

American International University-Bangladesh (AIUB)  
**Department of Computer Science  
Faculty of Science & Technology (FST)**

**Centralized Medical Application**

A Software Requirement Engineering Project Submitted

By

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Semester: Spring\_22\_23** | | **Section:** | **Group Number:** | |
| SN | Student Name | Student ID | Contribution (CO1+CO2) | Individual Marks |
| 19 | Abu Shaleh Md. Kaium | 20-42475-1 |  |  |
| 25 | Thouhida Tasnim | 20-43051-1 |  |  |
| 28 | Shariar Nihal | 20-43220-1 |  |  |
| 29 | Atunu Paul | 20-43227-1 |  |  |

The project will be Evaluated for the following Course Outcomes

|  |  |  |
| --- | --- | --- |
| Evaluation Criteria | Total Marks (50) | |
|  | |
| Revision History, Test Plan Identifier, Reference Materials, Problem Background, Solutions | [10 Marks] |  |
| Requirements Specification (System feature, Quality Attributes, System Interface, Project Requirements) | [10 Marks] |  |
| Item Not to be tested, Testing approach (Testing levels, tools, meetings), Test cases | [10 Marks] |  |
| Item pass/fail criteria, Test deliverables, Staffing and Training, Responsibilities, Scheduling, Risk | [10 Marks] |  |
| Approval, Format, Submission, and Defense | [10 Marks] |  |

Software Test Plan

for

Centralized Medical Application

Version 2.0 approved

Prepared by

Abu Shaleh Md. Kaium

Thouhida Tasnim

Atunu Paul

Shariar Nihal

Organization: Rise up Tech Ltd.

Date

Table of Contents

[Revision History 3](#_Toc126659480)

[1. TEST PLAN IDENTIFIER: AT-TP01.3 4](#_Toc126659481)

[2. REFERENCES 4](#_Toc126659482)

[3. INTRODUCTION 4](#_Toc126659483)

[3.1 Background to the Problem 4](#_Toc126659484)

[3.2 Solution to the Problem 4](#_Toc126659485)

[4. REQUEIREMNT SPECIFICATION 4](#_Toc126659486)

[4.1 System Features 4](#_Toc126659487)

[4.2 System Quality Attributes 5](#_Toc126659488)

[4.3 System Interface 5](#_Toc126659489)

[4.4 Project Requirements 5](#_Toc126659490)

[5. FEATURES NOT TO BE TESTED 5](#_Toc126659491)

[6. TESTING APPROACH 5](#_Toc126659492)

[6.1 Testing Levels 5](#_Toc126659493)

[6.2 Test Tools 6](#_Toc126659494)

[6.3 Meetings 6](#_Toc126659495)

[7. TEST CASES/TEST ITEMS 7](#_Toc126659496)

[8. ITEM PASS/FAIL CRITERIA 7](#_Toc126659497)

[9. TEST DELIVERABLES 7](#_Toc126659498)

[10. STAFFING AND TRAINING NEEDS 8](#_Toc126659499)

[11. RESPONSIBILITIES 8](#_Toc126659500)

[12. TESTING SCHEDULE 8](#_Toc126659501)

[13. PLANNING RISKS AND CONTINGENCIES 9](#_Toc126659502)

[14. APROVALS 9](#_Toc126659503)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Updated by | Update Comments |
| 0.1 | 2023.02.01 | Thouhida Tasnim | First Draft [Update solution of problem] |
| 0.2 | 2023.02.30 | Abu Shaleh Md. Kaium | Second Draft [Update system features] |
| 0.3 | 2023.03.15 | Atunu Paul | Third Draft [Update UI/UX design] |
| 0.4 | 2023.03.27 | Shariar Nihal | Fourth Draft [Update Testing Level] |
| 0.5 | 2023.04.10 | Thouhida Tasnim | Fifth Draft [Update Test Case] |
| 0.6 | 2023.04.18 | Abu Shaleh Md. Kaium | Final Draft [Revised full document and finalized] |

# TEST PLAN IDENTIFIER: CMA-001T

# REFERENCES

1. System Design Document (SDD)
2. User Manual
3. Test Strategy Document
4. Defect Report

# INTRODUCTION

## Background to the Problem

## Every citizen must go to the doctor lots of times throughout their lifetime. Also, scheduling an appointment to see the doctor and have the tests done takes a lot of time and presents difficulties for a variety of reasons. The citizen must manually make an appointment and then wait a long time for his meeting time. When a patient visits a doctor and the doctor prescribes him medicines or a test, the information is kept on a hard paper that is easily lost or damaged.And the result of this is that we have to go back to the doctor or have the subsequent tests done again, which is costly and time-consuming. These difficulties can be removed simultaneously by a digital web-based solution, which will save time and money.

