Time Healer

Clinic Management System

Team Members:

Name	ID
Menna Mohamed Islam	20221457144
Haneen Amr Abdelhameed	2106150
Tasneem Khaled Farouk	20221443994
Abdelrahman Waleed Mohamed	20221443348
Mamdouh Ahmed Amin	20221441410
Nour Eldeen Ahmed	2022447712

Table of Contents

Software Requirements Specification (SRS) for Clinic Management System

1.Introduction:

1.1 Purpose

The purpose of this document is to outline the software requirements for the development of a Clinic Management System to support the operations of a clinic.

1.2 Scope

The Clinic Management System will include features such as patient registration, appointment scheduling, medical record management and reporting. The system will be designed to accommodate the needs of a doctor, their staff, and the administrative personnel.

1.3 Methodology

The methodology we are using is the agile methodology, it is a project management approach that prioritizes cross-functional collaboration and continuous improvement. It divides projects into smaller phases and guides teams through cycles of planning, execution, and evaluation.

2.Overall Description:

2.1 Product Perspective

The Clinic Management System will operate as a standalone application, facilitating the day-to-day operations of a clinic with a focus on supporting the doctor and their respective workflows.

2.2 Product Features

- Staff Registration
- Patient Registration
- Appointment Scheduling
- Medical Record Management
- Reporting

2.3 User Classes and Characteristic

- **Administrator:**
  Manage user accounts and permissions.
- **Receptionist:**
  - Can register new patients.
  - Schedule appointments for patients.
  - Add appointments for patients.
- **Doctor:**
  - View his daily appointments.
  - Can write prescriptions.
  - Can view medical records.
  - Can update medical records.

3.Specific Requirements:

3.1 Functional Requirements

a. Use Cases

- **Administration Registration:**
  Administrators can register new users (receptionists, doctors) by entering their credentials and assigning appropriate roles.
  Users cannot login unless they have an account created by the administrator.
  They also can deactivate user accounts or update user roles as needed.
- **Staff Registration:**
  Login for clinic staff like receptionist and doctor.

- **Patient Registration:**
  Allows receptionists to enter patient details into the system.

- **Appointment Scheduling and Booking:**
  Enable receptionists to book and schedule appointments for patients.

- **Medical Record Management:**
  Permits doctors to view and update patient medical records.
  .

b. User Interfaces

- **Login Pages:**
  Username and password entry for secure access.

- **Patient Registration Form:**
  Fields for capturing patient's information.

- **Appointment Calendar:**
  Visual representation of patients' appointments.

- **Medical Record Interface:**
  User-friendly display of patient history and treatment plans.

c. Data Management

- **Database:**
  staff information, patients' records and appointments details stored and retrieved securely from Notepads.

3.2 Non-Functional Requirements

a. Performance

- **Response Time:**
  System should response within 2 seconds for most operations.

b. Reliability

- **Error Handling:**
  The system should provide clear and informative error messages and gracefully handle errors.

c. Security

- **User Authentication:**

- **Data Encryption:**
  Patient data should be encrypted during transmission and storage.

d. Usability

- **Intuitive Interface:**
  The system should have an easy-to-use interface for all user classes.

