Helwan University
Faculty of Computers and Artificial Intelligence
Software Engineering Program

MediBooki Healthcare and Pneumonia Detection System

Team Members: -

1. **Mohamed Ali GadAlhak Hassan** **ID: 20201871**
2. **Assem Mohamed AhmedSamy Mohamed** **ID: 20201836**
3. **Pola Osama Aziz** **ID: 20201823**
4. **Ahmed Salah Mohammed** **ID: 20201803**
5. **Ahmed Waleed Abdullah Ahmed** **ID: 20201815**
6. **Eslam Sabry Hassan** **ID: 20201849**

Supervised by: -

1st Assc. Prof. Mohamed Marie.

2nd Dr. Ahmed El Sayed.

June 2023

Table Of Contents

Table Of Figures

Abstract

We discuss medical booking system and its social benefits to fulfill patients' needs by connecting them to the appropriate doctor and we discuss the technical requirements for booking doctors in the easiest way. the presentation is not totally

completed, but it aims to give an idea of the system-level issues to be considered for real applications. The technology in this area is rapidly developing, and without doubt we will evidence emergence of these applications in the coming years in the market.

Chapter 1: Introduction

1.1. Overview

Our project is a medical information system for hospital which helps patients in their medical needs such as booking doctors and providing pneumonia prediction service for doctors.

1.2. Objectives

Project's objective is building a powerful system which provides services to patients to fulfill their requirements digitally such as booking doctors and buying medical supplies online using medical insurance; and provides services to admins such as managing users digitally without using hard copy papers by building ease-use dashboard.

- Now we can say that the most affected by the current system in hospitals are patients who find it difficult to seek medical advice, so they turn to search via the Internet. In a study conducted, it proved that websites and applications for examining symptoms are accurate about 34% of the time, while doctors, when given the same information, diagnosing the condition correctly 72% of the time.
- Doctors also, because their task is made difficult in the current system because there are no ways to facilitate the matter of meeting patients, for example, or knowing their medical history in asking each patient. Those related to the health system in Egypt, so by making it easier for them, they are more attracted to work in Egypt.

1.3. Purpose

- managing patients and their related information.
- Improving patients care by helping them in booking doctors easily and digitally.
- helping radiology doctors in detecting pneumonia using service.
- Helping doctors in managing their appointments.
- Helping admins in accessing users' information.
- Improving efficiency via taken care of processes automatically.
- Increasing data security & retrieve-ability.
- Accounting, laboratory, and pharmacy management.
- Buying medical supplies from pharmacy page online using electronic payment system allowing them to use medical insurance.
- Serving patients from multiple regions using multitenancy (Software as a Service).

1.4. Scope

Mention the scope or range of the project. Scope means the work involved to finish the project. For example planning, designing, coding, testing and documentation.

1.5. General constraints

Mention things that hindered (prevented) your project from being finished on time. This could be due to time constraint, the scope was not clear, collecting raw data for simulation was not easy to access.

Chapter 2: Project "Planning and analysis"

2.1. Project planning

In this section we will know everything about the project and study its aspects to understand it very well to start building the system.

2.1.1. Feasibility Study

A feasibility study is conducted to find out whether the proposed system is possible, affordable, and acceptable for organization. The financial, political, social and time constraints must be considered during this study.

- Possible: to build it with the given technology and resources
- Affordable: given the time and cost constraints of the organization
- Acceptable: for use by the eventual users of the system.

2.1.1.1. Technical Feasibility

The primary technical requirement includes the availability of a good version of operating system installed in the network. To develop programs, any good Integrated Development Environment is needed, which can be easily acquired after deciding. Reliability, access, power and data security are also available.

- Hardware Requirements:

  ➔ Computer Systems: 3 (Available)

  ➔ Processor: Core i3 Processor (minimum)

  ➔ RAM: Minimum 8 GB. (1 GB extra RAM is required to use Android emulator and Vs code)

  ➔ Disk Space: Using an SSD would be a wise decision, but 256GB SSD can be a good choice.

  ➔ Works on graphic card 4GB to 8GB

- Software Requirements:

1. Web apps can be developed using a number of different alternative languages and IDEs.

➔ Back-End

A. Xampp local host and Vs code “IDE”

B. Php V 7.4 “language”

➔ Front-End

A. HTML, CSS “tools”

B. Local host and Vs code “IDE”

C. Angular “frame work”

2. Android or IOS apps can be developed using a number of different alternative languages and IDEs.

➔Java Development Kit (JDK) and Android studio “IDE”

➔Git.

➔Dart “language”

➔ Flutter “frame work”

AI feature:

➔Anaconda environment

2.1.1.2. Economic Feasibility

Whether the MediBooki is cost effective or not? The benefits in the form of reduced cost?

MediBooki is economically Feasible. As the hardware cost on the project is low. Similarly. it’s cost is also under the budget. Moreover, some of the technical requirements are already available and some can be obtained by using a reasonable amount and effort.

2.1.1.3. Operational Feasibility

MediBooki is operationally feasible. it provides the necessary information to the user as how to enter the information, how to register, selecting the interests, giving permissions to the apps. Some prior knowledge is required for the management to go through the various operations. But for the user basic knowledge of computers is enough

2.1.2. Gantt Chart

dfsdfsfsdfsdf

2.2. Analysis and Limitation of existing system

- At the beginning of our study of the project, we found that the current manual medical system is difficult for the patient these days, so we decided to try to make it easier for patients and doctors as well by making an electronic system that would be an intermediary between them and also between them and the hospital. **We found the following:**
- There are many medical systems, but we did not find one of them that contains all the needs of the three categories patients, doctors and the hospital.
- The patient has to go to the hospital to book his doctor, and he finds it difficult because he sits and waits for a lot of time.
- The patient is forced to go to the hospital to book his doctor, and he finds it difficult because he sits and waits for a lot of time and also book the work of x-rays and medical tests and also receive them.
- We also found that the proportion of patients with pneumonia affects about 15% of children under the age of five around the world, according to the World Health Organization. https://www.who.int/ar/news-room/fact-sheets/detail/pneumonia.
- Under the spread of the Corona virus, the patient, if he suspects that he has the disease, tends to make Lung x-ray, where pneumonia appears.
- We found that these days, the state is working to reduce the circulation of currencies and dealing with them and towards electronic payment.
- We also found it difficult to organize between doctor's and patients' appointments.
- We also found that the doctor does not see the patient’s medical history, so the doctor is forced to ask each patient about his medical history and his details, but the medical

history is not recorded in order to be preserved if the same patient goes again in follow-up.

- We also found administrative and accounting problems in hospitals.
- We also found that there is a difficulty in dispensing medicines to patients and that they do not reach those who deserve them.

2.3. Need for the new system

To overcome problems found in existing system as mentioned above; then we will build new system that contains the following points: -

- Optimize, manage, and track personal and financial hospital resources.
- **No** chance for **duplicated** patient files and data.
- Manage **lab tests**, and **consultation** of different specialties like cardiology and more.
- Build actionable treatment plans with reminders and targets for patients, staff, and doctors to enhance adherence.
- Manage appointment time slots and timings by lab, clinic, and doctor.
- Access to your portal through our mobile apps.
- You will find support in how to use our website in **Chatbot**.
- Fast **detection** of pneumonia disease.
- Cost effective and easily manageable.
- Easy access to patient data with correct patient history.
- Support Multilingual.
- Support multitenancy.
- Billing and Insurance Management.

2.4. Analysis of the new system

In this section we know who are stakeholders and collect all requirements they want in new system.

2.4.1. Identification of key stakeholders and End-Users

In this chapter we identify all persons who have an interest in the successful implementation of the system either they are inside or outside the organization. Stakeholders are consisting of 3 types: -

a. Internal-Operational:

persons within the organization and regularly interact with the system.

- **Doctor** is the person who examines and diagnoses the patient's condition, determines the optimal treatment, and follows up on his condition and treatment results. He also performs first aid for patients and injured people. He also trains and directs instructions to the nursing staff.
- **Analytics specialist** is the person who receives various samples of blood and other body fluids, marking and sorting and classifying blood samples. he also organizes and stores all chemicals, liquids and compressed gases in accordance with safety instructions. Designs and implements laboratory tests according to standard procedures and takes explanatory notes on the results. He also presents the results of tests and medical examinations to patients and providing specialized doctors with the necessary knowledge for treatment, taking into account the confidentiality of medical laboratory information related to patients.

- **Radiology doctor** is the person who makes sure that the x-ray examination is requested from the treating doctor and determines what part to be photographed and what conditions are required. He also informs patients or department nurses of all the necessary instructions for any examination, such as attending the patient without breakfast or taking a specific tablet. He also adjusts the x-ray tube and determines the x-ray package and the necessary imaging factors. For each patient's required situation, he chooses the appropriate film size and quality for each examination, puts the patient's letter and identification number on the clipboard, and prints the patient's name and examination date on the film, if possible, and his technical number, or write that on the x-ray film. He also develops the films from him or from other radiology assistants according to the order of the work schedule in the radiology department, prepares the chemicals for acidification and daily and periodic cleaning of the acidifying device, and then delivers the x-rays to the patient.

- **Accountant** is the person who Follow up the financial procedures of patients, whether cash payment patients or receivable patients, collecting them, settling the fund, and following up on closing outstanding bills. He also restricts cash payment patients' bills on the system, collects cash from cash payment patients, restricts health insurance patients' bills on the system, follows up on cash payment patients' bills, and follows up pending bills with doctors and administration. Also, organizes the patient's file upon discharge from the hospital and completes financial exit procedures. Also send invoices to public accounting and accounting insurance companies. Also participate in the annual or periodic inventory work in the hospital.

- **Pharmacist** is the person who Dispensing the patient's medication via the doctor's prescription only, as well as dispensing the medication through the health insurance. Educating the patient about the side effects of the drug, writing insurance forms, and communicating with insurance companies to verify that patients get the drugs they want. Also answer patients' inquiries about prescriptions for medicines. Also, organize the pharmacy and its medications in an effective and organized manner, and continuously monitor the expiration dates of the medications. Also, follow-up invoices for purchases and sales of medicines and medical supplies as well. Follow the correct means of storing medicines and try as much as possible to prevent problems of poor storage.

- **Administrator** He is the person responsible for the administrative things in the hospital such as organizing departments, appointing employees, adding departments and things like that.

b. External-Operational

persons outside the organization and regularly interact with the system.

- **Patients** They are the clients who come to the hospital in order to receive the services they request.
- **Pharmaceutical suppliers** They are the people who supply all medicines to the hospital pharmacy as well as all medical supplies.

c. Internal-Executive

persons within the organization and don't directly interact, but use the information or have financial interest.

- **hospital managers** They are the people who have financial interests and who take the highest decisions in the management of the hospital and follow up on all activities in the hospital.

2.4.2. User requirements

Requirements that system should provide to user.

a. User Interface

- Improved controls and displays interaction,
- Better user understanding of the device's status and operation,
- Better user understanding of a patient's current medical condition,
- More effective alarm signals management,
- Easier device maintenance and repair,
- Reduced user reliance on user manuals,
- Reduced need for user training and retraining,
- Reduced risk of use error,
- Reduced risk of adverse events, and
- Reduced risk of product recalls

b. User Population

- With population health management platforms, healthcare organizations and providers can better create patient care plans, track patient outcomes, and achieve measurable improvements in the health outcomes of a community.

c. Environments of Use

- State the environments in which the system is intended to be used – e.g. home, hospital, ambulance, factory and the great outdoors. Consider the difference between a clinical setting and a home setting with children, pets, etc.

2.4.3. System Requirements

System need Cloud-based to

- Allows recording of medical data for eventual use
- Allows data retrieval in real time,
- Allows patients to access and monitor their medical data,
- Does not allow patients to provide their health conditions,
- Supports data sharing only within the same hospital,

2.4.4. Domain Requirements

Defining the requirements for all foreseeable applications to be developed in the software product line.

Our system contains many integrated sub-systems such as pharmacy management system, reservation management system, accounting management systems, etc.

- Accuracy and reliability of diagnosis, treatment, monitoring, etc.
- Usability and accessibility for different users such as doctors, nurses, patients, administrators, etc.
- Performance and scalability for handling large volumes of data and transactions Adaptability and flexibility for changing needs and requirements.

2.4.5. Functional Requirements

They describe what the system/software must do; functionality or services (a function is a useful capability provided by one or more components of a system). Therefore, they specify an action that a system must be able to perform.

- **Login Function: -**
  - **Function:**

    Login
  - **Actors:**

    Administrator, Hospital Manager, Doctor, Pharmacist, Analytics Specialist, Radiology doctor, Accountant, Patient
  - **Priority:**

High

- **Description:**
  When the actor login, he can manage the system based on his roles.
- **Inputs:**
  The actor should write the email and password in the right way and correct data for a successful login
- **Outputs:**
  The actor can manage many functions based on roles that assign to him
- **Requirements:**
  Write the right data to can login
- **Pre-condition:**
  The actor already has an account
- **Post-condition:**
  The actor entered his account successfully

➢ View Profile: -

- **Function:**

  View profile
- **Actors:**

  Administrator, Hospital Manager, Doctor, Pharmacist, Analytics Specialist, Radiology doctor, Accountant, Patient
- **Priority:**

  Medium
- **Description:**
  An actor profile is a collection of settings and information associated with a user. It contains critical information that is used to identify an individual, such as their name, age, profile picture, email, and password
- **Inputs:**
  His account must be verified on the site after logging in.
- **Outputs:**
  View personal information successfully.
- **Requirements:**
  Actor information must be added by logging in to the site before viewing his profile.
- **Pre-condition:**
  The actor must be logged in to the system and has permission.
- **Post-condition:**
  The actor can view this profile.

➢ Edit Profile: -

- **Function:**

  Edit profile
- **Actors:**

  Administrator, Hospital Manager, Doctor, Pharmacist, Analytics Specialist, Radiology doctor, Accountant, Patient
- **Priority:**

  High
- **Description:**
  An actor profile is a collection of settings and information associated with a user. It contains critical information that is used to identify an individual, such as their name, age, profile picture, email, and password, and can edit this information based on this role
- **Inputs:**
  His account must be verified on the site after logging in.
- **Outputs:**
  edit personal information successfully.
- **Requirements:**
  Actor information must be added by logging in to the site before viewing his profile.
- **Pre-condition:**
  The actor must be logged in to the system and has permission.
- **Post-condition:**
  The actor can edit this profile.

➢ Add Doctor: -

- **Function:**

  Add Doctor
- **Actors:**

  Administrator
- **Priority:**

  High
- **Description:**
  When the administrator login, he can add the Doctor to the website database and give him an account to make the doctor can do many functions

- **Inputs:**
  Admin should write first name, last name, age, id, username, and password in the right way and correct data to add successfully;
- **Outputs:**
  The administrator adds doctors to the system and makes accounts for the doctors and gives them some permeations.
- **Requirements:**
  The data about a required doctor will be added.
- **Pre-condition:**
  The administrator had signed in to his profile (system), has permeation and the doctor give his data to the administrator
- **Post-condition:**
  The doctor is added and has an account.

➢ View Doctor: -

- **Function:**

  View Doctor
- **Actors:**

  Administrator
- **Priority:**

  Medium
- **Description:**
  When the administrator login, he can add the Doctor to the website database and give him an account to make the doctor can do many functions, and he can view doctor details like "see his appointments, personal information, and the number of patients who have examined him.”.
- **Inputs:**
  Admin should write first name, last name, age, id, username, and password in the right way and correct data to add successfully; can view the details by clicking “view info”.
- **Outputs:**
  The administrator adds doctors to the system and makes accounts for the doctors and gives them some permeations, views doctors’ details from the system.
- **Requirements:**
  The data about a required doctor will be added.
- **Pre-condition:**

The administrator had signed in to his profile (system), has permeation and the doctor give his data to the administrator

- **Post-condition:**
  The administrator views details about all the doctors.

➢ Delete Doctor: -

- **Function:**

  Delete Doctor

- **Actors:**

  Administrator

- **Priority:**

  High

- **Description:**
  When the administrator login, he can add the Doctor to the website database and give him an account to make the doctor can do many functions, and he can delete the doctor.

