**Analysis, Design, and/or Construction deliverables.**

**Working Code Demonstration:** <https://drive.google.com/file/d/1Y6VF9umrFSYD7L4wOcWUTOcgUiLqrZIB/view>

**Use Case Descriptions**

**1. Appointment Booking**

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Appointment Booking | |
| **Scenario:** | Patient books the appointment with one of the doctors online. | |
| **Trigger Event:** | A patient decided to book an appointment. | |
| **Brief Description:** | The use case is allowing patients to book appointments with doctors. After doctors schedule their availability, the patient is able to view it and pick a comfortable appointment time. The system then will notify both the patient and the doctor about an upcoming appointment. | |
| **Actors:** | Patient, Doctor, System | |
| **Related Use Cases:** | Appointment Viewing, Doctor Scheduling Availability | |
| **Stakeholders:** | Doctor, and Patient | |
| **Preconditions:** | Patient is logged into their account. Doctors have their availability set. | |
| **Postconditions:** | The appointment is booked. An email is sent to notify both the patient and the doctor about an upcoming appointment. | |
| **Flow of Activities:** | **Actor** | **System** |
| 1. Doctor sets their availability.  2. Patient selects “Book Appointment”.  3. Patient chooses the Doctor and available time slot.  4. Patient confirms the booking.  5. Both patient and doctor receive an appointment confirmation email. | 1.1 System saves doctor’s availability.  2.1 System displays available doctors and time slots.  3.1 System checks that the selected slot doesn’t conflict with anything.  4.1 Appointment is saved to the system.  4.2 Confirmation emails are sent.  5.1 The new appointment is added to their schedules. |
| **Exception Conditions:** | No available doctors/time slots, Invalid patient login, System couldn’t store the appointment. | |

**2. Appointment Viewing (in a calendar form)**

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Appointment Viewing (in a calendar form) | |
| **Scenario:** | Both doctors and patients access their future appointments. | |
| **Trigger Event:** | A user wants to check their upcoming appointments. | |
| **Brief Description:** | The use case allows both doctors and patients to view scheduled appointments through a calendar interface. The system retrieves and displays all upcoming appointments for the logged-in user. | |
| **Actors:** | Patient, Doctor, System | |
| **Related Use Cases:** | Appointment Booking, Doctor Scheduling Availability | |
| **Stakeholders:** | Doctor, Patient | |
| **Preconditions:** | The user (patient/doctor) is logged into their account. The user has scheduled appointment/s. | |
| **Postconditions:** | The user views their upcoming appointments. | |
| **Flow of Activities:** | **Actor** |  |
| 1. User selects “View Appointments.”  2. User navigates through the calendar.  3. User selects an appointment to view details. | 1.1 System retrieves the user’s upcoming appointments.  2.1 System updates the calendar view accordingly.  3.1 System displays appointment details. |
| **Exception Conditions:** | System couldn’t retrieve appointment data. | |

**3. Doctor Scheduling Availability**

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Doctor Scheduling Availability | |
| **Scenario:** | Doctors set their availability for appointments. | |
| **Trigger Event:** | A doctor decided to update their availability schedule. | |
| **Brief Description:** | The use case allows doctors to define their available hours and days for patient appointments. Once availability is set, patients can view and book appointments. | |
| **Actors:** | Doctor, System | |
| **Related Use Cases:** | Appointment Booking, Appointment Viewing | |
| **Stakeholders:** | Doctor, and Patient | |
| **Preconditions:** | Doctor is logged into their account. | |
| **Postconditions:** | Doctor’s availability (dates, time slots) is updated and saved in the system. | |
| **Flow of Activities:** | **Actor** |  |
| 1. Doctor selects “Manage Availability”.  2. Doctor sets available dates and time slots.  3. Doctor confirms the schedule. | 1.1 System retrieves the doctor’s current availability.  2.1 System validates the input and checks for conflicts.  3.1 System updates and saves the doctor’s availability. |
| **Exception Conditions:** | Time slots are overlapping with others, System couldn’t save the availability schedule. | |

