Software Requirements Specification

for

5.Restaurant Billing Management System

Version 1.0

Prepared by

Student Roll Number:	**B190011CS**	**Student Name:**	**Abin Jacob John**
Student Roll Number:	**B190368CS**	**Student Name:**	**Vishnu Ajay**
Student Roll Number:	**B191290CS**	**Student Name:**	**Tom Saju**

Project Owner: Akhila V H

Course: CS4097D Object Oriented Systems Laboratory

Date: November 07, 2022

Contents

1 Introduction **3**
1.1 Document Purpose 3
1.2 Product Scope 3
1.3 Definitions, Acronyms and Abbreviations 4
1.4 Document Conventions 4
1.5 References and Acknowledgments 4

2 Overall Description **5**
2.1 Product Overview 5
2.2 Product Functionality 6

3 Specific Requirements **7**
3.1 User Interfaces 7
3.1.1 Login Page 7
3.1.2 Main Menu 7
3.1.3 Item Management Page 8
3.1.4 Order Management Page 8
3.2 Functional Requirements 9
3.3 Use Case Model 10
3.2.1 Use Case #1 (Login) 11
3.2.2 Use Case #2 (Add Item) 11
3.2.3 Use Case #3 (Modify Item) 11
3.2.4 Use Case #4 (View Item) 11
3.2.5 Use Case #5 (Delete Item) 12
3.2.6 Use Case #6 (Add orders to cart) 12
3.2.7 Use Case #7 (View Bill) 12

1 Introduction

This SRS (Software Requirements Specification) document describes the requirements needed for the Restaurant Billing Management System. This SRS will help to understand the goals and objectives of the system. A clear understanding of the system will also help further stages in order to design and implement the system. This document aims to help users to know about the system and its features.

1.1 Document Purpose

This document is a Software Requirements Specification (SRS) for the Restaurant Billing Management System. It explains the functional, non-functional requirements. It includes a set of use cases that help users interact with the system and understand its features. The application aims to provide services like adding an item, modifying an item, viewing an item, deleting an item, adding orders to cart and viewing bills. The document is organized into several sections to help and assist the development of the system from the user perspective. It also describes non-functional requirements and other factors necessary to provide a complete and comprehensive description of the requirements for the software.

1.2 Product Scope

This Restaurant Billing Management system will be an application intended to manage the orders in a restaurant and provide billings. The purpose of the system is to minimize manual calculations among the employees at restaurants and save time. Also, the goal of this software system is to make the user experience comfortable and straightforward.

Following are the features provided by the system :

1. Manage the items in the restaurant.
2. Manage the orders from customers.
3. Calculate the bills accordingly.

1.3 Definitions, Acronyms, and Abbreviations

- **RBMS :** Restaurant Billing Management system will be an application intended to manage the orders in a restaurant and provide billings.

- **Login/ Authentication:** The system allows the admin to access the software using the login feature, which has a login id and password. When entered correctly, they'll get access to the application.

- **Use Case:** A use case is a methodology used in system analysis to identify, clarify and organize system requirements. The use case comprises a set of possible sequences of

interactions between systems and users in a particular environment and related to a specific goal.

- **Abbreviations:**
    a. App: Application
    b. DB: Database
    c. GUI: Graphical user interface
    d. H/D: Hardware
    e. RBMS : Restaurants Billing Management System
    f. SRS: Software requirement specifications
    g. S/W: Software

1.4 Document Conventions

This document follows the IEEE formatting requirements. Times New Roman font size 13, 14, and 18 have been used throughout the document for text. Used italics for comments. Use of bold for headings. The document text is single-spaced, and 1” margins are maintained in this document.

1.5 References and Acknowledgments

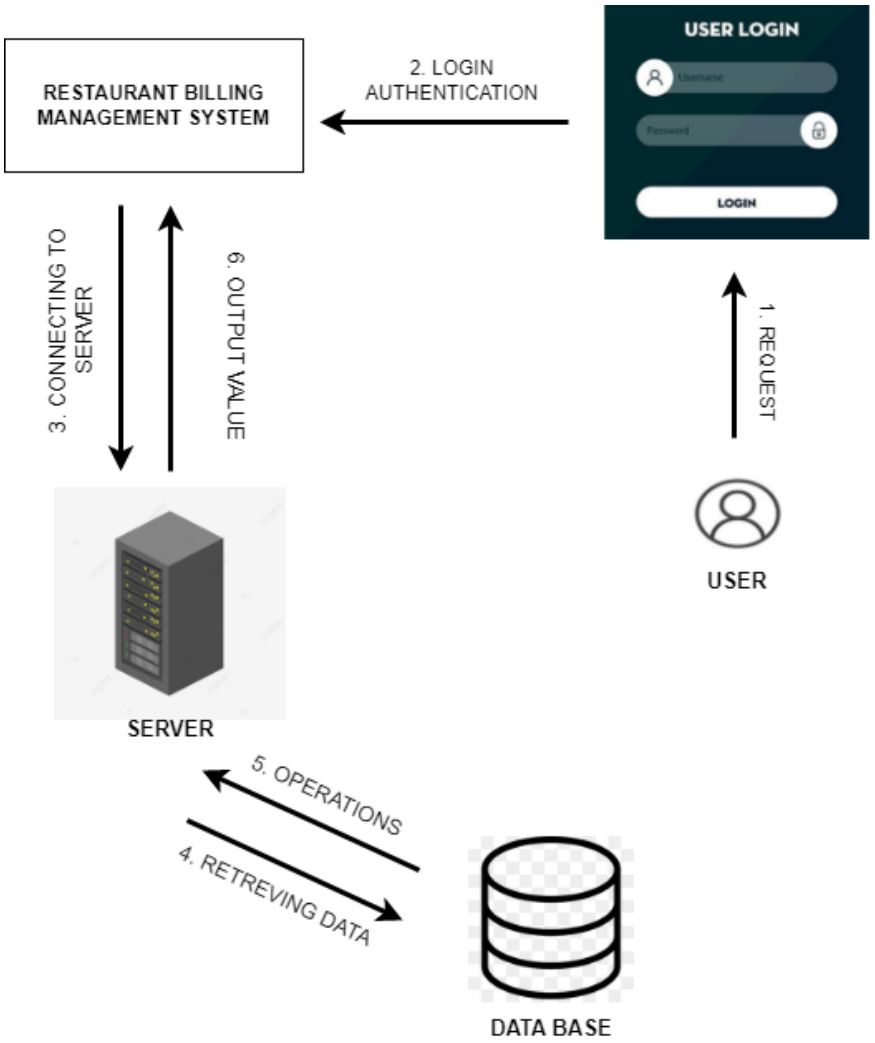
- IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998

- Object Oriented Modelling and Design with UML-Michael Blaha, James Rumbaugh.

2 Overall Description

2.1 Product Overview

This system will be an application with a simple GUI so that users need not have trouble accessing the application. This product serves users/ admins/ cashiers of restaurants to manage the number of items in the restaurant, so that they can add or delete items and modify them. This product also allows them to take orders from the available items and calculate bills. Today, we all are familiar with apps and online platforms, and we prefer to do things online rather than doing it on pen and paper. Calculating the bills for users can be done easily using this application. In other ways, we have to remember bills, expenses and do manual calculations, which is time consuming, difficult and now can be eliminated.



2.2 Product Functionality

Restaurant Billing Management System application allows

- **Login:** Login with userId and Password.
- **Add Item:** Allows users to add items in the restaurant which has the following details:
    a. Item Name.
    b. Item Price.
    c. Item Quantity.
- **Delete Item:** Allows users to delete an item based on the item name from the Inventory.
- **Modify Item:** Allows users to modify an item based on the item name from the Inventory.
- **View Items:** Allows users to view all the items that are present in the inventory.
- **Add Orders to Cart:** Allows users to add orders from customers to the cart.
- **Calculate Bills:** Calculates the bill of any particular order with discounts applied if any.
- **Simple GUI:** A simple GUI so that users can easily access and use the application.

3 Specific Requirements

3.1 User Interfaces

3.1.1 Login Page

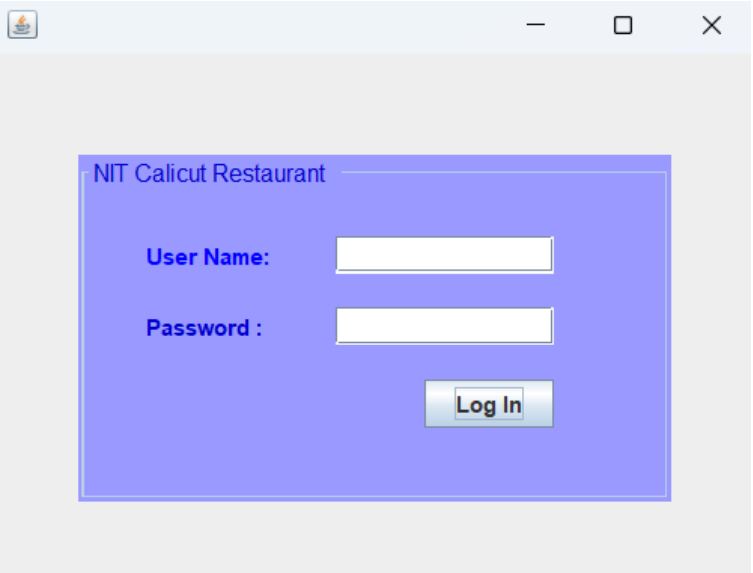


Figure 1: Login Page

3.1.2 Main Menu

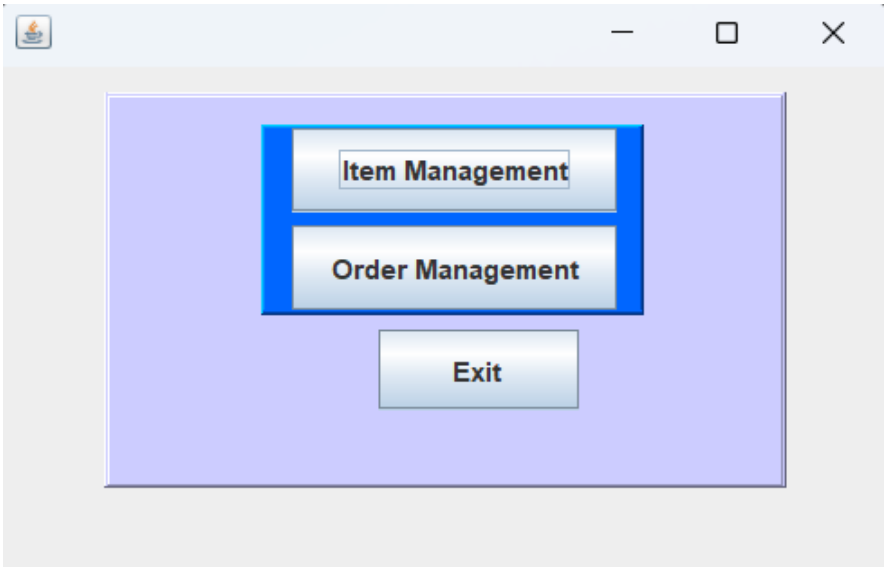


Figure 2: Main Menu

3.1.3 Item Management Page

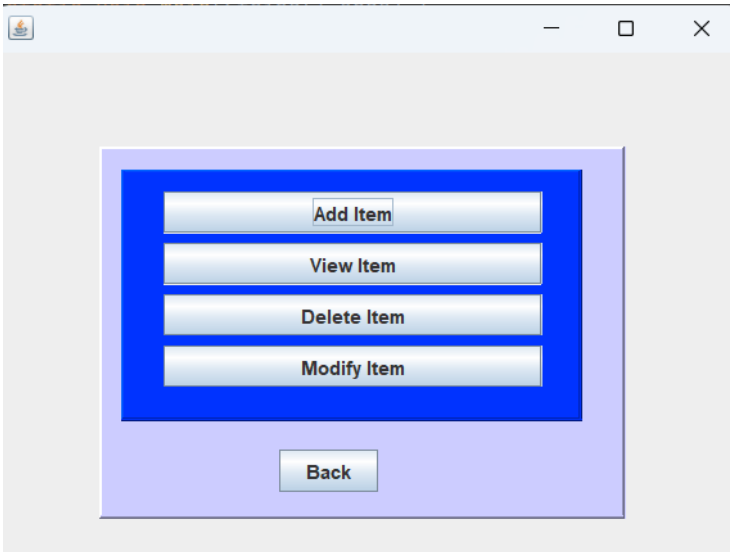


Figure 3: Item Management Page

3.1.4 Order Management

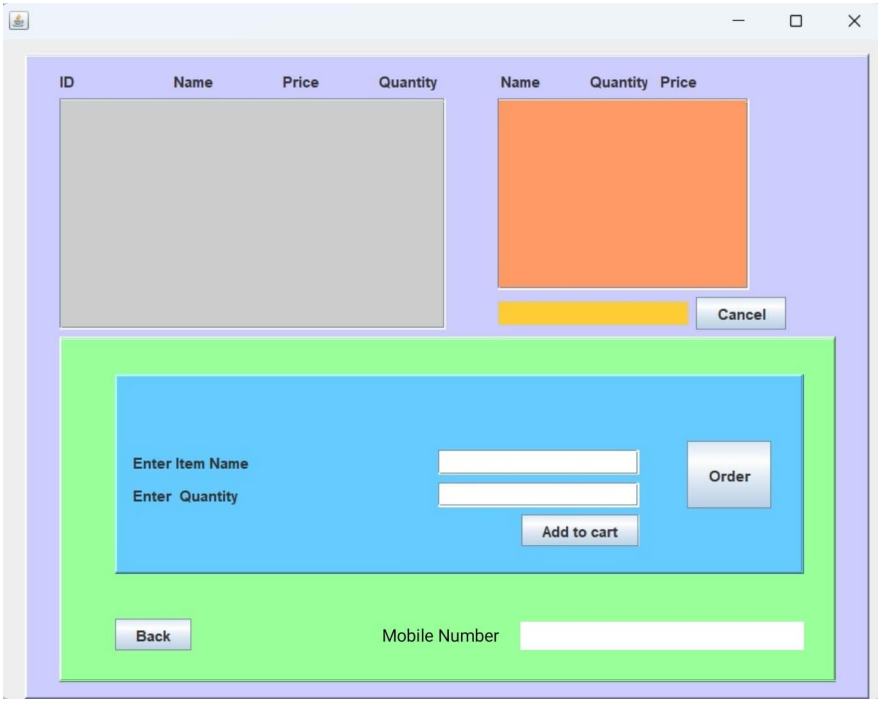
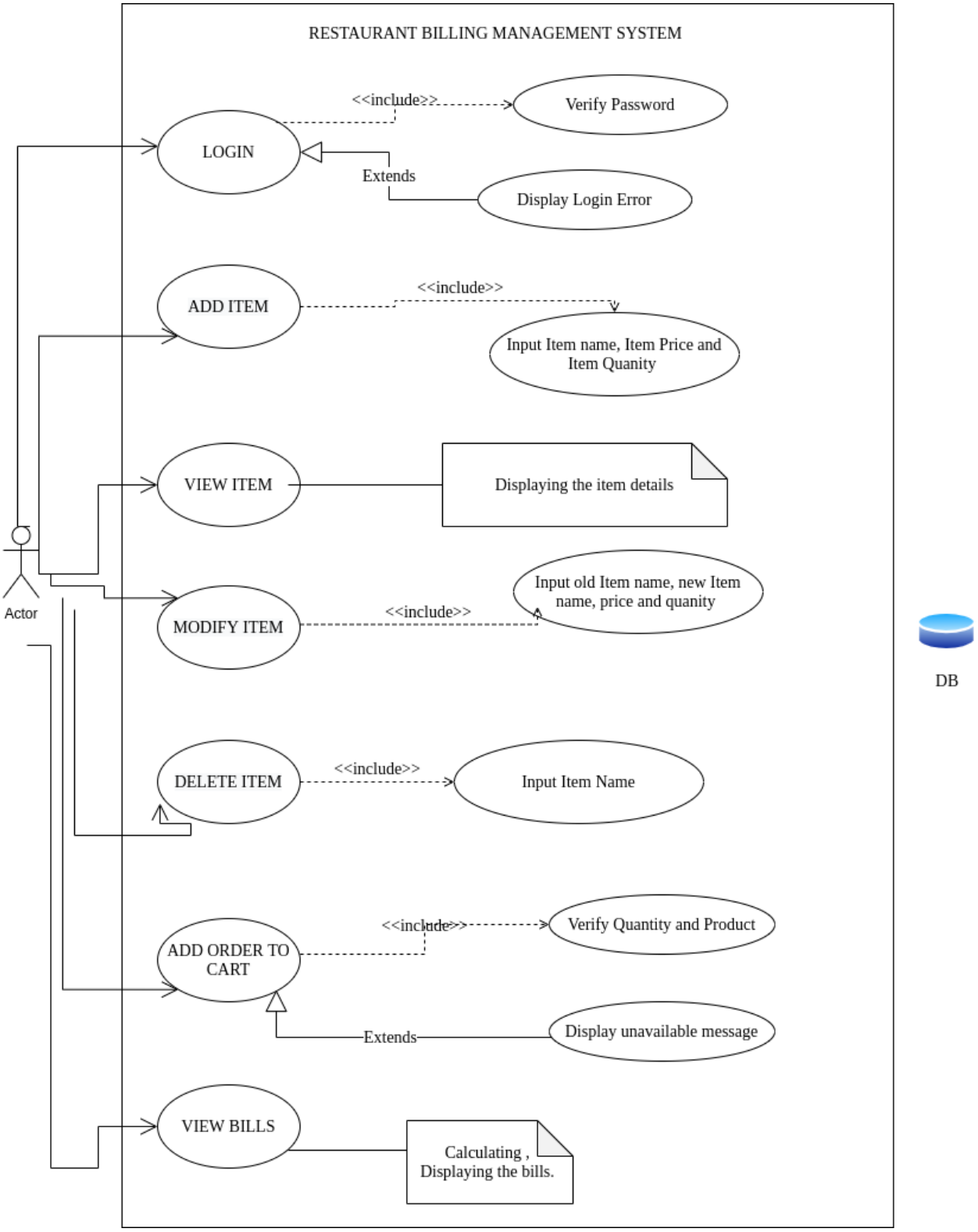


Figure 4: Order Management Page

3.2 Functional Requirements

F1: This system shall allow users to log in with their UserID and password.
F2: This system shall allow users to add Item Details such as name, cost, and quantity.
F3: This system shall allow users to modify the item details
F4: This system shall allow users to view the items and their details.
F5: This system shall allow users to delete the items.
F6: This system shall allow users to enter orders from customers based on their items in inventory.
F7: This system shall allow users to tell if the quantity ordered is available.
F8: This system shall check for any discounts applicable to the customer's order based on his number of visits.
F9: This system shall calculate the total bill of an order.

3.3 Use Case Model



3.3.1 Use Case #1 (Login – U1)

Purpose - To authenticate users to use the app.

Requirements Traceability – F1

Actors - User

3.3.2 Use Case #2 (Add Item – U2)

Purpose - To allow users to add the details of an item. The following details of an item can be added.

a. Item Name
b. Item Price
c. Item Quantity

Requirements Traceability – F2

Actors - User

3.3.3 Use Case #3 (Modify Item – U3)

Purpose - To allow users to modify the details of an item. It finds the item to modify by item name. The details: Name, Price, and quantity can be modified of the item that has been found.

Requirements Traceability – F3

Actors - User

3.3.4 Use Case #4 (View Item – U4)

Purpose - To allow users to view all the items along with their details.

Requirements Traceability – F4

Actors - User

3.3.5 Use Case #5 (Delete Item – U5)

Purpose - To allow users to delete items based on the item name.

Requirements Traceability – F5

Actors - User

3.3.6 Use Case #6 (Add orders to cart – U6)

Purpose - To allow users to add orders to the cart. Users can input the item name and the quantity that has been ordered by the customer.

Requirements Traceability – F6, F7, F9

Actors - User

3.3.7 Use Case #7 (View Bill – U7)

Purpose - To provide the bill of the orders with discounts applied (if any) that have been added to the cart from User Case 6 if it is valid.

Requirements Traceability – F6, F7, F8, F9

Actors - User

<Leave this page blank>

Design Document
for
Restaurant Billing Management System

Version 1.0

Prepared by Team 5:
(Based on SRS Version 1.0 prepared by Team 5)

Tom Saju	**B191290CS**	**tom_b191290cs@nitc.ac.in**
Abin Jacob John	**B190011CS**	**abin_b190011cs@nitc.ac.in**
Vishnu Ajay	**B190368CS**	**vishnu_b190368cs@nitc.ac.in**

Project Owner: **Akhila V H**

Course: **CS4097D Object Oriented Systems Laboratory**

Date: **November 14, 2022**

Glossary

RBMS	Restaurant Billing Management System
UML	Unified Modelling Language
DD	Design Document
DB	Data Base

Table of contents

Glossary **15**
Table of contents **16**

1. Detailed Design through UML diagrams **17**
1.1 System model using Class Diagram 17
1.1.1 Class Diagram 17
1.2 Responsibilities - Use Case Diagram 18
1.3 Static snapshot of the system - Object Diagram 19
1.4 System Interactions through Sequence Diagrams 20
1.4.1 Login 20
1.4.2 Add items into the inventory 21
1.4.3 View items in the inventory 22
1.4.4 Modify items in the inventory 23
1.4.5 Delete items in the inventory 24
1.4.5 Order items 25

2. Database Design **26**
2.1 ER Diagram 26

3. Implementation Plans **27**
3.1 Technology Stack 27
3.2 User Interface Prototyping 27

4. Test Cases **29**
4.1 Test Case #1 (TC_Login_001) 29
4.2 Test Case #2 (TC_Login_002) 30
4.3 Test Case #3 (TC_AddItem_001) 31
4.4 Test Case #4 (TC_AddItem_002) 32
4.5 Test Case #5 (TC_ViewItem_001) 32
4.6 Test Case #6 (TC_DeleteItem_001) 33
4.7 Test Case #7 (TC_DeleteItem_002) 34
4.8 Test Case #8 (TC_ModifyItem_001) 35
4.9 TestCase #9 (TC_ModifyItem_002) 36
4.10 Test Case #10 (TC_OrderManagement_001) 37

