AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY
School of Engineering

Doctor Appointment Booking System

Product Owner: Dr.Vivek Menon
Group Number: 07

Developed By:

- S.S.R.Sudhendra
- I.v.v.Raghavendra
- B. Siva Praveen
- N.Sai Sujan
- D.V.N.Sandeep

Contents

Weekly Status Report-I

Project Title: Twitter Sentiment Analysis
Product Owner: Dr.Vivek Menon

Date From: 16-12-2019 Date To: 20-01-2019

Team Members:

	Name	Roll Number
1	Sudheendra.S.S.R	AM.EN.U4CSE17062
2	Venkata Naga Sandeep.D	AM.EN.U4CSE17020
3	Sai Sujan.N	AM.EN.U4CSE17050
4	Raghavendra.I.V.V	AM.EN.U4CSE17032
5	Praveen.B.D.S	AM.EN.U4CSE17014

Question	Week: 1				
	Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
What did you do yesterday?					
What will you work on today?	made plans for the project and discussed various modules required for the project	went through various ways of implementing the modules by looking into blogs	installed the setup of the IDE and watched youtube lectures on how to use it		watched video lectures on the Naive Bayes classifier and went through probability concepts for it
Do you have any obstacles?		faced difficulties for finding the best classifier and suitable IDE for the project	faced some difficulties in using the platform which is difficult to use it in the first go		faced difficulties while understanding the concepts

Weekly Status Report-II

Project Title: Twitter Sentiment Analysis
Product Owner: Dr.Vivek Menon

Date From: 21-01-2020 Date To: 24-01-2020

Team Members:

	Name	Roll Number
1	Sudheendra.S.S.R	AM.EN.U4CSE17062
2	Venkata Naga Sandeep.D	AM.EN.U4CSE17020
3	Sai Sujan.N	AM.EN.U4CSE17050
4	Raghavendra.I.V.V	AM.EN.U4CSE17032
5	Praveen.B.D.S	AM.EN.U4CSE17014

Question	Week: 2				
	Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
What did you do yesterday?					
What will you work on today?	continued watching the lecture videos on naive Bayes classifier and requested to twitter developers for keys	Checked how to retrieve tweets data set using twitter API	we went through some articles on how to tokenize the tweets	started working on the UML diagram after knowing the flow of the project	based on the UML diagram we started writing the use cases for the project
Do you have any obstacles?		The process was completely new and was tough to understand as python was relatively new to us when compared to other programming languages.	The process of tokenizing was also new to us so we were a bit slow.		identifying the possible use cases

Weekly Status Report-III

Project Title: Doctor Appointment Booking System
Product Owner: Dr.Vivek Menon

Date From: 27-01-2020 Date To: 31-01-2020

Team Members:

	Name	Roll Number
1	Sudheendra .S.S.R	AM.EN.U4CSE17062
2	Venkata Naga Sandeep.D	AM.EN.U4CSE17020
3	Sai Sujan.N	AM.EN.U4CSE17050
4	Raghavendra.I.V.V	AM.EN.U4CSE17032
5	Praveen.B.D.S	AM.EN.U4CSE17014

Question	Week: 3				
	Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
What did you do yesterday?					
What will you work on today?	Search topics which suite software engineering requirements	Selected the project Hospital Management as it has relatively scope for a better UI	Started drawing use case diagrams		
Do you have any obstacles?	The magnitude of the project in selecting	What features should we include in?	Dividing the tasks between sprint 1 and sprint2		

Weekly Status Report-IV

Project Title: Doctor Appointment Booking System
Product Owner: Dr.Vivek Menon

Date From: 03-02-2020 Date To: 07-02-2020

Team Members:

	Name	Roll Number
1	Sudheendra .S.S.R	AM.EN.U4CSE17062
2	Venkata Naga Sandeep.D	AM.EN.U4CSE17020
3	Sai Sujan.N	AM.EN.U4CSE17050
4	Raghavendra.I.V.V	AM.EN.U4CSE17032
5	Praveen.B.D.S	AM.EN.U4CSE17014

Question	Week: 4				
	Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
What did you do yesterday?					
What will you work on today?	Worked on changes suggested in the use cases	Completed use cases,synopsis, and UML diagram	Learned how to draw sequence diagram on watching video lectures	Worked on the drawing sequence diagram	Continued working on the sequence diagram
Do you have any obstacles?	To alter the use cases according to the suggestions			Implementing the Learned concepts for the first time took us time	

Weekly Status Report-V

Project Title: Doctor Appointment Booking System
Product Owner: Dr.Vivek Menon

Date From: 10-02-2020 Date To: 14-02-2020

Team Members:

	Name	Roll Number
1	Sudheendra .S.S.R	AM.EN.U4CSE17062
2	Venkata Naga Sandeep.D	AM.EN.U4CSE17020
3	Sai Sujan.N	AM.EN.U4CSE17050
4	Raghavendra.I.V.V	AM.EN.U4CSE17032
5	Praveen.B.D.S	AM.EN.U4CSE17014

Question	Week: 5				
	Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
What did you do yesterday?					
What will you work on today?	Started Working Class Diagram and User Stories	Continued and completed Class Diagram and User Stories	Made the suggested changes in the use case diagram and sequence Diagram	Installed Netbeans IDE and went through how to work with it	Started Working on Sprint I
Do you have any obstacles?	Stucked a bit at deciding classes			Faced Some errors while Installing Netbeans	

Weekly Status Report-VI

Project Title: Doctor Appointment Booking System
Product Owner: Dr.Vivek Menon

Date From: 17-02-2020 Date To: 21-02-2020

Team Members:

	Name	Roll Number
1	Sudheendra .S.S.R	AM.EN.U4CSE17062
2	Venkata Naga Sandeep.D	AM.EN.U4CSE17020
3	Sai Sujan.N	AM.EN.U4CSE17050
4	Raghavendra.I.V.V	AM.EN.U4CSE17032
5	Praveen.B.D.S	AM.EN.U4CSE17014

Question	Week: 6				
	Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
What did you do yesterday?					
What will you work on today?	Started connecting with Database	Learned how to draw activity diagram and drew it	Completed Connecting with Databases	Completed Login and Signup pages	Completed Reset password and started Profile management
Do you have any obstacles?	Faced many errors while connecting it.			Faced Some errors while Installing Netbeans	Got errors while bringing objects from one class to another class

Weekly Status Report-VII

Project Title: Doctor Appointment Booking System Product Owner: Dr.Vivek Menon

Date From: 02-03-2020 Date To: 07-03-2020

Team Members:

	Name	Roll Number
1	Sudheendra .S.S.R	AM.EN.U4CSE17062
2	Venkata Naga Sandeep.D	AM.EN.U4CSE17020
3	Sai Sujan.N	AM.EN.U4CSE17050
4	Raghavendra.I.V.V	AM.EN.U4CSE17032
5	Praveen.B.D.S	AM.EN.U4CSE17014

Question	Week: 7				
	Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
What did you do yesterday?					
What will you work on today?	Worked on ways to write a good SRS documents	Started writing SRS	Continued writing SRS	Made suggested changes in the class diagram and worked on the Sprint 1	Completed writing SRS and made small changes in the working model
Do you have any obstacles?	Viewed a few templates and found difficulty in inferring it to our project				

PROJECT PLAN

Synopsis

PURPOSE

This application is useful in making an appointment with the doctor we choose to meet. Using these web interface users can see the list of available doctors and choose from them.

Features of this application

1)User can select a doctor from the list of available doctors
2)User can pay the appointment fee for the same using this application
3)User can see his previous appointments and his personal details and manage his profile like changing passwords etc.

Target Audience

Anyone who wants to make an appointment ie.public

USE CASES

SPRINT 1: Allow user to login to his account and manage his profile with facilities like previous appointments history and password change

- Upon clicking the register button redirect to the register page
    - Allow the user to enter his details in the fields given and create an account.

- Upon entering the credentials and clicking the login button
    - If the credentials match redirect him to the page where he can select the specialized doctor which he prefers
    - Upon failure redirect to try again / registration page.

- Upon clicking the **‘Manage Profile’** button
    - Display a page showing **appointment history**, **password change** options.
    - Upon clicking the appointment history button, show the user all the previous appointments he had.
    - Upon clicking the password change button, allow the user to change his/her password.

SPRINT 2: Allow the user to select the doctor that he prefers and complete the appointment process and the fee payment.

- Upon entering the appointment page
    - Show the various specialists available in the hospital like cardiologist, gynaecologist, dentist, etc
    - Allow the user to select the available doctor under the specified specialization.

- Upon selecting the available doctors
    - Display the content message showing his qualification, rating and appointment timings.
    - Show the rating of each doctor along with his display picture.

- Upon clicking '**take an appointment**' button
    - Redirect the user to the **Fee payment** page displaying the appointment fees.
    - Give the user **online** payment options
    - On clicking the online option display the online payment page.
    - Once the payment is over display a page showing **Appointment Successful** text

USER STORIES

- As a Product owner, I want a login/signup page, so that no unknown person can't use the application.
- As a Patient, I want to Manage details, so that I can change my details whenever I need it.
- As a Patient, I want to select a specialization of the doctor, so that I can choose the doctor easily.
- As a Patient, I want Edit Appointment, so that I can re-change the timing or date if there is any problem.
- As a Patient, I want to Cancel an Appointment, so that I can cancel the appointment if I can’t attend.
- As a Patient, I want to Forgot Password option, so that I can set a password whenever I forget it.
- As a Patient, I want my history of appointments, so that I can check my last visits.

