****

**Easy Clinic**

**health care personal system**

**Student :**

Rahmah Oudeh

Balqees Khanfar

Raneem Khalil

*Supervisor:*

Dr.Suhad Daraghmeh

**Table of Contents**

Table of Contents …………………………………………………………………………………………………….. ii

History of Revision........................................................................................................

**1. Introduction ……………………………………………………………………………………..** 1

1.1 Purpose

1.2 Project Scope

1.3 Definitions, acronyms, and abbreviations

1.4 References

**2. Overall Description ………………………………………………………………………..** 2

2.1 Product Perspective

2.2 Product Functions

2.3 User Classes and Characteristics

2.4 Operating Environment

2.5 Design and Implementation Constraints

**3. System Features……………………………………………………………………………….** 3

3.1 System Feature 1

3.2 System Feature 2 (and so on)

**4. External Interface Requirements …………………………………………………** 4

4.1 User Interfaces

4.2 Hardware Interfaces

4.3 Software Interfaces

4.4 Communications Interfaces

**5. Other Nonfunctional Requirements ……………………………………………** 5

5.1 Performance Requirements

5.2 Safety Requirements

5.3 Security Requirements

5.4 Software Quality Attributes

**6. Design Documentation……………………………………………………………………………..6**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

**1-Introduction**

**1.1 Purpose**

The purpose of this document is to describe all the software requirement specification (SRS) for the Easy Clinic. It illustrates the requirements of the system. It will also explain system constraints, interfaces, and interactions with other external applications.

## **1.2 Project Scope**

Due to the large number of patients, medical clinics have a number of problems. The spread of diseases and transmission of infection became more common with the increase in the number of patients in these clinics, and this problem was exacerbated during the Corona pandemic.

The Easy Clinic web-applications give solutions to the problem of waiting period inside the clinic. This system allows the patient to make an appointment and choose a doctor for the appropriate treatment. It may also store all patient information and perform analysis on it. The prescription and patient were entered by the physician. As a result, this saves time and effort, and the tests ensure that it is not lost. The goal of this system is to reduce the incidence of overcrowding within medical clinics even while helping patients make appointments.

**1.3 Definitions, acronyms, and abbreviations**

**Related Requirements:** REQ1, REQ3, REQ4.

**Appointment:** An arrangement to meet someone at a particular time and place.

**Web-based application:** An application that runs on the Internet.

**User:** Patient or doctor.

**Software Requirements Specification:** A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.

**1.4 References**

**1-**<https://stackoverflow.com/?fbclid=IwAR3iJx826ScAqKsxlLfesR3Glt65QclhTFidcco6igQDt29Ys8MzTpRk8x4>

**2**-<https://www.djangoproject.com/?fbclid=IwAR1lvjJDALz5slf0gHKdQ7fJReurlM4lQTb-LehxuVDD1Vx3eGS33_Htd9s>

**3-**<https://www.django-rest-framework.org/?fbclid=IwAR3oN4G_TucHLaOKRJWxCq5UfR_zSgEDJrl2ACVXp-pilVvkERYmSyrx_R0>

**4-**<https://www.w3schools.com/angular/?fbclid=IwAR3NmqqQTLcnEcVA1gYN2n24Vj9OVSJUTwy59Vbp8-uZ8PJuVSG-4F50kH0>

5-<https://angularjs.org/?fbclid=IwAR3nNP0BcEbEsPeQJSJGs0d22BKDlfP5oiKFPHaBirVeTk75U2iOHLdIaLo>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**2-Overall Description**

**2.1 Product Perspective**

This program is a stand-alone system that manages appointment scheduling and also the addition and analysis of patient examinations. The idea was developed out of people's needs and a visit to the Al-Rahma clinic in Nablus to learn what it requires.

This project uses a single database to store all of the data. This web application has various permissions for different users. A unique identifier is assigned to each user.

**2.2 Product Functions**

1. Provide an application that enables patients to book an appointment with any available doctor.
2. Provide an application that allows patients and doctors to save medical examinations and analyze them.
3. Doctors will be able to view appointments.
4. When necessary, the ability to send an email to the user.

**2.3 User Classes and Characteristics**

The main users will be the admin, doctors, and patients. The system is also designed to be simple to use.

* **Admin:** The system should be known to the administrator. The Admin has complete control of the system. His ability to add a doctor is one of his most important functions.
* **Doctors:** The doctor must have a good understanding of how to use the system. The appointments that have been scheduled can be viewed by doctors. Details and information on the patient can also be viewed. The results of patient visits can be included.
* **Patients:** The patient has access to clinic data. He can save, view, and analyze his medical examinations, as well as learn about doctors and

appointments and book an appointment online.

* **Casual users:** The clinic's information is available to anyone. He has access to basic information on doctors and the latest news. He can not, however, make an appointment without first logging in to the website.

**2.4** **Operating Environment**

This software will be used with the Windows 10 Pro edition of Windows. MySQL will be utilized to store patient and doctor information in the database.

* **Operating system**: Windows 10 Pro.
* **Processor**: Intel(R) Core i5.
* **Processor speed**: 2.70 GHz.
* **RAM**: 8GB.

**2.5**  **Design and Implementation Constraints**

We had a difficulty driving between pages while designing the frontend, thus we were limited as a result.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**3-System Features**

**3.1 Login Account**

**3.1.1 Description:**

Users must input login information to access their accounts.

**3.1.2 Stimulus and Response Sequences:**

* The user must first provide his or her login username and password.
* After entering the login information, the system verifies that the login username and password entered are correct.
* If it is valid, it is linked to the user account.
* If the user does not have an account, he or she must create one.

**3.1.3 Functional Requirements:**

**REQ-1:** The system should enable patients and doctors to login.

**REQ-2:** The system should enable the admin to login.

**3.2 Online Appointment**

**3.2.1 Description:**

Patients can schedule appointments online by first choosing a doctor and then selecting a day and time. To schedule an appointment online, the patient must first register or log in.

