****

**REPUBLIC OF CAMEROON**

**Peace-Work-Fatherland**

**MINISTER OF HIGHER EDUCATION**

**FACULTY OF ENGINEERING**

**AND TECHNOLOGY**

**REPUBLIQUE DU CAMEROON**

**PAIX-Travail-Patrie**

**MINISTRE DE L’ENSEIGNEMENT SUPERIEUR**

**FACULTE DE L’ENGINERIE**

**ET TECHGNOLOGIE**

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** UNIVERSITY OF BUEA *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

**DEPARTMENT OF COMPUTER ENGINEERING**

**COURSE: INTERNET PROGRAMMING (J2EE) AND MOBILE PROGRAMMING**

**COURSE CODE: CEF 440**

**DESIGN AND IMPLEMENTATION OF A MOBILE-BASED DISASTER MANAGEMENT SYSTEM**

**PRESENTED BY:**

GROUP 20

**INSTRUCTOR:** **DR. VALERY JULY 2024**

**GROUP MEMBERS**

|  |  |  |
| --- | --- | --- |
| S/N | **Name** | Matricule Number |
| 1 | DUESENBERRY MBIKANG AGBORTAR AKO | FE21A175 |
| 2 | FOTABONG FUALEFAC PUISSANCE | FE21A195 |
| 3 | ABANGMA ARRAH JESUS MCLOUIS | FE21A124 |
| 4 | TABOT JOEL EBANGHA | FE21A308 |
| 5 | EBOMESUMBE NGOLE DANDY BRADLEY | FE21A177 |

Table of Contents

[**TABLE OF FIGURES** 6](#_Toc171303635)

[**CERTIFICATION** 7](#_Toc171303636)

[**ACKNOWLEDGEMENT** 8](#_Toc171303637)

[**ABSTRACT** 9](#_Toc171303638)

[**CHAPTER 1. INTRODUCTION** 10](#_Toc171303639)

[**1.1 BACKGROUND** 10](#_Toc171303640)

[**1.2 PROBLEM STATEMENT** 10](#_Toc171303641)

[**1.3 OBJECTIVES OF THE STUDY** 10](#_Toc171303642)

[**1.4 METHODOLOGY** 11](#_Toc171303643)

[**1.5 SIGNIFICANCE OF THE STUDY** 11](#_Toc171303644)

[**CHAPTER 2: REQUIREMENT GATHERING** 12](#_Toc171303645)

[**2.1 VICTIMS** 12](#_Toc171303646)

[**2.1.1 Problems Faced** 12](#_Toc171303647)

[**2.1.2 Types of Assistance Sought** 12](#_Toc171303648)

[**2.1.3 Features Considered Useful** 12](#_Toc171303649)

[**2.2 RESPONDENTS** 13](#_Toc171303650)

[**2.2.1 Respondents' Specialties in Case of a Disaster** 13](#_Toc171303651)

[**2.2.2 Features to Enhance in the System** 13](#_Toc171303652)

[**2.2.3 Challenges Faced** 13](#_Toc171303653)

[**2.3 AUTHORITIES** 14](#_Toc171303654)

[**2.3.1 Features Prioritized in the Disaster Management App** 14](#_Toc171303655)

[**2.3.2 Challenges Faced** 14](#_Toc171303656)

[**CHAPTER 3: REQUIREMENT ANALYSIS** 15](#_Toc171303657)

[**3.1 REQUIREMENT GATHERING** 15](#_Toc171303658)

[**3.1.1 Primary Stakeholders for the disaster management system** 15](#_Toc171303659)

[**3.1.2 Goals of the Disaster Management System** 15](#_Toc171303660)

[**3.1.3 Problems faced** 15](#_Toc171303661)

[**3.1.4 Need for a New System** 15](#_Toc171303662)

[**3.2 FUNCTIONALITIES** 16](#_Toc171303663)

[**3.2.1 Functional Requirements** 16](#_Toc171303664)

[**3.2.2 Non-Functional Requirements** 16](#_Toc171303665)

[**3.3** **USER ROLES AND PERMISSIONS** 17](#_Toc171303666)

[**3.4 DATA REQUIREMENTS** 20](#_Toc171303667)

[**3.5 SYSTEM INTERFACES** 20](#_Toc171303668)

[**3.6 REPORTING AND ANALYTICS** 20](#_Toc171303669)

[**3.6.1 Importance of Reporting and Analytics in Disaster Management** 20](#_Toc171303670)

[**3.6.2 Types of Reports** 20](#_Toc171303671)

[**3.6.3 Dashboards and Visualizations** 21](#_Toc171303672)

[**3.6.4 Disaster-Related Data** 21](#_Toc171303673)

[**3.6.5 Performance Metrics and Key Performance Indicators (KPIs)** 21](#_Toc171303674)

[**3.7 CONSTRAINTS** 21](#_Toc171303675)

[**3.7.1 Technical Constraints** 21](#_Toc171303676)

[**3.7.2 Time Constraints** 21](#_Toc171303677)

[**3.7.3 Budget Constraints** 22](#_Toc171303678)

[**3.7.4 Organizational Constraints** 22](#_Toc171303679)

[**3.8 DEPENDENCIES** 22](#_Toc171303680)

[**3.9 ACCEPTANCE CRITERIA** 22](#_Toc171303681)

[**CHAPTER 4: SYSTEM MODELLING AND DESIGN** 23](#_Toc171303682)

[**4.1. CONTEXT DIAGRAM** 23](#_Toc171303683)

[**4.1.1 Actors and Their Roles** 23](#_Toc171303684)

[**4.2 USE CASE DIAGRAM** 24](#_Toc171303685)

[**4.2.1 Actors and Their Roles** 24](#_Toc171303686)

[**4.3 SEQUENCE DIAGRAM** 25](#_Toc171303687)

[**4.3.1 Actors and Their Roles** 25](#_Toc171303688)

[**4.4 CLASS DIAGRAM** 26](#_Toc171303689)

[**4.4.1 Actors and Their Roles** 28](#_Toc171303690)

[**4.5 DEPLOYMENT DIAGRAM** 28](#_Toc171303691)

[**4.5.1Actors and Their Roles** 29](#_Toc171303692)

[**CHAPTER 5: UI DESIGN AND IMPLEMENTATION** 30](#_Toc171303693)

[**5.1 UI DESIGN PROCESS** 30](#_Toc171303694)

[**5.2 FIGMA AS A DESIGN TOOL** 30](#_Toc171303695)

[**5.3 SCREENS** 31](#_Toc171303696)

[**5.6 UI Implementation Strategy** 37](#_Toc171303697)

[**5.7 Coding** 37](#_Toc171303698)

[**CHAPTER 6: DATABASE DESIGN AND IMPLEMENTATION** 38](#_Toc171303699)

[**6.1. CONCEPTUAL DESIGN** 38](#_Toc171303700)

[**6.1.1 Entities Identified:** 38](#_Toc171303701)

[**6.2. LOGICAL DESIGN** 39](#_Toc171303702)

[**6.2.2 Schema Definition:** 39](#_Toc171303703)

[**6.3. PHYSICAL DESIGN** 40](#_Toc171303704)

[**6.3.1 Physical Storage Considerations:** 41](#_Toc171303705)

[**6.3.2 Database Security Design:** 41](#_Toc171303706)

[**6.4. SQL DATABASE IMPLEMENTATION** 41](#_Toc171303707)

[**6.4.1 SQL Scripts:** 41](#_Toc171303708)

[**6.5 RESULTS:** 42](#_Toc171303709)

[**6.6. TECHNOLOGIES USED** 44](#_Toc171303710)

[**6.7. TESTING AND VALIDATION** 45](#_Toc171303711)

[**6.7.1 Testing Strategies:** 45](#_Toc171303712)

[**CHAPTER 7: CONCLUSION** 46](#_Toc171303713)

# **TABLE OF FIGURES**

[Figure 1: Context Diagram 23](#_Toc171303824)

[Figure 2: Use case diagram 24](#_Toc171303825)

[Figure 3: Sequence diagram 25](#_Toc171303826)

[Figure 4: Class Diagram 27](#_Toc171303827)

[Figure 5: Deployment Diagram 28](#_Toc171303828)

[Figure 6: Language screen 31](#_Toc171303829)

[Figure 7: Splash Screen 31](#_Toc171303830)

[Figure 8: user enter details 32](#_Toc171303831)

[Figure 9: Select role 32](#_Toc171303832)

[Figure 10: Get started 32](#_Toc171303833)

[Figure 11: alerts, user dashbaord, emergencies and contacts police screen for the user 33](#_Toc171303834)

[Figure 12: Additional screens for the user 33](#_Toc171303835)

[Figure 13: Screen for user 34](#_Toc171303836)

[Figure 14: Respondents screens 34](#_Toc171303837)

[Figure 15: Additional respondents screens 35](#_Toc171303838)

[Figure 16: Authorities sign up, login, dashboard and emergency screens 35](#_Toc171303839)

[Figure 17: Additional authorities screen 35](#_Toc171303840)

[Figure 18: Additional authorities screens 36](#_Toc171303841)

[Figure 19: Additional authorities screens 36](#_Toc171303842)

# 

# **CERTIFICATION**

This is to certify that the project report titled **“MOBILE APPLICATION FOR A DISASTER MANAGEMENT SYSTEM”** submitted by **GROUP 20 MEMBERS is** in fulfilment with the requirements of the Bachelor’s Degree in Engineering (B.Eng) in **COMPUTER ENGINEERING**, has been carried out under supervision and guidance.

# **ACKNOWLEDGEMENT**

We would like to extend our sincere gratitude to all those who have supported and contributed to the realization of this project. Without your assistance, guidance, and encouragement, this endeavor would not have been possible.

We are deeply grateful to my project supervisor, **DR. VALERY,** for his invaluable guidance, expertise, and unwavering support throughout the duration of this project.

we are indebted to our friends and colleagues for their support, collaboration, and insightful discussions.

we extend our gratitude to all the participants and volunteers who generously contributed their time, expertise, and insights.

We would like to acknowledge the Faculty of Engineering and Technology for providing the necessary resources, facilities, and conducive environment that have facilitated the smooth execution of this project.

**Keywords:**

* Disaster management: Disaster management is the organization and management of resources and responsibilities for dealing with all humanitarian aspects of emergencies.
* Alerts: Updates sent by authorities to users and respondents, mostly disaster which can be predicted
* Emergencies: Updates on disasters sent by different users of the application
* Emergency preparedness tips: tips on what do during, before and after a disaster
* Safety tips: Tips on what to do to avoid man disaster like fires
* First aid tips: Like the name implies, what to incase any injuries are sustained during a disaster.

# **ABSTRACT**

The increasing frequency and severity of natural disasters necessitate advanced solutions for efficient disaster management. This paper presents a comprehensive mobile application designed to enhance disaster preparedness, response, and recovery. The Disaster Management System (DMS) app serves as a multifaceted platform catering to individuals, emergency responders, and authorities. It integrates real-time alerts, safety tips, and emergency contacts to keep users informed and safe. The application features distinct dashboards for users, respondents, and authorities, enabling tailored functionalities such as reporting incidents, monitoring disaster status, and managing resources effectively. By leveraging location-based services, the app provides critical information on safe zones, shelters, and medical facilities. Furthermore, it facilitates communication between affected individuals and emergency services, ensuring prompt and coordinated responses. The DMS app aims to mitigate the impact of disasters by empowering communities with the tools and information needed to navigate emergencies efficiently. Through its user-friendly interface and robust features, the DMS app aspires to set a new standard in disaster management, fostering resilience and preparedness in the face of adversity.

# **CHAPTER 1. INTRODUCTION**

The increasing frequency and severity of natural disasters worldwide have highlighted the critical need for advanced and efficient disaster management solutions. This document outlines the requirements for designing and implementing a mobile-based Disaster Management System (DMS), serving as a comprehensive guide for stakeholders, including the development team, project managers, and potential users of the system. The goal is to provide a clear understanding of the objectives, functionalities, and scope of the DMS, ensuring a coordinated approach to disaster preparedness, response, and recovery.

## **1.1 BACKGROUND**

Recent years have seen a surge in interest and awareness regarding the importance of collective action in managing natural disasters. Communities and individuals recognize that effective disaster response requires seamless coordination and access to real-time information. The process of managing disaster-related activities, however, can be complex and time-consuming. The DMS aims to simplify this process by offering a user-friendly platform that enhances communication, resource allocation, and situational awareness during emergencies.

## **1.2 PROBLEM STATEMENT**

The primary issue that the DMS addresses is the fragmented and inefficient nature of current disaster management efforts. Individuals often struggle to access timely and relevant information, while emergency responders and authorities face challenges in coordinating their actions and mobilizing resources effectively. The DMS provides a centralized platform where users can receive real-time alerts, access safety tips, report incidents, and connect with emergency services, ensuring a unified and efficient response to disasters.

## **1.3 OBJECTIVES OF THE STUDY**

The objective of this project is to develop a mobile application that facilitates disaster management by connecting individuals, emergency responders, and authorities. The system aims to enhance disaster preparedness, streamline response efforts, and support recovery processes, ultimately reducing the impact of disasters on communities.

## **1.4 METHODOLOGY**

The development of the DMS is guided by principles of mobile application development and cloud computing. The solution involves using React Native for building a cross-platform mobile application, ensuring accessibility on both iOS and Android devices. Firebase will be utilized for backend services, including real-time data handling and database management. This approach guarantees scalability, efficiency, and a seamless user experience, enabling the platform to support diverse disaster management needs.

## **1.5 SIGNIFICANCE OF THE STUDY**

The DMS project holds significant importance in several key areas:

1. Enhancing Disaster Preparedness: By providing a centralized platform, the DMS simplifies access to vital information and resources, encouraging individuals to be better prepared for potential disasters.

2. Streamlining Response Efforts: The system empowers emergency responders and authorities to mobilize resources effectively and coordinate their actions, leading to a more efficient and timely disaster response.

3. Supporting Community Resilience: The platform fosters community resilience by facilitating communication and collaboration among individuals, responders, and authorities, strengthening local communities' ability to withstand and recover from disasters.

4. Promoting Awareness and Education: The DMS includes features such as safety tips and educational content, helping raise awareness about disaster risks and the necessary steps to mitigate them.

This document serves as a blueprint for developing a robust and comprehensive mobile Disaster Management System, aiming to create a safer and more resilient world in the face of natural disasters.

# **CHAPTER 2: REQUIREMENT GATHERING**

## **2.1 VICTIMS**

### **2.1.1 Problems Faced**

* Lack of Awareness: Victims were unaware of appropriate actions, safety protocols, available resources, and ongoing response operations, leading to confusion, panic, and potential harm.
* Network Issues: Network-related challenges hindered communication, access to information, and overall functionality during disasters.
* Lack of Finance and Needs: Financial challenges severely impacted victims' ability to secure basic necessities, access healthcare, and recover from the disaster.

### **2.1.2 Types of Assistance Sought**

* Medical Assistance: Victims faced challenges in accessing timely medical support due to disrupted transportation, damaged healthcare infrastructure, and overwhelmed medical facilities.
* Safety Tips: Lack of information on safety tips and guidelines hindered victims' ability to protect themselves and minimize risks during and after disasters.
* Evacuation Routes: Poor planning and lack of access to accurate and up-to-date evacuation routes increased the danger to victims' lives.
* Shelters: Absence of adequate shelters for hiding or carrying injured victims led to more damage and loss of human life.

### **2.1.3 Features Considered Useful**

* Easy-to-Use Interface: The application should have a simple, intuitive, and visually appealing user interface with clear navigation and easily accessible features.
* Offline Functionalities: The application should provide offline access to essential information, such as safety tips, emergency contacts, and previously accessed data.
* Quick Access to Emergency Contacts: The application should offer a dedicated feature for storing and accessing important emergency contacts.
* Real-Time Updates on Disaster Situations: The application should provide real-time updates on disaster situations, including alerts, warnings, and updates from relevant authorities.

## **2.2 RESPONDENTS**

### **2.2.1 Respondents' Specialties in Case of a Disaster**

* Medical Training: The majority of respondents have medical training and expertise.
* Technological Training: A significant portion of respondents have technological expertise and training.

### **2.2.2 Features to Enhance in the System**

* Real-Time Maps and Data Visualization: Provide real-time maps displaying affected areas, critical infrastructure, evacuation routes, and other relevant data.
* Live Video Feeds: Incorporate live video feed capabilities to stream and share footage of affected areas, incidents, and ongoing rescue operations.
* Incident Reporting Tools: Include incident reporting tools to document incidents, attach relevant media, and track the status and resolution of reported incidents.
* Historical Data Analysis: Provide tools for analyzing historical data, including past incidents, response efforts, and outcomes, to identify trends, improve preparedness strategies, and enhance decision-making.

### **2.2.3 Challenges Faced**

* Blocked Roads/Infrastructure Damage
* Lack of Accurate Information on Affected Areas
* Resource Limitations
* Lack of Communication Platforms
* Language Barrier

## **2.3 AUTHORITIES**

### **2.3.1 Features Prioritized in the Disaster Management App**

* Real-Time Data Analytics: Robust real-time data analytics capabilities to collect, analyze, and visualize data related to the disaster situation, response efforts, and resource utilization.
* Emergency Alert System: Incorporate an emergency alert system to send timely and targeted alerts, notifications, and warnings to the affected population and relevant response teams.
* Resource Allocation and Tracking: Provide features for authorities to allocate and track resources, including personnel, equipment, supplies, and facilities, to ensure efficient utilization and coordination.
* Communication Methods: Integrate various communication methods, such as voice and video conferencing, instant messaging, and broadcast messaging, to facilitate seamless communication and coordination.

### **2.3.2 Challenges Faced**

* Network Issues: Network-related challenges were the main issue faced by authorities.

# **CHAPTER 3: REQUIREMENT ANALYSIS**

## **3.1 REQUIREMENT GATHERING**

### **3.1.1 Primary Stakeholders for the disaster management system**

* Disaster Management Authorities
* Emergency Responders
* Victims and Affected Population
* System Developers and Administrators

### **3.1.2 Goals of the Disaster Management System**

* Improve efficiency
* Enhance communication
* Provide situational awareness
* Enable resource management
* Promote public awareness

### **3.1.3 Problems faced**

* Manual and paper-based systems
* Lack of real-time information
* Limited accessibility and reach
* Fragmented communication channels
* Inadequate resource allocation

### **3.1.4 Need for a New System**

* Seamless and real-time communication
* Enhanced data collection and analysis
* Improved accessibility and reach
* Streamlined resource management
* Public awareness and preparedness

## **3.2 FUNCTIONALITIES**

### **3.2.1 Functional Requirements**

* Incident Management
* Resource Management
* Communication and Collaboration
* Situational Awareness
* Response Coordination
* Decision Support

### **3.2.2 Non-Functional Requirements**

* Performance
* Security and Privacy
* Usability and Accessibility
* Interoperability
* Regulatory Compliance

## **USER ROLES AND PERMISSIONS**

|  |  |  |
| --- | --- | --- |
| **USERS** | **ROLES** | **PERMISSIONS** |
| Disaster Model User | These users send disaster analysis requests as descriptive text to execute the model services hosted on the Disaster Management Platform (DMP) via the network | * Access disaster models. * Send disaster analysis requests. * Modify their own requests. * Remove their own requests. * Run disaster models. |
| Disaster Model Provider | These users provide the disaster models that are hosted on the DMP. They are responsible for ensuring that these models are accurate and up-to-date | * Add new disaster models. * View all disaster models. * Modify the disaster models they provided. * Remove the disaster models they provided. * Test run disaster models. |
| Admins | Admin users have complete access rights, allowing them to manage all aspects of the application, such as adding new users, modifying settings, and viewing/editing all content | * Add new users, modify settings, add content. * View all content and user information. * Modify all content, user information, and system settings. * Remove any content, user, or system settings. * Run system-wide operations. |
| Supervisors | Supervisors are responsible for managing content created by other users but may not have full administrative privileges | * Add new content. * View all content. * Modify content created by other users. * Remove content created by other users. * Run operations related to content management. |
| Users | Regular users are able to create content and view their own content and content created by others and can also request help if need be. | * Access public disaster information and response efforts. * Report incidents or needs. * Modify their own reports. * Remove their own reports. * Request assistance. |
| Emergency Respondents | These users are typically first responders or other emergency personnel who are or who have been first at the scene of the disaster to coordinate their response efforts | * Add new response plans. * View all response plans and disaster information. * Modify their own response plans. * Remove their own response plans. * Implement response plans. |
| Government officials | These users are government officials who use the system to coordinate disaster response efforts at a higher level | * Add new disaster response policies. * View all disaster information and response efforts. * Modify disaster response policies. * Remove disaster response policies. * Implement disaster response policies. * Notify public about upcoming disasters and what to do |
| Healthcare Professionals | In the context of a health emergency and disaster management system, healthcare professionals use the system to respond to and cope with health-related hazards | * Add new health emergency plans. * View all health-related disaster information. * Modify their own health emergency plans. * Remove their own health emergency plans. * Implement health emergency plans. * Provide safety drills in case of a disaster |
| Geographic Information System (GIS) Users | In a GIS-based disaster management system, users input their location and needs (like hospital or refuge areas nearby) into the system, which then displays relevant information on a map | * Input their location and needs. * Access relevant information on a map. * Modify disaster locations. * Remove affected areas which have been dealt with * Request for nearby services |

## **3.4 DATA REQUIREMENTS**

* Geospatial Data
* Meteorological and Hydrological Data
* Historical Data
* Social Data
* Infrastructure Data
* Emergency Services Data
* Sensor Data
* Decision Support Systems

## **3.5 SYSTEM INTERFACES**

* User Interface (UI)
* Communications Interface
* Application Programming Interfaces
* Sensor Interface
* Resource Management Interface
* Emergency Alert System Interface
* Geospatial Interface

## **3.6 REPORTING AND ANALYTICS**

### **3.6.1 Importance of Reporting and Analytics in Disaster Management**

* Reporting and analytics enable informed decision-making by providing insights into disaster-related data, performance metrics, and KPIs.
* They facilitate the identification of trends, patterns, and anomalies, helping stakeholders to mitigate risks, allocate resources efficiently, and improve response strategies.
* Reporting and analytics provide a comprehensive view of the system's performance, enabling continuous evaluation and improvement.

### **3.6.2 Types of Reports**

* Regular Reports
* Ad Hoc Reports
* Executive Reports

### **3.6.3 Dashboards and Visualizations**

* Dashboards: Dashboards provide real-time visual representations of key metrics and performance indicators.
* Interactive Visualizations: Interactive visualizations allow users to explore and analyze data intuitively.

### **3.6.4 Disaster-Related Data**

* Incident Data
* Resource Data
* Performance Data

### **3.6.5 Performance Metrics and Key Performance Indicators (KPIs)**

* Response Time
* Resource Utilization
* Incident Resolution
* Cost Efficiency
* Public Satisfaction

## **3.7 CONSTRAINTS**

### **3.7.1 Technical Constraints**

* Hardware limitations
* Software compatibility
* Performance requirements
* Security constraints
* Technology limitations

### **3.7.2 Time Constraints**

* Project deadlines
* Time-to-market

### **3.7.3 Budget Constraints**

* Cost limitations
* Cost-effectiveness

### **3.7.4 Organizational Constraints**

* Regulatory compliance
* Organizational policies
* Stakeholder requirements

## **3.8 DEPENDENCIES**

* Third-Party Software
* Hardware
* Data Sources
* Regulatory Changes

## **3.9 ACCEPTANCE CRITERIA**

* Functional Requirements
* Non-Functional Requirements
* Performance
* Usability
* Security
* Reliability
* Testing and Validation

# **CHAPTER 4: SYSTEM MODELLING AND DESIGN**

## **4.1. CONTEXT DIAGRAM**

The context diagram for our disaster management system illustrates the system at the center, with lines connecting to external entities like local residents, disaster response teams, and local government.