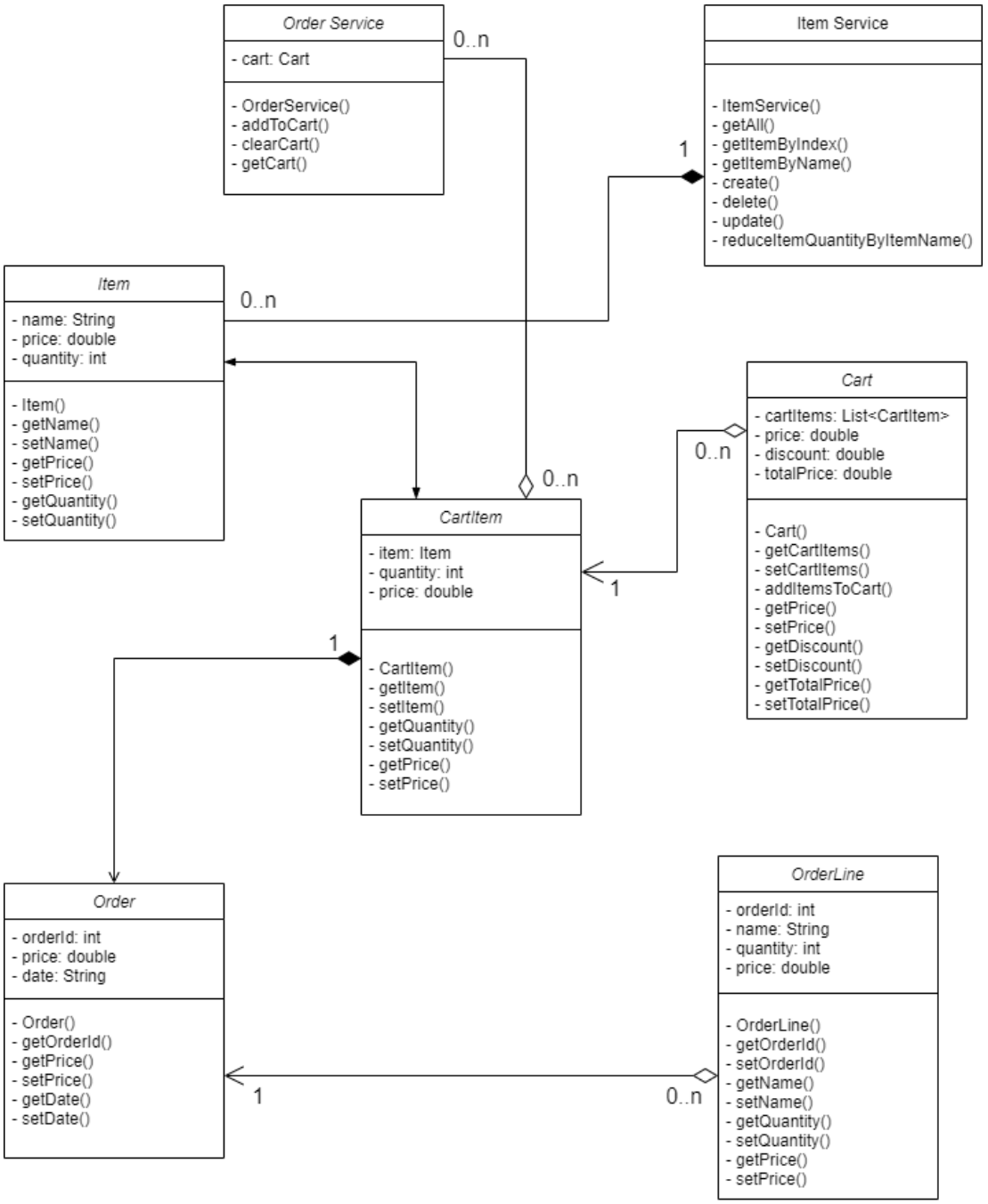
References **38**

1. Detailed Design through UML diagrams

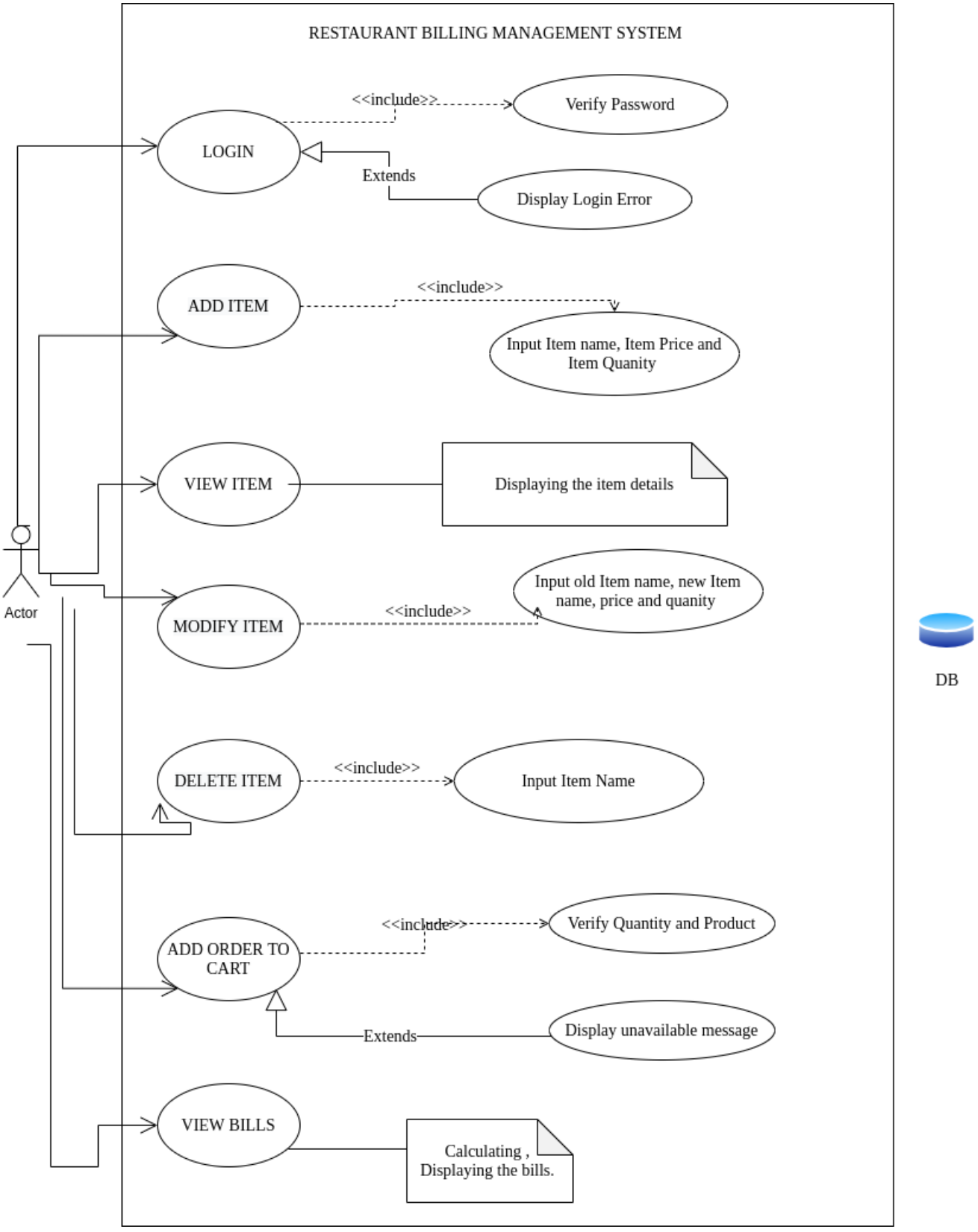
1.1 System model using Class Diagram

Class Diagram in the Unified Modelling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods) and the relationships among classes.

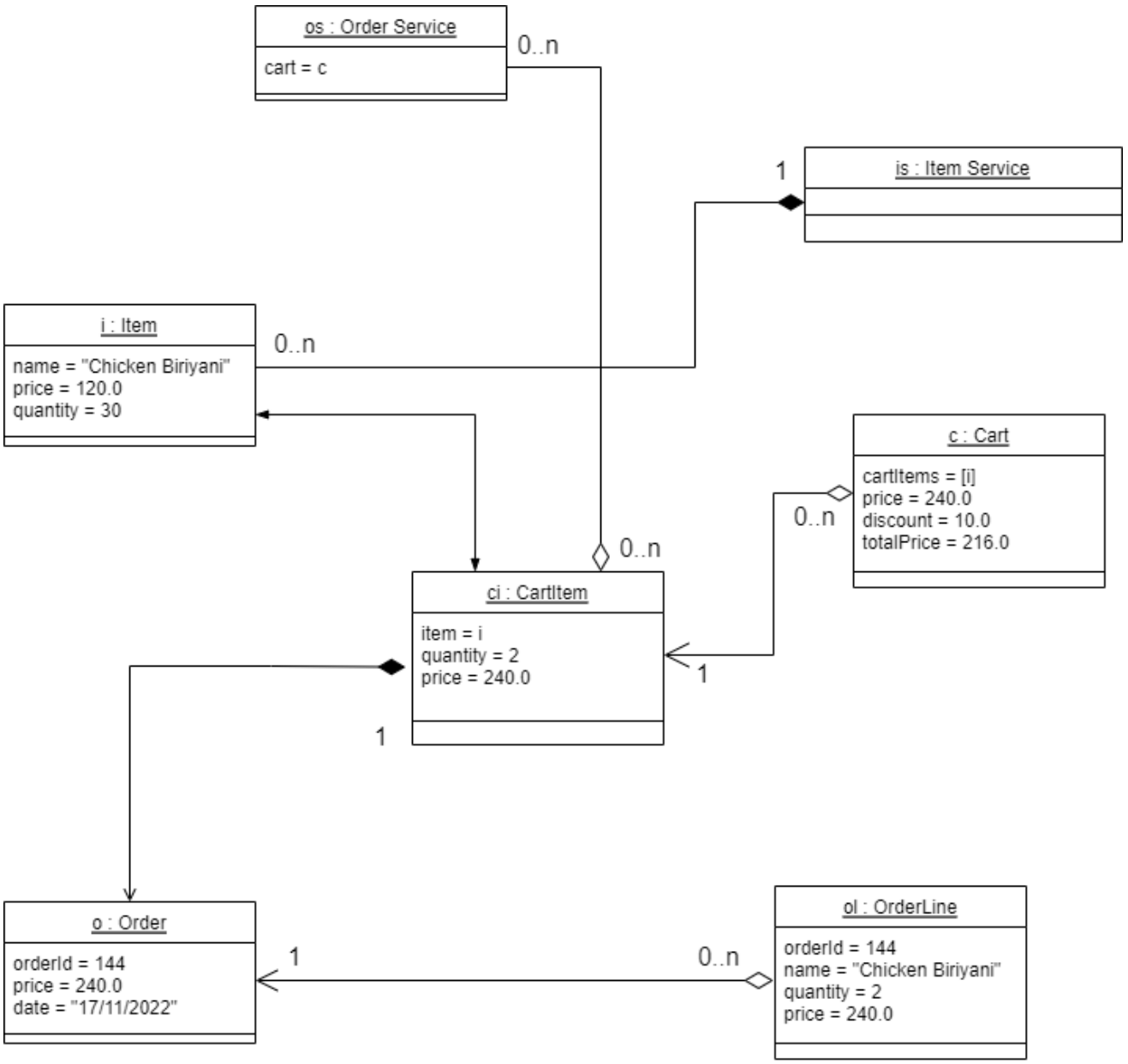
1.1.1 Class Diagram



1.2 Responsibilities - Use Case Diagram



1.3 Static snapshot of the system - Object Diagram

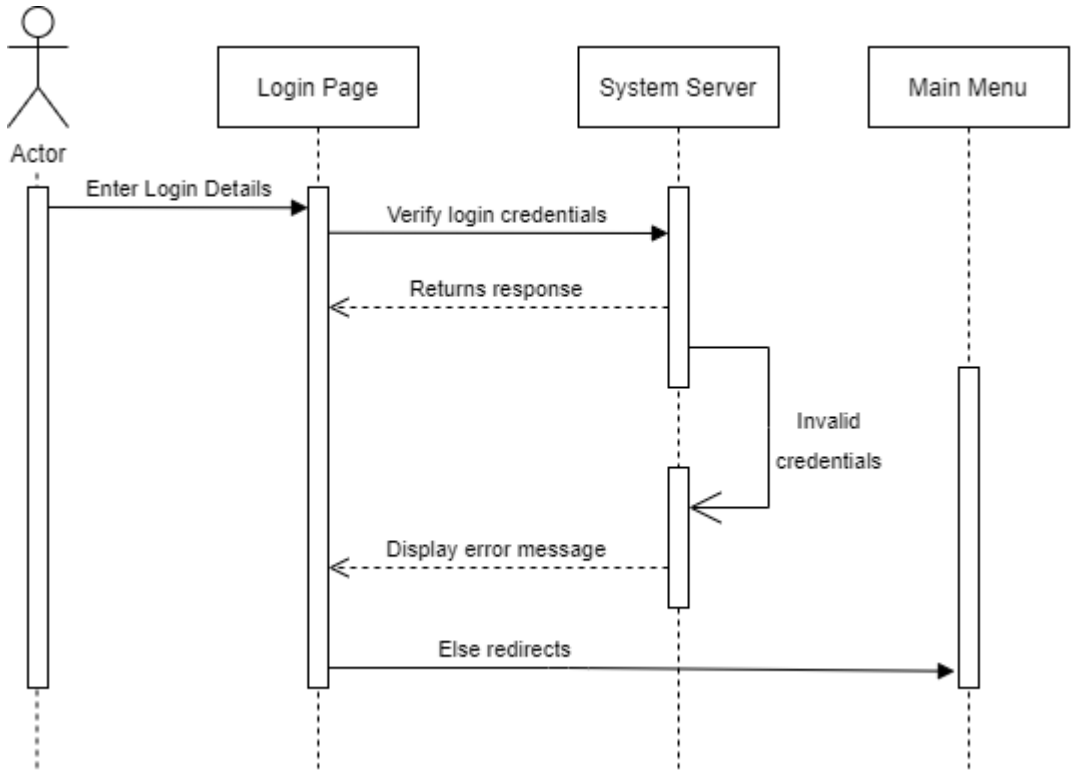


1.4 System Interactions through Sequence Diagrams

Sequence diagrams are interaction diagrams that show the sequence of messages exchanged by the set of objects performing a certain task. A sequence diagram shows, as parallel vertical lines (lifeline), different processes or objects that live simultaneously, and as horizontal arrows, the messages exchanged between them, in the order in which they occur.