e. Testing Requirements

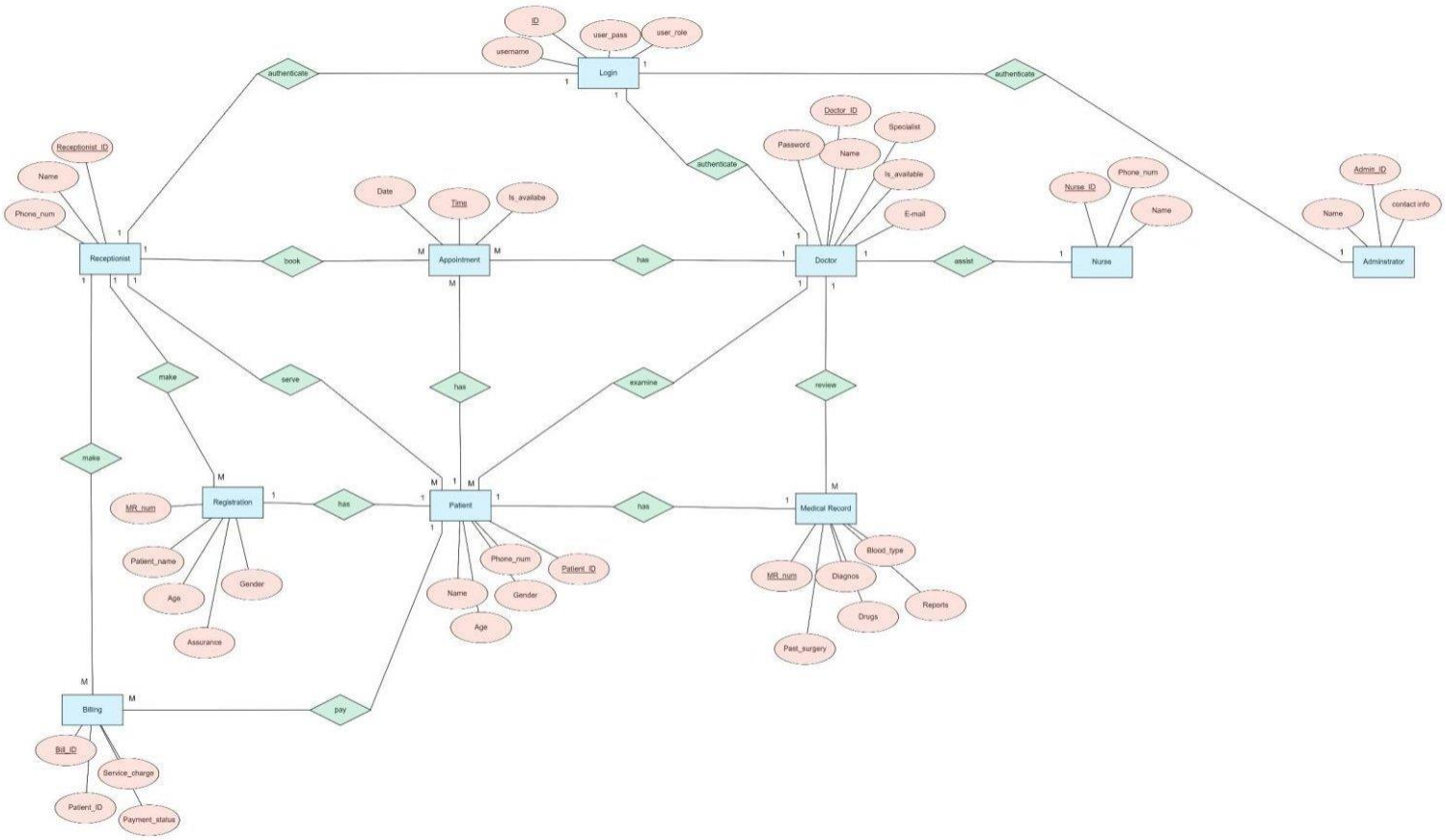
- **Testing plan:**
  Create a testing plan that includes performance testing, security testing and user acceptance testing.