**3.2.2 Stimulus and Response Sequences:**

* The patient must register or login to the website.
* After logging in, the patient selects the doctor with whom he wishes to schedule an appointment.
* The doctor's availability days are then shown.
* The days of the doctor's availability are then displayed, and the patient selects the day that is most suitable for him.
* The hours that the doctor is available are then displayed, and the patient selects the time that is most suitable for him.
* Following that, the doctor's availability hours are displayed, the patient selects a suitable time for him, and the patient confirms the reservation.
* Confirmation of booking is displayed in case it conflicts with another appointment with the same doctor.

**3.2.3 Functional Requirements:**

**REQ-1:** The system should allow patients to book appointments**.**

**REQ-2:** The system should allow patients to view available doctors.

**REQ-3:** The system should allow patients to modify or cancel their appointments.

**3.3 Admin**

**3.3.1 Description:**

A superuser is an admin. He or she has complete control over the system.

One of the most important jobs for the administration is to add a doctor.

**3.3.2 Stimulus/Response Sequences :**

* The admin logs into the system.
* Following that, he can add and remove doctors, as well as delete doctor appointments.
* When a patient misses an appointment, the admin writes next to his name "absent."
* When the patient's absences reach three, the admin can add a block date for the patient.

**3.3.3 Functional Requirements :**

**REQ-1:** The system should allow the admin to add and delete doctors**.**

**REQ-2:** The system should allow the admin to delete doctor appointments.

**REQ-3:** The system should allow the admin to add the patient's block date.

**REQ-4:** The system should allow the admin to enter a patient's absences.

**3.4 Doctor Module**

**3.4.1 Description:**

The doctor can see the scheduled appointments. Patient details and information can also be viewed. The results of patient visits can be included.

**3.4.2 Stimulus/Response Sequences :**

* The doctor logins to the website.
* Following that, the doctor will be able to see the patient's booking.
* The doctor can view the patient's information and add the visit results.
* The doctor displays the analysis as a graph and can search for any analysis at any time.

**3.4.3 Functional Requirements:**

**REQ-1:** The system should allow the doctor to be able to see the patient's booking.

**REQ-2:** The system should allow the doctor to view the patient's information.

**REQ-3:** The system should allow the doctor to add the visit results.

**3.5 Analyze Medical Examination**

**3.2.1 Description:**

The patient uploads or enters the medical analysis as an image, and the image data is analyzed and displayed in a table by a specific algorithm.

**3.2.2 Stimulus/Response Sequences :**

* The patient uploads or enters the medical analysis as an image.
* A specific algorithm is applied to the image data, which is then analyzed and converted into a table.
* The table can be sorted by name, value, or date.
* In the table, the patient and doctor can look for the analysis. The patient and doctor display the analysis as a graph, such as blood percentage progress.

**3.2.3 Functional Requirements:**

**REQ-1:** The system should enable patients to upload medical analyses.

**REQ-2:** The system should enable patients and doctors to look for the analysis.

**REQ-3:** The patient and doctor display the analysis as a graph.

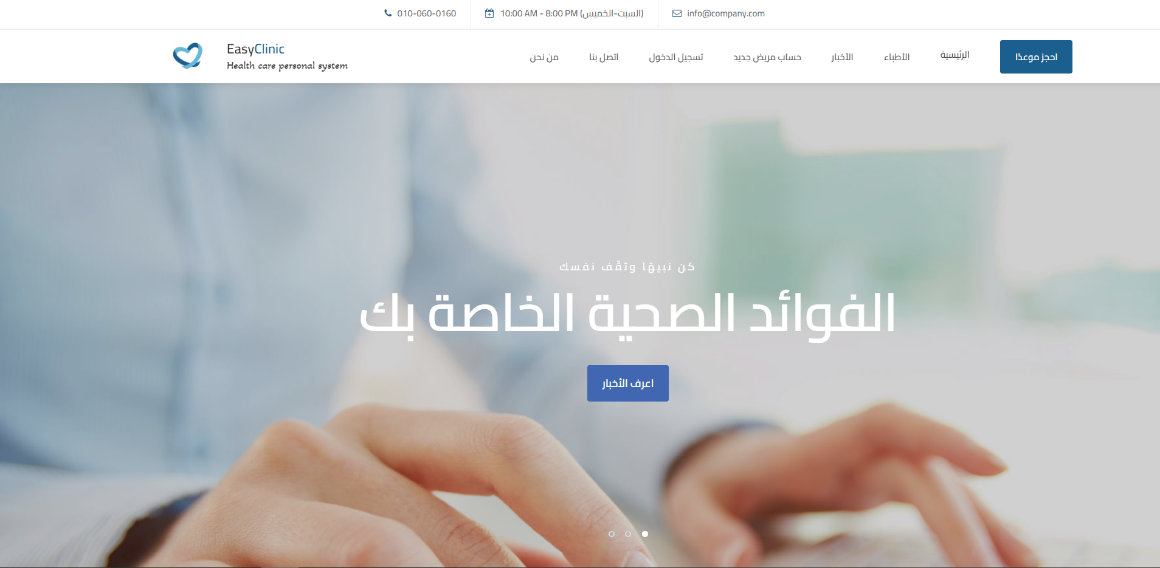
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**4-External Interface Requirements**

This section of the SRS describes the interface requirements for the system. Requirements for user hardware, software, and communication interfaces are defined.

**4.1 User Interfaces**

The user interface is designed using Bootstrap, HTML, CSS, JS and uses framework angularjs . The main page appears when the user first visits a website. To book an appointment, the patient must first log in. He creates a new account if he is not logged in, as illustrated in the accompanying images.







**4.2 Hardware Interfaces**

There is no hardware interface required for this system. The screen, keyboard, mouse, laptop, and mobile phone are the ones used here.

The system should have these hardware requirements:

* **Processor**: Intel(R) Core i5 Windows 10 pro 2.70 GHz.
* **Memory**: 350 GB.

**4.3 Software Interfaces**

The front end is developed with bootstrap, html, CSS, and JS, and the backend is built with Django (python) and MySQL. They are also distinct from one another. The API was used to connect with the backend and the frontend.

