

Software design document

Sandra Fares, Nour Ghoniem, Samiha Hesham, Mariam Hesham

May 11, 2020

SDD Version	Date	Reason for Change
1.0	12-April-2020	SDD First version's specifications are defined
1.1	20-April-2020	Scope updated

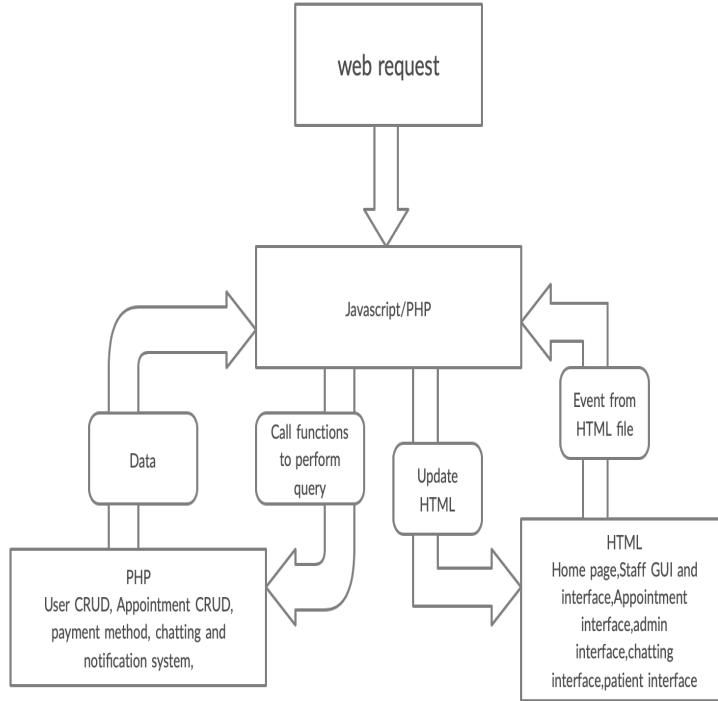
Table 1: Document version history

GitHub: <https://github.com/SW-Project/ClinicSystem/blob/master/SDD>

1 Introduction

1.1 Overview

The proposed software system will be made for a knee and orthopedic clinic. The system should be used by the clinic staff with all their roles. Also by the patients that will interact with the clinic. The system should make appointments to be easy for simple users. Also the system will manage the clinic's users, appointments, operations, notifications and reports.



1.2 Scope

The scope of this document is to talk about the technicality of this software system. It will be an application for Dr ahmed ghoniem clinic. This system will be designed to help the clinic to have an easier system of appointment. A full database instead of paper usage and more faster way to manage the appointments ,generating reports and adding operations if necessary. More over, the system is designed to help the doctors to know the medical history of the patients and to access their reports easily .The system must be build in incremental model and agile model.

1.3 Purpose

The purpose of software design document is to present detailed description of project system architecture and design. The document will explain the features of the clinic management system. Also, this document will help the project team and the customer to have a full overview about the interface and the functions of the project. The customer can review this document to be aware of the requirements that is finished.

1.4 Intended audience

Mainly we have five stakeholders:

- Admin
- Receptionists
- Doctor
- Assistants
- Patients

1.5 Abbreviations

- SMS = Short Message Service
- ER=Entity Relationship diagram
- Tb=table
- Rtb=relation table
- API=application program interface

2 Project Overview

The outcome of this project is a Clinic software product that is used to assist patients in booking appointments effortlessly. This software collects data about the patients and their medical history, which helps the Doctor to get a better view of the Patients' case and view their medical history easily any time necessary. Therefore, offering accurate Diagnosis in less time and with less confusion. Moreover, this product shall offer better communication with the Dr and his Patients. The Doctor can write reports for his patients and keep them updated. On the other hand, Patients also have the ability to reach out to the Doctor's assistant using a chatting system, learning more about their case and the operations scheduled, if any.

2.1 Project Scope

The project's target is to help Patients book appointments and assist the Dr to view his patients' data effortlessly. This target shall be reached using certain methods. These methods include offering the patient an easy way to fill his information and book his appointments in any time suitable for him without reaching out to receptionists. On the other hand, the receptionists shall be automatically notified when a patient books an appointment. Moreover the patients' data, including their medical history, shall be saved for the Dr to view easily any time. API tools shall also be used to offer a chatting system for the Patients and the Doctor's assistant to communicate. By successfully building these methods and making them function well, there shall be no more confusion with retrieving Patients' data and booking appointments. The patients and the Dr shall be updated with any new progress shortly and effortlessly.

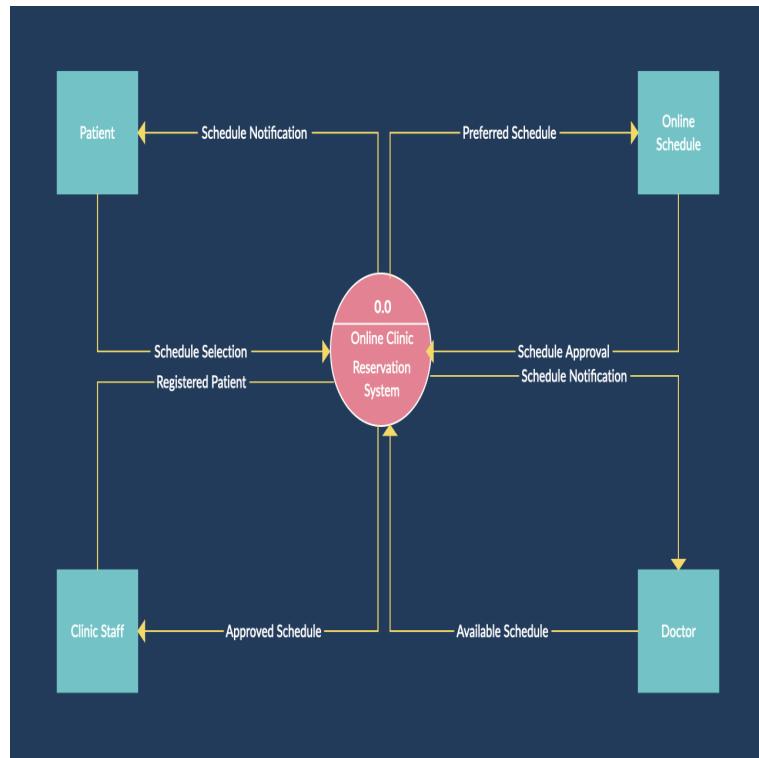
2.2 Goals and objectives

1. The Patients shall be able to book appointments effortlessly.
2. Receptionists and the Doctor shall be able to view appointments.
3. The Patients shall be able to contact the Doctor's assistant through a chatting system.
4. The Doctor shall be able to view Patients' medical history.
5. The Doctor shall be able to add Reports concerning his Patients' cases.
6. The Doctor shall be able to add an operation.
7. The Doctor's assistant shall be able to schedule operations and contact patients.

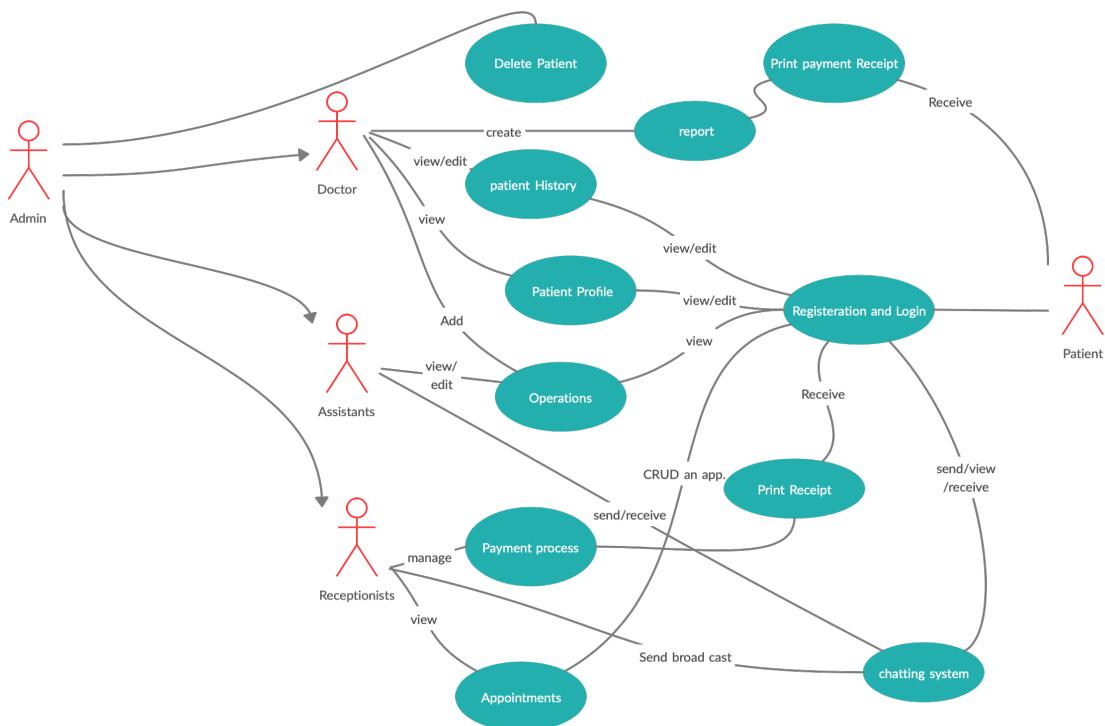
8. The admin shall be able to add staff, view appointments and manage everything concerning the system.
9. The Patients shall be able to register into the system.
10. The Patients and The Staff shall be able to login into the system, if verified, and shall be forwarded to their main pages.

