Medicine Management System (MMS)

Project Code: BSCS51F21R009

Project Manager: BSCS51F21R009

Submission Date: 06 - 02 - 2023

Version : 1.0

_____

Project Manager's Signature

Table of Contents

1. Introduction

1.1 Purpose of Document

The Medicine Management project is an online pharmacy. This document describes the functional and non-functional requirements for the C++ Medicine Management project. This document's goal is to provide a clear understanding of the project's scope and objectives.

The project, which will be a console-based program, will be implemented in C++ using OOP techniques and Data Structures.

1.2 Project Overview

The C++ Medicine Management project is a simulation of an online pharmacy. The user will be able to browse a list of medicines and purchase items based on their requirements. Inventory will be tracked and sales reports will be generated by the application.

1.3 Scope

- **Inventory Management**: A comprehensive database of all available medicines with updates to track stock levels and manage medicines.
- **Data Security**: A secure system to protect data and ensure the confidentiality of sensitive information.
- **User Management:** A system to manage user access and permissions to ensure only authorized personnel can access sensitive information.
- **Reporting and Analytics**: A reporting and analytics system to provide real-time insights into medicine sales.

2. Overall System Description

2.1 User characteristics

- The intended users of the MMS are individuals who want to manage and sell their medicines or wanna buy some medicines.
- Users should have basic knowledge of using a computer and running console based applications.
- Users should be familiar with shopping or managing.

2.2 Operating environment

- System should be on and running.
- The application will be a console-based program. System will have access to any console(bash recommended).
- System will have .exe supported system (windows 12 recommended).
- The application is written in C++ and will require a C++ compiler to be installed on the system if .exe file format not supported.
- System will not resist any file creation in any directory.
- System should support .dat files.
- The application will not have any specific hardware requirements.

2.3 System constraints

- The application will not have a graphical user interface; it will be a console-based program.
- The application is developed for concept representation of Data Structures , OOPs and Programming Fundamentals.
- The application will be written in C++ Language.
- The application will not have any integration with payment gateways.
- The application will not have any external dependencies or connections to external systems.

- The application will must have any security features such as user authentication.
- The application will have capability to add unlimited inventory items.
- The application will not have any reporting features other than sales reports.

3. External Interface Requirements

3.1 Hardware Interfaces

The physical connections and devices that allow communication between the MMS and the hardware components are referred to as hardware interfaces. This may include items such as a computer keyboard, monitor, and RAM. These interfaces enable software to control and interact with hardware by reading input and controlling output.

3.2 Software Interfaces

The ways in which different software components of the MMS interact with one another are referred to as software interfaces. APIs, libraries (iostream, string), and modules are examples of this. These interfaces enable various software components to communicate and share data with one another, allowing the project to function as a whole.

3.3 Communications Interfaces

This project does not include any communication interfaces, but for real-time projects like this, communication interfaces refer to how the MMS communicates with external devices or systems. Networking protocols, serial communication protocols, and communication protocols for specific types of devices are examples of this. These interfaces enable the project to communicate with other systems, such as sending and receiving data over a network.

4. Functional Requirements

Here are some functional requirements that may be relevant to MMS:

- The system should allow users to buy, search, and view medicine information.
- The system should allow admin to store, update, and delete medicines information.
- The system should allow admin to track sales and invoices.
- The system should allow administrative users to manage user accounts and settings.
- The system should generate invoice based on sale data.

5. Non-functional Requirements

- The system should provide a user-friendly interface.
- The system should use authentication to verify user identities.
- The system should use encryption to protect data at rest.
- The system should use access control to limit user access to sensitive data.
- The system should use data backup to protect data.
- The system should be able to handle large amounts of data.
- The system should have high performance and scalability.
- The system should be secure and reliable.
- The system should be easy to maintain and update.

5.1 Performance Requirements

The ability of the system to meet specific performance targets such as response time, throughput, and scalability is referred to as performance requirements. For example, a non-functional requirement in a MMS might be to respond to user input within a certain time frame or to handle a certain number of requests per second.

5.2 Safety Requirements

This Project have Authentication system. Make sure that no normal user can enter the admin area and prevent any user to make themselves admin.

Also no user can access to any functionality without login or signup except just viewing list of medicines

5.3 Security Requirements

The project should be designed to prevent software errors, which may include employing appropriate programming practices, thoroughly testing the software, and ensuring that it is free of malware or other malicious code.

5.4 User Documentation

An example of type of user documentation that might be included in MMS is:

User manual is a comprehensive document that provides detailed information on how to navigate through the project via Menu, including step-by-step instructions for common tasks, Info about Invalid Choices and functionality.

6. Assumptions and Dependencies

Examples of assumptions in a C++ MMS include:

- The project assumes that the user has a certain level of technical knowledge and experience.
- The project assumes the user's computer meets certain system requirements, such as a specific amount of RAM or an operating system.
- The project assumes that the user's computer has certain libraries installed and configured, such as iostream or f-stream.
- Dependencies in a C++ MMS may include: The project is dependent on a specific version of a C++ compiler or runtime environment.
- The project is dependent on libraries, such as one for saving data into .dat files and another for working with specific types of data.

7. System Architecture

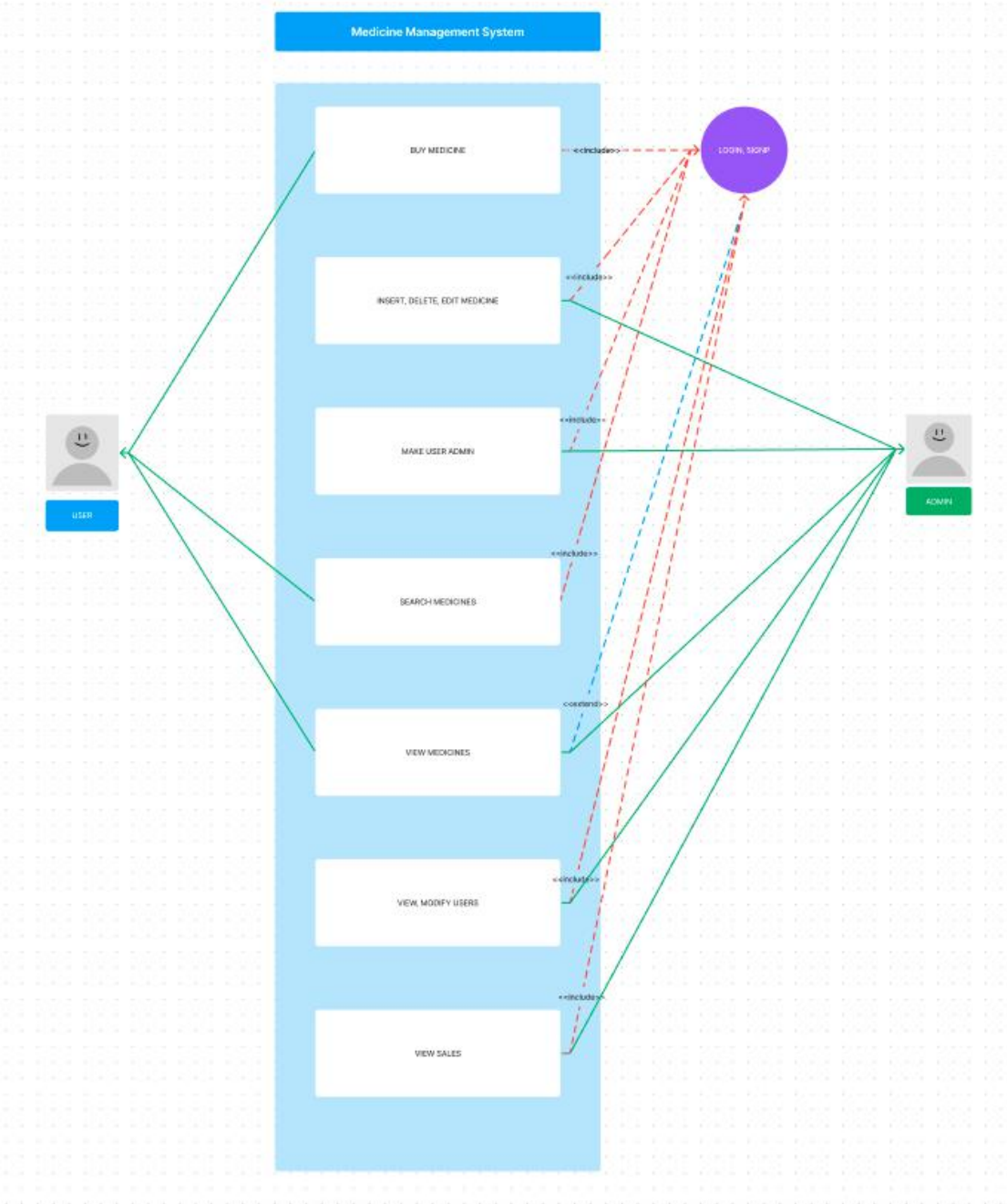
Well-designed system architecture can help ensure that the project is efficient, maintainable, and scalable.

Typical system architecture for a C++ MMS includes the following components:

- *The user interface component is in charge of displaying the project's features and functionality to the user via a console. A C++ compiler could be used to create the user interface.*
- *The data management component stores and retrieves data such as product information, customer information, and sales data. To store and retrieve data, we can use File Handling as a Data management component.*
- *Business logic: This component is in charge of implementing the project's core functionality, such as order processing and inventory management. C++ classes and functions could be used to implement the business logic component.*
- *External libraries: The project may use external libraries for specific functionality such as data storage, choice input, and menu display.*

8. Use Cases

8.1 Use Case Diagrams



8.2 Use Case Description

<1: Authentication>		
Actors: users , admin		
Feature:	Login/Sign up	
Use case Id:	1	
Pre-condition:	System should be running and have access to terminal. Also keyboard should be in working conditions.	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Select Login option from menu and enter username and password	if credential not present in users file then error message should be displayed. if its present it will check if isadmin flag is on or off and redirect user according to this flag either user panel or admin panel.
2.	Select Signup option from menu and enter name,age and set and confirm password	if password and confirm password matches and user doesn't exist already then sign up goes well and success message displayed and user is registered in record file if conditions not met i will show error message.
Post Conditions		
Step#	**Description**	
1	in Login user will be redirected to panel either admin panel or normal usage based on user condition.	
2	in Sign up user should not be redirected to panel ad reqired to login first with new credentials	
Use Case Cross referenced	<Related use cases, which use or are used by this use case>	
User Interface reference	List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements.	
Concurrency and Response **E**stimate of the following Number of concurrent users are 1000-2000 Expected response time of the use case is 0.002s		

<2: View Medicines>		
Actors: users , admin		
Feature:	View Medicines	
Use case Id:	2	
Pre-condition:	System should be running and have access to terminal. Also keyboard should be in working conditions.	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Select View Medicines option from menu	it will show list of all medicines with their prices using Medicines Object which is fetching data from file.
Post Conditions		
Step#	**Description**	
1	Medicines will show up if you press enter it will magically disappeared	
Use Case Cross referenced	<Related use cases, which use or are used by this use case>	
User Interface reference	List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements.	
Concurrency and Response **E**stimate of the following Number of concurrent users are 1000-2000 Expected response time of the use case is 0.002s		

<3: Search >		
Actors: users		
Feature:	Search Medicines	
Use case Id:	3	
Pre-condition:	System should be running and have access to terminal. Also keyboard should be in working conditions. User should be login in his account	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Select Search option from menu and enter name of medicine	if medicine not present in medicine file then error message should be displayed. if its present it will show that and ask if user wanna buy it or not.
Post Conditions		
Step#	**Description**	
1	after that user can choose either he wanna buy this medicine or not	
Use Case Cross referenced	<Related use cases, which use or are used by this use case>	
User Interface reference	List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements.	
Concurrency and Response **E**stimate of the following Number of concurrent users are 1000-2000 Expected response time of the use case is 0.002s		

<4: Buy Medicine>		
Actors: users		
Feature:	Buy Medicine	
Use case Id:	4	
Pre-condition:	System should be running and have access to terminal. Also keyboard should be in working conditions. user should be login in his account.	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Select Buy option from menu and enter medicine name	if medicine not present in medicine file then error message should be displayed. if its present it will ask for quantity customer want to buy if it is valid and medicine is not out of stock then it will create invoice for customer
Post Conditions		
Step#	**Description**	
1	after that system will add this sale to sales record with proper time	
Use Case Cross referenced	<Related use cases, which use or are used by this use case>	
User Interface reference	List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements.	
Concurrency and Response **E**stimate of the following Number of concurrent users are 1000-2000 Expected response time of the use case is 0.002s		

<5: View Sales>		
Actors: admin		
Feature:	View Sales	
Use case Id:	5	
Pre-condition:	System should be running and have access to terminal. Also keyboard should be in working conditions. admin should be login in his account.	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Select option from menu	it will fetch data from file and show all sales in list
Post Conditions		
Step#	**Description**	
Use Case Cross referenced	<Related use cases, which use or are used by this use case>	
User Interface reference	List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements.	
Concurrency and Response **E**stimate of the following Number of concurrent users are 1000-2000 Expected response time of the use case is 0.002s		

<6: Delete User>		
Actors: admin		
Feature:	Delete user	
Use case Id:	6	
Pre-condition:	System should be running and have access to terminal. Also keyboard should be in working conditions. admin should be login in his account.	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Select delete option from menu and enter user name and his password	if user not present in medicine file or password is wrong then error message should be displayed. if its present and password is correct it will delete user and display success message.
Post Conditions after that system will update file with new data		
Step#	**Description**	
1	after that user data removed from file permanently	
Use Case Cross referenced	<Related use cases, which use or are used by this use case>	
User Interface reference	List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements.	
Concurrency and Response **E**stimate of the following Number of concurrent users are 1000-2000 Expected response time of the use case is 0.002s		

<7: Show Users>		
Actors: admin		
Feature:	Show Users	
Use case Id:	7	
Pre-condition:	System should be running and have access to terminal. Also keyboard should be in working conditions. admin should be login in his account.	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Select Show option from admin menu	it will fetch data from file and show all users in list
Post Conditions		
Step#	**Description**	
Use Case Cross referenced	<Related use cases, which use or are used by this use case>	
User Interface reference	List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements.	
Concurrency and Response **E**stimate of the following Number of concurrent users are 1000-2000 Expected response time of the use case is 0.002s		

<8: Add Medicine>		
Actors: admin		
Feature:	Add Medicine	
Use case Id:	8	
Pre-condition:	System should be running and have access to terminal. Also keyboard should be in working conditions. admin should be login in his account.	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Select Add option from admin menu and enter name,quantity,price	if medicine already present in medicine file then error message should be displayed. if its not it will add medicine to file
Step#	**Post Conditions** after that system will update file with new data	
Use Case Cross referenced	<Related use cases, which use or are used by this use case>	
User Interface reference	List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements.	
Concurrency and Response **E**stimate of the following Number of concurrent users are 1000-2000 Expected response time of the use case is 0.002s		

<9: Delete Medicine>		
Actors: admin		
Feature:	Buy Medicine	
Use case Id:	9	
Pre-condition:	System should be running and have access to terminal. Also keyboard should be in working conditions. admin should be login in his account.	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Select Delete option from menu and enter medicine name	if medicine not present in medicine file then error message should be displayed. if its present it will delete it.
Step#	**Post Conditions**	
	after that system will update file with new data	
Use Case Cross referenced	<Related use cases, which use or are used by this use case>	
User Interface reference	List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements.	
Concurrency and Response **E**stimate of the following Number of concurrent users are 1000-2000 Expected response time of the use case is 0.002s		

<10: Update Price>		
Actors: admin		
Feature:	update price	
Use case Id:	10	
Pre-condition:	System should be running and have access to terminal. Also keyboard should be in working conditions. admin should be login in his account.	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Select option from admin menu and enter medicine name and new price	if medicine not present in medicine file then error message should be displayed. if its present it will update price with new price.
Post Conditions after that system will update file with new data		
Step#	**Description**	
1		
Use Case Cross referenced	<Related use cases, which use or are used by this use case>	
User Interface reference	List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements.	
Concurrency and Response **E**stimate of the following Number of concurrent users are 1000-2000 Expected response time of the use case is 0.002s		

<11: Update Quantity>		
Actors: admin		
Feature:	update quantity	
Use case Id:	11	
Pre-condition:	System should be running and have access to terminal. Also keyboard should be in working conditions. admin should be login in his account.	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Select option from admin menu and enter medicine name and new quantity	if medicine not present in medicine file then error message should be displayed. if its present it will update quantity with new given quantity.
Post Conditions after that system will update file with new data		
Step#	**Description**	
1		
Use Case Cross referenced	<Related use cases, which use or are used by this use case>	
User Interface reference	List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements.	
Concurrency and Response **E**stimate of the following Number of concurrent users are 1000-2000 Expected response time of the use case is 0.002s		

<12: Make User Admin>		
Actors: admin		
Feature:	Make user admin	
Use case Id:	12	
Pre-condition:	System should be running and have access to terminal. Also keyboard should be in working conditions. admin should be login in his account.	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Select option from admin menu and enter username and his password	if user not present in medicine file or password is wrong then error message should be displayed. if its present and password is correct it will make user admin and display success message.
Post Conditions		
Step#	**Post Conditions**	
1	after that system will update file with new data	
Use Case Cross referenced	<Related use cases, which use or are used by this use case>	
User Interface reference	List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements.	
Concurrency and Response Estimate of the following Number of concurrent users are 1000-2000 Expected response time of the use case is 0.002s		

9. Risk Analysis

There is some risk of failing this MMS system on front of client or evaluator. There is some risk of higher response time. There is also some risk of failing hardware, dependency issues or any external module issue. Or there is a chance of human error.

10. Cost Estimation Sheet

1.	**Software development cost**	100
2.	**Packaged software**	0
3.	**Hardware**	0
4.	**Network**	160
5.	**Client**	0
6.	**Misc.**	0

		Total cost = 260

11. References

C++ How to Program 8
C++ Robert Leford
FreeCodeCamp -Youtube
CodewithHarry -Youtube