Logo

Description automatically generated with medium confidence

### Jordan University of Science and Technology

### College of Computer Sciences & Information Technology

# Health Care Tracking System

A picture containing company name

Description automatically generated

A project submitted in partial fulfillment of the

requirements for the degree of Bachelor in

Software Engineering

|  |  |
| --- | --- |
| Student names | Id number |
| Haneen Fakhri JARADAT | (125590) |
| Sewar Emad Alfdawi | (123412) |
| Selina Mustafa Alish | (140609) |
| Sara Sameh Aldhoun | (126744) |

**Supervised by**

Dr.Hamza Alkofahi

Note: SANI MEDICO is mean health and medicine

### ACKNOWLEDGEMENT

We would like to express our deepest appreciation to all those who provided us the possibility to complete and develop our system "SANI MEDICO Health care system".

Special gratitude to our supervisor Dr.Hamza alkofahi for his persuaded and continuously guided us during the whole course of our project.

We would also like to thank our families for their endless support.

### UNDERTAKING

We are site below sewar, Haneen, Selina and Sara developed (SANI MEDICO Health Care System) is an original work done by undersigned, in partial fulfilment of the requirements for the degree -bachelor’s in software engineering at Software Engineering Department, College of Computer and Information Technology, Jordan University of Science and Technology.

All the analysis, design and system development has been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

|  |  |
| --- | --- |
| Student Name | Student signature |
| Haneen Fakhri JARADAT | A picture containing text, businesscard  Description automatically generated |
| Sewar Emad Alfdawi | Text, whiteboard  Description automatically generated |
| Sara Sameh Aldhoun | Text, letter  Description automatically generated |
| Selina Mustafa Alish | A drawing on a piece of paper  Description automatically generated with low confidence |

## Table of Contents

**LIST OF TABLES** .........................................................................................................................6

**LIST OF FIGURES** .......................................................................................................................7

**CHAPTER 1: INTRODUCTION** ....................................................................................................9

1.1 Overview ...........................................................................................................................10

1.2 Motivation .........................................................................................................................10

1.3 Problem Statement ...........................................................................................................11

1.4 Project Aim and Objectives ...............................................................................................11

1.5 Project Scope ....................................................................................................................12

1.6 Project Software and Hardware Requirements ............................................................... 15 Hardware requirements .....................................................................................................15

Software Requirements …...................................................................................................15

1.7 Project Limitations ….........................................................................................................15

1.8 Project Expected Output ..................................................................................................15

1.9.1 Project Schedule ............................................................................................................16

1.9.2 Gantt Chart ....................................................................................................................16

1.10 Project, product, and schedule risks .............................................................................. 17

**CHAPTER 2: Related Existing System** .................................................................................... 18

2.1 Introduction ..................................................................................................................... 19

2.2 Existing Systems ............................................................................................................... 19

2.3 Overall Problems of Existing Systems .............................................................................. 22

2.4 Overall solutions of Existing Systems ............................................................................... 22

**CHAPTER 3: Recruitment Engineering and Analysis**.............................................................. 23

3.1 Stakeholders ..................................................................................................................... 24

1. System users .................................................................................................................. 24
2. Development team ........................................................................................................ 24

3.2 Use Case Description ........................................................................................................ 26 Use Case 1 ……….................................................................................................................. 26 Use Case 2 …….................................................................................................................... 27 Use Case 3 …....................................................................................................................... 28

Use Case 4 …....................................................................................................................... 29

Use Case 5 …....................................................................................................................... 29

Use Case 6 …....................................................................................................................... 30

Use Case 7 …....................................................................................................................... 31

Use Case 8/9 ...................................................................................................................... 31

Use Case 10 ........................................................................................................................32

Use Case 11 ....................................................................................................................... 32

Use Case 12 ….................................................................................................................... 32

3.2.1Activity Diagram ........................................................................................................ 34

3.3 Functional Requirements …...............................................................................................36 System Requirements …......................................................................................................36 patient Requirements .........................................................................................................36

Medical staff Requirements ................................................................................................36

Admin Requirements ….......................................................................................................36

3.4 Non-functional requirements ...........................................................................................37 Execution qualities …...........................................................................................................37

1.usability .......................................................................................................................36

2.privacy …………..............................................................................................................37

3.efficiency .....................................................................................................................37

4. Availability ..................................................................................................................37

5.security ….….................................................................................................................37

6.responsabolity .............................................................................................................37

Evolution qualities …...........................................................................................................37

4.maintainability ….........................................................................................................37

5. testability ………...........................................................................................................37

3.5 Constraints ….....................................................................................................................38

**CHAPTER 4: Architecture and design**......................................................................................39

4.1 Overview ……......................................................................................................................40 Logical view ……...................................................................................................................40

Process view ….....................................................................................................................40

Physical view …....................................................................................................................40

4.2 Software architecture .......................................................................................................41

4.2.1 Logical view ...............................................................................................................41 4.2.2 Process view …...........................................................................................................42

4.2.3 Physical view …...........................................................................................................43

4.3 Software design ................................................................................................................44

4.3.1 Sequence diagram .....................................................................................................44

4.3.2 Class diagram …..........................................................................................................47 4.3.3 ER diagram …..............................................................................................................48

4.3.4 State transition diagram ............................................................................................49

4.4 User Interface (Prototype) ................................................................................................51

**CHAPTER 5: Implementation Plan ……………………….**...............................................................59

5.1 Description of implementation …......................................................................................60

5.2 Programming language and technology ...........................................................................62

5.3 part of implementation if possible ....................................................................................63 **CHAPTER 6: Test planning** .....................................................................................................63

6.1 Black-box ...........................................................................................................................65

6.2 White-box ……....................................................................................................................71

6.3 Testing Automation …........................................................................................................72

**CHAPTER 7: Conclusion and result**..........................................................................................77

7. Conclusion and result ….......................................................................................................78

**CHAPTER 8: Reference** ………………...........................................................................................79

8. Reference ......................................................................................................................... 80

**LIST OF TABLES**

**Table 1-1**: Table of Main Objectives and Problems………………………………………………….12

**Table 1-2**: Project Schedule…………………………………………………………………………………….16

**Table 1-3**: Risk probability and impact…………………………………………………………………….17

**Table 2-1**: Sani medico Comparing with existing systems……………………………………….21

**Table 3-1**: Description of use case Register Patient…………………………………………………26

**Table 3-2**: Description of use case login………………………………………………………………….27

**Table 3-3**: Description of use case schedule appointment………………………………………27

**Table 3-4**: Description of use case view medical record………………………………………….28

**Table 3-5**: Description of use case Edit Personal Information………………………………….28

**Table 3-6:** Description of use case search for appointment…………………………………….29

**Table 3-7**: Description of use case login staff………………………………………………………….30

**Table 3-8/9**: Description of use case edit medical file for patient……………………………30

**Table 3-10**: Description of use case patient accept…………………………………………………31

**Table 3-11**: Description of use case Send reminder for patient……………………………….31

**Table 3-12**: Description of use case View all appointment………………………………………32

**Table 6-1**: register test case…………………………………………………………………………………….66

**Table 6-2**: login test case…………………………………………………………………………………………66

**Table 6-3**: schedule appointment test case…………………………………………………………….68

**Table 6-4**: view medical history test case……………………………………………………………….68

**Table 6-5**: edit personal information test case……………………………………………………….69

**Table 6-6**: login staff test case……………………………………………………………………………….69

**Table 6-7**: edit medical record for patient test case……………………………………………….70

**Table 6-8**: add to medical file test case………………………………………………………………….70

**LIST OF FIGURES**

**FIGURE 1-1**: Nurse uses a book to do record……….….…………………….………………………11

**FIGURE 1-2**: iterative model life cycle……………………….….……………………………………….14

**FIGURE 1-3**: Gant chart………………………………………….………………………………………………17

**Figure 2-1**: home page……………………………………………………………………………………………19

**Figure 2-2:** Schedule a Demo…………………………………………………………………………………20

**FIGURE 3-1**: HealthCare Use Case Diagram……………….……………….………………………….25

**FIGURE 3-2**: activity diagram of Use Case Register………………………………………………...34

**FIGURE 3-3**: activity diagram of Use Case personal information………………………...….35

**Figure 4-1**: logical view……………………………………………………………………………………………41

**Figure 4-2**: sequence diagram………………………………………………………………………………...42

**Figure 4-3**: communication diagram of edit personal information………………………….42

**Figure 4-4**: communication diagram of edit register patient………………………………....43

**Figure 4-5**: communication diagram of edit register patient………………………………….43

**Figure 4-6**: Healthcare tracking system Sequence Diagram for medical staff………….44

**Figure 4-7**: Healthcare tracking system Sequence Diagram for admin…………………….45

**Figure 4-8**: Healthcare tracking system Sequence Diagram for patient………………….46

**Figure 4-9:** Healthcare tracking system Class diagram…………………………………………….47

**Figure 4-10**: Healthcare tracking system ER Diagram…………………………………………….48

**Figure 4-11**: State diagram for register patient………………………………………………………49

**Figure 4-12**: State diagram for personal information…………………………………………….50

**Figure 4-13**: Sani medico interface…………………………………………………………………………52

**Figure 4-14**: login form for medical staff…………………………………………………………………52

**Figure 4-15**: login for admin……………………………………………………………………………………52

**Figure 4-16**: login form for patient………………………………………………………………………….53

**Figure 4-17**: personal information for patient…………………………………………………………53

**Figure 4-18**: medical information for patient………………………………………………………….54

**Figure 4-19**: department of treatment……………………………………………………………………54

**Figure 4-20**: name of doctor……………………………………………………………………………………55

**Figure 4-21**: doctor schedule…………………………………………………………………………………..55

.

**Figure 4-22**: patient manage account……………………………………………………………………..56

**Figure 4-23**: about Sani Medco……………………………………………………………………………….56

**Figure 4-24**: patient booking…………………………………………………………………………………..57

**Figure 4-25**: medical file for patient………………………………………………………………………..57

**Figure 4-26**: patient account (communication with doctor) …………………………………..58

**Figure 4-27**: appointment schedule for doctor……………………………………………………….58

**Figure 5.3.1**-: Download Xammp …………………………………………………………………………...60

**Figure 5.1.1**-: Start Apache and start MYSQL database……………………………………………61

**Figure 6-1**-: Black Box Testing Techniques………………………………………………………………65

**Figure 6-2**:login………………………………………………………………………………………………………67

**Figure 6-3**-: white box testing…………………………………………………………………………………71

# CHAPTER ONE

A picture containing graphical user interface

Description automatically generated

**Introduction**

**1.1 overview**

Health care is considered one of the necessary priorities in life. Human health is the most precious of what we have. Any person in us may be exposed to a disease. Therefore, we undertook this project to shorten the time for patients who suffer from long waiting and overcrowding that occurs in the hospital, as through this system the patient can communicate with the doctor easily and can make for the doctor to view the patient’s file, diagnose his condition, and set a suitable date for treatment.

**1.2 Project Motivation**

The technology is taking a large space in all industries and business, the medical technology also, took a commendable advance to manage the healthcare system and to adjust the huge numbers of data flowing from patients in several health sectors this will reduce the pressure laying on the medical staff and provide a better medical service. **figure 1.1**

To achieve this goal, we thought that building a website that facilitates and provides a convenient way to share the information with the physicians who treat them.

This website will contain the medical history which will reduce the misdiagnosis that would happen and provide a better understanding of the patient condition by the physicians and a better outcome to patient condition. in addition to previous, the website will Keep data confidential by making username, a password for the patient file, encrypted patient history to prevent unauthorized access.

On the other hand, the website is helpful because it helps the patient who has poor medical literacy, the patient who cannot express themselves properly, who have specials condition (Alzheimer’s disease, deafness) and make it easier to the patient family to understand his/her condition.

Our project is important because it reduces time and effort, makes better communication between doctor-patient to reach high-quality medical services.



**Figure 1-1 Nurse uses a book to do record**

**1.3 Problem Statement**

Most patients have difficulty explaining their health status and treatment methods we believe that we should find a solution for those patients to make their lives easier, the patient owns a file that contains all chronic disease and way of treatment, allergy.

Also, the physician can understand the medical history that reduces misdiagnosis That saves time and effort.

**1.4 Project Aim and Objectives**

The main objectives of the Health Care Tracking System are to improve the currently available system that lacks some innovations which are more likely required in the clinic.

|  |  |
| --- | --- |
| Objectives | Problems |
| 1-To provide a systematic scheduling system. | 1) Patients get frustrated for waiting longer in the clinic because there is no system keep tracks their waiting time.  2) Some patients which are quite suffer and making appointment with doctor cannot be schedule properly. |
| 2) To provide a medical file for patient | 1-patients do not remember information about their cases (Information Daily Staff Writer), which leads to losing valuable information.  2- The patient is not able to properly communicate the information to the doctor |
| 3) To reduce the workloads and mistakes made by the nurse by providing information received from Identity Card. | 1) Filling up the patient details manually consumes a lot of time. 2) The nurse might key in the wrong details of patient. |

**Table 1-1: Table of Main Objectives and Problems**

**1.5 Project Scope**

**1.5.1 Feature**

The system comes up with the following features, a patient can register in system the (username, password) also doctor register in the system while patient login profile a patient can see medical history (cannot edit), can edit personal information, give accessibility medical history to doctor.

On other hand, while a doctor access to history he can add, edit in medical history.

Admins make and organize an appointment by phone or by website.

\*\*note medical history include (chronic disease, way of treatment, allergy, blood type)

**1.5.2 Not included**

Medical history does not include x-ray, disorder genetic the system cannot detect patient location.

**1.5.3 Responsibilities**

In " Health care tracing system “there is a role for each team member:

**Supervisor**

DR-Hamza al\_kofahi

**Project Manager**

Sewar Alfdawy

Haneen Jaradat

**Developer**

Sewar alfdawy

Haneen jaradat

Selina alush

Sarah aldhoun

**Analysts**

Sewar alfdawy

Haneen jaradat

Selina alush

Sarah aldhoun

**Tester**

Sewar alfdawy

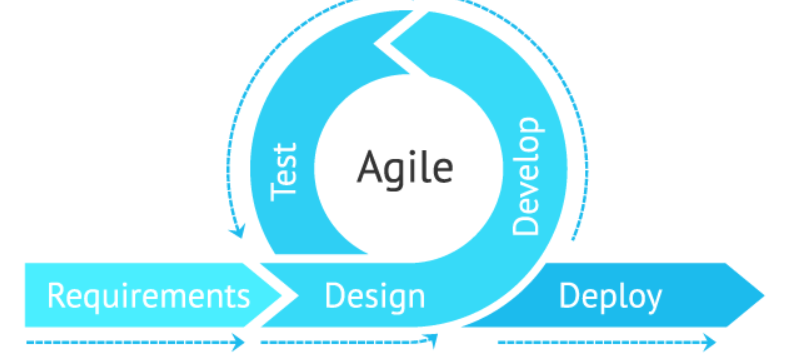
Haneen jaradat

Selina alush

Sarah aldhoun

**1.5.4 Methodology used.**

**We follow SDLC methodology to develop our project.**



**Figure 1-2 iterative model life cycle**

In our project, we used Agile Methodology it is a modern software methodology. Agile so flexible we change the scope of work according to new requirement the project will divide into a small cycle, team member have a clear vision about the project, and the customer can change requirement or accept team suggestion, so Agile is mindset productivity that consists the following stage.

* Requirement phase what project support to do. its gathering and analysis phase with manger user stockholder.
* Design phase based on user requirement the logical design system convert to a physical system.
* Implementation phase the developer get start write code using a different language.
* Testing phase after development test software and provide feedback to the developer team.
* Deployment phase After the testing gets completed, the product is handed over to the client
* Maintenance phases make changer to software are require making.

**1.6 Project Software and Hardware Requirements**

**Hardware needed (Hosting websites):**

A-web server

B-sql server.

c-internet connection

**software requirement:**

A-PHP8

B-MySQL

C- HTML5

D-CSS3

C-MYSQL

**User:**

A-smart phone or(laptop)

B-web browser

C-internet connection

**1.7 Project limitations**

1. Concerns about accuracy of the information.
2. In emergency situation the website cannot detect patient location.
3. The medical history does not include X-RAY, disorder genetic.
4. Communication problem between target server and web browser that’s mean the server may taking long time to send response to client so opening website may take some time.
5. Inability to use facial recognition or fingerprint detection to login patient.
6. Poor technology skills with users.
7. GUI will be in one language (Arabic).

**1.8 Project Expected Output**

1. Clearly Graphical User Interface.
2. Creating website that provide health care service.
3. Easily to make appointment.
4. patient Communication with her/his doctor that increase efficiency of health life.
5. Patient can access to his/her medical history anywhere.
6. Patient history will be secured from unauthorized access.
7. Our website will help patients with special need.
8. Medical history will reduce stress on physician.

**1.9 Project Schedule**

Start Day: 28-October-2020

Due Date: 13-January-2021

Project Client: Software Engineering department -JUST

Project Name: Health Care Tracking System (Sani Medico)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Finish Date | Start Date | Duration(day) | Task name | Task number |
| 31/10/2020 | 28/10/2020 | 3 | Introduction | 1 |
| 31/1/2020 | 31/10/2020 | 1 | Related Existing  System | 2 |
| 13/11/2020 | 1/11/2020 | 12 | Requirement  engineering and  analysis | 3 |
| 21/11/2020 | 13/11/2020 | 8 | Architecture and  Design | 4 |
| 2/12/2020 | 22/11/2020 | 10 | Implementation  Plan | 5 |
| 9/12/2020 | 2/12/2020 | 7 | Testing plan | 6 |
| 20/12/2020 | 10/12/2020 | 10 | Conclusion and  Result | 7 |
| 31/12/2020 | 21/12/2020 | 10 | References | 8 |
| 13/1/2021 | 1/1/2021 | 12 | Reviewing and  Approval | 9 |

**Table 1**-2**: Project Schedule**

**Chart, waterfall chart

Description automatically generated**

**Figure 1-3: Gant chart**

**1.10 Project, product, and schedule risks**

In all projects there will be some risks affecting the progression of the work and quality I will point to some of the risks that we faced and passed by team working and Patience.

1- Changing the requirement in the implementation phase.

2-The conflict between the project members.

3-Choose inappropriate person for specific task.

4-One of the team member faced a situation prevent her from complete her tasks.

|  |  |  |
| --- | --- | --- |
| Impact | Probabilities | Risk Number |
| High | 50% | 1 |
| Medium | 15% | 2 |
| Low | 5% | 3 |
| Medium | 12% | 4 |

**Table 1-**3**: Risk probability and impact**

**Chapter two**

A picture containing graphical user interface

Description automatically generated

**Related Existing System**

**2.1 introduction**

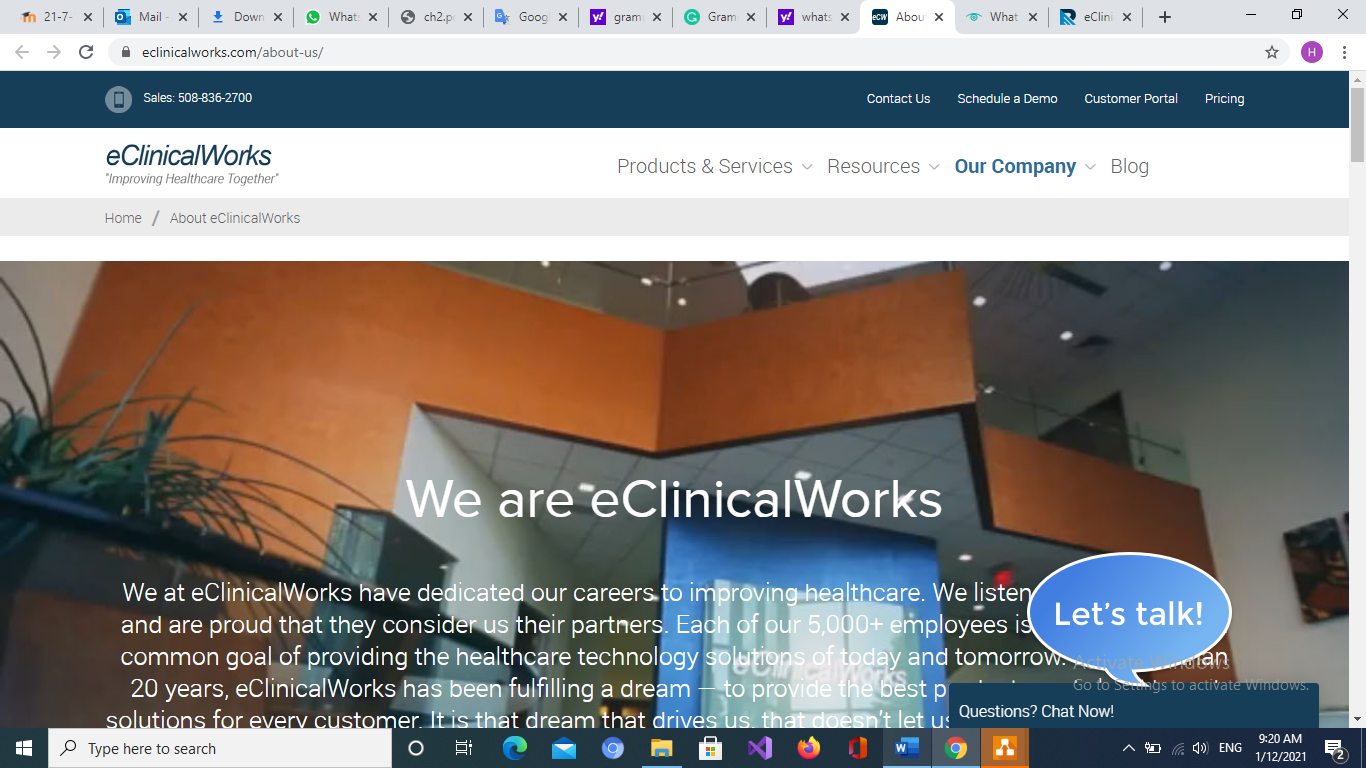
In this part, we are going to give an example of existing systems that in a way or another are like the proposed system and the problems that they solved and how our project solves some problems that the existing systems failed to solve and other extra features in our system. we worked much to provide a suitable solution to fill the gap that happens in the communication between the patient and the health care provider.

