

Software Engineering (SE)
Project Proposal
Laboratory Management System



Submitted by:

Affan Tahir 2023-CS-709

Submitted to:

Dr. Zeeshan

Dated: October 19th, 2024.

Department of Computer Science
University of Engineering and Technology Lahore, New Campus.

Table of Contents

Abstract.....	3
Introduction.....	3
Tools & Technologies.....	4
Software Tools & Technologies	4
Hardware Tools & Technologies	4
Vision	5
Functional Requirements.....	5
Non-Functional Requirements	6
Business Requirements	7
Business Rules.....	8
Physical Product Requirements	8
Development Constraints	8
User Requirements.....	10
User Stories.....	23
Use Cases	24
Story Board.....	29
Proposed Methodology	29
Timeline/Gantt Chart	31

Laboratory Management System

Abstract

The **Laboratory Management System** is a software designed to improve the daily operations of a medical lab. It has two main user roles: Admin and Laboratorian. Admins can manage staff, add new tests, monitor equipment, and view financial reports. Laboratorians can add patient details, record test results, and generate reports. The system enhances the workflow of a laboratory by automating the tasks and providing an attractive and user-friendly interface.

Introduction

Managing laboratory operations can be time-consuming and complex, especially when dealing with multiple patients, tests, and equipment. Traditionally, labs handle patient data, tests, and reports manually, which can lead to delays, errors, and inefficiency. To simplify this process, we developed a **Laboratory Management System**, which brings all lab activities into one easy-to-use platform.

With LMS, admins can manage laboratorians, add new tests, view machine inventories, and access financial reports—all from one place. Laboratorians can easily add patients, record test results, and generate reports quickly and accurately.

This system not only saves time but also enhances accuracy and efficiency in lab operations. While the system might not replace some advanced tools used in larger labs, it offers a streamlined and user-friendly solution to managing everyday lab tasks.

Problem Statement

How can we address the inefficiency, delays, and errors caused by the manual management of laboratory operations, including patient data and test results?

Objectives

1. **Automate Laboratory Operations:** Streamline processes like patient data entry, test recording, and report generation to reduce manual effort and errors.
2. **Improve Efficiency:** Enhance the speed and accuracy of handling patient tests and managing lab resources, such as equipment and test kits.
3. **Centralize Data Management:** Provide a single platform for managing all laboratory information, from patient records to machine inventories and financial reports.
4. **Enhance Patient Care:** Ensure timely and accurate delivery of test results to patients and medical staff, improving overall healthcare quality.
5. **Support Decision-Making:** Provide admins with tools to monitor lab performance, income, and inventory for better operational decisions.

Tools & Technologies

1. Microsoft Word:

- To Document the Proposal in a formatted way

2. Visual Studio:

- To Code and develop the App in C#

3. Mendeley:

- To write and link References

4. Pencil:

- To make Wireframes
- A **wireframe** is a simple, visual blueprint that outlines the layout and structure of a website or app, focusing on functionality and user interface without detailed design elements.

5. Languages:

- C# (Programming Language)
- It provides framework to develop desktop app more conveniently.

Software Tools & Technologies

1. MS Word:

- To Document the Proposal in a formatted way

2. Visual Studio:

- To Code and develop the App in C#

3. Mendeley:

- To write and link References

4. Pencil:

- To make Wireframes

Hardware Tools & Technologies

1. Server:

- To handle the database of the system and to connect all the laboratories together.

2. Power Supply:

- To ensure continuous service in the event of power outages.

3. Disaster Recovery Equipment:

- To ensure quick restoration in the event of a system failure, system should include disaster recovery tools, such as portable storage devices or cloud-based recovery solutions, backup servers.

4. Computer Systems/Laptops:

- For the development, testing, and maintenance of the desktop app.

Vision

My vision is to create a smooth and streamlined laboratory management system to enhance and optimize the daily operations of a laboratory.

Scope

The scope of this **Laboratory Management System** focuses on both Admin and Laboratorian functionalities to create a user-friendly platform for managing lab operations. Admins will be responsible for managing laboratorians, adding new tests, monitoring machine inventory, and viewing income reports. They will have secure access to these tools to oversee the lab's performance. On the laboratorian side, users will be able to add patient details, record and add test results, generate patient reports, and view statistics of the total tests. This system will simplify lab management for admins and laboratorians, ensuring smooth and efficient lab operations.

Functional Requirements

Actors:

1. Admin
2. Laboratorian

Admin:

1. Admin shall sign-in to the system.
2. Admin shall view all users.
3. Admin shall create, delete or edit a laboratorian user.
4. Admin shall add, edit or delete a test.
5. Admin shall view the machines inventory.
6. Admin shall add, delete or edit a machine for maintenance.
7. Admin shall view the profit from tests.
8. Admin shall view the maintenance cost of all machines.

Laboratorian:

1. Laboratorian shall be able to sign-In.
2. Laboratorian shall view all patients.
3. Laboratorian shall add, delete or edit a patient to the system.
4. Laboratorian shall view the test records of all patients.

5. Laboratorian shall add, delete or edit the test record of a patient.
6. Laboratorian shall generate report of the total tests done within selected timeline.

Non-Functional Requirements

1. Response Time:

The system must response within 2.5 seconds for any user interaction.

2. Throughput:

The system must process 200 requests within 1 second.

3. Utilization:

The system must not exceed 85% of CPU.

4. Scalability:

The system must have the ability to scale up to 10,000 users in future.

5. Capacity:

The system must accommodate up to 10 TB of user data.

6. Availability:

The system must be available 24/7.

7. Reliability:

The system must have an error rate of less than 0.01% during record entering and transactions to ensure high reliability.

8. Recoverability:

The system must recover from failure within 24 hours.

9. Maintainability:

The system must support regular updates and patches without causing service disruption.

10. Serviceability:

The system should diagnose issues or failures within an hour.

11. Security:

The system must ensure 100% encryption of sensitive data e.g. passwords and personal identifiable information (PII) of the patients and laboratorians.

12. Regulatory:

The system must ensure regulations regarding data privacy.

13. Manageability:

Admins must be able to add new laboratorians, manage their accounts, and update tests within 2 minutes for each task.

14. Environmental:

The system must function within a temperature range of 10°C to 50°C.

15. Data Integrity:

The system must maintain 99.99% accuracy in test records and data updates.

16. Usability:

The app must achieve a minimum of 85% user satisfaction in usability surveys.

17. Interoperability:

The system must integrate smoothly with external systems, such as payment gateways and report generation software.