**4. Prescription Management**

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Prescription Management | |
| **Scenario:** | Doctor creates and manages a patient's prescription. | |
| **Trigger Event:** | The doctor decided to give their patient a prescription. | |
| **Brief Description:** | The use case is for when a doctor wants to make, update, or delete a patient's prescription. All related information is stored in the systems database. User authentication will prevent a patient from changing these details. | |
| **Actors:** | Doctor | |
| **Related Use Cases:** | Medical Record Access, Billing and Payment | |
| **Stakeholders:** | Doctor, and Patient | |
| **Preconditions:** | Doctor is logged into their account. A patient's profile already exists.The patient was present for the appointment. | |
| **Postconditions:** | The prescription is saved to the patient's medical records. The patient is able to view their prescription. | |
| **Flow of Activities:** | **Actor** | **System** |
| 1. Doctor picks their patient and clicks "Prescription Management".  2. The doctor enters the prescription details.  3. The prescription is reviewed and confirmed by the doctor.  4. Patients log into their account to view prescriptions. | 1.1 Patient profile is displayed.  1.2 The system validates that their profile is valid.  2.1 The info is validated by the system.  3.1 The prescription is saved to the patient records.  3.2 The patient's prescription record is updated.  4.1 The patient's prescription is displayed by the system. |
| **Exception Conditions:** | No patient found, or Inputs are not valid. | |

**5. Medical Record Access**

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Medical Record Access | |
| **Scenario:** | Patients and doctors access medical records. | |
| **Trigger Event:** | Patient or doctor request to view medical records. | |
| **Brief Description:** | The doctor is able to view all current and past patients medical records and a patient can view only their own records. | |
| **Actors:** | Doctor, and patient | |
| **Related Use Cases:** | Prescription Management, 1-on-1 Sessions with Doctors | |
| **Stakeholders:** | Doctor, patient, administrator | |
| **Preconditions:** | User is authenticated and has the right authorization to view the record. The record exists in the database. | |
| **Postconditions:** | The record is displayed to the user. | |
| **Flow of Activities:** | **Actor** | **System** |
| 1. User logs into the system.  2. The doctor picks a patient and requests to view their records.  3. The patient clicks "View Medical Records". | 1.1 The user credentials are verified to ensure proper user access.  1.2 The user is granted access to the deserted records.  2.1 The patient's records are displayed for the doctor.  3.1 The patient's medical records are displayed for the patient. |
| **Exception Conditions:** | Unauthorized access, or no record found. | |

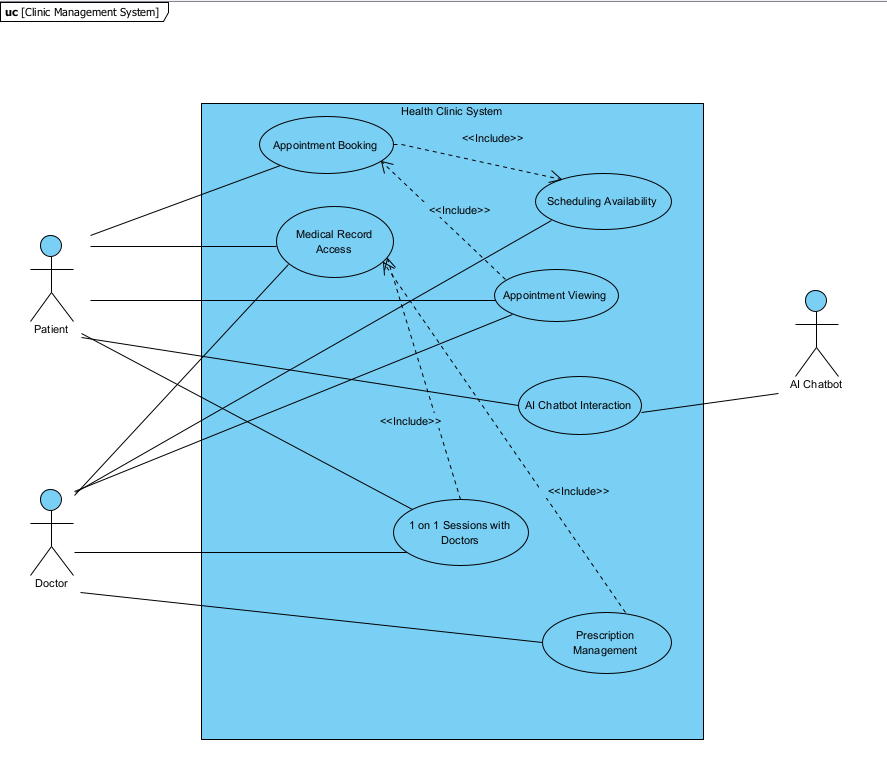
**6. Patients interact with AI Chatbot**

