Faculty of Computing & Information Technology

SOFTWARE DESIGN AND ARCHITECTURE

ASSIGNMENT-02

SUBMITTED BY

GROUP MEMBERS

BSEF21M003-KHADIJA TARIQ

BSEF21M022-RUBAISHA ZAIDI

BSEF21M027-AREEBA QAMAR

SUBMITTED TO: MAM FATIMA SABIR

Project Topic: Hospital Management System

Description: The Hospital Management System (HMS) is designed to assist hospitals and Patients. It helps doctors, nurses, and other staff keep everything organized and running smoothly. With the HMS, they can easily keep track of patient information, schedule appointments, talk to patients, manage supplies, handle billing and payments, and make reports to help make better decisions. It's designed to make hospitals work better, so patients get better care and everyone stays happy.

Scope: The Hospital Management System (HMS) aims to make running a hospital easier by providing a computer program that helps manage everything from appointments to billing. It's designed for hospitals and clinics to keep track of patients, schedule appointments, communicate with staff, manage supplies, handle billing and payments, respond to emergencies, and keep everything organized. The goal is to make hospitals run more smoothly and improve patient care, no matter where they are or what their background is.

Features :

1. User Registration
2. User login
3. User profile management
4. Search for doctor
5. Book an appointment
6. Dashboard for doctors, patients, medical staff
7. Dashboard for admin
8. Communication between doctors and patients
9. Call or request for an ambulance
10. Order Medicine and inventory
11. Payment and billing
12. Feedback and rating
13. Notification System
14. FAQs
15. Logout

Requirement details with actors (External Entity)

Requirement #	External Entity	Initial Requirement
1	System	The System “shall” only allow the registered user to access the website platform.
1.1	Patient	Patient “shall” Register to access the platform, by providing a username, valid email, password, address, age, gender

1.2	Doctor	Doctor “shall” Register to access the platform, by providing a username, email, password, and additional details such as information about his degree, post, experience, and educational background.
1.3	System	The system “shall” validate the provided information to ensure accuracy and completeness.
1.4	System	The system “shall” generate a unique user ID for each registered patient
2	System	The System “shall” allow the user to render the home page of the website after successful login
2.1	Patient/Doctor	Registered user “shall” be able to Log in with their username/email and password
2.2	Admin	Admin “shall” log in to his account.
3	System	The system “shall” process different types of updating e.g. updating of his personal details, profile details, or upgrading of his status from registration, or updating of his payment details.
3.1	System	The system “shall” accept the Patient's Request to change the Password and profile details.
3.1.1	Patient	A Patient “shall” login to the system and can change his/her password for security reasons
3.1.2	Patient	The patient “shall” create, update, and manage profiles, including their field of study, skills, goals, and business plan information
3.2	System	The system “shall” accept the Patient’s Request to change the Password and profile details.
3.2.1	Doctor	A Doctor “shall” login to the system and can change his/her password for security reasons.
3.2.2	Doctor	Doctor ”shall” create, update, and manage profiles, and information about his degree, post, experience, and educational background.
4	Patient	The Patient “shall” a search doctors based on various criteria like names, categories, duty hours, specialization, and experience.
4.1	System	The system “shall” display relevant doctor profiles with details such as expertise, qualifications, and reviews
5	Patient	The patients “shall” book appointments with selected doctors.
5.1	Doctor	The Doctor “shall” ” see the appointments taken by patients at the platform.