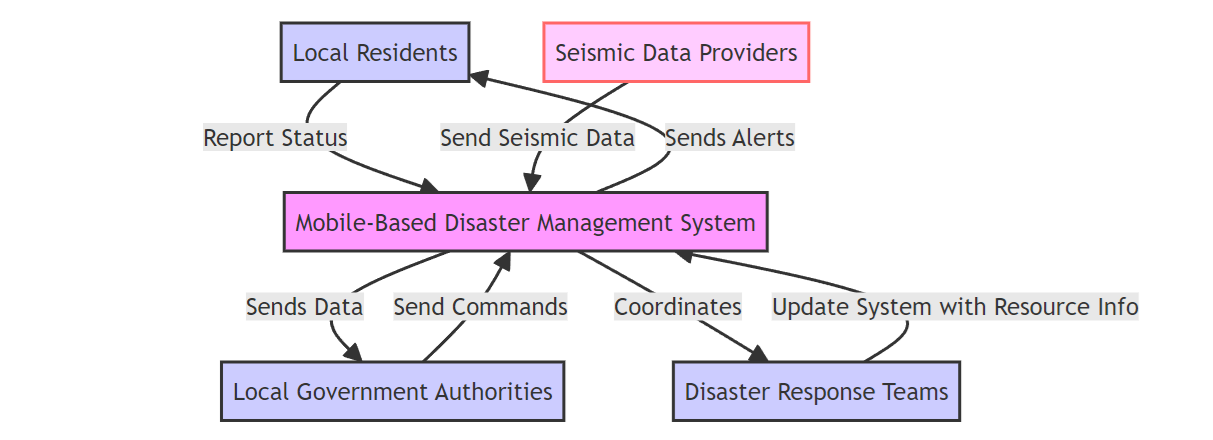


Figure 1: Context Diagram

### **4.1.1 Actors and Their Roles**

• Local Residents: Users who will interact with the system to receive alerts and provide updates about their status during a disaster.

• Disaster Response Teams: Groups that coordinate and execute emergency response strategies; they use the system for communication and resource management.

• Local Government Authorities: Government bodies responsible for overseeing disaster management at a policy and operational level. They issue commands and receive aggregated data for decision-making.

• Seismic Data Providers: Entities that provide real-time seismic data, crucial for the system to detect and assess earthquake events.

## **4.2 USE CASE DIAGRAM**

The use case diagram includes actors such as residents and response teams, and use cases like receive alerts, report status, and access emergency info.

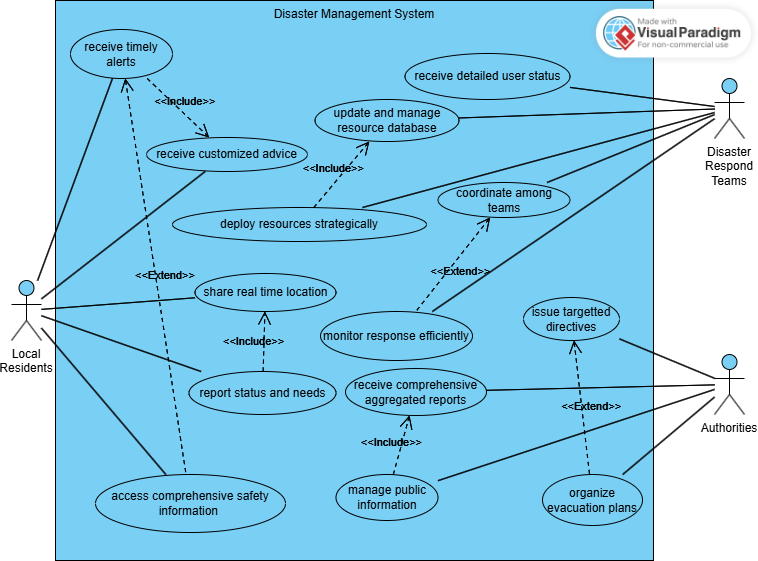


Figure 2: Use case diagram

### **4.2.1 Actors and Their Roles**

• Residents: Interact with the system to receive information and provide personal status updates during disasters.

• Response Teams: Manage and coordinate emergency responses using the system to access realtime data from residents and other sources.

• Government Officials: Use the system to monitor disaster management efforts and communicate with both the public and response teams.

## **4.3 SEQUENCE DIAGRAM**

A sequence diagram for the "Disaster Alert Issuance" scenario shows interactions starting from the seismic activity detection, system processing, and alert dispatch to user devices.

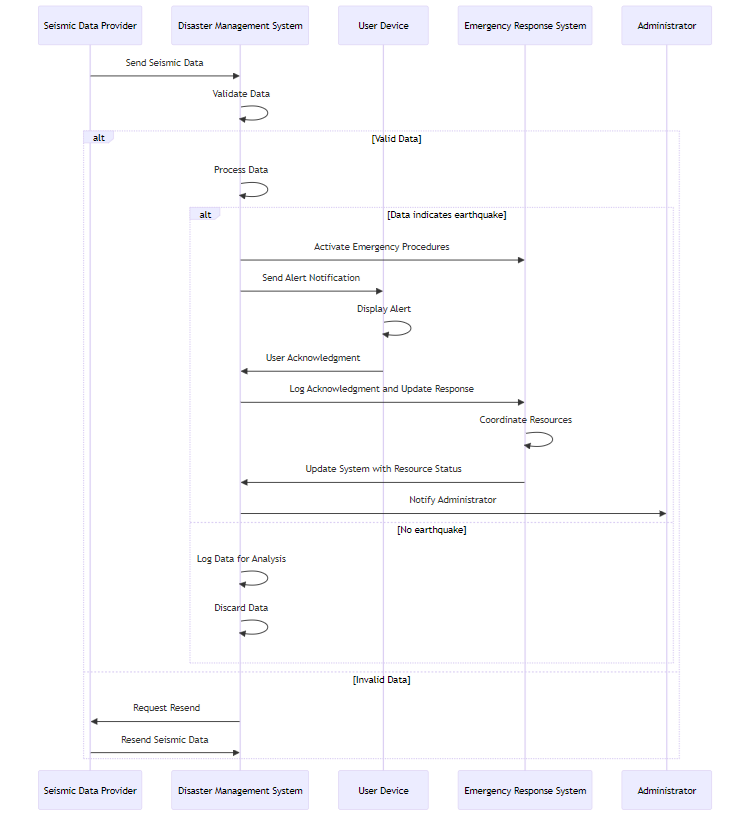


Figure 3: Sequence diagram

### **4.3.1 Actors and Their Roles**

• Seismic Data Provider: Supplies initial data that triggers the system's alert processes.

• Disaster Management System: Processes seismic data, determines the necessity of alerts, and manages data flow to other components.

• User Device: Receives alerts and displays them to the user; sends back user acknowledgments or requests for help.

## **4.4 CLASS DIAGRAM**

The class diagram would include classes such as User, Alert, Emergency Response, each with their respective attributes and methods.

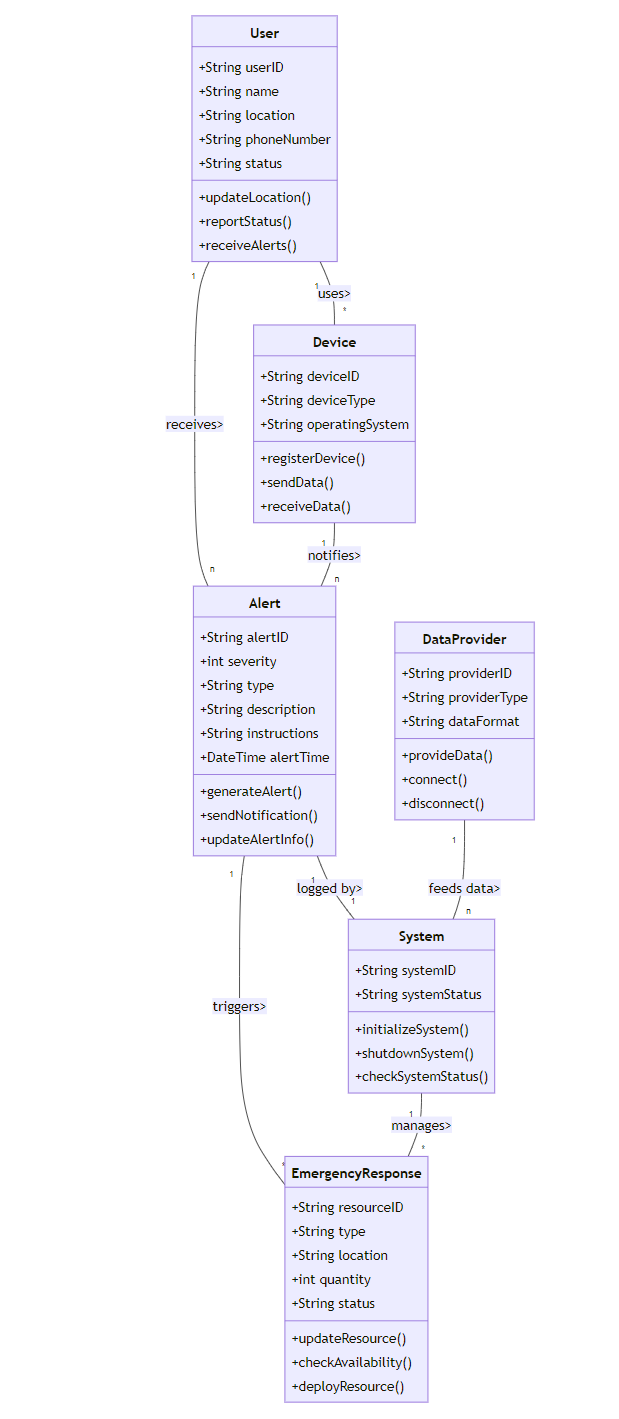


Figure 4: Class Diagram

### **4.4.1 Actors and Their Roles**

• User Class: Represents the end-users of the system, storing information such as location and status.

• Alert Class: Manages the creation and distribution of alert notifications to users.

• Emergency Response Class: Manages resources and responses during emergencies.

## **4.5 DEPLOYMENT DIAGRAM**

The deployment diagram shows the distribution of the mobile application across user devices, connected through secure networks to centralized servers that manage data and interactions.

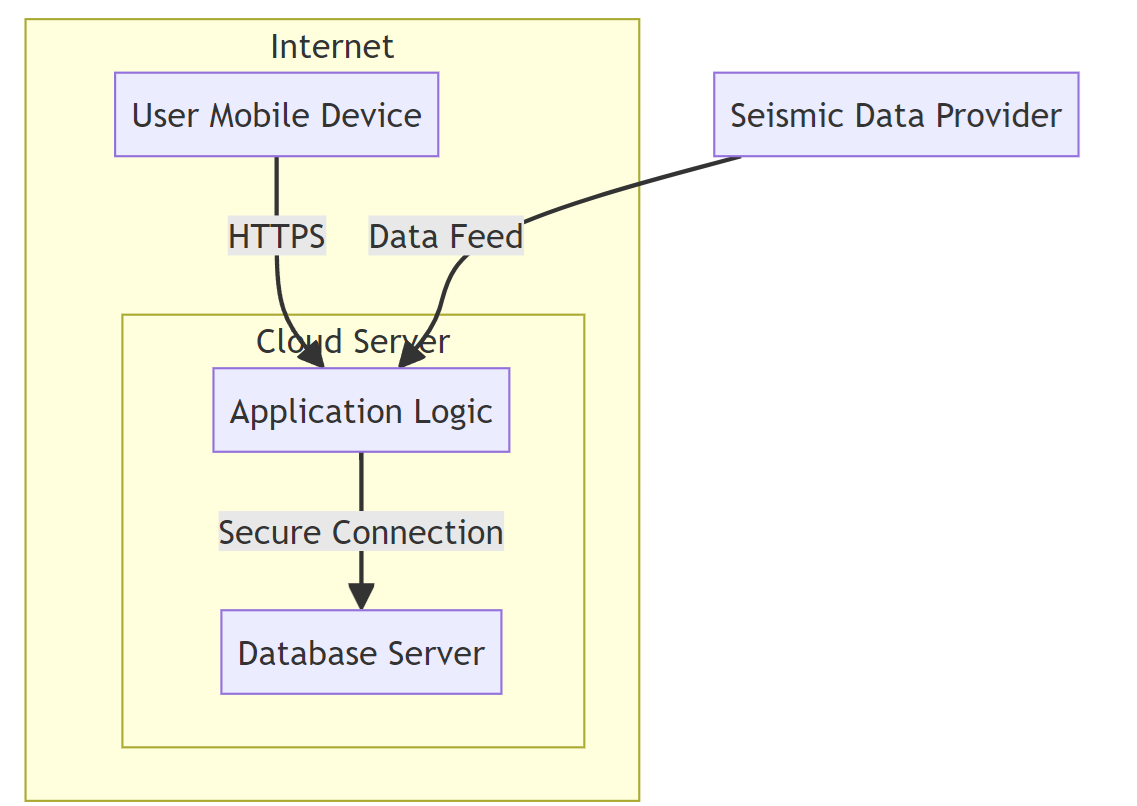


Figure 5: Deployment Diagram

### **4.5.1Actors and Their Roles**

• User Mobile Device: Where the application is installed and used by local residents.

