A

MINOR PROJECT REPORT

ON

**BOOK YOUR DOCTOR: ANDROID**

A REPORT SUBMITTED IN PARTIAL FULFILMENT OF THE ACADEMIC

REQUIREMENT FOR THE AWARD OF DEGREE OF

**MASTER OF COMPUTER APPLICATION**

**Submitted**

**By**

**AMIYA RANJAN ROUT (1805106002)**

**Under the esteemed guidance of**

**Dr. JIBITESH MISHRA**

**Head of Department**

**Department of CSA**



**DEPARTMENT OF COMPUTER SCIENCE & APPLICATION**

**COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(Techno Campus ,PO-Ghatikia, Mahalaxmi Vihar,Bhubaneswar-751029)**

**2019-2020**

**COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(Techno Campus, PO- Ghatikia, Mahalaxmi Vihar,Bhubaneswar-751029)**

**DEPARTMENT OF COMPUTER SCIENCE & APPLICATION**



**CERTIFICATE**

This is to certify that the minor project entitled **BOOK YOUR DOCTOR: ANDROID** is being submitted by **AMIYA RANJAN ROUT** with registration no. **1805106002** in partial fulfilment of the requirement for the award of the degree of MCA in College of Engineering & Technology under our guidance and supervision from July to December 2019.

The results presented in this report have been verified and are found to be satisfactory. The results embodied in this report have not been submitted to any other university for the award of any other degree.

**INTERNAL GUIDE HOD**

**Dr. Jibitesh Mishra Dr. Jibitesh Mishra**

*(***Associate Professor)** *(***Associate Professor)**

**COLLEGE OF ENGINEERING & TECHNOLOGY**

**(Techno Campus, PO- Ghatikia, Mahalaxmi Vihar,Bhubaneswar-751029)**

**DEPARTMENT OF COMPUTER SCIENCE & APPLICATION**



**ACKNOWLEDGMENT**

I am extremely grateful to **Professor Dr.Jibitesh Mishra, H.O.D** as well as our internal guide, Department of CSA, College of Engineering & Technology for their encouragement and valuable guidance in bringing shape to this project report.

I will be failing in duty if I do not acknowledge with thanks to the author of the references and other literature referred in this project.

I express my thanks to all staff members and friends for all the help and coordination extended in bringing out this project successfully in time.

Finally, I am very much thankful to my parents who guided me for every step.

**Date:**

**AMIYA RANJAN ROUT**

**(1805106002)**

**Place:**

**COLLEGE OF ENGINEERING & TECHNOLOGY**

**(Techno Campus, PO- Ghatikia, Mahalaxmi Vihar,Bhubaneswar-751029)**

**DEPARTMENT OF COMPUTER SCIENCE & APPLICATION**



**DECLARATION**

I,  **Amiya Ranjan Rout** bearing **Roll No: 1805106002** is a bonafide student of **College of Engineering and Technology,** would like to declare that the project titled **“BOOK YOUR DOCTOR: ANDROID”** is a partial fulfilment of MCA Degree course of College of Engineering & Technology is my original work in the year 2019 under the guidance of **Dr. Jibitesh Mishra,** Associate Professor, Computer Science & Application Department and it has not previously formed the basis for any degree or diploma or other any similar title submitted to any university.

**Date: Student Name**

**ABSTRACT**

Now a day’s healthcare issue is one of the tough challenges for our country. For metro cities, smart cities it may not be a major issue but for small cities and especially for rural areas this issue is enhancing day by day because of lack of experienced doctors. So, our main focus is to develop this project to reduce health care issues in those areas. We aim to develop an application that would act like a medium between Doctor and Patient. As we know some experienced doctor from bigger cities used to visit their private clinic or some pharmacies situated in small cities and rural area once or more than once in a week. But people in their respective areas are unable to know about Doctor's availability. And even if they get to know the Doctor's availability, they are facing a lot of problems to get a ticket. For e.g. they have to stand in queue from dawn to get an early ticket. Through our platform, our main target is to collaborate with those pharmacies and provides some better solution for this issue. This application will provide availability of registered Doctors. And Patients can also search doctors as per their specialty and book an appointment for the same. And if later on they changed their mind they can also cancel the appointment.

**INDEX**

**Chapter 1: Introduction------------------------------------------------------------------------8**

* 1. Introduction**-----------------------------------------------------------------------9**
  2. Project Scope**----------------------------------------------------------------------9**
  3. Existing System**-------------------------------------------------------------------10**
  4. Proposed System**-----------------------------------------------------------------10**

**Chapter 2: Literature Survey/Review of Literature-------------------------------------11**

2.1 Introduction--------------------------------------------------------------------------12

2.2 Related Work------------------------------------------------------------------------12

**Chapter 3: System Requirement-------------------------------------------------------------14**

3.1. Software Requirement**------------------------------------------------------------15**

3.2. Hardware Requirement-----------------------------------------------------------15

**Chapter 4: Software Requirement Analysis----------------------------------------------16**

4.1. System planning and the initial Investigation-------------------------------17

4.2. Information Gathering------------------------------------------------------------17

4.3. Feasibility Study--------------------------------------------------------------------18

4.3.1. Technical Feasibility-----------------------------------------------------18

4.3.2. Economic Feasibility-----------------------------------------------------19

4.3.3. Cost Benefit Analysis---------------------------------------------------19

4.3.4. Legal Feasibility----------------------------------------------------------20

**Chapter 5: Project Plan & Approach--------------------------------------------------------21**

**Chapter 6: Project Schedule------------------------------------------------------------------23**

**Chapter 7: Software Design--------------------------------------------------------------------32**

7.1. Scope-----------------------------------------------------------------------------------33

7.2. System model specification-------------------------------------------------------33

7.3. Class Diagram-------------------------------------------------------------------------35

7.4. Use Case Diagram-------------------------------------------------------------------36

7.5. ER Diagram----------------------------------------------------------------------------43

7.6. System Architecture----------------------------------------------------------------44

**Chapter 8: Use Case Description--------------------------------------------------------------45**

**Chapter 9: Coding /Code Templates---------------------------------------------------------45**

**Chapter 10: Testing-------------------------------------------------------------------------------52**

**Chapter 11: Database Design View----------------------------------------------------------60**

**Chapter 12: Output Screens--------------------------------------------------------------------64**

**Chapter 13: Conclusion-------------------------------------------------------------------------72**

**Chapter 14: References/Bibliography-------------------------------------------------------75**

**Chapter 1. INTRODUCTION**

* 1. **INRODUCTION**

**BookUrDoctor** is more than an app for finding doctors. Whether it's viewing doctors’ profile, booking doctor appointments, confirming appointments**,** cancelling appointments, **BookUrDoctor** helps you find online medical services & solutions to enable you to take better care of yourself. Briefly, the main goal of this project is: Finding the right doctors for your good health can be a challenge. With an expansive list of verified doctors, specialists, surgeons from neighborhood clinics & top hospitals to choose from, we help you get appointments with the doctors using the **BookUrDoctor** app.

**1.2. Project Scope**

* The system allows a doctor to add their details such as their name along with their designation, specialization, experiences and also with a brief description of them.
* User can search for top experienced doctors as per their specialization.
* User can search for top experienced doctors as per their nearby location.
* User can find right doctors by viewing the doctor’s profile.
* User can book an appointment with their specialized doctors.
* User can see their all appointment letters at a one place.
* User can also cancel their appointments.

**1.3. EXISTING SYSTEM**

In the existing system it only working on metro cities and smart cities. Then what about for small town and especially for rural area? And also, it’s main focus on doctor’s private clinic which is very rare in rural areas.

The following are the drawbacks present in the existing system:

1. Only for smart cities and for metro cities.
2. No scope for small towns.
3. No scope for rural areas.
4. Only focus on doctor’s private clinic.

**1.4. Proposed solution**

In the proposed system our main focus is to collaborate with those pharmacies in which some experienced doctors from bigger cities used to visit once or more than once in a week which are situated in small cities and rural areas. Our proposed system provides solution to existing system by extending its facilities as follows:

1. Extending the facility to the small-town areas.
2. Extending the facility to the rural areas as well.
3. Collaborating with pharmacies.

**Chapter 2: Literature Survey/Review of Literature**

**2.1. Introduction**

A literature review is a search and evaluation of the available literature in your given subject or chosen topic area. A lot of projects and researches has been done aiming in proper solving the problem of users.

