Name: Christopher Nowacki Class: CIS 280 Assignment: Module 6

1 Part I

1. **What is O-O analysis?**

   Objected Oriented Analysis describes an information system via object identification. This gives you an overview of what entities, functions, messages, and transactions, and data will be required to create the information system.

2. **Define an object in an information system and provide three examples.**

   An object represents a person, place, item, event, or transaction. It contains attributes and/or methods.

   (a) Object: Librarian
   (b) Object: Library
   (c) Object: Book Reading

3. **Define an attribute and provide three examples.**

   An **attribute** is a characteristic that describes an object.

   (a) Object: Librarian
   Attributes: *Name, Employee_ID*
   (b) Object: Library
   Attributes: *Name, Location, Location_ID, Open?*
   (c) Object: Book Reading
   Attributes: *Reader, Date, Time*

4. **Define a *method* and provide three examples**

   Methods are functions that the object performs when it receives a message.

   (a) Object: Librarian
   Methods: *Update Book Status, Issue Library Card*
   (b) Object: Library
   Methods: *Schedule Event, List Events, Change Hours*
   (c) Object: Book Reading
   Methods: *Add Reading, Delete Reading, Update Reading*

5. **Define *encapsulation* and explain how it is used in OO analysis.**

   With **encapsulation**, *data and functions are self-contained* in an object class. IT is useful in the OOP concept as it allows objects to be used as modular components and prevents its internal code from being changed.

6. **Define a *class, subclass, and superclass* and provide three examples of each.**

   A **class** is a blueprint for creating *objects.* A **subclass** is a class that inherits the properties from another class, called a **superclass** or **parent class**.

   class **record** in *superclass* **recorded media**
   class **7"** in *superclass* **record**
   class **12"** in *superclass* **record**

   Superclass would be **recorded media**, it's *child class* in this case would be **record, cassette, cd, etc**.

   record would be the *parent class* for: **7", 10", 12", etc**, making the mentioned record sizes the **subclass**.

7. **Explain the concept of *inheritance* in object relationships. Inheritance** enables an object to derive one or multiple attributes from its **parent class**. The class inheriting the attributes is the **child class**.

   For example, an **animal** class could have methods like eat_food() and drink_water(), and a **tiger** subclass could inherit those methods.

8. **Draw an *object relational diagram* for a typical library system.**

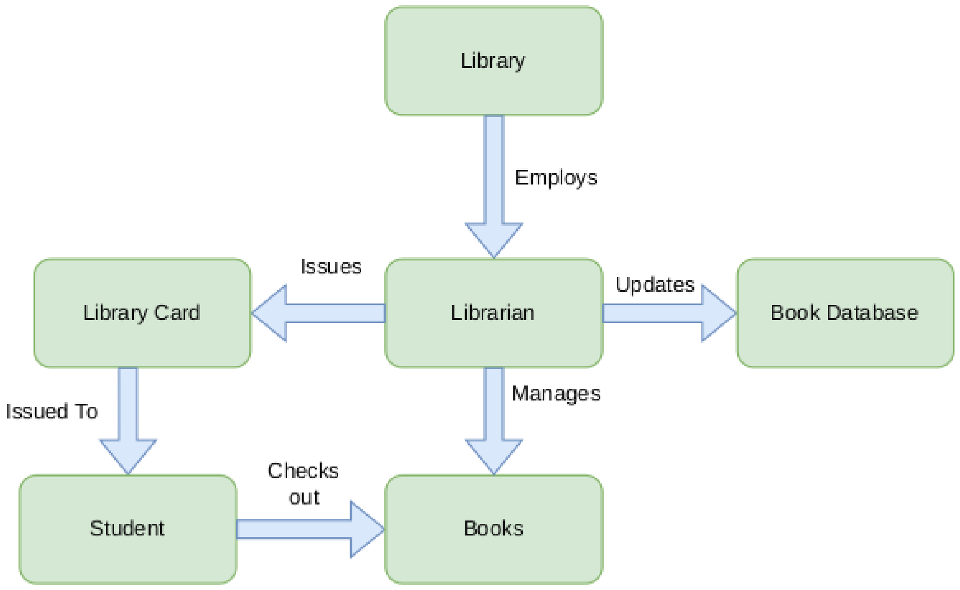

Figure 1: Objected Relational Diagram

9. **Define a *use case* and a *use case diagram* and prepare a sample of each.**

   A **use case** is a description of a way that a user can interact with a system or product.
   A **use case diagram** is a visual diagram that shows the process, entities, and "touch points", i.e. where an entity is involved in a process.

