Patient Scheduling Subsystem

Domain Model Class Diagram and Specific Use Case Diagrams by Team One

Summary of Project

Our project aims to create an online appointment system where hospital patients can search for a doctor, see the doctor's availability, and schedule an appointment through the web.

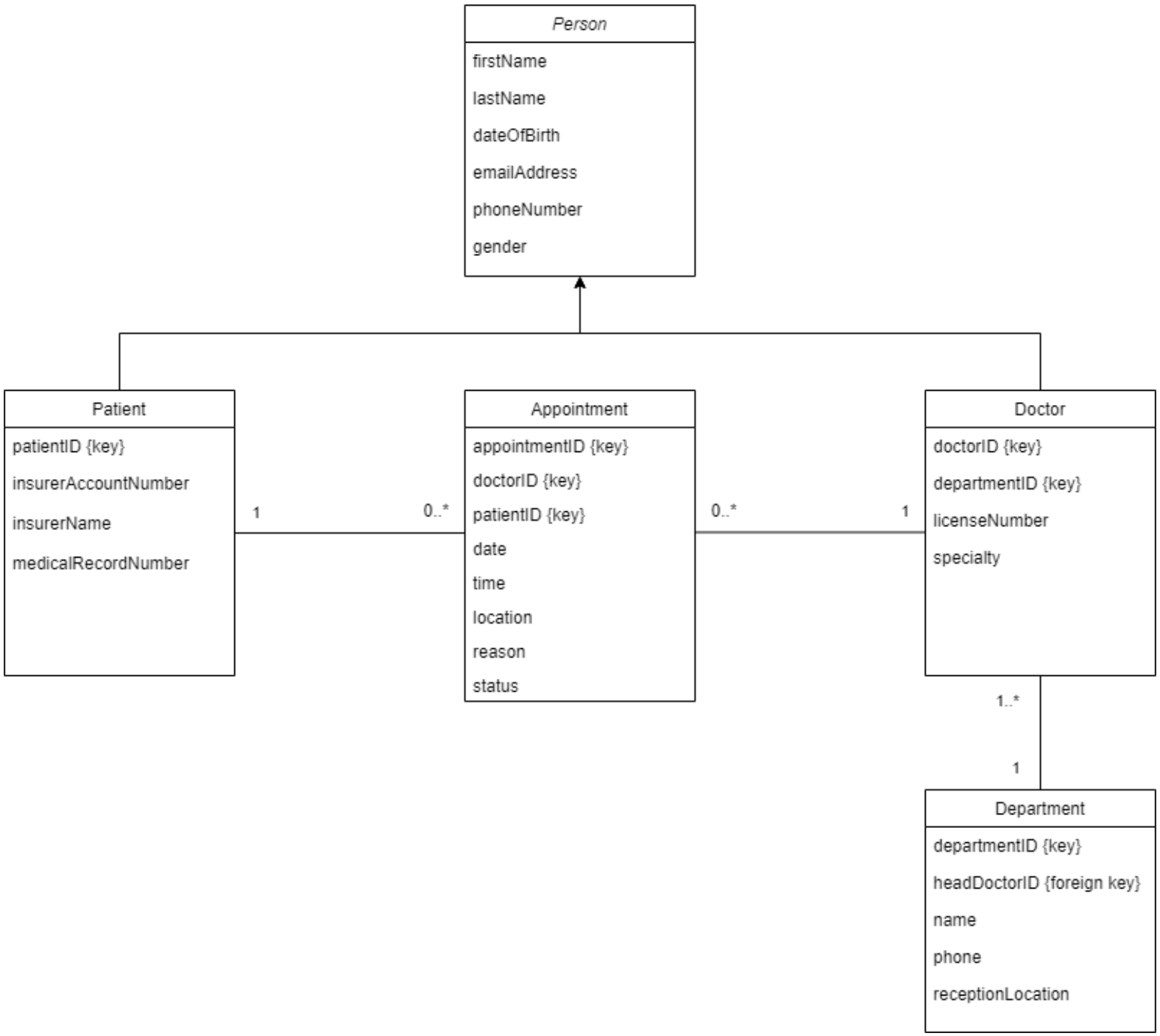
We present several design figures and tables to detail our proposed system. First we provide a Domain Class Diagram that presents the names and attributes of the object classes in the system and their multiplicity relationships. For example, the abstract Person class is the super class to the concrete Doctor and Patient classes. The latter share an association class called Appointment that models the appointment record between a doctor and a patient. Next we provide details for the "Create Appointment" use case with an Activity Diagram and a System Sequence Diagram. Finally we provide links to our team's organization page and repositories on GitHub. These figures and tables are as follows:

Part I - Domain and Use Case Modes

Section A

For classes that are related to your system: develop and present your domain model class diagram using UML tools. Be sure to identify each class, all its related attributes and how the classes are related by multiplicities. Also identify any super and subclass relationships.

