Cairo University

Faculty of Computers and Artificial Intelligence

Department of Computer Sciences

Monqez Application

Supervised by

Dr. Mohamed El-Ramly

TA. Sarah El-Nady

Implemented by

20170093	Hussien Ashraf Abdul-Hamid
20170072	Ehab Fawzy Ibrahim
20170085	Hatem Mamdoh Saed
20170098	Khaled Ezzat Ahmed

Graduation Project

Academic Year 2020-2021

Final Document

Acknowledgement

First and foremost, we would like to thank Allah for His grace and for giving us the strength to complete this work. Secondly, we would like to express our sincere gratitude to our supervisor Dr. Mohamed El-Ramly and TA Eng. Sarah El-Nady for their continuous support during our graduation project researches and implementation, for their patience, motivation, enthusiasm, and immense knowledge. Their guidance helped us during all phases of the graduation project. Finally, we would like to thank our families and friends for all the support and encouragement during our graduation project as without them, this work would not have been possible.

Abstract

First aid is vital for saving lives and averting the victim's condition from deteriorating. That's why the world could use a fast and easy method of communication connecting people with paramedics when emergencies occur. Some websites and applications exist that offer such functionality but unfortunately, they have their limitations. So, we propose developing an application, "Monqez", that aims to help people in need of first aid and connect users requesting help to the nearest available paramedics. Several tools and techniques such as video conferencing and location tracking will be used to build this application and enable its functionalities. In this document, we will discuss the features, tools, design, and implementation aspects of Monqez as well as our concluding remarks.

Table of Contents

List of Figures

List of Tables

List of Acronyms & Abbreviations

API	Stands for **A**pplication **P**rogramming **I**nterface which is a computing interface that defines interactions between multiple software intermediaries.
App	Stands for **App**lication.
CPR	Stands for **C**ardio **P**ulmonary **R**esuscitation.
EMS	Stands for **E**mergency **M**edical **S**ervices.
ERD	An **E**ntity **R**elationship **D**iagram, also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts, or events within an information technology system.
ETA	Stands for **E**stimated **T**ime of **A**rrival.
GPS	**G**lobal **P**ositioning **S**ystem is a global navigation satellite system that provides location, velocity, and time synchronization.
GUI	The **G**raphical **U**ser **I**nterface is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicator such as primary notation.
IDE	Stands for **I**ntegrated **D**evelopment **E**nvironment which is a software application that provides comprehensive facilities to computer programmers for software development.
Monqez	The Arabic word for “savior” who, in our system, is a paramedic / helper that arrives to help the person calling for help.
MVC	MVC is an architectural pattern which stands for **M**odel, **V**iew, and **C**ontroller. MVC separates an application into three components - Model, View, and Controller.
SDK	A **S**oftware **D**evelopment **K**it is a set of software development tools that allows the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platform

Chapter 1: Introduction

1.1. Background

Several people die when involved in an accident because they don't get the first aid they need in time. For many years, people tried to solve this problem by calling an ambulance or, if the person in need of first aid is lucky, there might be a person with specialized first aid skills nearby. Lately, some websites and applications have been developed to inform nearby paramedics of emergencies but unfortunately, they only handle specific medical emergencies or have other limitations. Our approach for solving this problem is having a mobile application that, among other features, allows a video or voice call with a qualified specialist "Monqez" or requests a nearby specialist to help onsite and provide the needed first aid.

To help connect the person in need of first aid with the Monqez, we will need to use video conferencing technology which uses the camera in a smart phone. This technology needs about 2 mega bit per second and requires a stable internet connection. Moreover, location tracking is used when the Monqez is attempting to reach the victim as it helps him/her find the target's location faster and it helps the user request an onsite Monqez faster as well. The location tracking can be implemented as long as the smartphone has a Global Positioning System "GPS".

1.2. Motivation

On April 12, 2010, The Guardian stated that over 150,000 deaths could have been saved if the right first aid was applied to them [1]. Moreover, the University of Manchester reported that 59% of deaths that occurred before arriving to the hospital could have been saved [2]. With the rapid advances in mobile app development, a few applications were developed that aim at handling emergencies that require medical attention. However, these applications have their shortcomings and rely on hospitals to save the victims rather than offer on-site first aid.

1.3. Problem Definition

Accidents occur every day all around the world and a lot of people fall victim to them. It could take some time for an ambulance to arrive even though every minute is crucial to the victim. That's why first aid is vital for saving lives! Although 93% of people will call for an ambulance if they find someone with an injury, first aid intervention of any kind is infrequent [2]. This partly because many people lack the basic first aid knowledge and partly because sometimes the situation requires a trained professional to handle the injuries. So there lies the problem; how can we help people who just witnessed an accident save the victim's life?

It's unquestionable that we need a fast and easy method of communication with paramedics and what better way to have this communication than to have a mobile app that notifies paramedics and connects us with them through voice and video calls? Some applications that provide EMS exist and have gained popularity. However, not all these applications are cross platform or free. Furthermore, they don't notify the nearest first-aid-certified person to the accident's location or offer video calls with paramedics or just provide contact with a first aid chatbot. Even the applications that do exist offering these features target certain emergencies only (CPR emergencies only for instance).

1.4. Suggested Solution

We propose developing a cross platform mobile application called “Monqez” which helps people who need first aid by informing the closest available paramedics to the patient’s location in order to help the patient and save his/her life.

The main goal of Monqez is to provide quick help to the person in need until an ambulance arrives. In order to achieve this goal, the app must be able to connect the user requesting help to the nearest available paramedic “Monqez” as well as provide some basic first aid tips either via pictures or voice and video calls.

Of course, all registered paramedics must be certified and need to have passed a first aid course from accredited institutions.

1.5. Project Time Plan

The following figures show the time plan we have followed / will follow to develop Monqez application:

