SAAD ZAFAR
01-134182-046
FATIMA SAEED KHAN
01-134182-015

Smart Home Application for Emergency Services

Bachelor of Science in Computer Science

Supervisor: Dr. Muhammad Asfandeyar

Department of Computer Science
Bahria University, Islamabad

June 2022

Abstract

Smart Homes initially consisted of automating systems to increase human comfort such as cooling, lighting, ventilation, etc. Over time, it has expanded to cater almost all home gadgets, improving the overall standard of living. Home safety is one such category that effectively utilizes the Internet of Things (IoT). Commercially available system are still quite expensive for middle and lower class residents. However, with the release of cheap microcontrollers, it has become ever so simple to implement smart systems at a low price.

In this report, we propose a low-cost smart home solution for emergency situations. The highly-scalable Arduino-based system detects three emergency types: intruder, fire, and gas leaks, which can be monitored via an Android application installed on the user's device. The parameters from these sensors are collected by the Arduino and stored on the cloud server, which enables it to be accessible in real-time from anywhere.

Acknowledgments

We would like to express our profound gratitude to our supervisor, Dr. Muhammad Asfand-e-yar, who played a major role in bringing our final year project to a great success. He showered us with his constant guidance and creative ideas along with relentless motivation to push us to keep working hard to achieve our goals. Secondly, we would like to thank Sir Gohar Mumtaz who approved our proposal defense. He made that defense gratifying accompanied by vivid comments and suggestions to improve in areas we lacked.

Moreover, we would like to pay our special regard to our family and friends who proved to be our greatest support and strength throughout this journey. Your prayers and love are what helped us sail through this project.

Last but not the least, we would like to thank our Lord, Allah(S.W.T), for letting us come out of every one of the troublesome stages and hard times. Whenever we got stuck in a stage, it was that continuous faith in Allah (S.W.T) that He will make a way for us so, we did not let ourselves down and kept on going. None of this would have been conceivable without Allah (S.W.T), the greatest of all.

FATIMA SAEED KHAN

RAJA SAAD ZAFAR

Islamabad, Pakistan

Saturday, April 30, 2022

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In our fast-paced world, personal safety has become a common problem with very limited solutions. The need for prompt emergency response has become integral. The Internet of Things is the new era for providing smart solutions for society. Internet of Things (IoT) is a system that allows devices to be connected and monitored from anywhere over the Internet. The integration of the Internet of Things (IoT) in various home automation has become popular over the years. These include automated light switches, irrigation systems, temperature control, appliances, etc. Automation of such systems has greatly reduced energy consumption as well as made resident's lives easier and more comfortable.

1.1 Overview

In this project, we propose a smart home solution for emergencies. To make emergency response real-time, the Internet of Things (IoT) enhances the way first responders communicate and take the necessary action. It diminishes many of the challenges to emergency response such as a weak communication network. The aim is to develop a full-scale smart emergency system that will maximize the safety of residents.

The system will detect motion, gas, or fire in real time and an alarm or alert should notify the user on the mobile app, even when the resident is away from home. The user will be given the option to press the help button to call the concerned authorities of the accident and reach the location of the user. Furthermore, a panic button will be displayed for immediate action in-case the user feels that their personal safety is being threatened.

1.2 Problem Description

When faced with an urgent situation, many people tend to be unaware of the correct and safe approach to take. Instead, they end up panicking about the situation, which sometimes, causes things to get worse. Most of the Pakistani population is unacquainted with helpline services. Due to this lack of awareness, they cannot inform the authorities of the incidents on time.

The emergency services application uses IoT sensors to detect occurrences of urgent situations. For example, a smoke sensor, upon detection of a certain level of smoke in the house, will alert the user through the Android application. The user can then make a call to alert the concerned first responders. If the user is in the house or building, they will be provided with guidelines to distinguish the smoke or to evacuate safely.

According to statistics, incidents at home are a major cause of injuries and deaths. The U.S. National Fire Protection Association states that there were at least 2500 deaths caused by 360,000 home fires, as well as 11,000 injuries. Around 60 percent of these deaths were in homes with no smoke alarms. Hence, the proposed system is also expected to reduce the number of fatalities and severe injuries.

This system will allow authorities to reach the site of emergency much quicker than the user having to manually call the helpline numbers, which is also not really an efficient approach in case of emergencies. In extreme cases, the user is unable to physically dial for help due to inhalation of toxic substances or injury. Any emergency contacts that the individual added will also be alerted through text. Hence, the system ensures greater safety and security for citizens by reducing the emergency services response time.

The goal for the user is to be able to access emergency services on a single platform as quickly as possible without hassle. The sensor data is being sent to the cloud server, so even when the resident is away, they can be carefree and monitor their homes from anywhere. Moreover, it is aimed to have a cheap, reliable, and user-friendly system because in such critical situations, no fault in the system can be afforded.

Lastly, our aim is to build a fault-tolerant system since the breakdown of the app or system may cost precious lives. Therefore, it will be made sure that system continues to operate despite any issues that may occur.

1.3 Objective

To design and develop a smart system that will detect various home emergencies (e.g., smoke, intruder, gas leak) and communicate effectively with an Android application to alert authorities of the incident.

1.4 Project Scope

This project aims to assist people in emergency cases at home. However, it is assumed that the user will have an either Wi-Fi connection or a mobile data connection available to use the services provided in the application. Similarly, the location sent to the concerned authorities via GPS would be most accurate up to 15 meters of error and it cannot send the exact location. However, people in critical condition or people in pain might not be able to speak up due to injuries which will help the responders to at least reach the destination without having to talk to them.

1.5 Chapter Summary

This chapter thoroughly explained the crisis faced by people where most of them panicked and were unaware of the corrective action to take due to a lack of awareness of helpline services. The study showed at least 2500 deaths were caused by 360,000 home fires and almost 60 percent of these deaths in homes were caused to due no smoke alarms. Therefore, our objective is to design and develop a smart home system that will detect various home emergencies such as motion, gas, and fire and communicate effectively with an Android application to alert authorities of the incident.

Chapter 2

Literature Review

Due to the security threats present in Pakistan, the need for home automation has risen greatly over the years, with several companies shifting to smart home services over the years. Several papers regarding automated systems have been published. Still, many market gaps exist as international standards are still being articulated. We analyzed a few of these papers, picking out the ones closest to our desired aim of providing emergency services, and will discuss the approaches described in them, as well as how our system will be similar or different.

The first paper to be discussed is [1]. It focuses on providing maximum accuracy in case of a fire hazard. The paper describes using Zigbee, a wireless technology designed to monitor and control low-cost and low-power IoT devices. Data communication is performed using Texas Instruments' RF chip. To avoid any errors in measurements and avoiding false alarms, the paper proposes the use of multiple sensors, which is a good solution that we will be implementing in our system. In addition, classification of smoke information by safe and unsafe values is also a useful approach to avoid low accuracy detections.

The next paper, [2], the authors use the ESP-32, an advanced microcontroller board. Its functionality includes both Bluetooth and Wi-Fi, and provides more reliability. However, due to the complexity of using the ESP-32, we have chosen to go with the Arduino Uno with ESP8266 for Internet. Furthermore, this system lacks a user interface for the user to monitor and control their homes. It uses a GPS module to obtain the coordinates of the emergency location.

The authors of [3] implemented an additional security feature that used RFID card and pin code that would allow residents to enter the home. However, they also chose to alert the user of any emergency through email, which is not very convenient since people normally do not check emails so frequently. The user should be notified directly on their smartphone so they can take action as soon as possible.

The system closest to our desired application is presented in [4], where the system is built around

the Arduino Uno board. All the sensors are connected to the Arduino. The paper presents a solution to the challenge of determining the best way to communicate with an application. It establishes this communication using the USB cable that connects the Arduino board to the computer. Furthermore, it does not receive the data on an LCD shield connected to the Arduino, since it requires 9 pins and would leave limited number of pins to connect the sensors. By receiving the data on a PC, the board has space to add more sensors.

Table 2.1: Summary of Literature Review

S#	**Title of Paper**	**Author**	**Methodology**	**Analysis**
[1]	Intelligent Smoke Alarm System with Wireless Sensor Network Using ZigBee	Qin Wu et al.	Implements Zigbee technology. Uses multiple fire hazard related sensors for maximum accuracy. Classifies smoke readings into safe and unsafe values.	The project has a motion sensor in addition. Our system also has a threshold for smoke data, above which the smoke is considered hazardous.
[2]	Efficient Smart Emergency Response System for Fire Hazards using IoT	Lakshmana Phaneendra et al.	Microcontroller used: ESP-32. Supports both Bluetooth and Wi-Fi. GPS module to acquire location coordinates.	ESP-32 is an advanced and complex microcontroller board. We chose the Arduino with ESP-8266. System lacks user interface for monitoring.
[3]	Iot-based Integrated Home Security and Monitoring System	Taryudi et al.	Used RFID card and a pin code for entering home. Users are alerted of unauthorized entry via email.	Email is not the most efficient approach for alerting user.
[4]	Arduino based Smart Home Automation System	Daniel Chioran et al.	Sensors connected to Arduino Uno. Data received on PC.	Approach is closest to our desired system. Data will be received on Firebase, not locally.
[5]	Bluetooth Based Smart Automation System	Ms. Poonam V. Gaikwad et al.	Uses Arduino Uno with a Bluetooth module. Communication is done via. Bluetooth.	Bluetooth limits the range at which the system works; hence, Wi-Fi would make truly make it a smart system.

Chapter 3

Requirement Specifications

3.1 Existing System

As discussed in the literature review, similar systems have been proposed and implemented. However, each of them have their own drawbacks or limitations. The main issue is the connection between the Arduino and the Android app. One system, as presented in [4], uses the Arduino USB cable to connect to the computer and receive data directly. Wired connections restricts one of the key aspects of portability, which is essential in modern applications.

In [5], communication is done via Bluetooth. Although it is better than wired connections, it requires the user to keep their mobile phone within a certain range of the system, even inside the home, since Bluetooth has an approximate range of 30 feet. Our system's target is to provide complete remoteness.

3.2 Proposed System

The proposed system is Arduino-based, which makes the process of adding, removing, and communicating with sensors becomes simplified through the provided I/O pins. However, establishing effective communication between the Arduino and ESP8266, as well as between ESP8266 and a server, is where it gets challenging.

