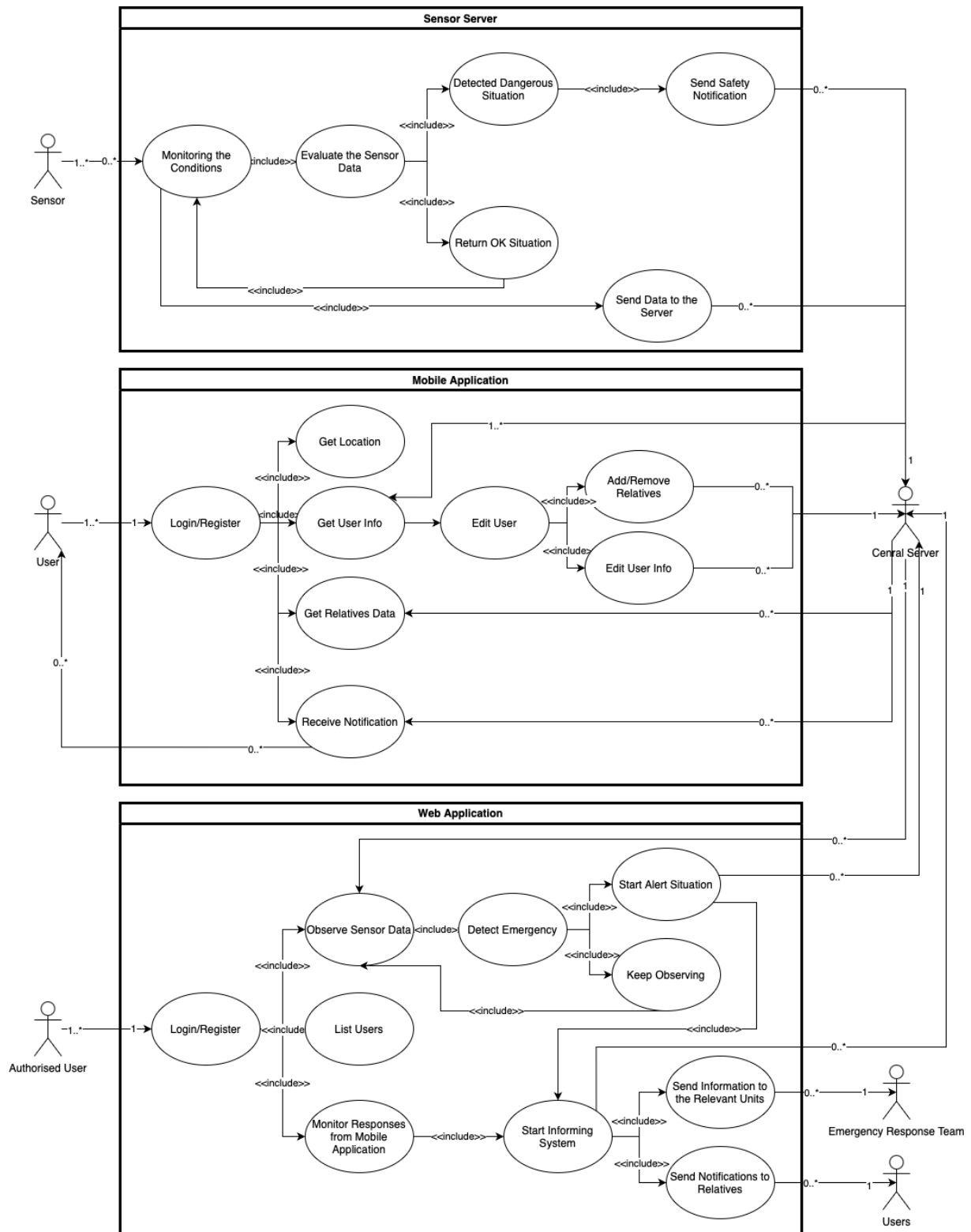


Figure 2 - Use Case Diagram

2.5.3 Object and Class Model

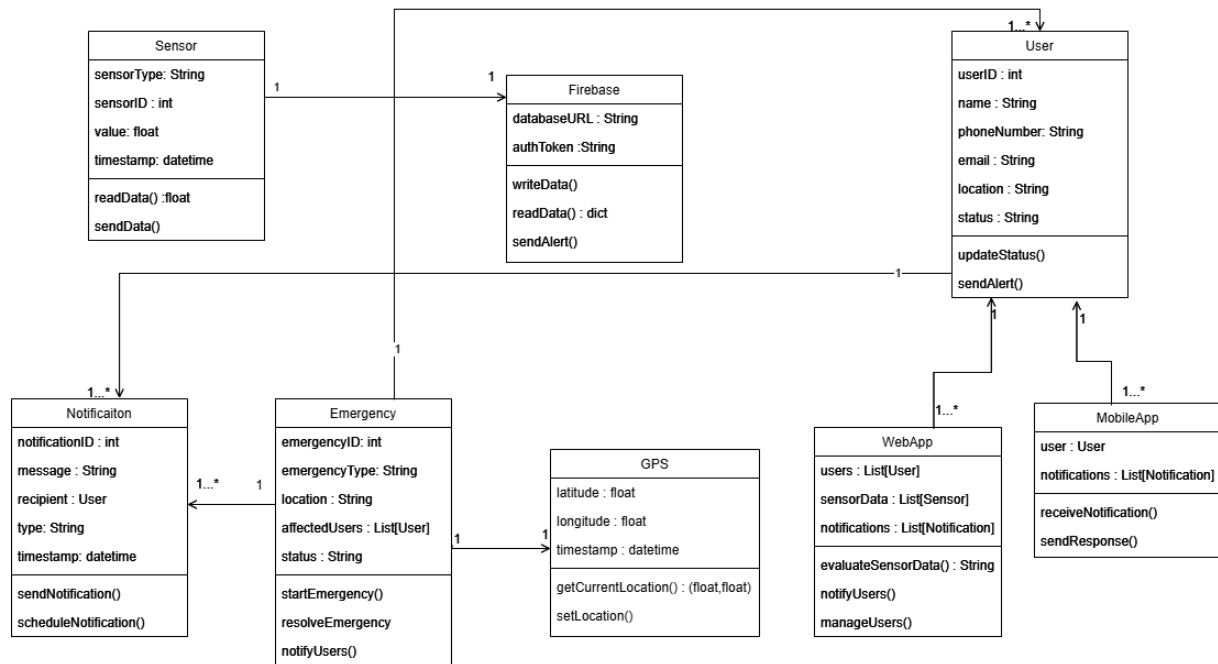


Figure 3 - Class Diagram

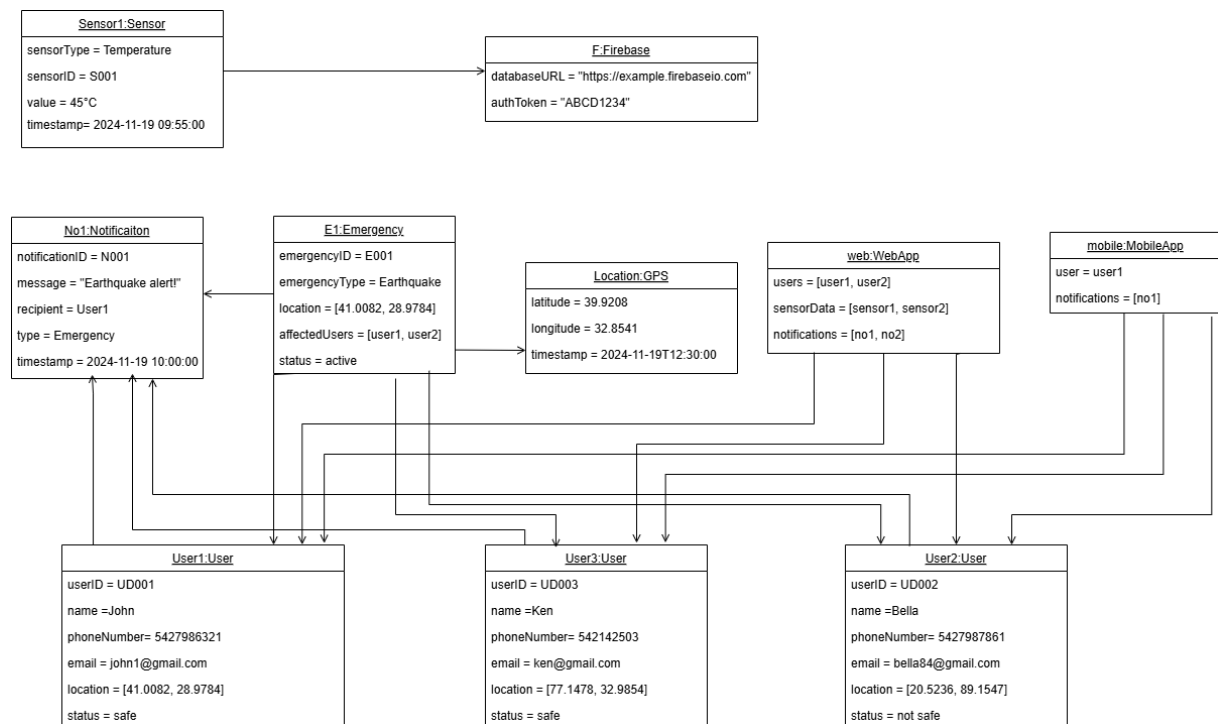


Figure 4 - Object Diagram

2.5.4 Dynamic Models

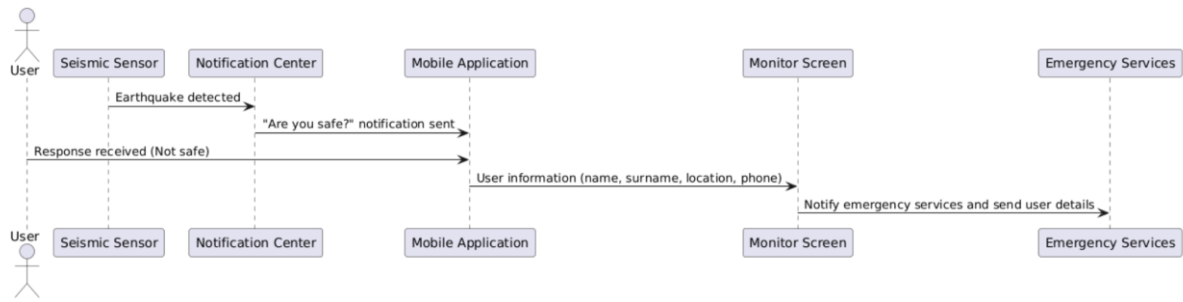


Figure 5 - Sequence Diagram I

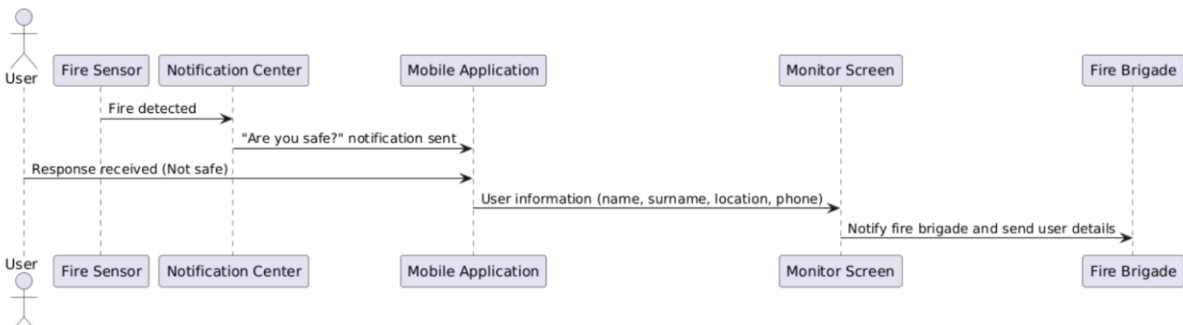