Use Case Name	Checkout Book
Actor	**Librarian**, *Student*
Preconditions	*Student* has **library card**
Postconditions	**book** status is changed to *checked out*
Description	A student wants to check out a book, and the Librarian helps check it out to them manually.
Steps	1.1. Student selects book for checkout. 1.2. Librarian confirms that the book is available for checkout. 1.3. Librarian confirms there are no holds on student's account. 1.4. Library updates book status to **CHECKED OUT**. 1.5. Student is issued a receipt with a **DUE BY** date.
Alternate Flow	1.3.1. If Student has holds on account, they must settle holds before more books can be checked out.

Table 1: Use Case Definition for "Checkout Book"

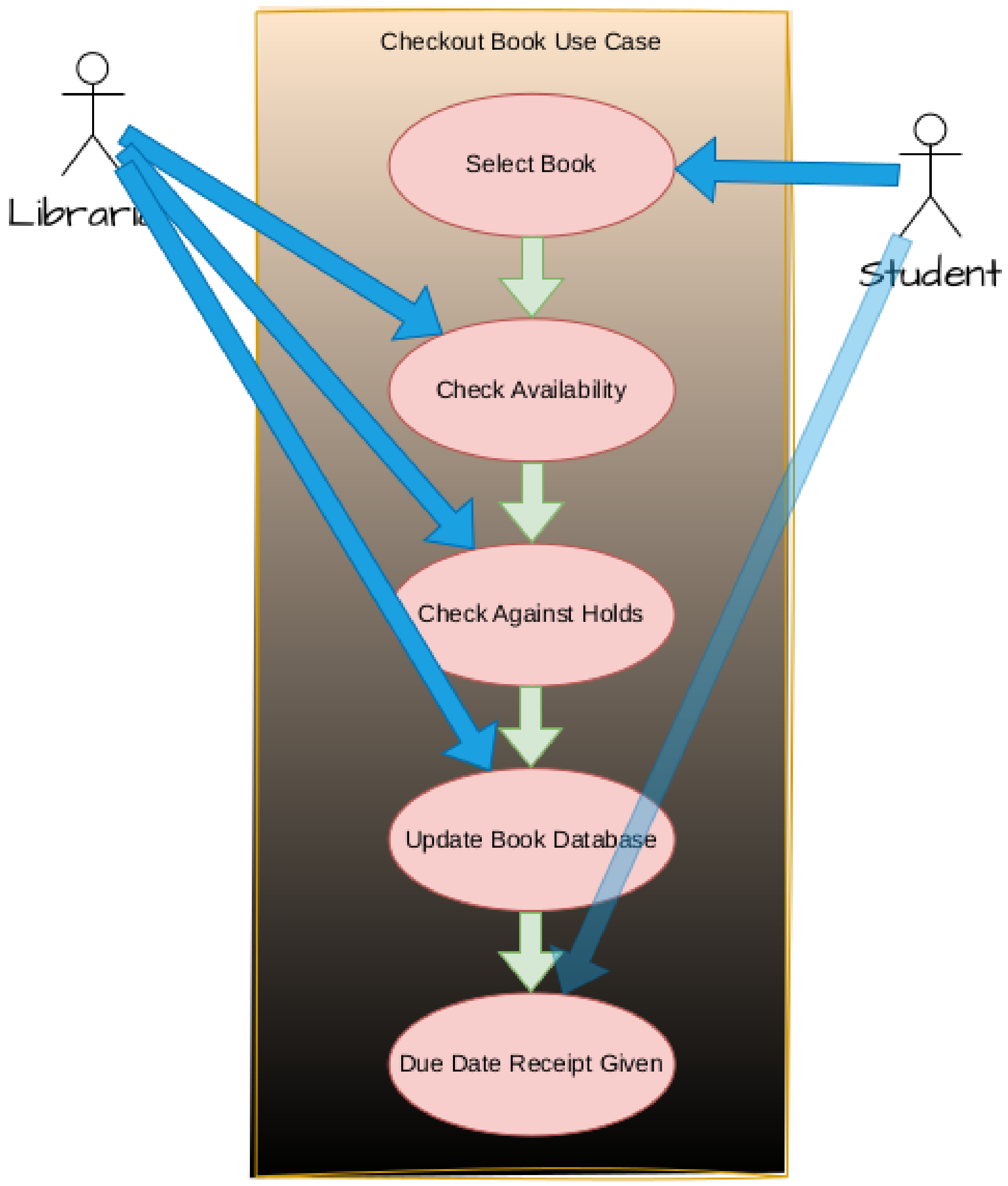

Figure 2: Book Checkout Use Case Diagram

10. **Why is it important to use a modeling tool and not just a diagramming tool during O-O analysis?**

    Modeling tools provide a solutions to assist analysts in documenting system components. They also improve modularity. Once an object is described in one section, it can be reused with little extra effort, where in a diagramming tool you would have to re-draw and create the link every time. A modeling tool will also adhere to UML guidelines so it will be harder to ”break the rules”.