Software Requirements Specification

Introduction

1.1Purpose

This document is to describe all the software requirement specification (SRS) for the Doctor Appointment Booking System. The system aims to help the patients to make appointments online through the internet and track their records through it. Doctor Appointment Booking System has been facing problems due to its paper-based appointment system. With the increase in the number of patients visiting, it has become difficult to manage the appointment system manually. The purpose of this project is to solve these complications by creating custom-built database software to manage the appointment system. For the user/patient, it makes it easy to set a date and time for the treatment of the patient to the relevant doctor. It also helps to maintain the doctor's consultation fee. Above all, it helps to develop a simple user interface to help the user with the doctors' arrangement.

1.2 Intended Users and Reading Suggestions

- **Developers:** In order to be sure they are developing the right project that fulfils requirements provided in this document.
- **Testers: I**n order to have an exact list of the features and functions

that have to respond according to requirements and provided diagrams.

- **Users:** In order to get familiar with the idea of the project and suggest other features that would make it even more functional.
- **Documentation writers:** to know what features and in what way they have to explain. What security technologies are required, how the system will respond in each user's action etc.
- **Patients:** in order to know exactly what they have to expect from the system, right inputs and outputs, and response in error situations.

1.3 Project Scope

The system has been facing problems due to its paper-based appointment system. So The Doctor Appointment booking system is a web application that gives a solution to the polyclinic patients. This system manages complete booking details in a single application and in a single database. The users will use this system to handle all the functionalities easily. The intentions of the system are to reduce overtime pay and increase the number of patients that can be treated accurately. Requirements statements in this document are both functional and non-functional.

1.4 References

Books: Software Engineering: A Practitioner's Approach, 7/e by Roger S. Pressman

Websites: https://netbeans.org/kb/74/index.html

https://www.smartdraw.com/uml-diagram/

2. Overall Description

2.1 Product Perspective

This software is a totally independent system that manages activities of the Appointment system such as taking appointments, generating patient reports, and profile management.

In this project all the records are stored in a single database. Different users have different permission to access this web application. Each user has a unique id making it more secure.

2.2 Product Features

- Authentication for different users.
- Real-time validation of all fields and databases to prevent errors.
- History of patients recorded in the database.
- Maintaining the billing section of the appointment.

2.3 User Classes and Characteristics

Patients: Patients will be the main users. The system is also designed to be user-friendly.
Patients can view their appointment history and doctor's details and timings. And also can make appointments online.

2.4 Assumptions and Dependencies

- The code should be free with compilation errors/syntax errors.
- The product must have an interface that is simple enough to understand.

2.5 System Features

2.5.1 Login Account

2.5.1.1Description:

To open the user account the users have to enter login information.

2.5.1.2 Stimulus/response

Users must enter a valid user id and password to open the user page. If it is valid then it links to the user account page. If the user is new he/she has to register.

2.5.1.3 Basic data flow

- Here first the user enters login id and password.
- After entering the login information system checks whether entered login id and password are valid or not.
- If it is valid then it is linked to the user account.

  If the user doesn't have a user account then the user needs to register.

2.5.2 Online appointment:

2.5.2.1 Description:

Patients can make appointments online based on the availability of the doctor on the specified date and time. Patients have to register or login to take appointments online.

2.5.3 Basic data flow

- Patient first logs into the website.
- After logging in, the patient gets multiple options like canceling, making an appointment, editing the appointment.
- After selecting a make appointment, the patient can confirm his

appointment by selecting doctors according to the timings.

- After selecting the edit option, the patient can edit his/her appointment according to doctor timings.
- He can also cancel his appointment.

2.5.3.1 Functional requirements

- Patients can make appointments online.
- patients can view the old appointment details and their records.

3. External Interface Requirement

3.1 User Interfaces

The user interface is designed in JAVA Swings and Applets. The user of the product will get a very user-friendly web page which will be very easy to work with.

3.2 Hardware Interfaces

The system should have these hardware requirements:

- Processor: Intel Pentium4 2.00GHz or above
- Memory: 512MB or above
- Hard Disk Drive: 40MB or above

3.3 Software Interfaces

- Front end: Swings, Applets
- Language: JAVA
- Back End: MySQL

3.4 Communications Interfaces

Communication is done through the internet and intranet.

4.Other Non-functional requirements

4.1 Performance Requirements

Doctor Appointment Booking System manages facilities required by casual users quickly and easily. It offers to take appointments faster through online. It takes appointment details from the patients and sends the appointment date and timings to the particular patient.

4.2 Safety Requirements

- In case the user forgets his Password, the repair functionality helps by choosing the
  "forgot password" option in the main login window.
- To avoid this kind of situation, backups can be done regularly.

4.3 Security Requirements

This system is provided with authentication without which no user can pass. So only the legitimate users are allowed to use the application. If the legitimate users share the authentication information then the system is open to outsiders.

4.4 Software Quality Attributes

Reliability: Good validations of user inputs will be done to avoid incorrect storage of records.

Maintainability: During the maintenance stage, SRS documents can be referred for any validations.

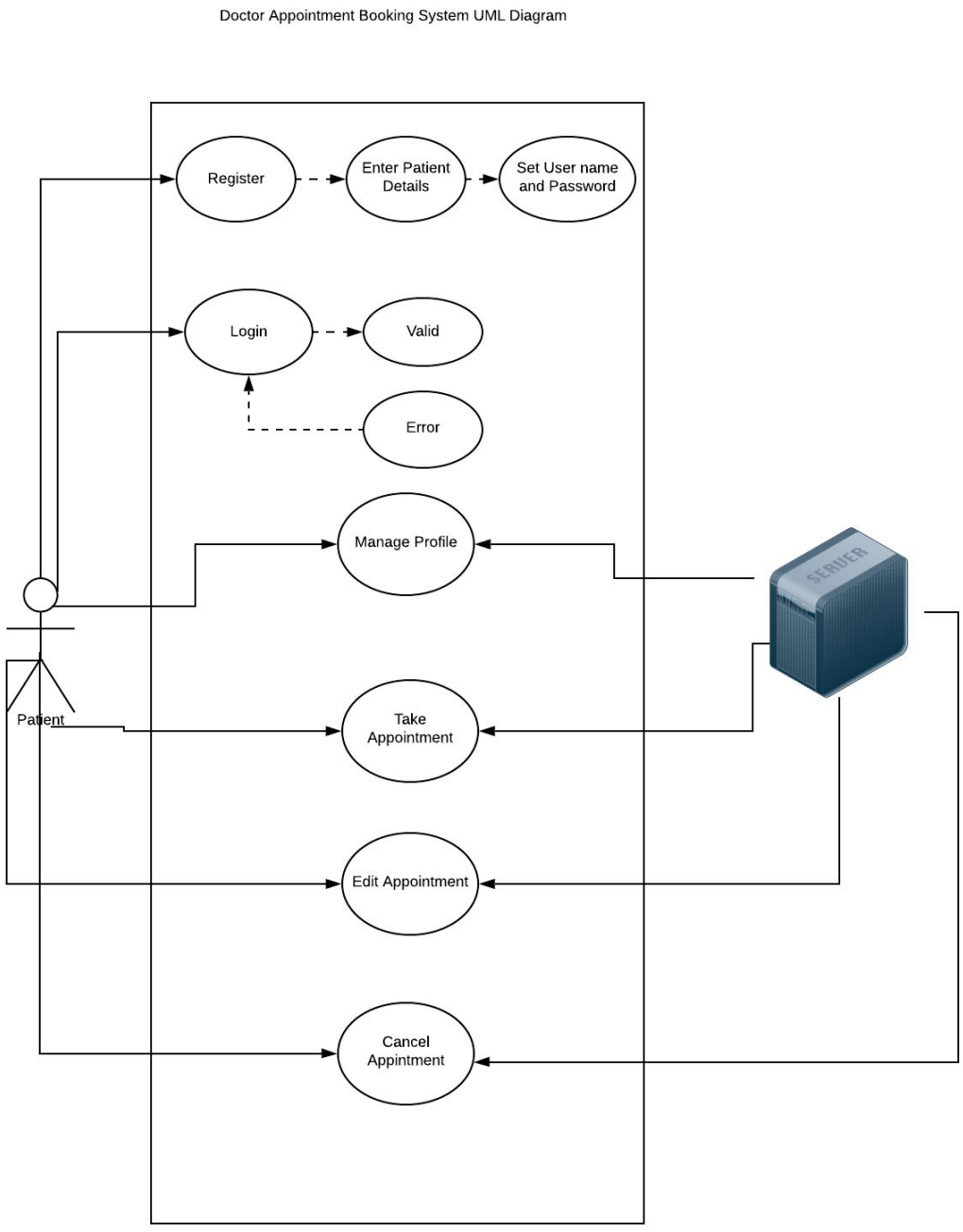
Flexibility: The system keeps on updating the data according to the transactions that take place.