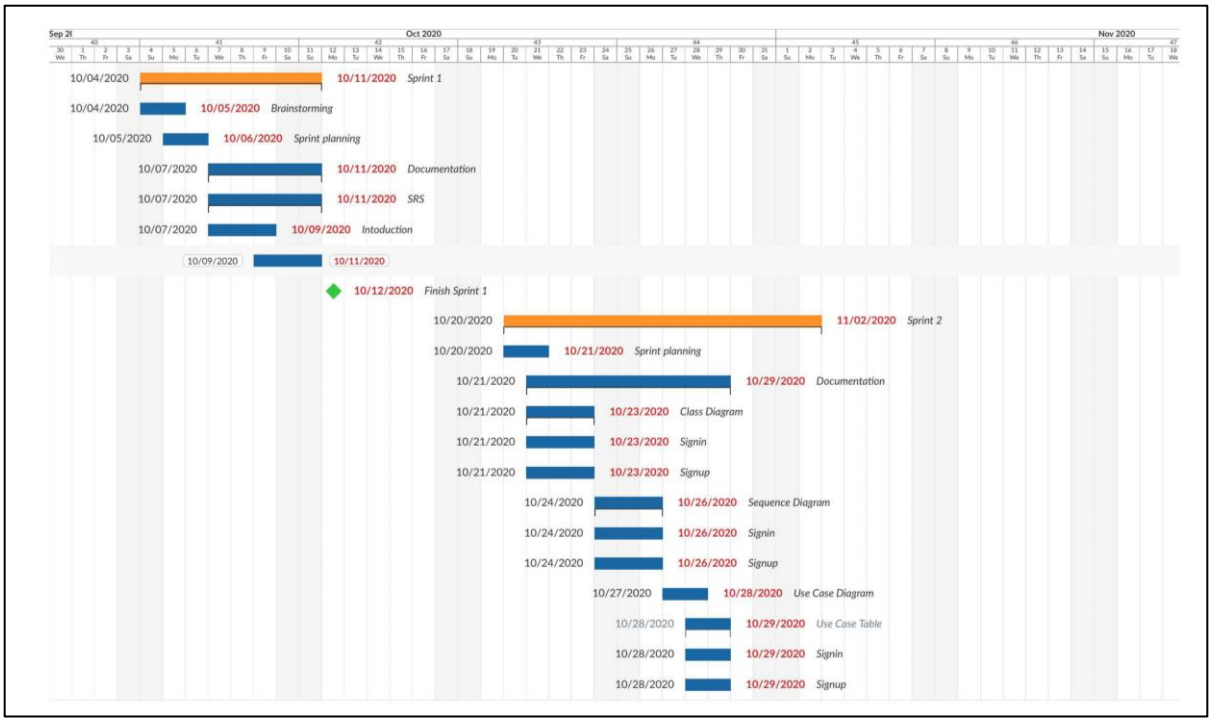

Figure 1: Gantt chart from 4th of October to 29th of October

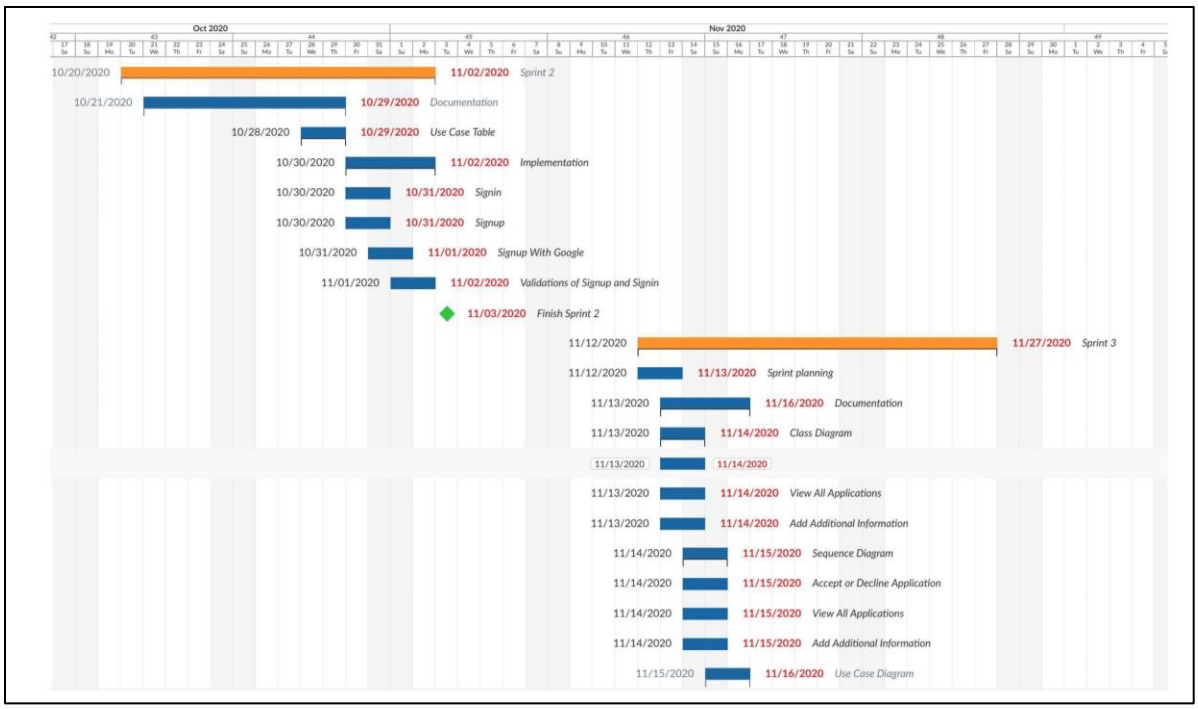

Figure 2: Gantt chart from 29th of October to 16th of November

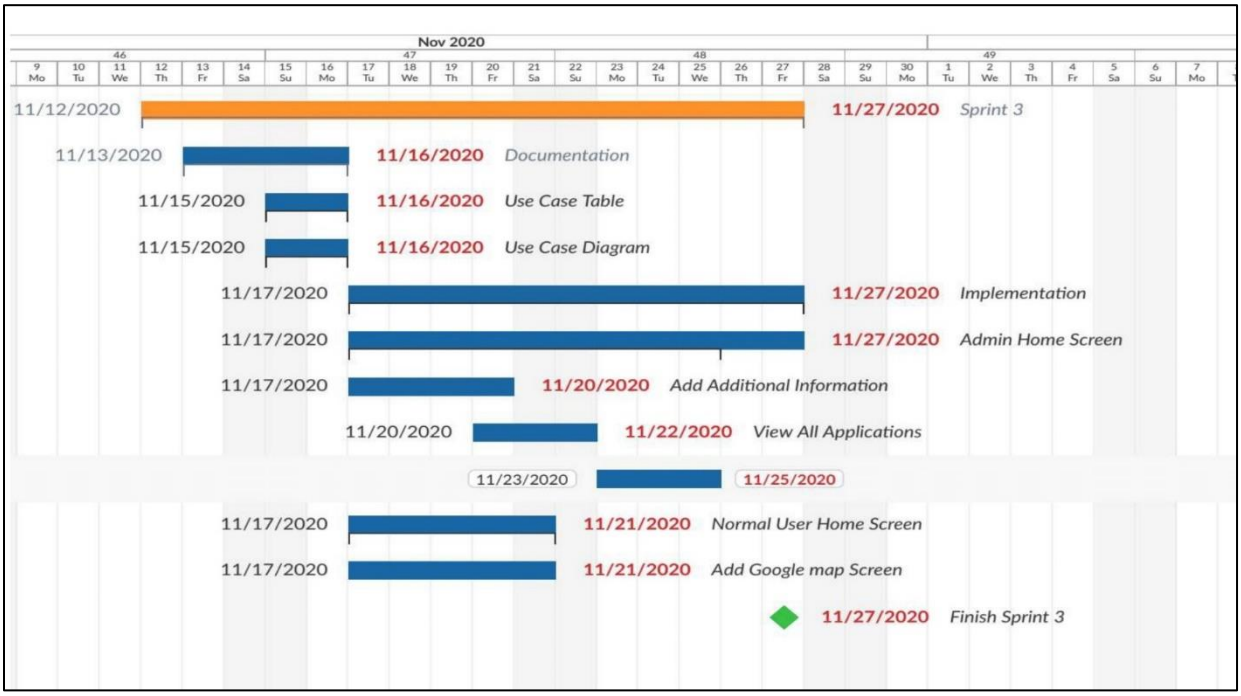

Figure 3: Gantt chart from 16th of November to 27th of November

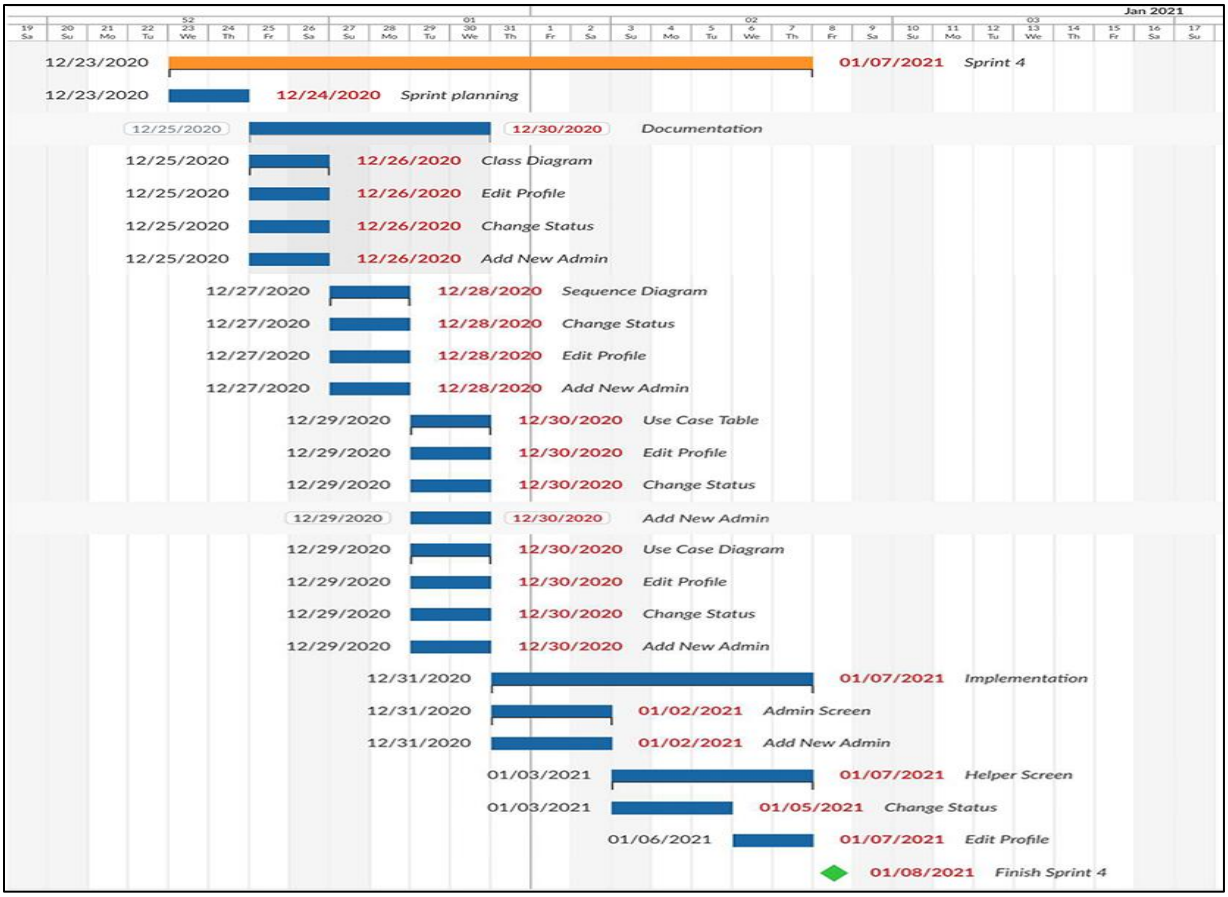

Figure 4: Gantt chart from 23rd of December to 8th of January

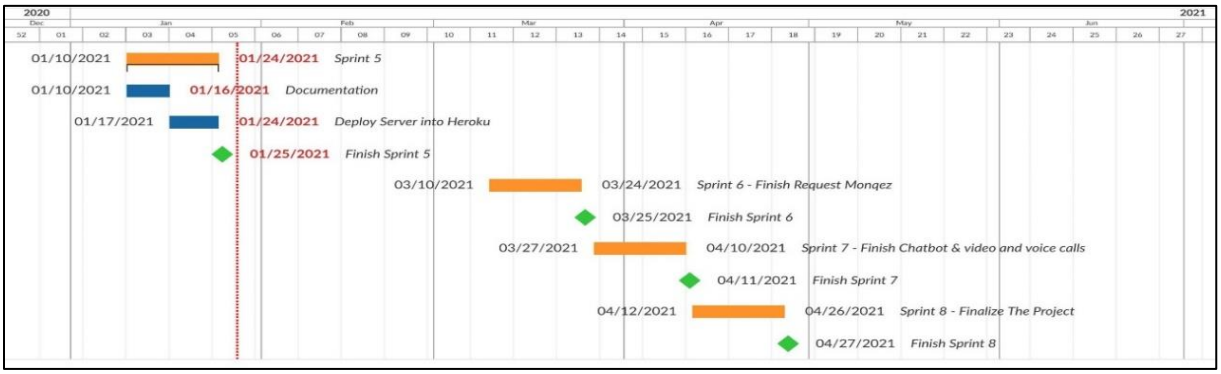

Figure 5: Gantt chart from 10th of January to 27th of April

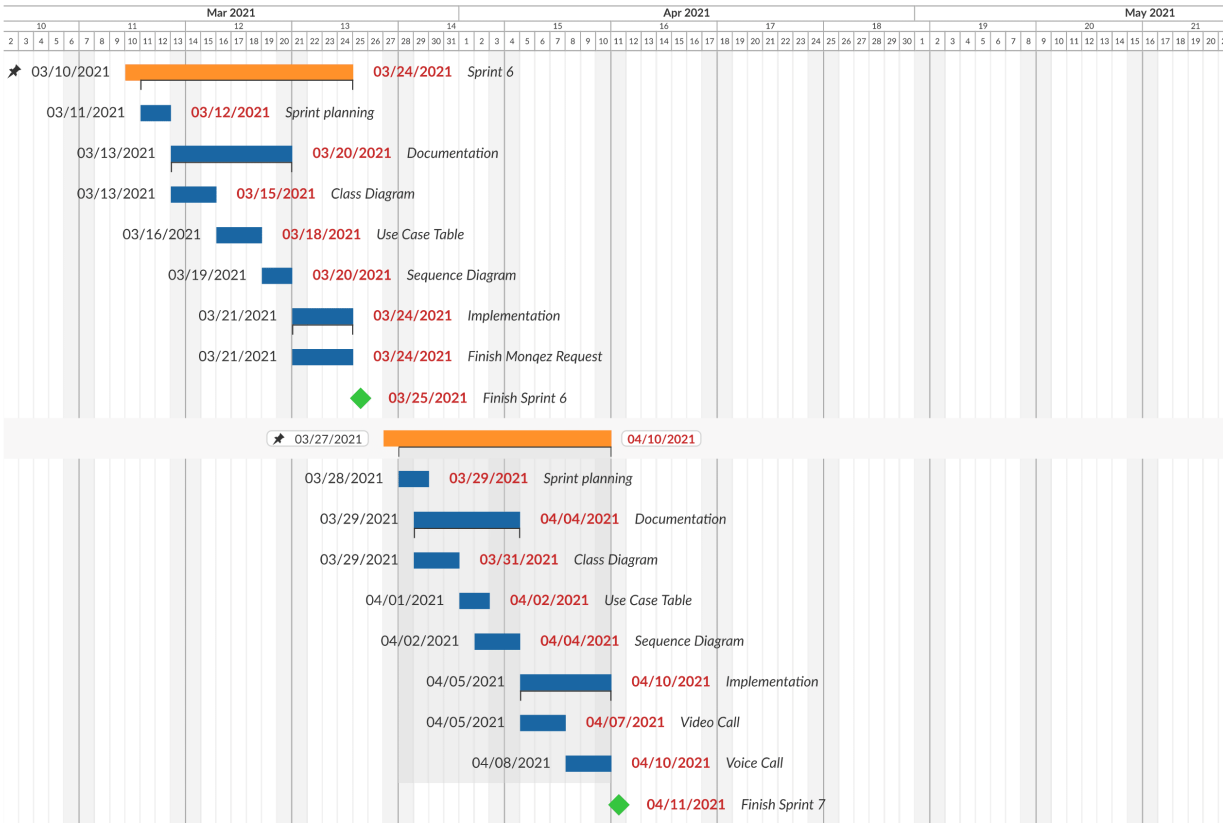

Figure 6: Gantt chart from 10th of March to 11th of April

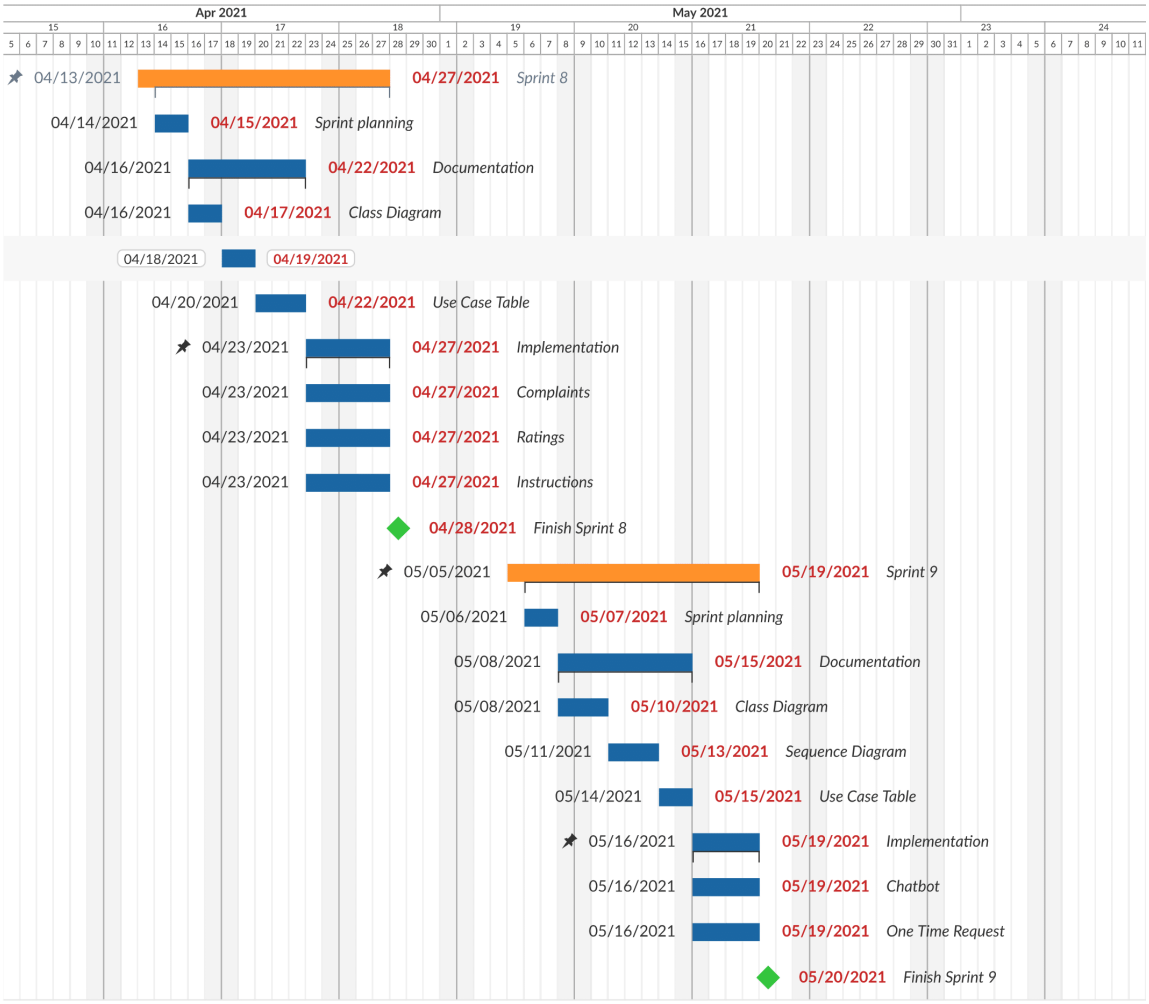

Figure 7: Gantt chart from 13th of April to 20th of May

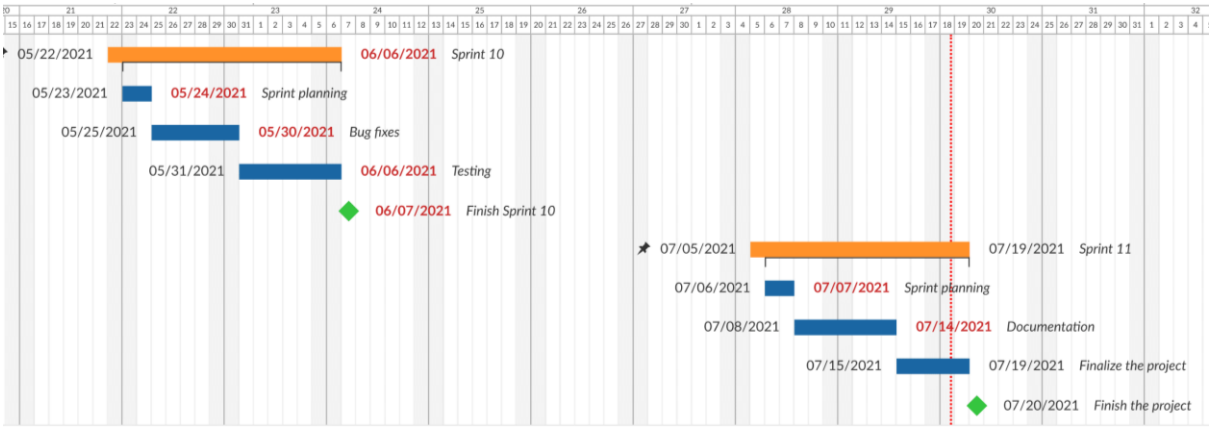

Figure 8: Gantt chart from 22th of May to 20th of July.

1.6. Project Development Methodology

For this project, we decided to follow the Agile Software Development method. “Agile” is a term used to describe approaches of software development emphasizing incremental delivery, team collaboration, continual planning, and continual learning instead of trying to deliver the project all at once near the end.

Agile development focuses on keeping the process lean and creating minimum viable products that go through a number of iterations before anything is finalized. Feedback is also gathered and implemented continually, so overall, it is a much more dynamic process where everyone is working together towards one goal.

Among the agile frameworks is scrum. Scrum breaks down the development phases into stages or cycles called sprints. The development time for each sprint is maximized and dedicated, thereby managing one sprint at a time. It focuses on continuous deliverables.

So, why did we use scrum for this project? There are several reasons that lead us to that decision including:

1. Scrum lets the team self-organize.
2. Scrum work is done simultaneously rather than sequentially. Everything is flexible and changeable during the life of the project and even after.
3. Scrum is more suitable for small teams, makes all the team members work cooperatively resulting in faster code completion time as well as better quality.
4. Scrum lets the team to prioritize the tasks by the order of importance, which means that tasks to be completed first will probably affect return on investment the most.

1.7. Tools

In this section, we will discuss the tools and technologies that we will be using to develop Monqez application.

- **Node.js:** It is an open source, cross-platform runtime environment for developing ***server-side*** and networking applications. Node.js applications are written in JavaScript and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux. Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.
- **Dart Programming Language:** Dart is a client-optimized programming language for apps on multiple platforms. It was developed by Google and is used to build mobile, desktop, server, and web applications. Dart is an object-oriented, class-based, garbage-collected language with C-style syntax. It can compile to either native code or JavaScript and it supports interfaces, abstract classes, reified generics, and type inference.
- **Flutter:** It is a free and open-source mobile UI framework created by Google and released in May 2017. In a nutshell, it allows us to create a native mobile application with only one codebase. This means that we can use one programming language and one codebase to create two different apps (for iOS and Android). Flutter uses Dart programming language and consists of two main parts:
    1. An SDK (Software Development Kit) which is a collection of tools that helps us develop applications. This includes tools to compile code into native machine code (code for iOS and Android).

2. A framework (UI Library based on widgets) which is a collection of reusable UI elements (buttons, text inputs, sliders, and so on) that can be personalized to one's own needs/liking.

- **Firebase:** It is a Backend-as-a-Service (Baas) that provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google's infrastructure. Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents.
- **Heroku:** It is a cloud platform as a service (PaaS) supporting several programming languages. It has features that allow a developer to build, run and scale applications across most languages.
- **Google Maps Platform:** The Google Maps Platform is a set of APIs and SDKs that allows developers to embed Google Maps into their mobile apps and web pages, or to retrieve data from Google Maps.
- **Agora Platform:** Agora is a platform as a service which provides the SDKs and building blocks to enable a wide range of real-time engagement possibilities. It has voice and video calls SDK which lets you integrate high-quality, stutter-free interactive voice chat into your app and makes it easy for you to add new features like voice effects and 360-degree surround sound, noise cancellation, and active-speaker recognition.
- **Dialogflow:** Dialogflow is a natural language understanding platform by Google used to design and integrate a conversational user interface into mobile apps, web applications, devices, bots, interactive voice response systems and related uses. It uses rule-based grammar matching as well as machine learning matching.
- **Postman:** Postman is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so you can create better APIs – faster.

1.8. Report Organization

The next chapters will cover the design and implementation aspects of Monqez. First, chapter 2 will provide an overview of some similar applications. Then, chapters 3 and 4 will dive into the design of Monqez application. Chapter 3 details all the functional and quality requirements of Monqez while chapter 4 contains specific information about the expected input, output, classes and functions. In addition, the interactions between classes to meet the desired requirements will be outlined in several figures in chapter 4. Chapter 5 will talk about how the application was implemented and tested. Finally, chapter 6 will include a discussion of what has been achieved so far as well as some concluding remarks.

The intended reader groups for this document are the project manager, developers, supervising professor, and any person interested in developing an application like Monqez.

Chapter 2: Related Works

As we mentioned before, a few applications exist that aim at handling emergencies which require medical attention. Each one of these apps offers a variety of features. We surveyed these apps and determined their advantages and disadvantages from a user's perspective. In the next few paragraphs, we will discuss some of these apps in more detail.