4.UML diagrams

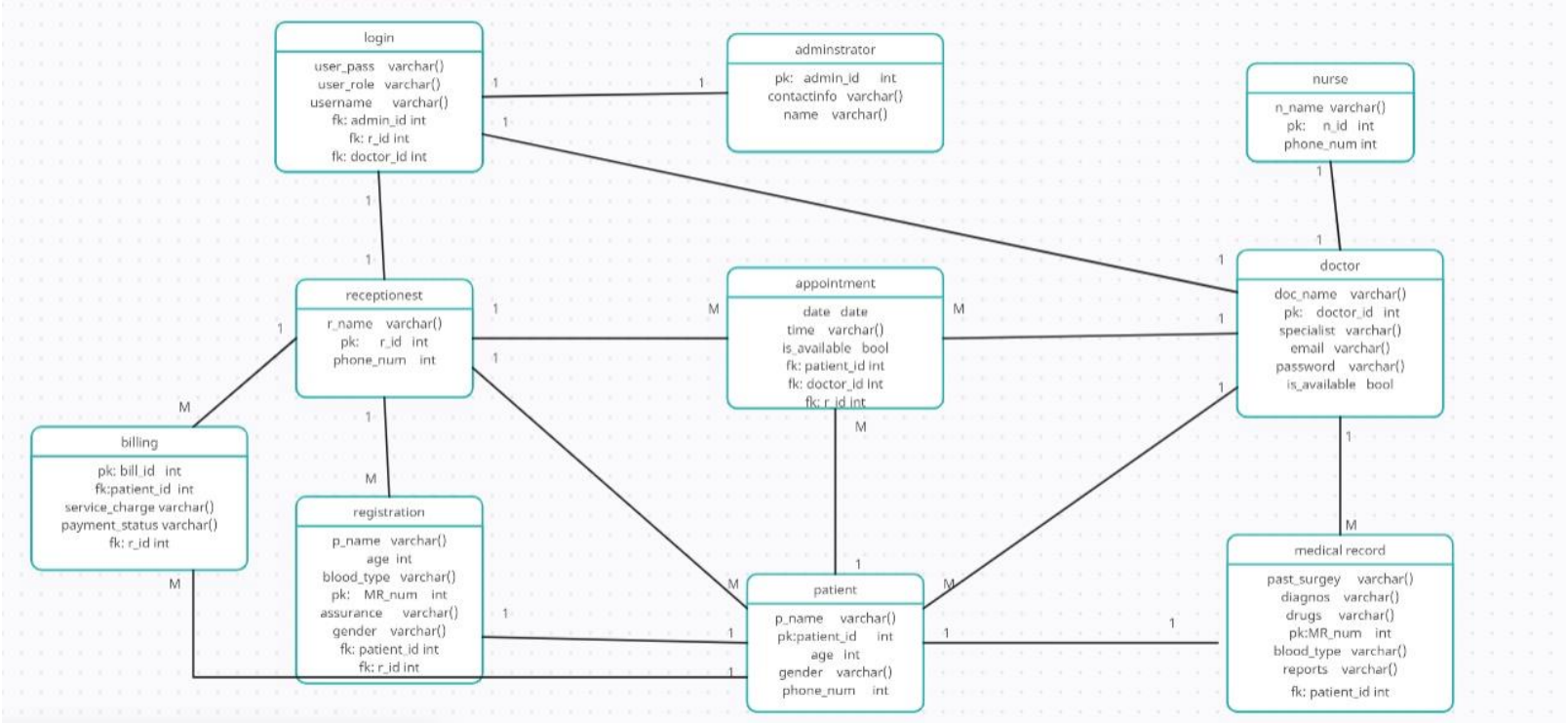
4.1 Entity Relationship Diagram:

- An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how "entities" such as people, objects or concepts relate to each other within a system. An ERD contains different symbols and connectors that visualize two important information: The major entities within the system scope**,** and the inter-relationships among these entities**.**

a.