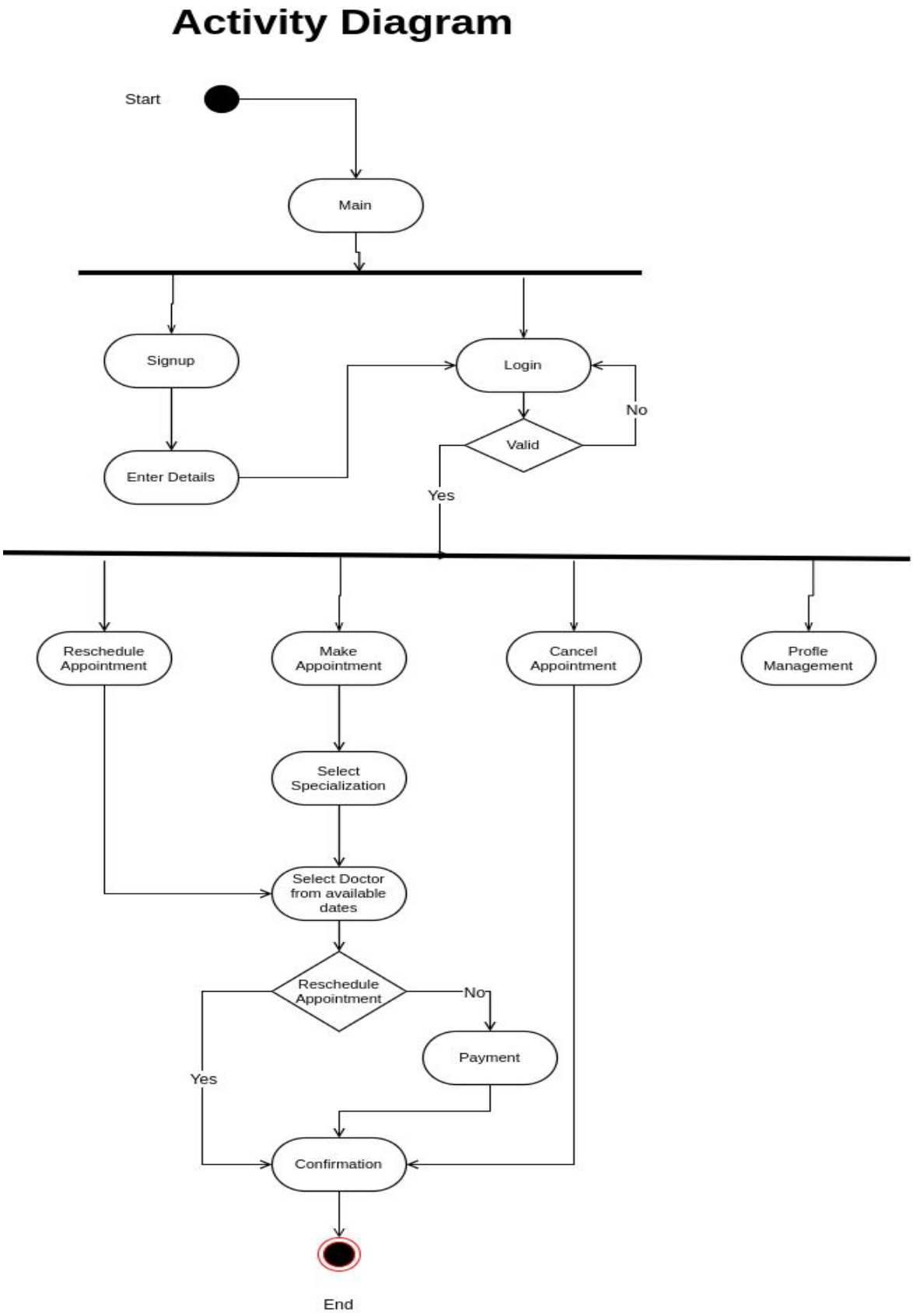
Timeless: The system carries out all the operations with consumption of very little time.

Security: The security of the system is maintained by giving access to only authenticated user id and password.

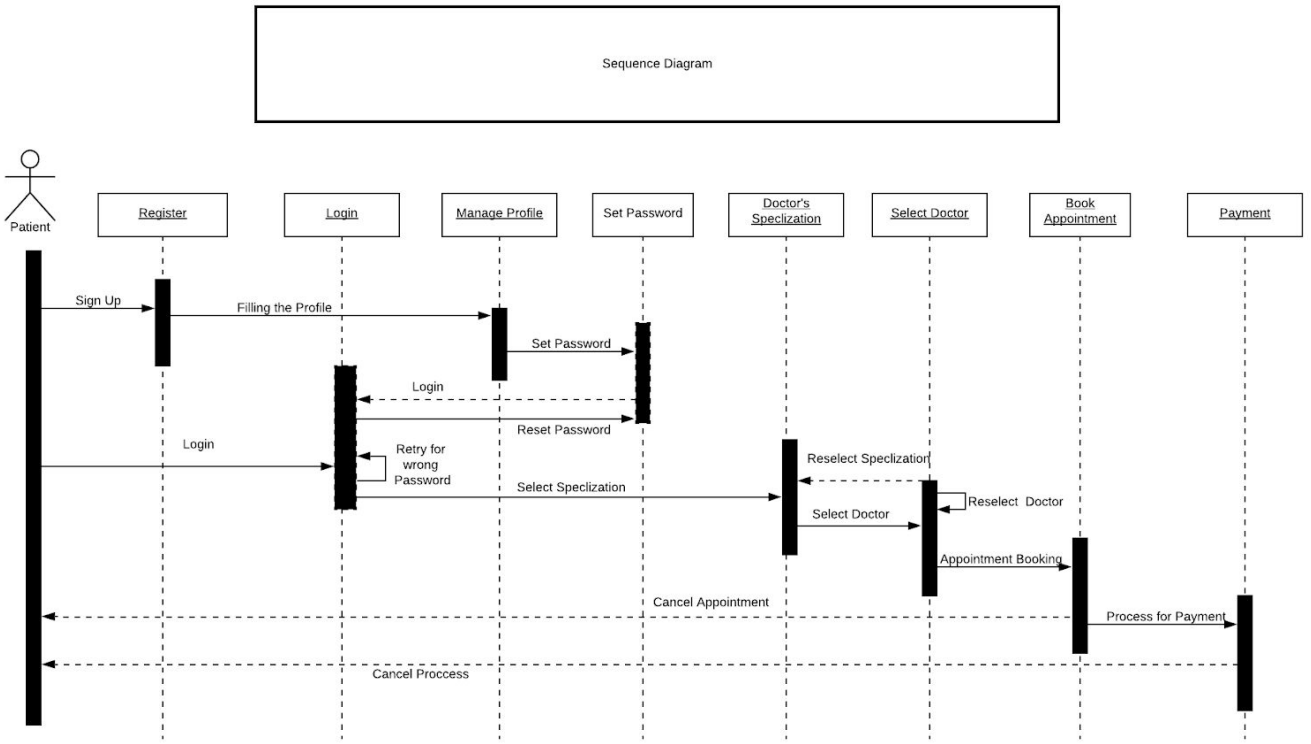
DIAGRAMS

USE CASE DIAGRAM

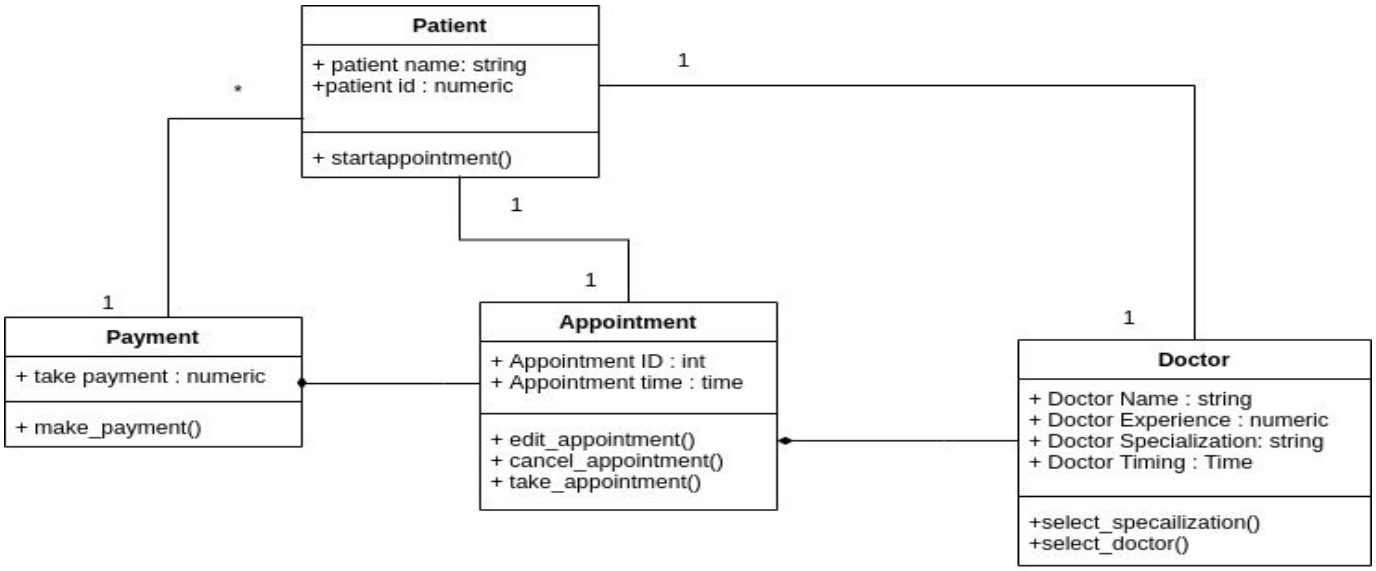



Activity Diagram
Start
Main
Signup
Login
No
Valid
Enter Details
Yes
Reschedule Appointment
Make Appointment
Cancel Appointment
Profle Management
Select Specialization
Select Doctor from available dates
Reschedule Appointment
No
Payment
Yes
Confirmation
End