**4.4 Communications Interfaces**

Communication is done through the internet and the Localhost server.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**5-Other Non-functional Requirements**

**5.1 Performance Requirements**

The performance of storing information, booking appointments, showing appointments, and displaying visit results is fast and good. And the system must have a good response time.

**5.2 Safety Requirements**

The source code is saved in a secure location, backed up, and can be restored at any time if the code is lost or the device is damaged.

**5.3 Security Requirements**

The security was used in the hash, when a password has been “hashed” it means it has been turned into a scrambled representation of itself. A user’s password is taken and – using a key known to the site – the hash value is derived from the combination of both the password and the key, using a set algorithm. To verify a user’s password is correct it is hashed and the value compared with that stored on record each time they login.

You cannot directly turn a hashed value into the password, but you can work out what the password is if you continually generate hashes from passwords until you find one that matches, a so-called brute-force attack, or similar methods.

**5.4 Software Quality Attributes**

**Flexibility:** The code is flexible because the backend and frontend are separated, making it simple to work with.

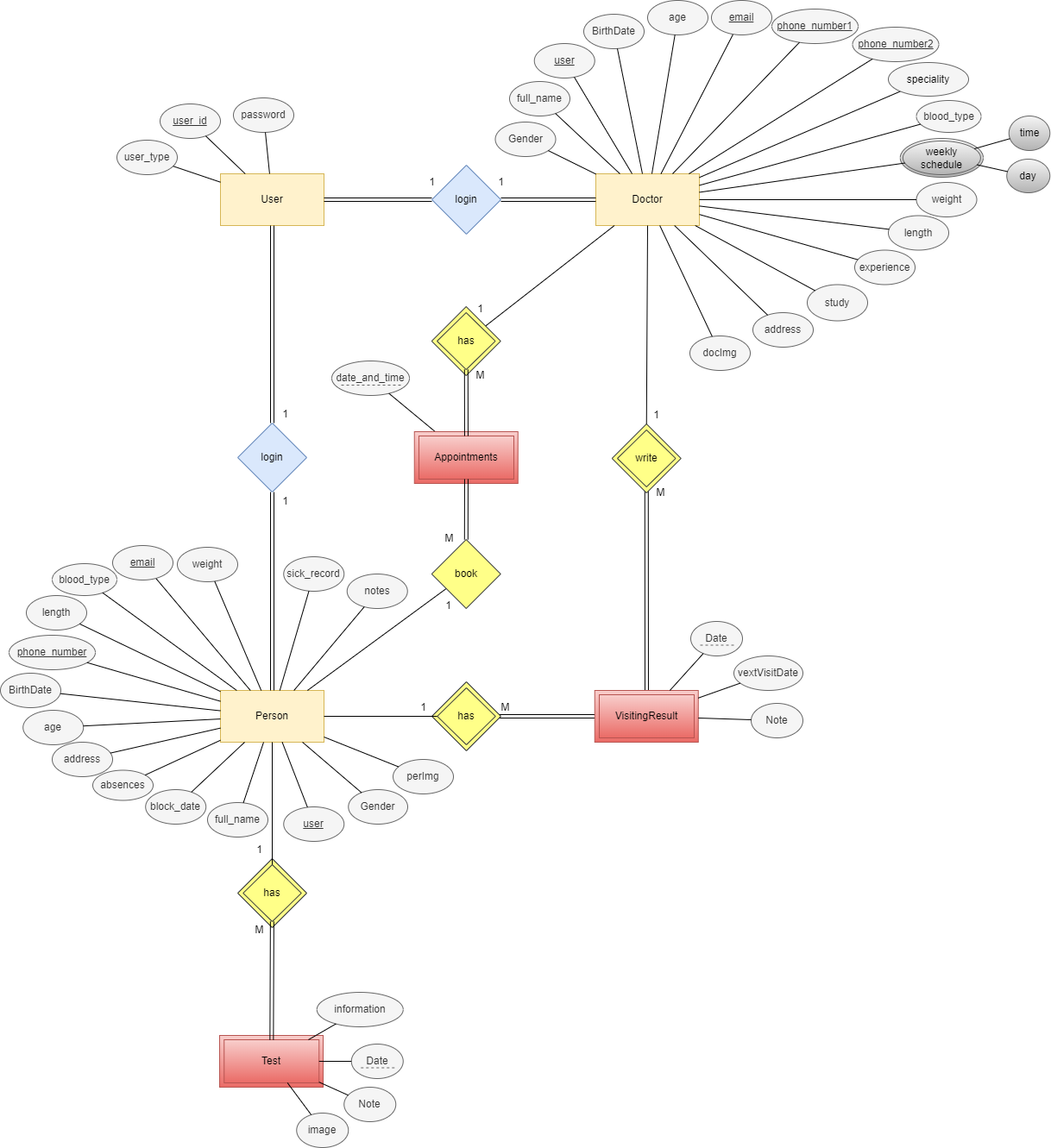
**Portability:** The system can be used on any system or device, and with our source code, we can easily connect the backend to multiple frontends, such as a mobile application.