## Solution to the Problem

People frequently struggle with the decision of whether or not to go to the hospital when they consider the inconvenience of seeing a doctor. Today, getting a serial in a government hospital is significantly more challenging. More than an hour must be spent waiting in line. Then, after receiving the serial, some people engage in political maneuvering to meet the doctor early. Again, taking tests and other tasks are challenging. Also, the sufferer will endure great suffering if they lose their previous paperwork. Sometimes, doctors and hospital administrators overcharge patients for prescriptions, medications, procedures, etc. These problems can be solved by standardizing these rates. At that time, they go for a test and consult with the doctor.

These problems can be carefully resolved by a web-based application. It will help people do their tasks more quickly and reduce their worry of getting health care. At the same time, it will also decrease discrimination in regard to the cost of medical care, scheduling appointments with physicians, etc. The existing paper-based process takes a long time, causes a lot of issues, and causes citizens to be treated differently depending on their needs. The web-based solution will get rid of all the hassles and give people an effective means to get the care they need. Citizens will be able to use this method to diagnose mild illnesses without seeing a doctor. With the help of this system, discrimination between social classes will be eliminated, and everyone will receive equal treatment.

There are a few independent applications, including DIMS, DocTime, Arogga, SeekMed, and others. Each of them has unique qualities and was created to fulfill particular requirements. Also, they provide some unique features and offers, but these are insufficient to address these problems. We require a health application that will be created to address each of the issues mentioned.

# REQUEIREMNT SPECIFICATION

## System Features

**1.User Registration:**

**Functional Requirements:**

* 1. user needs to register in this system first by giving a valid email address or phone number

and password.

1.2 A user also needs to provide valid information to get the right suggestions from the system.

1.3 The user must verify their email or phone number for completing the registration.

1.4 The user can add previous medical history (if any).

**Priority Level:** High

**Precondition:** User have valid NID card and phone number.

**Cross reference:** N/A

**2. Doctor Appointment**

**Functional Requirements:**

2.1 The system will recommend a doctor to the user based on their specific diseases.

2.2Additionally, a user can search for doctors manually by name.

2.3 The user will then be able to view the doctor's name, degree, specialty, and

Qualifications and  appointment time.

2.4 The user may book an appointment with any doctor by clicking "Book" next to the

doctor name.

2.5 The user can also find information about hospitals and their doctors on this site.

**Priority Level:** High

**Precondition:** user must need to log in with valid id and password.

**Cross reference**: 1.1, 1.2, 1.3, 1.4,4.1,4.2,4.3,4.4

**3.Doctor’s View:**

**Functional Requirements:**

3.1 Can view patient list who are requested for appointment.

3.2 Can check patient’s old medical history and reports.

* 1. Can communicate with patient if necessary.

**Priority Level:** Medium

**Precondition:** user must need to log in with valid id and password.

**Cross reference**: 1.1, 1.2, 1.3, 1.4, 2.3, 2.4, 2.5

1. **Disease Prediction**

**Functional Requirements:**

**4.1** Disease prediction requires inputs from the user, such as blood pressure and pulse.

**4.2** The user must select the checkbox next to the disease symptoms.

**4.3** The user is required to fill out the 'Duration' and 'Disease State' columns for any

symptoms for which the checkbox has been selected.

**4.4** The user must then select "Submit" for disease prediction, after which doctors will be

suggested based on the disease prediction.

**Priority Level:** Medium

**Precondition:** The user must accurately enter all required information.

**Cross-reference:** 2.3, 2.4, 2.5

1. **Custom Exercise Guide to Patients**

**Functional Requirements:**

5.1 In this option, a doctor may advise the patient to engage in physical activity.

5.2 The doctor can provide the patient with the necessary documents or videos.

5.3 Patients can download or verify the available documents and videos.

5.4 Additionally, the user is able to communicate with the doctor in real-time via live

chatting, and the doctor is able to monitor the patient's progress.

**Priority Level:** Medium

**Precondition:** The user must log in with their valid email and password and must have

doctor’s consultation subscription.

**Cross-reference:** 1.4, 2.3, 2.4, 2.5,4.1,4.3

## System Quality Attributes

**QA 1- Availability:** The system must be 97% available during 8 p.m. to 8 a.m. hours and 99% available between the hours of 8 a.m. and 8 p.m. local time.

**Priority Level:** High

**Precondition:** The users must have enough internet connection.

**Cross-reference:** N/A

**QA 2- Testability:** Software must have the ability to recognize when a system is at danger of failing. There shouldn't be any cyclomatic complexity greater than 15.

