clinic Web Application System

A CSCI 313 project

Presented to: DR. Mourad Raafat

Eng / Rahma

**SHAHD AMR ZAKI 221001747**

**FARIDA ELHAUSSINY 221001958**

**YASSIN HAMAD 2210000948**

**YASSIN IZER 221001868**

Table of Contents

1.Introduction4

1.1Abstract………………………………………………………………………………………………….4

1.2 Project Idea …………………………………………………………………………………………...4

1.3 Problem Statement…………………………………………………………………………………...5

‎1.4 User Personas………………………………………………………………………………………...5

‎**2. Overall Description………………………………………………………………………….6**

‎2.1 Purpose‎....................................................................................................................................................6

2.2 Scope of the project.7

2.3 Technologies Used...7

2.4 Intended Audience.8

2.5 Overview of the document.8

**3. Functional Requirements............................................................................9**

3.1 Main Window………………………………………………………………………….9

3.2 Dashboard …………………………………………………….9

3.3 Weekly Schedule ……………………………………………………....9

3. 4 Appointment Management ………………………………………………………......10

3. .5 Prescription Management………………………………………………………….........10

**4.Non-Functional Requirements.........................................................................14** 4.1Usability............................................................................................................................14

4.2 Reliability...................................................................................................................14

4.3Performance........................................................................................................................14

4.4Availability..........................................................................................................................14

4.5Maintainability....................................................................................................................15

4.6 Compatibility....................................................................................................................15

**5.Diagrams.........................................................................................................16**

5.1. Use case Diagram .......................................................................................................... 16

5.1.1. Use case scenarios................................................................................................ 17

5.2. Sequence Diagram ......................................................................................................... 26

5.3 Class Diagram..................................................................................................................27

5.4 Architecture Diagram……………………………………………………………………29

5.5Activity Diagram……………………………………………………………………30

5.6State Diagram……………………………………………………………………31

**6.Interfacae…………………………………………………………………………………**

**7.Test Cases…………………………………………………………………………………….**

**8**. **Risk Identification…………………………………………………………………..32**

**9.Future Work………………………………………………………………………..**33

**1. Introduction**

**1.1 Abstract**

SWproject is a WPF application aimed at managing user login, appointments, and prescriptions in a healthcare context. It connects to a MySQL database to fetch and manipulate data, providing a user-friendly interface for both patients and healthcare providers.

**1.2 Project Idea**

The project aims to create a desktop application that allows users to log in, manage appointments, view schedules, and handle prescription records efficiently.

**1.3 Problem Statement**

Managing patient information and appointments can be cumbersome in traditional settings. This application seeks to streamline these processes, improving accessibility and efficiency for both patients and healthcare providers.

**1.4 User Personas**

#### Persona 1: The Doctor

* **Name:** Admin
* **Age:** 26
* **Occupation:** General Practitioner
* **Goals:**
  + Efficiently manage patient appointments .
  + Efficiently manage prescriptions..
* **Needs:**
  + A system to log patient visits and manage prescription data.

**Persona 2: secartary**

* **Name:** shahd amr
* **Age:** 20
* **Occupation:** student
* **Goals:**
  + Easily book appointments .
  + view medical prescriptions.
* **Needs: A system to log patient visits and manage prescription data.**

**2. Overall Description**

**2.1 Purpose**

The purpose of the SWproject application is to facilitate the management of appointments and prescriptions, providing a seamless experience for users.

**2.2 Scope**

The application allows users to log in, view a dashboard with summary data, manage appointments, and handle prescription information. It automates data retrieval from a MySQL database, ensuring up-to-date information.

**2.3 Technologies Used**

**Programming Language:** C

**Framework : WPF (Windows Presentation Foundation)**

**Other Tools:** GitHub for version control

**Data Base:** MySQL light

**Libraries** : MySql.Data for database connectivity

**2.4 Intended Audience**

* **- Healthcare providers looking to streamline appointment management.**
* **- Patients seeking to manage their healthcare information efficiently.**
* **- Developers interested in healthcare application development**
* **2.5 Overview**

This project results in a clinic that is scalable . This documentation outlines the system architecture, functional and non-functional requirements, and user interface details to support development and testing efforts.