**Availability:** The user can access the data at any time and from any location.

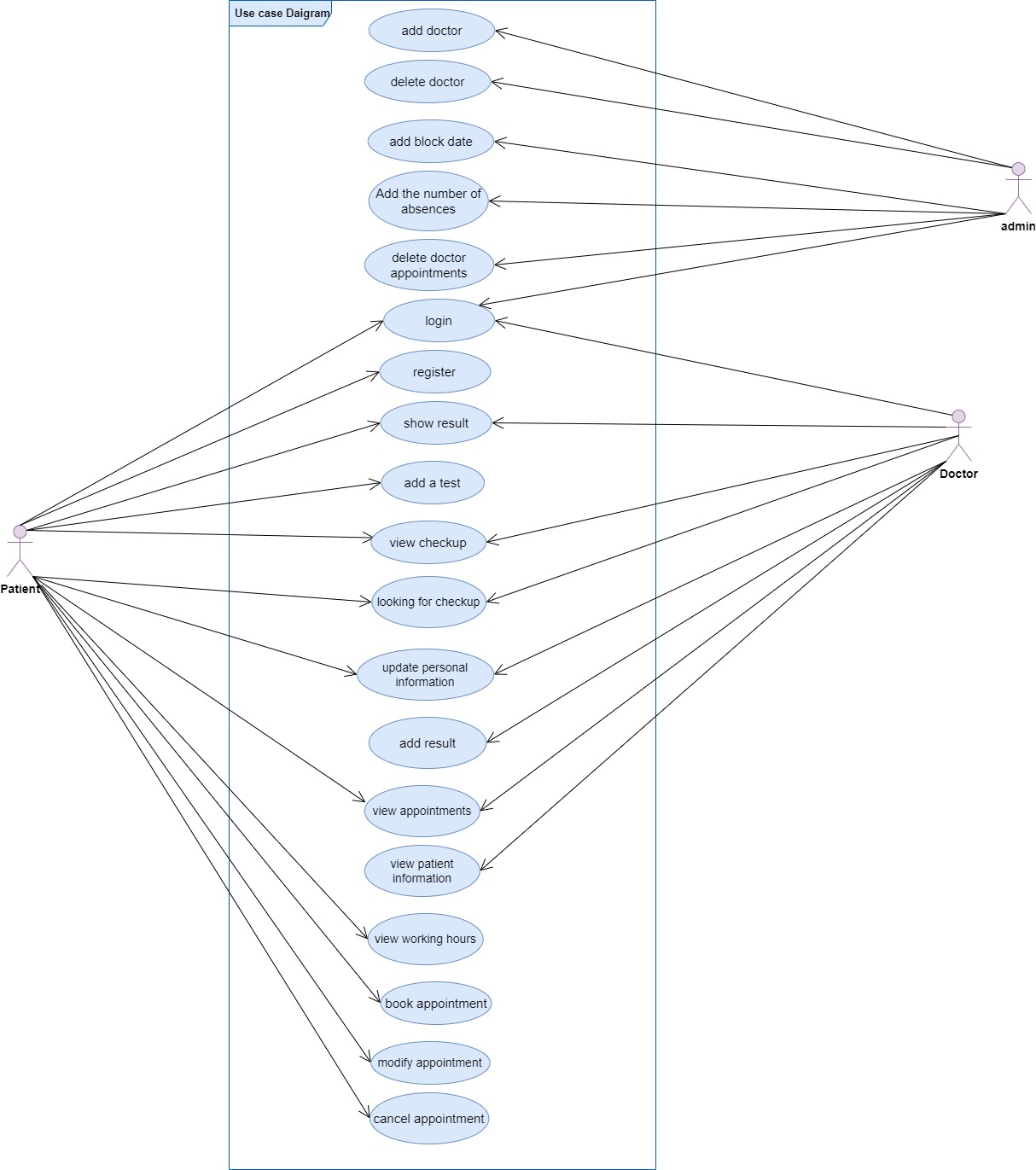
**Maintainability:** The system is easy to maintain because the backend and frontend interfaces are separate and can be changed independently of each other without affecting each other.