Figure 6 - Sequence Diagram II

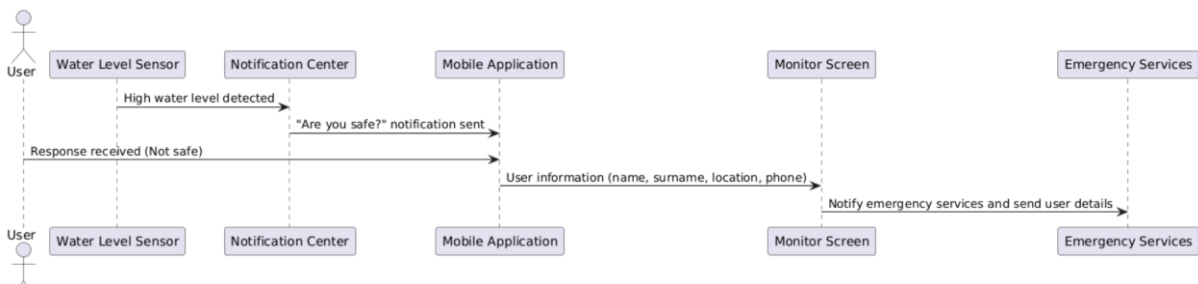


Figure 7 - Sequence Diagram III

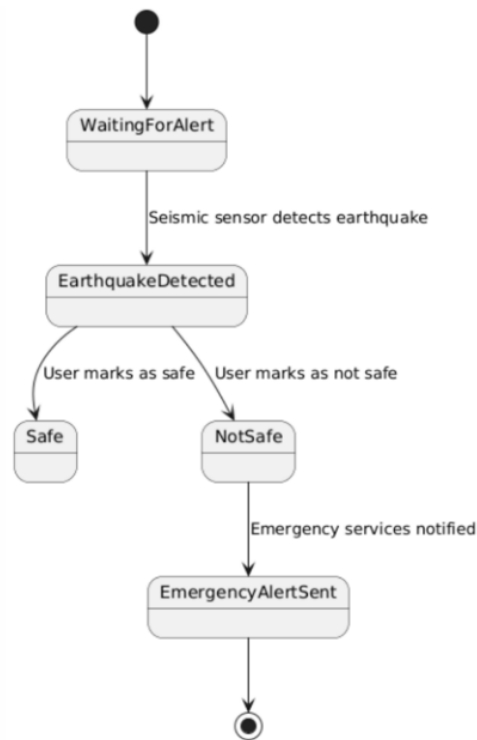


Figure 8 - State Diagram for Earthquake Alert

