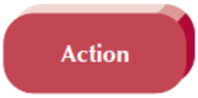
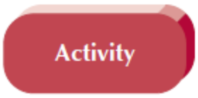
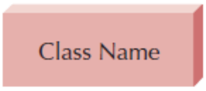
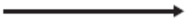
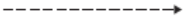
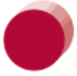
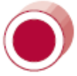
Software Engineering Lab #10
Software Analysis and Design with UML (Part 1)

Business Process Modeling with Activity Diagrams

Activity diagrams are used to model the behavior in a business process independent of objects. It is a business work flow diagram. Activity diagrams include notation that addresses the modeling of parallel, concurrent activities and decision making processes. Activity diagrams portray the primary activities and the relationships among the activities in a process. It typically has the following elements:

- **Actions and Activities** represent manual or computerized behavior. They should have a short name that begins with a verb and ends with a noun. An activity can be decomposed further into a set of activities and/or actions, whereas an action cannot be decomposed. In most cases **each activity is associated with a use case**.
- **Object Nodes** represent information which flows from one activity to another.
- **Control Flows and Object Flows:** Control flows model the paths of execution through a business process and can be attached only to actions or activities. Object flows model the flow of objects through a business process. Object flows are used to show the actual objects that flow into and out of the actions or activities.
- **Control Nodes**: There are seven different types of control nodes in an activity diagram, i.e. initial, final-activity, final-flow, decision, merge, fork, and join.

There are many circumstances where an activity diagram can be broken up so that it can be used to assign responsibility to many objects which perform activities. This is accomplished through the use of **swimlanes**.

An Action: ■ Is a simple, non-decomposable piece of behavior ■ Is labeled by its name	
An Activity: ■ Is used to represent a set of actions ■ Is labeled by its name	
An Object Node: ■ Is used to represent an object that is connected to a set of Object Flows ■ Is labeled by its class name	
A Control Flow: ■ Shows the sequence of execution	
An Object Flow: ■ Shows the flow of an object from one activity (or action) to another activity (or action)	
An Initial Node: ■ Portrays the beginning of a set of actions or activities	
A Final-Activity Node: ■ Is used to stop all control flows and object flows in an activity (or action)	

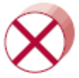
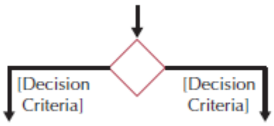
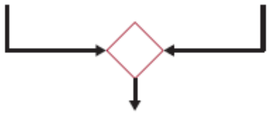
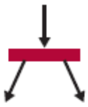
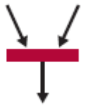
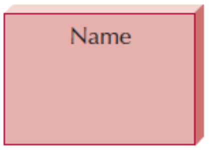
A Final-Flow Node: ■ Is used to stop a specific control flow or object flow	
A Decision Node: ■ Is used to represent a test condition to ensure that the control flow or object flow only goes down one path ■ Is labeled with the decision criteria to continue down the specific path	
A Merge Node: ■ Is used to bring back together different decision paths that were created using a decision-node	
A Fork Node: ■ Is used to split behavior into a set of parallel or concurrent flows of activities (or actions)	
A Join Node: ■ Is used to bring back together a set of parallel or concurrent flows of activities (or actions)	
A Swimlane: ■ Is used to break up an activity diagram into rows and columns to assign the individual activities (or actions) to the individuals or objects that are responsible for executing the activity (or action) ■ Is labeled with the name of the individual or object responsible	

Figure 1: Syntax for an Activity Diagram

Case Study 1: Restaurant System

The system to develop is intended to support the day-to-day operations of a restaurant by improving the processes of making reservations and allocating tables to customers. The restaurant staff can book a reservation for a customer if there is a suitable table available at the requested date and time. The booking will include the customer's name and phone number, the time of the booking, the number of covers (no. of diners expected) and the table number. When a party arrives and is seated at their reserved table, the corresponding booking will be marked as the customer being arrived. The customer can also cancel a booking. Diners can also walk-in and seat at an available table without having to book in advance. In this case, no record of the customer's name and telephone, but only the covers and table number are recorded.