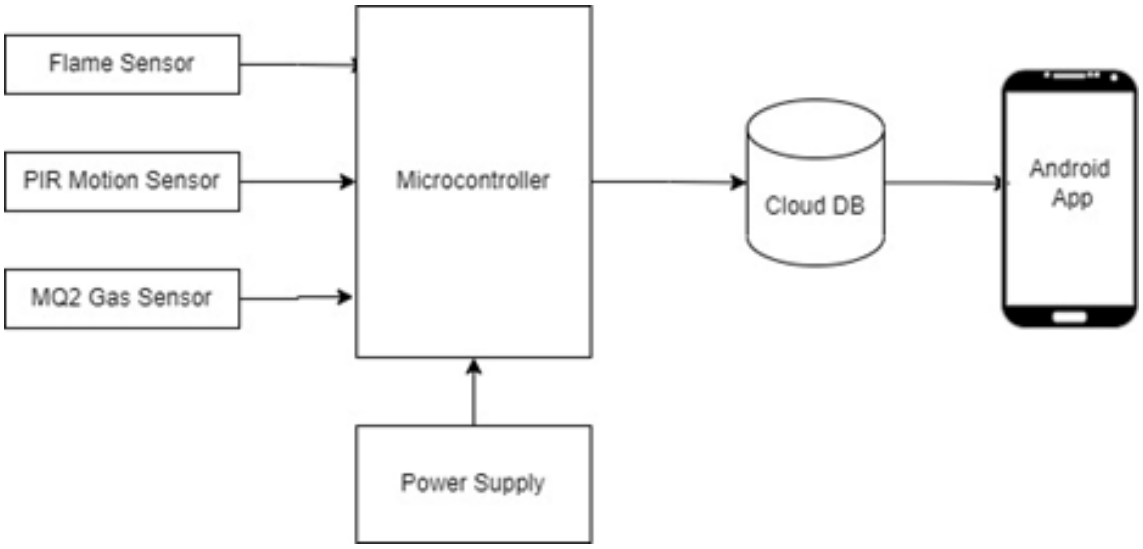

Figure 3.1: Overview of Proposed System

The above figure shows a block diagram of the Smart Home Emergency System. The three sensors are connected and configured on the microcontroller, which contains an Arduino Uno ATMega328 chip for processing sensor information, and an ESP8266 chip for Internet connectivity. The data from the microcontroller is pushed to the cloud database. The real-time database is integrated with our custom Android application. The mobile app listens for changes to the real-time database and reads the values that get uploaded there. Based on some standard values for each sensor, the system identifies the critical conditions of each phenomenon and alerts the user on their mobile application.

3.3 Requirement Specification

3.3.1 Customer Requirements

In terms of functionality, the goal is to design a system that can notify the user of an emergency as soon as it is detected in their home. The product should be cost-effective and user-friendly, and should not make the user do more work than necessary. It should also be reliable; hence, any chances of false alarms occurring should be prevented. The application design should be simple with the important icons clearly visible on the screen.

3.3.2 Functional Requirements

- Registration

  If a new user opens the application, he/she should be able to make a new account by entering their credentials.

- Login

If a user has successfully registered an account, then he/she should be able to log in to their account via username and password.

- Panic button

  Upon successful login, the user is able to contact concerned authorities (for example, Rescue 1122) with just a single click.

- Emergencies data

  Once logged in, the user shall be able to check the data of any emergency that has taken place.

- Data transfer

  The device prototype should send the sensor data to the Firebase server at the provided time intervals.

- Data retrieval

  The Android application should listen for changes in the real-time database and retrieve the values.

- Notifications

  On detection of emergency from any of the three sensors, the application should generate a notification to the user immediately.

- Action taken The user should be able to take action directly from the notification.

3.3.3 Non-Functional Requirements

- Accuracy of sensor readings

  Sensor readings should be accurate enough to be relied upon and must be tested by setting different intensities of sensors and testing them in several different ranges to make sure that they are detecting as they should be and it must be concluded that those readings are near to optimal.

- Power supply to sensors

  An adequate amount of power supply should be sent to the sensors from the Arduino board; for example, the motion sensor works to its optimal when supplied with 5 volts of power. The power supply which is more than required may cause harm to the board or the hardware components attached to it. Similarly, less power supply than required can cause the sensor's reliability to decrease.

- The efficiency of the system

The efficiency of a system depends on the internet connection. Therefore, it should perform to its best when the internet connection is stable. Fluctuations in the speed of an internet connection may result in poor performance. Hence, it has to be made sure that the internet connection is reliable before installing the system.

- Time interval of sensor readings

  We should set the time interval of 2 seconds to fetch the data from sensors to make it as realistic as possible and to make the system capable of getting the data on a real-time basis. Since we are dealing with emergencies, therefore delays more than that can cause harm to a greater extent which may include human life as well.

3.4 Use Cases

3.4.1 Login

The login page will be the initial start up page that will be launched when the user first opens the application. It will consist of username and a password fields. Upon pressing the login button, the application will search the database for the provided credentials, and if they match, redirect the user to the home page.

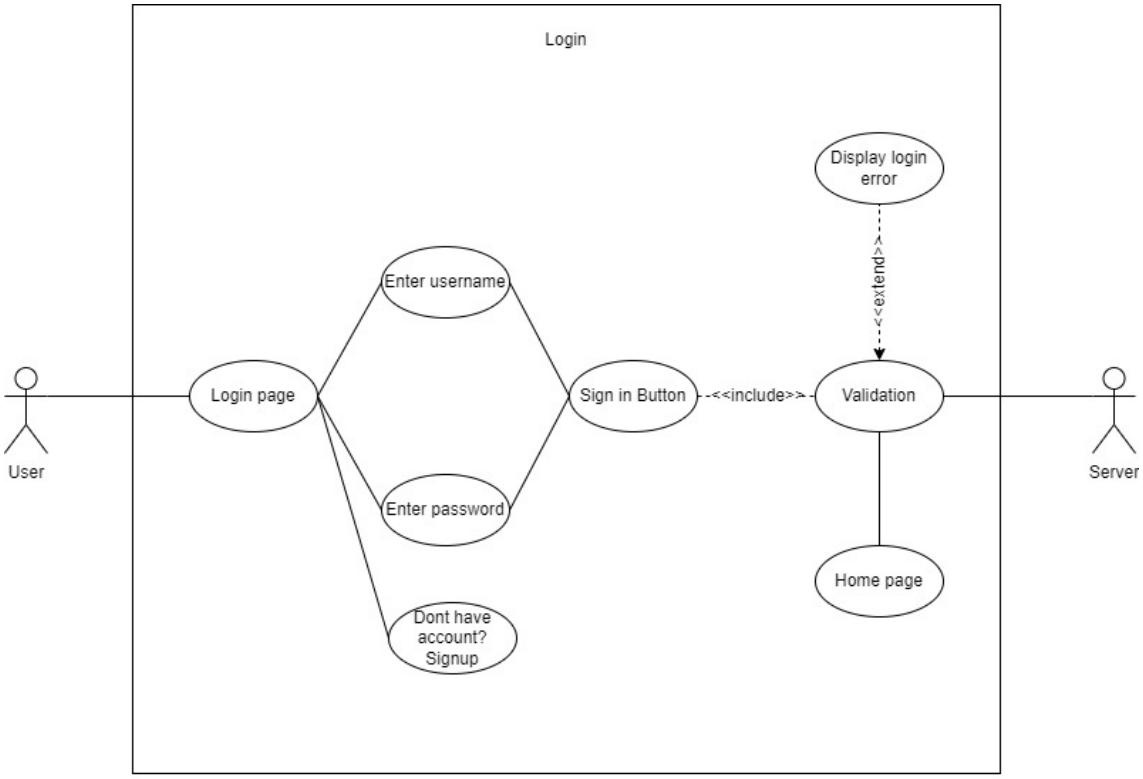

Figure 3.2: Use case - Login

Table 3.1: Tabular Description of Login

Use Case Name	Login
Use Case ID	L01
Description	Login page allows existing users to sign in and access the information on their sensors.
Primary Actors	• Raja Saad Zafar • Fatima Saeed Khan • Users
Pre conditions	User has installed and opened the application. User has enough RAM to run this application smoothly.
Post Conditions	User has successfully logged in and is directed to the home page.
Main Scenario	Provides the user interface in order to login.
Additional information	• In the event if user does not have an account, he/she is able to create a new account. • It is assumed that user has compatible version of android in order to run the application.
Basic Flow	Step 1: Fields of information is displayed. Step 2: User enters the information in order to sign in. Step 3: The data of this user is validated by checking the database. Step 4: User is taken to the home page.
Alternate Flow	• User enters email and password. • No such record exists in the database. • User is shown an error.

3.4.2 Registration

If the user is not registered on the app, a Sign Up button is provided at the bottom of the login screen which directs the user to the register page. This page consists of the fields name, username, email, and password. On signup, the application will check if any similar user exists, and if not, register a new user to the app.

Table 3.2: Tabular Description of Register

Use Case Name	Register
Use Case ID	R01
Description	Register page includes the basic information of the user such as name, username, email and so on. This page also leads to login page where you enter username and password in order to login.
Primary Actors	• Raja Saad Zafar • Fatima Saeed Khan • Users
Pre conditions	User has installed and opened the application. User has enough RAM to run this application smoothly.
Post Conditions	User has successfully Registered
Main Scenario	Provides the user interface in order to register.
Basic Flow	Step 1: Fields of information is displayed. Step 2: User enters the information in order to register. Step 3: The data of this user is saved into database.
Alternate Flow	• User enters their information in the fields. • The username already exists. • The user is not allowed to register unless they provide unique details.

3.4.3 Homepage

Figure 3.4 shows the home page which will display all the sensors that the user has installed in their home. Clicking on any of them will show the user if there is any emergency present in their home at a certain time. There is also a panic button at the bottom of the page that the user can press in urgent situations. This button will generate a phone call to the Rescue Helpline. The tabular description of the homepage is shown in Table 3.3.

3.4.4 Notification Service

Figure 3.5 describes how the application will receive data from the server and push a notification on the smart-phone when the value received indicates an emergency situation. The user can then click on the notification that they received, which will open the phone's dialer with the respective helpline's number.

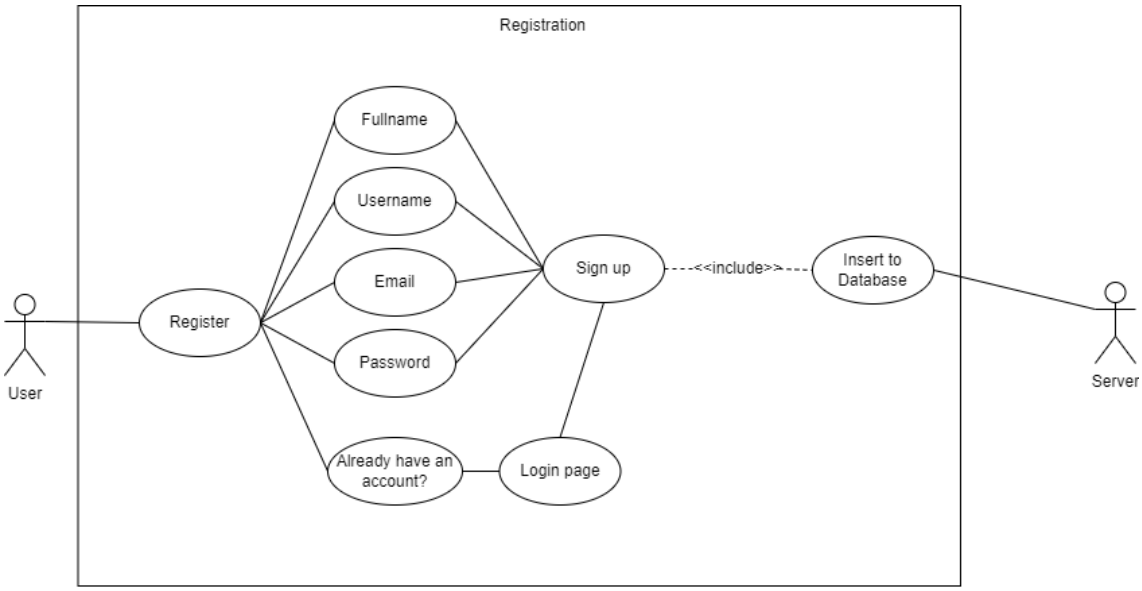

Figure 3.3: Use case - Register

Table 3.3: Tabular Description of Homepage

Use Case Name	Home Page
Use Case ID	HP12
Description	Home page shows a menu where different options are available to the user. This include basic features of the application such as emergency call, first aid and so on.
Primary Actors	Registered users.
Pre conditions	User has successfully logged into the application.
Post Conditions	User may have used one of our services.
Main Scenario	Provides the user interface to the user which involves our services of application.
Additional information	User is not necessarily the one in emergency who is about to use of services.
Basic Flow	**Note**: User is about to make an emergency call. Step 1: Menu of home page is displayed. Step 2: User selects the required service. Step 3: Dialer is opened to make emergency call.
Alternate Flow	**Note**: User has already created an account. • User clicks the option of menu cards. • Information received by the sensor is displayed.