SEQUENCE DIAGRAM



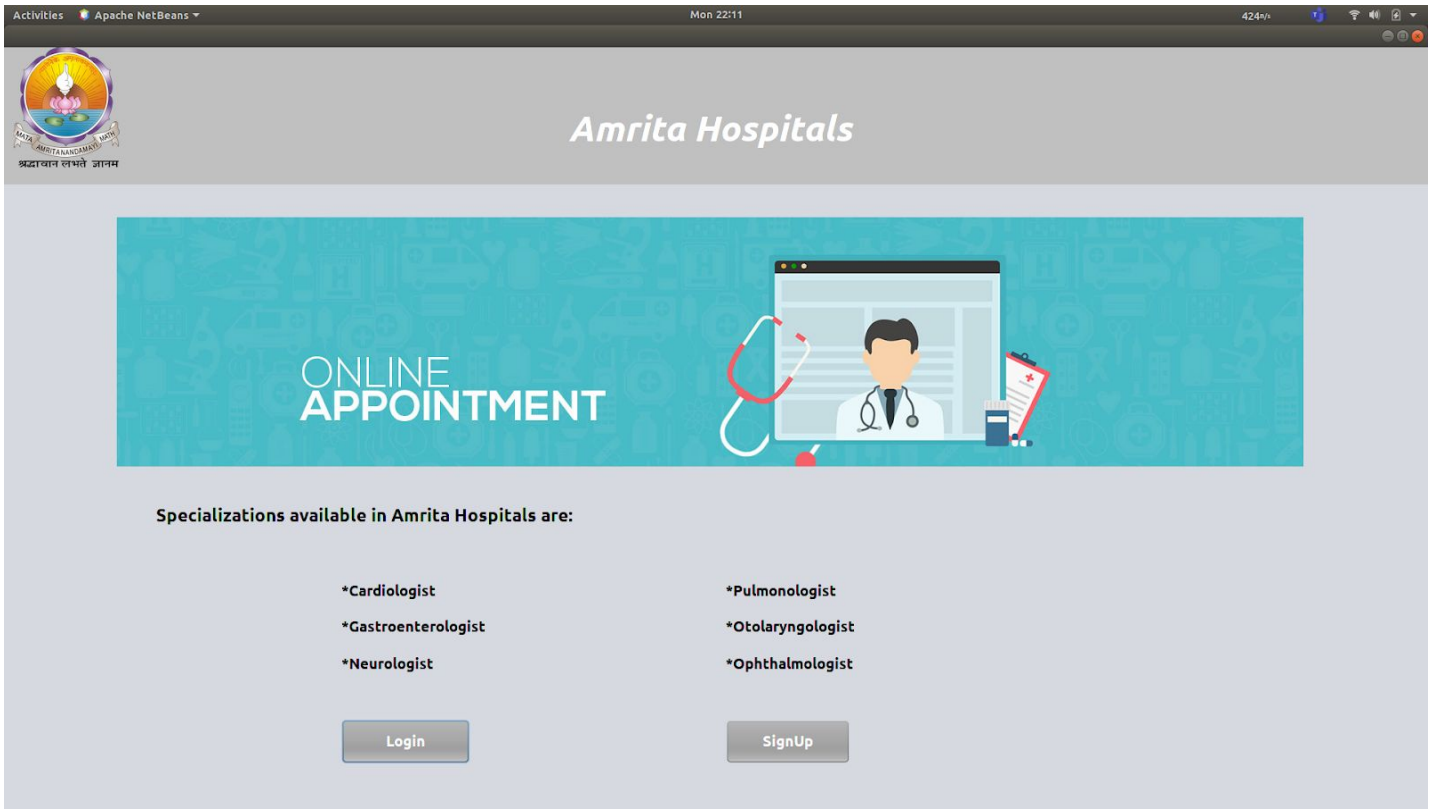
CLASS DIAGRAM



Project UI

Link of Project: DoctorAppointment Booking System

Main Page :

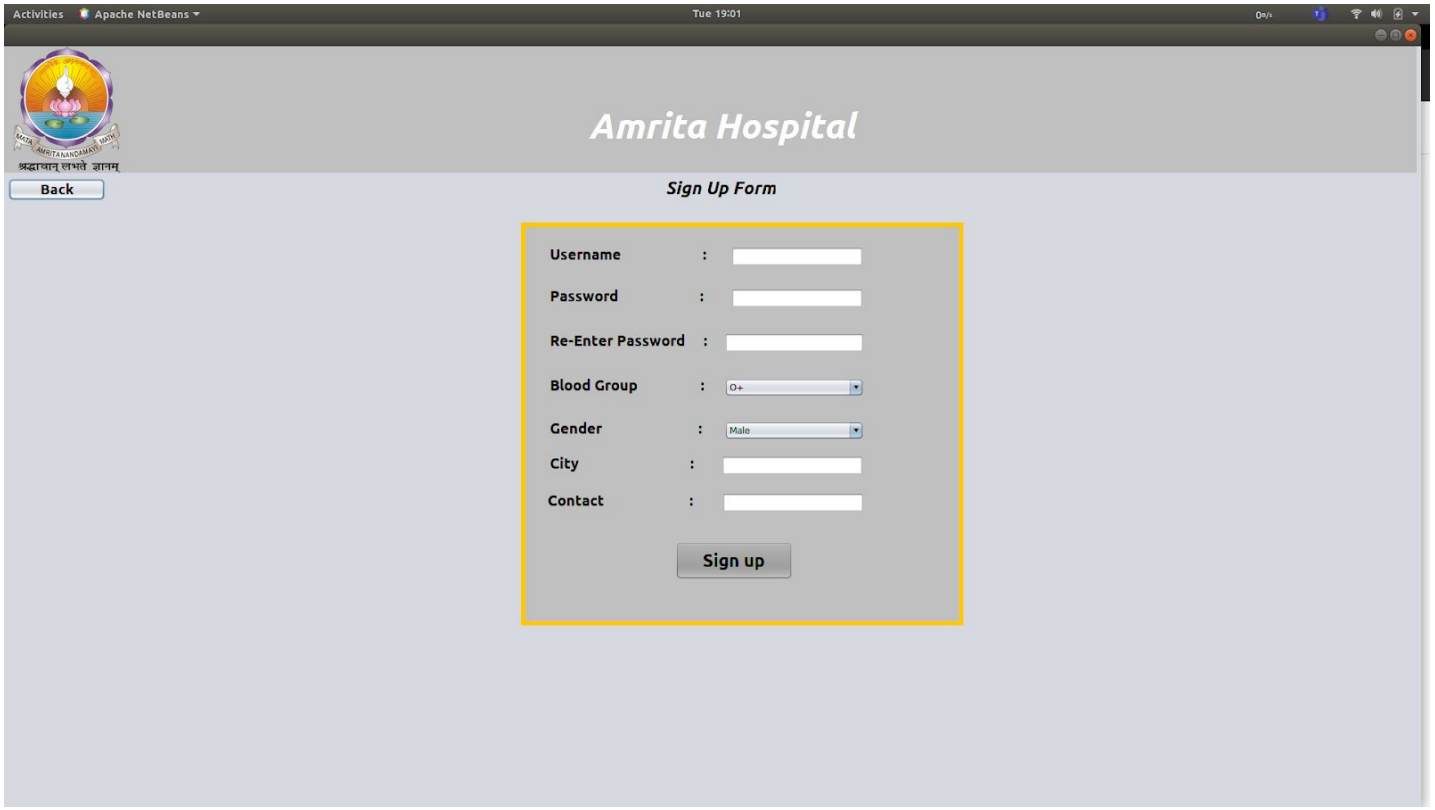


Link of the Code: Main Page

Login Page :

Link of the Code: [Login Page]

Signup Page:



Link of the Code: [SignUp Page]

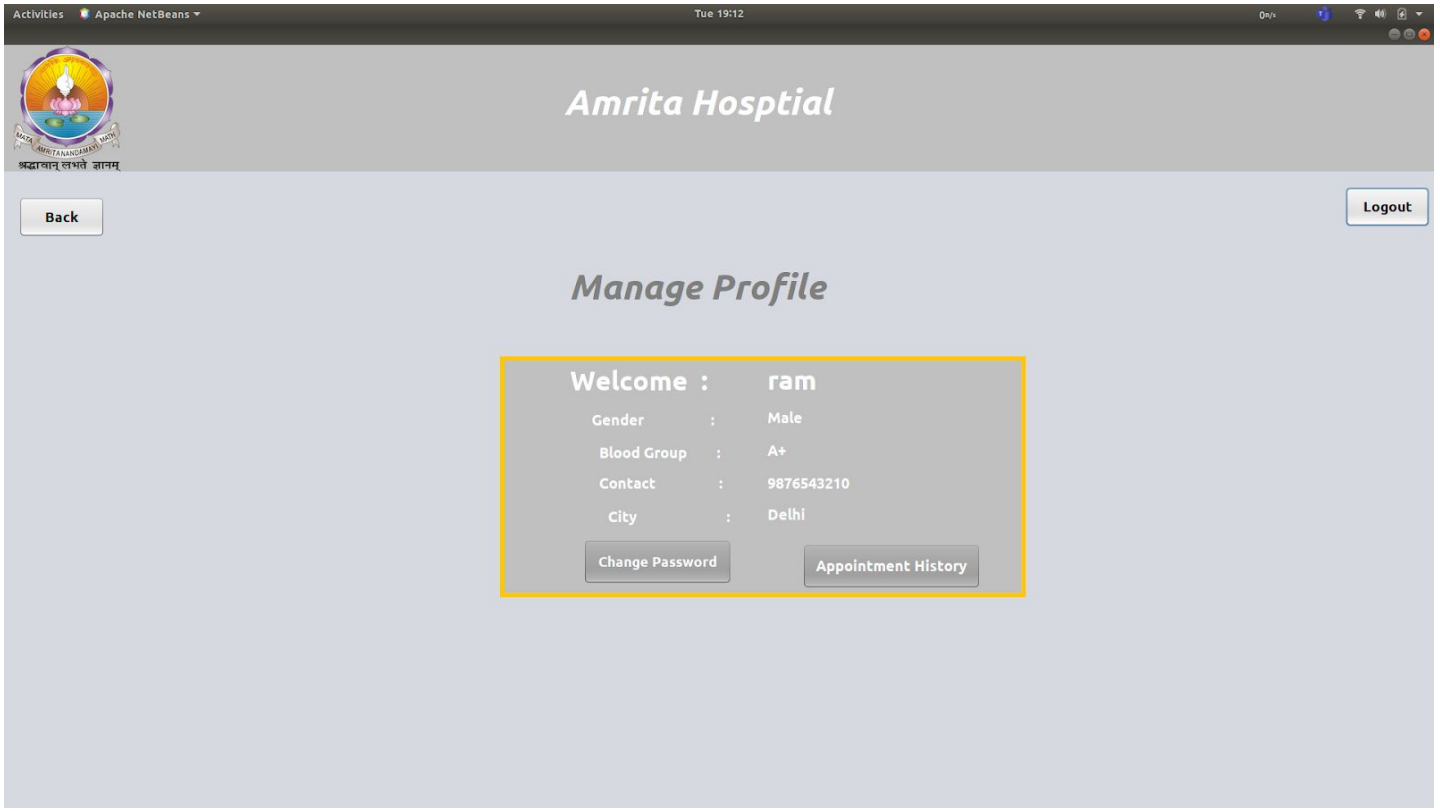
SQL Query of SIgnup:

insert into user(username,password,gender,blood_group,Contact,city)
values('username1','pass','gender1','blood',contact1,'city1');

