**Project Report**

# Casebook For Lawyers

##### Submitted by

**Muhammad Faraz Khalil Mughal**

(FA17-BSSE-158)

**Muhammad Ahmad** (FA17-BSSE-168)

Session 2017-2021

##### Supervised by

**Miss Kiran Amjad**

**Department of Computer Science Lahore Garrison University Lahore**

Casebook

A project submitted to the Department of Software Engineering In

Partial Fulfillment of the Requirements for the Bachelor’s Degree in Software Engineering By

##### Muhammad Faraz Khalil Mughal

**&**

##### Muhammad Ahmad

**Internal Supervisor**

##### Miss Kiran Amjad

Senior Lecturer

Department of Software Engineering

**External Examiner** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Chairperson**

**Prof. Nadeem Asif \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Head of Department

Department of Software Engineering

# COPYRIGHTS

This is to certify that the project titled “**Case Book”** is the genuine work carried out by **Muhammad Faraz Khalil Mughal & Muhammad Ahmad,** understudy of BSSE of Computer Science Department, Lahore Garrison University, Lahore. During the scholarly year 2017-21, in fractional satisfaction of the necessities for the honor of the level of Bachelor of Science in Software Engineering and that the task has not framed the reason for the honor already of some other degree, confirmation, association or some other comparable title.

##### Muhammad Faraz Khalil Mughal

**Muhammad Ahmad**

# DECLARATION

This is to declare that the project entitled “Casebook” is an original work done by undersigned, in partial fulfillment of the requirements for the degree “Bachelor of Science in Software Engineering” at Software Engineering Department, Lahore Garrison University, Lahore.

All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

##### Muhammad Faraz Khalil Mughal

**Muhammad Ahmad**

# ACKNOWLEDGEMENTS

The warmest thank of my heart goes to Allah Almighty who provided all what was needed to reach the goal of project’s completion and the program, for the cause of which this project was taken under development. Throughout the entire development of the project, he was the one who took care of everything that proved to be negative for the project that would have diverted us from the track and the project would have never reached to its accomplishment. But in fact, he was the one to strengthen us in our most difficult times.

Registrar Brig Raheel Ashraf (Retd) can never be forgotten

as he proved to be a true mentor for the young leaders. We do appreciate our supervisor the Miss Kiran Amjad without whom it was really not possible to develop the project. We are also very thankful to the whole academic staff of the Software Engineering who were majorly or minorly involved in the development of the project.

# DEDICATION

We would like to dedicate this project to Allah Almighty the creator, strong pillar, my source of inspiration, wisdom, knowledge and understanding. He is only the one who has been mine source of strength and inspiration completely throughout this project and on whose wings we have only soared. Also, to our beloved brothers who have affected our project in every possible way by this quest. At last I would like to thanks to all those who affected the project positively, may Allah Almighty bless them all.

# Table of Content

[List of Tables xi](#_bookmark0)

[List of Figures xii](#_bookmark1)

[List of Abbreviation xiii](#_bookmark2)

[Abstract xiv](#_bookmark3)

[Chapter 1 1](#_bookmark4)

[Introduction 1](#_bookmark5)

* 1. [Background 1](#_bookmark6)
  2. [Objective 1](#_bookmark7)
  3. [Organizational Report 2](#_bookmark8)
  4. [Milestone Chart 2](#_bookmark9)

[Chapter 2 4](#_bookmark12)

[Problem Definition 4](#_bookmark13)

* 1. [Problem Statement 4](#_bookmark14)
  2. [Solution 4](#_bookmark15)
  3. [Mission Statement 5](#_bookmark16)
  4. [Vision Statement 5](#_bookmark17)

[Chapter 3 6](#_bookmark18)

[Software Requirement Specification 6](#_bookmark19)

* 1. [Introduction 6](#_bookmark20)
     1. [Purpose 6](#_bookmark21)
     2. [Intended Audience and Reading Suggestions 6](#_bookmark22)
     3. [Product Scope 7](#_bookmark23)
  2. [Overall Description 7](#_bookmark24)
     1. [Product Perspective 8](#_bookmark25)
     2. [Communication Interface 8](#_bookmark26)
     3. [Software Interface 9](#_bookmark27)
     4. [Hardware Interface 9](#_bookmark29)
     5. [User interface 9](#_bookmark30)
  3. [Product Functions 10](#_bookmark31)
  4. [User Classes and Characteristics 10](#_bookmark32)
  5. [Operating Environment 10](#_bookmark33)
  6. [Design and Implementation Constraints 10](#_bookmark34)
  7. [User Documentation 10](#_bookmark35)
     1. [Assumptions and Dependencies 10](#_bookmark36)
  8. [External Interface Requirements 11](#_bookmark37)
     1. [User Interfaces 11](#_bookmark38)
     2. [Hardware Interfaces 12](#_bookmark39)
     3. [Software Interfaces 12](#_bookmark40)
     4. [Communications Interfaces 12](#_bookmark41)
  9. [System Features 12](#_bookmark42)
     1. [User Registration / Sign Up: 12](#_bookmark43)
        1. [Stimulus/Response Sequences 13](#_bookmark44)
        2. [Functional Requirements 13](#_bookmark46)
     2. [Login 13](#_bookmark47)
        1. [Stimulus/Response Sequences 14](#_bookmark49)
        2. [Functional Requirements 14](#_bookmark51)
     3. [Case adding 14](#_bookmark52)
        1. [Stimulus/Response Sequences 15](#_bookmark54)
        2. [Functional Requirements 15](#_bookmark56)
     4. [Update case 15](#_bookmark57)
        1. [Stimulus/Response Sequences 16](#_bookmark59)
        2. [Functional Requirements 16](#_bookmark61)
     5. [Inbox 16](#_bookmark62)
        1. [Stimulus/Response Sequences 17](#_bookmark64)
        2. [Functional Requirements 17](#_bookmark66)
  10. [Other Nonfunctional Requirements 17](#_bookmark68)
      1. [Performance Requirements 17](#_bookmark69)
      2. [Safety Requirements 18](#_bookmark70)
      3. [Security Requirements 18](#_bookmark71)
      4. [Software Quality Attributes 18](#_bookmark72)
      5. [Business Rules 19](#_bookmark73)

[Chapter 4 20](#_bookmark74)

[Methodology 20](#_bookmark75)

* 1. [Introduction 20](#_bookmark76)
  2. [Planning 21](#_bookmark79)
  3. [Data Collection 21](#_bookmark80)
  4. [Hardware and software requirements 21](#_bookmark81)
     1. [Hardware requirements 21](#_bookmark82)
     2. [Software requirements 21](#_bookmark83)
  5. [Implementation 21](#_bookmark84)
  6. [Analysis 22](#_bookmark85)

[Chapter 5 23](#_bookmark86)

[Detailed Design and Architecture 23](#_bookmark87)

* 1. [System Architecture /Initial Design 23](#_bookmark88)
     1. [Architecture Design Approach 23](#_bookmark89)
        1. [Model View Controller 24](#_bookmark90)
     2. [Architecture Design 25](#_bookmark92)
     3. [Subsystem Architecture 25](#_bookmark94)
     4. [Functional Description 26](#_bookmark95)
  2. [Detailed System Design 26](#_bookmark96)
     1. [Sign Up 26](#_bookmark97)
        1. [Responsibilities 26](#_bookmark98)
        2. [Constraints 27](#_bookmark99)
     2. [Log In 27](#_bookmark100)
        1. [Responsibilities 27](#_bookmark101)
        2. [Constraints 27](#_bookmark102)
     3. [Home Page 27](#_bookmark103)
        1. [Inbox 28](#_bookmark104)
        2. [Sent Box 28](#_bookmark105)
        3. [Compose 28](#_bookmark106)
        4. [Profile 28](#_bookmark107)
        5. [Responsibility 28](#_bookmark108)
        6. [Constraints 29](#_bookmark109)
     4. [Uses/Interactions 29](#_bookmark110)
     5. [Resources 29](#_bookmark111)
     6. [Processing 29](#_bookmark112)
     7. [Interface/Exports 29](#_bookmark114)
     8. [Detailed Subsystem Design 29](#_bookmark113)
  3. [Use Case Diagram 30](#_bookmark115)
  4. [ER Diagram 31](#_bookmark117)
  5. [Architectural Diagram 32](#_bookmark119)
  6. [Activity Diagram 33](#_bookmark121)
  7. [Component Diagram 34](#_bookmark122)
  8. [Data Flow Diagram 35](#_bookmark124)
  9. [Sequence Diagram 36](#_bookmark126)
  10. [Class Diagram 37](#_bookmark128)
  11. [State Diagram 38](#_bookmark130)

[Chapter 6 39](#_bookmark132)

[Implementation and Testing 39](#_bookmark133)

* 1. [Tools 39](#_bookmark134)
  2. [Testing Methods 39](#_bookmark135)
     1. [Test plan for Registration Activity 39](#_bookmark136)
     2. [Test plan for Login Activity 40](#_bookmark137)
     3. [Test plan for Logout Activity 41](#_bookmark138)
     4. [Test plan for Home Page Activity 42](#_bookmark139)
  3. [Testing 43](#_bookmark140)
     1. [Purpose of Testing 43](#_bookmark141)
     2. [Testing the Registration and Home page activities 43](#_bookmark142)
        1. [Black Box Testing 43](#_bookmark143)
        2. [User Testing 43](#_bookmark144)
        3. [Unit Testing 44](#_bookmark145)
        4. [Integration Testing 44](#_bookmark146)

[Chapter 7 44](#_bookmark147)

[Results and Discussion 45](#_bookmark149)

* 1. [Results and Test Evaluation 45](#_bookmark148)
  2. [Final User Interface and Working 45](#_bookmark150)

[Login Page 47](#_bookmark153)

[Chapter 8 59](#_bookmark166)

[Conclusion and Future Work 59](#_bookmark167)

* 1. [Future Work 59](#_bookmark168)
  2. [Conclusion 59](#_bookmark169)

[References 60](#_bookmark170)

# List of Tables

**[Table 1- Milestone Chart 1](#_bookmark10)** [2](#_bookmark10)

**[Table 2- Milestone Chart 2](#_bookmark11)** [3](#_bookmark11)

**[Table 3- Software Interface](#_bookmark28)** [9](#_bookmark28)

**[Table 4- Use Case 1](#_bookmark45)** [13](#_bookmark45)

**[Table 5- Functional Requirement 1](#_bookmark48)** [13](#_bookmark48)

**[Table 6- Use Case 2](#_bookmark50)** [14](#_bookmark50)

**[Table 7- Functional Requirement 2](#_bookmark53)** [14](#_bookmark53)

**[Table 8- Use Case 3](#_bookmark55)** [15](#_bookmark55)

**[Table 9- Functional Requirement 3](#_bookmark58)** [15](#_bookmark58)

**[Table 10- Use Case 4](#_bookmark60)** [16](#_bookmark60)

**[Table 11- Functional Requirement 4](#_bookmark63)** [16](#_bookmark63)

**[Table 12- Use Case 5](#_bookmark65)** [17](#_bookmark65)

**[Table 13- Functional Requirement 5](#_bookmark67)** [17](#_bookmark67)

# List of Figures

**Figure 1-Use Case1** 13

**Figure 2- Use Case2** 14

**Figure 3- Use Case3** 15

**Figure 4- Use Case4** 16

**Figure 5- Use Case5** 17

**[Figure 6- Methodology](#_bookmark77)** [20](#_bookmark77)

**[Figure 7- Development Steps](#_bookmark78)** [20](#_bookmark78)

**[Figure 8- MVC](#_bookmark91)** [24](#_bookmark91)

**[Figure 9- Architecture Design](#_bookmark93)** [25](#_bookmark93)

**[Figure 10- Use Case Diagram](#_bookmark116)** [30](#_bookmark116)

**[Figure 11- ER Diagram](#_bookmark118)** [31](#_bookmark118)

**[Figure 12- System Architecture](#_bookmark120)** [32](#_bookmark120)

**Figure 13- Activity Diagram** 33

**[Figure 14- Component Diagram](#_bookmark123)** [34](#_bookmark123)

**[Figure 15- Data Flow Diagram](#_bookmark125)** [35](#_bookmark125)

**[Figure 16- Sequence Diagram](#_bookmark127)** [36](#_bookmark127)

**[Figure 17-Class Diagram](#_bookmark129)** [37](#_bookmark129)

**[Figure 18- State Diagram](#_bookmark131)** [38](#_bookmark131)

# List of Abbreviation

|  |  |  |
| --- | --- | --- |
| **Short Form** | **Complete Form** | **Description** |
| API | Application Programming Interface | It is a software intermediary that allows two applications to talk to each other. |
| JS | Java Script |  |
| SDK | Software Development Kit |  |
| App | Application |  |
| User |  | A person used the application. |
| Android |  | A mobile device operating system developed by Google Inc.  Helps the reader identify key information. |
| Developers |  | A person who build something. |
| DB | Database |  |
| CPU | Central Processing Unit |  |
| MB | Mega Byte |  |
| OS | Operating system |  |
| GUI | Graphical User Interface |  |
| UI | User Interface | UI is the series of screens, pages, and visual elements. |
| APK | Android Application Kit |  |
| SMTP | Simple Mail Transfer Protocol | The Simple Mail Transfer Protocol is an internet standard communication protocol for electronic mail transmission. |
| FR | Functional Requirements |  |
| UC | Use Case |  |
| ER diagram | Entity Relation Diagram |  |

# Abstract

Casebook is a mobile application based portal for lawyers and clients, to provide better communication between the two and to make the court process efficient. Lawyers wont have to deal with the manual filesystem. Clients will be completely updated with their cases, app will be covering cases repository, client management, courts management, emails, sms, cases reminders, adjourned dates and authorization.

The main purpose of this software is to speed-up and digitalize the work of judicial system. The software will be used by lawyer’s and their clients to stay updated and to communicate better. It is aimed on replacing the file system of the courts.

The app will be cross-platform working on both android and ios operating systems. The development is in React Native. It is a library of javascript to create dynamic interactive cross-platform mobile apps. The backend and database will be Firebase.

# Chapter 1

# Introduction

## Background

Right now our judicial system is based on the manual filing system, we are initiating a step toward digitizing it, right now we are offering a solution for lawyers through our app. The Lawyers can easily manage all their cases in a smart way. They can keep their clients updated. In their current approach they need to keep track of all their files and documents which can get a bit messed up or the could lose sensitive information about their cases. And on the other hand clients also faces difficulty on keeping up with progress of their cases.