we have researched to find similar system have a lot of weakness and we tried to convert these weaknesses into strengths, we worked much to provide suitable solution to fill the gap that happen in the communication between the patient and the health care provider.

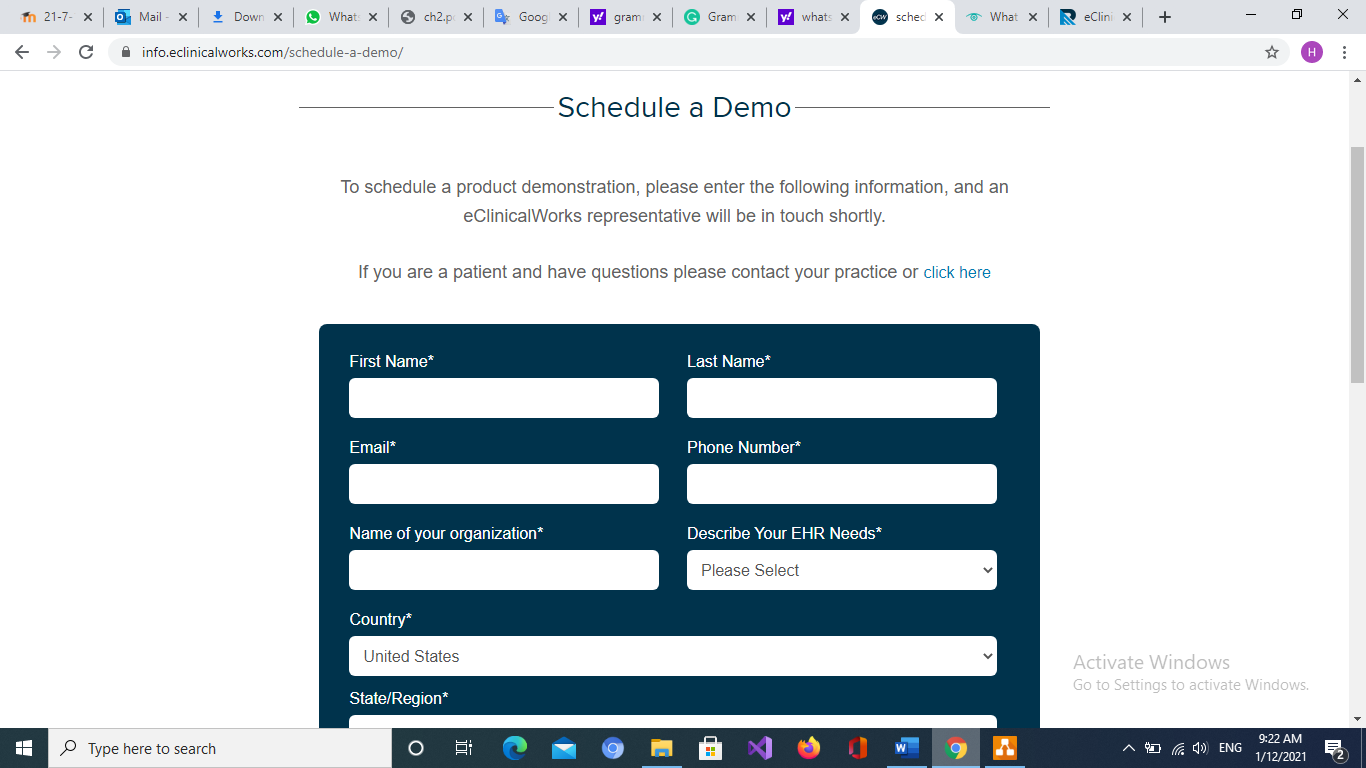
**2.2 Existing system**

**• eClinicalworks**

**eClinicalworks** Practice Management (PM) system is ideal for practices of any size. Manages appointments and schedules streamline billing and stores demographic data. eClinicalWorks is the market leader among independent, ambulatory practices. Our leading-edge technology and innovation ensure every aspect of patient care is under control, from scheduling and check-in through documentation, labs, prescribing, billing, and follow-up. And V11 comes with free clinic interoperability through the Common Well and Care quality nationwide networks.



**Figure 2-1: home page**



**Figure 2-2: Schedule a Demo**

This Table is a summary of the existing systems we searched for.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Features** | **eClinicalworks** | **Webpt** | **ClearCare** | **Careo** | **athenaClinicals** | **Sani medico**  **Our website** |
| Chat between patient and doctor | **×** | x | **×** |  |  | × |
| Choose the doctor |  | **×** | **×** |  |  | × |
| Send reminder for patient |  |  | **×** |  |  | × |
| Medical history. |  |  |  |  |  | × |
| Easy to use |  |  | **×** |  |  | × |
| Schedule appointment | **×** | **×** | **×** | **×** | **×** | × |

**Table 2-1: Sani medico Comparing with existing systems.**

**2.3 Over all problem existing system**

The primary problem in this existing system (eClinicalWorks) is the patient does not have a file for medical history to view to the doctor and when sending messages to the user, the eClinicalWorks program present failure in the direction of synchronization of user profile links, because when generating such action of sending, usually not all the necessary messages from the user’s display to notify, change of medication, a reminder of hospitalization payments or cancellation of bills for transfer of medicines, condition of care assessment of health progress, import of documentation to doctors on call, including problems of access to patient history, are evident within this programming when using their main tools of collaborative work, but our system brings more features to the existing systems.

**2.4 Over all solution approach**

Our System is featured by that it responds quickly, the patient has a medical history, send reminder for patient, remote order patient information management continuity, file upload functionality, organize appointments as appropriate for the doctor and patient, fill the gap between physician and patient by the doctor can access medical recorded and understand and diagnose the case, the patient can access their medical recorded every want and everywhere, patient give accessibility to the doctor to see history. In addition to the ease of communication between the doctor and the patient, which saves time.

**Chapter Three**

A picture containing graphical user interface

Description automatically generated

**Requirement Engineering and Analysis**

**3.1 Stakeholders**

A stakeholder is anybody who can affect or is affected by an organization, strategy, or project. They can be inner or outer and they can be at higher or lower levels.

**3.1.1 System users**

* Patient.
* Doctor.
* administrator.

**3.1.2 Development team**

* System Analyst.
* Programmer.
* Project manager.
* Software Tester.
* Software architecture.

**3.2 Use Case Diagram**

**3.2.1 Use Case Section**

**Diagram

Description automatically generated**

**Figure 3-1: HealthCare Use Case Diagram**

**3.2.2 Use Case Description**

* Use case #1:

Table 3-1: Description of use case Register Patient

|  |  |
| --- | --- |
| Use case name | Register Patient |
| ID | 1 |
| Brief Description | The system registers a new patient |
| Precondition One | None |
| Basic Flow | 1. The patient enters the site’s URL. 2. The use case starts when the user selects register on the main system menu. 3. The patient enters the following information:   3.1 Create own username and password.  3.2 first name, last name, password, phone number, personal health number, address, Date of birth, Gender, Address.   * 1. In the text box, the patient places his notes if he has any chronic diseases, blood type, allergy.  1. The user fills out the registration form and submits it. 2. The system verifies the submitted information. If the data is valid, the system returns the confirming information to the user, shows the user a welcome message, the user account, and the password and ID (personal health number) |
| Alternative Flow | 1. If the user’s email address has already existed in the system, the system shows an error message. Then the system displays the login form or add new email. 2. If the user’s password and re-entry password are not identical, the system shows an error message. Then the system displays the registration form and re-try. |
| Post-condition | After a new patient register to the system successfully, the patient can use the user account or ID and password to login the system and perform other tasks. |
| Special Requirement | If the patient registers from the same phone, more than two accounts, the "I'm not Report" window appears |

* Use case #2:

Table 3-2: Description of use case login

|  |  |
| --- | --- |
| Use case name | Login |
| ID | 2 |
| Brief Description | After register on system patient can login on their account |
| Precondition One | Patient should make account in system |
| Basic Flow | 1. The patient enters id and password. 2. Press login. |
| Alternative Flow | 1. If patient enter wrong id the system displays the message “wrong id, please enter your id”. 2. If the user enters the same wrong password, the system displays the message "this password incorrect”. |
| Post-condition | Login successfully |
| Special Requirement | If the patient enters error current password more than 3 times, he delivers a message on the entered phone at the time of registration. "Someone tried to change your password." |

* Use case #3:

Table 3-3: Description of use case schedule appointment

|  |  |
| --- | --- |
| Use case name | Scheduled appointment |
| ID | 3 |
| Brief Description | The system allow patient to view own appointment after login |
| Precondition One | The user must log in the system successfully. |
| Basic Flow | 1.successful login  2.the patient choose to view own schedule appointment |
| Alternative Flow | If patient did not find any appointment, they can call clink |
| Post-condition | Appointment will display on page |
| Special Requirement | - |

* Use case #4:

Table 3-4: Description of use view medical recorded

|  |  |
| --- | --- |
| Use case name | View medical record |
| ID | 4 |
| Brief Description | Allows to show patient’s history |
| Precondition One | The user must log in the system successfully. |
| Basic Flow | Select view history |
| Alternative Flow | Attempt to editing it by patient, the system displays the message "You do not have permission to editing the information" |
| Post-condition | The history displays on page |
| Special Requirement | - |

* Use case #5:

Table 3-5: Description of use case Edit Personal Information

|  |  |
| --- | --- |
| Use case name | Edit personal information |
| ID | 5 |
| Brief Description | allows a registered patient to update his or her personal information, such as an address, phone number etc. |
| Precondition One | The user must log in the system successfully. |
| Basic Flow | 1. The user accesses the system and chooses “view and edit my profile” view and edit the user’s personal information. 2. The system gets the information of the login user and displays the information to the user. 3. The user chooses the “edit my profile”. 4. The user edits the data on the form and submits it. 5. The system checks the validity of the user’s input. If the data is valid, the system returns the confirming information to the user |
| Alternative Flow | 1. If the user’s password and re-entry password are not identical, the system shows an error message. Then the system displays the registration form, and the user goes to step 3.  2.If the user enters the same current password, the system displays the message "this password is the same old password", and the user goes to step 3 |
| Post-condition | Update personal information for patient |
| Special Requirement | If the patient enters error current password more than 3 times, he delivers a message on the entered phone at the time of registration. "Someone tried to change your password." |

* Use case 6#:

Table 3-6: Description of use case search for appointment

|  |  |
| --- | --- |
| Use case name | Search for appointment |
| ID | 6 |
| Brief Description | The patient can see the appointment schedule based on |
| Precondition One | The patient should select the Department of Clinics |
| Basic Flow | patient should search for appropriate appointment and appropriate doctor |
| Alternative Flow | None |
| Post-condition | None |

* Use case 7#:

Table 3-7: Description of use case login staff

|  |  |
| --- | --- |
| Use case name | Login staff |
| ID | 7 |
| Brief Description | Staff login system |
| Precondition One | Medical staff / admin will enter username, password |
| Basic Flow | 1.Staff will enter website.  1.1-enter username, password.  1.2-press login |
| Alternative Flow | None |
| Post-condition | None |

* Use case #8/9:

Table 3-8/9: Description of use case Edit medical file for patient.

|  |  |
| --- | --- |
| Use case name | add/Edit medical file for patient |
| ID | 8/9 |
| Brief Description | Doctor can add or edit medical file for patient depend on patient visit |
| Precondition One | Doctors send request to patient / patient accept |
| Basic Flow | 1-doctor login to the website  2-doctor sent request for patient.  3-patient accept or reject the doctor request.  4-doctor add or edit medical file for patient |
| Alternative Flow | Patient may reject doctor access history, on result doctor cannot access to history |
| Post-condition |  |

* Use case #10:

Table 3-10: Description of use case Patient accept.

|  |  |
| --- | --- |
| Use case name | Patient accept |
| ID | 10 |
| Brief Description | Medical staff or admin sent request for patient and then patient accept or reject the request |
| Precondition One | Medical staff or admin sent request for patient |
| Basic Flow |  |
| Alternative Flow | None |
| Post-condition | Medical staff add/edit medical file for patient |

* Use case #11:

Table 3-11: Description of use case Send reminder for patient.

|  |  |
| --- | --- |
| Use case name | Send reminder to patient |
| ID | 11 |
| Brief Description | to post a reminder to the user about a specific event in their schedule |
| Precondition One | The patient has appointments on pending |
| Basic Flow | The system send notification to patient for reminder using email. |
| Alternative Flow | None |
| Post-condition | A new alarm is added to the system at the specified date/time(s) |

* Use case #12:

Table 3-12: Description of use case View all appointment

|  |  |
| --- | --- |
| Use case name | View all appointment |
| ID | 12 |
| Brief Description | allows the doctors and admin to view all appointments. |
| Precondition One | The doctor must be authenticated |
| Basic Flow | After doctors have logged into the system and have chosen “View ALL Appointments”, doctors will be able to see all the appointments that are pending. |
| Alternative Flow | None |
| Post-condition | System will generate a report. |
|  |  |

**3.2.3 Activity Diagram:**

**Diagram

Description automatically generated**

**Figure 3-2: Activity diagram of Use Case Register**

**Diagram

Description automatically generated**

**Figure 3-3: Activity diagram of Use Case personal information**

**3.3 functional requirements**

System requirements

SR1: The system should be running in php system.

SR2: The system should view a description of each healthcare item.

SR3: The system should request password and username from user.

SR4: The system should view different categories of item.

SR5: The system can view available appointment for patient.

SR6: The system shall enable user to book an appointment.

SR7: The system should enable patient to view his medical file.

SR8: The system should enable user to manage his profile.

patient requirements

PR1- The patient should be able to sign up for system by entering (username, password, email, address).

PR2- The patient should be able to sign in for system by entering (username, password).

PR3 - The patient should be able to schedule their appointment.

PR4- The patient should be able to view their medical history.

PR5- The patient should be able to edit their personal information.

PR6- The patient should be able to search for available appointment.

PR7- The patient should be able to accept the doctor request to edit their medical history.

Medical staff requirements

MSR1- The medical staff should be able to sign in for system by entering (username, password).

MSR2- The medical staff should be edit, add to medical record for patient.

MSR3 - The medical staff should be able to view their personal information.

Admin requirements

AR1- The admin should be able to sign in for system by entering (username, password).

AR2- The admin should be able to view their personal information.

AR3 - The admin should be able to view all appointment.

AR4 - The admin should be able to send reminder for patient.

**3.4 Non-functional requirements**

* **Execution qualities**

1.Usability

The system shall be easy to use.

No technical skills shall be required to use the system.

The system shall provide easy navigation for the user.

2.praivicy

Medical history of each patient should be visible by authorized people.

3.Efficiency

the system should response in 2 second or less.

4.Availability

system is required to have 99.99% available

**5.Security**

The Email and password shall be required to the active screen,

only authorized people can access to their accounts.

**6.Responsiveness**

For every invalid input from the user the system shall display meaningful error message explain input format expected.

* **Evolution qualities**

7.Maintainability

The system components shall be able to change without break the whole system.

8.Testability

The user interface shall not contain any component that would prevent automated testing.

**3.5 Constraints**

1-Time constraint is one of most important factors to measure project success (how long it will take to deliver).

2-Scope that refer what include in project and excluded.

3-Privacy its important factor to protect user information from external resource.

**Chapter four**

A picture containing graphical user interface

Description automatically generated

**Architecture and Design**

**4.1 Overview**

The software architecture of a system describes the system’s plan or structure and describes how it works. A system represents the collection of components that achieve a specific function or set of functions.

Software architecture is very important in any software system to help well.

understanding of system, we have many views such as:

**Logical view:**

The logical view is used to decompose systems into logical components that

represent the structural integrity that supports functional and nonfunctional

requirements.

**Process view:**

Shows how, at run-time, the system is composed of interacting processes.

Represent the dynamic or behavioral aspects of software systems, where the main units of analysis are processes and threads.

**Physical view:**

shows the system hardware and how software components are

distributed across the processors in the system.

**4.2 Software architecture**

**4.2.1 Logical view:**

the figure below shows the app user interface contains a category and the controller contains doctor and admin, App user interface asks access data from the controller and provides details to the patient then database store data and provide details to the controller.

**Diagram

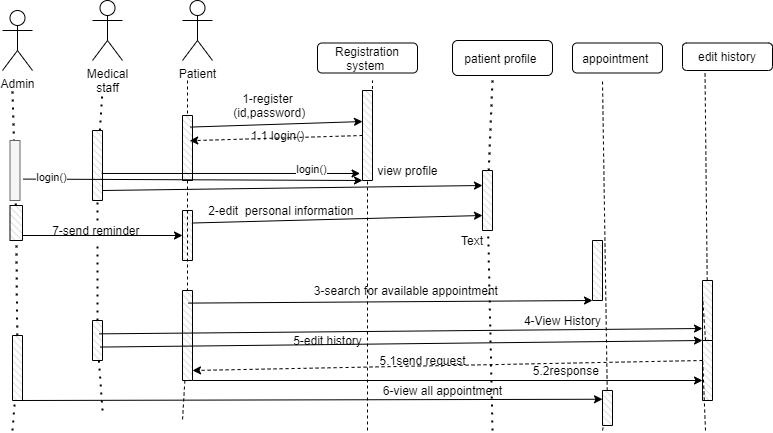
Description automatically generated**

**Figure 4-1: logical view**

**4.2.2 Process view:**

As showed in **Figure 4-2** we describe the interaction between objects how

User will be register in the system, who has access to the patient profile, how make an appointment, and who can edit history.

****

**Figure 4-2: sequence diagram**

**Diagram

Description automatically generated**

**Figure 4-3: communication diagram of edit personal information**

**Diagram

Description automatically generated**

**Figure 4-4: communication diagram of edit register patient**

**4.2.3 Physical view:**

the below figure shows the server that contains the database that contains a controller, the data will be stored in the database which receives TCP/IP request and we have a web browser (healthcare tracking system) that receives the HTTP request.

**Graphical user interface, diagram

Description automatically generated**

**Figure 4-5: physical view diagram of edit register patient**

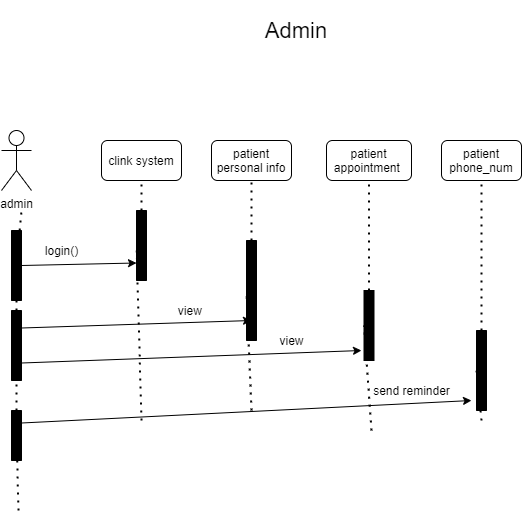
**4.3 Software design**

Software design is the process of imagining the software requirements into software implementation. This is the initial phase within the software development life cycle SDLC, moving the attention from the problem to the solution.