Table 3.4: Tabular Description of Notification Service

Use Case Name	Notification Service
Use Case ID	NS12

Description	Notifications are generated when the data received by the Firebase database indicate a positive emergency case.
Primary Actors	Registered users.
Pre conditions	Value received is either 1 (for digital sensors) or above a threshold (for analog sensors)
Post Conditions	User may have used one of our services.
Main Scenario	Allows user to respond to the situation as quickly as possible.
Basic Flow	Step 1: Firebase sends realtime data to the application. Step 2: Application receives data and upon detecting emergency value, pushes notification on the device. Step 3: User sees notification, and clicks on it to open dialer.
Alternate Flow	The data received is not an emergency value. Hence, no notification is generated.

3.5 Chapter Summary

This chapter elaborated the requirement specifications of the system from three perspectives: customer requirements, functional requirements, and non-functional requirements. It also discussed similar, existing implementations or methodologies of our project, and a comparison of our project with those systems. Furthermore, we provided an overview of the system architecture, discussing the various hardware and software components and how they will work together to achieve the desired functionality.

In order to visualize the functional requirements, we have created use case diagrams of how the application will work on the user's end. It allowed us to analyze how the system will interact with actors. These use cases will be translated into the design phase of the project and allow us to prioritize certain features during the development phase.

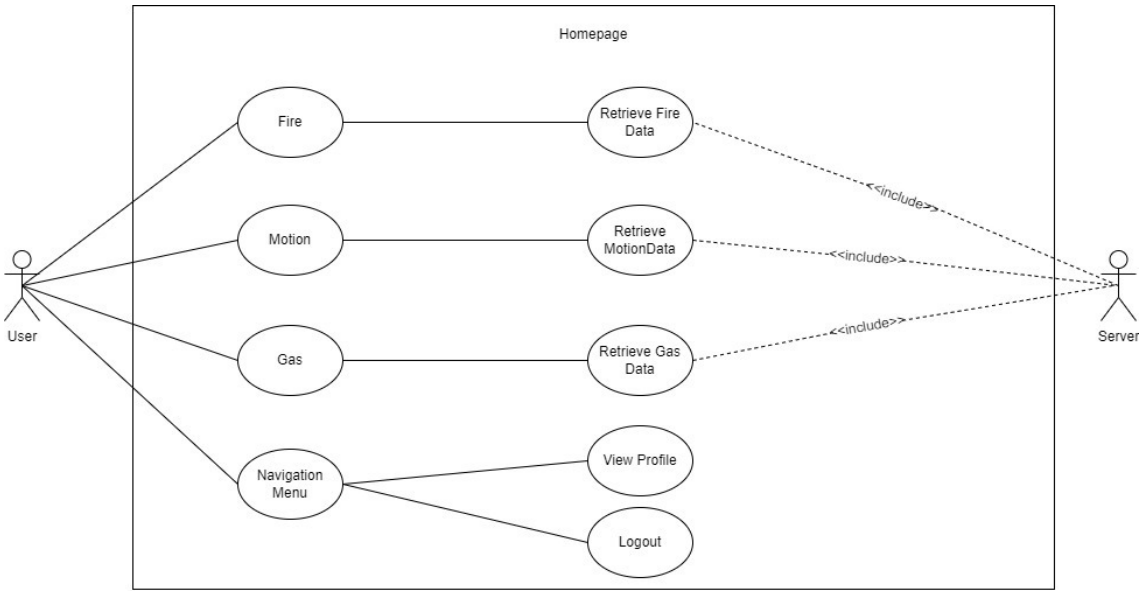

Figure 3.4: Use case - Homepage

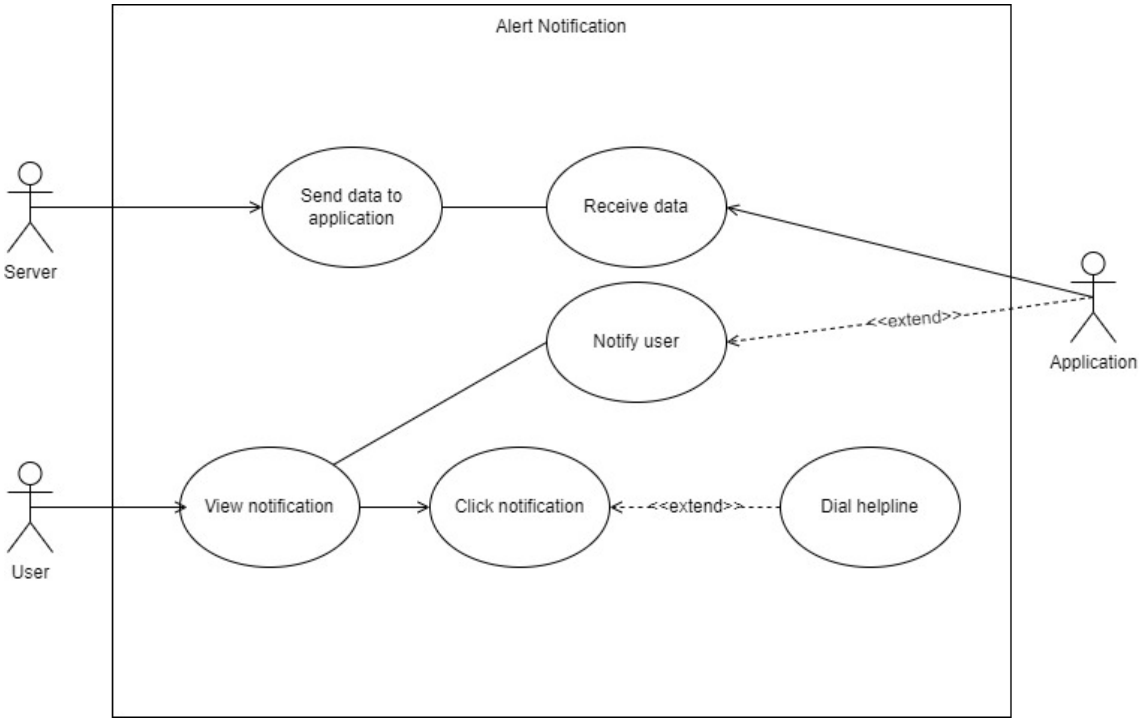

Figure 3.5: Use case - Notification Service

Chapter 4

Design

System design refers to the process of formulating the components, architecture, interfaces, and data of a system so that it can work together to satisfy the user requirements. Our proposed system consists of both hardware and software aspects which will be discussed in detail in this chapter. In addition, this chapter gives an abstract view of the proposed system architecture.

4.1 System Architecture

Because our system is Arduino-based, the process of adding, removing, and communicating with sensors becomes simplified through the provided I/O pins. However, establishing effective communication between the Arduino and ESP8266, as well as between ESP8266 and a server, is where it gets challenging.

The figure below shows a block diagram of the Smart Home Emergency System. The three sensors are connected and configured on the microcontroller. The microcontroller processes this sensor information and pushes it to the cloud database. Based on some standard values for each sensor, the system identifies the critical conditions of each phenomena and alerts the user on their mobile application.

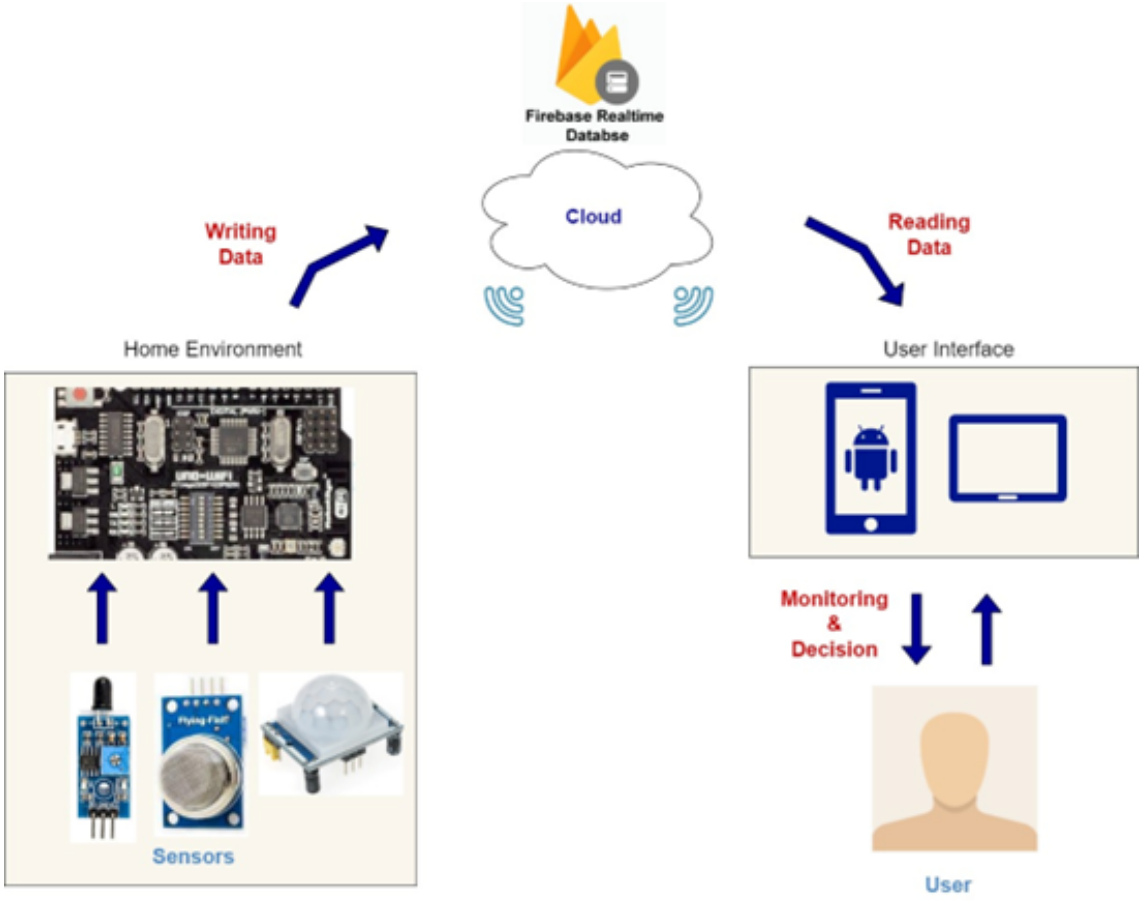

Figure 4.1: System Architecture

4.2 Design Constraints

The Arduino UNO Wi-Fi is a retired product and replaced by UNO Wi-Fi Rev2. Due to the unavailability of this product in Pakistan, we had to purchase a custom board by RobotDyn, which comes with two processors, Atmel ATmega328 microcontroller, and ESP8266 WiFi chip with 8 MB flash memory. This board comes with dip switches to connect the processors. In addition to the limitations of development boards, the use of wiring to implement the hardware causes a trade-off in the communication speed.