2 Part II

2.1 Use Cases/Use Case Diagram

Use Case Name	Schedule an Appointment
Actor	**Patient**, *Admin*
Preconditions	• The doctor's schedule is open. • Patient's records are accessible to the admin.
Postconditions	Appointment is scheduled and both the patient and doctor have details of appt.
Description	A patient wishes to schedule an appointment with a doctor. The patient can either call the office and the admin will schedule it, or the patient can use the online portal to book it.
Steps	1.1 Patient contacts the office or accesses the online portal. 1.2 Patient provides desired date and time or selects from available slots. 1.3 Administration checks doctor's availability. 1.4 Appointment is scheduled. 1.5 Patient received a confirmation.
Alternate Flow	1.3.1 During step 1.3., admin finds doctor has no open appointments available. 1.3.2 Admin offers alternate provider. 1.3.3 Patient selects a new slot or does not move forward with booking. 1.3.4 If new slot is selected, proceed to step 1.4 of the basic flow.
Exceptions	• If the system or online portal malfunctions, patient will be advised to contact later or use another method of booking. • If the doctor is on an extended leave, admin might suggest scheduling with an alternate physician.

Table 2: Use Case Definition for ”Schedule an Appointment”

Use Case Name	Update Medical Information
Actor	**Patient**, *Admin*
Preconditions	• Patient has an account within the system.
Postconditions	Medical information is up-to-date.
Description	Patient's medical profile is out-of-date, and needs to be updated. Patient can log-in and update through a portal, or could fill out updated information at their next in office visit via pen and paper, which will be manually updated by admin.
Steps	1.1 Patient comes into the office for a visit or logs into the portal. 1.2 Patient updates and saves medical history and billing details. 1.3 Notification of update is sent to administration. 1.4 Administration reviews the changes and confirms updated details in the system.
Exceptions	• If the online system malfunctions, admin will encourage patient to update the information at their next visit or try again later.

Table 3: Use Case Definition for ”Update Medical Information”