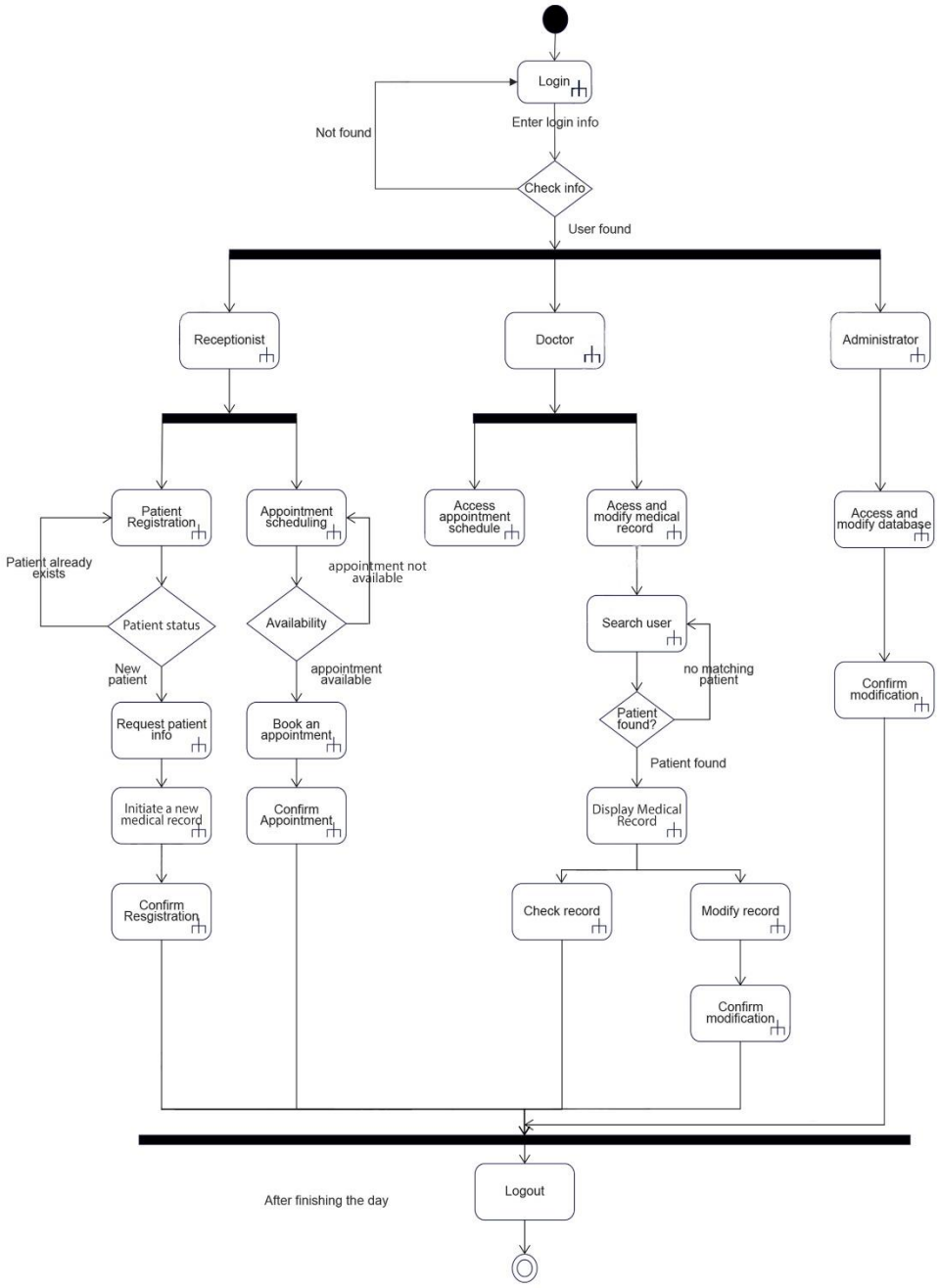


b.


login
user_pass varchar()
user_role varchar()
username varchar()
fk: admin_id int
fk: r_id int
fk: doctor_id int
adminstrator
pk: admin_id int
contactinfo varchar()
name varchar()
nurse
n_name varchar()
pk: n_id int
phone_num int
receptionest
r_name varchar()
pk: r_id int
phone_num int
appointment
date date
time varchar()
is_available bool
fk: patient_id int
fk: doctor_id int
fk: r_id int
doctor
doc_name varchar()
pk: doctor_id int
specialist varchar()
email varchar()
password varchar()
is_available bool
billing
pk: bill_id int
fk:patient_id int
service_charge varchar()
payment_status varchar()
fk: r_id int
registration
p_name varchar()
age int
blood_type varchar()
pk: MR_num int
assurance varchar()
gender varchar()
fk: patient_id int
fk: r_id int
patient
p_name varchar()
pk:patient_id int
age int
gender varchar()
phone_num int
medical record
past_surgey varchar()
diagnos varchar()
drugs varchar()
pk:MR_num int
blood_type varchar()
reports varchar()
fk: patient_id int

4.2 Activity Diagram:

- A UML activity diagram helps to visualize a certain use case at a more detailed level. It is a behavioral diagram that illustrates the flow of activities through a system. They are like a flowchart, but with more specific symbols and notations. The purpose of an activity diagram is to model the behavior of a system or process in a clear and structured way, making it easier to understand and analyze.