4.3 Hardware Aspects

- RobotDyn Uno+WiFi Board

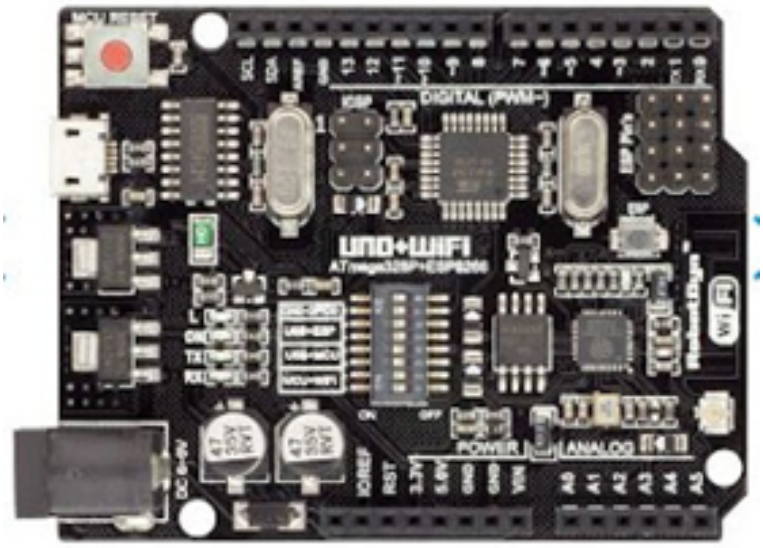

Figure 4.2: RobotDyn Uno+WiFi Board

The RobotDyn UNO+WIFI R3 is a customized board that combines the Arduino UNO R3 with the ESP8266 Wi-Fi chip. The advantage of having a board with integrated Wi-Fi is that there is no need for any external connections. This board has three microchips, 1) ATmega328 for loading the programs to the Arduino, 2) CH340 to connect the computer to the board via USB connection, and 3) ESP8266, which is the Wi-Fi chip. The board has dip switches to allow for uploading code on to both ATmega328 and ESP8266, but not at the same time.

	1	2	3	4	5	6	7	8
CH340 connect to ESP8266 (upload sketch)	OFF	OFF	OFF	OFF	ON	ON	ON	NoUSE
CH340 connect to ESP8266 (connect)	OFF	OFF	OFF	OFF	ON	ON	OFF	NoUSE
CH340 connect to ATmega328 (upload sketch)	OFF	OFF	ON	ON	OFF	OFF	OFF	NoUSE
Mega328+ESP8266	ON	ON	OFF	OFF	OFF	OFF	OFF	NoUSE
Both MEGA328P and ESP8266work independently	OFF	OFF	OFF	OFF	OFF	OFF	OFF	NoUSE

Figure 4.3: Dip Switch Configurations for the Board

- Flame Sensor

  The infrared IR flame sensor module was used to identify if fire is present.

- Gas Sensor

  The MQ2 gas sensor was used to detect flammable gas such as hydrogen, propane, and liquified petroleum gas, or LPG.

- PIR Motion Sensor

  These sensors detect heat energy to sense whether a person or animal has moved in the surrounding environment.

4.4 Software Aspects

- Firebase

  Google Firebase is a software that allows development of Android, iOS, and web applications. The Firebase Realtime Database is the tool that we use in this project.

- Java Programming Language

  Android Studio is the official IDE created for the development of Android applications. The language used to develop the app was Java.

- Arduino Programing Language

  The Arduino IDE is used to program the Arduino board and read input from the sensors. The code is written in C++ with special functions included as an addition.

4.5 Database Design

Two separate databases were required for this system. A local database was created on SQLite that stored for login and registration. This database consists of the following columns: username, name, email address, and password.

The other database is for monitoring the sensor data which will be sent by Arduino system. For this, we have used the Firebase Realtime Database, a no-SQL database which follows a document database model and stores information in a JSON format. The reason we have chosen a non-relational database as our main database is because SQL databases need to have a pre-planned, rigid structure. No-SQL databases offer more flexibility in terms of scalability and adjusting to changing scenarios.

4.6 GUI Design

A user-friendly and simple GUI was developed for this system as complex GUIs may be counteractive in emergency situations. The goal was for the user to be able to quickly find the options that they require at the moment. Large icons and card views were used along with images of each icon for the ease of the user and improved visibility.

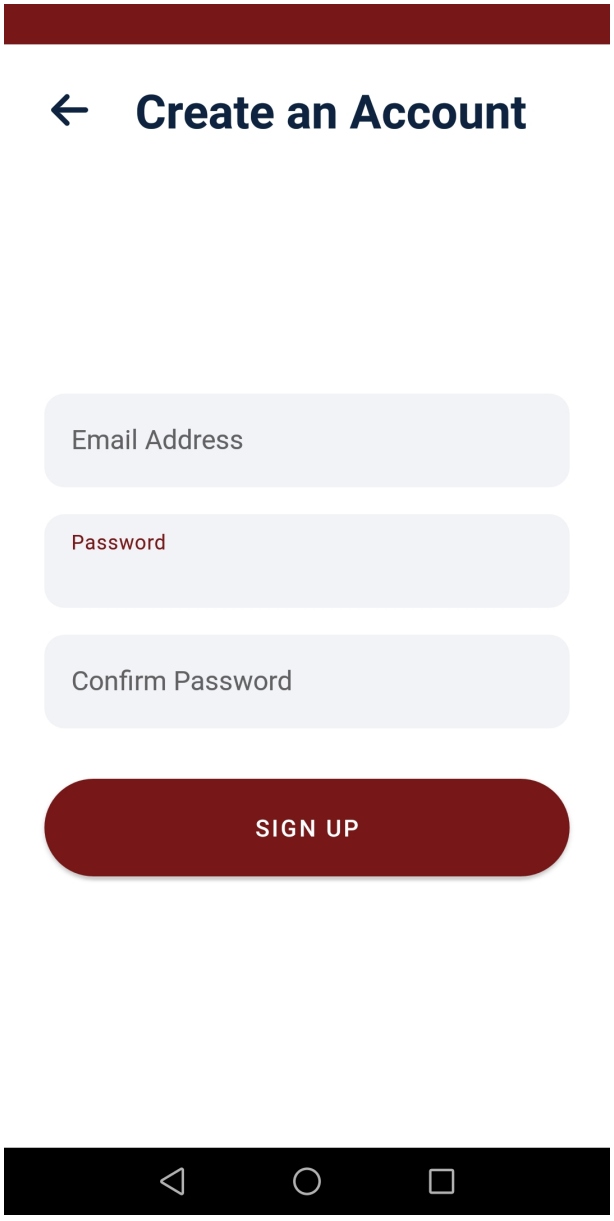

Figure 4.4: Registration Page

When the user first opens the application, they will be required to create an account in order to login. This is a simple registration page of our mobile app where new users can enter the mentioned credentials in fields and get registered on the system. Once the information is entered, the signup

button is clicked which will successfully register the user in our database. Once registered, user can login to use the application.

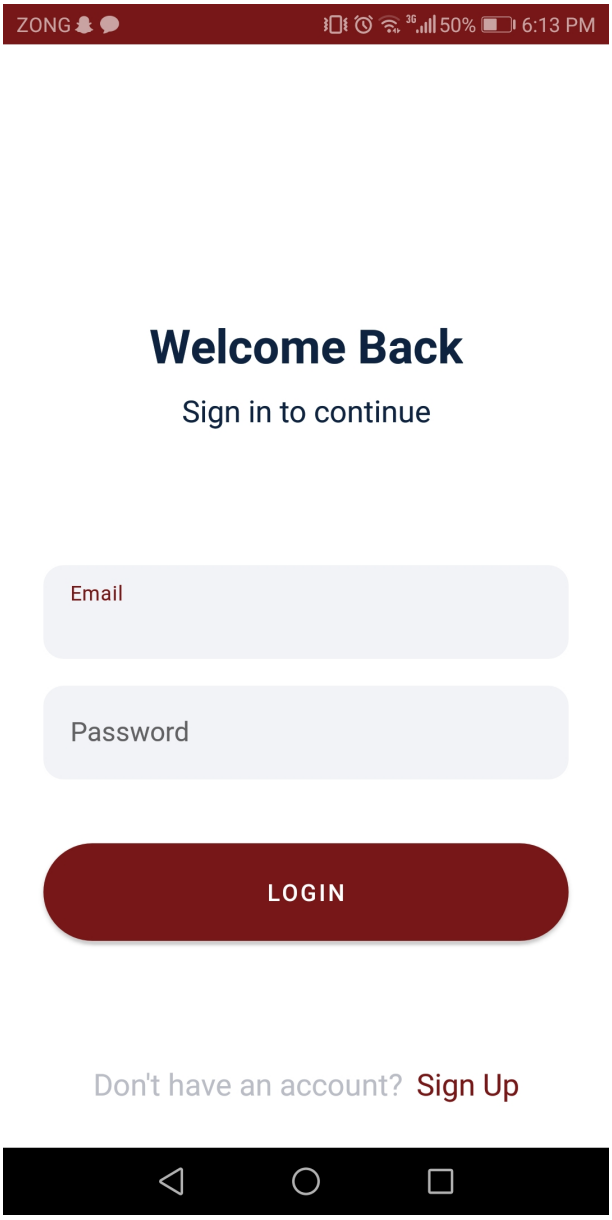

Figure 4.5: Login Page

Once the registration process is complete, users will be taken to this sign-in page where they can enter their username along with password. Upon clicking on the login button, the user can enjoy the amazing features of our application.

