Chapter II

**PRESENTATION OF FINDINGS, ANALYSIS AND INTERPRETATION OF DATA**

**Existing Operation and Practices**

The Bilar Search and Rescue Unit (BISARU) operates around the clock to provide assistance with emergency concerns throughout the area of Bilar, Bohol. BISARU continues to employ manual processes for the identification and location of emergency concerns. BISARU acknowledges the need for advancements and is actively exploring opportunities to integrate modern technologies for more efficient and effective emergency response.

# Receiving Emergency Through Communication Channels

The traditional operational workflow of emergency response agencies involves a manual sequence of steps, beginning with the reception of emergency calls through various communication channels such as mobile phones, telephones, or frequency radios.

# Emergency Verification

Upon receiving a distress call, the specialized response team promptly initiates a thorough verification process to confirm the authenticity and urgency of the reported incident.

# Deployment

The response team promptly proceeds to the incident location, where they actively engage in the process of confirming essential details related to the accident,

carefully assessing factors such as its nature and impact, and ensuring accuracy in pinpointing the precise location of the incident.

# Generation of Reports.

The generation of reports involves systematically documenting and recording pertinent details in a dedicated logbook, highlighting the ongoing need to refine and enhance the process for maintaining a comprehensive and organized record of relevant information.

These comprehensive verification steps are essential for ensuring accurate and effective emergency response. However, inherent delays are incurred in the overall emergency response due to the manual nature of these processes and the constraints imposed by travel time to reach the incident site promptly.

# Needs of the Existing Operation

The present process will employ a cross-platform application-based approach, which cn be customized to better suit user needs. The researcher’s observations led to the identification of the following needs:

1. To develop an app with location tracking capabilities for users in need of assistance, ensuring that responders have information about their situation. The goal is to enhance safety through effective communication and provide location details.

# BERA: BILAR EMERGENCY RESPONSE APPLICATION USING DYNAMIC CLUSTERING PROTOCOL

After all the data and information that was gathered from the current system of BISARU Bohol and developed the “BILAR EMERGENCY RESPONSE APPLICATION”, a system that would handle emergency concerns more efficiently, optimize the verification process, and automate the generation of reports.

# Administration

Authorized individuals are required to log in to access the system by entering their email and password in the login form. This security measure is implemented to prevent unauthorized access and potential breaches, safeguarding responder and user personal information from loss or compromise. Within the administration module, authorized personnel can adjust settings and oversee user roles.

# Emergency Notification and Dispatch

The new system can generate alerts efficiently notifying designated responders. It provides details to ensure a well-informed response, and the system logs and tracks the entire alert and response process for comprehensive record- keeping and continuous improvement of emergency response procedures.

# Location Tracking

The system can pinpoint the location of an incident through the use of geolocation technology. The location tracking feature allows for the quick identification of the exact site, facilitating timely assistance and intervention.

# Generation of Reports.

The system will generate reports containing data on responders, including registered app users. This encompasses the utilization of graphical data visualization methods, such as daily, weekly, monthly, and annual incident reports. These reports may be formatted in tables, ready for convenient printing.

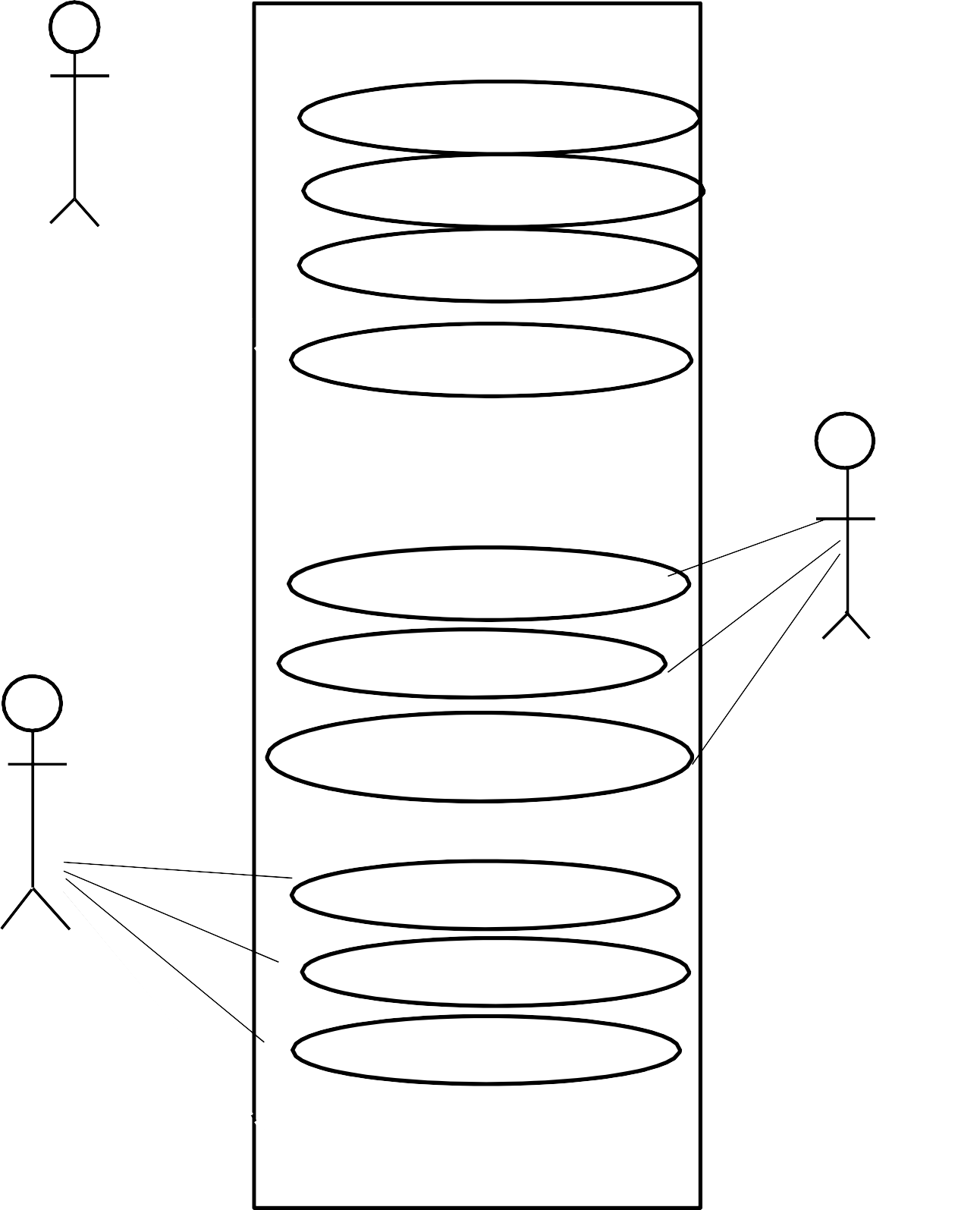
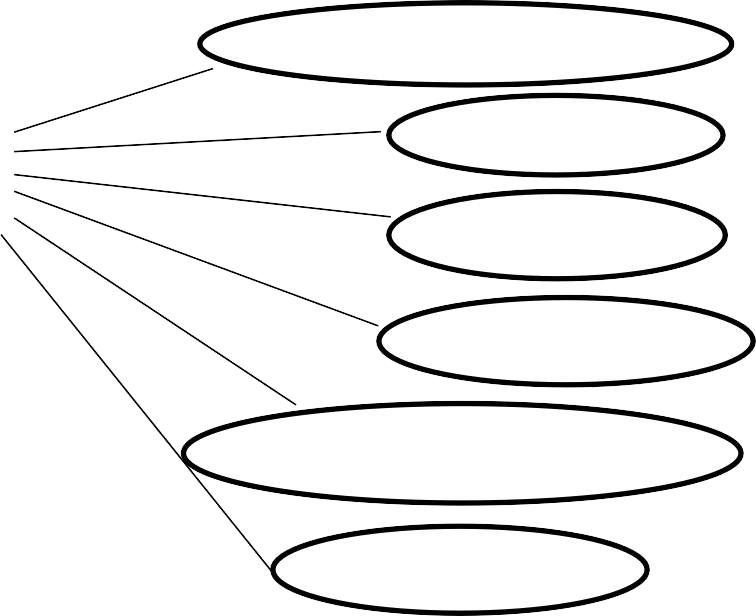
# Used Case Diagram

According to Erikkson and Pemker (2000), a use case diagram illustrates the particular utilization of the system by one or more actors. The use case entails a series of actor or event steps, typically outlining the interactions between a role (referred to as an actor in the Unified Modeling Language) and a system to accomplish a goal. UML Use Case diagrams are commonly known as behavior diagrams and are employed to depict a set of actions (use cases) that a system or systems (subject) should or can execute in collaboration with one or more external users of the system (actors).

Figure 4 displays the use-case diagram of the BERA application, comprising the actors namely: emergency requestor, emergency message receiver, and emergency responder.