Business Requirements

1. Lab Operations Management:

Create an efficient platform where admins and laboratorians can manage patient data, tests, and lab resources seamlessly.

2. Patient and Test Management:

Enable laboratorians to add, update, and manage patient details, test results, and reports in a streamlined and secure manner.

3. Admin Control Panel:

Provide admins with easy-to-use tools to manage laboratorians, oversee machine inventory, add new tests, and access financial reports efficiently.

Business Rules

1. User Verification:

Admin must add a new user's before to ensure the authenticity of accounts.

2. User Roles:

User must access and manage the functionalities that are assigned to it.

3. Data Privacy:

All user data, including personal and payment information, must be encrypted and stored securely.

Physical Product Requirements

2. Server:

- To handle the database of the system and to connect all the laboratories together.

5. Power Supply:

- To ensure continuous service in the event of power outages.

6. Disaster Recovery Equipment:

- To ensure quick restoration in the event of a system failure, system should include disaster recovery tools, such as portable storage devices or cloud-based recovery solutions, backup servers.

7. Computer Systems/Laptops:

- For the development, testing, and maintenance of the desktop app.

Development Constraints

1. Technology Stack:

The platform must be built using specific technologies, such as C# for developing the desktop application, and SQL Server for database management and storage.

2. Operating System Compatibility:

The desktop application must support Windows 10 and newer versions, restricting the choice of development tools and libraries to those that are compatible with the Windows environment.

3. Development Timeline:

The project must be completed within a 2-month timeframe, limiting the number of features that can be implemented and reducing the time available for extensive testing and debugging.

4. Budget Limitations:

The project has a set budget, limiting resources for hiring extra developers, purchasing software licenses, or investing in premium development tools and cloud services.

5. Version Control System:

The development process must adhere to the company's standard version control procedures, with specific branching strategies dictating how code is managed, merged, and deployed throughout the development lifecycle.

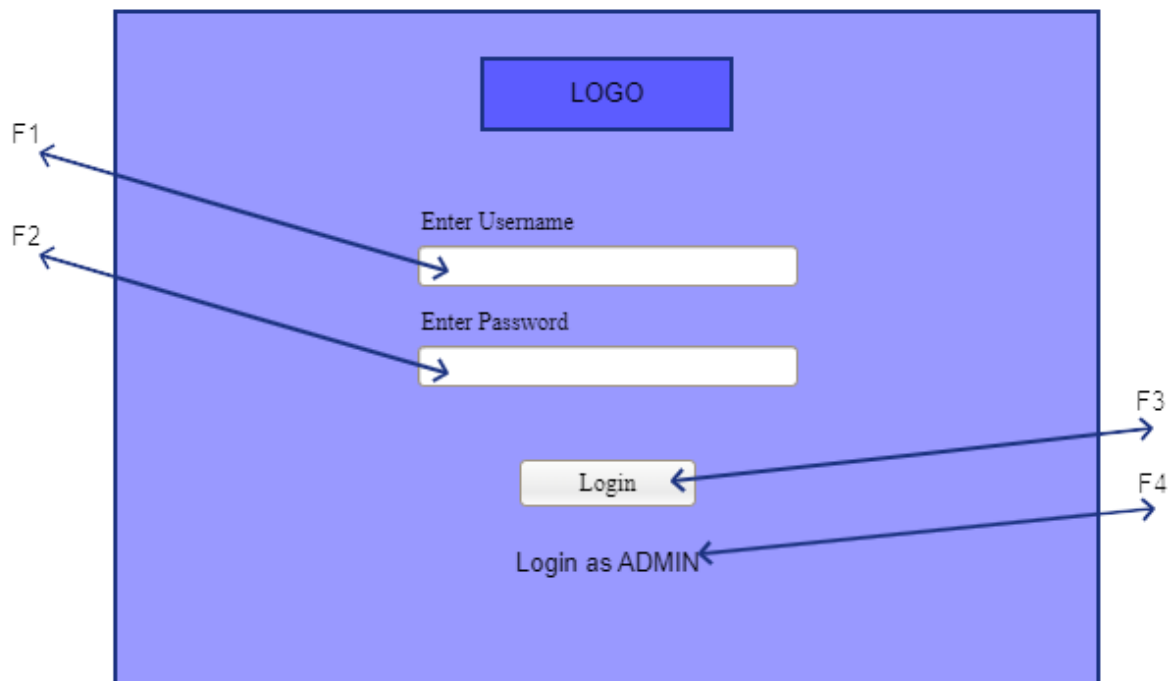
6. Coding Guidelines:

The project must follow the organization's predefined coding guidelines and standards, including specific naming conventions and documentation practices, ensuring consistency in code quality and structure across the project.

User Requirements

Wireframes

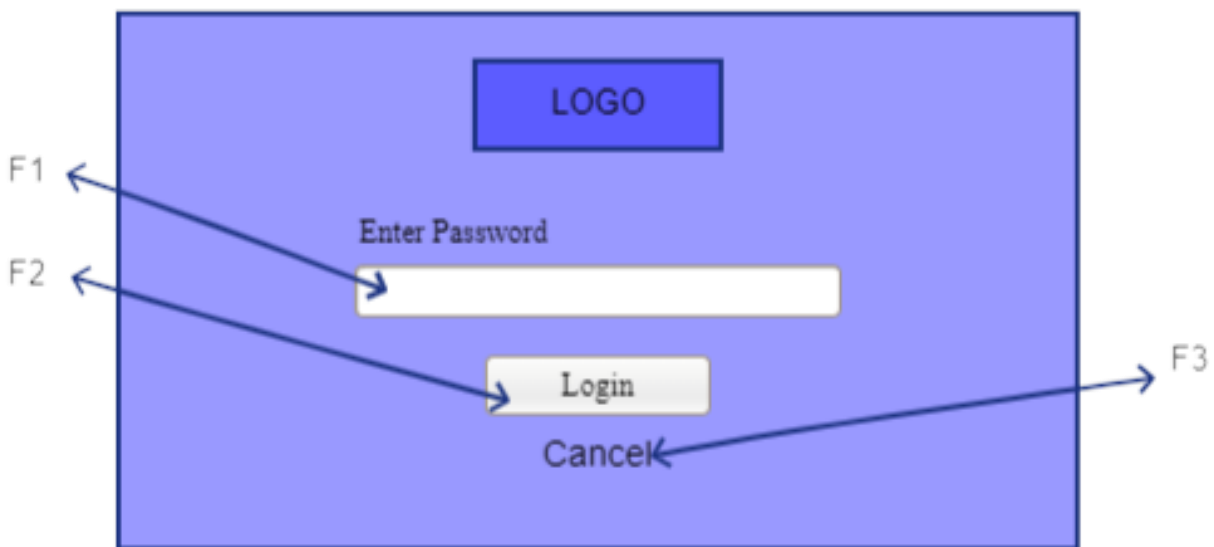
Login Page:



Requirements	Laboratorian	Admin
F1	As a laboratorian, I shall be able to enter my username.	NA
F2	As a laboratorian I shall be able to enter my password.	NA

F3	As a laboratorian, I shall jump to the laboratorian home page when I click login.	NA
F4	NA	As an admin, I shall jump to my admin login page when I click on this text.