- **Inputs:**
  Admin should write first name, last name, age, id, username, and password in the right way and correct data to add successfully, and can delete the doctor by clicking “delete doctor”

- **Outputs:**
  The administrator adds doctors to the system and makes accounts for the doctors and gives them some permeations, and can delete doctors’ details from the system

- **Requirements:**
  The data about a required doctor will be added.

- **Pre-condition:**
  The administrator had signed in to his profile (system), has permeation and the doctor give his data to the administrator

- **Post-condition:**
  The administrator deletes all the doctors.

➢ Add Patient: -

- **Function:**

  Add Patient

- **Actors:**

  Administrator

- **Priority:**
- High
- **Description:**
  When the administrator login, he can add the patient to the website database and give him an account to make the patient can do many functions.
- **Inputs:**
  Admin should write first name, last name, age, id, username, and password in the right way and correct data to add successfully;
- **Outputs:**
  The administrator adds patients to the system and makes accounts for the patients and gives them some permeations.
- **Requirements:**
  The data about a required patient will be added.
- **Pre-condition:**
  The administrator had signed in to his profile (system), has permeation and the patient give his data to the administrator
- **Post-condition:**
  The patient is added and has an account.

➢ View Patient: -

- **Function:**

  View Patient
- **Actors:**

  Administrator
- **Priority:**

  Medium
- **Description:**
  When the administrator login, he can add the patient to the website database and give him an account to make the patient can do many functions, and he can view patient details like " Viewing the dates he booked, personal information".
- **Inputs:**
  Admin should write first name, last name, age, id, username, and password in the right way and correct data to add successfully; can view the details by clicking "view info".
- **Outputs:**

The administrator adds patients to the system and makes accounts for the patients and gives them some permeations, views patients' details from the system.

- **Requirements:**
  The data about a required patient will be added.
- **Pre-condition:**
  The administrator had signed in to his profile (system), has permeation and the patient give his data to the administrator
- **Post-condition:**
  The administrator views details about all the patients.

➢ Delete Patient: -

- **Function:**

  Delete Patient
- **Actors:**

  Administrator
- **Priority:**

  High
- **Description:**
  When the administrator login, he can add the patient to the website database and give him an account to make the patient can do many functions, and he can delete the patient.
- **Inputs:**
  Admin should write first name, last name, age, id, username, and password in the right way and correct data to add successfully, and can delete the patient by clicking "delete patient"
- **Outputs:**
  The administrator adds patients to the system and makes accounts for the patients and gives them some permeations, and deletes patients' details from the system
- **Requirements:**
  The data about a required patient will be added.
- **Pre-condition:**
  The administrator had signed in to his profile (system), has permeation and the patient give his data to the administrator.
- **Post-condition:**
  The administrator deletes all the patients.

➢ Add Analytics Specialist: -

- **Function:**

  Add Analytics Specialist

- **Actors:**

  Administrator

- **Priority:**

  High

- **Description:**
  When the administrator login, he can add the analytics specialist to the website database and give him an account to make the analytics specialist can do many functions.

- **Inputs:**
  Admin should write first name, last name, age, id, username, and password in the right way and correct data to add successfully;

- **Outputs:**
  The administrator adds analytics specialists to the system and makes accounts for the analytics specialists and gives them some permeations.

- **Requirements:**
  The data about a required analytics specialist will be added.

- **Pre-condition:**
  The administrator had signed in to his profile (system), has permeation and the analytics specialists give his data to the administrator

- **Post-condition:**
  The analytics specialist is added and has an account.

➢ View Analytics Specialists: -

- **Function:**

  View analytics specialists

- **Actors:**

  Administrator

- **Priority:**

  Medium

- **Description:**
  When the administrator login, he can add the analytics specialist to the website database and give him an account to make the analytics specialist can do many functions, and he can view analytics specialist details like " Viewing the personal information".

- **Inputs:**
  Admin should write first name, last name, age, id, username, and password in the right way and correct data to add successfully; can view the details by clicking “view info”.
- **Outputs:**
  The administrator adds an analytics specialist to the system and makes accounts for the analytics specialists and gives them some permeations, views analytics specialist details from the system.
- **Requirements:**
  The data about a required analytics specialist will be added.
- **Pre-condition:**
  The administrator had signed in to his profile (system), has permeation and the analytics specialist give his data to the administrator
- **Post-condition:**
  The administrator views details about all the analytics specialists.

➢ Delete Analytics Specialist: -

- **Function:**

  Delete analytics specialist

- **Actors:**

  Administrator
- **Priority:**

  High
- **Description:**
  When the administrator login, he can add the analytics specialist to the website database and give him an account to make the analytics specialist can do many functions, and he can delete the analytics specialist.
- **Inputs:**
  Admin should write first name, last name, age, id, username, and password in the right way and correct data to add successfully, and can delete the analytics specialist by clicking “delete pharmacist”
- **Outputs:**
  The administrator adds analytics specialists to the system and makes accounts for the analytics specialists and gives them some permeations, and deletes the analytics specialist’ details from the system
- **Requirements:**
  The data about a required analytics specialist will be added.
- **Pre-condition:**

The administrator had signed in to his profile (system), has permeation and the analytics specialist give his data to the administrator

- **Post-condition:**
  The administrator deletes all the analytics specialists.

➢ Add Pharmacist: -

- **Function:**

  Add Pharmacist

- **Actors:**

  Administrator

- **Priority:**

  High

- **Description:**
  When the administrator login, he can add the Pharmacist to the website database and give him an account to make the Pharmacist can do many functions.

- **Inputs:**
  Admin should write first name, last name, age, id, username, and password in the right way and correct data to add successfully.

- **Outputs:**
  The administrator adds Pharmacists to the system and makes accounts for the Pharmacists and gives them some permeations.

- **Requirements:**
  The data about a required pharmacist will be added.

- **Pre-condition:**
  The administrator had signed in to his profile (system), has permeation and the Pharmacist give his data to the administrator.

- **Post-condition:**
  The Pharmacist is added and has an account.

➢ View Pharmacist: -

- **Function:**

  View Pharmacists

- **Actors:**

  Administrator

- **Priority:**

  Medium

- **Description:**
  When the administrator login, he can add the Pharmacist to the website database and give him an account to make the Pharmacist can do many functions, and he can view Pharmacist details like " Viewing the personal information".
- **Inputs:**
  Admin should write first name, last name, age, id, username, and password in the right way and correct data to add successfully; can view the details by clicking "view info".
- **Outputs:**
  The administrator adds Pharmacists to the system and makes accounts for the Pharmacists and gives them some permeations, views Pharmacist details from the system.
- **Requirements:**
  The data about a required Pharmacist will be added.
- **Pre-condition:**
  The administrator had signed in to his profile (system), has permeation and the Pharmacist give his data to the administrator.
- **Post-condition:**
  The administrator views details about all the pharmacists.

➢ Delete Pharmacist: -

- **Function:**

  Delete Pharmacist
- **Actors:**

  Administrator
- **Priority:**

  High
- **Description:**
  When the administrator login, he can add the Pharmacist to the website database and give him an account to make the Pharmacist can do many functions, and he can delete the pharmacist.
- **Inputs:**
  Admin should write first name, last name, age, id, username, and password in the right way and correct data to add successfully, and can delete the pharmacist by clicking "delete pharmacist"
- **Outputs:**

The administrator adds pharmacists to the system and makes accounts for the pharmacists and gives them some permeations, and deletes the pharmacist' details from the system

- **Requirements:**
  The data about a required Pharmacist will be added.
- **Pre-condition:**
  The administrator had signed in to his profile (system), has permeation and the pharmacist give his data to the administrator
- **Post-condition:**
  The administrator deletes all the pharmacists.

➢ Add Radiology Doctor: -

- **Function:**

  Add Radiology Doctor
- **Actors:**

  Administrator
- **Priority:**

  High
- **Description:**
  When the administrator login, he can add the Radiology Doctor to the website database and give him an account to make the Radiology Doctor can do many functions.
- **Inputs:**
  Admin should write first name, last name, age, id, username, and password in the right way and correct data to add successfully.
- **Outputs:**
  The administrator adds Radiology Doctors to the system and makes accounts for the Radiology Doctors and gives them some permeations.
- **Requirements:**
  The data about a required Radiology Doctor will be added.
- **Pre-condition:**
  The administrator had signed in to his profile (system), has permeation and the Radiology Doctor give his data to the administrator.
- **Post-condition:**
  The Radiology Doctor is added and has an account.

➢ View Radiology Doctor: -

- **Function:**

View Radiology Doctor

- **Actors:**

  Administrator
- **Priority:**

  Medium
- **Description:**
  When the administrator login, he can add the Radiology Doctor to the website database and give him an account to make the Radiology Doctor can do many functions, and he can view Radiology Doctor details like " Viewing the personal information".
- **Inputs:**
  Admin should write first name, last name, age, id, username, and password in the right way and correct data to add successfully; can view the details by clicking "view info".
- **Outputs:**
  The administrator adds Radiology Doctors to the system and makes accounts for the Radiology Doctors and gives them some permeations, views Radiology Doctors' details from the system.
- **Requirements:**
  The data about a required Radiology Doctor will be added.
- **Pre-condition:**
  The administrator had signed in to his profile (system), has permeation and the Radiology Doctor give his data to the administrator
- **Post-condition:**
  The administrator views details about all the Radiology Doctors.

➢ Delete Radiology Doctor: -

- **Function:**

  Delete Radiology Doctor
- **Actors:**

  Administrator
- **Priority:**

  High
- **Description:**
  When the administrator login, he can add the Radiology Doctor to the website database and give him an account to make the Radiology Doctor can do many functions, and he can delete the Radiology Doctor.

- **Inputs:**
  Admin should write first name, last name, age, id, username, and password in the right way and correct data to add successfully, and can delete the radiology doctor by clicking “delete radiology doctor”
- **Outputs:**
  The administrator adds radiology doctors to the system and makes accounts for the radiology doctors and gives them some permeations, and deletes the radiology doctor’ details from the system
- **Requirements:**
  The data about a required Radiology Doctor will be added.
- **Pre-condition:**
  The administrator had signed in to his profile (system), has permeation and the radiology doctor give his data to the administrator.
- **Post-condition:**
  The administrator deletes all the radiology doctors.

➢ Add Accountant: -

- **Function:**

  Add Accountant

- **Actors:**

  Administrator
- **Priority:**

  High
- **Description:**
  When the administrator login, he can add the Accountant to the website database and give him an account to make the Accountant can do many functions.
- **Inputs:**
  Admin should write first name, last name, age, id, username, and password in the right way and correct data to add successfully.
- **Outputs:**
  The administrator adds Accountants to the system and makes accounts for the Accountants and gives them some permeations.
- **Requirements:**
  The data about a required Accountant will be added.
- **Pre-condition:**
  The administrator had signed in to his profile (system), has permeation and the Accountant give his data to the administrator

- **Post-condition:**
  The Accountant is added and has an account.

➢ View Accountant: -

- **Function:**

  View Accountant

- **Actors:**

  Administrator

- **Priority:**

  Medium

- **Description:**
  When the administrator login, he can add the Accountant to the website database and give him an account to make the Accountant can do many functions, and he can view Accountant details like " Viewing the personal information".

- **Inputs:**
  Admin should write first name, last name, age, id, username, and password in the right way and correct data to add successfully; can view the details by clicking "view info".

- **Outputs:**
  The administrator adds Accountants to the system and makes accounts for the Accountants and gives them some permeations, views the Accountant' details from the system.

- **Requirements:**
  The data about a required Accountant will be added.

- **Pre-condition:**
  The administrator had signed in to his profile (system), has permeation and the Accountant give his data to the administrator

- **Post-condition:**
  The administrator views details about all the Accountants.

➢ Delete Accountant: -

- **Function:**

  Delete Accountant

- **Actors:**

  Administrator

- **Priority:**

High

- **Description:**
  When the administrator login, he can add the Accountant to the website database and give him an account to make the Accountant can do many functions, and he can delete the Accountant.
- **Inputs:**
  Admin should write first name, last name, age, id, username, and password in the right way and correct data to add successfully, and can delete the Accountant by clicking "delete Accountant"
- **Outputs:**
  The administrator adds Accountants to the system and makes accounts for the Accountants and gives them some permeations, and deletes the Accountant' details from the system
- **Requirements:**
  The data about a required Accountant will be added.
- **Pre-condition:**
  The administrator had signed in to his profile (system), has permeation and the radiology doctor give his data to the administrator.
- **Post-condition:**
  The administrator deletes all the Accountants.

➢ Accept Join Requests: -

- **Function:**
  admin accepts Join requests.
- **Actors:**
  Administrator
- **Priority:**
  High
- **Description:**
  The system is available to accept 'doctors requests by the admin, and requests for the doctor to join the site to work on it.
- **Input:**
  Click on the "accept" button on the requested doctor
- **Output:**
  Admin accepts doctor join request.
- **Requirements:**
  Central database to store all doctor join request information.

- **Pre-condition:**
  The system is allowed to accept doctor join request.
  The admin must be log in system.
  The admin has permission to do this.
- **Post-condition:**
  Admin accepts doctor join request.

➢ Reject Join Requests: -

- **Function:**
  admin rejects Join requests.
- **Actors:**

  Administrator
- **Priority:**

  High
- **Description:**
  The system is available to reject 'doctors requests by the admin, and requests for the doctor to join the site to work on it.
- **Input:**
  Click on the "Reject" button on the requested doctor
- **Output:**
  Admin rejects doctor join request
- **Requirements:**
  Central database to store all doctor join request information.
- **Pre-condition:**
  The system is allowed to reject doctor join request.
  The admin must be log in system.
  The admin has permission to do this.
- **Post-condition:**
  Admin rejects doctor join request.

➢ Add Specialization: -

- **Function:**
  Add Specialization
- **Actors:**

  Administrator

- **Priority:**

  High
- **Description:**
  When the admin login, he can add a specialization to the website database, and make patient can access to view it
- **Inputs:**
  Admin should write name, id, and some information about the specialization in the right way and correct data to add successfully
- **Outputs:**
  Admin adds specializations to system and patient can access to view it
- **Requirements:**
  The data about a required specialization will be added
- **Pre-condition:**

Admin had signed into his profile (system) and has a permeation

- **Post-condition:**
  specialization is added.

➢ View Specialization

- **Function:**
  View specialization
- **Actors:**

  Administrator
- **Priority:**

  Medium
- **Description:**
  When the System User login, he can view specialization details like “name and some other info”
- **Inputs:**
  System User should write the id or specialization name in the right way and correct data to be able to view the details and click “view info”
- **Outputs:**
  System User views specializations details from the system
- **Requirements:**
  The data about a required specialization that System User wants to view
- **Pre-condition:**

The system user had signed into his profile (system) and has a permeation

- **Post-condition:**

System User view details about all the specializations.

➢ Update Specialization

- **Function:**
  Update specialization
- **Actors:**

  Administrator
- **Priority:**

  Medium
- **Description:**
  Updating Specialization is a behavior done by an Administrator. When the Administrator updates a specialization, the old information of the specialization will be changed to new information.
- **Inputs:**
  The Administrator chooses specialization to make updates on him.
- **Outputs:**
  The specialization will be successfully updated.
- **Requirements:**
  Only the administrator can update the specialization that exists in the system.
- **Pre-condition:**
  The administrator must log in system and verified to update the specialization is existing in the system
- **Post-condition:**
  The specialization will be updated in the database.

➢ Delete Specialization

- **Function:**
  Delete specialization
- **Actors:**

  Administrator
- **Priority:**

  Medium
- **Description:**
  The admin is available to delete a Specialization with his information
- **Inputs:**
  The admin must enter the information about the Specialization system with everything the Specialization contains and with more accurate details.

- **Outputs:**
  The specialization will be deleted successfully.
- **Requirements:**
  Only the administrator can delete this existing specialization from the system.
- **Pre-condition:**
  The administrator must log in system and verify to delete the specialization is existing in the system
- **Post-condition:**
  The specialization will be deleted from the database.