# Use Case Diagram

**Login With their Credentials**



**Edit Profile**

**Emergency Requestor**

**Send Photos for Evidence**

**Activate GPS**

**Receive Emergency Message**

**Emergency Responder (Mobile Phone)**

**Receive Emergency Calls**

**Access Google Map to track the location of emergency requestor.**

**Track GPS location of the phone**

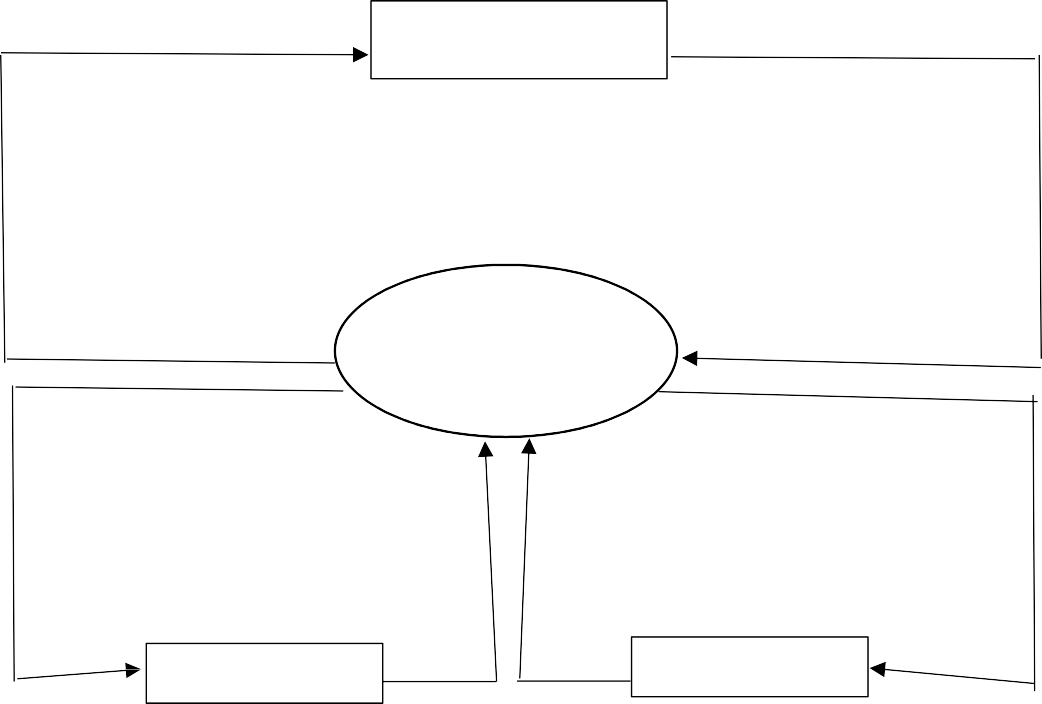
**Capture Snapshots of the Scene**

**Send emergency message**

**Emergency Message Receiver**

Figure 4: Use Case Diagram of Bilar Emergency Response Application

**Data Flow Diagram**



Admin/Responder

* Clients Identification
* Emergency Reports
* Rescue

Emergency Response System

* Clients Information  Call
* Monitoring
* Request Help

Admin

Client

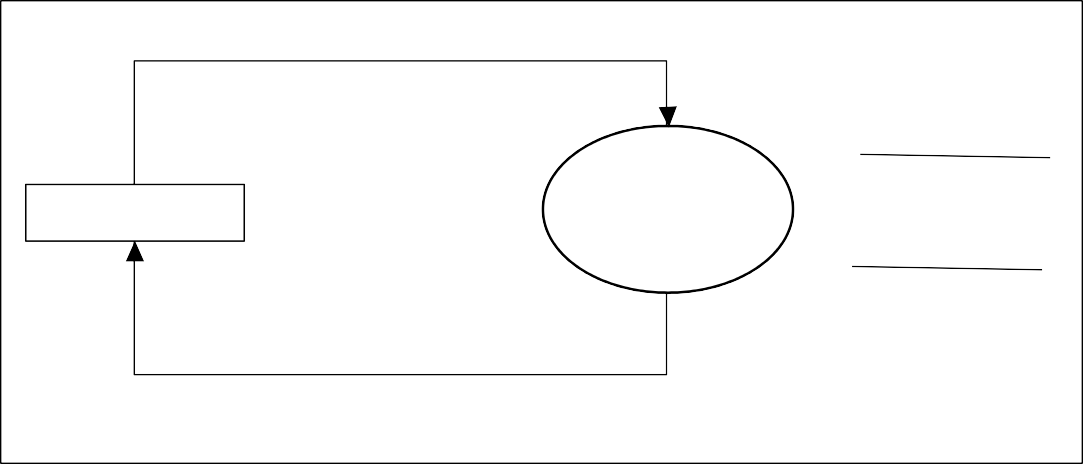
**Figure 5. Contextual Diagram of the Present System Event Specifications:**

**Event list:**

* 1. Requesting Help Process
  2. Recording of Requestor Data
  3. Locating the Requestors
  4. Generation of Reports

# Event List Diagram

Figure 6 show the event diagram of inquiry process of the present system.



- Request

Client/Admin

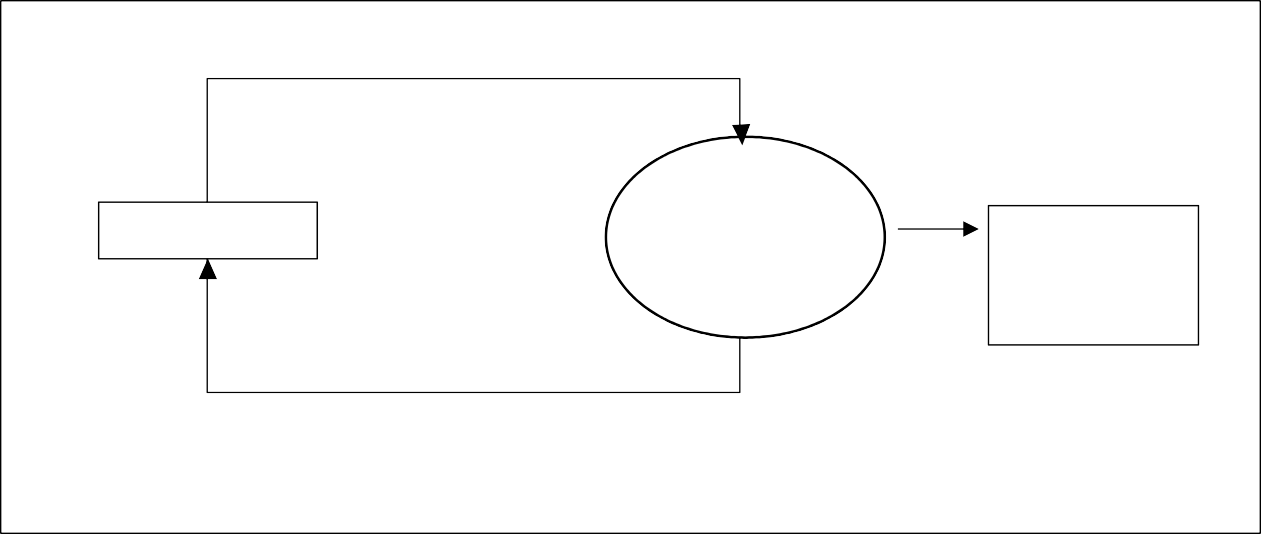
Request Help

Service

- Response

# Figure 6. Request Help Process

Figure 7 show the event diagram of the recording of requestor data of the present system.



-

Client

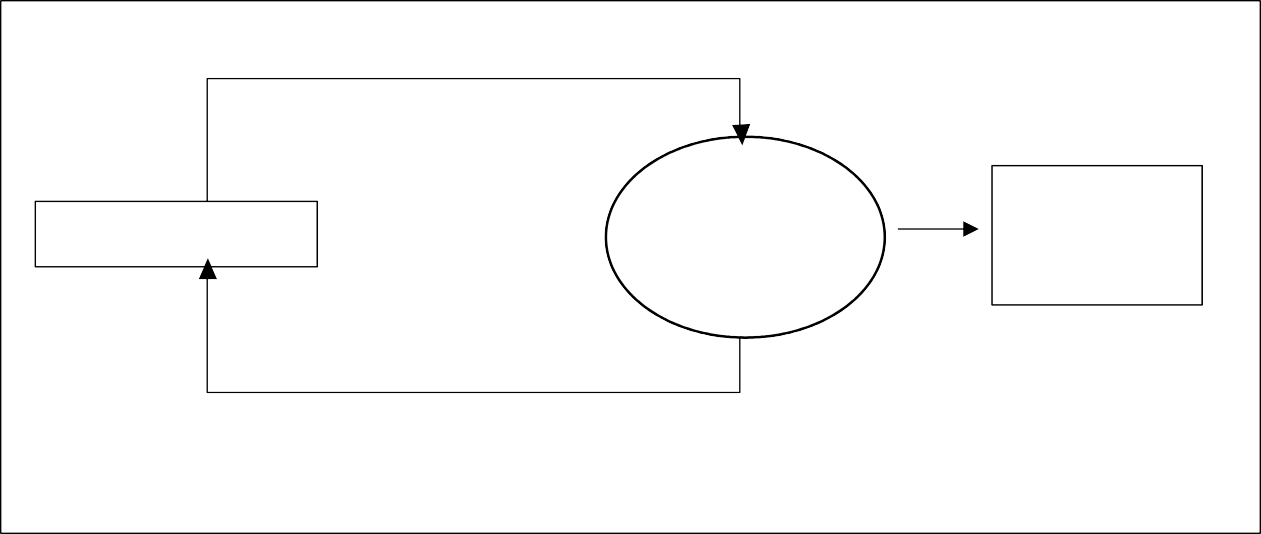
Calling and asking for information through calls

Recording of Requestor Data

Collection of clients data

# Figure 7. Recording Requestor Data

Figure 8. show the event diagram of Locating the Requestors of the present system



- Asking Location through calls

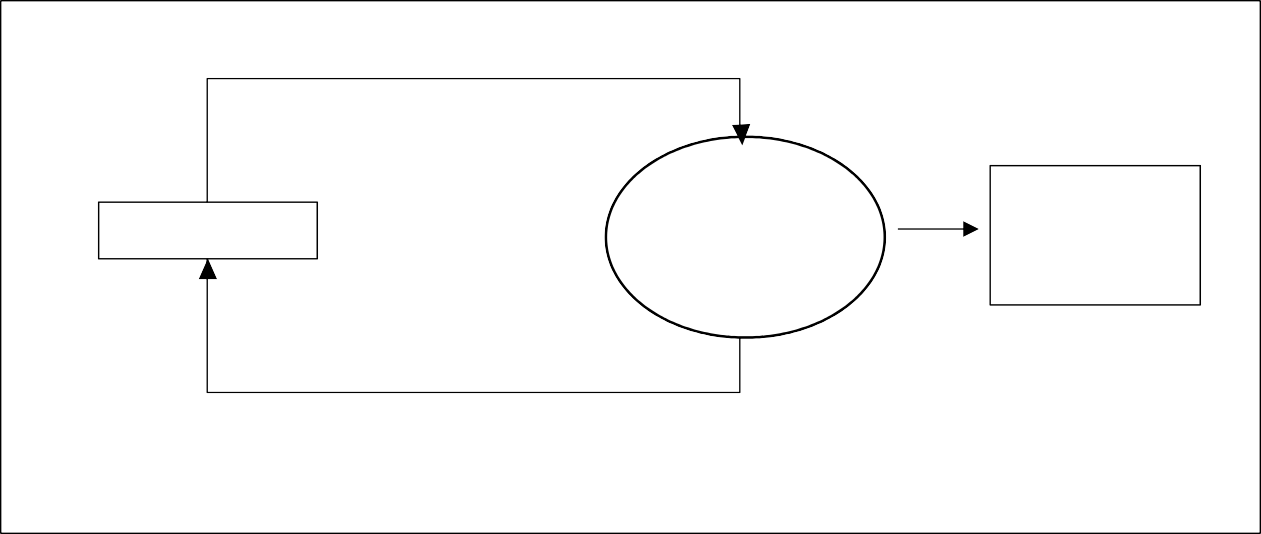
Admin/Responder

Locating Requestors

Service

# Figure 8. Locating the Requestors

Figure 9 show the event diagram of the Generation of Reports of the present system.



