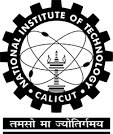
Software Requirements Specification

for

12. Cafeteria Management System

**Version 0.1**

**Prepared by**

**M210694CA Joel L Ralte**

**M210678CA Mayank Mewar**

**M210659CA Ravinder Singh**

|  |  |
| --- | --- |
| **Project Owner:** | ***Ms. Neethu*** |
| **Course:** | **CS4097D Object Oriented Systems Laboratory** |
| **Date:** | **3rd Nov,2022** |

[**1 Introduction**](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.q7s9eh1cg4p)3

[1.1 Document Purpose](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.3znysh7) 3

[1.2 Product Scope](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.tyjcwt) 3

[1.3 Definitions, Acronyms and Abbreviations](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.2s8eyo1) 3

[1.4 Document Conventions](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.3rdcrjn) 3

[1.5 References and Acknowledgments](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.lnxbz9) 4

[**2 Overall Description**](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.35nkun2)4-5

[2.1 Product Overview](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.1ksv4uv) 4

[2.2 Product Functionality](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.2jxsxqh) 5

[**3 Specific Requirements**](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.nsn84txrpv8l)5-16

[3.1 External Interface Requirements](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.1ci93xb) 5-9

[3.1.1 User Interfaces](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.3whwml4) 5-9

[3.2 Functional Requirements](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.2p2csry) 10

[3.3 Use Case Model](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.3o7alnk) 11-19

[3.3.1 Use Case #1(Admin Login)](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.23ckvvd) 12

[3.3.2 Use Case #2](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.ihv636)(Customer Registration) 12

[3.3.3 Use Case #3](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.ihv636)(Customer Login) 13

[3.3.4 Use Case #4](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.ihv636)(Verify Credentials) 13

[3.3.5 Use Case #5](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.ihv636)(Invalid Credentials) 14

[3.3.6 Use Case #6](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.ihv636)(Add Item) 14

[3.3.7 Use Case #7](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.ihv636)(Edit Item) 15

[3.3.8 Use Case #8](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.ihv636)(Delete Item) 15

[3.3.9 Use Case #9](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.ihv636)(Filter Items by Category) 16

[3.3.10 Use Case #10](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.ihv636)(View all Items) 16

[3.3.11 Use Case #11](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.ihv636)(View Customer Bills) 17

[3.3.12 Use Case #12](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.ihv636)(View All Bills) 17

[3.3.13 Use Case #13](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.ihv636)(Add Item to Bill) 18

[3.3.14 Use Case #14](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.ihv636)(Generate Bill) 18

[3.3.15 Use Case #15](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.ihv636)(Logout) 19

[3.3.16 Use Case #16](file:///C:\Users\Mayank\AppData\Local\Microsoft\Windows\INetCache\IE\AYG1B2V6\srsfinal21212%5b1%5d.docx#_heading=h.ihv636)(Exit Application) 19

# 

# 

# Introduction

## Document Purpose

This SRS describes the software functional and non-functional requirements for release 1.0 of the Cafeteria Ordering System (COS). This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here are high priority and committed for release 1.0.

## Product Scope

The Cafeteria Management System permits the Customer to view meals from the cafeteria menu and place order. CMS calculates the bill for each Customer and provide them with the bill in the form of a Word or pdf file. The Admin in the CMS can add new meals, delete meals or edit a meal from menu.

## Definitions, Acronyms and Abbreviations

CMS = Cafeteria Management System

Admin = Application Administrator

IEEE = The Institute of Electrical and Electronics Engineers

Item= Meal

## Document Conventions

**Formatting Convention**

1. Arial font style of size 11 or 12 is used throughout the document. For heading font size of 33, 25, 18, 14 is used.
2. Italics is used for comments.
3. Bold is used for Headings.
4. 1’’ margin is maintained throughout the document.
5. Lists are represented using numbered list.

**Naming Convention**

1. Heading are all starting with capital letter and rest small letters.
2. Functions calls are written in camelCase.

## References and Acknowledgments

1. [Star uml usage guide video](https://www.youtube.com/watch?v=wR1SQqcIJhM)

2. [Swing Tutorials](https://www.javatpoint.com/java-swing)

3. [Java swing with mysql tutorial](https://www.javaguides.net/2019/07/java-swing-jdbc-mysql-example.html)

# 2 Overall Description

## 2.1 Product Overview

The Cafeteria Ordering System is a new system that replaces the current manual processes for ordering and picking up meals in the Process Impact cafeteria. The system is expected to evolve over several releases, ultimately connecting to the Internet ordering services for several local restaurants and to credit and debit card authorization services.

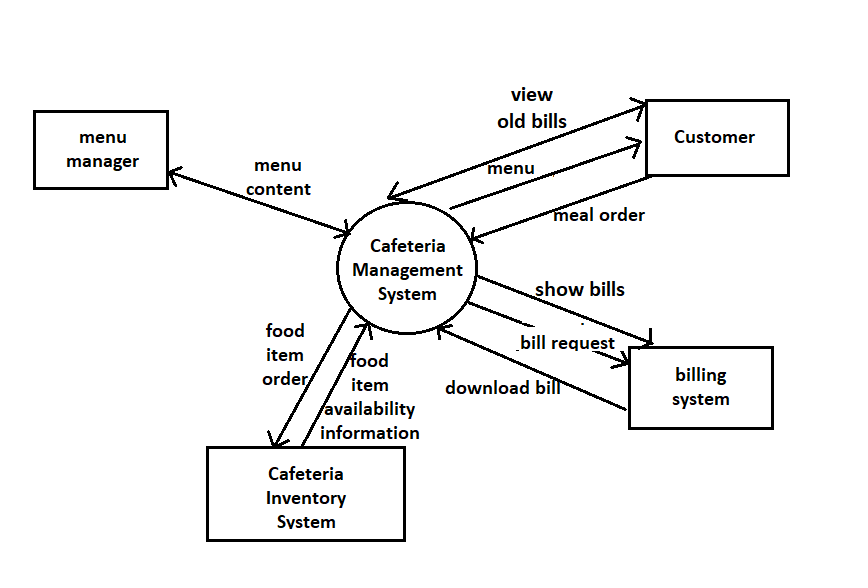


Figure 1

Context diagram for release 1.0 of the Cafeteria Management System.