**Priority Level:** High

**Precondition:** N/A

**Cross-reference:** QA1**,** QA-4, QA-5

**QA 3- Portability**: The web-based platform shall be used to run the system. Using a web browser on any device, the user can access the system.

**Priority Level:** Medium

**Precondition:** N/A

**Cross-reference:** QA-2, QA-4, QA-5

**QA 4 – Maintainability:** It shouldn't take more than two hours for a maintenance programmer to make changes to an existing form. Any system problem must be fixed effectively by the maintenance programmers in less than three hours of manual effort.

**Priority Level:** High

**Precondition:** The system should detect any errors.

**Cross-reference:** QA-1, QA-2, QA-5

**QA 5–Flexibility:**The operation of this system will be easy and straightforward. If anything has to be added or updated, a maintenance programmer may work on the software and generate a new version, including code modifications and testing, in less than 3-4 hours of labor.

**Priority Level:** High

**Precondition :** System should identify an error.

**Cross-reference:** QA-2, QA-3

**QA 6-Performance:** The system must be able to handle a large number of concurrent users without performance issues, with an average response time of two seconds and at least 1000 simultaneous users without crashing or slowing down.

**Priority Level:** High

**Precondition:** The system must be functional and stable, with all components integrated.

**Cross Reference:** N/A

**QA 7**-**Efficiency:** The system should be designed to use resources efficiently and quickly to ensure an optimal user experience. It should be able to handle a large number of simultaneous users without slowing down or crashing. Performance metrics should be measured and optimised regularly to ensure peak efficiency.

**Priority Level:** High

**Precondition:** All of the parts of the system have to work together smoothly and reliably.

**Cross-reference:** QA-1, QA-2, QA-3, QA-5.

**QA 8- Security:** The system should have a well-defined authorization system to control user access to various system resources and restrict access to sensitive information.

**Priority Level:** High

**Precondition:** The system must have an effective authentication mechanism and well-defined user classifications and access levels.

**Cross-reference:** N/A

**QA 9- Reliability:** The system must be reliable, meaning that it should perform its functions correctly and consistently. It must be able to handle errors, prevent data loss, and recover quickly from any failure. The system must have a mean time between failures of at least 10,000 hours, and the mean time to repair must not exceed four hours.

**Priority Level:** High

**Precondition:** The system must be designed with redundancy and fault tolerance mechanisms in place.

**Cross-reference:** QA1, QA2, QA4, QA5, QA6, QA7

**QA 10- Usability:** The system must have a simple and understandable user interface. It needs to be simple to use, provide helpful feedback when something goes wrong, and be accessible to people with different abilities. Standardized usability testing requires a system usability score of 80% or above.

**Priority:** High

**Precondition:** The system must be fully functional, with all features implemented and tested.

**Cross reference:** QA-1, QA-3, QA-5

**QA 11- Reusability:** The system should be built so that its parts can be simply reused and incorporated into other programmers. For maximum reusability, the system should be developed using common practices in the field, such as object-oriented design and the separation of concerns.