-

Staff

Recording of Incidents in a logbook

Generation of Reports

Report

# Figure 9. Generation of Reports

**Use Case Narrative**

A use case narrative outlines a scenario that necessitates a comprehensive understanding of the use case framework, illustrating the interaction between the user (actor or use case) to attain a goal with observable value. It should encompass more than a basic sequence of user-to-system interactions, incorporating essential elements. Every use case narrative includes pre-conditions, processes, and post- conditions. Table 5 presents the use-case narrative for accessing the system by logging in.

# Table 3. Logging in (Emergency Requestor/Emergency Responder)

|  |  |
| --- | --- |
| **Use case name** | Emergency Requestor/ Emergency Responder Log-In |
| **Actor** | Emergency Requestor/ Emergency Responder |
| **Precondition** | The app is installed on the user's device, and the requestor has a registered account. |
| **Description** | This use case describes the process of both users  logging into the app to access emergency services. |
| **Typical Course of Action** | |
| **Actor Action** | **System Response** |
| **Step 1**  The user opens the app on their mobile device. | **Step 2**  Enters their registered username and password |
| **Step 3**  The app verifies the entered credentials. | **Step 4**  Upon successful authentication, the app grants access to the main dashboard for submitting emergencies and checking incident status. |
| **Alternate Paths**   * Unsuccessful Login:   + If credentials are incorrect, the app prompts the user to re-enter. | |

Table 4 shows use case narrative for submitting an incident, outlining the user's interactions with the system.

# Table 4. Incident Submission

|  |  |
| --- | --- |
| **Use case name** | Emergency Requestor Submits Incident |
| **Actor** | Emergency Requestor |
| **Precondition** | The requestor has already logged into the app. |
| **Description** | This use case outlines the process of an emergency  requestor submitting details of an incident through the app. |
| **Typical Course of Action** | |
| **Actor Action** | **System Response** |
| **Step 1**  The app prompts the requestor to select what kind of assistance is needed (Police Emergency, Fire Emergency, or Health  Emegency) | **Step 2**  The requestor presses the “Help” button to submit the details of the incident |
| **Step 3**  The app sends an instant notification to the admin and responders, alerting them  about the submitted emergency. | . |
| **Alternate Paths**   * Incomplete Information:   If the requestor fails to provide necessary details, the app prompts them to complete the form.   * Confirmation:   The app may provide a confirmation message to the requestor upon successful submission. | |

Table 5 outlines the use-case narrative for signing up for user roles, explaining the step-by-step process for users to register and take on specific roles within the

system**.**

# Table 5. Signing Up of User Roles - User Registration and Role Assignment (Responder/Requestor)

|  |  |
| --- | --- |
| **Use case name** | User Registration and Role Assignment |
| **Actor** | New User (Emergency Responder or Requestor) |
| **Precondition** | The user has downloaded and installed the application. |
| **Description** | This use case illustrates the process a new user goes through when signing up for the application and  selecting their role, either as an Emergency Responder or an Emergency Requestor. |
| **Typical Course of Action** | |
| **Actor Action** | **System Response** |
| **Step 1**  The user clicks on the "Sign Up" button and provides essential information such as name, email, phone number, valid Id, and password. | **Step 2**  After basic registration, the user is prompted to select their role:  Emergency Responder: Individuals interested in providing assistance during emergencies.  Emergency Requestor: Individuals who may need emergency assistance. |
| **Step 3**  Depending on the selected role, the user might be asked to provide additional information. For instance, an Emergency Responder may input skills or certifications, while an Emergency  Requestor may input any relevant medical information. | **Step 4**  The admin reviews the provided information, confirms their role, and creates the user account with the provided details and assigns the chosen role. |
| **Alternate Paths**   * None | |

Table 6 shows the use-case narrative of a responder receiving an alert explains the series of actions and interactions between the system and the responder during the alert reception process.

# Table 6. Receiving Incident Notification and Dispatch – (Responder/Admin)

|  |  |
| --- | --- |
| **Use case name** | Responder and Admin Receives Incident Notification |
| **Actor** | Responder/Admin |
| **Precondition** | The responder or administrator has logged into the  application, installed it, or opened in a web environment. |
| **Description** | This use case illustrates the process a new user goes  through when signing up for the application and |

|  |  |
| --- | --- |
|  | selecting their role, either as an Emergency Responder or an Emergency Requestor. |
| **Typical Course of Action** | |
| **Actor Action** | **System Response** |
| **Step 1**  Both admin and responder  receive an emergency notification. | **Step 2**  Clicks the notification and displays the profile of the patient and their location. |
| **Step 3**  The responder takes necessary actions, such as moving to the incident location. | **Step 4**  Responder take photos of the incident and send to the admin. |
| **Alternate Paths**   * None | |

Table 7 shows the use-case narrative for managing user accounts in the admin details the process of overseeing and controlling user accounts for both regular users and administrators.

# Table 7. Account Management – (Administrator)

|  |  |
| --- | --- |
| **Use case name** | Admin Manages User Accounts |
| **Actor** | Admin |
| **Precondition** | The admin is logged into the admin portal of the application. |
| **Description** | This use case outlines the process when an admin  manages user accounts within the system. |
| **Typical Course of Action** | |
| **Actor Action** | **System Response** |
| **Step 1**  The admin navigates to the user management section of  the admin dashboard. | **Step 2**  The admin views a list of all registered users, including responders and requestors. |
| **Step 3**  The admin selects a specific  user account for management. | **Step 4**  The admin selects a specific user account for management. |
| **Step 5**  After making modifications, the admin saves the changes to the user account. |  |
| **Alternate Paths**   * None | |

Table 8 use-case narrative illustrates the editing of profile information for both requestors and responders.

# Table 8 Editing of Profile Information (Requestor/Responder)

|  |  |
| --- | --- |
| **Use case name** | Profile Editing |
| **Actor** | Requestor/Responder |
| **Precondition** | User is already logged in and in the account module |
| **Description** | Allow user to view, edit and update account |
| **Typical Course of Action** | |
| **Actor Action** | **System Response** |
| **Step 1**  User selects “Profile” tab | **Step 2**  Displays user info and list of action menu. |
| **Step 3**  User clicks the edit profile icon. | **Step 4**  Displays user info form and save button. |
| **Step 5**  Users fill in the form and click the save button. | **Step 6**  If the user clicks the save button, data stored in the database will be updated and displays a confirmation message.  **Go back to Step 4.** |
| **Alternate Paths**  Step 5   * If the form is not filled completely, A form invalidation label will display in each field. | |

Table 9 shows the use case narrative of logging out from the system.

# Table 9. Logging Out (All)

|  |  |
| --- | --- |
| **Use case name** | Logging in |
| **Actor** | All |
| **Precondition** | Users are logged in |
| **Description** | Allow users to leave the system |
| **Typical Course of Action** | |
| **Actor Action** | **System Response** |

|  |  |
| --- | --- |
| **Step 1**  User clicks Profile tab. | **Step 2**  Displays user information with a list of actions. |
| **Step 3** | **Step 4** |
| User presses Sign Out | The system will clear cached history of the |
| button | previously signed in account and redirects to sign in screen. |
| **Alternate Paths**   * None | |

**Database Design**

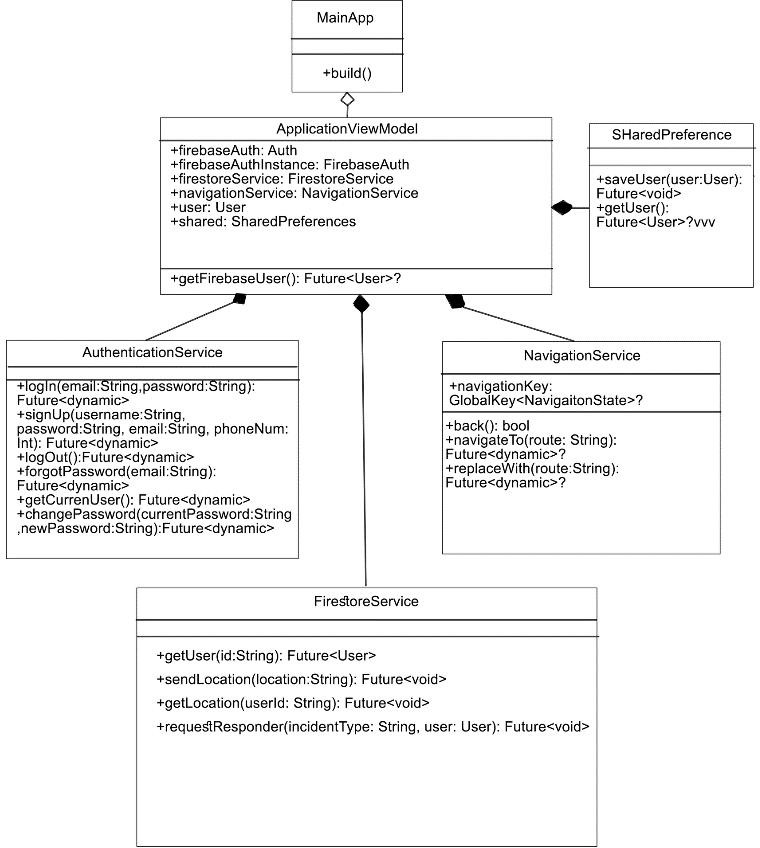
Database design involves creating a data model for the database, encompassing the essential logical and physical design decisions as well as the storage parameters necessary to formulate a comprehensive data design. definition language, which can then be used to create a database.

System design is the procedure of specifying the components, modules, interfaces, and data for a system to meet the specified requirements of the Bilar Search and Rescue Unit. The researchers' objective is to develop a new system tailored for office use. In order to meet the client's needs, various improvements were made to BISARU's existing infrastructure and operational processes.

# Class Diagram

A class diagram in the Unified Modeling Language (UML) is a graphical representation that depicts the structure of a system. It provides an overview of the classes present in the system, their attributes, operations, and the relationships or interactions among them. It provides visual representation of the system’s structure

and facilitate communication across stakeholders. Figure 8, refers to the instances depicting the most utmost processes of BERA.



# Figure 8. Class Diagram of the BERA

**Data Structure**

The following tables below were the database tables that were used in storing the information that was inputted in the system together with a collection of requirements that facilitate searching, sorting, and similar activities. It is a particular

way of organizing data on a computer so that it can be used effectively. Table 10 shows the data structure for user account credential.

# Table 10. User Account Credential

User Account Credential

|  |  |  |  |
| --- | --- | --- | --- |
| **Field No.** | **Field Name** | **Type** | **Description** |
| 1 | uid | String | User Id |
| 2 | email | String | Email |
| 3 | password | Hash | password |
| 4 | phone number | int | phone |

Table 21 shows the data structure for system users. It is organized in a way that is efficient to index and to retrieve.