**2.2. Related Work**

Here is some related work of the book your doctor system

**2.2.1. Android Civil Administration Reporting**

***Internal Journal of Advance Research, Ideas And Innovation In Technology***

***(ISSN:2454-132X,Impact Factor:4.295)***

In this project user must register an account and launch a complaint to the authority with the generated unique ID. The system allow the user to register and as soon as they register an account with unique id is created in the system. The user must require an Android application to register a complaint by mobile or desktop. The status of the complaint is updated only by respective authority. The user has to wait till the authority changes the status of the complaint. Complaints can be easily registered by the user and can be tracked with the unique complaint ID. The user can then view the progress of the complaint. This system will not only save the time of the complaint fillers but also lead to track their complaint and its status in a regular interval of time. Once the reply of the complaint is given, the user will get the update on his smartphone and email address as well.

**2.2.1. Android Civil Administration Reporting**

***International Journal for Research in Applied Science & Engineering Technology(IJRASET)***

***(*ISSN:2321-9653;IC Value: 45.98; Impact Factor: 6.887)**

In India people don’t have any direct communication between the government and public in an efficient way for solving the problems i.e. for getting a problem solved in our place we have to bribe the officials and get them solved in 2 months which can be solved actually in 1 month of time. In order to overcome this problem previously National Informatics Centre has launched a site named Prajavani through which public can post the petitions or complaints in the site and get them solved in a specified time and can also know the status of the complaint or petition he has lodged at any time.

NIC has launched this site with the goal of Right TO Information Act (RTI Act) i.e. providing the complete information of a place to the user at any time. But it failed in providing the complete information to the public and is providing only the complaint lodging facility to the public. In order to make the goal of NIC come true we are going to develop a system which will be able to provide the complete information to the public at any point of time regarding the problems they are facing currently and what is the impact of it and then how effectively the funds are utilized for the development purpose can be known by public which also includes the online discussion forums and feedback forms which will help them to communicate well with the government. It concludes by publishing a newsletter and a magazine to the registered users of the system which gives the complete details of the district for every month.

**Chapter 3. System Requirement**

* 1. **Software Requirement**

**For Android Platform**

**3.1.1.** **Frontend**

3.1.1.1. Android Development Tools

3.1.1.2. Advance Java

3.1.1.3. XML

**3.1.2. Backend**

3.1.2.1. Firebase

3.1.3.IDE

3.1.3.1. Android Studio

3.2. **Hardware Requirement**

3.2.1. **Application Requirement**

3.2.1.1. Android version 6.0 (Marshmallow onwards)

3.2.1.2. 4GB RAM

3.2.1.3. 1.2 GHZ Processor

3.2.2. Development Environment Requirement

3.2.2.1. Intel i3 5th Generation Processor

3.2.2.2. Windows 10 / UBUNTU 14.0.4

3.2.2.3. Android Studio Installed

**For Web Platform**

**3.1.1.** **Frontend**

3.1.1.1. Web Development Tools

3.1.1.2. HTML, CSS

3.1.1.3. BOOTSTRAP

**3.1.2. Backend**

3.1.2.1. MySQL

3.1.3.IDE

3.1.3.1. Brackets, Sublime

3.2.1. Development Environment Requirement

3.2.1.1. Intel i3 5th Generation Processor

3.2.1.2. Windows 10 / UBUNTU 14.0.4

3.2.1.3. Brackets or Sublime Installed

**Chapter 4: Software Requirement Analysis**

**4.Software Requirement Analysis**

Analysis is a detailed study of the various operations performed by a system and their relation within and outside the system. A key question is what must be done to solve the problem. One aspect of the analysis defining the boundaries of the system and determining whether or not a candidate system should consider other related system. During analysis data are collected on the available files decision points and transaction handled by the parent system. Some logical system models and tools are used in the analysis. Data flow diagrams, interviews, onsite observation and questionnaires are examples, the interview is commonly used in analysis. It requires special skill and sensitivity to the subject being interview bias in data collection and interpretation can be problem.

Training experience and common sense are required for collections of the information are needed to do analysis. Once analysis is completed, the analyst has firm understanding of what is to be done. The next step is to decide how the problem might be solved. Thus, in system design, we be move from the logical to the physical of the life cycle.

**4.1. System Planning and the initial investigation**

The most critical phase of managing system projects is planning to launch a system investigation, we need plan detailing the steps to be taken, the people to be questioned and they outcome expected. The initial investigation has the objective of determining whether the users request has potential merit. The major steps are defining user requirements. When the initial investigation is completed. The user receives a proposal summarizing the finding the recommendation of the analyst.

**4.2. Information Gathering**

A key part of feasibility analysis is gathering information about the present system. The analyst knows what information to gather, where to find it, how to collect it and what to make of it. The proper use of tools for gathering information is the key to successful analysis. The tools are the traditional interview, questionnaire, and on-site observation. We need to Know, for example how to structure an interview, what makes up a questionnaire, and what to look for on-site observations. These tools when learned help analysis assess the effectiveness of the present system and provide the groundwork for recommending a billing system.

**4.3. Feasibility Study**

The main objective of the preliminary analysis is to identify the problem, evaluate the system concept of feasibility, and perform the economic and technical analyses perform the cost benefit analysis. After the clarification analysis the solution proposed it is checked that it is practical to implement that solutions. This is done through the feasibility study. It is checked for various aspect whether the proposed solution is technically or economically feasible or not. On the basis of which it has been categorized into four classes viz

1. Technical
2. Economic
3. Legal
4. Operational

The outcome of the preliminary analysis should be clear so that an alternate way to do the job can be found out.

**4.3.1. Technical Feasibility**

During the technical feasibility studies following issues are taken into consideration

1.Whether the required technology is available or not?

2. Required resources are available or not? (Manpower, programmer, software and hardware etc)

Once the technical feasibility is established, it is important to consider the monetary factors also. Since it might happen that developing a particular system may be technically possible but it may be require huge investments and benefits may be less. For evaluating this, economic feasibility of the proposed system is carried out.

As in our proposed system our team has technically trained manpower with knowledge of developing the system. We are going to use web technology in our system, which is readily available. Software to be used is also available easily. So technically the project is feasible.

**4.3.2. Economic Feasibility**

For any system if the expected benefits equal or exceed the expected costs, the system can be judged to be economically feasible. In economic feasibility, cost benefit analysis is done in which expected costs and benefits are evaluated.

Economic analysis is used for evaluating the effectiveness of the proposed system. In economic feasibility, the most important is cost benefit analysis. As the name suggests, it is an analysis of the cost to be incurred in the system and benefits derivable out of the system.

As in our institute the hardware and software required for this type of system is already available so economically our project is feasible.

**4.3.3. Cost Benefit Analysis**

Developing an IT application is an investment. Since after developing that application it provides the organization with profits. Profits can be monetary or in the form of the improved working environment. However, it carries risks, because in some cases an estimate can be wrong. And the project might not actually turn out to be beneficial.

Cost Benefit Analysis helps to give management a picture of cost, benefits and risks. It usually involves comparing alternate investments.

Cost Benefit determines the benefits and saving that are expected form the system and compare them with the expected costs.

The cost of an information system involves the development cost and maintenance cost. The development costs are one-time investments whereas the maintenance cost is recurring. The development cost is basically the cost incurred during the various stages of the system development.

Each phase of the life cycle has a cost. Some examples are:

1. Personal
2. Equipment
3. Supplies
4. Overheads etc.

In proposed system all hardware and Software are available in the company so cost is 0%.

**4.3.4. Legal Feasibility**

It includes study concerning contracts, liability, violation and legal other traps frequently unknown to the technical staff.

**Chapter 5: Project Plan & Approach**

**5. Project Plan & Approach**

Software development team consists of 3 members.

***Phase-1: Requirement Gathering and Analysis***

Gathering system requirement and prepare a System Requirement Specification document.

After collect the information it is analysed that the available resources can fulfil all the requirements. And it is also be examined that what resource will be used.

***Phase-2: System Design***

Make a detailed analysis of the system and prepare a System Design Document on the basis of SRS.

***Phase-3: Prepare UTC/STC for testing the software***

Prepare Unit Test Cases document. This document will be used to verify whether the functional requirements of the system have been met.

***Phase-4: Develop the software***

Develop the planned system

***Phase-5: Test the software using the prepared UTC/STC and Rework if needed***

Run your software programs using the respective UTC to verify & test the software.