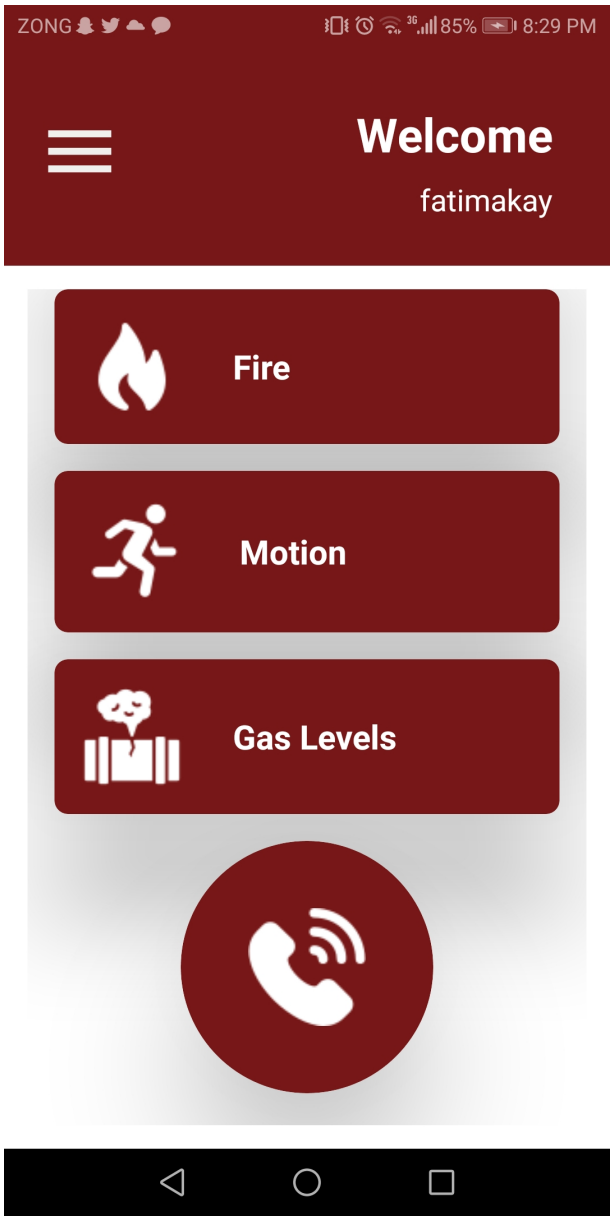

Figure 4.6: Home Page

This is a simple home page of the mobile app where you can see different emergencies. Red was chosen as the primary color because it depicts danger and emergency. You can open these emergencies to see the recent data received from the server. For example, the “Fire” card will hold the data obtained from the Infrared IR Flame sensor. Lastly, the panic button at the bottom can be seen which takes us to the dialer to immediately contact the Rescue 1122 service.

Figure 4.7: Profile Page

This page shows the details of profile information of the user who is currently logged in. It follows a simple format where full name of the user comes at the top in bold followed by username and email of the user.

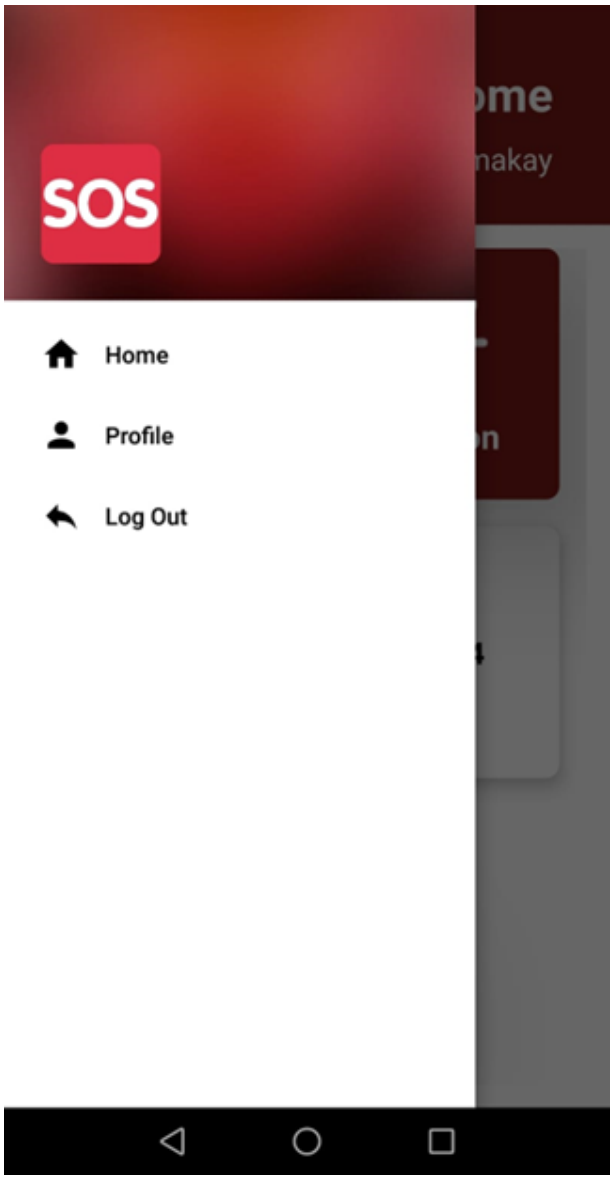

Figure 4.8: Navigation Drawer

This is the navigation drawer menu where you get different options such as home, profile, and logout. The logout option will take you back to sign in page and the user's session gets expired.

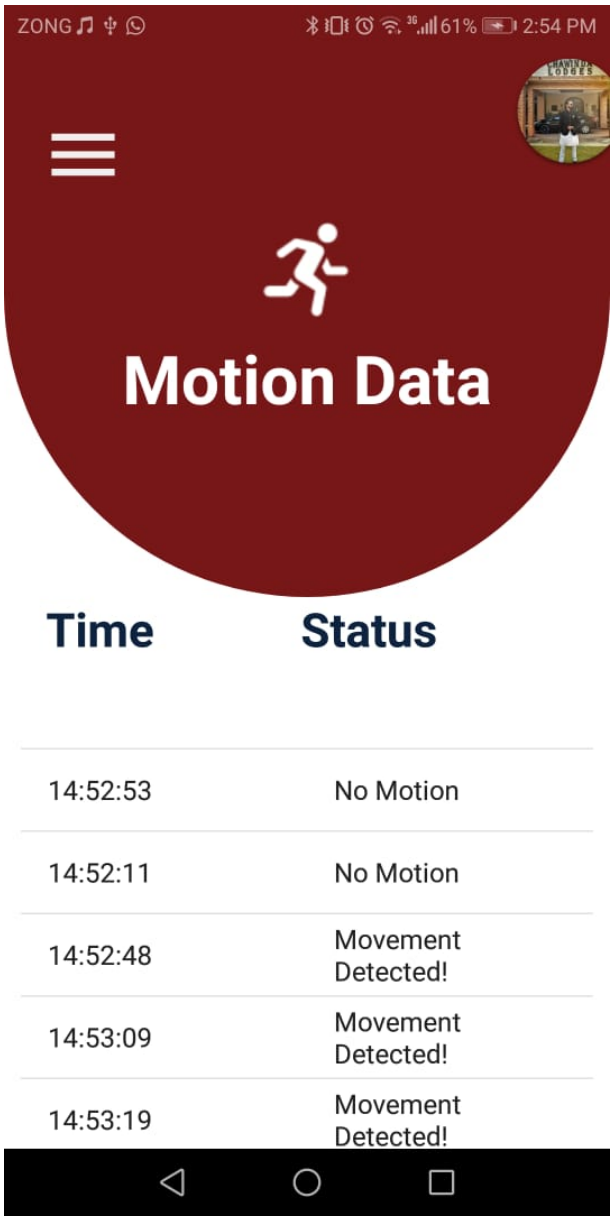

Figure 4.9: Motion Data

This is how the user will view the incoming data from the Firebase Realtime Database. The data is ordered by the most recent data changes.

4.7 Chapter Summary

In this chapter, the components, modules, GUI, and the data of the system are broken down and discussed. We also outlined how the components are linked together to work as a system. Next, we described the hardware, software, and databases used.

Chapter 5

System Implementation

The following chapter discuss the stages involved in designing and implementing the system. Similarly, all the details of tools and techniques will be discussed along with the procedures used in order to implement the requirements of the system that were specified in earlier phases.

5.1 System Overview

Firstly, the hardware components were to be identified such as the microcontroller, sensors, etc. that we were going to use in the system. The configuration logic was designed and implemented as shown in Figure 5.1. The sensors shown are motion, MQ-2 gas, and IR flame sensor. Next, the sensors were programmed through the Arduino software to read the values.

Next, we created a JSON object to send the readings to the ESP8266 using the serial communication. To do this, the data received by the ATMega328 was serialized, and printed to the serial monitor. Then, ESP8266 was programmed to read the data from the serial monitor and deserialize it. Finally, the readings were stored in strings and pushed onto Firebase Realtime Database.

The next phase involves the development of the Android app that will be used to read and monitor data on the user's end. To make the mobile application, we used the most commonly used software for this purpose, i.e. Android Studio. The app has a login functionality, after which the user is shown three options and a panic dial button. The options are for monitoring the three sensors data, and the panic button is for making a direct call to the rescue helpline for any situation regarding personal safety. Making the app as quick and simple as possible was the main priority during development.

Lastly, we worked on retrieving the data from Firebase Realtime Database into the application so that if the app receives data indicating a hazard, it can immediately generate a notification to

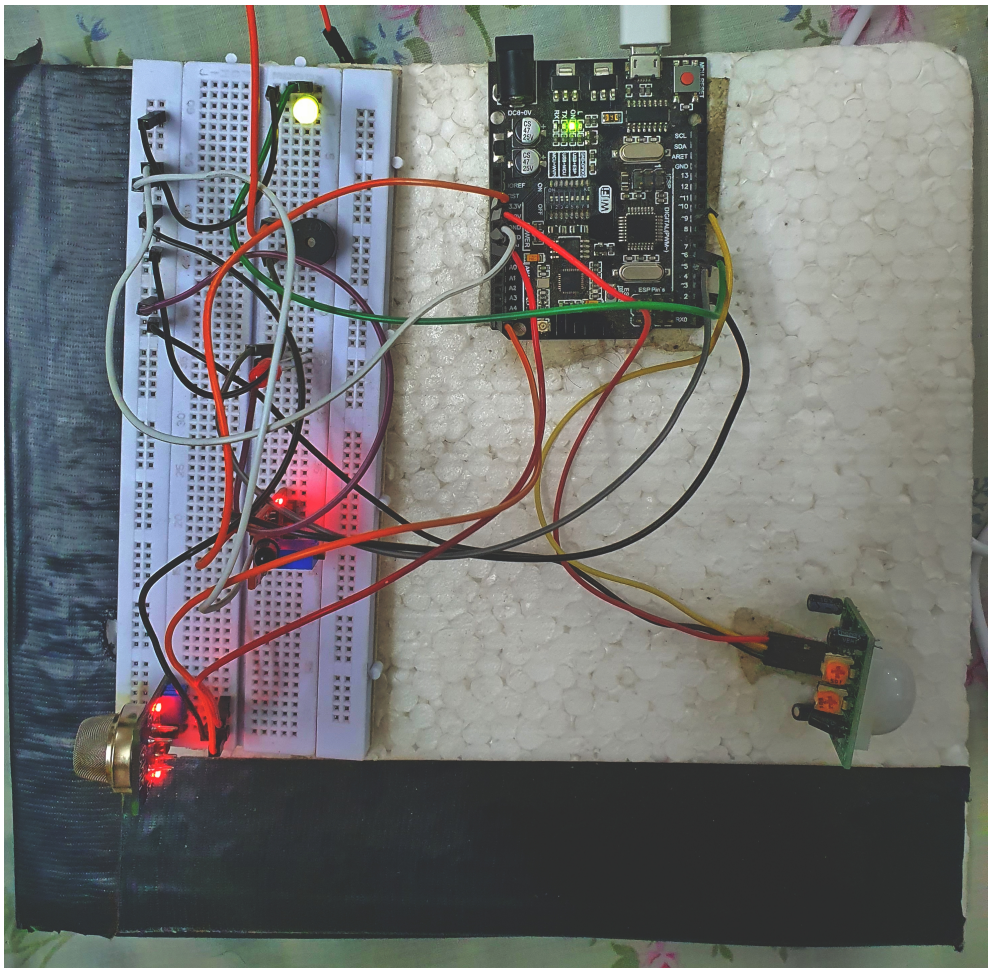

Figure 5.1: Photo of Device Prototype

the user's device and allow them to take action accordingly. Since the data is stored on a server, this process can occur regardless of the user's whereabouts, as long as they are connected to the Internet.