**3. Functional Requirements**

**3.1 Main window**

**3.1.1 Login Functionality:**

* **Description** **Users can enter their username and password to access the application.**

**3.1.2 Error Handling:**

* **Description: Displays appropriate messages for invalid login attempts or connection issues**

**3.2 Dashboard**

**3.2.1 Title: Data Display**

* **Description: Shows summary information about new and old patients, total appointments, and today's requests.**

**3.2.2 Title: Dynamic Updates**

* **Description: Retrieves data from the MySQL database to provide real-time updates.**

**3.3 Weekly Schedule**

**3.3.1 Schedule Management**

* **Description: Displays weekly class schedules for different days.**

**3.3.2 Data Binding**

* **Description: Uses Observable Collections to bind schedule data to the user interface.**

**3.4 Appointment Management**

**3.4.1 Add Appointment**

**• Description: Users can schedule appointments with validation for necessary fields.**

**3.4.2 Navigation**

**• Description: Allows users to navigate back to the dashboard or other pages.**

**3.5 Prescription Management**

**3.5.1 Submit Prescription**

**• Description: Users can enter prescription details with validation checks.**

**3.5.2 Clear Fields**

**• Description: Resets input fields after successful submission to enhance user experience.**

**4. Non-Functional Requirements**

**4.1 Usability**

* The application should have a clean, intuitive interface that allows users to navigate easily between different functionalities.

**4.2 Reliability**

* The system should maintain a high uptime, handling user requests without failures during peak usage times.

**4.3 Performance**

* The application should execute database queries efficiently, displaying results in under 3 seconds.

**4.4 Availability**

* The application should be available for use whenever the user needs to access it, assuming the system is running.

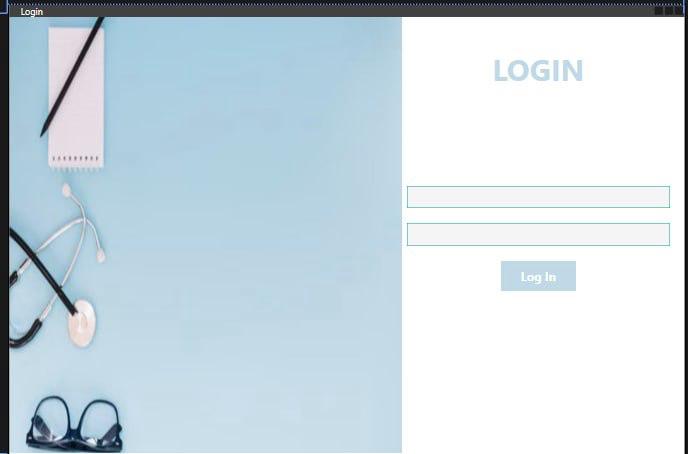
**4.5 maintainability**

* The code should be well-documented, allowing for easy updates and feature additions.

**4.6 Compatibility**

* The application should be available for use whenever the user needs to access it, assuming the system is running.

