



National University of Computer and Emerging Sciences, Lahore



Muhaafiz

Aiza Khurram 21L-5771 BS(SE)

Minaal Khurram 21L-5828 BS(SE)

Umamah Hussain 21L-1858 BS(CS)

Supervisor: Mr. Fraz Yousaf

Final Year Project

October 10, 2024

Anti-Plagiarism Declaration

This is to declare that the above publication was produced under the:

Title: Muhaafiz

is the sole contribution of the author(s), and no part hereof has been reproduced as it is the basis (cut and paste) that can be considered Plagiarism. All referenced parts have been used to argue the idea and cited properly. I/We will be responsible and liable for any consequence if a violation of this declaration is determined.

Date:

Name:Aiza Khurram.....

Signature:

Name:Minaal Khurram.....

Signature:

Name:Umamah Hussain.....

Signature:

Author's Declaration

This states Authors' declaration that the work presented in the report is their own, and has not been submitted/presented previously to any other institution or organization.

Abstract

Recognizing the urgent need for an all-inclusive emergency application, we developed "Muhaafiz." Our project offers a comprehensive system that integrates diverse features. These features range from emergency reporting mechanisms to acquiring real-time information on healthcare facilities. Muhaafiz aims to enhance public safety and reduce the disconnect between users and emergency services. Furthermore, the application addresses public unawareness of first aid and healthcare information. This is done by ensuring that users can access vital information effectively. To sum up, Muhaafiz is designed to improve emergency reporting and response mechanisms in the country.

Executive Summary

Our project Muhaafiz aims to address the need for a comprehensive and all-inclusive emergency application. Currently, there is an absence of applications aimed at effective emergency reporting and safety systems. Furthermore, the general public faces unawareness regarding first aid information and a disconnect in healthcare related information. All these factors combined results in insufficient emergency reporting and processing.

To address these factors, Muhaafiz provides an all inclusive emergency application. This application covers all avenues by providing an efficient and cohesive system to manage all operations. To begin with, the application provides various mechanisms for reporting emergencies and tracking location of emergency responders so that the user can stay informed and prepared. In addition, the project integrates real-time information from hospitals, to reduce the information disconnect between users and healthcare. This is done by syncing real time data given by hospitals (such as equipment availability, specialized doctors etc) with the application. By doing so, our application will help users under distress make critical decisions more easily. Furthermore, the application also provides an AI powered chatbot trained to provide first aid guidance, provide relief in minor emergencies and while the user waits for emergency responders. Lastly, various other features such as location tracking, safety timers, medical history management and more are included in the application.

Initially, the project involves a mobile application for the general users, built using android studio with kotlin and jetpack compose. Following that, the project also features a web based dashboard for admin and hospital staff management activities. The website is built using part of the MERN stack. Furthermore, the system uses a central real time database to integrate both platforms seamlessly. This ensures real-time data synchronization and secure storage of user and hospital information. Moreover, using a client-server architecture with a singular central database, will not only help in data-synchronization but also provide room for future scalability and improvements. Lastly, The application's frontend is built with user friendliness and ease in mind ensuring an efficient and satisfactory user experience.

To conclude, the system provides critical software for managing emergency situations, with the potential for future scalability and improvements. The final report elaborates on the system's architecture, design considerations, implementation strategies, and testing outcomes, offering insights into each phase of the project's development.

Table of Contents

List of Figures	x
List of Tables	xi
1 Introduction	1
1.1 Purpose of this Document	1
1.2 Intended Audience	1
1.3 Definitions, Acronyms, and Abbreviations	1
1.4 Conclusion	2
2 Project Vision	3
2.1 Problem Domain Overview	3
2.1.1 Emergency Services	3
2.1.2 Hospital Integration	3
2.2 Problem Statement	3
2.3 Problem Elaboration	3
2.3.1 Lack of Safety apps	4
2.3.2 Lack of Real time hospital data	4
2.3.3 Lack of available first aid knowledge	4
2.3.4 Time consuming hospital paperwork	4
2.4 Goals and Objectives	4
2.4.1 Goals	4
2.4.2 Objectives	5
2.5 Project Scope	5
2.6 Sustainable Development Goal (SDG)	6
2.6.1 SDG-3 (Good health and well being)	6
2.6.2 SDG-5 (Gender Equality)	6
2.6.3 SDG-10 (Reduced Inequality)	7
2.6.4 SDG-11 (Sustainable Cities and Communities)	7

2.6.5	SDG-16 (Peace and Justice Strong Institutions)	7
2.7	Constraints	8
2.8	Business Opportunity	8
2.9	Stakeholders Description/ User Characteristics	8
2.9.1	Stakeholders Summary	8
2.9.2	Key High-Level Goals and Problems of Stakeholders	9
3	Related Work	10
3.1	Definitions, Acronyms, and Abbreviations	10
3.2	Detailed Literature Review	10
3.2.1	Rescue 1122	10
3.2.2	Women Safety App by Punjab Police	11
3.2.3	Edhi Foundation	12
3.2.4	Punjab Police App	13
3.2.5	Pak Life Saver	15
3.2.6	PAK NDMA-Disaster Alert Application	16
3.2.7	The Punjab Safe Cities Authority (PSCA)	17
3.3	Literature Review Summary Table	18
3.4	Conclusion	18
4	Software Requirement Specifications	19
4.1	List of Features	19
4.2	Functional Requirements	19
4.3	Quality Attributes	20
4.3.1	Usability	20
4.3.2	Maintainability	20
4.3.3	Reliability	20
4.3.4	Availability	21
4.3.5	Testability	21
4.3.6	Integrity	21
4.3.7	Reusability	21
4.3.8	Robustness	21
4.3.9	Correctness	21
4.4	Non-Functional Requirements	21
4.5	Assumptions	22

4.6	Use Cases	23
4.6.1	Use Cases for Mobile Application	23
4.6.2	Use Cases for Admin Dashboard	35
4.6.3	Use Cases for Hospital Dashboard	38
4.7	Hardware and Software Requirements	47
4.7.1	Hardware Requirements	47
4.7.2	Software Requirements	47
4.8	Graphical User Interface	47
4.8.1	Welcome Pages	47
4.8.2	Login Page	48
4.8.3	Sign up	48
4.8.4	Home Page	49
4.8.5	Emergency Page	50
4.8.6	Medical Situation	50
4.8.7	Consult a Doctor	51
4.8.8	Call an Ambulance	52
4.8.9	Nearby Hospitals	52
4.8.10	Hospitals Page	53
4.8.11	AI Assistant	54
4.8.12	Profile Page	54
4.8.13	Personal Information	55
4.8.14	Emergency Contacts	56
4.8.15	Safety Timer	57
4.8.16	Admin Dashboard - Homepage	58
4.8.17	Admin Dashboard - User's Page	58
4.8.18	Admin Dashboard - Hospital's page	58
4.8.19	Admin Profile Page	59
4.8.20	Hospital Dashboard – Homepage	59
4.8.21	Hospital Dashboard – Department's Page	60
4.8.22	Hospital Dashboard – Doctor's Page	60
4.9	Database Design	61
4.9.1	ER Diagram	61
4.9.2	Data Dictionary	63
4.10	Risk Analysis	63

4.10.1	Technical Risks	63
4.10.2	Business Risks	64
4.10.3	Operational Risks	65
4.10.4	Development Risks	65
4.10.5	User Risks	65
5	High-Level and Low-Level Design	66
5.1	System Overview	66
5.1.1	Client Side	66
5.1.2	Server Side	66
5.2	Design Considerations	66
5.2.1	Assumptions and Dependencies	67
5.2.2	General Constraints	67
5.2.3	Goals and Guidelines	68
5.2.4	Development Methods	69
5.3	System Architecture	69
5.3.1	Subsystem Architecture	70
5.4	Architectural Strategies	73
5.4.1	Programming Language and Frameworks	73
5.4.2	Database Storage	73
5.4.3	User Interface Paradigms	73
5.5	Domain Model/Class Diagram	74
5.6	Policies and Tactics	74
5.6.1	Product Choices	74
5.6.2	Engineering Tradeoffs	75
5.6.3	Design Patterns	75

List of Figures

2.1	SDG-3: Good Health and well being	6
2.2	SDG-5: Gender Equality	6
2.3	SDG-10: Reduced Inequality	7
2.4	SDG-11: Sustainable Cities and Communities	7
2.5	SDG-16: Peace and Justice Strong Institutions	8
4.1	Welcome Pages	47
4.2	Login Page	48
4.3	Sign Up page	49
4.4	Home Page	49
4.5	Emergency Page	50
4.6	Medical Emergency page	51
4.7	Consultation Page	51
4.8	Ambulance Tracking Page	52
4.9	Nearby hospitals	53
4.10	Hospitals Page	53
4.11	AI Assistant	54
4.12	Profile Page	55
4.13	Personal Information	56
4.14	Emergency Contacts	57
4.15	Safety Timer	57
4.16	Admin Dashboard - Homepage	58
4.17	Admin Dashboard - User's Page	58
4.18	Admin Dashboard - Hospital's page	59
4.19	Admin Profile Page	59
4.20	Hospital Dashboard – Homepage	60
4.21	Hospital Dashboard – Department's Page	60
4.22	Hospital Dashboard – Doctor's Page	61
4.23	ER Diagram for Admin Dashboard	61

4.24 ER Diagram for Hospital's Admin Dashboard	62
4.25 ER Diagram for Mobile Application	62
5.1 System Architecture design	70
5.2 Subsystem component design for users	71
5.3 Subsystem component design for Admin	71
5.4 Subsystem component design for Hospital Admin	72
5.5 Class Diagram for our application	74

List of Tables

3.1	Related Work Review Summary Table	18
4.1	Use Case for Login	23
4.2	Use Case for Sign Up	24
4.3	Use Case for Incident Reporting	25
4.4	Use Case to set Safety Timer	26
4.5	Use Case for AI Assistant- First Aid Guide	27
4.6	Use Case for calling an ambulance	28
4.7	Use Case to find the nearest hospital	29
4.8	Use Case to view hospital information	30
4.9	Use Case for consulting a doctor	31
4.10	Use Case to add emergency contacts	32
4.11	Use Case to view emergency contacts	33
4.12	Use Case to manage user profile	34
4.13	Use Case to add hospitals	35
4.14	Use Case to remove hospitals	36
4.15	Use Case to remove users	37
4.16	Use Case to add beds	38
4.17	Use Case to remove beds	39
4.18	Use Case to view beds	40
4.19	Use Case to view doctors	41
4.20	Use Case to add doctors	42
4.21	Use Case to remove doctors	43
4.22	Use Case to view departments	44
4.23	Use Case to add departments	45
4.24	Use Case to remove departments	46
4.25	Data Dictionary Table	63

Chapter 1 Introduction

In Pakistan , there is critical insufficiency of cohesive platforms related to reporting emergency. The general public faces a lack of effective emergency reporting , first aid knowledge and gap in healthcare information. Our project ”Muhaafiz” aims to address such issues by making reporting emergencies more efficient, providing real time hospital information along with an integrated chat bot for minor emergencies and first aid queries. Through the project, we aim to provide a cohesive all in one application to the general public of Pakistan, making first aid, and healthcare more widely accessible.

1.1 Purpose of this Document

The purpose of this document is to provide a detailed overview of the technical and general aspects of your project, Muhaafiz. This document will go over all aspects of the project including goals, objectives, design, challenges, implementation and more. This project aims to provide a cohesive emergency reporting and safety application, complete with an AI powered chat bot and real-time hospital data.

1.2 Intended Audience

This document is intended for academic audiences which is mainly composed of professors and educationalists looking to understand and evaluate this project. In addition to this, any concerned person who wants to comprehend the system’s operations can review certain parts of this report.

1.3 Definitions, Acronyms, and Abbreviations

Abbreviations used in this document are as follows:

SDG: Sustainable Development Goal

FYP: Final Year Project

UI: User Interface

UX: User Experience

Agile: Agile Development

SCRUM: Scrum Development

DB: Data Base

MERN: Mongo, Express, React, Nodejs

API: Application Programming Interface

TDD: Test Driven Development

CI: Continuous Integration

IDE: Integrated Development Environment

CSS: Cascading Styles Sheet

JS: Javascript

GUI: Graphical User Interface

GPS: Global Positioning System

OS: Operating System

AI: Artificial Intelligence

GIT: Global Information Tracker (commonly referred to as a distributed version control system)

XML: Extensible Markup Language

MVC: Model View Controller

1.4 Conclusion

To conclude, Pakistan currently faces a lack of comprehensive emergency application widely available to the general public. This report provides an analysis of the technical details, the design approach and addresses challenges related to the project. Each chapter contributes to this by providing details of various aspect of the system and its development process

The design considerations were explored in detail, explaining the system's architecture and how different subsystems collaborate to manage data flow and ensure real-time updates. The report then discussed the implementation, covering the technology stack and the methodologies used which played a significant role in ensuring quality and adaptability. Lastly, the report examined security measures, applying key information security principles to safeguard sensitive data and ensure the integrity, confidentiality, and availability of the system.

In summary, this report provides a clear road map from concept to implementation, ensuring that the emergency management system is both functional and secure, with room for future improvements.

Chapter 2 Project Vision

2.1 Problem Domain Overview

Our system “Muhaafiz” aims to be a comprehensive emergency management application, complete with hospital integration. The application targets the problem of delay in emergency responses, lack of safety systems in the country and information gap between patients and hospitals.

2.1.1 Emergency Services

The application aims to streamline emergency response efforts in Pakistan by providing a comprehensive and centralized system for reporting emergencies, finding assistance and resource coordination. The application system connects users to emergency services such as rescue teams, hospitals and fire departments. Furthermore, the application includes features such as AI trained chatbot and live location tracking to further help in general emergencies and finding safety.

2.1.2 Hospital Integration

The application also aims to make getting healthcare effortless and accessible by giving user access to real time information on hospital beds, doctors, and medical equipment in the city. By doing so, the application aims to help facilitate quicker and more informed decisions during emergencies. The system’s integration with hospital databases and use of location-based services ensures that the nearest and most suitable help is dispatched promptly, significantly improving response times and resource allocation during crises.

2.2 Problem Statement

The application addresses the problem of the lack of comprehensive emergency response and safety systems in Pakistan. People facing critical situations face delays and struggle to access emergency services in time. This causes inefficiencies in both receiving care and reporting emergencies and puts people at a great safety and health risk. The proposed system also addresses the gap between the general man and hospitals, the real-time data will ensure faster and better decision-making under critical situations.

