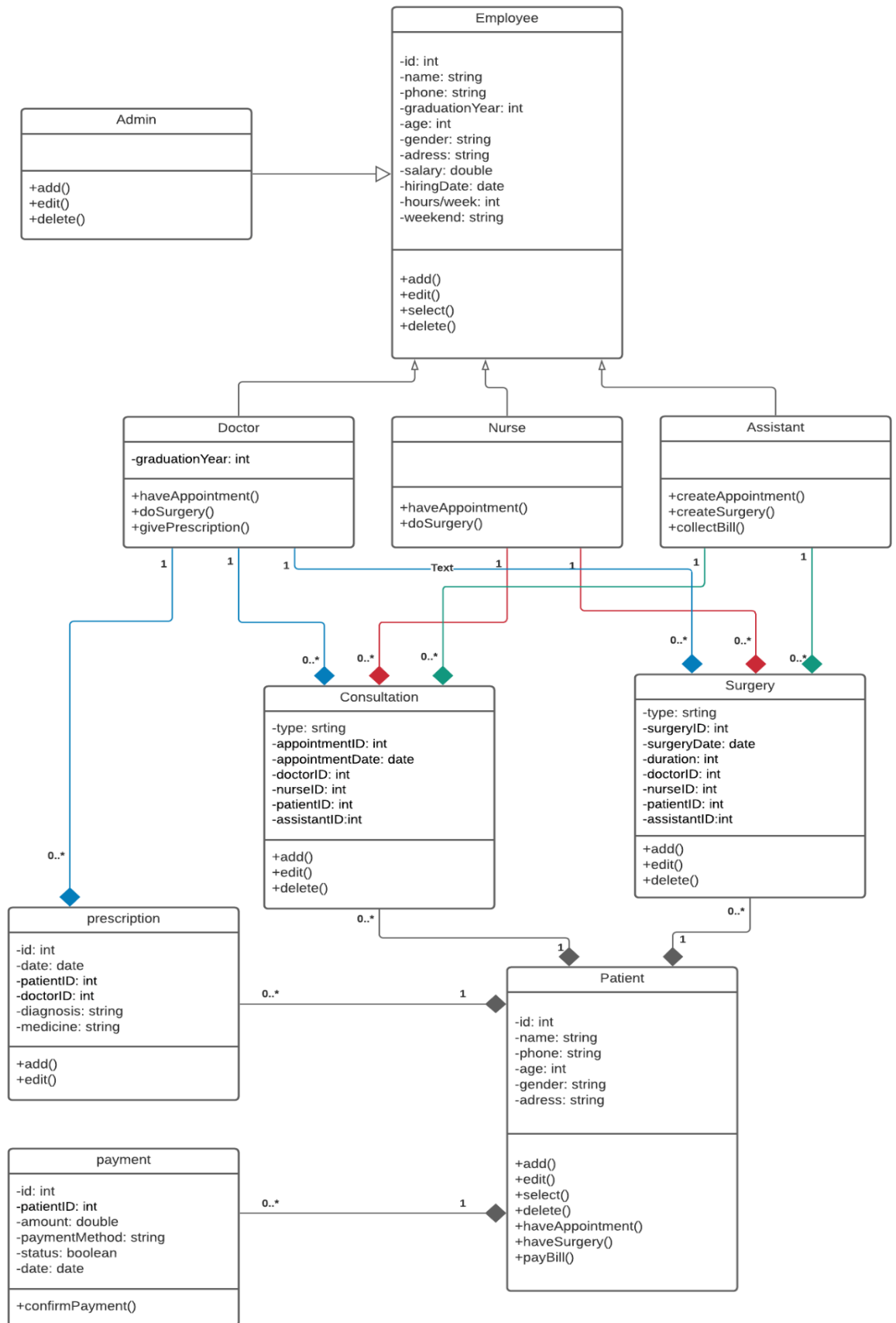# Clinic System Analysis

1- Description:

An application used in a clinic by the assistant to manage the appointments and surgeries of several doctors. The nurses are there to help the doctors. The assistant starts by searching for patient information in the database by using patient ID. If the patient has no profile, the assistant creates a new one then advances to make an appointment for the patient. The assistant opens the consultation/surgery page and a detailed calendar is viewed with all reserved appointments and makes the patient choose the suitable date and time. Then, the assistant creates a new appointment in the consultation/surgery database with the patient desired details. At the same page the assistant can edit all appointments and the changes affect the database. When it is time for the patient appointment, the assistant notifies the doctor with patient information then the doctor accesses the prescription UI and searches for patient medical history by patient ID. The patient enters the consultation/surgery room and the doctor starts the process then writes a new prescription for the patient. The patient exits the room and the assistant accesses the payment database through payment UI to check the patient's bill then asks the patient whether he pays in cash or by credit card. If he wants to pay by credit card the assistant verifies the card information and makes the transaction. At the end of the payment process, the assistant makes sure to confirm payment details in the database and give the patient a receipt.
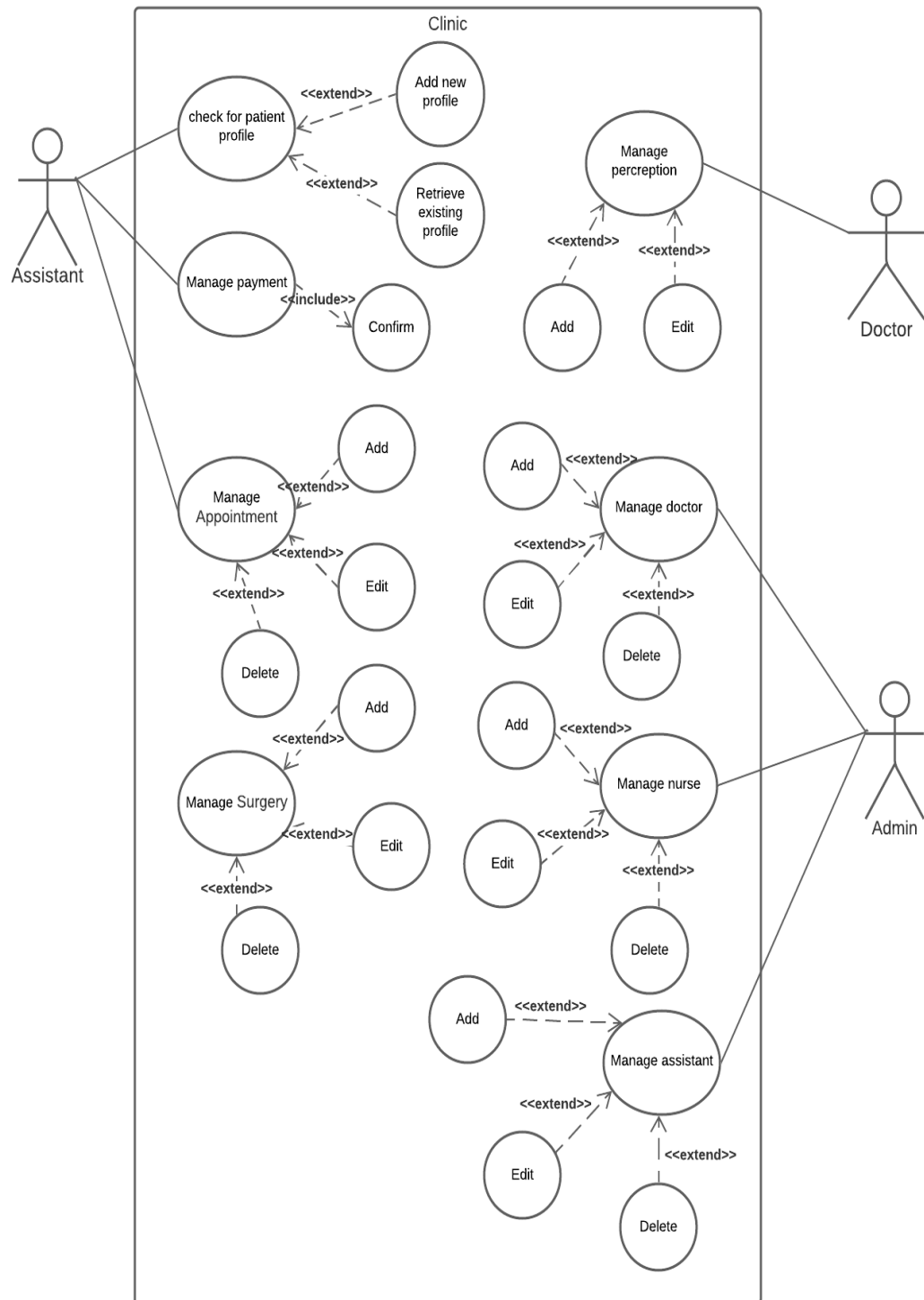
2- Actors:

- Admin: is responsible for managing the system data such as add or remove or edit the details of the staff.

- Assistant: the main actor in the system that manages reservations.

- Doctor: is responsible for consultation and surgery.