**6-Design Documentation**

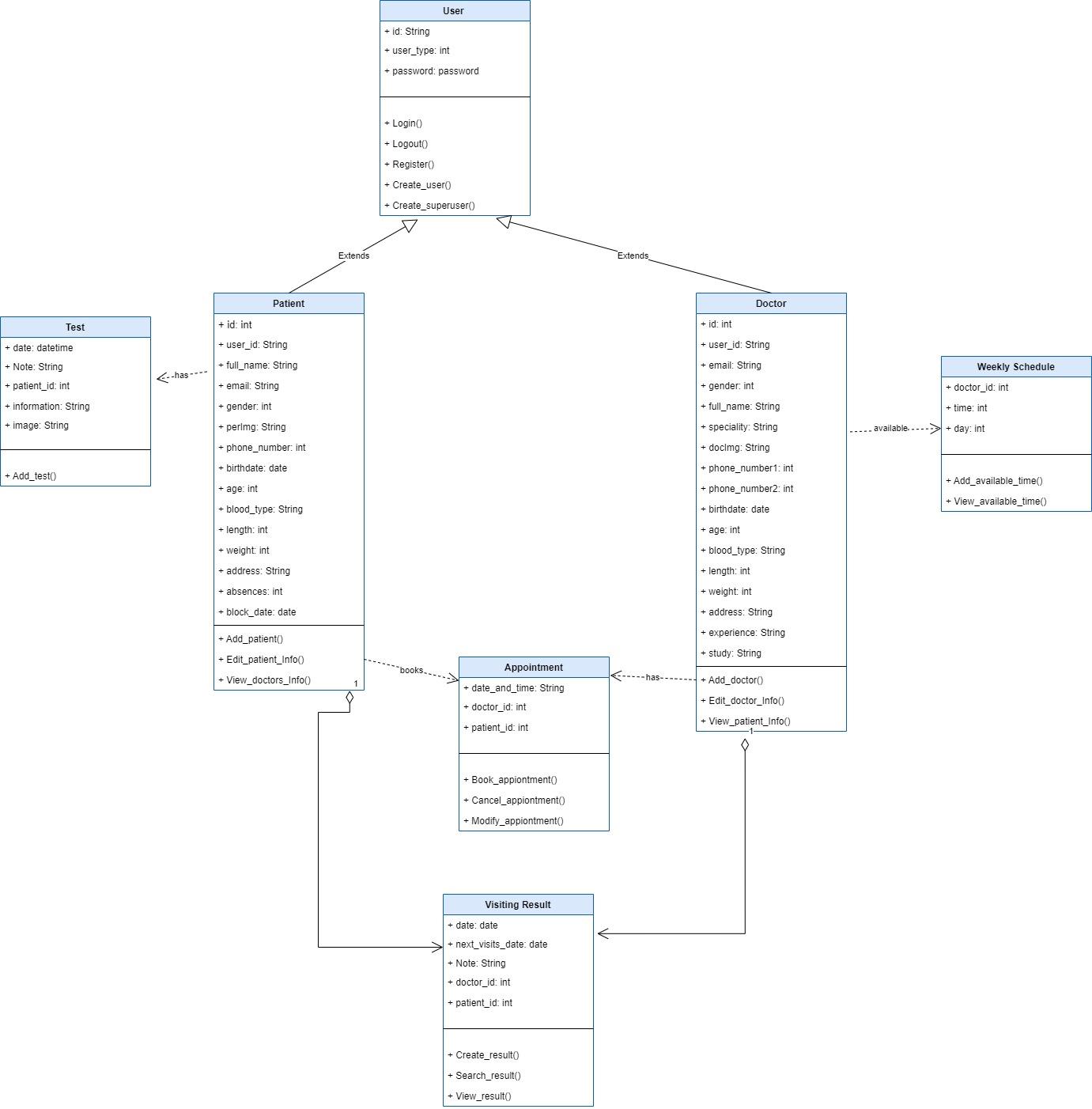
**1- ER-Diagram**

****

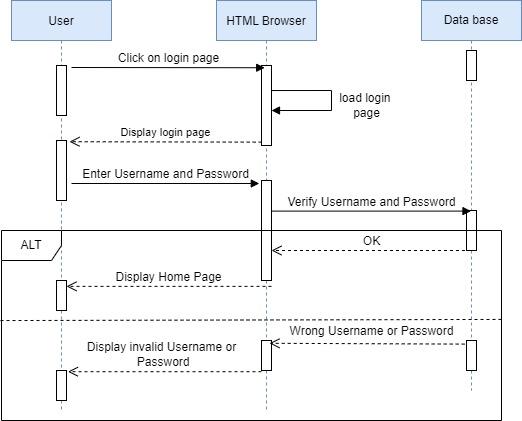
**2- Use Case Diagram**

****

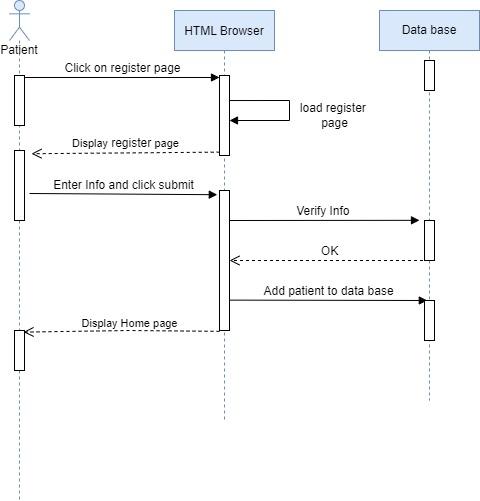
**3- Class Diagram**

****

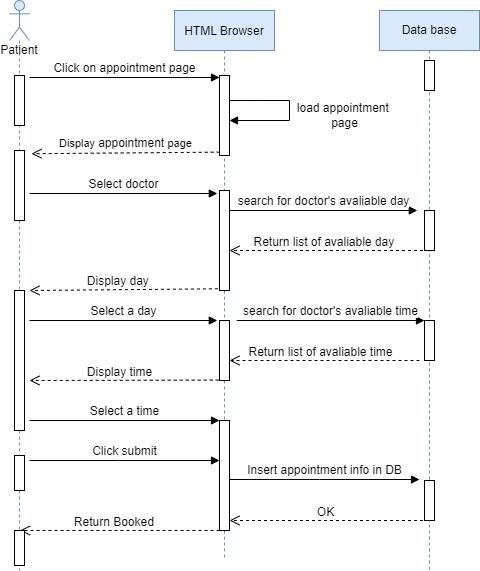
**4- Sequence Diagram on login**

****

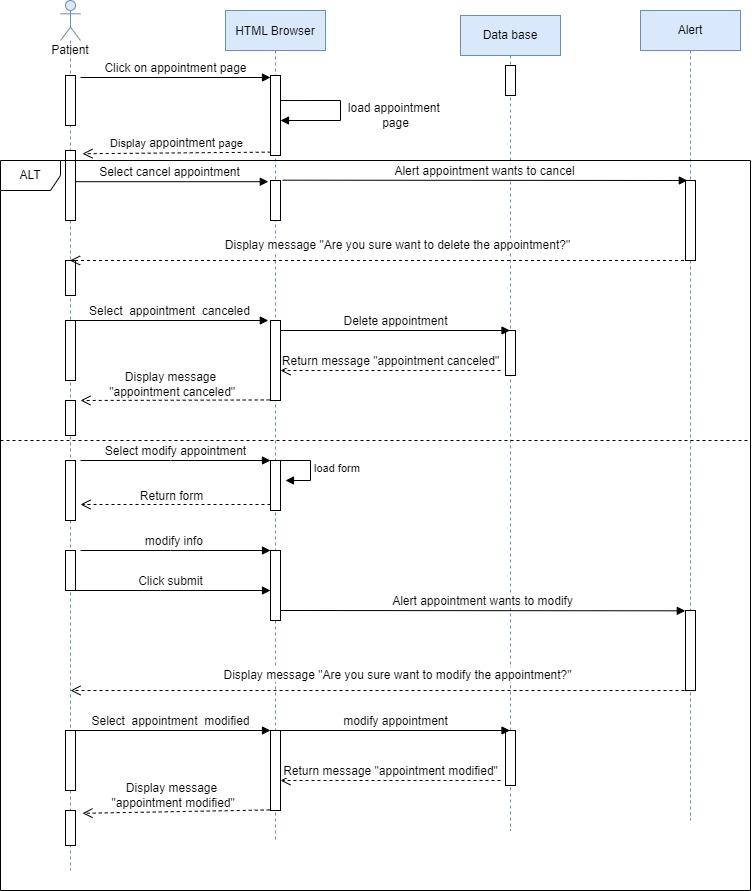
**5-Sequence Diagram on registration**

****

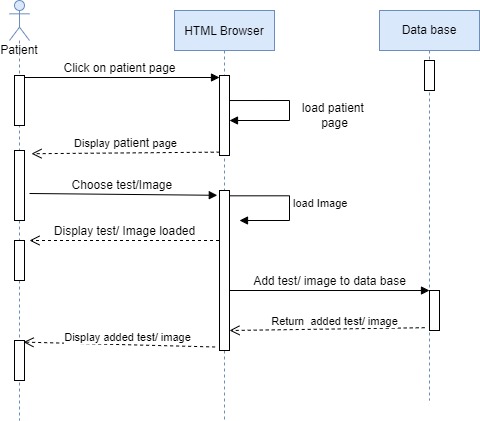
**6- Sequence Diagram on book appointment**

