

Software Requirement Specification Document

V2 Hospital System Web Application

abdelrahman ehab,ziad mohammed,youssef mohammed,omar saeed

March 2020

1 Introduction

1.1 Purpose of this document

the purpose of this Software Requirements Specification document is to show our hospital system specifications and how it shall be used and who will be using it .

1.2 Scope of this document

Currently Hospital system is using a manual system to handle the hospital process . instead of patients going to the hospital to reserve an arrangement with the doctor. Then the patients diagnosed symptoms related disease details and other necessary details are being recorded and those are being recorded in a file. Calculation of bills are done manually.these files can be easily destroyed by fire and natural disasters, Also admin and receptionist can view data of users and patients. they may lose data and information.the amount of files used to save the information waste a huge space.it takes time to check previous record of patients ,or add new records ,or update records.its also tough.its hard to relay on manual calculations done by the staff and receptionists. our system shall overcome the above mentioned problems.

1.3 overview

our system shall help patients to arrange an appointment in the hospital through the web application to consult the chosen doctor .firstly the user creates an account then sign in to the system where the user shall find different types of treatment specialists , their descriptions and available times then the user shall choose the doctor they desire after the user reserve the doctor the user shall receive the description with the price, the appointment time and number of patients waiting before him for the reservation. some kind of staff are allowed to view data or Edit, add, and delete it.

1.4 business context

the purpose of this application is to help patients reserve an appointment fast ,avoid wasting time in the reservation process. manage data of users and patients on the system by adding, deleting, and editing it,financially it reduces the amount of paper work.

2 General Description

2.1 Product Functions

- Users Data Management : (View data / add new data / Delete data)
- Patient Management : (Inpatient: appointments, reservations, registrations, discharge, payment history)
- Patient Record Editing / updating / deleting : (Patient History, treatment required, preferred doctor)
- Clinical Management : (Operation theatre availability, medical records, personnel management, laboratory management)
- Billing Management : (generate a bill, transmit bill to insurance, and record payment)
- Personnel Management : (Doctors records, operations schedule, Doctors Daily Schedule, Doctor availability i.e. day date and time)
- Rooms Availability Management : (add / delete ward, add / delete room from ward including number of beds in room)
- Doctor attendance : (doctor shall mark beside his name in his profile)
- Staff management : (update job description of Doctors / tracks recruiting records)
- Reporting : (Manage and edit your lab test reports)
- Patient Portal : (patients information plus their reports)

2.2 Similar System Information

systems that our system is similar to:

- Saudi German hospital [2]
- mayo clinic [3]
- Indian University health [1]
- northwestern medicine [5]
- mount Sinai hospital [4] But it can be different in functionalities.

2.3 User Characteristics

1. Admin :

- Admin has the full access to the system which means he is able to manage any activity with regard to the system. He is the highest privileged user who can access to the system.

Key functions:

- Manage Staff, patients and equipment
- Manage the web application
- Allocate resources
- Administer the charges
- Monitor Performance
- Provide Technical Support
- Generate reports
- Manage doctors
- Employee Interacts with the systems most often to supply service to customers
- Keep track of patient details
- Keep tract of test details
- Make ambulance reservations

2. Doctor :

- Doctors has a very limited access to the system, as he has few control on it.

Key functions:

- Edit account details
- Manage reservations
- Mark his attendance
- writes reports about the condition of the patient

3. Receptionist :

Key functions :

- receptionist help patients to confirm their reservation
- confirm appointment
- calculate and print bill

4. patient:
 - reserve appointment
 - Manage reservation(cancel the reservation)
 - see receipt of his service.
 - Register with site
 - view appointment history

2.4 User Problem Statement

This system provides solutions for all above mentioned problems, the client shall want the system to move from paper based to electronic medical records to help them to provide premium health care to their patients and improve the medical services offered. It provides customer comfort for reservations, bill payment, and decrease the waiting time by knowing how many people are waiting before his appointment. All of which should be easy to use, and handle the system crash.

2.5 User Objectives

1. Reservations management :
 - Fast and easy to reserve an appointment
 - Different appointment time to each patient
2. Analysis System :
 - analysis on Each user information
 - analysis on all reservations happened on system
3. Accounts Management :
 - full access on users accounts (Add, Edit or Delete)
 - Manage reservations (Delete)

2.6 General Constraints

Each user should have individual Username and password,
Only administrator can access the whole system (Admins),
the System is only accessible within the hospital premises only and User should have an account to access the system

3 Functional Requirements

- The user access system process takes place either by Sign-up or Registration or can sign out from system by just one click.