# Table 11. System User

User Information

|  |  |  |  |
| --- | --- | --- | --- |
| **Field No.** | **Field Name** | **Type** | **Description** |
| 1 | uid | String | User Id |
| 2 | firstname | String | First Name Middle |
| 3 | middlename | String | Name |
| 4 | lastname | String | Last Name |
| 5 | email | String | Email |
| 6 | role | String | User Role |
| 7 | image | String | Image Uri |
| 8 | phoneNum | String | User Contact No. |

# Table 12. Incident Information

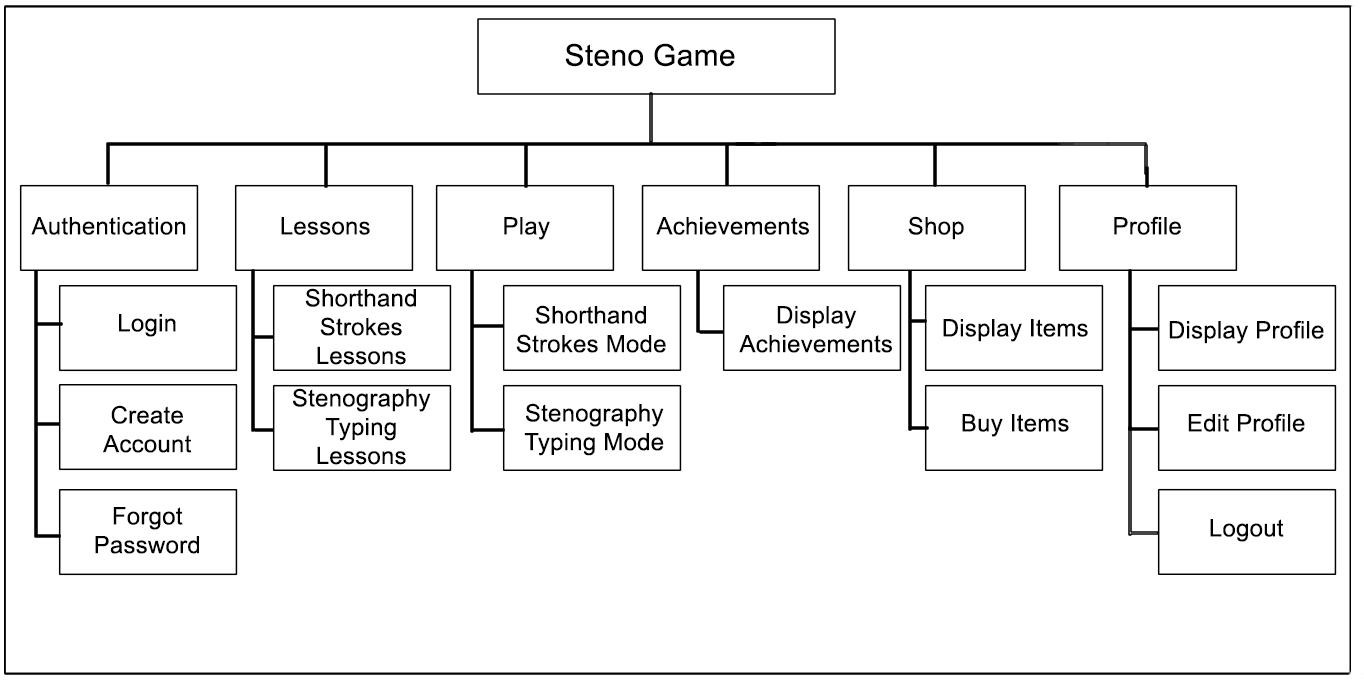
Incident Information

|  |  |  |  |
| --- | --- | --- | --- |
| **Field No.** | **Field Name** | **Type** | **Description** |
| 1 | uid | String | User Id |
| 2 | incident | String | Type of Incident |

|  |  |  |  |
| --- | --- | --- | --- |
| 3  4  5 | location username phoneNum | String String String | location Username  User Contact No. |

# Program Hierarchy

A program hierarchy is a diagram that displays the system's breakdown into its most basic, controllable layers. Each Module is symbolized as a box that holds the Module within. The high-level design or architecture of a computer program is described using a program hierarchy. Figure 12 show the program hierarchy of the BERA Application



BERA Application

Emergency Send Notification

Location Tracking

User Management

Reports

Settings

Enable to send emergency notification

Locate the place of an incident

Add and Delete users

Generate Reports

Notifies Responders

Respond to an incident

Record Daily, Weekly, and monthly incidents

# Functional Requirements

The function of a software system or component is specified by its functional requirements. Three components make up a function: inputs, behavior, and outputs. Functional requirements include processes like calculations, technical specifications, data processing and manipulation, and other particular functionality that outlines what

a system is intended to do. A prototype was used to assist determine the functional requirements. A functioning prototype will be developed in close cooperation with the staff numbers of BISARU and a few of its clients in order to strategically identify functional requirements. Also, functionalities and modules will be based upon the existing standard operating procedures of Bohol Veterinary. The resulting functionalities are as follows:

# Process Log in

FREQ 1: Access to the system should allow authorized Staff only to login FREQ 2: The system should limit module accessibility corresponding to their access rules and privileges.

FREQ 3: All information gathered should be saved and secured

# Process Requestor Identification

FREQ 4: The system should allow the recording of client’s information. FREQ 5: All information gathered should be saved and secured

# Process Sending Incident Information

FREQ 6: The system should send incident information.

FREQ 7: All information gathered should be validated by the admin

# Process of Location Tracking

FREQ 8: The system should send location to the responders FREQ 9: The system should identify the location of the incident.

# Generation of Reports

FREQ 10: The system should allow the personnel in-charge to search, view and print reports.

FREQ 11: The system should allow the personnel in-charge to generate daily, weekly, monthly and annual sales in graphical ang tabular return.

# User Account Management

FREQ 12: The system should automatically display save user account. FREQ 13: The system should allow the personnel-in-charge to add client account.

FREQ 14: The system should allow its users to modify corresponding account.

# Log out

FREQ 15: The system should execute log-out process when invoked and should clear residual caches preventing data leak.

# Non-Functional Requirement

A non-functional requirement specifies criteria that can be used to evaluate a system’s functioning rather than specific actions. Functional requirements, on 54 the other hand, describe specific behaviors or functions. The system design will include a plan for implementing functional requirements.

1. The system should be implemented with internet connection.
2. Classified modules must run on either browser or mobile device.

# Technical Requirements

Smartphones and computers have revolutionized industries, enabling efficient communication, streamlined operations, and enhanced productivity. From facilitating global business transactions to empowering educational institutions with online

learning tools, these technologies have enabled innovation and driving progress across industries, businesses, government, and education.

To ensure effective operation, computer systems and mobile applications must carefully select the appropriate hardware and software components as well as the users who will be involved in system operation. When utilized appropriately, these parts will optimize performance and allow the system to reach its maximum potential.

The term "hardware" refers to the parts of the computer's CPU that are physically located; these parts include the CPU's lower casing, microprocessor, hard drive, RAM, UPS, monitor, mouse, and keyboard. The hard drive, RAM, and microprocessor are three of these parts that are essential for processing data and enabling efficient system operations.

The physical portion of the mobile phone is referred to as the hardware component. It has a display, speaker, microphone, battery, SIM card, USB port, memory unit, camera, Bluetooth/GPS capabilities, network connectivity, and more. However, the system-on-chip, also known as the CPU, and random-access memory, also known as RAM, were the only two components that assisted in data processing.

Computer applications are commonly referred to as software. However, the computer requires this set of instructions in order to process, store, and retrieve data. Software is another term for the programming language that the researcher will use to create the aforementioned system.

# Minimum Hardware Specifications

Table 29 shows the minimum android hardware specifications need by the system. This covers the minimum hardware requirements for the system to

work as intended and expected. These specifications were chosen based on what is already available on the market and what most mobile phone systems provide.

# Table 14. Minimum Android Hardware Specifications

|  |  |
| --- | --- |
| **COMPONENT** | **SPECIFICATION** |
| CPU | DUAL CORE AND UP |
| RAM | 2 GB |
| ROM | 4GB |
| CONNECTIVITY | WIFI or LTE |

Table 30 shows the minimum desktop hardware specifications need by the system.

# Table 15. Minimum Desktop Hardware Specifications

|  |  |
| --- | --- |
| **COMPONENT** | **SPECIFICATION** |
| OS | WINDOWS, LINUX, MAC |
| RAM | 2 GB |
| CPU | INTEL CELERON, AMD A4 |

**Minimum Software Requirements**

Various software is needed for BERA to operate correctly. The minimal software specification can be found in Table 31. The Android units used to develop the system served as the foundation for the aforementioned specifications.

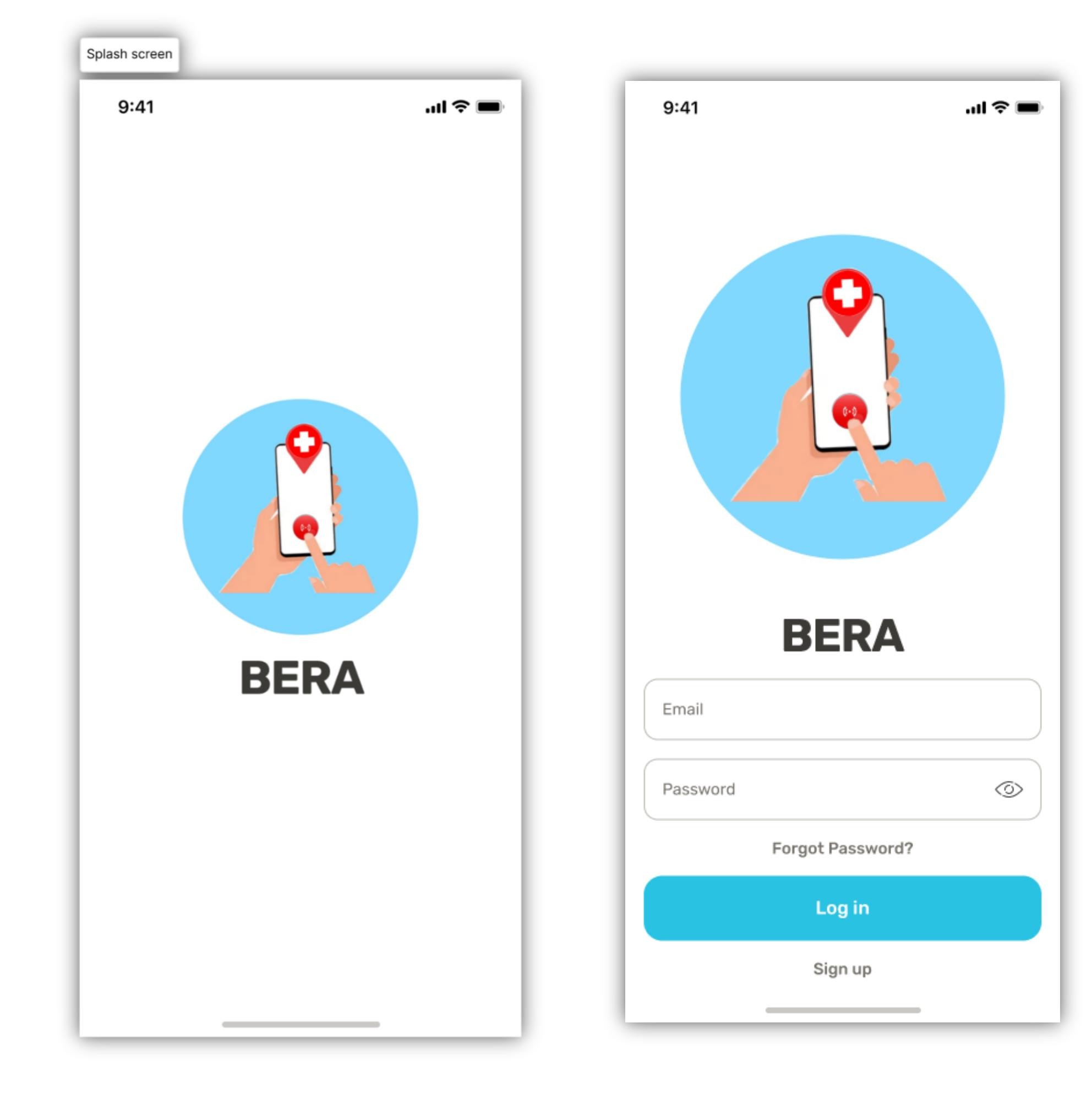
# Table 16. Minimum Software Specifications