- **New Zealand Red Cross First Aid [3]:** This web application contains courses and step-by-step instructions for many emergencies including CPR, burns, heart attacks, open wounds and more. However, it doesn't send medic to help the victim right away; it just relies on normal users to understand and follow the first aid instructions shown.
- **E-Medic [4]:** This application allows receiving any new communication to report an accident as well as accepting or canceling the report, drawing the best paths to the location of the incident and taking notes. In addition, it allows sending any amendments to the report to the operating room so that the operating room is aware of the emergency teams and following them on an ongoing basis. The application also allows emergency teams to communicate with hospitals and choose the most appropriate hospital for the case. The only drawback we found is that it does not send a paramedic or help the user perform the first aid right away; it just sends an ambulance to take the victim to the hospital.
- **eMedic (for iPhones) [5]:** It is an EMS app for iPhones. Used by more than 20,000 paramedics, EMTs, and nurses, eMedic provides quick access to basic anatomical illustrations, medical calculators, emergency quick reference cards, ACLS algorithms, popular medicines, and medical acronyms. Unfortunately, eMedic is not a free app and it only targets users with a strong medical background.
- **PulsePoint Respond [6]:** It is a 911-connected app that can immediately inform users of emergencies occurring in their community and can request their help when CPR is needed nearby. This app helps create an informed and engaged community that drives a "Culture of Action," a key strategy in strengthening the Chain of Survival for cardiac arrest victims. In addition to nearby "CPR-needed" notifications, the user can choose to be notified of significant events that may impact him/her and his/her family.
- **Asifny - إسعفني [7]:** It is an application developed by Saudi Red Crescent Authority and includes multiple features such as opening an emergency communication with the Saudi Red Crescent Authority and increasing the accuracy of the location, sending urgent distress in the extreme emergencies of both the Red Crescent and people close to you by SMS service and recording the details of medical history, diseases and medicines to understand the patient's condition as soon as possible. The app also informs the user of close medical facilities with directions and charts the route on the map of the facility you want to go to. This application sends an ambulance - not any nearby person with a first aid certificate - to the patient in order to save his life and this makes the process of reaching the patient slower.
- **Inaa'sh [8]:** It is a combined first aid and CPR Accreditation Program made especially for government and commercial organizations as well as community at large. It provides an up to date first aid and CPR knowledge and skills information through certified courses and provides the required equipment for organizations which helps perform CPR and first aid procedures during emergencies. It doesn't send a paramedic to the person in need of help nor does it provide online help that the user can contact.

Chapter 3: System Analysis

3.1. Stakeholders

This project's stake holders are the team developing the application, the administrators of the application, the certified paramedics that will be registered in our system as well as any user using the application.

3.2. Project Specification

3.2.1. Functional Requirements

The following are the functional requirements that the system should be able to do:

1. Allow the user to create a normal or Monqez account in the database.
2. Log the user into his/her account and allow the user to edit his/her information and to log out.
3. Show the first aid instructions for any type of injury.
4. Allow the normal user to order a Monqez for himself or for someone else in need of first aid.
5. Notify 3 nearby Monqez users when a first aid request is received.
6. Allow helper user to accept or decline the request.
7. Connect the normal user to a Monqez so that the Monqez shall arrive to the emergency site.
8. Provide video and voice calls between a normal user and any available Monqez.
9. Allow the normal user to make complaints about any incident that happened during the process of fulfilling a request.
10. Allow the normal user to rate the Monqez that has fulfilled his request.
11. Allow an admin user to add another admin.
12. Allow an admin user to either approve or deny a Monqez application.
13. Show the complaints about a specific request to an admin.
14. Show the Monqez applications and their certificates to an admin.
15. Allow an admin to ban any account he/she finds inappropriate.
16. Add points to a Monqez when he completes a request. These points can be redeemed when a sponsor provides pointing system.
17. Allow a normal user to chat with chatbot that helps him to use any functionality in Monqez application.

3.2.2. Non-functional Requirements

The following are the quality requirements that the system should be able to achieve:

1. **Performance:** Minimal searching time for nearby Monqez since the duration is an important factor to save the patients' lives.
2. **Safety:** During the registration process, the user is asked to attach his/her national ID to be used in the event of any illegal or unethical activity.
3. **Reliability:** The system should be able to work with full functionality in worst cases (peak demand).
4. **Availability:** The application should always be available to serve the users (at least 95% of the time). The maximum tolerated down time is 72 minutes per day and the meantime between failure and recovery must not exceed 30 minutes.
5. **Security:** All user information must be encrypted to protect it in the event of any breach. The system will protect the data and services from unauthorized access.
6. **Portability**: This software should be usable on different environments such as iOS, web and Android.
7. **Usability:** The application should be very easy to use with a user-friendly interface. It should allow the user to do all the basic features by 5 taps as a maximum.
8. **Robustness:** The system should be robust and handle invalid user input (invalid emails, passwords, etc.) without crashing.
9. **Fault tolerance:** The system should continue to operate properly even if a service fails.

Regarding the application platform, the client can use this software on different environments such as (iOS 10.0+ and Android 7.0+). Also, 4G connection needs to be established from the user's mobile for stable video and voice call.

The server-side components of the software system must operate within a Windows operating system environment running Node.JS.

3.3. Use Case Diagram and Use Cases

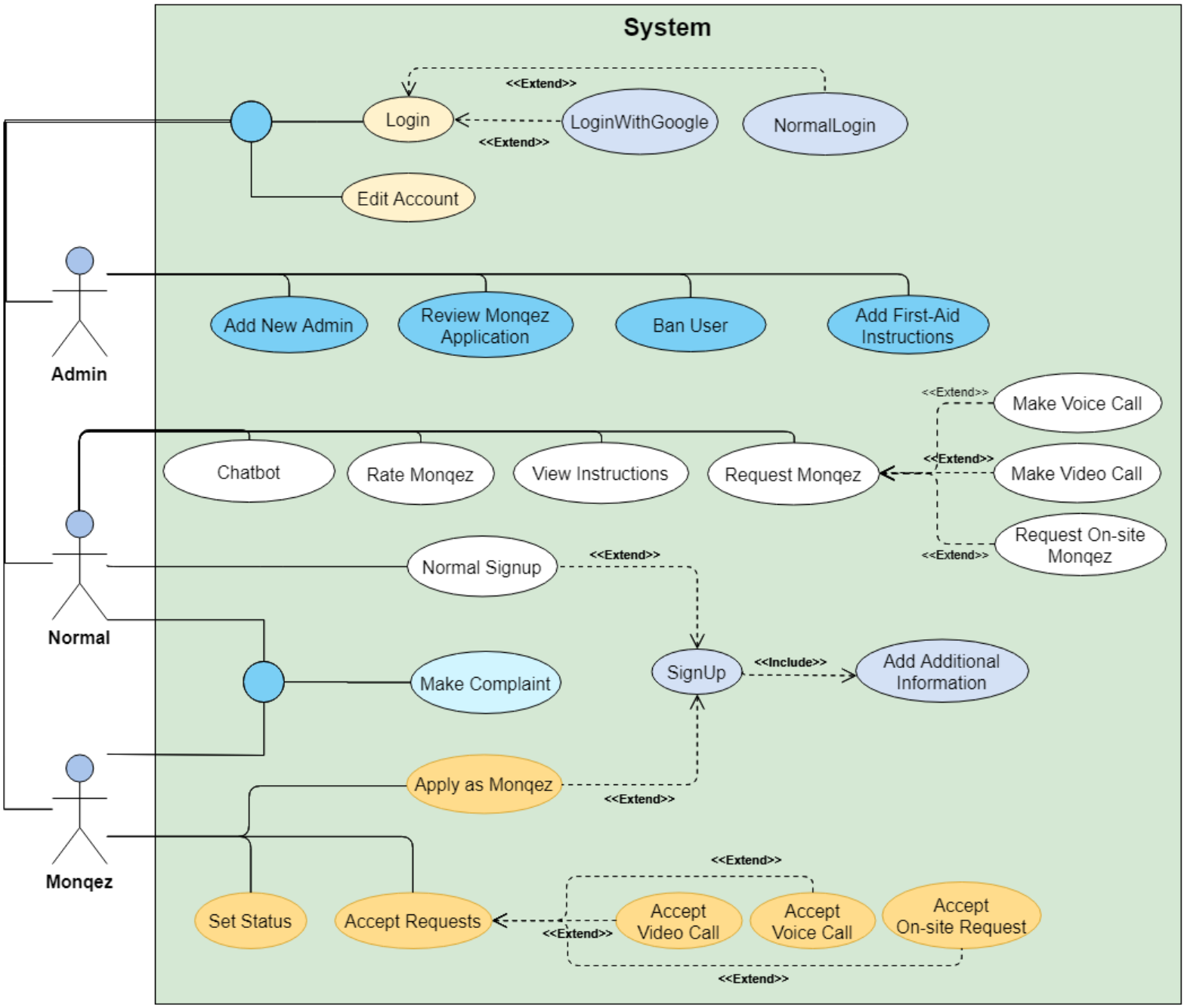

Figure 9: Use Case Diagram

Use Case ID:	1
Use Case Name:	Signup
Brief Description:	User creates a new account in the system.
Actors:	Normal user, Helper (Monqez)
Preconditions:	No account exists for the user in the database.
Postconditions:	A new account is added to the database with the user information.
Main Scenario:	1- System displays the signup page. 2- User enters the credentials (email and password). 3- System stores the credentials. 4- System redirects the user to the "Add Additional Information" page.
Exception Flow:	1- System displays an error message. 2- System displays the signup page.

Table 1: Signup Use Case

Use Case ID:	2
Use Case Name:	Add additional information
Brief Description:	User fills in data about himself into the system.
Actors:	Normal user
Preconditions:	User is signed up.
Postconditions:	Additional information is added to the existing user.
Main Scenario:	1- System displays the "Add Additional Information" page. 2- User enters the required data (full name, phone number, national ID, date of birth, address and gender). 3- System stores the information in the database. 4- System redirects the normal user to his/her home page.
Exception Flow:	1- System displays an error message.

Table 2: Add additional information Use Case

Use Case ID:	3
Use Case Name:	Apply as Monqez
Brief Description:	Monqez fills the signup application into the system.
Actors:	Helper (Monqez)
Preconditions:	User is signed up.
Postconditions:	Application is submitted and pending approval.
Main Scenario:	1- System displays the “Add Additional Information” page. 2- User enters the required data (full name, phone number, national ID, date of birth, address, gender, and first-aid certificate). 3- System stores the information in the database. 4- System adds the application to the Monqez Application Queue. 5- System redirects to the login page.
Exception Flow:	1- System displays an error message. 2- System displays “Add Additional Information” page.

Table 3: Apply as Monqez Use Case

Use Case ID:	4
Use Case Name:	Login
Brief Description:	Actors authenticate and get into the system.
Actors:	Normal user, Helper (Monqez), Administrator
Preconditions:	User must exist in the database and has not yet authenticated.
Postconditions:	User is authenticated and allowed to use the system functions.
Main Scenario:	1- System displays the login page. 2- User enters the credentials (email and password). 3- System validates the credentials. 4- System checks that the user has added his additional information. 5- System displays the appropriate user home page.
Alternative Scenario:	1- System displays the login page. 2- User enters the credentials (email and password). 3- System validates the credentials. 4- System checks and finds that the user has not added his additional information. 5- System displays “Add Additional Information” page.
Exception Flow:	1- System displays an error message. 2- System displays the login page.

Table 4: Login Use Case

Use Case ID:	5
Use Case Name:	Login with Google
Brief Description:	Actors authenticate and get into the system.
Actors:	Normal user, Helper (Monqez), Administrator
Preconditions:	-
Postconditions:	User is authenticated and allowed to use the system functions.
Main Scenario:	1- System displays the login page. 2- User selects his Google account. 3- System validates the credentials. 4- System checks that the user has added his additional information. 5- System displays the appropriate user home page.
Alternative Scenario:	1- System displays the login page. 2- User selects his Google account. 3- System validates the credentials. 4- System checks and finds that the user has not added his additional information. 5- System displays “Add Additional Information” page.
Exception Flow:	1- System displays an error message.

Table 5: Login with Google

Use Case ID:	6
Use Case Name:	Signup with Google
Brief Description:	User creates a new account in the system.
Actors:	Normal user, Helper (Monqez)
Preconditions:	No account exists for the user in the database.
Postconditions:	User is authenticated and a new account is added to the database with the user information.
Main Scenario:	1- System displays the login page. 2- User selects his Google account. 3- System validates the credentials. 4- System checks and finds that the user has not added his additional information. 5- System displays the "Add Additional Information" page.
Alternative Scenario:	1- System displays the login page. 2- User selects his Google account. 3- System validates the credentials. 4- System checks that the user has added his additional information. 5- System displays the appropriate user home page.
Exception Flow:	1- System displays an error message.

Table 6: Signup with Google Use Case

Use Case ID:	7
Use Case Name:	Add New Admin
Brief Description:	An administrator adds a new admin to the system
Actors:	Admin
Preconditions:	Admin is logged in.
Postconditions:	A new admin is added to the database.
Main Scenario:	1- System displays the "Add New Admin" page. 2- Admin enters the email and password of the new admin. 3- System stores the credentials in the authentication database. 4- System marks the created user as an admin. 5- System redirects the original admin to the "Admin Home Screen" page.
Exception Flow:	1- System displays an error message.

Table 7: Add new admin Use Case

Use Case ID:	8
Use Case Name:	Review Monqez Application
Brief Description:	An administrator reviews a Monqez application and either accepts or declines the Monqez
Actors:	Admin
Preconditions:	Admin is logged in. A user has applied as a Monqez.
Postconditions:	Monqez application is reviewed.
Main Scenario:	1- System displays all unreviewed Monqez applications. 2- Admin selects a Monqez Application. 3- System displays all the details of the application (name, age, phone number, gender, and the first aid certificate). 4- Admin chooses to accept the application. 5- System emails the Monqez reporting his/her acceptance and allows him/her to login to the system as well as mark the application as reviewed.
Alternative Scenario:	1- System displays all unreviewed Monqez applications. 2- Admin selects a Monqez Application. 3- System displays all the details of the application (name, age, phone number, gender, and the first aid certificate). 4- Admin chooses to reject the application. 5- System emails the Monqez reporting his/her rejection and marks the application as reviewed.
Exception Flow:	1- System displays an error message. 2- System displays all unreviewed Monqez applications.

Table 8: Review Monqez Application Use Case

Use Case ID:	9
Use Case Name:	Edit Account
Brief Description:	User edits information at his profile.
Actors:	Normal user, Helper (Monqez), Admin
Preconditions:	User is logged in.
Postconditions:	User's information is updated.
Main Scenario:	1- System displays the "Edit Account" page. 2- User enters the data (full name, phone number, date of birth, address) that he/she wants to update. 3- System updates the data in the database.
Exception Flow:	1- System displays an error message.

Table 9: Edit Account Use Case

Use Case ID:	10
Use Case Name:	Make a voice call
Brief Description:	Normal user creates a voice call and any free Monqez can answer it.
Actors:	Normal user
Preconditions:	User is logged in.
Main Scenario:	1- System displays the Normal User's home screen. 2- Normal user taps on make a voice call and enters any additional information (if exists). 3- System adds call to the call queue and redirects Normal User to the call screen.
Exception Flow:	1- System displays an error message.

Table 10: Make a voice call Use Case

Use Case ID:	11
Use Case Name:	Accept a voice call
Brief Description:	Helper users selects a call from the call queue to answer it and connects to the normal user via voice call.
Actors:	Helper user (Monqez)
Preconditions:	User is logged in
Main Scenario:	1- System displays the Helper user's home screen. 2- Helper user selects to view all call queue. 3- System displays all call queue to the helper. 4- Helper selects the call he wants to enter. 5- System connects the helper user with the normal user via voice call.
Exception Flow:	1- System displays an error message.

Table 11: Accept a voice call Use Case

Use Case ID:	12
Use Case Name:	Make a video call
Brief Description:	Normal user creates a video call and any free Monqez can answer it.
Actors:	Normal user
Preconditions:	User is logged in.
Main Scenario:	1- System displays the Normal User's home screen. 2- Normal user taps on make a video call and enters any additional information (if exists). 3- System adds call to the call queue and redirects Normal User to the call screen.
Exception Flow:	1- System displays an error message.

Table 12: Make a video call Use Case

Use Case ID:	13
Use Case Name:	Accept a video call
Brief Description:	Helper user selects a call from the call queue to answer and connects to the normal user via video call.
Actors:	Helper user (Monqez)
Preconditions:	User is logged in.
Main Scenario:	1- System displays the Helper user's home screen. 2- Helper user selects to view the call queue. 3- System displays the call queue to the helper. 4- Helper selects the call he wants to answer. 5- System connects the helper user with the normal user via video call.
Exception Flow:	1- System displays an error message.