| | |
|----------------|---|
| Function | sign up |
| Description | Add new user to the system so he could login with same identity later on and do system's operations which is reserve an appointment |
| Input | username, password , phone, full name |
| Output | save data on database and tell user whether the process was accepted or not |
| action | checks if the username entered by the user is already registered , if it is ,it tells the user to enter another username , if it isn't its proceeds to home page. |
| precondition | the username must not be already registered on the database, and Server must be connected to the Database. |
| post-condition | the username information should be saved in the database according to the validations required |

| | |
|----------------|---|
| Function | sign in |
| Description | user shall be able to sign in after checking his username and password are in the database |
| Input | username, password entered by user |
| Output | directs to the hospital system main page |
| action | check if username and password are in the database |
| precondition | the username that the user enters should be registered on the database, and Server must be connected to the Database. |
| post-condition | the user is signed in, and can do system's operations |

| | |
|----------------|---|
| Function | Password Hashing |
| Description | after the user register to the system, the password is hashed so it could be secured |
| Input | Password of user |
| Output | Encrypted password |
| action | check if username and password are in the database, and Server must be connected to the Database. |
| precondition | the user enters his password |
| post-condition | None |

| | |
|----------------|---|
| Function | sign out |
| Description | the user shall be able to sign out |
| Input | the user presses the sign out button |
| Output | returns the user to the sign in page |
| precondition | the user should be signed in, and Server must be connected to the Database. |
| post-condition | the user isn't signed in anymore |

- Insert new user information to the system either by sign-up or add the information by the administrator (admin). This process takes various functions to be done.

| | |
|----------------|---|
| Function | Insert user |
| Description | This function is accessed by admin, he can be able to create new user to system |
| Input | Insert user |
| Output | User ID of the user to be entered |
| action | these data shall be added to the database |
| precondition | the username 'shouldn't be in database already, and Server must be connected to the Database. |
| post-condition | the username shall be added to the database, username and password will be sent to user |

| | |
|----------------|--|
| Function | Insert user value |
| Description | Insert user information in the database table called "Users" according to user's entered automatically ID in database. |
| Input | The form information and user previous entered automatically ID from database |
| Output | create new row in database with the new user's information added by administrator (admin) |
| action | input all required user information and follow the validations |
| precondition | input of information should match the Validations, and Server must be connected to the Database. |
| post-condition | User information shall be added to the database, so user can sign-in later |

- Deleting user from the system by the administrator (admin) or deleting account by the user itself. This process takes various functions to be done.

| | |
|----------------|--|
| Function | Delete user from system |
| Description | only the administrator shall be able to delete any user from the system and database |
| Input | pressing on delete user button and confirm deletion |
| Output | Delete user value |
| precondition | the user shall have an account registered in the database, and Server must be connected to the Database. |
| post-condition | the user information shall be deleted from the database |

| | |
|----------------|---|
| Function | Delete user value from system |
| Description | only the administrator shall be able to delete any user from the system and database |
| Input | Selecting user ID |
| Output | a pop-up saying that the user has been deleted successfully |
| precondition | the user shall have an account registered in the database, Server must be connected to the Database, and user Data must be already existed in the Database. |
| post-condition | the user account information shall be deleted from the database |

| | |
|----------------|---|
| Function | Deleting account |
| Description | User only can delete his account if he want to |
| Input | pressing Deleting account button |
| Output | a pop-up to confirm deletion and the another pop-up saying that the account has been deleted successfully |
| precondition | the user shall have an account registered in the database, Server must be connected to the Database, and user Data must be already existed in the Database. |
| post-condition | the user account information shall be deleted from the database |

- Update user information either by administrator (admin) or by user itself.
This process takes various functions to be done.

| | |
|----------------|---|
| Function | Update user information(administrator) |
| Description | Admin can edit the user's information id assigned to each user in table users in the database |
| Input | User ID and the section is to be edited |
| Output | the selected row in database is updated |
| action | Confirm Editing user's information |
| precondition | pressing confirm button after filling the section selected to be edited, Server must be connected to the Database, and user Data must be already existed in the Database. |
| post-condition | User's information is updated successfully according to each row validations |

| | |
|----------------|--|
| Function | Update user profile (user) |
| Description | User update his own information by himself in his profile and the this information is automatically updated in the database |
| Input | username, password , phone, full name |
| Output | pop-up message saying information is updated successfully |
| action | User updates his own information in profile |
| precondition | user must be authenticated, Server must be connected to the Database, and user Data must be already existed in the Database. |
| post-condition | None |

- view all users only by administrator (admin). admin can select a specific user to view all his information

| | |
|----------------|--|
| Function | view user profile |
| Description | Admin view any user profile he selects on the system |
| Input | User ID / pressing edit button beside user's information on the system |
| Output | User information / profile |
| action | admin views user's profile |
| precondition | only admin can make this action, and the selected user must not be already deleted, Server must be connected to the Database, and user Data must be already existed in the Database. |
| post-condition | None |

| | |
|----------------|---|
| Function | view all users in system |
| Description | Admin view all users profile on the system |
| Input | None |
| Output | Array of all users |
| action | admin views all users |
| precondition | only admin can make this action, and the selected user must not be already deleted, and Server must be connected to the Database. |
| post-condition | None |

- view all doctors on system available with days and times, so it will be easy for the user to choose a suitable appointment with any doctor. This reservation will be added to the user's profile so he can check date and time and doctor information as long as he sign-in.