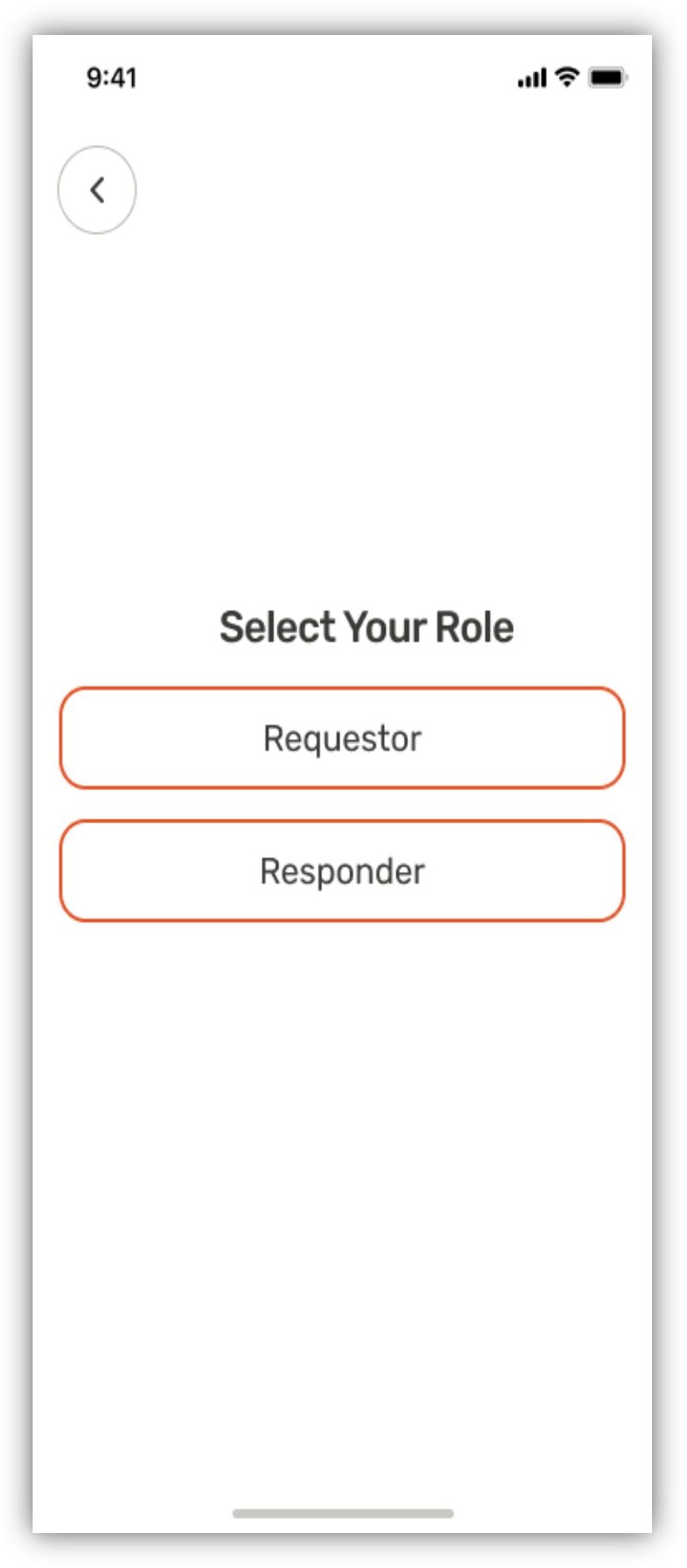
|  |  |
| --- | --- |
| **COMPONENT** | **SPECIFICATION** |
| Operating System | Windows, Ubuntu Linux, Android |
| Windows Version | At least Windows 7 |
| Ubuntu Version | Ubuntu 22.10 (Kinetic Kudu) |
| Browser | Edge, Chrome, Safari, Firefox |

**Screen Layout**

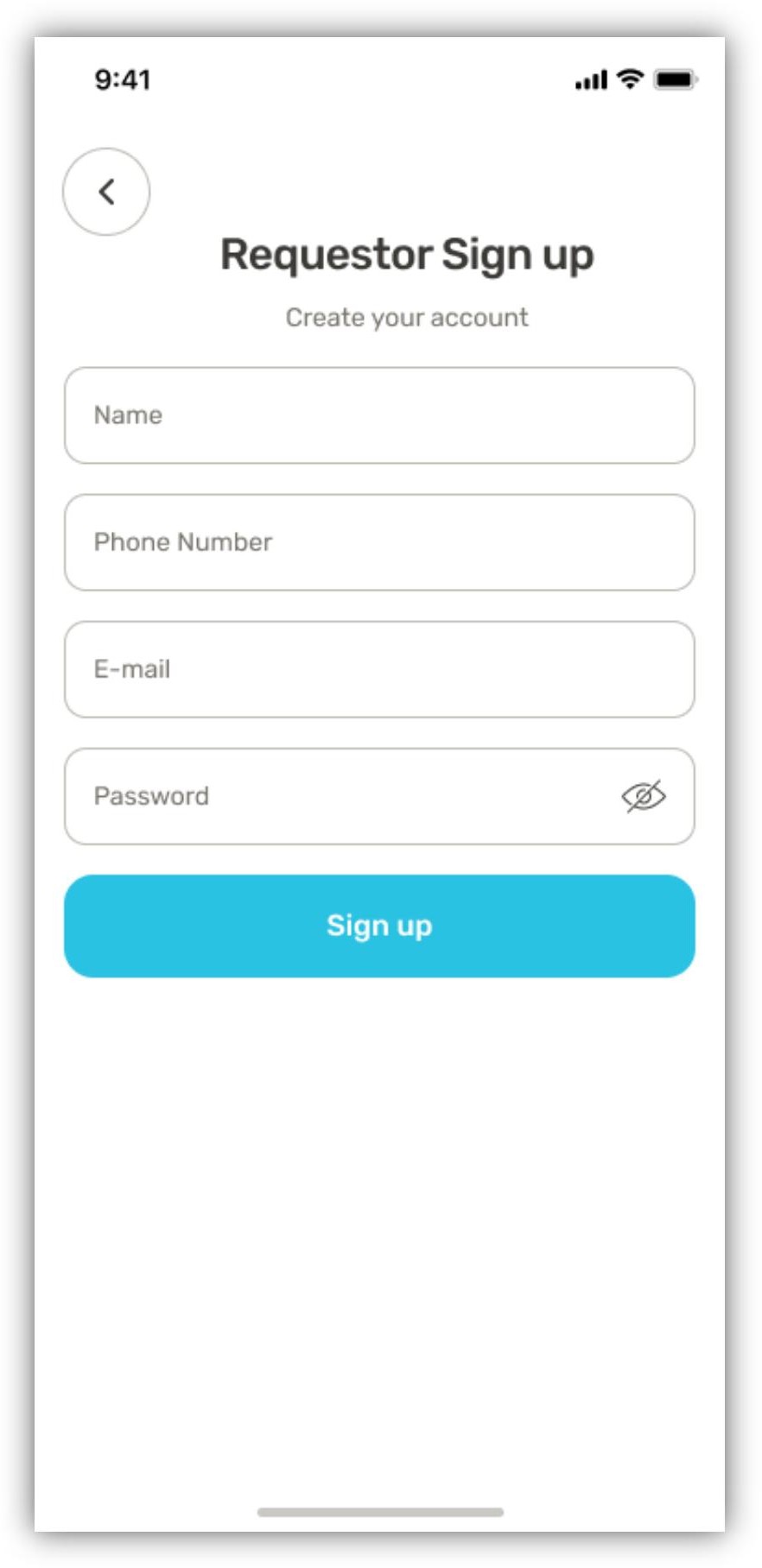
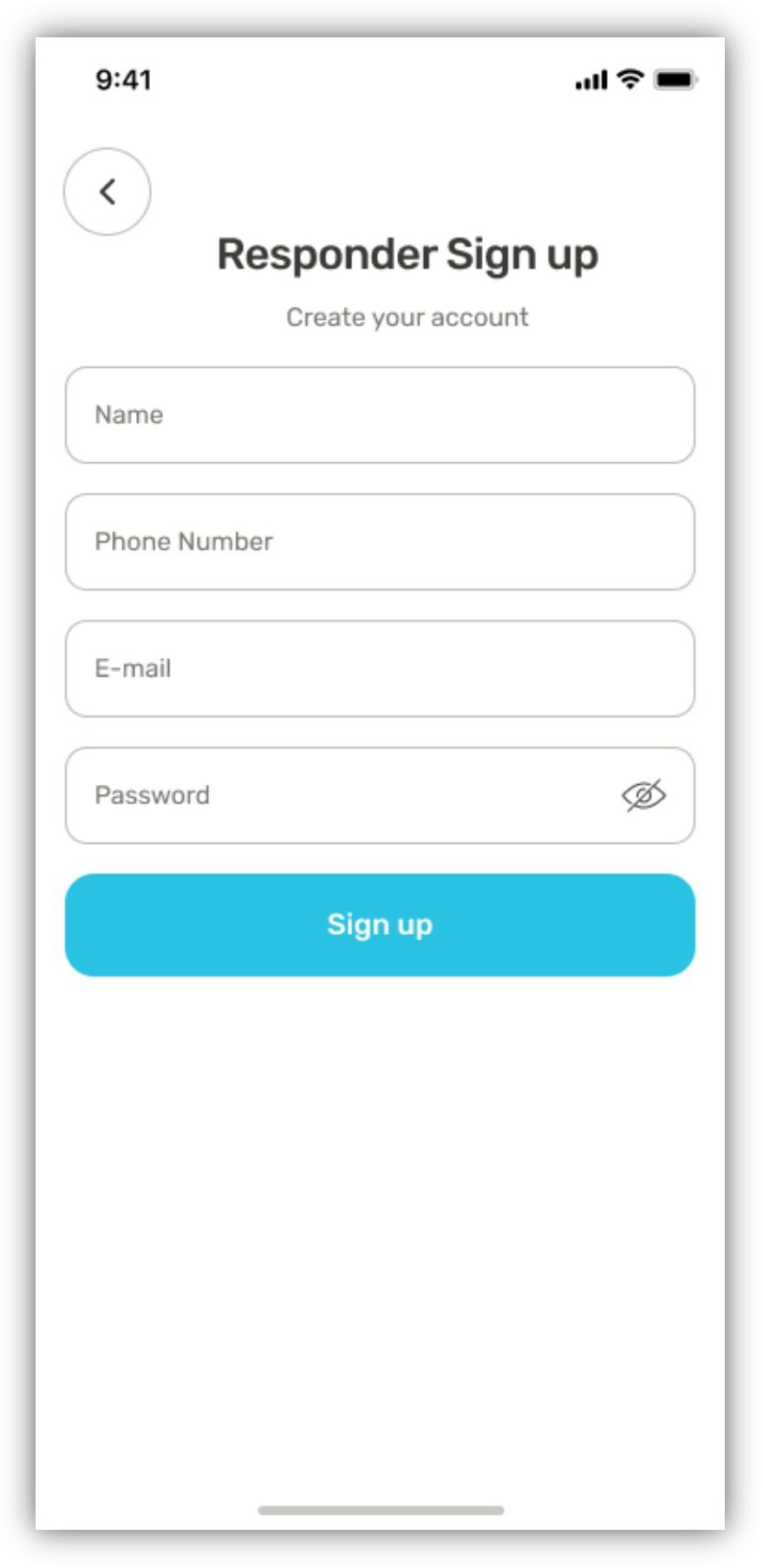
The graphical user interface's design is explained by the screen layout. It should be separated from the graphical user interface's features and encompasses a range of applications where screens can be utilized in human-machine interaction.