****

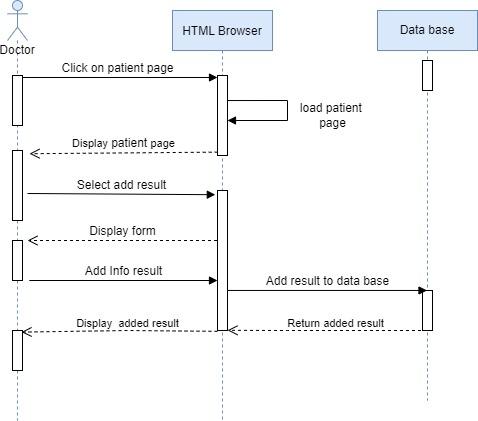
**7- Sequence Diagram on cancel and modify appointment**

****

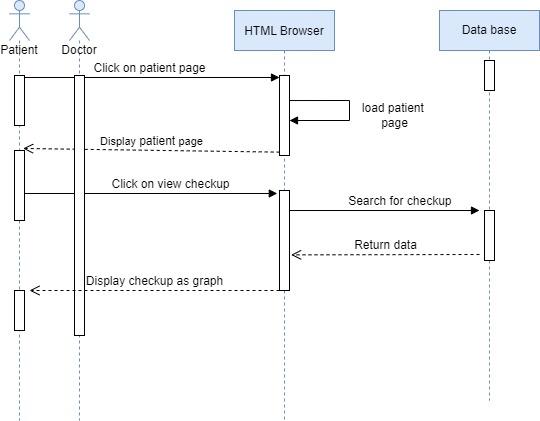
**8- Sequence Diagram on add test**

****

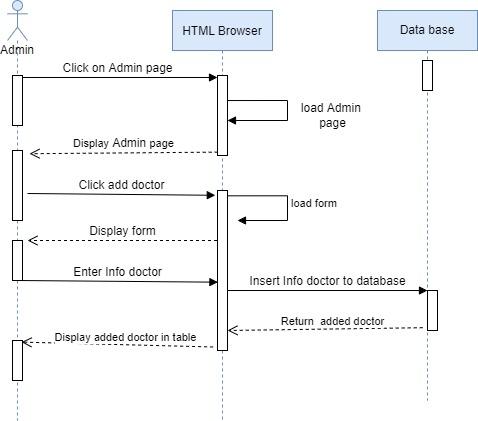
**9- Sequence Diagram on add result**

****

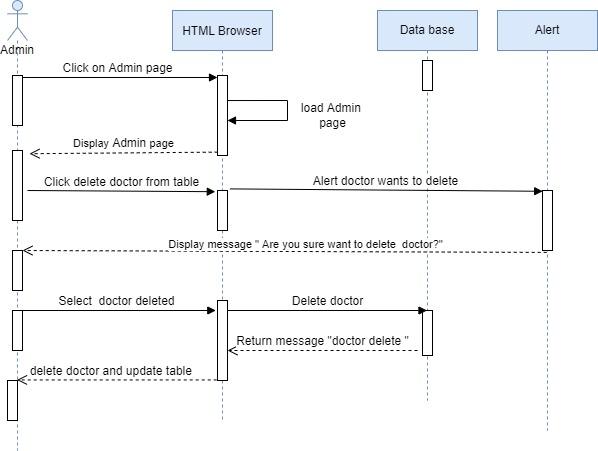
**10-Sequence Diagram on view checkup**

****

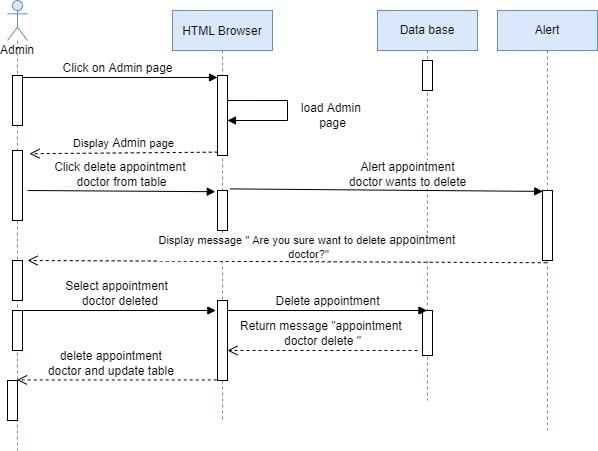
**11-Sequence Diagram on add doctor**

****

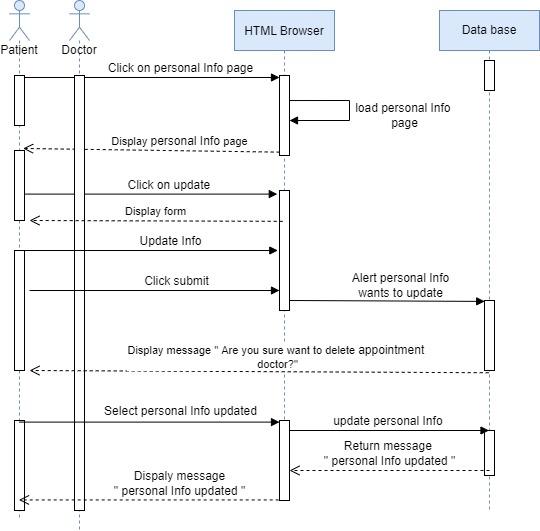
**12-Sequence Diagram on delete doctor**