Admin Login



Requirements	Admin
F1	As an Admin, I shall be able to enter my password in the text box.
F2	As an Admin, I shall jump to Admin Home Page when I click Login button.
F3	As an Admin, I shall go back to the main login page when I click Cancel.

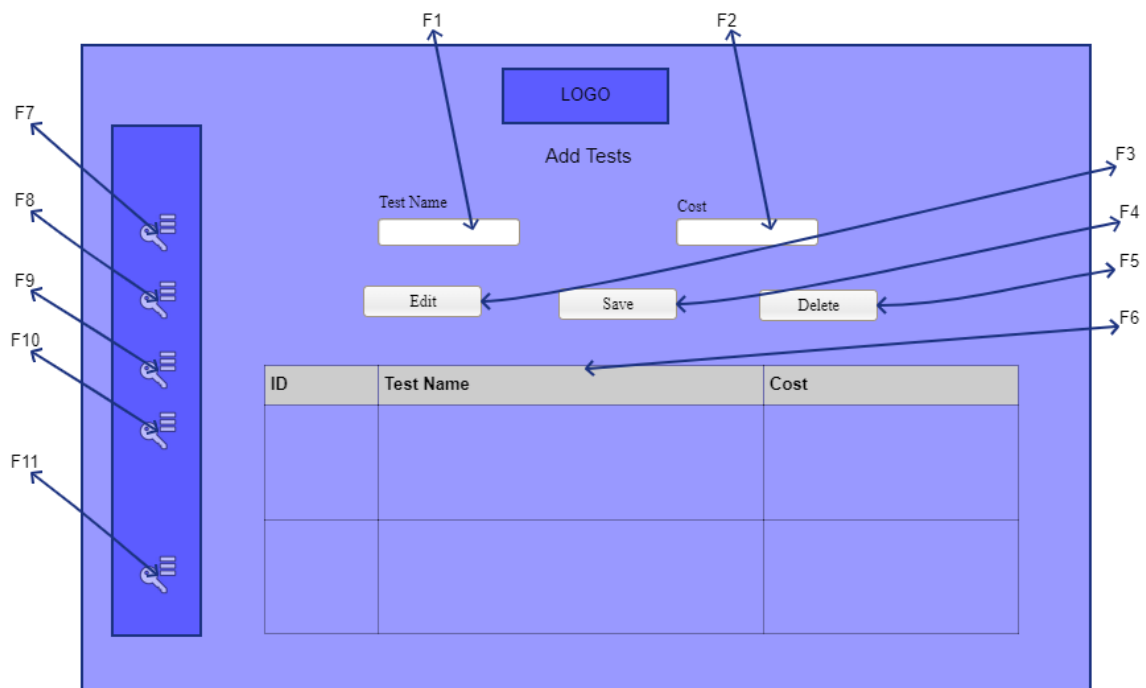
Add Laboratorians:

The screenshot shows a web form titled "Add Laboratorians" with a blue header bar containing a "LOGO" and the title. The form has several input fields: "Name" (F1), "Date of Birth" with a "Date Picker" (F2), "Phone" (F3), "Gender" (F4, a dropdown menu), "Username" (F5), "Password" (F6), and "Address" (F7). Below these fields are three buttons: "Edit" (F8), "Save" (F9), and "Delete" (F10). At the bottom is a table with 8 columns: ID, Name, DOB, Phone, Gender, Address, Username, and Password. The table has 3 empty rows. On the left side of the form is a vertical sidebar with four menu items, each with a key icon (F11, F12, F13, F14). At the bottom left is a footer area with four icons (F15, F16).

Requirements	Admin
F1	As an admin, I shall be able to enter the name of laboratorian.
F2	As an admin, I shall be able to pick the date of birth of the laboratorian.
F3	As an admin I shall be able to enter the phone number of the laboratorian.

F4	As an admin I shall be able to pick the gender of the laboratorian.
F5	As an admin I shall be able to assign a username to the laboratorian.
F6	As an admin I shall be able to assign a password to the laboratorian.
F7	As an admin, I shall be able to enter the address of the laboratorian
F8	As an admin, I shall be able to save the details of the laboratorian.
F9	As an admin, I shall be able to delete a laboratorian record from the database.
F10	As an admin, I shall be able to view the details of all laboratorians.
F11	As an admin, I shall be able to edit the details of an existing laboratorian.
F12	As an admin, I shall jump to the Add laboratorian page when I click the icon.
F13	As an admin, I shall jump to the Add Tests page when I click the icon.
F14	As an admin, I shall jump to the Inventory page when I click the icon.
F15	As an admin, I shall jump to the Income Reports when I click the icon.
F16	As an admin, I shall logout of the system when I click the icon.