One of the many aspects of the system's user-friendliness is the screen layout. It should be made so that users can quickly and easily navigate the system and that the tasks they need to complete are clearly identified.

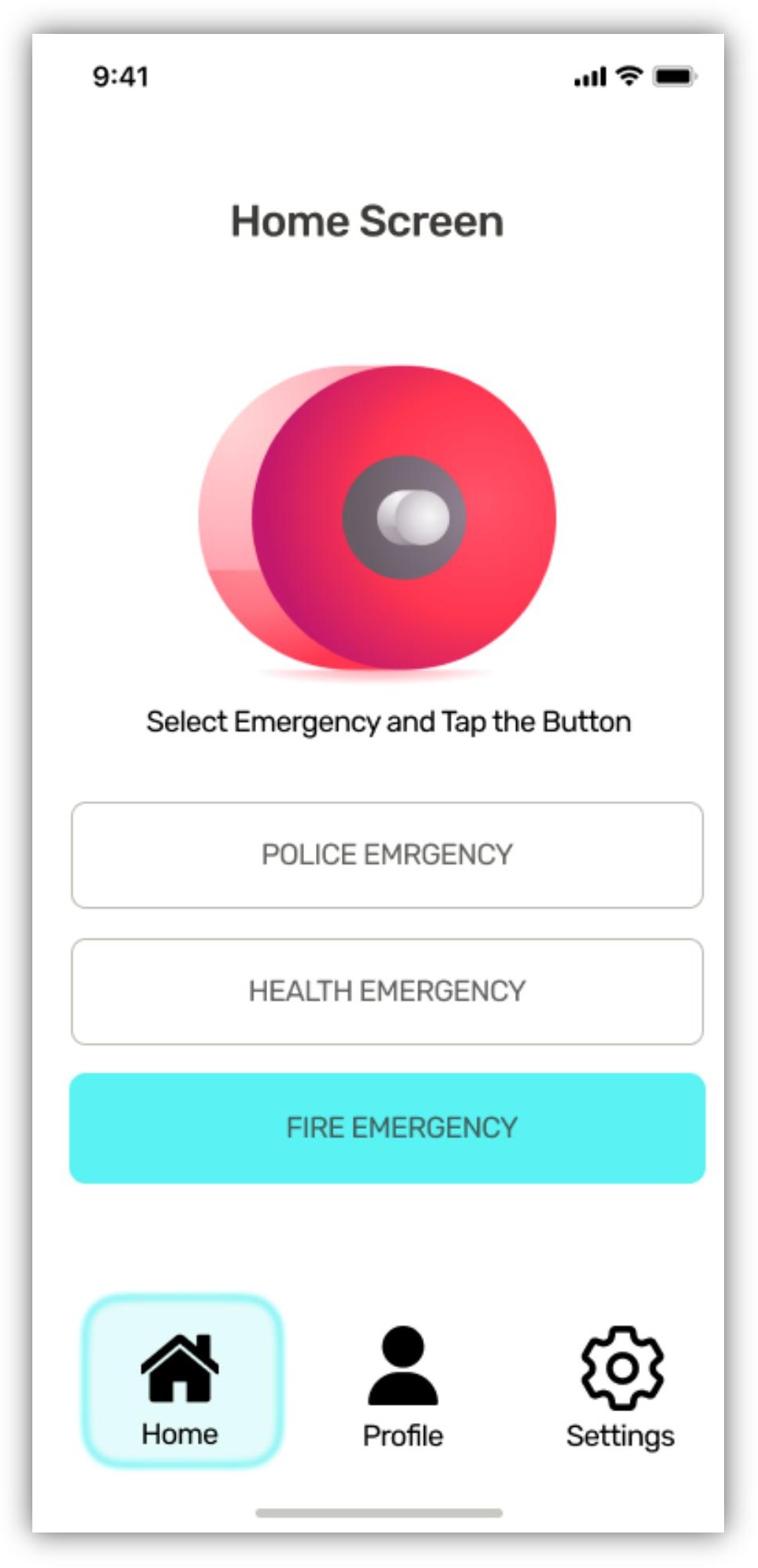
Preview 1: Splash Screen & Login Page