## Objective

The main purpose of this software is to speed-up and digitalize the work of judicial system. The software will be used by lawyer’s and their clients to stay updated and to communicate better. It is aimed on replacing the file system of the courts.

Casebook is Lawyer’s portal app used to digitalize the lawyer’s work and cases and help provide better communication to the client’s with their lawyers. It is a new product and in present time there is no product like this.

## Problem Statement

In most of the cases; people with low vision has overcome this loss by using an assistance of a third person for help to compose an email. This process has multiple constraints like time consumption, availability of third person, sensitivity of email content. No doubt they have the option of using different mailing platforms by using some features available in android phones for blinds but these features are not completely effective and easy for them to use. They can use it in their phones for basic tasks but there is no proper facility available for communication or for sending mails. So after the research and discussion with our target users [visually impaired] we have proposed this system.

we are offering a solution for lawyers through our app. The Lawyers can easily manage all their cases in a smart way. They can keep their clients updated. In their current approach they need to keep track of all their files and documents which can get a bit messed up or the could lose sensitive information about their cases. And on the other hand clients also faces difficulty on keeping up with progress of their cases.

No doubt they have the option of using different mailing platforms by using some features available in android phones for blinds but these features are not completely effective and easy for them to use.

They can use it in their phones for basic tasks but there is no proper facility available for communication or for sending mails. So after the research and discussion with our target users [visually impaired] we have proposed this system

## Organizational Report

In this chapter, we have discussed project without technical details brief overview of project aim. About what is the product is and how it will be helpful for people or students. In upcoming chapters, we will discuss:

* + - Problem Identification and Solution of that problem
    - Software Requirement Specification
    - Methodology
    - System Architectural Design
    - Implementation of Testing
    - Software Quality
    - Result and Discussion
    - Conclusion and Future Work

## Milestone Chart

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Activity** | **Semester7 Weeks** | | | | | | | | | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| Project Proposal |  |  |  |  |  |  |  |  | M I  D |  |  |  |  |  |  |  |  | F I  N |
| Proposal Defense |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | A L |
| Setting up Environment  on PC |  |  |  |  |  |  |  |  | E X |  |  |  |  |  |  |  |  | E X |
| Gathering  Data |  |  |  |  |  |  |  |  | A  M |  |  |  |  |  |  |  |  | A  M |
| Code Part-1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Table 1- Milestone Chart 1**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Activity** | **Semester 8 Weeks** | | | | | | | | | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| Code Part-2 |  |  |  |  |  |  |  |  | M I  D |  |  |  |  |  |  |  |  | F I  N |
| Implementation Of complete  code |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | A L |
| Testing and  Debugging |  |  |  |  |  |  |  |  | E  X |  |  |  |  |  |  |  |  | E  X |
| Finalization |  |  |  |  |  |  |  |  | A  M |  |  |  |  |  |  |  |  | A  M |

**Table 2- Milestone Chart 2**

# Chapter 2

# Problem Definition

## Problem Statement

In most of the cases; people with low vision has overcome this loss by using an assistance of a third person for help to compose an email. This process has multiple constraints like time consumption, availability of third person, sensitivity of email content. No doubt they have the option of using different mailing platforms by using some features available in android phones for blinds but these features are not completely effective and easy for them to use. They can use it in their phones for basic tasks but there is no proper facility available for communication or for sending mails. So after the research and discussion with our target users [visually impaired] we have proposed this system.

## Solution

we are offering a solution for lawyers through our app. The Lawyers can easily manage all their cases in a smart way. They can keep their clients updated. In their current approach they need to keep track of all their files and documents which can get a bit messed up or the could lose sensitive information about their cases. And on the other hand clients also faces difficulty on keeping up with progress of their cases.

No doubt they have the option of using different mailing platforms by using some features available in android phones for blinds but these features are not completely effective and easy for them to use. They can use it in their phones for basic tasks but there is no proper facility available for communication or for sending mails. So after the research and discussion with our target users [visually impaired] we have proposed this system.

## 2.3 Objective

The main purpose of this software is to speed-up and digitalize the work of judicial system. The software will be used by lawyer’s and their clients to stay updated and to communicate better. It is aimed on replacing the file system of the courts.

Casebook is Lawyer’s portal app used to digitalize the lawyer’s work and cases and help provide better communication to the client’s with their lawyers. It is a new product and in present time there is no product like this.

## 2.4 Mission Statement

This software is to speed-up and digitalize the work of judicial system. The software will be used by lawyer’s and their clients to stay updated and to communicate better. It is aimed on replacing the file system of the courts.

## 2.5 Vision Statement

The contribution made by this product enabled the visually impaired and normal person to send receive voice based E-mail message in efficient and time saving environment by just using their android phones. [4]

# Chapter 3

# Software Requirement Specification

## Introduction

Right now our judicial system is based on the manual filing system, we are initiating a step toward digitalizing it, right now we are offering a solution for lawyers through our app. The Lawyers can easily manage all their cases in a smart way. They can keep their clients updated. In their current approach they need to keep track of all their files and documents which can get a bit messed up or the could lose sensitive information about their cases. And on the other hand clients also faces difficulty on keeping up with progress of their cases.

### Purpose

Primarily this app is made for the lawyers to manage the day to day works. The lawyers can add all their cases to a single platform online cloud based platform. Additionally the app also consist a client side where a client can mange all his cases with every lawyers with a simple signup or lawyer can let any client keep tab on his case without authentication with a single token based system.

* + 1. **Intended Audience and Reading Suggestions**

The SRS document is basically written for the general audience, this document is intended for individuals, developers that directly involved in the mobile and web app development.

This included:

•Software Developers,

•Project Consultants,

•All the Stakeholders.

Audience are not bound to read this SRS sequentially, they can also directly jump to the relevant section. It contains a brief description of every part of the project or product. It has the following details about the project; • Introduction

•Overall Description

•External Interface Requirements

•System Features

•Other Non-Functional Requirements

We’ll suggest the reader’s that they should be read this document sequentially but this is

understandable because every part of the document is explained deliberately so, readers can

also read only a particular section of the document

* + 1. **Product Scope**

Our target is professional who suffer a lot, software is to speed-up and digitalize the work of judicial system. The software will be used by lawyer’s and their clients to stay updated and to communicate better. It is aimed on replacing the file system of the courts.

## Overall Description

* + 1. **Product Perspective:**

Casebook is Lawyer’s portal app used to digitalize the lawyer’s work and cases and help provide better communication to the client’s with their lawyers. It is a new product and in present time there is no product like this.

* + 1. **Product Functions:**

Casebook has the following functions:- Lawyer Signup

Client Signup Lawyer Login Client Login

Case Updates without Authorization Case Management

Client Management Court Management Case Events Reminders Document Scanner Emails

Sms

Details of all the functions is described later in the document.

Casebook is Lawyer’s portal app used to digitalize the lawyer’s work and cases and help provide better communication to the client’s with their lawyers. It is a new product and in present time there is no product like this.Casebook is Lawyer’s portal app used to digitalize the lawyer’s work and cases and help provide better communication to the client’s with their lawyers. It is a new product and in present time there is no product like this.

we are offering a solution for lawyers through our app. The Lawyers can easily manage all their cases in a smart way. They can keep their clients updated. In their current approach they need to keep track of all their files and documents which can get a bit messed up or the could lose sensitive information about their cases. And on the other hand clients also faces difficulty on keeping up with progress of their cases.

No doubt they have the option of using different mailing platforms by using some features available in android phones for blinds but these features are not completely effective and easy for them to use. They can use it in their phones for basic tasks but there is no proper facility available for communication or for sending mails. So after the research and discussion with our target users [visually impaired] we have proposed this system

#### Product Perspective

Casebook is Lawyer’s portal app used to digitalize the lawyer’s work and cases and help provide better communication to the client’s with their lawyers. It is a new product and in present time there is no product like this.

#### Product Functions

Casebook has the following functions:- Lawyer Signup

Client Signup Lawyer Login Client Login

Case Updates without Authorization

Case Management Client Management Court Management Case Events Reminders Document Scanner Emails

Sms

Details of all the functions is described later in the document.

##### Hardware Requirements:

These are the hardware interfaces that support the use of

Casebook:-

•Any Android Mobile Phone not older than Android 4.0.

•Any Apple Mobile Phone

•Any system that has Web Browser  
Any Android Mobile Phone not older than Android 4.0.

•Any Apple Mobile Phone

•Any system that has Web Browser.

#### Software Interface

Following are the softwares used for Casebook.

•Android OS, iOS

•React Native, MongoDB, Node JS

#### Software Interface

Following are the softwares used for Casebook.

•Android OS, iOS

•React Native, MongoDB, Node JS

**Communications Interfaces**

This project is going to support Android as well as iOS. You can also communicate with admin and send data so your data and chats will be secure.

**Hardware Interface**

FTP is the best to send files over internet. The data can be send to the service provider only if the user’s device has an active internet connection. For service provider there should also be an active internet connection.

##### Hardware Requirements:

These are the hardware interfaces that support the use of

Casebook:-

•Any Android Mobile Phone not older than Android 4.0.

•Any Apple Mobile Phone

•Any system that has Web Browser.

#### User interface

The user interfaces will be divided into two portions that you are user or service provider. Both have separate modules. After entering into their specific module, they can access their data through their accounts and can use the functions of the product. User interface consists of login and sign-up screen and then they can enter their account and at the main screen they have the functionalities to send their documents to the service provider. The chat option is also for the user.

The Admin panel for those who provide the service, they have their account and can login to it and see the request send by the users and then they can print their documents on user request and send responses through their account.

FTP is the best to send files over internet. The data can be send to the service provider only if the user’s device has an active internet connection. For service provider there should also be an active internet connection.

These are the hardware interfaces that support the use of casebook:-

•Any Android Mobile Phone not older than Android 4.0.

•Any Apple Mobile Phone

•Any system that has Web Browser.

## Product Functions

Casebook has the following functions:-

Lawyer Signup, Client Signup, Lawyer Login, Client Login, Case Updates without Authorization, Case Management, Client Management, Court Management, Case Events, Reminders, Document Scanner, Emails, Sms

Details of all the functions is described later in the document.

## User Classes and Characteristics

This product has two main users, one is who is providing the services and other who wants the services. Both have different view of the product, user who wants services only has access to its account and can get updates of the service provider. User can just read all the status. User can also communicate.

On the other hand the service provider has the access to read and edit the data. He can change the status of the user and status of document.

## Operating Environment

The product is based on mobile and web application. Web application can be access anywhere by web browser. Mobile application will be cross platform it means the mobile application can work on Android and IOS. For android and IOS we do not need to code seperate for both application because the beauty of React Native is that you can run single-base code and use in IOS and Android application.

So the product can be used either through Web, Android or IOS.

## Design and Implementation Constraints

The information about the document will be stored against each document. Firebase will be used as database. The application can be access anytime. Any system that has internet and web browser can be used to get access to the aplication. All users and service provider must have their account credentials to access their respective account.  
The user interfaces will be divided into two portions that you are user or service provider. Both have separate modules. After entering into their specific module, they can access their data through their accounts and can use the functions of the product. User interface consists of login and sign-up screen and then they can enter their account and at the main screen they have the functionalities to send their documents to the service provider. The chat option is also for the user.

The Admin panel for those who provide the service, they have their account and can login to it and see the request send by the users and then they can print their documents on user request and send responses through their account.

## User Documentation

The user documentation explains the use of application that how can you create your account as a user and how to use the services of Casebook. All the functionalities will be explained in details for better understanding for users so that they can use the application with full knowledge of it.

The whole documentation of the can be found on the playstore and appstore for help in usage of the application and the application itself also contains short tutorials for its new users to make it easier to get 100% benefit out of it.

## Assumptions and Dependencies

The product need the following third party products:-

Firebase CloudFirestore is the database to store data. React and React Native to develop the product.

Node JS will be run-time environment. The application will depend entirely on internet, because through internet the server receive their request and send their response. So these are the dependencies of the product.

## External Interface Requirements

#### User Interfaces

The user interfaces will be divided into two portions that you are user or service provider. Both have separate modules. After entering into their specific module, they can access their data through their accounts and can use the functions of the product. User interface consists of login and sign-up screen and then they can enter their account and at the main screen they have the functionalities to send their documents to the service provider. The chat option is also for the user.  
  
The Admin panel for those who provide the service, they have their account and can login to it and see the request send by the users and then they can print their documents on user request and send responses through their account.

After collecting and executing the data, this is the last stage of development methodology and first stage of Testing. In this phase app is analyzed and bugs will be removed. After developing the app structure we first analyze the app by own and then analyze by some random users. The remarks that we got from random users are the appreciating but few bugs are highlighted in phase that we remove after. Then we analyze our database structure by adding random entries in database.

User must be able to register for the application through email address. After installing the application, user will register themselves by adding email address, password username and profile picture. If user skips this step, application will be close.

#### Hardware Interfaces

FTP is the best to send files over internet. The data can be send to the service provider only if the user’s device has an active internet connection. For service provider there should also be an active internet connection.

These are the hardware interfaces that support the use of casebook:-

•Any Android Mobile Phone not older than Android 4.0.

•Any Apple Mobile Phone

•Any system that has Web Browser.

#### Software Interfaces

Following are the softwares used for Casebook

•Android OS, iOS

•React Native, MongoDB, Node JS

#### Communications Interfaces

This project is going to support Android as well as iOS. You can also communicate with admin and send data so your data and chats will be secure.

## System Features

Features for Casebook are described according to their priority ratings.

#### User Registration / Sign Up:

User must be able to register for the application through email address. After installing the application, user will register themselves by adding email address, password username and profile picture. If user skips this step, application will be close.

**Priority:** Priority of this feature is High.

The remarks that we got from random users are the appreciating but few bugs are highlighted in phase that we remove after. Then we analyze our database structure by adding random entries in database.

The login screen allows registered users to login to the site to access all of the features that their account gives them access to. If they type email address and password and click

login the user’s credentials are validated and if correct, they are logged into the application

FTP is the best to send files over internet. The data can be send to the service provider only if the user’s device has an active internet connection. For service provider there should also be an active internet connection.