1.4.1 Login

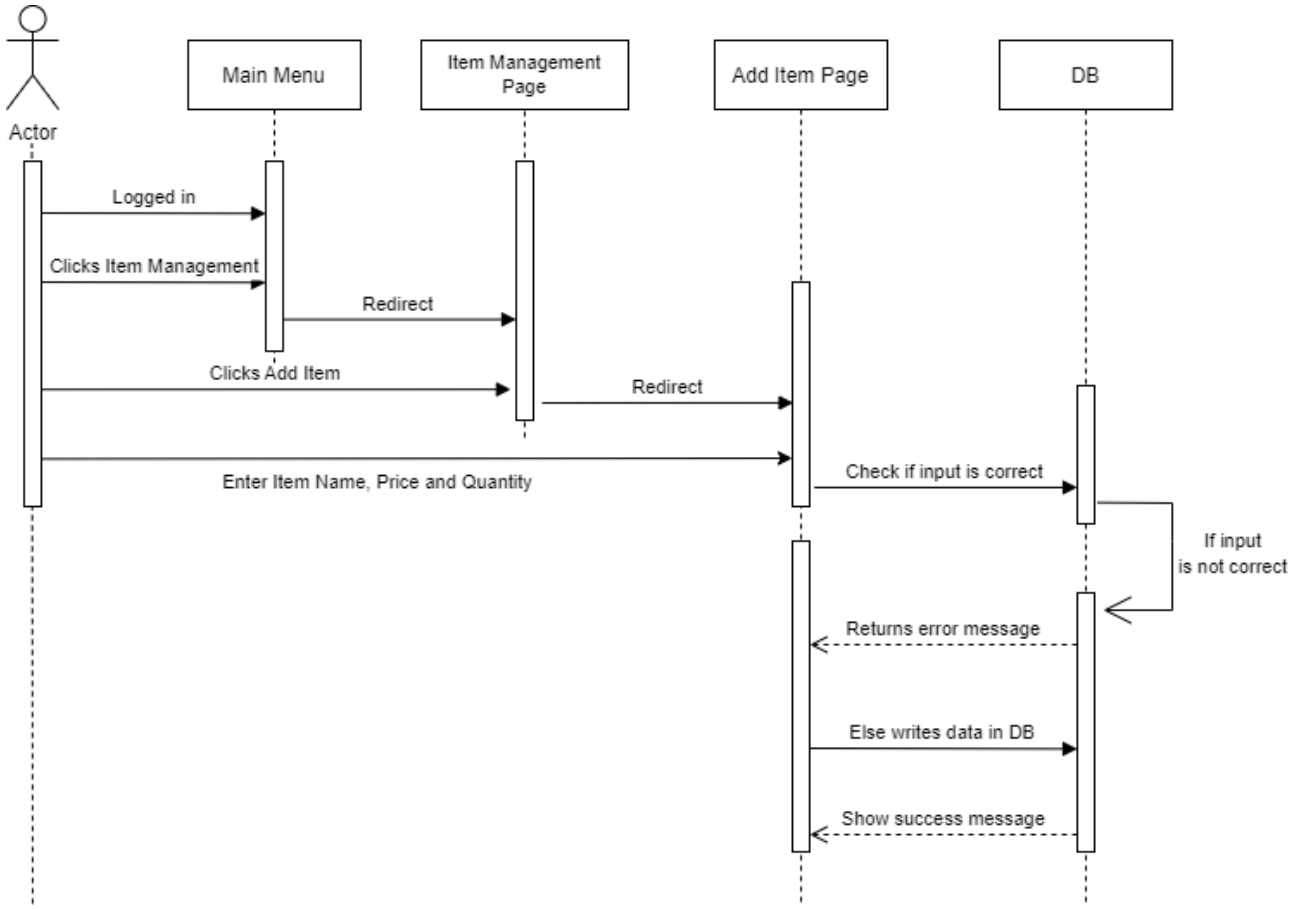
The login feature is provided for the biller (actor) of the restaurant. The user can log in using the user Id and password provided. The login credentials are then verified, and if valid then they are returned to the main menu where they have access to Item management and order management. If the credentials are invalid then an error message is displayed and the user stays on the login page.



1.4.2 Add items into the Inventory

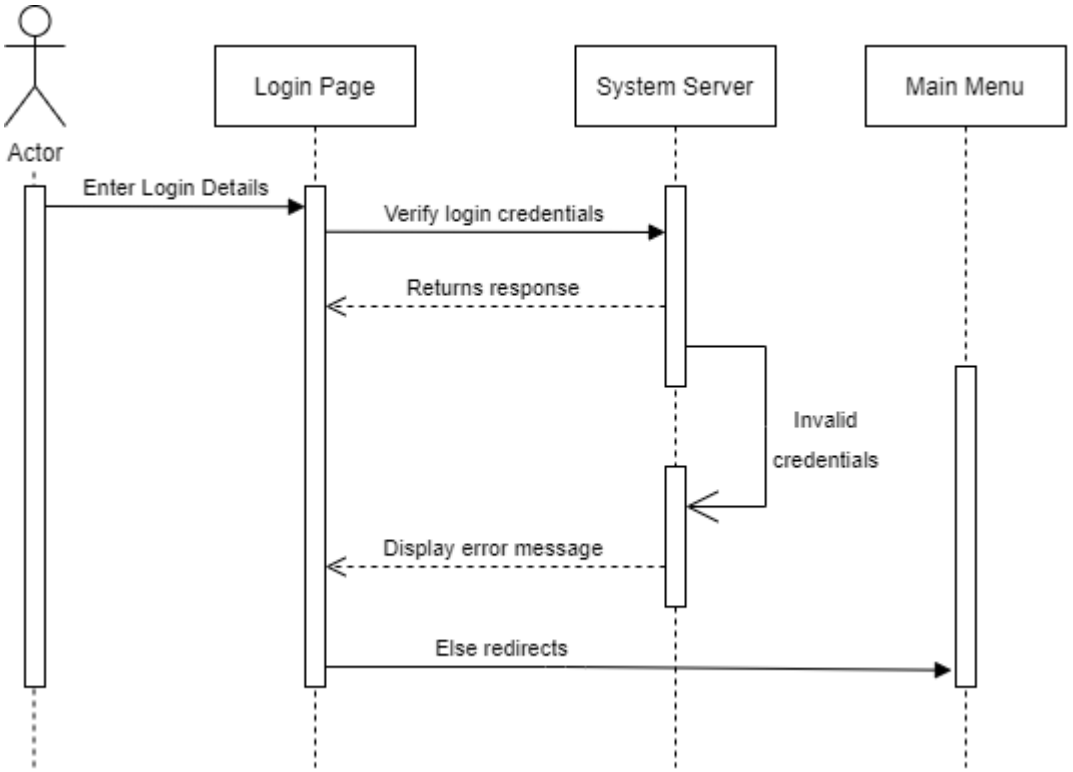
This feature allows the user to add items to the inventory after the user is logged in. User can add items to the inventory from the main menu by

selecting Item Management and selecting ‘Add Item’ and have to enter the item name, price, and quantity of items available for the day. The user can later select the ‘View Items’ option to see all the items added to the inventory.



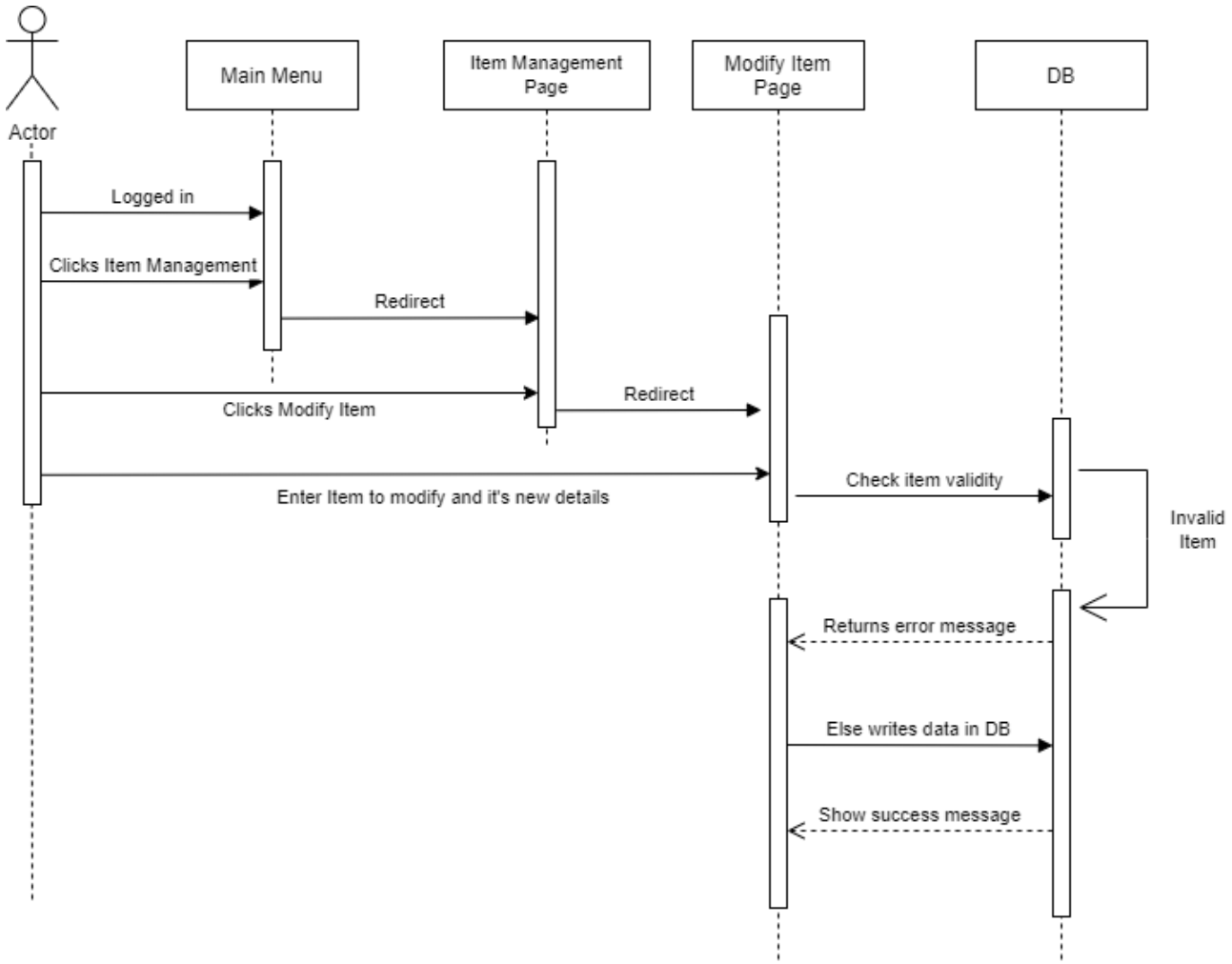
1.4.3 View items in the Inventory

This feature allows the user to view items in the inventory for the current day after the user is logged in. User can access the View Items from the main menu by selecting Item Management and selecting ‘View Item’.