3 Context viewpoint

3.1 Context Diagram



3.2 Use case scenario



4 System Architecture Design

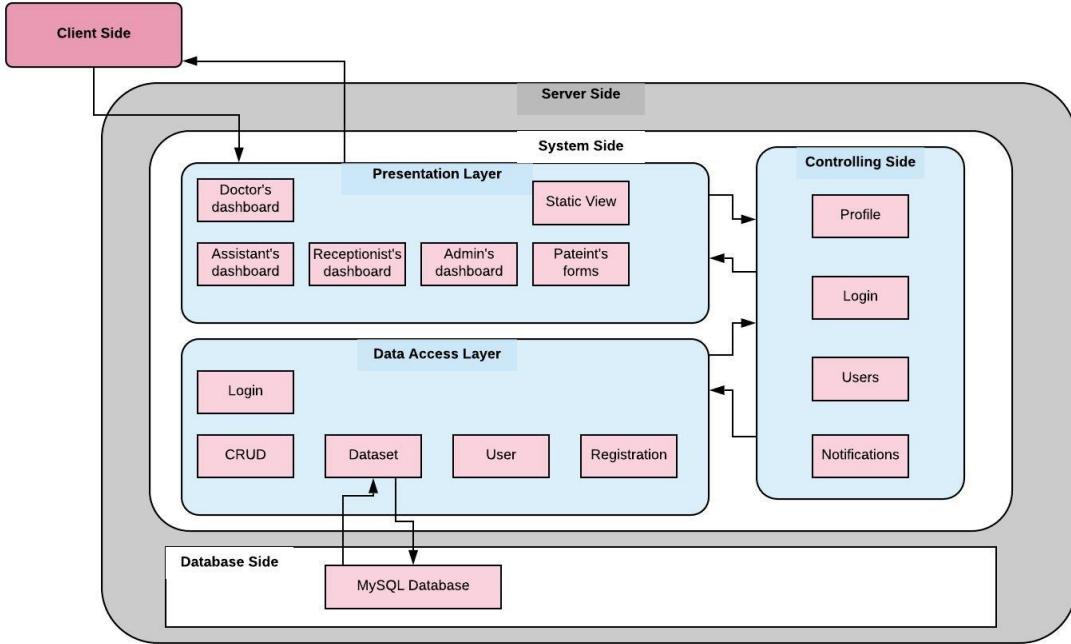


Figure 1: Software Architecture Diagram

4.1 Logical viewpoint

Our system is divided into five models.

- Admin: The Admin is the owner of the system. He has the ability to add Staff and view their data, in addition to viewing everything in the system.
- Receptionists: They are mainly responsible for viewing the appointments.
- Doctor: The system gives the Dr. a lot of privileges, such as viewing appointments, adding reports, viewing Patients' medical history and view operations.
- Assistants: The assistant's main job is to manage operations and contacting Patients through a chatting system.
- Patients: The system gives a lot of privileges to Patients too. Some of the functionalities that the system provides to Patients include, booking appointments, viewing reports, and contacting the Doctor's assistant.

Our system is designed using the MVC architecture, which consists of three partitions: Model (Data access layer), View (Presentation Layer), and Controller (Controlling Layer).

1. Model Layer: Contains only the pure application data, it contains no logic describing how to present the data to a user. By means, it contains the functions which interact directly with the database.
Example:

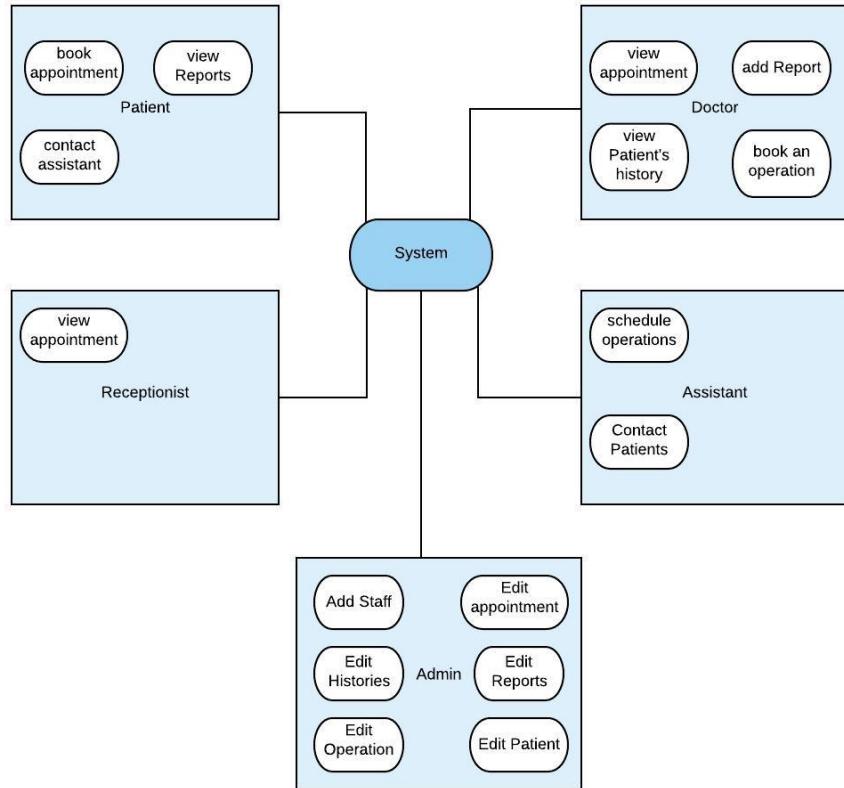


Figure 2: Logical Architecture Diagram

- CRUD: all the functions that are implemented for our users and objects (Create, Read, Update, Delete) deal with the model layer as our functions interact directly with the database.
2. View Layer: presents the model's data to the user. The view knows how to access the model's data, but it does not know what this data means or what the user can do to manipulate it. Simply, the view layer is concerned with what the users see on their screens. Example:
 - Doctor's dashboard: it is the page that appears for the Doctor when he logs into the system.
 3. Controller Layer: It is a mid layer. It exists between the view and the model. It listens to events triggered by the view and executes the appropriate reaction to these events. In most cases, the reaction is to call a method on the model. Example:
 - Login: by getting the data entered by the user and checking into the database if the user exists or if the data are valid.

4.1.1 Design Rationale

MVC is a quite useful architecture to use for our system, as the system is huge. It will help us develop our system easily and in a simple way. Moreover, it helps us update our system easily if needed, making our system more flexible and reliable. It also makes it easier for multiple developers to work together, which is a great advantage in our case.

4.2 Patterns use viewpoint

4.2.1 MVC design pattern

Clinic management system is build on MVC form. MVC is the separation of Model, View and Controller classes. Main advantage of MVC architecture is differentiating the layers of a project in Model, View and Controller for the Re-usability of code and better maintenance. MVC is applied in most of the tables in our project. Examples:

- Appointment
- Reports
- patients system
- Notification system

4.2.2 Singleton Design Pattern

Singleton design pattern is used to allow access to one and only one instance of a particular class. It provides global access to that single instance from anywhere. In our project we used singleton in connection to the database class which is the database helper class, so a single connection is made to the database by each user.

4.2.3 Observer Design Pattern

The Observer Design Pattern is used in the application for the notifications feature due to that we notify in different formats emails and SMS.

4.2.4 Facade Design Pattern

Facade pattern creates an easy use interface by hiding multiple interfaces in one class. We will use facade on all APIs used to hide the complexity of the code.