2.3 Problem Elaboration

In Pakistan, numerous challenges are prevailing due to the lack of comprehensive safety applications and inefficient emergency handling strategy. Through our system, we aim to target all issues ranging

from emergency reporting, online first aid and access to real-time hospital information. The following subsections further elaborate.

2.3.1 Lack of Safety apps

Currently, Pakistan faces a lack of emergency apps that are comprehensive and effective. Most emergency handling efforts launched by the government are in the form of call helplines or applications handling only a subdomain. Our system aims to be a comprehensive application for emergency where all issues are solved within a single application

2.3.2 Lack of Real time hospital data

The healthcare industry in Pakistan faces a shortage of readily available online information regarding hospital facilities, such as the availability of beds, equipment and doctors. This lack of real-time data makes it difficult for individuals in emergencies to make informed decisions about where to pursue medical care. Having this information on hospitals' available capacity of ICU beds and other critical medical equipment would streamline emergency responses, allowing patients to be directed to the appropriate hospitals with the available resources.

2.3.3 Lack of available first aid knowledge

Majority of Pakistani individuals do not have basic first aid knowledge, this knowledge can be critical in managing emergencies before professionals arrive. The absence of easily accessible first aid resources can lead to poor outcomes in emergency situations.

2.3.4 Time consuming hospital paperwork

During emergencies, when time is of the essence, handling hospital paperwork can be very time consuming and inconvenient. These procedures can delay medical care and add stress to the patient and their families. Our system aims to address this by providing electronic documentation and simplifying the hospital administrative process.

2.4 Goals and Objectives

Following are some goals and objectives we aim to achieve through this project

2.4.1 Goals

- Improve access to emergency services

- Enhance Reporting strategies
- Promote public safety awareness

2.4.2 Objectives

- Develop a comprehensive and user friendly safety app
- Ensure the app is accessible in all areas of Pakistan
- Integrate hospital data systems to provide real-time updates
- Ensure a smooth and effective emergency reporting strategy

2.5 Project Scope

The scope of this project is to design and implement a user-friendly mobile application for handling a range of emergency related problems, ranging from emergency reporting, Artificial intelligence assistance and hospital integration.

The application will enable users to:

- Report emergencies
- Share real time location information with personal emergency contacts
- Access AI chatbot trained to handle minor emergencies and first aid guidance
- Obtain real-time information on real time hospitals, including availability of special equipment, doctors and beds
- Schedule appointments and manage hospital paperwork directly through the app
- Call ambulances, firefighters and other emergency responders
- View safety guidelines and disaster preparedness tips
- Handle Users medical history

The project's main components include:

Firstly, A mobile based application for normal users. The mobile based application will be developed on android studio , using jetpack compose and kotlin for frontend and backend respectively.

Secondly, A website for admin and hospitals The web based application mainly consists of a dashboard and control options for the admin and the hospitals, the website will be developed using React, Node js and Express

The system will use firebase as the database and backend storage, this will allow easy cross platform

integration and real-time data updates. Relevant APIs will be used to facilitate seamless communication between the app, database and emergency services

2.6 Sustainable Development Goal (SDG)

Our project aligns with 5 of the given Sustainable Development Goals. The following sections discuss this in detail :

2.6.1 SDG-3 (Good health and well being)

Our project aims to provide accessible emergency reporting and safety mechanisms to the public. Furthermore, users will be able to access real-time hospital information and first aid guidance. Thus, the accessibility and awareness related to healthcare and safety mechanisms will improve community well being and health outcomes.



Figure 2.1: SDG-3: Good Health and well being

2.6.2 SDG-5 (Gender Equality)

The gender equality goal is addressed by offering a safe and judgment free space for all vulnerable individuals, such as women, transgender and disabled people in distress to report emergencies.



Figure 2.2: SDG-5: Gender Equality

2.6.3 SDG-10 (Reduced Inequality)

Through the app users can access services such as healthcare information, first aid guidance, emergency reporting services and more without facing any bias. Therefore, The application reduces healthcare disparities targeting minorities by bringing healthcare information and serving under-served areas



Figure 2.3: SDG-10: Reduced Inequality

2.6.4 SDG-11 (Sustainable Cities and Communities)

)Our project will help enhance urban resilience through its emergency response system, which allows hospitals and admins to coordinate in real-time, ensuring that communities are safer and better prepared for emergencies. This contributes to making cities more inclusive and sustainable by improving access to essential services and reducing inequalities in emergency care.



Figure 2.4: SDG-11: Sustainable Cities and Communities

2.6.5 SDG-16 (Peace and Justice Strong Institutions)

The system promotes transparent and efficient public service delivery by providing a platform that supports accountable emergency management. Its ability to ensure safety and security during emergencies fosters more peaceful and resilient communities, aligning with the goal of strong, effective institutions.



Figure 2.5: SDG-16: Peace and Justice Strong Institutions

2.7 Constraints

Some constraints of this project include the challenge of obtaining real-time hospital data, as hospitals will have to incorporate our website to integrate real-time data. Technical difficulties, such as ensuring reliable communication between the mobile app and emergency services, especially in areas with limited internet access, may also arise. Regulatory compliance with healthcare data privacy laws and limited development resources may further restrict the project's scope

2.8 Business Opportunity

The business opportunities regarding this project lies in dominating a critical gap in healthcare accessibility and emergency services in Pakistan. This app can help revolutionize emergency handling strategies in Pakistan by providing a comprehensive emergency application complete with AI-base assistance. There is a growing demand for efficient, integrated solutions in healthcare, and the app can target underserved areas with limited access to emergency care. The project has the potential to collaborate with hospitals, emergency services, and public safety institutions, creating a sustainable business model.

2.9 Stakeholders Description/ User Characteristics

The key stakeholders involved in our system and their roles are as follows

2.9.1 Stakeholders Summary

- Users
 - The main users of this application is the general public, individuals who use the app to report emergencies, get real-time guidance and access hospital information

- Healthcare Provides
 - Institutions responsible for providing real-time data on doctors and equipment's
- Emergency Responders
 - Ambulance services, firefighters, and law enforcement who respond to emergency alerts generated by the app.
- Admin/Developers
 - Responsible for maintaining the system and managing the hospital dashboard and emergency integration's.

2.9.2 Key High-Level Goals and Problems of Stakeholders

- Users
 - Require a reliable and efficient way of reporting emergencies and accessing healthcare information. General public struggles with delays in emergency responses and lack of information
- Hospitals
 - Deliver a streamlined system for sharing real-time information and managing appointments efficiently
- Emergency Responders
 - location data and accurate alerts to reduce response time
- Admin/Developers
 - Focus on maintaining the system's reliability, handling data privacy concerns, and ensuring smooth integration between stakeholders.

Chapter 3 Related Work

This chapter includes some of the previous work done that is related to our project and explores their methodologies. There are some hospital's own applications/websites that give us information about nearby hospitals and doctors available in that particular hospital but there is no comprehensive application that provides real-time data about nearby hospitals, available beds and doctors, first-aid guidance and efficient emergency reporting techniques. Moreover, there are applications that we can access during an emergency but they are not available for everyone.

3.1 Definitions, Acronyms, and Abbreviations

KPK: Khyber Pakhtunkhwa

PSCA: Punjab Safe Cities Authority

CPR: Cardiopulmonary Resuscitation

PAK: Pakistan

NDMA: Natural Disaster Management Authority

IGP: Inspector General of Police

PPIC3: Punjab Police Integrated Command, Control, and Communication.

3.2 Detailed Literature Review

Following is the detailed literature review relevant to our project.

3.2.1 Rescue 1122

Rescue 1122 is an emergency response service that serves almost all regions of Pakistan including KPK, Sindh, Balochistan and Punjab. It is accessible in any region by calling 1122 from any phone within Pakistan.

3.2.1.1 Summary of Rescue 1122

Rescue 1122 was launched as an emergency service in different provinces in Pakistan at different years. It provides quick help in case of emergencies and accidents. It was launched to cater different emergencies like medical emergencies, fires and road accidents etc efficiently. It is available 24/7 throughout Pakistan and is known for its well trained staff. The aim of this service is to save lives and reduce the impact of emergencies by providing quick and effective responses.

In addition to emergency services, Rescue 1122 also focuses on community safety and preparedness. They offer training programs to educate people on how to handle emergencies and prevent accidents. Over the years, Rescue 1122 has expanded its reach to remote areas, making sure everyone is able to get help. This service has helped in reducing the number of critical situations, making it a crucial part of Pakistan's public safety infrastructure.

3.2.1.2 Critical analysis of Rescue 1122

The strengths of Rescue 1122 service includes fast response time, well trained staff, public awareness like different programs to train people regarding first aid guidance and preventing accidents and a wide range of services like firefighters, ambulances and disaster management. The weaknesses of Rescue 1122 include limited resources like equipment, slower response time during peak periods and dependence on government funding.

3.2.1.3 Relationship to Rescue 1122

Our project will also cater to emergency situations like fires, thefts, medical emergencies similar to the rescue 1122 service but it will also provide real time data for hospitals and AI based first aid guidance system instead of organizing training programs.

3.2.1.4 Conclusion

In conclusion, Rescue 1122 service initiated after 2004 in Punjab has helped in reducing the number of critical situations like medical emergencies, fires, road accidents by providing immediate help yet it still has some drawbacks like limited resources, no real time access to hospitals information like beds etc, and dependency on government funding.

3.2.2 Women Safety App by Punjab Police

Women safety app by Punjab Police is launched for women to feel safer and get help in case of emergency. It is an application built primarily for females and is only accessible in Punjab at the moment.

3.2.2.1 Summary of Women Safety App

Women safety app is launched by Punjab Police specifically for women in emergencies. This app allows women to notify police through alerts if they are in danger, like being followed or harassed. By pressing a button, the police can track the women's location and send help immediately. It also has a feature to notify nearby family members or friends for immediate support through call. This app helps women feel secure when going out alone in unsafe areas.

Additionally, this app contains all the useful information regarding women's rights, laws and awareness material/documents. It also enables users to report harassment or unsafe situations directly to the authorities. Moreover, it enables women to determine the place by location reviews before going to that specific place and fill a survey on how safe, somewhat safe, and non-safe a location was to other women who may wish to visit the same area. This initiative is a part of the government's effort to make the environment safer for women.

3.2.2.2 Critical analysis of Women Safety App

The strengths of this application include quick response time, user friendliness, sending alerts to police, nearby family members and friends and containing information about women's rights and laws. The weaknesses of this application are that it is gender specific and region specific i.e it is available only for women of Punjab. Additionally, it does not cater to medical emergencies, fires, theft etc.

3.2.2.3 Relationship to Women Safety App

Our project will allow women to share their live location and send alerts to police and family members in a similar way as the women safety app but it will also cater other emergencies like medical, theft, fire and natural disasters. Moreover, our project will not be gender specific allowing everyone to use it.

3.2.2.4 Conclusion

In conclusion, the Women Safety app launched by Punjab police is a great initiative to help women in danger. Its features including, location sharing, location feedback, alerting police and family members, access of information about women's rights and reporting emergencies directly to authorities has provided a safer environment for women yet it is not a comprehensive solution as it does not cater medical emergencies and is a gender specific application available only in Punjab.

3.2.3 Edhi Foundation

Edhi Foundation is the largest and the most effective welfare organization in Pakistan which has been working on non-commercial, non-political and non-sectarian basis round the clock without any partiality of color, class and creed.

3.2.3.1 Summary of Edhi Foundation

Edhi foundation was started in 1951 by Abdul Sattar Edhi. The foundation operates a vast network of services that include emergency medical assistance, orphanages, homeless shelters, and free hospitals.

One of its key services is the Edhi Ambulance, which is the world's largest volunteer-run ambulance network, providing quick and reliable transportation for the sick and injured. The organization is dedicated to helping the poor and marginalized.

The Edhi Foundation also runs rehabilitation centers, maternity homes, and programs for abandoned children, aiming to support those in need across the country. Its operations are entirely funded by donations, and it has gained immense trust and respect both locally and internationally for its transparency and dedication. Abdul Sattar Edhi's philosophy of selflessness and humanitarianism continues to inspire the foundation's work, making it a lifeline for millions of people in Pakistan.

3.2.3.2 Critical analysis of Edhi Foundation

The strengths of this service is that it operates across the whole nation, provides a wide range of services like healthcare, shelters, orphanages, burial services and disaster reliefs and is highly trusted for its transparency and non-discriminatory services.

The weaknesses of this service is that it entirely depends on donations, has limited resources, and it does not cater to emergency situations like fires, thefts, harassments etc.

3.2.3.3 Relationship to Edhi Foundation

Our project will be built upon Edhi's ambulance service by incorporating technology i.e users will be able to track the real time location of the ambulance and will have real time access to nearby hospitals data. It will also contain AI chatbot to provide first-aid guidance in case of critical situations

3.2.3.4 Conclusion

In conclusion, the Edhi Foundation is a crucial part of Pakistan's welfare system, providing essential services like shelters, ambulances, disaster reliefs, rehabilitation services, burial services and orphanages. However, its dependency on donations and limited resources can hinder the services sometimes. Moreover, due to the emerging world in technology, we need a comprehensive online emergency service that will ensure quick response time and better coordination during emergencies.

3.2.4 Punjab Police App

Punjab Police Pakistan offers a user-friendly digital platform that simplifies access to a range of police services for the residents of Punjab. Users can conveniently file complaints, report crimes, and utilize services such as police verification and tenant registration directly from their mobile devices. The app is designed to enhance public convenience and improve the overall delivery of police services.

3.2.4.1 Summary of Punjab Police App

Punjab Police Pakistan is one of those comprehensive digital platforms which has been designed to offer numerous police services to citizens in the province of Punjab, Pakistan. The application essentially increases the availability of law enforcement services as citizens can communicate with their police department straight from their personal mobile phones without any hassle. This will include online complaint filing, reporting of crimes, seeking character certificates, police verification, and registration of private employees and tenants. Other services include traffic advisory, "Find My Police Station," and filling of complaints at IGP, which finally ensure a complete user experience with the app. The aim is to simplify citizen-police interactions and further enable improvement in service delivery.

3.2.4.2 Critical analysis of Punjab Police App

Punjab Police Pakistan reduces the number of visits physically, elements like online complaints, reporting of crimes are used, and also promotes transparency through direct access to higher officials while offering versatile services like tenant registration and employee verification. Probably its user interface and navigation need to be improved. It would also face challenges reaching citizens who don't have access to smartphones or are digitally not so literate. As it handles sensitive data, the storage security has to be strong

3.2.4.3 Relationship to Punjab Police App

Both Muhaafiz and the Punjab Police App are designed to enhance emergency reporting and facilitate better communication between users and authorities. The Punjab Police App primarily targets law enforcement and crime reporting, whereas Muhaafiz broadens this idea by incorporating healthcare emergency reporting and first aid advice, providing a more comprehensive approach to public safety.

3.2.4.4 Conclusion

Punjab Police Pakistan reduces the number of visits physically, elements like online complaints, reporting of crimes are used, and also promotes transparency through direct access to higher officials while offering versatile services like tenant registration and employee verification. Probably its user interface and navigation need to be improved. It would also face challenges reaching citizens who don't have access to smartphones or are digitally not so literate. As it handles sensitive data, the storage security has to be strong

3.2.5 Pak Life Saver

The Pak Life Saver Program is a groundbreaking initiative by Punjab Emergency Services designed to equip citizens with vital CPR skills using digital platforms. It offers a combination of online courses and hands-on training to enhance survival rates in cases of cardiac arrest. This program aims to cultivate leadership and a sense of civic duty among the youth of Pakistan.

3.2.5.1 Summary of Pak Life Saver App

The Pak Life Saver Program is an initiative by Punjab Emergency Services that leverages ICT to equip citizens, especially the youth, with essential CPR skills, enhancing safety and saving lives. Through its mobile app and web portal, individuals can register, access vital course materials, take online exams, and participate in hands-on training at local CPR centers to earn certification. The program's goal is to boost survival rates for cardiac arrest, teach lifesaving skills, and encourage civic responsibility among young people in Pakistan.

3.2.5.2 Critical analysis of Pak Life Saver app

The program provides extensive access to CPR training via online courses, making essential lifesaving skills easier to obtain. It encourages leadership and civic responsibility, with the goal of enhancing Pakistan's global reputation. However, depending on digital platforms might restrict access for individuals lacking internet or technological skills, and the need for in-person certification could create obstacles in remote regions. It's also vital to maintain consistent quality across all training centers.

3.2.5.3 Relationship to Pak Life Saver

Both Muhaafiz and the Pak Life Saver App are dedicated to empowering people with vital lifesaving skills and information. The Pak Life Saver focuses on CPR and first aid training, while Muhaafiz introduces an AI-assisted chatbot that delivers quick first aid guidance when emergencies arise. This collaboration ensures that both platforms offer valuable support in critical healthcare situations.

3.2.5.4 Conclusion

The program offers broad access to CPR training through online courses, simplifying the acquisition of crucial lifesaving skills. It promotes leadership and civic responsibility, aiming to improve Pakistan's standing on the global stage. However, reliance on digital platforms may limit access for those without internet or technological proficiency, and the requirement for in-person certification could pose challenges in remote areas. Additionally, ensuring consistent quality across all training centers is essential.

3.2.6 PAK NDMA-Disaster Alert Application

The NDMA's disaster management application aims to enhance public safety by offering early warning alerts and encouraging community involvement. It provides timely risk communication, comprehensive alert descriptions, and guidelines for before, during, and after emergencies. This platform is essential in equipping users to handle potential disasters, adhering to international standards.

3.2.6.1 Summary of PAK NDMA

This comprehensive platform, created by NDMA, delivers early warning alerts and encourages community involvement to improve public safety and disaster preparedness. It provides timely risk communication, clear color-coded alerts, and consistent updates during and after emergencies. By following international best practices, the platform acts as an essential resource for disaster management and public awareness, offering guidelines, public service messages, and videos to enhance understanding.

3.2.6.2 Critical analysis of PAK NDMA

The platform stands out by providing timely alerts, comprehensive descriptions, and practical steps for users in times of disaster. Its commitment to international standards and the inclusion of video clips for better understanding are significant advantages. However, it might encounter difficulties in reaching remote areas that lack internet access or technological familiarity, and there may be valid concerns regarding user privacy and data security during emergencies.

3.2.6.3 Relationship to NDMA

Muhaafiz and NDMA both prioritize preparedness and efficient emergency response. NDMA is responsible for issuing early warning alerts for natural disasters and offering public safety guidelines, while Muhaafiz addresses everyday emergencies like medical situations by providing real-time hospital information and first aid advice to assist users in managing crises. Both platforms aim to minimize response times and improve safety awareness.

3.2.6.4 Conclusion

This platform is an essential initiative by NDMA aimed at enhancing public safety and disaster response. By providing timely alerts and straightforward guidelines, it enables users to make informed choices during emergencies. To increase its effectiveness, efforts should concentrate on improving accessibility and security for all users, especially in at-risk areas.

3.2.7 The Punjab Safe Cities Authority (PSCA)

The Punjab Safe Cities Authority (PSCA) is enhancing the public safety by adopting information technology for establishing a common command, control and communication center for police across the Punjab province. The app enables verified users to get rapid responses at specific locations and provides live traffic updates to facilitate smoother travel. By pressing the panic button, users can quickly connect to emergency services for immediate help.

3.2.7.1 Summary of PSCA

The PSCA app merges law enforcement services with smart technology to ensure fast and accurate emergency responses. It registers users as verified citizens, which allows for precise location-based assistance, and provides real-time traffic updates to enhance travel planning. The authority's broader goal includes the installation of surveillance cameras at over 2,000 locations to improve public safety across major cities in Punjab through its integrated command center (PPIC3).

3.2.7.2 Critical analysis of PSCA

This app offers rapid and accurate emergency responses through location-based services and verified user registrations. It also provides live traffic updates for added convenience, and its extensive surveillance network boosts public safety. However, the app's effectiveness could be compromised in areas with limited internet connectivity, and large-scale surveillance might raise privacy concerns. Furthermore, expanding the network to additional cities could introduce logistical challenges.

3.2.7.3 Relationship to PSCA

Muhaafiz works alongside PSCA to promote public safety through technology. It integrates real-time location tracking for emergency responders and employs a centralized system to effectively manage critical information. Similar to how PSCA improves response times with its integrated command systems, Muhaafiz focuses on connecting users directly with emergency responders in real time, leading to quicker and more informed decisions during emergencies.

3.2.7.4 Conclusion

The PSCA app is an important resource for boosting public safety in Punjab, merging technology with law enforcement to provide quick, real-time responses. With its easy-to-use features and broad surveillance reach, the app has the ability to significantly improve security and emergency services. However, it will be crucial to resolve connectivity problems and privacy concerns for it to succeed in the long run.

3.3 Literature Review Summary Table

The related work summary table is shown in table 3.1

Table 3.1: Related Work Review Summary Table

Application	Features	Relevance	Limitations
Rescue 1122 [1]	Emergency services, community training	Provides real-time information on emergencies and ambulance services	Limited geographic availability and may require continuous technical updates for accuracy
Women Safety App [2]	Safety alerts, GPS tracking	Provides quick emergency alerts and location tracking for safety	Often restricted to specific areas, may depend on internet access, and raises concerns about data privacy
Edhi Foundation [3]	Donation management, Service listings	Enables users to request ambulance services	Limited functionality regarding expansion and service updates, it depends significantly on manual updates
Punjab Police App [4]	Crime reporting, Alerts	It allows users to report crimes and receive notifications about local incidents	May experience delays in response times and have restricted integration with local law enforcement systems
Pak Life Saver[5]	First aid instructions, Emergency contacts	Offers immediate access to first aid guidelines and links users to emergency services	Limited medical coverage, particularly in sparsely populated areas, and it may have incomplete information for contacts
PAK NDMA-Disaster Alert Application[6]	Disaster alerts, Risk maps	Issues timely alerts for natural disasters and offers assessments of disaster risks	Coverage may be limited in rural areas and depends on a strong communication network for timely alerts
PSCA [7]	Surveillance, Crime alerts	Uses surveillance data to issue crime alerts and improve public safety monitoring	Limited access in regions beyond urban centers and possible privacy issues associated with surveillance

3.4 Conclusion

The literature review discusses various emergency services and applications available in Pakistan, such as Rescue 1122, the Women Safety App, the Edhi Foundation, the Punjab Safe Cities Authority, NDMA, Pak Life Saver, and the Punjab Police app. Each of these services provides essential support, ranging from medical help to law enforcement and public safety. However, they face challenges like limited resources, restricted geographic or gender-specific access, and reliance on donations or government funding. The proposed Muhaafiz app seeks to fill these gaps by offering a comprehensive emergency response system that includes real-time hospital information, AI-driven first aid guidance, and a wider range of services to address different types of emergencies.

Chapter 4 Software Requirement Specifications

This section offers a detailed and comprehensive assessment of the Software Requirements for our project which include: list of features, the functional and non-functional requirements, quality attributes, assumptions, hardware and software requirements, relevant use cases, graphical user interfaces, and a thorough risk analysis.

4.1 List of Features

Our system will provide the following features:

- A comprehensive incident categorization system.
- Automated emergency calls during emergencies.
- Real-time hospital navigation for users and ambulances.
- Hospital dashboard to update the availability of medical resources.
- A specialized chat-bot that answers queries related to the application and provides first aid information.
- Streamlined hospital admission procedure through paperwork automation.
- Facial recognition for identification during emergencies.
- A safety timer that alerts relevant authorities if not deactivated in the designated time.
- Online consultancy with the best doctors in Pakistan.
- A user-friendly appointment scheduling system.
- Upload and save medical data and history for future reference.

4.2 Functional Requirements

The following functional requirements shall be provided by our system:

- The user shall be able to report various emergencies quickly.
- The system shall trigger automated calls to relevant authorities with the user's live location.
- The user shall be able to receive real-time navigation to the nearest hospitals with necessary medical facilities.
- The user shall be able to receive first aid information and guidelines via a specialized chat-bot.

- The user shall be able to use a safety timer that alerts the authorities and close contacts if the user has not reached the specified destination on time.
- The user shall be able to schedule online appointments with relevant doctors.
- The system shall allow doctors to accept and cancel an appointment
- The system shall allow the users to fill the hospital admission paperwork through the app.
- The system shall allow the users to upload their medical data and history in the app.
- The system shall employ facial recognition to promptly recognize individuals in distress or emergency situations.
- The system shall facilitate online medical consultations.
- The system shall allow hospitals to update the availability of medical resources (beds, doctors, equipment) via the dashboard.
- The system shall allow users to edit their personal information
- The system shall allow users to add and remove emergency contacts

4.3 Quality Attributes

The system will comprise the following quality attributes that are critical to its efficient operation.

4.3.1 Usability

The system will provide a user friendly and intuitive interface so that the user can easily navigate the application.

4.3.2 Maintainability

The system will be easy to maintain and manage, the integration of new functionalities and modules can be done seamlessly due to the modular structure of the code.

4.3.3 Reliability

The system will be reliable and will provide accurate and real-time information regarding the availability of doctors and medical equipment at hospitals.

4.3.4 Availability

The system will be available 24 hours and it will readily cater to the user's requests during emergencies.

4.3.5 Testability

The system will be divided into various modules to facilitate quality assurance and testing, the modules will be highly cohesive and loosely coupled which will ensure that each module can be tested independently.

4.3.6 Integrity

The system will provide access only to authorized personnel to view a user's medical history and personal information in emergency situations.

4.3.7 Reusability

The system will have a modular design, this will enable the reuse of components and modules. The code will also be clean and maintainable.

4.3.8 Robustness

The system will be able to cater to a large number of users i.e., 100 without crashing and compromising the quality of the application.

4.3.9 Correctness

The system will provide accurate real-time information regarding hospitals, it will also navigate the user to the nearest hospital with available resources.

4.4 Non-Functional Requirements

The following non-functional requirements shall be provided by our system:

- The system shall provide accurate real-time information regarding hospitals.
- The system shall be scalable and will cater to a 100 users without crashing.
- The system shall be easily usable and will have a user-friendly interface.
- The system shall allow only authorized personnel to view a user's personal information.
- The system shall be divided into modules and will be easy to maintain, manage and test.

- The system shall be available 24 hours a day to address all kinds of emergencies.
- The system shall cater 100 users without performance degradation.
- The system shall provide real time information and accurate navigation promptly.
- The system shall be extensible and will offer cohesive integration of new modules and functionalities.
- The system shall be made reusable because of the modular nature of the code.

4.5 Assumptions

The following assumptions have been made for the users of this system:

- The user shall be able to understand either English or Urdu in order to use the system.
- The user shall have a basic understanding of technology in order to use the system.
- The user shall have a smart phone with an active internet connection and Location Services enabled.
- The user shall provide a valid identity card/contact number so that the system can verify the user's identity.

4.6 Use Cases

4.6.1 Use Cases for Mobile Application

4.6.1.1 Login

Name		Login	
Actors		User, Doctor	
Summary		The user shall provide their email/contact number and password on the login form, and after successful verification, the user is redirected to the home page.	
Pre-Conditions		The user must be in the database records The user must not already be logged in.	
Post-Conditions		The user’s session is successfully established and shall be redirected to the home page.	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	The user opens the login page.	2	The login page is displayed asking for email/contact number and password.
3	The user enters valid credentials.	4	The system verifies the credentials, and establishes a session for the user and redirects the user to the home page.
Alternative Flow			
3	The user enters invalid email/ contact number or password.	4-A	The system responds with an error message: Incorrect email/contact number or password entered.

Table 4.1: Use Case for Login

4.6.1.2 Signup

Name		Signup	
Actors		User, Doctor	
Summary		The user shall provide their valid email/contact number, name, and password on the signup form, and after successful verification, the user is redirected to the login page.	
Pre-Conditions		The user must not be in the database records.	
Post-Conditions		The user’s account is registered and their credentials are saved in the Database.	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	The user opens the signup page.	2	The signup page is displayed asking for email/contact number, name and password.
3	The user enters valid credentials.	4	The system verifies the credentials, and stores them in the Database and redirects the user to the login page.
Alternative Flow			
3	The user enters invalid email/ contact number	4-A	The system responds with an error message: Invalid email/contact number entered.

Table 4.2: Use Case for Sign Up

4.6.1.3 Incident Reporting

Name		Incident Reporting	
Actors		User	
Summary		The user shall open the home page and choose the emergency type from an intuitive list after which the authorities will be alerted and the user’s live location will be shared.	
Pre-Conditions		The user must have an active internet connection and location services enabled	
Post-Conditions		The relevant authorities are alerted and help is dispatched.	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	The user opens the home page.	2	The home page is displayed with an intuitive incident categorized list.
3	The user selects an emergency.	4	The system alerts the authorities, and redirects the user to the emergency page where user can view the location of the responders.
Alternative Flow			
3	The user cancels the emergency request	4-A	The system alerts the authorities about the false alarm.