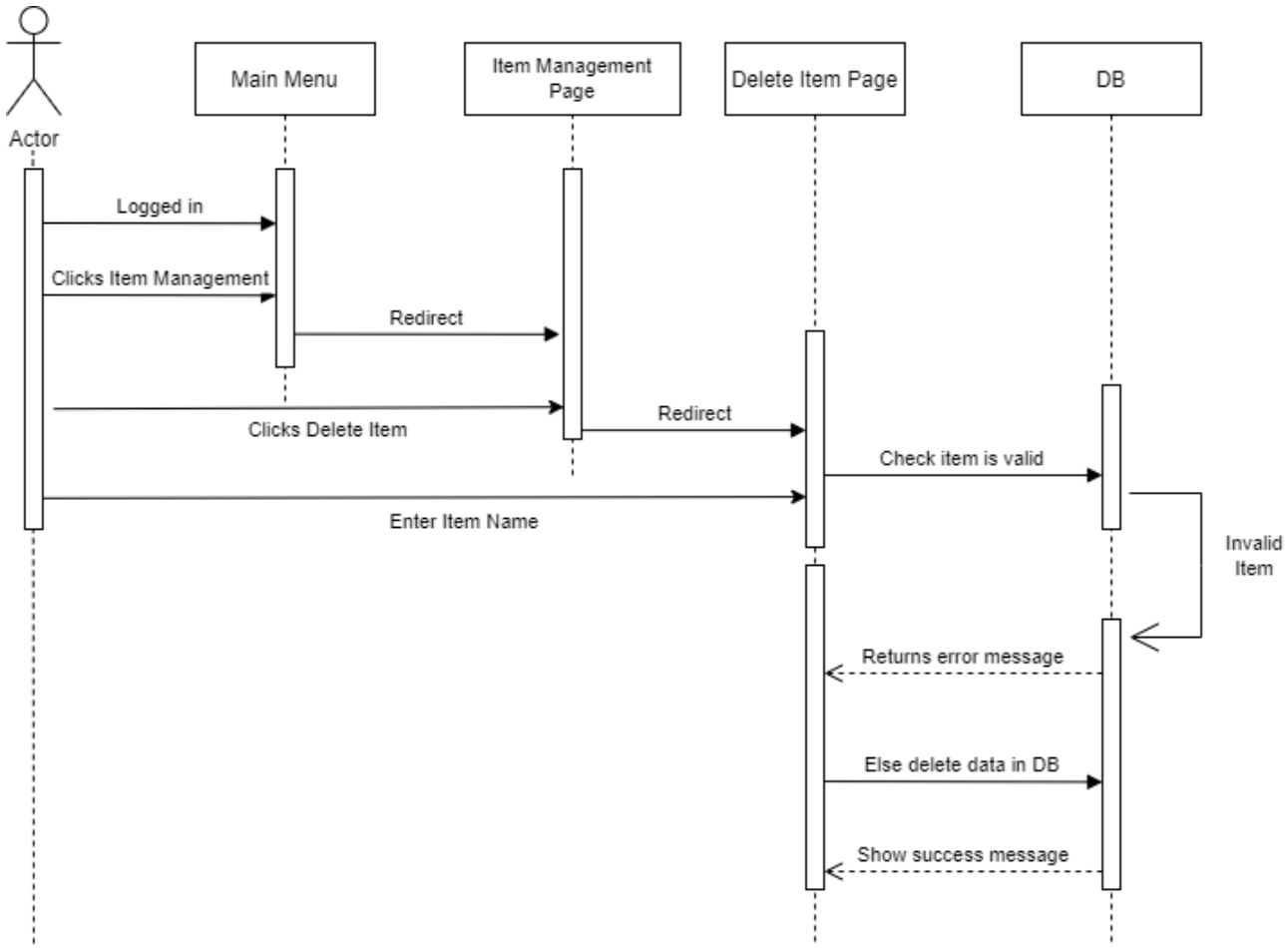
1.4.4 Modify Items in the Inventory

This feature allows the user to modify items in the inventory after the user is logged in. User can modify existing items from the main menu by selecting Item Management and selecting ‘Modify Item’ and have to enter the new item name, price, and quantity of an existing item in the inventory.



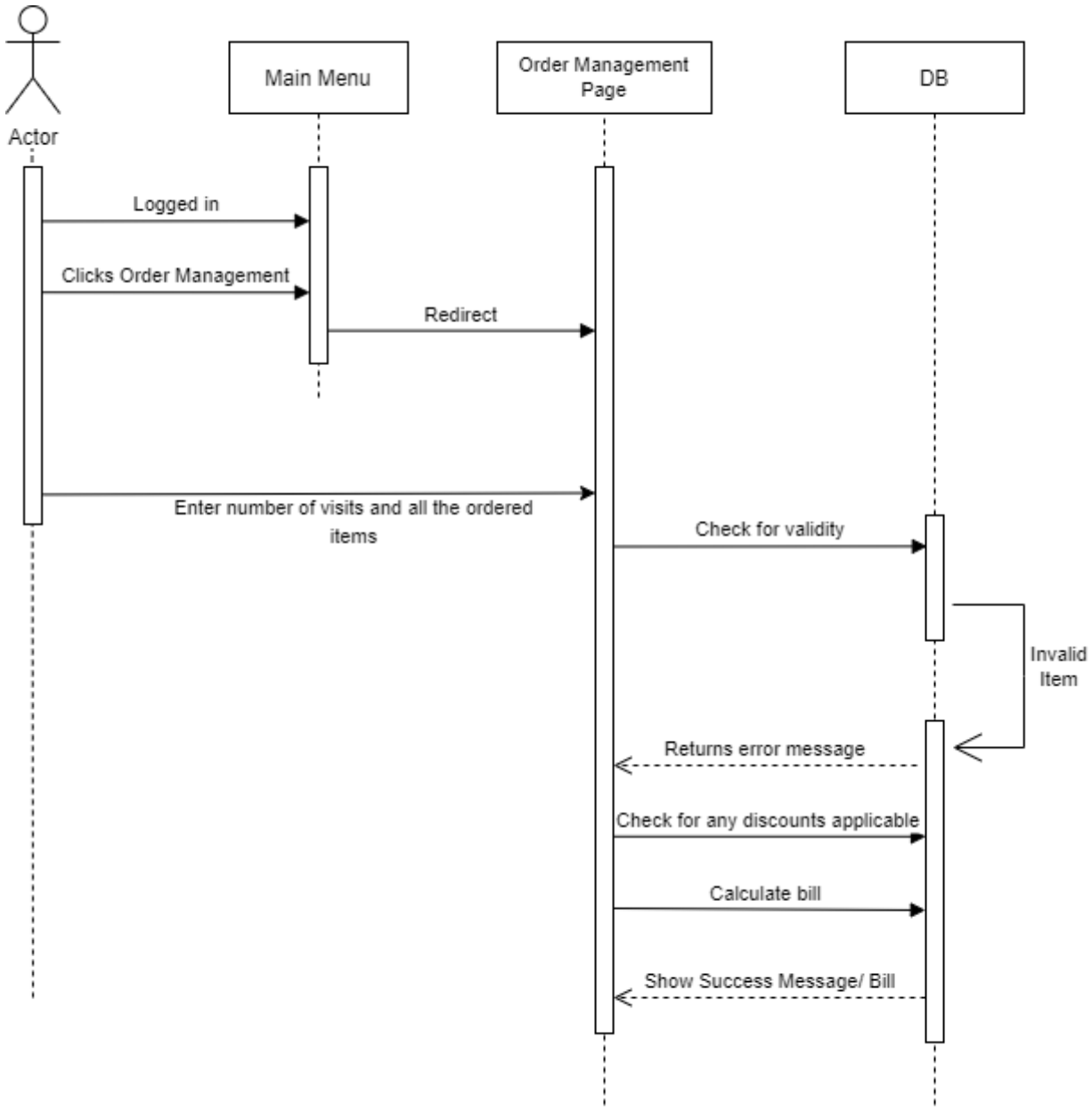
1.4.5 Delete Items in the inventory

This feature allows the user to delete items in the inventory after the user is logged in. User can delete existing Items from the main menu by selecting Item Management and selecting ‘Delete Item’ and have to enter the item name to be deleted from the inventory list. The user can later select the ‘VIEW ITEMS’ option to see the updated items in inventory.