**Priority:** Medium

**Precondition:** The system must have a clear and well-documented architecture with standardized APIs and data formats.

**Cross reference:** QA-2, QA-4, QA-5

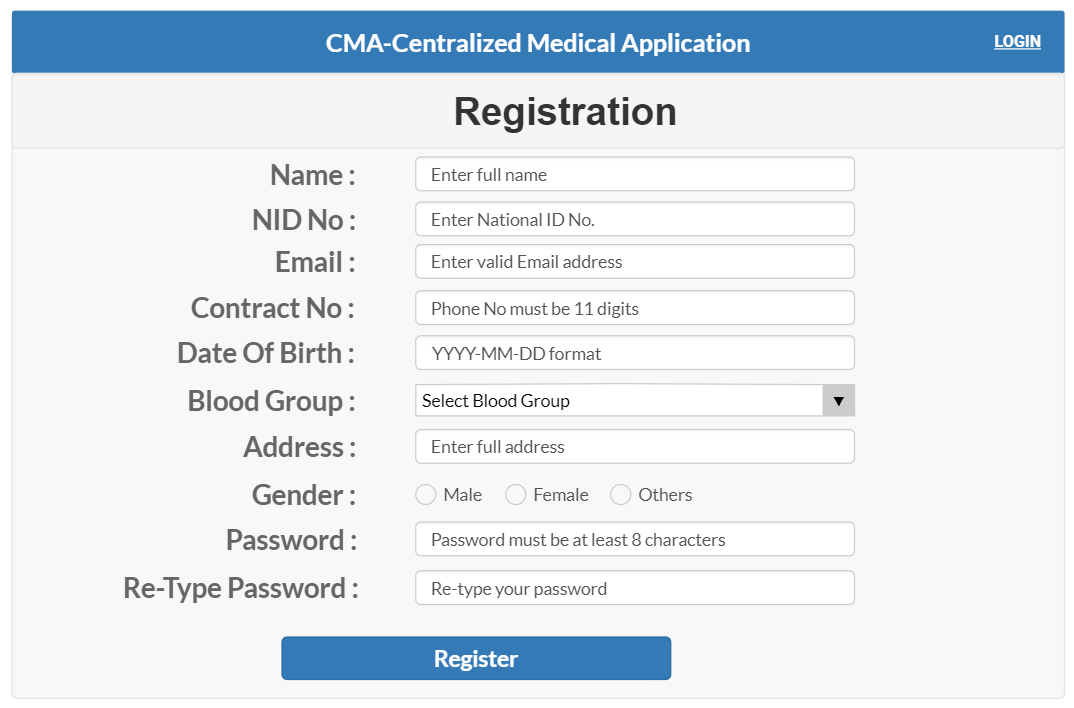
**QA 12- Interoperability:** The system must be compatible with common data formats and protocols so that it can share information with other applications. Data import/export functionality and compatibility with other widely used applications and platforms are essential features.

