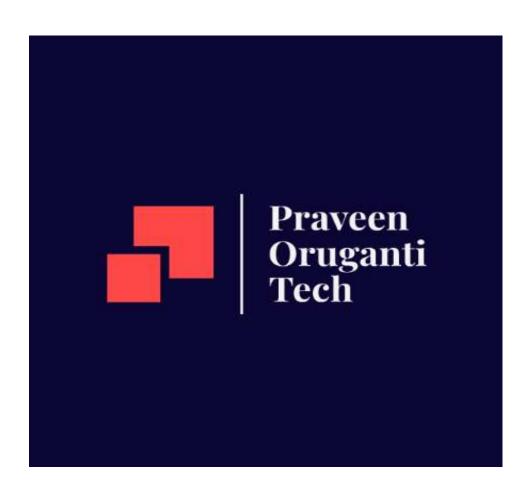
Book My Doctor App



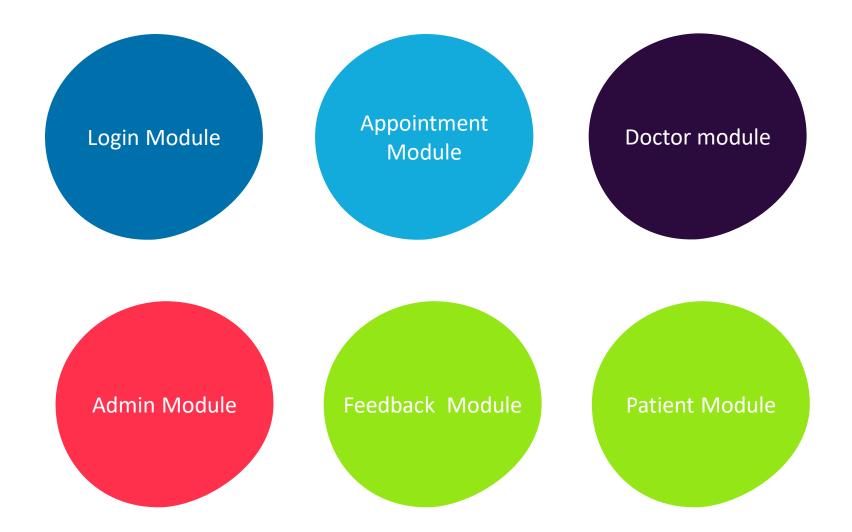
Problem Statement

- To Create an Application for online appointment with Doctor
- Customer can Login/Signup to application.
- Customer should be able to take appointment with Doctor
- Customer should be able view appointment status
- Customer can give the feedback about Doctor.
- Admin should be able to approve or reject the appointment.
- Admin can add/modify/view/remove patient
- Admin can add/modify/view/remove Doctor
- Admin can view list patient
- Doctor can login and view the list of appointment, list of patients,
- Doctor can cancel appointment

Plan

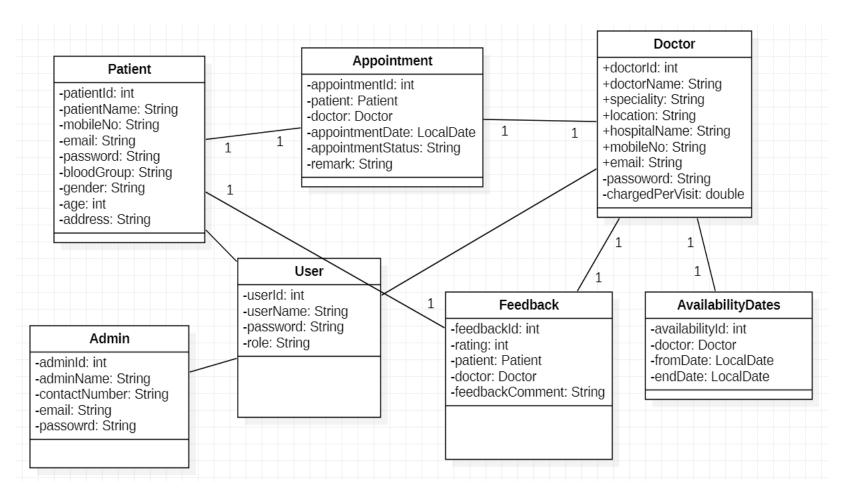
- 1. Design
 - 1. Generate the skeleton as per class design provided
- 2. TDD approach
 - 1. Identify test cases for each module
 - 2. Write Failing Test cases
 - 3. Convert the Test Cases in GREEN
- 3. Implement the case study in incremental manner