5.2	System	The system “shall” confirm appointment bookings and send notifications to patients and doctors.
6	Doctor, Patient, Medical Staff	Doctor, Patient, Medical Staff “shall” present personalized dashboards for doctors, patients, and medical staff upon login.
6.1	System	The system “shall” display relevant information such as appointments, notifications, and tasks.
6.2	System	The system “shall” allow customization of dashboard layouts and preferences.
7	Admin	Admin “shall” manage system settings and resources.
7.1	System	The system “shall” provide an administrative dashboard for administrators to manage system settings and resources.
7.2	System	The system ”shall” ensure that admin dashboards are accessible only to authorized personnel.
8	Patient	Patient “shall” take advice from doctors.
8.1	System	The system “shall” support real-time chat, video calls, and file-sharing functionalities.
9	Patient	Patient “shall” request ambulance services in case of emergencies.
9.1	System	The system “shall” notify relevant authorities and dispatch ambulances promptly.
10	Admin	Admin “shall” manage inventory levels and order medicines as needed
10.1	System	The system “shall” track inventory transactions and update stock levels in real-time.
11	Patient	Patient “shall” pay bills through the platform
11.1	System	The system “shall” generate invoices and process payments securely.
11.2	System	The system “shall” provide options for online payments and insurance claims processing.
12	Patient	Patient “shall” give feedback about the website and services provided by the hospital.
12.1	Doctor	Doctor “shall” give feedback about website performance
12.2	System	The system “shall” analyze feedback data to identify areas for improvement.
13	System	The system “shall” send notifications to users regarding appointments, medication reminders, etc.
14	Patient	Patient “shall” read frequently asked questions from the website to know about the hospital and its services.
14.1	System	The system “shall” allow users to log out of their accounts securely.
15	System	System “shall” provide Logout functionality.

15.1	Patient/Doctor/Admin	Users who are logged in to the web application “shall” be able to Log out from the website when they’re done.
15.2	System	Upon logging out, the system “shall” terminate the user’s session and return them to the login or home page as appropriate.

Use Case Diagram

A use case diagram is a vital blueprint for visualizing interactions among system entities.

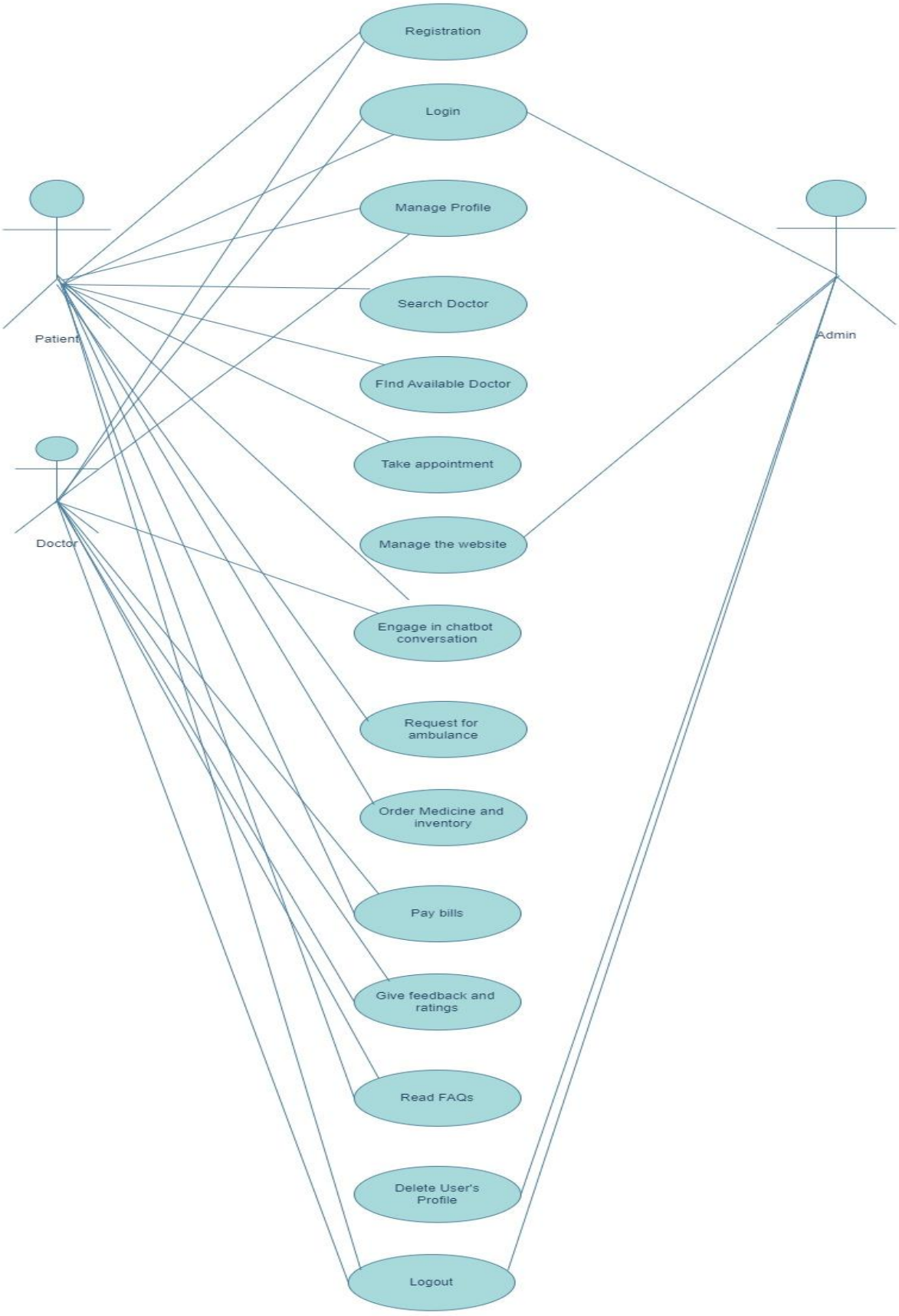
In the case of HMS, this platform connects patients across the country with doctors.They can get medical assistance at any time through the platform. Below is the use case diagram of the Hospital Management System.