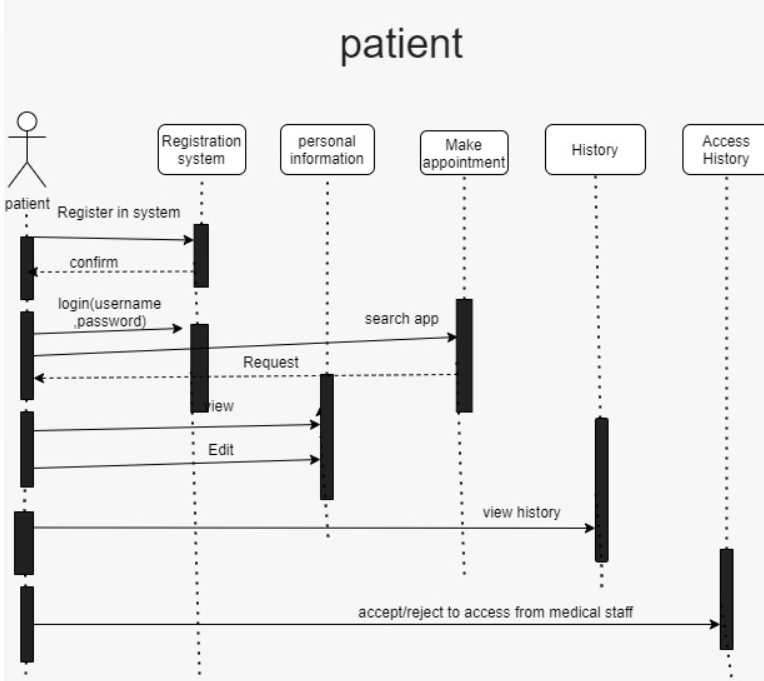
**4.3.1 UML sequence/communication diagram**

****

**Figure 4-6: Healthcare tracking system Sequence Diagram for medical staff**

****

**Figure 4-7: Healthcare tracking system Sequence Diagram for admin**

****

**Figure 4-8: Healthcare tracking system Sequence Diagram for patient**

**4.3.2 Class diagram**

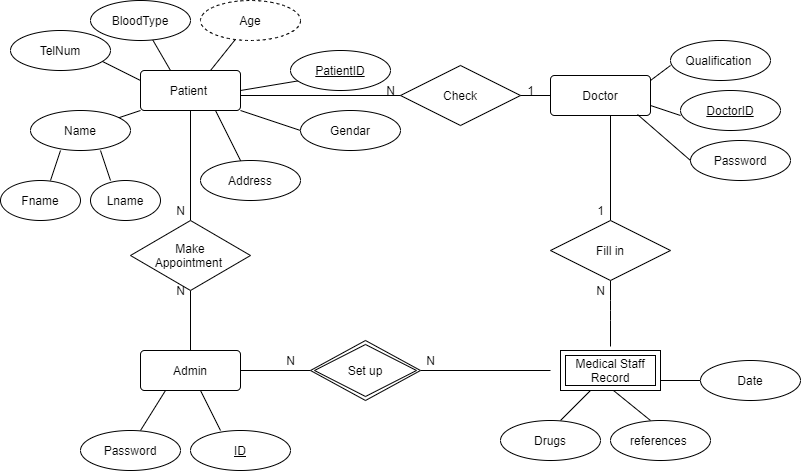
The class diagram is the main building block of object-oriented modelling. It is used for general conceptual modelling of the structure of the application, and for detailed modelling translating the models into programming code.

**Diagram

Description automatically generated**

**Figure 4-9: Healthcare tracking system Class diagram**

**4.3.3 ER diagram (if any)**

****

**Figure 4-10: Healthcare tracking system ER Diagram**

**4.3.4 State transition diagram**

**Diagram

Description automatically generatedFigure 4-11: State diagram for register patient**

**Diagram

Description automatically generated**

**Figure 4-12: State diagram for personal information**

**4.4 User interface design (prototype)**

User interface (UI) prototyping is an iterative analysis procedure in which users are actively included in the simulation of the UI for a system. UI prototypes have several purposes:

* As an analysis artifact that enables you to explore the problem space with your stakeholders.
* As a requirements artifact to initially version the system.
* As a design artifact that enables you to explore the solution space of your system.
* A vehicle for you to communicate the possible UI design(s) of your system.
* A potential foundation from which to continue developing the system (if you intend to throw the prototype away and start over from scratch then you do not need to invest the time writing quality code for your prototype).