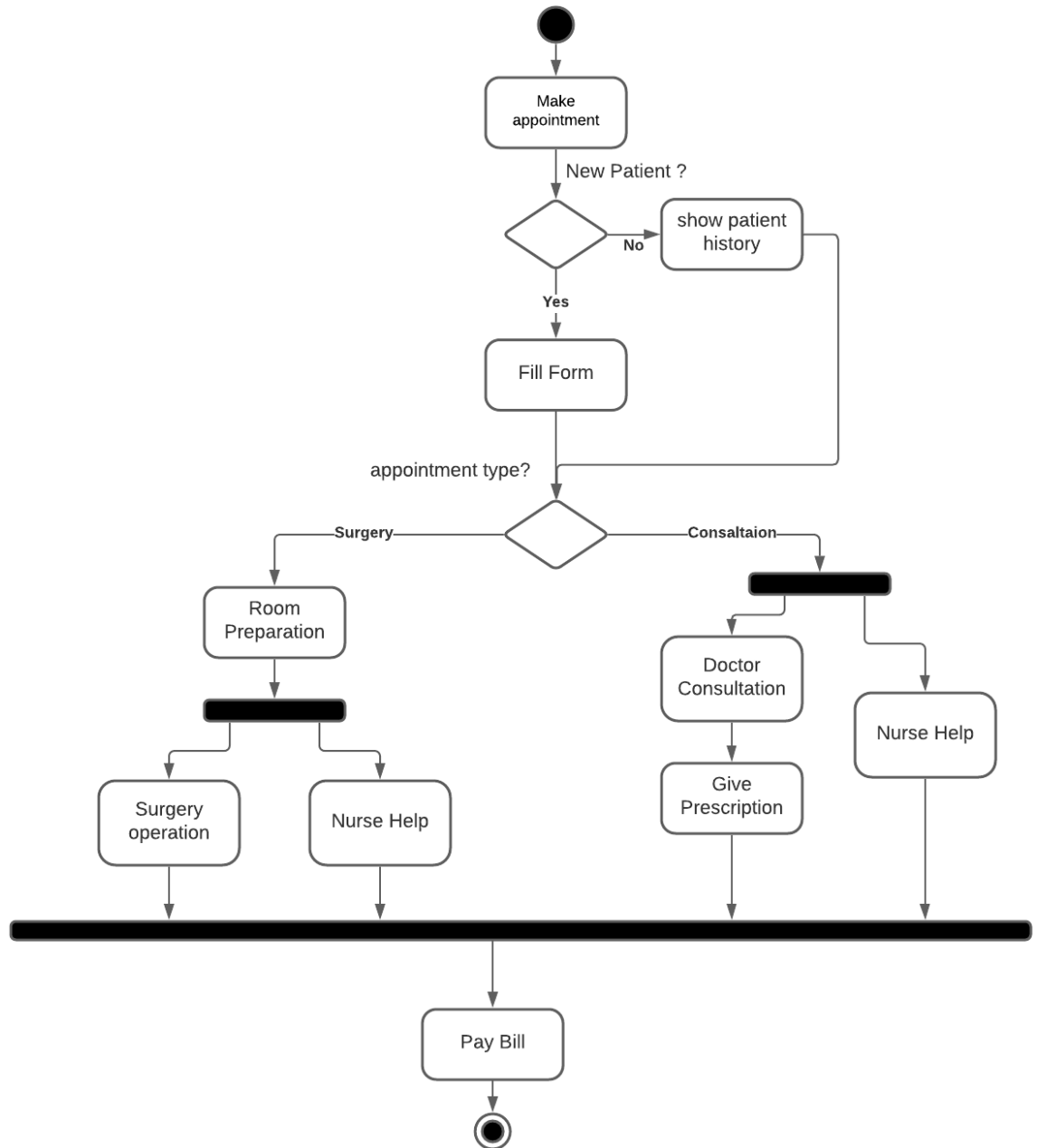- Patient: receives consultation