1. Creating an activity diagram in Enterprise Architect

1.1 Starting a new project

Assume that you have installed Enterprise Architect in your system. Run the program.

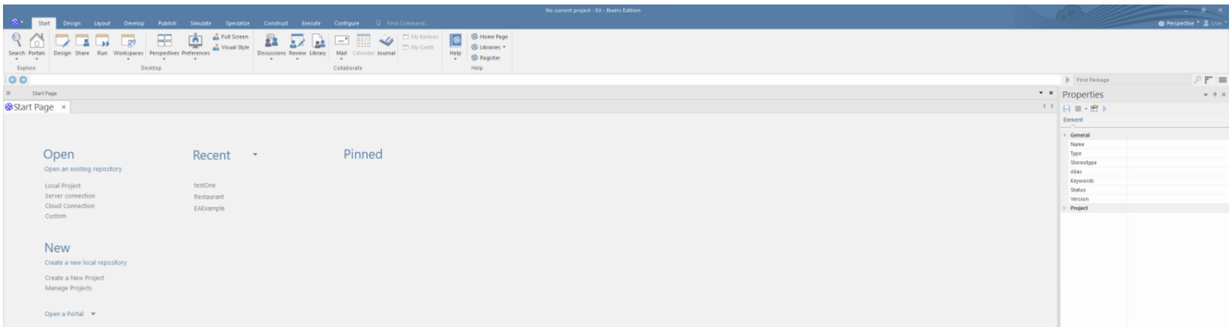
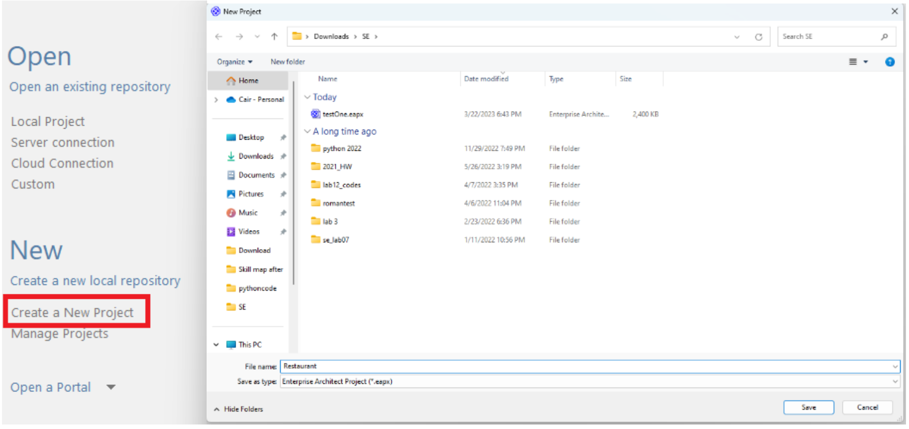


Figure 2: Enterprise architect Start Page

Then to create a new project, select "**Create New Project**" from the start page. For this exercise, create a new project named "**Restaurant**". Afterwards, mark a Business Process model on "Select model(s)" window (Fig. 2 & 3).



Name	Date modified	Type	Size
Today			
testOne.eapx	3/22/2023 6:43 PM	Enterprise Archite...	2,400 KB
A long time ago			
python 2022	11/29/2022 7:49 PM	File folder	
2021_HW	5/26/2022 3:19 PM	File folder	
lab12_codes	4/7/2022 3:35 PM	File folder	
romantest	4/6/2022 11:04 PM	File folder	
lab 3	2/23/2022 6:36 PM	File folder	
se_lab07	1/11/2022 10:56 PM	File folder	

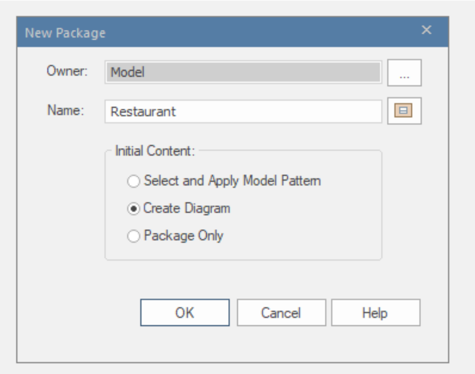
File name: Restaurant

Save as type: Enterprise Architect Project (*.eapx)

Figure 3: EA creates a new project window

1.2 Creating a new package

From the Project browser window, right click on the **Business Process Model** folder then select "**Add ->Add Package…**". For this exercise, create a new package named "**Restaurant**".