Libraries/Dependencies Used

- FirebaseArduino library

  FirebaseArduino is a library for the ESP8266 board. It makes it easier to make API calls and connect your arduino client to Firebase through C++. It works with the help of the ArduinoJSON library which is described below.

- ArduinoJSON library

  ArduinoJSON is a library compatible with all Arduino boards. It is used for storing and transmitting data in JSON format. It allows serialization, de-serialization, filtering, and much more.

- ESP8266Wifi library This is used for connecting your board to Wi-Fi or mobile hotspot using only a few lines of code.

The following dependencies and plugins were added to the Android Studio project for data retrieval from Firebase:

```
implementation 'com.google.firebase:firebase-database:20.0.0'
implementation platform('com.google.firebase:firebase-bom:30.1.0')
implementation 'com.google.firebase:firebase-analytics'
classpath 'com.google.gms:google-services:4.3.10'
```

5.2 Activity Diagram

Figure 5.2 shows the the basic flow of the application, which is initiated by registering an account for a new user. Nevertheless, if the account has already been made then you can navigate to the login page directly. The user now enters his/her credentials which are verified before granting access to the application's homepage. If the verification is positive, then the user will successfully log into the application.

The homepage will have different options which the user can select. If the user intends to access the sensor's data, then he will open the menu card of the respective sensor where the timestamp and current status (received by the firebase server) will be updated and displayed in real-time. In doing so, if the sensor sends any data to the application which needs to be taken care of then the notification alert will be generated and sent to a mobile device to let the user know about an emergency. Now, the user can either tap the notification banner to alert the authorities or swipe it away to ignore it. Therefore, if the user taps the notification, it will direct him/her to the dialer app with the contact number of a certain authority to be alerted.

However, to enlighten the alternate flow, the user has an option to press the panic button visible at the bottom of the homepage which will take the user to the dialer with the contact number of Rescue 1122 to deal with the emergency. Lastly, the user can log out of the application for the process to come to an end at that point.

5.3 Flow Diagram

The diagram in Figure 5.3 shows a detailed version of the activity diagram described above. It portrays the user side of activities as well as the operations of data retrieval at the back-end. Hence, we can clearly visualize the working of our entire project using this diagram.

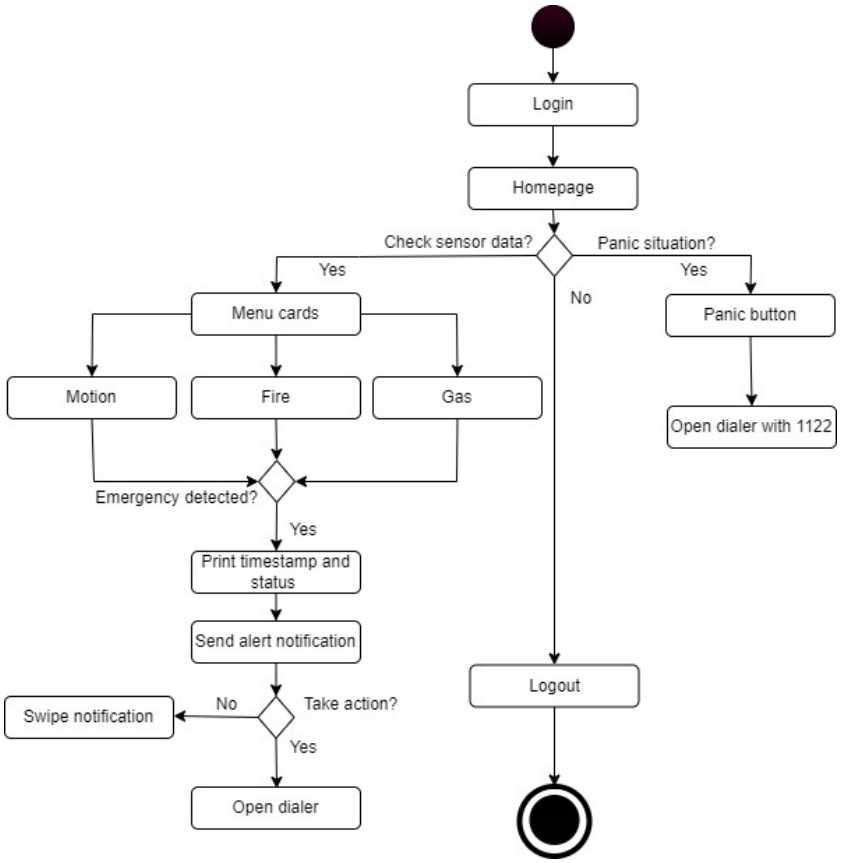

Figure 5.2: Activity Diagram

5.4 Chapter Summary

Chapter 5 was dedicated to the development phase of our project. We elaborated each step involved in creating the project, including both the hardware and the software side of things. A photo of the device prototype is also shown. In addition, the libraries used to program the sensors were discussed with their functions. In the application side, we have mentioned the dependencies and plugins that were required to use the Firebase API. Lastly, activity and flow diagrams are used to depict the general and specific flow of events of the system.

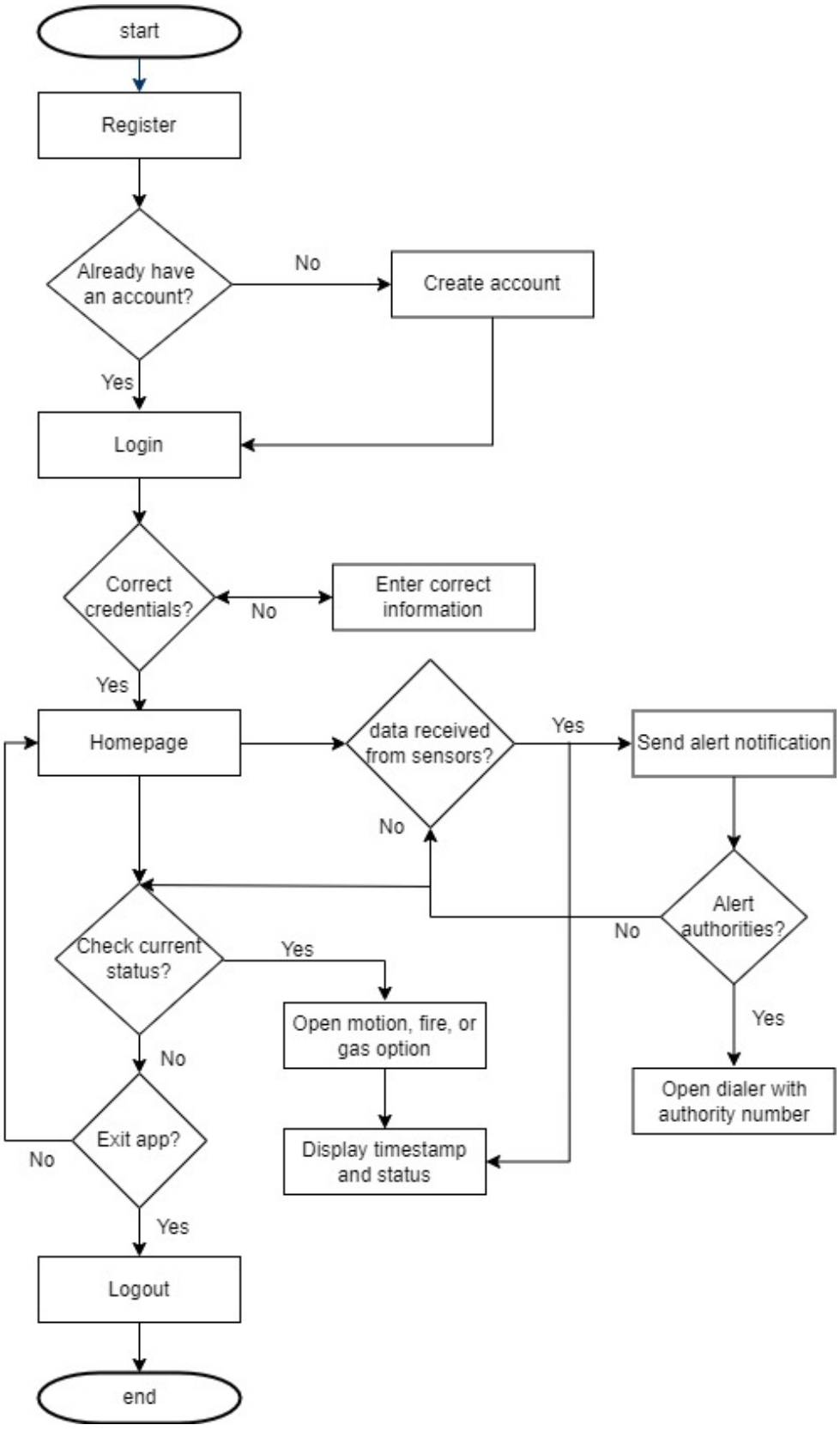

Figure 5.3: Flow Chart of System Events

Chapter 6

System Testing and Evaluation

In this chapter, the testing and validation of a product will be discussed. System testing consists of a certain level of tests to make sure that the system designed is working as it was intended to be. This chapter will moreover highlight how we did a test to identify errors and make the system fault tolerable.

6.1 Verification

The verification of the product ensures that the final product designed meets the requirements/specifications of the product.

6.2 Validation

Validation is a process of testing where we identify whether the system has managed to fulfill the requirements of the customer.

6.3 Reasons for testing

We performed the testing to make sure that the real-time data is being sent to the mobile application which is being uploaded by the sensors on the cloud to avoid any disaster. Moreover, it has to be made sure that the application is very user-friendly so that users do not find any difficulty in using the application since this app is used by several people from different age groups.