**12 |** P a g e

#### Stimulus/Response Sequences

|  |  |
| --- | --- |
| **Use case number** | UC-1 |
| **Use case name** | Registration |
| **Actor Name** | User |
| **Description** | To register yourself in the application |
| **Preconditions** | Application must be installed in phone |
| **Post conditions** | User successfully registered |

**Table 4- Use Case 1**

User, admin

Register

**Figure 1-Use Case1**

#### Functional Requirements

|  |  |
| --- | --- |
| **ID** | **FR1** |
| Title | Registration |
| Description | To register yourself for application |
| Input | Name, email, password, picture |
| Output | Registered |

* + 1. **Login**

**Table 5- Functional Requirement 1**

The login screen allows registered users to login to the site to access all of the features that their account gives them access to. If they type email address and password and click

login the user’s credentials are validated and if correct, they are logged into the application.

**Priority:** Priority is high.

#### Stimulus/Response Sequences

|  |  |
| --- | --- |
| **Use case number** | UC-2 |
| **Use case name** | login |
| **Actor Name** | User, Admin |
| **Description** | User login for main menu |
| **Preconditions** | Must be registered |
| **Post conditions** | Login Successful |

**Table 6- Use Case 2**

User

Login

**Figure 2- Use Case2**

#### Functional Requirements

|  |  |
| --- | --- |
| **ID** | **FR2** |
| Title | login |
| Description | To use application user must have to login |
| Input | email, password |
| Output | Home page will appear |
| Dependency | FR21 |

* + 1. **Compose Mail:**

**Table 7- Functional Requirement 2**

After logging successfully into the app, user will check their all cases in it

#### Stimulus/Response Sequences

|  |  |
| --- | --- |
| **Use case number** | UC-3 |
| **Use case name** | Adding Cases |
| **Actor Name** | User |
| **Description** | Lawyers are adding their cases |
| **Preconditions** | User must be login |
| **Post conditions** | Case added successfuly and showing in draft |

**Table 8- Use Case 3**

User

Adding Case

**Figure 3- Use Case3**

#### Functional Requirements

|  |  |
| --- | --- |
| **ID** | **FR3** |
| Title | Updating the case |
| Description | To update the date of case |
| Input | next date of case |
| Output | Case updated |
| Dependency | FR2 |

* + 1. **Sent Box**

**Table 9- Functional Requirement 3**

User can update his case accordingly and can add more cases if he want.He can sent mails to clients

And aware his clients about there cases

#### Stimulus/Response Sequences

|  |  |
| --- | --- |
| **Use case number** | UC-4 |
| **Use case name** | Mail the client |
| **Actor Name** | User |
| **Description** | User can check sent messages. |
| **Preconditions** | Message should be sent successfully. |
| **Post conditions** | Messages read or listen Successfully. |

**Table 10- Use Case 4**

User

Sent Box

**Figure 4- Use Case4**

#### Functional Requirements

|  |  |
| --- | --- |
| **ID** | **FR4** |
| Title | Sent Box |
| Description | Sent message list |
| Input | Press on the message whom you want to check. |
| Output | Message Checked. |
| Dependency | FR3 |

* + 1. **Inbox**

**Table 11- Functional Requirement 4**

User can check their received mails by swiping right the screen from home page. And after entering into the inbox user can open the mail by simple click and listen the sent message. Same like sent box, Inbox also has the option of mail deletion.

#### Stimulus/Response Sequences

|  |  |
| --- | --- |
| **Use case number** | UC-5 |
| **Use case name** | Sent Box |
| **Actor Name** | User |
| **Description** | User can check received messages. |
| **Preconditions** | Should be login with valid mail address. |
| **Post conditions** | Messages read or listen Successfully. |

**Table 12- Use Case 5**

User

Inbox

**Figure 5- Use Case5**

#### Functional Requirements

|  |  |
| --- | --- |
| **ID** | **FR5** |
| Title | Inbox |
| Description | Received message list |
| Input | Click on the message whom you want to check. |
| Output | Message Checked. |
| Dependency | FR2 |

**Table 13- Functional Requirement 5**

## Other Nonfunctional Requirements

#### Performance Requirements

We will try to improve performance of Casebook by using different techniques to improve speed, throughput etc. Techniques like normalization to eliminate data redundancy from database. Database Offline Persistance to make it work on cache.

#### Safety Requirements

If there are some damages like database failure or something else so there will be a backup system of all the important information.

#### Security Requirements

When it comes to security so you have to think about how can you make your data secure it is all about database where you store your information. So vendors must choose their database partner carefully to make their system secure.

#### Software Quality Attributes

##### Accessibility:

Casebook application is easily accessible for users; its development structure is just like a simple mobile app and its features are similar to traditional applications

**Efficiency:**

Casebook is very efficient if the user have some know how about mobile applications

##### Performance:

Performance of this application is based on its efficiency, because if user gives correct commands then it works properly and its performance rate increases.

##### Scalability:

We are trying to develop this application in a scalable environment. Whenever with the passage of time its user rate increases it will not collapse and work proper and efficiently.

##### Availability:

Casebook will be accessible every minute of every day if you just have an android phone and good internet connection.

##### Portability:

This application is portable like an android phone; you can use it anywhere/anyplace at any time. It is a mobile application.

## Planning

In the First step we need to investigate all the data and necessities of equipment and programming, which are utilized in the venture. Information Collection, necessary equipment and programming are its two primary components.

#### Business Rules

•Lawyers cannot delete any case, the cases will be just soft deleted.

•Lawyers can provide secret token to their user’s through email or sms, the token can be used to get updates about cases without Client’s login.

•Clients need to create account to get access to all their data.

•A particular case information can only be viewed by case’s lawyer and whoever the lawyers want to allow access.

•All data and document will be secure in google firebase and can only be viewed through authorization.

**Chapter 4**

**Methodology**

## Introduction

For accomplishing our goals that shows us the ideal outcome we utilize this procedure. For assessing this task, we utilized the SDLC system, which contains three significant advances, Implementing, Planning and Analysis, as appeared in figure 4.1.

**Planning**

**Analysis**

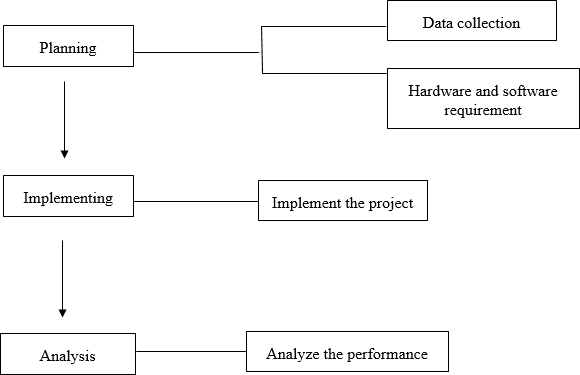
**Design**

**Implementation**

**Maintenance**

**Figure 6- Methodology**

Following are the major three steps shows in figure 4.2. That’s starts from planning, implementing and testing. Through these three steps we accomplish our project and get the final results.



**Figure 7- Development Steps**

## Planning

In the First step we need to investigate all the data and necessities of equipment and programming, which are utilized in the venture. Information Collection, necessary equipment and programming are its two primary components.

## Data Collection

Information Collection is the most significant piece of any undertaking. In first stage we assemble the data from the Internet and Books.

## Hardware and software requirements

The hardware and software requirements are as follow.

#### Hardware requirements

* + - * Laptop, Pc, Smartphone

#### Software requirements

* + - * Java Development Kit, JDK
      * Visual studio code
      * Firebase

## Implementation

After planning and collecting information, In this stage the collected information is executed by utilizing the product prerequisites to build up the Android Application. After viable structure the Android App needs to accomplish the primary target we pushed ahead to conclusive stage.

This framework design is completely depended to three fundamental modules that has been likewise talked about at above. These modules are probably going to be worked by joining their functionalities and the yield created by top dimension interfaces. All the functionalities are interlinked with each other in the architecture and working on their specific time.

The first functionality that comes in architecture is the sign up activity that come immediate after the user open the app and then after sign up the second activity is login and its both are highly interlink. And after successful completion of first two the third activity is the home page and it’s totally depend on the first two activities of architecture.

## Analysis

After collecting and executing the data, this is the last stage of development methodology and first stage of Testing. In this phase app is analyzed and bugs will be removed. After developing the app structure we first analyze the app by own and then analyze by some random users. The remarks that we got from random users are the appreciating but few bugs are highlighted in phase that we remove after. Then we analyze our database structure by adding random entries in database.

# Chapter 5

# Detailed Design and Architecture

## System Architecture /Initial Design

This framework design is completely depended to three fundamental modules that has been likewise talked about at above. These modules are probably going to be worked by joining their functionalities and the yield created by top dimension interfaces. All the functionalities are interlinked with each other in the architecture and working on their specific time.

The first functionality that comes in architecture is the sign up activity that come immediate after the user open the app and then after sign up the second activity is login and its both are highly interlink. And after successful completion of first two the third activity is the home page and it’s totally depend on the first two activities of architecture.

#### Architecture Design Approach

The methodology utilized in this entire framework is absolutely relies upon the functionalities of the parts that work freely and give the availability between the interfaces that are talked about above. The segments on which this methodology of complete framework depends are following:

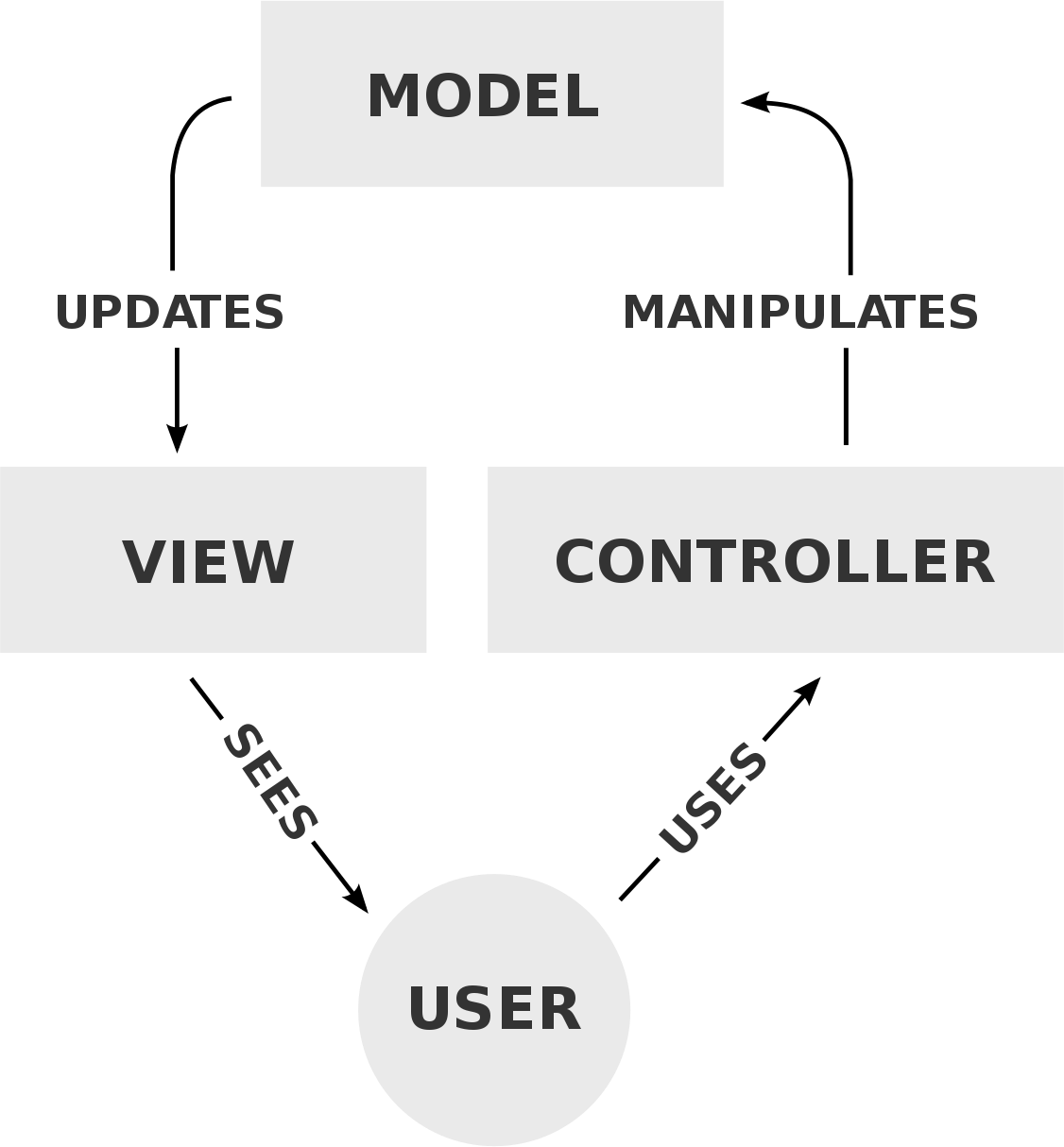
Android based Application is designed to control the management activities of the Voice Mail Service and designed using the approach of the following architecture that consist on

* + - * Model

Component corresponds to all the data-related logic that the user works with

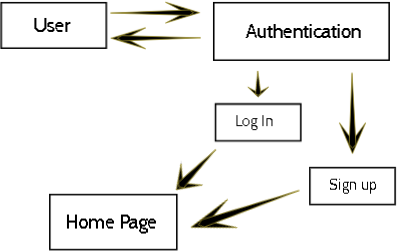
* + - * View  
        This components is use for all UI logic of application.
      * Controller  
        It acts as interlink between Model and view
      * User  
        They are the person who is going to use the application

#### 5.1.1.1Model View Controller



**Figure 8- MVC**

#### Architecture Design



**Figure 9- Architecture Design**

Above mentioned diagram clarify the basic structure of application, in the system there are three main activities that are completely linked with each other starting from user activity. User can enter into the app with proper authentication of their credentials and if is will recognize by system then he/she has access of further application. In this application we have three major modules three subsystem in Home module:

1. Sign Up
2. Login
3. Home page
   * Lawyer
   * Client
4. Sub-activities of tastks

Login is a subsystem of Registration if registration is done then user go to the login page and access it. Further functionalities of application is based on authentication from both user’s end (Lawyer and client). Both will communicate effectively if they registered successfully.

#### Subsystem Architecture

Subsystem architecture is described with functions description. This system is interconnected with each other to perform the functions activities and if one of the following

is spoiled or unlinked with the other then the functions are not going to perform in a well- ordered manner There is Login user page if the user have sign in firstly then the user enter in the list of web viewer by giving its email and password.

#### Functional Description

The system architecture is divided into the following architecture that are linked together to perform each functionality of every activity.