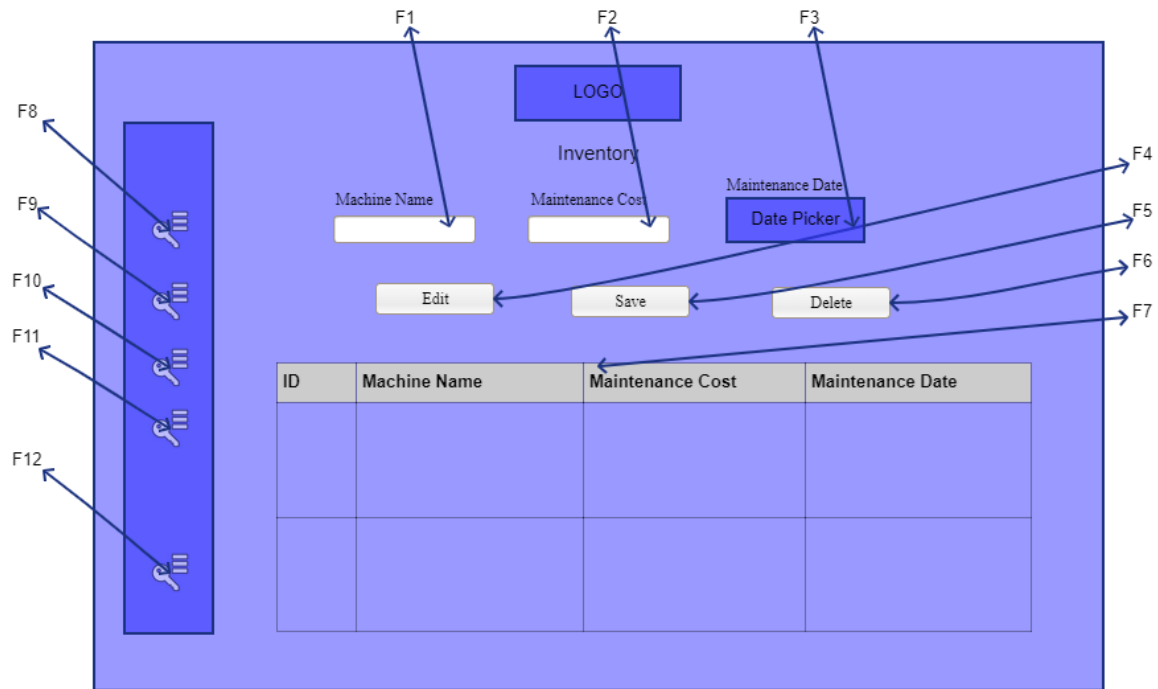
Add Tests:



Requirements	Admin
F1	As an admin, I shall be able to enter the name of the test.
F2	As an admin, I shall be able to enter the cost of the test.
F3	As an admin, I shall be able to edit the details of existing tests.
F4	As an admin, I shall be able to save the details of new test in the database.

F5	As a n admin, I shall be able to delete a test record from the database.
F6	As an admin, I shall be able to view all the existing tests in the database.
F7	As an admin, I shall jump to the Add laboratorian page when I click the icon.
F8	As an admin, I shall jump to the Add Tests page when I click the icon.
F9	As an admin, I shall jump to the Inventory page when I click the icon.
F10	As an admin, I shall jump to the Income Reports when I click the icon.
F11	As an admin, I shall logout of the system when I click the icon.

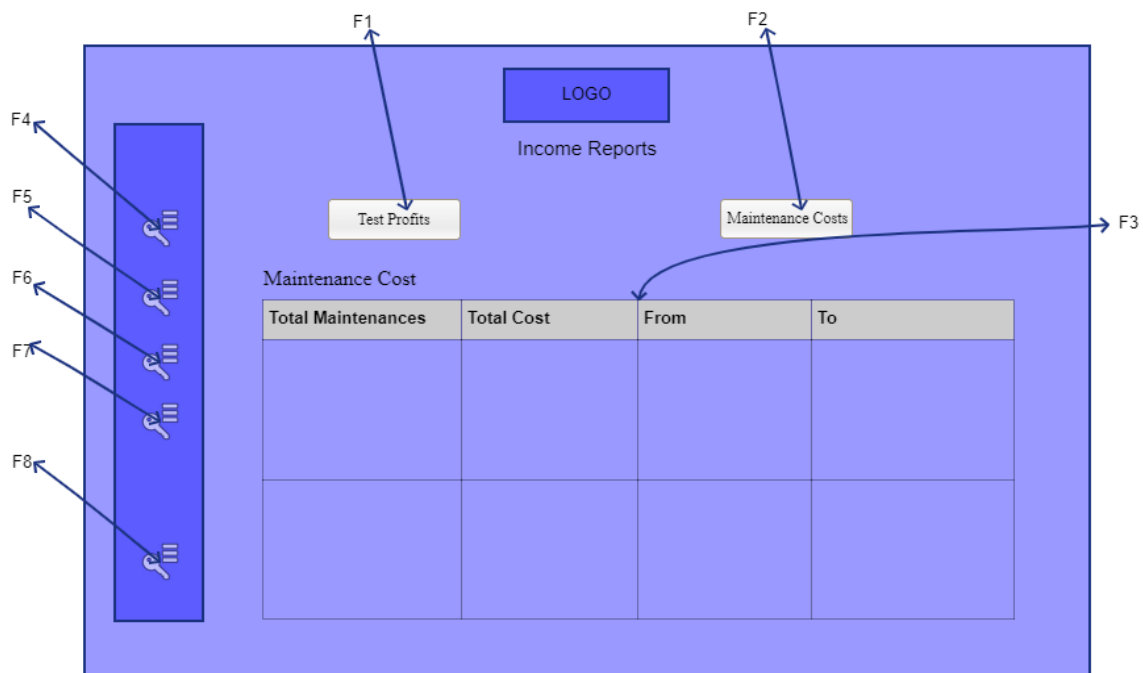
Inventory:



Requirements	Admin
F1	As an admin, I shall be able to enter the machine name.
F2	As an admin, I shall be able to enter the maintenance cost of machine.
F3	As an admin, I shall be able to select the data of maintenance of the machine.
F4	As an admin, I shall be able to edit the details of existing machine records by clicking the button.
F5	As an admin, I shall be able to save the details of machine maintenance by clicking the Save button.