➢ Add Ambulance: -

- **Function:**
  Add Ambulance
- **Actors:**
  Administrator
- **Priority:**
  High
- **Description:**
  When the admin login, he can add an Ambulance to the website database, and make patient can access to view it
- **Inputs:**
  Admin should write name, id, and some information about the Ambulance in the right way and correct data to add successfully
- **Outputs:**
  Admin adds Ambulances to the system and patients can access to view it
- **Requirements:**
  The data about a required Ambulance will be added
- **Pre-condition:**

Admin had signed into his profile (system) and has a permeation

- **Post-condition:**
  Ambulance is added.

➢ View Ambulance

- **Function:**
  View Ambulance

- **Actors:**

  Administrator
- **Priority:**

  Medium
- **Description:**
  When the System User login, he can view Ambulance details
- **Inputs:**
  System User should write the id or Ambulance name in the right way and correct data to be able to view the details and click “view info”
- **Outputs:**
  user views ambulances detail from the system
- **Requirements:**
  The data about a required Ambulance that System wants to view
- **Pre-condition:**
  The system user had signed into his profile (system) and has a permeation
- **Post-condition:**
  System User view details about all the Ambulances.

➢ Update Ambulance

- **Function:**
  Update ambulance
- **Actors:**

  Administrator
- **Priority:**

  Medium
- **Description:**
  Updating an ambulance is a behavior done by an Administrator. When the Administrator updates an ambulance, the old information about the ambulance will be changed to new information.
- **Inputs:**
  The Administrator chooses an ambulance to make updates on him.
- **Outputs:**
  The specialization will be successfully updated.
- **Requirements:**
  Only the administrator can update the ambulance that exists in the system.
- **Pre-condition:**
  The administrator must log in system and verified to update the ambulance is existing in the system

- **Post-condition:**
  The ambulance will be updated in the database.

➢ Delete Ambulance

- **Function:**
  Delete ambulance
- **Actors:**

  Administrator
- **Priority:**

  Medium
- **Description:**
  The admin is available to delete an ambulance with his information
- **Inputs:**
  The admin must enter the information about the ambulance system with everything the ambulance contains and with more accurate details.
- **Outputs:**
  The ambulance will be deleted successfully.
- **Requirements:**
  Only the administrator can delete this existing ambulance from the system.
- **Pre-condition:**
  The administrator must log in system and verify to delete the ambulance is existing in the system
- **Post-condition:**
  The ambulance will be deleted from the database.

➢ Add Insurance: -

- **Function:**
  Add Insurance
- **Actors:**

  Administrator
- **Priority:**

  High
- **Description:**
  When the admin login, he can add Insurance to the website database, and make patient can access to view it
- **Inputs:**
  Admin should write name, id, and some information about the Insurance in the right way and correct data to add successfully

- **Outputs:**
  Admin adds Insurance to the system and patients can access to view it
- **Requirements:**
  The data about a required Ambulance will be added
- **Pre-condition:**
  Admin had signed into his profile (system) and has a permeation
- **Post-condition:**
  Insurance is added.

➢ View Insurance

- **Function:**
  View Insurance
- **Actors:**

  Administrator
- **Priority:**

  Medium
- **Description:**
  When the System User login, he can view Insurance details
- **Inputs:**
  System User should write the id of Insurance in the right way and correct data to be able to view the details and click "view info"
- **Outputs:**
  user views Insurance detail from the system
- **Requirements:**
  The data about required Insurance that System wants to view
- **Pre-condition:**

The system user had signed into his profile (system) and has a permeation

- **Post-condition:**
  System User view details about all the Insurance.

➢ Update Insurance

- **Function:**
  Update Insurance
- **Actors:**

  Administrator
- **Priority:**

  Medium

- **Description:**
  Updating Insurance is a behavior done by an Administrator. When the Administrator
  Update Insurance, the old information about the Insurance will be changed to new information.
- **Inputs:**
  The Administrator chooses an Insurance to make updates on him.
- **Outputs:**
  The Insurance will be successfully updated.
- **Requirements:**
  Only the administrator can update the Insurance that exists in the system.
- **Pre-condition:**
  The administrator must log in system and verify to update the Insurance is existing in the system
- **Post-condition:**
  The Insurance will be updated in the database.

➢ Delete Insurance

- **Function:**
  Delete Insurance
- **Actors:**
  Administrator
- **Priority:**
  Medium
- **Description:**
  The admin is available to delete an Insurance with his information
- **Inputs:**
  The admin must enter the information about the Insurance system with everything the Insurance contains and with more accurate details.
- **Outputs:**
  The Insurance will be deleted successfully.
- **Requirements:**
  Only the administrator can delete this existing Insurance from the system.
- **Pre-condition:**
  The administrator must log in system and verify to delete the Insurance is existing in the system
- **Post-condition:**
  The Insurance will be deleted from the database.

➢ View medical Analysis

- **Function:**
  View medical analysis
- **Actors:**
  Administrator
- **Priority:**
  medium
- **Description:**
  When the User login, he can view medical analysis details
- **Inputs:**
  Users should write medical analysis in the right way and correct data to be able to view the details and click “view info”
- **Outputs:**
  The user views medical analysis details from the system
- **Requirements:**
  The data about required medical analysis that the user wants to view
- **Pre-condition:**
  The user had login into his profile (system) and has permission
- **Post-condition:**
  User view details about all filtered the medical analysis.

➢ View X-rays

- **Function:**
  View x-rays
- **Actors:**
  Administrator
- **Priority:**
  medium
- **Description:**
  When the User login, he can view the details of the x-ray
- **Inputs:**
  User should write x-rays in the right way and correct data to be able to view the details and click “view info”
- **Outputs:**
  The user views x-ray details from the system
- **Requirements:**
  The data about required x-rays that the user wants to view
- **Pre-condition:**
  The user had login into his profile (system) and has permission

- **Post-condition:**
  User view details about all filtered the x-rays.

➢ View Invoices

- **Function:**
  View invoices
- **Actors:**
  Administrator
- **Priority:**
  Medium
- **Description:**
  When the User login, he can view invoices details
- **Inputs:**
  Users should write invoices in the right way and correct data to be able to view the details and click “view info”
- **Outputs:**
  The user views invoices details from the system
- **Requirements:**
  The data about required invoices that the user wants to view
- **Pre-condition:**
  The user had login into his profile (system) and has permission
- **Post-condition:**
  User view details about all filtered the invoices.

➢ Add Medicine: -

- **Function:**
  Add Medicine
- **Actors:**
  Pharmacist.
- **Priority:**
  High
- **Description:**
  Adding Medicine is behavior done by a pharmacist. When the pharmacist adds a medicine, He’ll add details for the medicine such as the name and image etc.
- **Input:**
  The pharmacist chooses category he wants to put the medicine in.
- **Output:**

The medicine will be successfully added.

- **Requirements:**
  Only the pharmacist can add the medicine.
- **Pre-condition:**
  The pharmacist must log in system and verified to add pharmacist.
- **Post-condition:**
  The medicine will be saved to database.

➢ Update Medicine: -

- **Function:**
  Update Medicine
- **Actors:**

  Pharmacist.
- **Priority:**

  medium
- **Description:**
  Updating Medicine is behavior done by a pharmacist. When the pharmacist Update a medicine, the old information of medicine will be changed to new information.

- **Input:**
  The pharmacist chooses medicine to make update on him.
- **Output:**
  The medicine will be successfully updated.
- **Requirements:**
  This medicine exists in system and only the pharmacist can update it.
- **Pre-condition:**
  The pharmacist must log in system and verified to update pharmacist and medicine is exist in system.
- **Post-condition:**
  The medicine will be updated in database.

➢ Delete Medicine: -

- **Function:**
  Delete Medicine
- **Actors:**

  Pharmacist.
- **Priority:**

high

- **Description:**
  Delete Medicine is behavior done by a pharmacist. When the pharmacist delete a medicine, the medicine will be deleted from the medicine list.
- **Input:**
  The pharmacist chooses medicine to delete.
- **Output:**
  The medicine will be successfully deleted.
- **Requirements:**
  This medicine exists in system and only the pharmacist can delete it.
- **Pre-condition:**
  The pharmacist must log in system and verified to delete medicine and medicine is exist in system.
- **Post-condition:**
  The medicine will be deleted from database.

➢ Upload Medical Analysis: -

- **Function:**
  Upload medical analysis.
- **Actors:**
  Analytics specialist.
- **Priority:**
  medium
- **Description:**
  After Analytics specialist finish medical analysis, he uploads result of medical analysis as file on system.
- **Input:**
  The doctor must send prescription about what type of medical analysis that he does.
- **Output:**
  The medical analysis result will be successfully Added.
- **Requirements:**
  Only the Analytics specialist can Add the result of medical analysis.
- **Pre-condition:**
  The Analytics specialist must log in system and type of medical analysis is exist.
- **Post-condition:**
  The medical analysis result will be saved to database.

➢ Upload X-Rays: -

- **Function:**
  Upload x-ray.
- **Actors:**

  Radiology doctor.
- **Priority:**

  medium
- **Description:**
  After Radiology doctor finish x-ray, he uploads result of x-ray as file on system.
- **Input:**
  The doctor must send prescription about what type of x-ray that he does.
- **Output:**
  The x-ray result will be successfully Added.
- **Requirements:**
  Only the Radiology doctor can Add the result of medical analysis.
- **Pre-condition:**
  The Radiology doctor must log in system and type of x-ray is exist.
- **Post-condition:**
  The x-ray result will be saved to database.

➢ Add Admin: -

- **Function:**
  Add Admin.
- **Actor:**

  The Hospital Manager
- **Priority:**

  Critical
- **Description:**
  The Hospital Manager is available to add admin and describe his privileges.
- **Input:**
  Enter Admin Information
- **Output:**
  The Hospital Manager adds admin
- **Requirements:**
  The Hospital Manager **must add admin.**

- **Pre-condition:**
  The system is allowed to add admins
  The Hospital Manager must be logged in the system
  Admin hasn't been created yet.
- **Post-condition:**
  Admin is added successfully.

➢ View Admin: -

- **Function:**
  View Admin.
- **Actor:**

  The Hospital Manager
- **Priority:**

  Critical
- **Description:**
  The Hospital Manager can view the admins he created.
- **Input:**
  Click on the list button.
- **Output:**
  The Hospital Manager views list of admins.
- **Requirements:**
  The Hospital Manager **must view admins list.**
- **Pre-condition:**
  The system is allowed to view admins.
  The Hospital Manager must be logged in the system.
  Admin has been created.
- **Post-condition:**
  Admin is viewed successfully.

➢ Update Admin: -

- **Function:**
  Update Admin.
- **Actor:**

  The Hospital Manager
- **Priority:**

  Critical
- **Description:**

The Hospital Manager can update admins he created.

- **Input:**
  Admin ID
- **Output:**
  The Hospital Manager updates admin.
- **Requirements:**
  The Hospital Manager **can update admin description and privileges.**
- **Pre-condition:**
  The system is allowed to update admin.
  The Hospital Manager must be logged in the system.
  Admin has been created.
- **Post-condition:**
  Admin is updated successfully.

➢ Delete Admin: -

- **Function:**
  Delete Admin.
- **Actor:**
  The Hospital Manager
- **Priority:**
  Critical
- **Description:**
  The Hospital Manager can delete admins he created.
- **Input:**
  Admin ID.
- **Output:**
  The Hospital Manager deletes admins.
- **Requirements:**
  The Hospital Manager **can delete admins.**
- **Pre-condition:**
  The system is allowed to delete roles.
  The Hospital Manager must be logged in the system.
  Admin have been created.
- **Post-condition:**
  Admin is deleted successfully.

➢ Add Role: -

- **Function:**
  Add Role.

- **Actor:**

  The Hospital Manager
- **Priority:**

  Critical
- **Description:**
  The Hospital Manager is available to add roles and describe the privilege for each user. The role is the validities that the user takes, whoever (Admin, Pharmacist, Doctor, Analytics Specialist, Radiology Doctor, Accountant or Patient) to perform certain tasks.
- **Input :**
  The Hospital Manager **create roles and describe privileges for each role**
- **Output :**
  The Hospital Manager adds the roles
- **Requirements :**
  The Hospital Manager **must add roles and describe their privileges**
- **Pre-condition :**
  The system is allowed to add roles
  The Hospital Manager must be logged in the system
  Role hasn't been created yet.
- **Post-condition:**
  Role is created successfully.

➢ View Role: -

- **Function:**
  View Role.
- **Actor:**

  The Hospital Manager
- **Priority:**

  Critical
- **Description:**
  The Hospital Manager can view roles he created.
- **Input:**
  The Hospital Manager click on list button.
- **Output:**
  The Hospital Manager view role page.
- **Requirements:**
  The Hospital Manager **can view roles and privileges.**

- **Pre-condition:**
  The system is allowed to view roles.
  The Hospital Manager must be logged in the system.
  Role has been created.
- **Post-condition:**
  Role is viewed successfully.

➢ Update Role: -

- **Function:**
  Update Role.
- **Actor:**

  The Hospital Manager
- **Priority:**

  Critical
- **Description:**
  The Hospital Manager can update roles he created.
- **Input:**
  Role name and Role ID
- **Output:**
  The Hospital Manager updates role
- **Requirements:**
  The Hospital Manager **can update roles and privileges**
- **Pre-condition:**
  The system is allowed to update roles
  The Hospital Manager must be logged in the system
  Role has been created.
- **Post-condition:**
  Role is updated successfully.

➢ Delete Role: -

- **Function:**
  Delete Role.
- **Actor:**

  The Hospital Manager
- **Priority:**

  Critical
- **Description:**
  The Hospital Manager can delete roles he created.

- **Input:**
  Enter Role ID or Role Name
- **Output:**
  The Hospital Manager deletes roles
- **Requirements:**
  The Hospital Manager **can delete roles**
- **Pre-condition:**
  The system is allowed to delete roles
  The Hospital Manager must be logged in the system
  Roles have been created.
- **Post-condition:**
  Roles deleted successfully.

➢ Add Invoice: -

- **Function:**
  Add Invoice.
- **Actor:**

  Accountant
- **Priority:**

  High
- **Description:**
  Accountant must add invoices and manage the financial part of the hospital.
- **Input:**
  Add Invoice details and for whom this invoice belong to.
- **Output:**
  Accountant creates invoice
- **Requirements:**
  Accountant **should write the** invoice **details**
- **Pre-condition:**
  The system is allowed to create invoices
  The Accountant must be logged in the system
  Patients and Doctors have been created.
- **Post-condition:**
  Invoice is added successfully.

➢ Request to join hospital: -

- **Function:**
  Join doctor.

- **Actor:**

  Doctor
- **Priority:**

  Medium.
- **Description:**
  When doctor register by filling his information, he/she will wait until Request will be Accepted then he/she can login into the system.
- **Input:**
  Add doctor`s information.
- **Output:**
  The Doctor`s information saved successfully in database.
- **Requirements:**
  Doctor must write the right data can register.
- **Pre-condition:**
  Doctor don't have an account.
- **Post-condition:**
  Doctor register successfully and can login into the system.

➢ Diagnose Patient: -

- **Function:**
  Diagnose Patient.
- **Actor:**

  Doctor
- **Priority:**

  High
- **Description:**
  Doctor must add patient diagnosis and can send them to the specialist.
- **Input:**
  Add patient details and his/her diagnosis.
- **Output:**
  The Doctor adds patient diagnosis
- **Requirements:**
  Doctor **must diagnose patient and write the diagnosis details , send patient to specialist if needed**
- **Pre-condition:**
  The system is allowed to add diagnosis
  The Doctor must be logged in the system

Patient, Radiology Doctor, Analytics Specialist and Doctors have been created.
The Doctor should view patient history.

- **Post-condition:**
  Diagnosis is added successfully.

➢ Create Patient Prescription: -

- **Function:**
  Create prescription
- **Actor:**
  Doctor
- **Priority:**
  High
- **Description:**
  Doctor must create prescription after diagnose patient.
- **Input:**
  Add patient details, date and medicines needed.
- **Output:**
  The Doctor adds prescription.
- **Requirements:**
  Doctor **must diagnose patient and write the diagnosis first and then write the prescription**
- **Pre-condition:**
  The system is allowed to add prescription
  The Doctor must be logged in the system
  Patients and Doctors have been created.
  Patient Diagnosis has been created
- **Post-condition:**
  Prescription is added successfully

➢ Add Service: -

- **Function:**
  Add service
- **Actor:**
  Doctor
- **Priority:**
  Medium
- **Description:**

Doctor must add service. Service is what the doctor will do when the patient books him.