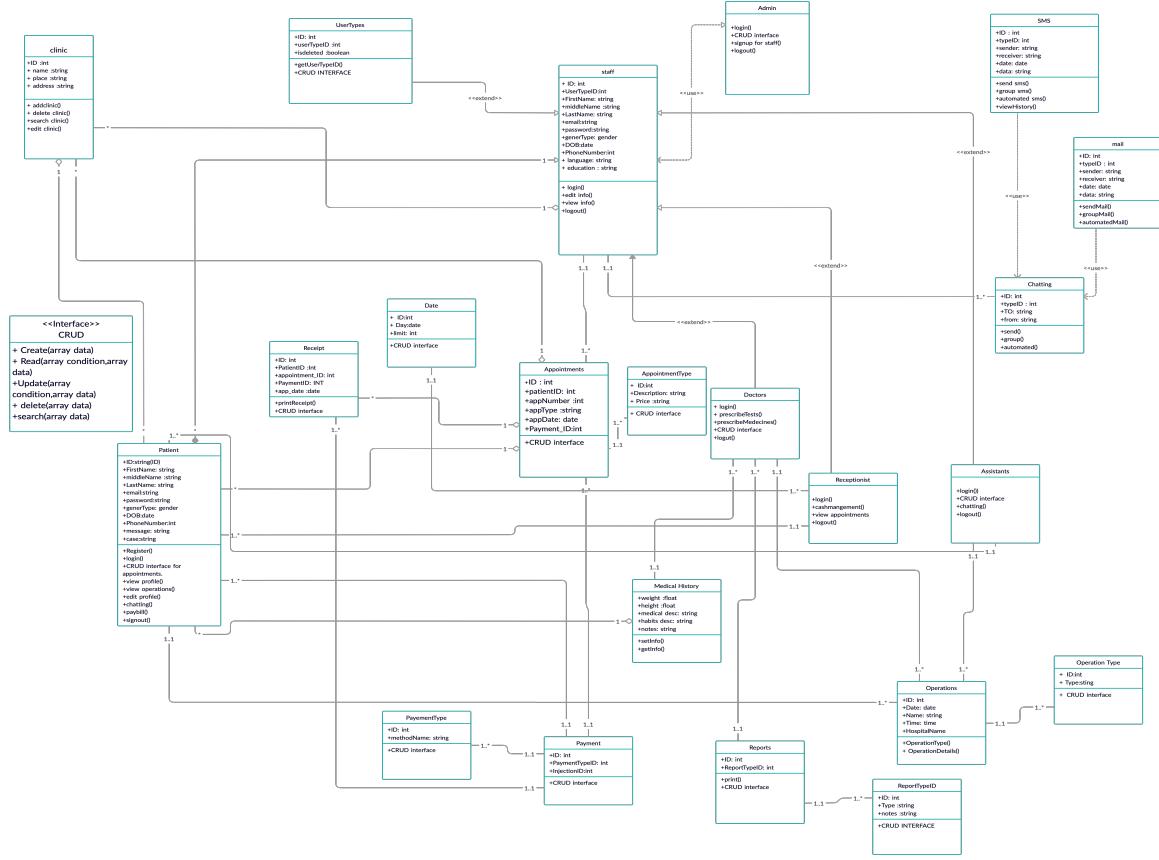
4.3 Composition viewpoint

to be continued

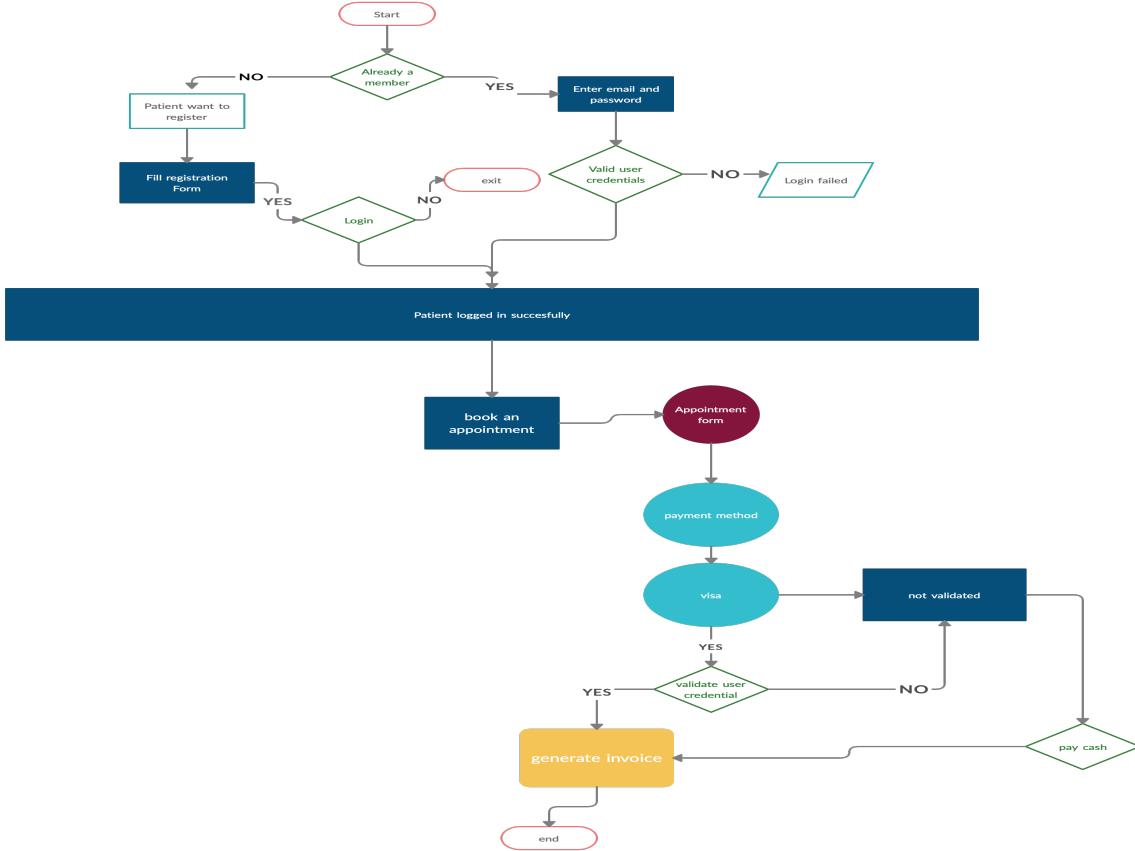
4.4 Structure viewpoint

4.4.1 Class diagram

- CRUD interface: This will contain the functions which draw the forms and tables to contain the data of the users for viewing, searching, inserting, deleting and editing.
- Login: In this part the files are meant to make the validations and verification on all the data inserted for protection and insurance during login.



4.4.2 process architecture



4.5 Interaction viewpoint

4.5.1 Patient

The Patients' sequence diagram shows the sequence of events that happen and appear to the Patient, once he logs into the system. If the Patient managed to successfully log into the system, the Patient's main page pops up, which has many options for the patient. If the Patient chose to book an appointment, the appointment page appears to him, so he can start booking. Once he's done booking, an invoice page appears to him. The Patient can also view Reports written to him by the Doctor from his main page. Once he chooses this option, a page appears with all his Reports. Moreover, the Patient can contact the assistant through the Services page, and be informed when the assistant replies to his message.

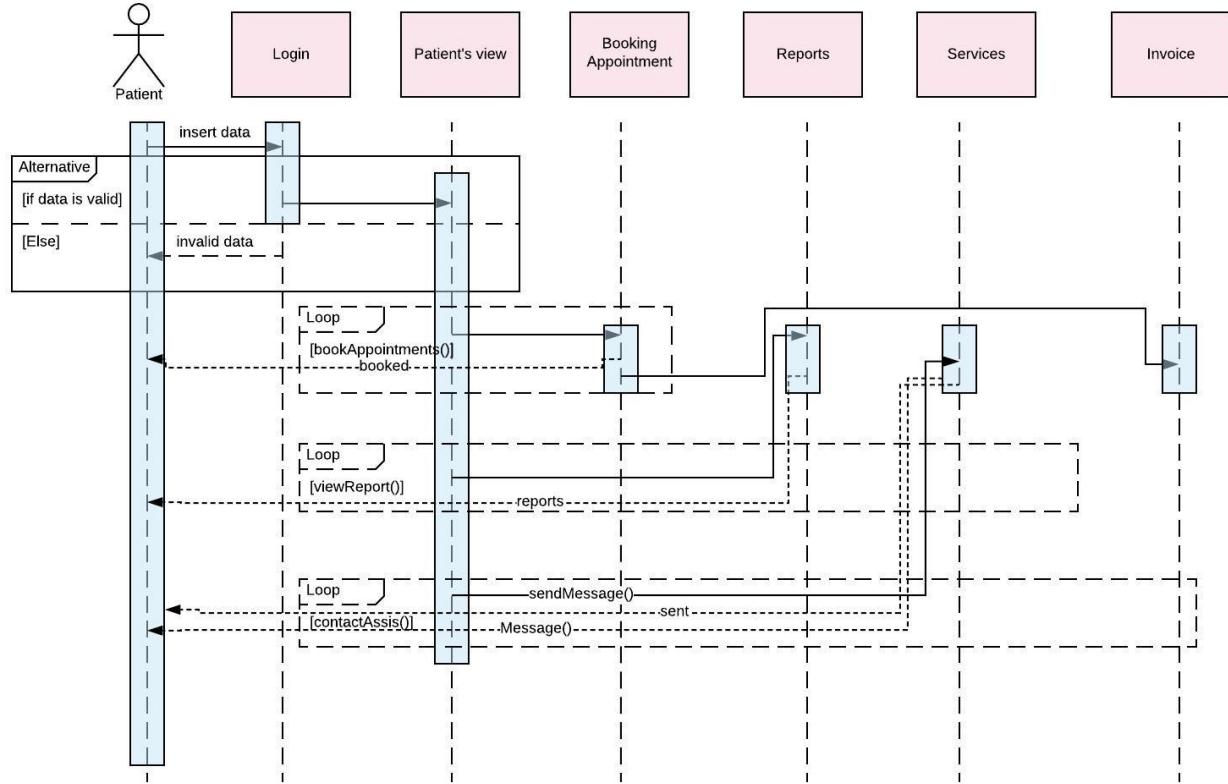


Figure 3: Patient's Sequence Diagram

The Doctor's sequence diagram shows the sequence of events that happen and appear to The Doctor, once he logs into the system successfully. The first things that appears to the Doctor is that Doctor's dashboard that, like the Patients', has various alternative pages for the Doctor to choose from. If the Doctor wished to view appointments, the appointments page, which has all the appointment's for the day, will pop up to him. The Doctor can also open the Reports' page, which will allow him to add a Report for any Patient he chooses. Moreover, he can view Patients' histories and book operations, leading him to the pages required for each of them.