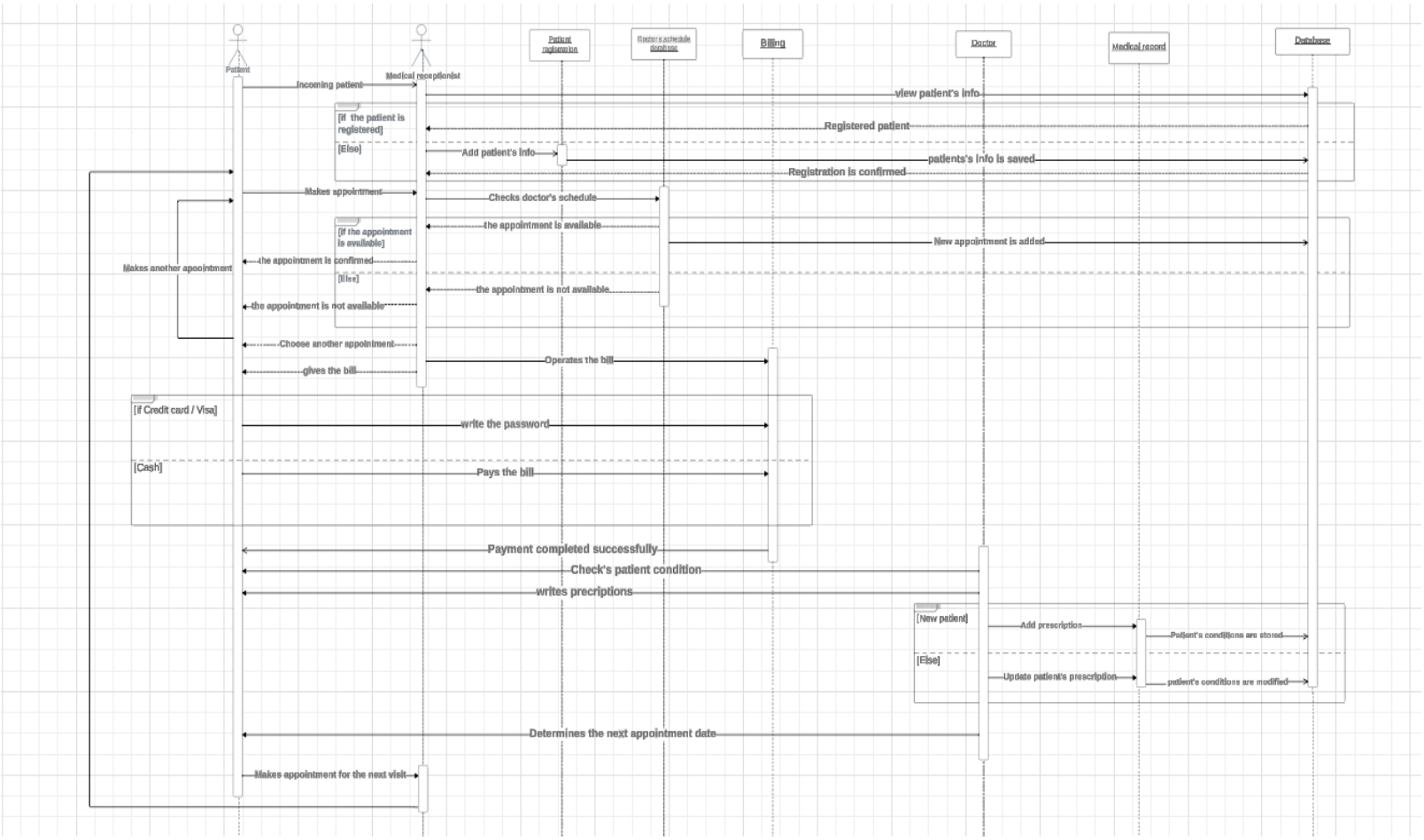


4.3 Sequence Diagram:

- A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together.