• Cloud Server (Application Logic, Database Server): Hosts the application logic and the database storing all system data, ensuring that data is processed and stored centrally for access by all other system components.

# **CHAPTER 5: UI DESIGN AND IMPLEMENTATION**

## **5.1 UI DESIGN PROCESS**

The UI design process for the disaster management mobile application follows a structured approach:

• Requirement Gathering: Engage with potential users, respondents, and authorities to understand their needs and preferences.

• Wireframing: Develop basic layouts for each screen, focusing on functionality and user flow.

• Prototyping: Create interactive prototypes using Figma to visualize the user experience and gather feedback.

• User Testing: Conduct usability testing with a diverse group of users to identify and resolve any issues.

• Iteration: Refine the designs based on feedback and testing results to ensure the final product meets user expectations.

Link to figma design: https://www.figma.com/design/MmQwhmaSH4pXAiKShlACRi/Untitled?node-id=0-1&t=ictCbqtb4vm4XwlI-1

## **5.2 FIGMA AS A DESIGN TOOL**

Figma was chosen for the UI design due to its robust features and collaborative capabilities:

• Real-Time Collaboration: Multiple designers can work on the same project simultaneously, making it ideal for team-based design work.

• Prototyping Capabilities: Figma allows the creation of interactive prototypes, helping visualize user flows and gather feedback.

• Accessibility: Being a cloud-based tool, Figma ensures that design files are easily accessible from anywhere.

• Design Consistency: Figma supports the creation of design systems and component libraries, ensuring consistency across the application.

## **5.3 SCREENS**



Figure 6: Language screen

Figure 7: Splash Screen

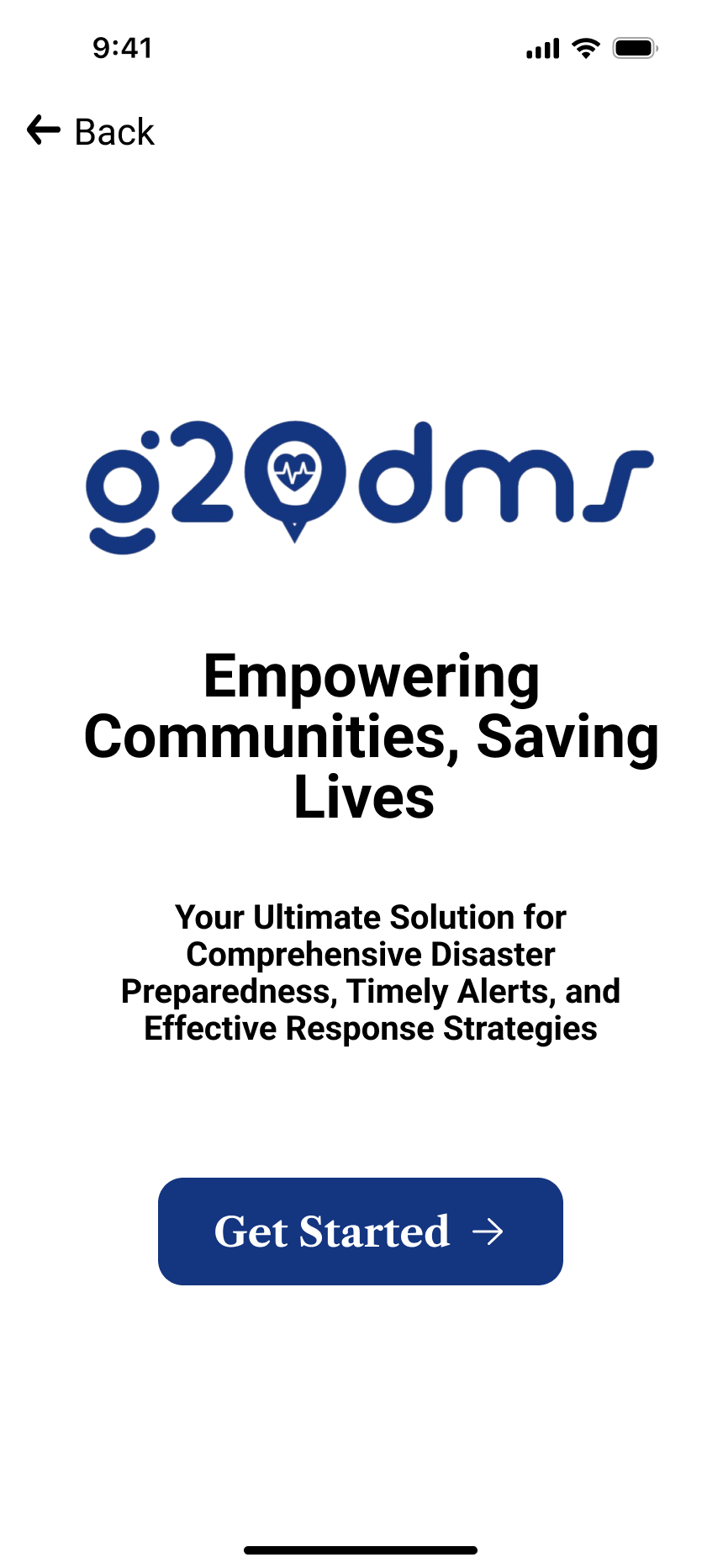
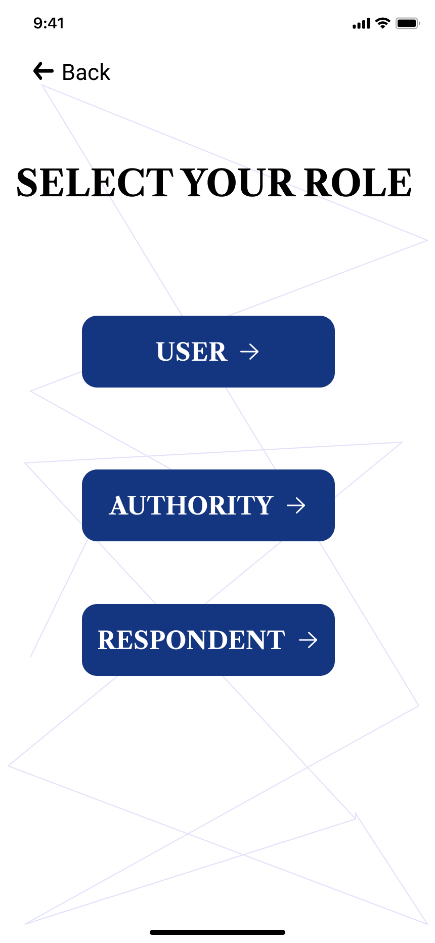
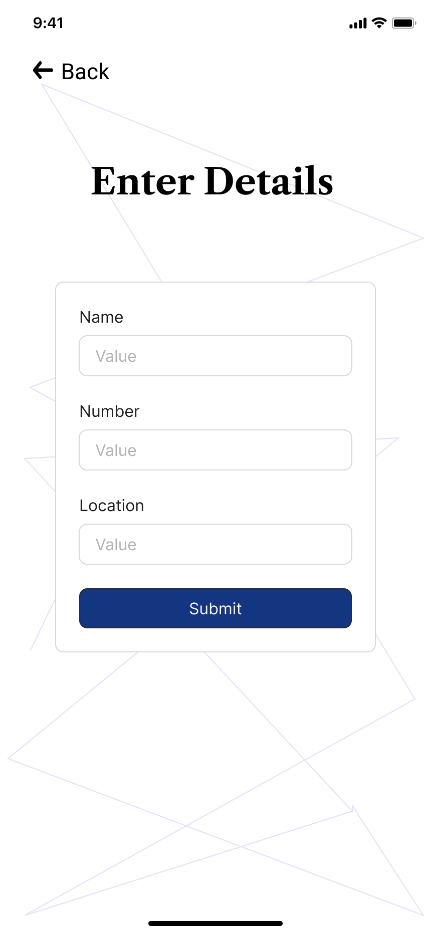


Figure 8: user enter details

Figure 9: Select role

Figure 10: Get started

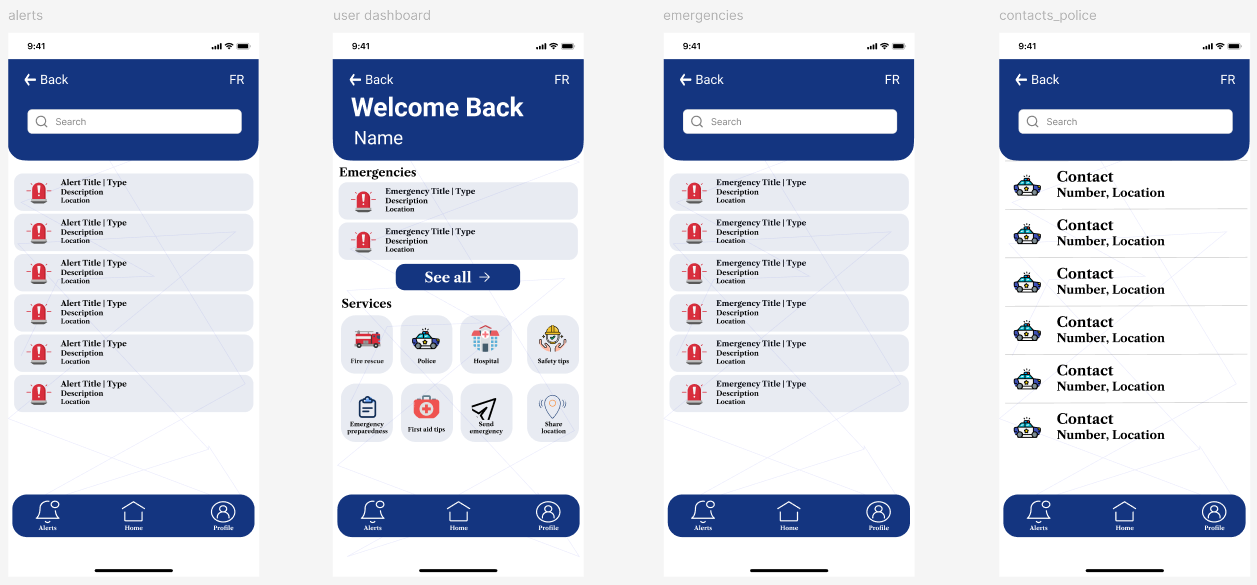


Figure 11: alerts, user dashbaord, emergencies and contacts police screen for the user