Home Page:

Link of the Code: Home Page

Profile Page:

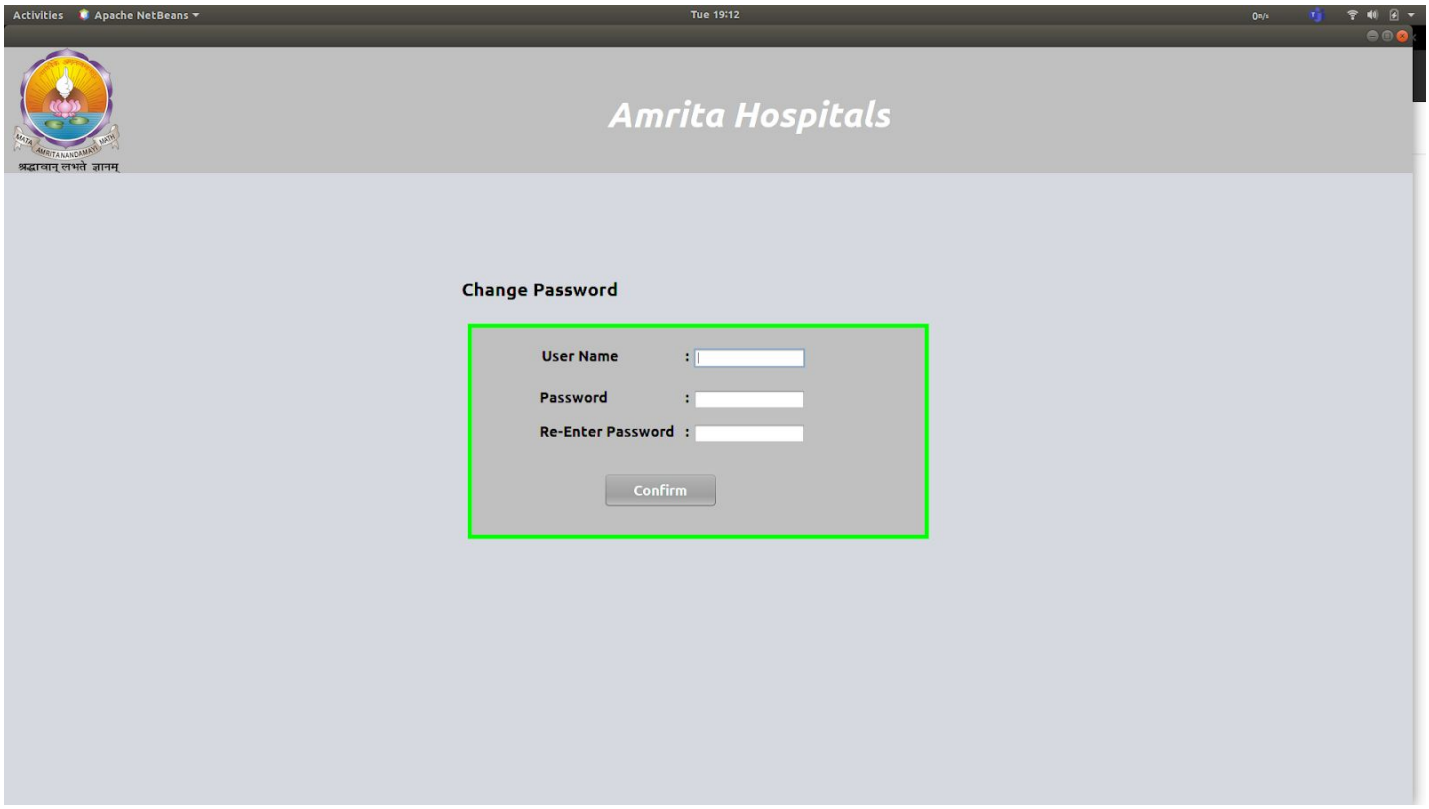


Link of code: [Profile page]

SQL Query of SIgnup:

select * from user where username='user1';

Forgot Password or Change Password :



Link of Code: [Change Password or Change Password Page]

SQL Query for Change Password :

update user set password = 'pass2' where username='username1';

Specialization Page :

Link of Code: Specialization Page

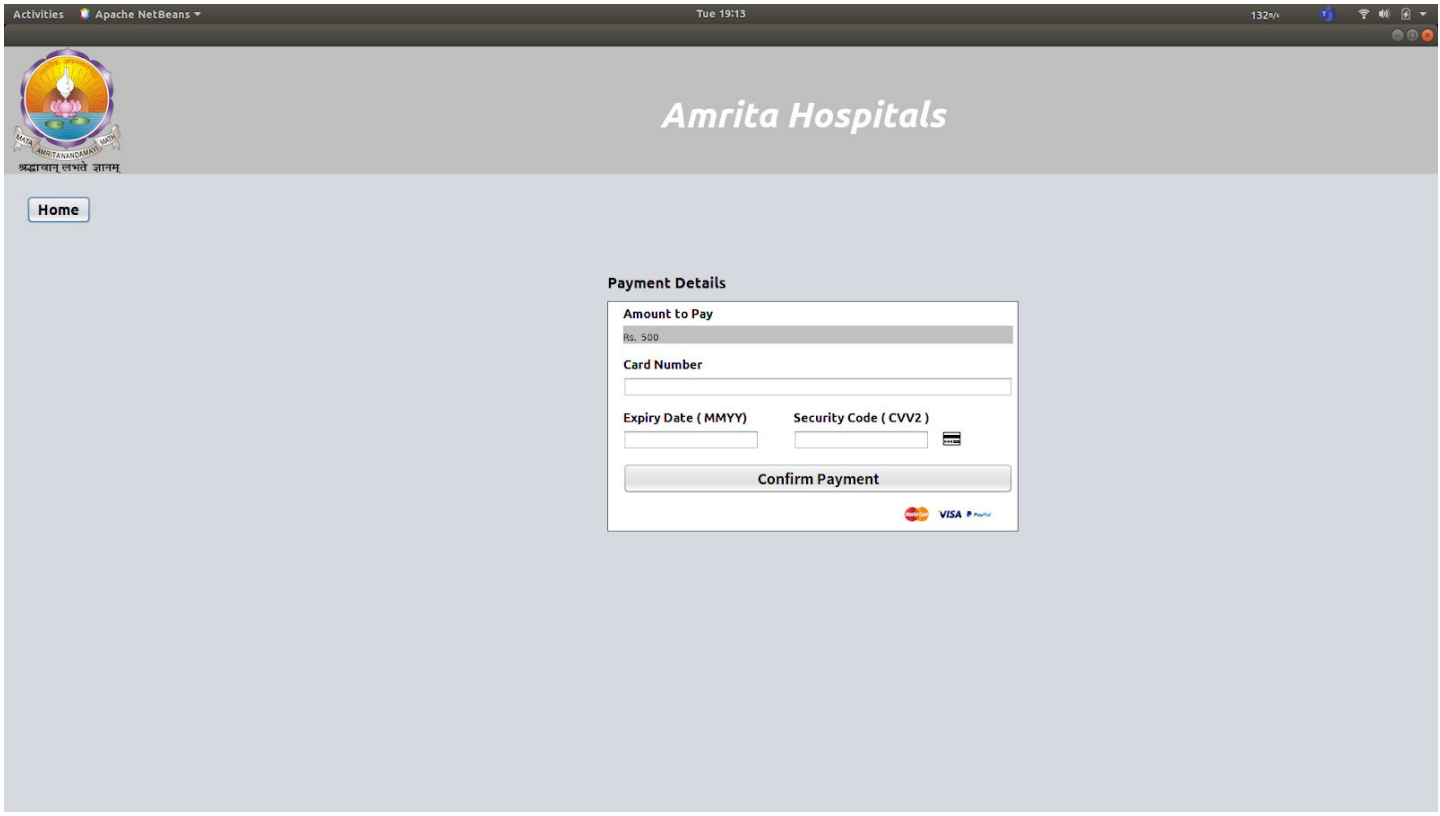
Select Doctor Page :

Link of Code: Select Doctor Page

SQL Query for View Doctors:

select * from doctor where specailization='docname1';

Payment Page :