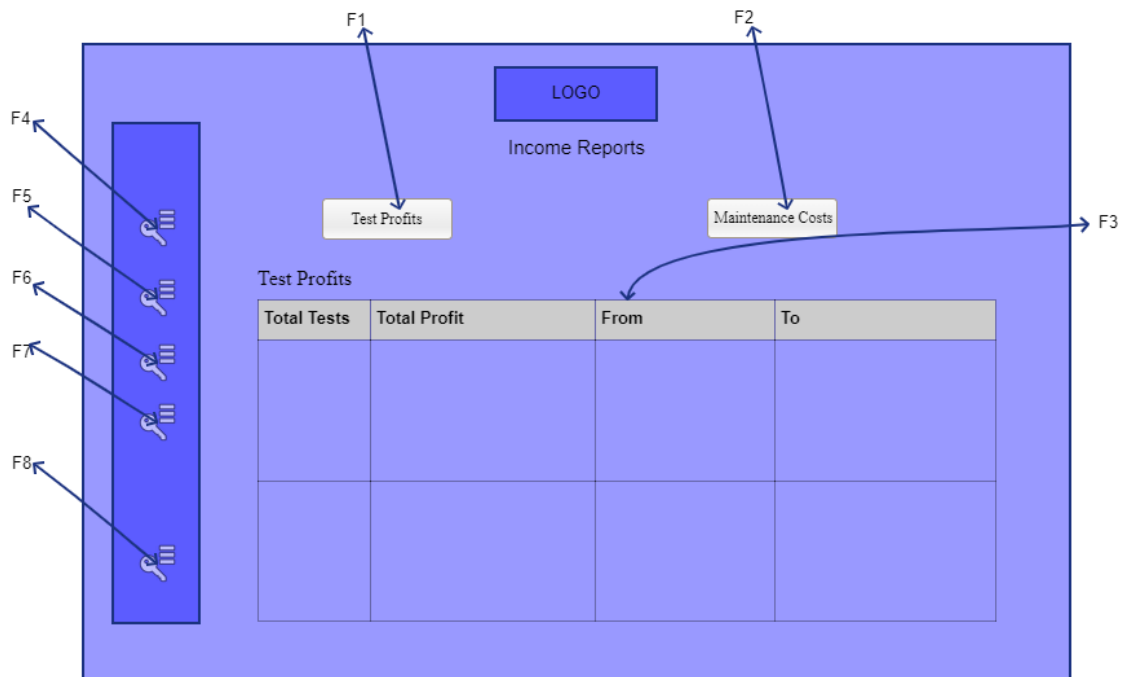
F6	As an admin, I shall be able to delete the record of a machine from database.
F7	As an admin, I shall be able to view the existing machine records.
F8	As an admin, I shall jump to the Add laboratorian page when I click the icon.
F9	As an admin, I shall jump to the Add Tests page when I click the icon.
F10	As an admin, I shall jump to the Inventory page when I click the icon.
F11	As an admin, I shall jump to the Income Reports when I click the icon.
F12	As an admin, I shall logout of the system when I click the icon.

Maintenance Reports:



Requirements	Admin
F1	As an admin, I shall be able to view the test profits and details.
F2	As an admin, I shall be able to view the maintenance cost and details.
F3	As an admin, I shall be able to view the records in the table.
F4	As an admin, I shall jump to the Add laboratorian page when I click the icon.
F5	As an admin, I shall jump to the Add Tests page when I click the icon.
F6	As an admin, I shall jump to the Inventory page when I click the icon.
F7	As an admin, I shall jump to the Income Reports when I click the icon.
F8	As an admin, I shall logout of the system when I click the icon.

Test Profits:



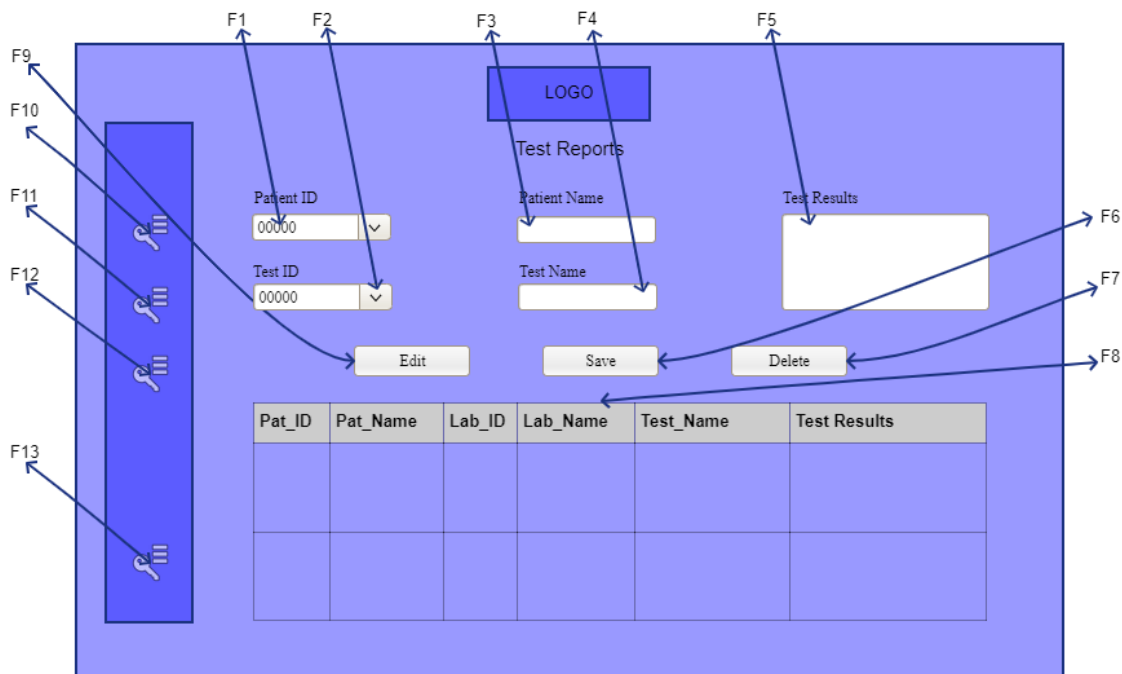
Requirements	User
F1	As an admin, I shall be able to view the test profits and details.
F2	As an admin, I shall be able to view the maintenance cost and details.
F3	As an admin, I shall be able to view the records in the table.
F4	As an admin, I shall jump to the Add laboratorian page when I click the icon.
F5	As an admin, I shall jump to the Add Tests page when I click the icon.
F6	As an admin, I shall jump to the Inventory page when I click the icon.
F7	As an admin, I shall jump to the Income Reports when I click the icon.
F8	As an admin, I shall logout of the system when I click the icon.

Add Patients:

Requirements	Laboratorian
F1	As a laboratorian, I shall be able to enter the name of the patient.
F2	As a laboratorian, I shall be able to select the date of birth of the patient.
F3	As a laboratorian, I shall be able to enter the phone number of the patient.
F4	As a laboratorian, I shall be able to pick the gender of the patient.
F5	As a laboratorian, I shall be able to enter the address of the patient.
F6	As a laboratorian, I shall be able to save the details of the patient to the database.
F7	As a laboratorian, I shall be able to delete a record of a patient from the database.
F8	As a laboratorian, I shall be able to view all the patients in the database.
F9	As a laboratorian, I shall be able to edit the record of a patient in the database.

F10	As a laboratorian, I shall jump to the Add Patient page.
F11	As a laboratorian, I shall jump to the Add Test Report page.
F12	As a laboratorian, I shall jump to the Statistical Report page.
F13	As a laboratorian, I shall logout of the system.