## Product Functionality

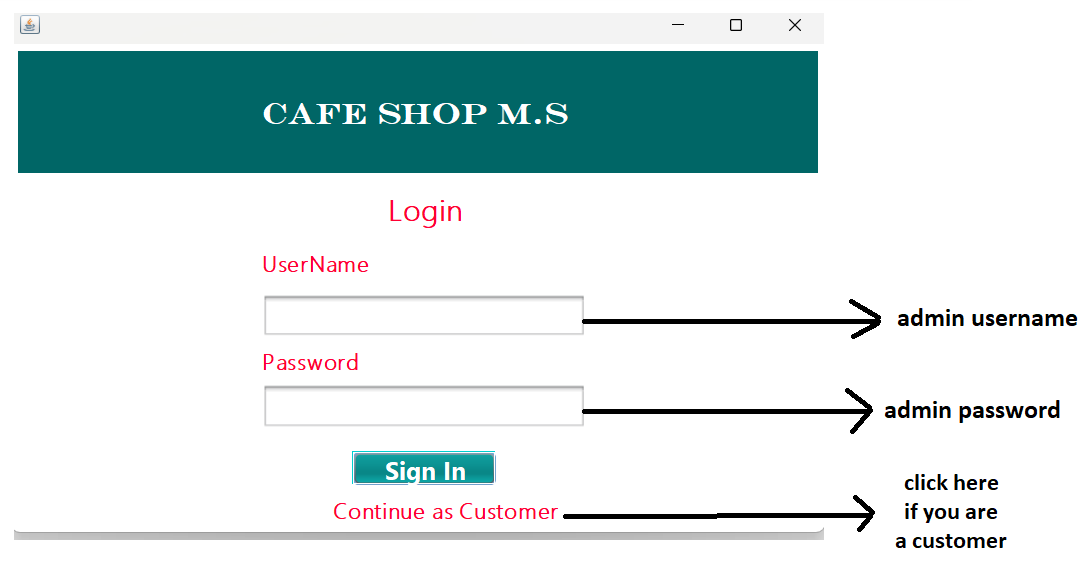
* User can Order Meals from Seller.
* Generate bill in form of DOCX or PDF.
* Admin can add, edit, delete meals from the Menu.
* Admin has access to total earning in a day.