* + - * User need to sign up for using the application.
      * After signing up the user enter into home page, and login in the application again if they use logout application.
      * User has to login app with valid email address.
      * Lawyer and clients can manage their cases

## Detailed System Design

#### Sign Up

Lawyers and clients can make account will create their accounts and can login to their respective accounts after entering validate information.

#### Responsibilities

•Create lawyer’s / client’s account.

•Authorization

##### Classification

The Lawyers can lost track of their cases. They don’t have any way to keep track of all their clients and courts. Everything is file-based.

Clients don’t have a suitable way to get update about the cases.

##### Definition

This feature will help users to manage and get access to their personal data provide security and and also make the communication with the database very smooth.

#### Constraints

* + - * + The user need a valid email address to login and get verified account to start using the app.

#### Log In

After Sign up, the login activity is second higher priority function. Only registered users can use this functionality and user doesn’t need to login their account again and again when they want to use app because we use the one time login service in this but if user will log out their account then they need to log in again for accessing the application. In this functionality two inputs are required from the user, one is email and the other is password and same facility of speech to text also available here for user’s ease of access. User will go to home page after adding valid email and password. After pressing Login button user will enter into home and have access to their account data and functionalities.

#### Responsibilities

* + - * + Authenticate user from database.
        + Give account access to user.

#### Constraints

* + - * + Without adding all credentials, user cannot signup.

#### Case Management

After successful sign up and login, user enters into the home page and in this page user will

manage his cases

##### Classification

Lawyers can sign-up to the app and start adding their current cases. All the cases will be shown to him in chronological order. The cases can be disposed or adjourned to a new date if needed. Complete history of all the cases will be saved in the application. Lawyer will be notified about his upcoming dates. Each case will have a unique code through which the lawyer can give access to client about his cases in the app.

##### Definition

Primarily this app is made for the lawyers to manage the day to day works. The lawyers can add all their cases to a single platform online cloud based platform. Additionally the app also consist a client side where a client can mange all his cases with every lawyers with a simple signup or lawyer can let any client keep tab on his case without authentication with a single token based system.

##### Responsibilities

•Manage all cases

•Upload data of cases

•Communicate with client

•Give managed access to different users

•Adjourn Cases

•Inform Users about upcoming cases

•Dispose cases

•Email, SMS or Call to the clients

#### Constraints

The lawyer will need to integrate google calendar into the app for getting properly updated through google events.

#### Composition

This is the core feature of the app, whole app is built around it.

Though this app we are trying to end the Manual file system for the lawyers and provide him with the managed cloud access to perform his day to day works.

##### Uses

As this functionality is a core feature of the app, its is bound with all the other subsystems Client view is also intact with the same lawyers view and the users without login are also seeing the same view as the lawyers and client.

Every user’s access the managed and authorized in a certain way.

#### Resources

•Firebase Authentication is used for primary authentication purposes.

•Firebase CloudFirestore for database

•React Native for Development

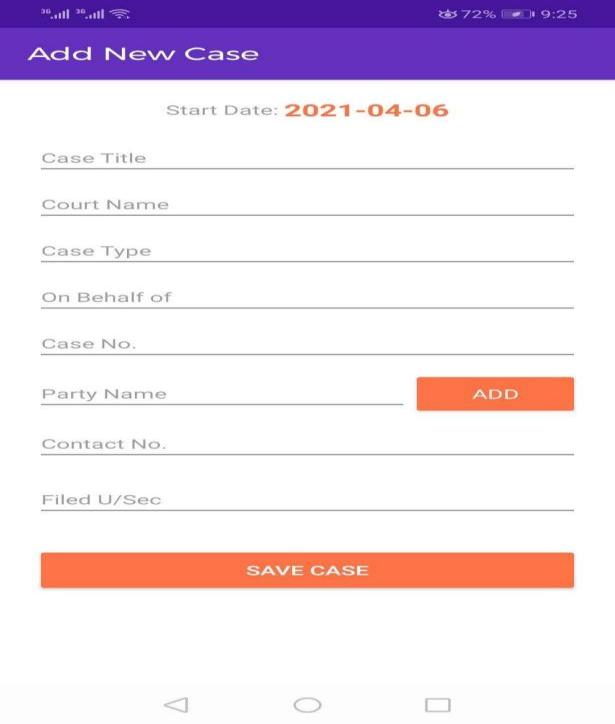
•Expo CLI for development environemnt in windows machine.

•Redux for state handling

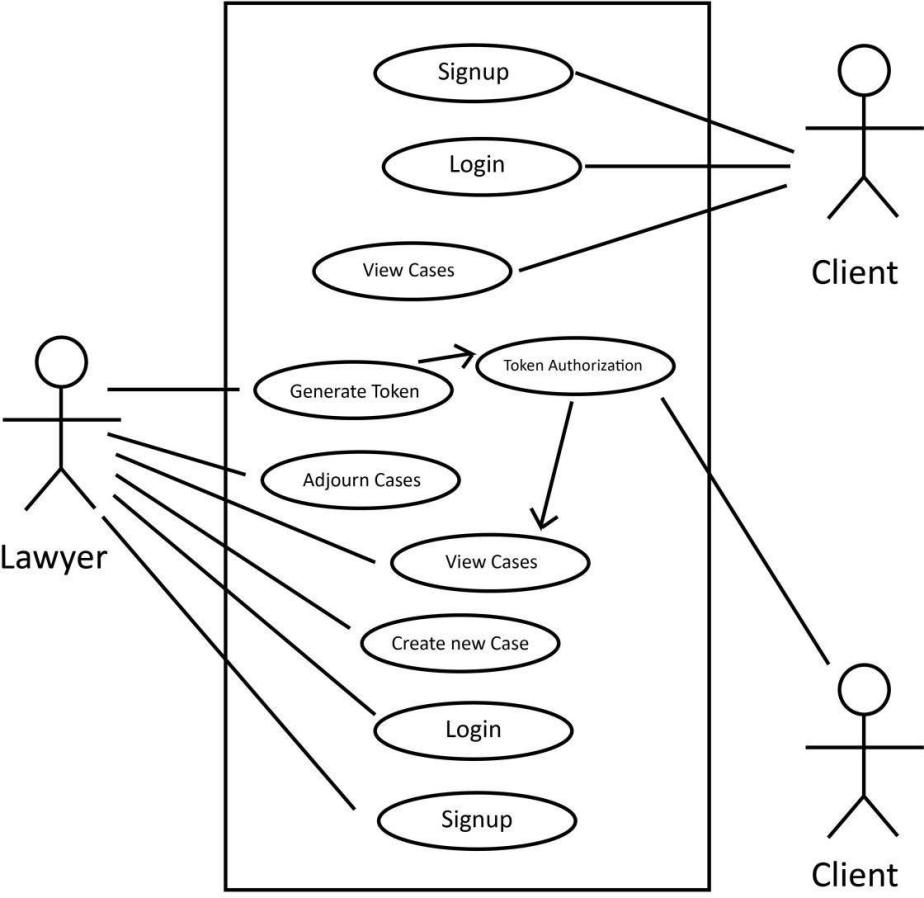
#### Processing

Lawyer needs to fill the forn to create a new case.

**5.2.7 Interface/Exports**

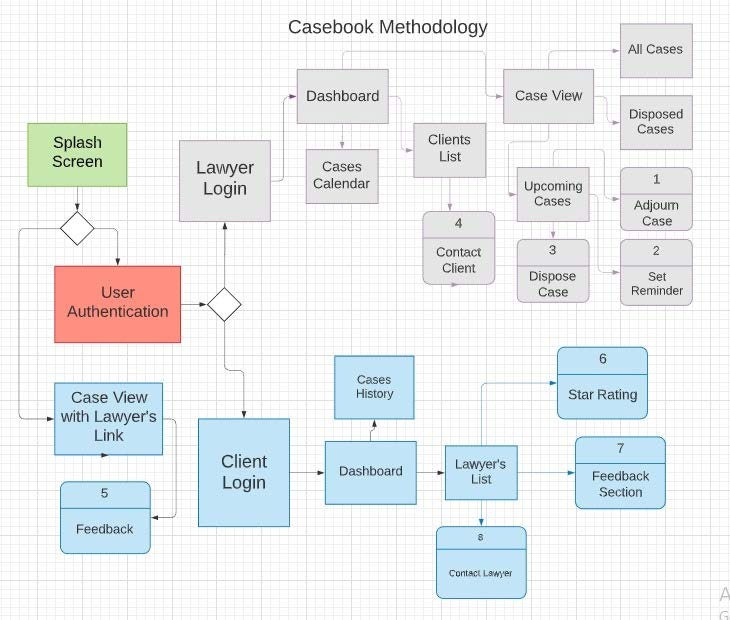


**Use Case Diagram**



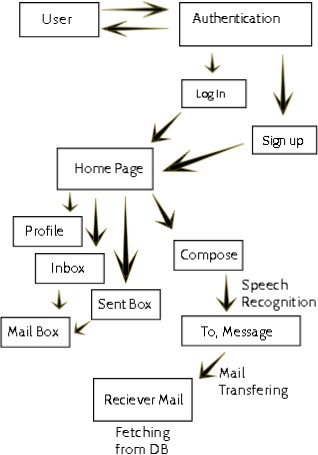
**Figure 10- Use Case Diagram**

## ER Diagram



**Figure 11- ER Diagram**

## Architectural Diagram



**Figure 12- System Architecture**

## Activity Diagram

Start

Speak username and password

Speech to Text Using API

DB

NO



Check validity with DB

Valid

Yes

Home



Goto

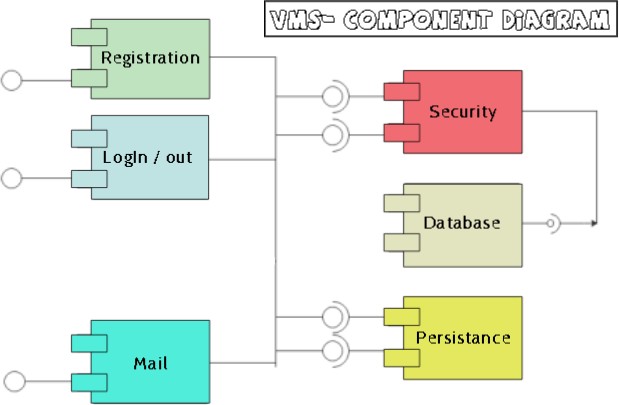
Compose

Inbox

Sent Mail

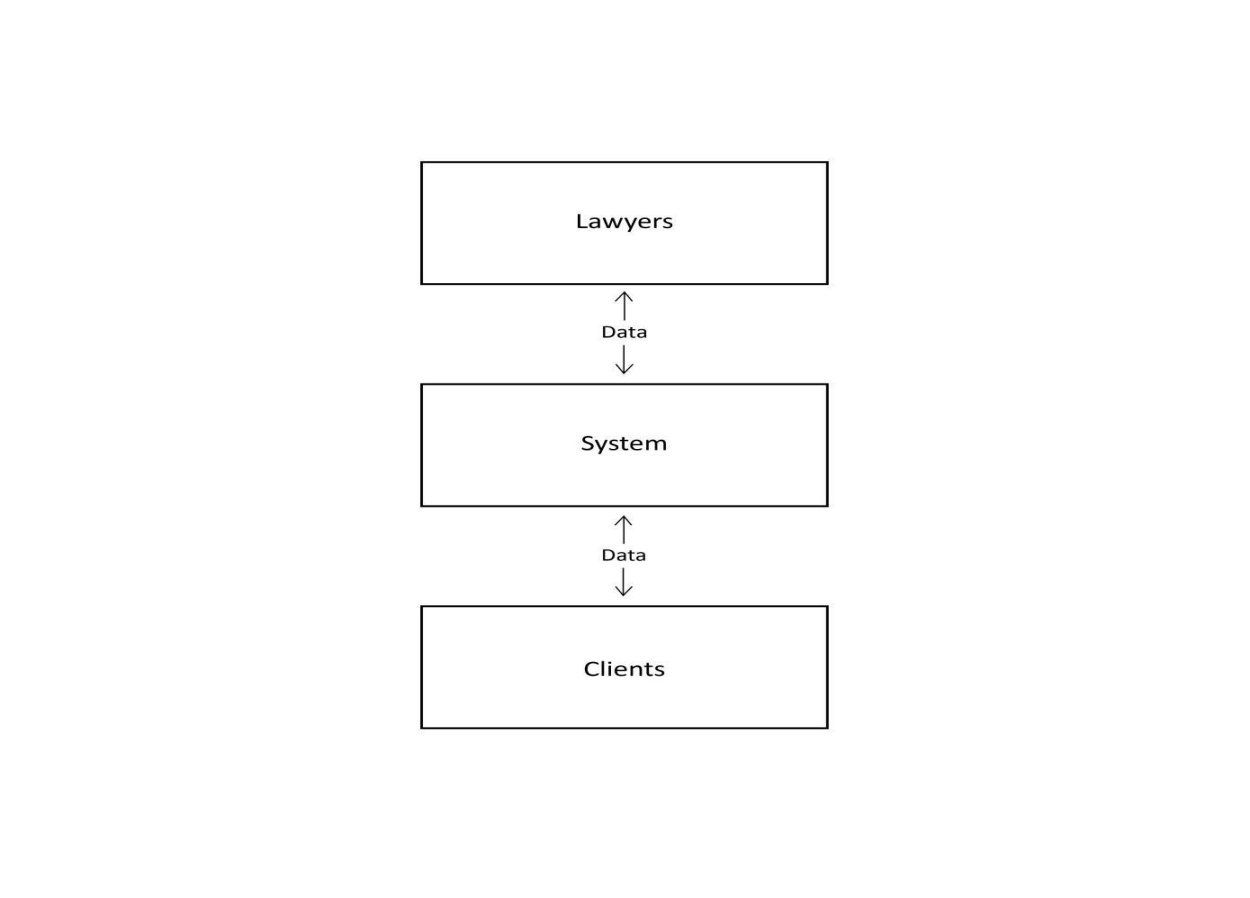
**Figure 13- Activity Diagram**

## Component Diagram



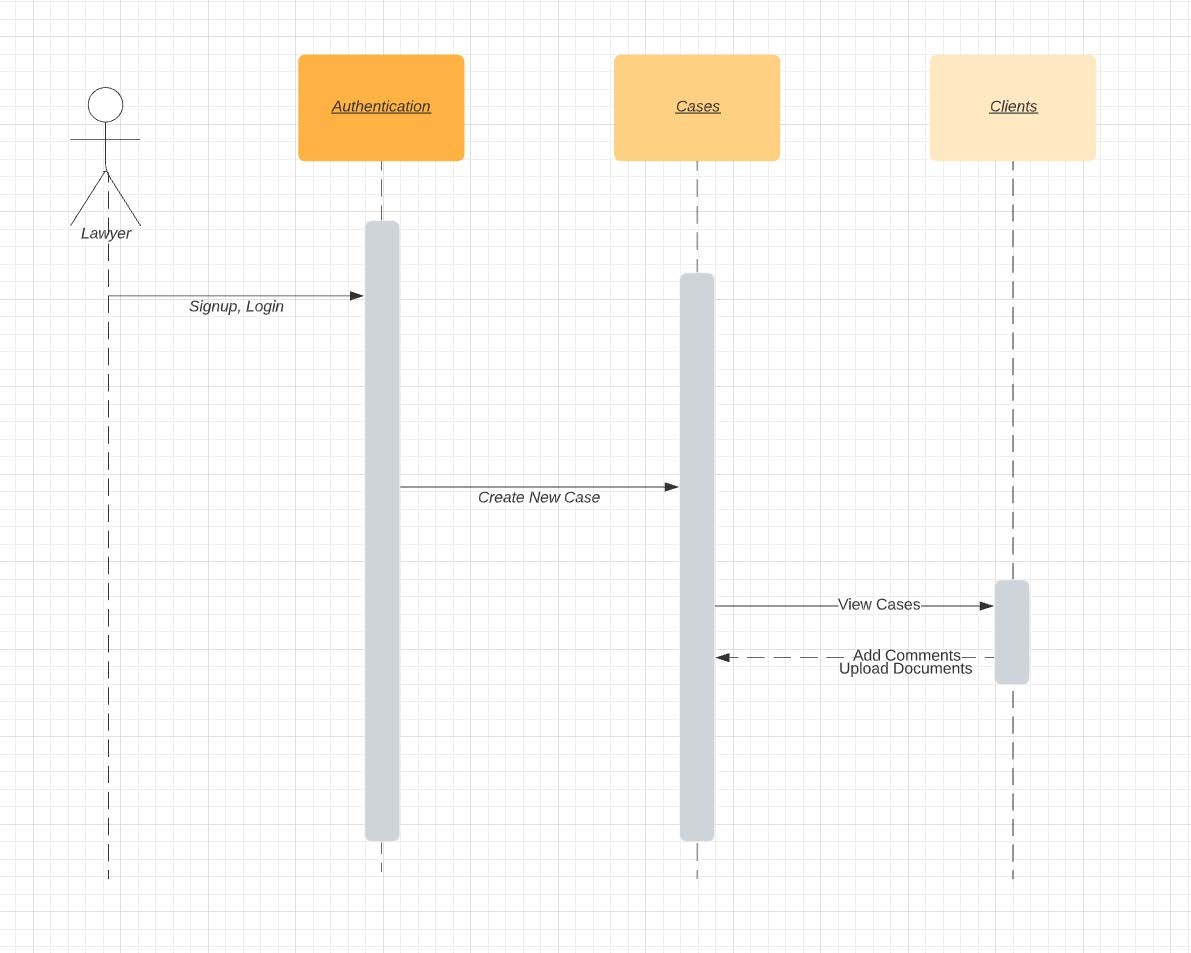
**Figure 14- Component Diagram**

***5.7 Data Flow Diagram***



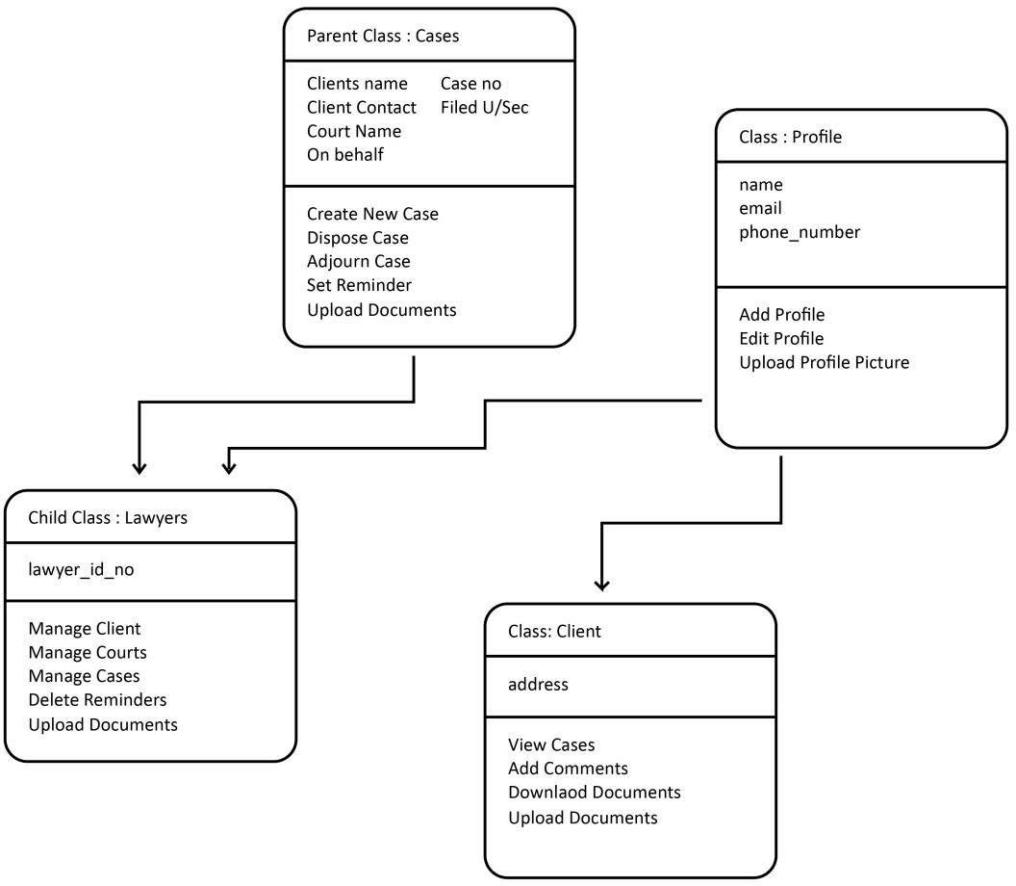
**Figure 15- Data Flow Diagram**

## Sequence Diagram



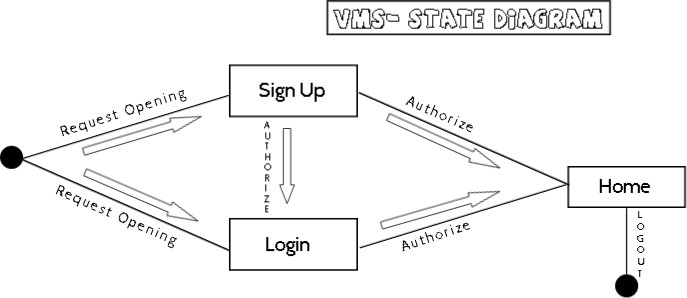
**Figure 16- Sequence Diagram**

## Class Diagram



**Figure 17-Class Diagram**

## State Diagram



**Figure 18- State Diagram**

## Tools

•Android OS, iOS

•React Native, MongoDB, Node JS

## Testing Methods

**Chapter 6**

**Implementation & Testing**

In this process we test three main Activities

1:Sign up 2:Login

3:Adding or updating cases

#### Test plan for Registration Activity

Screen Name: Registration. Test Plan ID: 001

Application Name: Casebook. QA: Faraz / Ahmad

Test no: 01

###### Purpose

Checking the Registration/Sign up activity is working properly.

###### Scenario

Application is responsive to generate account of the user.

###### Environment

Expo go

###### Pre-Requisite

* Download the application from play store.

###### Strategy

* Enter name, email password phone number
* Click on Sign up button

###### Expected Result

* Selected Activity Successfully generates the account.
* The user is connected with the Application for the further updates.

###### Observation

Application makes a connection with their users.

#### Test plan for Login Activity

Screen Name: Login.

Test Plan ID: 002

Application Name: Casebook. QA: Faraz / Ahmad

Test no: 02

###### Purpose

Checking the Login activity is working properly.

###### Scenario

Application is responsive to sign in accounts of the user.

###### Environment

Expo go

###### Re-Requisite

* Download the application from play store.
* User has to register itself before login.

###### Strategy

* Enter Email password
* Click on login button

###### Expected Result

* Selected Activity Successfully generates the account.
* The user is connected with the Application for the further updates.

###### Observation

Application makes a connection with their users.

#### Test plan for Logout Activity

Screen Name: Logout.

Test Plan ID: 003

Application Name: Casebook. QA: Faraz / AhmadTest no: 03

###### Purpose

Checking the logout activity is working properly.

###### Scenario

Application is responsive to generate account of the user.

###### Environment

Expo go

###### Pre-Requisite

* Download the application from play store.
* Register yourself
* Logged in to application.

###### Strategy

* Go to the profile page from home.
* Click on the logout button

###### Expected Result

* Selected Activity Successfully generates the account.
* The user is ready again for sign up and login.

###### Observation

Application makes a connection with their users.

#### Test plan for Home Page Activity

Screen Name: Home page. Test Plan ID: 004 Application Name: Casebook QA: Faraz / Ahmad

Test no: 04

###### Purpose

Checking the logout activity is working Properly.

###### Scenario

Application is ready for main activities.

###### Environment

Expo go

###### Pre-Requisite

* Download the application from play store.
* Register yourself
* Log in to application.

###### Strategy

* Go to home page.
* Click or swipe for respective activity.

###### Expected Result

* Selected activities successfully responding.
* The functionalities are ready to use for user.

###### Observation

Application makes a connection with their users.

## Testing

Testing phase is the most import phase of the project after development. It is on of the main crunch of the project. Project is tested in different environments and scenarios. The testing of the product is started during the development phase by Software Quality Assurance Team. SQA test the product in different time period by performing different type of tasks.

#### Purpose of Testing

Purpose behind testing is that if mistakes/bugs occur or any problem arises during testing. It should be resolve before handing over the project to the actual user. Another purpose is to check whether the project is working according to the requirements or not.

#### Testing the Registration and Home page activities

During the testing phase of our application we perform four types of testing.

* + - * Black Box Testing,
      * User Testing,
      * Unit Testing
      * Integration Testing.

#### Black Box Testing

This testing technique is most widely used today. In this technique SQA or the developer put his project into the specified program as an input and that program check the project irrespective of the internal structure and after checking highlights the errors or problems in that project. We use this testing technique during our development process and also after the project is completely built.

#### User Testing

This testing technique is also widely used. In this technique the project is given to random users to use as like they are using the complete application. This testing technique can be used after the completion of one module and at the end when all the modules are completed they are combined and again tested.

#### Unit Testing

This testing technique is used especially when we have to check the individual component of the project. When one module is completed it is tested using Unit Testing Technique. At the end all the modules are integrated and again this technique is used to test the project as a whole system.

#### Integration Testing

In this technique of testing project as a whole system is tested. When unit testing is completed then all the components are integrated and tested using Integration technique. If any error or a bug occurred during the testing phase it is corrected and removed immediately.

# 

# Chapter 7

# Results and Discussion

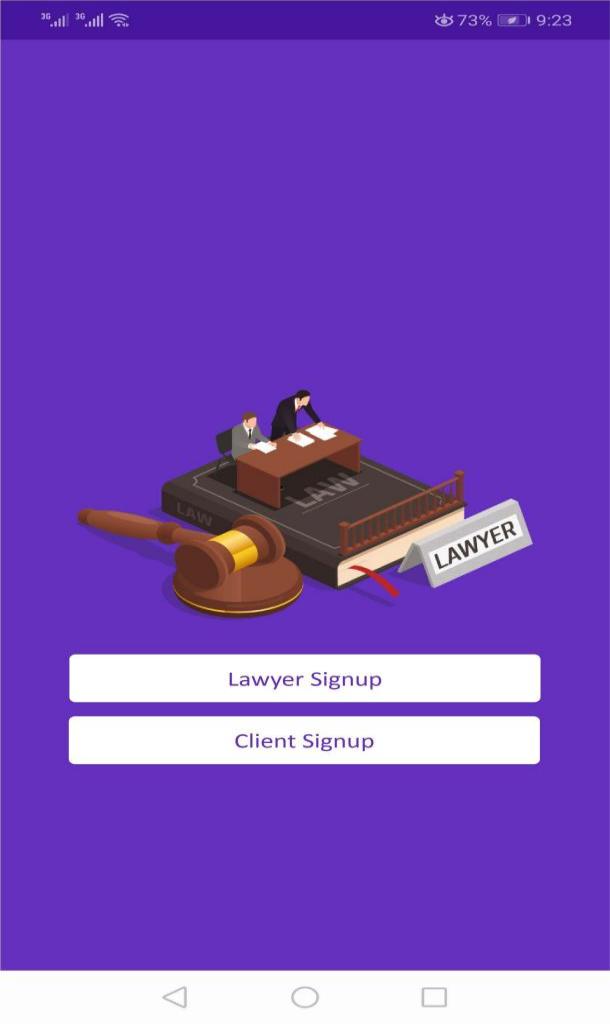
## Results and Test Evaluation

In previous chapter, we explain the testing techniques that we use to test our application during and after the development phase. In this chapter we will discuss the results that are obtained after the completion of the project.