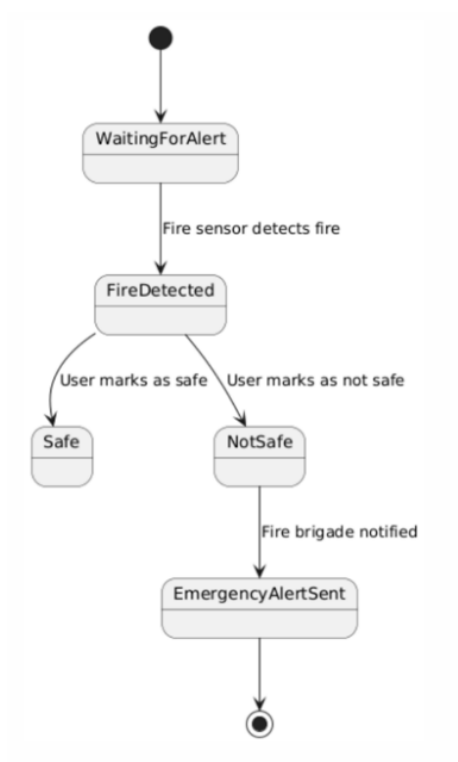


Figure 9 - State Diagram for Fire Alert

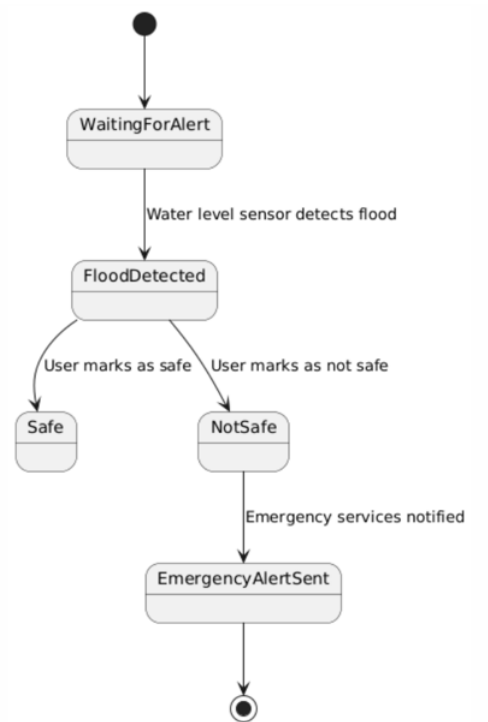
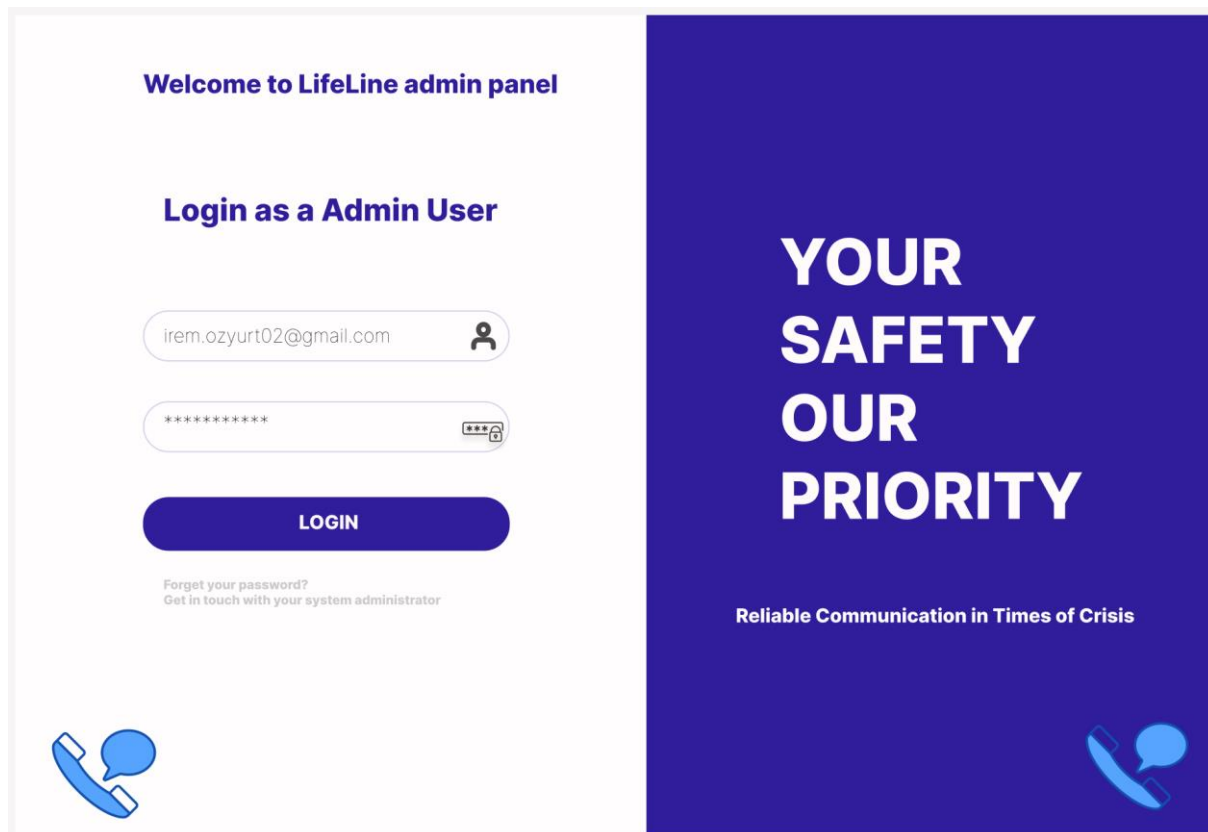