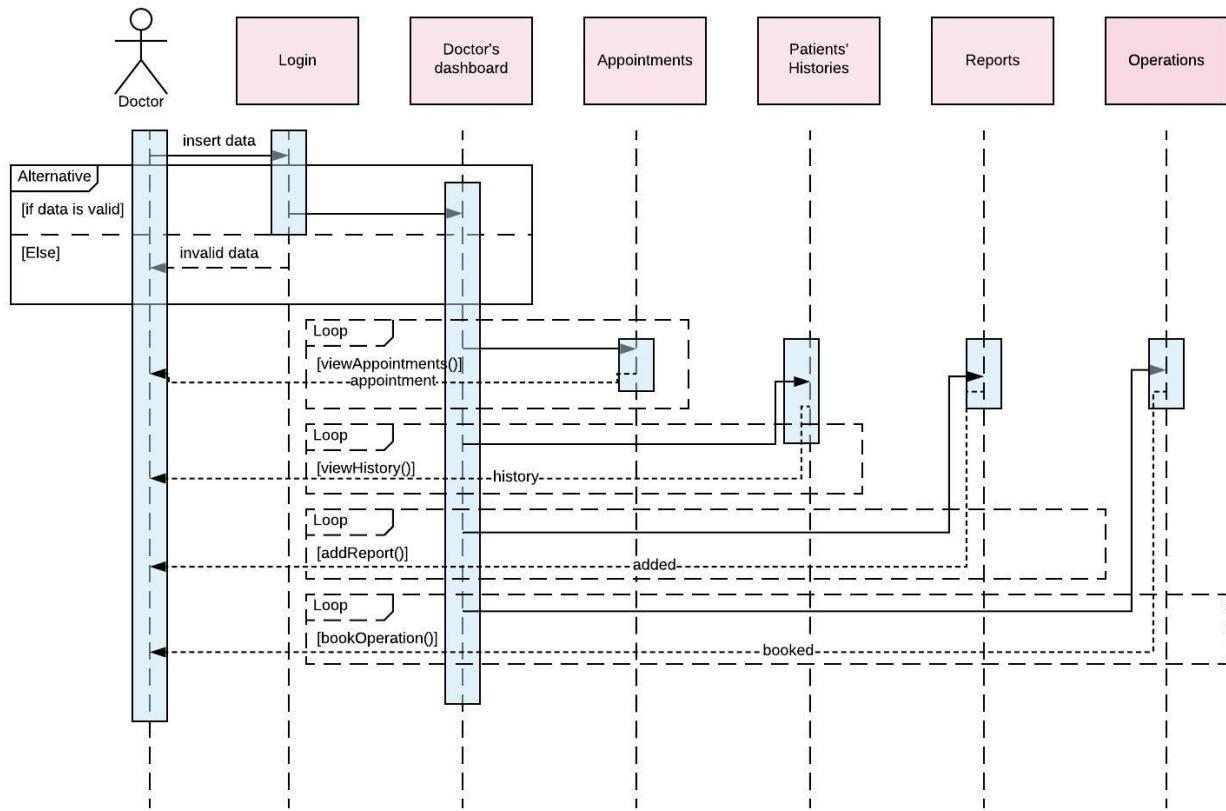


Figure 4: Doctor's Sequence Diagram

- The Assistant's sequence diagram shows the sequence of events that happen and appear to the Assistant, once he logs into the system successfully. The first things that appears to the Assistant is the Assistant's dashboard that, gives him the right to view the appointments, the histories of the Patients', and the Reports written by the Doctor. The Assistant can also chat with Patients through the chatting system and reply to their messages. Moreover, he can book operations for the Patients. For each of these options, appears a page for the Assistant work on.

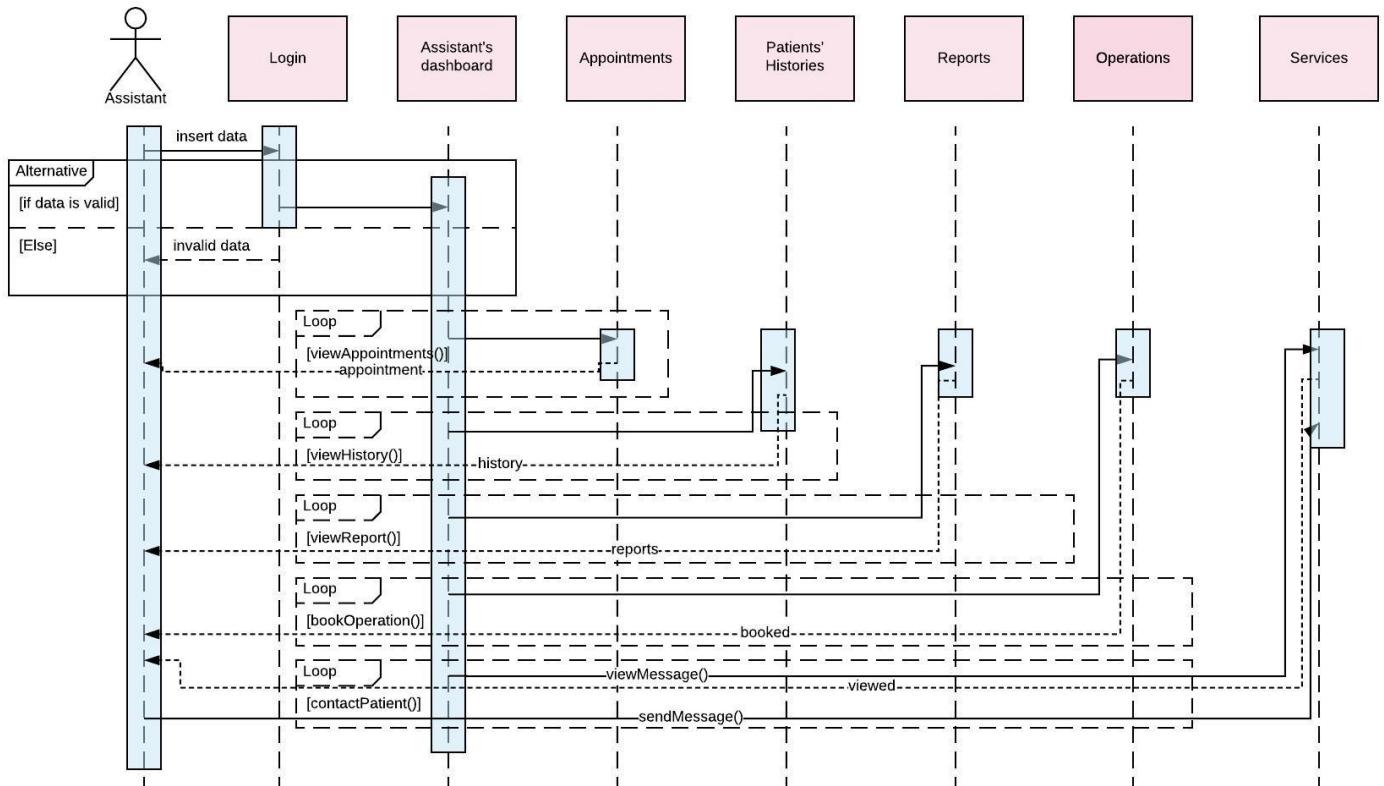


Figure 5: Assistant's Sequence Diagram

- The Admin's sequence diagram shows the sequence of events that happen and appear to the Admin, once he logs into the system successfully. The Admin's dashboard appears once he logs in. The Admin has the privilege of updating everything in the system including, Appointments, Patients' histories, Reports, and Operations. Moreover, he has the ability to add Staff and delete Patients.