Please see the following Domain Model Class Diagram:


Person
firstName
lastName
dateOfBirth
emailAddress
phoneNumber
gender
Patient
patientID {key}
insurerAccountNumber
insurerName
medicalRecordNumber
1
0..*
Appointment
appointmentID {key}
doctorID {key}
patientID {key}
date
time
location
reason
status
0..*
1
Doctor
doctorID {key}
departmentID {key}
licenseNumber
specialty
1..*
1
Department
departmentID {key}
headDoctorID {foreign key}
name
phone
receptionLocation

Section B

B1. Revise your system use cases according to any feedback from the previous assignment.

We did not have to revise any of our system use cases per your feedback on the previous assignment.
These are our system use cases so far with their descriptions from the previous assignment:

Description of Event	**Type of Event**	**Actor(s)**	**Use Case**	**Description of Use Case**
An actor requests a doctor's upcoming appointments.	External	Doctor, Secretary	Request Schedule	When a doctor or secretary requests a doctor's upcoming appointments, the system retrieves a doctor's upcoming appointments.
The system displays a list of a doctor's upcoming appointments.	Internal (State)	System	Display Schedule	The system displays a doctor's upcoming appointments.
An actor requests a patient's upcoming appointments.	External	Patient, Secretary	Request Upcoming Appointments	When a patient or secretary requests a patient's upcoming appointments, the system retrieves a patient's upcoming appointments.
The system displays a list of the patient's upcoming appointments.	Internal (State)	System	Display Upcoming Appointments	The system displays a patient's upcoming appointments.
The system sends a notification for an appointment coming in a week.	Internal (Temporal)	System	Send Notification	The system retrieves a patient's upcoming appointments and sends a notification to the patient about those that occur within a week.

Description of Event	**Type of Event**	**Actor(s)**	**Use Case**	**Description of Use Case**
An actor searches for a doctor.	External	Patient, Secretary	Search for Doctor	When a patient or secretary wants to schedule a new appointment by searching for a doctor, the system retrieves doctor information according to the search parameter(s).
The system displays a list of available doctors.	Internal (State)	System	Display Doctors	The system displays a list of doctors that match the search parameter(s).
An actor views a doctor's information.	External	Patient, Secretary	View Doctor Information	When a patient or secretary selects a doctor whose information they want to view, the system displays the doctor's information.
An actor selects a doctor from the doctor search results.	External	Patient, Secretary	Select a Doctor	When a patient or secretary selects a doctor from a list of doctors, the system retrieves that specific doctor's information.
The system provides a list of dates and times when the chosen doctor is available.	Internal (State)	System	Filter Doctor Availability	When a specific doctor has been chosen, the system verifies that doctor's availability and processes that doctor's schedule for open dates and times. This ensures that this specific doctor will not have overlapping appointments.

Description of Event	**Type of Event**	**Actor(s)**	**Use Case**	**Description of Use Case**
The system displays a list of dates and times for an appointment.	Internal (State)	System	Display Doctor Availability	When the system has filtered for a doctor's availability, the system displays those available dates and times.
An actor chooses an appointment date and time.	External	Patient, Secretary	Create Appointment	When a patient or secretary chooses an appointment date and time, the system creates a new appointment with a unique Appointment ID. This appointment is a relationship between the doctor and the patient.
An actor requests that an appointment be canceled.	External	Patient, Secretary	Cancel Appointment	When a patient or secretary chooses to cancel an appointment, the system retrieves that appointment using the unique Appointment ID and cancels that appointment.
An actor requests that an appointment be rescheduled.	External	Patient, Secretary	Reschedule Appointment	When a patient or secretary chooses to reschedule an appointment, the system retrieves that appointment using the unique Appointment ID, filters for a doctor's availability, and displays that new availability to the patient or secretary. Then the patient or secretary can choose a different appointment date and time.

B2. Write a fully developed description table for one use case.