Table 4.3: Use Case for Incident Reporting

4.6.1.4 Set Safety Timer

Name		Set safety timer	
Actors		User	
Summary		The user opens the safety timer page, sets a destination, and the app calculates the estimated time. If the user fails to reach on time the authorities will be alerted with the user’s live location	
Pre-Conditions		The user must be logged in The user must have an active internet connection and location services enabled	
Post-Conditions		The relevant authorities are alerted and help is dispatched.	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	The user opens the timer page.	2	The timer page displays an intuitive timer and location tracking map
3	The user sets the destination.	4	The system estimates the time to reach the destination the timer is deactivated upon successful arrival
Alternative Flow			
3	The user does not reach on time	4-A	The system alerts the authorities with the user’s live location.

Table 4.4: Use Case to set Safety Timer

4.6.1.5 First Aid Guidance

Name		First aid guidance	
Actors		User	
Summary		The user opens the chat-bot page, enters a query and the chat-bot responds accordingly.	
Pre-Conditions		The user must have an active internet connection	
Post-Conditions		The specialized chat-bot responds accordingly	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	The user opens the chat-bot page and enters a query.	2	The page displays an intuitive, easy-to-understand response
Alternative Flow			
1	The user does not enter a query	2-A	The system does not respond.

Table 4.5: Use Case for AI Assistant- First Aid Guide

4.6.1.6 Call an Ambulance

Name		Call an Ambulance	
Actors		User	
Summary		The user opens the homepage, then clicks on the medical emergency situation button and press on call the ambulance button which will call the ambulance.	
Pre-Conditions		The user must have an active internet connection and should be logged in the application	
Post-Conditions		Ambulance is called and the location is shared with the user	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	The user clicks on the "Medical emergency" button on the Homepage .	2	User is redirected to the Medical Emergency Page
3	The user clicks on the "Call An Ambulance" button on the Homepage .	4	Ambulance is called and user is redirected to the page where location is shared
Alternative Flow			
1	The user does not click on the button on homepage	2-A	The system does not respond.

Table 4.6: Use Case for calling an ambulance

4.6.1.7 Find Nearest Hospital

Name		Find Nearest Hospital	
Actors		User	
Summary		The user opens the homepage, then clicks on the medical emergency situation button and press on find nearest hospital which will give the user a list of nearby hospitals.	
Pre-Conditions		The user must have an active internet connection, should be logged in the application and location access should be enabled	
Post-Conditions		User is directed to the selected hospital	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	The user clicks on the "Medical emergency" button on the Homepage .	2	User is redirected to the Medical Emergency Page
3	The user clicks on the "Find nearest Hospital" button on the Homepage .	4	System shows a list of nearby hospitals with details like beds available
Alternative Flow			
1	The user does not click on the button on homepage	2-A	The system does not respond.

Table 4.7: Use Case to find the nearest hospital

4.6.1.8 View Hospital Information

Name		View Hospital Information	
Actors		User	
Summary		User opens the homepage, then clicks on the medical emergency situation button and press on find nearest hospital and click on any hospital to view its information.	
Pre-Conditions		The user must have an active internet connection and should be logged in the application	
Post-Conditions		User is given the details of the hospital being searched	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	The user clicks on the "Medical emergency" button on the Homepage .	2	User is redirected to the Medical Emergency Page
3	User clicks on the "Find nearest Hospital" button on the Homepage .	4	System shows a list of nearby hospitals.
5	User clicks on specific hospital .	6	System shows the details like beds, doctors, departments etc in that hospital.
Alternative Flow			
1	The user does not click on the button on homepage	2-A	The system does not respond.

Table 4.8: Use Case to view hospital information

4.6.1.9 Consult a Doctor

Name		Consult a doctor	
Actors		User	
Summary		User opens the homepage, then clicks on the medical emergency button and press consult a doctor button to find all the doctors that are registered in the application with ratings.	
Pre-Conditions		The user must have an active internet connection and should be logged in the application	
Post-Conditions		User is given the details of the doctors being searched	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	User clicks on the "Medical emergency" button on the Homepage .	2	User is redirected to the Medical Emergency Page
3	User clicks on the "Consult a Doctor" button on the Homepage .	4	System shows a list of doctors registered with their ratings.
5	User clicks on specific doctor .	6	System shows doctors detail.
7	User clicks on book appointment .	8	System books the appointment with the doctor and doctor is notified
Alternative Flow			
3	The user does not click on the button on homepage	2-A	The system does not respond.

Table 4.9: Use Case for consulting a doctor

4.6.1.10 Add Emergency Contacts

Name		Add Emergency Contacts	
Actors		User	
Summary		User can add a new emergency contact.	
Pre-Conditions		The user must have an active internet connection and should be logged in the application	
Post-Conditions		User can add emergency contacts	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	User clicks on profile icon on home page .	2	User is redirected to the Profile Page
3	User clicks on the "Emergency Contacts" button.	4	User is redirected to the emergency contacts page
5	User writes the contact number and click on add button .	6	Emergency contact is added
Alternative Flow			
5	User writes irrelevant number.	6-A	System shows an error message.
5	User clicks on add button without writing contact number.	6-A	System shows an error message.

Table 4.10: Use Case to add emergency contacts

4.6.1.11 View Emergency Contacts

Name	View Emergency Contacts		
Actors	User		
Summary	User can view emergency contacts.		
Pre-Conditions	The user must have an active internet connection and should be logged in the application		
Post-Conditions	User can view emergency contacts		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	User clicks on profile icon on home page.	2	User is redirected to the Profile Page
3	User clicks on the "Emergency Contacts" button.	4	User is redirected to the emergency contacts page where emergency contacts are listed

Table 4.11: Use Case to view emergency contacts

4.6.1.12 Manage User Profile

Name		Manage User Profile	
Actors		User	
Summary		User can view and edit its profile by navigating to the profile page.	
Pre-Conditions		The user must be logged in The user must have an active internet connection	
Post-Conditions		User Profile is managed according to users needs.	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	User click on the profile icon on homepage.	2	System redirects the user to the profile page
3	User clicks on ”personal information” button.	4	System opens the personal information page
5	User edits the personal information and click ”save”.	6	System edits the personal information according to users needs
Alternative Flow			
5	User does not click on ”save”	6-A	System does not save the changes.

Table 4.12: Use Case to manage user profile

4.6.2 Use Cases for Admin Dashboard

4.6.2.1 Add Hospitals

Name		Add Hospitals	
Actors		Admin	
Summary		Admin shall be able to add hospitals	
Pre-Conditions		Admin must be logged in and should have a stable internet connection.	
Post-Conditions		Hospitals will be added according to admins preference.	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Admin navigates to the Hospitals page on admin dashboard.	2	Hospitals page is opened where all the registered hospitals are listed.
3	Admin clicks on 'add' button.	4	System opens the form to add hospitals.
5	Admin fills the form and clicks on add.	6	Systems adds the hospital and redirects to the hospitals page.
Alternative Flow			
5	Admin does not click on add.	6-A	The system does not add the hospital

Table 4.13: Use Case to add hospitals

4.6.2.2 Remove Hospitals

Name		Remove Hospitals	
Actors		Admin	
Summary		Admin shall be able to remove hospitals	
Pre-Conditions		Admin must be logged in and should have a stable internet connection.	
Post-Conditions		Hospitals will be removed according to admins preference.	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Admin navigates to the Hospitals page on admin dashboard.	2	Hospitals page is opened where all the registered hospitals are listed.
3	Admin clicks on 'remove' button of the respective hospital that is to be deleted.	4	System shows a confirmation message.
5	Admin clicks on "confirm".	6	Systems removes the hospital and redirects to the hospitals page.
Alternative Flow			
5	Admin clicks on cancel when confirmation message is shown.	6-A	The system does not remove the hospital

Table 4.14: Use Case to remove hospitals

4.6.2.3 Remove Users

Name		Remove Users	
Actors		Admin	
Summary		Admin shall be able to remove users that are a mis-using the application	
Pre-Conditions		Admin must be logged in and should have a stable internet connection.	
Post-Conditions		Users will be removed according to admins preference.	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Admin navigates to the Users page on admin dashboard.	2	Users page is opened where all the registered users are listed.
3	Admin clicks on 'remove' button of the respective user that is to be deleted.	4	System shows a confirmation message.
5	Admin clicks on "confirm".	6	Systems removes the user and redirects to the users page.
Alternative Flow			
5	Admin clicks on cancel when confirmation message is shown.	6-A	The system does not remove the user

Table 4.15: Use Case to remove users

4.6.3 Use Cases for Hospital Dashboard

4.6.3.1 Add Beds

Name		Add Beds	
Actors		Hospital Admin	
Summary		Hospital Admin shall be able to add number of beds currently available	
Pre-Conditions		Hospital Admin must be logged in and should have a stable internet connection.	
Post-Conditions		Bed count will be increased	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Hospital Admin navigates to the beds page from the menu bar opens the page on dashboard.	2	Bed’s page is opened where the bed count is displayed
3	Hospital Admin clicks on ’add’ button corresponding to beds and the count is incremented	4	System shows a confirmation message.
5	Admin clicks on ”confirm”.	6	Systems updates the count
Alternative Flow			
5	Hospital Admin clicks on cancel when confirmation message is shown.	6-A	The system does not change the bed count

Table 4.16: Use Case to add beds

4.6.3.2 Remove Beds

Name	Remove Beds																
Actors	Hospital Admin																
Summary	Hospital Admin shall be able to reduce number of beds currently available																
Pre-Conditions	Hospital Admin must be logged in and should have a stable internet connection.																
Post-Conditions	Bed count will be decreased																
Special Requirements	None																
Basic Flow																	
<table><tr><th colspan="2">Actor Action</th><th colspan="2">System Response</th></tr><tr><td>1</td><td>Hospital Admin navigates to the beds page from the menu bar opens the page on dashboard.</td><td>2</td><td>Bed’s page is opened where the bed count is displayed</td></tr><tr><td>3</td><td>Hospital Admin clicks on ’remove’ button corresponding to beds and the count is updated</td><td>4</td><td>System shows a confirmation message.</td></tr><tr><td>5</td><td>Hospital Admin clicks on ”confirm”.</td><td>6</td><td>Systems updates the count</td></tr></table>		Actor Action		System Response		1	Hospital Admin navigates to the beds page from the menu bar opens the page on dashboard.	2	Bed’s page is opened where the bed count is displayed	3	Hospital Admin clicks on ’remove’ button corresponding to beds and the count is updated	4	System shows a confirmation message.	5	Hospital Admin clicks on ”confirm”.	6	Systems updates the count
Actor Action		System Response															
1	Hospital Admin navigates to the beds page from the menu bar opens the page on dashboard.	2	Bed’s page is opened where the bed count is displayed														
3	Hospital Admin clicks on ’remove’ button corresponding to beds and the count is updated	4	System shows a confirmation message.														
5	Hospital Admin clicks on ”confirm”.	6	Systems updates the count														
Alternative Flow																	
5	Hospital Admin clicks on cancel when confirmation message is shown.	6-A	The system does not change the bed count														

Table 4.17: Use Case to remove beds

4.6.3.3 View Beds

Name		View Beds	
Actors		Hospital Admin	
Summary		Hospital Admin shall be able to view the number of beds currently available	
Pre-Conditions		Hospital Admin must be logged in and should have a stable internet connection.	
Post-Conditions		Bed count and information will be displayed	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Hospital Admin navigates to the beds page from the menu bar opens the page on dashboard.	2	Bed’s page is opened where the bed information is displayed
3	Hospital Admin views the page	4	System shows bed count and related information

Table 4.18: Use Case to view beds

4.6.3.4 View Doctors

Name	View Doctors		
Actors	Hospital Admin		
Summary	Hospital Admin shall be able to view all doctors associated with the hospital		
Pre-Conditions	Hospital Admin must be logged in and should have a stable internet connection.		
Post-Conditions	Doctors and related information will be displayed		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	Hospital Admin navigates to the doctors page from the menu bar and opens the page on dashboard.	2	Doctor’s page is opened where the doctors and related information is displayed
3	Hospital Admin views doctors and reviews their corresponding information	4	System shows doctors and their related information

Table 4.19: Use Case to view doctors

4.6.3.5 Add Doctor

Name		Add Doctors	
Actors		Hospital Admin	
Summary		Hospital Admin should be able to add doctors to the system	
Pre-Conditions		Hospital Admin must be logged in and should have a stable internet connection.	
Post-Conditions		New doctor will be added to the system	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Hospital Admin navigates to the doctors page from the menu bar and opens the page on dashboard.	2	Doctor’s page is opened where the doctors and related information is displayed
3	Hospital Admin clicks ”add” button	4	System displays a pop-up form
5	Hospital Admin fills relevant information	6	System verifies all fields
7	Hospital Admin clicks submit form	8	System adds new doctor and updates page
Alternative Flow			
5	Hospital Admin enters invalid information	6-A	The system displays an error
7	Hospital Admin clicks on cancel when form is displayed	8-A	The system does not add new doctor

Table 4.20: Use Case to add doctors

4.6.3.6 Remove Doctor

Name		Remove Doctors	
Actors		Hospital Admin	
Summary		Hospital Admin shall be able to remove doctors from the current doctors	
Pre-Conditions		Hospital Admin must be logged in and should have a stable internet connection.	
Post-Conditions		Selected doctors will be removed from the system	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Hospital Admin navigates to the doctors page from the menu bar and opens the page on dashboard.	2	Doctor’s page is opened where the doctors and related information is displayed
3	Hospital Admin selects doctor and presses ”remove” button	4	System displays a confirmation pop-up
5	Hospital Admin selects ”confirm” button	6	System removes the selected doctor and updates the page
Alternative Flow			
5	Hospital Admin clicks on cancel when confirmation message is shown.	6-A	The system does not remove the doctor

Table 4.21: Use Case to remove doctors

4.6.3.7 View Department

Name	View Department		
Actors	Hospital Admin		
Summary	Hospital Admin shall be able to view departments		
Pre-Conditions	Hospital Admin must be logged in and should have a stable internet connection.		
Post-Conditions	All departments of the related hospital will be displayed		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	Hospital Admin navigates to the departments page from the menu bar and opens the page on dashboard.	2	Department’s page is opened where the departments and related information is displayed
3	Hospital Admin views departments	4	System displays a list departments

Table 4.22: Use Case to view departments

4.6.3.8 Add Department

Name		Add Department	
Actors		Hospital Admin	
Summary		Hospital Admin shall be able to add a department	
Pre-Conditions		Hospital Admin must be logged in and should have a stable internet connection.	
Post-Conditions		A new department will be added to the system	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Hospital Admin navigates to the departments page from the menu bar and opens the page on dashboard.	2	Department’s page is opened where the departments and related information is displayed
3	Hospital Admin clicks ”add” button	4	System displays a pop-up form
5	Hospital Admin fills form with department information	6	System verifies all fields
7	Hospital Admin presses confirm	8	System add a new department and updates the page
Alternative Flow			
3	Hospital Admin clicks on cancel when pop-up form is shown.	4-A	The system does not add any department
5	Hospital Admin enters invalid information	6-A	The system displays and error message

Table 4.23: Use Case to add departments

4.6.3.9 Remove Departments

Name	Remove Departments		
Actors	Hospital Admin		
Summary	Hospital Admin shall be able to remove a department from the current departments		
Pre-Conditions	Hospital Admin must be logged in and should have a stable internet connection.		
Post-Conditions	Selected department will be removed from the system		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	Hospital Admin navigates to the departments page from the menu bar and opens the page on dashboard.	2	Department’s page is opened where the departments and related information is displayed
3	Hospital Admin selects a department and presses ”remove” button	4	System displays a confirmation pop-up
5	Hospital Admin selects ”confirm” button	6	System removes the selected department and updates the page
Alternative Flow			
5	Hospital Admin clicks on cancel when confirmation message is shown.	6-A	The system does not remove the department

Table 4.24: Use Case to remove departments

4.7 Hardware and Software Requirements

The following hardware and software requirements are needed for the development and deployment of this system.