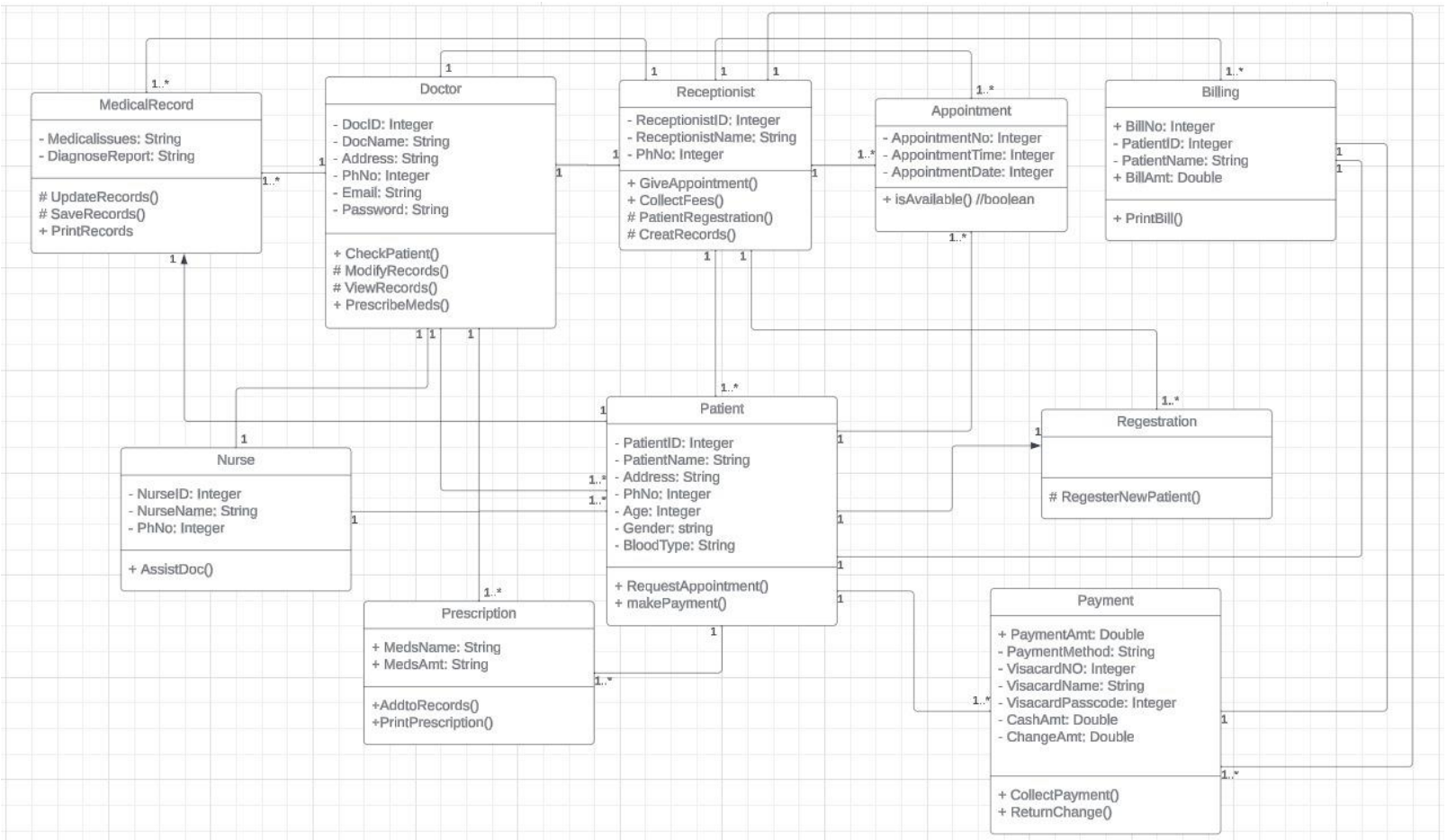
Sequence Diagrams captures:

- the interaction that takes place in a collaboration that either realizes a use case or an operation (instance diagrams or generic diagrams)
- high-level interactions between user of the system and the system, between the system and other systems, or between subsystems (sometimes known as system sequence diagrams)