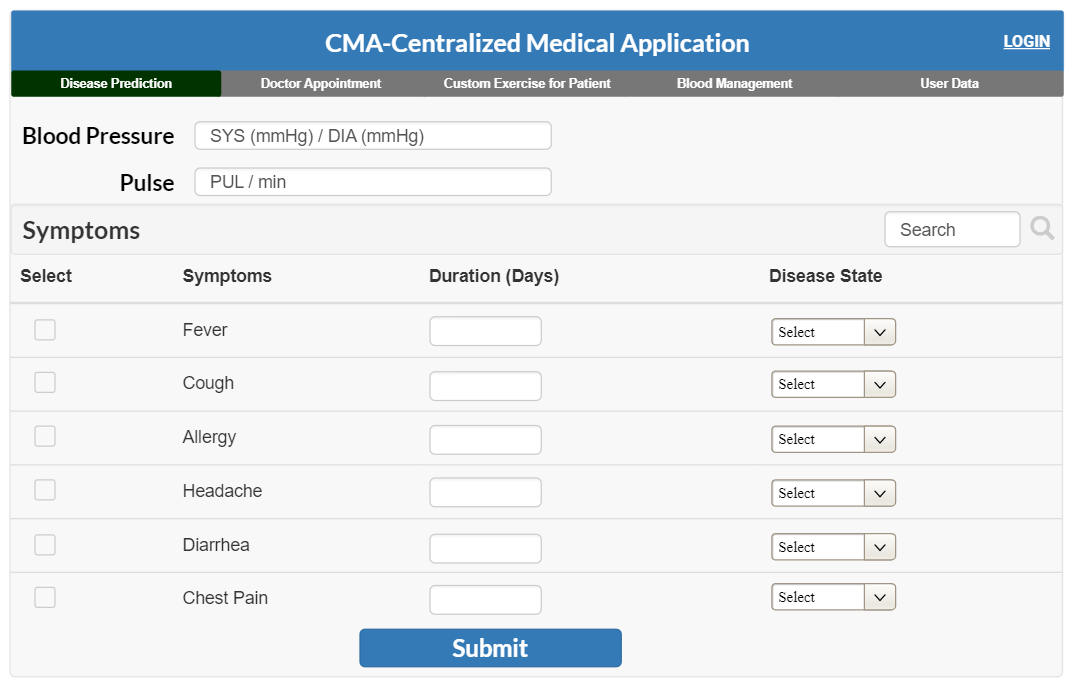
**Priority:** Medium

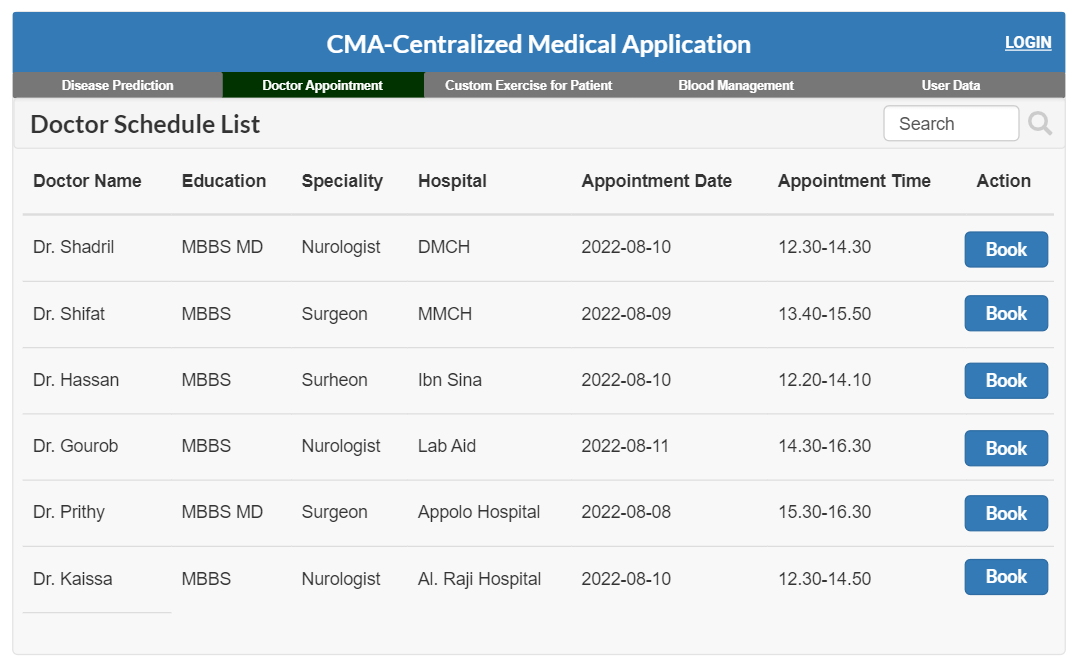
**Precondition:** The system must have a well-defined data model and API, with clear documentation and support for standard data formats.

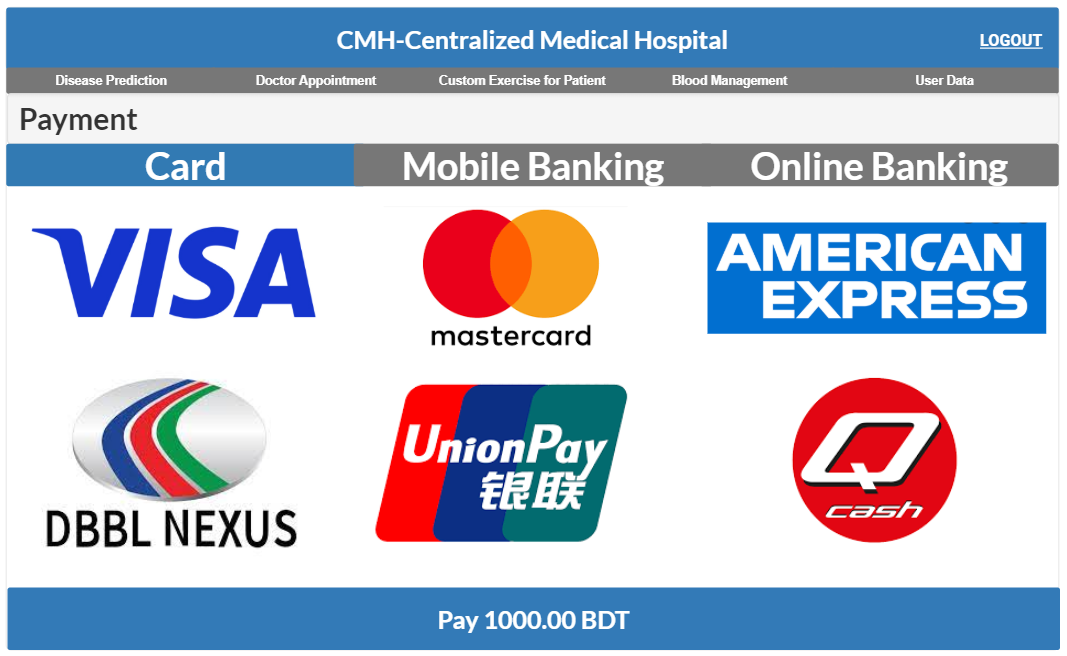
**Cross reference:** QA-3, QA-4, QA-5, QA-11

## System Interface

**UI/UX design:**







## 

## Project Requirements

**Time:**

Constructive Cost Model:

* Project type : Organic (Basic COCOMO Model)
* Coefficient<effort factor>  : 2.40 [P=1.05, T=0.38]
* SLOC : 10000 Lines
* Person Months : (2.40\* 101.05) = 26.93
* Dev. time, DM : (2.50\* 26.930.38) = 8.74 = 9 Months = 2160 WH