and surgery.
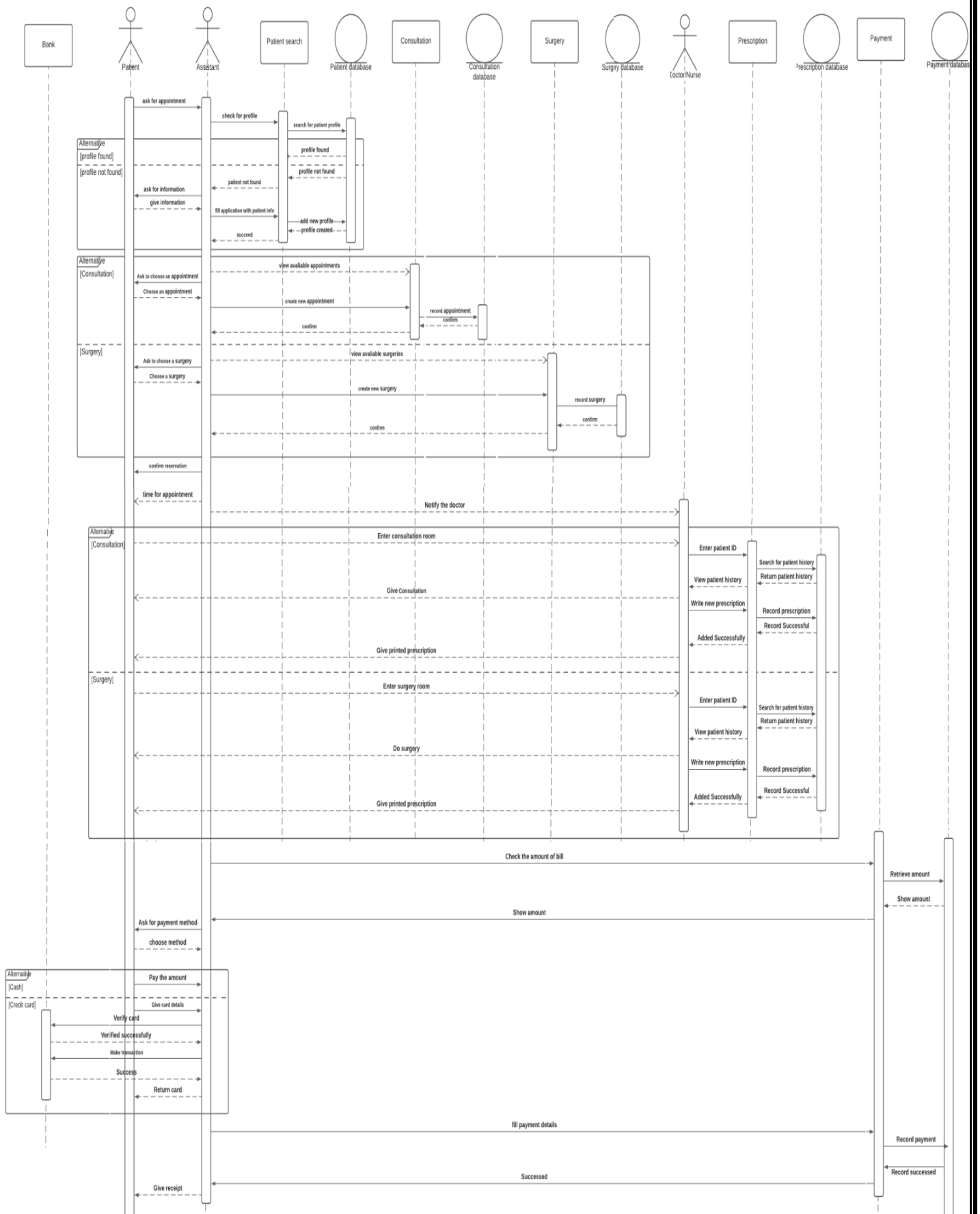
3-UML Diagram:

- ## Class Diagram

Admin

+add()
+edit()
+delete()

Employee

-id: int
-name: string
-phone: string
-graduationYear: int
-age: int
-gender: string
-adress: string
-salary: double
-hiringDate: date
-hours/week: int
-weekend: string

+add()
+edit()
+select()
+delete()

Doctor

-graduationYear: int

+haveAppointment()
+doSurgery()
+givePrescription()

Nurse

+haveAppointment()
+doSurgery()

Assistant

+createAppointment()
+createSurgery()
+collectBill()

1   1   1   Text   1   1   1   1

0..*   0..*   0..*   0..*   0..*   0..*

Consultation

-type: srting
-appointmentID: int
-appointmentDate: date
-doctorID: int
-nurseID: int
-patientID: int
-assistantID:int

+add()
+edit()
+delete()

Surgery

-type: srting
-surgeryID: int
-surgeryDate: date
-duration: int
-doctorID: int
-nurseID: int
-patientID: int
-assistantID:int

+add()
+edit()
+delete()

0..*

prescription

-id: int
-date: date
-patientID: int
-doctorID: int
-diagnosis: string
-medicine: string

+add()
+edit()

0..*   1   0..*   1

Patient

-id: int
-name: string
-phone: string
-age: int
-gender: string
-adress: string

+add()
+edit()
+select()
+delete()
+haveAppointment()
+haveSurgery()
+payBill()

payment

-id: int
-patientID: int
-amount: double
-paymentMethod: string
-status: boolean
-date: date

+confirmPayment()

0..*   1

● Use Case Diagram

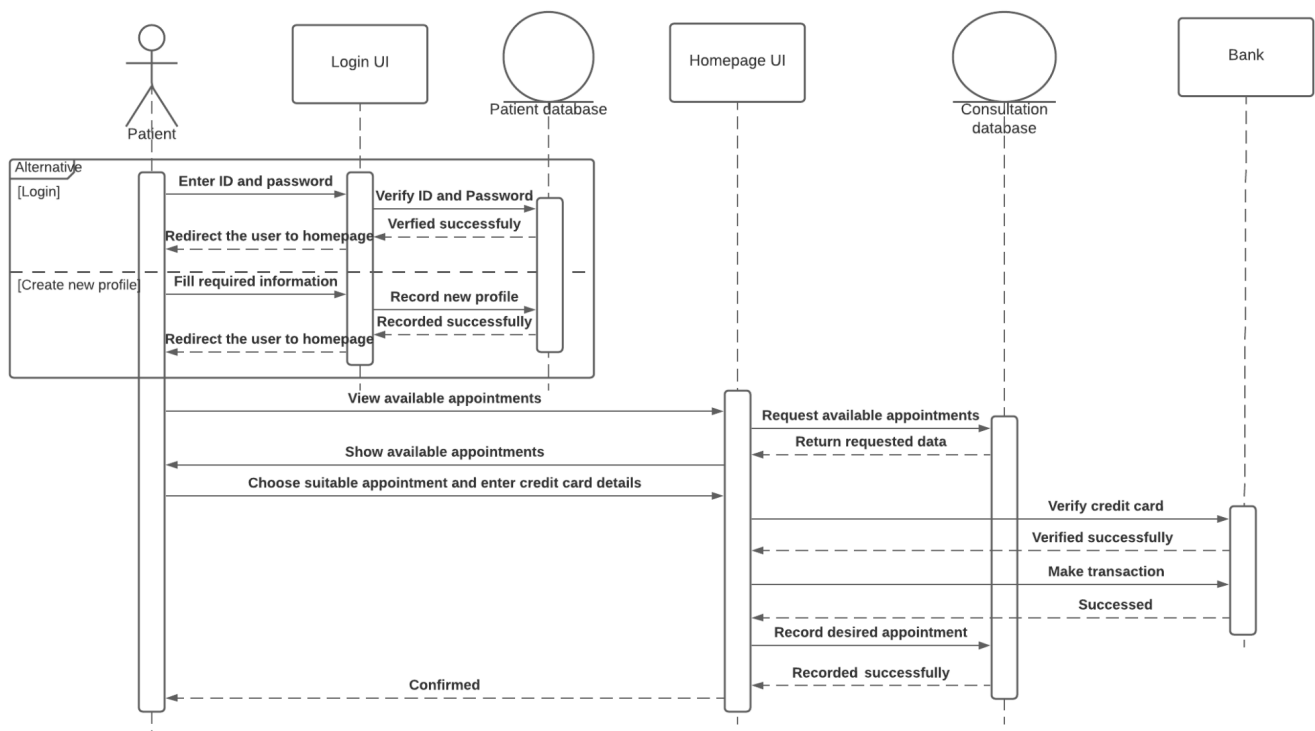● **Activity Diagram**

● Sequence Diagram

4- Development:

Providing the patient with an online application that allows him to make an appointment and confirm the registration without the need to go to the clinic for direct reservation.

The patient is required to login with ID and password or create a new profile. At the homepage of the application the patient can view available appointments and choose a suitable one. For assurance reasons, the patient is required to pay a part of the consultation bill when reserving the appointment. The patient enters the credit card details and the application connects to the bank to verify the credit card and make the transaction. Finally the reservation is recorded in the database and a confirmation message is shown.

Sequence Diagram

5- Risk Management:

Risk	Risk type	Probability	Effects
Lose connection with the database during using the application duo to server failure.	Technology	Moderate	Catastrophic
Low skilled staff may cause data loss or conflict.	People	Low	Serious
The database size may become huge and that makes the application slow.	Technology	High	Serious
The time required to develop the software is underestimated.	Estimation	Moderate	Serious
The size of the software is underestimated.	Estimation	Low	Tolerable