****

**Figure 4-13: Sani medico interface**

****

**Figure 4-14: login form for medical staff**

****

**Figure 4-15: login for admin**

****

**Figure 4-16: login form for patient**

****

**Figure 4-17: personal information for patient**

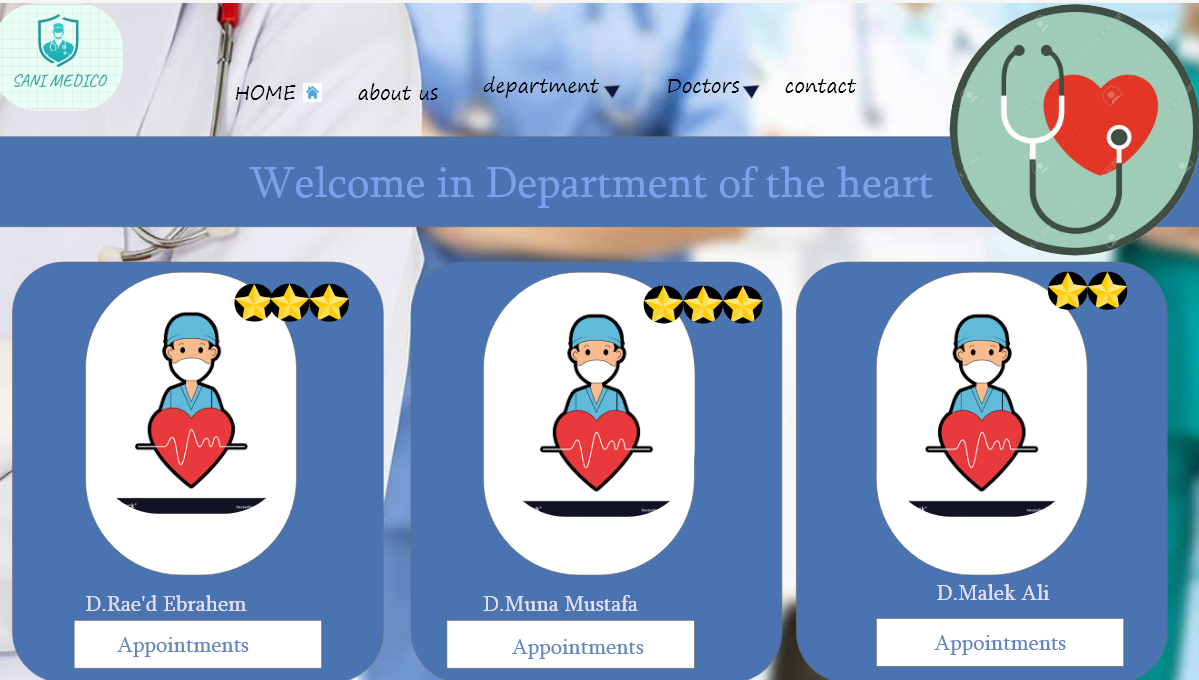
****

**Figure 4-18: medical information for patient**

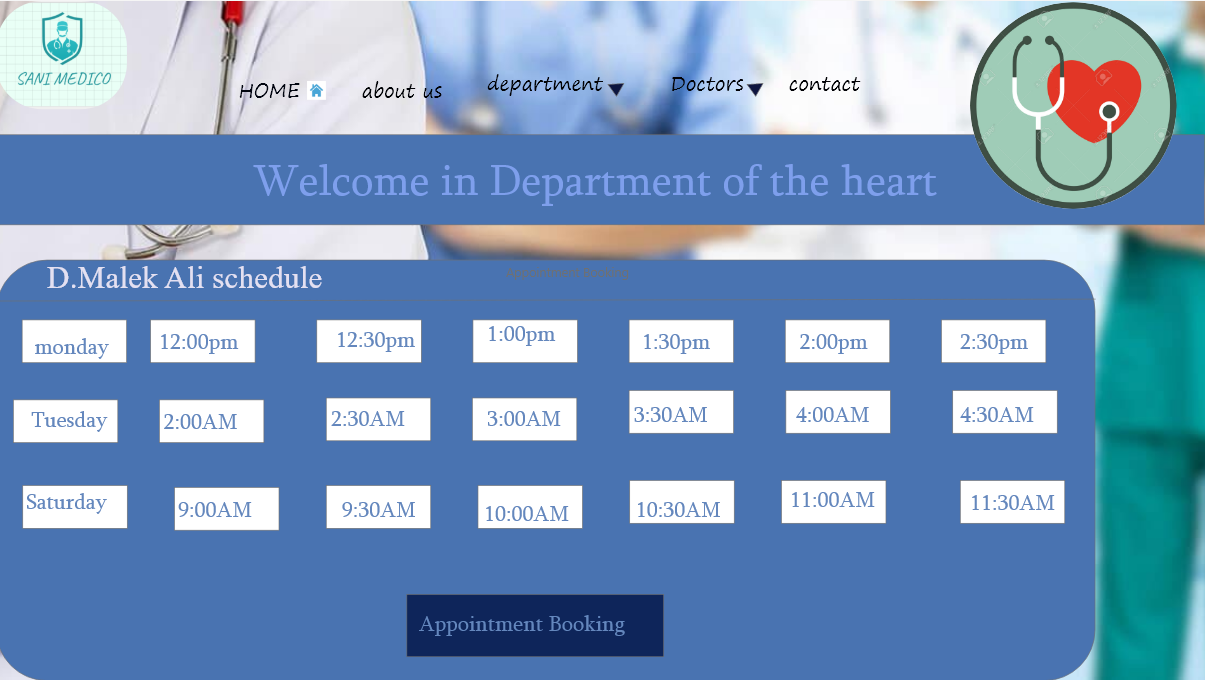
****

**Figure 4-19: department of treatment**

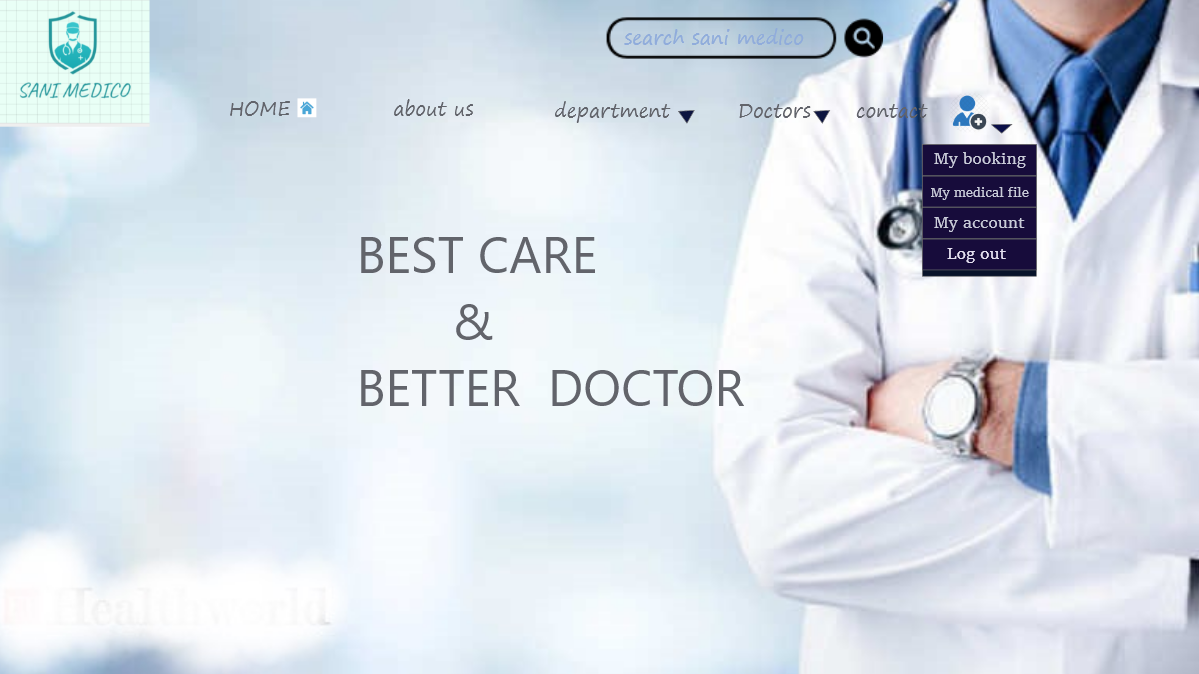
**.**

****

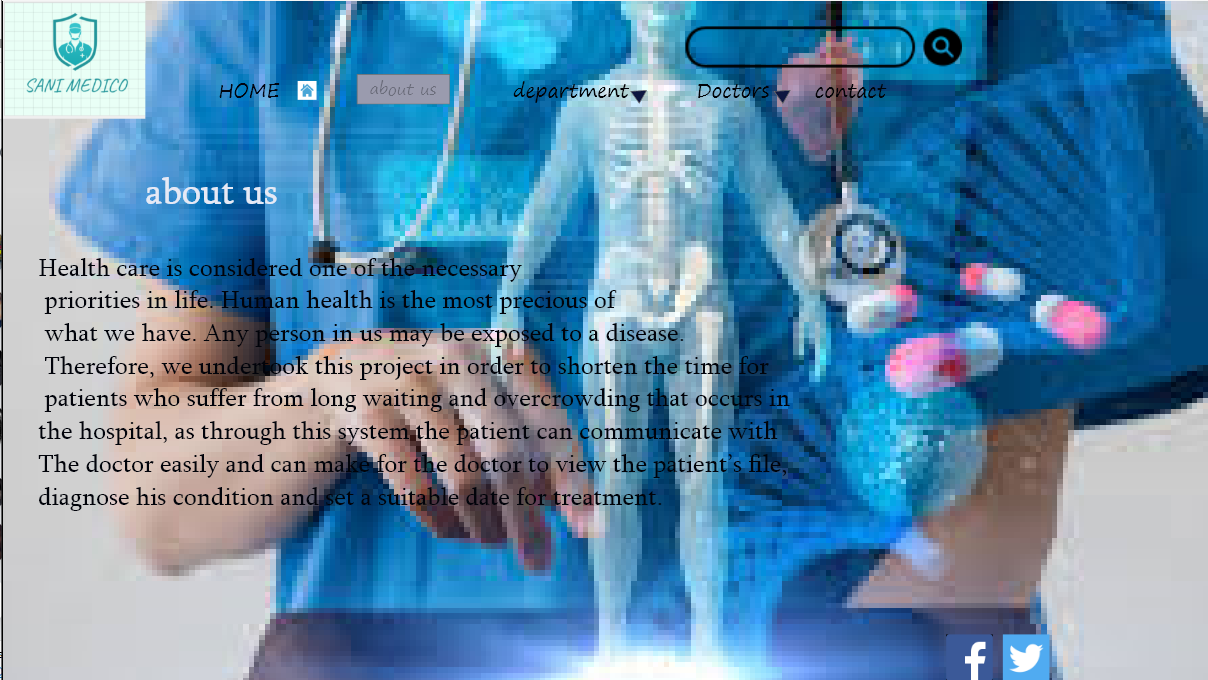
**Figure 4-20: name of doctor**

****

**Figure 4-21: doctor schedule**

****

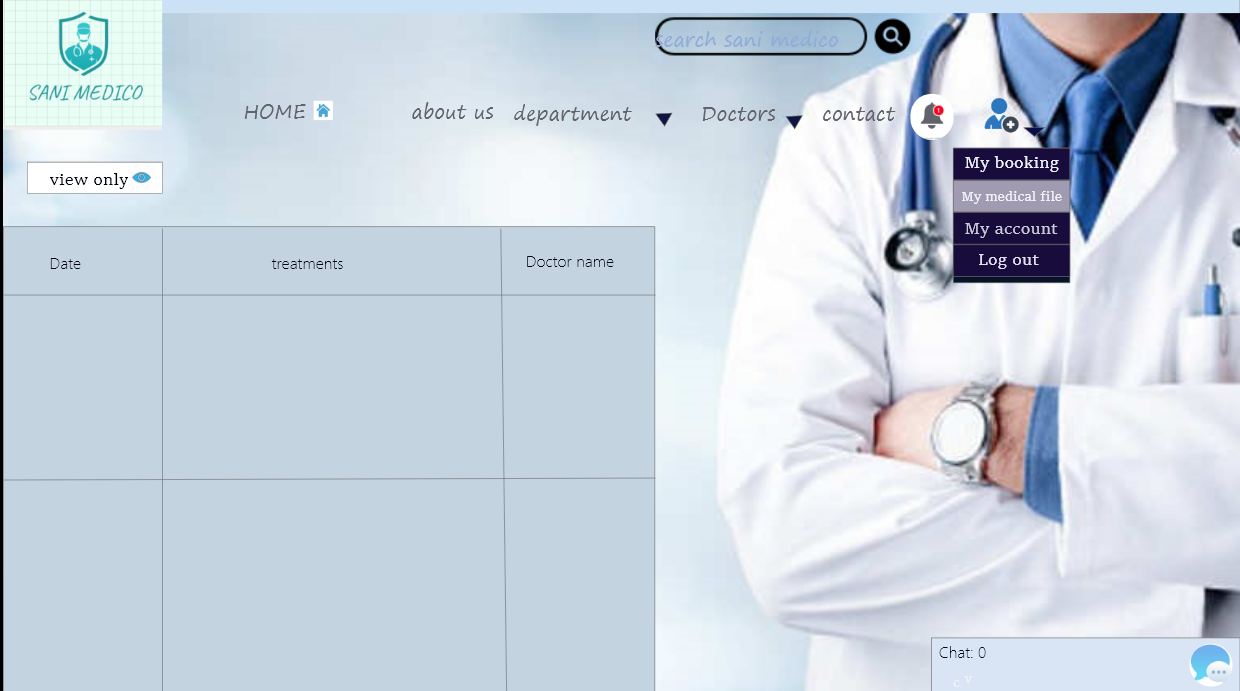
**Figure 4-22: patient manage account.**

****

**Figure 4-23: about Sani Medco**

****

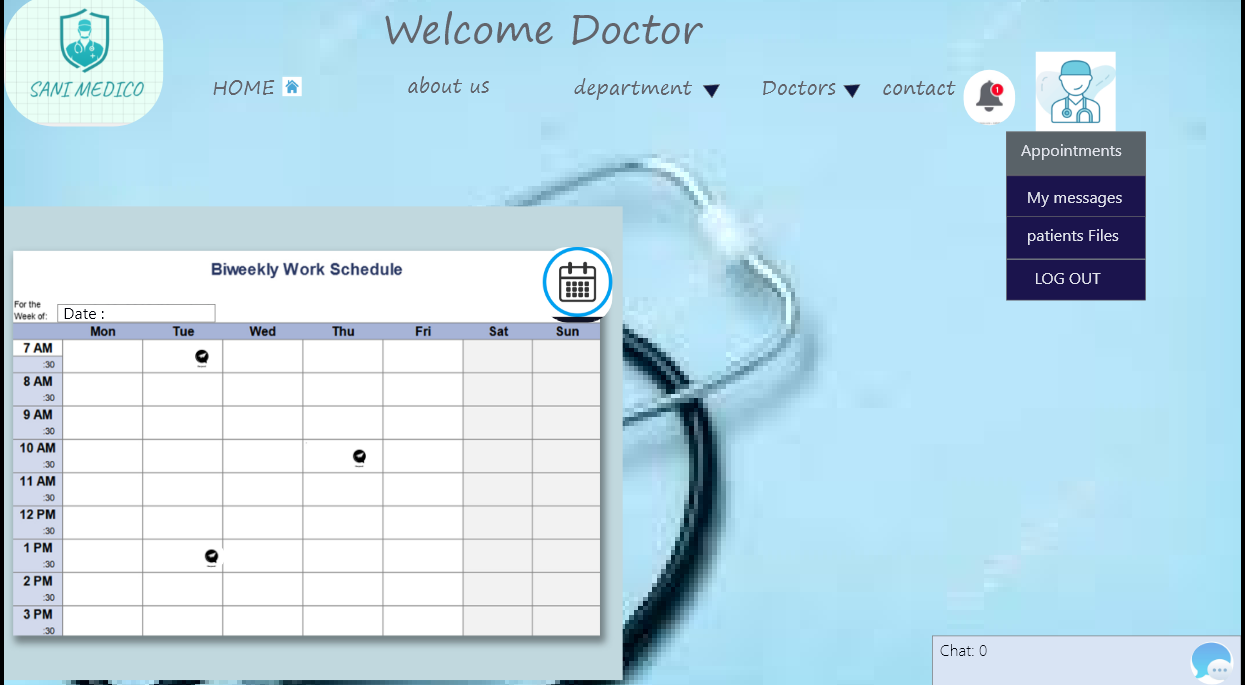
**Figure 4-24: patient booking**

****

**Figure 4-25: medical file for patient**

****

**Figure 4-26: patient account (communication with doctor)**

****

**Figure 4-27: appointment schedule for doctor**

**Chapter Five**

A picture containing graphical user interface

Description automatically generated

**Implementation plan**

* 1. **Description of implementation**

Our plan to implementation project first download XAMMP to get local server apache, Database MYSQL and we can rum PHP file

1-Download last version of XAMPP for windows operating system 8.0.0(PHP 8.0.0) after we complete downloaded, we start server apache and MySQL. Apache (server connect code with database) they have processed port number we can changed it from config. MYSQL database is first open scours database available, MySQL is an RDBMS (Relational Data Base Management System) to store, retrieve, modify, and administrate a database using SQL.