**Budget:**

Required People, ST: PM/DM = 26.93/8.74 = 3.80 = 4

Developer Salary in 9 months:

Per Developer salary Per working Hour = 700 Taka

Total Developer salary = 700\*2160 = 15,12000 Taka

Requirement Analysis:

Time Needed: 5 weeks (25 working days = (25\*12) =300 WH)

Req Analyst Person’s Hourly wage = 300 Taka

Total Req. Analyst salary = 300\*300= 90,000 Taka

Hardware Expense Estimation: 100000 Taka

Rent Expense:

Room per Month = 10,000 Taka

Total in 9 Months = 90,000 Taka

Total Utilities in 9 Months (including miscellaneous): 20,000 Taka

Maintenance (Till 6 Months after Delivery):

Expense per Hour: 1000 Taka

Total Estimated Time needed for Maintenance 60 Hours

Total Estimated Maintenance Expense = 60\*1000 = 60,000 Taka

Total Estimated Expense:

1512000 + 90000 + 100000 + 90,000 + 20,000 + 60,000 = 18,72,000Taka

Profit:

20% of Total Estimated Expense = 20%\*18,72,000 = 3,74,400 Taka

**Project Budget:** 18,72,000+ 3,74,400 = **22,46,400 Taka**

**Resources:**

|  |  |  |
| --- | --- | --- |
| **Hardware/Software** | **Resources** | **Quantity** |
| Hardware | Server | 2x rack-mounted servers, 1x tower servers |
| Hardware | Workstations | 10x desktop computers, 2x laptops |
| Hardware | Network equipment | 1x router, 2x switches, 1x firewall |
| Hardware | Printers | 1x laser printers, 1x inkjet printer |
| Software | Operating system | 10x Windows 10 licenses, 2x Linux licenses |
| Software | Database | 1x MySQL database |
| Software | Web servers | 2x Apache web servers, 1x Nginx web server |
| Software | Development tools | 5x Visual Studio licenses, 2x Eclipse licenses |
| Software | Testing tools | 1x Selenium testing suite, 1x JMeter load testing tool |

**Environment:**

**Development environment:** Software developers use a development environment as a tool to create, test, and debug new software prior to its release.

**Testing environments:** They should be similar to the production environment, but with reduced data volumes to avoid impacting the production environment. Examples include testing new software releases before they are deployed to production.

**Production environments:** They must be stable and reliable to protect end-users and the business, as any issues could lead to lost revenue and damage.

# FEATURES NOT TO BE TESTED

# Notification: When a patient request for appointment or a doctor accept appointments, Notification goes to their account. These features are not much important. So, this feature are not need to test right that version.

# Custom exercise for patient: Here, a user can see about heath related advice. Since that feature is not much important , so it’s not necessary for testing.

# TESTING APPROACH

## Testing Levels

## ● Unit Testing: Our project now still in the implementation phase so we assume that we will do the Unit Testing during our system development. In this testing we will tests individual software modules and see whether the individual system module has error or not. This testing methodology is done by the software developers and QA staff. This testing goal is to ensure that each unit of software code works as intended. In this step, we will follow “White Box Testing” technique.

● **Integration Testing:** Our project now still in the implementation phase so we assume

that in this integration testing we will make sure that all the software modules are

integrated logically and tested as a group and working correctly. The goal of this level

of testing is to find flaws in the way various software modules interact when they're

integrated. In this step, we will follow the “Bottom-up Integration” technique.

● **System Testing:** Our project now still in the implementation phase so we assume that

through the system testing we will test of full-featured, fully integrated system. Then

we will verify if it meets all the requirement. Black-box testing falls under this

condition. So, in this level, we will follow “Black Box Testing” technique.

● **Acceptance Testing:** Our project now still in the implementation phase so we assume

that we will do this acceptance testing for checking the acceptability of our product.

This test will be done to check whether any defect missed during the functional

testing phase. In this level, we will follow the “Black Box Testing” technique. After

that, we may run unit tests again.

## Test Tools

**Selenium:** It is an open-source tool that is used for automating web browsers. It allows a tester to write tests in various programming languages and run them on multiple browsers.

**Postman:** It is an API testing tool that is used for testing RESTful APIs. It allows a tester to test APIs without writing any code.

**LoadRunner:** It is a performance testing tool that is used to test the performance and scalability of applications.

**Visual Studio Test Professional:** It is a testing tool that is used for manual and automated testing of applications developed using Microsoft technologies.