Owner: Model

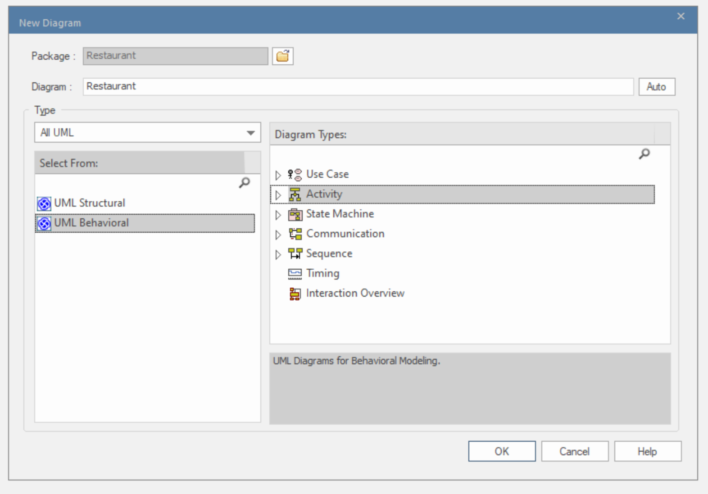
Name: Restaurant

Initial Content:

○ Select and Apply Model Pattern

● Create Diagram

○ Package Only



Package : Restaurant

Diagram : Restaurant

Type

All UML

Select From:

UML Structural

UML Behavioral

Diagram Types:

Use Case

Activity

State Machine

Communication

Sequence

Timing

Interaction Overview

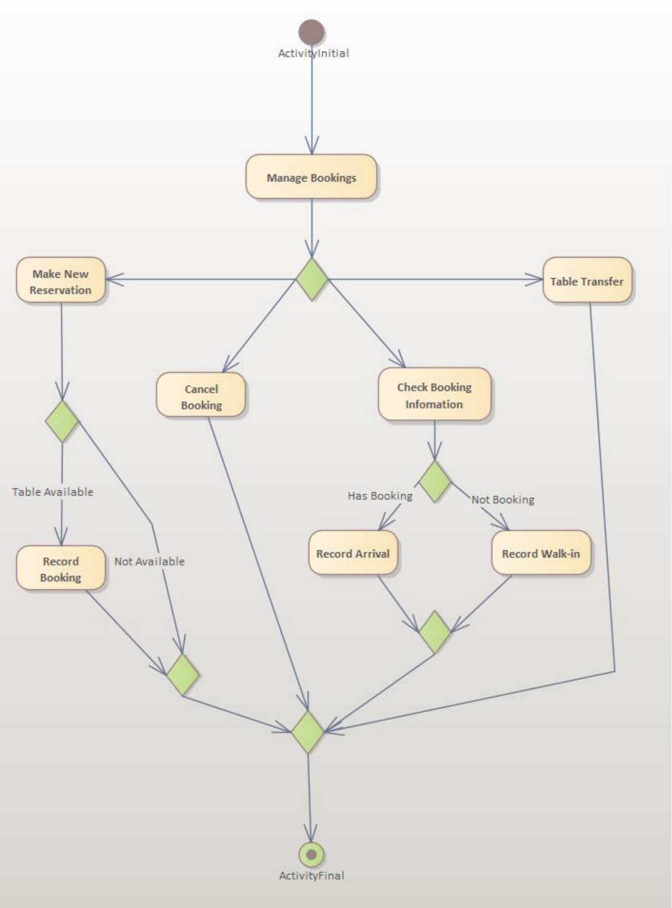
UML Diagrams for Behavioral Modeling.