- **Input:**
  Add service details.
- **Output:**
  The Doctor adds service.
- **Requirements:**
  Doctor **must add service so patient books him**
- **Pre-condition:**
  The system is allowed to add service.
  The Doctor must be logged in the system
  The Doctor has been created.
- **Post-condition:**
  Service is added successfully.

➢ Update Service: -

- **Function:**
  Update service
- **Actor:**
  Doctor
- **Priority:**
  Medium
- **Description:**
  Doctor can update the service which were added.
- **Input:**
  Write service ID. Add service details want to be updated.
- **Output:**
  The Doctor updates service.
- **Requirements:**
  Doctor **must has created service so he could update it.**
- **Pre-condition:**
  The system is allowed to update service.
  The Doctor must be logged in the system
  The Doctor has been created.
  The Service has been created
- **Post-condition:**
  Service is updated successfully.

➢ Delete Services: -

- **Function:**
  Update services
- **Actor:**
  Doctor
- **Priority:**
  Medium
- **Description:**
  Doctor can delete the added services.
- **Input:**
  Write service ID. Delete service.
- **Output:**
  The Doctor deletes the service.
- **Requirements:**
  Doctor **must has created service so he could delete it.**
- **Pre-condition:**
  The system is allowed to delete service.
  The Doctor must be logged in the system
  The Doctor has been created.
  The Service has been created.
- **Post-condition:**
  Service deleted successfully.

➢ View Services: -

- **Function:**
  View services
- **Actor:**
  Doctor
- **Priority:**
  Medium
- **Description:**
  Doctor can view services which were added.
- **Input:**
  Click on button service list
- **Output:**
  The Doctor's services view.
- **Requirements:**
  Doctor **must has created service so he could view it.**

- **Pre-condition:**
  The system is allowed to view service.
  The Doctor must be logged in the system
  The Doctor has been created.
  The Service has been created.
- **Post-condition:**
  Service is viewed successfully.

➢ Add Appointments:

- **Function:**
  Add appointments
- **Actors:**
  Doctor
- **Priority:**
  Medium
- **Description:**
  The doctor must add appointments. Appointments are the time that the patient books with the doctor for the examination to take place.
- **Inputs:**
  Add appointments details.
- **Outputs:**
  The Doctor adds appointments, user views appointment details from the system
- **Requirements:**
  The doctor must add appointments so the patient books him
- **Pre-condition:**
  The system is allowed to add appointments.
  The Doctor must be logged in to the system
  The Doctor has been created.
- **Post-condition:**
  Appointments are added successfully.

➢ Update Appointments:

- **Function:**
  Update appointments
- **Actors:**
  Doctor
- **Priority:**
  Medium
- **Description:**

Doctor can update the appointments which were added.

- **Inputs:**
  Write appointments ID. Add appointments details want to be updated.
- **Outputs:**
  The Doctor updates appointments.
- **Requirements:**
  Doctor **must has created** appointments **so he could update it.**
- **Pre-condition:**
  The system is allowed to update appointments.
  The Doctor must be logged in the system
  The Doctor has been created.
  The appointments have been created
- **Post-condition:**
  Appointments are updated successfully.

➢ View Appointments:

- **Function:**
  View appointments
- **Actors:**
  Doctor
- **Priority:**
  Medium
- **Description:**
  When User login, he can view appointment details like “name, date and some another info”
- **Inputs:**
  User should write appointment id or date in right way and correct data to be able to view the details and click “view info”
- **Outputs:**
  User views appointment details from the system
- **Requirements:**
  The data about required appointment that user wants to view and has permission
- **Pre-condition:**
  User had login into his profile (system) and has permission
- **Post-condition:**
  User view details about all filtered the appointment

➢ Delete Appointments:

- **Function:**
  Delete appointments
- **Actors:**
  Doctor
- **Priority:**
  Medium
- **Description:**
  Doctor can delete the appointments which were added.
- **Inputs:**
  Write appointments ID. Delete appointment.
- **Outputs:**
  The Doctor deletes appointments.
- **Requirements:**
  Doctor **must has created** appointments **so he could** delete **it.**
- **Pre-condition:**
  The system is allowed to update appointments.
  The Doctor must be logged in the system
  The Doctor has been created.
  The appointments have been created
- **Post-condition:**
  Appointments are deleted successfully.

➢ View Patient History: -

- **Function:**
  View Patient History
- **Actor:**
  Doctor
- **Priority:**
  High
- **Description:**
  Doctor can view patient history to help doctor in diagnoses.
- **Input:**
  Write patient ID.
- **Output:**
  The medical history of patient.
- **Requirements:**
  Doctor must view **patient history to help in diagnosis.**

- **Pre-condition:**
  The system is allowed to view patient history.
  The Doctor must be logged in the system
  The Doctor and patient have been created.
- **Post-condition:**
  List of patient's history is viewed successfully.

➢ Pneumonia Detection: -

- **Function:**
  Detect Pneumonia
- **Actor:**

  Radiology Doctor
- **Priority:**

  High
- **Description:**
  Radiology Doctor can upload patient lung`s x-ray to help doctor in pneumonia diagnoses.
- **Input:**
  upload lung`s x-ray.
- **Output:**
  The pneumonia Positive or Negative.
- **Requirements:**
  The Radiology Specialist did the X-ray to the patient so it can be uploaded.
- **Pre-condition:**
  The Radiology Doctor must be logged in the system.
  The X-ray is existed and uploaded.
- **Post-condition:**
  pneumonia diagnoses result and diagnoses x-ray saved in database.

➢ Register function:

- **Function:**

  Register
- **Actors:**
  Patient
- **Priority:**
  High
- **Description:**

When user register by filling his information, he can login into the system, use its services in system and user`s information is added to the database

- **Inputs:**
  User should write first name, last name, phone number, user name (E-mail) and password in right way and correct data for successfully register
- **Outputs:**
  User can login into system
- **Requirements:**
  Write the right data to can register
- **Pre-condition:**
  User don't have an account
- **Post-condition:**
  User register successfully and can login into the system

➢ Book a doctor:

- **Function:**
  Book a doctor
- **Actors:**
  Patient
- **Priority:**
  High
- **Description:**
  When user login, he can book a doctor and make an appointment based on specialty
- **Inputs:**
  User should choose the doctor who can help him and enter the date and time which suit the user to make the appointment
- **Outputs:**
  User book a doctor and make an appointment
- **Requirements:**
  Write the right data (doctor, date, time)
- **Pre-condition:**
  Users have an account and login into the system
- **Post-condition:**
  Appointment is made

➢ View Doctors

- **Function:**
  View doctors
- **Actors:**
  Patient, Administrator
- **Priority:**
  Low
- **Description:**
  When User login, he can filter and view doctor details like “name, specialty and some another info”
- **Inputs:**
  User should write specialty or doctor name in right way and correct data to be able to view the details and click “view info”
- **Outputs:**
  User views doctor details from the system
- **Requirements:**
  The data about required doctor that user wants to view
- **Pre-condition:**
  User had login into his profile (system) and has permission
- **Post-condition:**
  System User view details about all filtered the doctors

➢ View medicines

- **Function:**
  View medicines
- **Actors:**
  Patient, pharmacist
- **Priority:**
  Medium
- **Description:**
  When User login, he can filter and view medicine details like “name, benefits and some another info” and can buy it online by adding to card
- **Inputs:**
  User should write benefit or medicine name in right way and correct data to be able to view the details and click “view info”
- **Outputs:**
  User views doctor details from the system
- **Requirements:**
  The data about required medicine that user wants to view

- **Pre-condition:**
  User had login into his profile (system) and has permission
- **Post-condition:**
  User view details about all filtered the medicines

➢ Buy medicines

- **Function:**
  Buy medicines.
- **Actors:**
  Patient.
- **Priority:**
  high
- **Description:**
  When Patient login, he can buy medicines using digital wallet or cash on delivery.
- **Inputs:**
  Patient should choose the medicines to buy.
- **Outputs:**
  The patient paid for the medicine and took it.
- **Requirements:**
  The required medicine and the medicine price are available.
- **Pre-condition:**
  Patient logged in into his system and has permission to buy and view medicines.
- **Post-condition:**
  Patient took the medicine he wanted.

➢ View Specialization:

- **Function:**
  View specialization
- **Actors:**
  Patient, Administrator
- **Priority:**
  medium
- **Description:**
  When user login, he can filter and view specialization details like “name”
- **Inputs:**

User should write specialization name in right way and correct data to be able to view the details and click “view info”

- **Outputs:**
  User views specialization details from the system
- **Requirements:**
  The data about required specialization that user wants to view
- **Pre-condition:**
  User had login to his profile (system) and has permission
- **Post-condition:**
  User view details about all filtered the specializations

➢ Call Emergency:

- **Function:**
  Call emergency
- **Actors:**
  Patient
- **Priority:**
  High
- **Description:**
  When user login, he can call emergency to help him
- **Inputs:**
  User should write emergency number in right way to call it
- **Outputs:**
  User is helped by emergency
- **Requirements:**
  The number of emergency
- **Pre-condition:**
  User had login to his profile (system) and has permission
- **Post-condition:**
  User view number of emergency and call it

➢ Use medical Insurance:

- **Function:**
  Use medical insurance
- **Actors:**
  Patient
- **Priority:**
  Medium

- **Description:**
  When user login, he can use medical insurance to have a discount on price of medicines
- **Inputs:**
  User should enter your medical insurance in right way
- **Outputs:**
  User dispends the medicine from the system
- **Requirements:**
  The data about required medical insurance that user wants to use
- **Pre-condition:**
  User had login to his profile (system) and has permission
- **Post-condition:**
  User dispends the medicine from the system

➢ Talk to chatbot:

- **Function:**
  Talk of chatbot
- **Actors:**
  Patient
- **Priority:**
  Medium
- **Description:**
  When user login, he can use medical insurance to have a discount on price of medicines
- **Inputs:**
  User should open chatbot to be guided
- **Outputs:**
  User dispends the medicine from the system
- **Requirements:**
  Open the chatbot in right way and has permission
- **Pre-condition:**
  User had login to his profile (system) and has permission
- **Post-condition:**
  User talk to chatbot and be guided by it

➢ View medical Analysis

- **Function:**
  View medical analysis

- **Actors:**
  Administrator
- **Priority:**
  Medium
- **Description:**
  When User login, he can view medical analysis details
- **Inputs:**
  User should write medical analysis in right way and correct data to be able to view the details and click “view info”
- **Outputs:**
  User views medical analysis details from the system
- **Requirements:**
  The data about required medical analysis that user wants to view
- **Pre-condition:**
  User had login into his profile (system) and has permission
- **Post-condition:**
  User view details about all filtered the medical analysis

➢ View X-rays

- **Function:**
  View x-rays
- **Actors:**
  Administrator
- **Priority:**
  Medium
- **Description:**
  When User login, he can view x-rays details
- **Inputs:**
  User should write x-rays in right way and correct data to be able to view the details and click “view info”
- **Outputs:**
  User views x-rays details from the system
- **Requirements:**
  The data about required x-rays that user wants to view
- **Pre-condition:**
  User had login into his profile (system) and has permission
- **Post-condition:**
  User view details about all filtered the x-rays

➢ **View Invoices:-**

- **Function:**
  View invoices
- **Actors:**
  Administrator
- **Priority:**
  Medium

- **Description:**
  When User login, he can view invoices details
- **Inputs:**
  User should write invoices in right way and correct data to be able to view the details and click "view info"
- **Outputs:**
  User views invoices details from the system
- **Requirements:**
  The data about required invoices that user wants to view
- **Pre-condition:**
  User had login into his profile (system) and has permission
- **Post-condition:**
  User view details about all filtered the invoices

2.4.6. Non- Functional Requirements

It specifies system/software properties (such as reliability and safety), and constraints on the services or functions offered by the system (such as timing constraints, response-time), or constraints on the development process.

➢ **Usability & Humanity.**

- The product shall be easy to use on the first attempt by a member of the public without training.
- **Intuitiveness:** the interface is easy to learn and navigate; buttons, headings, and help/error messages are simple to understand

➢ **Performance.**

- **Response Time:** The system provides a fast acknowledgment.

- **User-Interface:** The user interface acknowledges fast as we are using single page application.

➢ **Maintainability & Support.**

- Expected changes, and the time allowed to make them.
- **Back-Up:** The system offers efficiency for data backup.
- **Errors:** The system must be support error handling and will track every mistake as well as keep a log of it.

➢ **Security.**

- **Logon ID: -** Any user who uses the system shall have a Logon ID and Password (Authentication).
- **Modification: -** Any modification (inert, delete, update) for the Database shall be synchronized and only by the role that user has in the ward (Authorization).

➢ **Availability.**

- The system shall be available all the time.

➢ **Software Quality.**

- Good quality of the framework= produces robust, bug free software which contains all necessary requirements Customer satisfaction.

➢ **Reusability.**

- Is part of the code going to be used elsewhere= produces simple and independent code modules that can be reused.

2.5. Advantages of the new system

- **Validation**: usage of validation and regex when logging into the system and registering for the first time.
- **Verification:** Email verification will be sent to patient when registered.

- **Roles & Permissions:** Each user has his own permission so based on user permission he can does any modification on specified tables in the database (insert, delete, update, etc.).
- **Response Time:** The system provides a fast acknowledgment.
- **User-Interface:** The user interface acknowledges fast as we are using single page application.
- **Back-Up:** The system offers efficiency for data backup.
- **System Tracking:** The system will track every mistake as well as keep a log of it.
- **Availability:** The system is available all the time.
- **Support Multilingual:** The system supports two languages (Arabic and English).
- **Support Multitenancy:** Instead of forcing you to change how you write your code, the system by default bootstraps tenancy automatically, in the background. Database connections are switched, caches are separated, file systems are prefixed.