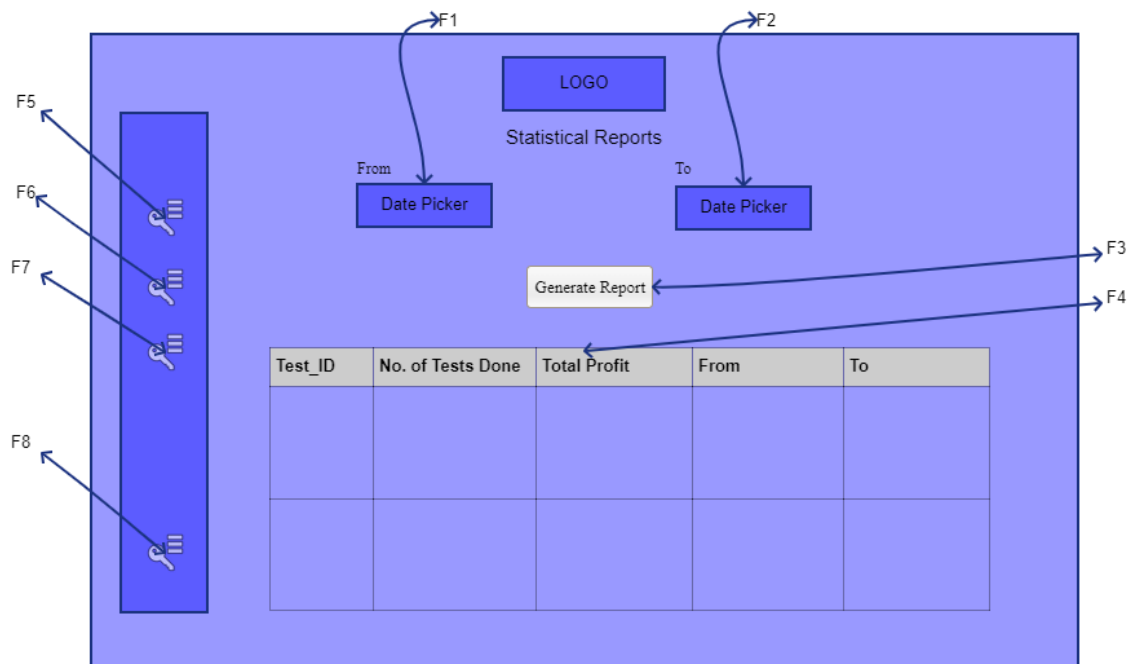
Test Reports:



Requirements	Laboratorian
F1	As a laboratorian, I shall be able to select the patient id.
F2	As a laboratorian, I shall be able to select the Test ID.
F3	As a laboratorian, I shall see the Patient name in the box.
F4	As a laboratorian, I shall see the test name in the box.
F5	As a laboratorian, I shall be able to add the Test results of the patient.

F6	As a laboratorian, I shall be able to save the details of the test.
F7	As a laboratorian, I shall be able to delete a record from the database.
F8	As a laboratorian, I shall be able to view the patient test records in table.
F9	As a laboratorian, I shall be able to edit a patient's test record in the database.
F10	As a laboratorian, I shall jump to the Add Patient page.
F11	As a laboratorian, I shall jump to the Add Test Report page.
F12	As a laboratorian, I shall jump to the Statistical Report page.
F13	As a laboratorian, I shall logout of the system.

Statistical Reports:



Requirements	Laboratorian
F1	As a laboratorian, I shall be able to select the start date of the report.
F2	As a laboratorian, I shall be able to select the end date of the report.
F3	As a laboratorian, I shall be able to generate the report of tests done and profit.
F4	As a laboratorian, I shall be able to view the data in the table.
F5	As a laboratorian, I shall jump to the Add Patient page.
F6	As a laboratorian, I shall jump to the Add Test Report page.
F7	As a laboratorian, I shall jump to the Statistical Report page.
F8	As a laboratorian, I shall logout of the system.

User Stories

1. As an Admin, I want to sign into the app so that I can manage lab operations effectively.
2. As an Admin, I want to add new tests to the system so that laboratorians can perform a wider range of analyses.
3. As an Admin, I want to manage laboratorian accounts so that I can control access and roles within the system.
4. As an Admin, I want to view machine inventory so that I can ensure all equipment is available and functional.
5. As an Admin, I want to access financial reports so that I can track the lab's income and expenses.
6. As a Laboratorian, I want to sign in to the app so that I can access patient records and manage tests.
7. As a Laboratorian, I want to add patient details so that I can keep accurate records of all patients.
8. As a Laboratorian, I want to record test results so that I can generate reports for patients and doctors.
9. As a Laboratorian, I want to generate patient reports so that I can provide clear and concise information about test outcomes.

Use Cases

Use Case 1	
Name	Sign-in
Actor	Admin, Laboratorian
Goal	The actor wants to successfully log in to their account.
Trigger	When the user clicks on the sign-in option from the navigation bar.
Pre-Condition	<ul style="list-style-type: none"> - User must have internet access. - User must have an account.
Post-Condition	<ul style="list-style-type: none"> - The actor successfully logs in. - User is redirected to the dashboard.
Basic Flow	<ol style="list-style-type: none"> 1. User opens the application. 2. User selects the sign-in option. 3. User enters their email and password. 4. User clicks the sign-in button.
Alternate Flow	<ol style="list-style-type: none"> 1. User opens the application. 2. User selects the sign-in option from another page. 3. User enters their email and password. 4. User clicks the sign-in button.
Exception	<ul style="list-style-type: none"> - User does not have an account. - User must sign up.
Qualities	<ul style="list-style-type: none"> - User can log in with minimal clicks. - User data is secure.

Use Case 2	
Name	Add New Tests
Actor	Admin
Goal	The admin wants to add new tests to the system.
Trigger	When the admin selects the option to add new tests from the dashboard.
Pre-Condition	<ul style="list-style-type: none"> - Admin must be logged in. - Admin must have permissions.
Post-Condition	The new test is successfully added.
Basic Flow	<ol style="list-style-type: none"> 1. Admin navigates to "Add New Test." 2. Admin enters required details. 3. Admin clicks "Save."
Alternate Flow	Admin reviews existing tests before adding a new one.

Exception	<ul style="list-style-type: none"> - Admin fails to enter required fields. - Admin lacks permission.
Qualities	<ul style="list-style-type: none"> - Admin can add tests quickly. - Input validation is in place.