Figure 10 - State Diagram for Flood Alert

2.5.5 User interface - Navigational Paths and Screen Mock-ups



The mock-up for the login page is divided into two main vertical sections. The left section has a white background and contains the following elements: a header 'Welcome to LifeLine admin panel' in blue; a sub-header 'Login as a Admin User' in blue; an email input field with the placeholder 'irem.ozyurt02@gmail.com' and a user icon; a password input field with masked characters '*****' and a lock icon; a blue 'LOGIN' button; and a link 'Forget your password? Get in touch with your system administrator' in small grey text. The right section has a solid blue background and contains the text 'YOUR SAFETY OUR PRIORITY' in large white letters, followed by 'Reliable Communication in Times of Crisis' in smaller white text. Both sections feature a blue telephone handset icon in the bottom left corner.

Welcome to LifeLine admin panel

Login as a Admin User

irem.ozyurt02@gmail.com

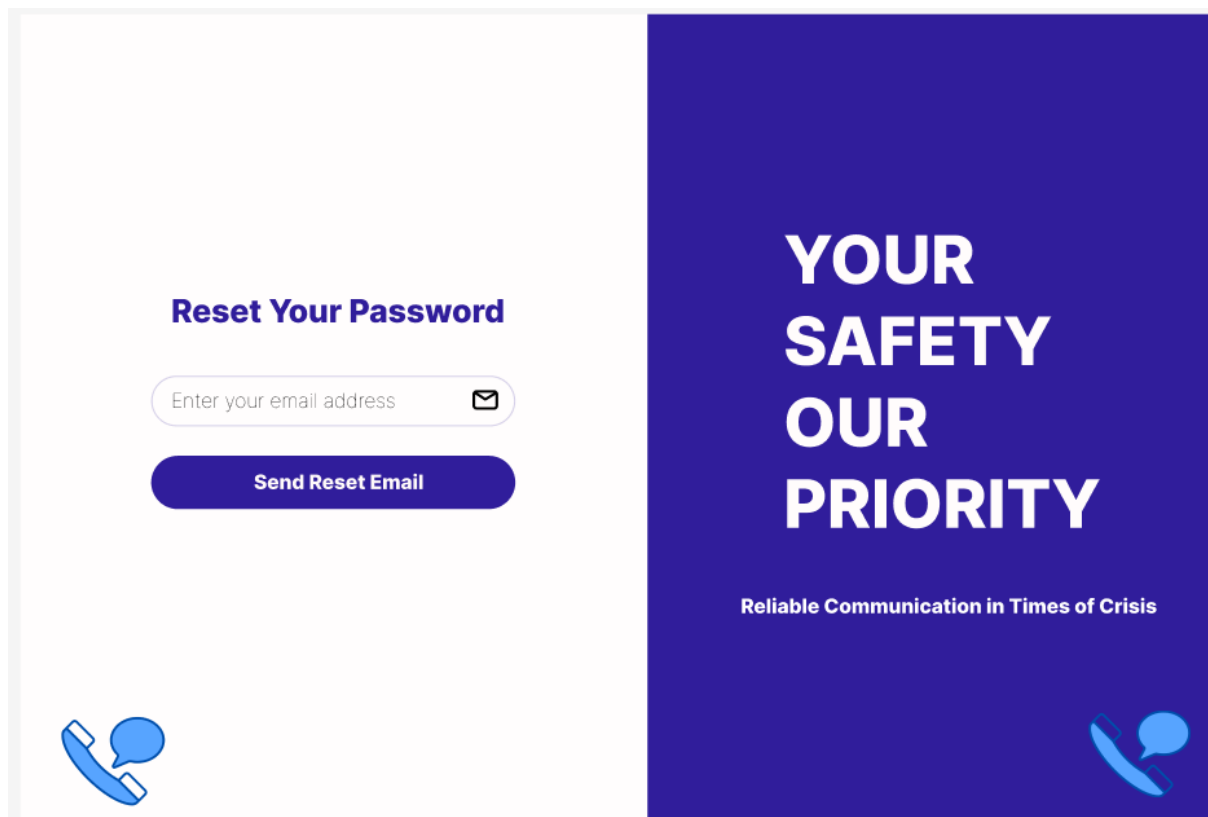
LOGIN

Forget your password?
Get in touch with your system administrator

YOUR SAFETY
OUR
PRIORITY

Reliable Communication in Times of Crisis

Figure 11 - Website Login Page Mock-up



The mock-up for the reset password page follows the same layout as the login page. The left white section contains: a sub-header 'Reset Your Password' in blue; an email input field with the placeholder 'Enter your email address' and an envelope icon; a blue 'Send Reset Email' button; and the same 'Forget your password?' link in small grey text. The right blue section contains the same large white text 'YOUR SAFETY OUR PRIORITY' and 'Reliable Communication in Times of Crisis', with the blue telephone handset icon in the bottom left corner.