***Phase-6: Demonstrate the software to users & Implement it***

**Chapter 6: Project Schedule**

**PROJECT SCHEDULE**

|  |  |  |
| --- | --- | --- |
| ***Step #*** | ***Time Frame*** | |
| **Phase 1**. **Requirement Gathering** | **18 Days** | |
| **Phase 2. System Design** | **25 Days** | |
| **Phase 3. Prepare UTC & STC for Testing the software** | **15 Days** | |
| **Phase 4. Develop the software** | **32 Days** | |
| **Phase 5. Test the software** | **13 Days** | |
| **Phase 6. Demonstrate the software to users & implement it** | | **12 Days** | |

**Chapter 7: Software Design**

**7.1. Scope**

In this section we define the scope of the design effort. The design phase is an important part of the system development phase. A good design of the system needs creativity and flair from the designer and is the key to effective and successful engineering. The following are the basic objectives of the software design process:

* To describe the process of software design where informal ideas are transformed into detailed implementation description.
* Introduction of different stages in the design process.
* Understanding whether an Object Oriented or a Functional Oriented approach or both should be applied to the software.
* Determining and improving, cohesion control and coupling within subsystems.

In the project, the design phase has been identified as one of the most crucial documents. In this phase, we have identified the various aspect of the “USER MANAGEMENT”, which have to be implemented as subsystems and their further components.

**7.2. System model specification**

The system model chosen for the software project is the prototyping model. It begins with requirement gathering. The overall objectives of the software are well defined, known requirements are identified and an area where further definition is mandatory has been outlined. After this the software process is initiated and a prototype is built and then it is evaluated by the customer and used to refine requirements for the software to be developed. A prototype is a working system that is developed to test the ideas about the new system and prototyping is a process of building a model of the system to be developed.

This approach is used in our project because it is difficult to know all the requirements in the beginning of the project. Such situation arises when no other system like proposed one is built earlier.

The complete system of our project is based on iterative model where the user keeps specifying the requirements according to the model made available and the changes are incorporated into the system to build a better model. In this type of approach, we will develop a prototype and show it to the user. The user verifies the prototype. In case if there are some suggestions from user then again that functionality is added to the system and again the user evaluates it. This cycle gets repeated till the time the user is completely satisfied with the prototype. Then the actual project is built and implemented thereafter.

**7.3. Class Diagram**

**7.4 Use Case Diagram**

Admin

Authority

**Civil Administrative System**

User

**7.5 ER Diagram**

**7.6. Architectural Design**

Large system is generally decomposed to smaller subsystems that account for functionality of the complete software system.

This process of identifying the sub systems and establishing a framework for sub system control and communication is called architectural design.

Architectural design is an important phase, as a bad architectural design cannot be rescued by good implementation. Following activities were performed during this stage:

* Principle sub systems that are functionally independent were identified and distinguished.
* A general model of control relationships between system parts was established.

Each subsystem was further decomposed into their sub function.

Assign Complaint

Gives Complaints

**Administration**

Gives status

Gives response

**Authority**

**User**

Uses

Uses

**Database**

Register details

**Fig. Architecture of Civil Administration**

**8. Use case Description**

**Use Case Name: Signup**

**Actors:** Doctor & Patient

**Description:** Doctor & Patient must sign up using their name, mobile no, email & password.

**The flow of Events:**

**A. Basic Flow**

1)The user chooses to Sign up Now Button.

2)The system prompts for registration from.

3)The user must fill the form using their name and their unique mobile no, email & password.

4)The system verifies the unique mobile no & email if those are never been used in that system before and send an OTP to that mobile no & email. After getting the correct OTP the system will create a new account of that user.

**B. Alternative Flows**

1) Invalid Mobile No/Email: If the mobile no or email entered by Doctor or Patient have already registered before or if it's invalid then it displays registration unsuccessful.

**Special Requirements:**

1) Minimum password length is 8 characters and must include a combination of characters including at least one non-alphabetic character.

**Pre-conditions:**

User must have a valid email and mobile no.

**Post-conditions:**

1. Signup Success: The system displays the main interface from which additional actions can be initiated by the user.

2. Signup Failure: If the Signup fails as described in any of the alternative flows above, an appropriate message is displayed and the user is not considered authenticated.

**Use Case Name: Login**

**Actors:** Doctor & Patient

**Description:** A registered user can login to the system and once authenticated, can initiate subsequent actions.

**Flow of Events**

A. **Basic Flow**

1) The registered user chooses to Login.

2) The system prompts for username (registered email/mobile no) and password.

3) The registered user submits username and password.

4) The system verifies the username and password combination. It authorizes the registered user according to the role(s) to which the user has been assigned. Then it displays the main interface and awaits subsequent action from the user.

B. **Alternative Flows**

1) **Invalid Username/Password:** The system displays the Authentication Failed message.

**Pre-conditions**

User must have registered.

**Post-conditions**

1. **Login Success:** The system displays the main interface from which additional actions can be initiated by the user.

2. **Login Failure:** If the Login fails as described in any of the alternative flows above, an appropriate message is displayed and the user is not considered authenticated.

**Use Case Name: Update Details**

**Actors**: Doctor

**Description**: Doctors should be able to add their details such as their name, age, gender, phone no, email and unique registration no given by MCI (Medical Council of India) along with their designation, specialization, experiences and also with a brief description of them. Apart from those the doctor must update their time table that on which day and on which date they are coming to which clinic.

**The flow of Events:**

**A. Basic Flow**

1)The user chooses to Login Now Button.

2)The system displays the main interface from which doctors could choose the update details button.

3)The user must fill the add your details form using their name, unique registration no and along with their designation, specialization, experiences and also with a brief description of them.

4)The user must update time to time that on which day and on which date they are coming to which clinic.

5)After filling the form doctors should press the add details button to save their details in the system.

**B. Alternative Flows**

1) Invalid Registration No: If the Registration No entered by Doctor is invalid then the system should notify him.

**Pre-conditions:**

1)The doctor must have registered.

2)The doctor must have unique registration no.

**Post-conditions**

1. Add Details Successfully: The system displays a toast message as add details successfully.

2. Add Details Failure: If the Add details fail as described in any of the alternatives flows above, an appropriate message is displayed and the doctor is not considered authenticated.

**Use Case Name: Search Doctor**

**Actors**: Patient

**Description**: Patient can search for top experienced doctors from dermatologists, medicine specialists, gynecologists, oncologists and more as well as the renewed surgeon using this app or by popular doctor’s name or by their nearby location.

**The flow of Events:**

**A. Basic Flow**

1)The user chooses to Login Now/Skip Now Button.

2)The system displays the main interface from which the patient could search for the doctor by their name.

3)The patient could search for doctors according to their specialists.

4)The patient could search for doctors according to their nearby location.

**B. Alternative Flows**

1) Search Unsuccessful: If the name entered by the patient is not found in the database then the system will show a message as not found.

**Pre-conditions:**

1) The patient should enter a valid name in the search engine.

**Post-conditions:**

1. Search Successful: The system displays the profile of those doctors

searched by the patient.

2. Search Failure: If the searching fails as described in any of the alternatives flows above, an appropriate message is displayed as please input the correct name.

**Use Case Name: View Profile**

**Actors: Patient**

**Description:** Patients could find the right doctor by viewing the doctor's profile.

**Flow of Events**

1. **Basic Flow:**

* The registered user chooses to view profile
* The system prompts for adding the name of the doctor
* The user submits the details
* The system generates a list of doctors who are already registered with the system.
* The user chooses one of the doctors from the list to see the complete details of the doctor.

**B. Alternate Flow**

The system will generate error blank field.

**Special Requirements:**

        The doctor must have added his details.

**Pre-Conditions:**

        The user must be a registered user.

**Post-Conditions:**

The system directs the user to the profile page of the doctor.

**Use Case Name: Book Appointment**

**Actors:** Patient

**Description:** A registered user can book an appointment for his required date and time.

**Flow of Events**

**A. Basic Flow**

1.register user chooses on book appointment.

* system prompts a search icon to enter either disease name or the doctor name and also below the icon the system displays some of the frequent disease or doctor name.
* user submits either the name of the doctor or the disease name.
* system generates a list of available slots for booking an appointment.
* The user chooses on a slot of his requirement among the list and proceed further.
* The system generates appointment no. for future reference.
* The system asks for another appointment if any.