Use Case Diagram for *X-Ray Procedure*

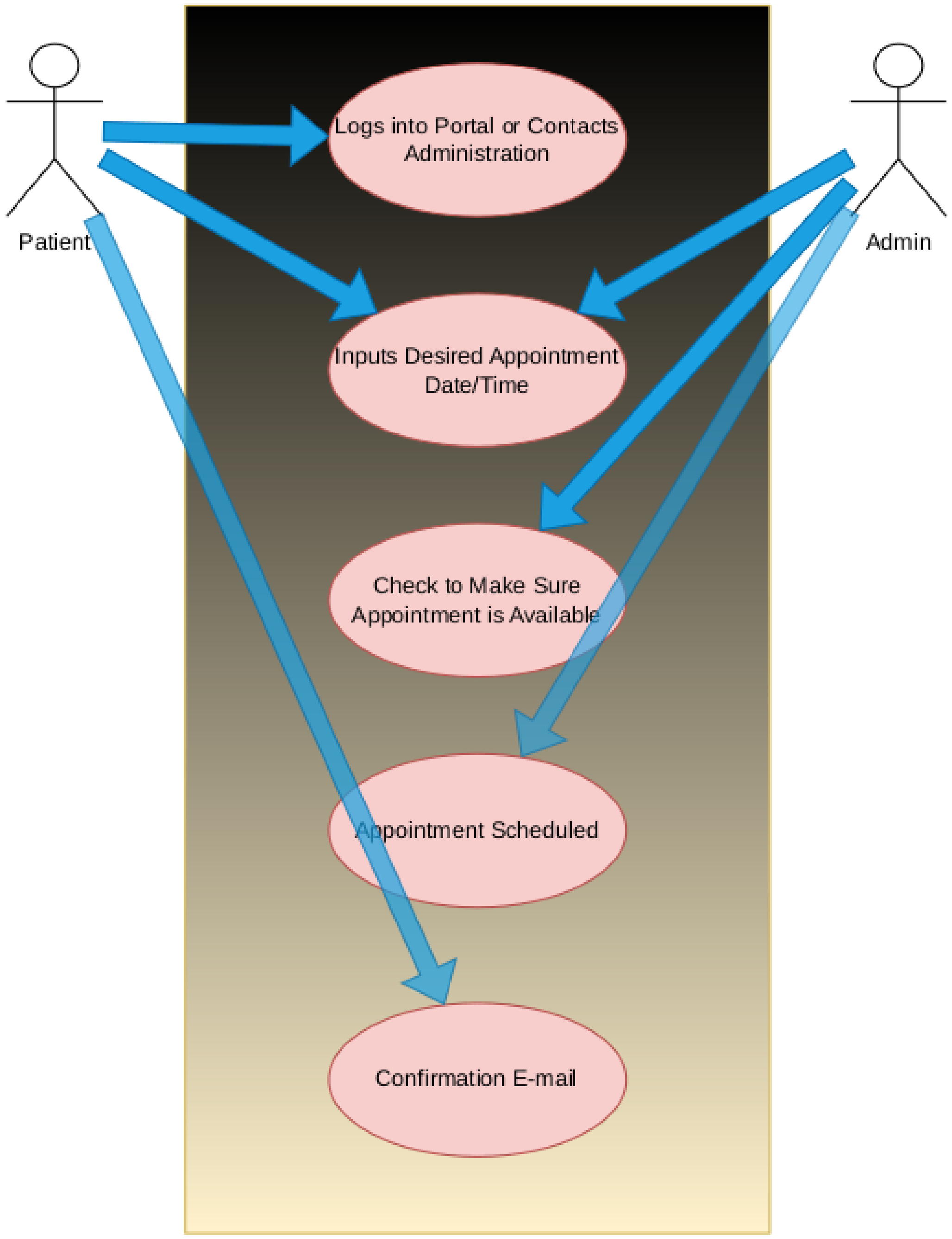
Patient
Admin
Logs into Portal or Contacts Administration
Inputs Desired Appointment Date/Time
Check to Make Sure Appointment is Available
Appointment Scheduled
Confirmation E-mail

Figure 3: Use Case Diagram

2.2 Domain Class Diagram

Domain Class Diagram for *Appointment Booking*.