**6.3 Meetings**

We will meet once after every one or two days through MS Teams to evaluate progress to date

and to identify error trends and problems as early as possible. Our test team leader will meet with

development and the project manager once after every two days as well. These two meetings will

be scheduled on different days. Additional meetings can be called as required for emergency

situations.

# TEST CASES/TEST ITEMS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Name: Centralized Medical Application -CMA | Test Designed by: Abu Shaleh Md. Kaium | | | |
| Test Case ID: FR\_1 | Test Designed date: 19-Apr-2023 | | | |
| Test Priority : Medium | Test Executed by: | | | |
| Module Name: Disease prediction | Test Execution date: | | | |
| Test Title: Verify diseases prediction by suggesting doctor. | | | | |
| Description: Test website disease prediction page. | | | | |
| Precondition: The user must accurately enter all required information in the disease prediction page. | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Go to the diseases   prediction option.   1. Enter Blood pressure and pulse. 2. Check option of symptoms. 3. Enter duration days. 4. Enter disease state.   Click on submit. | Blood pressure:120/80  Pulse: 70  Symptoms: Check Fever  Duration: 7  Disease state: High | The interface should suggest some doctors for that disease. |  |  |
| Post condition: User’s must need to log in with their valid ID and Password. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Centralized Medical Application - CMA | | Test Designed by: Thouhida Tasnim | | | |
| Test Case ID: FR\_2 | | Test Designed date: 19-Apr-2023 | | | |
| Test Priority: High | | Test Executed by: | | | |
| Module Name: Blood Management | | Test Execution date: | | | |
| Test Title: Verify Blood Management | | | | | |
| Description: Test web Blood Management | | | | | |
| Precondition: The user needs to log in to the system | | | | | |
| Test Steps | Test Data | | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Navigate to the website and login using valid credentials. 2. Click on the "Blood Management" tab in the menu. 3. Click on search box. 4. Give desired blood group and location. 5. Click on action button. | Username: "Piash@123" Password: "123@iL".  Blood group type: "O-".  Location: "Dhaka".  Action: Click on action button. | | The system should display a list of relevant blood donors based on the entered location and blood group. The displayed information should be accurate, and the contact information of the selected donor should be correct. |  |  |
| Post Condition: The system should display a list of relevant blood donors based on the entered location and blood group. | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Name: Centralized Medical Application - CMA | | Test Designed by: Abu Shaleh Md. Kaium | | |
| Test Case ID: FR\_3 | | Test Designed date: 19-Apr-2023 | | |
| Test Priority: High | | Test Executed by: | | |
| Module Name: Doctor Appointment | | Test Execution date: | | |
| Test Title: Verify Doctor Appointment | | | | |
| Description: Test web Doctor Appointment | | | | |
| Precondition: The user needs to log in to the system. | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Go to website and click on doctor appointment.  2. Click on search option.  3. Type a doctor’s name or select from suggested doctors list.  4. Check doctors name, specialty, appointment date, hospital name, available time.  5. Click on book option. | **For search option-**  Doctor’s name: Dr. Shadril.  Specialty : Neurologist  Action: **Click on Book** | Go to payment option successfully. |  |  |
| Post Condition: This user's time slot will be reserved on database. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Name: Centralized Medical Application - CMA | | | Test Designed by: Abu Shaleh Md. Kaium | |
| Test Case ID: NFR\_1 | | | Test Designed date: 06-Aug-22 | |
| Test Priority: High | | | Test Executed by: | |
| Module Name: Maintainability | | | Test Execution date: | |
| Test Title: Verify the responsiveness of System to solve problem within 3 hours | | | | |
| Description: Test if system can solve the problem within 3 hours. | | | | |
| Precondition: User must Login with valid username and password. | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1.Go to the application and login.  2.Click on Doctor’s appointment.  3. click on Book for taking appointment of a doctor. | **For search option-**  Doctor’s name: Dr. Shadril  Speciality : Nurologist  Action: **Click on Book** | Doctor’s appoinment problem should be solved within 2 hours or less. |  |  |
| Postcondition: N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Name: Centralized Medical Application- CMA | | | Test Designed by: Abu Shaleh Md. Kaium | |
| Test Case ID: NFR\_2 | | | Test Designed date: 06-Apr-2022 | |
| Test Priority: High | | | Test Executed by: | |
| Module Name: Availability | | | Test Execution date: | |
| Test Title: Verify the availability of the system | | | | |
| Description: Test the availability of the system between 8:00 a.m. to 8:00 p.m. | | | | |
| Precondition: User must login to the system | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Go to the website and click on disease prediction. 2. Use the system from 8:00 a.m. to 8:00 p.m. For 10000 times with automated software. | Blood pressure:120/80mmHg  Pulse: 70  Symptoms: Fever  Duration: 7  Disease state: Medium  (Use different data on every test) | The system must be 99% available between 8.00 a.m. and 8:00 p.m. local time |  |  |
| Post Condition: The system should be able to handle 10000 requests from the automated software without any downtime or performance issues. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | |  | |
|  | | |  | |
|  | | |  | |
|  | | |  | |
|  | | | | |
|  | | | | |
|  | | | | |
|  |  |  |  |  |
|  |  |  |  |  |
|  | | | | |