****

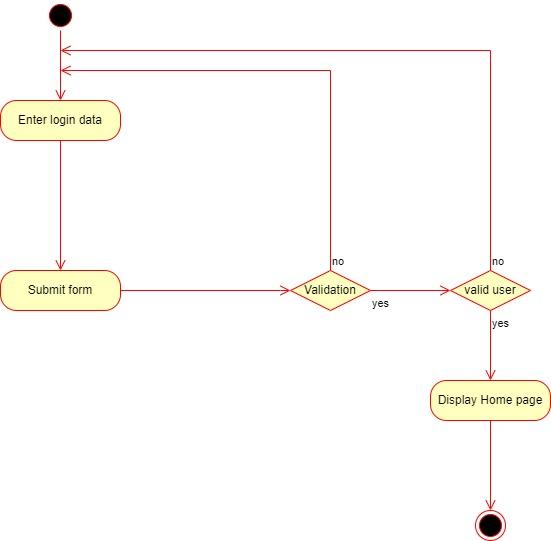
**13- Sequence Diagram on delete appointment doctor**

****

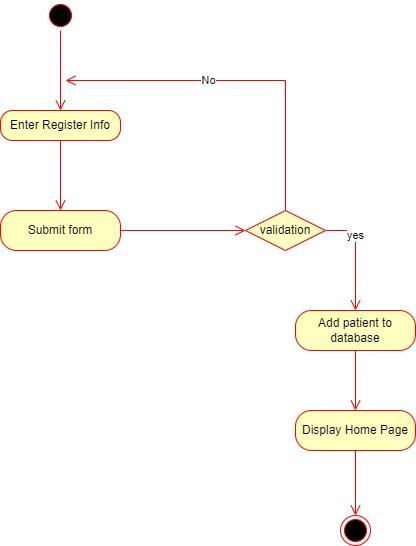
**14- Sequence Diagram on update information**