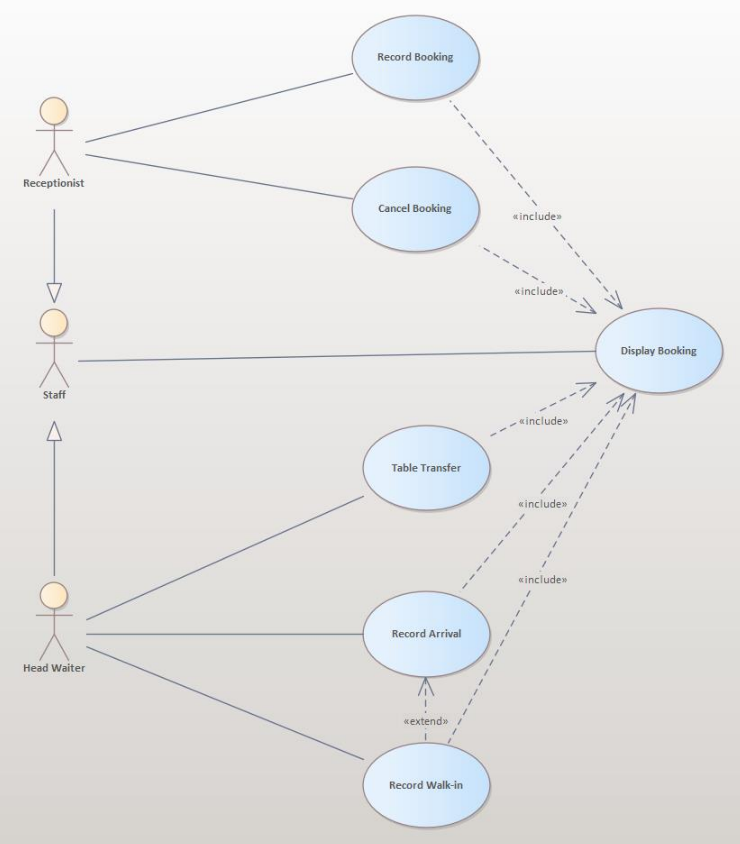
Figure 4: Insert Additional Package **Figure 5**: Enterprise architect "New Diagram" window

1.3 Creating a new activity diagram

From the Enterprise architect browser window, right click on the Restaurant package created on the previous step. Select "**Add > Add Diagrams…**". Choose "**UML Behavioral->Activity**" on "**New Diagram**" window. After creating an activity diagram, you can right click on it and select "**Properties…**" to change the diagram features such as name, description, and properties. To start adding components into the activity diagram, if the drawing toolbar is not already shown, open the drawing toolbar by press "**Alt+5**". You can select a component, from the diagram tools in the drawing toolbar, to add into the activity diagram. You can right-click on a component in the diagram and then select "**Appearance…**" from the menu if you want to customize the component appearance and select "**Properties…**" to add to it any further detail description.

- **Experiment with the drawing toolbar and draw the following activity diagram**





2. Creating a use case diagram

From EA "Project Browser" window, right-click on the Restaurant package. Select "**Add -> Add Diagrams…**" then choose "**UML Behavioral->Use Case**" on "New Diagram" window. After creating a use case diagram, you can right-click on it and select "**Properties…**" to change the diagram features such as name, description, and properties. To start adding components into the use case diagram, if the drawing toolbar is not already shown, open the drawing toolbar by press "**Alt+5**". You can select a component, from the diagram tools in the drawing toolbar, to add into the use case diagram. You can right-click on a component in the diagram and then select "**Appearance…**" from the menu if you want to customize the component appearance and select "**Properties…**" to add to it any further detail description.

- **Draw the following use case diagram (see above)**

Domain Modeling

Use cases are intended to be comprehensible to both developers and users of the system. They are described using terminology taken from the business domain rather than from implementation or computer-oriented vocabularies. A useful activity that is often carried out in parallel with use case modeling is to describe systematically the business concepts that are used in the use case descriptions.