\*My laptop specification

Windows 10

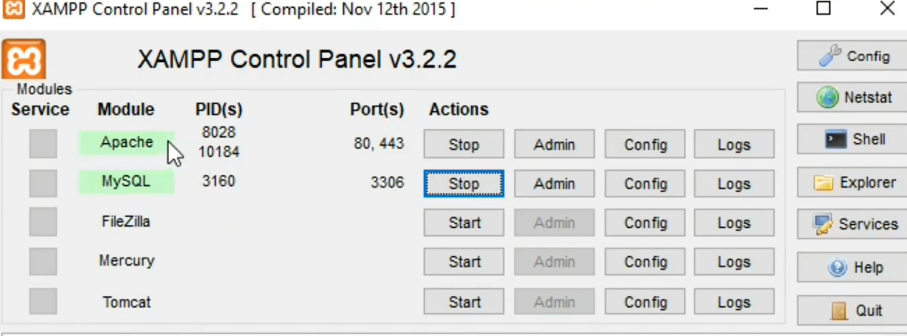
Processor CORE i7

Ram 8GB

64-bit operating system



**Figure 5.3.1-: Download Xammp**



**Figure 5.1.1-: Start Apache and start MYSQL database**

2-After that we are preparing Database:

A-create Database.

B-create table.

C-Capturing User Input

D-Adding Records to the Database

E-Retrieving Records from the Database

F-Deleting Records and Security

G-Updating Records

H-Database Connection (in web page $connection =new MySQLi (server name ‘host’, username, password, database name)).

3-we put our website in HTDOC, we create folder “sani-medco” we use Notepad++(PHP8, HTML5,CSS3), We design interface using HTML5 CSS3 to make the website dynamic we use PHP8 to connect with a database that allows us to update, delete, retrieve, edit ...etc. Open website <https://localhost/sani-medco.com/>.

So far, our website will be implemented in localhost, and we try to find free hosting to upload PHP site.

Our coding plan we design GUI using HTML5, PHP8 the first interface welcome in Sani Medco user have a lot of option, we design doctor login ,admin login and patient login , if patient have not account he/she make account and enter all information required after that patient can login her/his account, we design interface that enable patient search about doctor and make appointment ,design interface medical history, patient can just view and doctor require accessibility from patient , design interface to login admin that check appointment .

We will make database and table for patient, doctor, admin, appointment, and connect database we will use php to make dynamic website.

**Team responsibility**

haneen, silina will design interface

sewar, Sarah will make database

All member groups will be making implementation and testing.

**5.2 Programming language and technology**

To develop this website, we use PHP8, MySQL (database), HTML5, CSS3, and MYSQL. To use this website, you should have a smart device (mobile phone, laptop ...) internet connection and web browser.

**5.3Part of implementation if possible**

The code in "sani medico"/folder "code"every page in the text document.

and we create Database :

CREATE TABLE patient-information (

"PATIENT\_ID" NUMBER NOT NULL ENABLE,

PATINENT-ID NUMBER NOT NULL

PATIENT\_NAME VARCHAR2(40),

AGE NUMBER,

Gender VARCHAR2(12),

DOC-ID NUMBER,

PHONE\_NUMBER NUMBER,

Birth-day DATE,

Address VARCHAR2(30),

City VARCHAR2(20),

Emergency-contact VARCHAR2(40),

Emergency- num NUMBER,

Email VARCHAR (320)

Weight INT,

Height INT,

Medical-history VARCHAR2(600),

Blood-type varchar2(50),

CONSTRAINT PATIENT\_PK PRIMARY KEY "PATIENT\_ID"

CONSTRAINT DOCID\_FK FORIGN KEY "DOC-ID" REFERENCE “Doctor” (“DOC-ID”) Enable

);

CREATE TABLE Doctor (

DOC\_ID Number Not Null,

DOC\_Name varchar (15),

DOC\_SPECIALTY varchar (15)

CONSTRAINT PK-DOCTORID PRIMARY KEY” DOC\_ID”

);

CREATE TABLE Admin (

Admin-name VARCHAR (20),

ADMIN-ID NUMBER not Null,

CONSTRAINT Admin-PK PRIMARY KEY " ADMIN-ID ");

CREATE TABLE appointment (

DOCTOR\_ID int references DOCTOR(DOC\_ID)

PATIENT\_ID int references PATIENT(PATIENT\_ID)

VISIT\_DATE DATE,

VISIT\_ID int,

CONSTRAINT visit-pk PRIMARY KEY (VISIT\_ID )

);

CREATE TABLE Login (

DOCTOR\_ID int references DOCTOR(DOC\_ID),

PATIENT\_ID int references PATIENT(PATIENT\_ID),

ADMIN-ID ID int references PATIENT(ADMIN-ID),

Password VARCHAR2(20));

INSERT INTO patient-information (PATIENT-ID, PATIENT-NAME, AGE, Gender, DOC-ID, PHONE-NUMBER, BIRTH-DAY, Address, City, Emergency-contact, Emergency- num, Email, weight, height, medical-history, blood-type)

VALUES (125, 'AhmadAli', 36, 'Male', 4, 0797745500,'15/12/1984','almadenastreet','irbid','salehAli',0255687, ‘AhmadAli785@gmail.com',77,178,'he have diabetes type2 hypertension and lactose intolerance', 'A+');

INSERT INTO Doctor (DOC-ID, DOC-NAME, DOC- SPECIALTY)

VALUES (4, 'Mohammed', 'internist');

INSERT INTO Admin (Admin-name, ADMIN-ID)

VALUES('Mustafa',78);

INSERT INTO appointment (DOCTOR\_ID, PATIENT\_ID, VISIT\_DATE, VISIT\_ID)

VALUES (4,125, '20/2/2020',77);

**Chapter six**

A picture containing graphical user interface

Description automatically generated

**Test planning**

**6.Test planning**

Test Plan detailed document that describes the test strategy, objectives

And the way of how to write/create a detailed software test plan.

**6.1 Black box**

## Black Box testing is a software test method in which internal structure/design/implementation is not known to the tester, also known as Behavioral testing.

## Black Box Testing Techniques

Following are the prominent test strategy amongst the many used in Black box Testing

* **Equivalence Class Testing:** It is used to minimize the number of possible test cases to an optimum level while maintains reasonable test coverage.
* **Boundary Value Testing:** Boundary value testing is focused on the values at boundaries. This technique determines whether a certain range of values are acceptable by the system or not. It is very useful in reducing the number of test cases. It is most suitable for the systems where an input is within certain ranges.
* **Decision Table Testing**: A decision table puts causes and their effects in a matrix. There is a unique combination in each column.

A picture containing graphical user interface

Description automatically generated

**Figure 6-1: black box testing techniques**

## the following tables describe the valid and invalid boundaries and test cases:

## Register

**Table 6-1: Register test case**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Condition | Valid equivalence partitioning | Tag | Invalid  Equivalence  partitioning | Tag | Valid  Boundary | Tag | Invalid  Boundary | Tag |
| Username | 1-15 char | V1 | Less than 1  Greater than 15 | X1  X2 | 1 char  15 char | B1  B2 | 0 char  16 char | Y1  Y2 |
| Password | 4-15 char | V2 | Less than 4  Greater than 15 | X3  X4 | 4 char  15 char | B3  B4 | 3 char  16 char | Y3  Y4 |
| Email | 4-24 char | V3 | Less than 4  Greater than 24 | X5  X6 | 4 char  24 char | B5  B6 | 3 char  25 char | Y5  Y6 |
| Phone number | 10 | V4 | Less than 10  Greater than 10 | X7  X8 | 10 char | B7  B8 | 9 char  11 char | Y7  Y8 |

**Login:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Condition | Valid equivalence partitioning | Tag | Invalid  equivalence partitioning | Tag | Valid  Boundary | Tag | Invalid  Boundary | Tag |
| Username | 4-10 char | V4 | Less than 4  Greater than 10 | X7  X8 | 4 char  24 char | B7  B8 | 3 char  11 char | Y7  Y8 |
| Password | 4-15 char | V5 | Less than 4  Greater than 15 | X9  X10 | 4 char  15 char | B9  B10 | 3 char  16 char | Y9  Y10 |

**Table 6-2: Log in test case**

**t**he figure below describes the login test, you can try to enter the password three times if the password incorrect the system locked.