| | |
|----------------|---|
| Function | Reserve appointment |
| Description | user shall check the doctors available then select the doctor he wants to reserve with |
| Input | User name and password |
| Output | the system outputs a receipt with the information of the reservation |
| action | |
| precondition | the doctor should be available for reservations, and Server must be connected to the Database. |
| post-condition | a reservation time should be saved on the database with the doctor name and patient name and time for the reservation |

| | |
|----------------|--|
| Function | view reservation |
| Description | user shall be able to view his reservation |
| Input | pressing reservations button on his profile or in navigation bar |
| Output | pop-up message with the user's reservations |
| action | |
| precondition | user must reserve an appointment with a doctor or no results will appear in the pop-up message , Server must be connected to the Database, and Reservation Data must be already existed in the Database. |
| post-condition | None |

| | |
|----------------|--|
| Function | delete reservation |
| Description | user shall be able to delete his reservation |
| Input | pressing cancel appointment button on the system |
| Output | a pop-up telling the user that the reservation has been deleted successfully |
| action | |
| precondition | there should be a reservation saved on the database, and Server must be connected to the Database. |
| post-condition | the reservation should be deleted from the database successfully |

- only Administrator (admin) can Add, edit or delete Doctors accounts from system and also from database automatically. Also admin can view the Doctor's list and update it frequently.

| | |
|----------------|--|
| Function | Doctors List |
| Description | a list of the doctors that shall be updated frequently |
| Input | the admin shall input the information of the doctors and update it |
| Output | a list of the doctors with information about them |
| action | |
| precondition | doctors information should be saved on the database, and Server must be connected to the Database. |
| post-condition | if the admin updates the list it should be updated in the database |

| | |
|----------------|--|
| Function | Edit Doctor information |
| Description | a list of the doctors that shall be updated frequently |
| Input | the admin shall input the information of the doctors and update it |
| Output | a list of the doctors with information about them |
| action | |
| precondition | doctors information should be saved on the database ,Server must be connected to the Database, and Doctor information must be already existed in the Database. |
| post-condition | if the admin updates the list it should be updated in the database |

| | |
|----------------|--|
| Function | Delete Doctor value from system |
| Description | only the admin shall be able to delete any Doctor from the system and database |
| Input | Selecting Doctor ID |
| Output | a pop-up saying that the Doctor information has been deleted successfully |
| precondition | the Doctor shall have an account registered in the database, and Server must be connected to the Database. |
| post-condition | the Doctor account information shall be deleted from the database |

- paying process has many functions to be done. when user pay by cash in the hospital, the receptionist will print a bill receipt for him as the hardware used is the printer.

| | |
|----------------|--|
| Function | payment |
| Description | |
| Input | receptionist shall enter the reservation price |
| Output | none |
| action | each doctor has it's own payment price save in database, this will appear when opening bill |
| precondition | user must have done a reservation for a specific doctor, and Server must be connected to the Database. |
| post-condition | Go to print bill page |

| | |
|----------------|--|
| Function | print bill |
| Description | Receptionist will print the bill using the hardware which is the printer |
| Input | user name, Doctor name, and date, time of reservation, and price form database |
| Output | a printed receipt given to user |
| action | printing process |
| precondition | printer connected to the computer, and Server must be connected to the Database. |
| post-condition | give the patient the receipt of the payment |

- front desk Receptions make new profiles for patients who didn't register yet, also they print reports for patients

| | |
|----------------|--|
| Function | Receptionist sign in |
| Description | each Receptionist sign in with his username and password are in the database |
| Input | username, password entered by receptionist in his shift |
| Output | directs to the user's data, reservations, and check availability of Doctors list |
| action | None |
| precondition | Server must be connected to the Database, and Receptionist Data must be already existed in the Database. |
| post-condition | None |

| | |
|----------------|---|
| Function | new user profile |
| Description | Receptionist can create a profile instead of user at the hospital |
| Input | user sign-up form information as the patient tells the receptionist |
| Output | As when user sign-up |
| action | as when user sign-up |
| precondition | Server must be connected to the Database. |
| post-condition | receptionist gives the patient his username and password in case he wants to make another appointment later, receptionist can reserve an appointment as the patient tells him |

| | |
|----------------|---|
| Function | print Report tests |
| Description | Receptionist will print the Report using the hardware which is the printer |
| Input | None |
| Output | a printed Report given to user |
| action | printing process |
| precondition | printer connected to the computer, and Server must be connected to the Database |
| post-condition | give the patient the Report of his tests |

- More information by pressing buttons in the navigation bar which are contact us, emergency, and location of hospital.

| | |
|-------------|--|
| Function | contact us |
| Description | a way to help the user to contact with someone from the hospital if needed through the website |
| Input | the user should be signed in |
| Output | a response from the person assigned to answer the users |

| | |
|-------------|---|
| Function | emergency |
| Description | a button that shows the number of the hospital to contact with them if needed |
| Input | the user should press on the emergency button |
| Output | a pop-up showing the number of the hospital |