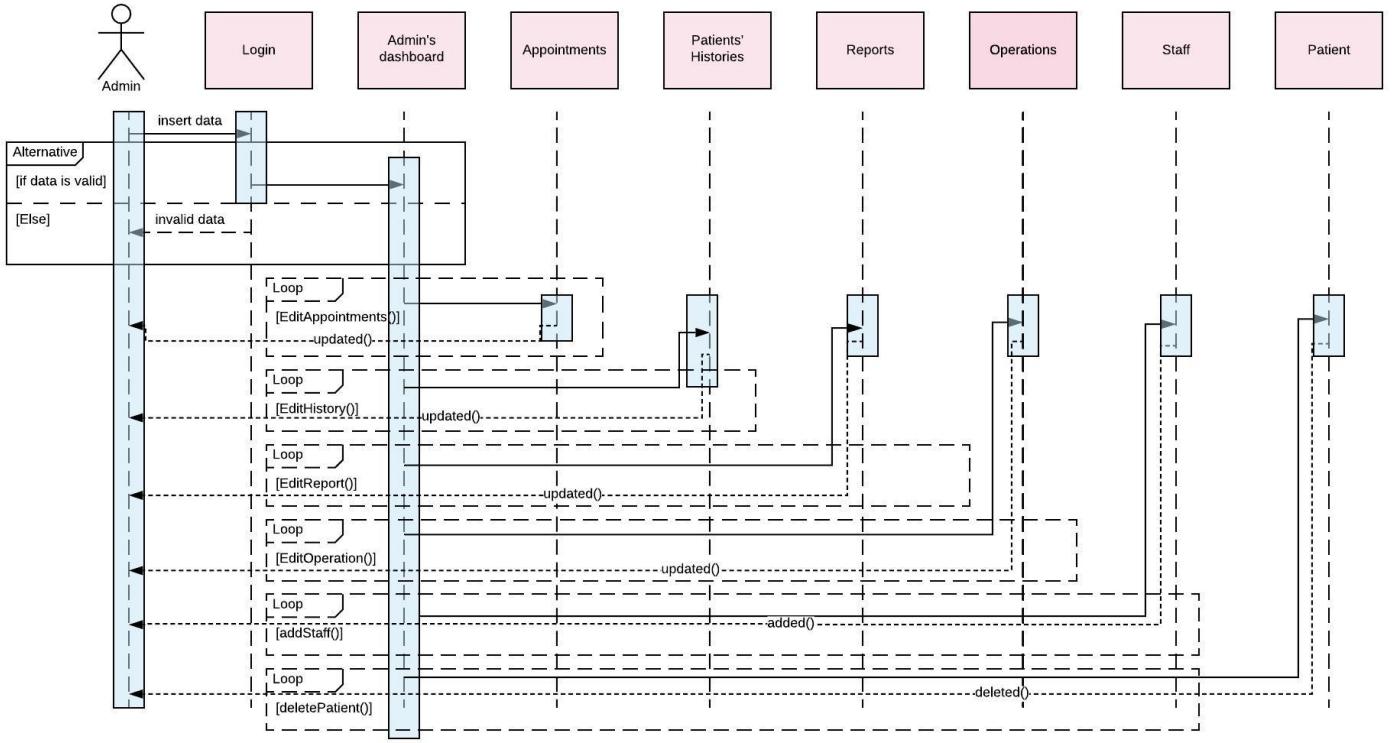


Figure 6: Admin's Sequence Diagram

- The Receptionists' sequence diagram shows the sequence of events that happen and appear to the Receptionists, once he logs into the system successfully. The Receptionist's dashboard pops up, which has less options than the rest. The Receptionists' can only view appointments and view invoices, so they can print them to the Patients.

5 Data Design

5.1 Data Dictionary

- Rtb option user type: the relation between the options the user types which is mainly the staff as Admin, doctor, assistant, receptionist.
- Rtb Payment : the table that contains the relations between the Payment Method ID and the description, where it saves the options ID required to be shown when selecting any kind of payment according to the appointment type.
- Rtb appointments: the relation between the clinic ID from class clinics that is chosen and the patient's specific time slot that is booked.
- Rtb patient's history: the relation between the patients and the doctor as the doctor is allowed to view every patients history to precisely write the report needed for this specific case.
- Tb clinics : stores the description and the id of each clinic branch.
- Tb option payment values: the table that carries the values that each user has paid by a certain payment method.

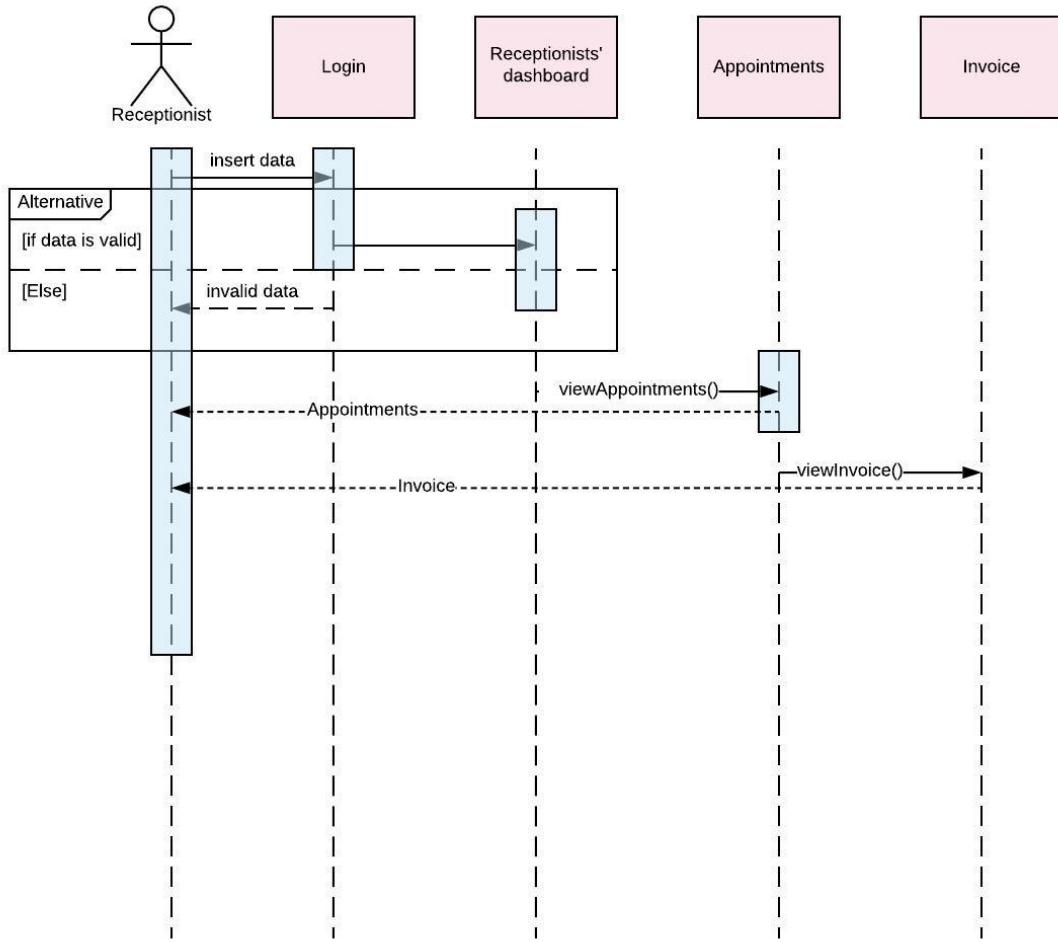


Figure 7: Receptionists' Sequence Diagram

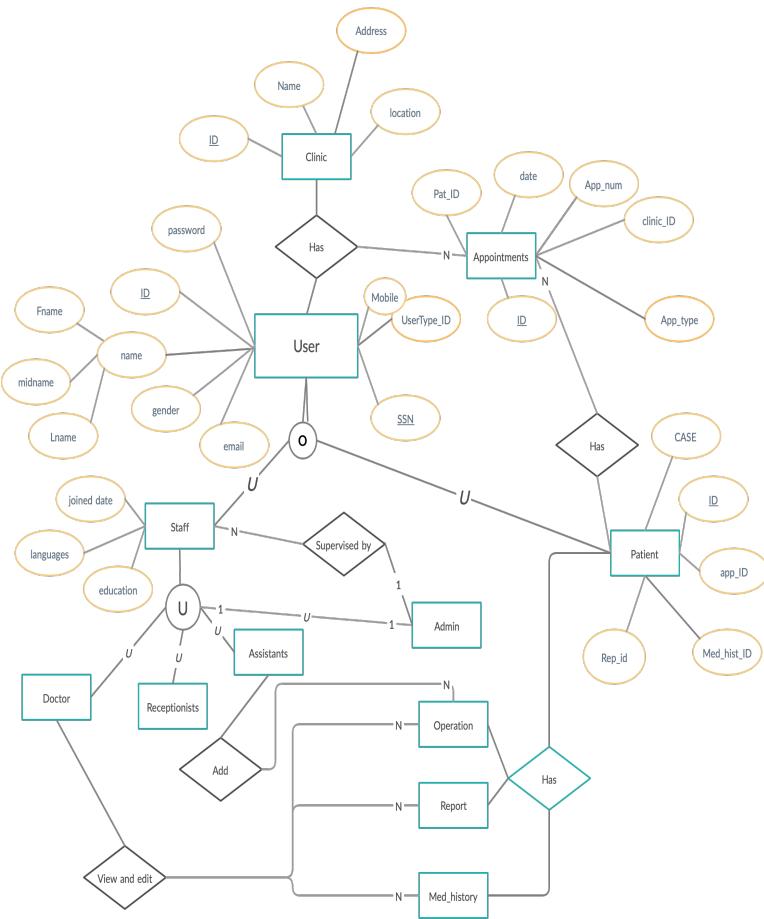
- Tb injections:saves the description and the price of the injection the doctor recommended to the patient.
- Tb appointments: the appointments that happened and its amount and category each appointment also stores the day and the patient's id.
- Tb day limit: that table that stores the day of appointments and the maximum limit of appointments for that specific day because it is dynamic and depends on the type of appointment that is reserved.
- Tb staff: the email and hashed password for each user signed up on the system and his/her own profile details.
- tb patients: storing the data of the patients while registering into the system.
-
- Rtb operations: the table that contains the relation between the patient and the assistant that the assistant has the ability to add operations if needed to this specific patient.
- Rtb operations type:the table that contains the relation between reports and operations that knows from tb reports if the doctor added an operation to this patient or not.

- Rtb reports: the table that contains relation between the patient and the doctor as the doctor has the ability to generate a report to each patient with specific description according to his/her illness.
- Tb reports: stores the date the report generated at, comments and the patient's ID.
- Tb message: the email subject and content that system can send to clients.
- Tb message type: determine either the message is via mail or SMS.
- Tb user type: all the staff types on the system and their IDs.

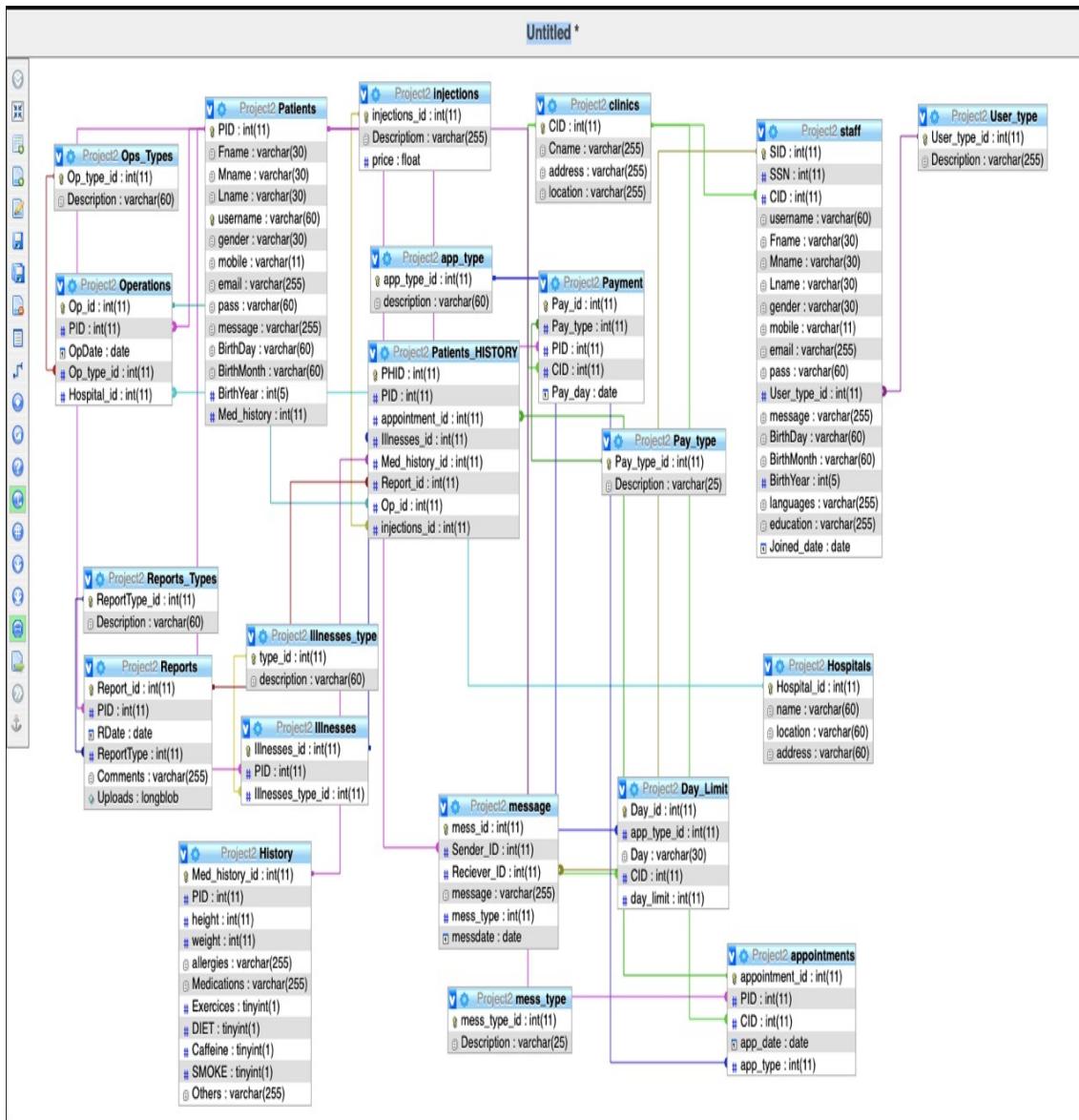
5.2 Database design description

MySQL database to communicate with the database that is installed locally on the server.

5.2.1 ER diagram



5.2.2 Database schema



6 Human Interface Design

6.1 User Interface

The proposed project allows the user to create an account. The system will allow the user to sign in with his account. The patient will choose which clinic branch he will book an appointment at. The system allows the clinic to have accounts for every patient to be easier to contact with them. Also from the staff' side it will help the doctor to know the medical history and generate report to each patient to follow up with them.

6.2 Screen Images



Figure 8: Main Page of website the patient use it

The screenshot shows a patient registration form on a web page. The form fields include:

- UserName: [Text input]
- First Name: [Text input]
- Middle Name: [Text input]
- Last Name: [Text input]
- Email: [Text input]
- Password: [Text input]
- Mobile: [Text input]
- Gender: [Text input]
- Birthdate:
 - Day: [Text input]
 - Month: [Text input]
 - Year: [Text input]

At the bottom left is a blue "Sign Up" button, and at the bottom right is a link "Already registered! Click Here!".

Figure 9: patient registration form

localhost/SW/Public/Profile.php

New message from Dr. Dalia Salama - sandra170...

Post Attendee - Zoom

Dr. Ahmed Ghoneim

(4) WhatsApp

User Name:

sandra5

First Name:

sandra

Middle Name:

fares

Last Name:

Wardkhan

Email:

sandra1@gmail.com

Password:

Mobile:

0100

Gender

Female

Birthdate

7

Month

February

1999

Edit

Figure 10: patient filling registration form

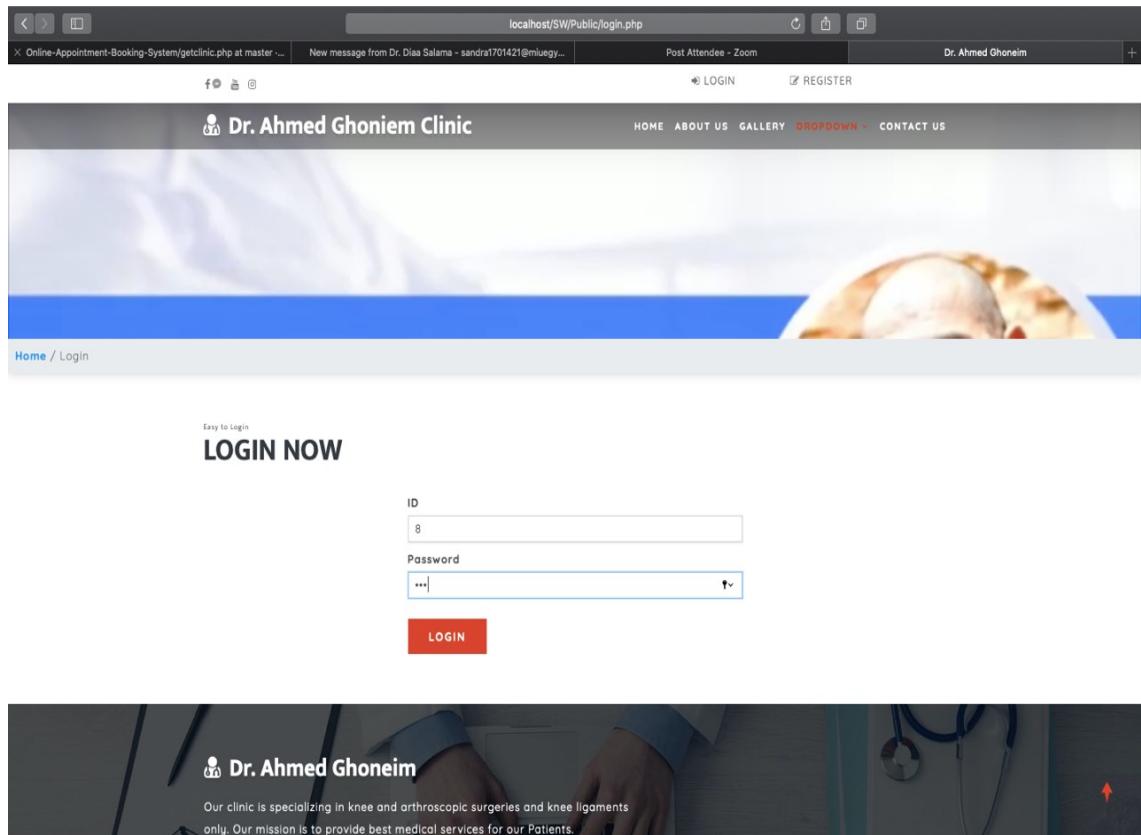


Figure 11: patient logging in

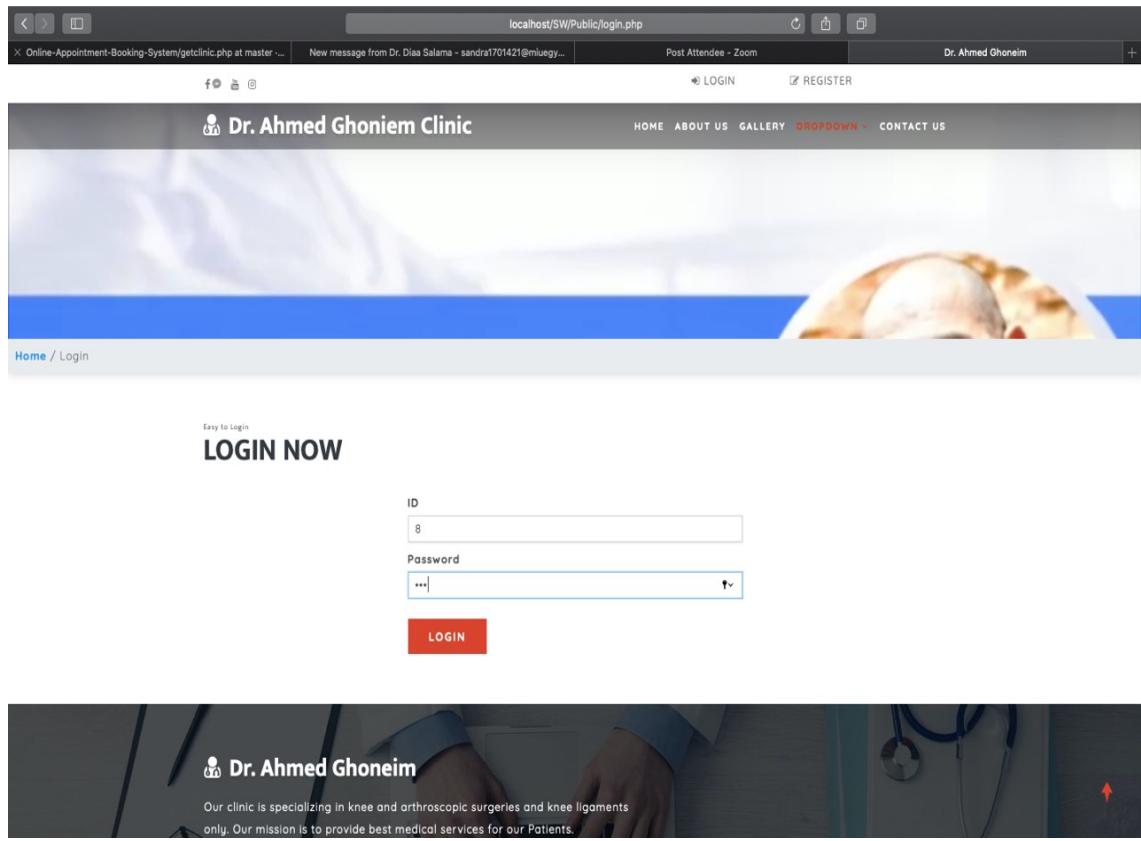


Figure 12: patient login with alert

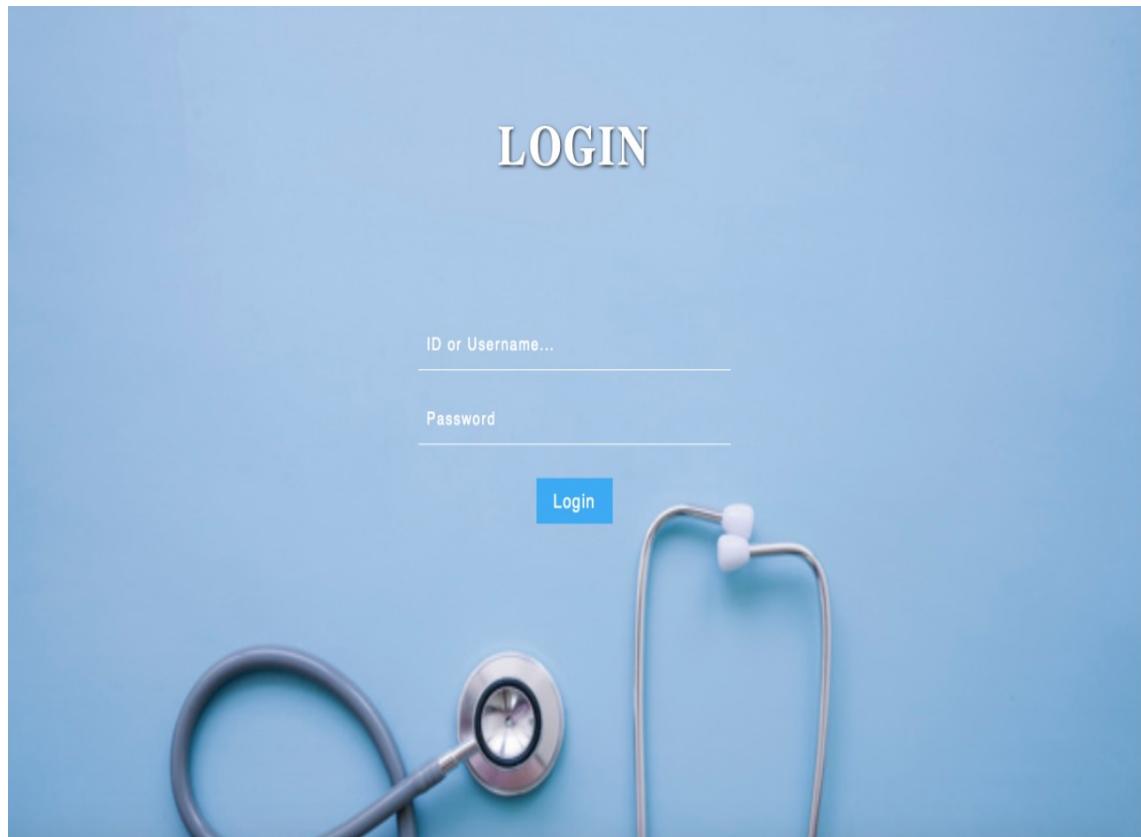


Figure 13: staff's login

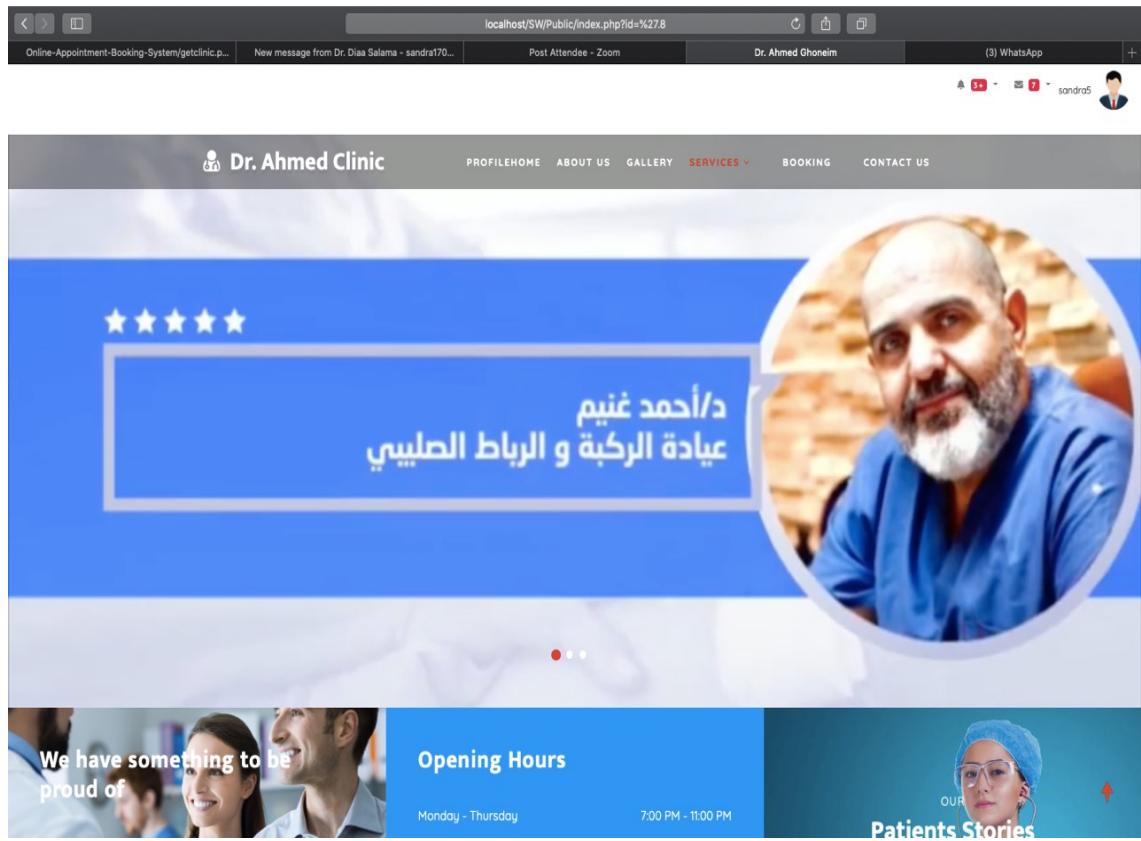


Figure 14: patient's dashboard

First Name

Middle Name

Last Name

Email

Password

Email

Mobile

Gender

Birthdate

Figure 15: Patient's page

Staff Registration

First Name

Middle Name

Last Name

National ID

Email

Password

Email

Mobile

Gender

Birthdate

Languages known

Qualifications

Employeed in which clinic ?