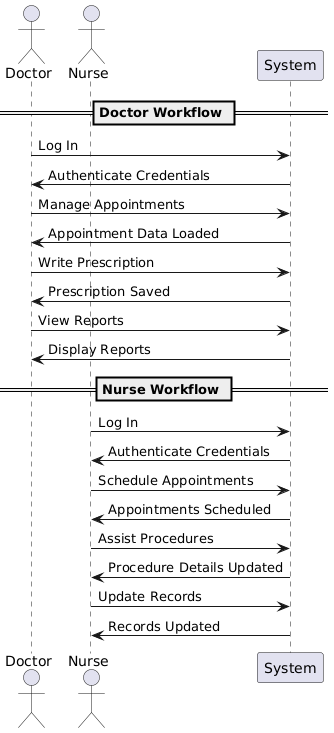
**System prototype:**



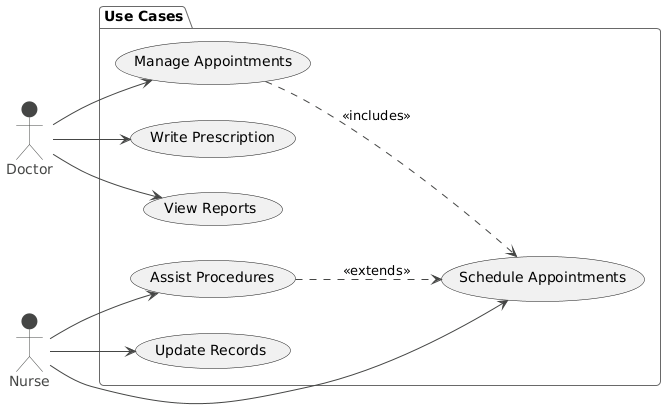


**5.Diagrams**

**5.1 Sequence diagram**



**5.2 Use Case Diagram**



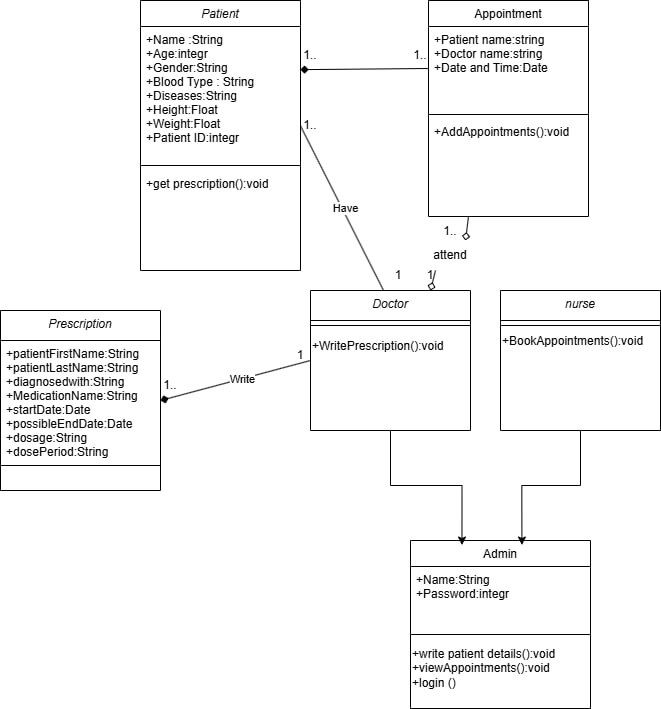
|  |  |
| --- | --- |
| **User case Name** | **Manage Appointments** |
| **Actors** | **Doctor, Nurse** |
| **Main Success scenario** | **1. The doctor views the list of existing appointments.**  **2. The doctor selects an appointment to manage.**  **3. The system displays the appointment details.**  **4. The doctor edits or cancels the appointment as needed.**  **5. The system updates the appointment details and confirms the changes.**  **6. The doctor receives a notification of the update.** |
| **Exceptions** | **- If the appointment cannot be found, the system displays an error message.**  **- If the doctor tries to cancel an already completed appointment, the system displays a restriction message.** |
| **Actions** | **Display the list of appointments. Allow selection of an appointment for management. Update or cancel the selected appointment.** |
| **Pre-Condition** | **The doctor must be logged into the system. The appointment must exist in the system.** |

|  |  |
| --- | --- |
| **User case Name** | **Write Prescription** |
| **Actors** | **Doctor** |
| **Main Success scenario** | **1. The doctor selects a patient.**  **2. The doctor writes a prescription.**  **3. The system validates the prescription details.**  **4. The doctor submits the prescription.**  **5. The system saves the prescription and confirms submission to the doctor.** |
| **Exceptions** | **- If the prescription details are incomplete, the system prompts for corrections.** |
| **Actions** | **Allow the doctor to write and submit prescriptions.** |
| **Pre-Condition** | **The doctor must be logged into the system. The patient must be selected from the system.** |