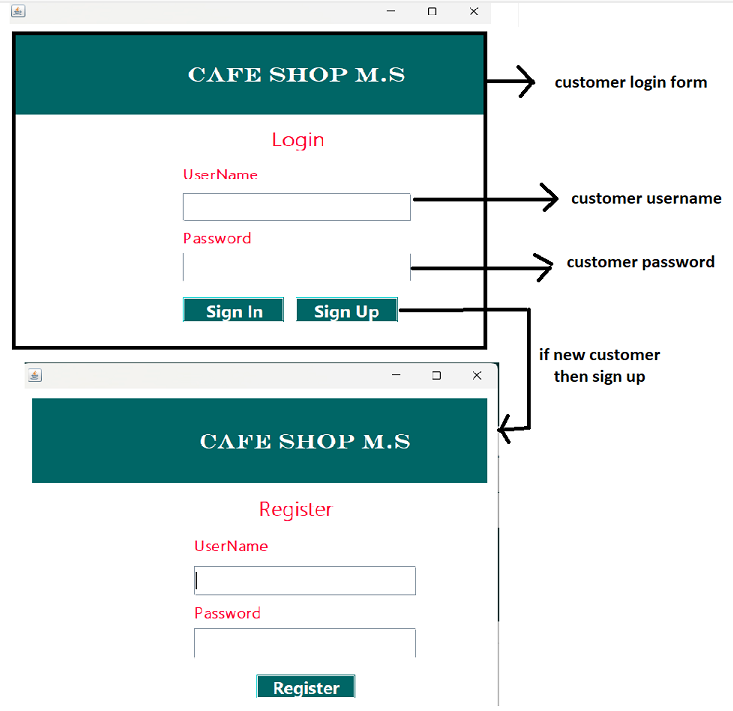
# 3 Specific Requirements

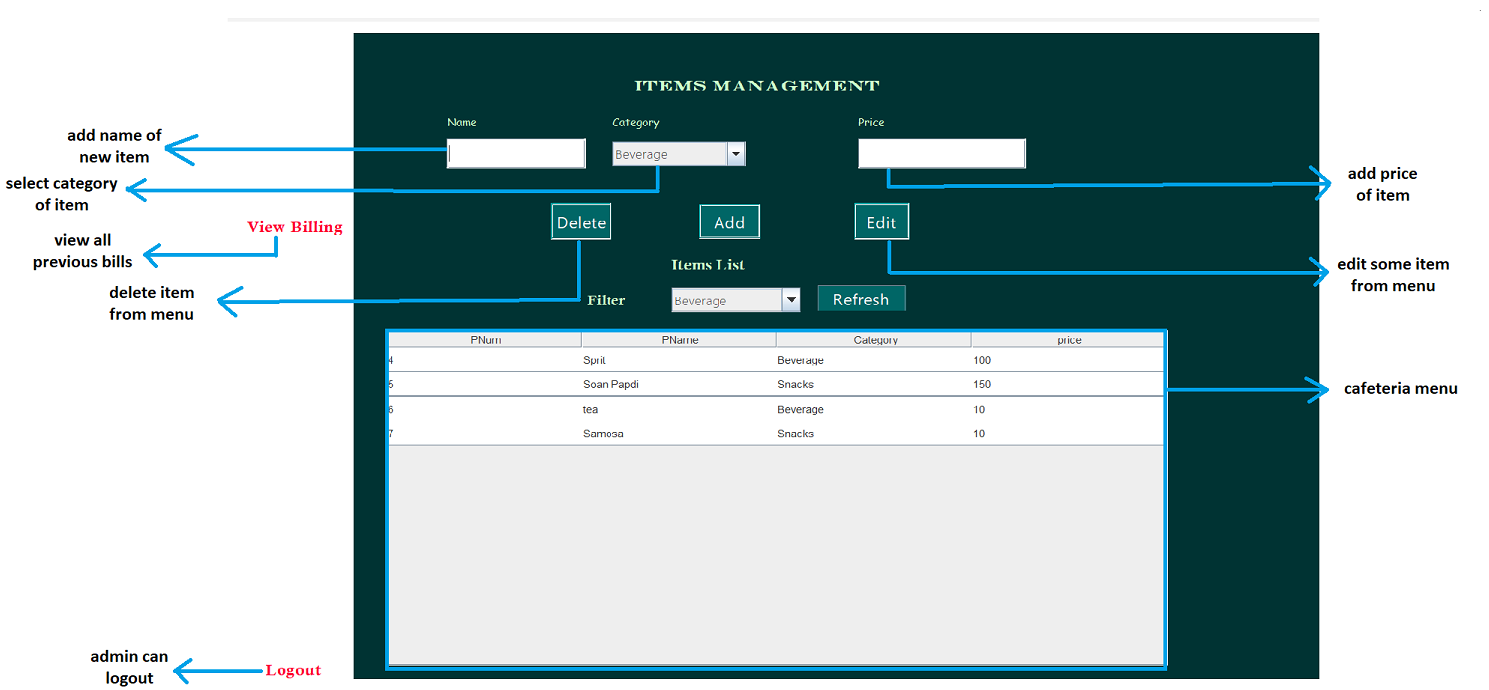
## 3.1 External Interface Requirements

**3.1.1 User Interfaces**

## 







## 

## 

## 

## 

## 3.2 Functional Requirements

**F1:** The system shall allow an administrator to login.

**F2:** The system shall allow a new customer to register.

**F3:** The system shall allow an existing customer to login.

**F4:** The system shall validate login credentials.

**F5:** On clicking the Exit button the application will be closed.

**F6:** On clicking the Logout button the user will be sent back to the login page.

**F7:** The system shall check if the item admin is adding has a Name, Category and Price field set.

**F8:** The system shall show all available meals in a table along with Category of meal and price of meal.

**F9:** On clicking any row of menu table the text fields Name, Category and Price get auto filled.

**F10:** On clicking the Add button the item is added (with fields Name, Category and Price).

**F11:** On clicking the Edit button Admin can update the Product Name, Category or Price.

**F12:** On clicking the Delete button the currently selected row of menu is deleted.

**F13:** Filter allows to view a particular category of meals in the menu.

**F14:** On clicking any row of menu table the text fields Name, Price gets auto filled.

**F15:** On button click the meal is added to a Bill table with Name, Price, Quantity and Total.

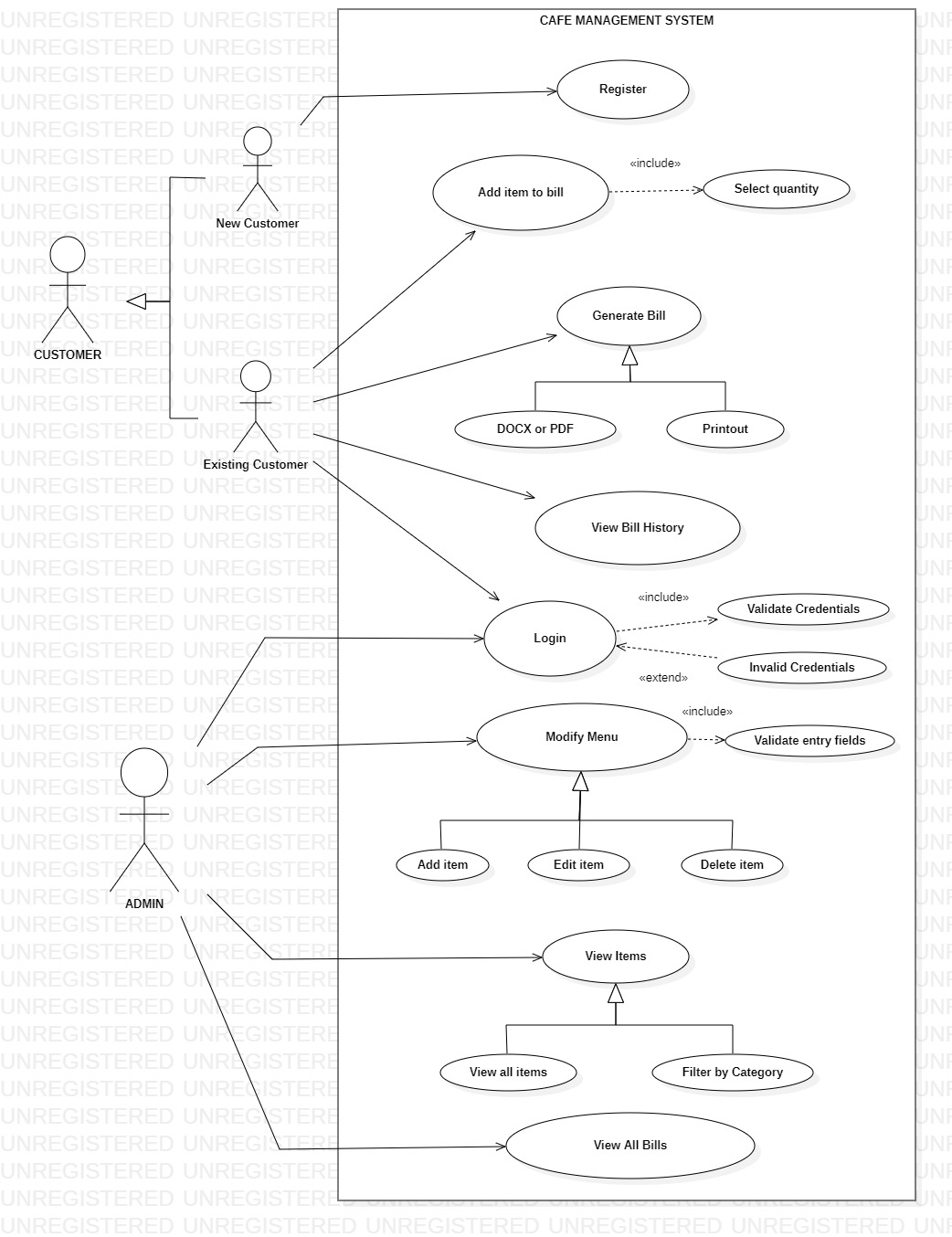
**F16:** On clicking Print button the system shall print the bill in the desired format – Physical printout or DOCX/PDF format.

**F17:** System shall save the bill in the database.

**F18:** System shall show previous bills of the logged in customer.

**F19:** System shall show all the previous bills.

## 3.2 Use Case Model

­

### 3.2.1 Use Case #1 (Admin Login)

**Author –** Joel L Ralte

**Purpose -** To login existing admins to the system using valid credentials.

**Requirements Traceability –** F1

**Priority -** High, admins need to login to add or modify the menu items which the seller can sell to the customers.

**Preconditions -** User must have an admin account for this system.

**Post conditions -** After username and password verification, admin will be logged in to the system.

**Actors –** Admin

**Extends –** Invalid Credentials

**Flow of Events**

1. Basic Flow - A user must enter a valid username and password to have access to the system. After submitting the valid credentials, the user will be logged into the system.
2. Alternative Flow - If the entered username or password is incorrect, then an error message will be displayed and the user will be asked to re-enter the credentials.