Use Case 3	
Name	Manage Laboratorian Accounts
Actor	Admin
Goal	The admin wants to manage laboratorian accounts.
Trigger	When the admin selects the option to manage accounts from the dashboard.
Pre-Condition	<ul style="list-style-type: none"> - Admin must be logged in. - Admin must have permissions.
Post-Condition	The admin can view/edit/delete laboratorian accounts.
Basic Flow	<ol style="list-style-type: none"> 1. Admin navigates to "Add Laboratorian" 2. Admin selects an account. 3. Admin makes edits or deletes the account. 4. Admin confirms changes.
Alternate Flow	1. Admin adds a new laboratorian instead of editing an existing one.
Exception	<ul style="list-style-type: none"> - Admin lacks permission. - Account does not exist.
Qualities	<ul style="list-style-type: none"> - Admin can manage accounts efficiently. - Changes are logged.

Use Case 4	
Name	View Machine Inventory
Actor	Admin
Goal	The admin wants to view the inventory of lab machines.
Trigger	When the admin selects the "View Machine Inventory" option.
Pre-Condition	Admin must be logged in.
Post-Condition	The admin can view the status of all lab machines.
Basic Flow	<ol style="list-style-type: none"> 1. Admin navigates to "Inventory." 2. Admin reviews the list of machines. 3. Admin add a new machine. 4, Admin click on Save.
Alternate Flow	<ol style="list-style-type: none"> 1. Admin edits an existing machine. 2. Admin deletes and existing machine.
Exception	<ul style="list-style-type: none"> - Admin lacks access.

Qualities	<ul style="list-style-type: none"> - Information is clear and organized. - Admin identifies machines needing maintenance.
------------------	---

Use Case 5	
Name	Access Financial Reports
Actor	Admin
Goal	The admin wants to access financial reports.
Trigger	When the admin selects the option to view financial reports.
Pre-Condition	Admin must be logged in.
Post-Condition	The admin can view detailed financial reports.
Basic Flow	<ol style="list-style-type: none"> 1. Admin navigates to "Income Reports." 2. Admin selects the report. 3. Admin reviews the data.
Alternate Flow	1. Admin downloads the report for offline analysis.
Exception	- Admin lacks access.
Qualities	<ul style="list-style-type: none"> - Reports generated quickly and accurately. - Data is user-friendly.

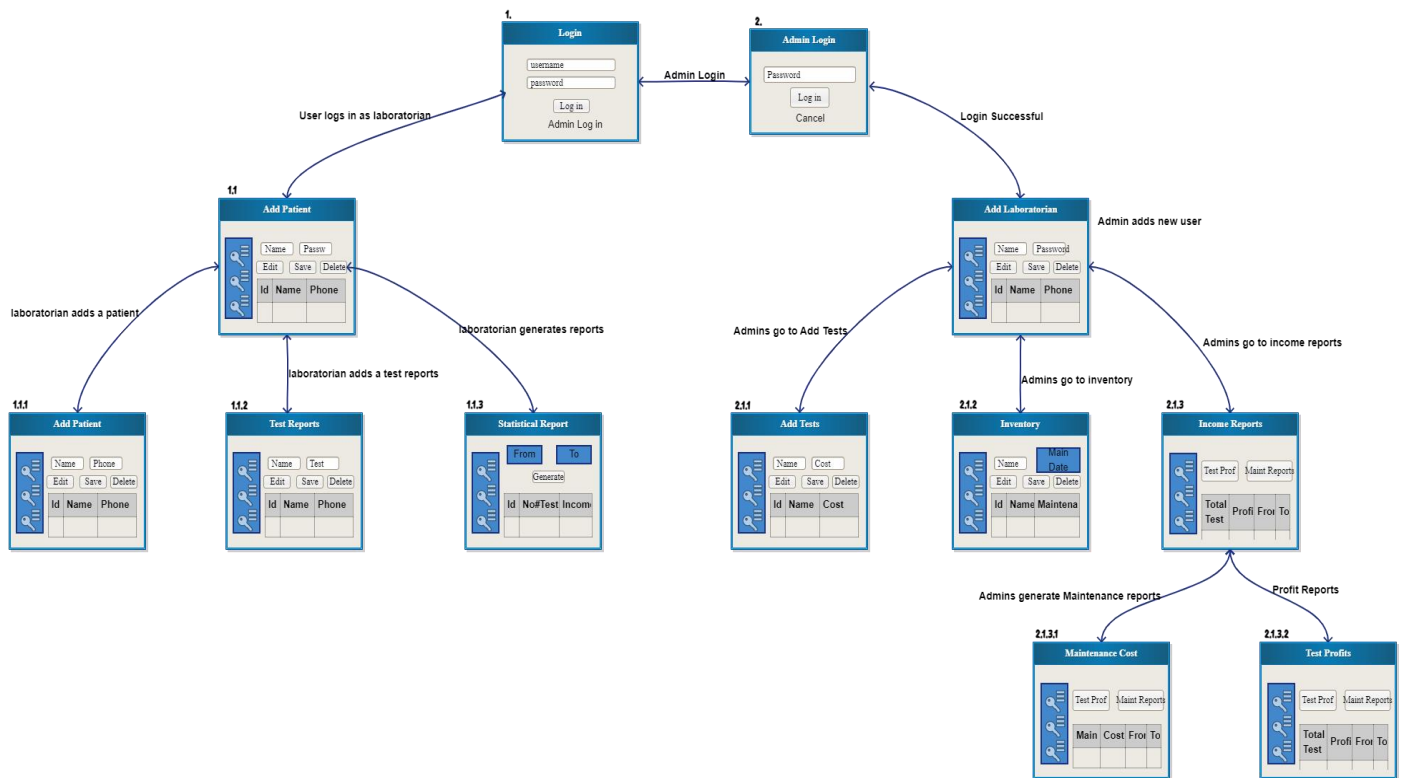
Use Case 6	
Name	Add Patient Details
Actor	Laboratorian
Goal	The laboratorian wants to add patient details.
Trigger	When the laboratorian selects the option to add a new patient.
Pre-Condition	Laboratorian must be logged in.
Post-Condition	Patient details are successfully saved.
Basic Flow	<ol style="list-style-type: none"> 1. Laboratorian navigates to "Add Patient." 2. Laboratorian fills in required information. 3. Laboratorian clicks "Save"
Alternate Flow	1. Laboratorian retrieves an existing record to update.
Exception	<ul style="list-style-type: none"> - Laboratorian fails to enter required fields. - Record exists.
Qualities	<ul style="list-style-type: none"> - Data is saved securely and accurately. - Laboratorian can add multiple patients easily.