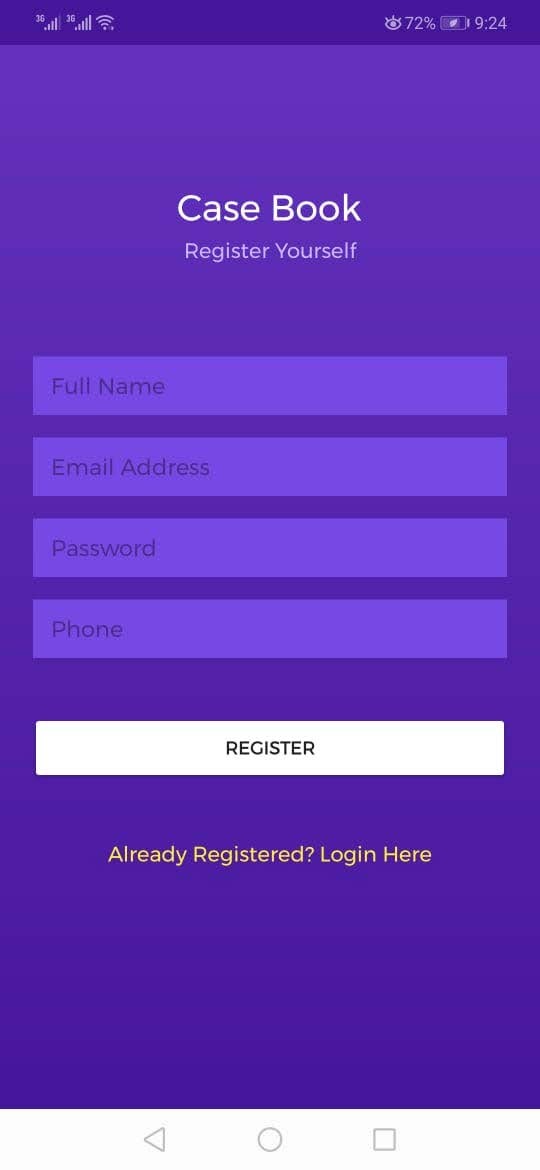
Screenshots of our Application interface are attached below after final testing and discussion with supervisor and other seniors of relevant field.

