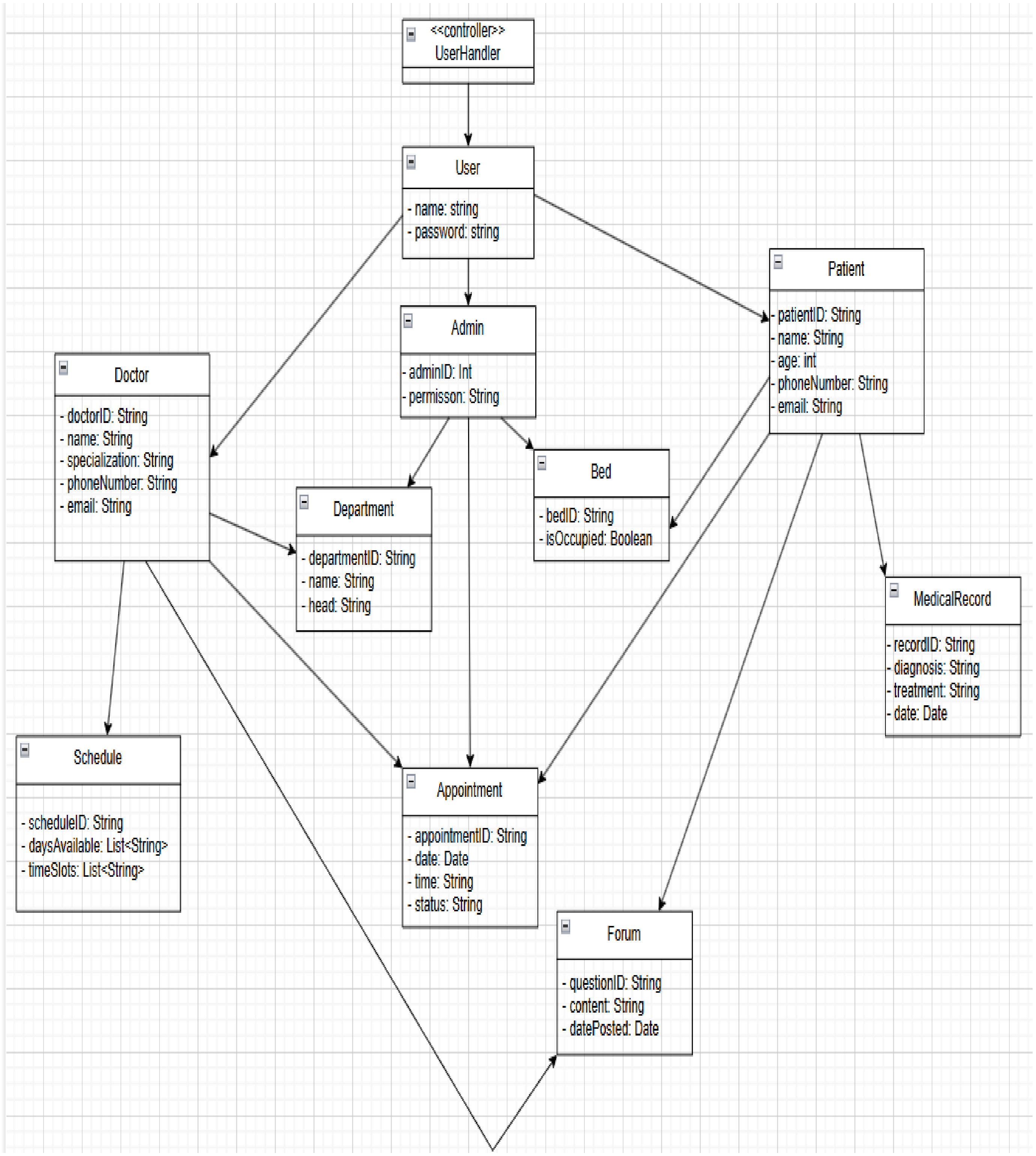
https://www.figma.com/proto/4GC53DGE5CVgkd4s5REO03/Hospital-Management-System?node-id=0-1&t=mA5RwcyQG2ykxvx1-1

a.



Admin

Attributes:

name: Represents the name of the hospital.

address: Represents the address of the hospital.

departments: Represents a list of departments in the hospital.

doctors: Represents a list of doctors working in the hospital. patients: Represents a list of patients in the hospital.

appointments: Represents a list of appointments scheduled in the hospital.

forum: Represents a forum for communication among staff or patients.

beds: Represents a list of beds in the hospital.

Functions:

Admin (name: string, address: string): Constructor to initialize the hospital with a name and address. Methods to get and manipulate hospital attributes such as name, address, departments, doctors, patients, appointments, forums, beds and administrative tasks, such as creating doctor emails, searching for patients and doctors, managing beds, and handling appointments.

Department

Attributes

name: Represents the name of the department.

doctors: Represents a list of doctors in the department.

Functions

Department (name: string): Constructor to initialize the department with a name. Methods to get and manipulate department attributes such as names and doctors. Methods to add and remove doctors from the department.

Doctor

<u>Attributes</u>

name: Represents the name of the doctor.

specialization: Represents the medical specialization of the doctor.

department: Represents the department to which the doctor belongs.

patients: Represents a list of patients assigned to the doctor.

email: Represents the email address of the doctor.

phoneNO: Represents the phone number of the doctor.

address: Represents the address of the doctor.

<u>Functions</u>

Doctor (name: string, specialization: string): Constructor to initialize the doctor with a name and specialization.

Methods to get and manipulate doctor attributes such as name, specialization, department, patients, and email, address, phone number.