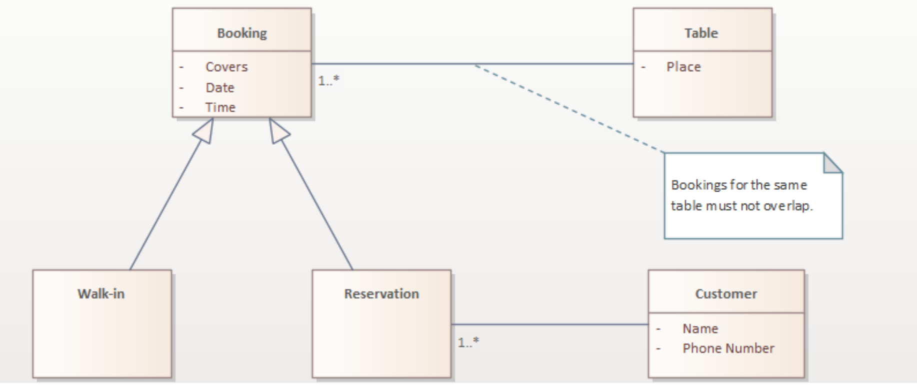
One common way of documenting this activity is by producing a class diagram showing the most important business concepts and the relationships between them. Such a class diagram is often referred to as a *domain model*.

Domain models typically do not make use of the full range of class diagram notation. The classes in a domain model usually represent entities or concepts that are significant in the real world context of the system. The data that the system must record is modeled as attributes of these classes, and a domain model also shows relationships between these concepts, using associations and generalization. Domain models do not usually contain operations: these are defined later, once the implementations of the use cases have been considered in more detail.

3. Creating a domain model

From EA "Project Browser" window, right-click on the Restaurant package. Select "**Add -> Add Diagrams…**" then choose "**UML Structural->Class**" on "New Diagram" window. After creating a class diagram, you can right-click on it and select "**Properties…**" to change the diagram features such as name, description, and properties. To start adding components into the class diagram.

- **Draw the following domain model:**



Case Study 2: Appointment System

The system to develop is an appointment system for a doctor's office. The system supports management of patients' information and their appointments. Patients can request for a new appointment at a available time slot. They can also change or cancel their appointment later. **Figure 6** shows an activity diagram that represents part of an appointment system. The diagram represents the business process being developed. It also documents requirements of the system.

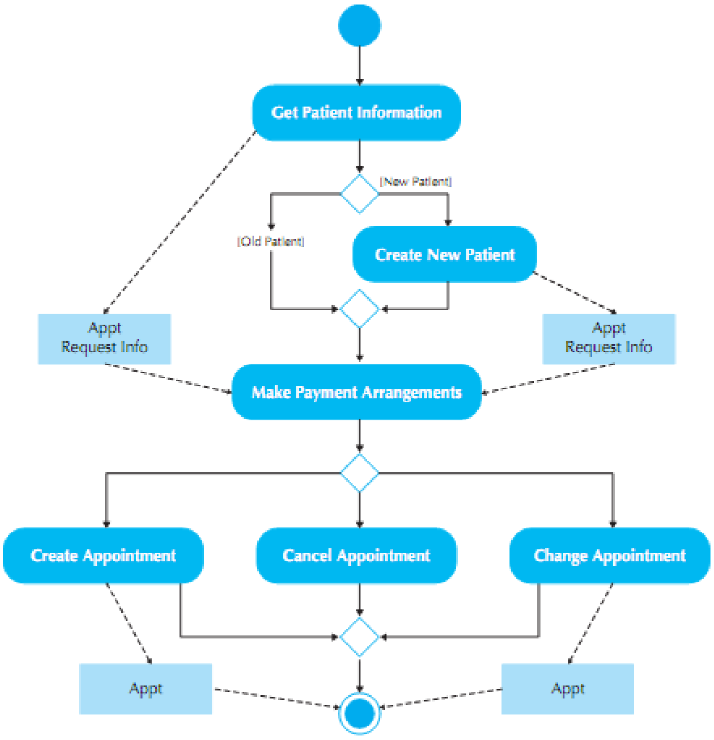


Figure 6: Activity Diagram for Appointment System

After doing a business modeling, you should be able to extract enough information to derive a use case description and a use case diagram from a business model. **Figure 8** shows a sample use case description for the "**Make Appointment**" use case.