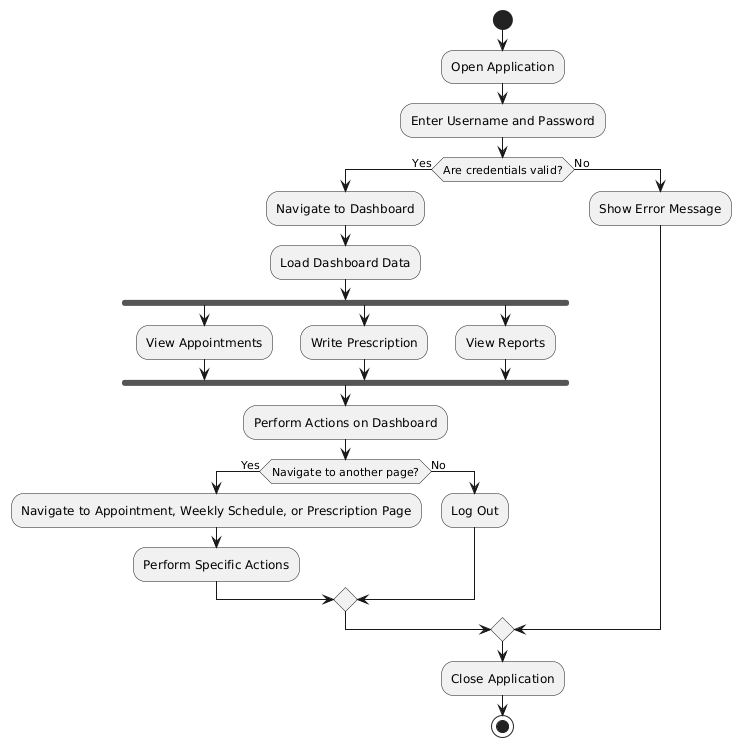
|  |  |
| --- | --- |
| **User case Name** | **View Reports** |
| **Actors** | **Doctor , Nurse** |
| **Main Success scenario** | **1. The actor selects the reports section.**  **2. The system displays available reports.**  **3. The actor selects a specific report to view.**  **4. The system generates and displays the report.** |
| **Exceptions** | **- If there are no reports available, the system displays a message indicating such.** |
| **Actions** | **Display available reports and generate selected reports.** |
| **Pre-Condition** | **The actor must be logged into the system.** |

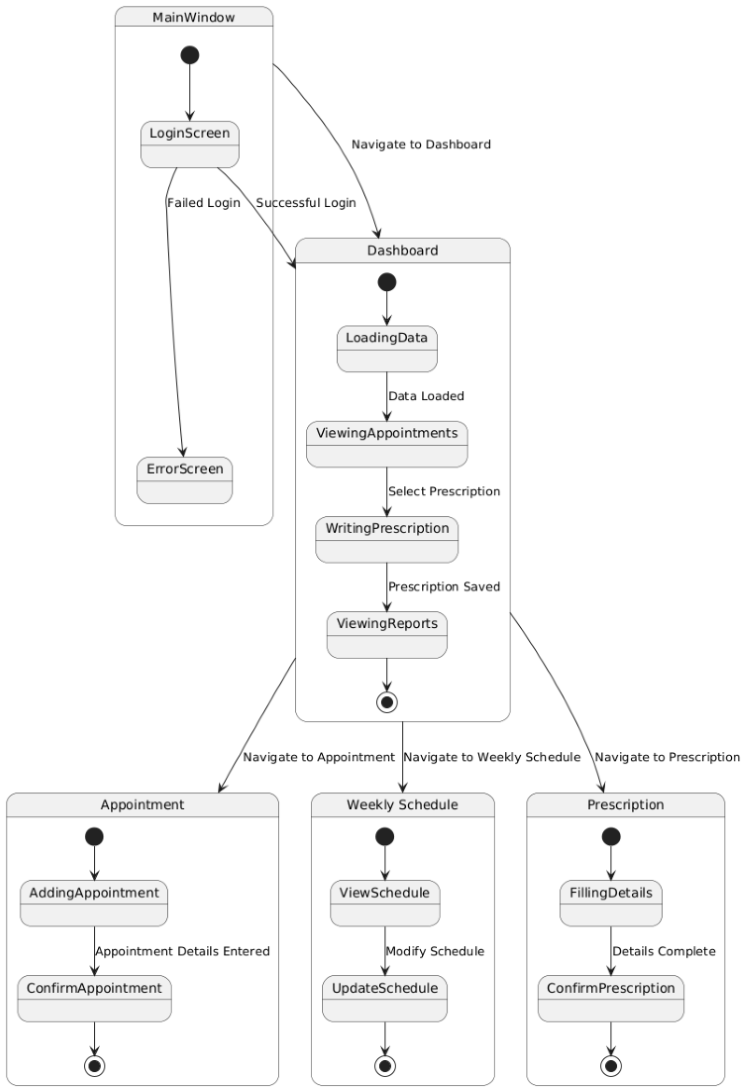
|  |  |
| --- | --- |
| **User case Name** | **Update Records** |
| **Actors** | **Nurse** |
| **Main Success scenario** | **1. The nurse selects a patient record to update.**  **2. The system displays the patient's current records.**  **3. The nurse makes necessary updates.**  **4. The system validates the updates.**  **5. The nurse submits the updates.**  **6. The system confirms the updates to the nurse.** |
| **Exceptions** | **- If the record cannot be found, the system displays an error message.** |
| **Actions** | **Allow the nurse to view and update patient records.** |
| **Pre-Condition** | **The nurse must be logged into the system. The patient record must exist in the system.** |