## Final User Interface and Working

#### Welcome Page

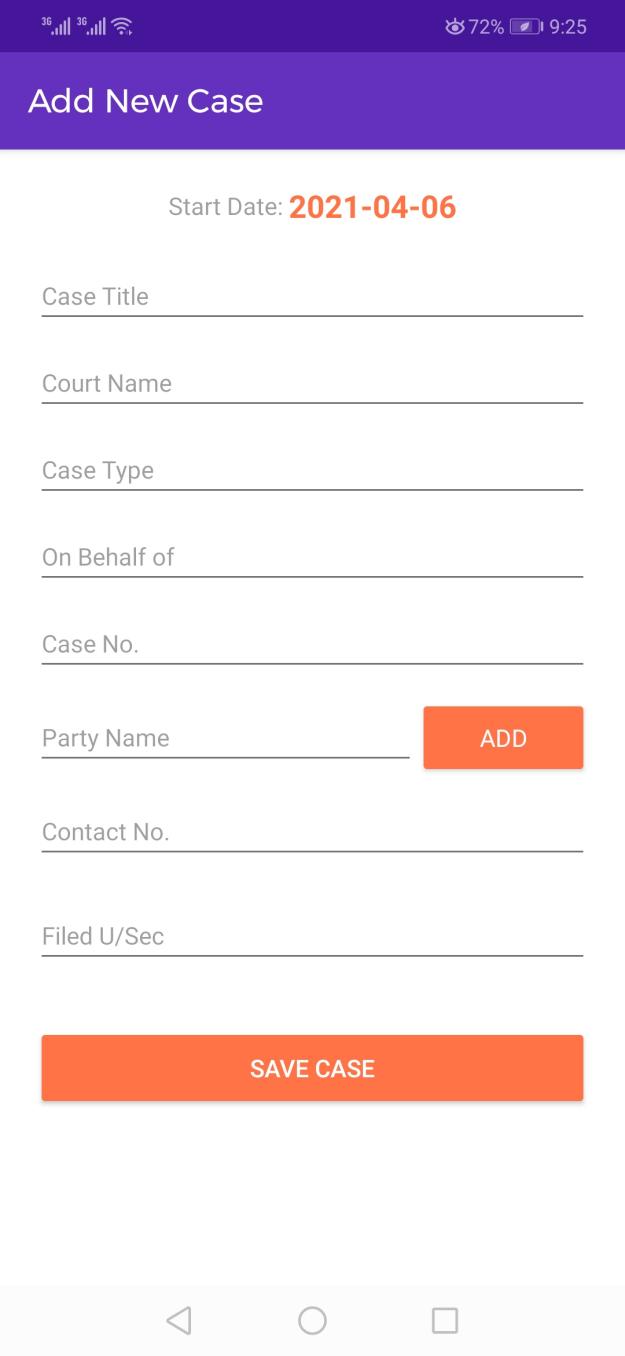


#### Sign Up Page

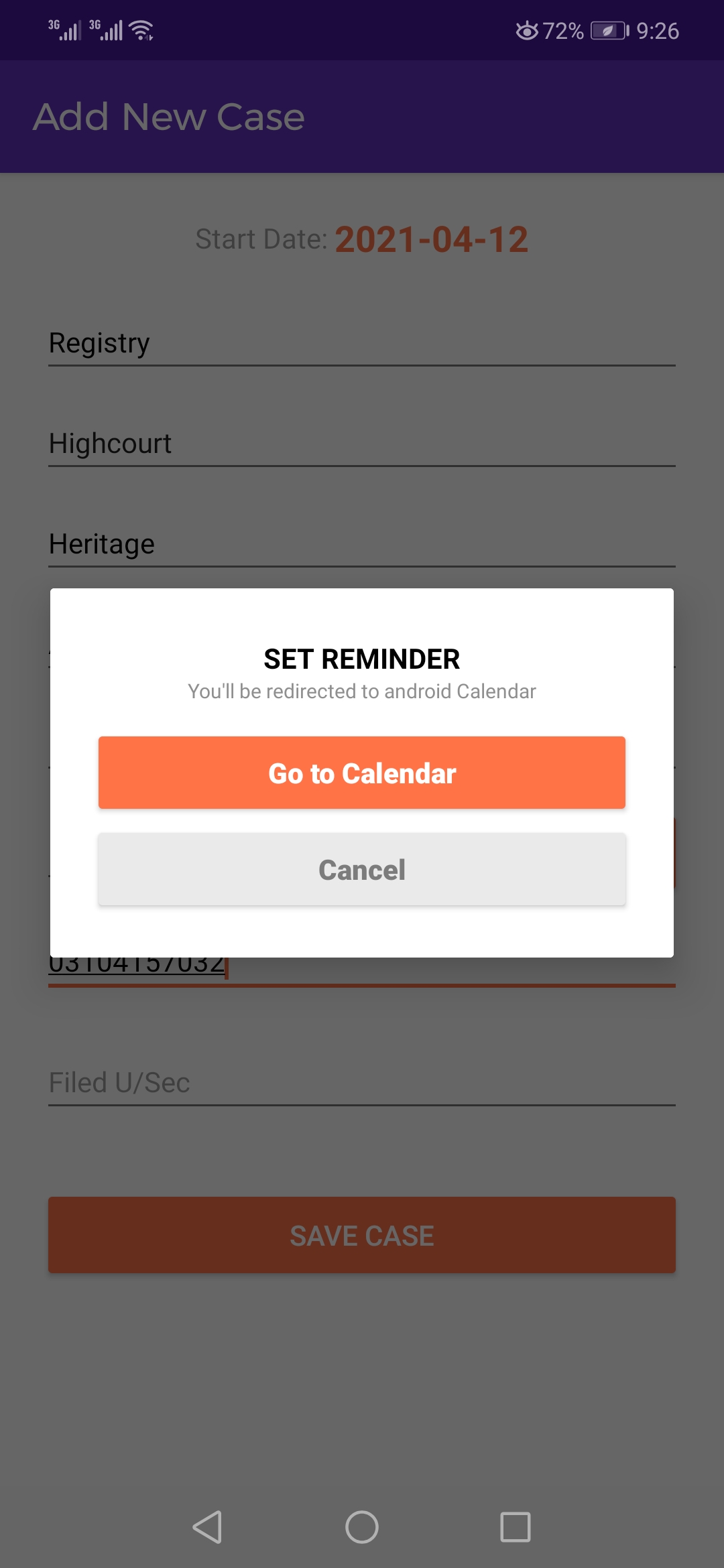


#### 

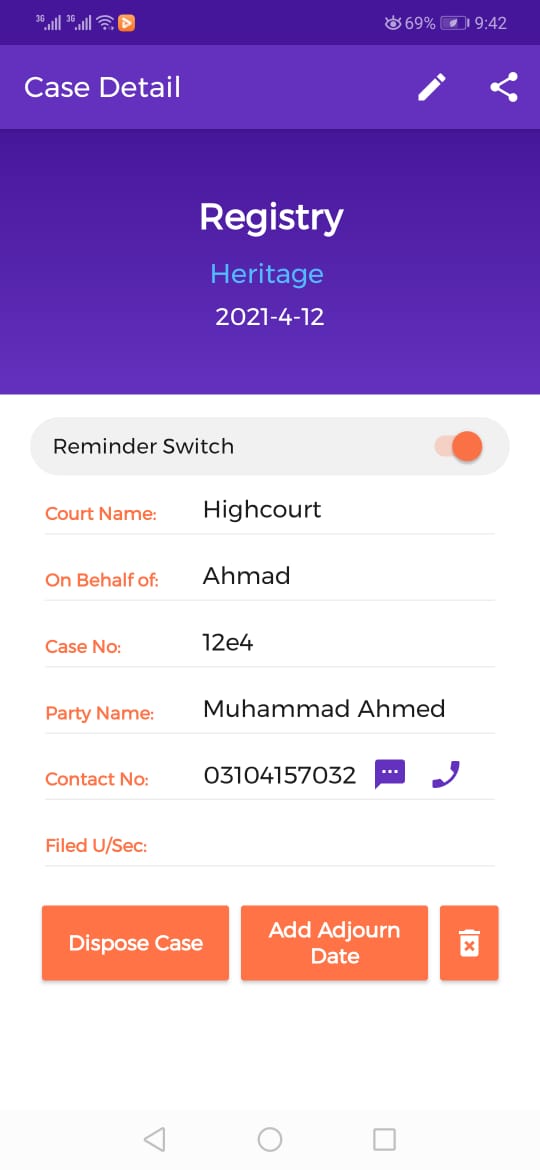
**Adding Case Page**



**Set reminder page**



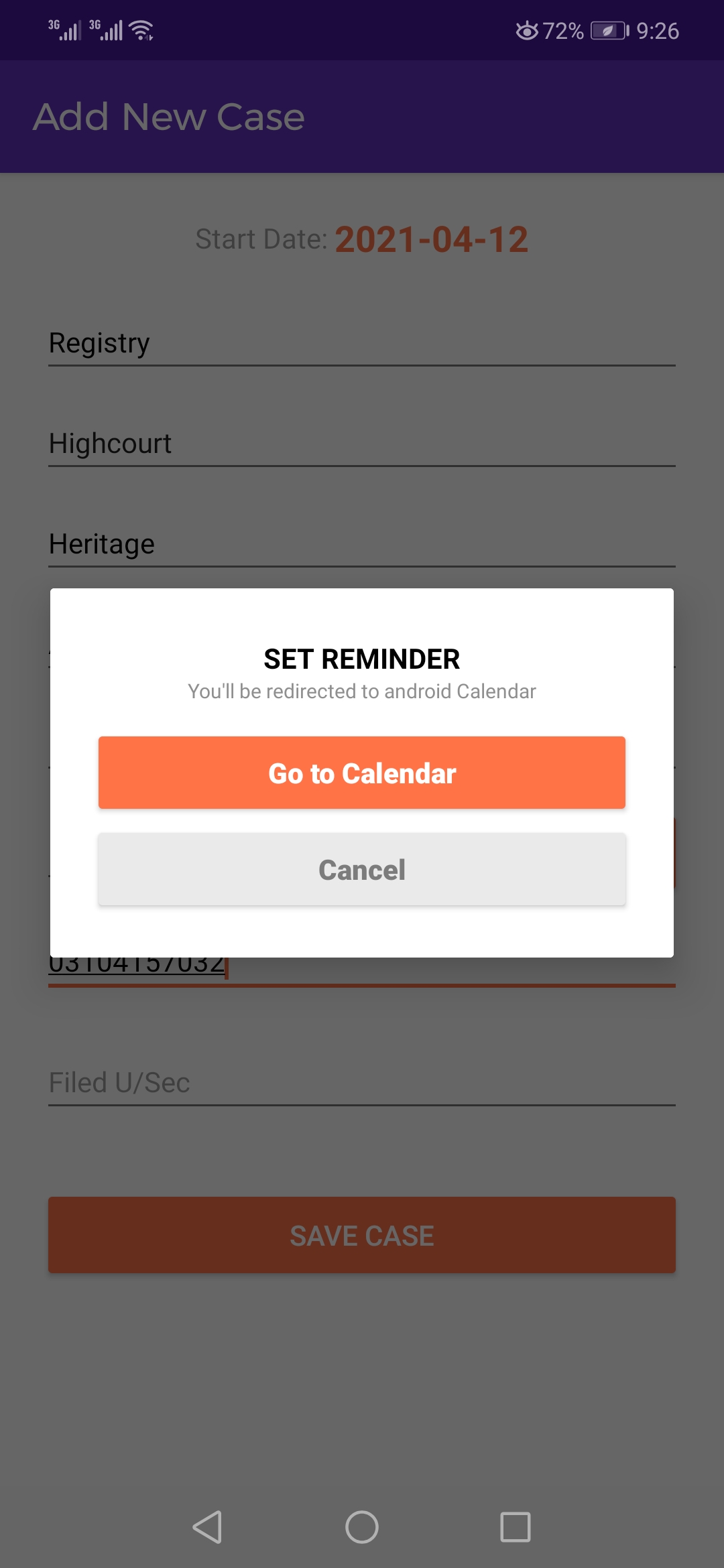
**Case Registry page**



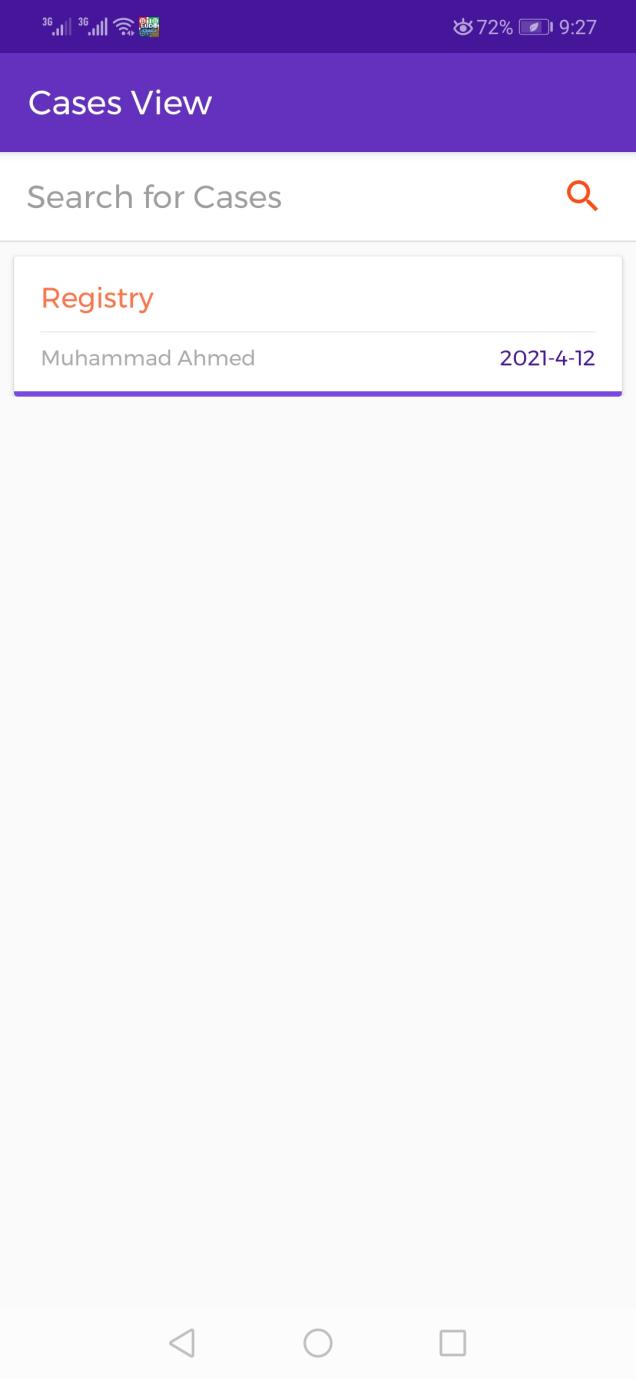
**Dashboard**



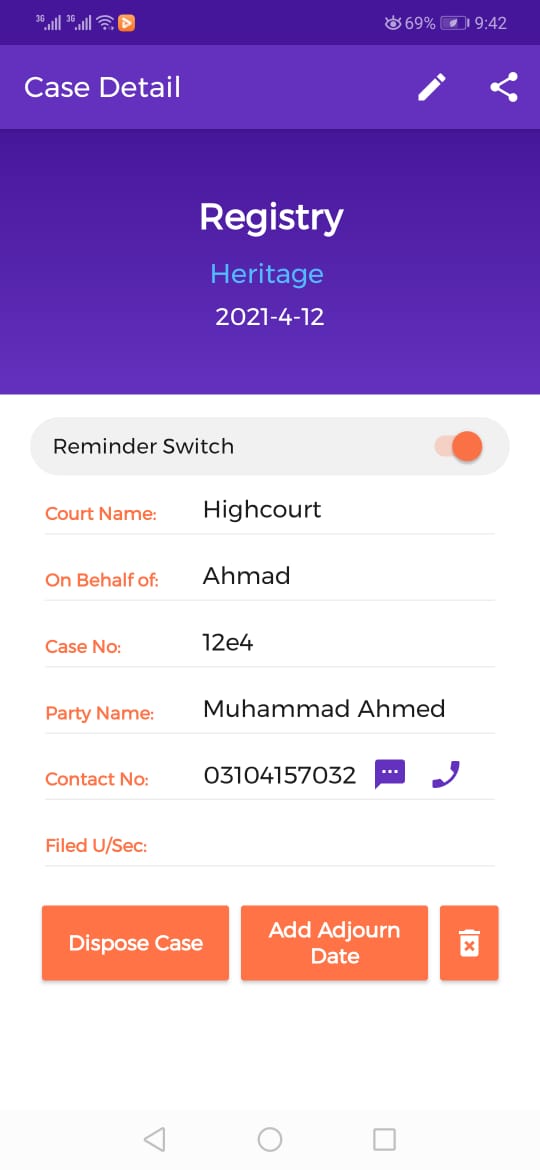
**Setting reminder 2**



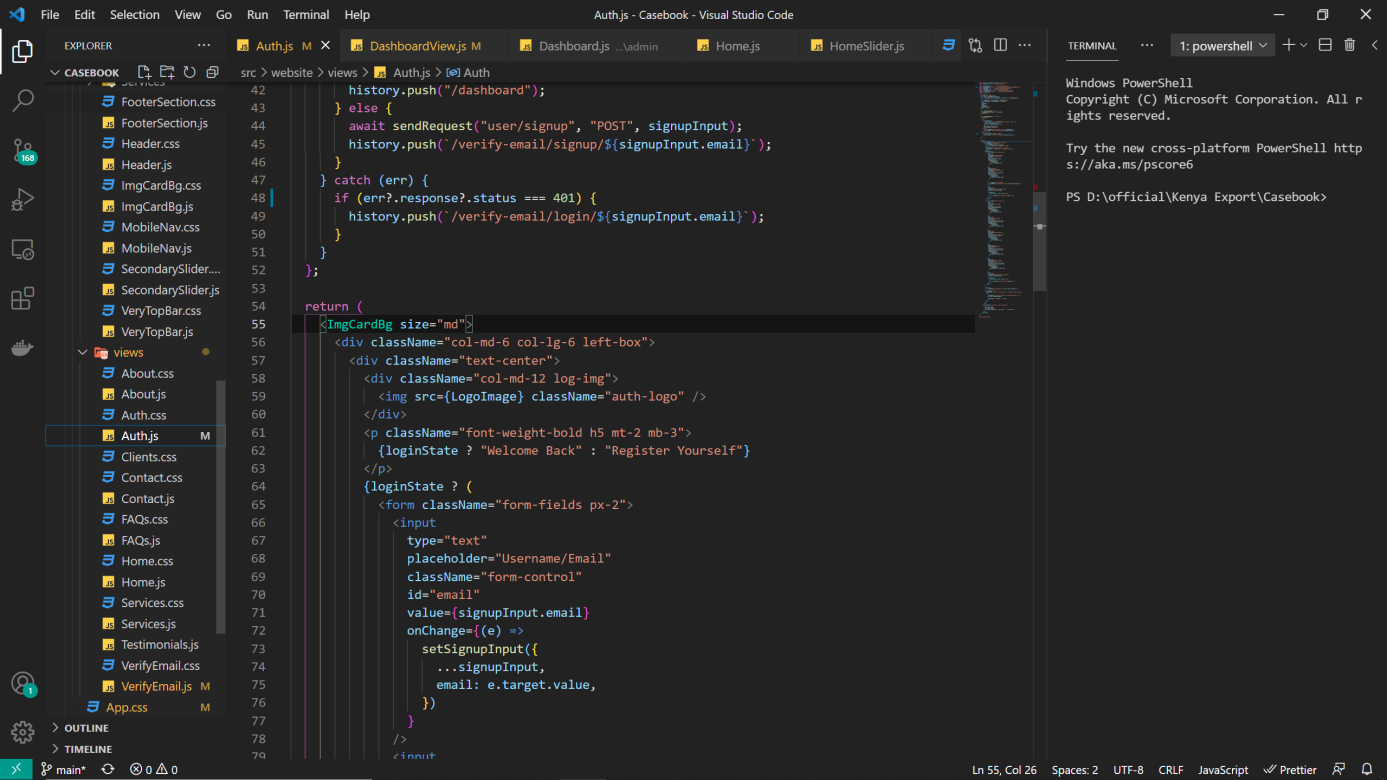
**View added cases**

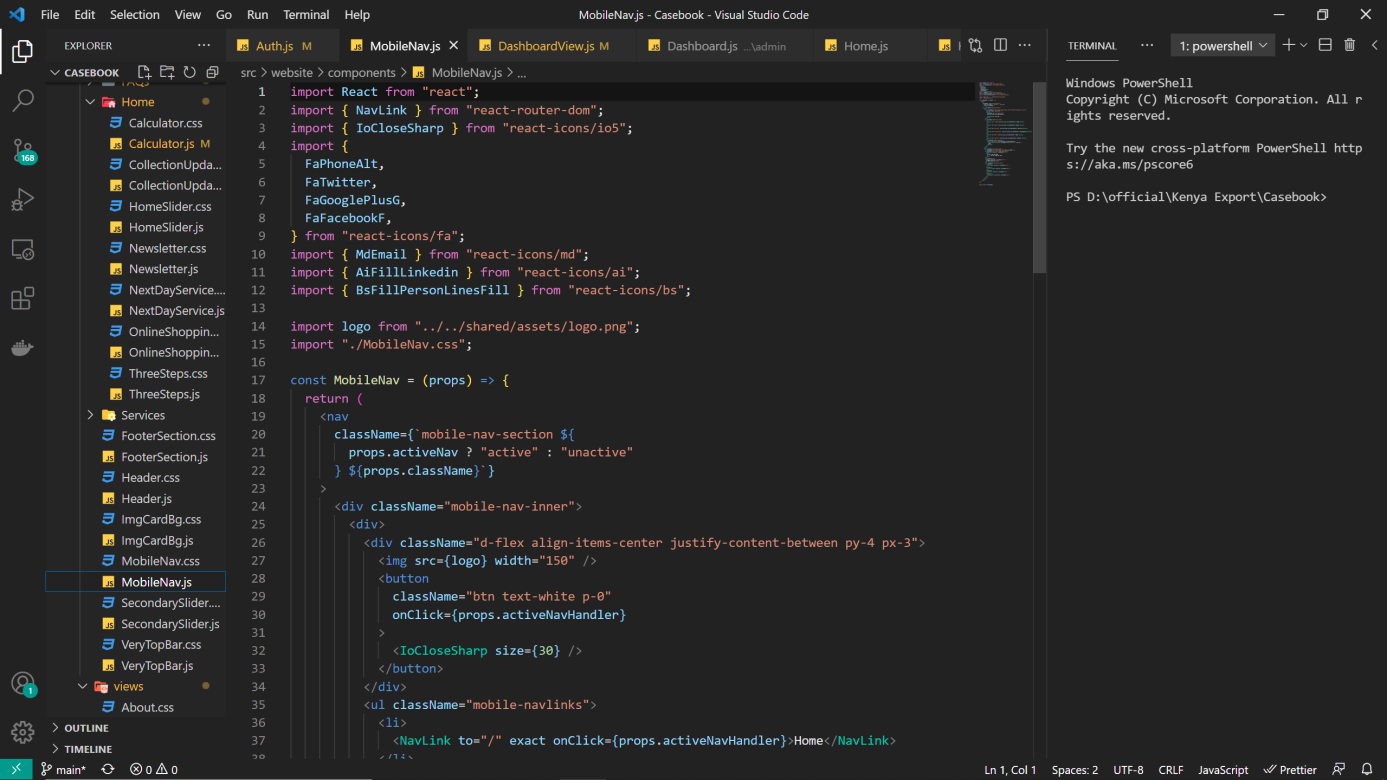


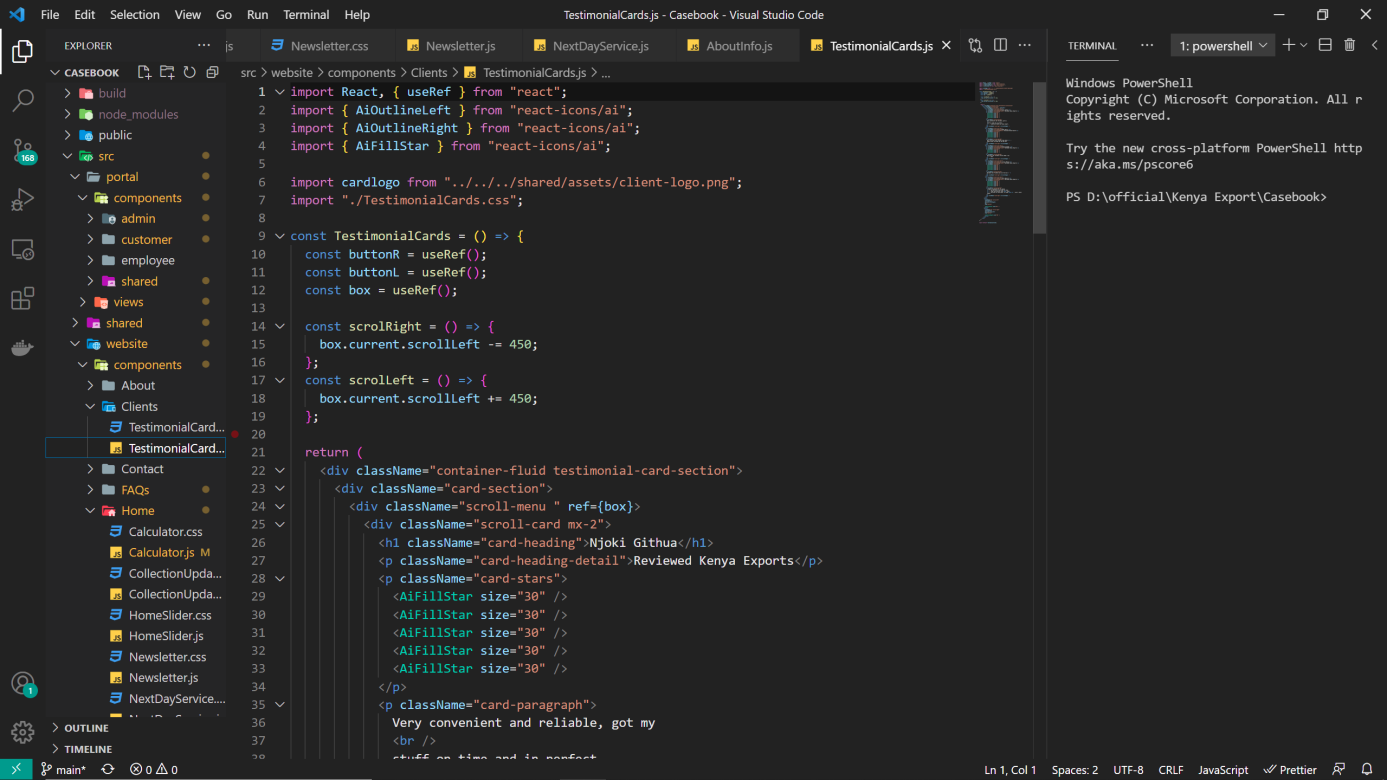
**Case description page**

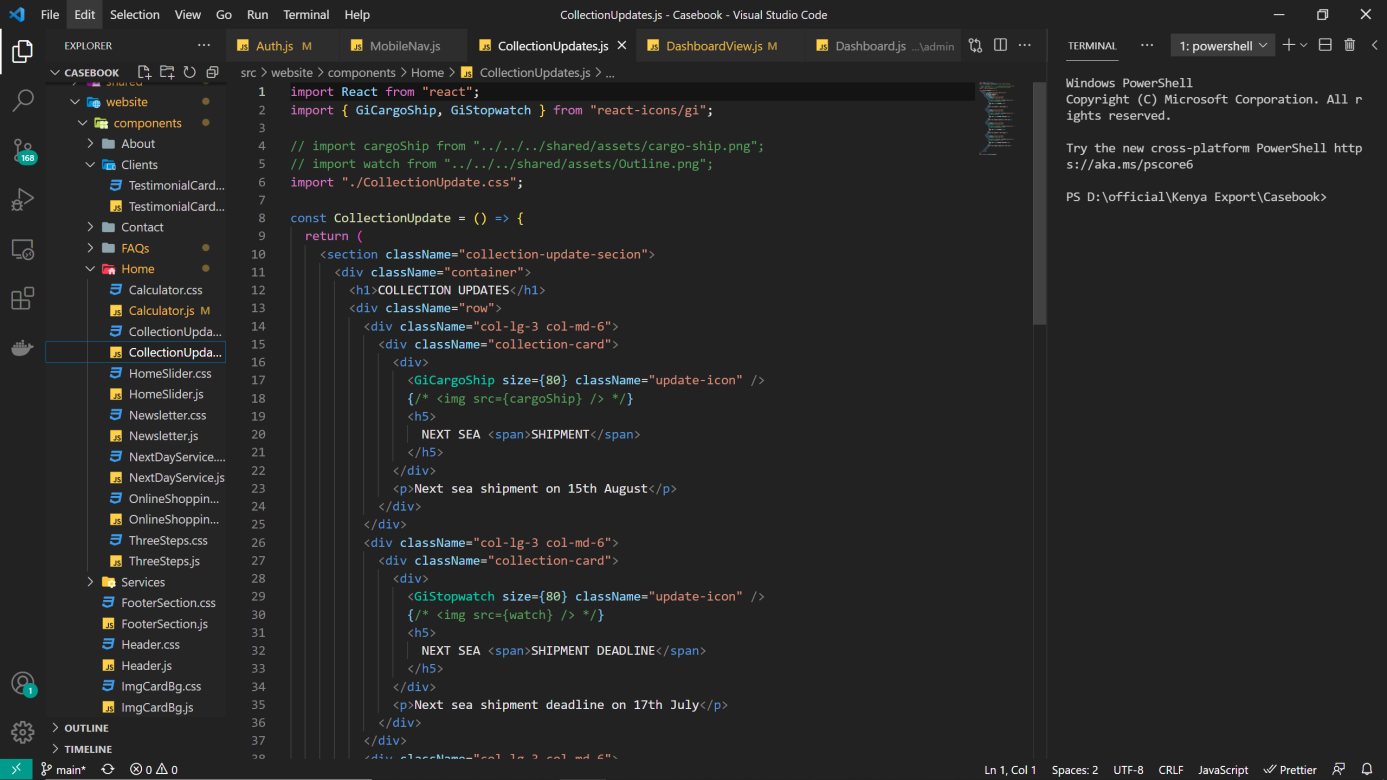


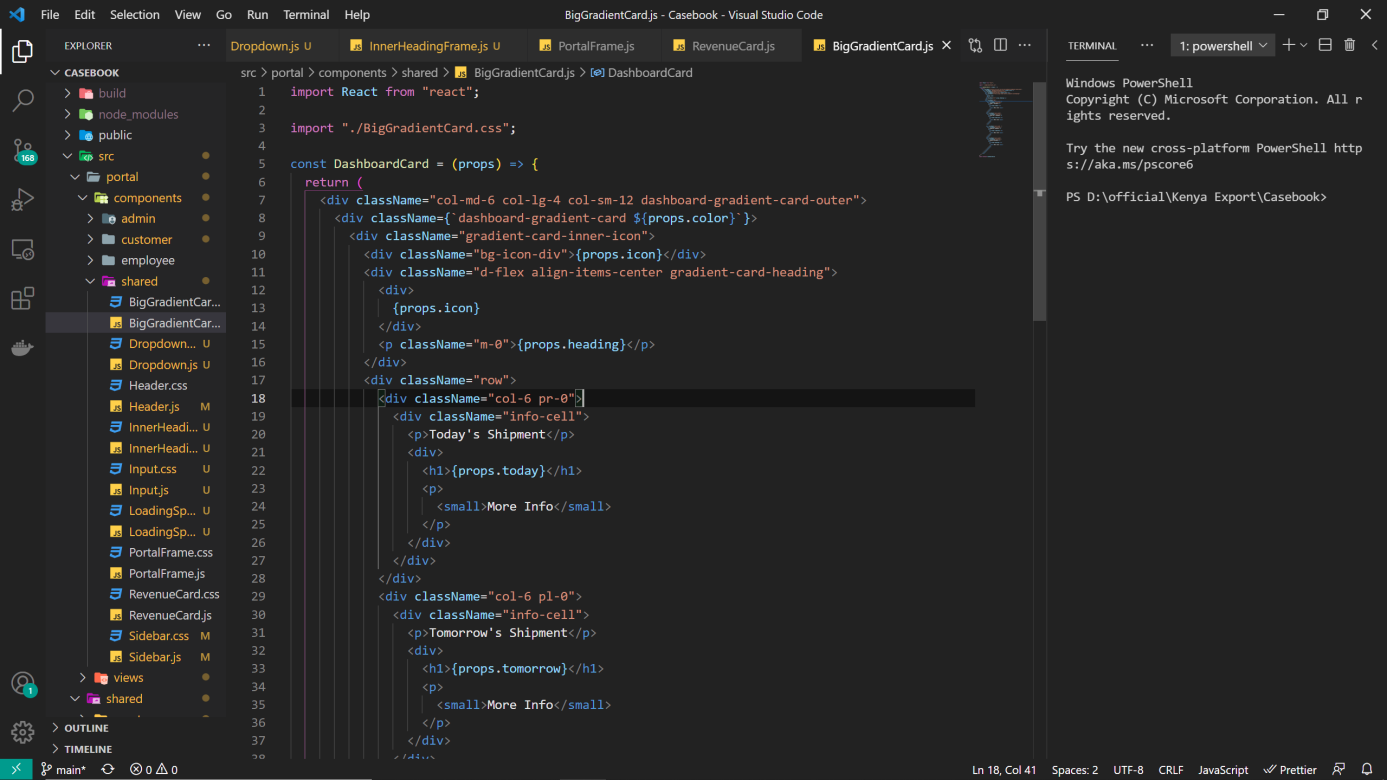
**Implementation Fig.**

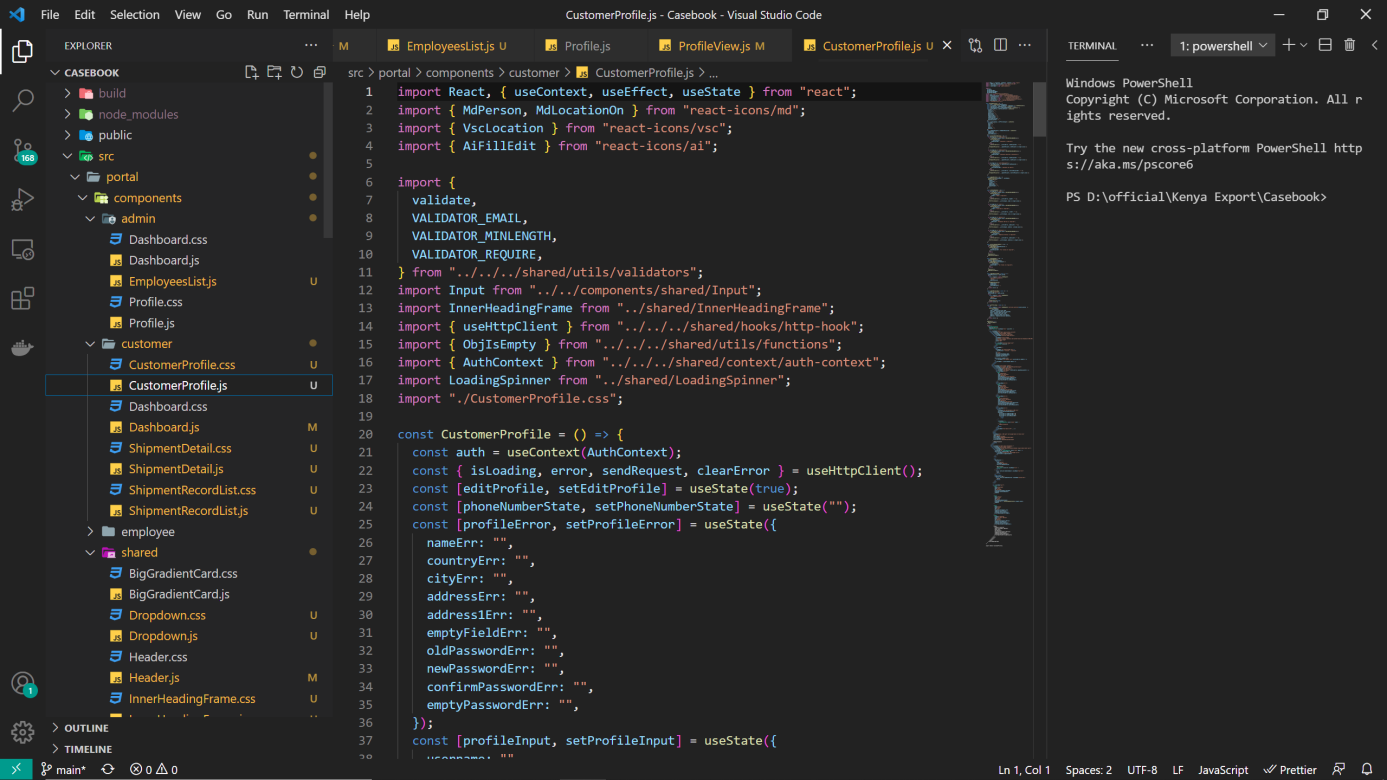


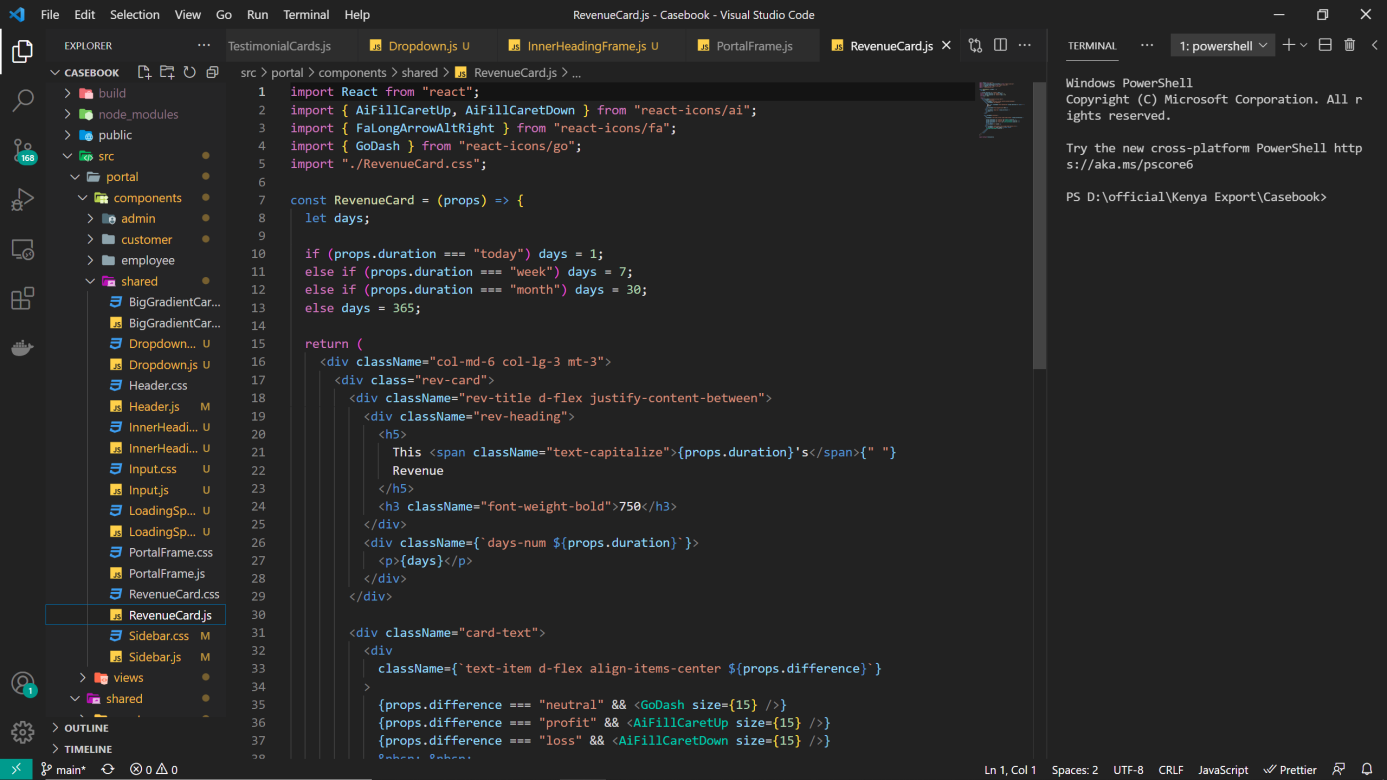












## Future Work

# Chapter 8

# Conclusion

As at the present due to lack of time and resources the app just focuses on the lawyer’s side of the but it could have extra functionalities covering all the bases of the other aspects of the cases.

We will use beta lawyer testers for this App and based on feedback we will keep working on this project and try to make to full fledged portal for lawyers judges and clients for implementation to our judicial system.

## Conclusion

The proposed system will really be helpful for the lawyers to save their precious hours of time. Primarily this app is made for the lawyers to manage the day to day works. The lawyers can add all their cases to a single platform online cloud based platform. Additionally the app also consist a client side where a client can mange all his cases with every lawyers with a simple signup or lawyer can let any client keep tab on his case without authentication with a single token based system.

# References

https://reactnative.dev https://material-ui.com

https://play.google.com/store/apps/details?id=com.pathlegal.app&hl=en&gl=US