**Includes -** Verify Credentials

**Notes/Issues –** None

### 3.2.2 Use Case #2(Customer Registration)

**Author –** Joel L Ralte

**Purpose –** To register a new customer into the system.

**Requirements Traceability –** F2

**Priority –** High, customer needs to register before he can login to the system.

**Preconditions –** None

**Post conditions –** New account created for customer.

**Actors –** New Customer

**Extends –** None

**Flow of Events**

1. Basic Flow – A user must enter username and password. After that the account is created.

**Includes –** None

**Notes/Issues –** None

### 3.2.3 Use Case #3(Customer Login)

**Author –** Joel L Ralte

**Purpose –** To log in existing customer into the system using valid credentials.

**Requirements Traceability –** F3

**Priority –** High, customer needs to login so that he/she can order items with desired quantity and generate the bill.

**Preconditions –** Customermust have an account for this system.

**Post conditions -** Customer gets logged in to the system.

**Actors –** Customer

**Extends –** Invalid Credentials

**Flow of Events**

1. Basic Flow – A user must enter a valid username and password to have access to the system. After submitting the valid credentials, the user will be logged into the system.
2. Alternative Flow - If the entered username or password is incorrect, then an error message will be displayed and the user will be asked to re-enter the credentials.

**Includes –** Verify Credentials

**Notes/Issues –** None

### 3.2.4 Use Case #4(Verify Credentials)

**Author –** Joel L Ralte

**Purpose –** To prevent unauthorized access.

**Requirements Traceability –** F1,F3

**Priority –** High, a login attempt will only be approved after the credentials have been verified by the database.

**Preconditions –** User needs to enter credentials into the system.

**Post conditions -** User gets logged in to the system.

**Actors –** Can beAdmin or Customer

**Extends –** None

**Flow of Events**

1. Basic Flow – If the credential input is verified the user is logged in to the system.

**Includes –** None

**Notes/Issues –** None

### 3.2.5 Use Case #5(Invalid Credentials)

**Author –** Joel L Ralte

**Purpose –** To prevent unauthorized access.

**Requirements Traceability –** F4

**Priority –** High, only verified users are allowed to login to the system.

**Preconditions –** User enters wrong credentials into the system.

**Post conditions –** A pop-up message is displayed showing the credentials are invalid.

**Actors –** Can be Admin or Customer

**Extends –** None

**Flow of Events**

1. Basic Flow – If the credential input is invalid the login attempt will fail.

**Includes –** None

**Notes/Issues –** None

### 3.2.6 Use Case #6(Add Item)

**Author –** Joel L Ralte

**Purpose –** To add an item to the menu

**Requirements Traceability –** F10

**Priority –** Medium, admin should be able to add new items to the menufor the customer to order.

**Preconditions –** Admin must be logged in to the system

**Post conditions –** A pop-up confirmation is displayed showing that it has been added.

**Actors –** Admin

**Extends -** None

**Flow of Events**

1. Basic Flow – Admin must fill in all the required details i.e., name of the item, its category and price to add the item.

**Includes –** Validation of entry fields

**Notes/Issues –** None

### 3.2.7 Use Case #7(Edit Item)

**Author –** Joel L Ralte

**Purpose –** To edit an item from the menu

**Requirements Traceability –** F9, F11

**Priority –** Medium, admin should be able to edit items in the menu in case of any changes proposed.

**Preconditions –** Admin must be logged in to the system, Item must be present

**Post conditions –** A pop-up confirmation is displayed showing that it has been edited.

**Actors –** Admin

**Extends –** None

**Flow of Events**

1. Basic Flow – Admin selects an item from the list and enters the new details. After pressing the Edit button, the new details are saved.

**Includes –** Validation of entry fields

**Notes/Issues –** None

### 3.2.8 Use Case #8(Delete Item)

**Author –** Joel L Ralte

**Purpose –** To delete an item from the menu

**Requirements Traceability –** F9, F12

**Priority –** Medium, admin should be able to delete items from the menu in case it is out of stock.

**Preconditions –** Admin must be logged in to the system, Item must be present.

**Post conditions –** A pop-up confirmation is displayed showing that it has been deleted.

**Actors –** Admin

**Extends –** None

**Flow of Events**

1. Basic Flow – Admin selects an item to be deleted from the list. After pressing the Delete button the item is removed from the database.

**Includes –** Validation of entry fields

**Notes/Issues –** None

### 3.2.9 Use Case #9(Filter Items by Category)

**Author –** Joel L Ralte

**Purpose –** To filter items by category and view them.

**Requirements Traceability –** F13

**Priority –** Medium, admin should be able to view items in the menu according to their desired filter.

**Preconditions –** Admin must be logged in to the system

**Post conditions –** Items are viewed according to the selected category.

**Actors –** Admin

**Extends –** None

**Flow of Events**

1. Basic Flow – Admins selects a category from the drop-down list. If items belonging to that category are present, only those will be shown to the admin.

**Includes –** None

**Notes/Issues –** None

### 3.2.10 Use Case #10(View All Items)

**Author –** Joel L Ralte

**Purpose –** To view all the items in the menu

**Requirements Traceability –** F8

**Priority –** Medium, admin should be able to view items present in the menu.

**Preconditions –** Admin must be logged in to the system

**Post conditions –** Entire list of items displayed.

**Actors –** Admin

**Extends –** None

**Flow of Events**

1. Basic Flow – All items that are present in the menu are shown.

**Includes –** None

**Notes/Issues –** None

### 3.2.11 Use Case #11(View Customer Bills)

**Author –** Joel L Ralte

**Purpose –** To view all the bills that have been generated so far by the customer.

**Requirements Traceability –** F18

**Priority –** Medium, customer should be able to view the all the bills he has made so far.

**Preconditions –** Customer must be logged in to the system

**Post conditions –** List of all previous bills generated by the customer.

**Actors –** Customer

**Extends –** None

**Flow of Events**

1. Basic Flow – All the bills that have been generated by the customer are listed with their customer name, bill date and total amount.

**Includes –** None

**Notes/Issues –** None

### 3.2.12 Use Case #12(View All Bills)

**Author –** Joel L Ralte

**Purpose –** To view all the bills that have been generated so far by all the customers.

**Requirements Traceability –** F19

**Priority –** Medium, admin should be able to view the all the orders made so far.

**Preconditions –** Admin must be logged in to the system

**Post conditions –** List of all bills.

**Actors –** Admin

**Extends –** None

**Flow of Events**

1. Basic Flow – All the bills that have been generated by the customers are listed with their customer name, bill date and total amount.

**Includes –** None

**Notes/Issues –** None

### 3.2.13 Use Case #13(Add Item to Bill)

**Author –** Joel L Ralte

**Purpose –** To add an item to the bill.

**Requirements Traceability –** F14, F15

**Priority –** High, Customer should be able to order items from the menu.

**Preconditions –** Customermust be logged in to the system.

**Post conditions –** Item name along with its total priceare added to the bill.

**Actors –**Customer

**Extends –** None

**Flow of Events**

1. Basic Flow – Customer chooses the item name and selects quantity which the Seller adds to the bill.

**Includes –** Select quantity

**Notes/Issues –** None

### 3.2.14 Use Case #14(Generate Bill)

**Author –** Joel L Ralte

**Purpose –** To generate the bill.

**Requirements Traceability –** F16, F17

**Priority –** High, Customer should be able to view the bill to know what he ordered and the amount he needs to pay.

**Preconditions –** Customermust be logged in to the system.

**Post conditions –** A bill stating the item names, is generated in the desired format (soft copy or printout) and is saved in the database.

**Actors –** Customer

**Extends –** None

**Flow of Events**

1. Basic Flow – Customer receives the generated bill.

**Includes –** None

**Notes/Issues –** None

### 3.2.15 Use Case #15(Logout)

**Author –** Joel L Ralte

**Purpose –** To log out of the system.

**Requirements Traceability –** F6

**Priority –** Medium, an user can log out after he is finished using the system.

**Preconditions –** User must be logged in.

**Post conditions –** Returned to login page.

**Actors –** Can beAdmin or Customer

**Extends –** None

### 3.2.16 Use Case #16(Exit Application)

**Author –** Joel L Ralte

**Purpose –** To quit the system.

**Requirements Traceability –** F5

**Priority –** Medium, the user can exit once he is finished using the system.

**Preconditions –** System must be opened.

**Post conditions –** System is closed.

**Actors –** Admin, Seller

**Extends –** None

**Flow of Events**

1. Basic Flow – The user exits the system.

**Includes –** None

**Notes/Issues –** None

<Leave this page blank>

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Design Document**

**for**

**Cafeteria Management System**

Version 0.1

Prepared by Team 12:

(Based on SRS Version 0.1 prepared by Team 12)

|  |  |  |
| --- | --- | --- |
| **Ravinder Singh** | **M210659CA** | [**ravinder\_m210659ca@nitc.ac.in**](mailto:ravinder_m210659ca@nitc.ac.in) |
| **Joel L Ralte** | **M210694CA** | **joel\_m210694ca@nitc.ac.in** |
| **Mayank Mewar** | **M210678CA** | **mayank\_m210678ca@nitc.ac.in** |
|  |  |  |
|  |  |  |

## 

|  |  |
| --- | --- |
| **Project Owner:** | **Ms Neethu** |
| **Course:** | **CS4097D Object Oriented Systems Laboratory** |
| **Date:** | **14-11-2022** |

# Glossary

|  |  |
| --- | --- |
| Admin | System Administrator |
| ack | Acknowledgement |
| GUI | Graphical User Interface |
| JDBC | Java Database Connectivity |
| IDE | Integrated Development Environment |

## **Table of contents**

[**Glossary**](#_heading=h.tyjcwt) **[22](#_heading=h.tyjcwt)**

[**Table of contents**](#_heading=h.3dy6vkm) **[23](#_heading=h.3dy6vkm)**

[**Detailed Design through UML diagrams**](#_heading=h.1t3h5sf) **[24-30](#_heading=h.1t3h5sf)**

[1.1 System model using Class Diagram](#_heading=h.4d34og8) [24](#_heading=h.4d34og8)

[1.1.1 Class Diagram](#_heading=h.2s8eyo1) [24](#_heading=h.2s8eyo1)

[1.2 Responsibilities - Usecase Diagram](#_heading=h.17dp8vu) [25](#_heading=h.17dp8vu)

[1.3 Static snapshot of the system - Object Diagram](#_heading=h.3rdcrjn) 2[6](#_heading=h.3rdcrjn)

[1.4 System Interactions through Sequence Diagrams](#_heading=h.26in1rg) [26-30](#_heading=h.26in1rg)

[1.4.1 Customer](#_heading=h.lnxbz9) Selling Interaction [26](#_heading=h.lnxbz9)

[1.4.2 Customer](#_heading=h.35nkun2) ViewSells Interaction [27](#_heading=h.35nkun2)

[1.4.3 Admin](#_heading=h.1ksv4uv) ViewSells Interaction [28](#_heading=h.1ksv4uv)

[1.4.4 Admin](#_heading=h.44sinio) Items Interaction 29

[**Database Design**](#_heading=h.4i7ojhp) **[31](#_heading=h.4i7ojhp)**

[2.1 ER Diagram](#_heading=h.2xcytpi) [31](#_heading=h.2xcytpi)

[**Implementation Plans**](#_heading=h.1ci93xb) **[32](#_heading=h.1ci93xb)**

[3.1 Technology Stack](#_heading=h.3whwml4) **[32](#_heading=h.3whwml4)**

[**3.2 User Interface Prototyping**](#_heading=h.ikt0rwhftybi) **32-36**

**Test Cases [37-38](#_heading=h.ikt0rwhftybi)**

[**4.1 Test Case #1 (Test Case ID)**](#_heading=h.23ckvvd) **[37](#_heading=h.23ckvvd)**