Methods to confirm if patients need bed after surgery.

Methods to know which patients are assigned to them.

Method for doctors to search for the patient by the name and his id.

Patient

Attributes

name: Represents the name of the patient.

age: Represents the age of the patient.

gender: Represents the gender of the patient.

medicalRecords: Represents a list of medical records of the patient.

doctor: Represents the doctor assigned to the patient.

email: Represents the email address of the patient.

PhoneNO: Represents the phone number of the patient.

address: Represents the address of the patient.

Functions

Patient (name: string, age: int, gender: string): Constructor to initialize the patient with a name, age, and gender.

Methods to get and manipulate patient attributes such as name, age, gender, medical records, doctor, and email, address, phone number.

Methods to add and remove medical records.

Methods to set, get and cancel the assigned doctor.

Appointment

Attributes

patient: Represents the patient involved in the appointment.

doctor: Represents the doctor involved in the appointment.

appointmentTime: Represents the scheduled time for the appointment.

notified: Represents whether the patient has been notified about the appointment.

Functions

Appointment (patient: Patient, doctor: Doctor, appointmentTime: DateTime):Constructor to initialize the appointment with patient, doctor, and time.

Methods to get and manipulate appointment attributes such as patient, doctor, appointment time, and notification status.

Bed

Attributes

bedNumber: Represents the identifier for the bed.

isOccupied: Represents whether the bed is currently occupied.

patient: Represents the patient assigned to the bed.

Functions

Bed (bedNumber: string): Constructor to initialize the bed with a bed number.

Methods to get and manipulate bed attributes such as bed number, occupancy status, and assigned patient.

Image

Attributes

data: Represents the binary data of the image.

Methods to get and set image data.

Functions

Image (data: byte []): Constructor to initialize the image with binary data.

Forum

Attributes

Doctors: Represents a list of doctors who participate in the forum.

Question: Represents a list of questions posted on the forum.

Functions

Forum (): Constructor to initialize an empty forum.

getDoctors (): Doctor []: Retrieves the list of doctors participating in the forum.

addDoctor (doctor: Doctor): void: Adds a doctor to the forum.

removeDoctor (doctor: Doctor): void: Removes a doctor from the forum.

getQuestions (): Question []: Retrieves the list of questions posted on the forum.

addQuestion (question: Question): void: Adds a question to the forum.

removeQuestion (question: Question): void: Removes a question from the forum.

answerQuestion (question: Question, doctor: Doctor, answer: string): void: Allows a doctor to provide an answer to a specific question on the forum.

Questions

<u>Attributes</u>

patient: Represents the patient who posted the question.

doctor: Represents the doctor who answered the question.

questionText: Represents the text of the question.

answer: Represents the answer provided by the doctor.

<u>Functions</u>

Question (patient: Patient, doctor: Doctor, questionText: string): Constructor to initialize a question with a patient, doctor, and question text.

getPatient (): Patient: Retrieves the patient who posted the question.

setPatient (patient: Patient): void: Sets the patient for the question.

getDoctor(): Doctor: Retrieves the doctor who answered the question.

setDoctor(doctor: Doctor): void: Sets the doctor for the question.

getQuestionText (): string: Retrieves the text of the question.

setQuestionText(text: string): void: Sets the text of the question.

getAnswer(): string: Retrieves the answer provided by the doctor.

setAnswer (answer: string): void: Sets the answer for the question.

Methods represent a question posted by a patient on the forum. It contains information about the patient, the doctor who answered the question, the question text, and the answer. The class allows for retrieving and modifying these details.

b.

Use Case	Complexity
Enable Registration for new patients.	Simple
Enable Login for Patients and Doctors.	Simple
Enable patients, doctors, and admins to view medical records.	Simple
Enable patients and admins to update/delete medical records.	Moderate
Enable patients to make appointments with the chosen doctor.	Moderate
Doctor may select appointment details.	Simple
Generate appointment date and timing confirmation by admin.	Moderate
Modification in schedule by patient and admin.	Complex
Anyone accesses a patient's record by QR code.	Moderate
The QR code has all patient profiles.	Moderate
Every patient has a QR code bracelet.	Simple
Enable Doctors to search for their patients and vice versa, and the admin can search both doctor and patient.	Moderate
Patients have a forum to ask their questions and any doctor in the same field can respond.	Complex
Allocating beds to patients on a priority basis by confirming and tracking available beds.	Complex

c. CRC cards:

1. **Enable Registration for New Patients**

RegistrationWindow	
-Display registration form -submit registration details	PatientHandler

PatientHandler	
-Validate registration details -save patient information	PatientDB

PatientDB	
-Store patient information	

2. **Enable Login for Patients and Doctors**

LoginWindow	
-Display login form -submit login credentials	AuthHandler

AuthHandler	
-Validate credentials -authenticate user	UserDB

UserDB	
-Retrieve user credentials	

3. **Enable Patients, Doctors, Admins to View Medical Records**

MedicalRecordViewer	
-Display medical records	RecordHandler

RecordHandler	
-Retrieve medical records	MedicalRecordDB

MedicalRecordDB	
-Store and retrieve medical records	

4. **Doctor May Select Appointment Details**

AppointmentSelector	
-Display appointment options -select appointment	AppointmentHandler

AppointmentHandler	
-Confirm appointment details	AppointmentDB

AppointmentDB	
-Store appointment details	

5. Every Patient Has a QR Code Bracelet

QRCodeGenerator	
-Generate QR code	PatientHandler

PatientHandler	
-Assign QR code to patient	PatientDB

PatientDB	
-Store patient information	