- **Primary Actors:**

1. **Patient**
2. **Doctor**

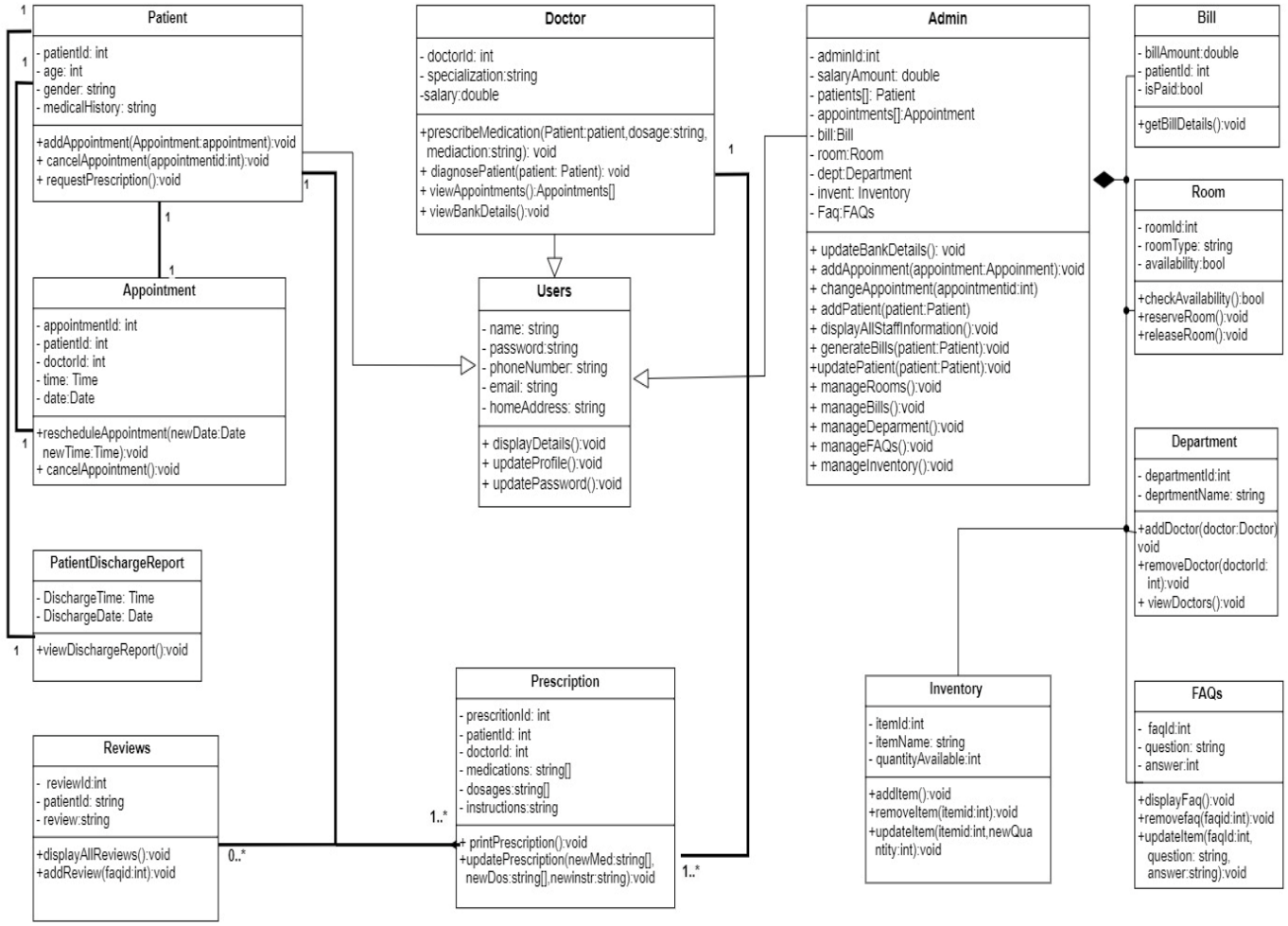
- **Secondary Actor**

1. Admin.

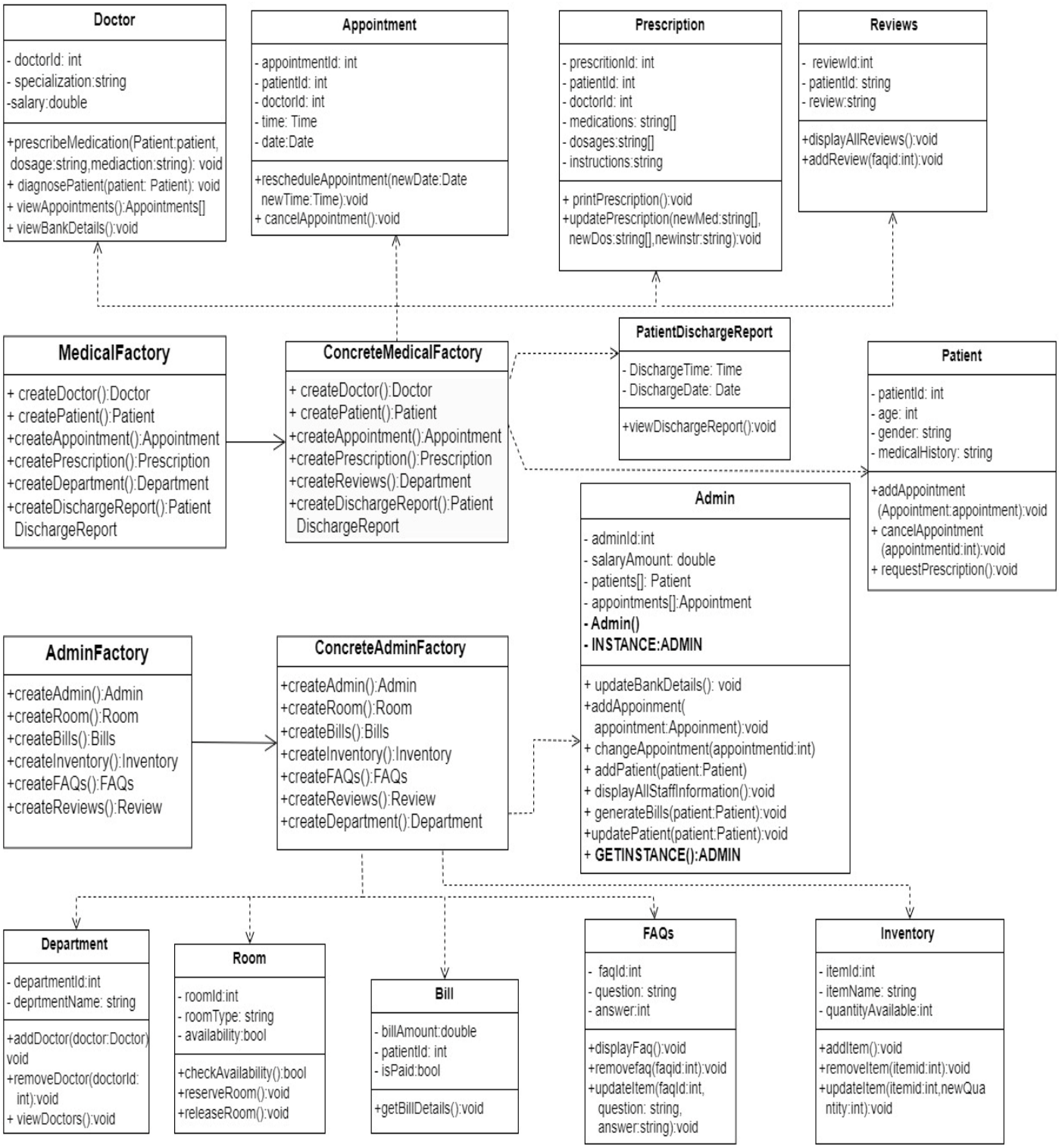

Registration
Login
Manage Profile
Search Doctor
FInd Available Doctor
Take appointment
Manage the website
Engage in chatbot conversation
Request for ambulance
Order Medicine and inventory
Pay bills
Give feedback and ratings
Read FAQs
Delete User's Profile
Logout
Patient
Doctor
Admin

Part # 2

Class Diagram



Refined Class Diagram


Doctor
- doctorId: int
- specialization:string
-salary:double
+prescribeMedication(Patient:patient, dosage:string,mediaction:string): void
+ diagnosePatient(patient: Patient): void
+ viewAppointments():Appointments[]
+ viewBankDetails():void
Appointment
- appointmentId: int
- patientId: int
- doctorId: int
- time: Time
- date:Date
+rescheduleAppointment(newDate:Date newTime:Time):void
+ cancelAppointment():void
Prescription
- prescritionId: int
- patientId: int
- doctorId: int
- medications: string[]
- dosages:string[]
- instructions:string
+ printPrescription():void
+updatePrescription(newMed:string[], newDos:string[],newinstr:string):void