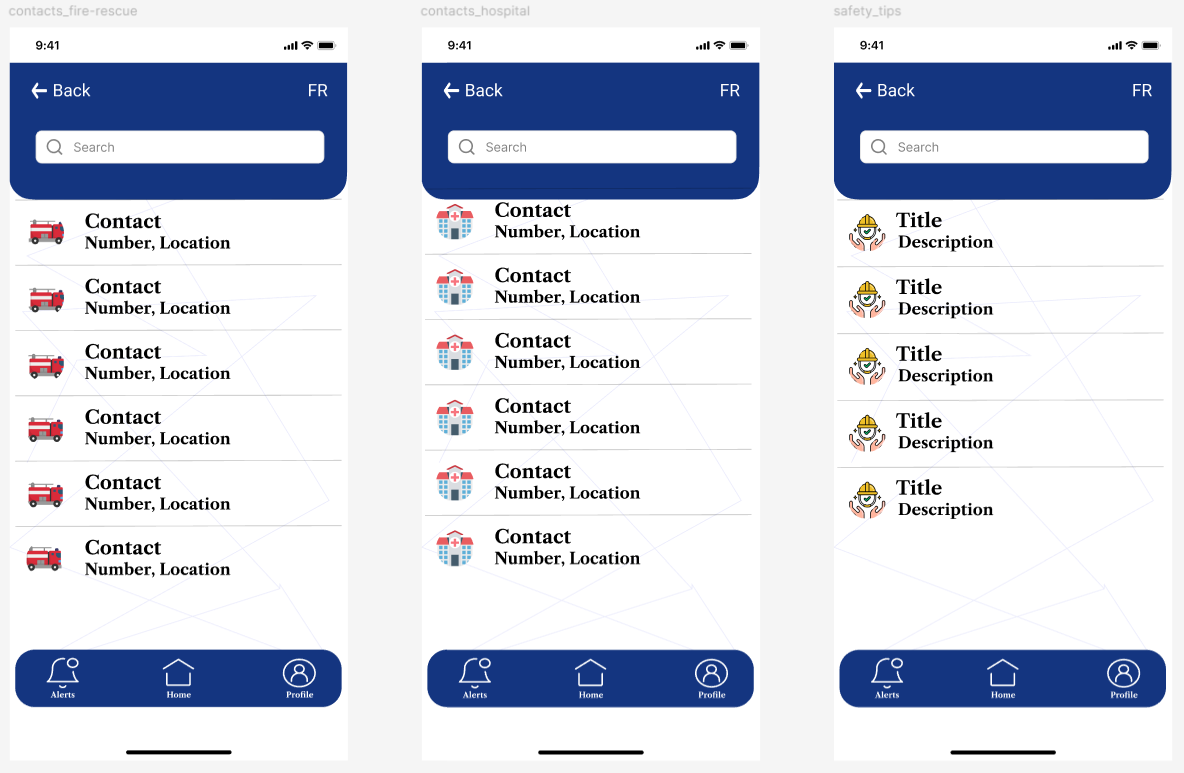


Figure 12: Additional screens for the user

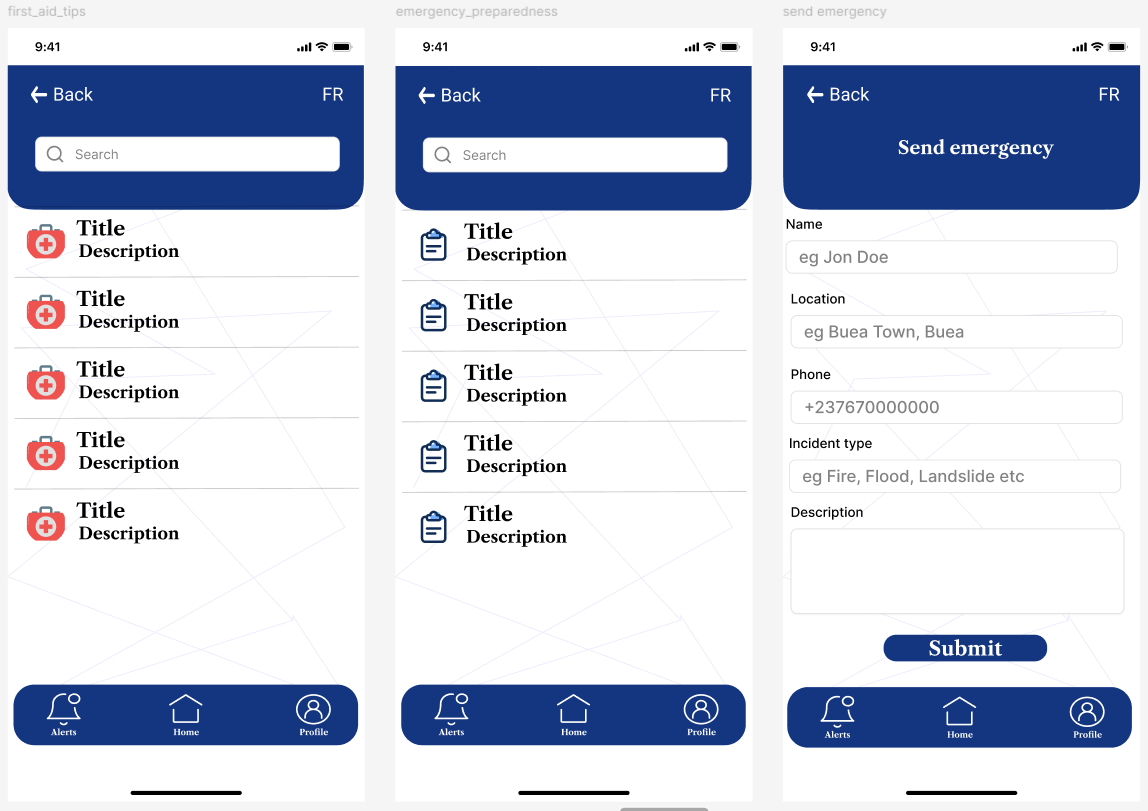


Figure 13: Screen for user

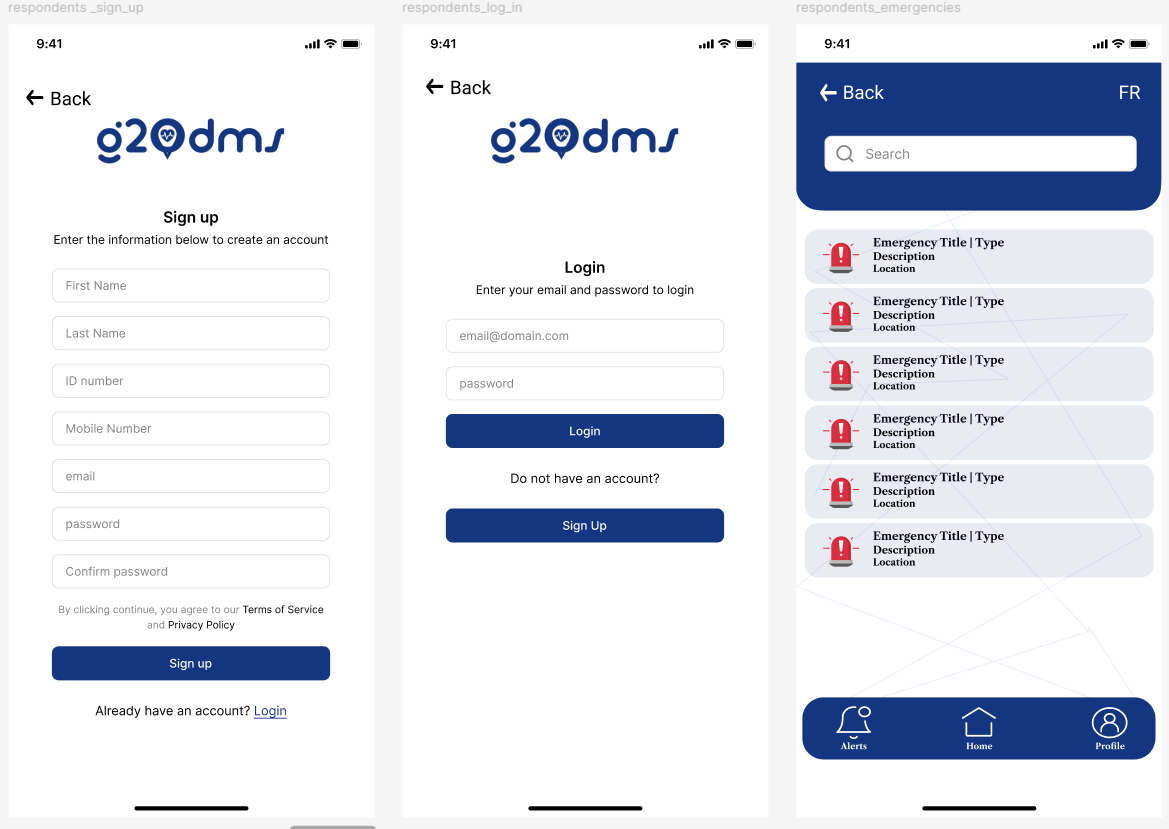


Figure 14: Respondents screens

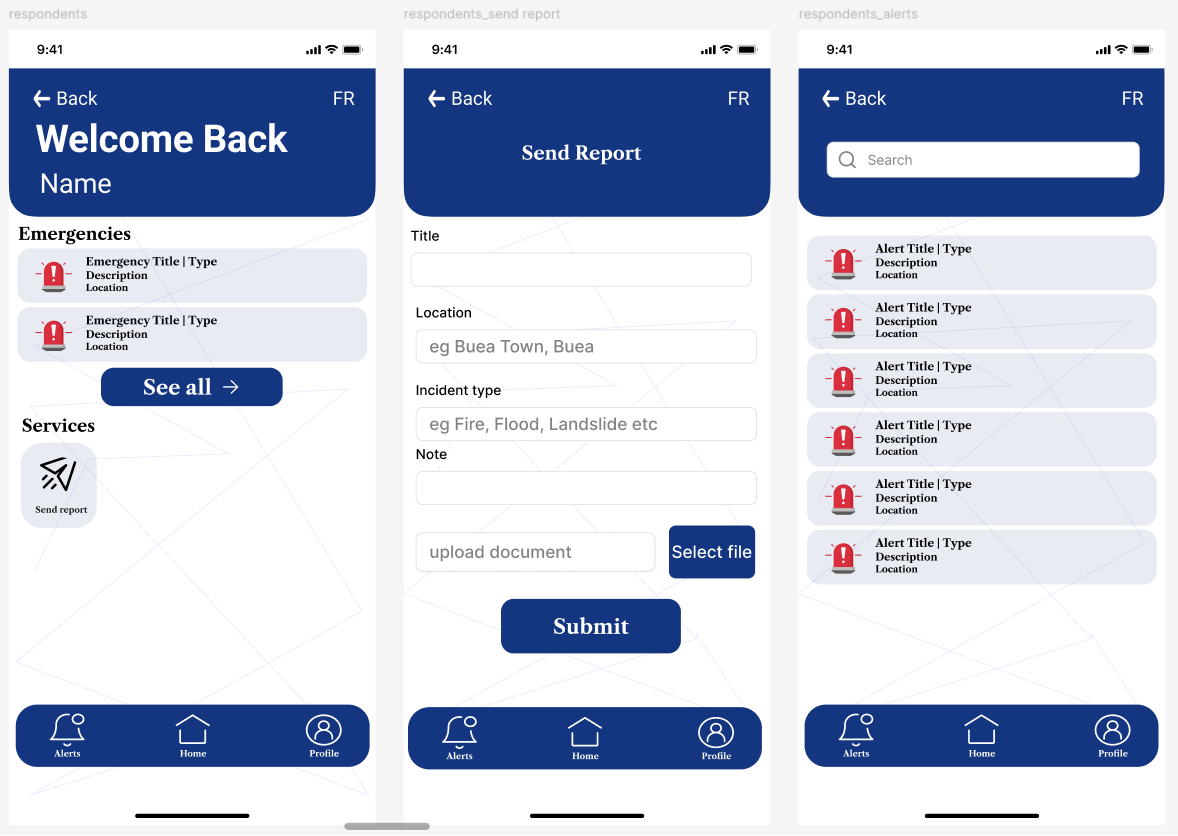


Figure 15: Additional respondents screens

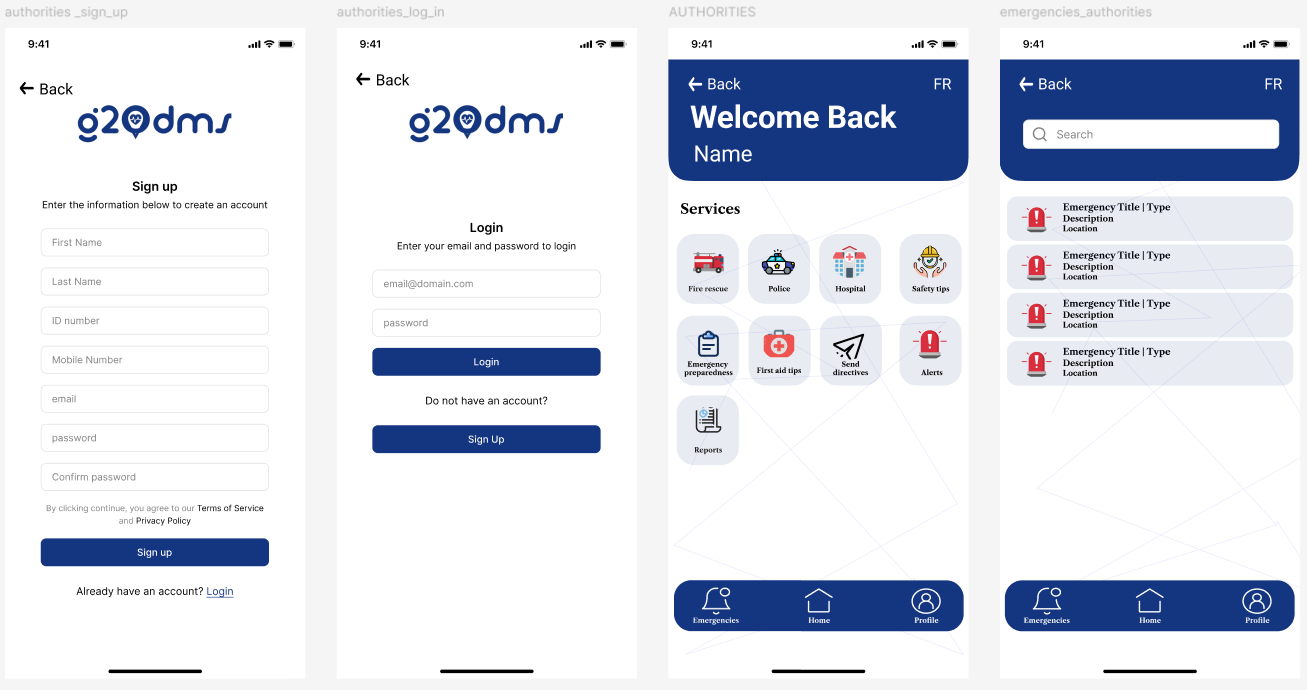


Figure 16: Authorities sign up, login, dashboard and emergency screens

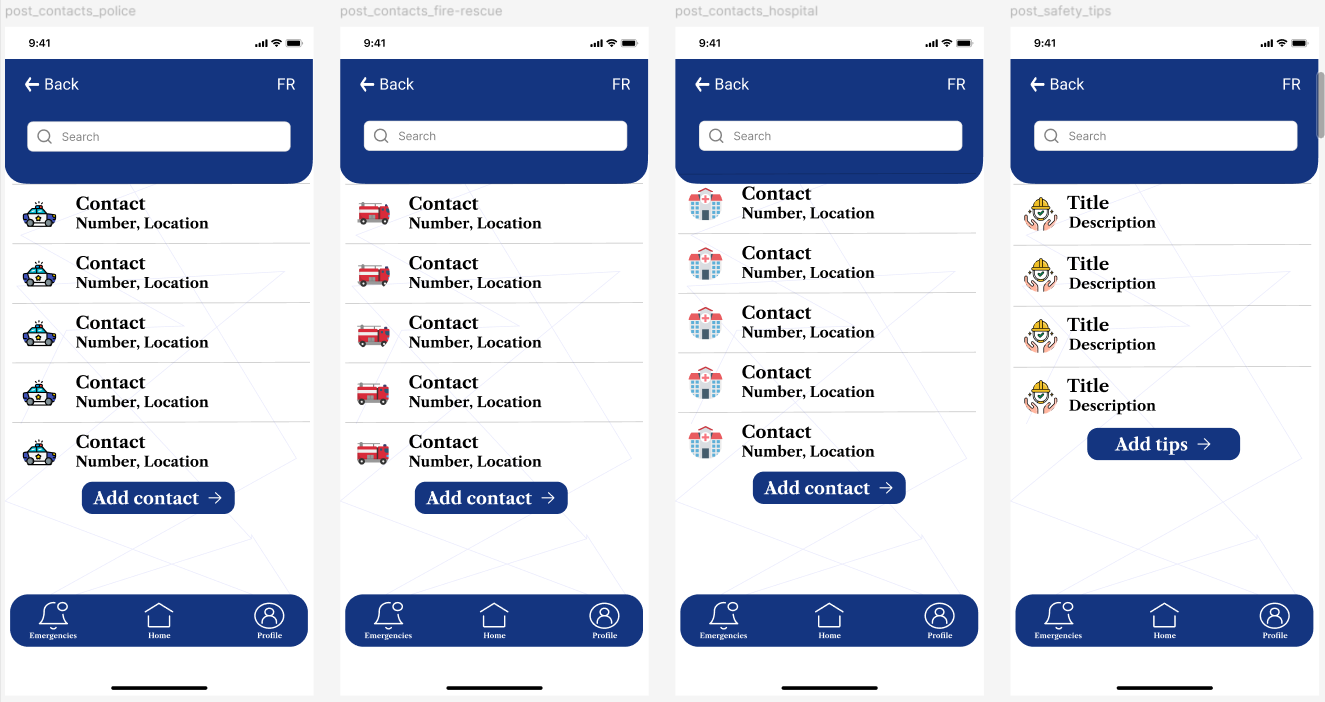


Figure 17: Additional authorities screen

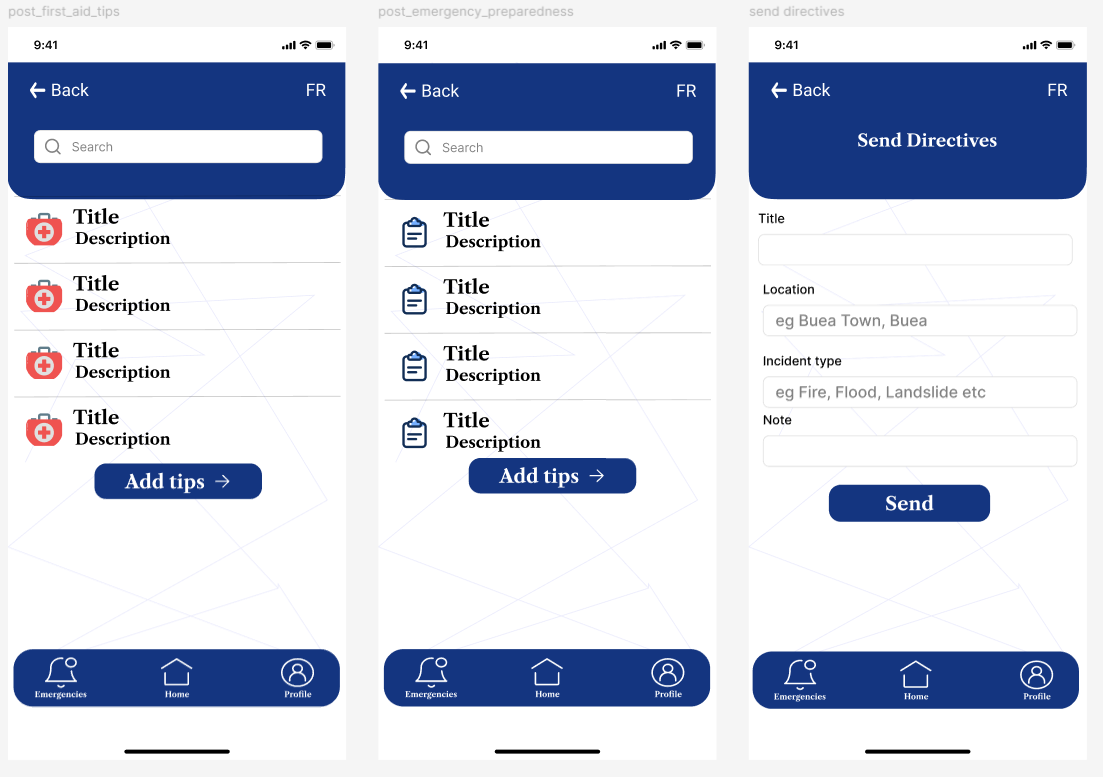


Figure 18: Additional authorities screens

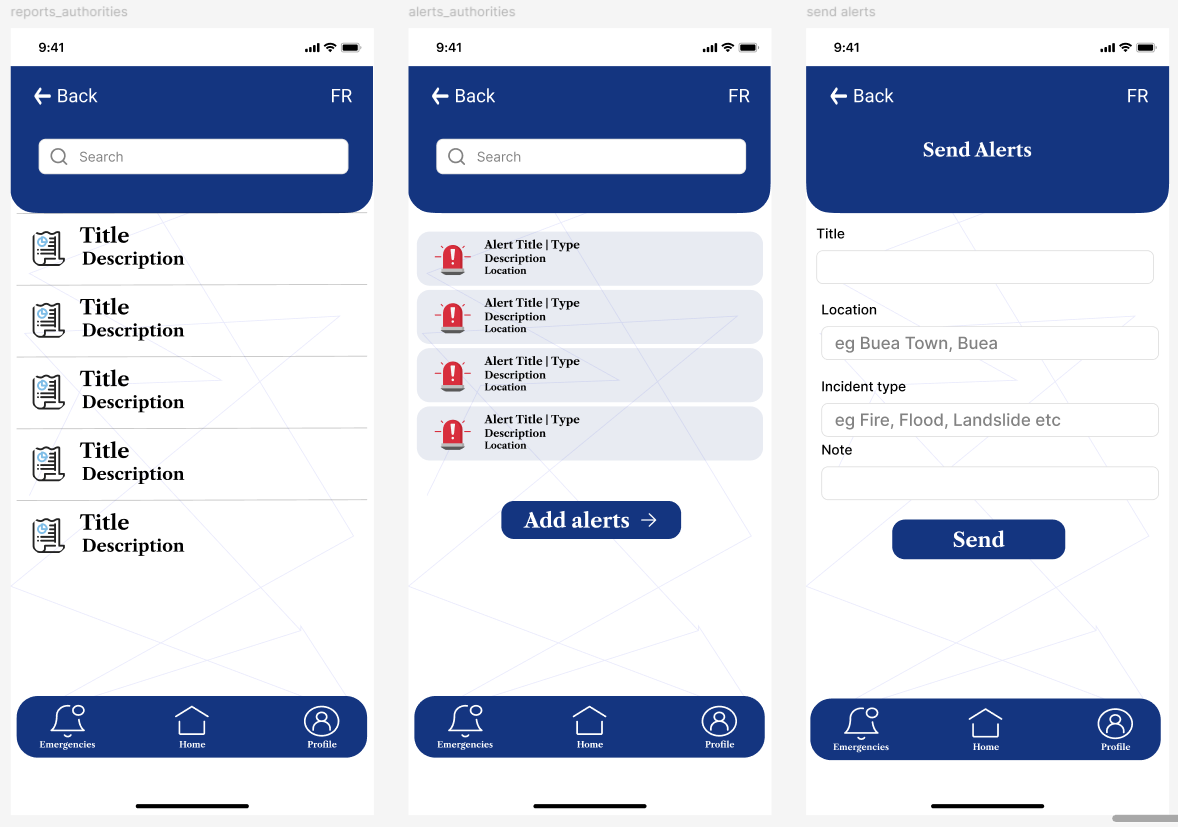


Figure 19: Additional authorities screens

## **5.6 UI Implementation Strategy**

The UI implementation strategy involves several key steps:

• Development: Using a cross-platform framework like Flutter to build the app, ensuring consistency across different devices and operating systems

• Testing: Conducting thorough testing, including usability testing, functional testing, and performance testing, to ensure the app is reliable and user-friendly

• Maintenance: Regularly updating the app to improve functionality, fix bugs, and enhance the user experience based on ongoing feedback and technological advancements.