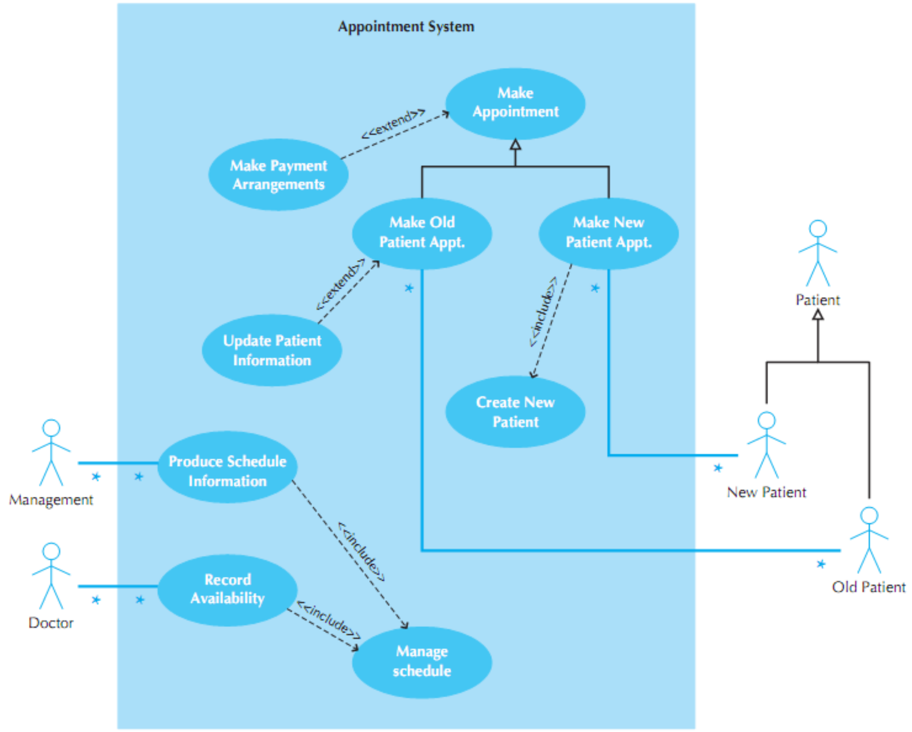
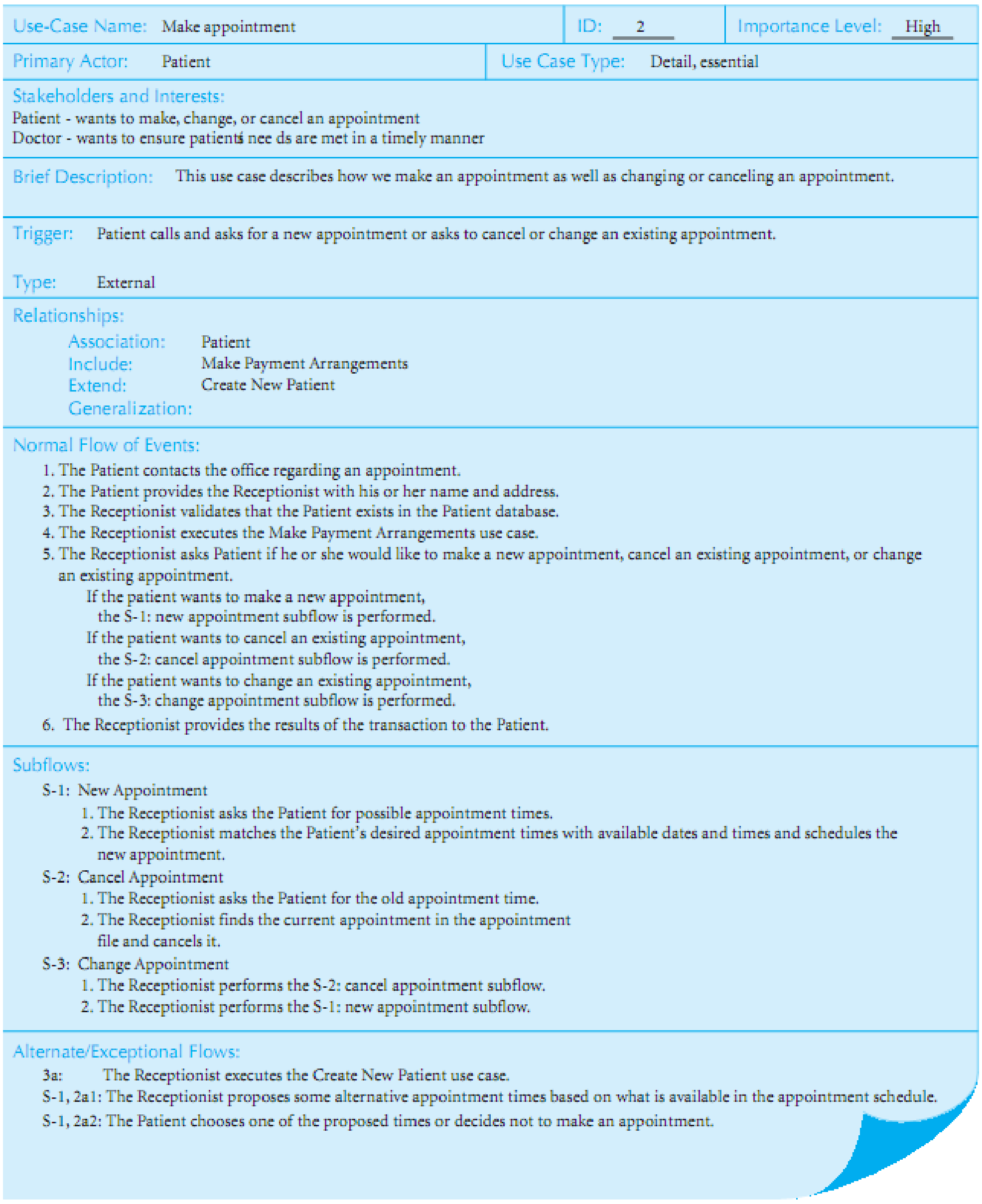