Reset Your Password

Enter your email address

Send Reset Email

YOUR SAFETY
OUR
PRIORITY

Reliable Communication in Times of Crisis

Figure 12 - Website Reset Password Page Mock-up

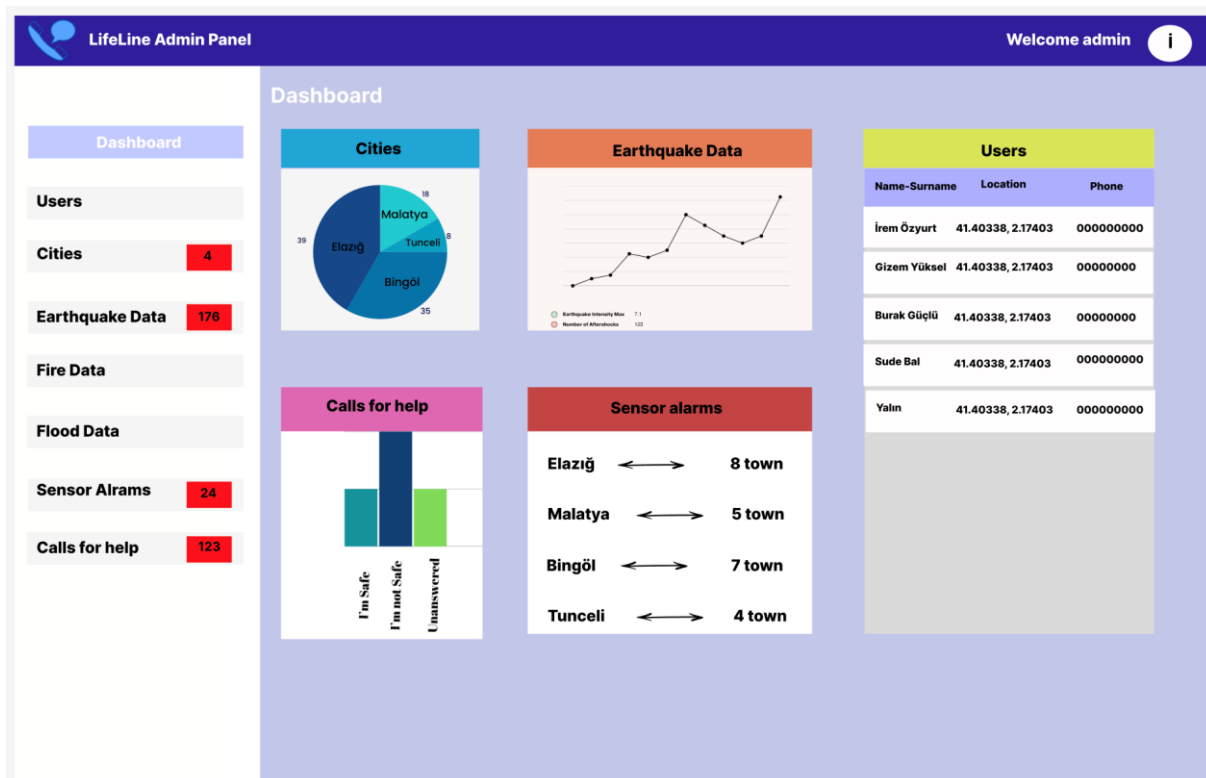


Figure 13 - Website Dashboard Mock-up

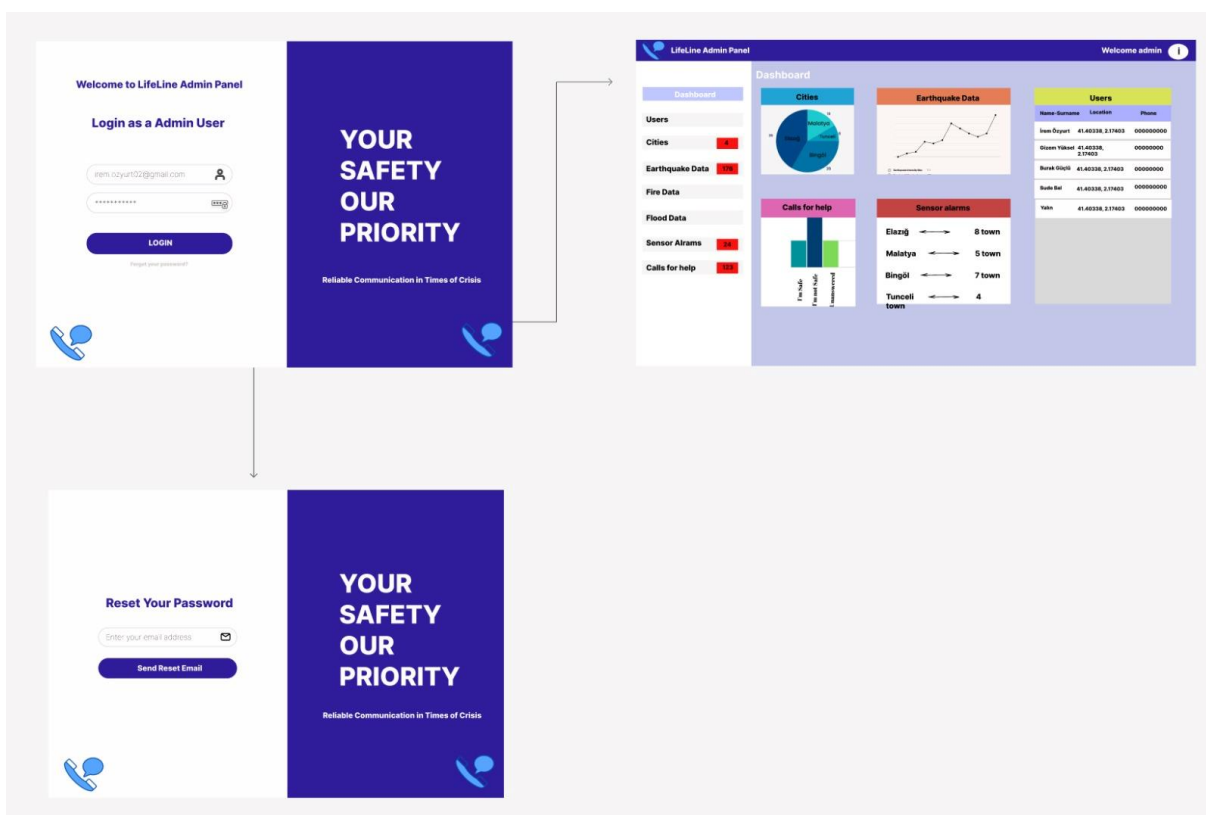


Figure 14 - Navigational Path for Website

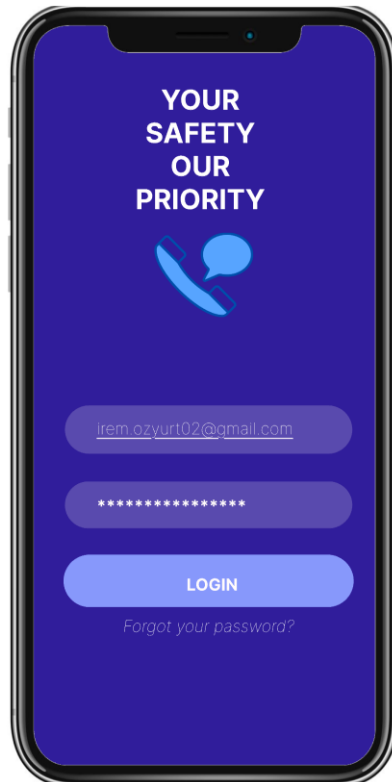


Figure 15 - Mobile App Login Screen Mock-up

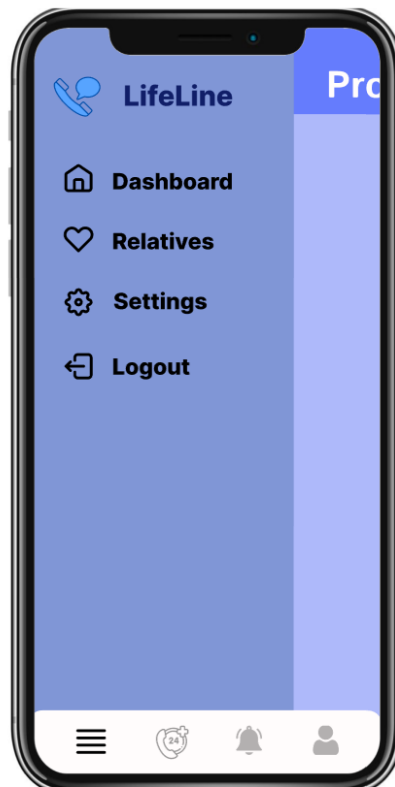


Figure 16 - Mobile App Menu Screen Mock-up

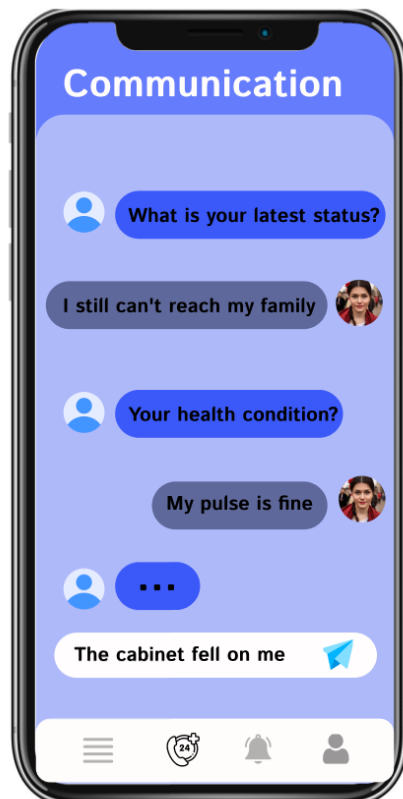


Figure 17 - Mobile App Communication Screen Mock-up



Figure 18 - Mobile App Notifications Screen Mock-up

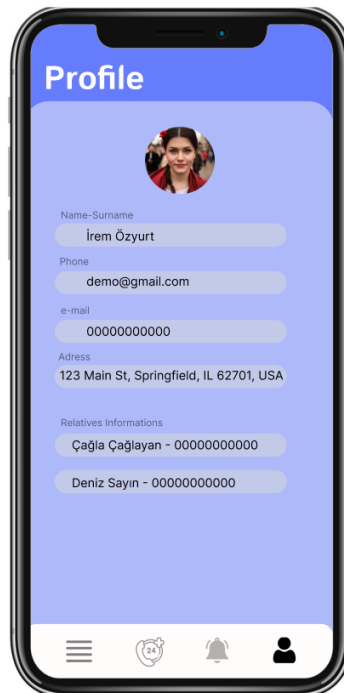


Figure 19 - Mobile App Profile Screen Mock-up



Figure 20 - Navigational Path for Mobile App

3. Glossary

Term	Definition
IoT (Internet of Things)	The technology that allows physical devices to connect to the internet to collect and share data.
Emergency Alert System	The system is used to quickly share information and send warnings during natural disasters.