Table 13: Accept a video call Use Case

Use Case ID:	14
Use Case Name:	Making a voice call from Chatbot
Brief Description:	User chatting with chatbot and asking it to make a voice call and optionally providing voice call additional information.
Actors:	Normal user
Preconditions:	User is logged in.
Postconditions:	User is directed to voice call additional information pop-up.
Main Scenario:	1- System displays the "Chatbot" page. 2- User types into chat any word related to voice call. 3- User may provide the chatbot with any additional information that describes what this call about. (e.g. "I want to make a voice call **about being ill**") 4- Dialogflow processes the data and detects that the action is "voice call" and the additional information is "about being ill". 5- Dialogflow returns the data to the application in this format [voice, about being ill]. 6- System redirects the user to the voice call additional pop-up filled with additional information that the user has provided.
Exception Flow:	2- System displays an error message.

Table 14: Making a voice call from Chatbot Use Case

Use Case ID:	15
Use Case Name:	Rate request
Brief Description:	User chooses a previous request that he has made and rates the Monqez with stars – from 1 to 5 – and optionally adds a comment.
Actors:	Normal user
Preconditions:	1- User is logged in. 2- User has at least one closed request that he hasn't rated before.
Main Scenario:	1- System displays the "Previous Requests Screen". 2- User chooses to rate any request. 3- User enters the number of stars and comment, then presses "Submit". 4- System sends the rating information to the server. 5- System stores the rating information in the database. 6- User is redirected to his home screen.
Exception Flow:	System displays an error message.

Table 15: Rate Request Use Case

Use Case ID:	16
Use Case Name:	Complain about a request
Brief Description:	User chooses a previous request that he has made and makes a complaint about it.
Actors:	Normal user
Preconditions:	1- User is logged in. 2- User has at least one closed request.
Postconditions:	User is redirected to his home screen.
Main Scenario:	1- System displays the "Previous Requests Screen". 2- User chooses to complain about any request. 3- User enters the subject and message, then presses "Submit". 4- System sends the complaint information to the server to be stored in the database.
Exception Flow:	System displays an error message.

Table 16: Complain on request Use Case

Use Case ID:	17
Use Case Name:	Accept onsite request
Brief Description:	Helper user accepts incoming request from normal user.
Actors:	Helper (Monqez)
Preconditions:	Helper's status is "available".
Postconditions:	System notifies the normal user that a helper accepted his request and shows the helper's name and phone number.
Main Scenario:	1- System shows SOS notification to helper. 2- Helper presses the accept button. 3- System checks if another Monqez has accepted the request already. 4- System finds that no other Monqez has accepted the request yet. 5- System displays "Request Screen" that shows the route between helper and the normal user who makes the request.
Alternative Scenario:	1- System shows SOS notification to helper. 2- Helper presses the accept button. 3- System checks if another Monqez has accepted the request already. 4- System finds that another Monqez has accepted the request already. 5- System displays the "Helper Home Screen".
Exception Flow:	1- System displays an error message. 2- System displays the "Helper Home Screen".

Table 17: Accept onsite request Use Case

Use Case ID:	18
Use Case Name:	Complete onsite Request
Brief Description:	Helper user completes the request after arriving and performing first aid.
Actors:	Helper (Monqez)
Preconditions:	Helper has accepted the request.
Postconditions:	System notifies the normal user that the request has been completed.
Main Scenario:	1- Helper presses the “complete” button. 2- System checks if the distance between the helper and normal user who makes the request is less than or equal 1 km. 3- System finds that the distance is less than or equal 1 km. 4- System marks the request as “completed” and displays “Helper Home Screen”.
Alternative Scenario:	1- Helper presses the “complete” button 2- System checks if the distance between the helper and normal user who makes the request is less than or equal 1 km. 3- System finds that the distance is greater than 1 km. 4- System displays “Helper Request Screen”.
Exception Flow:	System displays an error message.

Table 18: Complete onsite Request Use Case

Use Case ID:	19
Use Case Name:	Decline onsite request
Brief Description:	Helper user declines incoming request from normal user.
Actors:	Helper (Monqez)
Preconditions:	Helper's status is "available".
Postconditions	1- The request remains pending. 2- System displays the "Helper Home Screen "page.
Main Scenario:	1- System shows SOS notification to helper. 2- Helper presses the decline button. 3- System displays "Helper Home Screen".
Exception Flow:	System displays an error message.

Table 19: Decline onsite request Use Case

Use Case ID:	20
Use Case Name:	Request onsite Monqez
Brief Description:	Normal user requests helper on site.
Actors:	Normal user
Preconditions:	1- User is logged in. 2- User has no active request.
Postconditions:	1- The request is added to the database. 2- System notifies the nearest 3 helpers.
Main Scenario:	1- Normal user presses the "Get Help" button. 2- System shows "Add Additional Information" pop-up. 3- Normal user fills the request's additional information and presses "Submit". 4- System notifies the nearest 3 helpers.
Alternative Scenario:	1- Normal user presses the "Get Help" button. 2- System finds that no available Monqez is near the normal user. 3- System informs the normal user that no available Monqez exists at the moment.
Exception Flow:	System displays an error message.

Table 20: Request onsite Monqez Use Case

3.4. User Groups and Access Privileges

The following are the user groups that exist in the system:

1. **Normal user:** is a user who was at the scene of the accident reporting and asking a paramedic to aid for him/her or another person. The normal user can perform the following functions:
    - ***Sign up:*** The normal user can register his/her details to be able to call paramedics.
    - ***Log in:*** The normal user can log in using his/her phone number or email to be able to use the application functionalities.
    - ***Ask for Monqez:*** The normal user has the capability to ask for Monqez in case he/she or others were involved in an accident or need quick medical care. The normal user can ask for either onsite or online (video/voice) Monqez.
    - ***View first aid instructions:*** The normal user can view first aid instructions about many different types of injuries.
    - ***Rate the Monqez.***
2. **Monqez user:** is a user who registers in our system as a rescuer (Monqez) by attaching an accredited certificate of the first aid training course he/she took from any accredited organization and then awaits approval of his application to join the rescuer list until the certificate is reviewed. The Monqez can perform the following functions:
    - ***Sign up:*** The Monqez user can add his/her information attached with the first aid certificate to the rescuers queue to be reviewed.
    - ***Log in:*** The Monqez user can log in using his/her phone number or email to be able to use the application functionalities.
    - ***Specify working time:*** The Monqez user has the option to select his/her status and when he/she will be available for work.
    - ***Accept requests:*** The Monqez user can accept either onsite or online (video/voice) requests.
    - ***Request Description:*** The Monqez user can view the description of the request provided by the normal user who made it.
    - ***Make complaint:*** The Monqez user can complain about any inappropriate events that happened while fulfilling a request.
3. **Admin user:** The Admin user accepts or rejects requests to join the rescuers list based on the validity of the certificates that the rescuers attach when creating an account. The admin can perform the following functionalities:
    - ***Log in:*** The admin can log in using his/her phone number or email to be able to use the application functionalities.
    - ***Add another admin.***
    - ***View Monqez applications requests:*** The admin can retrieve all the non-reviewed Monqez applications requests and browse them.
    - ***Accept/decline Monqez application requests:*** The admin user has the capability of accepting or declining the Monqez application requests after reviewing the first aid certificate.
    - ***View ratings and complaints:*** The admin can review complaints and ratings.
    - ***Block users:*** The admin can block any user if there are many valid complaints against that user.

Chapter 4: System Design

4.1. System Architecture

We use two architectures in our project, the first one is **Client-Server** architecture which is used to connect the front-end application (Flutter) with the back-end server (Node.js), and the second is **MVC** architecture which is used to design the server. The server is designed to contain a lot of models which are used in other sections of the server and the entry point of our server is the controller layer which accepts any request, authorizes it, and routes the request to the correct function to be processed. Another layer in the server is the database where we used firebase's real-time database and the server is responsible for communicating with this layer through its models.

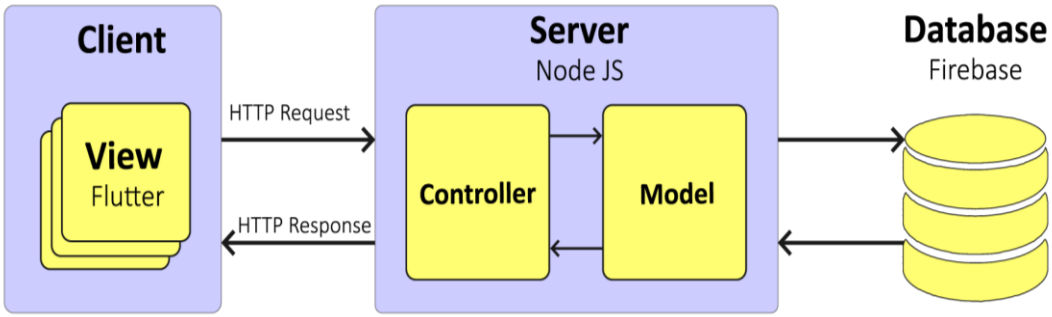

Figure 10: System Architecture

4.2. System Component Diagram

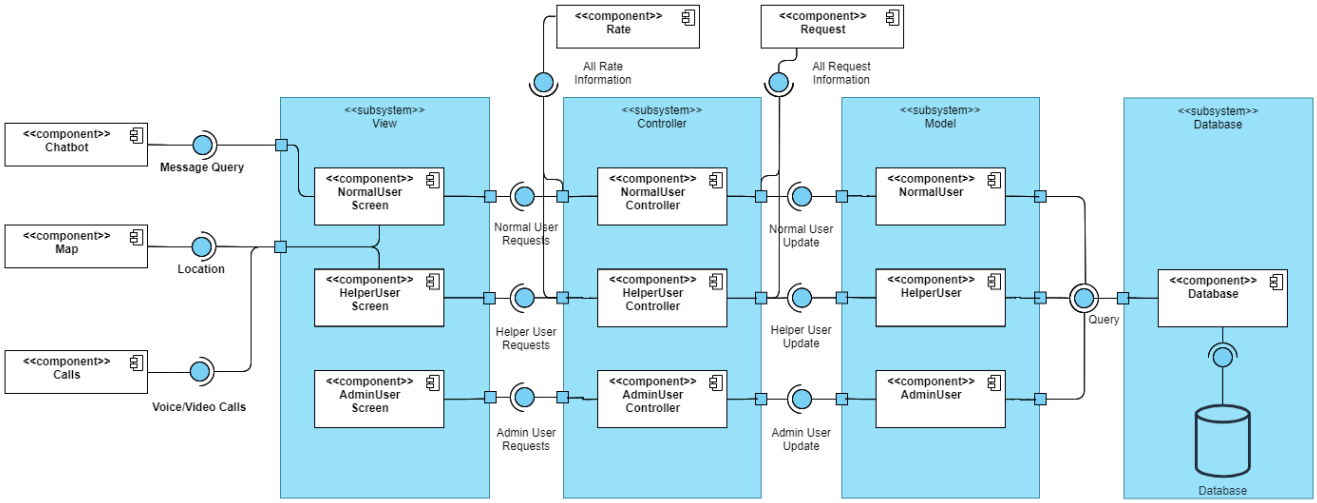

Figure 11: System Component Diagram

4.3. Class Diagram

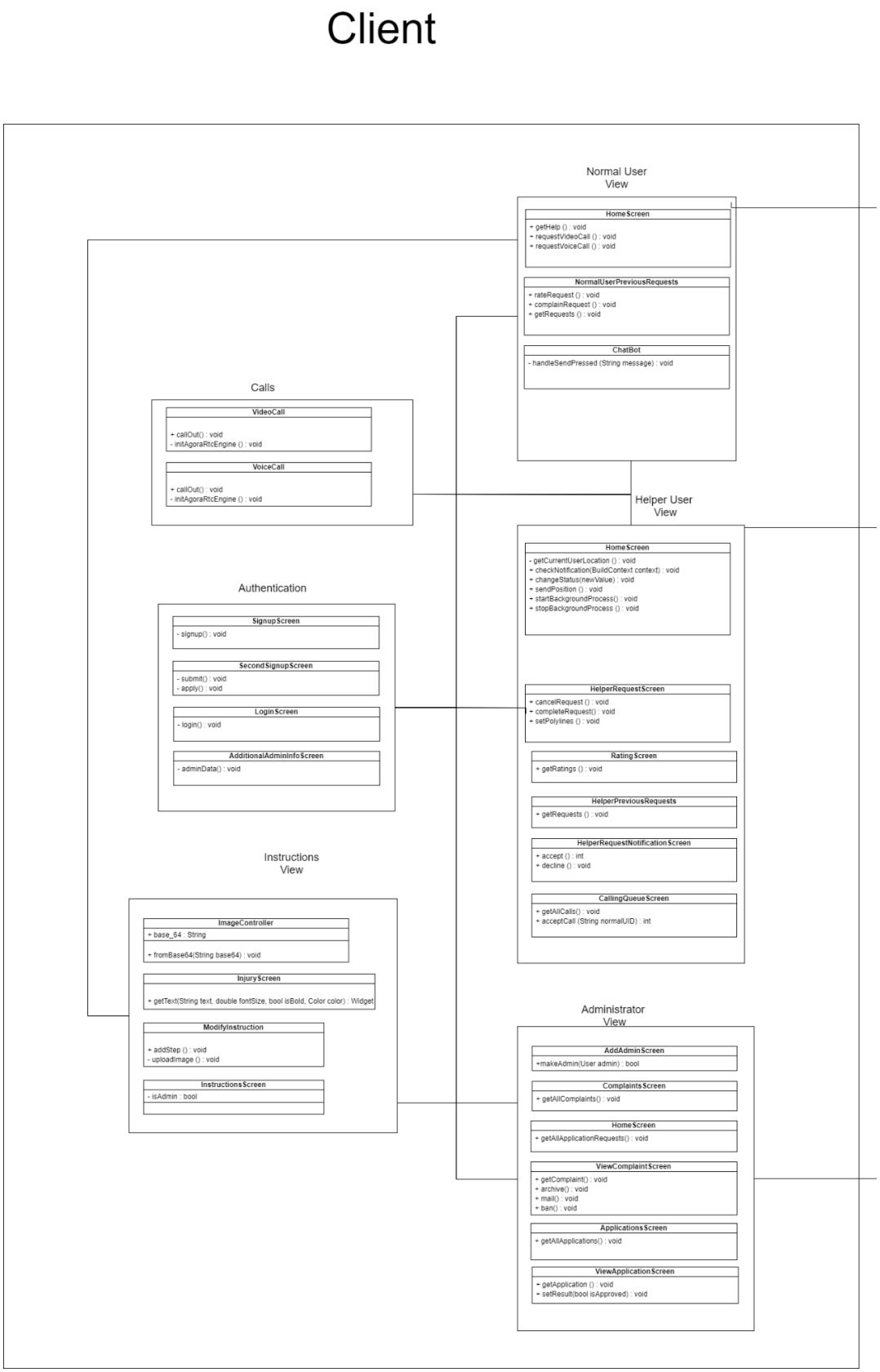

Figure 12: Class Diagram (Client)