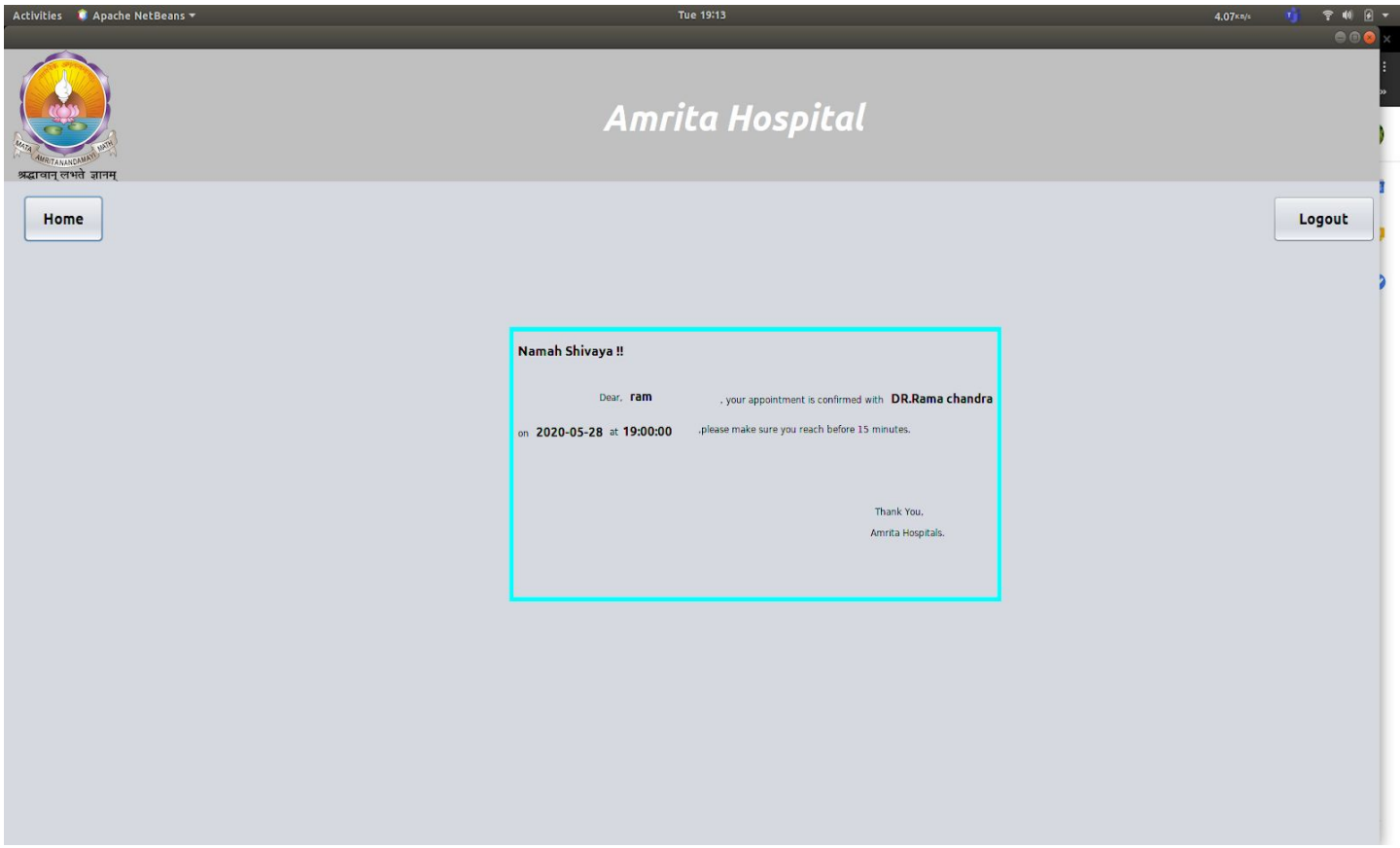


Link of Code: [Payment Page]

SQL Query after Payment :

insert into appointmentbooking(id,doc_id,dateofapp,time)
values(‘Login.id1’,r,'date2’,’rp.getTime(6)’);

Appointment Fixed Receipt :

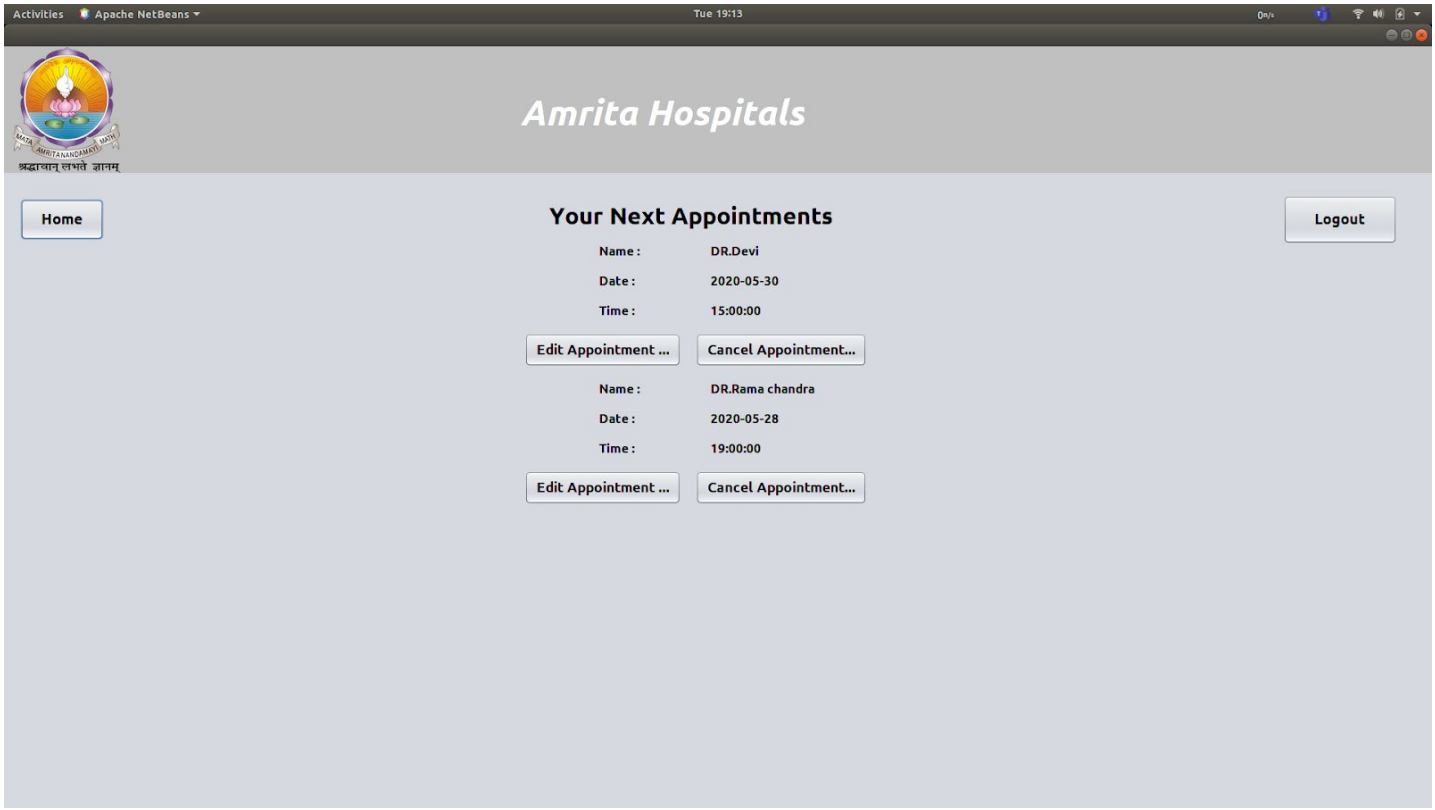


Link of Code: Appointment Receipt Page

SQL Query of Appointment Receipt:

select * from appointmentbooking where appoint_id=(select max(appoint_id) from appointmentbooking);

View Next Appointments :