6.4 System testing

Graphical User Interface Testing

Graphical User Interface Testing was done in order to make sure that the interaction of a user with the mobile application is as fine as possible. Since it is an important aspect of any mobile application, so we tested it thoroughly via clicking and testing every button present on the screen. Similarly, cards on the main menu were tested to make sure that every card leads to the desired screen.

Usability Testing

In usability testing, we ensured that the item is of phenomenal quality and very easy to understand. We ensured the client can use the application with least to no technical information. The designed application is very easy to use since it involves a very basic and simple user interface so that the user does not get lost in the internal world of the application. Similarly, users can simply click the panic button to call emergency responders with just a single click.

Software Performance Testing

Software performance testing was done to ensure that the response time of the application is as minimal as possible. It was identified that the notification is sent as soon as the sensor sends the data to the cloud. It was fetched in real-time and the user was notified very quickly. Since it deals with emergency cases, so the delays in access time can be very harmful.

Compatibility Testing

Compatibility testing was performed in order to make sure that the mobile application is designed is not only restricted to some android devices but rather works perfectly fine on every android device.

6.5 Test Cases

Test Case Field	Description
Test Case Name	LOGIN
Test-Case ID	LG01
Test-Case ID	Registered Users
Test-case Description	The user will log in to an account using his/her credentials.
Preconditions	The user should be registered already.
Status	Pass

Table 6.1: Test Case of Login

Test Case Field	Description
Test Case Name	REGISTRATION
Test-Case ID	RG02
Test-Case ID	New Users
Test-case Description	The user will create an account by entering the required details.
Preconditions	The same email should not be registered on this application.
Status	Pass

Table 6.2: Test Case of Registration

Test Case Field	Description
Test Case Name	HOMEPAGE
Test-Case ID	HP03
Test-Case ID	Registered Users
Test-case Description	Ensure that the application is displaying data retrieved from firebase.
Preconditions	The user should have the app logged in and connected to WiFi.
Status	Pass

Table 6.3: Test Case of Home page

Test Case Field	Description
Test Case Name	NOTIFICATION SERVICE
Test-Case ID	AN04
Test-Case ID	Registered Users
Test-case Description	Check if notification is sent when emergency value is received.
Preconditions	User should have the app logged in and connected to WiFi.
Status	Pass

Table 6.4: Test Case of Notification Service

6.6 Chapter Summary

This chapter discussed the concept of verification and validation along with reasons for testing a system. It further discusses the different tests performed such as GUI testing, Usability testing, Software Performance testing, Compatibility testing, etc in order to make sure that the system is as fault-free as possible. Similarly, test cases are mentioned in this chapter which were carried out in order to analyze the performance of our project.

Chapter 7

Conclusions

The IoT paradigm is becoming increasingly popular in all areas of the community, including the emergency sector. The SOS application initiative brings the concept of remote monitoring of the home and automated calls in case of serious situations, aiming to improve emergency response time and reduce fatalities. This paper proposes and implements such a system. Comparing the system to previous related works proves there are many advantages to the approaches taken in this project. The successful communication of data between two microcontrollers, as well as between the microcontroller and a real time database, shows that it is possible to develop a cheap automation system with simplicity. The user can then use their mobile application for monitoring purposes.

7.1 Future Work

This report is an important part in developing a complete Arduino-based emergency system that will make the home more secure and keep the residents safe. For future work, more modules can be added to the project to detect more types of emergencies. Aside from that, the system can also be used for non-emergency situations, such as automation of lights, fans, and other appliances. However, to do this, it must be ensured that system scalability is added so that no complications should occur while adding more modules.

Appendix A

User Manual

This is a user guide for you to use Smart Home Application. This manual is intended to ensure that the user's experience with the application is as fruitful as possible.

1. First of all, to use the application, you need to install and make sure that the installation process is completed successfully.

2. Once the installation is complete, you will be shown a screen in order to get you registered. You will be required to enter your credentials in the fields displayed such as full name, username, email, and password. Then press the **Sign Up** button.

3. This will get you registered on our application successfully and you will be directed to the login page. You will have to enter your username and password correctly and log in to your account. Then press the log-in button.

   However, if you see an **"Invalid Credentials"** message then this means that you have entered your credentials incorrectly and you might have to recheck and enter them again.

4. This will take you to the home page of the application which would look like this:

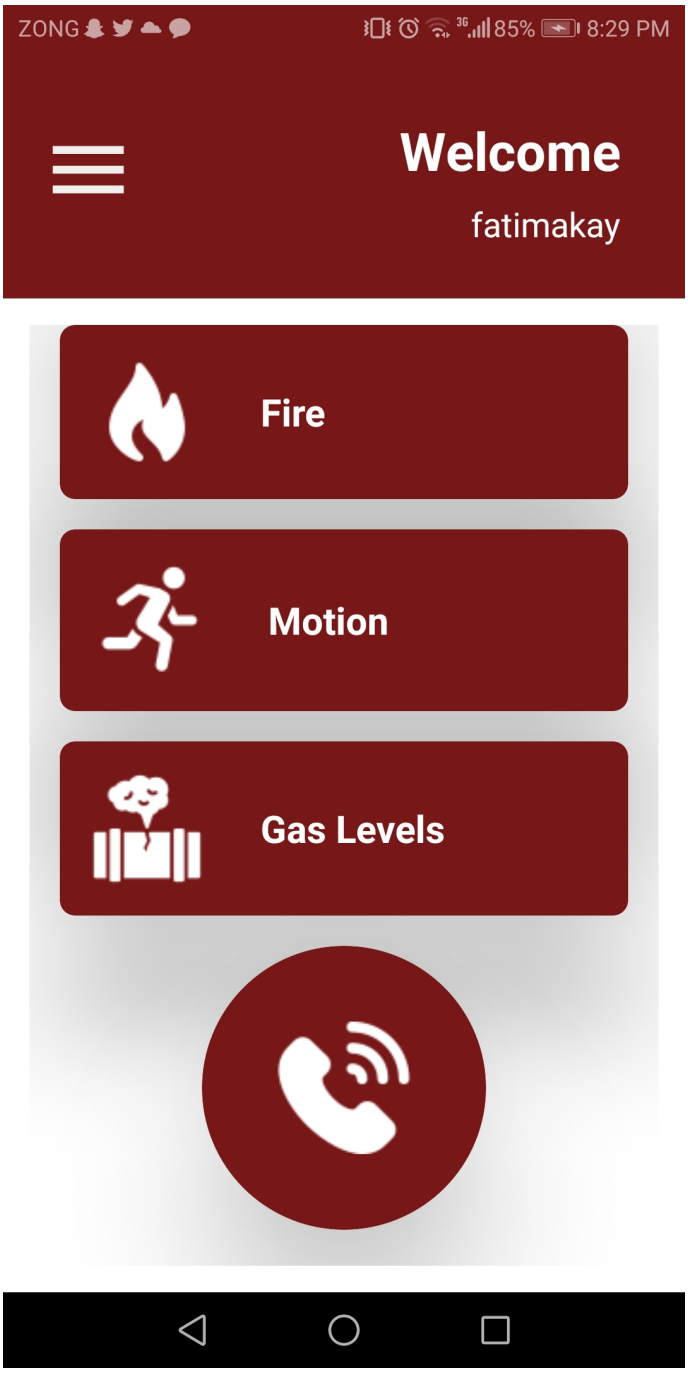

Figure A.1: Home Screen

You can see the different sensors in the menu and a call icon at the bottom of the screen. If you want to see the data which is being sent by the sensor, then you tap on the respective sensor displayed on the menu. It will have a timestamp which will indicate at what time the data is received. Secondly, it will display the current status to show the state of the sensor. Let's suppose you click on **"Fire"**, you will be able to see a screen like this:

Here you can see the first timestamp of "21.14.23" which means that the sensor has sent data

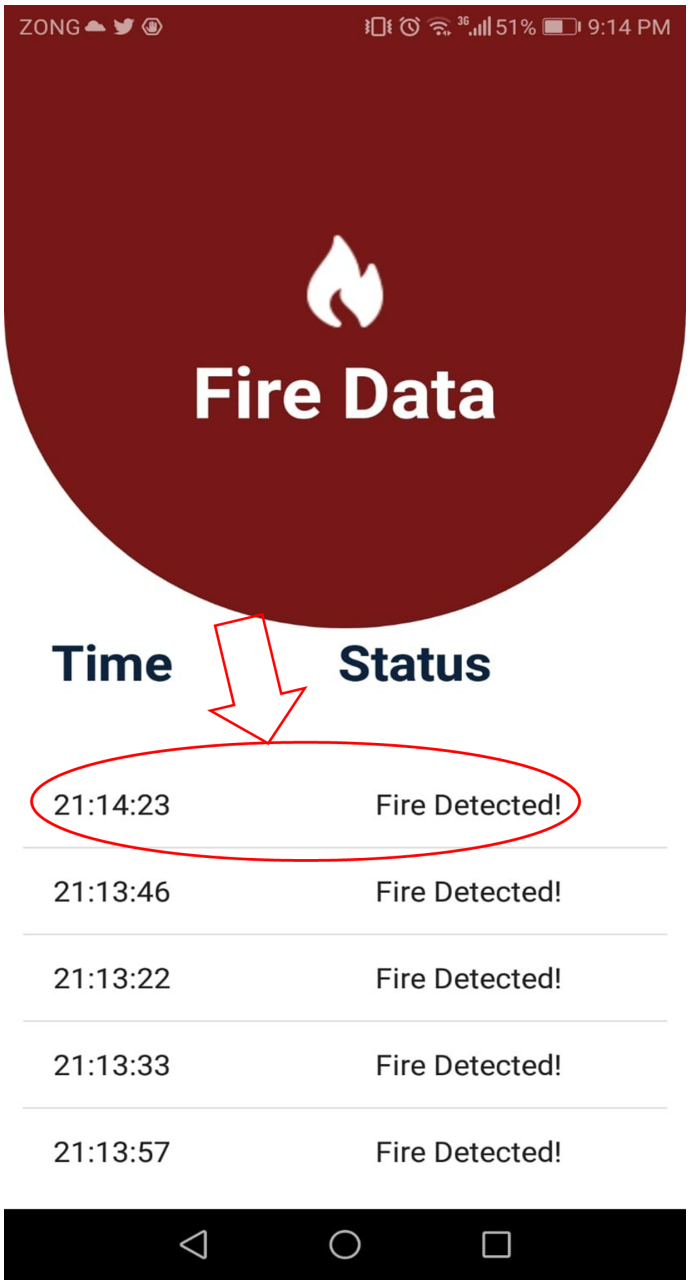

Figure A.2: Fire Information

at 9.14 PM and the status shows “Fire Detected” which means that the system detected a fire nearby.

5. If you see a notification like this, then it means that there is an emergency that needs to be handled as soon as possible.