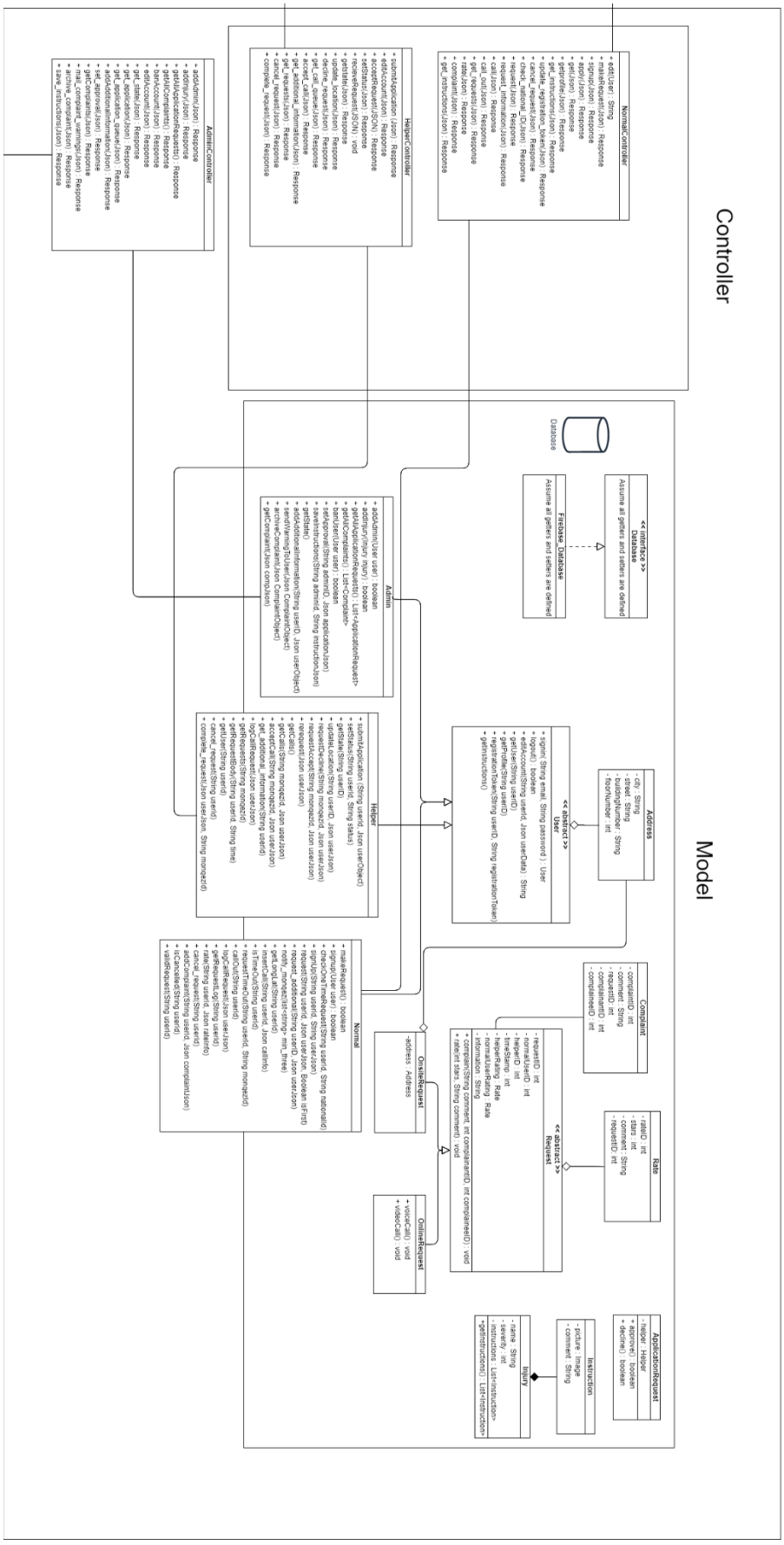

Figure 13: Class Diagram (Server)