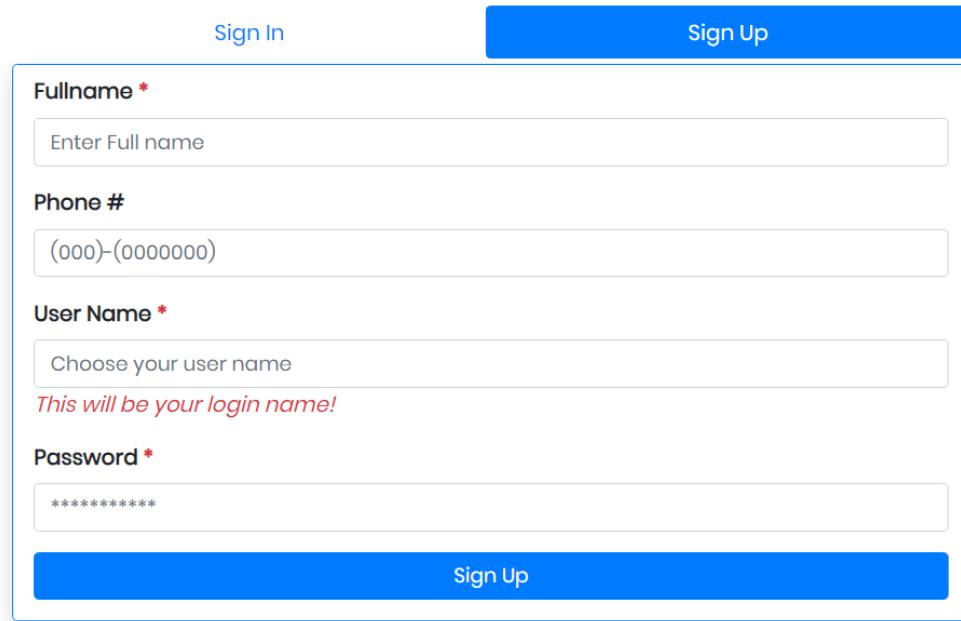
| | |
|-------------|--|
| Function | location of the hospital |
| Description | a button that shows the location of the hospital in maps |
| Input | the user should press the button named location |
| Output | a pop-up showing the location |

- Reports are done by doctors or admin but mainly by doctors

| | |
|----------------|---|
| Function | Write report |
| Input | User ID, and name |
| Output | Reporting page |
| action | Write report on the patient profile |
| precondition | Server must be connected to the Database. |
| post-condition | report will be available on patient's profile so he can view it |

4 Interface Requirements

4.1 GUI



The sign up form consists of several input fields and a large blue 'Sign Up' button at the bottom. At the top left is a 'Sign In' button. The first field is 'Fullname *' with a placeholder 'Enter Full name'. The second field is 'Phone #' with a placeholder '(000)-(0000000)'. The third field is 'User Name *' with a placeholder 'Choose your user name' and a note below it: 'This will be your login name!'. The fourth field is 'Password *' with a placeholder '*****'. A large blue 'Sign Up' button is located at the bottom right.

Sign In

Sign Up

Fullscreen *

Enter Full name

Phone #

(000)-(0000000)

User Name *

Choose your user name

This will be your login name!

Password *

Sign Up

Figure 1: Sign up



The sign in form consists of two input fields and a large blue 'Sign In' button at the bottom. At the top left is a 'Sign In' button. The first field is 'username *' with a placeholder 'Enter Full name'. The second field is 'Password *' with a placeholder '*****'. A large blue 'Sign In' button is located at the bottom right.