2.6. Risk and Risk Managements

Sfsdfdfsdfdf

2.7. Use Case: -

2.7.1. Use Case Diagram:

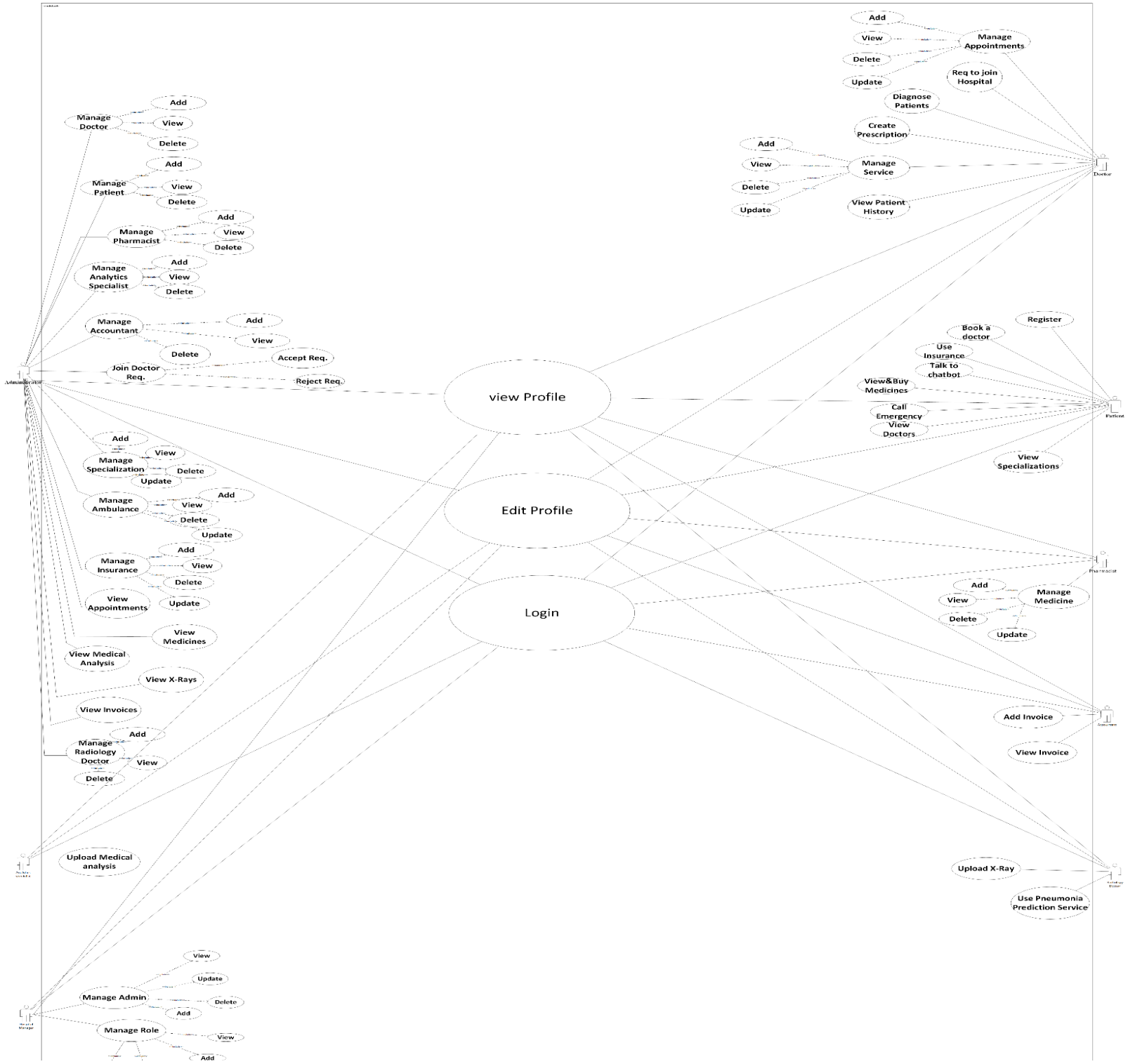
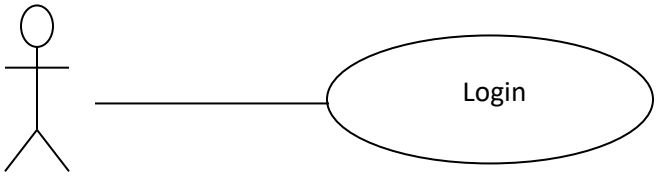


Figure 1: Use Case Diagram

2.7.2. Use Case Scenarios:

Use case: Login

Diagram :

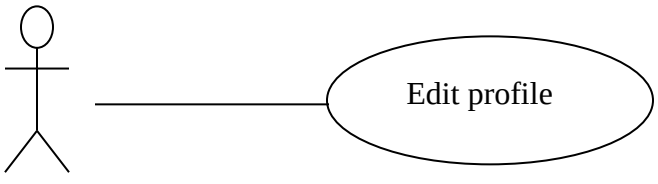


Scenario:

Log in to the system	
Actor who initiates the use case	System User
Pre-condition	The actor already has an account
Basic path	-The system user enters user ID and Password in the custom field. -click on the login button. -the system validation the entered username and Password and logs the user into the system.
Post condition	The actor entered his account successfully
Alternative Paths	If the system user enters invalid ID or Password, the system shall display an appropriate error message and returns them back to the first in the series of operation
Actor who benefits from the use case	System User

Use case: Edit profile

Diagram :

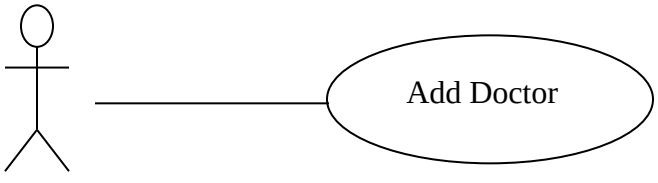


Scenario:

Edit profile	
Actor who initiates the use case	System User
Pre-condition	The actor must be logged in to the system and has permission.
Basic path	By click on edit profile in user profile can edit his data such as their name, age, profile picture, email, and password from the form opend.
Post condition	The actor can edit this profile.
Alternative Paths	If the system user enters invalid or wrong data, the system shall display an appropriate error message and returns them back to the first in the series of operation
Actor who benefits from the use case	System User

Use case: Add Doctor

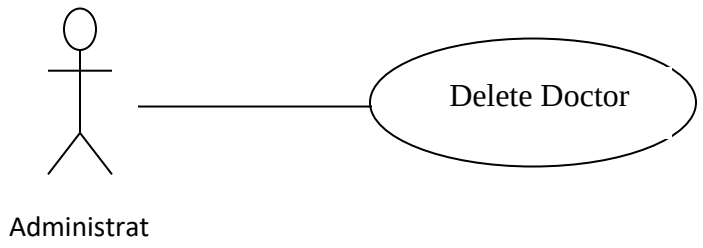
Diagram :



Scenario:

Add Doctor	
Actor who initiates the use case	Administrator
Pre-condition	The administrator had signed in to his profile (system), has permeation and the doctor give his data to the administrator.
Basic path	When the administrator login, he can add the Doctor to the website database and give him an account to make the doctor can do many functions.
Post condition	The doctor is added and has an account.
Alternative Paths	If the Administrator enters invalid doctor data, the system shall display an appropriate error message and returns them back to the first in the series of operation
Actor who benefits from the use case	Administrator

Use case: Delete Doctor
Diagram:

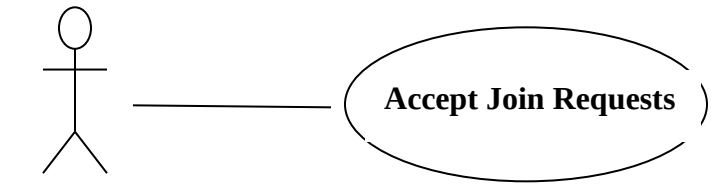


Scenario:

Delete Doctor	
Actor who initiates the use case	Administrator
Pre-condition	The administrator had signed in to his profile (system), has permeation and the doctor give his data to the administrator

Basic path	When the administrator login, he can Delete the Doctor from the website database by click on delete button on the doctor row.
Post condition	The administrator deletes all the doctors
Alternative Paths	If the Administrator in specific doctor page he can also delete doctor by click on delete button on the page.
Actor who benefits from the use case	Administrator

Use case: Accept Join Requests
Diagram:

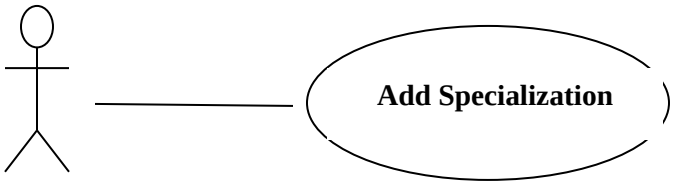


Administrat

Scenario:

Accept Join Requests	
Actor who initiates the use case	Administrator
Pre-condition	-The system is allowed to accept doctor join request -The admin must be log in system -The admin has permission to do this
Basic path	The system is available to accept 'doctors requests by the admin, and requests for the doctor to join the site to work on it
Post condition	Admin accepts doctor join request
Alternative Paths	If the doctor is already existing, the system shall display an appropriate error message.
Actor who benefits from the use case	Administrator

Use case: Add Specialization
Diagram:

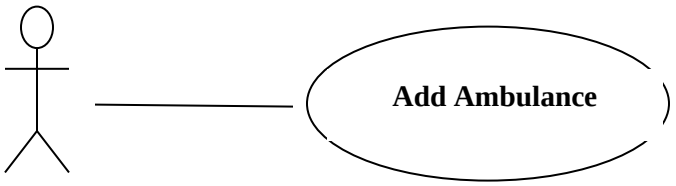


Administrat

Scenario:

Add Specialization	
Actor who initiates the use case	Administrator
Pre-condition	Admin had signed into his profile (system) and has a permeation
Basic path	When the admin login, he can add a specialization to the website database, and make patient can access to view it
Post condition	specialization is added
Alternative Paths	If the Administrator enters invalid Specialization data, the system shall display an appropriate error message and returns them back to the first in the series of operation
Actor who benefits from the use case	Administrator

Use case: Add Ambulance
Diagram:



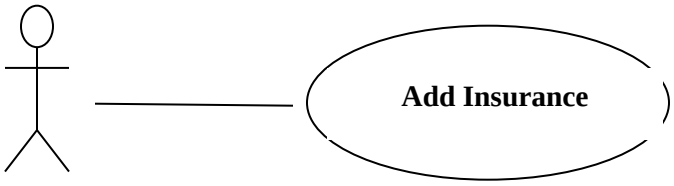
Administrat

Scenario:

Add Ambulance	
Actor who initiates the use case	Administrator
Pre-condition	Admin had signed into his profile (system) and has a permeation
Basic path	When the admin login, he can add an Ambulance to the website database, and make patient can access to view it
Post condition	Ambulance is added
Alternative Paths	If the Administrator enters invalid or wrong Ambulance data, the system shall display an appropriate error message and returns them back to the first in the series of operation
Actor who benefits from the use case	Administrator

Use case: Add Insurance

Diagram:



Administrat

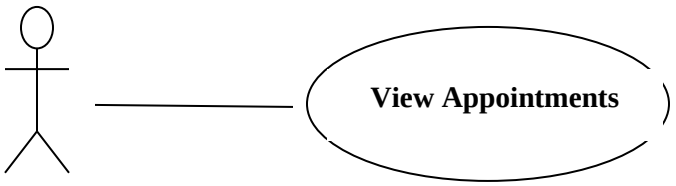
Scenario:

Add Insurance	
Actor who initiates the use case	Administrator
Pre-condition	Admin had signed into his profile (system) and has a permeation
Basic path	When the admin login, he can add Insurance to the website database, and make patient can access to view it
Post condition	Insurance is added

Alternative Paths	If the Administrator enters invalid Insurance data, the system shall display an appropriate error message and returns them back to the first in the series of operation
Actor who benefits from the use case	Administrator

Use case: View Appointments

Diagram:

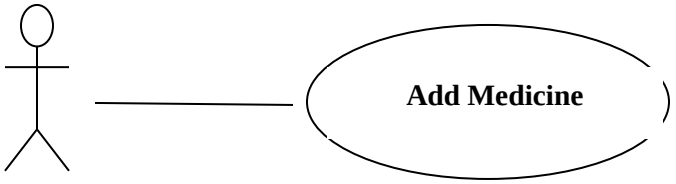


Administrat

Scenario:

View Appointments	
Actor who initiates the use case	Administrator
Pre-condition	The user had login into his profile (system) and has permission
Basic path	When the User login, he can view appointment details like "name, date and some other info
Post condition	User view details about all filtered the appointment
Alternative Paths	If the required Appointments is not found, the system should display not found message
Actor who benefits from the use case	Administrator

Use case: Add Medicine
Diagram:

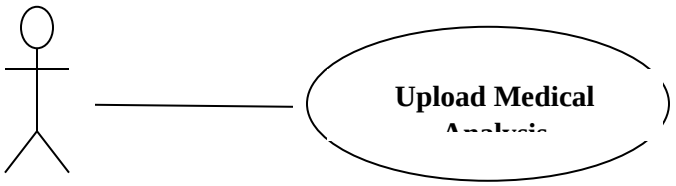


Pharmacist

Scenario:

Add Medicine	
Actor who initiates the use case	Pharmacist
Pre-condition	The pharmacist must log in system and verified to add pharmacist.
Basic path	Adding Medicine is behaviour done by a pharmacist. When the pharmacist adds a medicine, He'll add details for the medicine such as the name and image etc.
Post condition	The medicine will be saved to database.
Alternative Paths	If the Pharmacist enters invalid or wrong Medicine data, the system shall display an appropriate error message and returns them back to the first in the series of operation
Actor who benefits from the use case	Pharmacist

Use case: Upload Medical Analysis
Diagram:

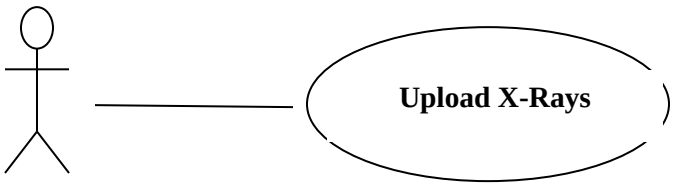


Analytics specialist

Scenario:

Upload Medical Analysis	
Actor who initiates the use case	Analytics specialist.
Pre-condition	The Analytics specialist must log in system and type of medical analysis is exist
Basic path	After Analytics specialist finish medical analysis, he uploads result of medical analysis as file on system
Post condition	The medical analysis result will be saved to database
Alternative Paths	If the Analytics specialist enters empty Medical Analysis, the system shall display an appropriate error message and returns them back to the first in the series of operation
Actor who benefits from the use case	Analytics specialist.

Use case: Upload X-Rays
Diagram:



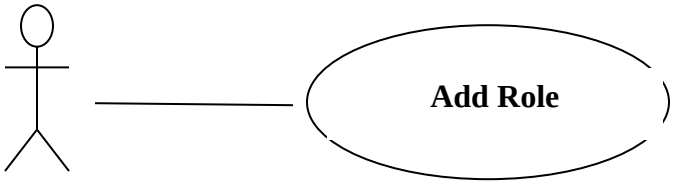
Radiology doctor.

Scenario:

Upload X-Rays	
Actor who initiates the use case	Radiology doctor.
Pre-condition	The Radiology doctor must log in system and type of x-ray is exist
Basic path	After Radiology doctor finish x-ray, he uploads result of x-ray as file on system
Post condition	The x-ray result will be saved to database

Alternative Paths	If the Radiology doctor enters empty X-Rays, the system shall display an appropriate error message and returns them back to the first in the series of operation
Actor who benefits from the use case	Radiology doctor.

Use case: Add Role
Diagram:

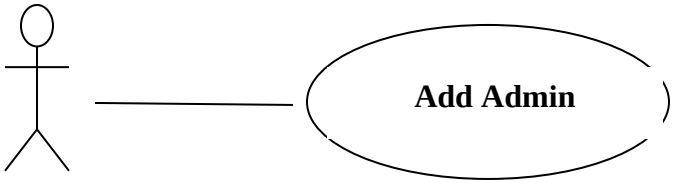


The Hospital Manager

Scenario:

Add Role	
Actor who initiates the use case	The Hospital Manager
Pre-condition	The system is allowed to add roles
Basic path	The Hospital Manager is available to add roles and describe the privilege for each user. The role is the validities that the user takes, whoever (Admin, Pharmacist, Doctor, Analytics Specialist, Radiology Doctor, Accountant or Patient) to perform certain tasks
Post condition	Role is created successfully
Alternative Paths	If the Hospital Manager enters invalid or wrong role data, the system shall display an appropriate error message and returns them back to the first in the series of operation
Actor who benefits from the use case	The Hospital Manager

Use case: Add Admin
Diagram:

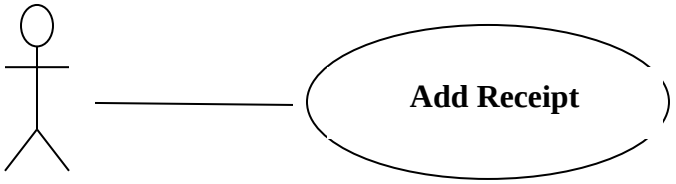


The Hospital Manager

Scenario:

Add Admin	
Actor who initiates the use case	The Hospital Manager
Pre-condition	-The system is allowed to add admins -The Hospital Manager must be logged in the system -Admin hasn't been created yet.
Basic path	The Hospital Manager is available to add admin and describe his privileges
Post condition	Admin is added successfully
Alternative Paths	If the Hospital Manager enters invalid or wrong Admin data, the system shall display an appropriate error message and returns them back to the first in the series of operation
Actor who benefits from the use case	The Hospital Manager

Use case: Add Receipt
Diagram:



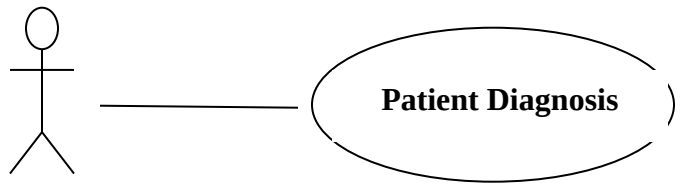
Accountant

Scenario:

Add Receipt	
Actor who initiates the use case	Accountant
Pre-condition	-The system is allowed to create receipts -The accountant must be logged in the system -Patients and Doctors have been created.
Basic path	Accountant must add receipts and manage the financial part of the hospital
Post condition	Receipt is added successfully
Alternative Paths	If the accountant enters invalid or wrong Receipt data, the system shall display an appropriate error message and returns them back to the first in the series of operation
Actor who benefits from the use case	Accountant

Use case: Patient Diagnosis

Diagram:

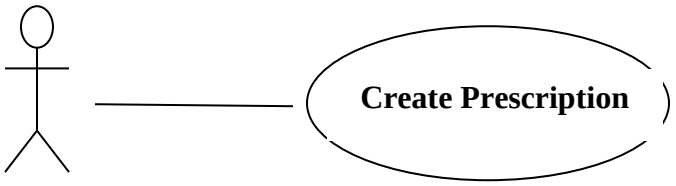


Scenario:

Patient Diagnosis	
Actor who initiates the use case	Doctor
Pre-condition	-The system is allowed to add diagnosis -The Doctor must be logged in the system -Patient , Radiology Doctor ,Analytics Specialist and Doctors have been created. -The Doctor should view patient history.
Basic path	Doctor must add patient diagnosis and can send them to the specialist.