Modules



Class Design

POJO Classes:



Class Design

Service Interface

IUserService

+validateUser(User user): User +addUser(User user): User +removeUser(User user): User +updateUser(User user): User

IAppointmentService

+getAllAppointments(): List<Appointment>
+getAppointment(int appointmentId): Appointment
+addAppointment(Appointment app): Appointment
+deleteAppointment(int appointmentId): Appointment
+updateAppointment(Appointment app): Appointment
+getAppointments(Doctor doc): List<Appointment>
+getAppointments(LocalDate date): List<Appointment>

IDoctorService

+addDoctor(Doctor bean): Doctor

+updateDoctorProfile(Doctor bean): Doctor

+addAvailability(AvailabilityDates bean): AvailabilityDates

+updateAvailability(AvailabilityDates bean): AvailabilityDates

+getDoctor(Doctor doc): Doctor

+removeDoctor(Doctor doc): Doctor

+getDoctorList(): List<Doctor>

+getDoctorList(String speciality): List<Doctor>

IAdminService

+addAdmin(Admin admin): Admin

+updateAdmin(Admin admin): Admin

+removeAdmin(Admin admin): Admin

+viewAdmin(Admin admin): Admin

IFeedbackService

+addFeedback(Feedback fbd): Feedback

+getFeedback(Feedback fbd): Feedback

+getAllFeedbacks(Doctor doc): List<Feedback>

IPatientService

+addPatient(Patient bean): Patient

+updatePatientDetails(Patient bean): Patient

+removePatientDetails(Patient bean): Patient

+getPatient(Patient bean): Patient

+getAllPatient(): List<Patient>

+getPatientListByDoctor(Doctor doctor): List<Patient>

+getPatientListByDate(LocalDate appdate): List<Patient>

Praveen Oruganti Tech

Class Design

Repository:

IUserRepository

+validateUser(User user): User +addUser(User user): User +removeUser(User user): User +updateUser(User user): User

IAppointmentRepository

+getAllAppointments(): List<Appointment>
+getAppointment(int appointmentId): Appointment
+addAppointment(Appointment app): Appointment
+deleteAppointment(int appointmentId): Appointment
+updateAppointment(Appointment app): Appointment
+getAppointments(Doctor doc): List<Appointment>
+getAppointments(LocalDate date): List<Appointment>

IDoctorRepository

+addDoctor(Doctor bean): Doctor

+updateDoctorProfile(Doctor bean): Doctor

+addAvailability(AvailabilityDates bean): AvailabilityDates

+updateAvailability(AvailabilityDates bean): AvailabilityDates

+getDoctor(Doctor doc): Doctor

+removeDoctor(Doctor doc): Doctor

+getDoctorList(): List<Doctor>

+getDoctorList(String speciality): List<Doctor>

IAdminRepository

+addAdmin(Admin admin): Admin +updateAdmin(Admin admin): Admin +removeAdmin(Admin admin): Admin +viewAdmin(Admin admin): Admin

IFeedbackRepository

+addFeedback(Feedback fbd): Feedback +getFeedback(Feedback fbd): Feedback +getAllFeedbacks(Doctor doc): List<Feedback>

IPatientRepository

+addPatient(Patient bean): Patient

+updatePatientDetails(Patient bean): Patient

+removePatientDetails(Patient bean): Patient

+getPatient(Patient bean): Patient

+getAllPatient(): List<Patient>

+getPatientListByDoctor(Doctor doctor): List<Patient>

+getPatientListByDate(LocalDate appdate): List<Patient>

Praveen Oruganti Tech