## **5.7 Coding**

A link to the code of our design was uploaded to github which can be found in the link below;

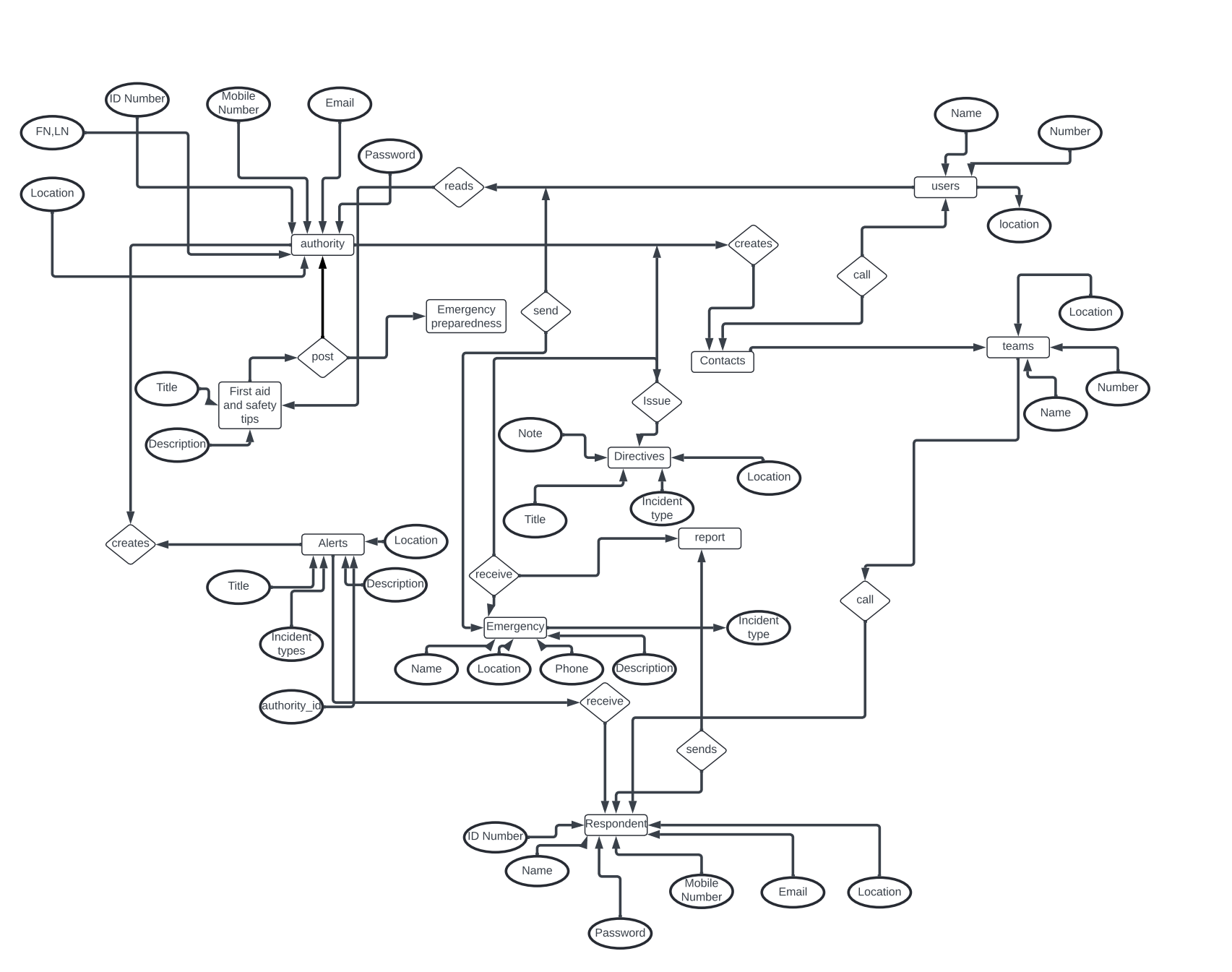
https://github.com/duesenberry55/CEF440-group-20/tree/main/dmsg20

# **CHAPTER 6: DATABASE DESIGN AND IMPLEMENTATION**

## **6.1. CONCEPTUAL DESIGN**

### **6.1.1 Entities Identified:**

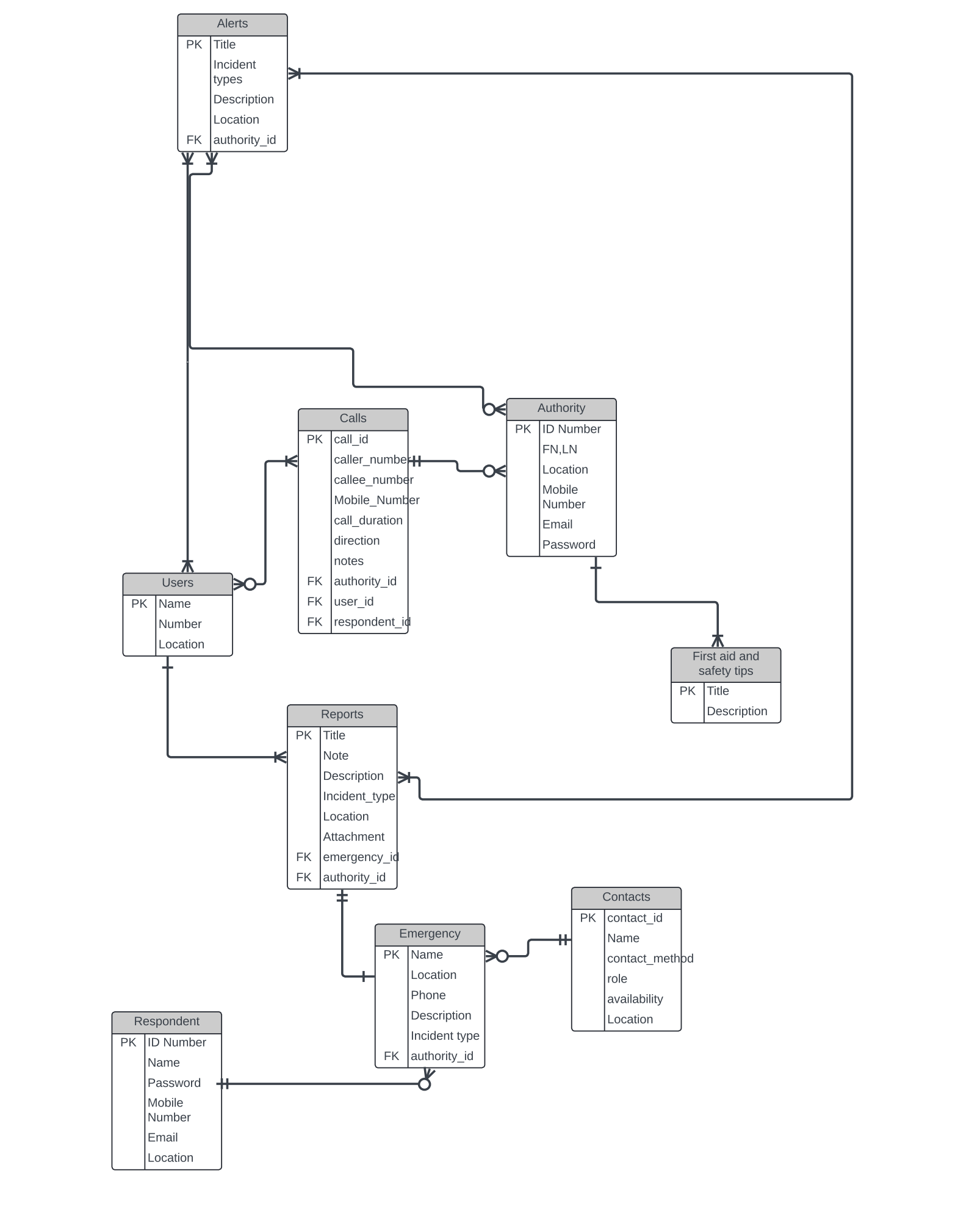
* Users: Individuals using the system, including the general public and disaster
* victims.
* Authority: Government and agency officials responsible for disaster response.
* Respondents: First responders and emergency services.
* Emergency: Types and details of emergencies.
* Contacts: Contact details for all users.
* Alerts, Calls, and Reports: Communication and data generated during
* emergencies.



## **6.2. LOGICAL DESIGN**

### **6.2.2 Schema Definition:**

* Created tables like Users, Authorities, Emergencies, each reflecting the system’s entities.
* Defined attributes such as ID Number, Name, Location, and Email, using data types appropriate for the content (e.g., VARCHAR for text, INT for numerical data).
* Established Primary Keys and Foreign Keys to maintain referential integrity across the relational database.



## **6.3. PHYSICAL DESIGN**

We selected MYSQL workbench as our Database Management System (DBMS) due to its robustness, scalability, and support for complex queries necessary for disaster management:

### **6.3.1 Physical Storage Considerations:**

* Implemented indexing on frequently accessed fields like User ID and Emergency Type to speed up queries.
* Employed partitioning on historical data to improve performance and manageability.

### **6.3.2 Database Security Design:**

* Implemented data encryption both at rest and in transit using mySQL workbench native support.
* Setup role-based access controls to ensure that data is only accessible to authorized personnel, crucial for protecting sensitive information.

## **6.4. SQL DATABASE IMPLEMENTATION**

Our SQL implementation phase involved the actual creation of the database schema based on our design:

### **6.4.1 SQL Scripts:**

* Executed SQL scripts to create tables, define relationships, and set constraints.
* Example SQL for creating the Users table:
* Scripts can be found here: <https://github.com/duesenberry55/CEF440-group20/tree/main/task%206>