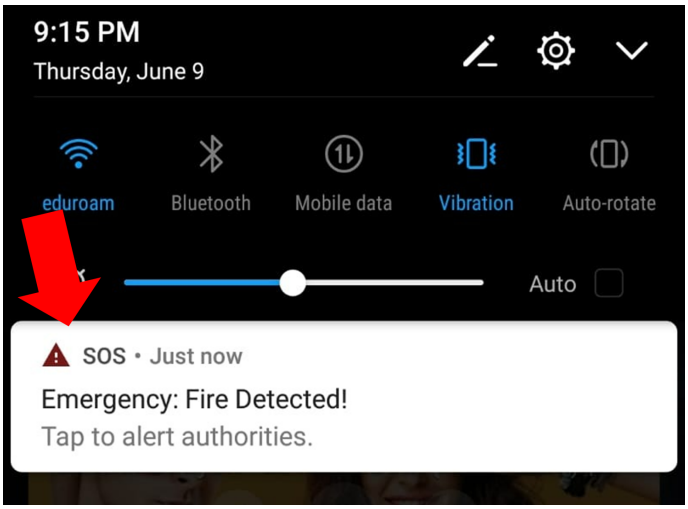

Figure A.3: Notification Displayed on Detection of Emergency

You can tap on this notification banner to alert the authorities.

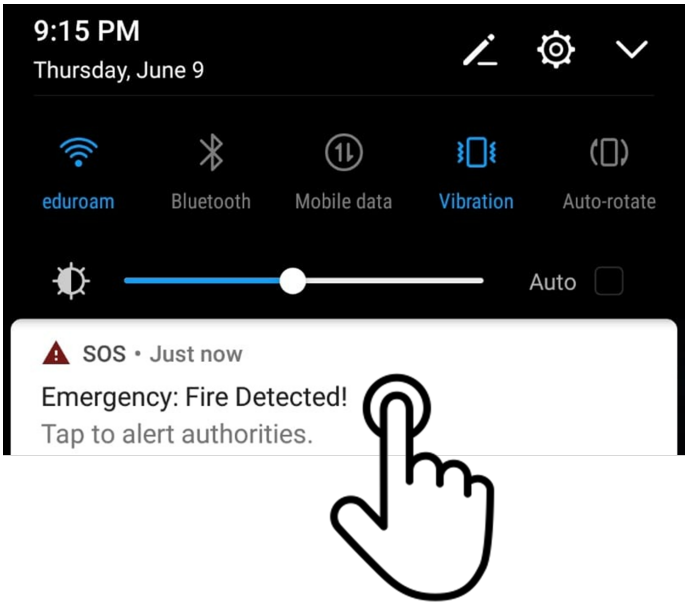

Figure A.4: Tap Notification

This will take you to the dialer screen with the contact number of the fire brigade to alert them.

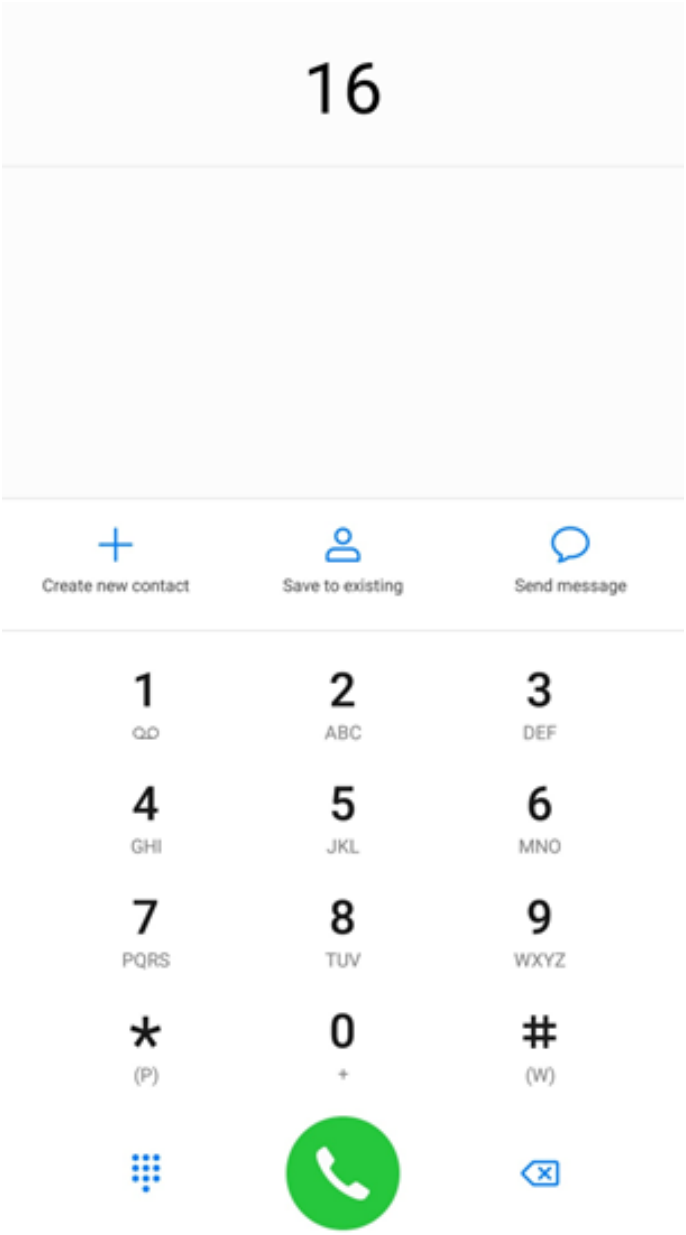

Figure A.5: Dialer for Fire Brigade

6. If you are in a crisis, then all you need to do is tap on the panic button displayed at the bottom of the home page Rescue 1122 number where you can contact them and ask for help.

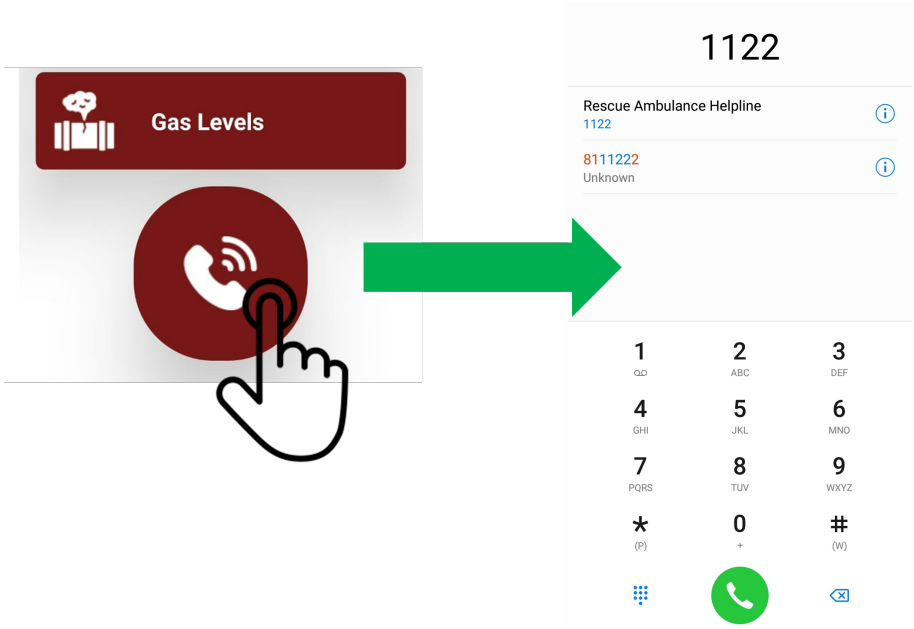

Figure A.6: Dialer for Rescue Helpline

7. If you want to check your account details, then slide the navigation bar right left to right to see the "profile" option there, and then click on profile. This will display the details of your account.

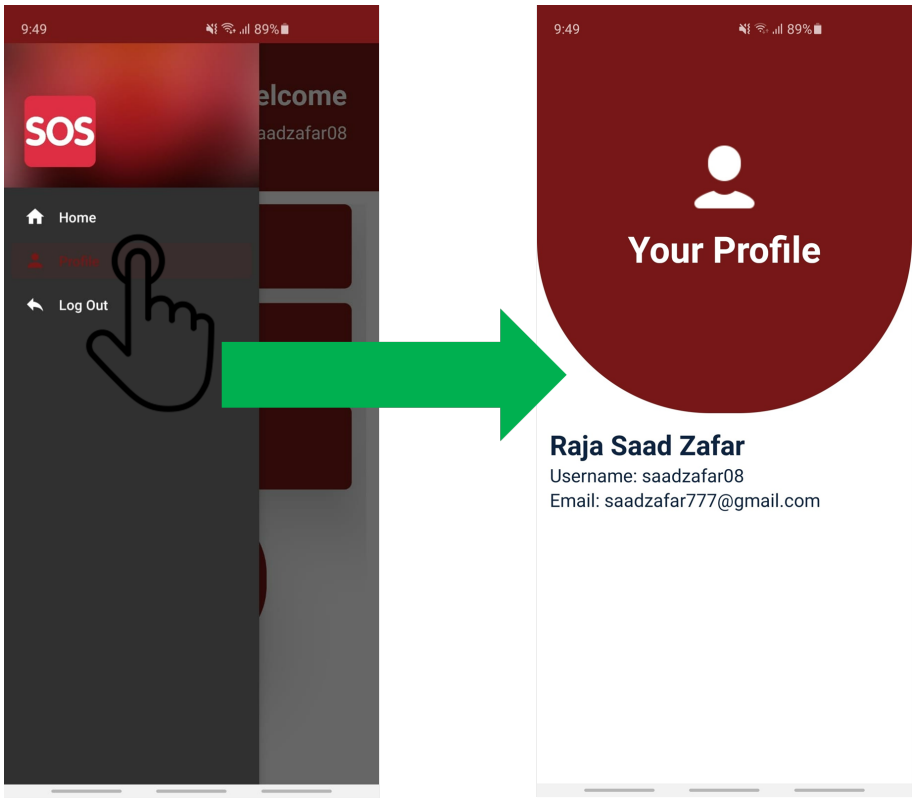

Figure A.7: Account Details

References

[1] Chuang Zhou Ji Huang Zhuo Li Shin-Ming Cheng Jun Cheng Qin Wu, Jiashuo Cao and Guanghui Pan. Intelligent smoke alarm system with wireless sensor network using zigbee. *Wireless Communications and Mobile Computing vol. 2018*, 2018. Cited on pp. 5 and 6.

[2] Maganti Syamala Nalini N.J Lakshmana Phaneendra, Tumma Srinivasarao. Efficient smart emergency response system for fire hazards using iot. *International Journal of Advanced Computer Science and Applications*, 2018. Cited on pp. 5 and 6.

[3] Davin Bagas Adriano Taryudi and Wahyu Apsari Ciptoning Budi. Iot-based integrated home security and monitoring system. *Journal of Physics: Conference Series*, 2018. Cited on pp. 5 and 6.

[4] Honoriu Valean Daniel Chioran. Arduino based smart home automation system. *International Journal of Advanced Computer Science and Applications*, 2020. Cited on pp. 5, 6, and 7.

[5] Prof. Mr. Yoginath R. Kalshetty Ms. Poonam V. Gaikwad. Bluetooth based smart automation system. *International Journal of New Innovations in Engineering and Technology*, 2017. Cited on pp. 6 and 7.