4.7.1 Hardware Requirements

- A computer with 8 GB Ram or more.
- A stable internet connection

4.7.2 Software Requirements

- A modern IDE i.e. VS Code for Web Development and Android Studio for App Development
- Front-end Libraries like React JS and Materialize CSS
- Back-end web application frameworks like Express JS
- Run time environments like Node JS
- Android Toolkits like Jetpack Compose for App Development
- Languages like JavaScript for Web Development and Kotlin for App Development
- A stable browser: Chrome or Firefox

4.8 Graphical User Interface

4.8.1 Welcome Pages

These are the main sliding pages that opens when we start our application

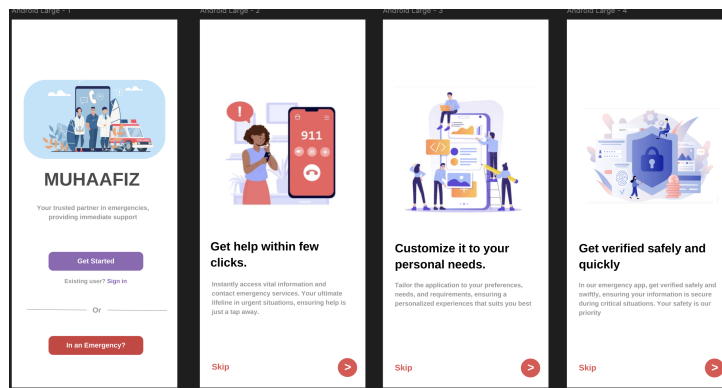
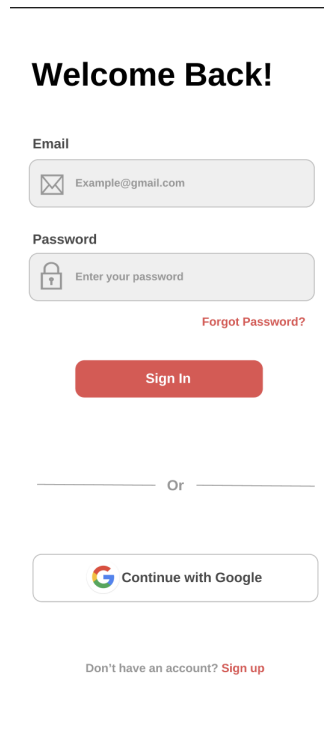


Figure 4.1: Welcome Pages

4.8.2 Login Page

Login page opens when user has decided to login into the application.



The login page features a clean, minimalist design. At the top, the heading "Welcome Back!" is displayed in a bold, black font. Below this, there are two input fields: "Email" and "Password". The "Email" field contains the placeholder text "Example@gmail.com" and is accompanied by an envelope icon. The "Password" field contains the placeholder text "Enter your password" and is accompanied by a lock icon. To the right of the password field, there is a link labeled "Forgot Password?". Below the input fields is a prominent red "Sign In" button. Further down, a horizontal line with the word "Or" in the center separates the login section from the social login section. Below this line is a button labeled "Continue with Google" with the Google logo. At the bottom of the page, there is a link that says "Don't have an account? Sign up".

Figure 4.2: Login Page


4.8.3 Sign up

Sign Up page opens when user wants to register into the application.


Sign Up

Personal Information

Full Name

 Enter full name


Phone Number

 Enter Phone Number


Date of Birth

Day Month Year


Email

 Example@gmail.com

Password

 Enter your password

Confirm Password

 Enter your password

Continue


Already have an account? [Sign in](#)


Figure 4.3: Sign Up page


4.8.4 Home Page


Home page opens when the user has signed up/ logged into the application.


username
we are here for you !





 **EMERGENCY** >

 Medical Emergency

 Motor Accident

 Theft & Harassment

 Fire Emergency

 Natural Disaster





   

Figure 4.4: Home Page

4.8.5 Emergency Page

This page is displayed when user clicks on the "emergency" button on Home page.

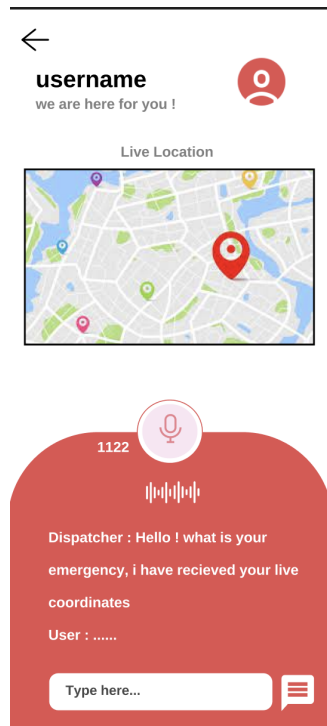


Figure 4.5: Emergency Page

4.8.6 Medical Situation

This page displays when user clicks on the "medical emergency" button on Home page.



Figure 4.6: Medical Emergency page

4.8.7 Consult a Doctor

This page is opened when user wants to consult a doctor. User clicks on "Consult a doctor" button on the medical emergency page.

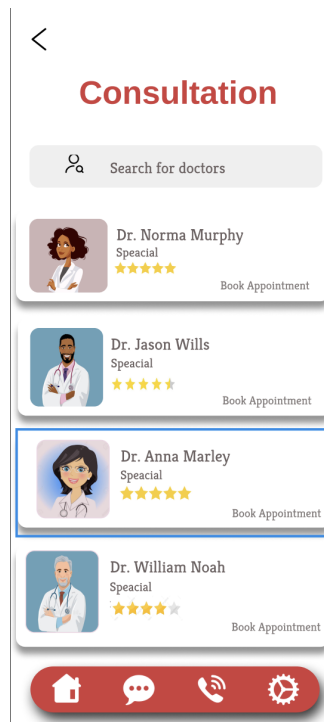


Figure 4.7: Consultation Page

4.8.8 Call an Ambulance

This page is displayed when the user clicks on the "Call an Ambulance" button on Medical emergency page.

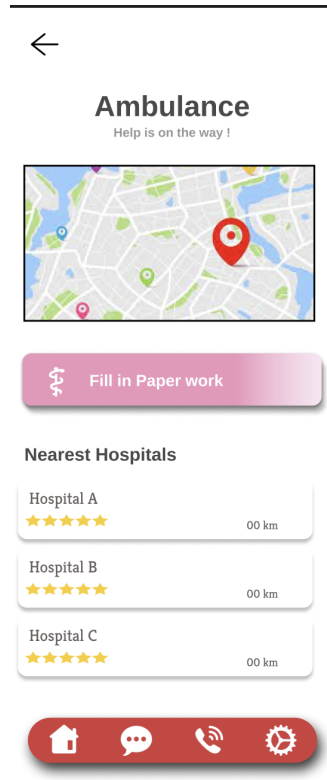


Figure 4.8: Ambulance Tracking Page

4.8.9 Nearby Hospitals

This page opens when user presses the "Nearby Hospitals" button on Medical emergency page.

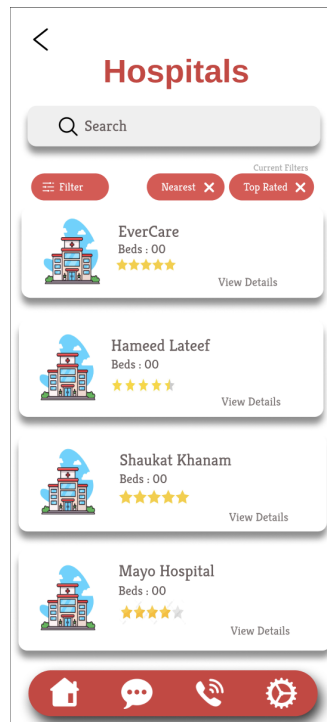


Figure 4.9: Nearby hospitals

4.8.10 Hospitals Page

Hospitals page opens when the user clicks on the any hospital on the Nearby Hospitals page or Ambulance tracking page.

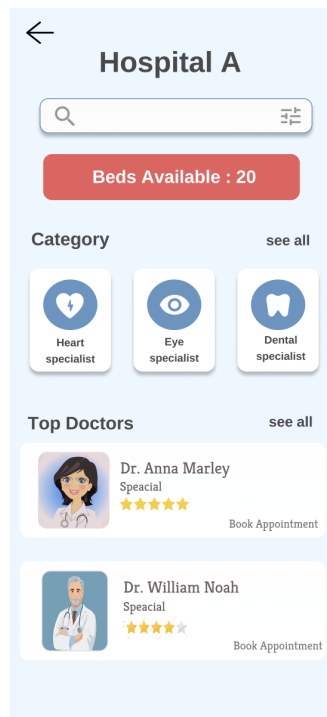


Figure 4.10: Hospitals Page

4.8.11 AI Assistant

AI assistant page will be displayed on clicking the "message" icon in the navigation/menu bar.

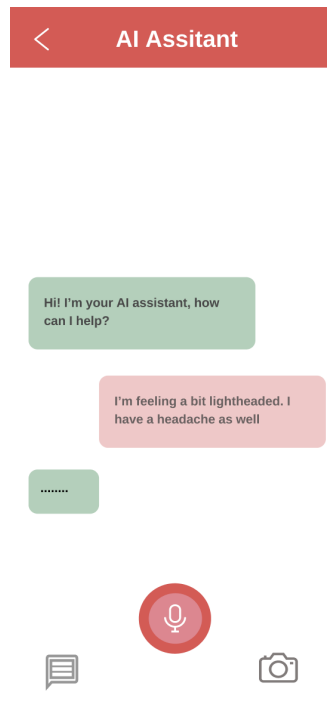


Figure 4.11: AI Assistant

4.8.12 Profile Page

Profile page opens when the user clicks on the profile icon on homepage.

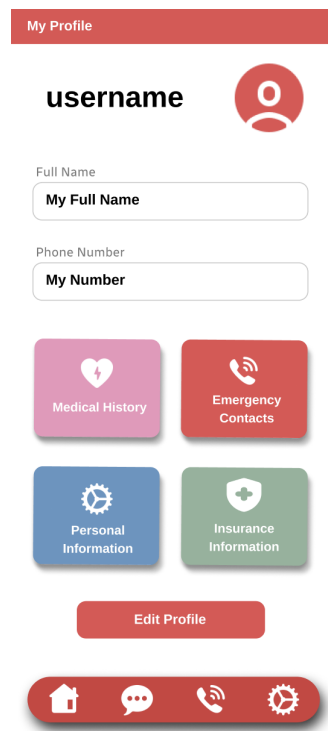


Figure 4.12: Profile Page

4.8.13 Personal Information

Personal Information page opens when the user clicks on the personal information button on the profile page

←

Personal Information

edit your personal information

Full Name

Enter full name

Address

Enter full address

Phone Number

Enter Phone Number

Date of Birth

Day Month Year

Email

Example@gmail.com

CNIC

Enter CNIC

Marital Status

☒ Single ☐ Married

Gender

☒ Male ☐ Female

Profile Picture

Add profile picture

Bio

Add bio

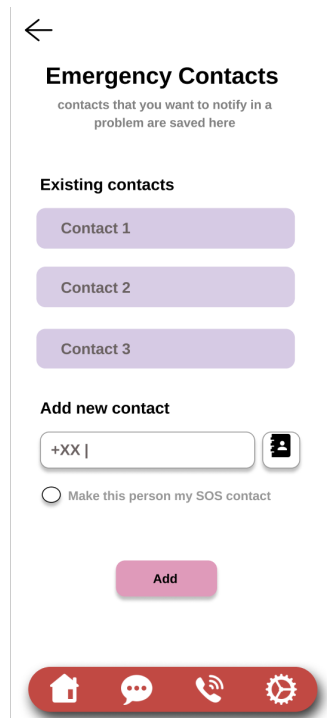
Save

Close without saving

Figure 4.13: Personal Information

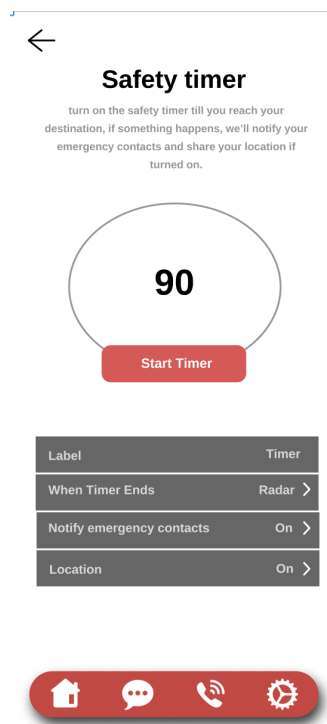
4.8.14 Emergency Contacts

This page appears when the button of the emergency contact located on the profile page is clicked.