1. **Alternative Flow**

If the user chooses to cancel the appointment process in between an appropriate message will be prompted.

**Pre-Conditions:**

1.The user must have registered.

**Post-Conditions:**

1.Success: The system generates an appointment letter.

2.Failure: The system cancels the appointment if any delay arises during the transaction.

**Use Case Name: Display Appointment Letter**

**Actor:** Patient

**Description:** A registered user will print the appointment letter.

**Flow of Events:**

**A. Basic Flow**

1. Once the slot is booked a preview of the booked slot will be displayed.

2. The preview will be having a print option.

3. The registered user selects to print and the appointment preview will be printed.

**B. Alternative Flow:**

1. If there is no printer the appointment preview will be downloaded as a softcopy.

**Special Requirements:**

1. The user should be having a printer.

**Pre-Condition:**

1. The user should have a booked slot.

**Post-Condition:**

1. Success: The document is downloaded or printed successfully.

2. Failure: The document is neither printed nor downloaded.

**Chapter 9. Coding/Code Templates**

**Android Platform:**

**Home Page:**

Frontend**:**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="#fff"  
 android:padding="10dp"  
 tools:context=".MainActivity">  
  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="90dp"  
 android:layout\_marginTop="30dp"  
 android:src="@drawable/logo"  
 android:scaleType="fitCenter"  
 ></ImageView>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:weightSum="2"  
 android:layout\_marginTop="50dp"  
 android:layout\_weight="3">  
  
  
 <LinearLayout  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:textAlignment="center"  
 android:orientation="vertical">  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:textAlignment="center"  
 android:fontFamily="@font/abhaya\_libre\_extrabold"  
 android:text="Are you a doctor? "  
 android:textColor="#050505"  
 android:textSize="20sp"  
 android:textStyle="bold" />  
  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_alignParentEnd="true"  
 android:layout\_centerVertical="true"  
 android:padding="10dp"  
 android:drawableTop="@drawable/ic\_doctor"  
 android:background="@drawable/round\_btn"  
 android:fontFamily="@font/maven\_pro\_medium"  
 android:onClick="doctorRegister\_btn"  
 android:textStyle="bold"  
 android:layout\_marginRight="2dp"  
 android:layout\_marginTop="5dp"  
 android:textColor="#FF6C02"  
 android:text="R e g i s t e r H e r e"  
 android:textAlignment="center"  
 android:textSize="13sp"/>  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:orientation="vertical">  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fontFamily="@font/abhaya\_libre\_extrabold"  
 android:text="Hello, User"  
 android:textAlignment="center"  
 android:textColor="#050505"  
 android:textSize="20sp"  
 android:textStyle="bold" />  
  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_alignParentEnd="true"  
 android:layout\_centerVertical="true"  
 android:layout\_marginLeft="2dp"  
 android:padding="10dp"  
 android:layout\_marginTop="5dp"  
 android:drawableTop="@drawable/ic\_man"  
 android:background="@drawable/round\_btn"  
 android:fontFamily="@font/maven\_pro\_medium"  
 android:id="@+id/main\_join\_now\_btn"  
 android:text="S i g n U p N o w"  
 android:textAlignment="center"  
 android:textStyle="bold"  
 android:textColor="#FF6C02"  
 android:textSize="13sp" />  
  
  
 </LinearLayout>  
  
  
  
 </LinearLayout>  
  
  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/abhaya\_libre\_extrabold"  
 android:text="OR"  
 android:layout\_marginBottom="10dp"  
 android:textAlignment="center"  
 android:textColor="#9B9FBB"  
 android:textSize="15sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/abhaya\_libre\_extrabold"  
 android:text="Already have an account ?"  
 android:layout\_marginBottom="10dp"  
 android:textAlignment="center"  
 android:textColor="#050505"  
 android:textSize="20sp"  
 android:textStyle="bold" />  
  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textAlignment="center"  
 android:layout\_centerVertical="true"  
 android:layout\_alignParentEnd="true"  
 android:fontFamily="@font/maven\_pro\_medium"  
 android:id="@+id/main\_login\_btn"  
 android:background="@drawable/round\_btn"  
 android:textColor="#FF6C02"  
 android:textStyle="bold"  
 android:text="L o g i n N o w"  
 android:layout\_marginBottom="80dp"  
 android:textSize="13sp" />  
  
 <!--  
 <Space  
 android:layout\_width="match\_parent"  
 android:layout\_height="10dp" />  
  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/abhaya\_libre\_extrabold"  
 android:text="Be My Visitor ?"  
 android:textAlignment="center"  
 android:textColor="#050505"  
 android:textSize="17sp"  
 android:textStyle="bold" />  
  
  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerVertical="true"  
 android:layout\_alignParentEnd="true"  
 android:fontFamily="@font/maven\_pro\_medium"  
 android:onClick="skip\_btn"  
 android:textAlignment="center"  
 android:background="@drawable/round\_btn"  
 android:textColor="#FF6C02"  
 android:layout\_marginBottom="30dp"  
 android:textStyle="bold"  
 android:text="s k i p N o w"  
 android:textSize="13sp" />  
  
-->  
  
 </LinearLayout>

Backend**:**

package com.example.demo;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.content.Context;  
import android.content.Intent;  
import android.net.ConnectivityManager;  
import android.net.NetworkInfo;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Button;  
import android.widget.Toast;  
  
import com.example.demo.Model.Users;  
import com.example.demo.Prevalent.Prevalent;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
  
import io.paperdb.Paper;  
  
public class MainActivity extends AppCompatActivity {  
  
 private Button joinNowButton, loginButton;  
 private ProgressDialog loadingBar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 checkConnection();  
  
 joinNowButton = (Button) findViewById(R.id.*main\_join\_now\_btn*);  
 loginButton = (Button) findViewById(R.id.*main\_login\_btn*);  
 loadingBar = new ProgressDialog(this);  
  
 Paper.*init*(this);  
  
  
 loginButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view)  
 {  
 Intent intent = new Intent(MainActivity.this, LoginActivity.class);  
 startActivity(intent);  
 }  
 });  
  
  
 joinNowButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view)  
 {  
 Intent intent = new Intent(MainActivity.this, SignupActivity.class);  
 startActivity(intent);  
 }  
 });  
  
 String UserPhoneKey = Paper.*book*().read(Prevalent.*UserPhoneKey*);  
 String UserPasswordKey = Paper.*book*().read(Prevalent.*UserPasswordKey*);  
  
 if (UserPhoneKey != "" && UserPasswordKey != "")  
 {  
 if (!TextUtils.*isEmpty*(UserPhoneKey) && !TextUtils.*isEmpty*(UserPasswordKey))  
 {  
 AllowAccess(UserPhoneKey, UserPasswordKey);  
  
 loadingBar.setTitle("Already Logged in");  
 loadingBar.setMessage("Please wait.....");  
 loadingBar.setCanceledOnTouchOutside(false);  
 loadingBar.show();  
 }  
 }  
 }  
  
 // Checking internet connection  
  
 public void checkConnection(){  
  
 ConnectivityManager manager = (ConnectivityManager)  
 getApplicationContext().getSystemService(Context.*CONNECTIVITY\_SERVICE*);  
  
 NetworkInfo activeNetwork = manager.getActiveNetworkInfo();  
  
 if (null != activeNetwork){  
 if(activeNetwork.getType() == ConnectivityManager.*TYPE\_WIFI*){  
 Toast.*makeText*(this, "Wifi Enabled", Toast.*LENGTH\_SHORT*).show();  
 }  
 else if(activeNetwork.getType() == ConnectivityManager.*TYPE\_MOBILE*){  
 Toast.*makeText*(this, "Data Network Enabled", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 }  
 else {  
 Intent intent = new Intent(MainActivity.this,NoInternetConnectionLottieAnimation.class);  
 startActivity(intent);  
 }  
 }  
  
  
  
  
 private void AllowAccess(final String phone,final String password) {  
  
 final DatabaseReference RootRef;  
 RootRef = FirebaseDatabase.*getInstance*().getReference();  
  
  
 RootRef.addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot)  
 {  
 if (dataSnapshot.child("Users").child(phone).exists())  
 {  
 Users usersData = dataSnapshot.child("Users").child(phone).getValue(Users.class);  
  
 if (usersData.getPhone().equals(phone))  
 {  
 if (usersData.getPassword().equals(password))  
 {  
 Toast.*makeText*(MainActivity.this, "Please wait, you are already logged in...", Toast.*LENGTH\_SHORT*).show();  
 loadingBar.dismiss();  
  