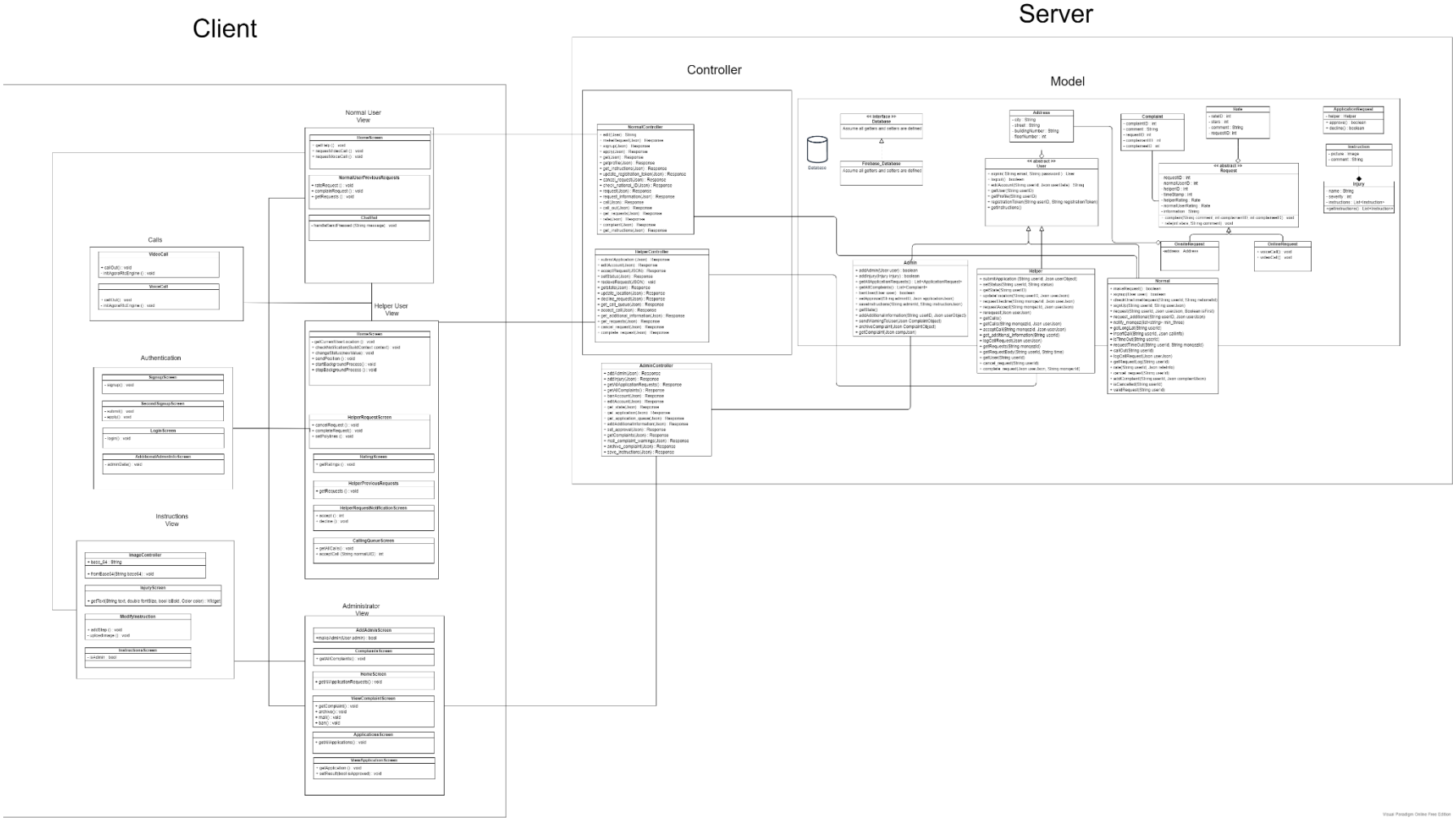

Figure 14: Class Diagram

4.4. Sequence Diagrams

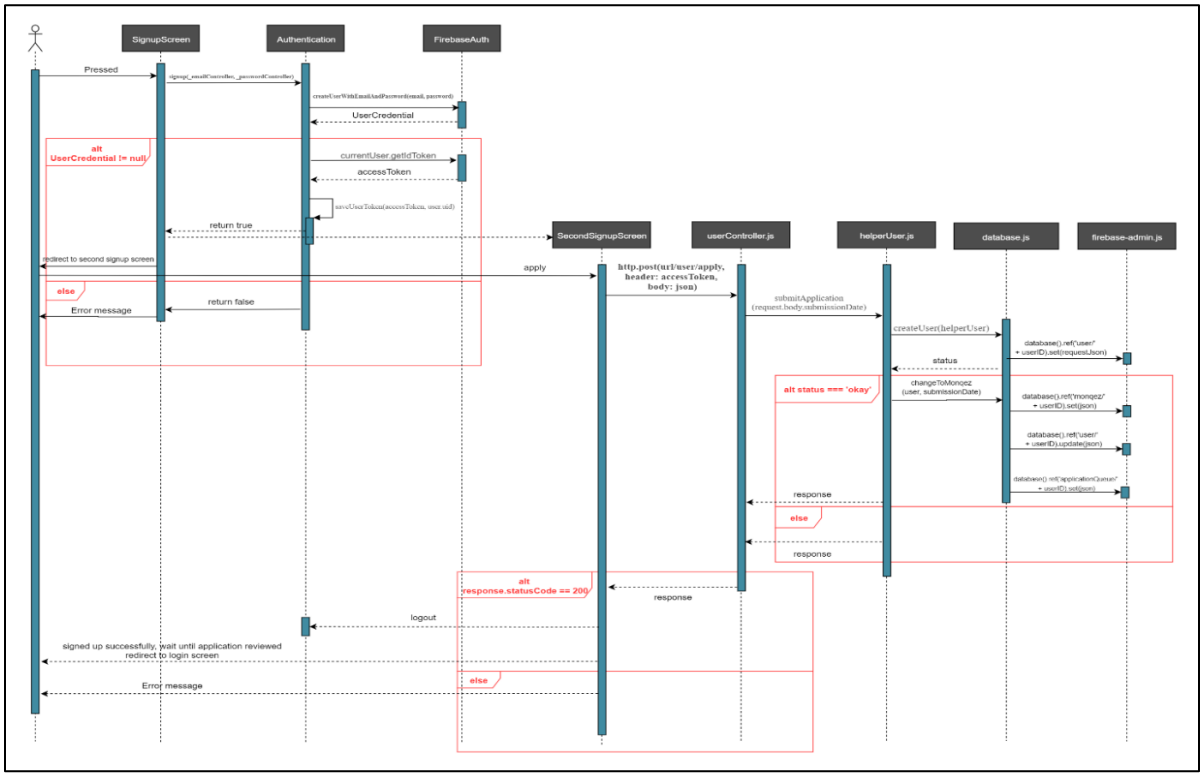

Figure 15: "Sign up as Monqez" Sequence Diagram

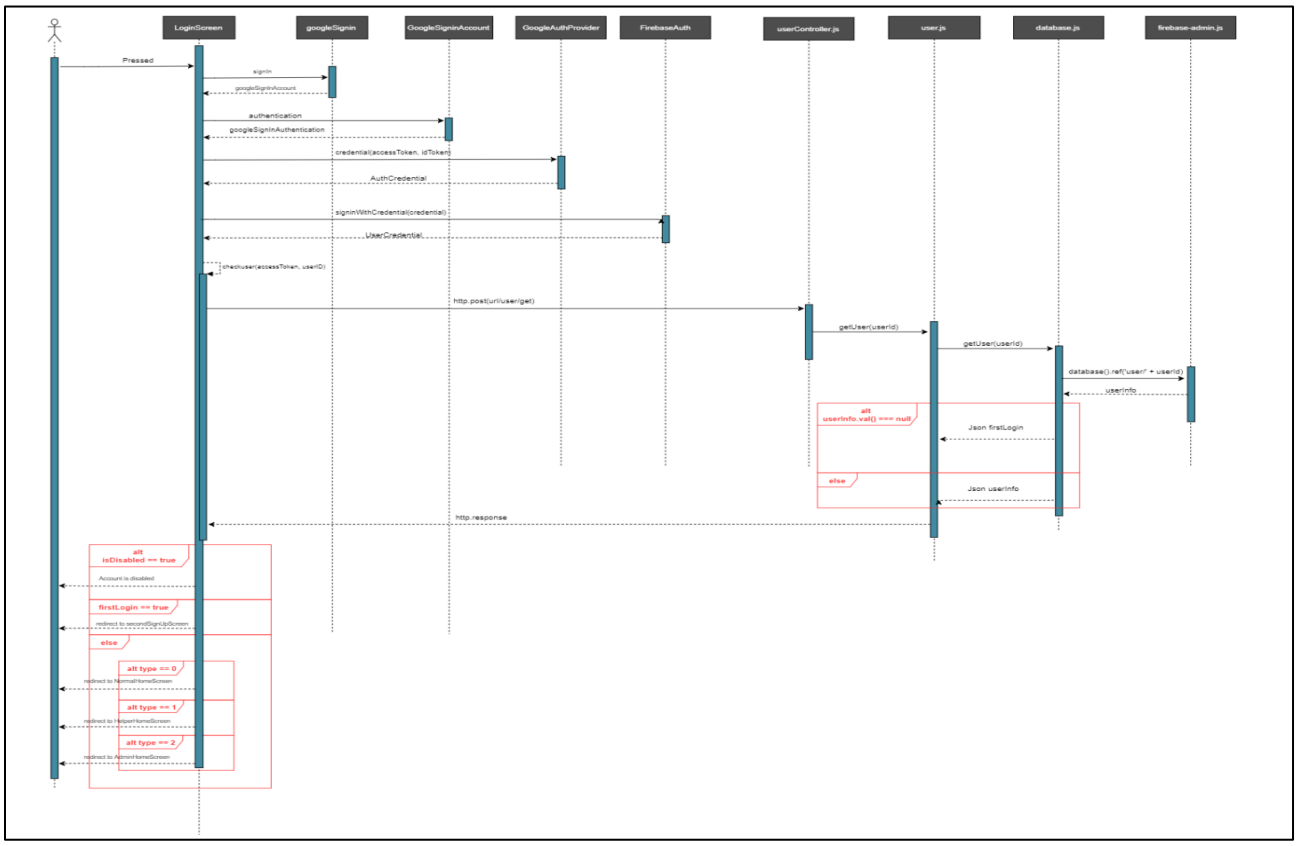

Figure 16: "Sign in with Google" Sequence Diagram

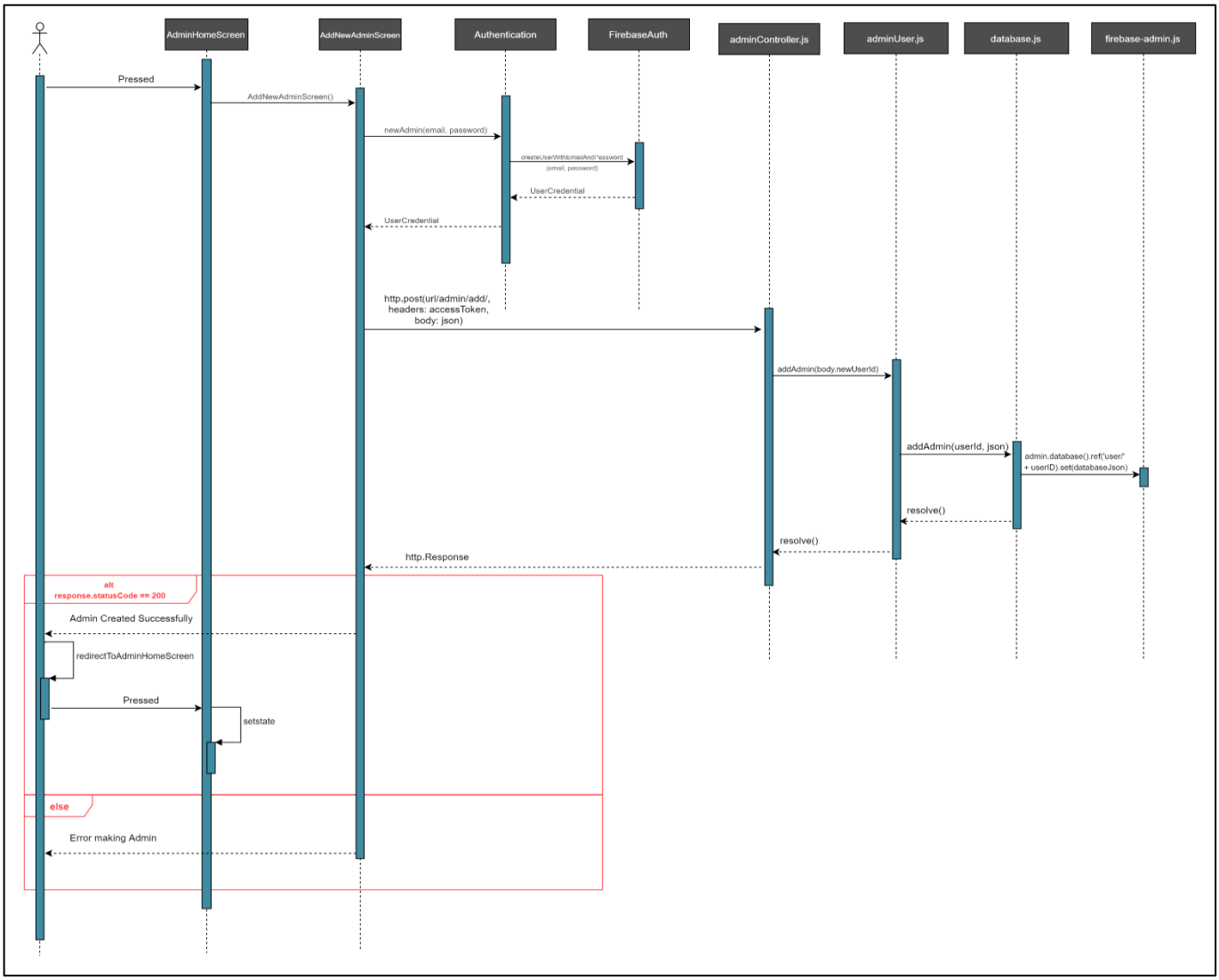

Figure 17: "Add new Admin" Sequence Diagram

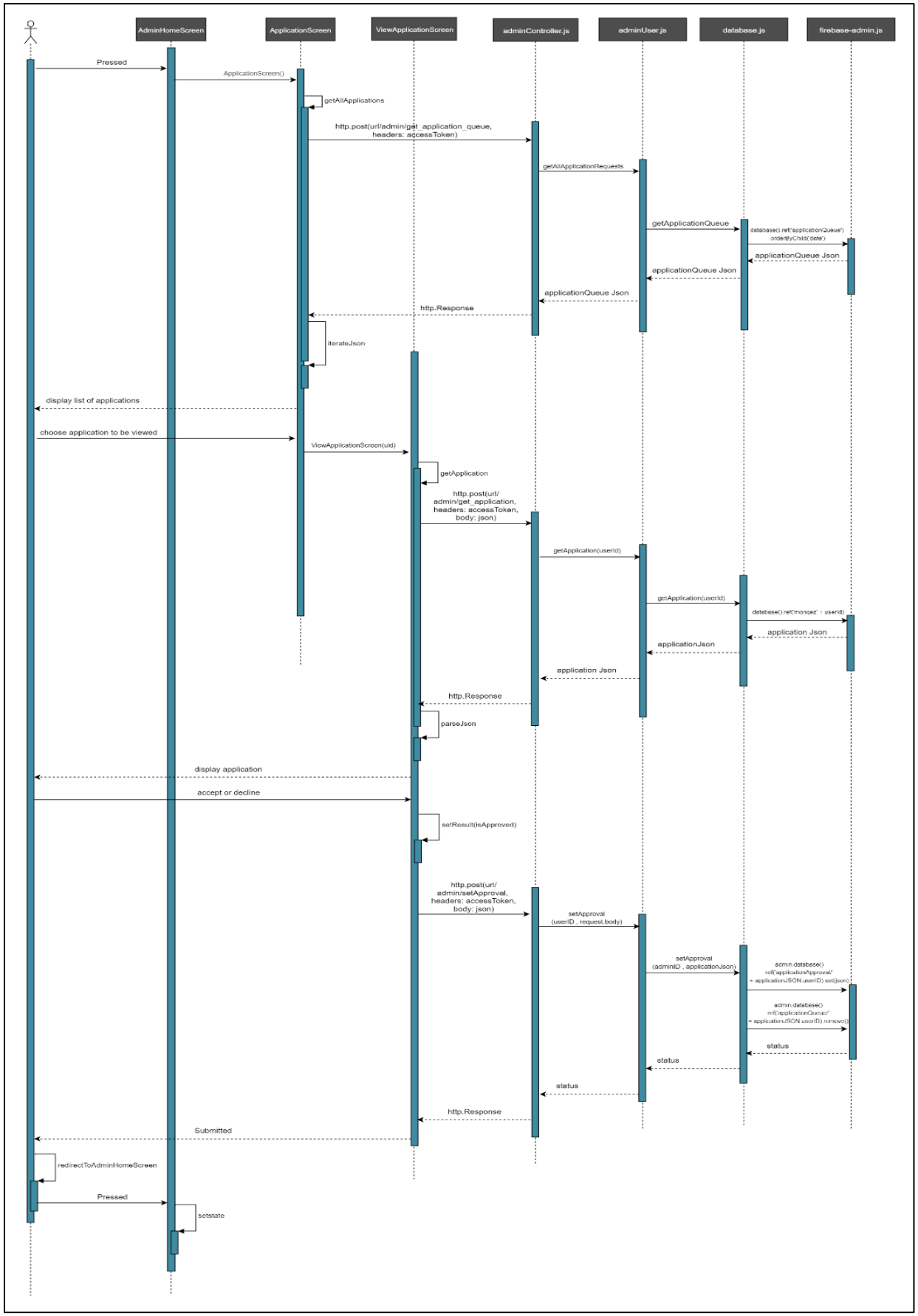

Figure 18: "Accept or Decline Monqez Application" Sequence Diagram

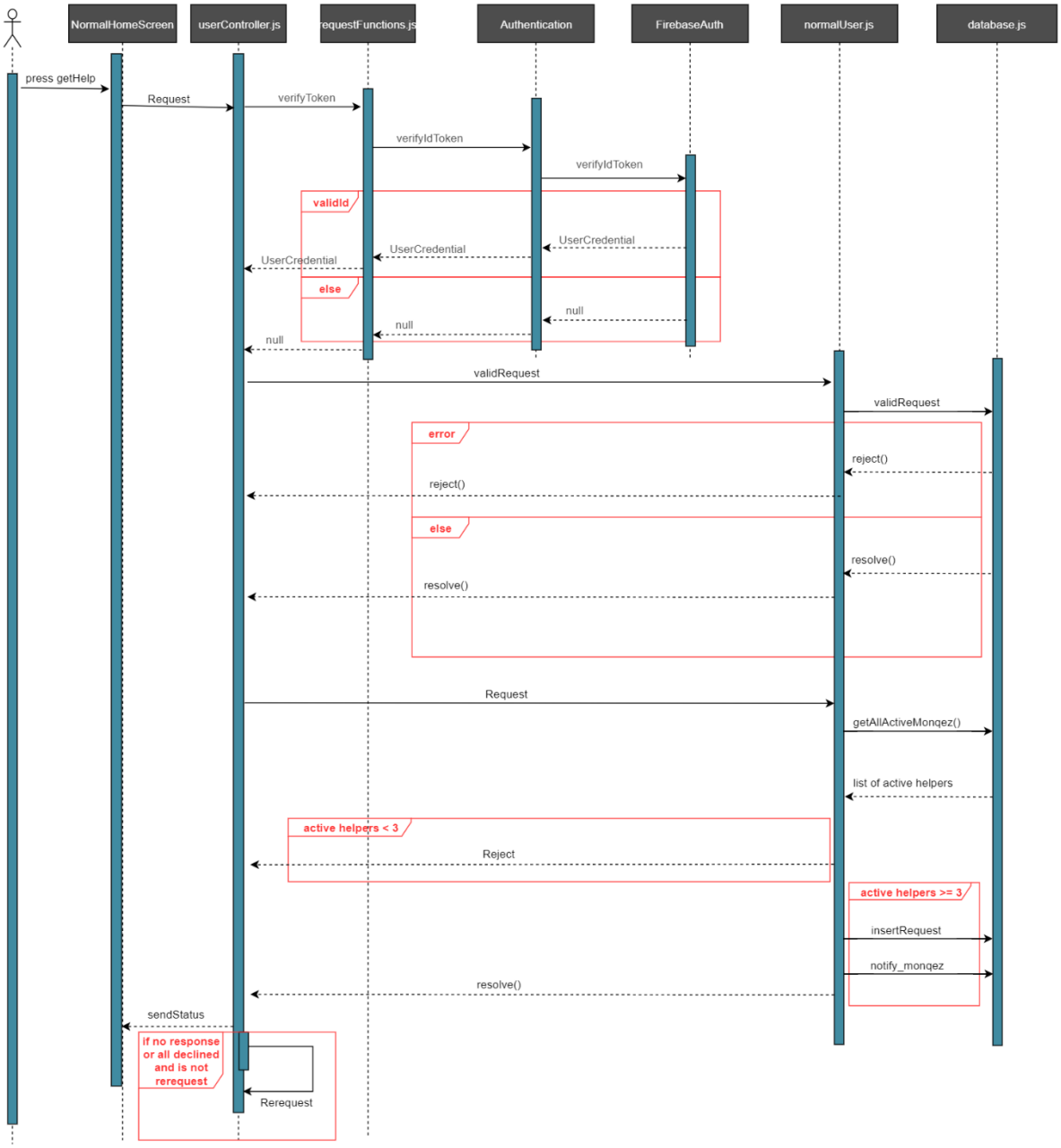

Figure 15: "Request Monqez" Sequence Diagram

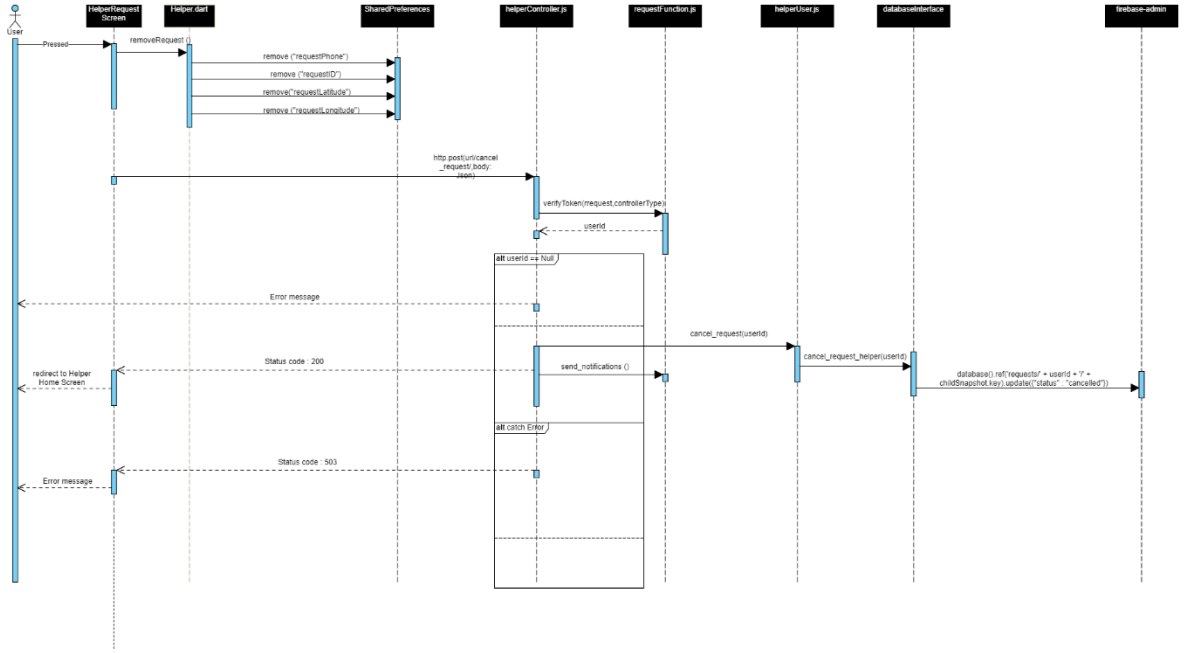

Figure 16: "Cancel Request" Sequence Diagram

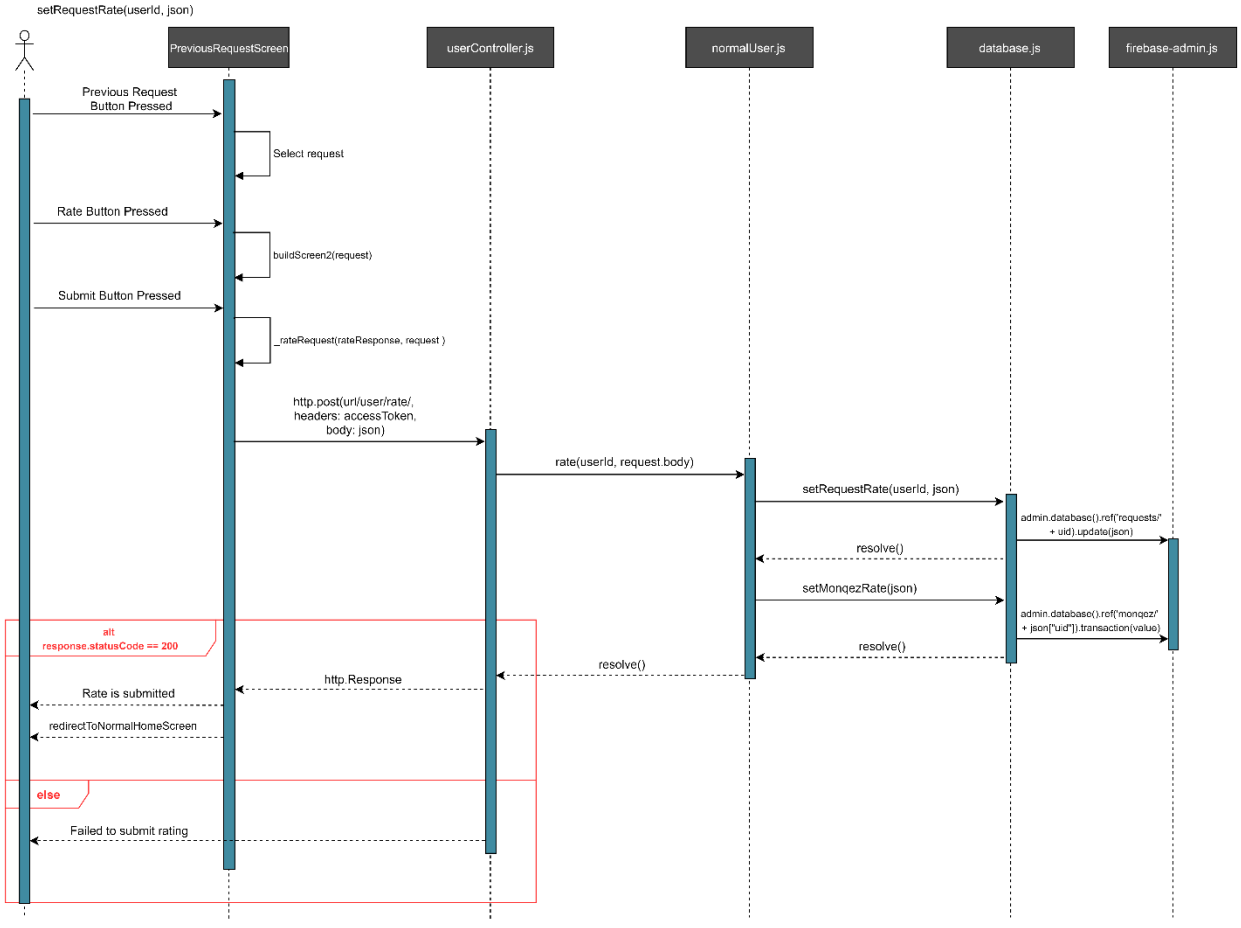

Figure 19: "Rate Monqez" Sequence Diagram

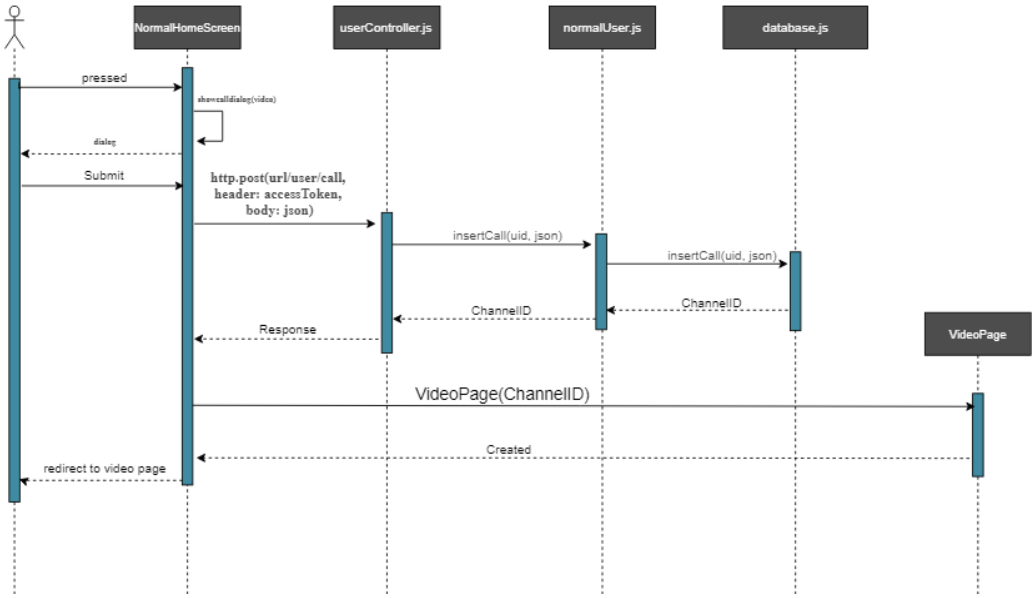

Figure 20: "Make Video Call" Sequence Diagram

4.5. Project ERD

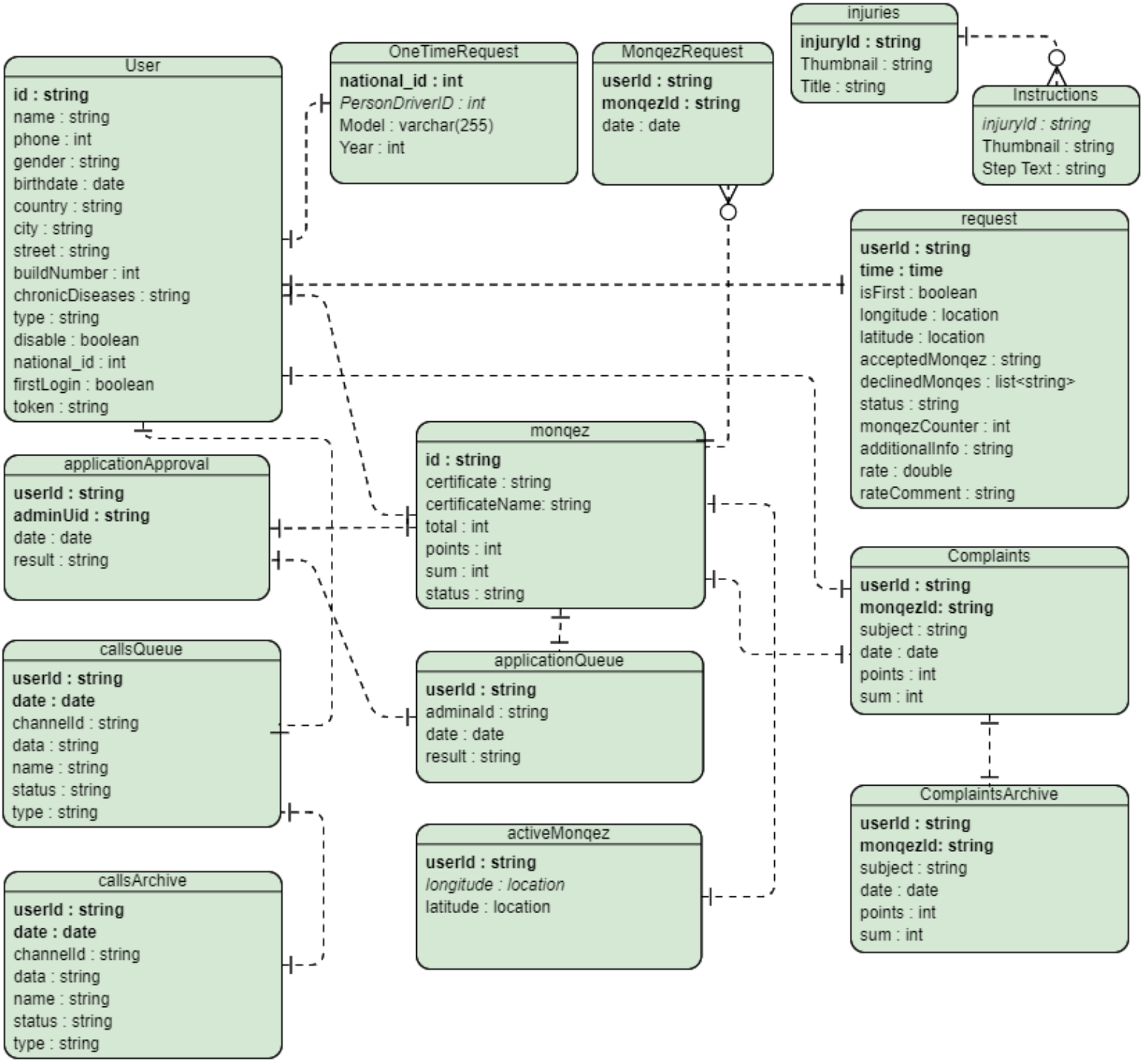

Figure 21: Application Entity Relationship Diagram

4.6. System GUI Design

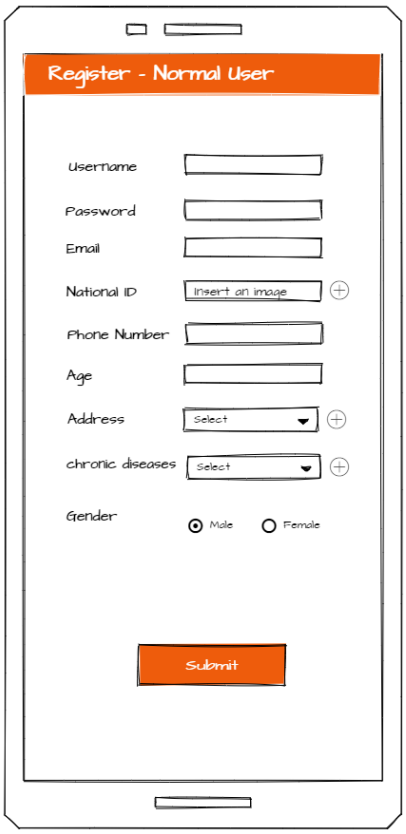

Figure 22: Normal User Signup Screen

Figure 23: Login Screen

Figure 24: Normal User Home Screen

Figure 25: Request's Additional Information Popup

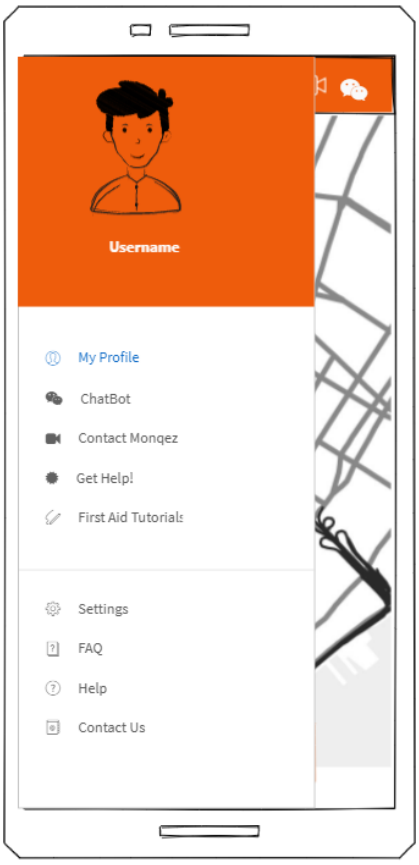

Figure 26: Normal User's Navigation Drawer

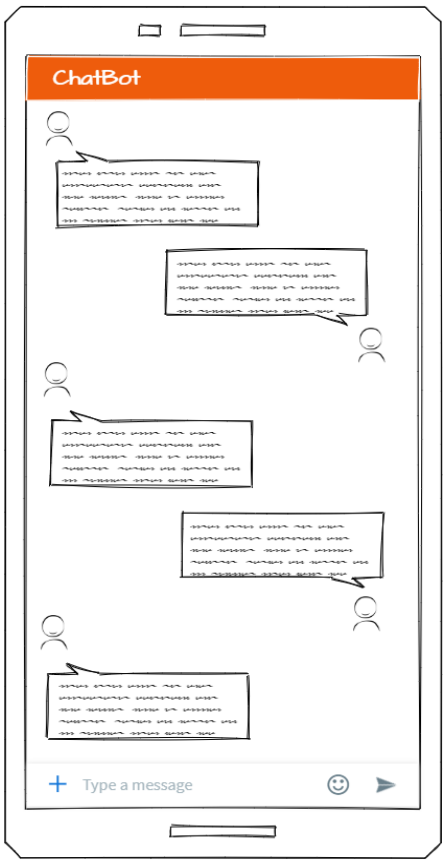

Figure 27: Chatbot Screen

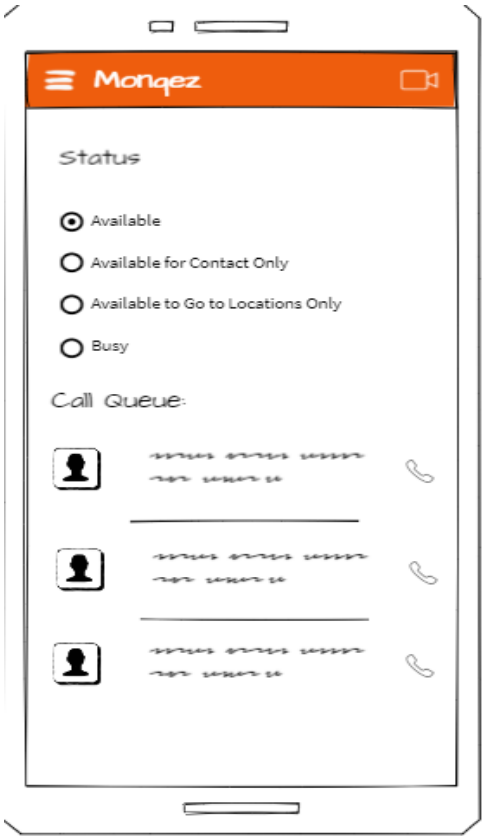

Figure 28: Monqez Home Screen

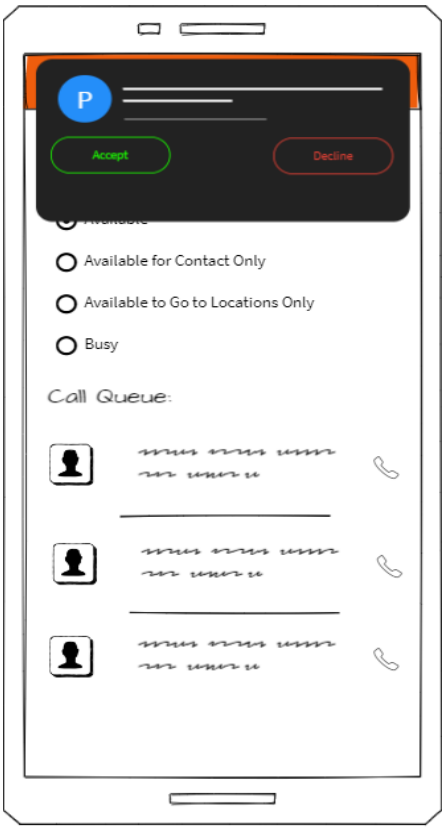

Figure 29: Monqez Request Notification

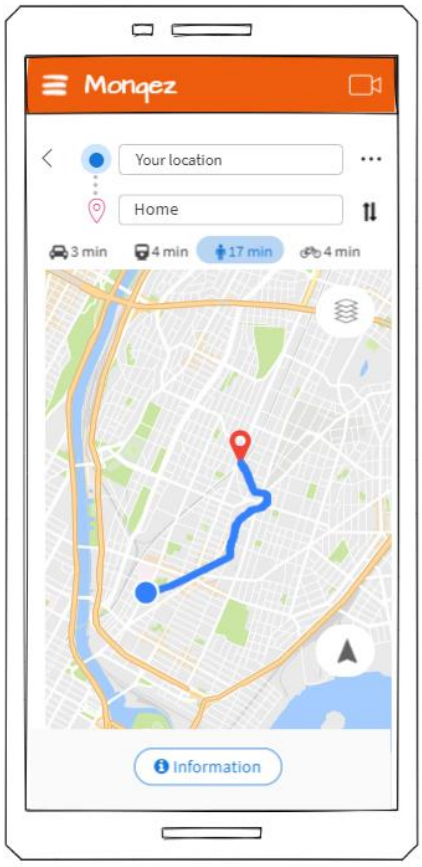

Figure 30: Monqez Navigation Screen

Figure 31: Monqez Display Request's Additional Information Popup

Chapter 5: Implementation and Testing

5.1. Methods

In our system we used two levels of authentication, this done in both “sign up” and “sign in”. In “sign up”, the first level of authentication is creating the user’s account in the Firebase Authentication where we take the user’s email and password. The second level is where we take all the user’s additional information and insert it in the real-time database. We also used two levels of authentication in “sign in”, the first one is authenticating the user credential used Firebase Authentication and the second one is retrieving the type of the user (for authorization, as well) from the real-time database.