**5. 3 Class diagram**

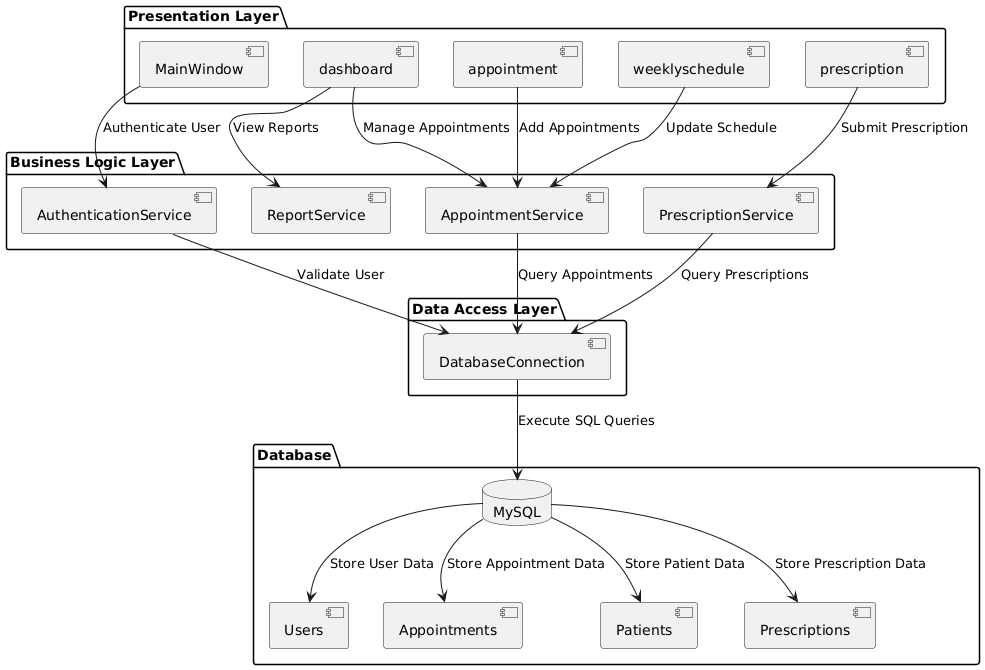


**5. 4 Activity Diagram**





**5. 6 Architecture diagram**



**6 Software Interface:**

The user interface consists of a main window for login, a dashboard to display summary information, pages for appointment management, weekly schedules, and prescription management.