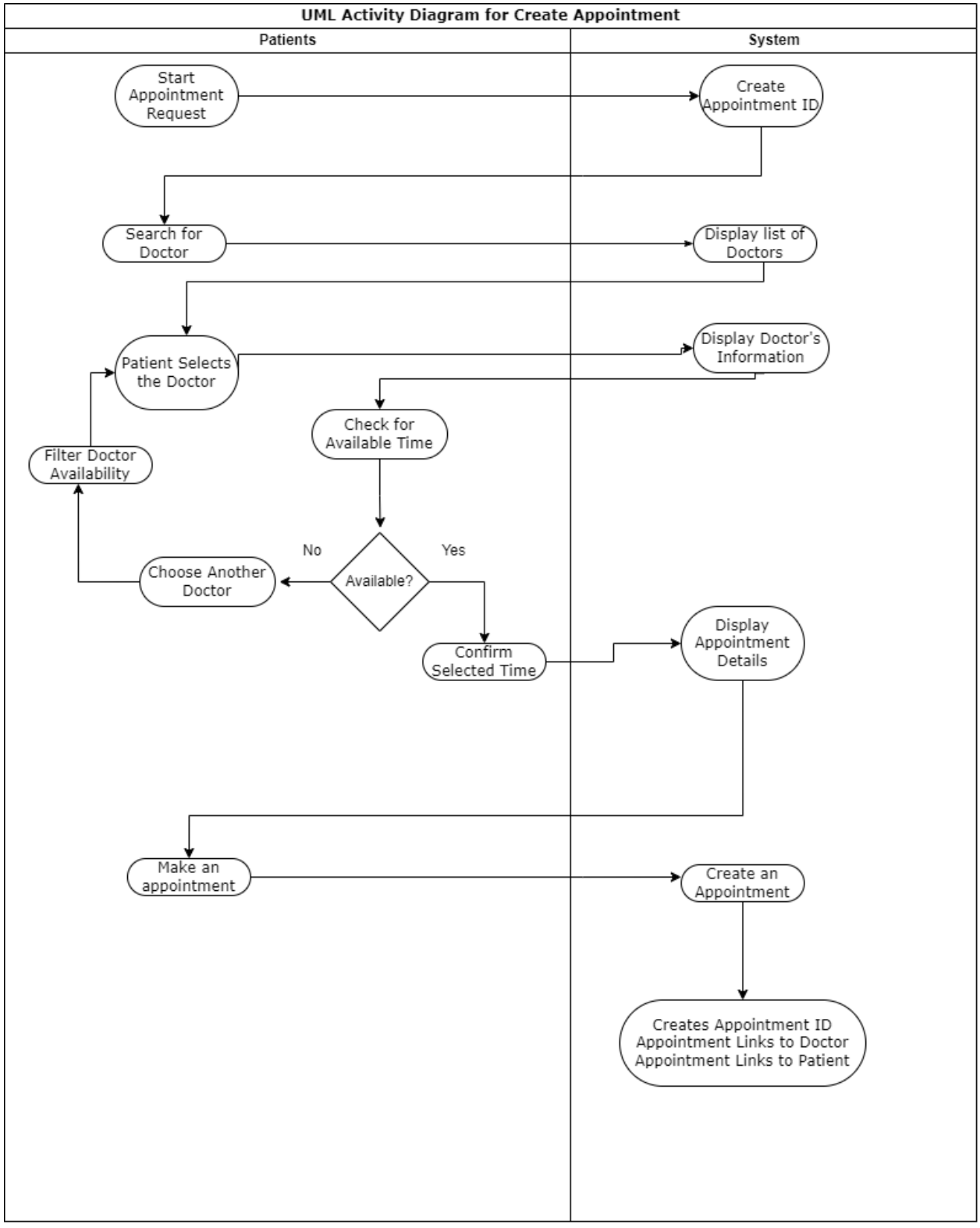
We choose the “Create Appointment” use case. Here is its fully-developed description table:

Use Case Name:	Create Appointment	
Scenario:	A Patient or Secretary creates an appointment for a given date and time.	
Triggering Event:	A Patient wants to create an appointment (In-Person through secretary or Online)	
Brief Description:	When a patient or secretary chooses an appointment date and time, the system crea appointment with a unique Appointment ID. This appointment is a relationship betwe the patient.	
Actors:	Patient, Secretary	
Related Use Cases:	May be invoked by Select Doctor Availability	
Stakeholders:	Accounting, Administration, Human Resources, Legal, Patient Services	
Pre-Conditions:	The external Content Server containing the doctor profile pictures is available.	
Post-Conditions:	Appointment will be created and saved. Appointment will be associated with the relevant patient and doctor. A unique ID will be created and assigned to the appointment.	
Flow of Activities:	**Actor**	**System**
	1. Patient indicates their desire to create a new appointment by clicking the “Schedule an Appointment” button.	1.1 System creates a new app the database. 1.2 System prompts with the c
	2. Patient enters doctor information: either their name or department.	2.1 System returns list of docto search criteria.
	3. Patient selects a specific doctor.	3.1 System returns a specific profile and their availability cal