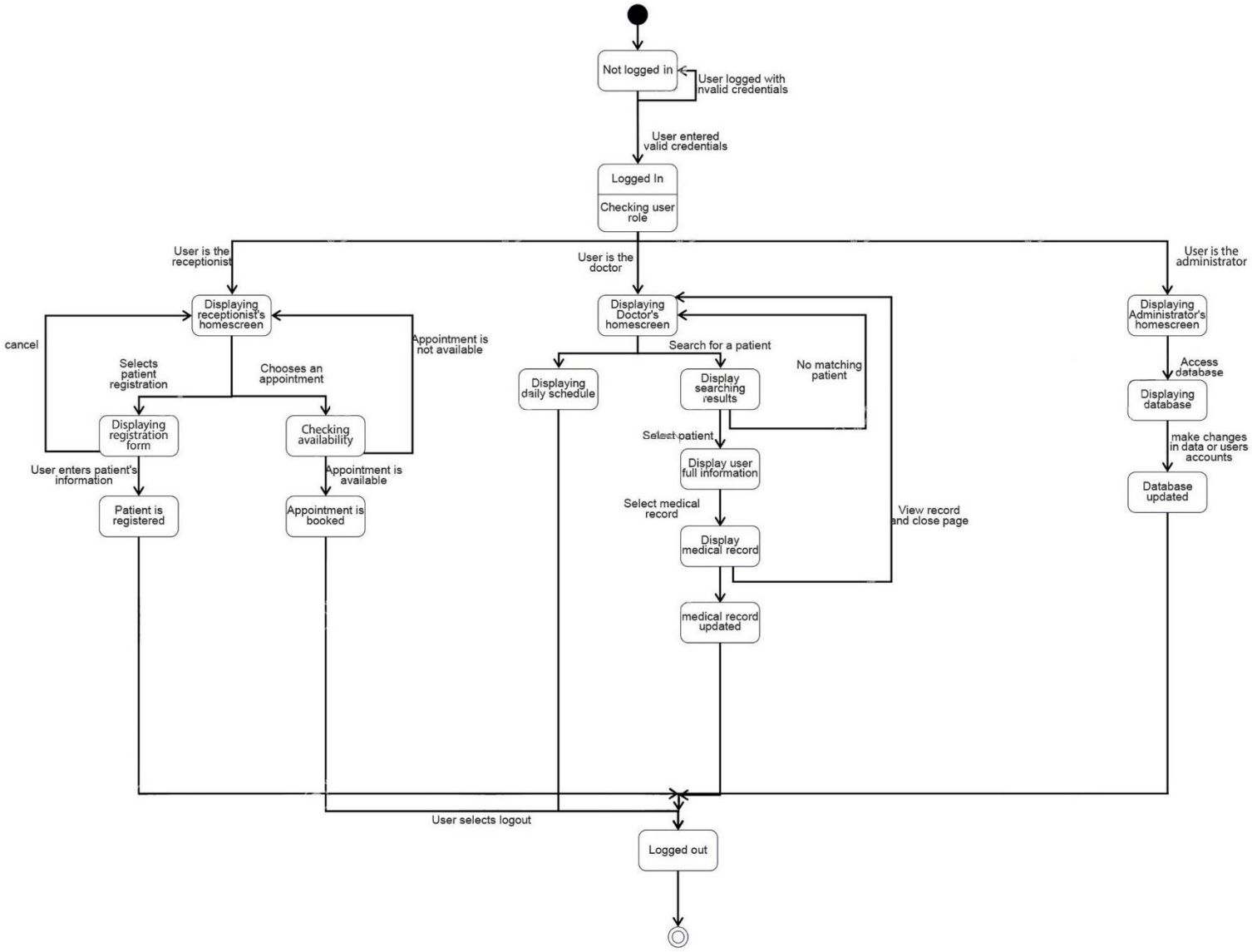
4.4 Class Diagram:

- In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.