Sign In

Sign Up

username *

Enter Full name

Password *

Sign In

Figure 2: Sign in

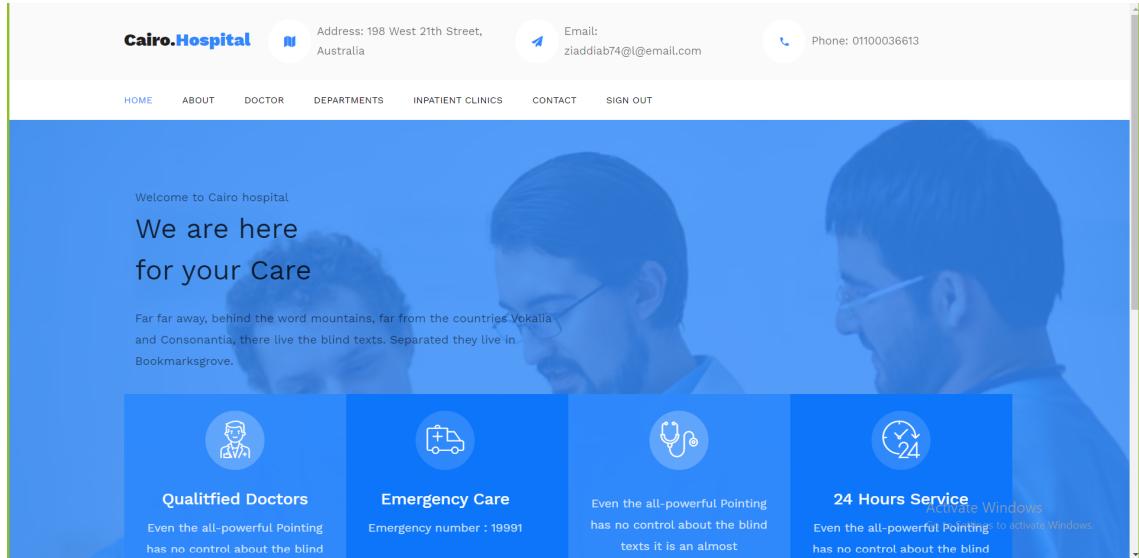


Figure 3: home page

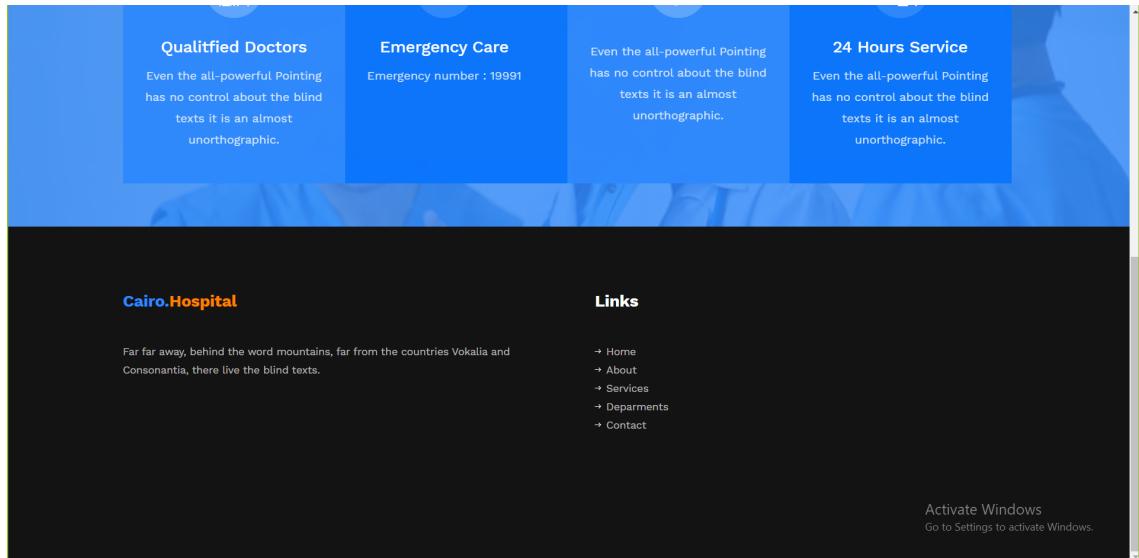


Figure 4: home page

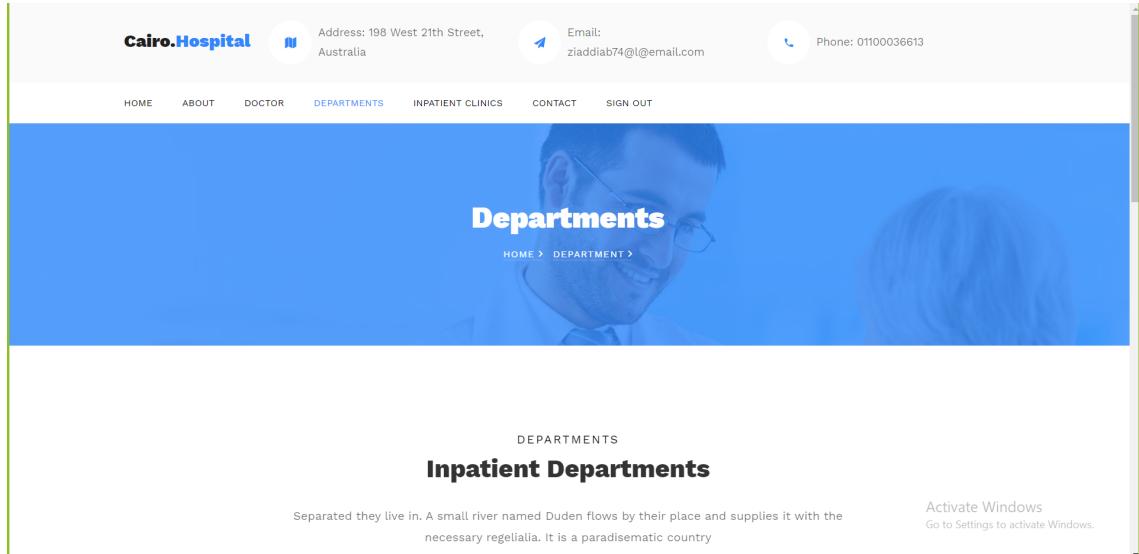


Figure 5: departments

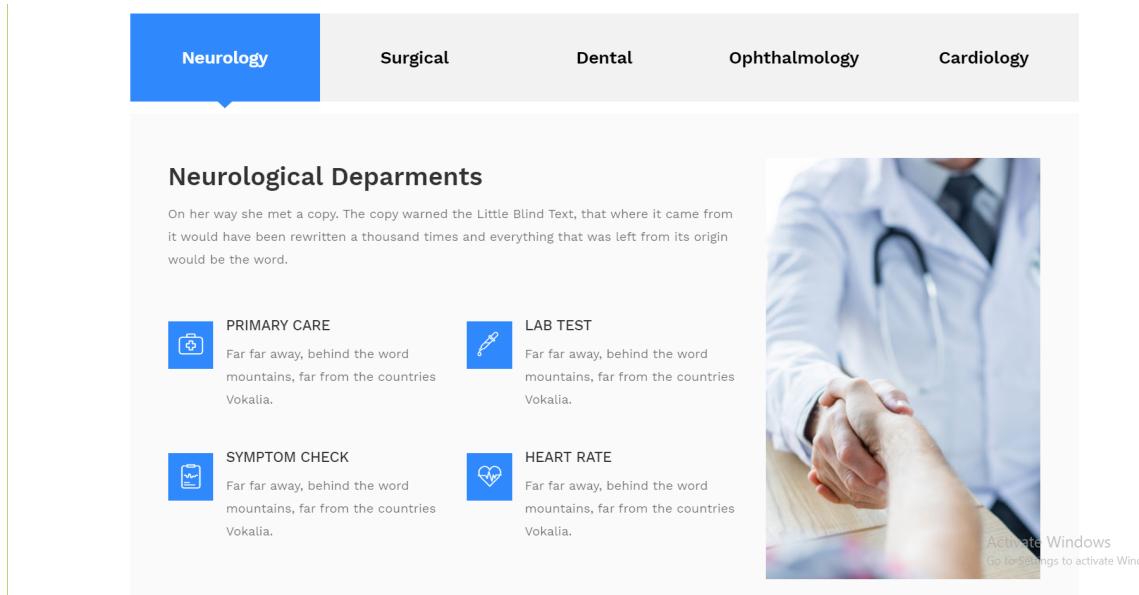


Figure 6: departments

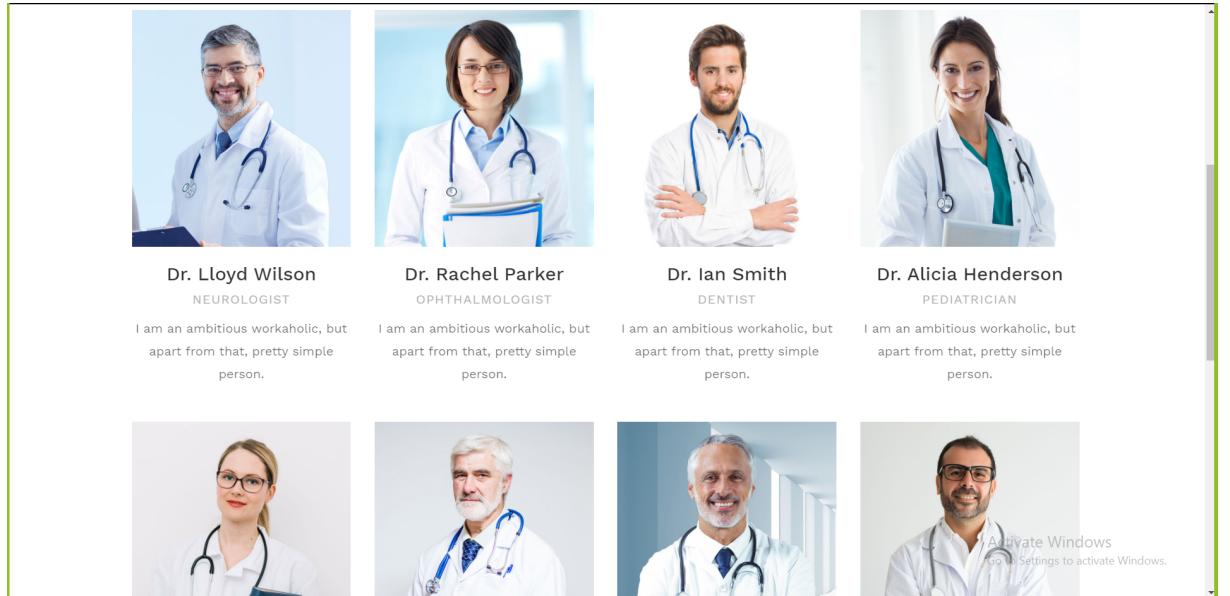


Figure 7: doctors

| DR.KHALED | DR.YOUSSEF | DR.ZIAD |
|---|-----------------------------------|-----------------------------------|
| Department:Skin | Department:Skin | Department:Skin |
| Designation:Specialist | Designation:Specialist | Designation:Consultant |
| Recent Positions: | Recent Positions: | Recent Positions: |
| Asst. Lecturer of surgery MedicineAsst. | Anaesthesia Specialist SGH-Jeddah | Anaesthesia Specialist SGH-Jeddah |
| Lecturer of surgery MedicineSuez Canal univ | Anaesthesia Resident, El Hela | Anaesthesia Resident, El Hela |
| Book | Book | Book |

| DR.MOHAMED | DR.AHMAD MOHAMED | DR.KHALED |
|------------------------|------------------------|------------------------|
| Department:Skin | Department:Skin | Department:Skin |
| Designation:Consultant | Designation:Consultant | Designation:Specialist |
| Recent Positions: | Recent Positions: | Recent Positions: |
| Book | Book | Book |