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | AI Chatbot Interaction | |
| **Scenario:** | Patient interacts with a mental health AI chatbot for guidance and support. | |
| **Trigger Event:** | The patient starts a conversation with the AI chatbot | |
| **Brief Description:** | The AI chatbot provides mental health assistance, answers general inquiries, and directs patients to professional help if needed. It ensures privacy and confidentiality while analyzing responses to give relevant guidance. | |
| **Actors:** | Patient and AI Chatbot | |
| **Related Use Cases:** | Medical Record Access, 1-on-1 Sessions with Doctors | |
| **Stakeholders:** | Patients, Doctors | |
| **Preconditions:** | The patient is logged into their own account.  The chatbot is online and active for use. | |
| **Postconditions:** | The patient receives responses from the chatbot.  If necessary, the chatbot recommends professional help. | |
| **Flow of Activities:** | **Actor** | **System** |
| 1. Patient selects the "Chat with AI" option.  2. The patient starts a conversation by entering a message.  3. The chatbot provides answers, coping strategies, or suggests a consultation with a doctor if needed.  4. The patient continues the conversation or ends the session. | 1.1. The chatbot interface is displayed.  2.1. The system analyzes the message and generates a response.  3.1. The chatbot checks sentiment and context before responding.  4.1. The conversation is logged securely if the patient consents. |
| **Exception Conditions:** | AI chatbot is unavailable.  Patient’s request cannot be processed.  The chatbot identifies a potential crisis and suggests emergency support. | |