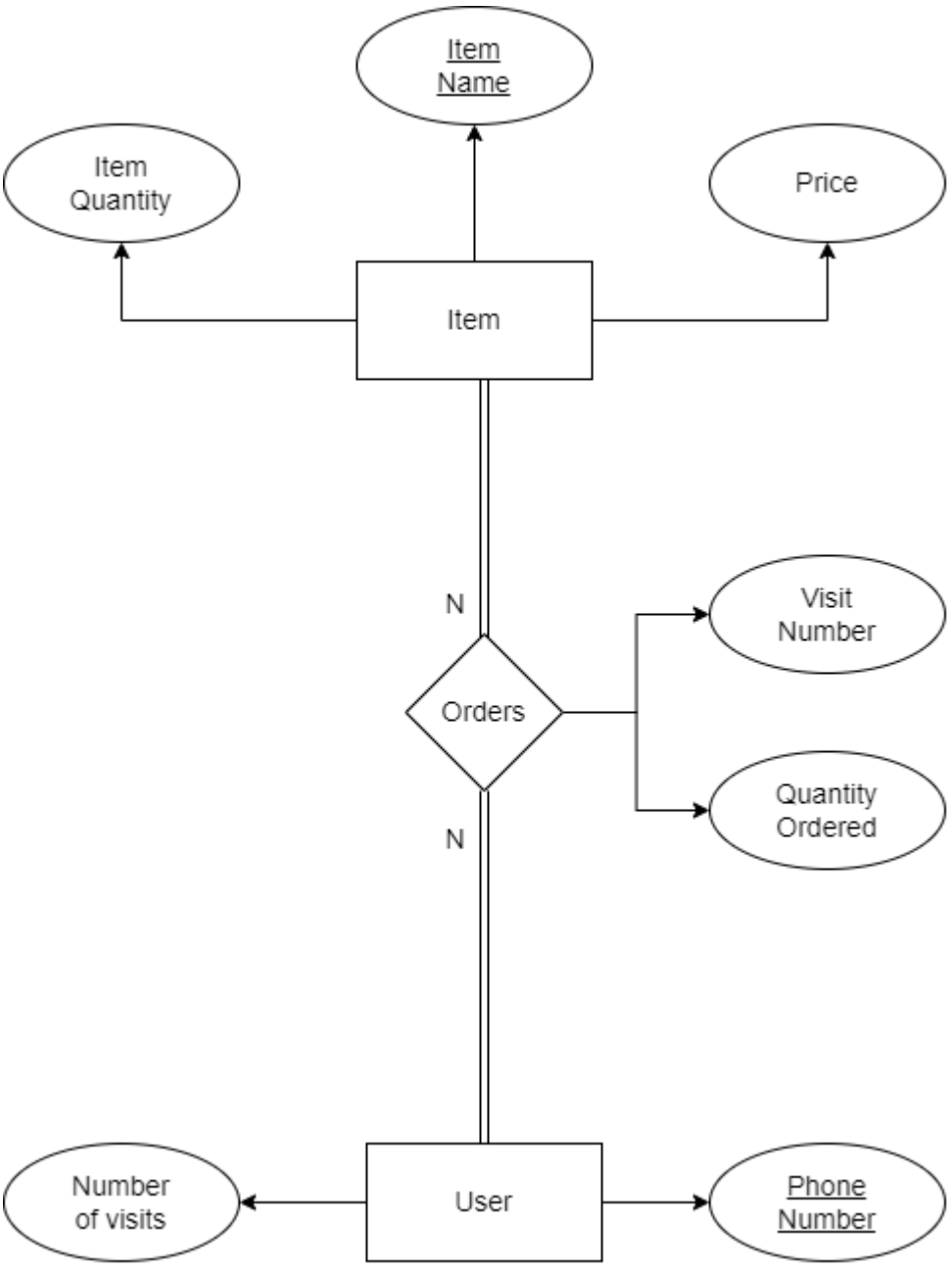
1.4.6 Order Items

This feature allows user to bill items that have been ordered by his customer. User can bill the items from the main menu by selecting order management and adding all the ordered items and their quantity to the cart. If the item and the respective quantity are available, the order can be placed and the bill is displayed.



2. Database Design

2.1 ER Diagram



3. Implementation Plans

3.1 Technology Stack

- Programming Languages - Java
- Framework - Swing
- Software Tools -
  - Draw.io for making various diagrams.
  - Google Docs for documentation.
  - Git for version control.
  - VS Code (IDE).