**Figure 4.14: Emergency Contacts**

4.8.15 Safety Timer

This page opens when the user clicks on the timer icon in the navigation bar

**Figure 4.15: Safety Timer**

4.8.16 Admin Dashboard - Homepage

This is the admin dashboard homepage

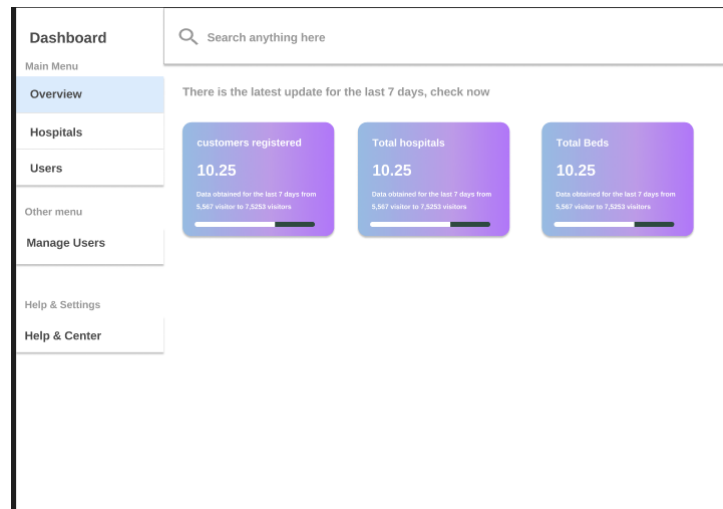


Figure 4.16: Admin Dashboard - Homepage

4.8.17 Admin Dashboard - User's Page

This is the admin dashboard users page, where admins can track users of the application.

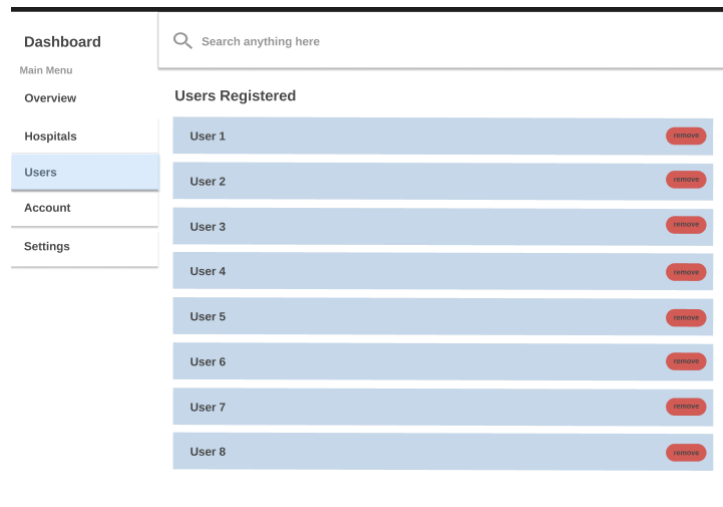


Figure 4.17: Admin Dashboard - User's Page

4.8.18 Admin Dashboard - Hospital's page

This is the admin dashboard hospitals page, where admins can track the registered hospitals.

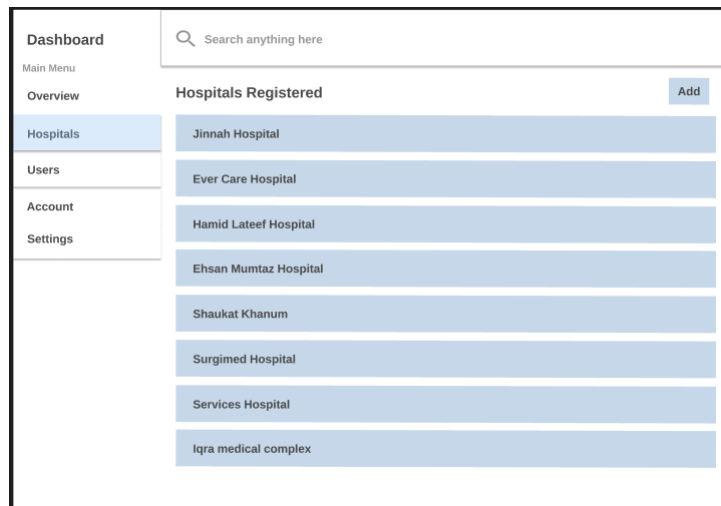


Figure 4.18: Admin Dashboard - Hospital's page

4.8.19 Admin Profile Page

This is the admin Profile page, where they can edit their personal information.

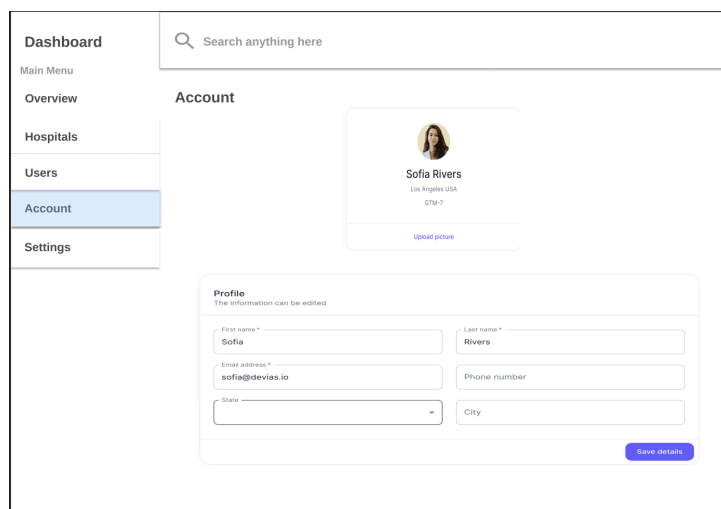


Figure 4.19: Admin Profile Page

4.8.20 Hospital Dashboard – Homepage

This is the Hospitals dashboard's homepage.

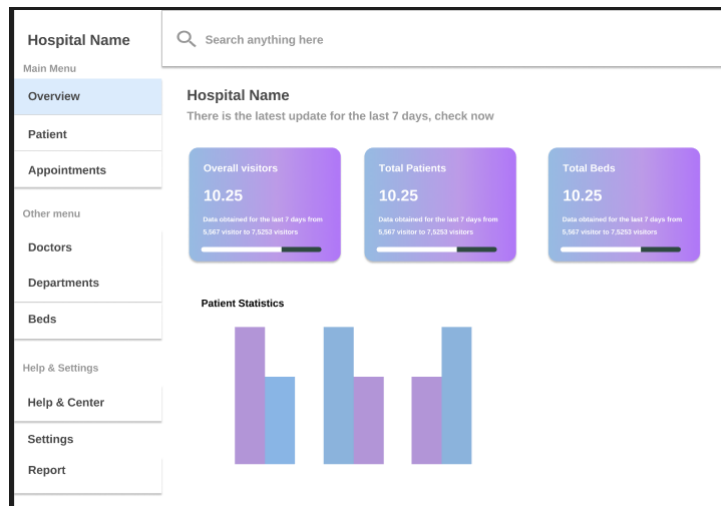


Figure 4.20: Hospital Dashboard – Homepage

4.8.21 Hospital Dashboard – Department’s Page

This is the Hospitals dashboard’s department page where all the departments are listed.

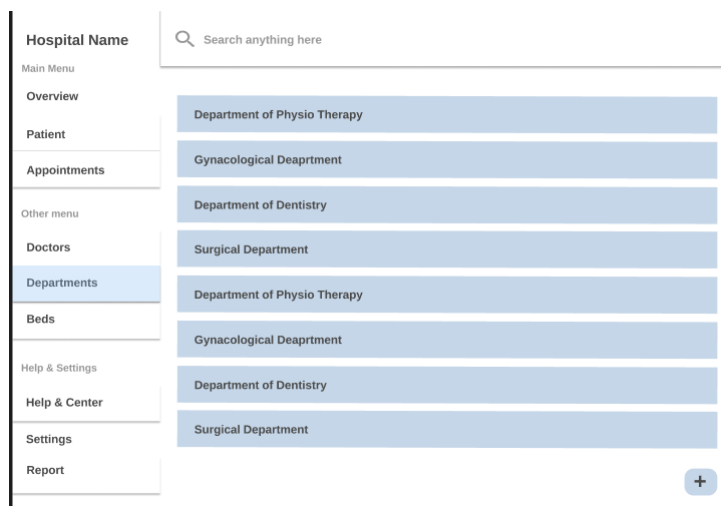


Figure 4.21: Hospital Dashboard – Department’s Page

4.8.22 Hospital Dashboard – Doctor’s Page

This is the Hospitals dashboard’s Doctors page where all the doctors working in the specific hospital are listed.

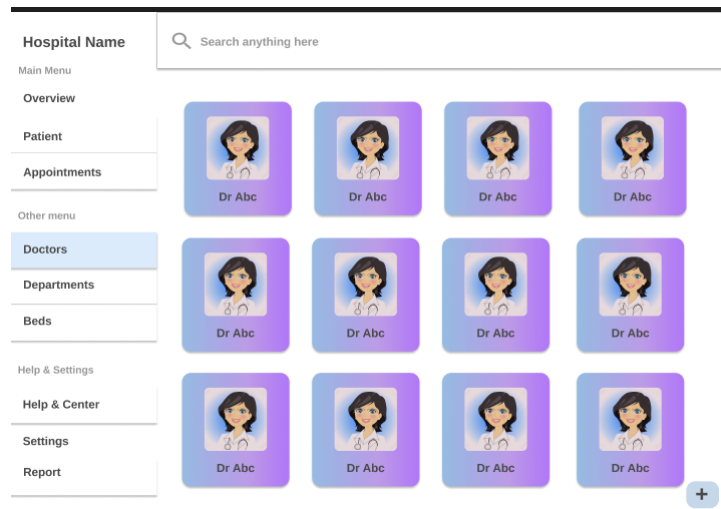


Figure 4.22: Hospital Dashboard – Doctor’s Page

4.9 Database Design

4.9.1 ER Diagram

The ER diagrams are as follows:

4.9.1.1 Admin Dashboard - ER Diagram

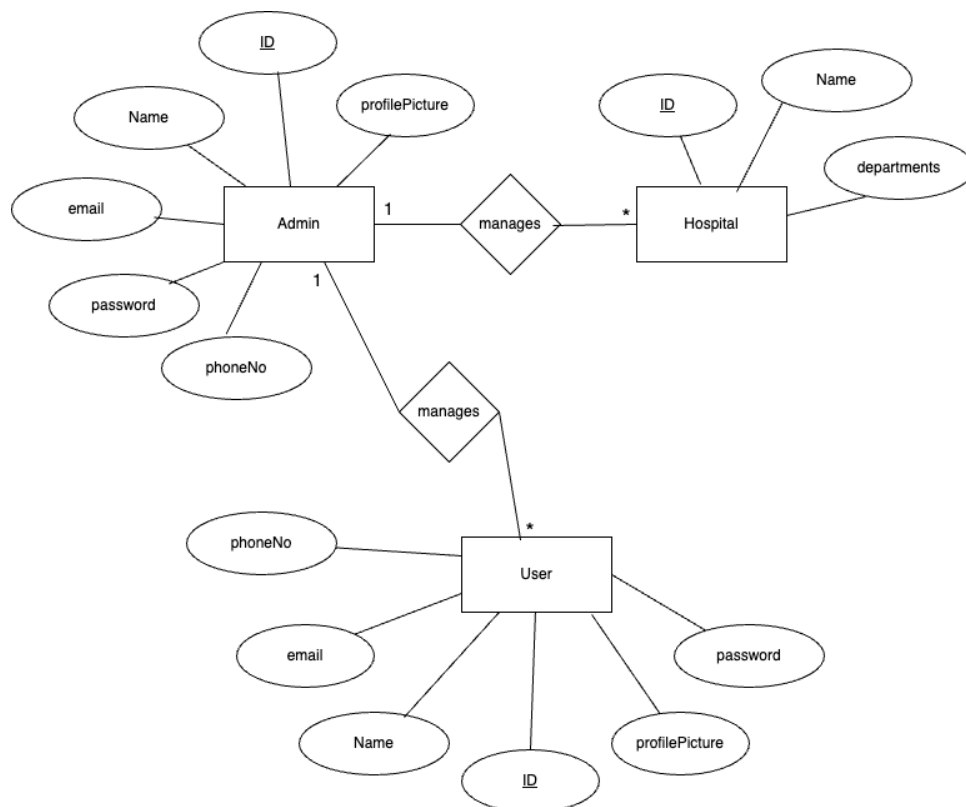


Figure 4.23: ER Diagram for Admin Dashboard

4.9.1.2 Hospital's Admin Dashboard - ER Diagram

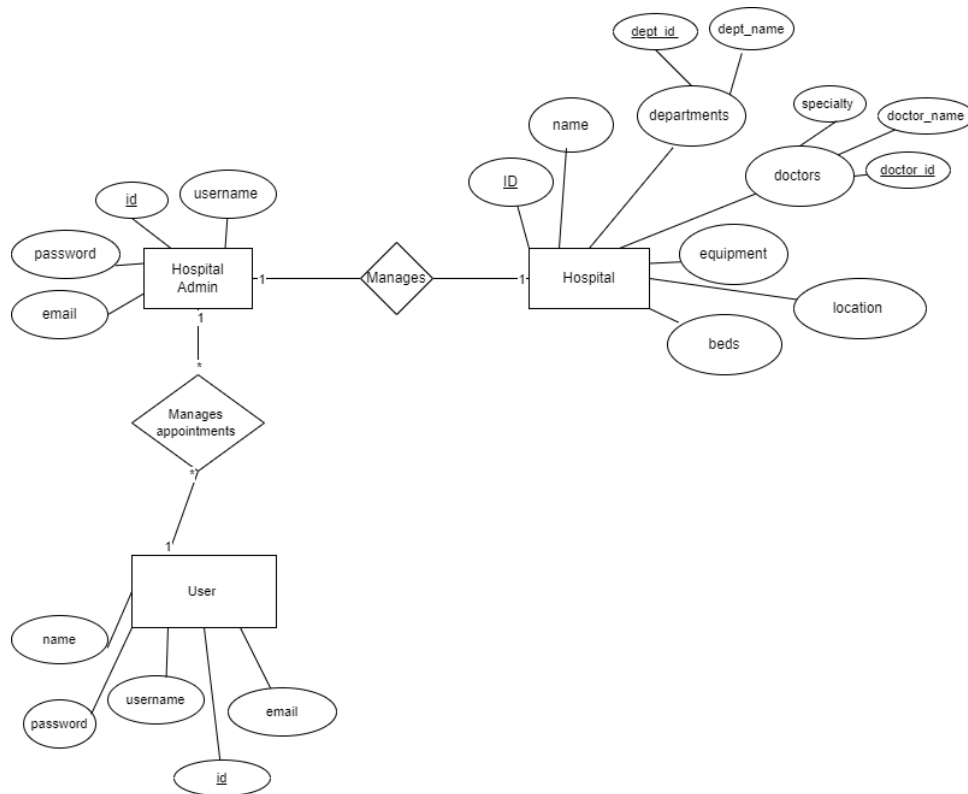


Figure 4.24: ER Diagram for Hospital's Admin Dashboard

4.9.1.3 Mobile Application - ER Diagram

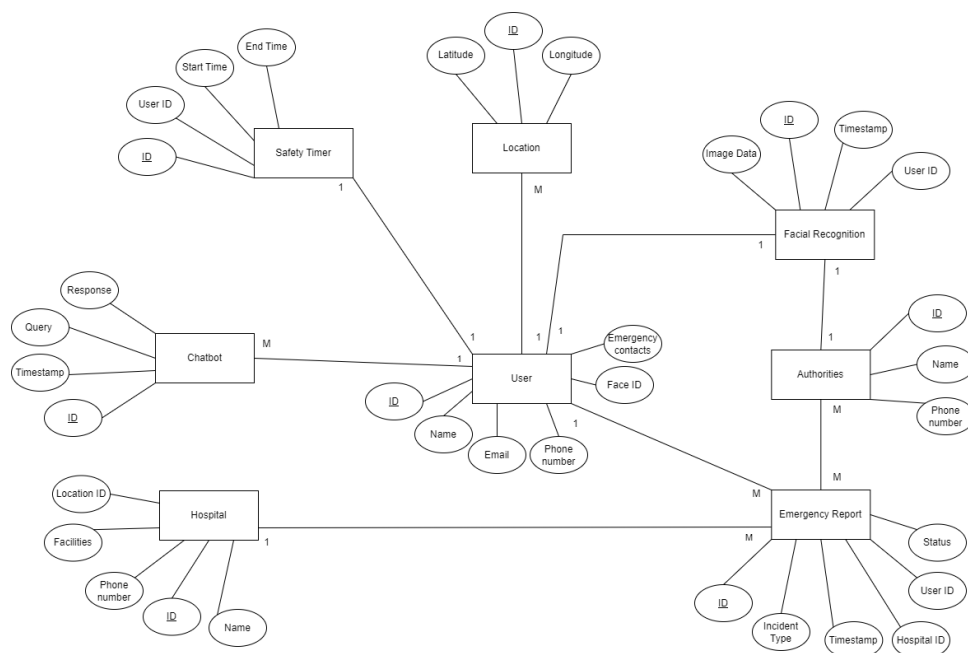


Figure 4.25: ER Diagram for Mobile Application

4.9.2 Data Dictionary

Entity	Attribute	Data Type	Description
Hospital Admin	id	Integer	Unique identifier for the hospital admin
Hospital Admin	username	Varchar	Username of the hospital admin
Hospital Admin	password	Varchar	Password of the admin account
Hospital Admin	email	Varchar	Email address of the hospital admin
Hospital	ID	Integer	Unique identifier for the hospital
Hospital	name	Varchar	Name of the hospital
Hospital	location	Varchar	Location of the hospital
Hospital	equipment	Varchar	Equipment available in the hospital
Hospital	beds	Integer	Number of beds in the hospital
Departments	dept_id	Integer	Department identifier
Departments	dept_name	Varchar	Name of the department
Doctors	doctor_id	Integer	Unique identifier for the doctor
Doctors	doctor_name	Varchar	Name of the doctor
User	id	Integer	Unique user identifier
User	name	Varchar	User's full name
User	username	Varchar	Username of the user
User	email	Varchar	Email address of the user
User	password	Varchar	User's account password
Safety Timer	ID	Integer	Unique identifier for the safety timer
Safety Timer	Start Time	Time	Start time of the safety timer
Safety Timer	End Time	Time	End time of the safety timer
Location	ID	Integer	Location identifier
Location	Latitude	Float	Latitude coordinates
Location	Longitude	Float	Longitude coordinates
Facial Recognition	ID	Integer	Facial recognition record ID
Facial Recognition	Image Data	Blob	Image data for facial recognition
Facial Recognition	Timestamp	Timestamp	Timestamp of the recognition
Authorities	ID	Integer	Authority identifier
Authorities	Name	Varchar	Name of the authority
Authorities	Phone Number	Varchar	Contact number of the authority
Emergency Report	ID	Integer	Report identifier
Emergency Report	Status	Varchar	Status of the emergency report
Emergency Report	Incident Type	Varchar	Type of incident reported
Emergency Report	Timestamp	Timestamp	Time when the incident was reported

Table 4.25: Data Dictionary Table

4.10 Risk Analysis

4.10.1 Technical Risks

4.10.1.1 System Downtime

If the application experiences downtime or technical failures, users will be unable to report emergencies which can potentially put their lives at risk.

4.10.1.2 Integration Issues

The application heavily relies on integration with third party services such as hospitals, emergency call centres, GPS tracking, ambulance services etc. In case, the administrators choose not to cooperate, it can lead to incomplete and limited functionality.

4.10.1.3 Data Security Breach

The application has the functionality to store a user's personal data, medical history and location. A data breach can compromise user's safety.

4.10.1.4 Scalability Issues

Due to the cost constraints, the app can cater to only a 100 users at a time. This can be problematic during mass emergencies when more than 100 users are simultaneously accessing the app. It can lead to unexpected crashes and performance degradation.

4.10.2 Business Risks

4.10.2.1 Regulatory Compliance

Considering the application is dependant on third party services and fetches data from various hospitals, ambulance and emergency services, government regulations regarding data handling must be strictly followed to prevent data theft and breaches. Non-compliance can cause legal repercussions.

4.10.2.2 Market Adoption Risk

The introduction to a comprehensive emergency application may be welcomed by many, but adapting and trusting a completely new technology with personal information when traditional emergency helplines exist may be a challenge users face.

4.10.2.3 Partnership Risks

The fundamental functionality of the app is reliant on information fetched from various third party services. The inability to reach a mutual consensus and agreement can reduce the efficiency of the application.

4.10.3 Operational Risks

4.10.3.1 Inaccurate Data

The application will use data provided by various hospitals, emergency and ambulance services. In instances, where the data provided is inaccurate, outdated, or does not align with real time data, the functionality of the app can be severely compromised.

4.10.4 Development Risks

4.10.4.1 High Maintenance Cost

The application requires GPS tracking, integration with Google Maps, call automation APIs, database management, and other third party services that become costly when usage scales up.

4.10.4.2 Time Constraint

The application along with the website will be developed by only three people, the comprehensive list of features span multiple domains, including Web Development, Computer Vision, App Development, Artificial Intelligence, Information Security and Database Management. The constraints on the time frame and budget can limit the functionality of the project altogether.

4.10.5 User Risks

4.10.5.1 User Misuse

There is a possibility that user's abuse the emergency features of the application. This can waste the time, energy and resources of the responders.

Chapter 5 High-Level and Low-Level Design

The system is designed to provide an extensive solution for emergency reporting and healthcare access through a mobile application for users and a website for administrative purposes .

5.1 System Overview

5.1.1 Client Side

As mentioned earlier , the client side interfaces are divided across a mobile application and a website.

5.1.1.1 Mobile Application

The mobile application caters to the general users and has all the fundamental features of the project. The application provides all essential features stated earlier , from emergency reporting to AI chatbot. This primary interface is an Android app , developed on android studio using kotlin as the backend and jetpack compose as the frontend. It enables users to report emergencies, receive first-aid guidance via an AI chatbot, and access real-time hospital data, such as the availability of beds, doctors, and equipment.

5.1.1.2 Website Portal

The administrative and hospital interface are web based platforms built using Express , React js and node. The website will contain a dashboard for admin and hospital staff to manage data , update information , provide real time data and handle appointments and requests .

5.1.2 Server Side

For both the mobile application and the website , a centralized database will be maintained using Fire-base. This centralized database will allow seamless integration across the application and website , handling all server side operations such as : authentication , real-time data updates and data synchronization between the application and website.

5.2 Design Considerations

When designing the emergency app and hospital dashboard system, several key issues need to be addressed to ensure the system's effectiveness and reliability.

5.2.1 Assumptions and Dependencies

The following sections describe the assumptions and dependencies regarding this project

5.2.1.1 Related Software

The mobile application part of the project will be made using android studio with Kotlin and Jetpack compose , while the website depends on React , Express and node js. Both the application and website rely on Firebase as the central database and real-time data management across platforms. For this project , we assume that the real-time data will be provided by the relevant hospitals and the system initially will be accessed through android devices .

5.2.1.2 Operating System

The mobile application is designed using android studio therefore, the operating system integrated will be the Android OS. Meanwhile, the website is accessible across platforms and integration with different operating systems (such as hospital systems) may be necessary.

5.2.1.3 End-Users

The end-user assumption regarding this project is that the general users will be familiar with smartphones but they might not have deep technical knowledge. Therefore, the mobile application interface will be kept as user friendly as possible. However, hospital staff may require minimal training to effectively update data through the website.

5.2.1.4 Functionality Changes

The future versions regarding this project , might need to scale to handle additional regions and services, requiring adjustments in infrastructure and APIs.

5.2.2 General Constraints

Some constraints and global limitations that might affect the system and its design are described below

5.2.2.1 Software Environment

The system relies on Firebase framework for the database storage and updates, possible storage or bandwidth limitation this could pose a design issue.

5.2.2.2 Security Requirements

Since the system handles sensitive information such as user medical history , user location, hospital data and resources data privacy is a major concern. Data Encryption is critical to ensure user data integrity

5.2.2.3 Interoperability Requirements

Our system integrates a mobile application and website , therefore , ensuring interoperability is a key constraint for the project. Hospitals may use different software or platforms for their internal operations, requiring the system to handle multiple data formats and protocols. This means developing flexible APIs and communication protocols that can adapt to different systems for seamless integration. Lack of interoperability could hinder real-time data updates and limit the system's ability to function effectively across different institutions.

5.2.2.4 Performance Requirements

The system must handle real-time updates from multiple hospital , it must also ensure no delays occur in contacting emergency Responders and accurately monitor user and responder location. Any performance issue of the system could pose critical threats to user during an emergency

5.2.2.5 Network Communications

Dependence on stable internet access for both the app and dashboard can be a constraint, particularly in regions with unreliable connectivity.

5.2.3 Goals and Guidelines

Some goals and guidelines that dominate the design of the system are described below

5.2.3.1 User-Centric Design

The mobile application of the system will be easy to use; it is important to note that people will be using this system in an emergency and will definitely not want to be overwhelmed with an interface of the application. The front end of the application will be designed keeping ease of use principles in mind.

5.2.3.2 Real Time Responsiveness

The system handles critical data and emergencies therefore, real-time data updates and synchronization are essential. Any downtime or delay could have critical consequences. Therefore, ensuring responsiveness and reliability is a major goal of the system design,

5.2.3.3 Scalability

The app should be salable to accommodate multiple hospitals and regions, ensuring it can handle increased traffic or demand.

5.2.3.4 Cross-Platform Compatibility

The system must be designed to function seamlessly across both the mobile app and web dashboard, ensuring interoperability between platforms.

5.2.4 Development Methods

The development of the system will follow the Agile Methodology, focusing on iterative and incremental delivery. The agile approach will ensure continuous feedback and modification of the software throughout the development process. This will be critical since the project may have evolving features. Furthermore, as mentioned earlier in this document, the mobile application concerning the system will be built using jetpack compose and Kotlin on android studio. The application software will focus on modularity, scalability, maintainability and user friendliness. However, the integrated website built for administrative purposes will be built using Express, react and node js. The website will focus on responsive design and ensure interoperability and scalability, Both the application and website will ensure compatibility with Firebase as the centralized database The TDD approach will be adopted for key modules such as data synchronization and API integration to ensure the reliability of critical features. CI tools will be used to automate testing and deployment.

5.3 System Architecture

The architecture of the system has a hybrid architecture since it is designed to manage both the mobile application and the website platform. The mobile app has all the features crucial to the project and will be targeted to the general public, whereas, the website hosts features necessary for administrative purposes such as managing and updating the data. Each subsystem plays a unique role, and they work together to ensure real-time data management and efficient emergency response handling.

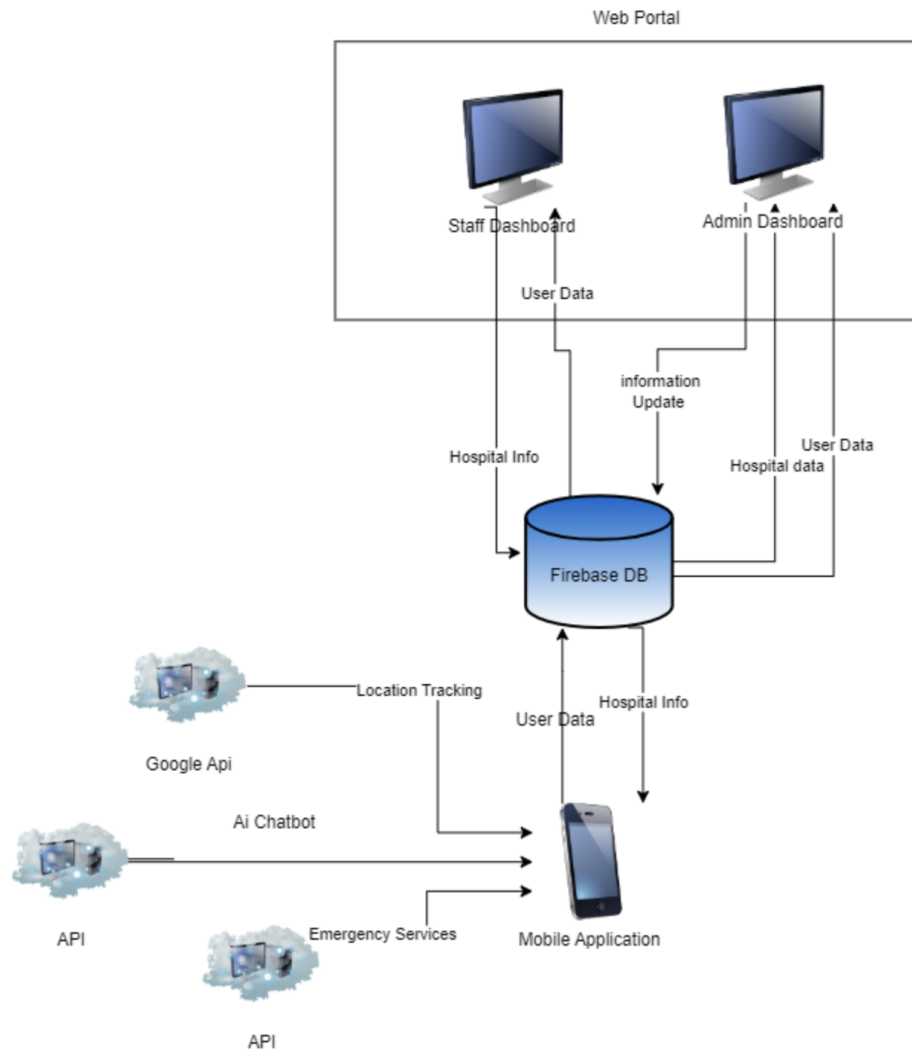


Figure 5.1: System Architecture design

5.3.1 Subsystem Architecture

The architecture of the system is divided into the following subsections

5.3.1.1 Mobile Application Module

The mobile application is built using kotlin and jetpack compose, it is the primary interface for our general users. The mobile application is the most vital part of the project as it hosts all intended features. It allows users to report emergencies, send location information, use AI chatbots for first aid advice, view real-time hospital data and all other proposed features. The application interacts with the database to store and retrieve data, it ensures users get accurate and updated information.