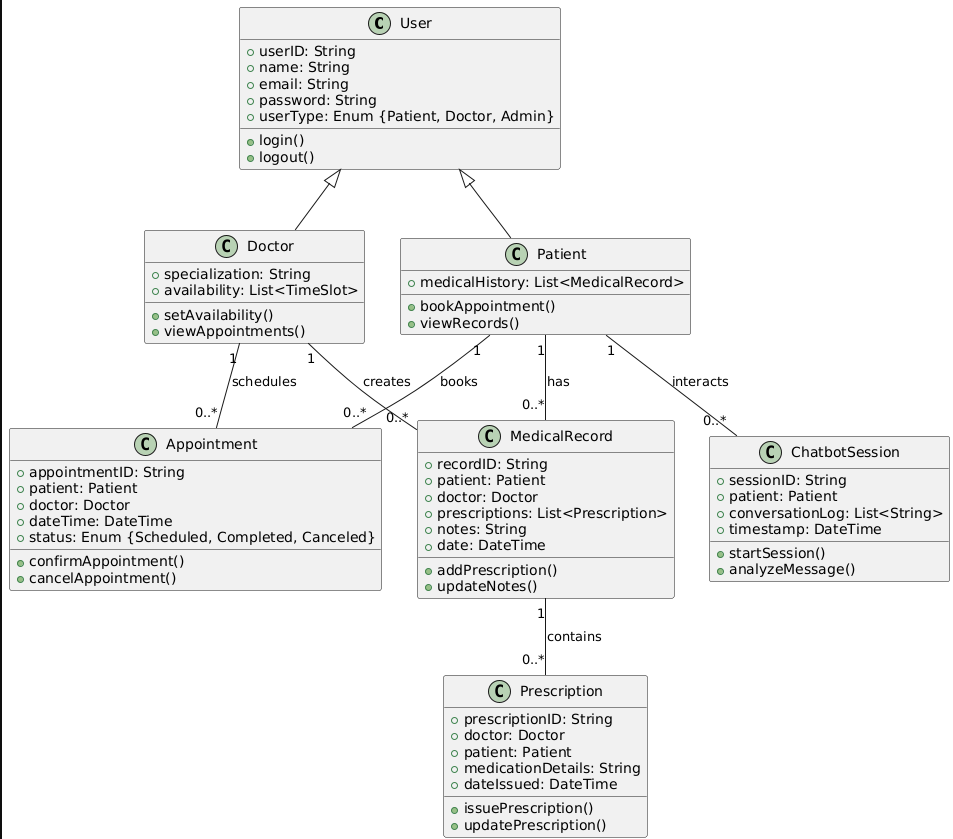
**7. 1 on 1 Sessions with Doctors**

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | 1 on 1 Sessions with Doctors | |
| **Scenario:** | A secure video consultation is scheduled between the patient and the doctor via Zoom. | |
| **Trigger Event:** | A patient or doctor schedules a 1-on-1 session. | |
| **Brief Description:** | The system generates a secure Zoom link for the virtual session, ensuring privacy and authentication. | |
| **Actors:** | Patient and Doctor | |
| **Related Use Cases:** | Medical Record Access, AI Chatbot Interaction | |
| **Stakeholders:** | Doctor, Patient, System Administrator | |
| **Preconditions:** | The patient and doctor are logged in.  A session appointment is scheduled. | |
| **Postconditions** | A secure Zoom link is generated  The session takes place. | |
| **Flow of Activities:** | **Actor** | **System** |
| 1. Patient or doctor requests a virtual session.  2. The system generates a secure Zoom link.  3. The patient and doctor receive notifications with the session details.  4. The session takes place at the scheduled time.  5. The session ends, and the doctor updates medical records if needed. | 1.1. The system verifies authentication and appointment details.  2.1. The Zoom link is securely stored and displayed to both users.  3.1. Email/SMS notifications are sent.  4.1. The system monitors session status.  5.1. The system securely logs the session details. |
| **Exception Conditions:** | Doctor or patient fails authentication.  Zoom link fails to generate.  Internet connection issues disrupt the session. | |