To implement the voice and video calls, the Agora SDK was used in order to make the real-time engagement possible. The normal user requests to make a call in which the server places the channel id in the call queue. This channel id is used to open a channel using Agora. When a helper is willing to join a call, he opens the call queue and selects the request he wants to join which gives him the channel id in order to join the same channel as the normal user is in. When both the normal user and the helper user are connected to the same channel, they can communicate with each other in real-time via voice or video until one user leaves the channel.

To make the application more usable, a chatbot was built using Dialogflow platform. Dialogflow is a natural language understanding platform used to design and integrate a conversational user interface which uses both rule-based grammar matching as well as machine learning matching. We provided the platform with our training phrases in both the entities and intents. Some entities were identified by machine learning, synonyms, and fuzzy matching such as the type of call, while others were identified using regular expressions such as the additional information entity. The chatbot opens a session with the user where it gets every query from the user and responds to it. Each response is checked to see if there is an action that should be made or not, and if an action exists, the response is parsed to know the required action and then apply this action.

5.2. Testing

5.2.1. Manual Testing (Sample Test Cases)

5.2.1.1. Make a request when an active request exists

- When a user presses the “Get Help” button, he can cancel the request if no Monqez accepted it.

Figure 32: Normal User Home Screen (Manual Test)

- When a Monqez named “Khaled” accepts the onsite request, the normal user can no longer cancel this request nor make another request.

Figure 33: Monqez Information in Normal Home Screen (Manual Test)

5.2.1.2. Helper logs in before an admin accepts the application

- When a helper user signs up, he enters his email and password.

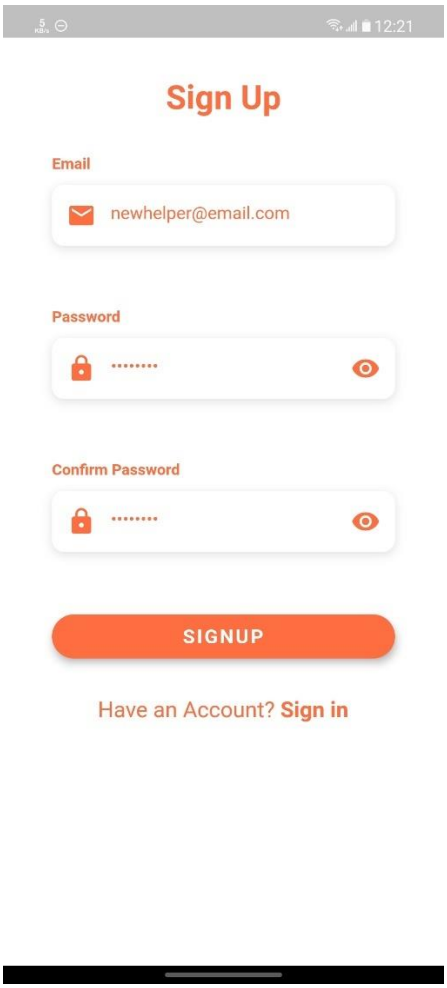

Figure 34: Signup Screen (Manual Test)

- He fills his application information and includes his certificate as PDF.

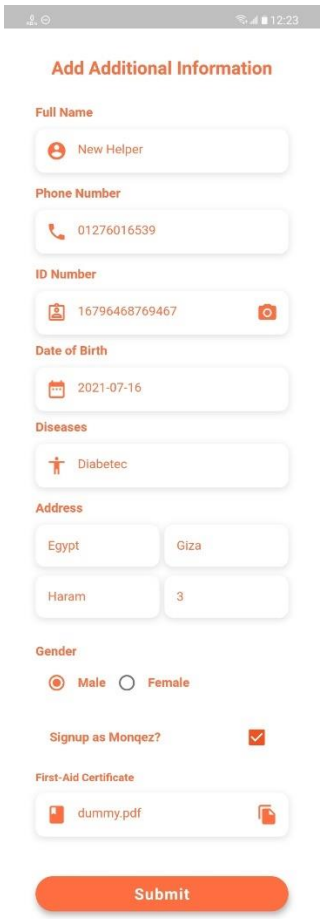

Figure 35: Add Additional Information Screen (Manual Test)

- When he tries to log in, a toast message appears that tells him to wait until the application is reviewed by an admin.