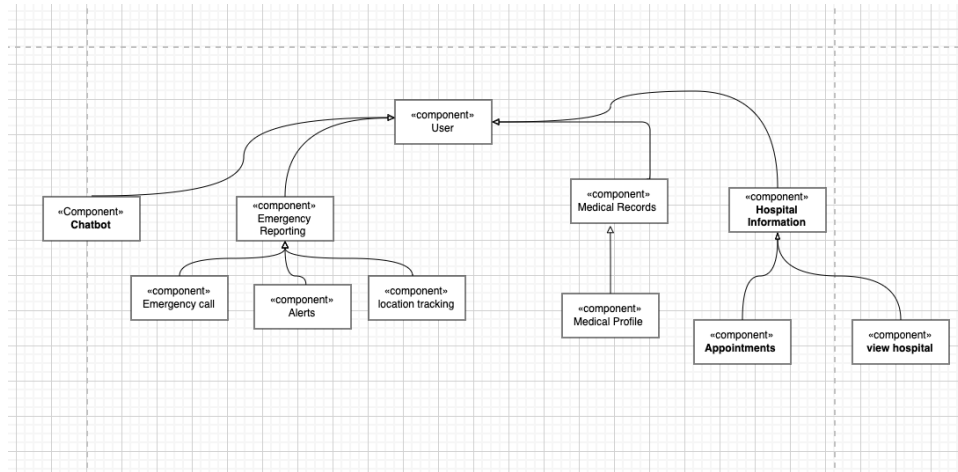


Figure 5.2: Subsystem component design for users

5.3.1.2 Website Platform

The website contains an admin dashboard and features relevant to uploading and managing data. This will be developed using part of the MERN stack mainly Express, React and Node JS. This serves as a control system for hospital staff and admins. Hospitals use the dashboard to update real-time data on the availability of beds, equipment, doctors, and other critical resources. Admins can manage overall system data, oversee emergencies, and ensure smooth operation by handling user inquiries and overseeing appointments. This subsystem also directly communicates with the centralized database to maintain and synchronize data across modules.

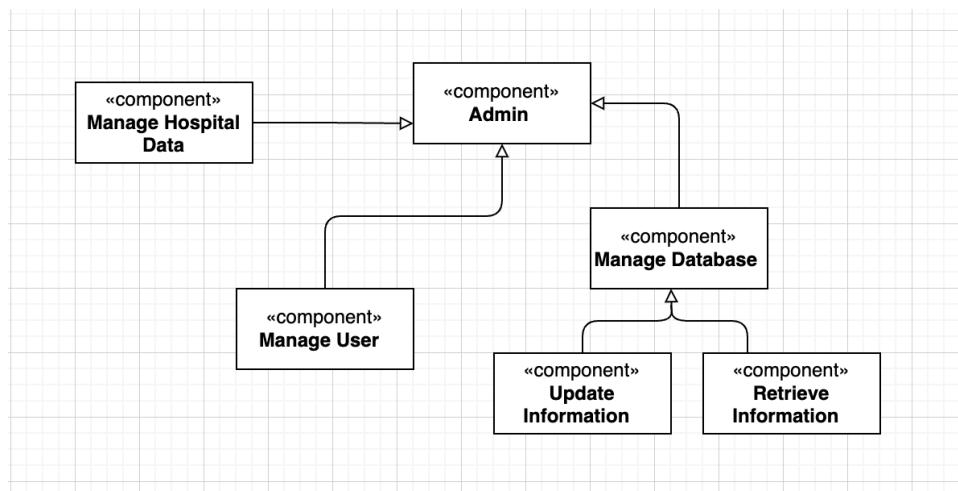


Figure 5.3: Subsystem component design for Admin

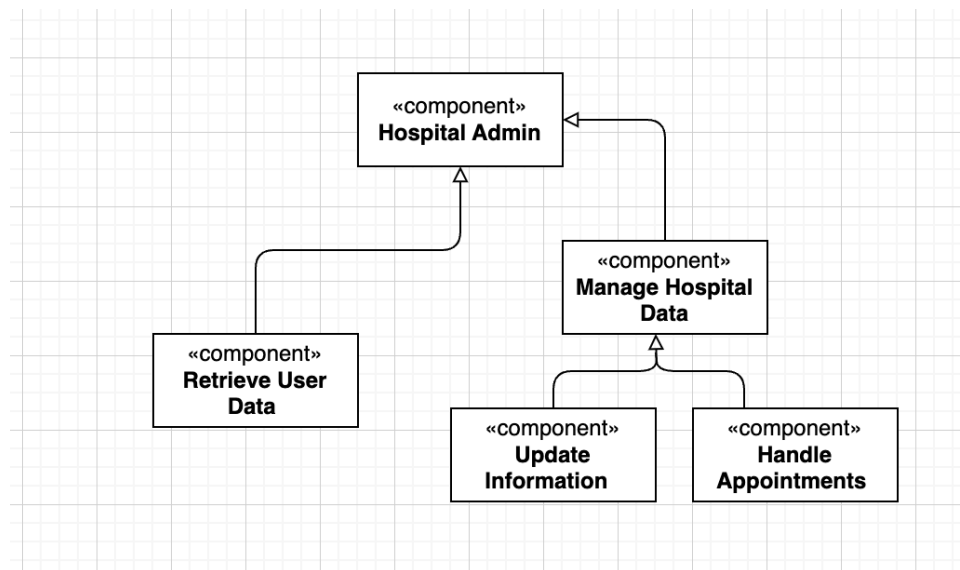


Figure 5.4: Subsystem component design for Hospital Admin

5.3.1.3 Central Database

The project has a singular DB that is centralized for access across the different modules. The database is set up using firebase , this acts as the central hub that stores and synchronizes data between the Android mobile app and the web dashboard. . Firebase also provides security, data storage, and APIs that allow seamless communication between the app and web-based components.

5.3.1.4 Subsystem Collaboration

The project requires seamless collaboration of the subsystems to achieve responsiveness and performance . Therefore , the system architecture provides seamless data flow and integration across platforms. The mobile application communicates with the central database through relevant firebase APIs , it stores and retrieves relevant data (such as user medical record, user location, hospital data etc) through the database itself. Similarly, on the other side the central database communicates with the website platform where hospital staff and admin monitor and manage the data. Hospital staff consistently update the availability of relevant resources through the website, these updates are pushed to the Firebase database. Since Firebase is a real-time database , the updates information is instantly available on users mobile application. Furthermore, the admins use the web platform to manage user data, system configuration and oversee the entire data flow and ensure the system is running smoothly.

5.4 Architectural Strategies

While designing the system various architectural strategies were kept in mind, following is the detail and the reason of choosing specific strategies

5.4.1 Programming Language and Frameworks

The languages for the mobile application are kotlin and jetpack compose. Kotlin language was adopted as the main implementing language for the android app since it is modern and relatively easy to use. Kotlin also provides efficient and effortless frameworks for APIs and integration with google services (such as google maps apis). Meanwhile, jetpack compose was chosen for developing the frontend as its UI toolkit requires lesser boilerplate code and allows smoother UI updates and easier management of changes. Due to this strategy we can ensure the application can not only be maintained efficiently but also allow future extensions without having to drastically change existing code.

On the other hand, the languages chosen for the web platform are Express, React and Node js (part of the MERN stack). This framework was chosen because of its robustness in handling data, robustness in handling data, its flexibility in building real-time applications, and the strong developer ecosystem around React.js for front-end development. The stack's non-blocking architecture is ideal for the admin dashboard, ensuring real-time updates and smooth handling of user queries. Future scalability was also a key consideration in this choice, allowing for potential expansion in terms of both user base and system complexity.

5.4.2 Database Storage

For the centralized database, Firebase was chosen due to its capabilities of handling real-time databases; this feature is crucial for handling data during emergencies and ensuring synchronization between the application and website. . Firebase also simplifies authentication and secure access, aligning with the project's goals of providing a seamless and secure platform. While other database systems like Sql Server were considered, Firebase's cross-platform support made it the best choice for integrating both the web and mobile interfaces effortlessly.

5.4.3 User Interface Paradigms

The mobile app and web dashboard were designed to prioritize usability, accessibility, and minimalism. The mobile app focuses on swift navigation and intuitive actions, critical in emergency scenarios, while the dashboard caters to hospital staff with detailed data management features.

5.5 Domain Model/Class Diagram

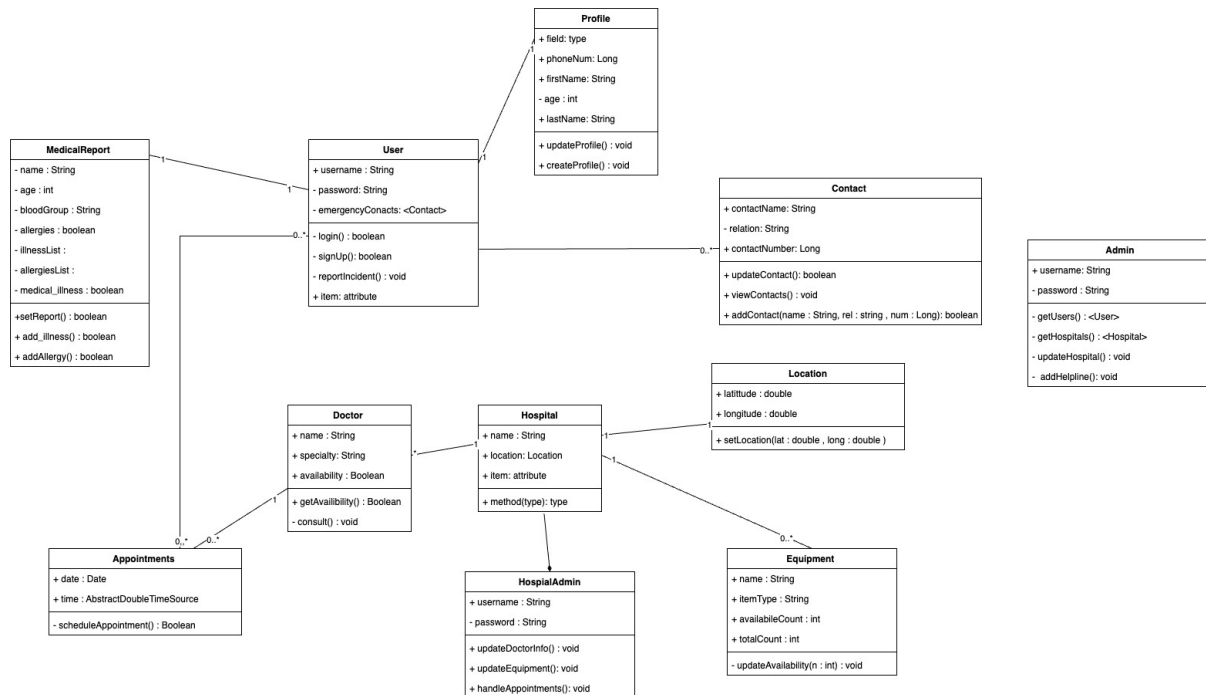


Figure 5.5: Class Diagram for our application

5.6 Policies and Tactics

To ensure a consistent , maintainable and efficient design, the following policies and tactics were followed throughout the system’s development.

5.6.1 Product Choices

- Android studio was chosen as the development IDE for the mobile application, the languages used were kotlin and jetpack compose.
- The website platform is developed using React for front end interface, Node and Express js for server execution. Visual Studio was picked as the IDE for developing the website due to its support for a range of frameworks , languages and integrated GIT support
- Firebase was chosen as the central database , since it is a real-time database and provides built in features such as authentication and data synchronization. Firebase also allows seamless integration with android studio and web applications.

5.6.2 Engineering Tradeoffs

For the development of mobile application Kotlin with jetpack compose was chosen over the traditional java and XML due to its modern and declarative approach to UI development , it makes it easier to build complex layouts with lesser code. However , in doing so one trade off is that jetpack compose is relatively new and has fewer resources and community support as compared to Java with XML. XML offers a well-established, stable environment with broader library support, but at the cost of needing more code and slower UI development .

Firebase was chosen as the central database over other databases such as PostgreSQL and MySQL due to its real-time capabilities , ease of use and seamless integration.. However, this means a trade-off limiting complex querying and scaling.

5.6.3 Design Patterns

The website platforms will be built using the Model View Controller (MVC) design pattern in mind. The react js will handle frontend separately, while server and database communication will be handled by Express and Node js. This promotes scalability and modularity in the system design.

Since the database is centralized for both the mobile and web platform, the repository pattern is used to abstract the data layer.

Bibliography

- [1] "Rescue 1122 - punjab emergency service." Available: <https://www.rescue.gov.pk/> [Accessed October 04, 2024].
- [2] "Women safety app - by punjab police." Available: <https://play.google.com/store/apps/details?id=com.pasca.ppic3.womensafety&hl=en> [Accessed October 04, 2024].
- [3] "Edhi foundation - serving humanity in the spirit of all regions." Available: <https://edhi.org/> [Accessed October 04, 2024].
- [4] "Punjab police app - by punjab it board." Available: <https://apkpure.com/punjab-police-pakistan/pk.pitb.gov.ppcma> [Accessed October 04, 2024].
- [5] "Pak life saver - by punjab it board." Available: <https://play.google.com/store/apps/details?id=com.PITB.RVMS.CPR&hl=en> [Accessed October 04, 2024].
- [6] "Pak ndma - disaster alert application." Available: <https://apps.apple.com/ae/app/pak-ndma-disaster-alert/id6503601052> [Accessed October 04, 2024].
- [7] "Psca - punjab safe cities authority." Available: <https://psca.gop.pk/> [Accessed October 04, 2024].