 Intent intent = new Intent(MainActivity.this, NavigationDrawerActivity.class);  
 Prevalent.*currentOnlineUser* = usersData;  
 startActivity(intent);  
 }  
 else  
 {  
 loadingBar.dismiss();  
 Toast.*makeText*(MainActivity.this, "Password is incorrect.", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 }  
 else  
 {  
 Toast.*makeText*(MainActivity.this, "Account with this " + phone + " number do not exists.", Toast.*LENGTH\_SHORT*).show();  
 loadingBar.dismiss();  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
  
 public void doctorRegister\_btn(View view) {  
 Intent intent=new Intent(MainActivity.this,DoctorRegistrationForm.class);  
 startActivity(intent);  
 }  
  
/\*  
  
 public void skip\_btn(View view) {  
 Intent intent=new Intent(MainActivity.this,CreateDoctorProfile.class);  
 startActivity(intent);  
 }  
  
  
 \*/  
}

**Doctor Add Details:**

Frontend**:**

<?xml version="1.0" encoding="utf-8"?>  
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/background"  
 android:orientation="vertical"  
 tools:context=".Upload\_Doctors">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:padding="30dp"  
 android:orientation="vertical"  
 android:layout\_height="wrap\_content">  
  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="200dp"  
 android:scaleType="centerCrop"  
 android:id="@+id/iv\_doctorImage"  
 android:src="@drawable/imagenotavailable"  
 />  
  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Full Name"  
 android:textSize="15sp"  
 android:layout\_marginTop="20dp"  
 android:id="@+id/txt\_doctor\_name"  
 />  
  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Specialization"  
 android:layout\_marginTop="10dp"  
 android:textSize="15sp"  
 android:id="@+id/text\_specialization"  
 />  
  
  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Gender"  
 android:textSize="15sp"  
 android:layout\_marginTop="10dp"  
 android:id="@+id/txt\_doctor\_gender"  
 />  
  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="15sp"  
 android:hint="Brief Description(Experience,Age,Achievements etc..)"  
 android:layout\_marginTop="10dp"  
 android:id="@+id/txt\_doctor\_location"  
 />  
  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Days"  
 android:textSize="15sp"  
 android:layout\_marginTop="10dp"  
 android:id="@+id/txt\_doctor\_days"  
 />  
  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Price"  
 android:inputType="number"  
 android:textSize="15sp"  
 android:layout\_marginTop="10dp"  
 android:id="@+id/txt\_doctor\_price"  
 />  
  
  
  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Locations(With Full Address Of That Pharmacy)"  
 android:layout\_marginBottom="10dp"  
 android:layout\_marginTop="10dp"  
 android:textSize="15sp"  
 android:id="@+id/text\_description"  
 />  
 <TextView  
 android:layout\_marginTop="5dp"  
 android:layout\_marginBottom="5dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="#FF6C02"  
 android:fontFamily="@font/amethysta"  
 android:text="\*All fields are mandatory"  
 android:textAlignment="textEnd"  
 android:textSize="10sp"  
 android:textStyle="bold" />  
  
 <Button  
 android:layout\_marginTop="13dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="S e l e c t I m a g e"  
 android:textSize="13sp"  
 android:textAlignment="center"  
 android:fontFamily="@font/maven\_pro\_medium"  
 android:background="@drawable/round\_button"  
 android:textColor="#fff"  
 android:onClick="btnSelectImage"  
 />  
  
 <Space  
 android:layout\_width="match\_parent"  
 android:layout\_height="10dp" />  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="U p l o a d"  
 android:textAlignment="center"  
 android:fontFamily="@font/maven\_pro\_medium"  
 android:textSize="13sp"  
 android:background="@drawable/round\_button"  
 android:textColor="#fff"  
 android:onClick="btnUploadDoctors"  
 />  
  
  
 </LinearLayout>  
  
</ScrollView>

Backend**:**

package com.example.demo;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.OnFailureListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.storage.FirebaseStorage;  
import com.google.firebase.storage.StorageReference;  
import com.google.firebase.storage.UploadTask;  
  
import java.text.DateFormat;  
import java.util.Calendar;  
  
public class Upload\_Doctors extends AppCompatActivity {  
  
 ImageView doctorImage;  
 Uri uri;  
 EditText txt\_name, txt\_specialization, txt\_gender, txt\_locations, txt\_days, txt\_price, txt\_description;  
 String imageUrl;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_upload\_\_doctors*);  
  
 doctorImage = (ImageView) findViewById(R.id.*iv\_doctorImage*);  
 txt\_name = (EditText) findViewById(R.id.*txt\_doctor\_name*);  
 txt\_specialization = (EditText) findViewById(R.id.*text\_specialization*);  
 txt\_gender = (EditText) findViewById(R.id.*txt\_doctor\_gender*);  
 txt\_locations = (EditText) findViewById(R.id.*txt\_doctor\_location*);  
 txt\_days = (EditText) findViewById(R.id.*txt\_doctor\_days*);  
 txt\_price = (EditText) findViewById(R.id.*txt\_doctor\_price*);  
 txt\_description = (EditText) findViewById(R.id.*text\_description*);  
  
 }  
  
 public void btnSelectImage(View view) {  
  
 Intent photoPicker = new Intent(Intent.*ACTION\_PICK*);  
 photoPicker.setType("image/\*");  
 startActivityForResult(photoPicker, 1);  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
  
 if (resultCode == *RESULT\_OK*) {  
 uri = data.getData();  
 doctorImage.setImageURI(uri);  
 } else Toast.*makeText*(this, "You Have Not Picked Any Image", Toast.*LENGTH\_SHORT*).show();  
  
 }  
  
 public void uploadImage() {  
  
 StorageReference storageReference = FirebaseStorage.*getInstance*().getReference().child("DoctorsImage")  
 .child(uri.getLastPathSegment());  
  
 final ProgressDialog progressDialog = new ProgressDialog(this);  
 progressDialog.setMessage("Details Uploading....");  
 progressDialog.show();  
  
 storageReference.putFile(uri).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {  
  
 Task<Uri> uriTask = taskSnapshot.getStorage().getDownloadUrl();  
 while (!uriTask.isComplete()) ;  
 Uri urlImage = uriTask.getResult();  
 imageUrl = urlImage.toString();  
 uploadDoctors();  
  
 progressDialog.dismiss();  
  
 }  
 }).addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 progressDialog.dismiss();  
 }  
 });  
  
  
 }  
  
 public void btnUploadDoctors(View view) {  
  
 if (TextUtils.*isEmpty*(txt\_name.getText().toString())) {  
 Toast.*makeText*(this, "Please provide appintro\_search\_doctor full name.", Toast.*LENGTH\_SHORT*).show();  
 } else if (TextUtils.*isEmpty*(txt\_specialization.getText().toString())) {  
 Toast.*makeText*(this, "Please provide appintro\_search\_doctor specialization.", Toast.*LENGTH\_SHORT*).show();  
 } else if (TextUtils.*isEmpty*(txt\_gender.getText().toString())) {  
 Toast.*makeText*(this, "Please provide gender.", Toast.*LENGTH\_SHORT*).show();  
 } else if (TextUtils.*isEmpty*(txt\_locations.getText().toString())) {  
 Toast.*makeText*(this, "Please provide a short description.", Toast.*LENGTH\_SHORT*).show();  
 } else if (TextUtils.*isEmpty*(txt\_days.getText().toString())) {  
 Toast.*makeText*(this, "Please provide on which days you are going to this clinic.", Toast.*LENGTH\_SHORT*).show();  
 } else if (TextUtils.*isEmpty*(txt\_price.getText().toString())) {  
 Toast.*makeText*(this, "Please provide your fee.", Toast.*LENGTH\_SHORT*).show();  
 } else if (TextUtils.*isEmpty*(txt\_description.getText().toString())) {  
 Toast.*makeText*(this, "Please provide the location of that pharmacy with its full address.", Toast.*LENGTH\_SHORT*).show();  
 } else {  
 uploadImage();  
 }  
 }  
  
 public void uploadDoctors(){  
  
  
  