Use Case 7	
Name	Record Test Results
Actor	Laboratorian
Goal	The laboratorian wants to record test results for a patient.
Trigger	When the laboratorian selects a patient from the records.
Pre-Condition	Laboratorian must be logged in.
Post-Condition	Test results are successfully recorded.
Basic Flow	<ol style="list-style-type: none"> 1. Laboratorian navigates to "Patient Records." 2. Laboratorian selects the patient. 3. Laboratorian enters the results. 4. Laboratorian clicks "Save."
Alternate Flow	1. Laboratorian reviews previous results before entering.
Exception	- Laboratorian fails to input required fields.
Qualities	<ul style="list-style-type: none"> - Results are saved promptly and accurately. - Access to previous results is easy.

Use Case 8	
Name	Generate Patient Reports
Actor	Laboratorian
Goal	The laboratorian wants to generate a report for a patient.
Trigger	When the laboratorian selects the option to create a report.
Pre-Condition	Laboratorian must be logged in. Patient must have test results.
Post-Condition	A report is successfully generated.
Basic Flow	<ol style="list-style-type: none"> 1. Laboratorian navigates to "Patient Reports." 2. Laboratorian selects the patient. 3. Laboratorian clicks "Generate Report." 4. Laboratorian reviews and prints/saves.
Alternate Flow	1. Laboratorian sends the report directly to the patient's email.
Exception	- Patient has no recorded test results.
Qualities	<ul style="list-style-type: none"> - Reports are generated accurately and timely. - Laboratorian can customize formats.

Use Case 9	
Name	Generate Statistical Reports
Actor	Laboratorian
Goal	The laboratorian wants to generate statistical reports based on selected date ranges.
Trigger	When the laboratorian selects the option to generate statistical reports.
Pre-Condition	Laboratorian must be logged in. There must be test data available for the selected date range.
Post-Condition	A statistical report is successfully generated for the selected date range.
Basic Flow	<ol style="list-style-type: none"> 1. Laboratorian navigates to "Statistical Reports." 2. Laboratorian selects a start date and an end date. 3. Laboratorian clicks "Generate Report." 4. The system compiles data and displays the report.
Alternate Flow	1. Laboratorian chooses to view summary statistics instead of detailed reports.
Exception	<ul style="list-style-type: none"> - No data available for the selected date range. - Laboratorian fails to select valid dates.
Qualities	<ul style="list-style-type: none"> - Reports are generated efficiently based on user-selected dates. - Data is presented clearly for analysis.

Story Board

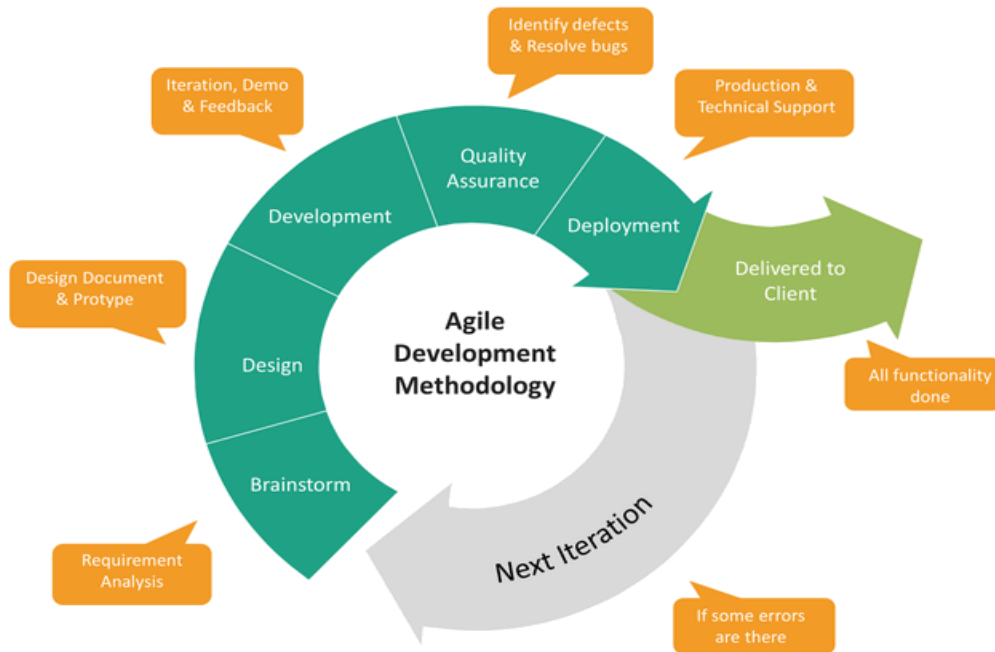


Proposed Methodology

- **System Architecture**
 - Utilize a client-server model.
 - Front-end application developed in C#.
 - Back-end data managed using SQL Server.
- **Development Methodology**
 - Adopt an Agile development approach for iterative progress and flexibility.
 - Facilitate regular feedback and adjustments based on user needs.

- **Tools and Technologies**
 - Use **C#** for application development.
 - Employ **SQL Server** for database management.
 - Consider additional front-end frameworks if applicable.
- **Database Design**
 - Design a database schema with key entities such as:
 - Users
 - Tests
 - Patients
 - Reports
 - Ensure robust relationships among these entities.
- **User Interface Design**
 - Focus on usability and accessibility for both Admin and Laboratorian roles.
 - Create intuitive navigation and clear layouts.
- **Testing and Quality Assurance**
 - Implement various testing strategies, including:
 - Unit testing
 - Integration testing
 - User acceptance testing (UAT)
 - Prioritize reliability and performance through thorough testing.
- **Deployment Strategy**
 - Plan for deployment that includes:
 - System setup
 - User training
 - Post-deployment support
- **Maintenance and Support**
 - Outline ongoing maintenance procedures post-launch.
 - Establish mechanisms for bug fixes and user feedback.
- **Project Timeline**
 - Create a high-level timeline with major development phases and expected completion dates.
- **Risk Management**
 - Identify potential risks (e.g., technical challenges, resource constraints).

- Develop strategies to mitigate these risks throughout the project lifecycle.



Timeline/Gantt Chart

Tasks	Week 01	Week 02	Week 03	Week 04
<i>Information Gathering</i>				
<i>Requirement Analysis</i>				
<i>System Design</i>				
<i>Prototyping</i>				