****

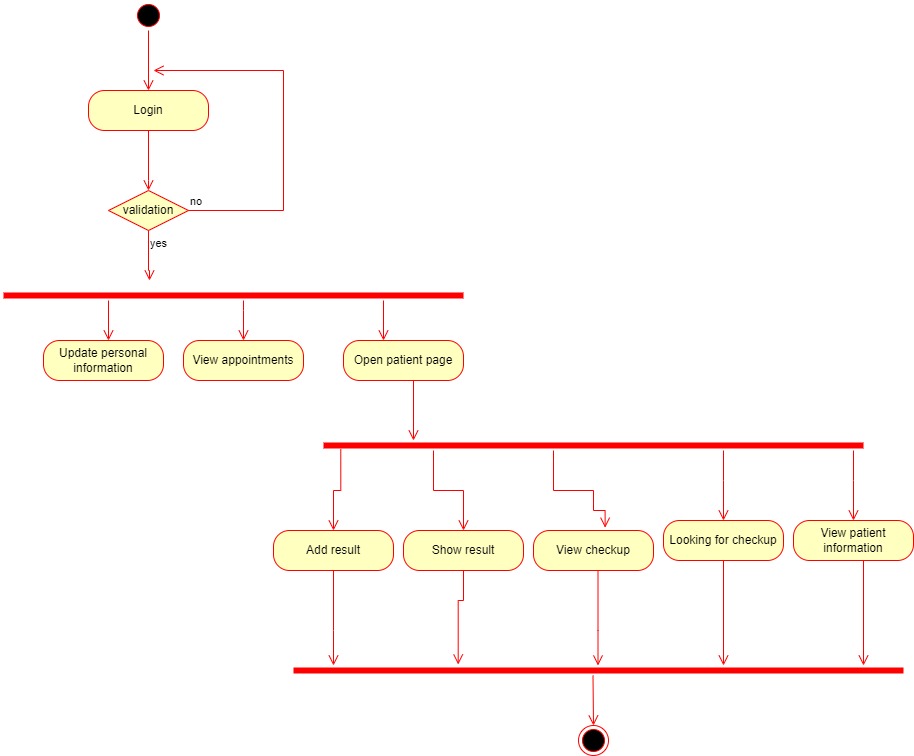
**15-Activity Diagram on login**

****

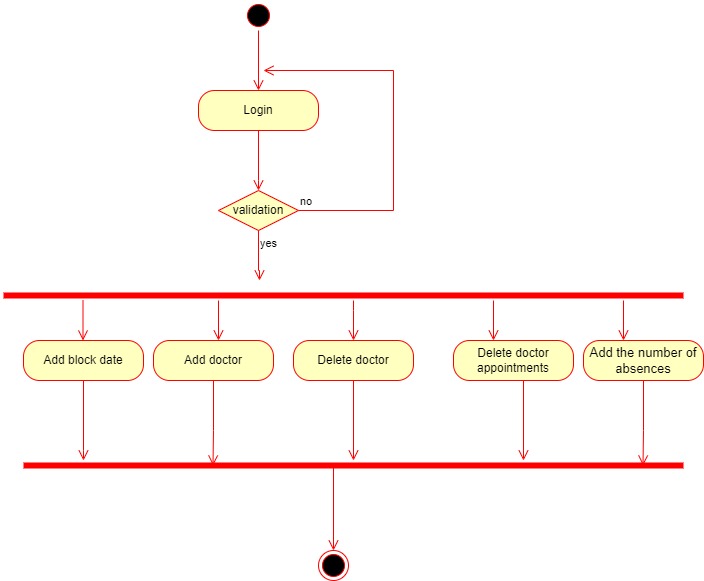
**16-Activity Diagram on registration**

****

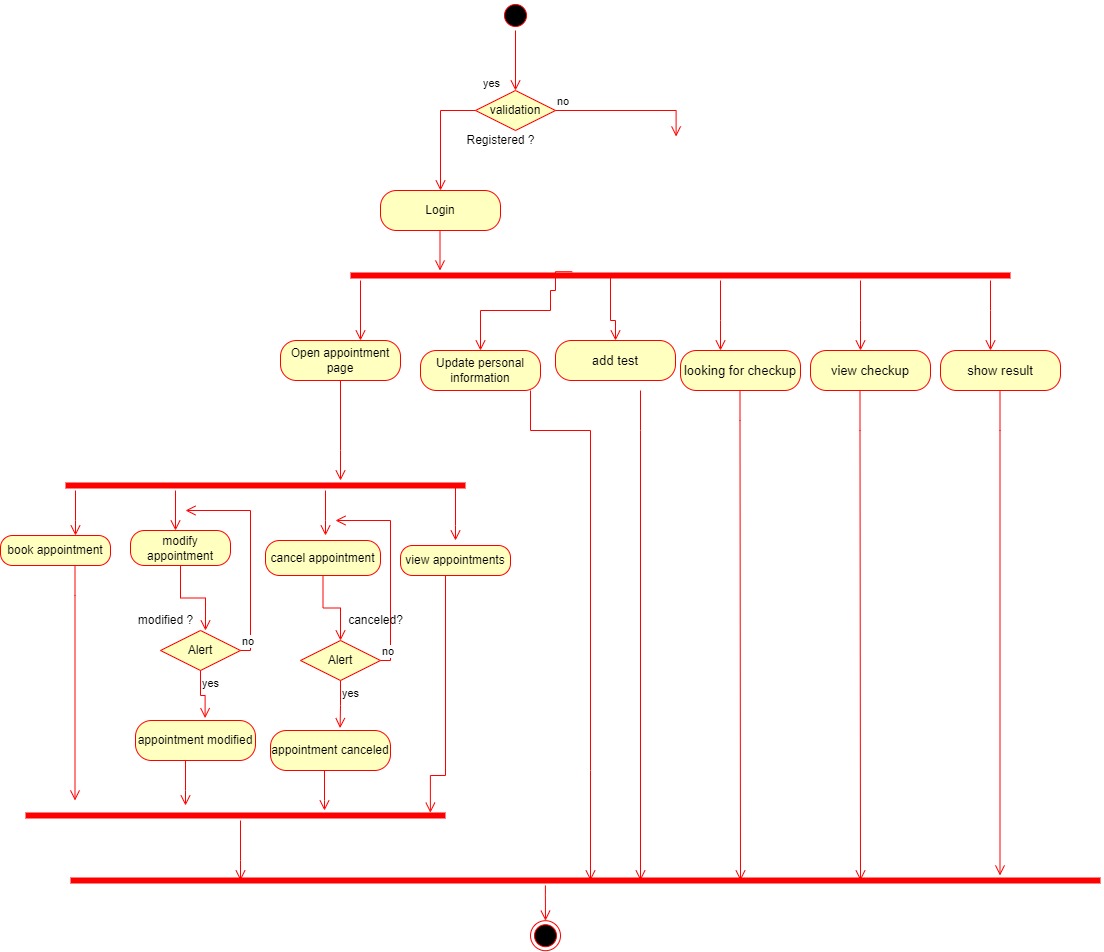
**17- Doctor’s Stimulus and Response Activity**

****

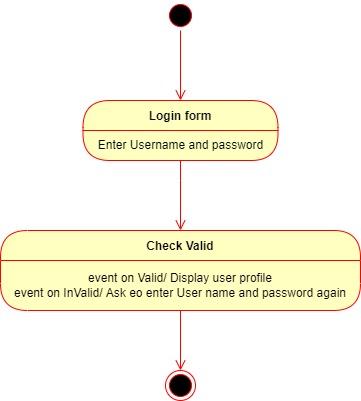
**18- Doctor’s Stimulus and Response Activity**

****

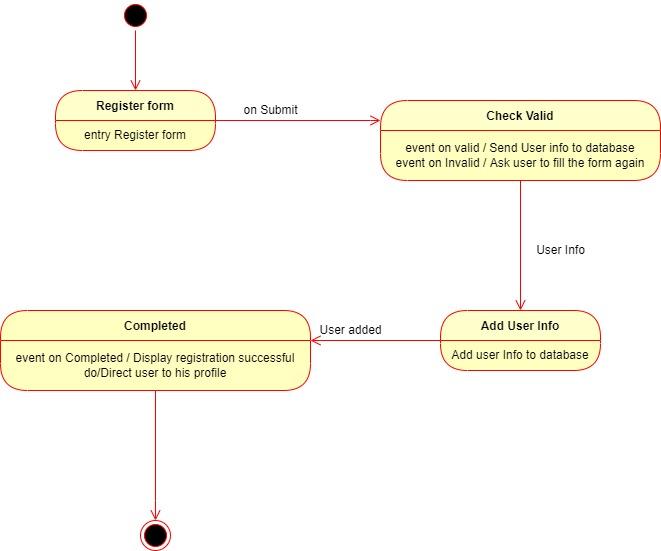
**19- Patient’s Stimulus and Response Activity**

****

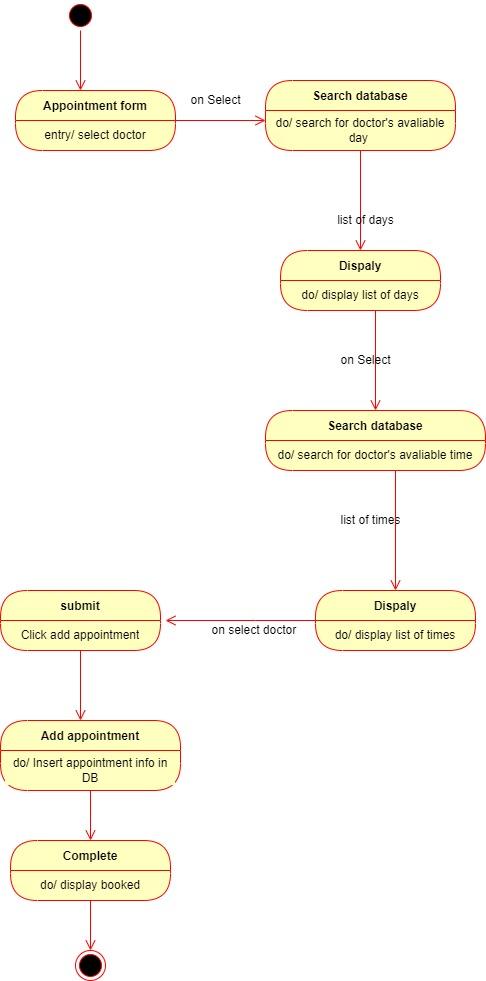
**20- State Diagram on login**

****

**21- State Diagram on registration**

****

**22-State Diagram on book appointment**

****