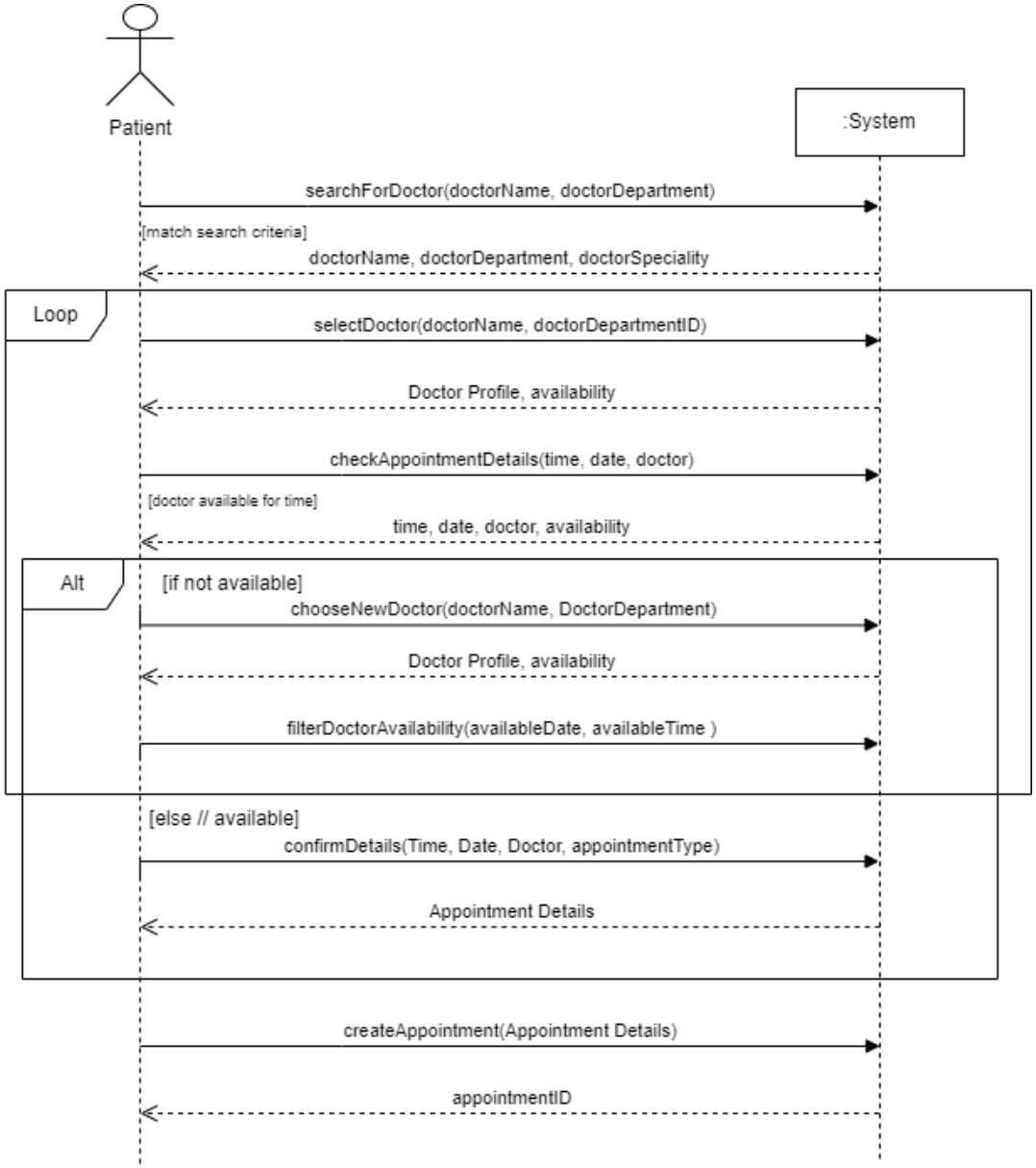
	4. Patient selects a date and time in the doctor's availability calendar.	4.1 System associates the app time, doctor, and patient by up appointment record in the data 4.2 System returns the appoin details to the patient, including number.
Exception Conditions:	2.1 No Doctors meet the search criteria	

Section C

C1. For the chosen use case in Section B: develop and present an activity diagram.



C2. For the chosen use case in Section B: develop and present a system sequence diagram.



Doctor Profile:
fullName, departmentName, specialties

Appointment details:
Assigned Doctor, Appointment Type, Date, Time, Location, status, appointmentID

Part II and III - GitHub Accounts and Organization

Personal Accounts

Please make sure that each team member has a GitHub account. Provide the URL to your team's GitHub organization.

The GitHub organization page and project repository is:
https://github.com/cs3560-03-01

Part IV - Project Skeleton

Develop a project skeleton for your system, including any class attributes, method parameters and returns, if any; be sure to comment your code and make it presentable.

We made some commits. Please see the URL in part (2) above and the specific repository.