Software Requirements Specification

for

Restaurant Management System

Version 1.2

Prepared by: MUHAMMAD UMAIS FAROOQ, ASAD JAHANGIR, AITZAZ-UL-HASSAN

Organization: Majeed Foods

Date: 08-11-2025

Table of Contents

Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Project Scope 1

Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 3

2.6 User Documentation 3

2.7 Assumptions and Dependencies 3

System Features (Functional Requirements) 3

3.1 Menu Management 3

3.2 Shift Management 4

3.3 Stock Management 5

3.4 Balance Mangement 6

External Interface Requirements 6

4.1 Hardware Interfaces 6

4.2 Software Interfaces 6

4.3 Communication Interfaces 7

Non-Functional Requirements 7

Analysis Model 7

Design Model 15

Appendix A — Glossary 19

Appendix B — References 19

Revision History 20

1. Introduction

1.1 Purpose

Purpose for this SRS is to understand the architecture of software and develop the software. Its intended audience is developers, testers, designers and managers.

1.2 Document Conventions

Functional requirements are labeled as FR-x and Non-functional requirements are labeled as NFR-x.

1.3 Intended Audience and Reading Suggestions

This document is intended for:

1. *Developers: To understand the and implement the software*

2. *Testers: To design and test the test cases for the software.*

3. *Managers: To check the progress of the software development.*

1.4 Project Scope

This software is for managing and automating the tasks at Majeed huts for saving the time of the managers and waiters and improve the services for the customers. Software will cover the issues related to stock, shifts and menus.

2. Overall Description

2.1 Product Perspective

This is standalone software with centralized database. Basically, its desktop application which will manage the shifts menu items stock and balance. It will use the Model View Controller architecture and provide interface to interact with the application

2.2 Product Functions

Software will cover the functionality of managing the stocks, tracking balance, updating menu items and managing shifts for waiters and managers.

2.3 User Classes and Characteristics

Managers: *Manager will manage menu items, balance checking, managing shits.*

Waiters: *Will check shifts and menu items.*

2.4 Operating Environment

Operating System: *Work on Windows (10,11).*

Programming Language: *Java (JDK 21) using Swing (latest version).*

Database: *SQL Server (8.4) for storing data.*

Network Requirements: *It can run offline with only database connection. Local network required for accessing if from multiple devices within the huts.*

2.5 Design and Implementation Constraints

1. *The software will be developed using Java (JDK 21) with Swing (latest version) for user interface.*

2. *The software architecture will base on MVC (Model–View–Controller) architecture*

2.6 User Documentation

User Manual provided for setting up the software and to use it.

2.7 Assumptions and Dependencies

1. *Huts have computers and terminals that supports the software.*

2. *Staff trained to use the software.*

3. System Features

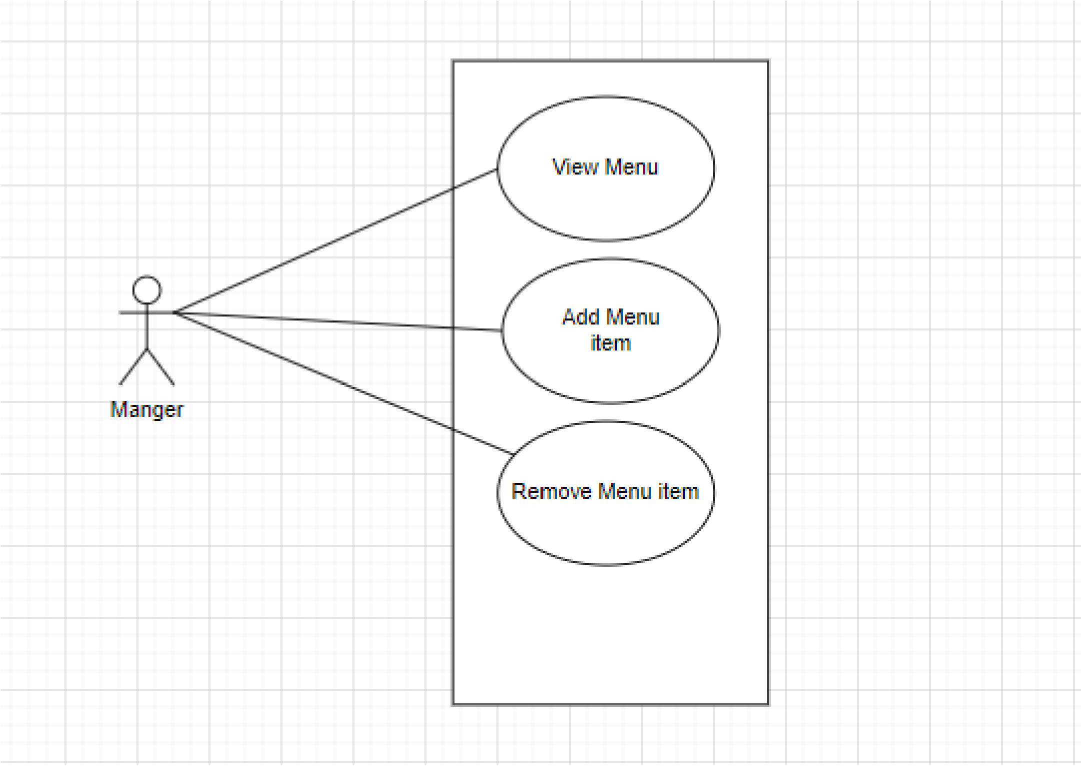
3.1 Menu Management:

Description:

This feature allow manger to remove and add menu items on daily basis. If the menu item is not available today, he will be able to remove it from the menu.

Functional Requirements:

FR	***Description***
FR-1	*Software will allow manager to view menu-items*
FR-2	*Software will allow manager to add menu-item*
FR-3	*Software will allow manager to remove menu-item*
FR-4	*Software will check if menu item exist it will not be added again*
FR-5	*If menu item didn’t exist it will tell manager that it doesn’t exist on try to remove from the menu*



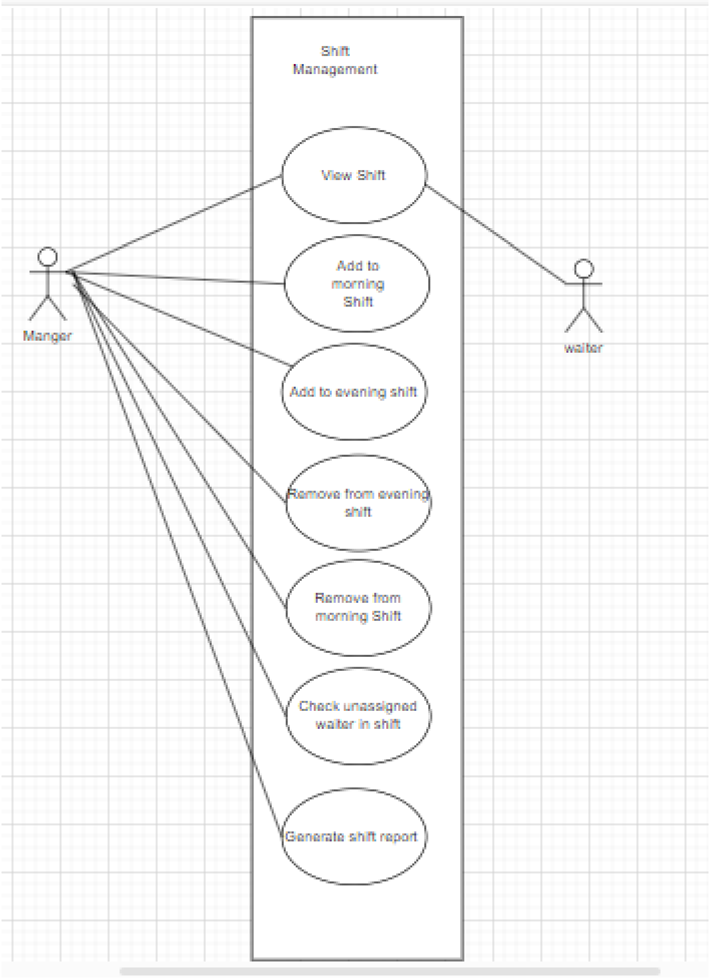
3.2 Shift Management:

Description:

This feature allow manger to update and manage the shift. Manager will be able to assign morning or evening shift to waiters. He will be able to make a proper schedule of the shifts using that feature.

Functional Requirements:

FR	***Description***
FR-1	*Software will allow manager and waiters to view shifts*
FR-2	*Software will allow manager to add waiters to shifts*
FR-3	*Software will allow manager to remove waiters from shifts*
FR-4	*Software will allow manager to add in morning or evening shifts*
FR-5	*Software will allow manager to check if any waiter has unassigned shift.*
FR-6	*Software will allow manager to generate a schedule of the shift*



3.3 Stock Management:

Description:

This feature allow manger to check the stock available for the menu items in the menu. Check the quantity of the stock's items. Check how much stock items are being consumed on daily basis.

Functional Requirements:

FR	***Description***
FR-1	*Software will allow manager to view stocks*
FR-2	*Software will allow manager to add items to the stock*
FR-3	*Software will allow manager to check which item is more used in menu*

FR-4	*Software will generate alert to manager if the item amount falls below specific category*

3.4 Balance Management:

Description:

This feature allow manger to check the account balance. Balance will be updated when items are purchased, and when sales values are entered manually. Orders are not part of the balance workflow.

Functional Requirements:

FR	***Description***
FR-1	*Software will allow manager to view account balance*
FR-2	*Software will allow manager to update balance if item is purchased*
FR-3	*Software will allow manager to update balance when manual sale amount is entered*
FR-4	*Software will show low balance alert*
FR-5	*Software will allow to check stats based on daily, weekly and monthly basis*
FR-6	*Software will allow to generate reports and track profit and loss*

4. External Interface Requirements

4.1 Hardware Interfaces

1.*Desktop.*

2. *Printer for reports generation*

3. *Local Network for multi device access.*

4.2 Software Interfaces

1.*Database: SQL Server.*

***2.**Language: Java SDK (21) support.*

4.3 Communications Interfaces

***1.**Local Network connection.*

5. Other Nonfunctional Requirements

NFR-1:

All data related to balance and stocks will be encrypted using AES (Advanced Encryption Standard).

NFR-2:

Software response time will be 2-3s in case of heavy load.

NFR-3:

Software will have consistent and clear look across all the screens.

NFR-4:

Software will perform its functionality without failing under heavy load for long period of time.

NFR-5:

Software will be divided into individual independent components to maintain modularity for system to easy to understand and update in future.

6. Analysis Model

Use Case UC1: Menu Management

Primary Actor: *Manager*

Stakeholders and Interests:

- *Manager: Wants to add menu item remove menu items without any duplication error of item.*
- *Waiters: Wants to view the updated menu to get orders according to it*

Preconditions: *Manger is identified and authenticated*

Postconditions: *Menu is updated and can be printed*

Main Success Scenario:

- *Manager checks the menu items list.*
- *Manager selects the add or remove menu item option.*
- *Now he can add the menu item name and details like price.*
- *If he selects the remove option then he can see the list and then remove item.*
- *System presents the updated menu.*
- *Manager leave after checking or printing the updated menu.*

Alternative Flow:

1. *Manager select the add option instead of removing item:*
    a. *System will provide the option to go back to previous discarding the steps.*
    b. *Same process with alternate scenario.*
2. *Database connection stops:*
    a. *Provide the error and load the cache data. Don't allow add or removal of menu item in this case.*

Use Case UC2: Shift Management

Primary Actor: *Manager*

Stakeholders and Interests:

- *Manager: Want to assign waiters shift (morning, evening).*
- *Waiters: Wants to view their shifts.*

Preconditions: *Manger is identified and authenticated*

Postconditions: *Shift updated and can be viewed or printed.*

Main Success Scenario:

- *Manager checks shift.*
- *Manager selects morning or evening shift.*
- *Now he can assign waiters to shift.*
- *He can remove waiter from any shift.*
- *System presents the updated shift.*
- *Manager leave after checking or printing the updated shift.*
- *Waiters can view their shift timings*
- *Manger can also check that if any waiter has unassigned shift*

Alternative Flow:

1. *Manager selects the wrong shift or waiter:*
    a. *System will provide the option to go back to previous discarding the steps.*
2. *Waiter is on leaver:*
    a. *Provide that waiter is not available. If a waiter is assigned morning and evening shift at same time system will let know the manager that waiter is unavailable. Remove first from the shift and then let the manager to add*

Use Case UC3: Stock Management

Primary Actor: *Manager*

Stakeholders and Interests:

- *Manager: Want to check if the stocks need to be purchased the number of stocks for menu item is available.*

Preconditions: *Manger is identified and authenticated*

Postconditions: *Stocks list updated successfully and menu can be managed on that basis.*

Main Success Scenario:

- *Manager checks stocks.*
- *Manager see if any stock is in low quantity.*
- *He can make purchase of item on that basis.*
- *When stocks are purchase, he can update the stocks quantity based on the purchase.*

Alternative Flow:

1. *Connection to database is interrupted:*
    a. *System will provide graceful error related to that and will terminate the process of adding the stocks.*

Use Case UC4: Balance Management

Primary Actor: *Manager*

Stakeholders and Interests:

- *Manager: Want to check if the current balance for purchases.*

Preconditions: *Manger is identified and authenticated*

Postconditions: *Balance updated successfully.*

Main Success Scenario:

- *Manager check the current balance.*

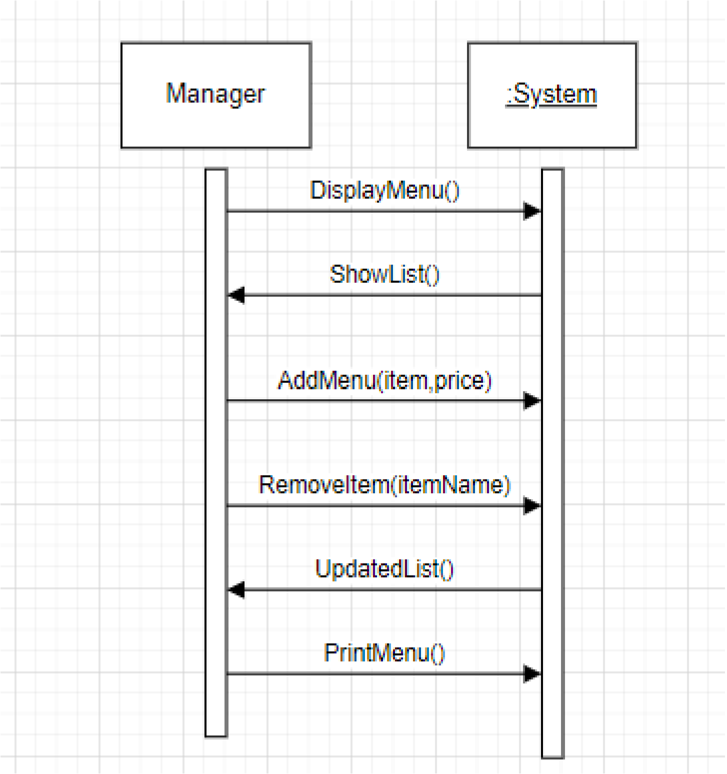
- *Manager add balance on basis of the sale.*
- *Update balance on basis of the purchase of the stocks.*
- *System will provide profit and loss reports based on the stats.*

Alternative Flow:

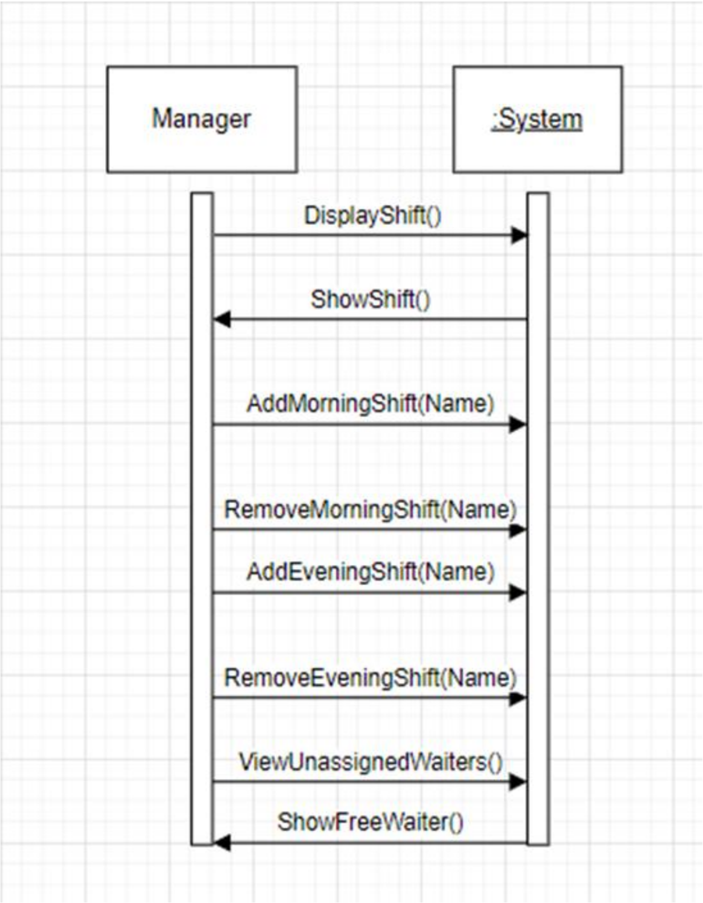
1. *Connection to database is interrupted:*
    a. *System will provide graceful error related to that and will terminate the process of balance updating.*
2. *If amount is entered in negative number or invalid strings it will show error and allow to reinput amount.*

System Sequence Diagrams:

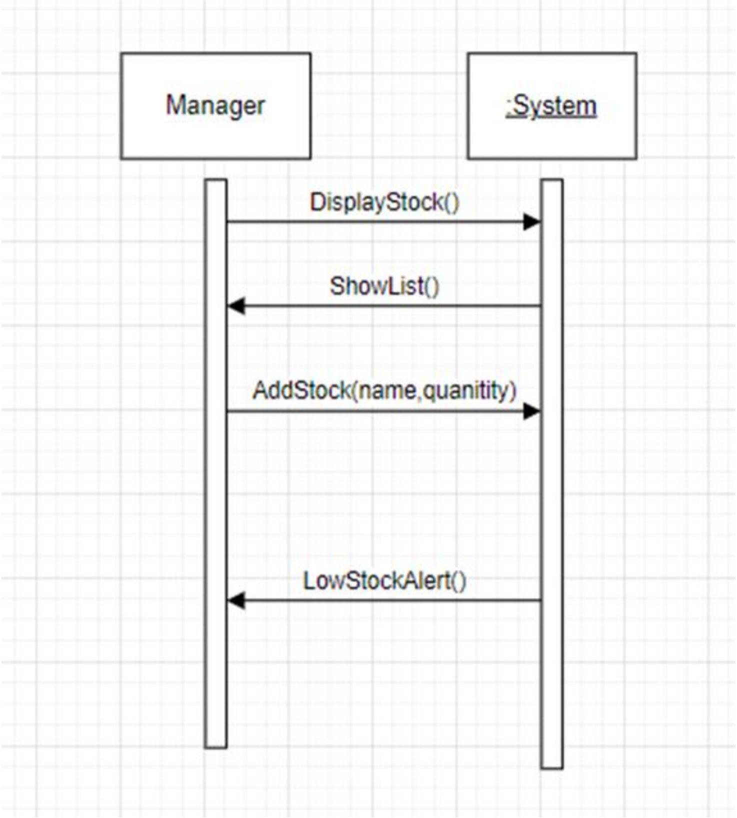
For Menu Management



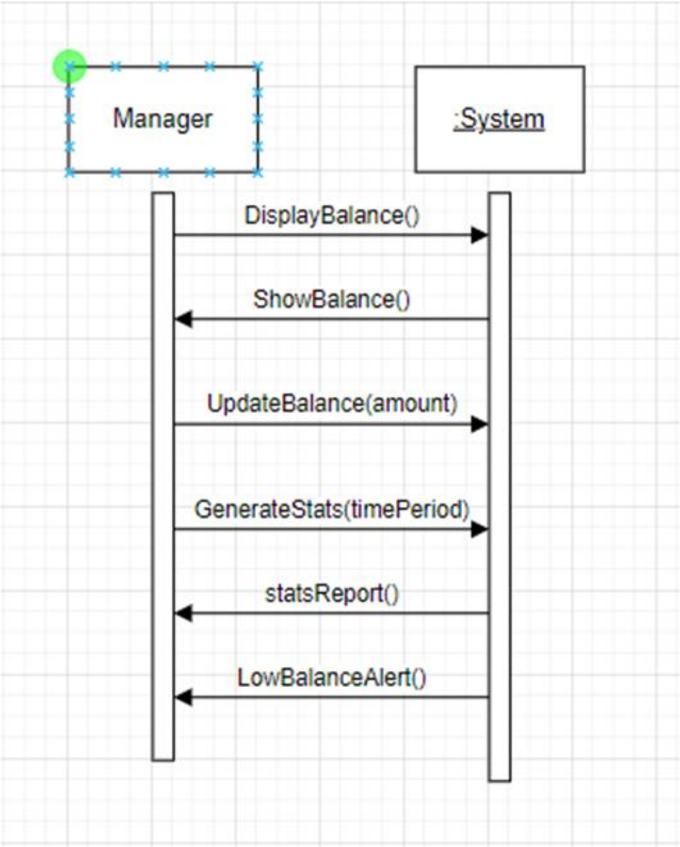
For Shift Management



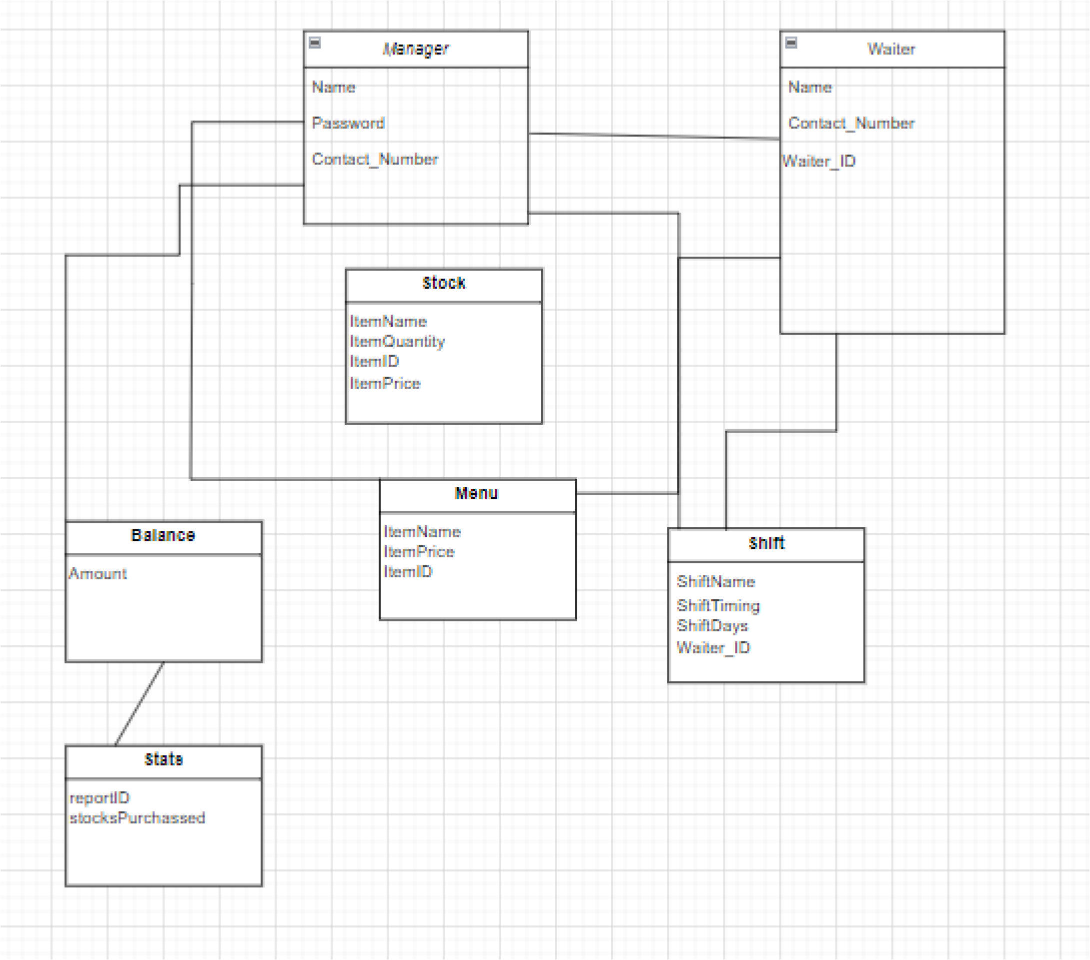
For Stock Management



For Balance Management

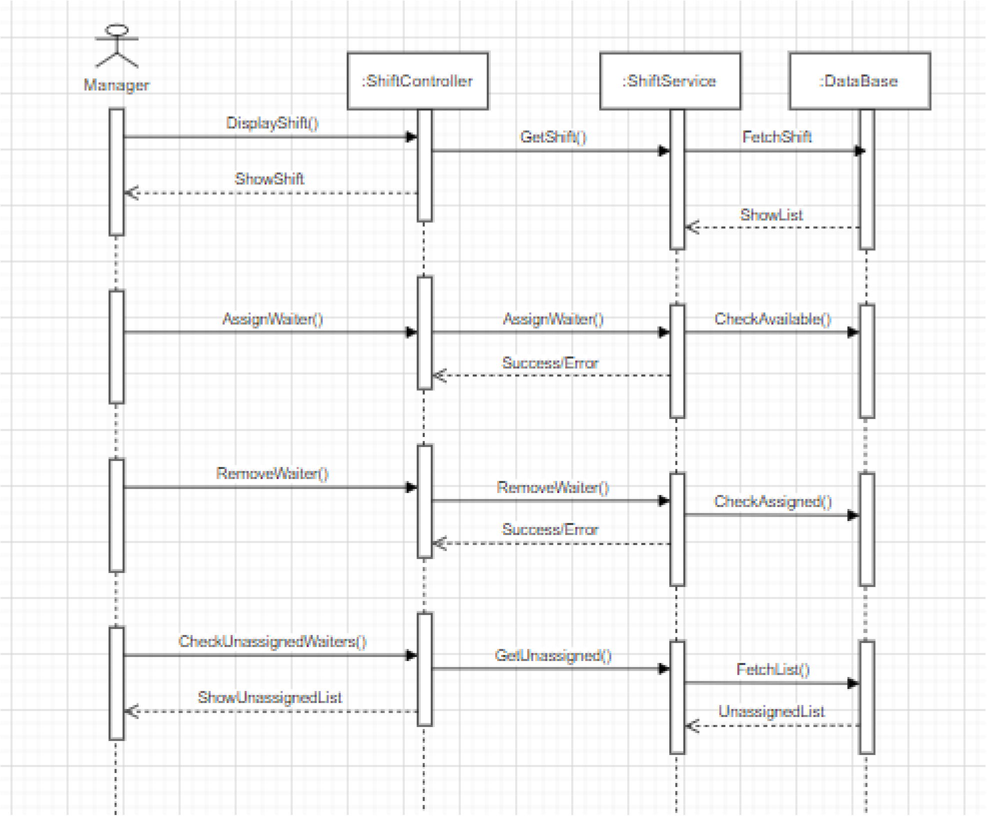


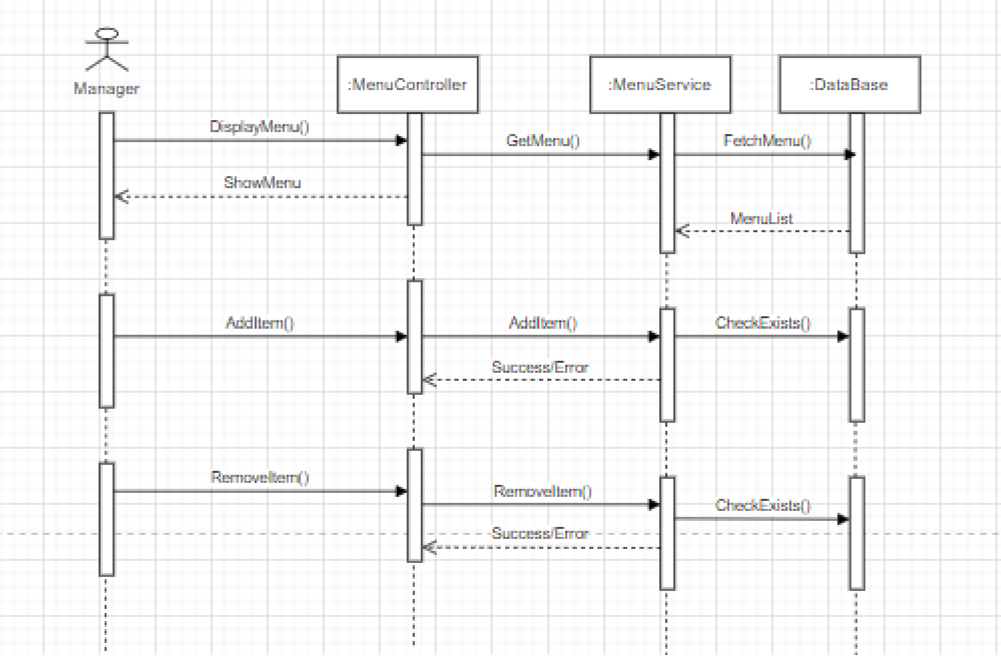
Domain Model:

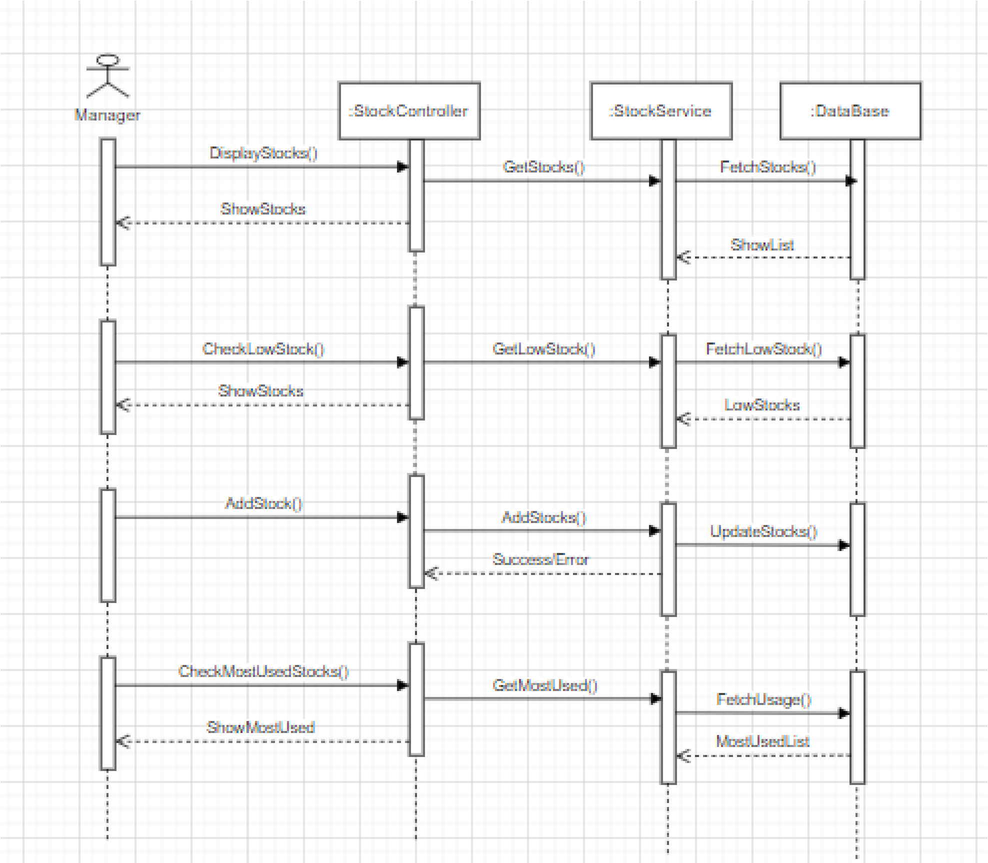


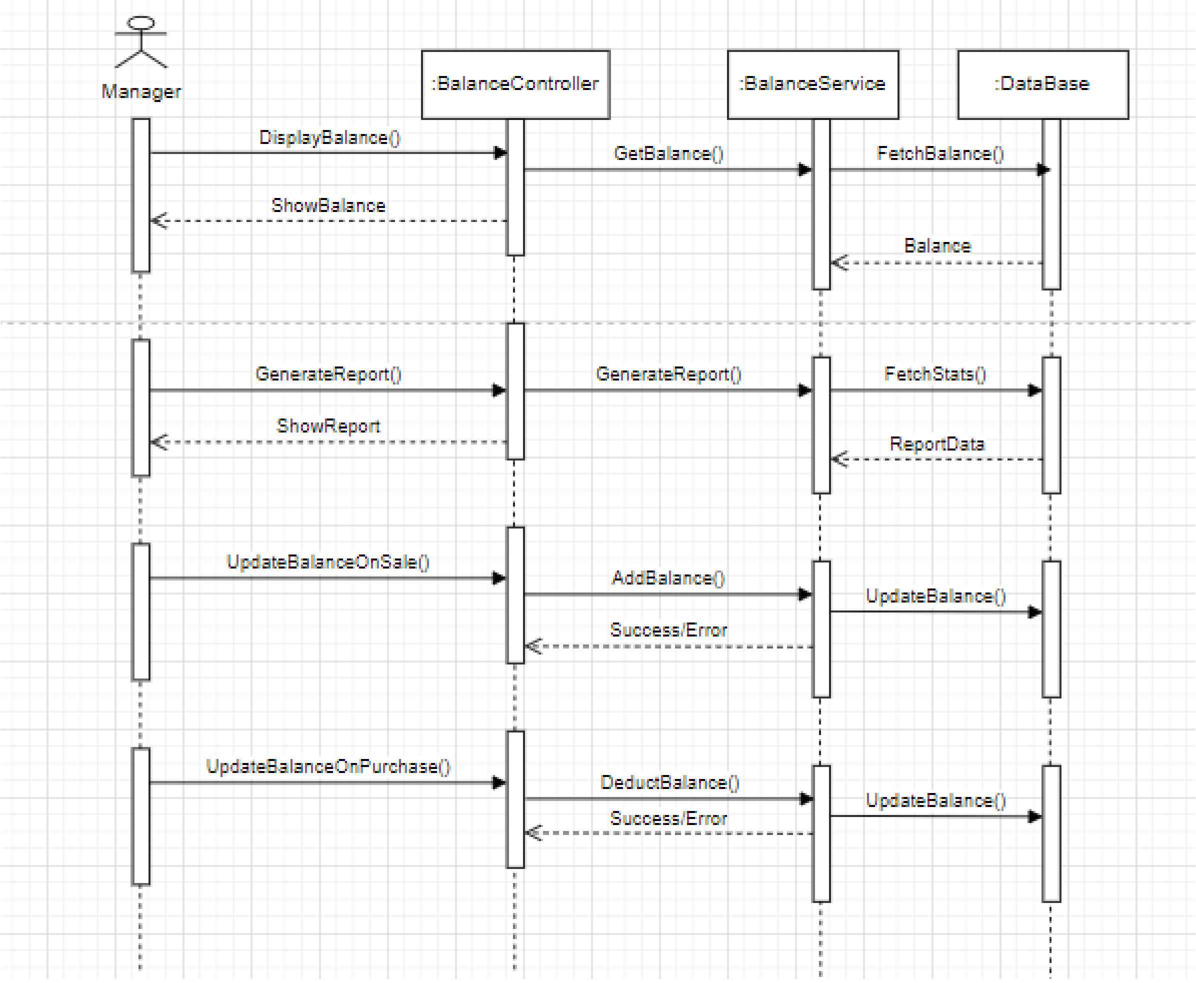
7. Design Model

Sequence Diagram:

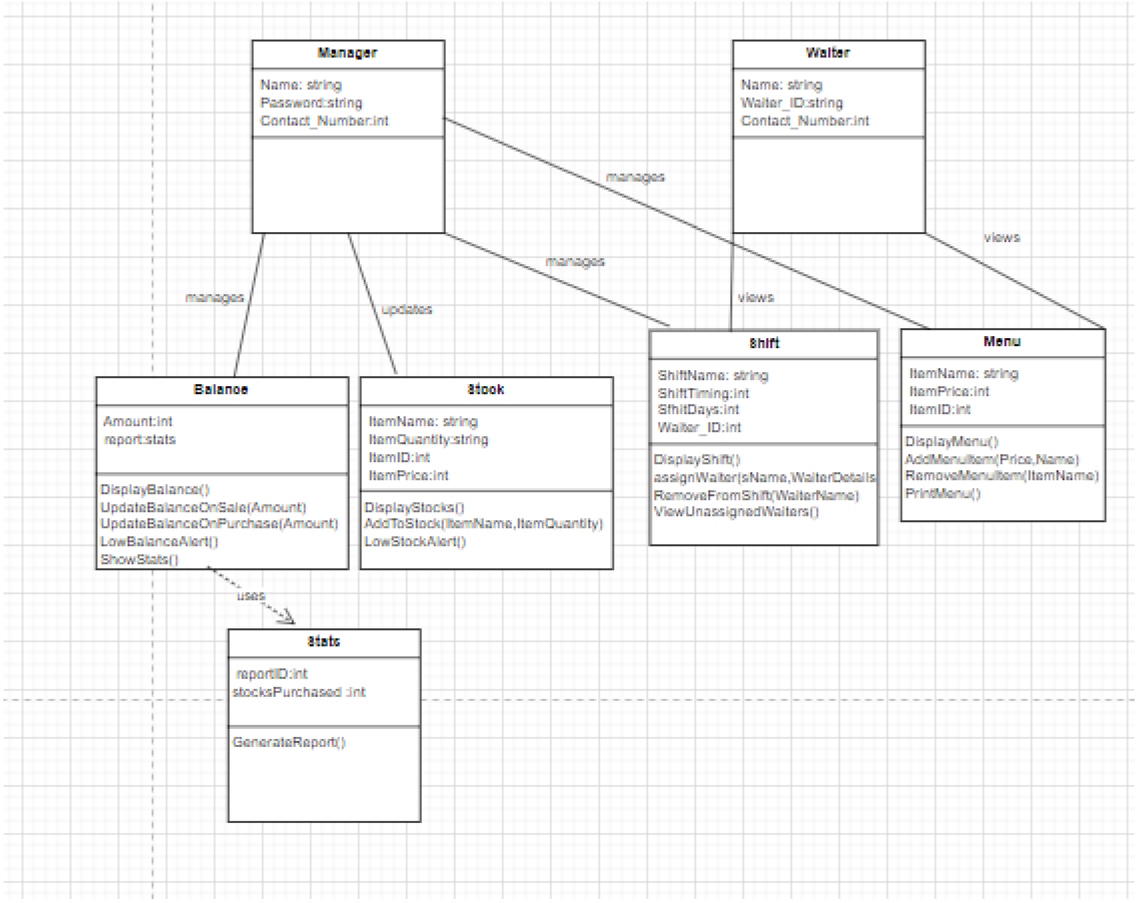

Manager
:ShiftController
:ShiftService
:DataBase
DisplayShift()
GetShift()
FetchShift
ShowShift
ShowList
AssignWaiter()
AssignWaiter()
CheckAvailable()
Success/Error
RemoveWaiter()
RemoveWaiter()
CheckAssigned()
Success/Error
CheckUnassignedWaiters()
GetUnassigned()
FetchList()
ShowUnassignedList
UnassignedList


Manager
:MenuController
:MenuService
:DataBase
DisplayMenu()
GetMenu()
FetchMenu()
ShowMenu
MenuList
AddItem()
AddItem()
CheckExists()
Success/Error
RemoveItem()
RemoveItem()
CheckExists()
Success/Error


Manager
:StockController
:StockService
:DataBase
DisplayStocks()
GetStocks()
FetchStocks()
ShowStocks
ShowList
CheckLowStock()
GetLowStock()
FetchLowStock()
ShowStocks
LowStocks
AddStock()
AddStocks()
UpdateStocks()
Success/Error
CheckMostUsedStocks()
GetMostUsed()
FetchUsage()
ShowMostUsed
MostUsedList


Manager
:BalanceController
:BalanceService
:DataBase
DisplayBalance()
GetBalance()
FetchBalance()
ShowBalance
Balance
GenerateReport()
GenerateReport()
FetchStats()
ShowReport
ReportData
UpdateBalanceOnSale()
AddBalance()
UpdateBalance()
Success/Error
UpdateBalanceOnPurchase()
DeductBalance()
UpdateBalance()
Success/Error

Design Class Diagram:



Appendix A: Glossary

FR: *Functional Requirement*

NFR: *Non-Functional Requirement*

MVC: *Model-View-Controller*

AES: *Advanced Encryption Standard*

Appendix B: Refernces:

- *IEEE Std 830-1998 for Software Requirements Specification document*
- *Software Engineering a Practitioner’s Approach (Roger S. Pressman, Ph.D.).*
- *Draw.io for use case diagrams*

Revision History:

Name	Date	Reason For Changes	Version
MUHAMMAD UMAIS FAROOQ ASAD JAHANGIR AITZAZ-UL-HASAN	30-10-2025	Initial Setup	1.0
MUHAMMAD UMAIS FAROOQ ASAD JAHANGIR AITZAZ-UL-HASAN	20-11-2025	Analysis and Design Model	1.1
MUHAMMAD UMAIS FAROOQ ASAD JAHANGIR AITZAZ-UL-HASAN	18-12-2025	Removed Order Dependency for simplifying scope	1.2