3.2 User Interface Prototyping

3.2.1 Login Page

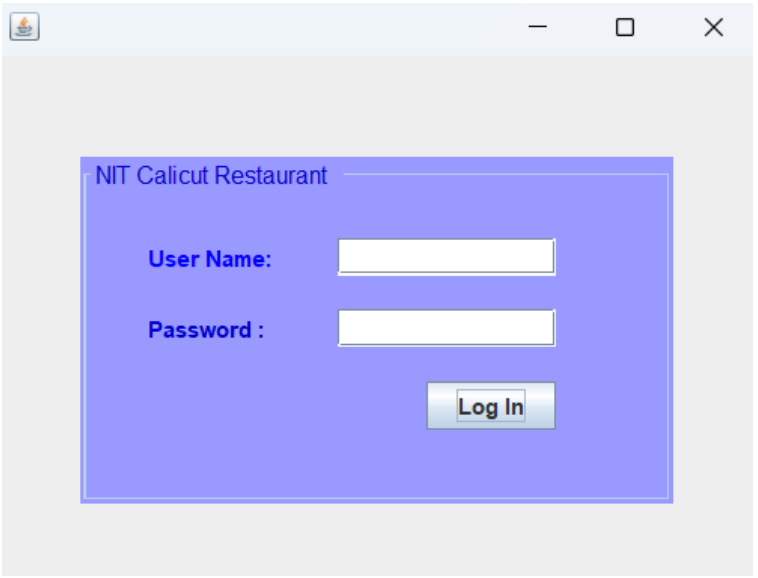


Figure 1: Login Page

3.2.2 Main Menu

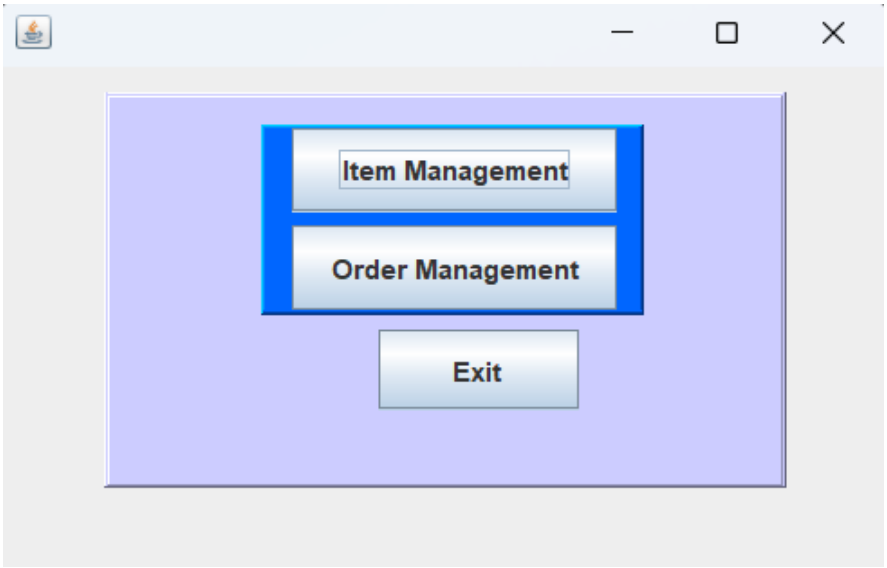


Figure 2: Main Menu

3.2.3 Item Management Page

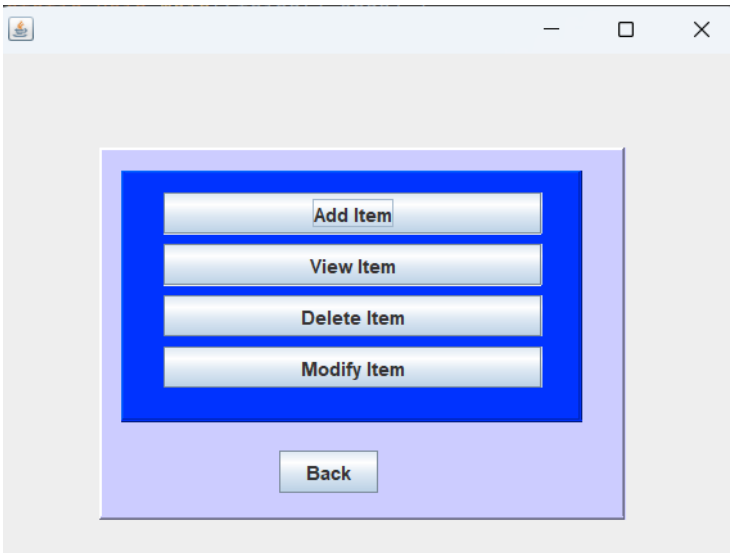


Figure 3: Item Management Page

3.2.4 Order Management

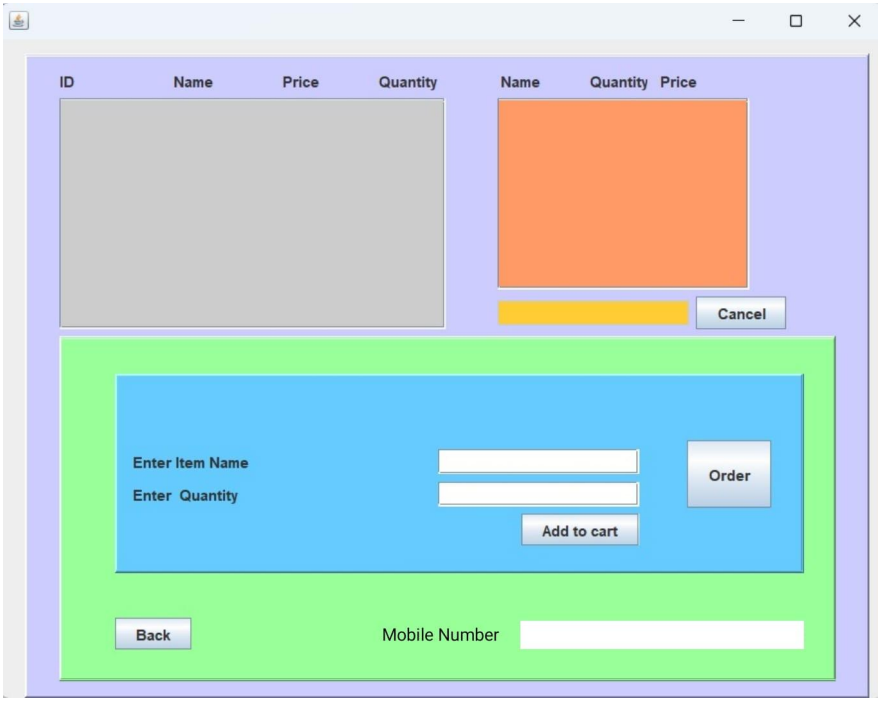


Figure 4: Order Management Page

4. Test Cases

4.1 Test Case #1 (TC_Login_001)

Author: Vishnu Ajay

Test Case Description:
Test scenario: Verify the admin login.
Test case: Enter a valid username and valid password provided.

Pre-Conditions:
Need a valid username and password provided by the developer of the application.

Test Steps:
Run the application.
Enter username
Enter password

Click Login

Test Data:
Username: admin123
Password: admin123

Expected Result:
Successful login and a message box is displayed showing “Access granted”.

Post Condition:
Main Menu is shown where the admin has access to item management and order management.

4.2 Test Case #2 (TC_Login_002)

Author: Vishnu Ajay

Test Case Description:
Test scenario: Verify the login with invalid credentials.
Test case: Enter an invalid username and invalid password.

Pre-Conditions:
Need a username and password.

Test Steps:
Run the application.
Enter username
Enter password
Click Login

Test Data:
Username: admin000
Password: admin000

Expected Result:
Login failed, and a message box is displayed showing “Access denied”.

Post Condition:
The user stays on the login page.

4.3 Test Case #3 (TC_AddItem_001)

Author: Abin Jacob John

Test Case Description:
Test scenario: Add items to the inventory.
Test case: Enter an item name, price, and quantity.

Pre-Conditions:
The user should be logged in.
Select Item Management from the main menu and select ‘Add Item’ from the Item Management menu.

Test Steps:
Run the application.
Enter username.
Enter password.
Click Login.
Select Item Management.
Select Add item.
Enter Name, Price, and Quantity.
Click Add.

Test Data:
Item Name: Rice
Price: 20
Quantity: 50

Expected Result:
The item will be successfully added to the inventory.

Post Condition:
The user stays on the Add Item page so that he can add more items (if any).

4.4 Test Case #4 (TC_AddItem_002)

Author: Abin Jacob John

Test Case Description:
Test scenario: Add items to the inventory.
Test case: Enter an item name, invalid price, and invalid quantity.

Pre-Conditions:
The user should be logged in.
Select Item Management from the main menu and select ‘Add Item’ from the Item Management menu.

Test Steps:
Run the application.
Enter username.
Enter password.
Click Login.
Select Item Management.
Select Add item.
Enter Name, Price, and Quantity.
Click Add.

Test Data:
Item Name: Rice
Price: abc
Quantity: abc

Expected Result:
The item will not be added to the inventory and a message box will be displayed requesting to enter valid input.

Post Condition:
The user stays on the Add Item page so that he can add a valid item.

4.5 Test Case #5 (TC_ViewItem_001)

Author: Vishnu Ajay

Test Case Description:

Test scenario: View items available in the inventory.
Test case: -

Pre-Conditions:
The user should be logged in.
Select Item Management from the main menu and select ‘View Items’ from the Item Management menu.

Test Steps:
Run the application.
Enter username.
Enter password.
Click Login.
Select Item Management.
Select View Items.

Test Data:
-

Expected Result:
All the items along with their name, price, and quantity will be displayed.

Post Condition:
The user stays on the View Item page and can navigate back to the Item Management page if needed.

4.6 Test Case #6 (TC_DeleteItem_001)

Author: Tom Saju

Test Case Description:
Test scenario: Delete items in the inventory.
Test case: Enter an item name to be deleted.

Pre-Conditions:
The user should be logged in.
Select Item Management from the main menu and select ‘Delete Item’ from the Item Management menu.

Test Steps:
Run the application.
Enter username.
Enter password.
Click Login.
Select Item Management.
Select Delete item.
Enter the name of the item to be deleted.
Click Delete Item.

Test Data:
Item Name: Rice

Expected Result:
The item will be deleted from the inventory, and a message box will be displayed showing the successful deletion of the item.

Post Condition:
The user stays on the Deleted Item page so that he can delete further items if required or he can navigate back to the Item Management Page.

4.7 Test Case #7 (TC_DeleteItem_002)

Author: Tom Saju

Test Case Description:
Test scenario: Delete items in the inventory.
Test case: Enter an item name that is not available in the inventory.

Pre-Conditions:
The user should be logged in.
Select Item Management from the main menu and select ‘Delete Item’ from the Item Management menu.

Test Steps:
Run the application.
Enter username.
Enter password.

Click Login.
Select Item Management.
Select Delete item.
Enter the name of an invalid item from the inventory to be deleted.
Click Delete Item.

Test Data:
Item Name: Beef Roast

Expected Result:
A message box displaying Item not found.

Post Condition:
The user stays on the Deleted Item page so that he can delete a valid item if required or he can navigate back to the Item Management Page.

4.8 Test Case #8 (TC_ModifyItem_001)

Author: Tom Saju

Test Case Description:
Test scenario: Modify an existing item in the inventory.
Test case: Enter new details of an existing item to be modified.

Pre-Conditions:
The user should be logged in.
Select Item Management from the main menu and select ‘Modify Item’ from the Item Management menu.

Test Steps:
Run the application.
Enter username.
Enter password.
Click Login.
Select Item Management.
Select Modify Item.
Enter the name of the item to be modified.
Enter the new details (name, price, and quantity) of the item.

Click Update Item.

Test Data:
Item Name to be modified: Rice
New Name: Lemon Rice
New Price: 25
New Quantity: 30

Expected Result:
The details of the item will be updated and a message will be displayed showing the successful updation of the item.

Post Condition:
The user stays on the Modify Item page so that he can modify any other item if required or he can navigate back to the Item Management Page.

4.9 Test Case #9 (TC_ModifyItem_002)

Author: Tom Saju

Test Case Description:
Test scenario: Attempt to modify an item that’s not available in the inventory.
Test case: Enter new details of an invalid item.

Pre-Conditions:
The user should be logged in.
Select Item Management from the main menu and select ‘Modify Item’ from the Item Management menu.

Test Steps:
Run the application.
Enter username.
Enter password.
Click Login.
Select Item Management.
Select Modify Item.
Enter the name of the item to be modified.
Enter the new details (name, price, and quantity) of the item.

Click Update Item.

Test Data:
Item Name to be modified: Biriyani
New Name: Chicken Biriyani
New Price: 150
New Quantity: 20

Expected Result:
A message will be displayed showing no item found to modify.

Post Condition:
The user stays on the Modify Item page so that he can modify an existing item if required or he can navigate back to the Item Management Page.

4.10 Test Case #10 (TC_OrderManagement_001)

Author: Tom Saju

Test Case Description:
Test scenario: Calculate Bill for an order.
Test case: Enter all the items ordered by the customer and his number of visits to the store to check for any discounts applicable.

Pre-Conditions:
The user should be logged in.
Select Order Management.

Test Steps:
Run the application.
Enter username.
Enter password.
Click Login.
Select Order Management.
Enter the number of visits to the store by the respected customer.
Add all the items ordered.
Click Generate Bill.

Test Data:
Enter number of visits: 4
Item Name: Biriyani
Quantity: 2

Expected Result:
Total bill will be calculated and displayed with discounts applied (if any).

Post Condition:
The user stays on the page so that he can generate a bill for the next customer or he can navigate back to the main page.

References

1. A. Bahrami, Object Oriented Systems Development, 1/e, McGraw-Hill, 2000.
2. B. D. McLaughlin, G. Police, and D. West, Head First Object-Oriented Analysis and Design, 1/e, O^Reilly.
3. R. S, Pressman, Software Engineering: A Practitioner's Approach, 6/e, McGraw Hill, 2008.