Figure 7: Sample use case diagram for the appointment system



Use-Case Name: Make appointment | ID: 2 | Importance Level: High

Primary Actor: Patient | Use Case Type: Detail, essential

Stakeholders and Interests:
Patient - wants to make, change, or cancel an appointment
Doctor - wants to ensure patient's nee ds are met in a timely manner

Brief Description: This use case describes how we make an appointment as well as changing or canceling an appointment.

Trigger: Patient calls and asks for a new appointment or asks to cancel or change an existing appointment.

Type: External

Relationships:
Association: Patient
Include: Make Payment Arrangements
Extend: Create New Patient
Generalization:

Normal Flow of Events:

1. The Patient contacts the office regarding an appointment.
2. The Patient provides the Receptionist with his or her name and address.
3. The Receptionist validates that the Patient exists in the Patient database.
4. The Receptionist executes the Make Payment Arrangements use case.
5. The Receptionist asks Patient if he or she would like to make a new appointment, cancel an existing appointment, or change an existing appointment.
   If the patient wants to make a new appointment,
   the S-1: new appointment subflow is performed.
   If the patient wants to cancel an existing appointment,
   the S-2: cancel appointment subflow is performed.
   If the patient wants to change an existing appointment,
   the S-3: change appointment subflow is performed.
6. The Receptionist provides the results of the transaction to the Patient.

Subflows:

S-1: New Appointment
1. The Receptionist asks the Patient for possible appointment times.
2. The Receptionist matches the Patient's desired appointment times with available dates and times and schedules the new appointment.

S-2: Cancel Appointment
1. The Receptionist asks the Patient for the old appointment time.
2. The Receptionist finds the current appointment in the appointment file and cancels it.

S-3: Change Appointment
1. The Receptionist performs the S-2: cancel appointment subflow.
2. The Receptionist performs the S-1: new appointment subflow.

Alternate/Exceptional Flows:
3a: The Receptionist executes the Create New Patient use case.
S-1, 2a1: The Receptionist proposes some alternative appointment times based on what is available in the appointment schedule.
S-1, 2a2: The Patient chooses one of the proposed times or decides not to make an appointment.