Disaster Response Teams	Teams that provide rapid intervention and assistance during natural disasters.
Cloud Storage	Storage of data on remote servers accessible over the internet.
Cybersecurity	Method of protecting digital systems against unauthorized access and threats.
Safety Confirmation	Notification of whether individuals are safe during a disaster.
Sensor Data	Data obtained from the physical environment.
Timeout	Automatic interruption that occurs when no action is taken within a certain period of time.
Dynamic Models	Models that represent system behaviors and processes in ways that change over time.
Functional Requirements	Basic functions that a system must perform.
Nonfunctional Requirements	Criteria that specify how a system will operate, such as performance and security.
User Interface	Visual and functional components through which users interact with a system.

4. References

- [1] Ahi, G. O., & Canpolat, B. (2021). Akıllı Telefonda Derin Öğrenme ile Deprem Erken Uyarı Sistemi Tasarımı. *Avrupa Bilim ve Teknoloji Dergisi*, (25), 23-27.
- [2] Chandrakumar, C., Prasanna, R., Stephens, M., & Tan, M. L. (2022). Earthquake early warning systems based on low-cost ground motion sensors: A systematic literature review. *Frontiers in Sensors*, 3, 1020202.
- [3] Chen, F. H., Shieh, H. L., & Tu, J. F. (2023). Development of Earthquake Detection and Warning System Based on Sensors. *Sensors & Materials*, 35.
- [4] Karacı, A. (2018). IoT-based earthquake warning system development and evaluation. *Mugla Journal of Science and Technology*, 4(2), 156-161.
- [5] Ray, N. K., & Turuk, A. K. (2017). A framework for post-disaster communication using wireless ad hoc networks. *Integration*, 58, 274-285.
- [6] Deepak, G. C., Ladas, A., Sambo, Y. A., Pervaiz, H., Politis, C., & Imran, M. A. (2019). An overview of post-disaster emergency communication systems in the future networks. *IEEE Wireless Communications*, 26(6), 132-139.
- [7] Matracia, M., Saeed, N., Kishk, M. A., & Alouini, M. S. (2022). Post-disaster communications: Enabling technologies, architectures, and open challenges. *IEEE Open Journal of the Communications Society*, 3, 1177-1205.