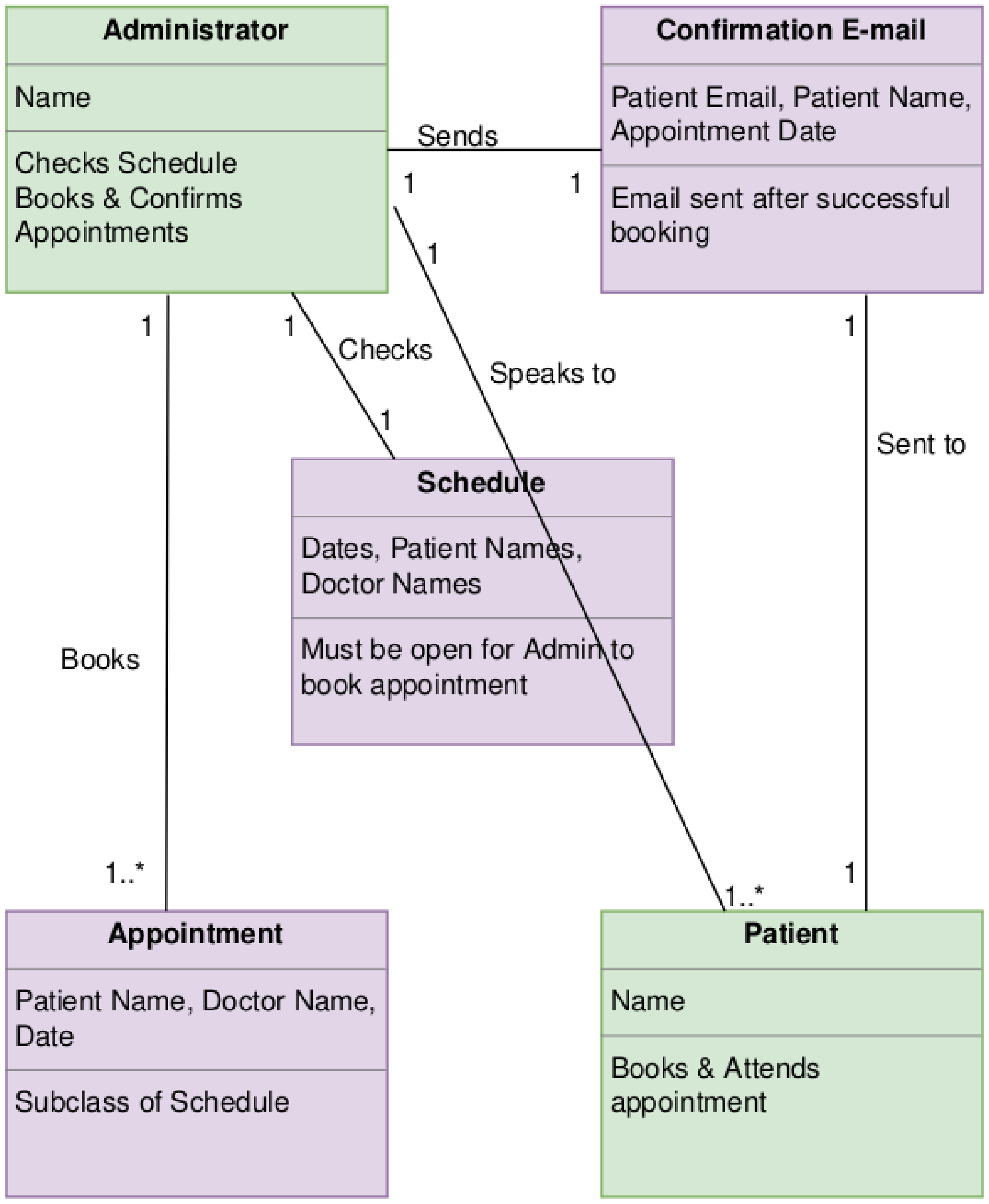

Figure 4: Domain Class Diagram

2.3 Sequence Diagram

Sequence Diagram for **Appointment Booking**

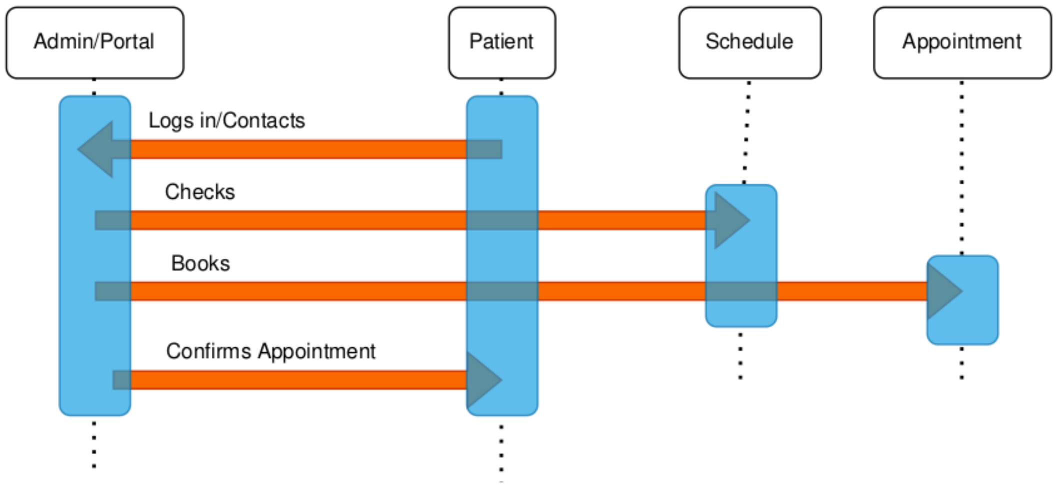

Figure 5: Sequence Diagram

2.4 State Transition Diagram

State Transition Diagram for Appointment Booking

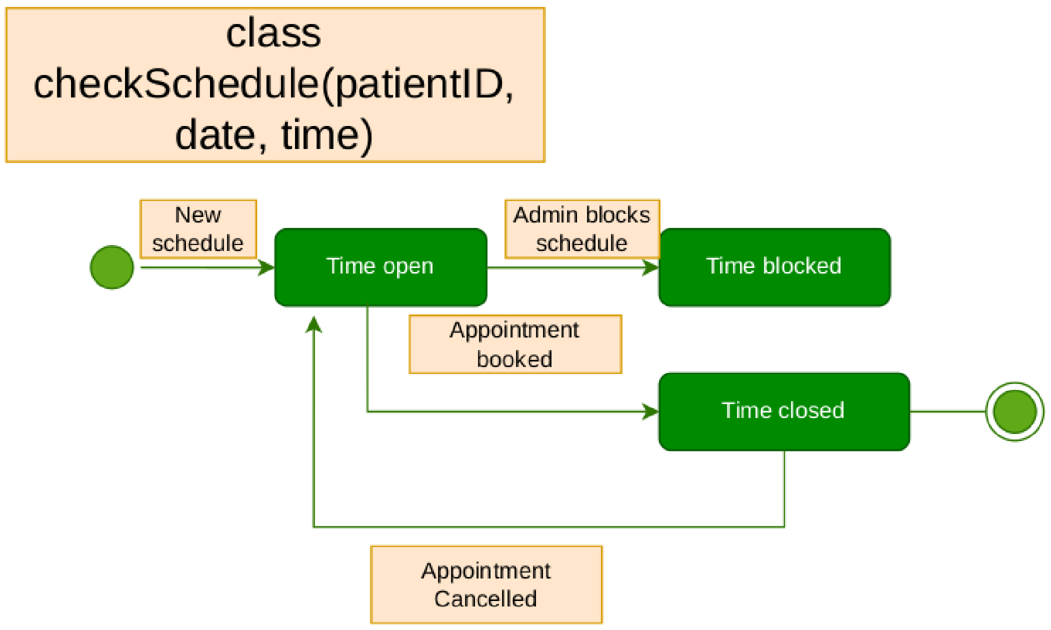

Figure 6: State Transition Diagram

2.5 Activity Diagram

Activity Diagram for Appointment Booking