 DoctorDetails doctorDetails = new DoctorDetails(  
  
 txt\_name.getText().toString(),  
 txt\_specialization.getText().toString(),  
 txt\_gender.getText().toString(),  
 txt\_locations.getText().toString(),  
 txt\_days.getText().toString(),  
 txt\_price.getText().toString(),  
 txt\_description.getText().toString(),  
 imageUrl  
 );  
  
 String myCurrentDateTime = DateFormat.*getDateTimeInstance*()  
 .format(Calendar.*getInstance*().getTime());  
  
  
 FirebaseDatabase.*getInstance*().getReference("Doctors")  
 .child(myCurrentDateTime).setValue(doctorDetails).addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
  
 if (task.isSuccessful()){  
 Toast.*makeText*(Upload\_Doctors.this, "Doctor Details Upload Successfully", Toast.*LENGTH\_SHORT*).show();  
 finish();  
 }  
  
 }  
 }).addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 Toast.*makeText*(Upload\_Doctors.this, e.getMessage().toString(), Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
}

**Web Platform:**

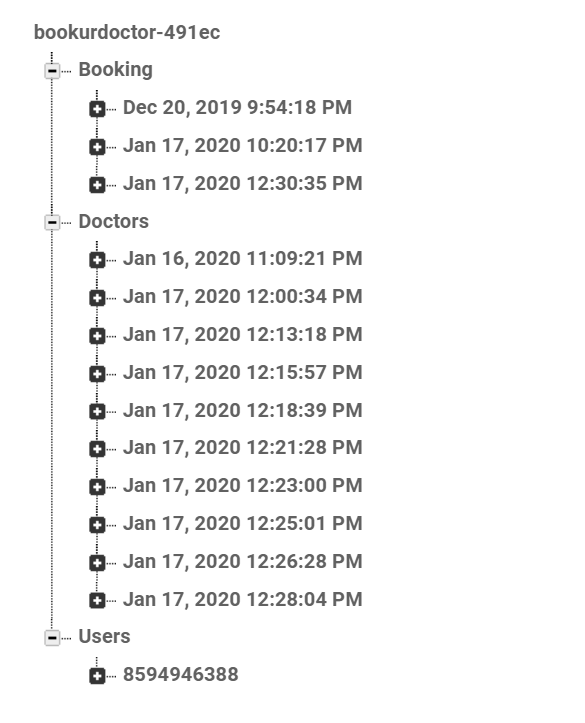
**Chapter 10: Testing**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Step Name** | **Step Description** | **Expected Result** | **Actual Result** | **Remarks** |
| TC\_001 | Verify that Password field does not allow only letters. | Step 1 | Launch the application of "Book Your Doctor System" | Home page should get displayed |  |  |
|  |  | Step 2 | Navigate to New User Registration UI | User Registration UI should get displayed |  |  |
|  |  | Step 3 | Enter valid values in all the mandatory fields and enter values in Password, Confirm Password fields as only lowercase or uppercase letters or combination of both and greater than or equals to 8 characters and click on Submit button | Validation error should get displayed as "Password is invalid. Should contain at least 1 uppercase letter, one lower case letter, 1 number and one special symbol and greater than or equals to 8 characters" | Validation error gets displayed as expected | Pass |
| TC\_002 | Verify that Password field does not allow only numbers. | Step 1 | Launch the application of "Book Your Doctor System" | Home page should get displayed |  |  |
|  |  | Step 2 | Navigate to New User Registration UI | User Registration UI should get displayed |  |  |
|  |  | Step 3 | Enter valid values in all the mandatory fields and enter values in Password, Confirm Password fields as only numbers [0-9] with greater than or equals to 8 characters and click on Submit button | Validation error should get displayed as "Password is invalid. Should contain at least 1 uppercase letter, one lower case letter, 1 number and one special symbol and greater than or equals to 8 characters" | Validation error gets displayed as expected | Pass |
| TC\_003 | Verify that Password field does not allow only special Characters. | Step 1 | Launch the application of "Book Your Doctor System" | Home page should get displayed |  |  |
|  |  | Step 2 | Navigate to New User Registration UI | User Registration UI should get displayed |  |  |
|  |  |  |  |  |  |  |
|  |  | Step 3 | Enter valid values in all the mandatory fields and enter values in Password, Confirm Password fields as only special symbols with greater than or equals to 8 characters and click on Submit button | Validation error should get displayed as "Password is invalid. Should contain at least 1 uppercase letter, one lower case letter, 1 number and one special symbol and greater than or equals to 8 characters" | Validation error gets displayed as expected | Pass |
| TC\_004 | Verify that Password field does not allow only letters and numbers | Step 1 | Launch the application of "Book Your Doctor System" | Home page should get displayed |  |  |
|  |  | Step 2 | Navigate to New User Registration UI | User Registration UI should get displayed |  |  |
|  |  | Step 3 | Enter valid values in all the mandatory fields and enter values in Password, Confirm Password fields as only lowercase or uppercase letters or combination of both along with numbers [0-9] with greater than or equals to 8 characters and click on Submit button | Validation error should get displayed as "Password is invalid. Should contain at least 1 uppercase letter, one lower case letter, 1 number and one special symbol and greater than or equals to 8 characters" | Validation error gets displayed as expected | Pass |
| TC\_005 | Verify that Password field does not allow only letters and numbers | Step 1 | Launch the application of "Book Your Doctor System" | Home page should get displayed |  |  |
|  |  | Step 2 | Navigate to New User Registration UI | User Registration UI should get displayed |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  | Step 3 | Enter valid values in all the mandatory fields and enter values in Password, Confirm Password fields as only lowercase or uppercase letters or combination of both along with numbers [0-9] with greater than or equals to 8 characters and click on Submit button | Validation error should get displayed as "Password is invalid. Should contain at least 1 uppercase letter, one lower case letter, 1 number and one special symbol and greater than or equals to 8 characters" | Validation error gets displayed as expected | Pass |
| TC\_006 | Verify that Password field does not allow only letters and special symbols | Step 1 | Launch the application of "Book Your Doctor System" | Home page should get displayed |  |  |
|  |  | Step 2 | Navigate to New User Registration UI | User Registration UI should get displayed |  |  |
|  |  | Step 3 | Enter valid values in all the mandatory fields and enter values in Password, Confirm Password fields as only lowercase or uppercase letters or combination of both along with special symbols with greater than or equals to 8 characters and click on Submit button | Validation error should get displayed as "Password is invalid. Should contain at least 1 uppercase letter, one lower case letter, 1 number and one special symbol and greater than or equals to 8 characters" | Validation error gets displayed as expected | Pass |
| TC\_007 | Verify that Password field does not allow only numbers and special symbols | Step 1 | Launch the application of "Book Your Doctor System" | Home page should get displayed |  |  |
|  |  |  |  |  |  |  |
|  |  | Step 2 | Navigate to New User Registration UI | User Registration UI should get displayed |  |  |
|  |  |  |  |  |  |  |
|  |  | Step 3 | Enter valid values in all the mandatory fields and enter values in Password, Confirm Password fields as numbers [0-9] along with special symbols with greater than or equals to 8 characters and click on Submit button | Validation error should get displayed as "Password is invalid. Should contain at least 1 uppercase letter, one lower case letter, 1 number and one special symbol and greater than or equals to 8 characters" | Validation error gets displayed as expected | Pass |
| TC\_008 | Verify that Password field does not allow only numbers and special symbols | Step 1 | Launch the application of "Book Your Doctor System" | Home page should get displayed |  |  |
|  |  | Step 2 | Navigate to New User Registration UI | User Registration UI should get displayed |  |  |
|  |  | Step 3 | Enter valid values in all the mandatory fields and enter values in Password, Confirm Password fields as numbers [0-9] along with special symbols with greater than or equals to 8 characters and click on Submit button | Validation error should get displayed as "Password is invalid. Should contain at least 1 uppercase letter, one lower case letter, 1 number and one special symbol and greater than or equals to 8 characters" | Validation error gets displayed as expected | Pass |
| TC\_009 | Verify that Password field allows combination of letters (Uppercase + lowercase), numbers and special symbols | Step 1 | Launch the application of "Book Your Doctor System" | Home page should get displayed |  |  |
|  |  | Step 2 | Navigate to New User Registration UI | User Registration UI should get displayed |  |  |
|  |  |  |  |  |  |  |
|  |  | Step 3 | Enter valid values in all the mandatory fields and enter values in Password, Confirm Password fields as combination of: 1. Uppercase letters 2. Lowercase letters 3. Numbers [0-9] 4. Special Symbols  with greater than or equals to 8 characters and click on Submit button | New user should get registered successfully and should get displayed with a unique candidate ID | Password is accepted and user registration is successful | Pass |
| TC\_010 | Verify that Password field does not allow combination of lowercase letters, numbers and special symbols | Step 1 | Launch the application of "Book Your Doctor System" | Home page should get displayed |  |  |
|  |  | Step 2 | Navigate to New User Registration UI | User Registration UI should get displayed |  |  |
|  |  | Step 3 | Enter valid values in all the mandatory fields and enter values in Password, Confirm Password fields as a combination of lowercase alphabets, numbers [0-9] along with special symbols with greater than or equals to 8 characters and click on Submit button | Validation error should get displayed as "Password is invalid. Should contain at least 1 uppercase letter, one lower case letter, 1 number and one special symbol and greater than or equals to 8 characters" | Validation error gets displayed as expected | Pass |
| TC\_011 | Verify that Password field does not allow combination of uppercase letters, numbers and special symbols | Step 1 | Launch the application of "Book Your Doctor System" | Home page should get displayed |  |  |
|  |  | Step 2 | Navigate to New User Registration UI | User Registration UI should get displayed |  |  |
|  |  | Step 3 | Enter valid values in all the mandatory fields and enter values in Password, Confirm Password fields as a combination of uppercase alphabets, numbers [0-9] along with special symbols with greater than or equals to 8 characters and click on Submit button | Validation error should get displayed as "Password is invalid. Should contain at least 1 uppercase letter, one lower case letter, 1 number and one special symbol and greater than or equals to 8 characters" | Validation error gets displayed as expected | Pass |
| TC\_012 | Verify that Password field does not allow any combination of letters (uppercase and lowercase), numbers [0-9] and special characters if the password length is less than 8 characters. | Step 1 | Launch the application of "Book Your Doctor System" | Home page should get displayed |  |  |
|  |  | Step 2 | Navigate to New User Registration UI | User Registration UI should get displayed |  |  |
|  |  | Step 3 | Enter valid values in all the mandatory fields and enter values in Password, Confirm Password fields as any combination of uppercase, lowercase alphabets, numbers [0-9] and special symbols with less than 8 characters in length and click on Submit button | Validation error should get displayed as "Password is invalid. Should contain at least 1 uppercase letter, one lower case letter, 1 number and one special symbol and greater than or equals to 8 characters" | Validation error gets displayed as expected | Pass |
| TC\_013 | Verify that appropriate validation error gets displayed when the values in Password and Confirm Password fields don't match | Step 1 | Launch the application of "Book Your Doctor System" | Home page should get displayed |  |  |
|  |  | Step 2 | Navigate to New User Registration UI | User Registration UI should get displayed |  |  |
|  |  | Step 3 | Enter valid values in all the mandatory fields and enter different values in Password, Confirm Password fields as any combination of uppercase, lowercase alphabets, numbers [0-9] and special symbols with greater than or equal to 8 characters length and click on Submit button | Validation error should get displayed as "Values of Password and Confirm Password must match" | Validation error gets displayed as expected | Pass |