# ITEM PASS/FAIL CRITERIA

The pass/fail criterion for that phase is the expected result (outcome or response). A test step is considered successful and its requirement validated if it was carried out, the predicted result occurred, and this was either observed or recorded. In any other case, it will be a failure. The application's primary features should function as designed, and more than 95% of test cases should pass without revealing any serious flaws.

# TEST DELIVERABLES

* **Acceptance Test Plan:** It is a document that details everyone on the team's roles and duties during the Acceptance Testing phase.
* **System/Integration Test Plan:** This document details the strategy for testing the system and the integration.
* **Unit Test Plans/Turnover Documentation:** This document describes the unit testing strategy and includes the unit testing documentation to be turned over to the client.
* **Screen Prototypes:** These prototypes represent system displays and their corresponding user interfaces.
* **Report Mock-ups:** These are the visual representations of the reports generated by the system.
* **Defect/Incident Reports and Summaries:** This document contains a record of all major and minor incidents that occurred during testing.
* **Test Logs and Turnover Reports:** These documents contain detailed information about the tests performed and their results.

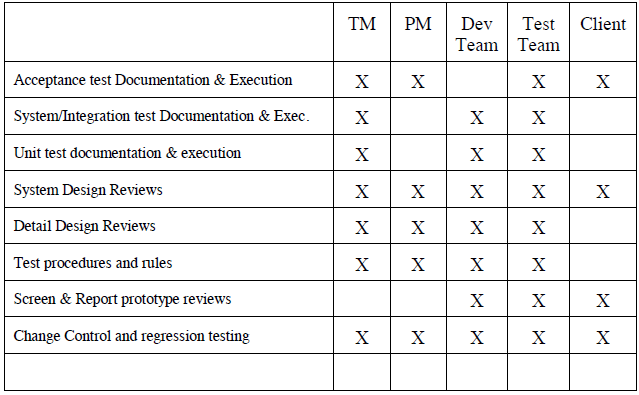
# STAFFING AND TRAINING NEEDS

To ensure proper testing, it is recommended to have a full-time tester assigned to the project during the system/integration and acceptance testing phases. At the start of the project, a person should be assigned part-time to participate in reviews, and after approximately three months into the project, they should be assigned full-time. If there is no dedicated tester available, the project manager or test manager can take on the role.

Training should be provided in the following areas to ensure complete and proper testing:

* The developers and testers need to be trained on the basic operations of the EDI (Electronic Data Interchange) interface.
* The operations staff will also require complete training on the EDI communications process prior to the final acceptance of the project.
* The sales administration staff will require training on the new screens and reports.

# RESPONSIBILITIES



# TESTING SCHEDULE

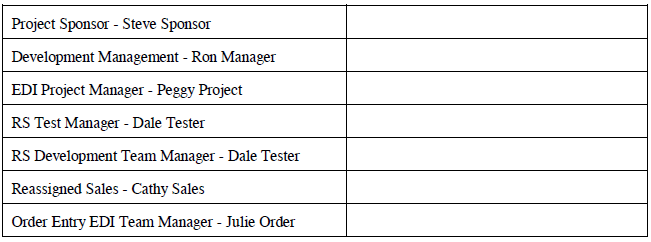
Time has been allocated within the project plan for the following testing activities. The specific dates and times for each activity are defined in the project plan timeline. The persons required for each process are detailed in the project timeline and plan as well. Coordination of the personnel required for each task, test team, development team, management and customer will be handled by the project manager in conjunction with the development and test team leaders.



# PLANNING RISKS AND CONTINGENCIES

* Limited Reassigned Sales staff. The Reassigned Sales administration staff currently has two positions unfilled. As a result of this staff shortage there may be delays in getting staff to review appropriate documents and to participate in the Acceptance test process. Should client staff become a problem, the appropriate dates for reviews and acceptance testing will slip accordingly. No attempt will be made to bypass any part of the review and testing processes.

# APROVALS



**Text Format**

* Style: Times New Roman
* Size: 12
* Line and Paragraph Spacing: 1.00
* Alignment: Justify