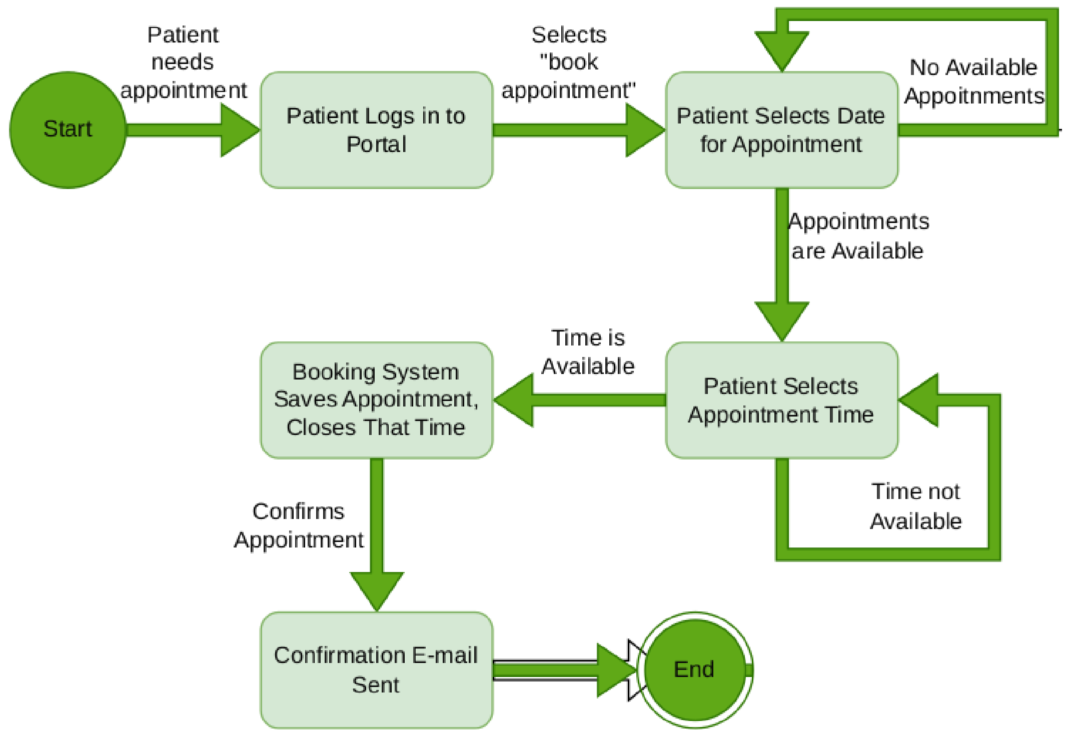
Start
Patient needs appointment
Patient Logs in to Portal
Selects "book appointment"
Patient Selects Date for Appointment
No Available Appoitnments
Appointments are Available
Patient Selects Appointment Time
Time is Available
Time not Available
Booking System Saves Appointment, Closes That Time
Confirms Appointment
Confirmation E-mail Sent
End

Figure 7: Activity Diagram