**sql**

CREATE TABLE Users (

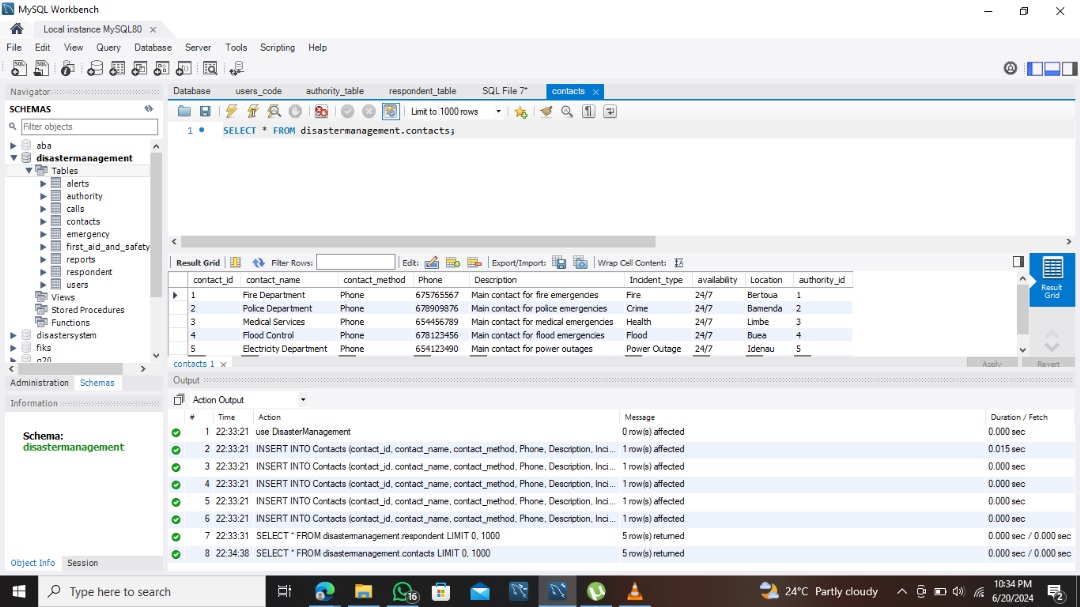
UserID INT PRIMARY KEY,

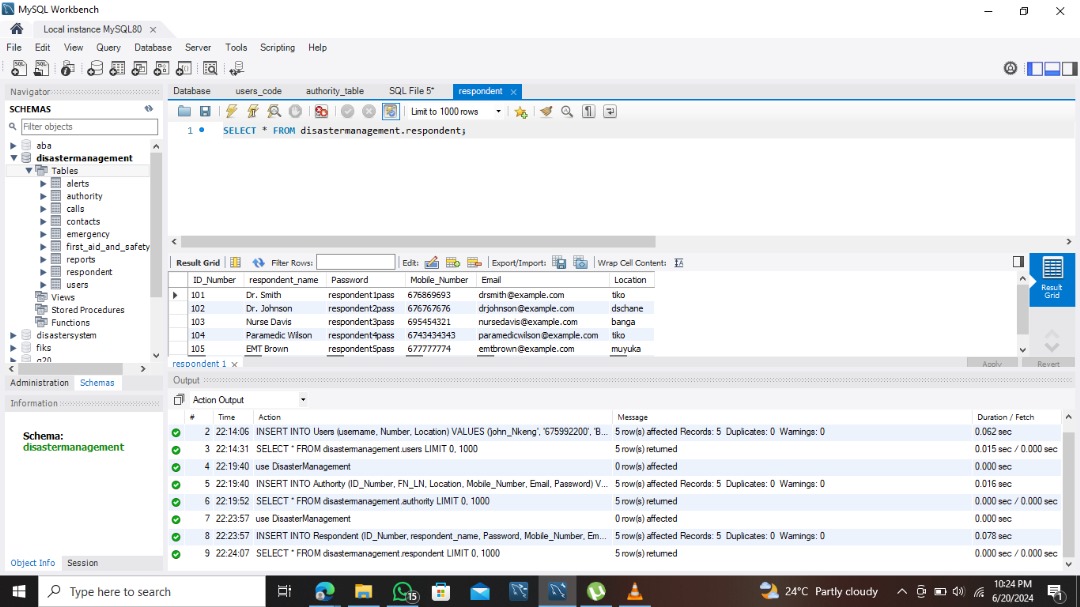
Name VARCHAR(100),

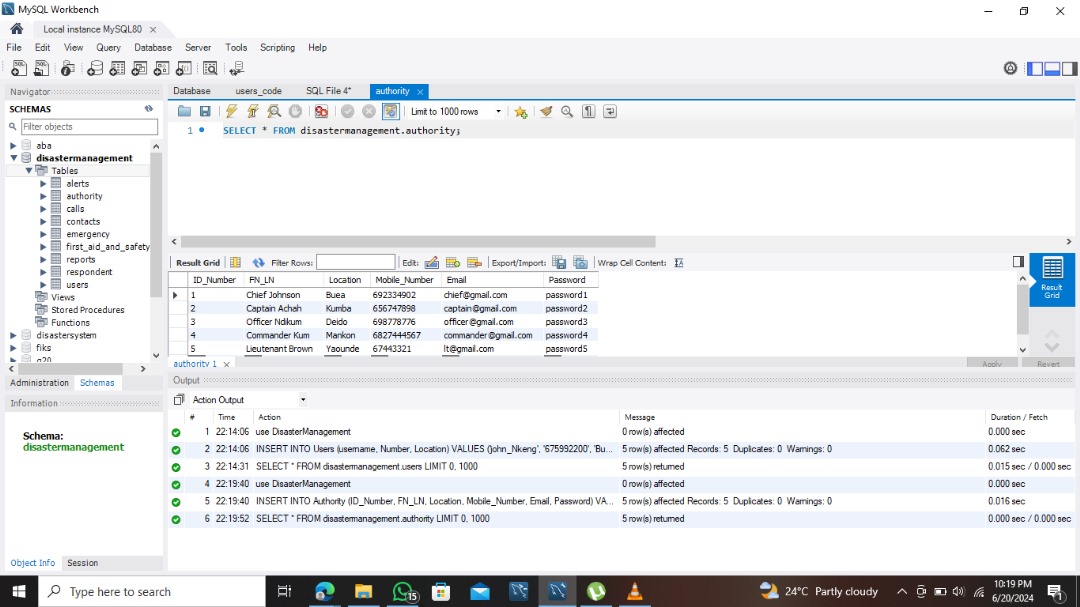
Location VARCHAR(100),

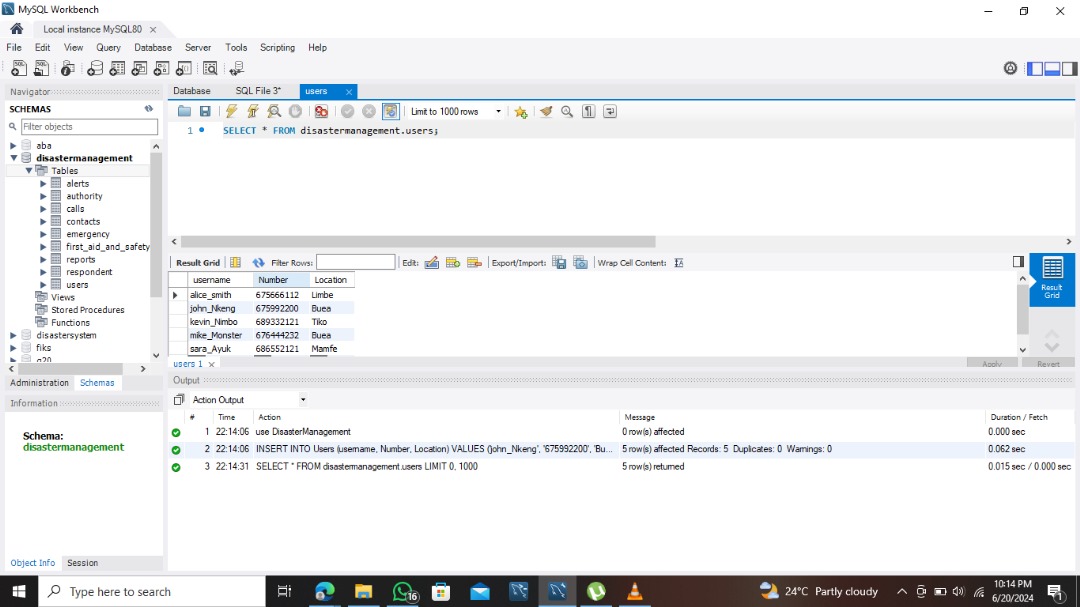
Email VARCHAR(100) );

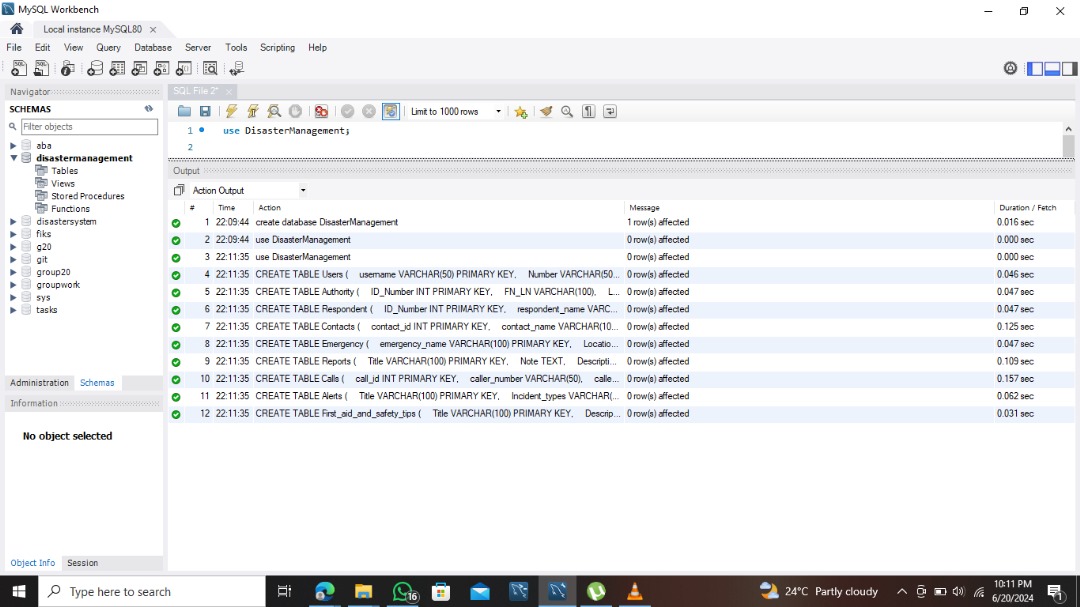
## **6.5 RESULTS:**

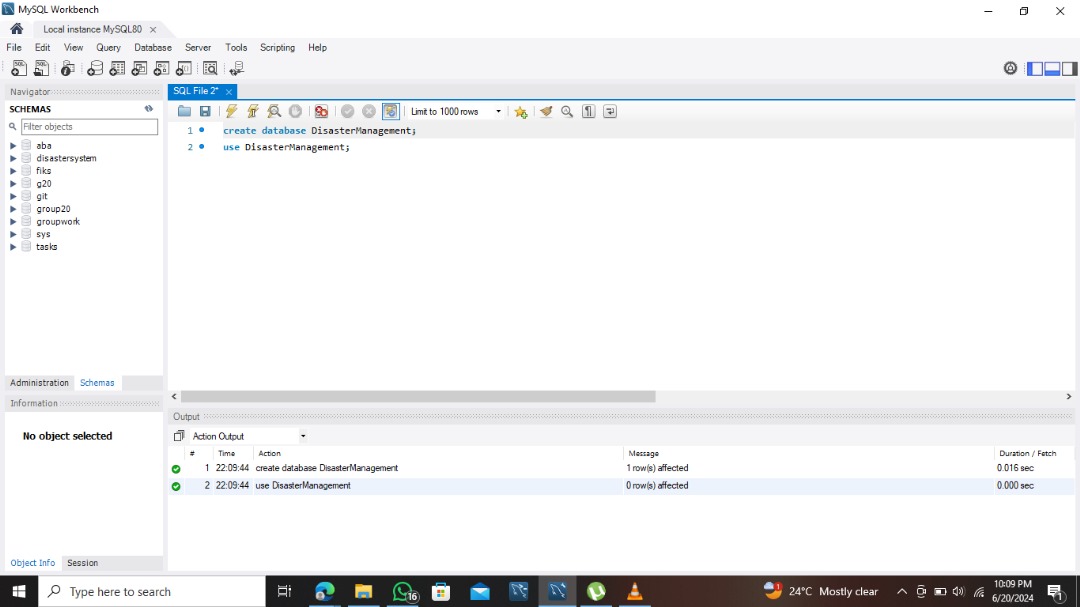












## **6.6. TECHNOLOGIES USED**

• Database Management System (DBMS): mySQl was chosen for its advanced features, reliability, and open-source nature.

• Development Environment: We utilized mySQL workbench, providing us with a graphical interface to facilitate database design and manipulation.

• Data Modeling Tools: Lucid chart was used to create detailed ER diagrams, supporting our data modeling and schema design processes.

• Security Technologies: We implemented AES encryption standards through PostgreSQL for data at rest and TLS for data in transit.

• Testing Tools: For load testing, we used mySQL workbench built-in benchmarking tool, to measure the system’s performance under various load conditions.

## **6.7. TESTING AND VALIDATION**

Extensive testing was conducted to ensure the integrity and performance of the database:

### **6.7.1 Testing Strategies:**

* Performed unit testing on individual tables.
* Conducted integration testing to ensure that table relationships were properly enforced.
* Load testing was implemented to simulate real-world usage scenarios.

# **CHAPTER 7: CONCLUSION**

In an era where natural disasters are becoming increasingly frequent and severe, the need for an efficient and advanced disaster management system is paramount. The proposed mobile-based Disaster Management System (DMS) addresses this critical need by providing a centralized platform that enhances disaster preparedness, response, and recovery. By integrating real-time alerts, safety tips, and emergency contacts, the DMS ensures that individuals, emergency responders, and authorities have the information and tools necessary to navigate emergencies effectively

The DMS stands to revolutionize disaster management by streamlining communication, resource allocation, and situational awareness. The project's methodology, rooted in mobile application development and cloud computing, guarantees a scalable, efficient, and user-friendly platform accessible to both iOS and Android users. Utilizing technologies such as React Native and Firebase ensures that the system can handle real-time data processing and provide seamless user experiences, critical for effective disaster management.

The significance of the DMS project extends beyond immediate disaster response. It fosters community resilience by encouraging preparedness, enhancing response coordination, and supporting recovery efforts. Moreover, the platform promotes awareness and education on disaster risks, contributing to a more informed and prepared public.

This document outlines the comprehensive requirements and strategic approach for developing the DMS, providing a clear guide for stakeholders involved in the project. Through the successful implementation of this system, we aim to mitigate the impact of disasters, protect lives and property, and build stronger, more resilient communities. The DMS aspires to set a new standard in disaster management, embodying a proactive and coordinated approach to addressing the challenges posed by natural disasters.