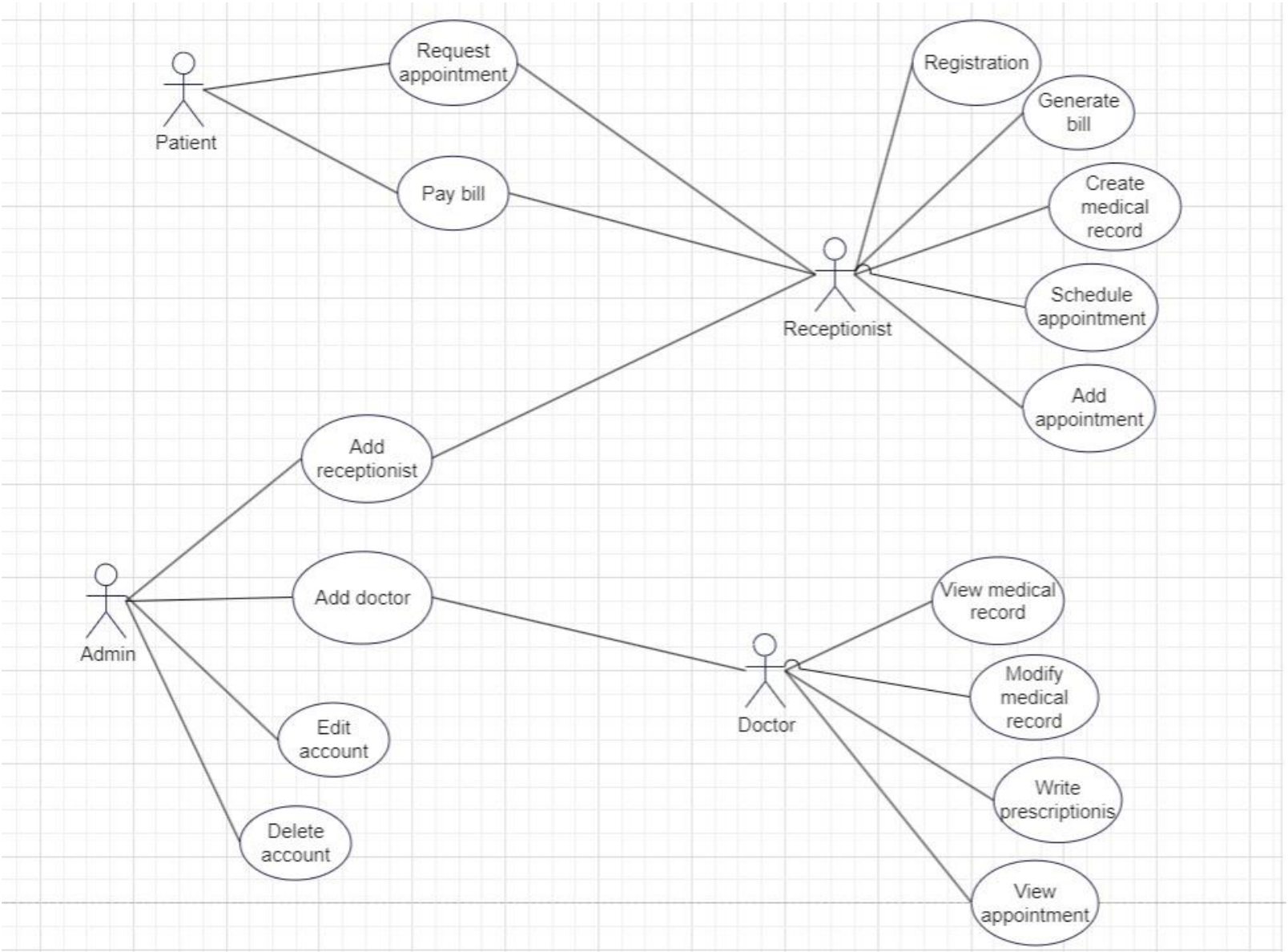
4.5 State Diagram:

- A state machine diagram models the behavior of a single object, specifying the sequence of events that an object goes through during its lifetime in response to events. A state diagram consists of states, transitions, events, and activities. You use state diagrams to illustrate the dynamic view of a system.



4.6 Use case Diagram:

- A use case diagram is the primary form of software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how).
  A key concept of use case modeling is that it helps us design a system from the end-user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.



5. Case testing:

Test case type	Test case	Precondition	Test step	Test data	Expected Result	Actual Results	Test case evaluation
Login_01	Verification of login with valid username and password	Account must be created with its credentials and stored in database by administrator first.	a. Type username b. Type password c. press on the login button	a. Valid username b. Valid password c. button clicked	Login successful	Login successful	Successful
Login_02	Verification of login with valid username and invalid password	Account must be created with its credentials and stored in database by administrator first.	a. Type username b. Type password c. press on the login button	a. Valid username b. Invalid password c. button clicked	“Invalid Credentials “ Message appears to the user	“Invalid Credentials “ Message appears to the user	Successful
Login_03	Verification of login with invalid username and invalid password	Account must be created with its credentials and stored in database by administrator first.	a. Type username b. Type password c. press on the login button	a. Invalid username b. Valid password c. button clicked	“Invalid Credentials “ Message appears to the user	“Invalid Credentials “ Message appears to the user	Successful
Login_04	Verification of login with empty username and valid or invalid password	Account must be created with its credentials and stored in database by administrator first.	a. Type username b. Type password c. press on the login button	a. Username -empty b. Valid or Invalid password c. button clicked	“Please complete all fields “ Message appears to the user	“Please complete all fields “ Message appears to the user	Successful
Login_05	Verification of login with valid or invalid username and empty password	Account must be created with its credentials and stored in database by administrator first.	a. Type username b. Type password c. press on the login button	a. Valid or Invalid username b. Password - empty c. button clicked	“Please complete all fields “ Message appears to the user	“Please complete all fields “ Message appears to the user	Successful