Reviews
- reviewId:int
- patientId: string
- review:string
+displayAllReviews():void
+addReview(faqid:int):void
MedicalFactory
+ createDoctor():Doctor
+ createPatient():Patient
+createAppointment():Appointment
+createPrescription():Prescription
+createDepartment():Department
+createDischargeReport():Patient DischargeReport
ConcreteMedicalFactory
+ createDoctor():Doctor
+ createPatient():Patient
+createAppointment():Appointment
+createPrescription():Prescription
+createReviews():Department
+createDischargeReport():Patient DischargeReport
PatientDischargeReport
- DischargeTime: Time
- DischargeDate: Date
+viewDischargeReport():void
Patient
- patientId: int
- age: int
- gender: string
- medicalHistory: string
+addAppointment (Appointment:appointment):void
+ cancelAppointment (appointmentid:int):void
+ requestPrescription():void
Admin
- adminId:int
- salaryAmount: double
- patients[]: Patient
- appointments[]:Appointment
- Admin()
- INSTANCE:ADMIN
+ updateBankDetails(): void
+addAppoinment(appointment:Appoinment):void
+ changeAppointment(appointmentid:int)
+ addPatient(patient:Patient)
+ displayAllStaffInformation():void
+ generateBills(patient:Patient):void
+updatePatient(patient:Patient):void
+ GETINSTANCE():ADMIN
AdminFactory
+createAdmin():Admin
+createRoom():Room
+createBills():Bills
+createInventory():Inventory
+createFAQs():FAQs
+createReviews():Review
ConcreteAdminFactory
+createAdmin():Admin
+createRoom():Room
+createBills():Bills
+createInventory():Inventory
+createFAQs():FAQs
+createReviews():Review
+createDepartment():Department
Department
- departmentId:int
- deprtmentName: string
+addDoctor(doctor:Doctor) void
+removeDoctor(doctorId: int):void
+ viewDoctors():void
Room
- roomId:int
- roomType: string
- availability:bool
+checkAvailability():bool
+reserveRoom():void
+releaseRoom():void
Bill
- billAmount:double
- patientId: int
- isPaid:bool
+getBillDetails():void
FAQs
- faqId:int
- question: string
- answer:int
+displayFaq():void
+removefaq(faqid:int):void
+updateItem(faqId:int, question: string, answer:string):void
Inventory
- itemId:int
- itemName: string
- quantityAvailable:int
+addItem():void
+removeItem(itemid:int):void
+updateItem(itemid:int,newQuantity:int):void