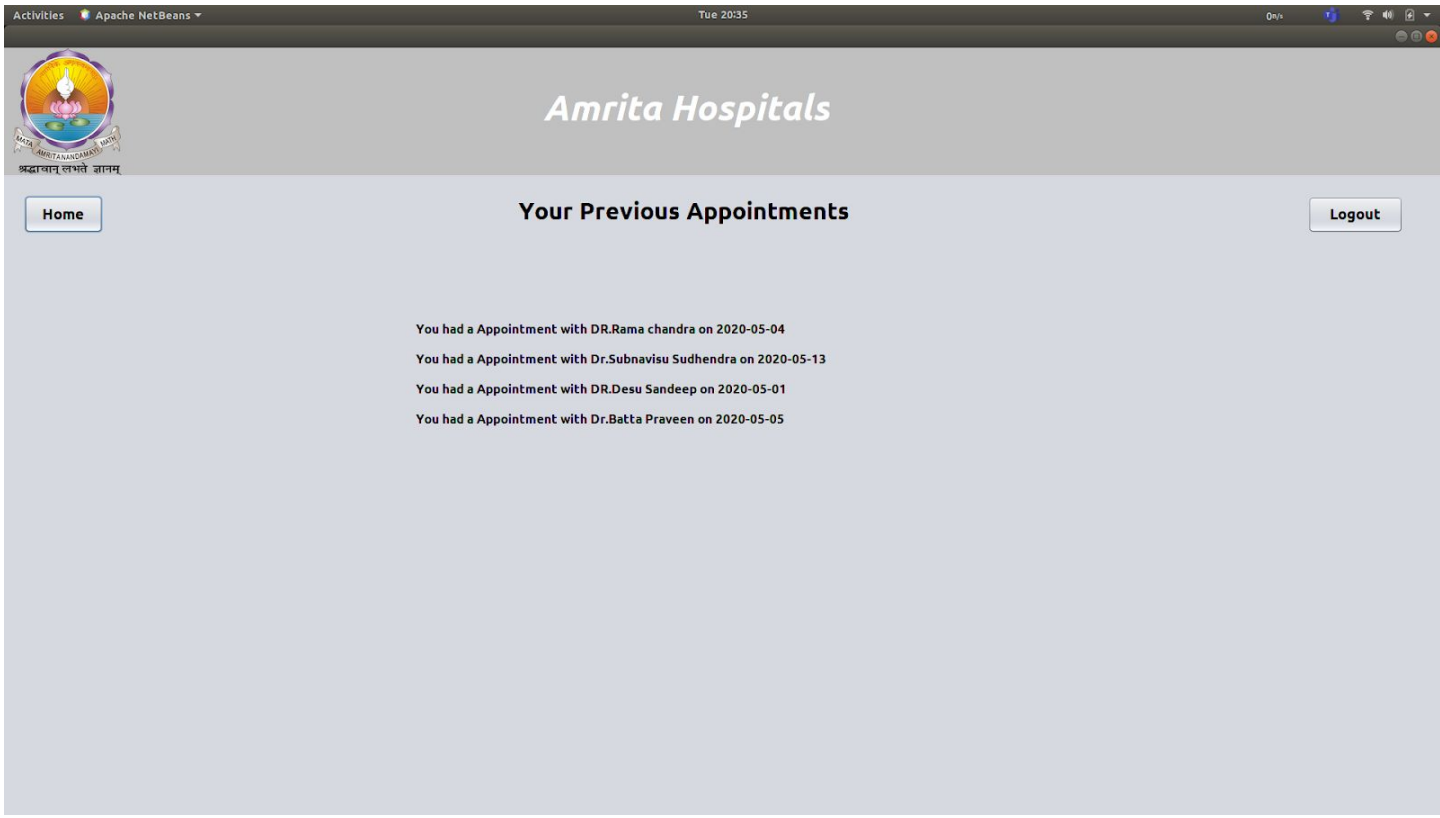


Link of Code: View Next Appointment Page

SQL Query of View Next Appointments:

select a.appoint_id,d.name,a.dateofapp,a.time from appointmentbooking as a,doctor as d where a.doc_id=d.doc_id and (a.dateofapp >=(select curdate()) and id=Login.id1);

View Previous Appointments :



Link of the Code: View Previous Appointments page

SQL Query of Previous Appointment:

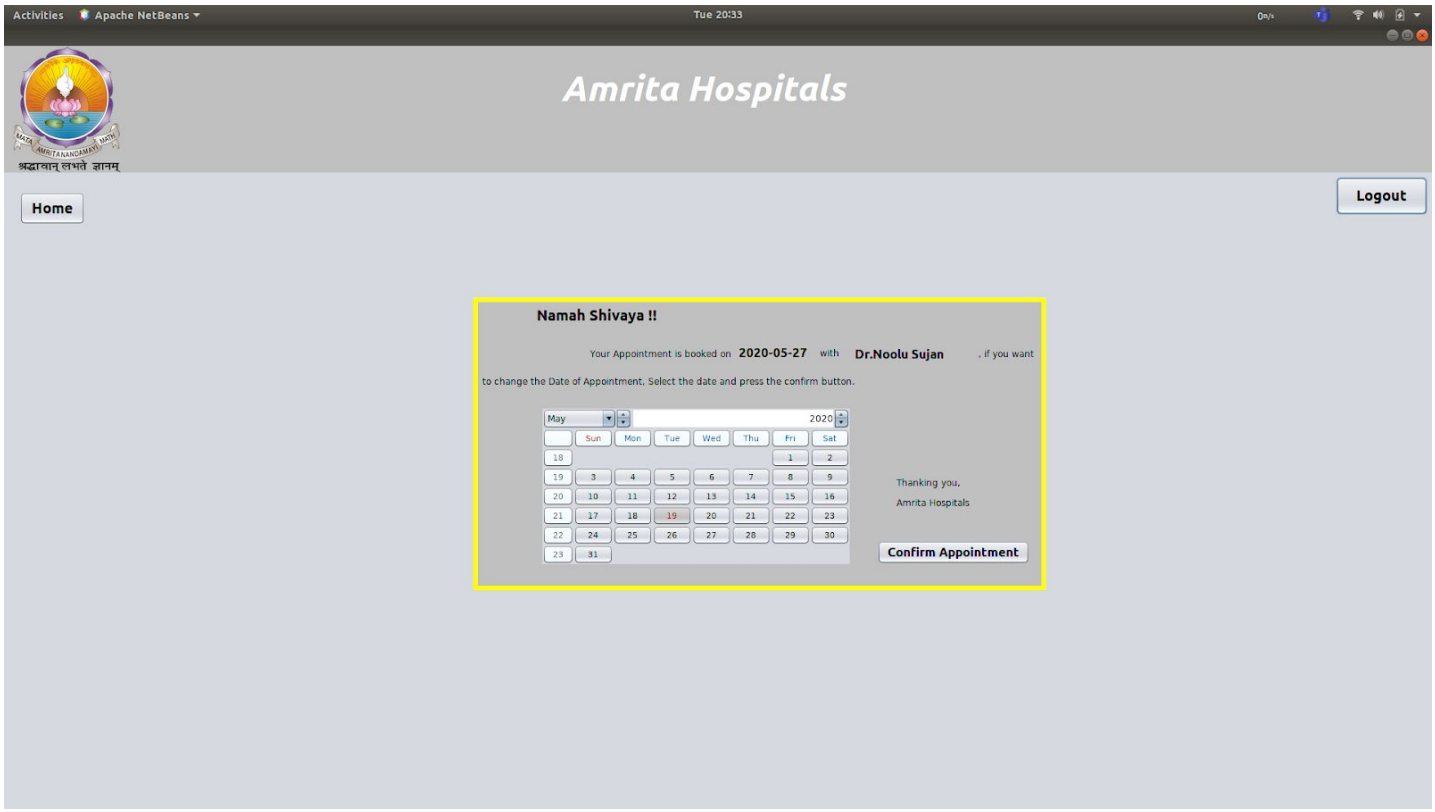
select doctor.name,appointmentbooking.dateofapp from appointmentbooking,doctor where doctor.doc_id=appointmentbooking.doc_id and (appointmentbooking.id=Login.id1 and appointmentbooking.dateofapp<curdate());

Cancel Appointment :

Link of Code: [Cancel Appointment Page]

SQL Query of Cancel Appointment :
delete from appointmentbooking where appoint_id=appid;

Edit Appointment:



Link of the Code: [Edit Appointment Page]

SQL Query of Edit Appointment:

update appointmentbooking set dateofapp=date2 where id=nop and dateofapp > (select curdate());

Database Tables in the Application :

- User Table
- Doctor Table
- Appointment Booking Table

Table Description:

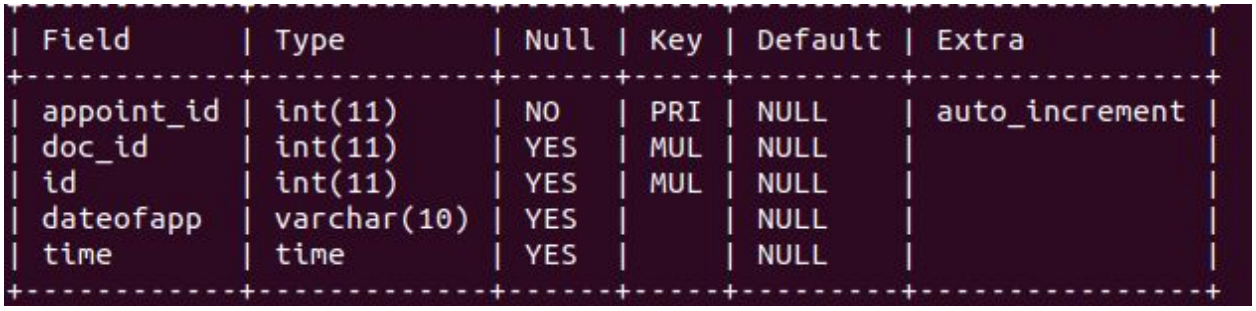
User Table:

```
mysql> describe user;
+--------------+---------------+------+-----+---------+----------------+
| Field        | Type          | Null | Key | Default | Extra          |
+--------------+---------------+------+-----+---------+----------------+
| id           | int(11)       | NO   | PRI | NULL    | auto_increment |
| username     | varchar(50)   | YES  |     | NULL    |                |
| password     | varchar(50)   | YES  |     | NULL    |                |
| gender       | varchar(20)   | YES  |     | NULL    |                |
| blood_group  | varchar(3)    | YES  |     | NULL    |                |
| Contact      | decimal(12,0) | YES  |     | NULL    |                |
| city         | varchar(15)   | YES  |     | NULL    |                |
+--------------+---------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```

Doctor Table:

```
mysql> describe doctor;
+----------------+---------------+------+-----+---------+----------------+
| Field          | Type          | Null | Key | Default | Extra          |
+----------------+---------------+------+-----+---------+----------------+
| doc_id         | int(100)      | NO   | PRI | NULL    | auto_increment |
| name           | varchar(50)   | YES  |     | NULL    |                |
| gender         | varchar(6)    | YES  |     | NULL    |                |
| specailization | varchar(20)   | YES  |     | NULL    |                |
| amount         | decimal(50,0) | YES  |     | NULL    |                |
| timmings       | time          | YES  |     | NULL    |                |
+----------------+---------------+------+-----+---------+----------------+
```

Appointment Booking Description:



Field	Type	Null	Key	Default	Extra
appoint_id	int(11)	NO	PRI	NULL	auto_increment
doc_id	int(11)	YES	MUL	NULL	
id	int(11)	YES	MUL	NULL	
dateofapp	varchar(10)	YES		NULL	
time	time	YES		NULL	

TESTING PLAN

TESTING STRATEGIES

The testing phase consists of

- Unit Testing
- Integration Testing
- System Testing

Unit Testing: In Unit testing, we tested every class in the application, all the errors have been solved.

Integration Testing: In integration Testing, we checked with combinations of two or more classes, no errors have occurred.
Checked with both

1. Top-down testing
2. Bottom-up testing

System Testing: After completion of unit testing and integration testing we went through system testing, we ran the application in Netbeans IDE, all the errors were solved, checked with a performance by opening multiple applications at a time the performance was good where it can manage more load at a time without any issues.

UNIT TESTING :

- We faced many challenges while connecting database and Application, we got many errors while unit testing the class with database connection like :
    - An error occurred while connecting to database mysql workbench:java.sql.SQLNonTransientConnectionException: Could not create a connection to the database server.
    - An Error occurred with connector issue, later our database was not able to retrieve data from the database properly.
- After Solving database issues we started building the middle-end and Front end of our Product.
- Then we started with the security of the application because any anonymous member should not able to use the website so we kept
    - Sign up
    - Login

For every user, so that hospital admin can understand if a patient is a genuine person or not.

- Sign up class: We have some challenges while getting details from the patient like
    - The name of the user should be only characters. If the user has given any other characters we will display this pop box, so that he may correct all characters.

    - For user confirmation of password, we have kept re-enter password so that if he types any letter by mistake he can

acknowledge it and change, here we are popping a box if he/her enters re-enter password equally.

- Later, we kept combo boxes for Blood group and gender selection, so that users can't type any wrong.
- For the Phone number, we checked whether the user has given 10 digits correctly. If at all, the user given digit less or more we will give a pop-up box of the wrong format.
- On the same, we kept while entering the city name whether all are characters or not.

- By the following ways, our unit testing was done in the Signup class.
- Login Class:
  - Here we are checking with every user in the database, whether the entered username and password matches if it matches we will redirect to Home page else, we are giving a pop-up box so, that he can re-enter details properly.

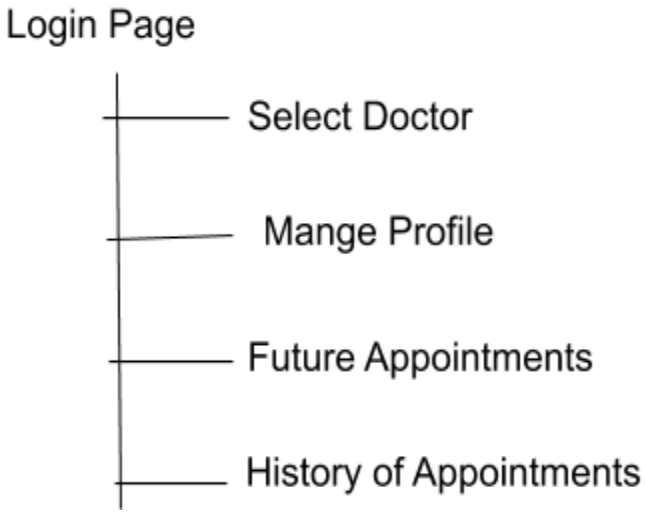
  - Here unit testing is done for login class.

- Home Class:
    - Here we are checking whether all the buttons are working when the user selects a particular button. It should redirect to the assigned page so that no error occurs.
    - Here Unit Testing is done, all the buttons are redirecting properly so that the user can see the information that he wants.
- Specialization Class:
    - Here the user should select the specialization of the doctor, so he can directly redirect to select doctor class by clicking on the button.
    - Here the Unit test is done while a selected Specialization of doctor is stored.
- Select Doctor Class:
    - Here the user will select the date of appointment and available doctor of specialization.
    - Here Unit testing is done whether the selected date is retrieving and the selected doctor is storing in the database.
- Payment Class:
    - Here we are taking credentials of the user for payment account number, expire date, expire month, CVV of the card.
    - Here we checked whether the user is given exact details like 16 digits of the account number, card month is less than 13 and year of 2 digits.
    - Here unit testing is done on all the text values.
- Fix Appointment Class:
    - Here the conformation of the Appointment letter is displayed.
    - Here unit testing is done whether the user name, doctor name and date of appointment is done.
- Edit Appointment Class :
    - Here all the Future appointments are displayed and the user can edit the appointment by selecting the date of appointment again.

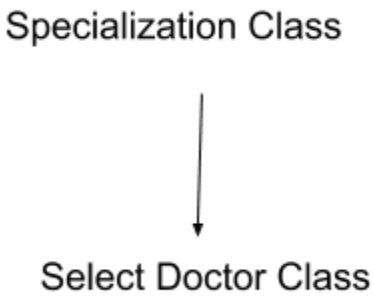
    - Here the unit test is done whether exact details of the patient are displayed over there and the patient is able to change the date of appointment.
- Cancel Appointment Class:
    - Here all the future Appointments will be displayed so that the user can cancel any of it.
    - Here unit testing is done whether a cancelled appointment is updated in the database and displayed all the future appointments.
- Profile Management Class:
    - Here all the details of the patient are displayed.
    - Unit testing is done whether the user details extracted are correct or not.
- Change Password Class:
    - Here unit testing is done whether the changed password is updated in the database or not.

INTEGRATION TESTING:

- Here whenever a user login the session of the user will be started i.e., the id of the user will be shared in the following pages.

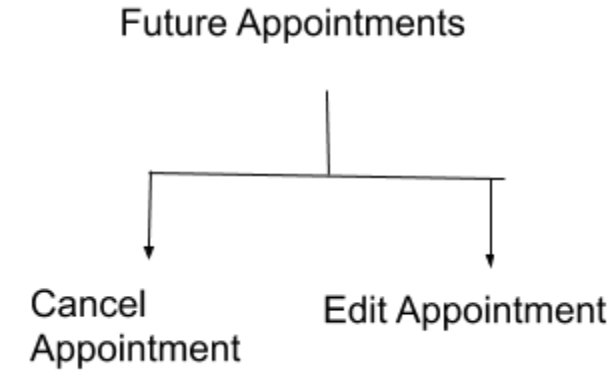


- We are taking selected specializations and sending the value to select doctor class for displaying all the doctor's specialized.



➔ Here we are checking variables that are passing from one class to another.

- While Editing appointment or cancelling appointment the user should select from the future appointments, so, the appointment id of the appointment should be passed to the Edit appointment class/ cancel appointment class



.

SYSTEM TESTING:

- All the class is run at a time and tested.

3. Determining the Test deliverables

Test deliverables are provided **before** the testing phase.

- Test plans document.
- Test cases documents
- Test Design specifications.

Test deliverables are provided **during** the testing

- Test Scripts.
- Test Data
- Error logs and execution logs.

Test deliverables are provided **after** the testing cycle is over.

- Test Results/reports
- Defect Report
- Installation/ Test procedures

Future Work:

- Developing GUI of the application.
- Gmail authentication for Register/Login (if needed).
- Phone number OTP confirmation.
- Able to Book appointments with one doctor within a period of time.
- Send Message to the mobile of the appointment and be able to set a reminder.
- Collecting Feedback from the Patients.

OUR TAKEAWAYS

As a Group: we learned how to work as a group, understanding each other while working, learning through mistakes and making time for working on this project daily during college days and during lockdown period, managed to work online in the Lockdown period, gathered in the Microsoft teams worked on this product, learned how to integrate the code, while working and developing the product we enjoyed a lot.

Technically we learned a lot, like how to use Github efficiently, how to do a project in different ways according to project needs, how to use plugins and APIs in the product, revised the database concepts, improved Java skills, learned how to design in the customer perspective.

Work Done individually:

S.S.R Sudhendra :

- Scrum master of the project.
- Maintaining team balance and work split among team members.
- Contributed to writing project plans, SRS documents, flow diagrams related to the project.
- Worked on Net beans IDE installation errors.
- Worked till phase1 of the project along with the team.
- Reporting product owner and updating him on the workflow.
- Meeting the deadlines as a team.
- Meeting via Microsoft team’s application and discussing the progress of the project.

I.V.V Raghavendra :

- Worked with teammates making weekly reports, Sequence diagram, Activity Diagram.
- Worked mostly on the middle-end and Backend of the project.
- Added calendar plugin into the application.
- Worked connection of database with the team.
- Coded Edit appointment, Cancel Appointment, View Appointment, History of Appointment along with the team.
- And also wrote some queries for extracting the data from the database to the Application,
- Been an Active Member for exchanging the ideas for project in the Team.
- Helped Team while writing documentation.
- Tested Project in multiple corner cases and Performed unit testing and integration testing.
- Worked on writing Testing Plan.

D.V.N Sandeep :

- The major contribution is the Middle end and UI part.
- worked on the Specialization page, Home page along with the team.
- worked connecting Database to the Middle End.
- worked on writing use cases of our project and designing SRS, user Stories
- documentation.
- Worked along with a team writing Weekly reports, Drawing diagrams.
- was active in the meetings and shared ideas in the implementation of the project.

N Sujan sai :

- A major contribution to the UI part.
- Worked on the Login page and Signup page along with the team.
- Worked connecting Database to the UI part.
- Creating SQL tables is the part that gave me the most exposure to Database Management.
- Worked on writing use cases of our project and designing SRS documentation.
- Worked on Integration Testing Documentation.
- Worked on writing Report and discussed about corner cases of the application.

B Siva Praveen :

- contributed towards the UI part in developing the application.
- Dealt with SQL queries while updating the database.
- Helped draw flow charts, writing weekly reports.
- Helped the plugin required for the Netbeans IDE.
- Worked on UI designing and tried different styles of designing.
- Worked on writing SRS and user stories.
- Performed System Testing and discussed some cases to test our project.
- Worked on writing Testing plans.

THANK YOU