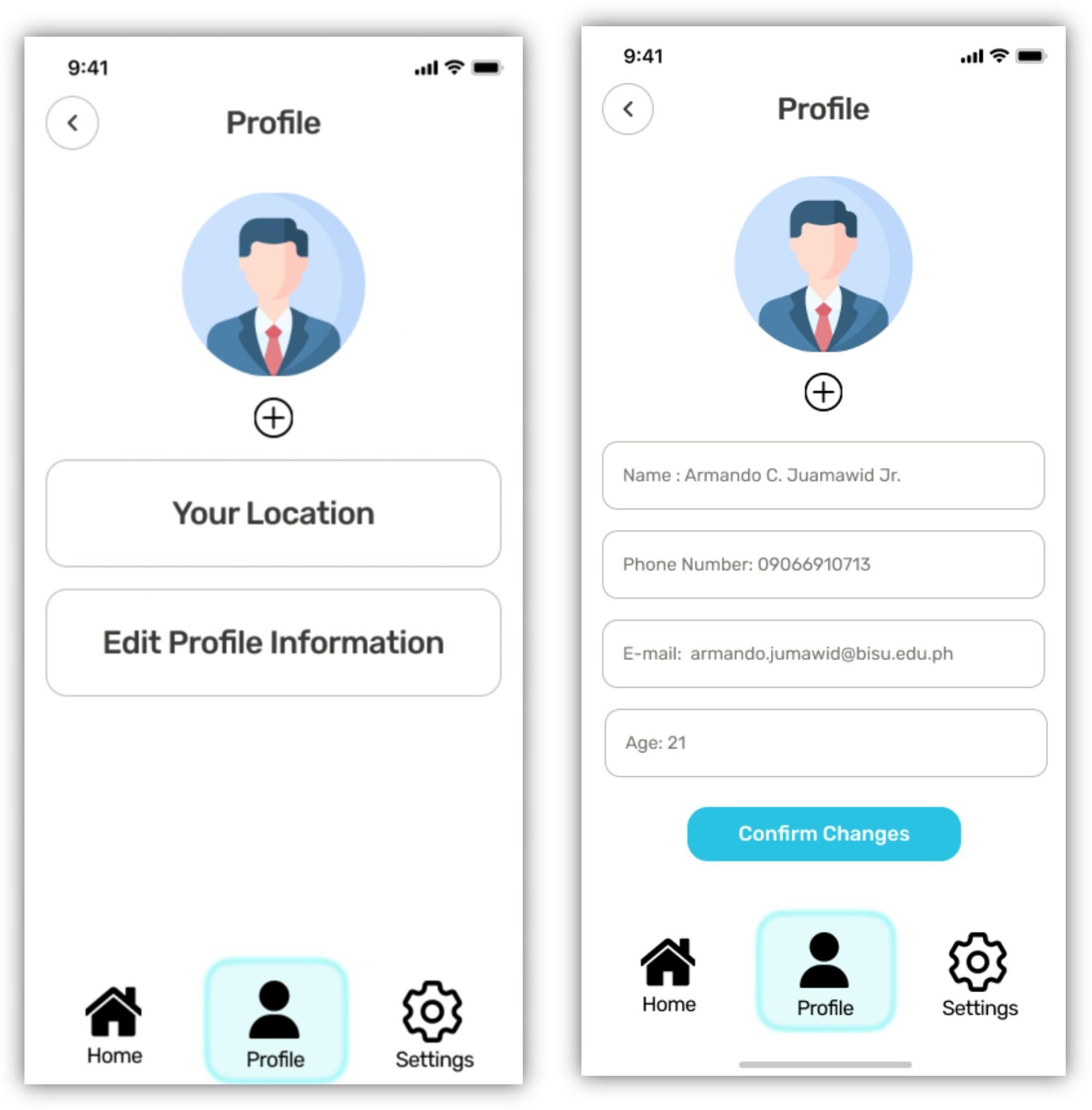
Preview 2: Role Selection

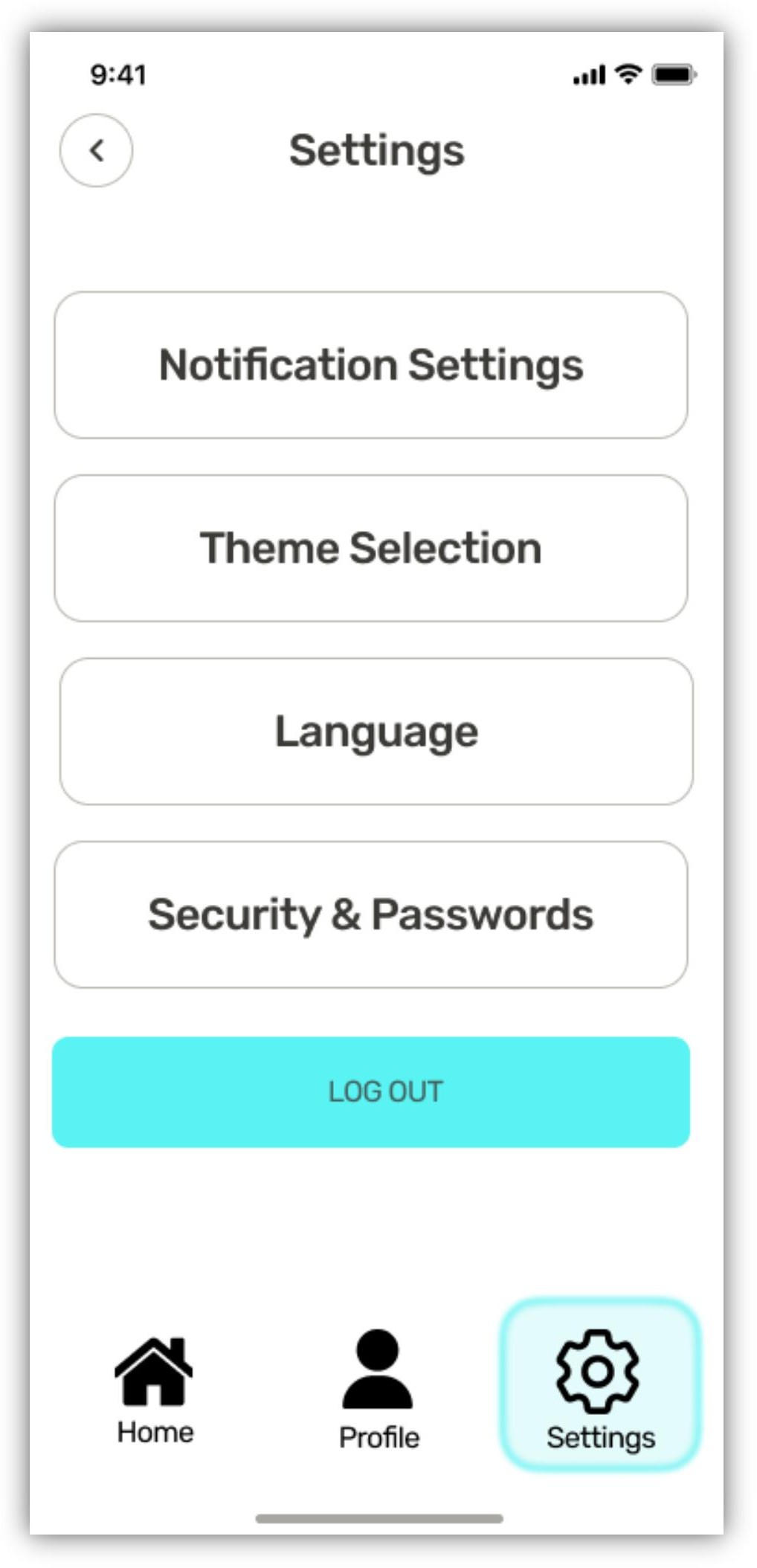
Preview 3: Sign Up Pages



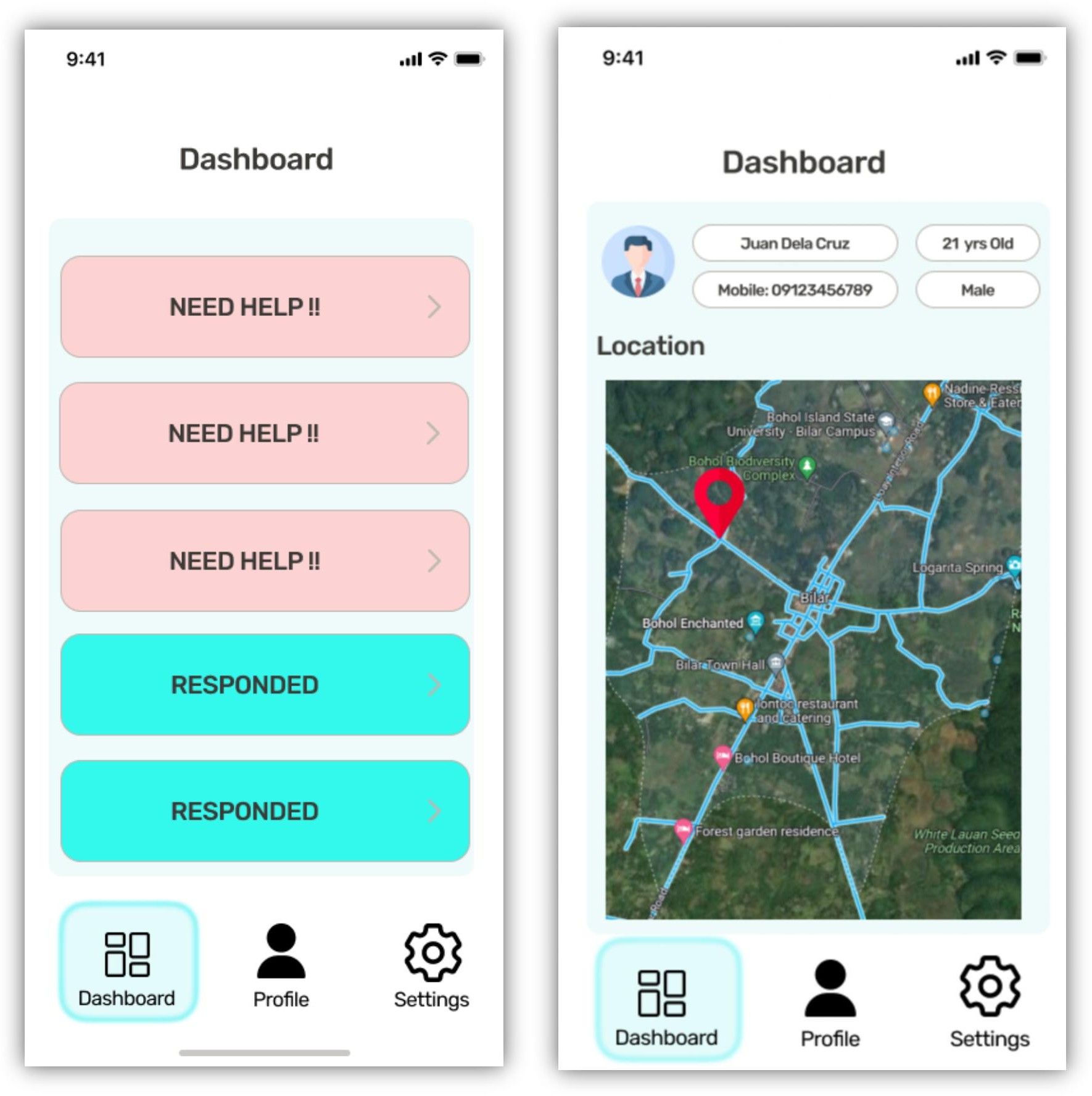
Preview: 4 Requestor Home Screen



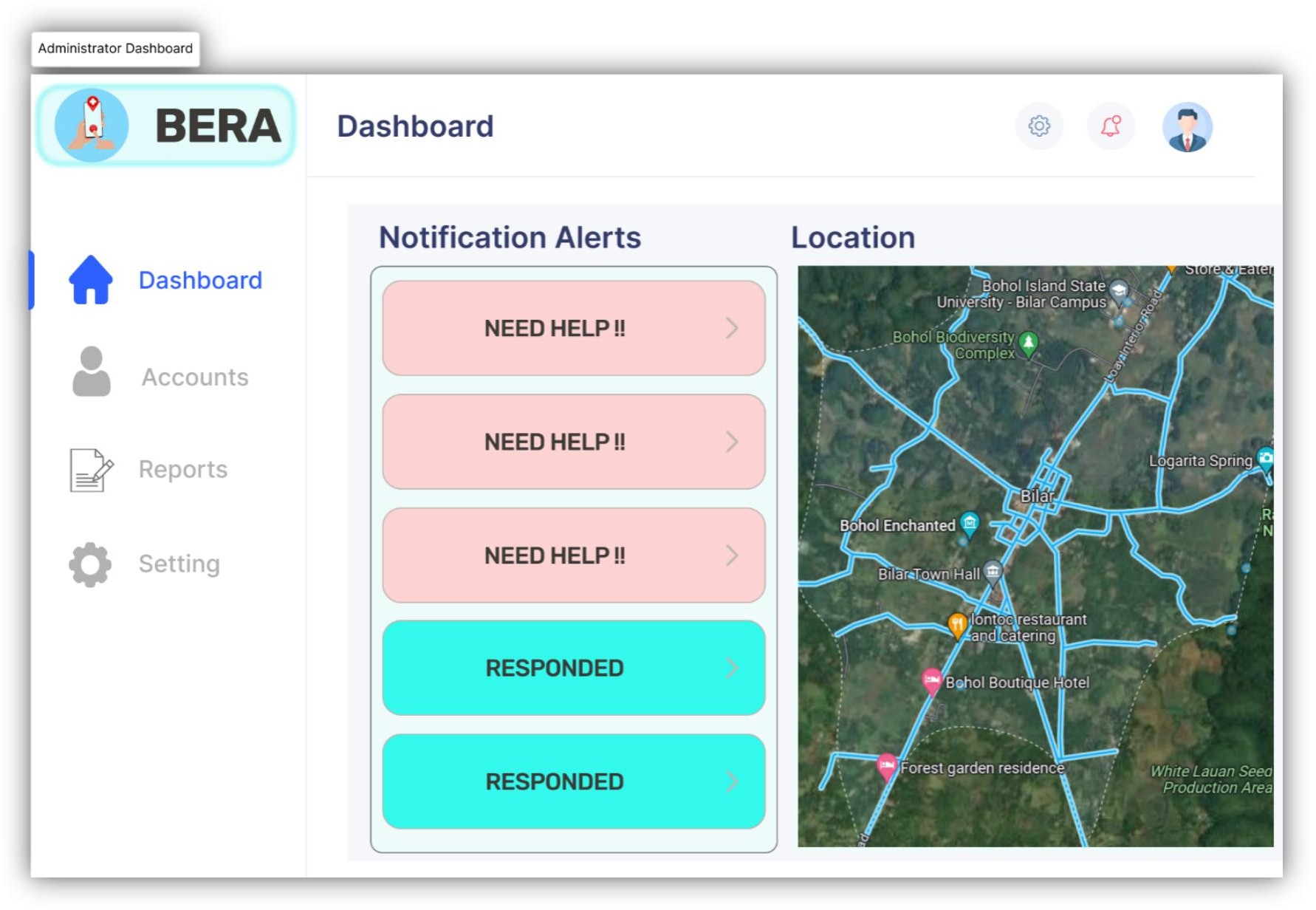
Preview 5: Requestor Profile & Edit Profile