|  |  |
| --- | --- |
| **7 Test Cases**  **7.1 Test Case 1:** **Login** | |
| **Test Case ID:** Login-0.1 | **Test Designed by:** <Name > |
| **Test Priority (Low/Medium/High):** High | **Test Designed date:** <Date> |
| **Module Name:** Doctor login screen | **Test Executed by:** <Name> |
| **Test Title:** Login | **Test Execution date:** <Date> |
| **Description:** Login successfully with valid username and password. |  |
|  |  |
|  |  |
| **Pre-conditions:** User has a valid account. | |
| **Dependence:** Stable internet connection.  **Post-conditions:** User is validated with database and successfully login to account. The account session details are logged in database. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Test Steps | Test Data | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
| 1 | Open login page |  | User can open login page | As Expected, | Pass |  |
| 2 | Enter valid username | User= “admin”. | User can enter username | As Expected, | Pass |  |
| 3 | Enter valid password | Password: “Password”. | User can enter password | As Expected, | Pass |  |
| 4 | Click on Login button |  | User should be able to login | As Expected, | Pass | Login successful |

|  |  |
| --- | --- |
| **Test Case 1** | |
| **Test Case ID:** Login-0.2 | **Test Designed by:** <Name > |
| **Test Priority (Low/Medium/High):** High | **Test Designed date:** <Date> |
| **Module Name:** Doctor login screen | **Test Executed by:** <Name> |
| **Test Title:** Login | **Test Execution date:** <Date> |
| **Description:** Failed login with invalid username or password. |  |
|  |  |
|  |  |
| **Pre-conditions:** User has an invalid username and invalid password | |
| **Dependence:** Stable internet connection. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | Open login page |  | User can open login page | As Expected | Pass |  |
| 2 | Enter invalid username | User= admie (invalid) | User can enter username | As Expected | Pass |  |
| 3 | enter invalid password | Password: 1233 ) invalid) | User can enter password | As Expected | Pass |  |
| 4 | Click on Login button |  | User should be able to login | Unsuccess login | Fail | User enters invalid username and password. |

|  |  |
| --- | --- |
| **Test Case 2** | |
| **Test Case ID:** Appointment-1 | **Test Designed by:** <Name > |
| **Test Priority (Low/Medium/High):** Medium | **Test Designed date:** <Date> |
| **Module Name** :**Appointment Management Module** | **Test Executed by:** <Name> |
| **Test Title:** Create Appointment | **Test Execution date:** <Date> |
| **Description:** Attempt to create a new appointment. |  |
|  |  |
|  |  |
| **Pre-conditions:** User must be logged in and on the appointment page. | |
| **Dependencies:** System must check for valid input fields.  **Post-conditions**: Appointment is created, and a confirmation message is displayed | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | Open the Appointment module page | Navigate to the appointment page. | Appointment page opens successfully. | As Expected | Pass |  |
| 2 | Enter a valid patient name | Patient Name: | Patient name is entered successfully. | As Expected | Pass |  |
| 3 | Select a valid appointment date | Appointment Date: | Date is selected successfully. | As Expected | Pass |  |
| 4 | Select a valid appointment time | Appointment Time: | Time is selected successfully. | As Expected | Pass |  |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7.3 Test Case 3: Error Handling** | | | | | | | |
| **Test Case ID:** Appointment-2 | | | | **Test Designed by:** <Name > | | | |
| **Test Priority (Low/Medium/High):** Medium | | | | **Test Designed date:** <Date> | | | |
| **Module Name:** Appointment Management Module | | | | **Test Executed by:** <Name> | | | |
| **Test Title:** Create Appointment with Missing Fields | | | | **Test Execution date:** <Date> | | | |
| **Description:** ttempt to create an appointment with missing fields. | | | |  | | | |
|  | | | |  | | | |
|  | | | |  | | | |
| **Pre-conditions:** User must be logged in and on the appointment page | | | | | | | |
| **Dependencies:** System must validate input fields.  **Post-conditions**: Appointment creation fails, and an error message is displayed | | | | | | | |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | | **Actual Result** | **Status (Pass/Fail)** | **Notes** | |
| 1 | Open the Appointment module page | Navigate to the appointment page. | Appointment page opens successfully. | | As Expected | Pass |  | |
| 2 | Leave patient name empty | Patient Name: "". | Error message displays: "Please fill in all fields correctly." | | As Expected | Pass |  | |
| 3 | Leave date field empty | |  | | --- | | Appointment Date: null! |  |  | | --- | |  | | Error message displays: "Please fill in all fields correctly." | | As Expected | Pass |  | |
| 4 | Select invalid time | Appointment Time: "Select Time" | Error message displays: "Please fill in all fields correctly." | | As Expected | Pass |  | |