[**4.2 Test Case #2 (Test Case ID)**](#_heading=h.exl6z521e60e) **38**

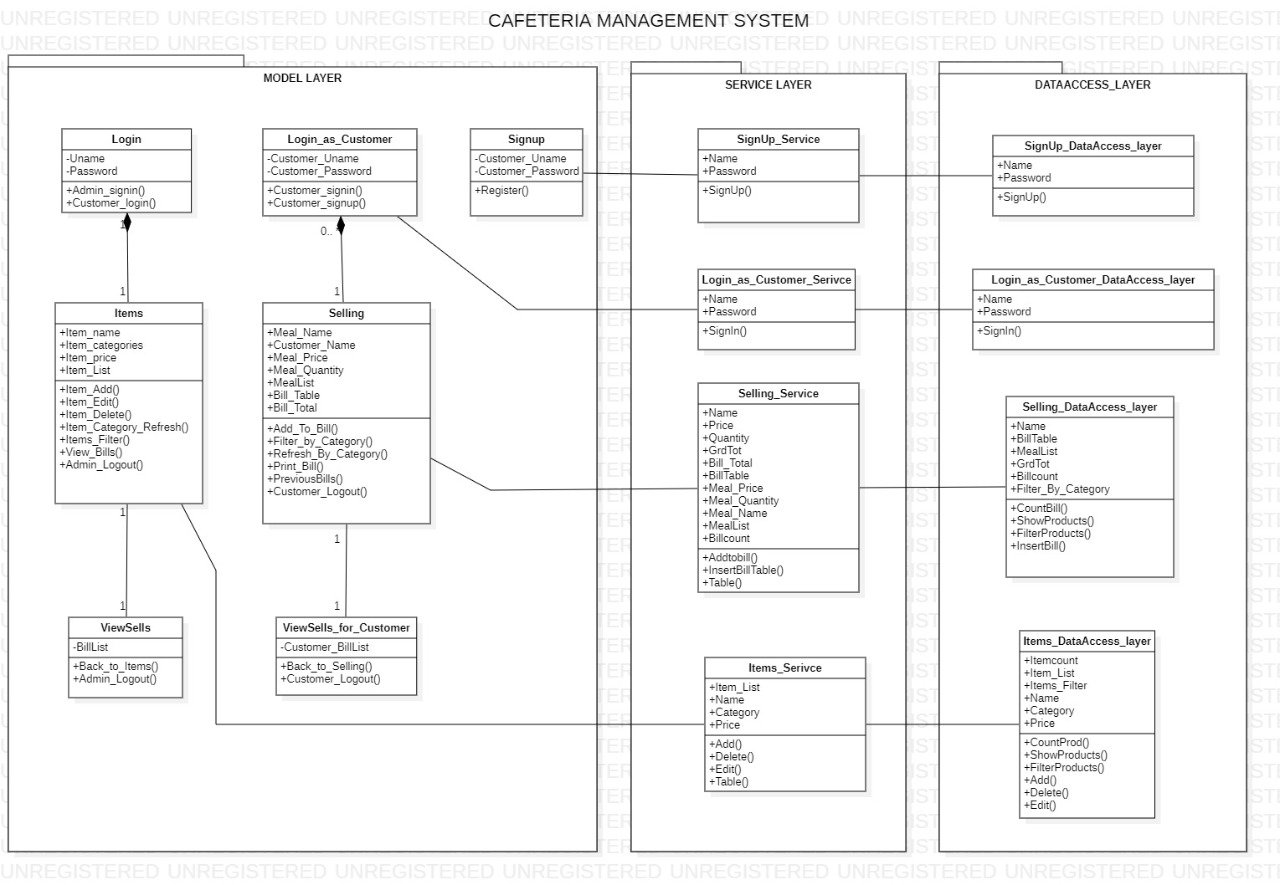
[**References**](#_heading=h.z4o9cmduz4k2) **38**

# Detailed Design through UML diagrams

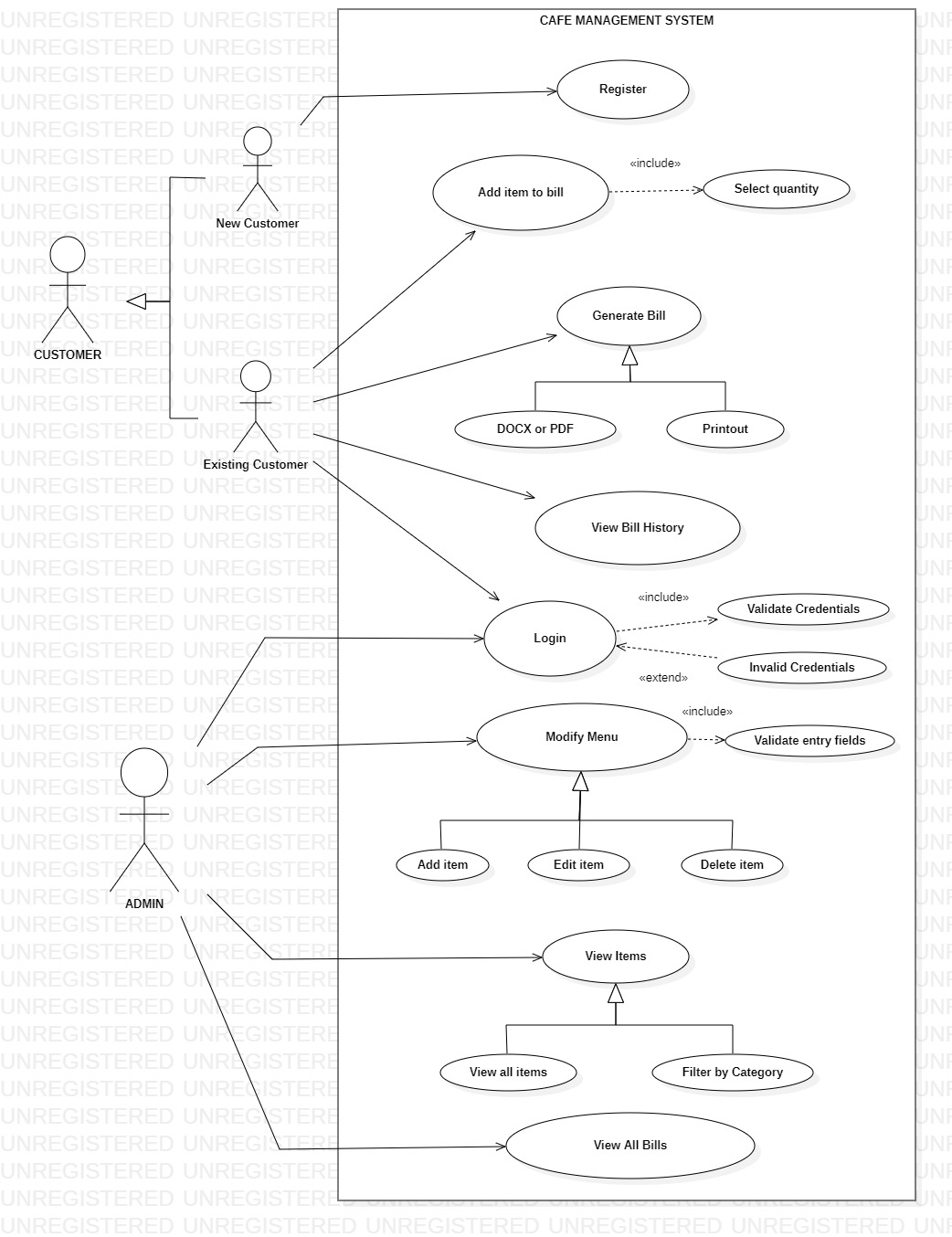
## 1.1 System model using Class Diagram

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.