Figure 8: inpatient

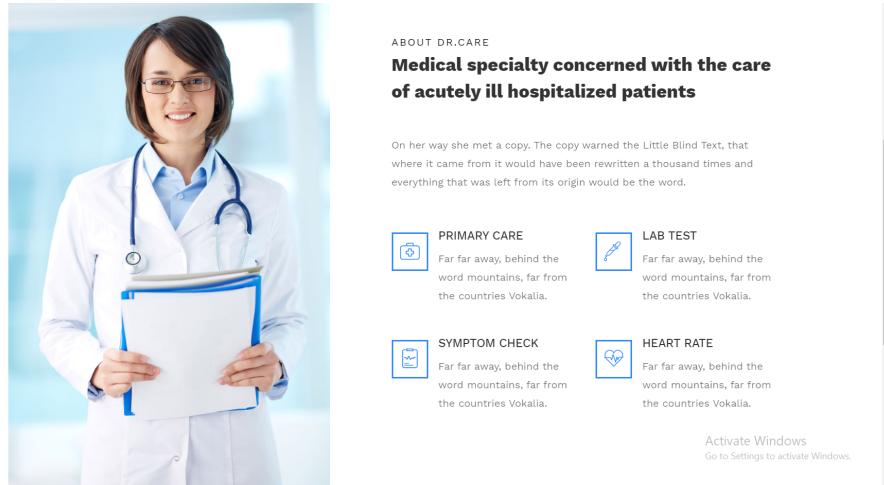


Figure 9: about us

A screenshot of a medical website's 'Contact Us' page. The background features a blue-tinted image of a doctor and a patient. The main title 'Contact Us' is centered at the top. Below it, a navigation bar shows 'HOME > CONTACT US'. The contact form starts with a 'Name' field and a 'Message' field, both represented by simple input boxes. At the bottom, a large blue button contains the text 'Send Message'. To the right of the button, there is a 'Activate Windows' button with the text 'Go to Settings to activate Windows.'

Figure 10: contact us

4.2 Hardware Interfaces

this system can run on ordinary computer and a printer is used to print bills and Reports for users from the system

5 Performance Requirements

the system should be using an accepted amount of memory from the client server also it should be reliable in performance and that its processes doesn't take much time.

6 Design Constraints

the system can run on any device .

7 Non-Functional Requirements

7.1 security

admins should have a unique username so that the system knows whose logged in to prevent intruders from having the administrator privileges

7.2 Maintainability

the system can maintain any Data despite how big it is. the code shall be written in an understandable way to help in the ability to maintain ,modify information, update and fix problems of the system,the system shall also provide capability to back-up the data

7.3 Re-usability

the system can be used again as much as the user and the admin wants without distortion ,the system shall be easy to use so that the user can learn how to use it fast and should be organised in such a way that user errors are minimized.

7.4 Availability

the system should be available at all the time,except a few minutes to maintain the system monthly .

7.5 reliability

the output of information should be accurate

8 Preliminary Object-Oriented Domain Analysis

8.1 Inheritance Relationships

see the inheritance relationships in figure 11

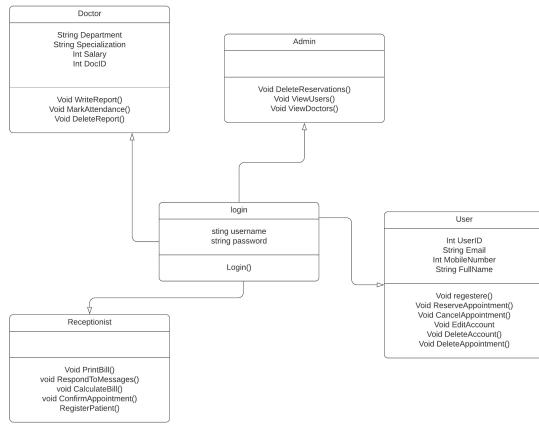


Figure 11: inheritance

8.2 Class descriptions

see the class diagram below at figure 12

9 Operational Scenarios

click to see the use case diagram in figure 13

10 Preliminary Schedule Adjusted

| task name | Finish date |
|-----------|-------------|
| SRS | 19/3/2019 |
| SRS V2 | 25/5/2020 |