**8. Risk:**

**Technical Risks**

* **System Integration Issues**

**Description: Difficulty in integrating various modules (e.g., appointment scheduling, prescription management).**

**Impact: Delays in deployment and functionality.**

**Mitigation: Establish clear integration protocols and conduct regular integration testing.**

* **Software Bugs and Defects**

**Description: Presence of untested or poorly tested code may lead to issues.**

**Impact: User dissatisfaction and potential system failures.**

**Mitigation: Implement rigorous testing protocols, including unit, integration, and user acceptance testing.**

* **Operational Risks**

**User Resistance to Change**

**Description: Staff may resist adopting new software due to comfort with existing processes.**

**Impact: Reduced utilization of the system and potential implementation failure.**

**Mitigation: Conduct comprehensive training sessions and provide ongoing support to ease the transition.**

**Insufficient Training**

**Description: Users may not be adequately trained on the new system.**

**Impact: Increased errors and inefficiencies in operations.**

**Mitigation: Develop a robust training program with materials and resources for ongoing education.**

**Project Management Risks**

* **Scope Creep**

**Description: Uncontrolled changes or continuous growth in project scope.**

**Impact: Delays and budget overruns.**

**Mitigation: Establish a clear scope definition and change management process.**

**Resource Availability**

**Description: Key personnel may become unavailable due to other commitments.**

**Impact: Project delays and knowledge gaps.**

**Mitigation: Develop a resource management plan and ensure backup resources are available.**

* **Financial Risks**

**Budget Overruns**

**Description: Project costs exceed initial estimates.**

**Impact: Funding shortfalls and potential project cancellation.**

**Mitigation: Maintain a contingency budget and regularly review financial status.**

**Unforeseen Costs**

**Description: Unexpected expenses related to software or hardware requirements.**

**Impact: Financial strain on the project.**

**Mitigation: Conduct thorough market research and risk assessments during the planning phase.**

* **Security Risks**

**Data Breaches**

**Description: Unauthorized access to sensitive patient data.**

**Impact: Legal liabilities and loss of trust.**

**Mitigation: Implement strong encryption, access controls, and regular security audits.**

**Compliance Violations**

**Description: Failure to comply with regulations such as HIPAA (Health Insurance Portability and Accountability Act).**

**Impact: Fines and legal repercussions.**

**Mitigation: Regularly review compliance requirements and conduct training on data protection laws.**

* **External Risks**

**Vendor Reliability**

**Description: Dependence on third-party vendors for software or hardware support.**

**Impact: Service disruptions if vendors fail to deliver.**

**Mitigation: Conduct thorough vendor assessments and establish service level agreements (SLAs).**

**Market Changes**

**Description: Rapid changes in technology or market demands.**

**Impact: The system may become obsolete quickly.**

**Mitigation: Stay informed about industry trends and be prepared to adapt the system as needed.**

* **Environmental Risks**

**Natural Disasters**

**Description: Events such as floods, earthquakes, or fires impacting operations.**

**Impact: Damage to physical hardware and data loss.**

**Mitigation: Implement disaster recovery plans and maintain offsite backups.**

# **9.Future Work:**

Future work on the project may involve several enhancements aimed at improving functionality and user experience. One key enhancement is the integration of a notification system that sends automated appointment reminders to both patients and healthcare providers, helping to reduce no-show rates and improve scheduling efficiency. Additionally, expanding the database to include more detailed patient records, such as medical history, allergies, and treatment plans, will provide healthcare professionals with comprehensive information for better decision-making. Implementing user role management will also be crucial, allowing for tailored access levels and permissions based on user types (doctors, nurses, administrative staff), thereby enhancing security and ensuring that sensitive information is only accessible to authorized personnel. These improvements will collectively contribute to a more robust and user-friendly system.