**Chapter 11: Database Design View**

**Android Platform:**

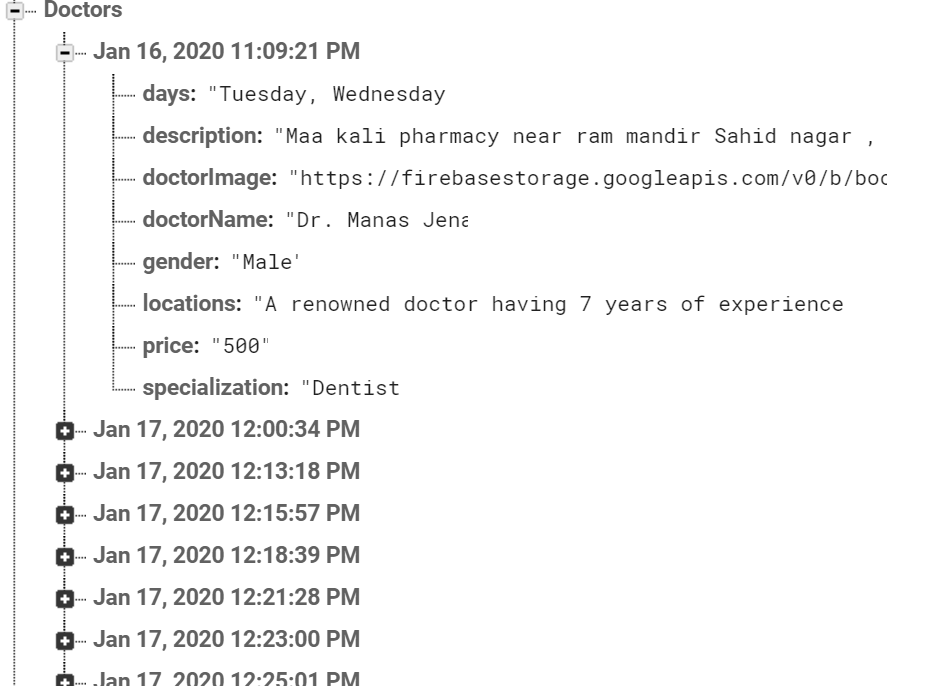
**OverView:**



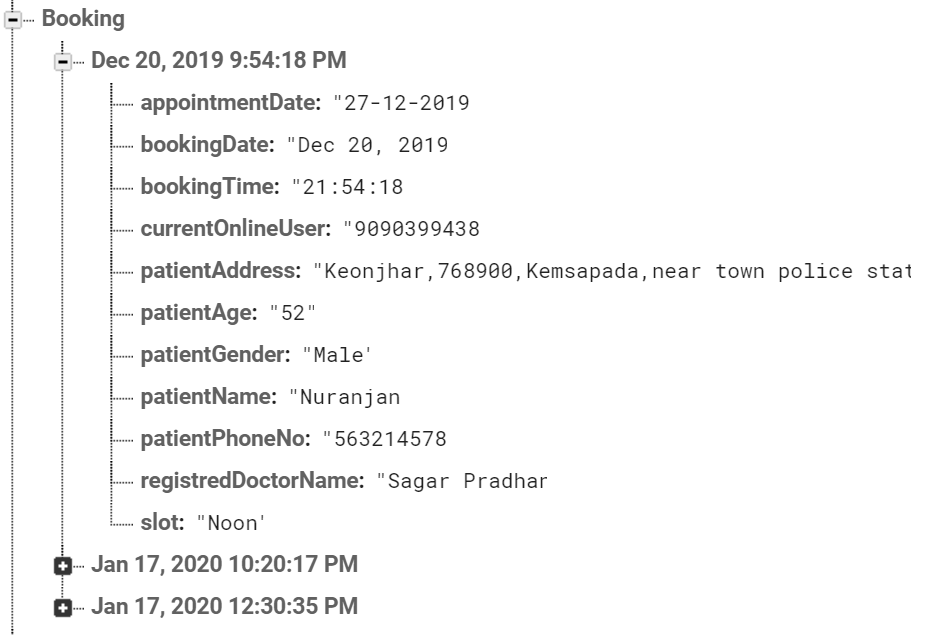
**Users:**



**Doctors:**



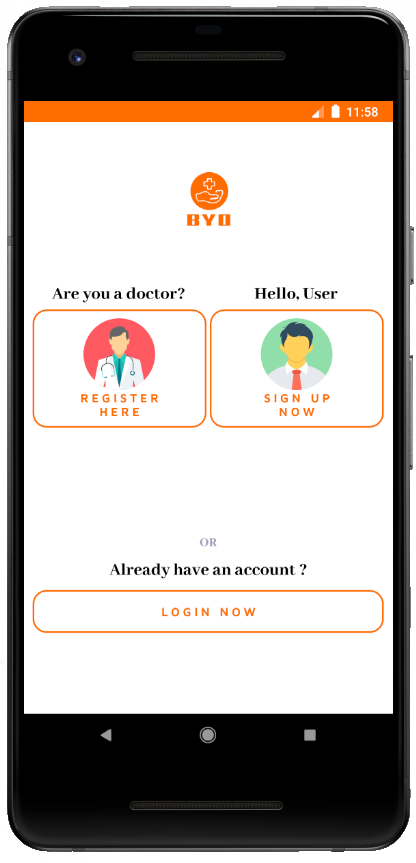
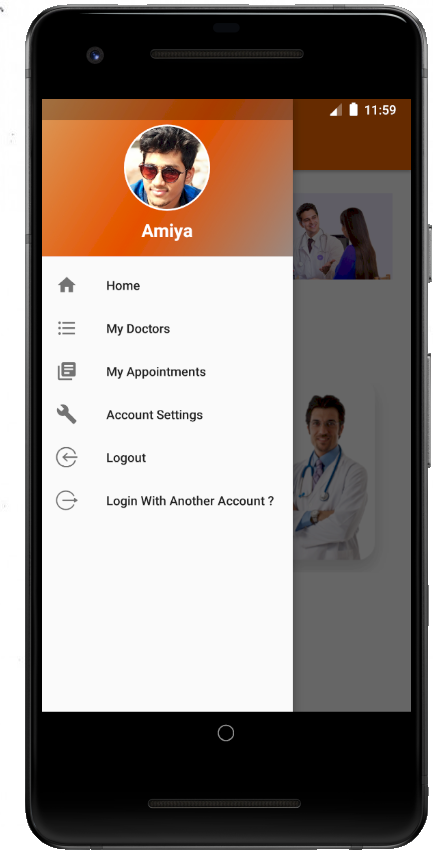
**Bookings:**

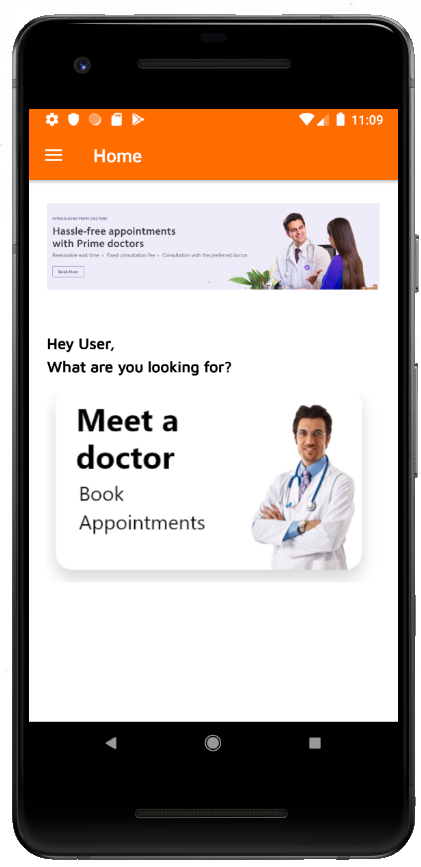
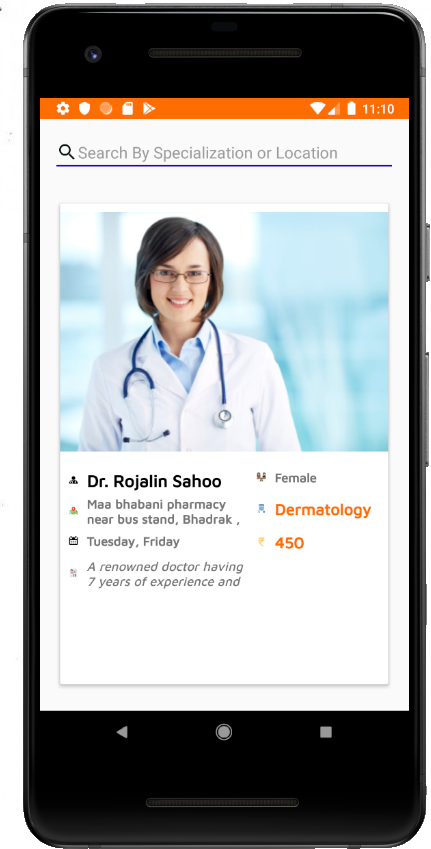


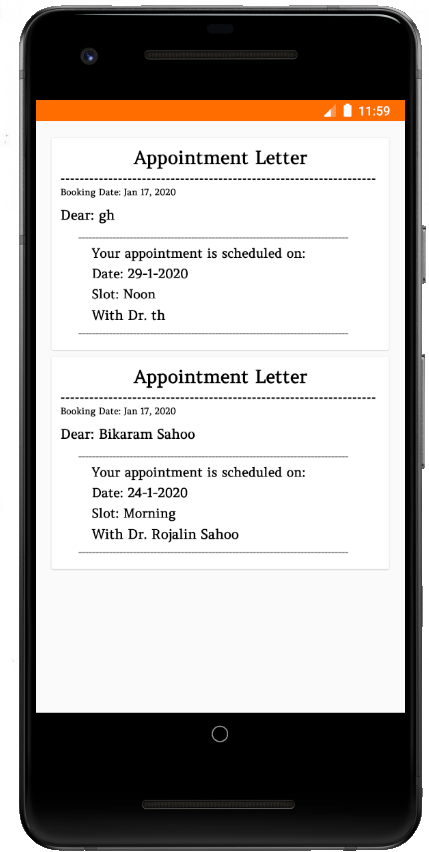
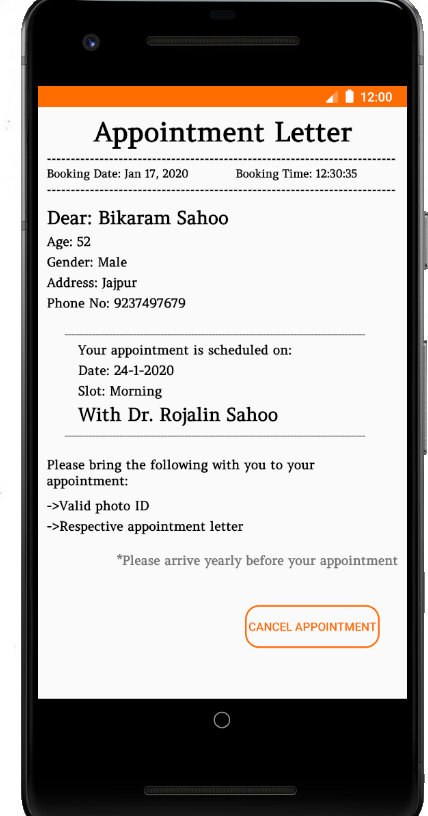
**Web Platform:**

**Chapter 12: Output Screens**

**Android Platform:**

**Web Platform:**

**Chapter 13: Conclusion**

**13.1 LIMITATIONSAND SCOPE FOR FUTURE ENHANCEMENTS:**

**13.1.1 Limitations of the System:**

* + System only works in specific platforms and its compatible environments.
  + Advanced techniques are not used to check the authorization.

**13.1.2 Future Enhancements:**

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that can be done to this system are:

* As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment.
* Because it is based on object-oriented design, any further changes can be easily adaptable.
* Based on the future security issues, security can be improved using emerging technologies.
* Sub admin module can be added
* GPS tracking can be implemented in order to track particular location from where the user registers the complain.

**13.1.3 PROJECT SUMMARY**

This application software has been computed successfully and was also tested successfully by taking “test cases”. It is user friendly, and has required options, which can be utilized by the user to perform the desired operations.

The software is developed using php as front end and mysql as back end in Windows environment . The goals that are achieved by the software are:

* Optimum utilization of resources.
* Efficient management of records.
* Simplification of the operations.
* Less processing time and getting required information.
* User friendly.
* Portable and flexible for further enhancement.

**13.2 Work Done**

**The Civil Administrative System** was successfully designed and is tested for accuracy and quality.

During this project we have accomplished all the objectives and this project meets the needs of the organization. The developed project will be used in searching, retrieving and generating information for the concerned requests.

**GOALS**

* + Reduced time needed previously for paper work
  + Easy retrieval of information
  + User friendly screens to enter the data
  + Portable and flexible for further enhancement
  + Web enabled.
  + Fast finding of information requested

**Chapter 14: References/Bibliography**

**Android Platform:**

<https://developers.google.com/android/>

<https://www.android.com/>

<https://www.youtube.com/>

<https://developer.android.com/studio/index.html>

<https://in.udacity.com/courses/android>

**Web Platform:**

<https://www.w3schools.com/>

<https://www.youtube.com/>

<https://in.udacity.com/courses/web>

**THE END**