DOCTOR APPOINTMENT BOOKING SYSTEM

E-DAC, SEPT 2020
KP- Bangalore

Names of the Project Team Members:

Sr. No.	Name	PRN No.
1.	Akash Kishore Borse	200950181025
2.	Pradnya Ravi Nakhale	200950181057
3.	Parag Laxman Patil	200950181064
4.	Vaibhav Sunil Wankhede	200950181115
5	Reddi Manoja Bommisetti	200950181024

SYSTEM DESIGN:

In this software we have developed some forms. The brief description about them is as follow:-

1. Proposed System

The Doctor appointment Booking System (DABS) is designed for Any Hospital to replace their existing manual, paper based system and make it on the online system so it can be easy to all. The new system is to control the following information; Patient information, Doctor details, Admin Details, staff and schedules. These services are to be provided in an efficient, cost effective manner, with the goal of reducing the time and resources currently required for such tasks.

3. Doctor Appointment Booking System Does:

- Make Doctor Appointment booking Online for Patients according to their feasibility.
- This system helps reduce the problems occurring when using the manual system and helps patients to skip endless queues.
- This System is a web application based on Spring Framework - Hibernate with Maven as a Project Management tool.
- System having Three types of Role Users Admin , Doctor and Receptionist.
- Admin can do all the transaction services like add, update, delete, view of Patient, Appointment, Doctor .
- Doctors can view the Patient details and Appointment Details.
- Receptionist can do all the transaction services like add, update, delete , view of Patients.
- These services are to be provided in an efficient, cost effective manner, with the goal of reducing the time

DOCTOR APPOINTMENT BOOKING SYSTEM - DESCRIPTION

Presentation Layer

In the Presentation Layer we write all the User Interface part of the Doctor Appointment Booking System.

- Home Page is a Landing page of the Project.
- All UI part of Sign in as Admin. It consists of Add, Update, Delete, View of Patient, Appointment, Doctor and Receptionist.
- All UI part of Sign in as Doctor. It consists of View of Patient details and Appointment details.
- All UI part of Sign in as Receptionist. It consists of Add, Update, Delete, View of Patient.

Service Layer

In the Service Layer we write the service business logic.

1.Sign In as Admin

- Admin can Add the Patient,Edit Patient, Update Patient, Delete Patient and View the Patient Details List.
- Admin can Add Appointment, Edit Appointment, Update Appointment, Delete Appointment and View Appointment Details List.
- Admin can Add Doctor, Edit Doctor, Update Doctor, Delete Doctor and View the Doctor Details List.
- Admin can Add Receptionist, Edit Receptionist, Update Receptionist, Delete Receptionist and View Receptionist Details List.

2.Sign In as Doctor

- Doctors can View Patient Details List and Appointment Details List.

3. Sign In as Receptionist

- Receptionist can Add Patient,Edit Patient,Update Patient,Delete Patient and View Patient Details List.

DAO Layer:

- In the DAO layer we are connecting to the database(MySQL).
- We created a database for this project.
- In the Database we created Patient, Appointment, User, Role table
- We can Add Patient, Edit Patient, Update Patient, Delete Patient and View the Patient Details List for Patient table.
- We can Add Appointment, Edit Appointment, Update Appointment, Delete Appointment and View Appointment Details List for Appointment table.
- We can Add Doctor, Edit Doctor, Update Doctor, Delete Doctor and View the Doctor Details List for Role table.
- We can Add Receptionist, Edit Receptionist, Update Receptionist, Delete Receptionist and View Receptionist Details List for Role table .
- User table has all the Details about Admin, Doctor and Receptionist.

Software Requirements:

Operating System **:**	Windows
Programming language**:**	Java
Web-Technology**:**	Spring MVC, html, css
Front-End**:**	html,Tiles
Back-End**:**	Java,Spring MVC,Hibernate , jdbc
Web Server**:**	Apache tomcat

PROBLEM FACED DURING THE PROJECT

1. We all five Project Group Member worked on the different differnet part of the project. Presentation , Service, DAO. In the presentation layer i.e. in the front end part . We got stuck because of How to connect it with the back end part. And further we solve this error by discovering, following some services related to how to connect the front end application to the backend part service.

2. While Merging the Project into one from different files developed by each team member we got into some errors caused due to different versions of dependencies and Different versions of Softwares installed on each Machine.
   - First of all we all tried to recognize what is the actual cause of the problem .We found out that every file is working fine but the problem is with different types of dependency used and different versions of software installed .
   - Then we had a meeting with team members and tried to find out what are the dependencies that are creating problems due to different versions and find out on which least version our project can work fine.
   - Then we updated the dependency into the best working dependency for each machine, which actually solved the problem.

WE LEARNED FROM THE PROJECT

1. **Complete Flow for making the project :** From this project we learned the complete Life cycle of making a Software Project. We learned collecting Requirements ,Analyzing the problem which is intended to solve, Planning for a particular project ,Designing the project and database structure, and also learned about Coding and Testing Phase.

2. **Spring Framework and Hibernate:** We have made use of Spring MVC Framework in this project so we have learned the complete workflow of Spring project and how the different layers work .

3. **Connect with the MySQL Database:** We have learned how to connect with MySql Database for the Spring Web Application Project.Create a Hibernate query for connecting the database with a working project.

4. **Postman Tool:** We have learned the Postman tool for testing the web services. And make it successfully test for every translation service.

5. **Agile Software development:** We have used an agile approach for making this software project so we understood how to work in small Sprints to achieve a final product ,We have made use of Jira Scrum board created sprints added issues,backlog and worked according to them.

6. **TeamWork:** This Project taught us the importance of Teamwork. We have completed all tasks as a team,We have learned that working alone on a project and Working as a team is totally different .Teamwork reduces the pressure and also increases the efficiency .

7. **Managing Version Control System(GIT) :** We have learned to manage the version control system for keeping the different versions of softwares .GIT has increased the efficiency and also made it simple for us to manage the software project well .