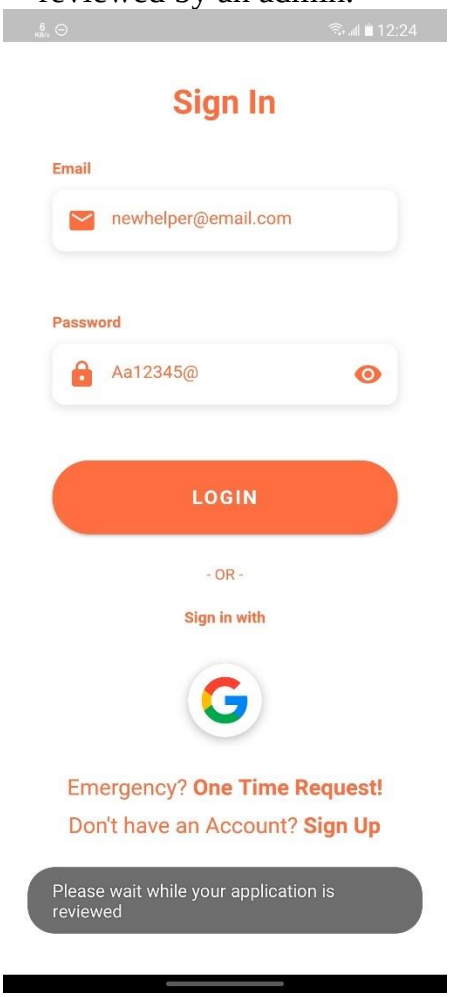

Figure 36: Login Screen (Manual Test)

5.2.2. Unit Testing

A unit test for each function in the backend server is made in order to verify that each and every function works properly. Each unit test has an input and expected return value that is compared to the actual returned value to verify that the function is working correctly. The following are sample unit tests' input and expected output as well as a summary of the unit tests' run.

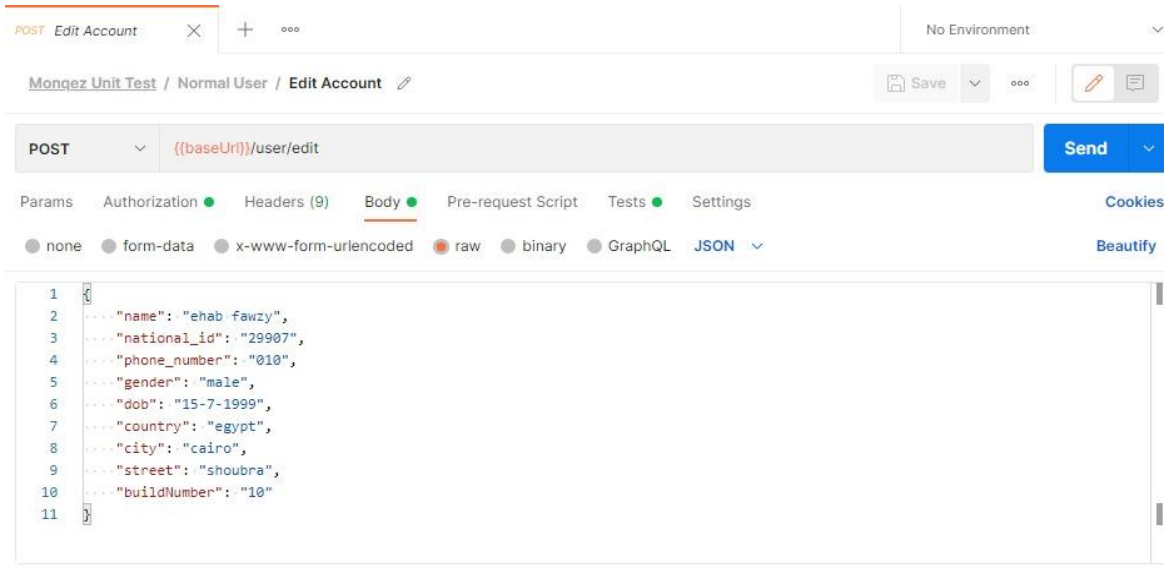

Figure 37: Edit Account Unit Test Input

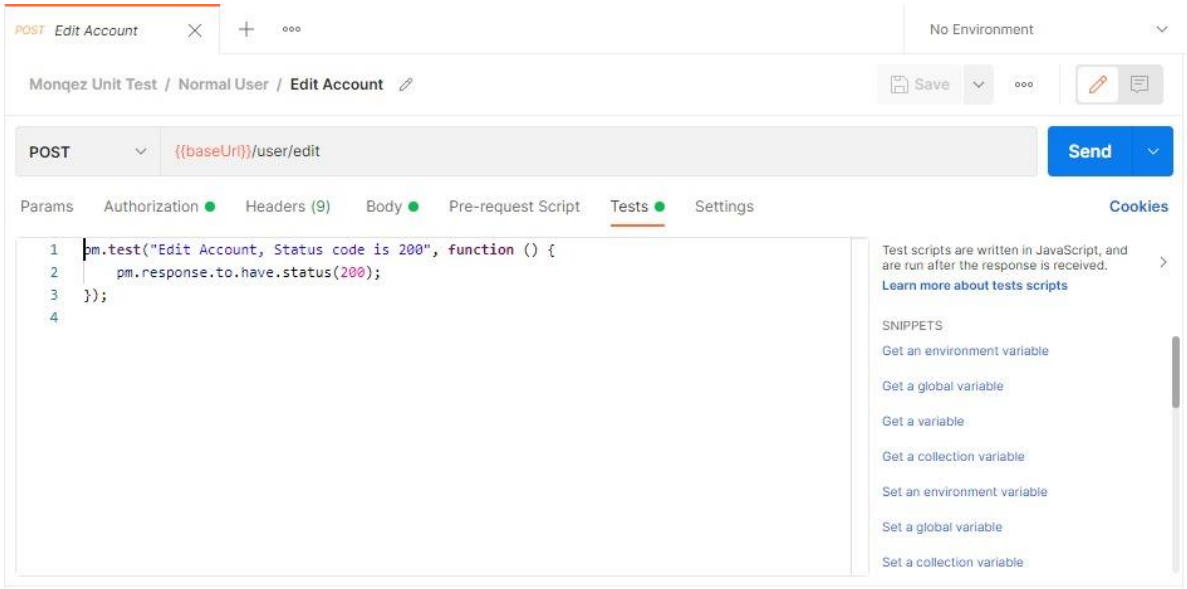

Figure 38: Edit Account Unit Test Expected Return Value

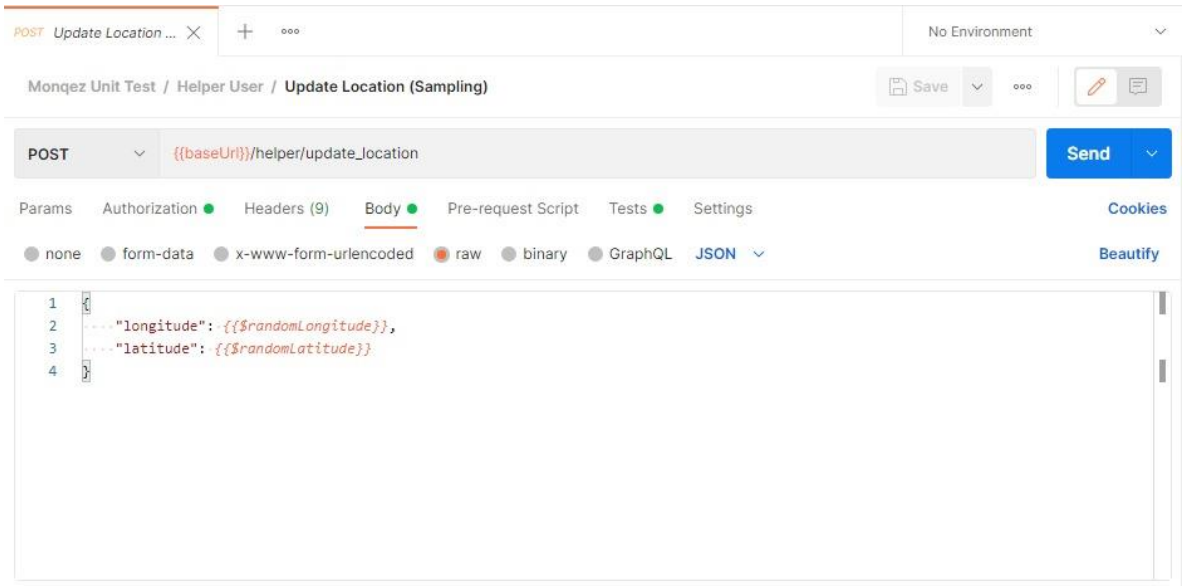

Figure 39: Update Location Unit Test Input

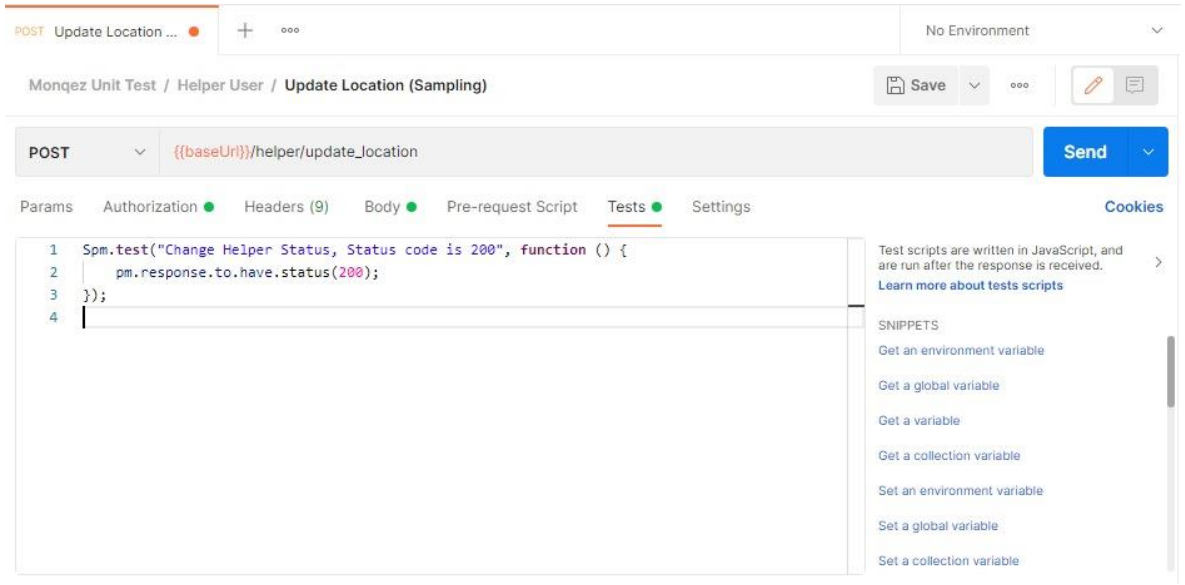

Figure 40: Update Location Unit Test Expected Return Value

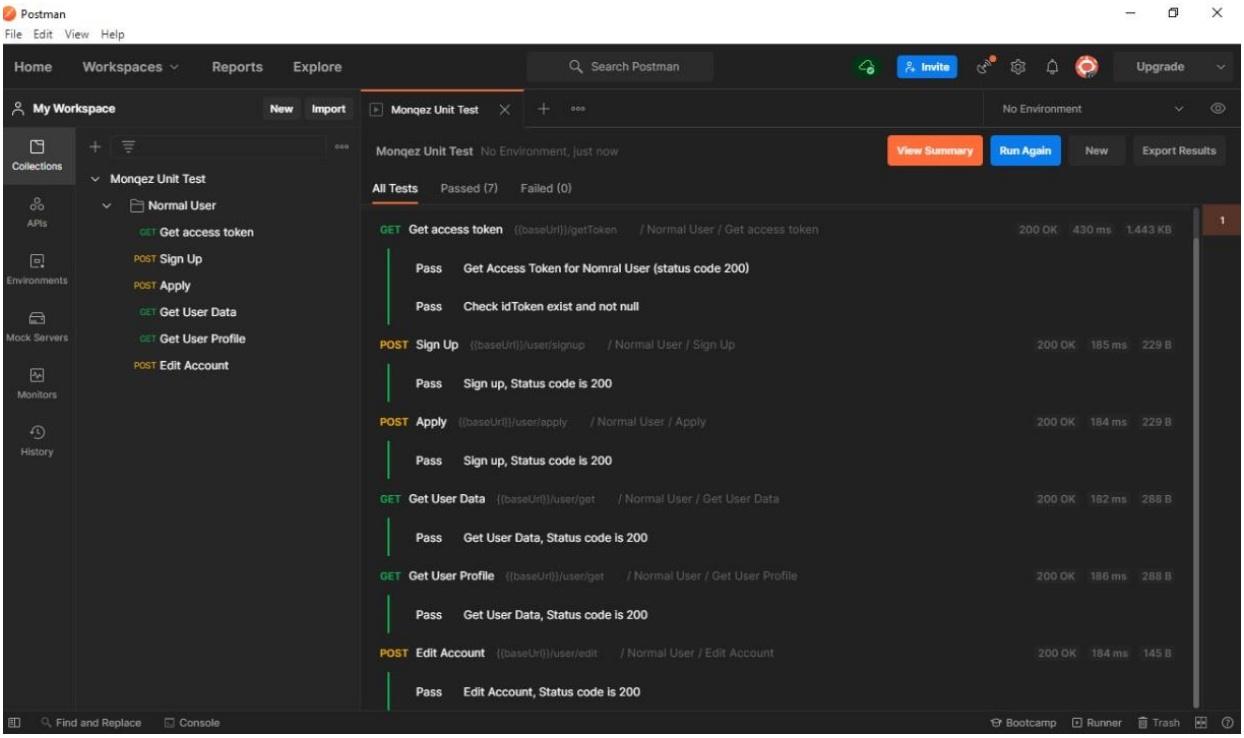

Figure 41: Unit Tests Summary

Chapter 6: Conclusion

To sum things up, Monqez is a mobile application developed to help people in need of first aid. It is expected to decrease the number of fatalities that occur before ambulances arrive by providing the appropriate first aid to the person in need of medical assistance in timely matter. Its support to voice and video call will ensure that anyone near the victim can help as much as possible until a paramedic arrives and the presence of a chatbot in the application makes the application more useable and easier to interact with. It is also worth mentioning that the application that was developed is rather versatile and with minor modifications, the scope of Monqez can be completely changed.

In the future, some features can be added to help make the application more powerful such as notifying the users whenever they are in an accident-prone zone using machine learning so that they would be careful. Image processing could also be used to automatically verify Monqez's first-aid certificate and national ID images. Moreover, a social network could be implemented where friends and family will be notified in case any harm happens to the user.

References

[1] The Guardian. 2021. *Lack of first aid skills endangers up to 150,000 lives*. [online] Available at: https://www.theguardian.com/society/2010/apr/12/first-aid-skills-deaths .

[Last visited: January 2021]

[2] University of Manchester. 2016. Over half of deaths from injury. [online] Available at: https://www.manchester.ac.uk/discover/news/over-half-of-deaths-from-injury-could-be-prevented-if-public-knew-first-aid/ .

[Last visited: January 2021]

[3] Redcross.org.nz. 2021. *All available courses | First aid courses | New Zealand Red Cross*. [online] Available at: https://www.redcross.org.nz/first-aid/all-available-courses/

[Last visited: January 2021]

[4]2021 .مجلة هي . *الهلال الأحمر السعودي يدشن المرحلة الثانية من تطبيق المسعف الإلكتروني*. [online] Available at: https://cutt.ly/qxxIs1e/

[Last visited: January 2021]

[5] App Store. 2021. *eMedic*. [online] Available at: https://apps.apple.com/app/emedic/id349527488

[Last visited: January 2021]

[6] PulsePoint Respond. [online] Available at: https://play.google.com/store/apps/details?id=mobi.firedepartment&hl=en_US

[Last visited: January 2021]

[7] Asifny (Google Play) [online] Available at: https://play.google.com/store/apps/details?id=com.srca.apps.sos&hl=en&gl=US

[Last visited: January 2021]

[8] App Store. 2021. *Inaa'sh*. [online] Available at: https://apps.apple.com/eg/app/inaash/id993321301

[Last visited: January 2021]

[9] Wikihow.com. 2021. wikiHow: How-to instructions you can trust. [online] Available at: https://www.wikihow.com/Main-Page

[Last visited: June 2021]