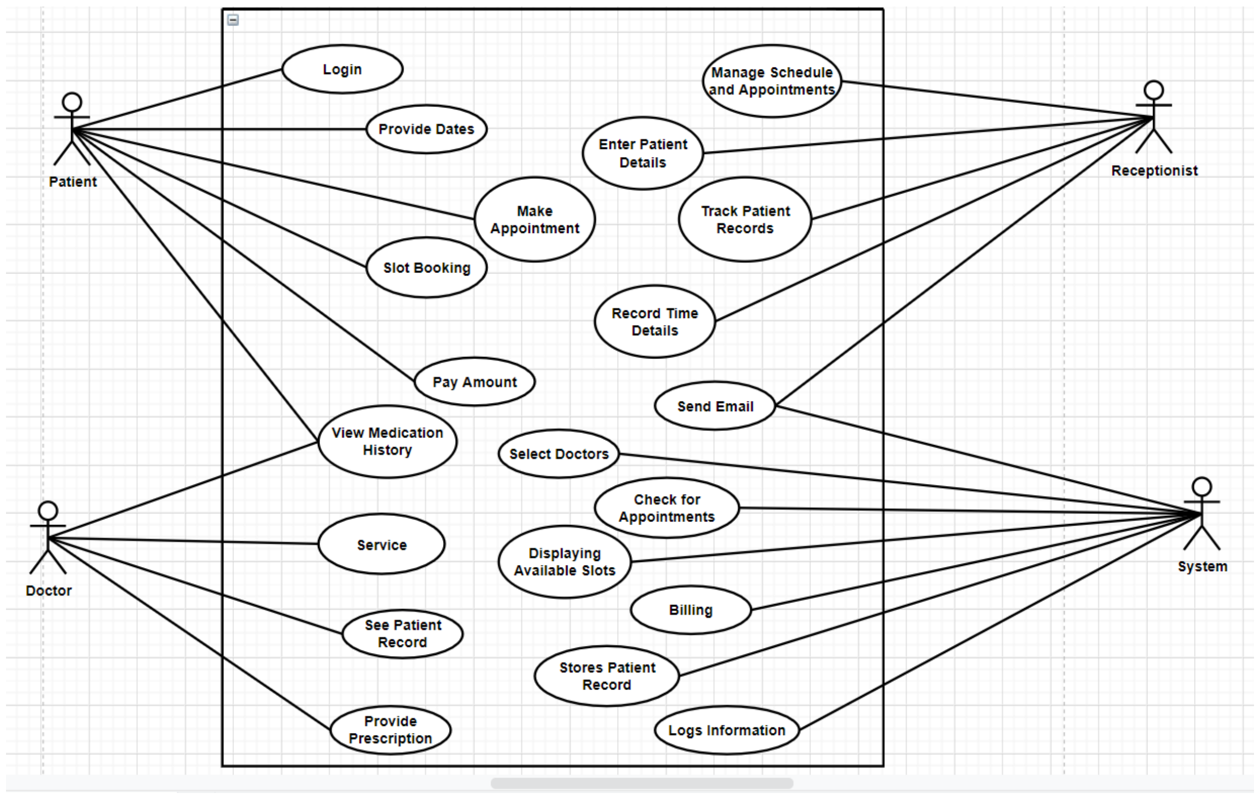
UML Use-Case Diagram and Descriptions:



UC1: Patient Registration
Description: This use case involves registering a new patient in the system, where the patient provides personal information, insurance details, etc. The system creates a new patient record with a unique ID.
UC2: Generate Medical Bill
Description: This use case is about generating a medical bill for a patient's services. The system retrieves the patient's medical record and generates a bill based on the services provided. The bill includes the patient's name, date of service, service code, description, and price.
UC3: Process Payment
Description: This use case covers processing a patient's payment for their medical services, which can be paid using cash, credit card, or insurance. The system verifies the payment and updates the patient's account.

UML Class Diagram and Descriptions:

```
+----------------------------------+
|          Patient                 |
+----------------------------------+
| - name : String                  |
| - date_of_birth : Date           |
| - address : String               |
| - contact_information : String   |
| - insurance_information : String |
+----------------------------------+
         ^
         |
         |
+----------------------------------+
|       MedicalService             |
+----------------------------------+
| - service_code : String          |
| - description : String           |
| - price : double                 |
+----------------------------------+
         ^
         |
         |
+----------------------------------+
|           Bill                   |
+----------------------------------+
| - patient_id : String            |
| - date_of_service : Date         |
| - service_codes : List<String>   |
| - descriptions : List<String>    |
| - prices : List<double>          |
| - total_amount : double          |
+----------------------------------+
         |
         |
         v
+----------------------------------+
|           Payment                |
+----------------------------------+
| - payment_type : String          |
| - amount : double                |
| - date_of_payment : Date         |
+----------------------------------+
```

Patient Class:
Attributes: name, date_of_birth, address, contact_information, insurance_information
MedicalService Class:
Attributes: service_code, description, price
Bill Class:
Attributes: patient_id, date_of_service, service_codes, descriptions, prices, total_amount
Payment Class: (not included in your code, but mentioned in the class descriptions)
Attributes: payment_type, amount, date_of_payment

Python Classes:
Below is the Python code

```
class Patient:
    def __init__(self, name, date_of_birth, address, contact_information, insurance_information):
        self.name = name
        self.date_of_birth = date_of_birth
        self.address = address
        self.contact_information = contact_information
        self.insurance_information = insurance_information

    # Getter and Setter methods for attributes
    def get_name(self):
        return self.name

    def set_name(self, name):
        self.name = name

    def get_date_of_birth(self):
        return self.date_of_birth

    def set_date_of_birth(self, date_of_birth):
        self.date_of_birth = date_of_birth

    def get_address(self):
        return self.address

    def set_address(self, address):
        self.address = address

    def get_contact_information(self):
        return self.contact_information

    def set_contact_information(self, contact_information):
        self.contact_information = contact_information

    def get_insurance_information(self):
        return self.insurance_information

    def set_insurance_information(self, insurance_information):
        self.insurance_information = insurance_information

class MedicalService:
    def __init__(self, service_code, description, price):
        self.service_code = service_code
        self.description = description
        self.price = price

    # Getter and Setter methods for attributes
    def get_service_code(self):
        return self.service_code

    def set_service_code(self, service_code):
        self.service_code = service_code
    def get_description(self):
        return self.description

    def set_description(self, description):
        self.description = description

    def get_price(self):
        return self.price

    def set_price(self, price):
        self.price = price

class Bill:
    def __init__(self, patient_id, date_of_service, service_codes, descriptions, prices, total_amount):
        self.patient_id = patient_id
        self.date_of_service = date_of_service
        self.service_codes = service_codes
        self.descriptions = descriptions
        self.prices = prices
        self.total_amount = total_amount

    # Getter and Setter methods for attributes
    def get_patient_id(self):
        return self.patient_id

    def set_patient_id(self, patient_id):
        self.patient_id = patient_id

    def get_date_of_service(self):
        return self.date_of_service

    def set_date_of_service(self, date_of_service):
        self.date_of_service = date_of_service

# You can add the Payment class following a similar pattern.
```

Github Link: