

Software Requirement Specification Document

V1.1 Dr. Ahmed Ghoniem Clinic

Mariam Hesham, Nour Ahmed ,Samiha Hesham, Sandra Fares

March 31, 2020

Contents

1	Introduction	3
1.1	Purpose of this document	3
1.2	Scope of this document	3
1.3	Overview	3
1.4	Business Context	3
2	General description	4
2.1	Product Functions	4
2.2	Product context	5
2.2.1	Context Diagram	5
2.2.2	Block Diagram	6
2.3	Similar System Information	6
2.4	User Characteristics	6
2.5	User Problem Statement	7
3	Functional Requirements	7
4	Interface Requirements	26
4.1	User Interfaces	26
4.1.1	GUI	26
4.1.2	Main page	27
4.1.3	Login pages	28
4.1.4	Registration page	30
4.1.5	Doctor's Dashboard	31
4.1.6	Booking Page	32
4.1.7	Appointment's Time Table Page	33
4.1.8	Admin's Page	34
4.1.9	CLI	34
4.2	API	34
5	Design Constraints	34

6	Other non-functional attributes	35
6.1	Security	35
6.2	Reliability	35
6.3	Maintainability	35
6.4	Portability	35
6.5	Flexibility	35
7	Preliminary Object-Oriented Domain Analysis	36
7.1	Class diagram	36
7.2	ER diagram	37
8	Operational Scenarios	39
8.1	Login and Registration System	40
8.2	Appointment System	40
8.3	Chatting System	40
8.4	Operation System	40
8.5	Administrator System	40
9	Project Plan	41
10	Abbreviations	41
11	Collected material	41
12	References	44

1 Introduction

1.1 Purpose of this document

The main purpose of this document is to clarify and demonstrate the requirements of a Clinic System. These requirements include booking appointments, contacting Doctors or Physician assistants, and viewing patients' medical information. Our aim is to assist patients' to reserve appointments and reach out to healthcare professionals easily. Moreover, our system shall make it easier for Doctors to view patients' medical information, in order to provide accurate diagnosis. This documentation shall provide an explanation of each phase, along with an illustration on how this system is going to function.

1.2 Scope of this document

The scope of this document is to break down our client's problems into chunks that can be solved more readily. It also helps to increase understanding of issues and makes them easier to be tackled through our system and tested properly to fulfil the optimum solutions.

1.3 Overview

The project clinic management is developed to facilitate the communication process between the doctor, the receptionist, the assistant and the patient. The software would be managed by two admins one is doctor and the other is receptionist. The doctor can view patient details and after checking up the patient, the recommended medicines for the patient are fed into the database by the doctor and are sent to receptionist. The patient can book an appointment and state the medical history. Receptionist would be responsible for viewing appointments and save it in the database along with their details. The receptionist can then generate bill and feed into the database. The system also maintains patient's history so that doctor or assistant can view them anytime. The assistant would be responsible for booking an operation and follow up with the patient after the surgery. The system can thus reduce complexity in maintaining patient's records.

1.4 Business Context

The project aims to build an application program to reduce the manual work for managing the clinic. According to Cheah Ming Lay's article on Clinic management systems [3], Clinic management systems help to end the complexities involved in accounting along with detailed profit and loss reports. The system can provide convenient access to information and making it less likely that mistakes would be made due to illegible writing. The internet-based access improves the ability to remotely access the data and making it easier to maintain and quicker to search through for relevant. The system also saves time for medical

staff to focus on more important tasks at hand and reduces wait times which is otherwise spent filling out forms in and reduces the waiting time for patients at the clinics.

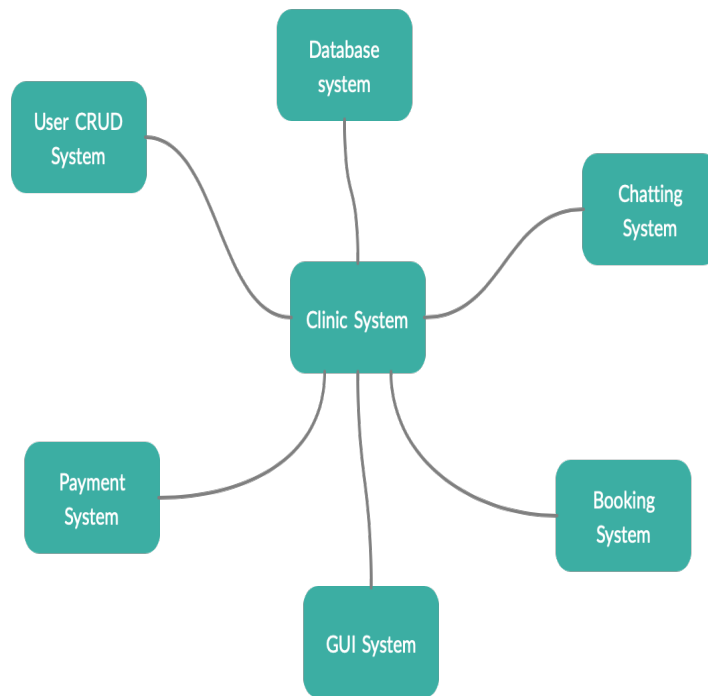
2 General description

2.1 Product Functions

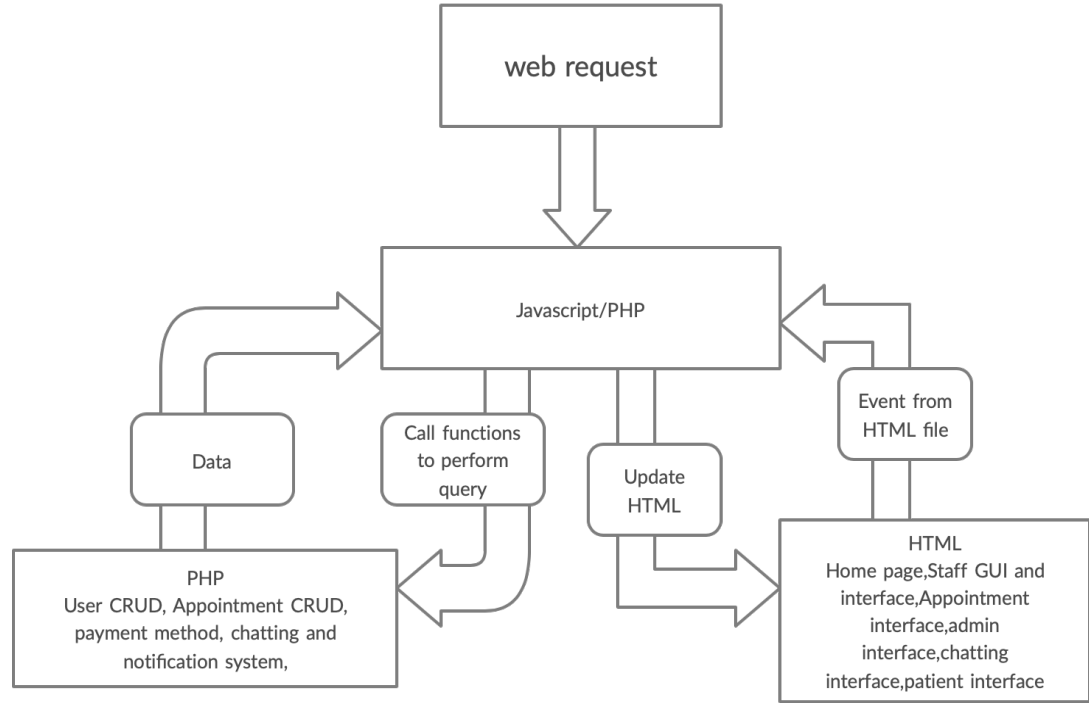
1. The system must be fully dynamic,in which the admin is fully controlling the system
2. The users(patients,doctor,receptionists,assistants) must be able to control their information
3. The patient must be able to reserve his/her appointment online and choose either paying online or cash.
4. The system should be able to send automatic emails and SMS to the patients.
5. The admin should have full authority to control all staff permissions.
6. The doctor must be able to see full information of the patient.
7. The doctor should has the ability to write a report to the patient's case with all details.
8. The doctor must be able to reserve operation for the patient if needed.
9. The assistant must organize the doctor's schedule and add,edit or delete operations to it.
10. The assistant should perform a broadcast en-forming patients of any cancellation in the doctor's schedule.
11. The patients must have the authority to edit their appointment details.
12. The receptionist must view patients' reservations and should deal with financial staff.

2.2 Product context

2.2.1 Context Diagram



2.2.2 Block Diagram



2.3 Similar System Information

This software is similar to various generic products. However, MedDNA [2] , myhCue [5], and Adroit's eClinic [1] are the most similar to our system (Adriot Infosystems is a Healthcare IT company which produced various products, including eClinic). These systems are stand-alone clinic management systems that are available for installation by any customer in need. Their purpose is to help in managing appointments, viewing patients' history, and offering effortless communication between the Doctors and their Patients, just like our system is intended to do.

2.4 User Characteristics

According to a Software Requirements Specification document written by Shiva Prasad [4] , there are four main users in the clinic management system. The admin, doctor, receptionists, and patients. However, in our system, we will have

an additional user who is the doctor's assistant. The system is also designed to be user-friendly.

- Admin
- Receptionists
- Doctor
- Assistants
- Patients

Admin: Admin should have prior knowledge of the system. Admin is able to control the whole system. He/she can add, delete, update and modify the system.

Receptionists: in order view the details of the patients come for the treatment and accordingly provides identity to them also, schedule the appointments of the day and deals with payment methods.

Doctor: Doctor should fairly know about the usage of the system. Doctor are able to see the respective appointments taken. And also, can view patient's details and records and set for them operations if needed.

Assistants: they are mainly managing the chatting system for the clinic with patients and they must view, edit ,delete and add operations in the doctor's schedule.

Casual users: Anyone can view the information of the polyclinic. Patients can view their own records and doctor's details and timings to take appointment online.

2.5 User Problem Statement

Our client needs to replace all the manual operations done daily in his clinic into automated ones. Starting from patient's booking their appointments, saving their history to be easier to retrieve on the follow up day to booking surgeries too .Also they need a chatting system in order to facilitate patients contact with our client or his assistant .All of which should be easy to use , without any complexities to provide fast interactions without consuming time while dealing with our software.

3 Functional Requirements

Table 1

Function:	Check email validation.
ID	FR00
Description	This Function is to check if the email is valid before inserting it in the database.
Type	Boolean
Input	Email
Action	Checks if the email entered in the text field is in the proper format and if the email is valid it returns true means that the email is valid to be stored if exists in the database.
Output	Boolean if email is valid
Precondition	User entered email into the required text field.
Post-condition	return to login function (FR03)
Dependencies	return to login function (FR03).

Table 2

Function:	Check password validation
ID	FR01
Description	This Function is to check if the password is valid before inserting it in the database.
Type	Boolean
Input	Password
Action	Checks if the password is valid in format. If yes it returns true means that the password is valid to be stored if exist in the database with its email.
Output	Boolean if password valid
Precondition	User entered password into the required text field
Post-condition	Return to login function (FR03)
Dependencies	FR03

Table 3

Function:	Check phone Number validation
ID	FR02
Description	This Function is to check if the phone number is valid before storing it in the database.
Type	Boolean
Input	Phone number
Action	Checks if the phone number entered in the field is in the proper format and if it is valid in it's format it returns true means that it is valid to be stored in the database.
Output	Boolean if phone number is valid.
Precondition	User enter the phone number into the required text field.
Post-condition	Return to login function (FR03)
Dependencies	FR03

Table 4

Function:	Login.
ID	FR03
Description	This Function is for the user to login into the system using his/her account.
Type	Boolean
Input	Email and Password
Action	Checks if the data are filled and compare the record that was entered before to the record in the database, The function returns true otherwise returns false.
Output	Boolean acceptance login.
Precondition	The user needs to into his email and password into the Text fields and that the inputs are verified (FR00, FR01)
Post-condition	Redirect to the home page
Dependencies	FR00, FR01

Table 5

Function:	Forget Password
ID	FR04
Description	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.
Type	Boolean
Input	Email
Action	Firstly the email is checked using FR00 then if the email exists in the database, the function sends email FR14 with the new password generated.
Output	Record updated
Precondition	The user enters his email in the Text field.
Post-condition	Notification is sent to the user with new password.
Dependencies	FR00, FR14

Table 6

Function:	Create User.
ID	FR05
Description	The functions get the user type ID according from which page the inserting process whether the patient's registration page or from the administrator is happening, then it uses the input data to insert a new row in users table in the system database
Type	void
Input	user type id and user data
Action	checks if the user's email,password and phone number are validated, if so it enters the user record in the database and returns true else it returns false and an error message will appear
Output	New User Record in Database
Precondition	User's information is right and validated
Post-condition	Creating a new row in database table with new ID for the user
Dependencies	FR00 , FR01 , FR02

Table 7

Function:	Delete User.
ID	FR06
Description	The Function is to remove the selected user row from users table
Type	Boolean
Input	ID of the user selected to be removed
Action	check if the user id exists in database
Output	A Success Message if User is deleted
Precondition	Administrator chooses which user to be deleted
Post-condition	User Record is deleted from database and database is up-dated
Dependencies	FR05

Table 8

Function:	Search User.
ID	FR07
Description	The Function returns the user with the same id or name that being searched for by one of our staff users
Type	Return Function
Input	Id or Name of the desired user
Action	check if user id or name matches any of records in database , if there is a user show it else returns null
Output	if Users details exists
Precondition	One of staff users enters users id or name needed to be found
Post-condition	user record is retrieved from database if is found
Dependencies	–

Table 9

Function:	Update User
ID	FR08
Description	The Function performs an sql query to users table to update a certain record according to a specific user ID
Type	Boolean
Input	User ID and the desired category to be updated with it's new record
Action	check if those data can be updated
Output	success message or failure message
Precondition	check if user exists in database and has the right to update
Post-condition	records in database are updated
Dependencies	–

Table 10

Function:	Create Usertype
ID	FR09
Description	this function is to add new user type to the system in the database
Type	Boolean
Input	User Type name
Action	Check if the user type name is not duplicated in the database , if so it enters the user type record in the database and returns true else return false
Output	Success Message if user type is created
Precondition	Check if user type doesn't exist
Post-condition	Creating a new database record with a new ID and add the new user type name in it
Dependencies	—

Table 11

Function:	List Usertype
ID	FR10
Description	this function is to List all existing user type in database according to the user type ID
Type	Returns array
Input	—
Action	Retrieving all user types in database
Output	Array by each user type
Precondition	Check if user type exists
Post-condition	—
Dependencies	FR09

Table 12

Function:	Delete UserType
ID	FR11
Description	This function is to delete the selected user type row from user type in database
Type	Boolean
Input	ID of which user type to be deleted
Action	Check if user Type id exists in the database to delete it
Output	Message if user type deleted
Precondition	The selected user type needed to be deleted
Post-condition	Updating the database by removing the user type id row from table
Dependencies	—

Table 13

Function:	Update UserType.
ID	FR12
Description	Admin can update the type of the user if there were any changes according to user ID
Type	Boolean
Input	User Type ID
Action	If User Type is updated return true, else return false and show error message
Output	if User Type update, success message
Precondition	There must be at least one User Type added to the storage and action must be made by the admin
Post-condition	The record of the user is updated in the database
Dependencies	FR09

Table 14: -

Function:	View Profile.
ID	FR13
Description	User can view his/her profile, admin can also view the profile
Type	Return array
Input	User ID
Action	Admin views user profile
Output	Array filled by each user's data
Precondition	Check if the user exists
Post-condition	The certain profile
Dependencies	FR03,FR05,FR09

Table 15

Function:	Send Notification.
ID	FR14
Description	The function for sending the notification to the patient
Type	Boolean
Input	Notification object
Action	Check for receiving the notificatin
Output	If true string notification, if false there will be an error message
Precondition	The user must be existed in the database
Post-condition	The user will receive the notification, the notification is filled in the database
Dependencies	FR09

Table 16: -

Function:	Add Appointment
ID	FR15
Description	This function adds a new record to the appointment table in the database
Type	Boolean
Input	Appointment data to be recorded. Including the ID, Patient's ID, appointment number, type, and date.
Action	If the appointment is added successfully, it gets recorded in the database. else, it returns an error message.
Output	New appointment is added in the database
Precondition	Patient books an appointment and fills up all the information needed to book an appointment.
Post-condition	The appointment has been booked successfully and the patient is expected to meet his Doctor at the time he specified.
Dependencies	FR03, FR05

Table 17: -

Function:	Edit Appointment
ID	FR16
Description	This function updates the appointment data in the database according to the appointment ID.
Type	Boolean
Input	Appointment ID to the appointment to be updated
Action	Checks if the appointment data has been updated. If yes, it returns true. else, it returns an error message.
Output	Appointment's data updated.
Precondition	Checks if the appointment exists.
Post-condition	Update in the appointment table the selected appointment.
Dependencies	FR15

Table 18: -

Function:	Delete Appointment
ID	FR17
Description	This function selects a specific row from the appointment table and removes it.
Type	Boolean
Input	ID for the desired appointment to be deleted.
Action	Checks if the appointment exists in the database. If yes, removes it. else, returns an error message.
Output	Appointment deleted successfully.
Precondition	Checks if the appointment exists and select it.
Post-condition	Updating the database by removing the appointment row from the appointment table.
Dependencies	FR15

Table 19: -

Function:	view notification.
ID	FR18
Description	viewing all the notification in the database
Type	return array
Input	
Action	fetching all the notifications in the database
Output	array of notifications
Pre condition	check the existence of notifications
Post condition	
Dependncies	FR09

Table 20: -

Function:	Delete notification.
ID	FR19
Description	The admin can delete the notifications from the database
Type	Boolean
Input	Notification object
Action	Check for the existence of the notification and delete it
Output	Confirmation message if true otherwise there will be an error message
Pre condition	Select the desired notification that will be deleted
Post condition	The desired notification record is deleted
Dependncies	FR18,FR20

Table 21: -

Function:	Send message.
ID	FR20
Description	The function for sending messages between the users
Type	Void
Input	Array of users
Action	Check if the array of users is not empty then send the message
Output	The desired message appears
Pre condition	The user send the message
Post condition	Messages are sent
Dependncies	FR20

Table 22

Function:	Create Report
ID	FR22
Description	This functions allows the Doctor to create or write reports to his patients if needed
Type	Boolean
Input	Doctor specified and wrote a Diagnosis
Action	If the report is created successfully, it gets recorded in the database. else, it returns an error message
Output	Report created successfully
Precondition	Doctor examines his patient and makes a Diagnosis
Post-condition	A new report is made for the patient and is successfully recorded in the database
Dependencies	FR03, FR06, FR15

Table 23: -

Function:	Print Receipt
ID	FR23
Description	This function prints a receipt for the patient of his booking fee and injection fee, if he needed any.
Type	float
Input	Booking and/or injection fees
Action	If any payment is required, the function returns the overall fee for the patient if there is any. else, it returns an error message
Output	Receipt containing the overall fee for the patient
Precondition	Patient books an appointment and/or uses an injection
Post-condition	Printing successfully a receipt for the patient to review
Dependencies	FR03, FR15

Table 24: -

Function:	Add Operation
ID	FR24
Description	This function adds an operation by getting the operation ID according to which operation is being inserted and inserting it in the database
Type	void
Input	The information needed for the operation such as, the operation's ID, location, date, hospital name, etc.
Action	If the operation is added successfully, it is recorded in the database. else, it returns an error message
Output	New operation added to the database
Precondition	Doctor examines the patient and determines that an operation must be made
Post-condition	An operation has been successfully recorded for the Doctor to perform
Dependencies	FR03, FR05, FR15

Table 25: -

Function:	Edit Operation
ID	FR25
Description	This function updates the operation data in the database according to the operation ID
Type	Boolean
Input	Operation ID to the operation to be updated
Action	Checks if the operation data has been updated. If yes, it returns true. else, it returns an error message
Output	Operation's data updated
Precondition	Checks if the operation exists
Post-condition	Update in the operation table the wanted operation
Dependencies	FR24

Table 26: -

Function:	Delete Operation
ID	FR26
Description	This function selects a specific row from the operation table and removes it.
Type	Boolean
Input	ID for the desired operation to be deleted
Action	Checks if the operation exists in the database. If yes, removes it. else, returns an error message
Output	Operation deleted successfully
Precondition	Checks if the operation exists and selects it
Post-condition	Updating the database by removing the operation row from the operation table
Dependencies	FR24

4 Interface Requirements

4.1 User Interfaces

4.1.1 GUI

4.1.2 Main page



Figure 1: Main Page of website the patient use it

4.1.3 Login pages

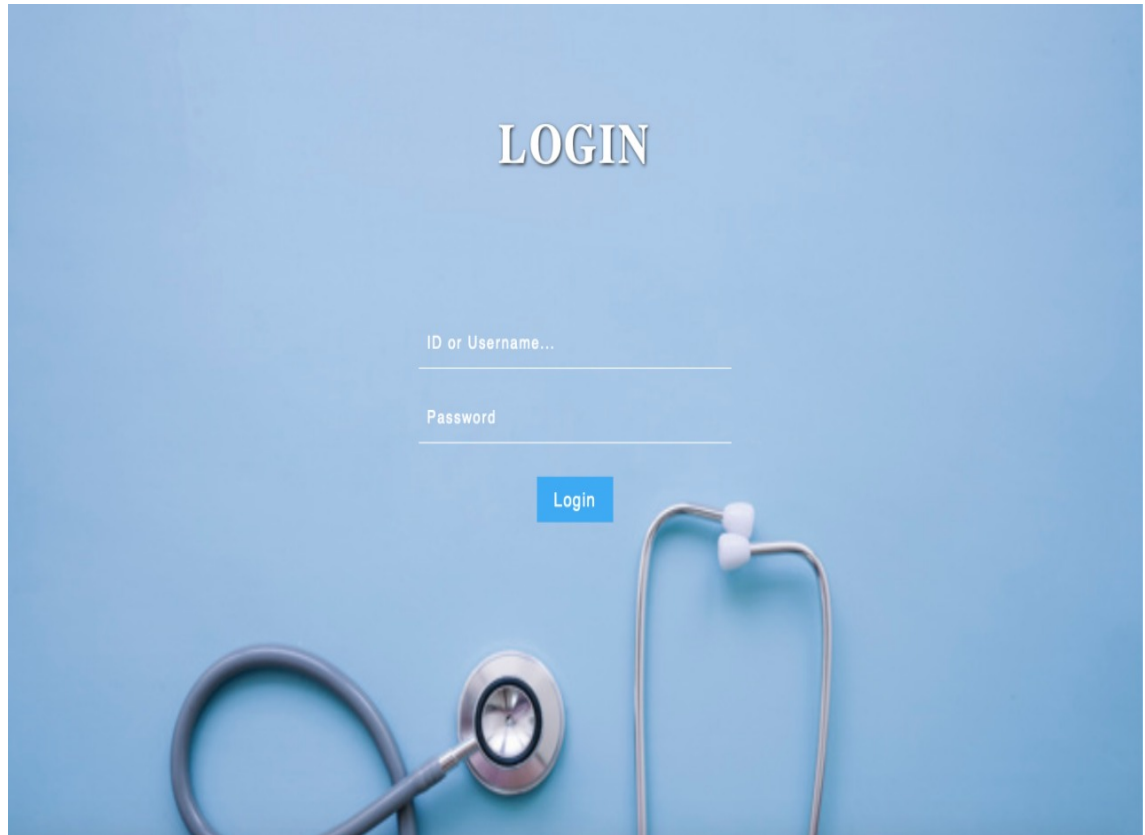


Figure 2: Staff's Login

[Home](#) / [Login](#)

Easy to Login

LOGIN NOW

ID

We'll never share your ID with anyone else.

Password

LOGIN

[Don't have an Account?](#)


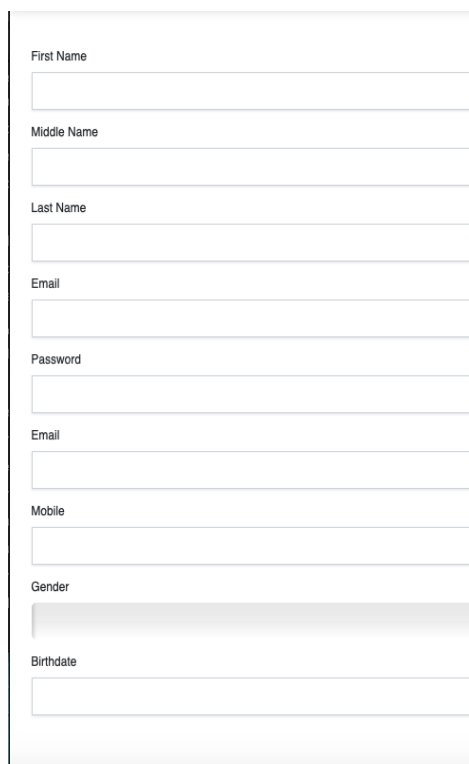


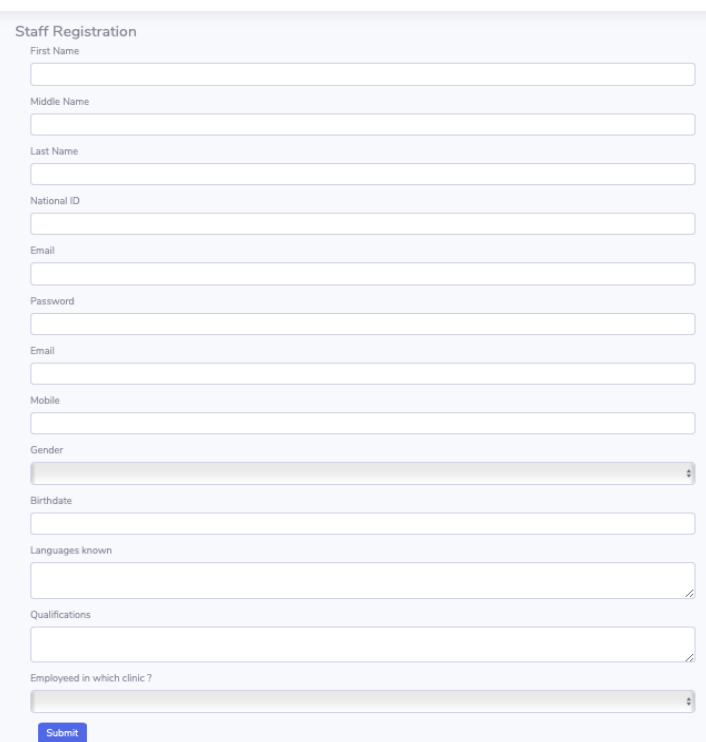
Figure 3: Patient's login

4.1.4 Registration page



A vertical registration form for patients. It contains the following fields from top to bottom: First Name, Middle Name, Last Name, Email, Password, Email, Mobile, Gender (a dropdown menu), and Birthdate. A blue vertical bar is positioned to the right of the form.

Figure 4: Patient's page



A horizontal registration form for staff. It is titled "Staff Registration" and contains the following fields from top to bottom: First Name, Middle Name, Last Name, National ID, Email, Password, Email, Mobile, Gender (a dropdown menu), Birthdate, Languages known, Qualifications, and Employed in which clinic? (a dropdown menu). A blue "Submit" button is located at the bottom left of the form.

Figure 5: Staff's page

4.1.5 Doctor's Dashboard

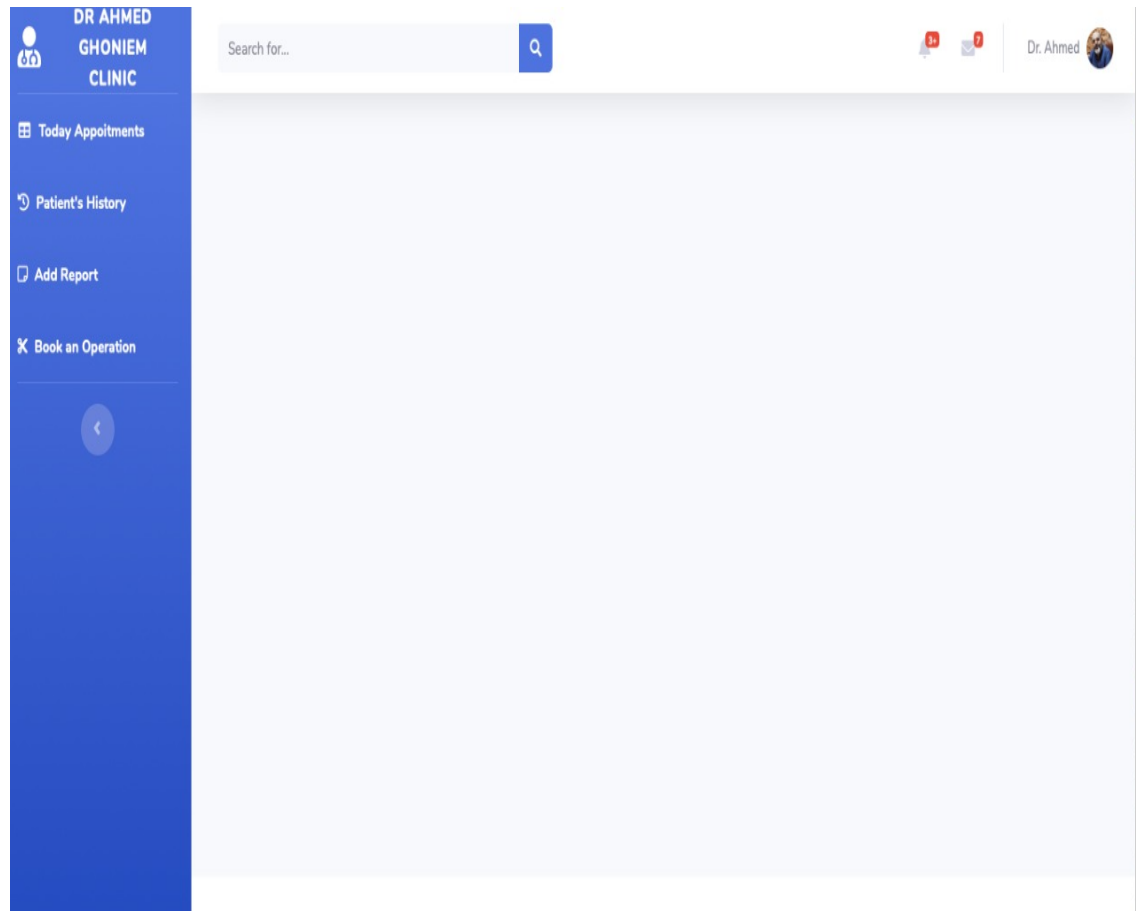


Figure 6: Doctor's dashboard

4.1.6 Booking Page

BOOK YOUR APPOINTMENT NOW

For Help Call : 01150001410

Which Clinic:

-----Clinics-----

Reasons For Seeing the Doctor:

-----Reasons-----

SUBMIT




Figure 7: Booking Page

4.1.7 Appointment's Time Table Page

DR AHMED
GHONIEM
CLINIC

Today Appointments

Search for...

Tables

Example

Show 10 entries

Search:

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

Showing 1 to 6 of 6 entries

Previous 1 Next

Figure 8: Appointment Page

4.1.8 Admin's Page

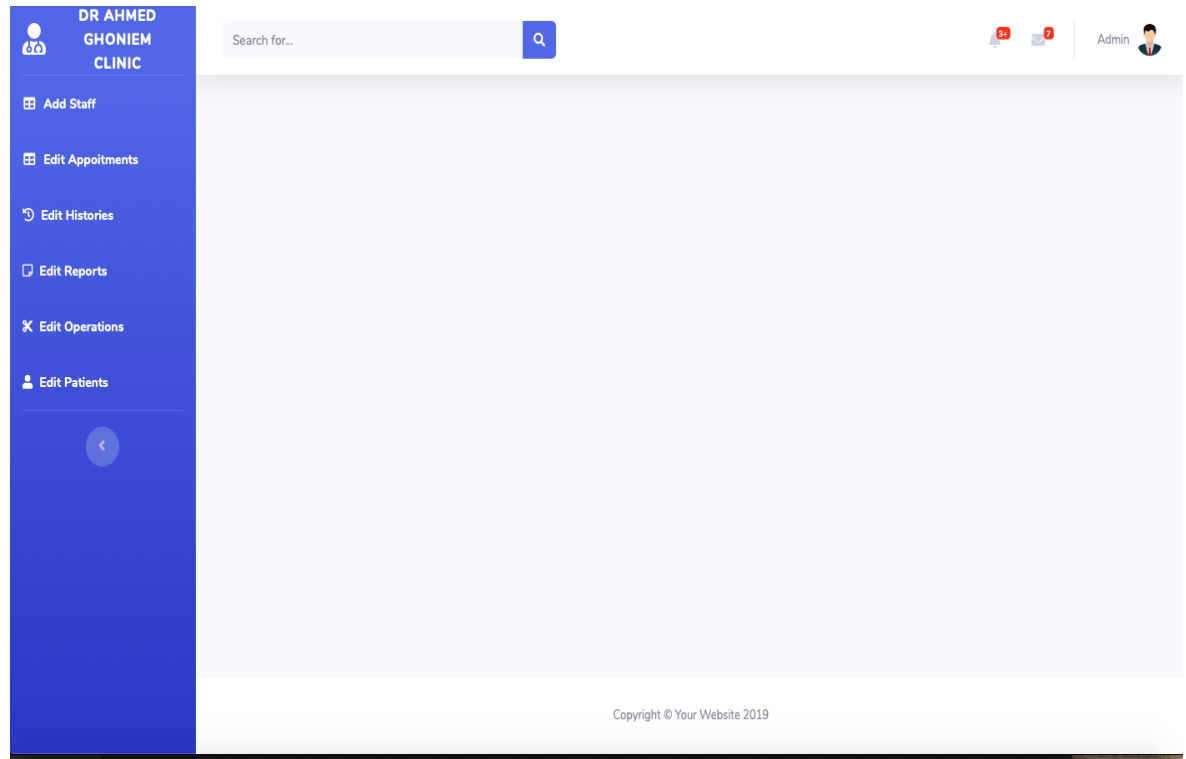


Figure 9: Admin Page

4.1.9 CLI

N/A

4.2 API

N/A

5 Design Constraints

The system should be adaptable to be used from different kinds of devices either tablets, laptops... etc. Also we had a constraint with which is the internet speed it varies from one user to another.

6 Other non-functional attributes

6.1 Security

This system is provided with authentication without which no user can pass. So only the legitimate users are allowed to use the application. If the legitimate user's share the authentication information then the system is open to outsiders.

6.2 Reliability

Good validations of user inputs will be done to avoid incorrect storage of records.

6.3 Maintainability

The software system should be designed to be easily to maintain and improved by software developers through the implementation of MCV design patterns so that helps a lot in maintaining the software system, and minimize the amount of changes that will be done in the code

6.4 Portability

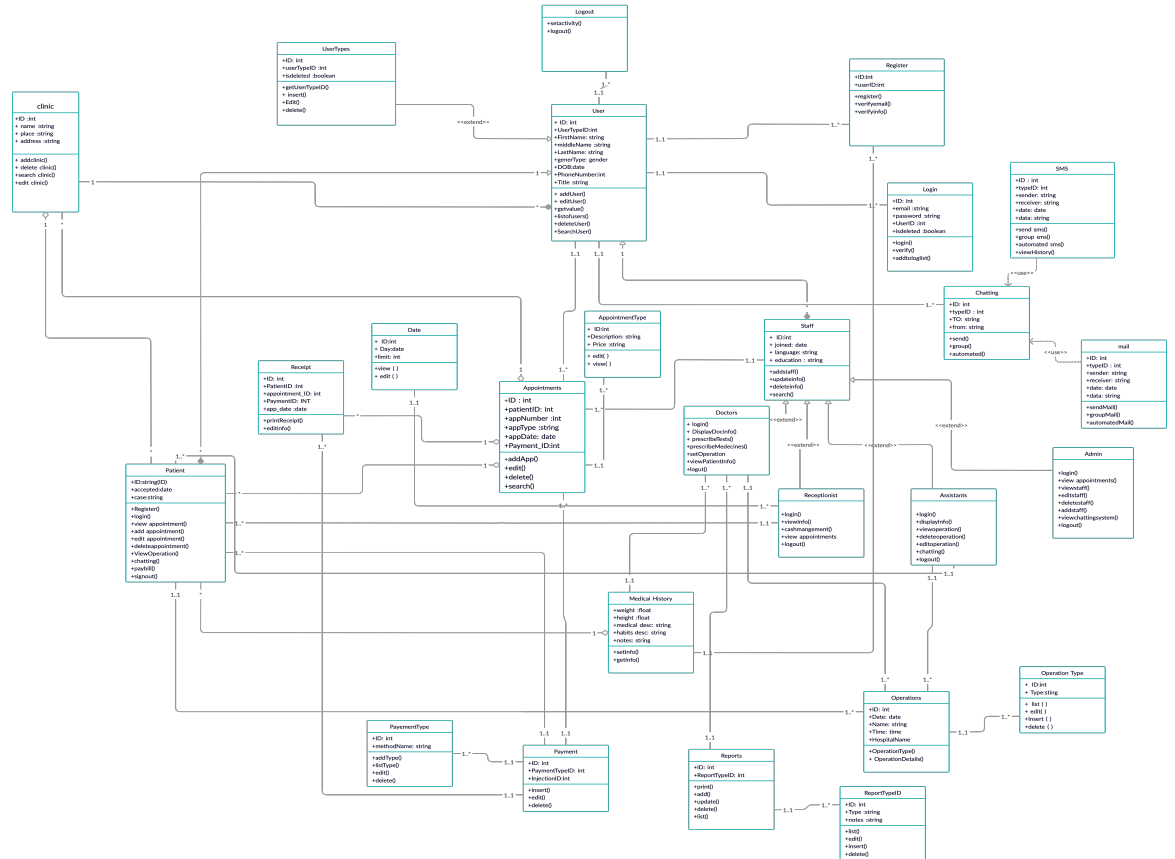
This system can be installed in any personal computers supporting windows operating system platform.

6.5 Flexibility

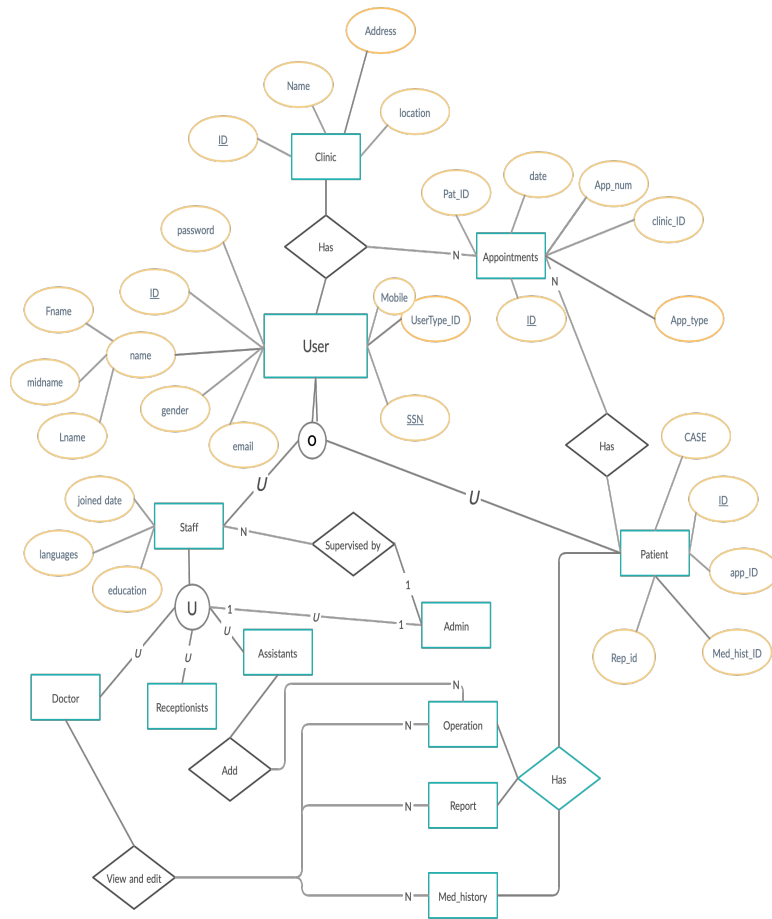
The system keeps on updating the data according to the transactions that takes place.

7 Preliminary Object-Oriented Domain Analysis

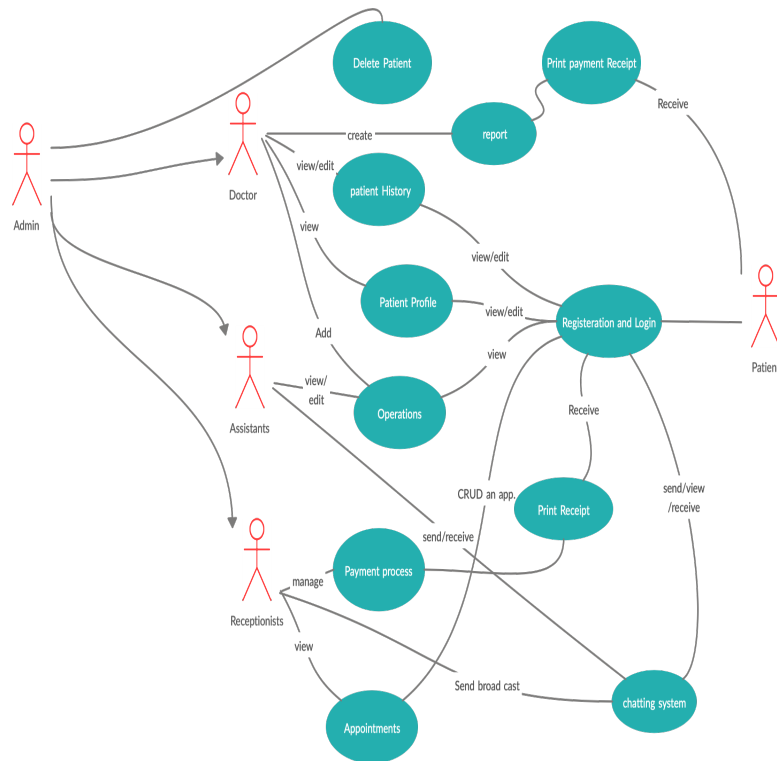
7.1 Class diagram



7.2 ER diagram



8 Operational Scenarios



The following is a brief explanation for some of the scenarios that present in our system:

8.1 Login and Registration System

- All Patients should view and edit their information including their medical description and personal habits
- Doctor can view patient's information but only can edit their medical history and his reports.

8.2 Appointment System

- Patients have the ability to choose the time of appointment they want to reserve . Moreover, they can edit or delete them and they can choose the branch(Location of the clinic) they want to book at.
- Receptionists have the ability to see the appointment's schedule for every day.

8.3 Chatting System

- Receptionists and assistants are responsible for managing the chatting system.
- Patients have the ability to send messages to the clinic to ask for anything and the assistant is responsible for handling their questions and requests.
- Patients can view their history messages .
- Admin has the ability to select specific patients to broadcast the message to them.
- Receptionists send notification either by mail or SMS to the patient to inform them if the doctor cancelled any reservations.

8.4 Operation System

- The doctor has the ability to add operation to the patient if needed.
- The assistant can view the operations and add or edit anything in them.

8.5 Administrator System

- The Admin can view everything in the system and has the authority to do all the staff's functions .

9 Project Plan

Table 27

Task	End Date
Html templates	17/3/2020
UserInterface	17/3/2020
Booking Interface	17/3/2020
User Full Control	4/4/2020
Staff Full Control	10/4/2020
Booking Full Control	10/4/2020
Messaging	15/4/2020
Project Full Control	5/5/2020

10 Abbreviations

- SMS = Short Message Service
- ER=Entity Relationship diagram

11 Collected material

250	رسوم كشف	1146903804	27/06/2019	سعاد علي	386
250	رسوم كشف	1064888272	01/07/2019	كريم عبد العزيز	387
250	رسوم كشف	1200179172	01/07/2019	ناهد رمضان	388
250	رسوم كشف	1229302964	01/07/2019	مومن فايز	389
250	رسوم كشف	1114364707	01/07/2019	عماد طهيه	390
250	رسوم كشف	1146394364	01/07/2019	زوان سليمان	391
250	رسوم كشف	100800585	01/07/2019	محمد يحيى	392
250	رسوم كشف	1012758598	04/07/2019	سارة احمد	393
250	رسوم كشف	1100424966	04/07/2019	علي يسرى	394
250	رسوم كشف	1113998209	04/07/2019	احمد مجدى	395
250	رسوم كشف	1118981211	04/07/2019	محمد شوقي	396
250	رسوم كشف	1144444551	04/07/2019	عمر محمد	397
250	رسوم كشف	1150809944	04/07/2019	انجي هشام	398
250	فيزيتا	1006822344	04/07/2019	زياد محمود	399
250	رسوم كشف	1155694442	08/07/2019	رضا سيد احمد	400
250	رسوم كشف	1021194664	08/07/2019	مصطفى محمد	401
250	رسوم كشف	1021190488	08/07/2019	حياء حسن	402
250	رسوم كشف	1114724532	08/07/2019	جمعة محمد	403
250	رسوم كشف	1147606365	11/07/2019	سعيد رجب	404
250	رسوم كشف	1113708498	11/07/2019	مشارى احمد	405
250	فيزيتا	1153005640	11/07/2019	هاتم علي اسماعيل	406
250	رسوم كشف	1282330841	11/07/2019	اسماعيل دويدار	407
250	فيزيتا	1007065124	11/07/2019	محمد الغنام	408
250	رسوم كشف	101327772	11/07/2019	تقريب سعيد	409
250	رسوم كشف	1017788601	15/07/2019	محمد ممدوح	410
250	فيزيتا	1211661824	15/07/2019	احمد عيد	411
250	فيزيتا	1142020838	15/07/2019	بالقاسم القوزى	412
250	رسوم كشف	1061305758	15/07/2019	ابراهيم السيد	413
250	رسوم كشف	112447212	15/07/2019	سارة نجران	414
250	رسوم كشف	1060488187	15/07/2019	خالد ابراهيم	415
250	رسوم كشف	1121205616	18/07/2019	عزة عبد العظيم	416
250	رسوم كشف	1277088216	18/07/2019	محمد زكى	417
250	رسوم كشف	1015661838	18/07/2019	واقي عبد الله	418
250	رسوم كشف	1008724101	18/07/2019	سعاد محمد	419
250	رسوم كشف	1113508498	18/07/2019	بسنت عطيفى	420
250	فيزيتا	116882808	18/07/2019	محمود التركى	421
0	رسوم كشف	1155013821	18/07/2019	اسامه عبد الله	422
250	رسوم كشف	1006420927	18/07/2019	محمد احمد الحداد	423
250	رسوم كشف	1114241912	18/07/2019	مصطفى ابو زيد	424
250	رسوم كشف	1004561058	22/07/2019	صفاء صلاح	425
250	رسوم كشف	1111410650	22/07/2019	امجد صبرى	426
250	رسوم كشف	1023482283	22/07/2019		

N : عيادة الركبة و الرباط الصليبي

د/ أحمد غنيم

كارت متابعة مريض

الاسم :
السن :
التاريخ :
التليفون :
الشكوى :

٢٠ / /



CS Scanned with CamScanner

12 References

References

- [1] “Adriot infosystems: eclinic,” 2008. [Online]. Available: <https://www.adroitinfosystems.com/products/eclinic-systems>
- [2] S. Khemka, “meddna,” 2010. [Online]. Available: <http://www.meddna.com>
- [3] C. M. Lay, “Clinic management system,” 2010.
- [4] S. Prasad, “clinic management srs,” oct 2010. [Online]. Available: <https://www.studentprojectguide.com/project-report/online-clinic-management-system-srs/>
- [5] G. RVR, “myhcue,” 2015. [Online]. Available: <https://www.myhcue.com/clinic-management-system>