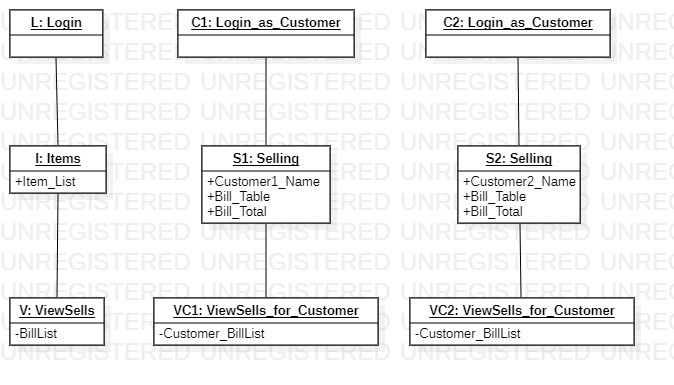
### 1.1.1 Class Diagram

**

## 1.2 Responsibilities - Usecase Diagram



## 1.3 Static snapshot of the system - Object Diagram

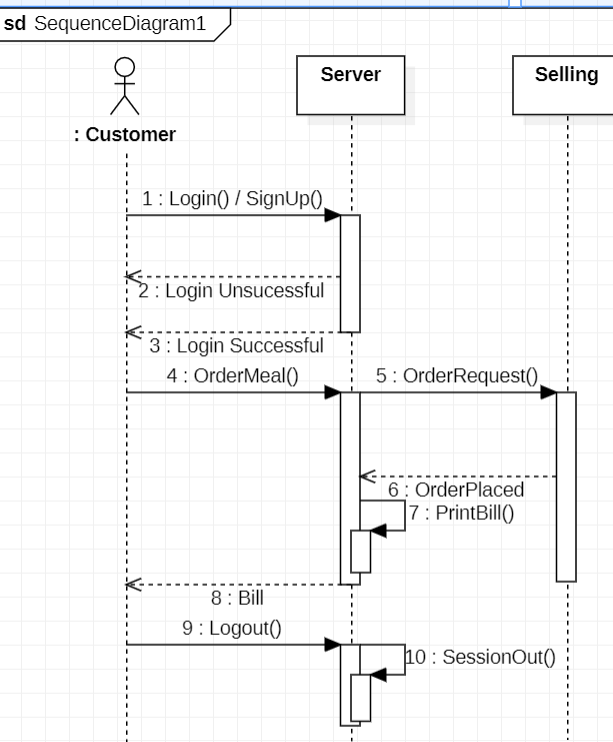
**

## 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 Customer Selling Interaction

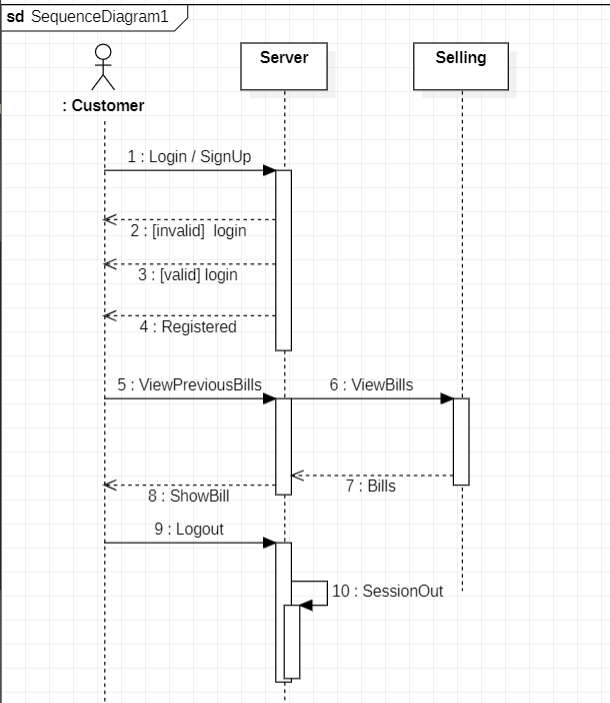
The Actor is Customer and 2 lifelines Server and Selling. First the Customer will either Login or Signup. Then Server will send ack if the Login was successful or not. If login was successful then Customer will order some meal and the request goes to Server. The Server will forward the request to Selling lifeline. The Selling lifeline will send ack to Server when order is placed. The Server will generate a bill and send it to Customer. Now Customer can logout.

**

### 1.4.2 Customer ViewSells Interaction

The Actor is Customer and 2 lifelines Server and ViewSells. First the Customer will either Login or Signup. Then Server will send ack if the Login was successful or not. If login was successful then Customer can make a request to view all his/her previous bills to Server. The Server

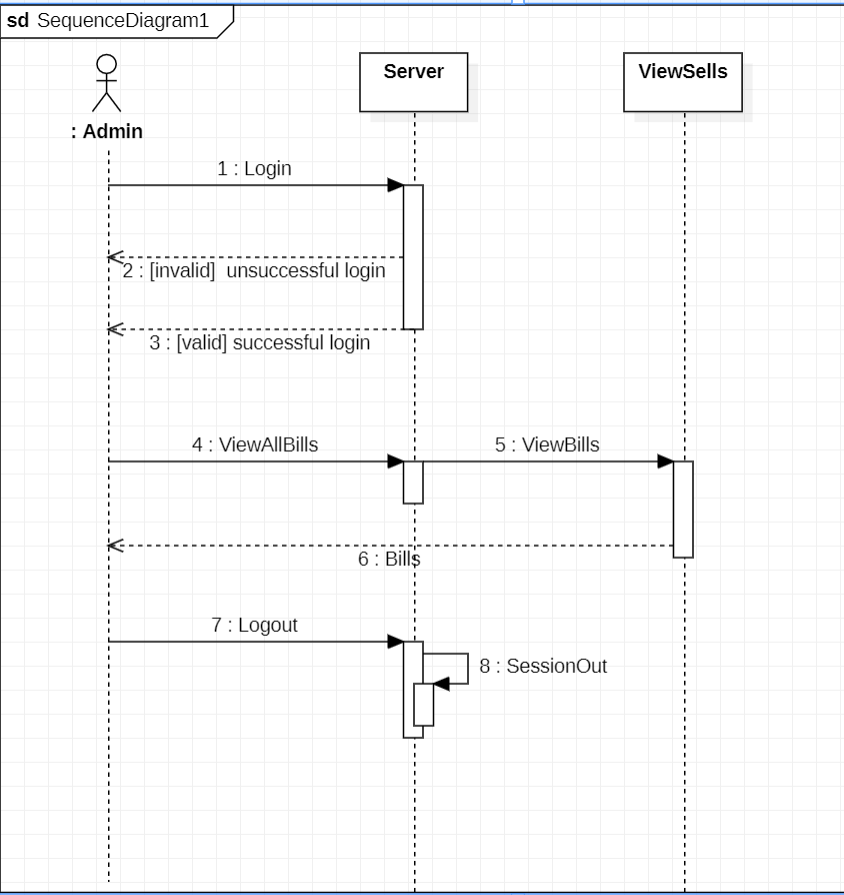
Sends a request to ViewSells. ViewSells will send a response of previous bills to Server. The Server will display the Bills to Customer. Now Customer can logout.