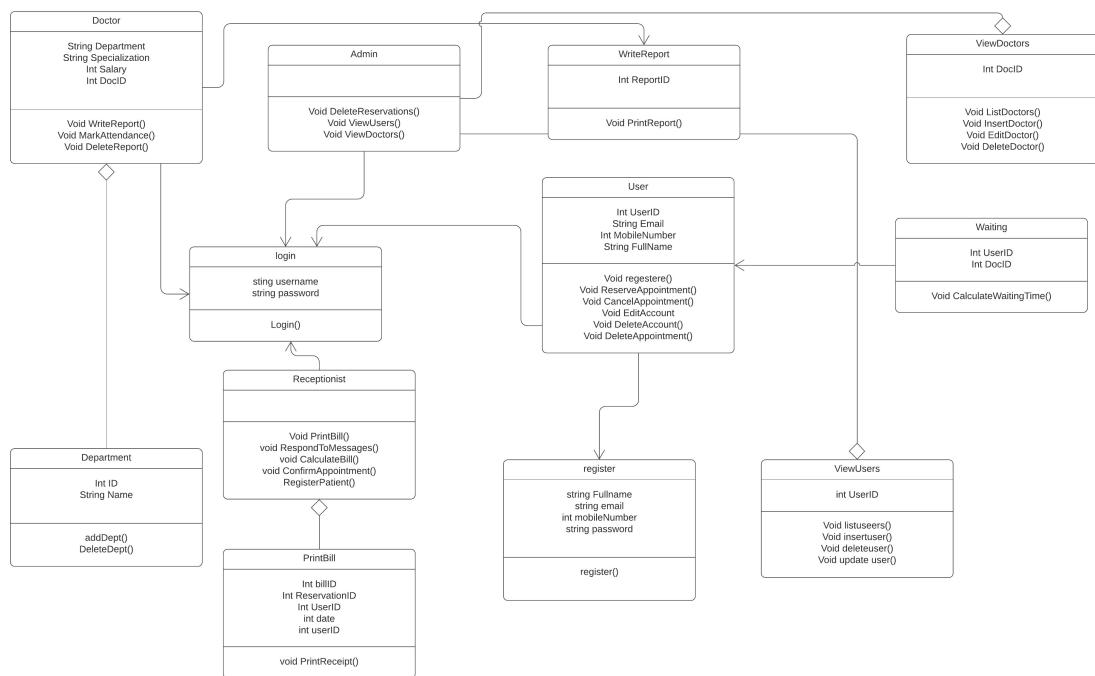


Figure 12: class diagram

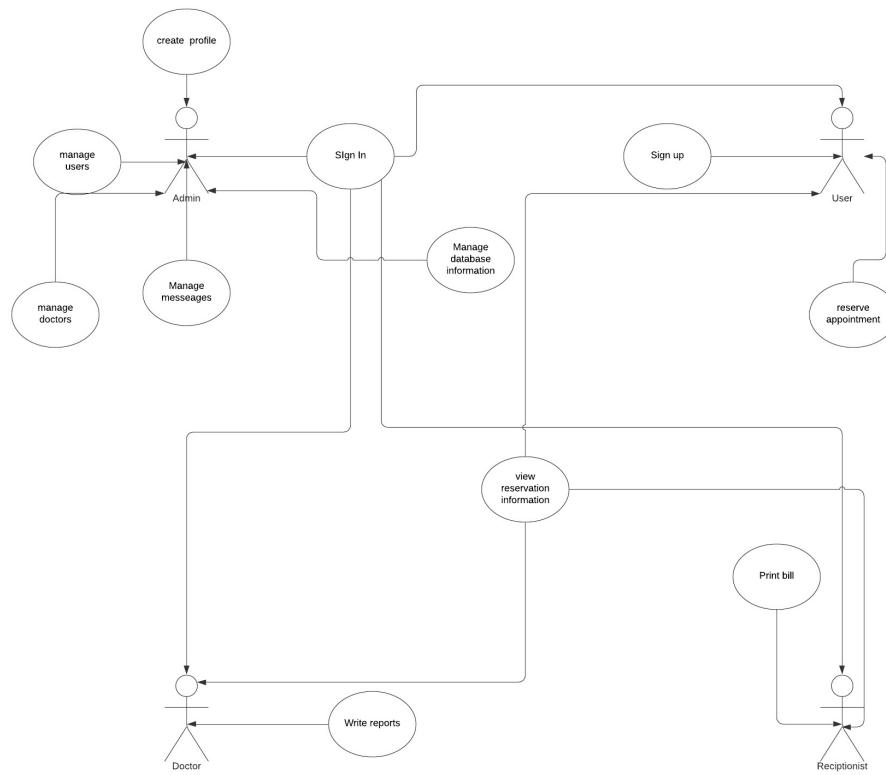


Figure 13: use case

11 References

References

- [1] [https://iuhealth.org/.](https://iuhealth.org/)
- [2] [https://sghcairo.com/en/.](https://sghcairo.com/en/)
- [3] [https://www.mayoclinic.org/.](https://www.mayoclinic.org/)
- [4] [https://www.mountsinai.org/locations/mount-sinai.](https://www.mountsinai.org/locations/mount-sinai)
- [5] [https://www.nm.org/.](https://www.nm.org/)