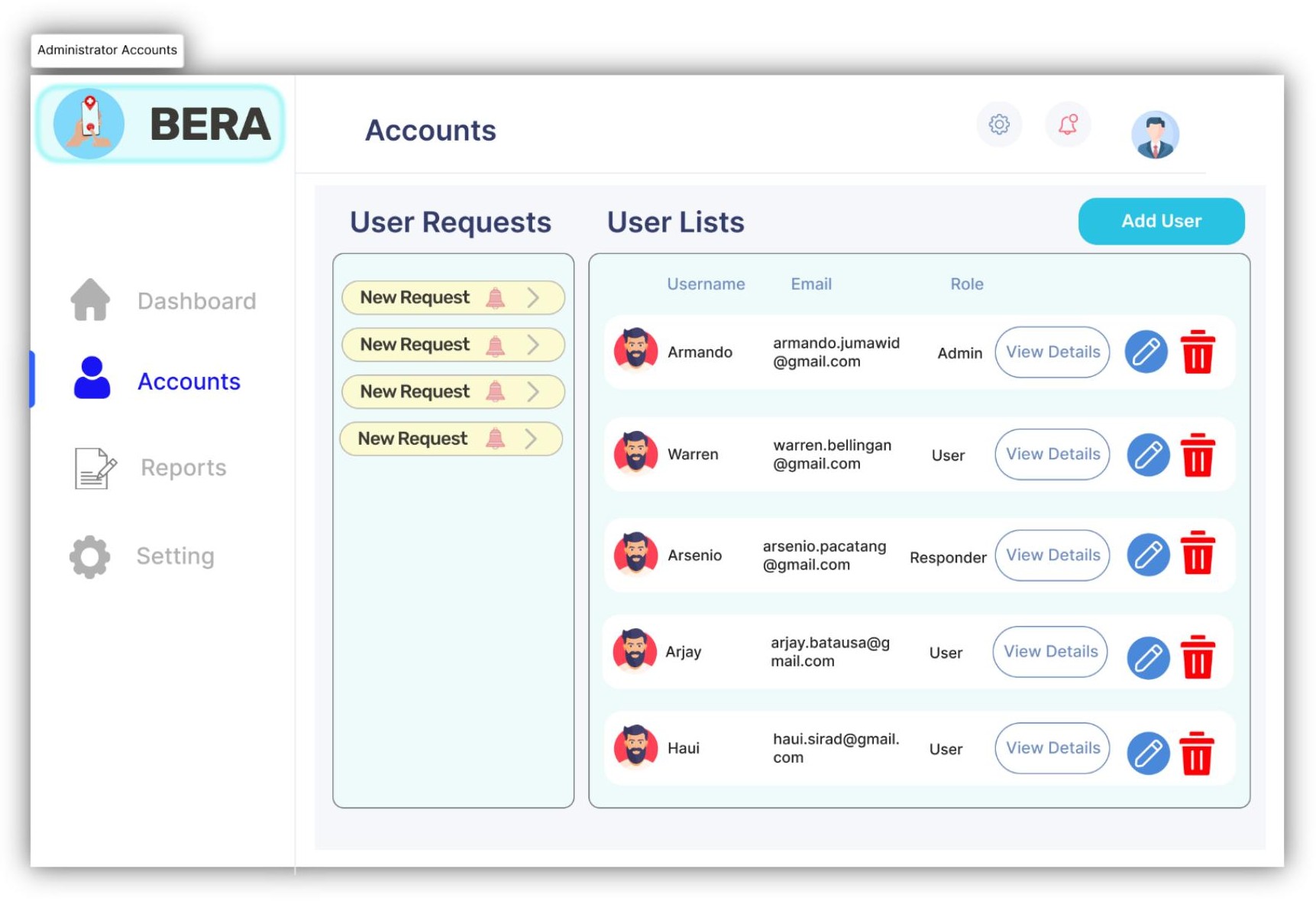
Preview 6: Settings



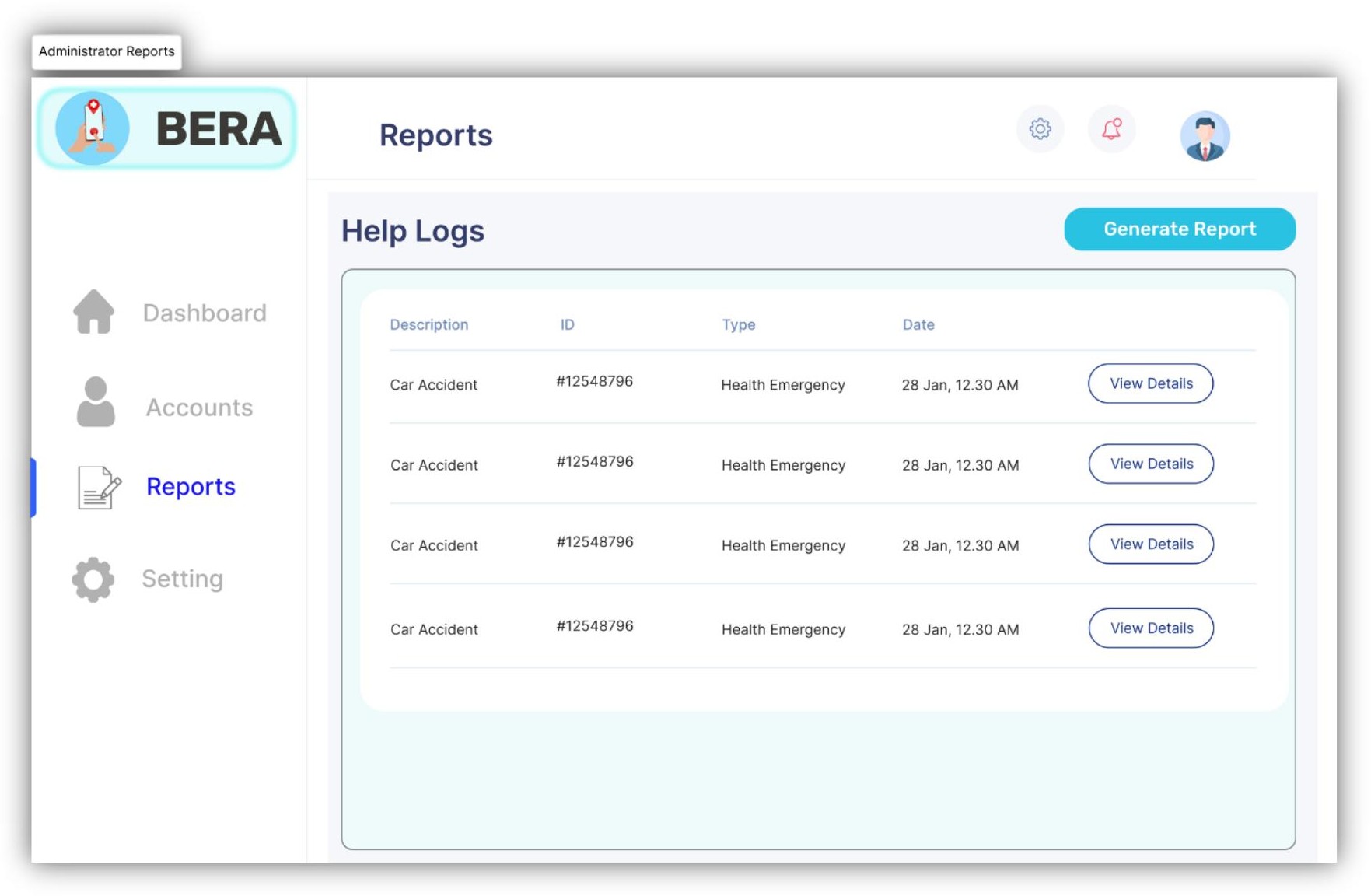
Preview 7: Responder Dashboard



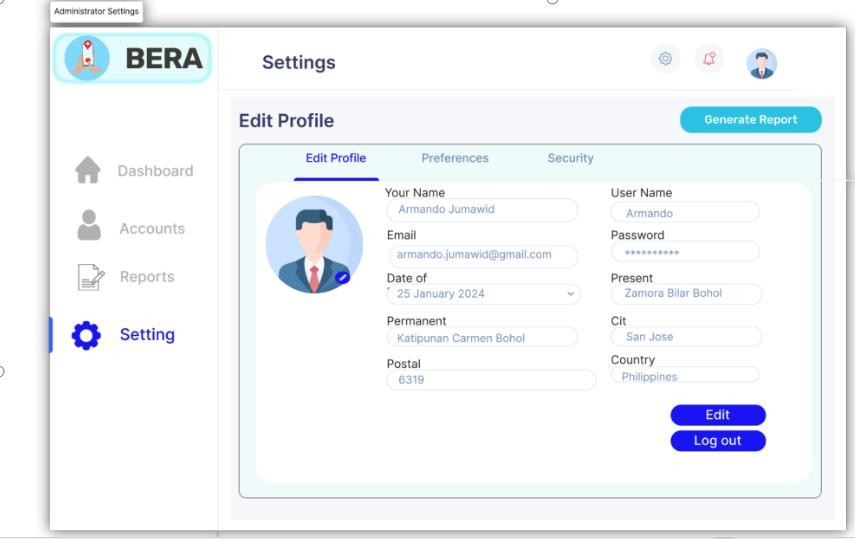
Preview:8 Admin Dashboard



Preview: 9 Admin Accounts



Preview: 10 Admin Reports



Preview: 11 Admin Settings