Post condition	Diagnosis is added successfully
Alternative Paths	If the doctor enters invalid or wrong Diagnosis data, the system shall display an appropriate error message and returns them back to the first in the series of operation
Actor who benefits from the use case	Doctor

Use case: Create Prescription
Diagram:

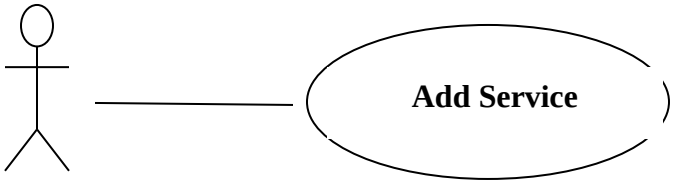


Doctor

Scenario:

Create Prescription	
Actor who initiates the use case	Doctor
Pre-condition	-The system is allowed to add prescription -The Doctor must be logged in the system -Patients and Doctors have been created. -Patient Diagnosis has been created
Basic path	Doctor must create prescription after diagnose patient and add it in patient profile.
Post condition	Prescription is added successfully
Alternative Paths	If the doctor enters invalid or wrong Prescription data, the system shall display an appropriate error message and returns them back to the first in the series of operation
Actor who benefits from the use case	Doctor

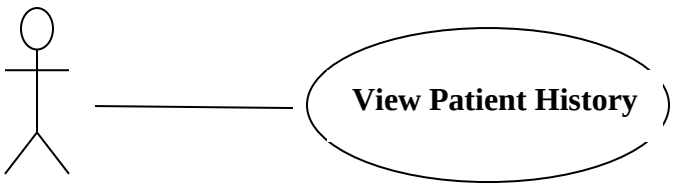
Use case: Add Service
Diagram:



Scenario:

Add Service	
Actor who initiates the use case	Doctor
Pre-condition	-The system is allowed to add service. -The Doctor must be logged in the system -The Doctor has been created
Basic path	By click on add service button will open form the doctor type service details and click submit then the service is available for patient.
Post condition	Service is added successfully
Alternative Paths	If the doctor enters invalid or wrong service data, the system shall display an appropriate error message and returns them back to the first in the series of operation
Actor who benefits from the use case	Doctor

Use case: View Patient History
Diagram:

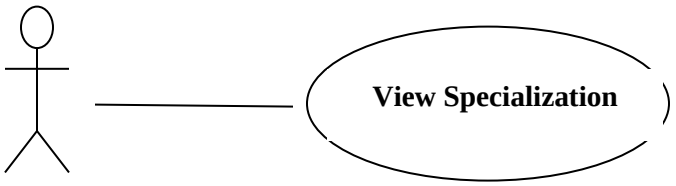


Scenario:

View Patient History	
Actor who initiates the use case	Doctor
Pre-condition	-The system is allowed to view patient history. -The Doctor must be logged in the system -The Doctor and patient have been created
Basic path	After open specific patient profile doctor can view patient history to help doctor in diagnoses
Post condition	List of patient's history is viewed successfully
Alternative Paths	If the required Patient History is not found, the system should display not found message
Actor who benefits from the use case	Doctor

Use case: View Specialization

Diagram:



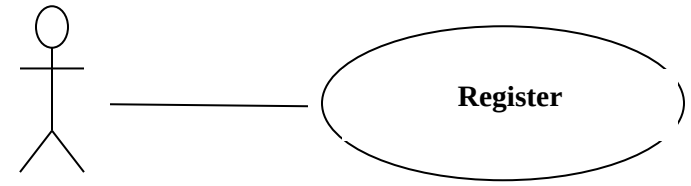
Patient, Administrator, Hospital

Scenario:

View Specialization	
Actor who initiates the use case	Patient, Administrator, Hospital manager
Pre-condition	User had login to his profile (system) and has permission
Basic path	When user login, he can filter and view specialization details like "name"
Post condition	User view details about all filtered the specializations
Alternative Paths	If the required Specialization is not found, admin should add Specialization to view his details

Actor who benefits from the use case	Patient, Administrator, Hospital manager

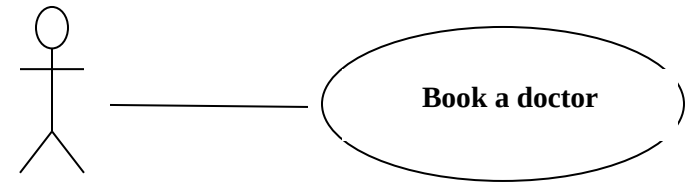
Use case: Register
Diagram:



Scenario:

Register	
Actor who initiates the use case	Patient
Pre-condition	Patient don't have an account
Basic path	By click on register Button on navbar will open the registration form the user type his data in every field and by click on submit button the account is created and user can login now.
Post condition	User register successfully and can login into the system
Alternative Paths	If the system user enters invalid ID or not strong Password or invalid user data, the system shall display an appropriate error message and returns them back to the first in the series of operation
Actor who benefits from the use case	Patient

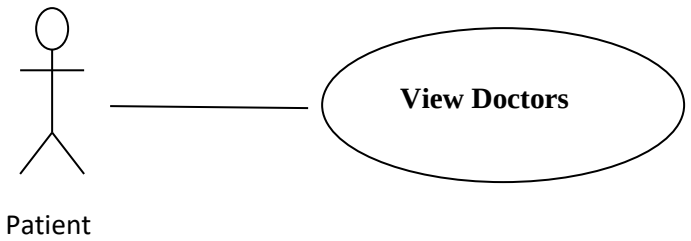
Use case: Book a doctor
Diagram:



Scenario:

Book a doctor	
Actor who initiates the use case	Patient
Pre-condition	Patient have an account and login into the system
Basic path	When user login, he can book a doctor and make an appointment based on specialty
Post condition	Appointment is made
Alternative Paths	If the doctor isn't available the system should display error message
Actor who benefits from the use case	Patient

Use case: View Doctors
Diagram:



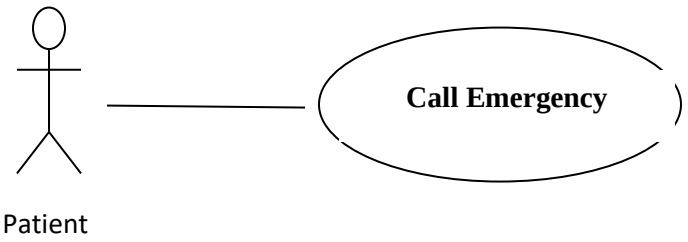
Scenario:

View Doctors	
Actor who initiates the use case	Administrator, Hospital manager

Pre-condition	User had login into his profile (system) and has permission
Basic path	When User login, he can filter and view doctor details like “name, specialty and some another info
Post condition	System User view details about all filtered the doctors
Alternative Paths	If the required doctor is not found, the system should display not found message
Actor who benefits from the use case	Administrator, Hospital manager

Use case: Call Emergency

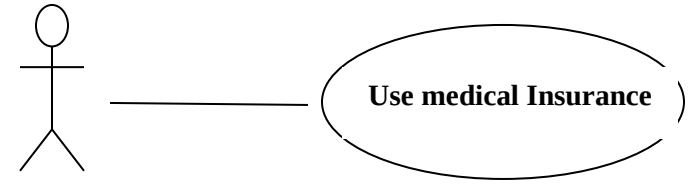
Diagram:



Scenario:

Call Emergency	
Actor who initiates the use case	Patient
Pre-condition	User had login to his profile (system) and has permission
Basic path	When user login, he can call emergency to help him
Post condition	User view number of emergency and call it
Alternative Paths	
Actor who benefits from the use case	Patient

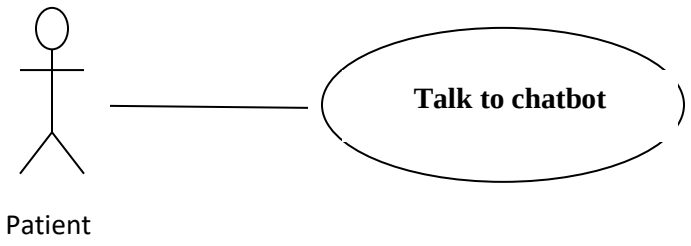
Use case: Use medical Insurance
Diagram:



Scenario:

Use medical Insurance	
Actor who initiates the use case	Patient
Pre-condition	User logged in to his profile (system) and has permission
Basic path	When user login, he can use medical insurance to have a discount on price of medicines
Post condition	User dispends the medicine from the system
Alternative Paths	If patient did not add insurance the system should ask user to add insurance
Actor who benefits from the use case	Patient

Use case: Talk to chatbot
Diagram:



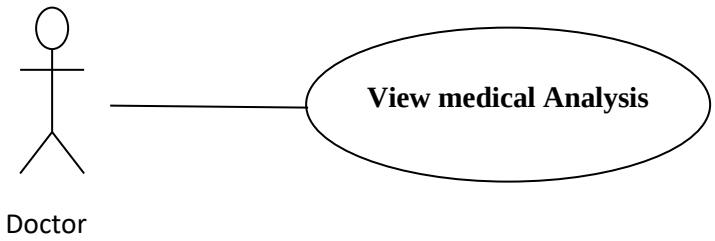
Scenario:

Talk to chatbot

Actor who initiates the use case	Patient
Pre-condition	User logged in to his profile (system) and has permission
Basic path	By click on chatbot icon chat will open and patient can type any question and bot will respond with the answer
Post condition	User talk to chatbot and be guided by it
Alternative Paths	If the user typed a wrong question, the system should ask the user to type another one
Actor who benefits from the use case	Patient

Use case: View medical Analysis

Diagram:



Scenario:

View medical Analysis	
Actor who initiates the use case	Doctor
Pre-condition	User logged in to his profile (system) and has permission
Basic path	When User login, he can view medical analysis details
Post condition	User view details about all filtered the medical analysis
Alternative Paths	If the required medical Analysis is not found, the system should display not found message
Actor who benefits from the use case	Doctor