**Diagram

Description automatically generated**

**Figure 6-2:login test**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Steps | Variable | Test Case1 | Test Case2 | Expected result | Actual result | Pass/fail |
| 1 | **Login to System** | Enter valid username and password | ---- | entering to the system | ---- |  |
| 2 | **Login to System** | ---- | Enter invalid username and password | Displaying Error message and staying in  login page | ---- |  |

**Table 6-3: Log in test case**

**Schedule appointment test case**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Steps | Variable | Test Case1 | Test Case2 | Expected result | Actual result | Pass/fail |
| 1 | **Schedule appointment** | Click on Schedule appointment | ---- | **Schedule appointment** | ---- |  |
| 2 | **Schedule appointment** | ---- | does not schedule appointment | ----- | ---- |  |

**Table 6-4: Schedule appointment test case**

**View medical history**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Steps | Variable | Test Case1 | Test Case2 | Expected result | Actual result | Pass/fail |
| 1 | **View medical history** | Click on view medical history | ---- | Show medical history | ---- |  |
| 2 | **View medical history** | ---- | does not Click on view medical history |  | ---- |  |

**Table 6-4: View medical history**

**Edit personal info.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Steps | Variable | Test Case1 | Test Case2 | Expected result | Actual result | Pass/fail |
| 1 | **Edit personal info** | Click on edit personal info | ---- | Edit personal info | ---- |  |
| 2 | **Edit personal info** | ---- | does not Click on edit personal info |  | ---- |  |

**Table 6-5: edit personal information**

**login staff**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Steps | Variable | Test Case1 | Test Case2 | Expected result | Actual result | Pass/fail |
| 1 | **Login staff** | Enter valid username and password | ---- | Entering to the system | ---- |  |
| 2 | **Login staff** | ---- | Enter invalid username and password | Displaying error message and staying in login page | ---- |  |

**Table 6-6: login staff**

**edit medical record for patient**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Steps | Variable | Test Case1 | Test Case2 | Expected result | Actual result | Pass/fail |
| 1 | **Edit Medical record for patient** | Patients accept the doctor request | ---- | Doctors edit medical file for patient | ---- |  |
| 2 | **Edit Medical record for patient** | ---- | Patients reject the doctor request | Doctors cannot access medical file for patient | ---- |  |

**Table 6-7: edit medical record for patient**

**add to medical file for patient**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Steps | Variable | Test Case1 | Test Case2 | Expected result | Actual result | Pass/fail |
| 1 | **Add to Medical file for patient** | Patients accept the doctor request | ---- | Doctors add medical file for patient | ---- |  |
| 2 | **Add to Medical file for patient** | ---- | Patients reject the doctor request | Doctors cannot access medical file for patient | ---- |  |

**Table 6-8: add to medical file for patient**

**6.2 white box testing**

White Box Testing: Is software testing technique, tester know about internal testing (structure, coding, implementation), white box testing have many advantage such as Ability to achieve complete code coverage, easy to automate.

Also known as clear box testing Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing.

**Diagram

Description automatically generated**

**Figure 6-3-: white box testing**

1-Unit tests: for testing path within a unit, is a level of software testing where individual unit or component of software are tested.

**2-**Integration testing: for testing paths between unit, is a level of software testing where individual units / components are combined and tested as a group.

3-System testing: for testing paths between a subsystem, is a level of testing that a complete and integrated software is tested.

**6.3 Test Automation**

The automation tools that have been used to control the execution of tests and the comparison of actual outcomes with predicted outcomes:

Selenium IDE.

**. Decide what test case automate**

Getting started with Selenium IDE requires no additional setup other than installing the extension on your browser. One of our driving philosophies is to provide an easy-to-use tool that will give instant feedback. We believe that the easier we can make it, the more likely people are to author tests, which in turn results in better tested apps.

**Selenium IDE- Login Test**

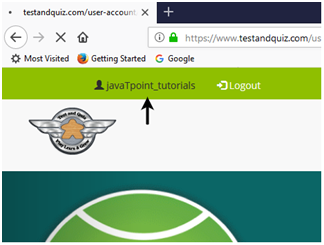
create a Login Test case in Selenium IDE:

* Fill-in the login credentials and click on the login button. Meanwhile you will get the notifications of the actions performed by IDE at the extreme right corner of your browser.

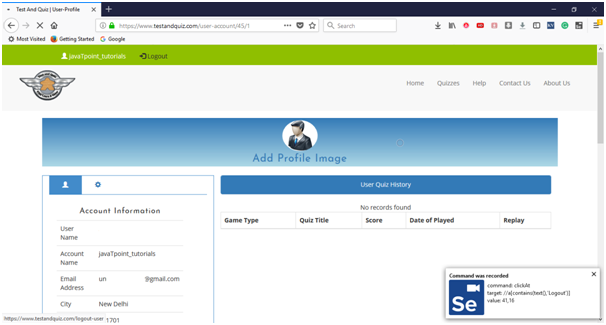
**Graphical user interface, application

Description automatically generated**

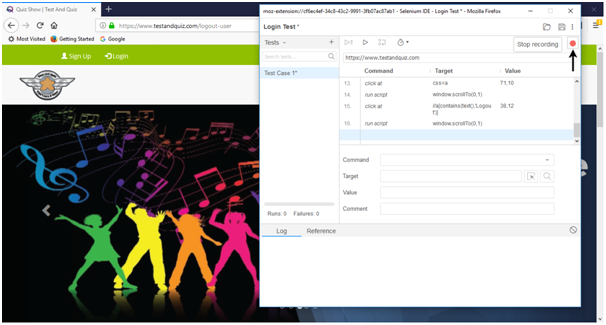
* Once you get logged-in, click on the username section to view your account details.



* It will redirect you to your account settings page, where you can edit your personal details.

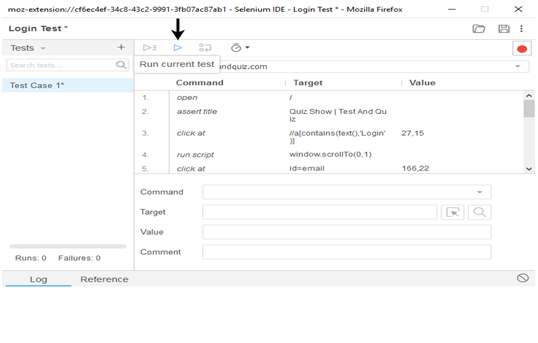


* Click on the Logout button.
* Now, go to the IDE and click on Stop Recording button to stop recording the test case.

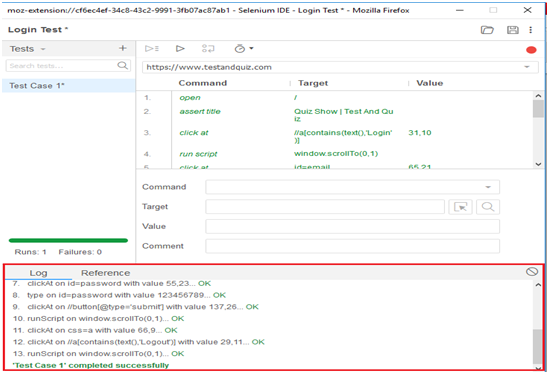


## 2. Playing Back

* Click on the "Run Current Test" button present on the tool bar menu of the IDE. It will execute all of your interactions with the browser and gives you an overall summary of the executed test script.

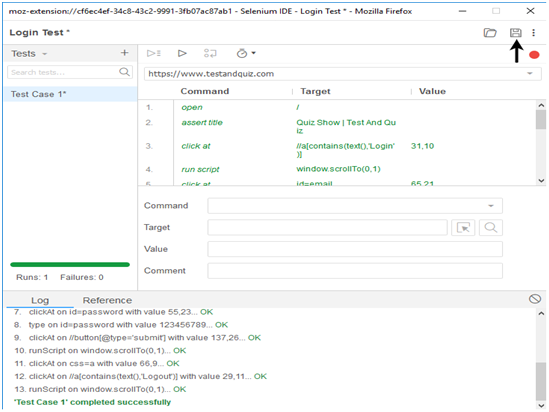


* The Log pane displays the overall summary of the executed test scripts.

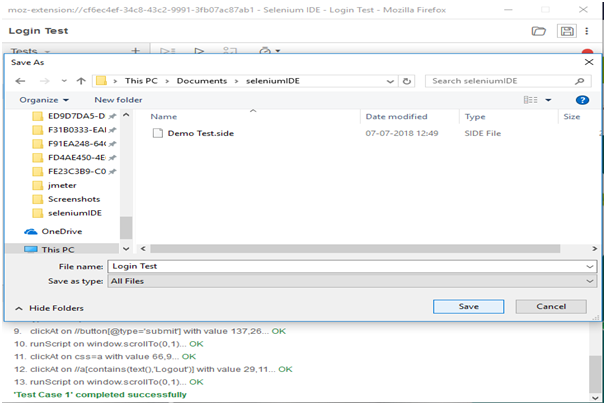


## 3. Saving the test suite

* Click on the save button present on the extreme right corner of the menu bar.



* Save the entire test suite as "Login Test".

  
-The test suite can be found at the location provided in the above steps

**Chapter seven**

A picture containing graphical user interface

Description automatically generated

**Conclusion and Results**

**7. Conclusion and Results**

The main goal of our project is to design the Health Care System that helps the patient find high quality of healthy life, by make it easier to access his/her medical records and check appointment.

By the notes we received from the health care providers about the problems they face with most of the patients that they have not the ability to explain their health conditions for the physicians. Also, the patients have problems in checking the appointments.

In order to exceed these problems and others we thought that this website will help all parties to save their time and effort.

**Chapter Eight**

A picture containing graphical user interface

Description automatically generated

**Reference**

**8.reference:**

**The references below show [1-5] books:**

**[1] Software Testing.**

**[2] Requirement Engineering Software.**

**[3] UML: Unified Modelling language.**

**[4] Software Engineering Design 1st Edition, Otero,Carlos E.**

**[5] Interaction design beyond human computer interaction,**

**Publisher John Wily & Sons, 3rd Edition, CS318.**

**[6] HTML/CSS /**[**https://www.w3schools.com/**](https://www.w3schools.com/)**.**

**[7] UML diagram/**[**https://www.lucidchart.com/blog/types-of-UML-diagrams**](https://www.lucidchart.com/blog/types-of-UML-diagrams)**.**

**[8] relationship between diagram /**[**https://www.smartdraw.com/entity-relationship-diagram/**](https://www.smartdraw.com/entity-relationship-diagram/)

**[9] sql /**[**https://www.guru99.com/sql-vs-mysql.html#:~:text=MySQL%20uses%20the%20SQL%20language%20to%20query%20the%20database**](https://www.guru99.com/sql-vs-mysql.html#:~:text=MySQL%20uses%20the%20SQL%20language%20to%20query%20the%20database)**.**

**[10] white box testing /** <https://softwaretestingfundamentals.com/white-box-testing>

**[11] Black box testing /** [**https://softwaretestingfundamentals.com/black-box-testing/#:~:text=BLACK%20BOX%20TESTING%2C%20also%20known,%2Dfunctional%2C%20though%20usually%20functional**](https://softwaretestingfundamentals.com/black-box-testing/#:~:text=BLACK%20BOX%20TESTING%2C%20also%20known,%2Dfunctional%2C%20though%20usually%20functional)**.**

**[12] unit testing/** <https://softwaretestingfundamentals.com/unit-testing/>

**[13] integration testing /** [**https://softwaretestingfundamentals.com/integration-testing/**](https://softwaretestingfundamentals.com/integration-testing/)

**[14] system testing /** <https://softwaretestingfundamentals.com/system-testing/>

**[15] Dr. Hamza Alkofahi Just- IT Department -Software Engineering**

**[16] pharmacist. Dina Alfdawi KHCC-former trainee**