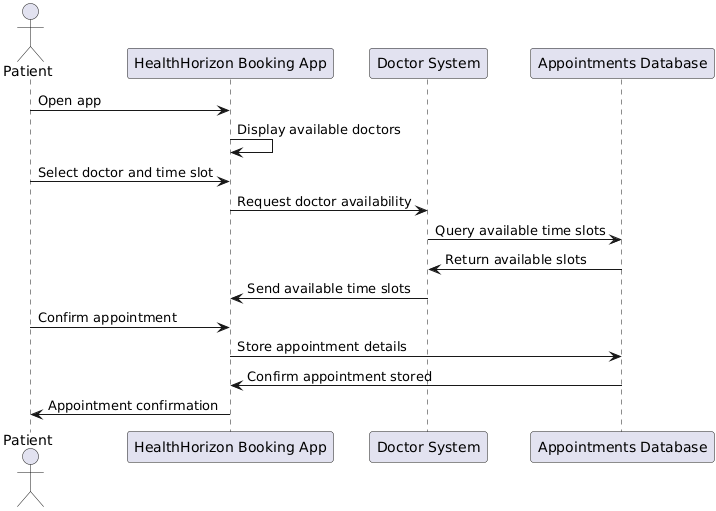
**Use Case Diagram**

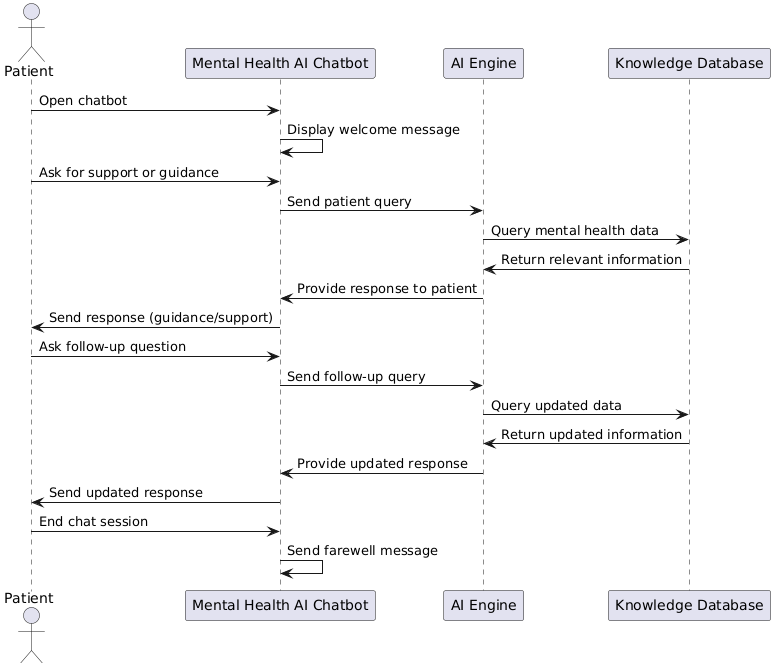
****

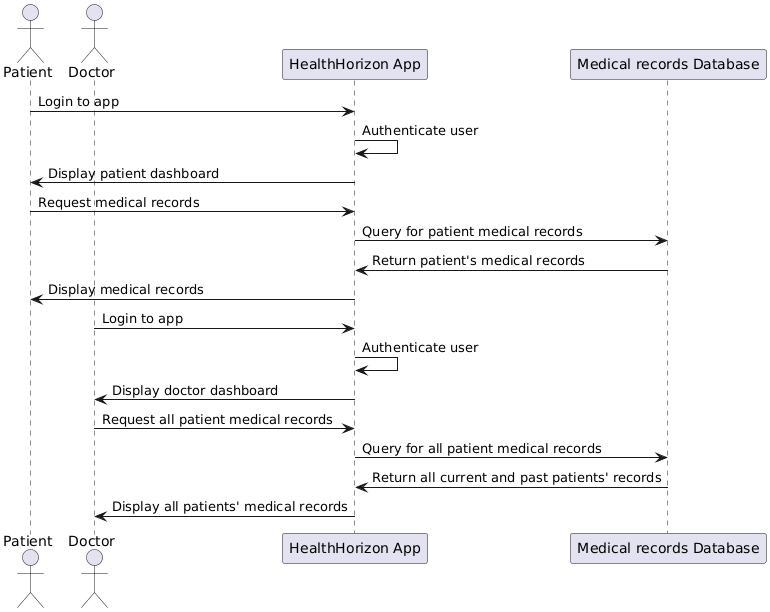
**Analysis Level Domain Class Diagram (7 classes)**

****

**System Sequence Diagrams (up to 3)**

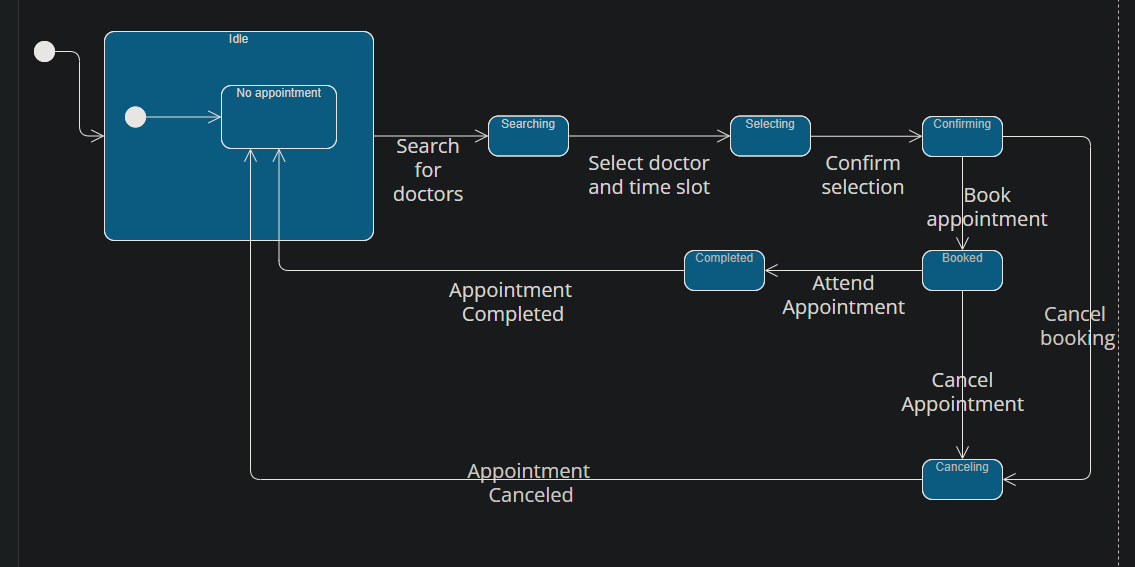
****

****

****

**\**

**State Machine Diagrams (Up to 3)**

****

The rest is TBD