2.8. Activity Diagrams

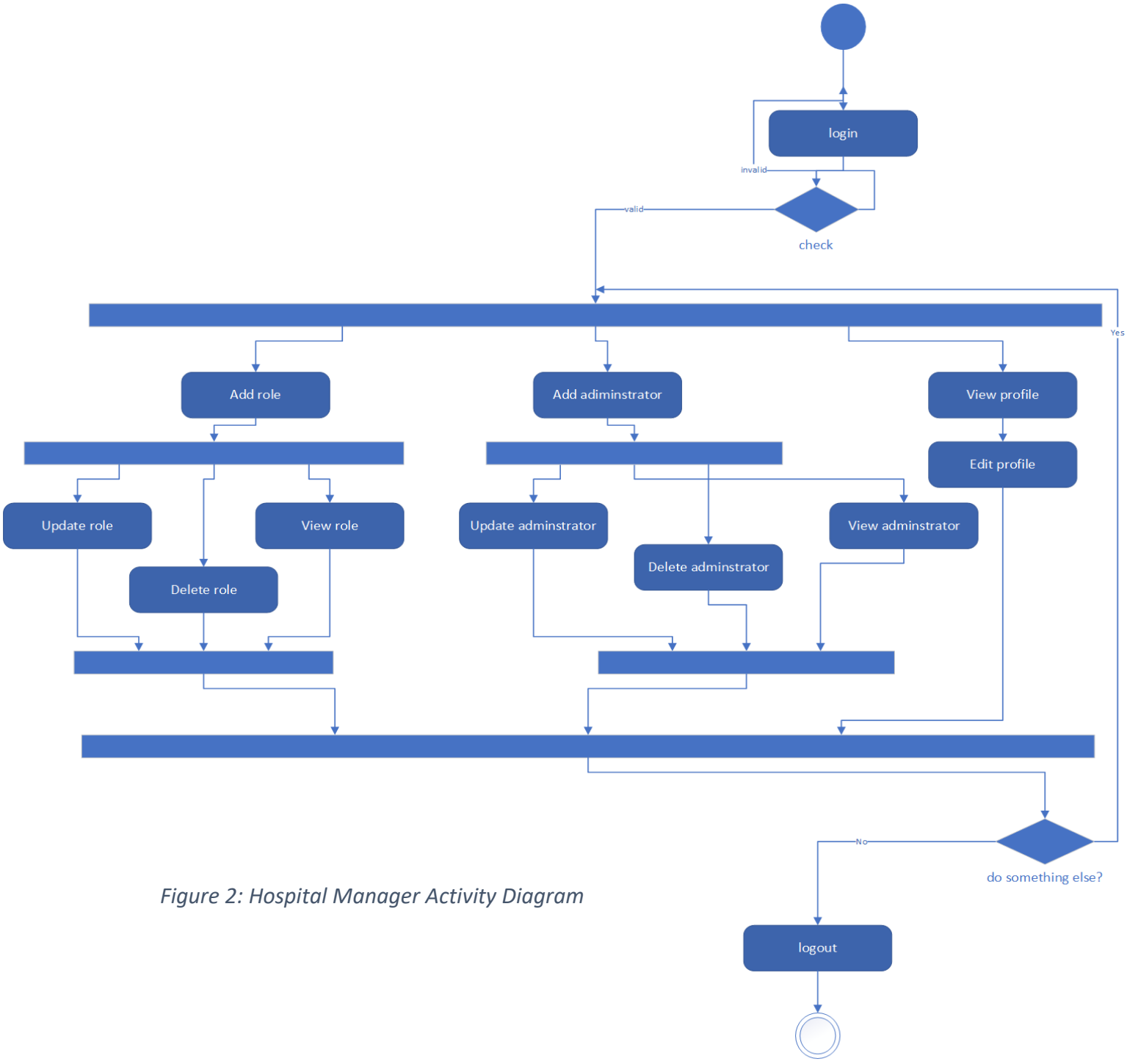


Figure 2: Hospital Manager Activity Diagram

Doctor

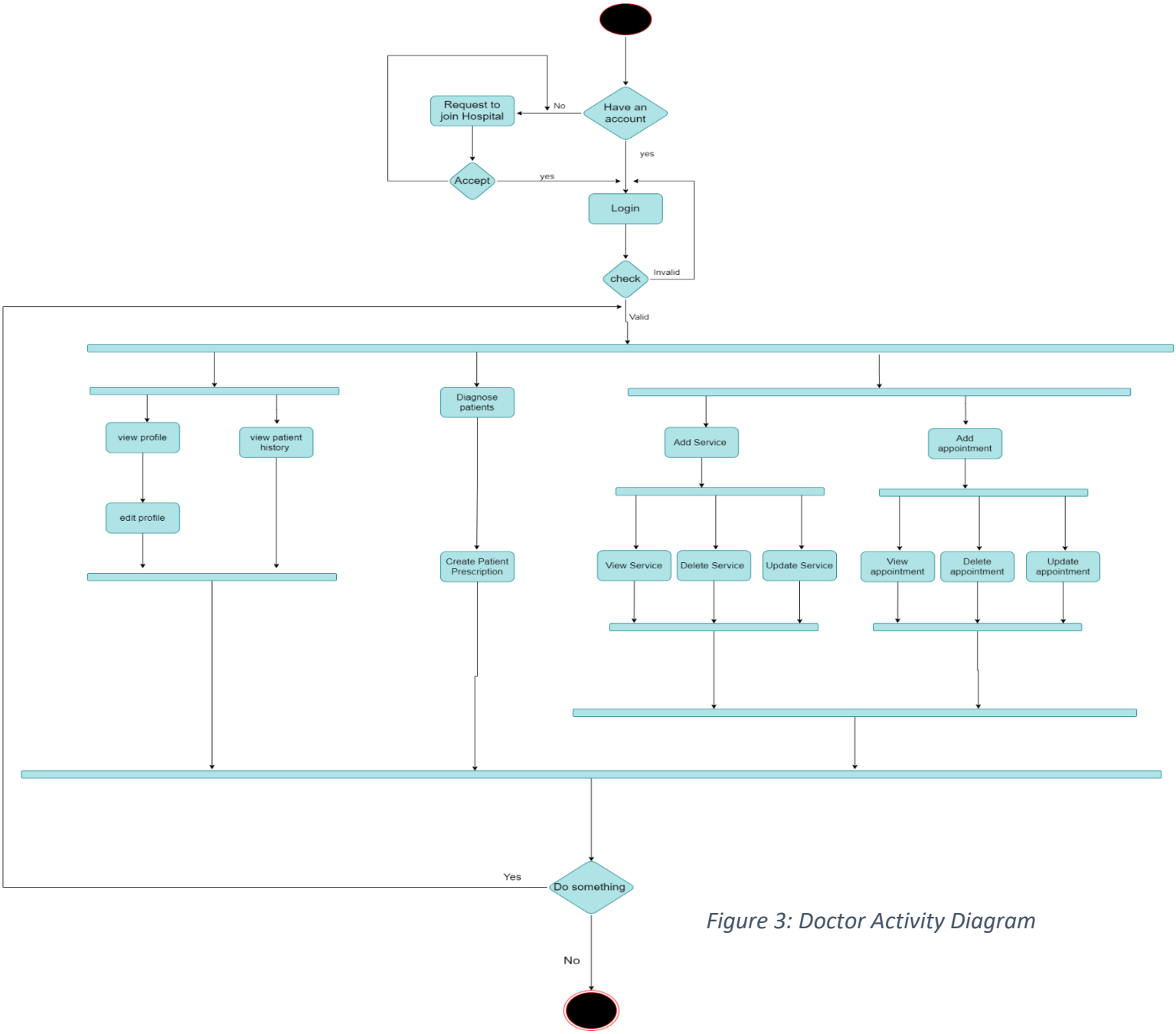


Figure 3: Doctor Activity Diagram

Admin

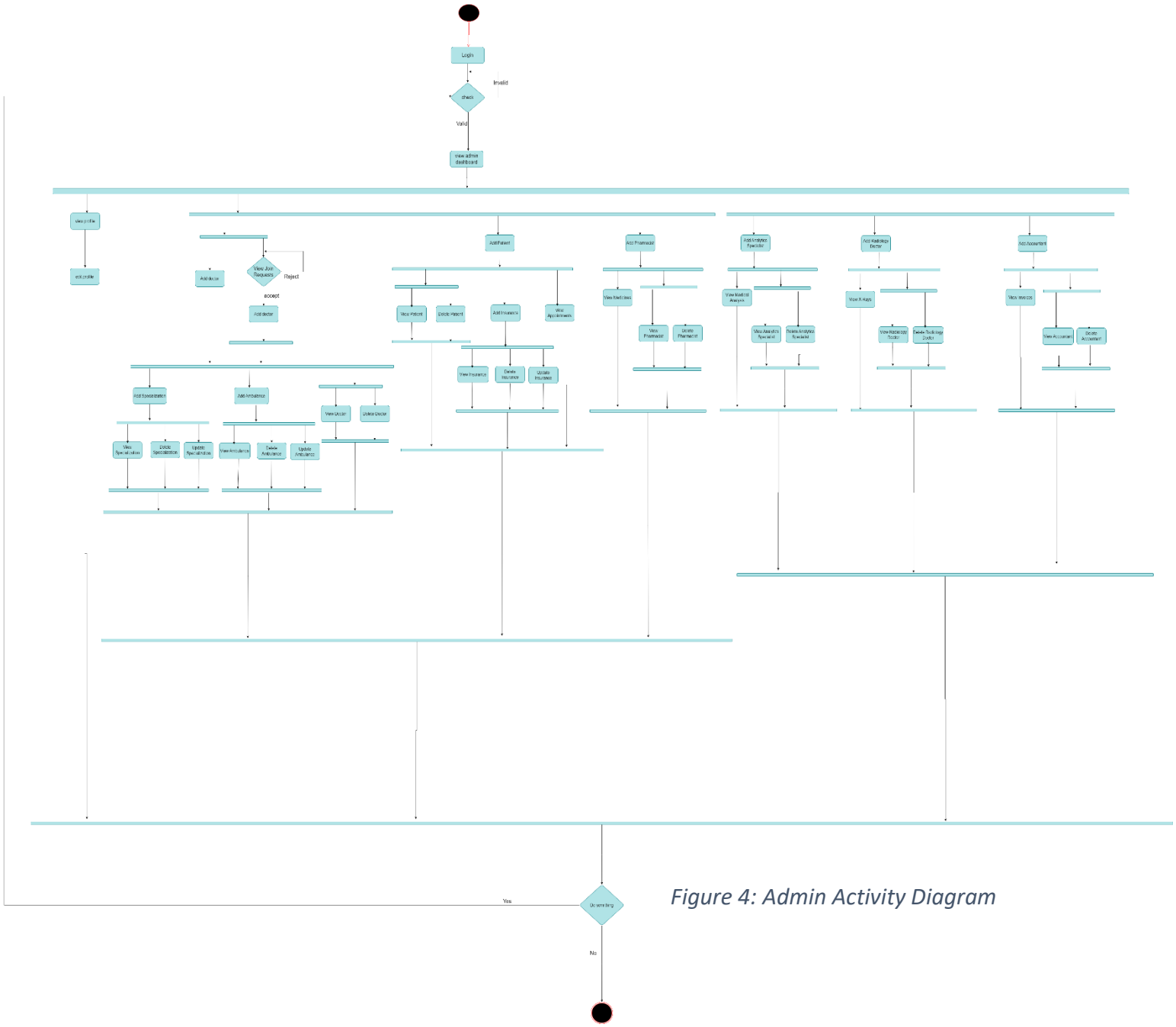


Figure 4: Admin Activity Diagram

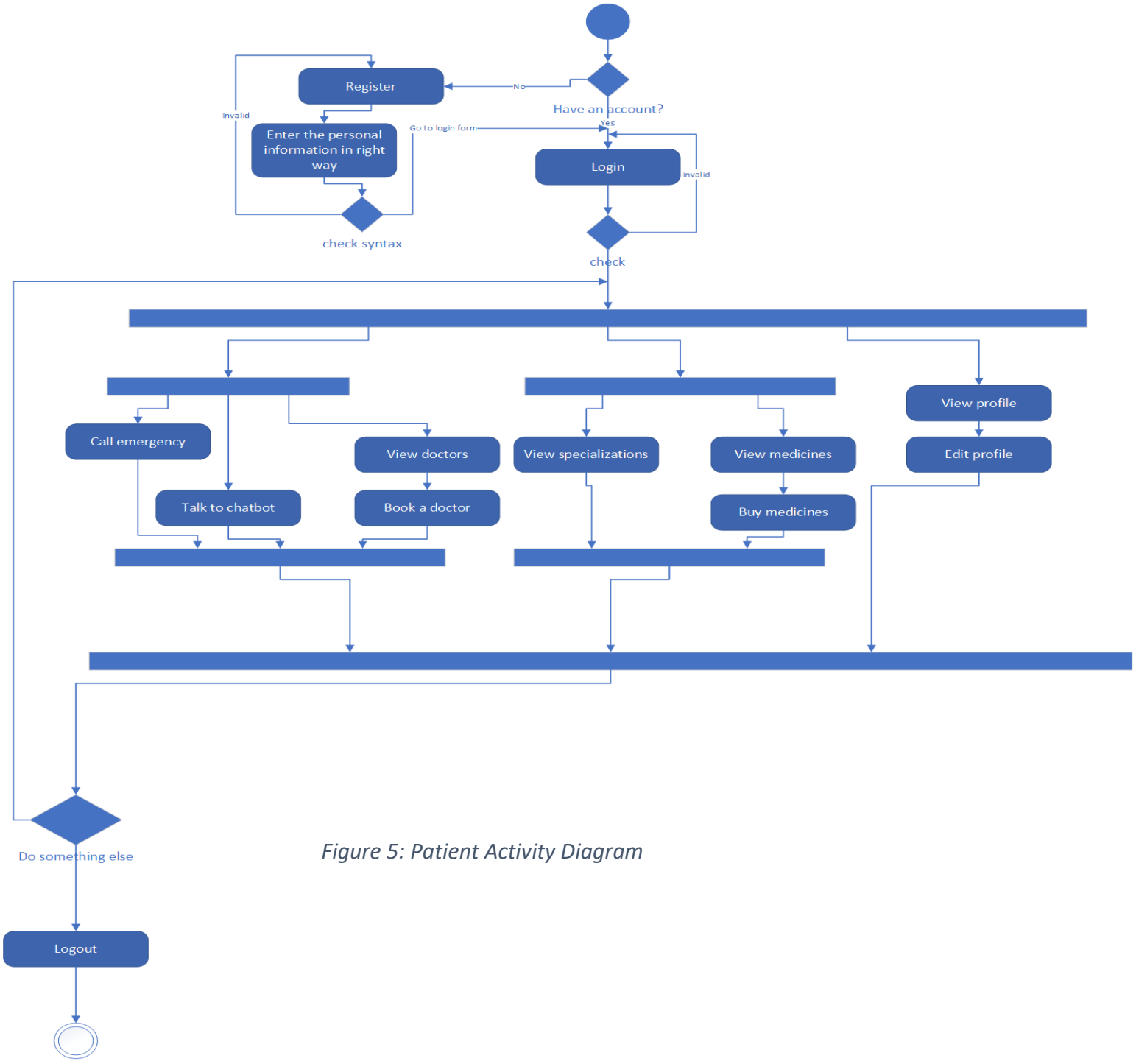


Figure 5: Patient Activity Diagram

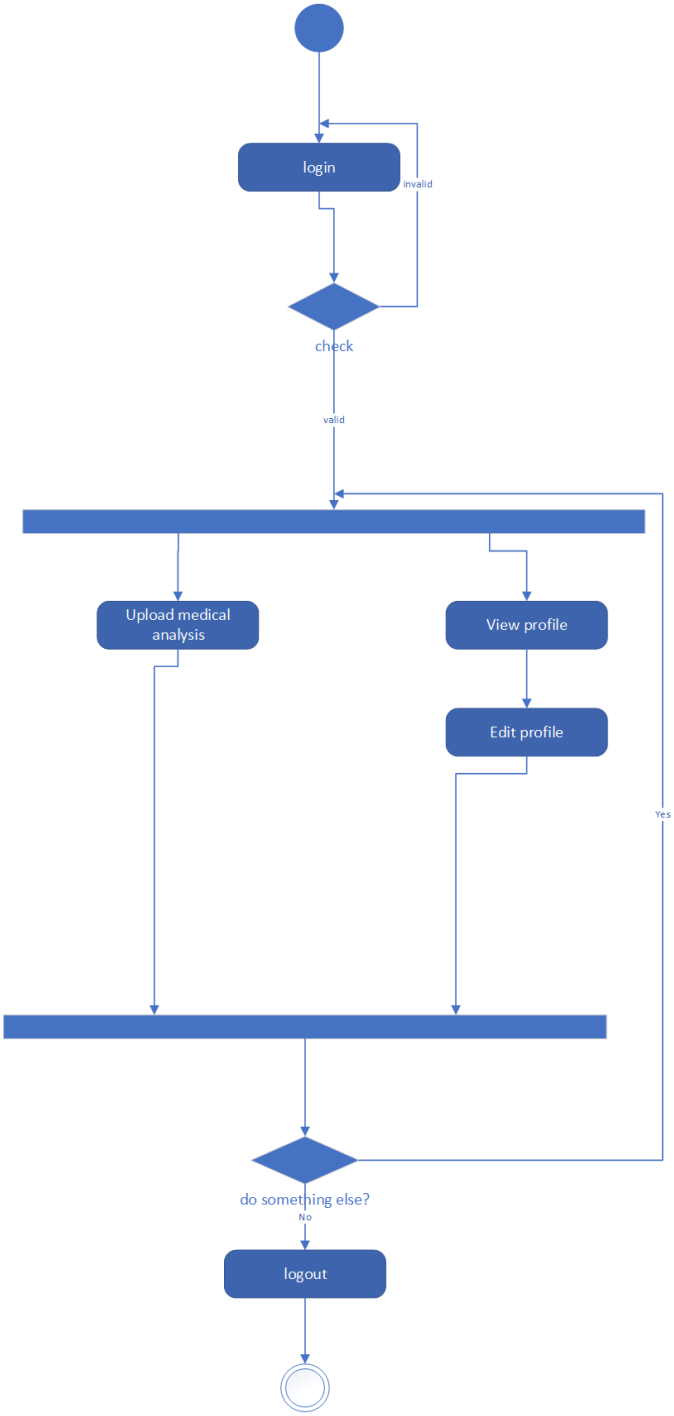


Figure 6: Analysis Specialist Activity Diagram

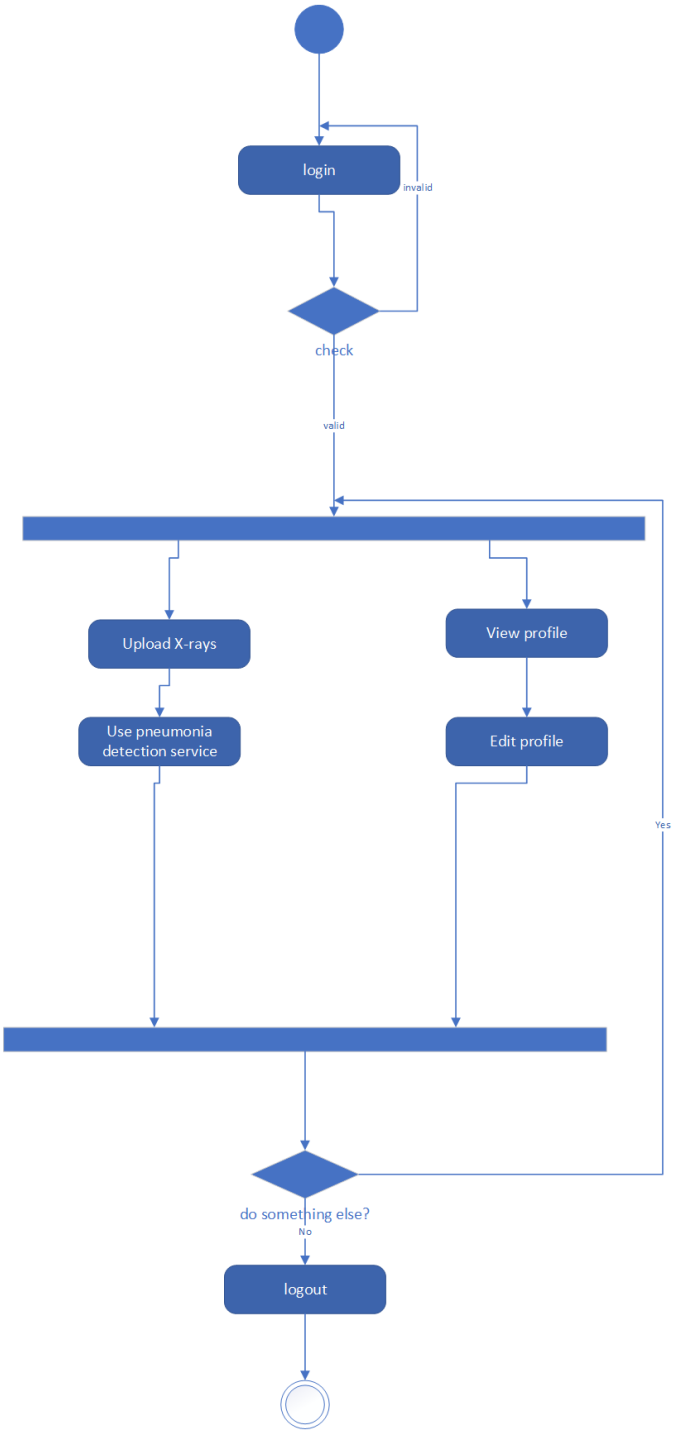


Figure 7: Radiology Doctor Activity Diagram

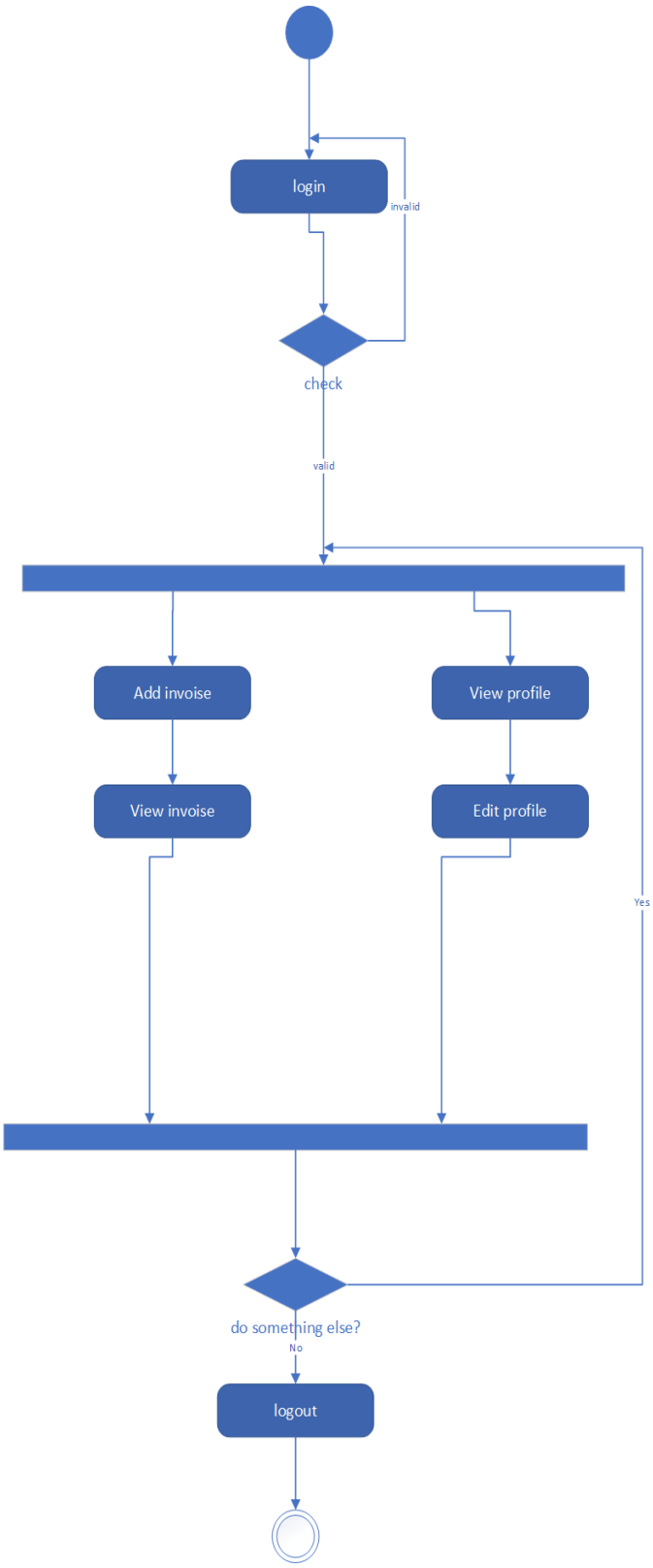


Figure 8: Accountant Activity Diagram

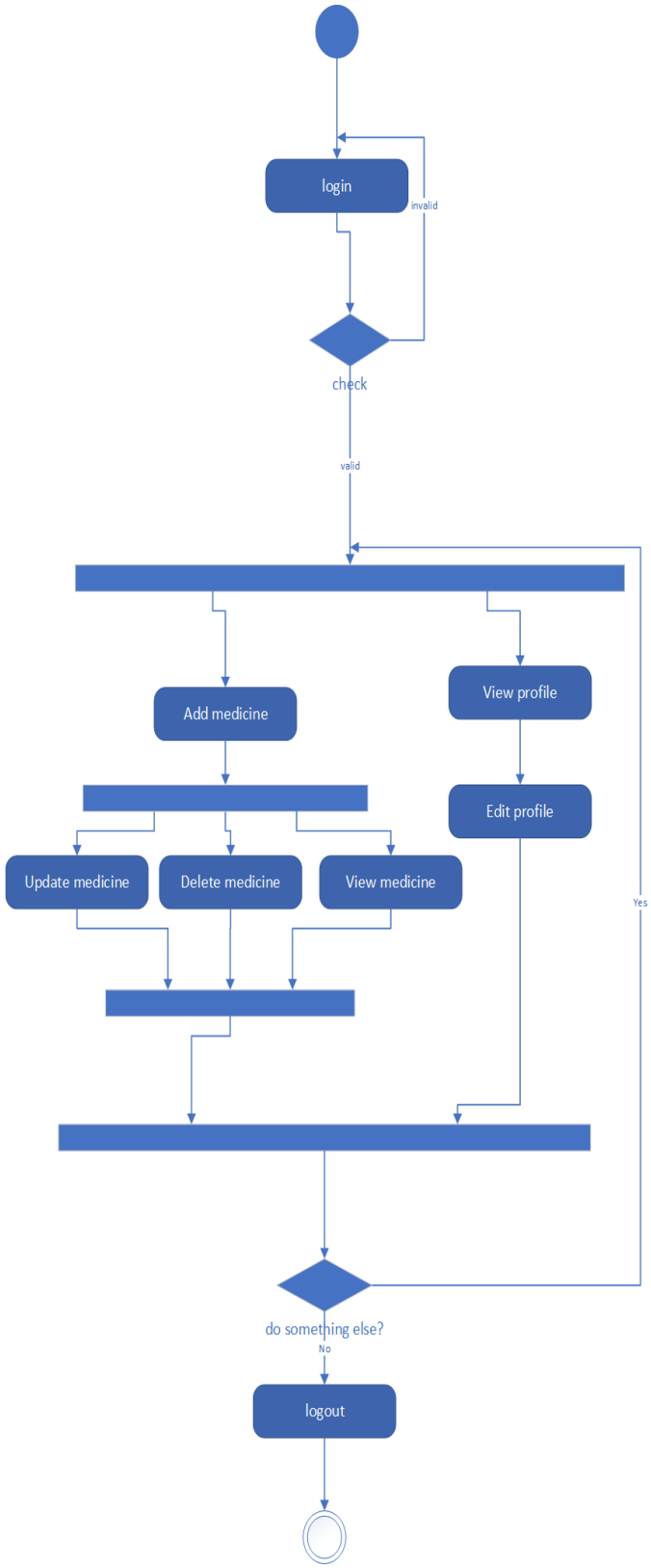


Figure 9: Pharmacy Activity Diagram

Chapter 3: Software Design

3.1. Design of database (Class Diagram)

3.1.1. Version 1: -

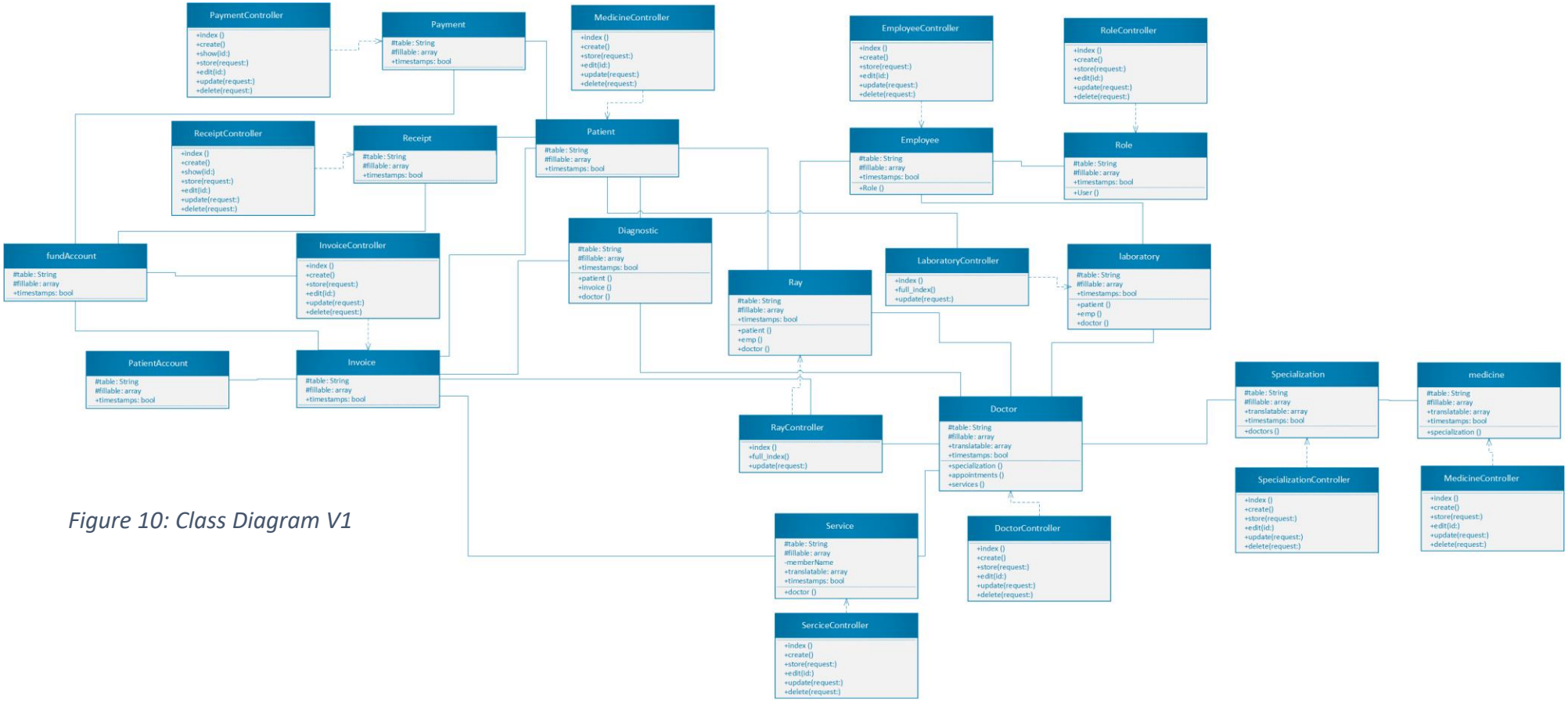


Figure 10: Class Diagram V1

3.1.2. Version 2: -

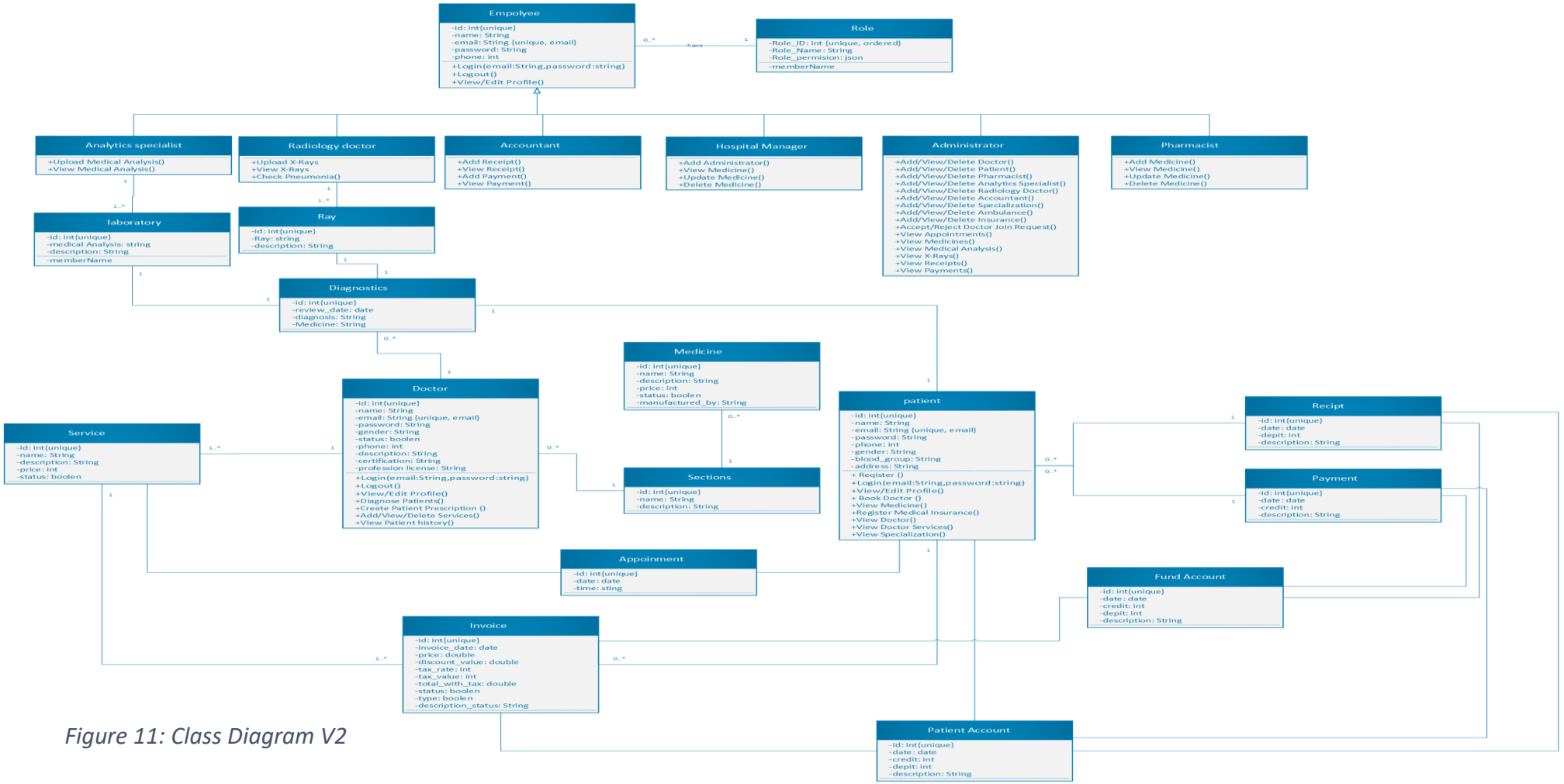


Figure 11: Class Diagram V2

3.2. sequence diagram

3.3. Software architecture

Here you have to mention about both the software in the client and server side. You have also to show the communication between the client and server diagrammatically. Briefly mention the technology used for client/server communication. Also try to explain the objectrelational mappings which facilitate data transfer between client and server.

Chapter 4: Implementation

Chapter 5: Testing

5.1. Unit Testing

Sgsdgsdgsdg

5.2. Integrated testing

Sdsbgsfbdfbdfbdf

5.3. Additional Testing

Sdvgsdgsdgdsgds

Chapter 6: Results and Discussion

6.1. Results

Sdfsgsdgsdsdg

7.1.1. Expected result

Mention what your project supposed to do. What is the theoretical result?. All what you have to mention here as if the project will work 100%

7.1.2. Actual results

This is the actual result what you have achieved from the project.

Chapter 7: Conclusion

Divide your conclusion into 2 paragraphs. First paragraph should include summary of your report including your achievement Second paragraph your recommendation how to enhance the project if you are given the right resources.

Chapter 8: Future work

Appendix

If your report has long codes, calculations, pictures, maps, graphs, illustrations, photographs, survey questionnaires, personal reflections, interviews, and other additional information put this as appendix. The purpose is the body of your report should remain clean