Figure 16: Staff's page

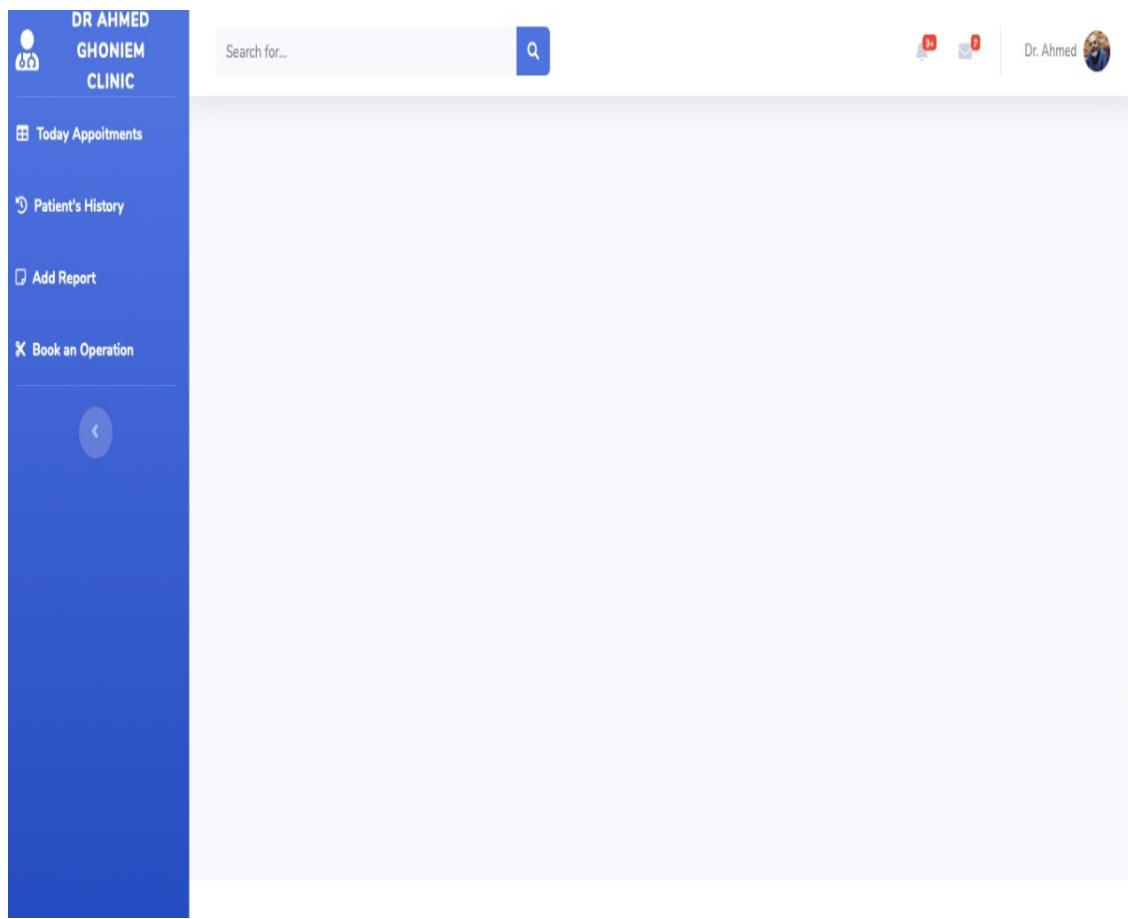


Figure 17: Doctor's dashboard

BOOK YOUR APPOINTMENT NOW

For Help Call : 01150001410

Enter your id

Enter Full name

Enter Mobile Number

Which Clinic:

.....Clinics.....

Reasons For Seeing the Doctor:

.....Reasons.....

Enter Date

Enter time

SUBMIT

Figure 18: Booking Page

The screenshot shows a web-based appointment management system. The top navigation bar includes a logo for "DR AHMED GHONIEM CLINIC" and a search bar. On the left, a sidebar lists various administrative functions: "Today Appointments", "Add Staff", "Edit Appointments", "Edit Histories", "Edit Reports", "Edit Operations", and "Edit Patients". The main content area is titled "Tables" and contains a table titled "Example". The table has columns for Id, Name, Case, Type of Appointment, Appointment Time, and Other. It displays 6 entries. At the bottom, it shows "Showing 1 to 6 of 6 entries" and includes navigation buttons for "Previous" and "Next".

Id	Name	Case	Type of Appointment	Appointment Time	Other
400	Airi Satou	Both knees	fellow up	2020/03/25 9:00	view
602	Brielle Williamson	Left knee	fellow up	2020/03/25 9:30	view
1001	Tiger Nixon	Left Knee	Examination	2020/03/25 7:00	view
1006	Cedric Kelly	Right knee	Examination	2020/03/25 8:30	view
1200	Garrett Winters	Both knees	Examination	2020/03/25 7:30	view
2100	Ashton Cox	Right Knee	Examination	2020/03/25 8:00	view

Figure 19: Appointment Page

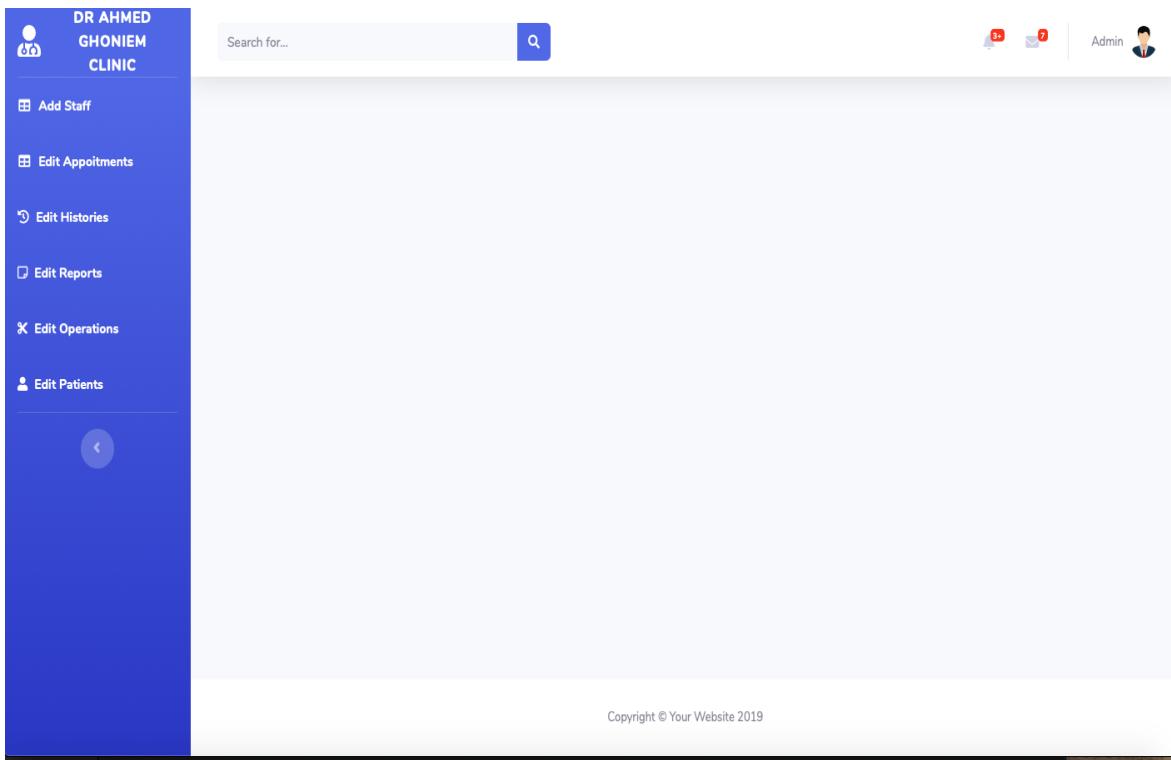


Figure 20: Admin Page

7 Requirements Matrix

Requirement ID	Requirement name	Requirement description	Status
FR00	Check email validation	This Function is to check if the email is valid before inserting it in the database.	In progress
FR01	Check password validation	This Function is to check if the password is valid before inserting it in the database.	In progress
FR02	Check phone Number validation	This Function is to check if the phone number is valid before storing it in the database.	In progress
FR03	Login	This Function is for the user to login into the system using his/her account.	Completed
FR04	Forget Password	This Function exists if the user forgets the password, it generates a new password and sends email with it to help him login successfully again.	In progress
FR05	Create User	The functions get the user type ID according from which page the inserting process whether the patient's registration page or from site administrator is happening, then it uses the input data to insert a new row in users table in the system database	Completed
FR06	Delete User.	The Function is to remove the selected user row from users table	In progress
FR07	Search User	The Function returns the user with the same id or name that being searched for by one of our staff users	In progress
FR08	Update User	The Function performs an sql query to users table to update a certain record according to a specific user ID	Completed
FR09	Create Usertype	this function is to add new user type to the system in the database	Completed
FR10	List Usertype	this function is to List all existing user type in database according to the user type ID	Completed
FR11	Delete UserType	This function is to delete the selected user type row from user type in database	In progress
FR12	Update UserType.	Admin can update the type of the user if there were any changes according to user ID	In progress
FR13	View Profile.	User can view his/her profile, admin can also view the profile	Completed
FR14	Send Notification.	The function for sending the notification to the patient	In progress
FR15	Add Appointment	This function adds a new record to the appointment table in the database	Completed
FR16	Edit Appointment	This function updates the appointment data in the database according to the appointment ID.	In progress
FR17	Delete Appointment	This function selects a specific row from the appointment table and removes it.	In progress
FR18	view notification.	viewing all the notification in the database	In progress
FR19	Delete notification.	The admin can delete the notifications from the database	In progress
FR20	Send message.	The function for sending messages between the users	In progress
FR22	Create Report.	This functions allows the Doctor to create or write reports to his patients if needed	In progress
FR23	Print Receipt	This function prints a receipt for the patient of his booking fee and injection fee, if he needed any.	Completed
FR24	Add Operation	This function adds an operation by getting the operation ID according to which operation is being inserted and inserting it in the database	In progress
FR25	Edit Operation	This function updates the operation data in the database according to the operation ID	In progress
FR26	Delete Operation	This function selects a specific row from the operation table and removes it.	In progress

Figure 21: matricies

8 References

- [1] [Online] Ros, Fabian, "Hospital Management System using PHP with Source Code", 01 2018, Source Code and Projects, Available at: <https://code-projects.org/hospital-management-system-using-php-source-code/>, Accessed:03/05/2020
- [2] [Online] Creately Templates, "Online Clinic Reservation System - Context Diagram (Data Flow Diagrams(YC))", Creately, Available at: <https://creately.com/diagram/example/intvotax1/OnlineClinicReservationSystem-ContextDiagram>, Accessed:10/05/2020