**

### 

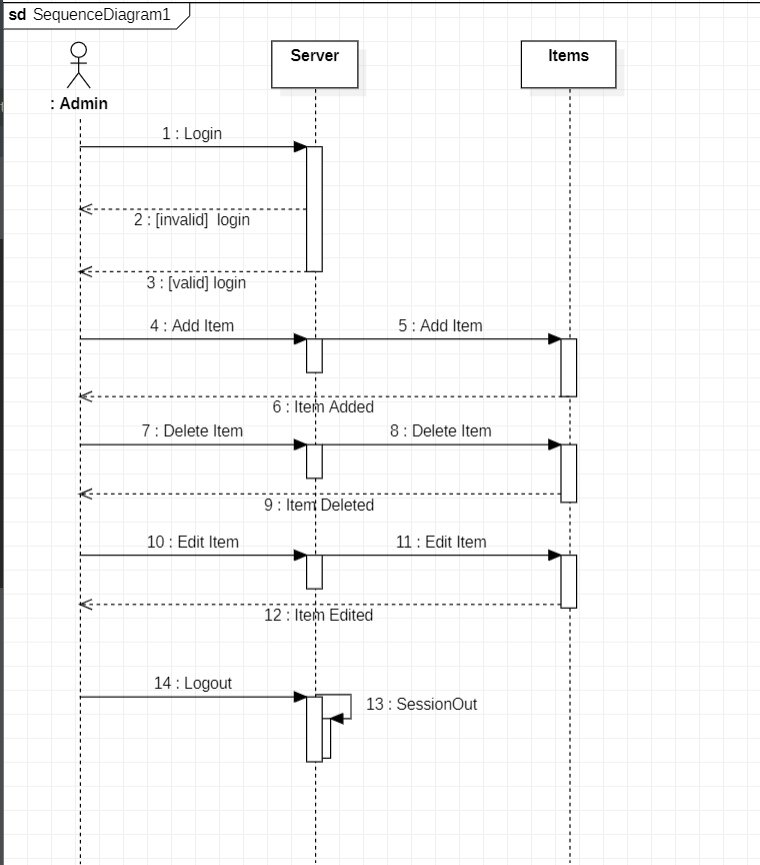
### 1.4.3 Admin ViewSells Interaction

The Actor is Admin. The lifelines in this sequence diagram are Server and ViewSells. The Admin will Login. Then Server will send ack if the Login was successful or not. If login was successful then Admin can make a request to view all previous bills of Customers to Server. The Server sends a request to ViewSells. ViewSells will send a response of previous bills to Admin. Now Admin can logout.



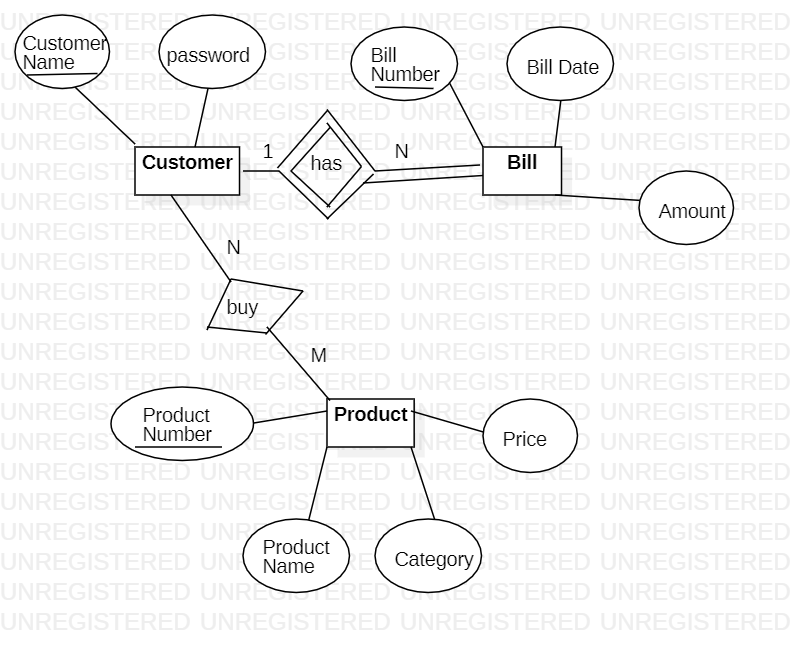
### 1.4.4 Admin Items Interaction

The Actor is Admin. The lifelines in this sequence diagram are Server and Items. The Admin will Login. Then Server will send ack if the Login was successful or not. If login was successful then Admin can Add, Delete or Edit and Item in Menu. Admin can logout after making necessary changes in menu.



# Database Design

## 2.1 ER Diagram



# Implementation Plans

## 3.1 Technology Stack

1. Star UML: Application to create UML diagrams. Used for creating the Use case, Class and Object Diagrams in this project.

2. MS Word: A document editor. Used for editing the SRS submission.

3. Apache NetBeans: It is an open-source integrated development environment (IDE) for developing with Java, PHP, C++, and other programming languages.

4. Java Swing: It is a lightweight Java graphical user interface (GUI) that is used to create various applications. Swing has platform-independent components.

5. MySQL: It is a tool used to manage databases and servers. It is not a database but is widely used in managing and organising data in databases.

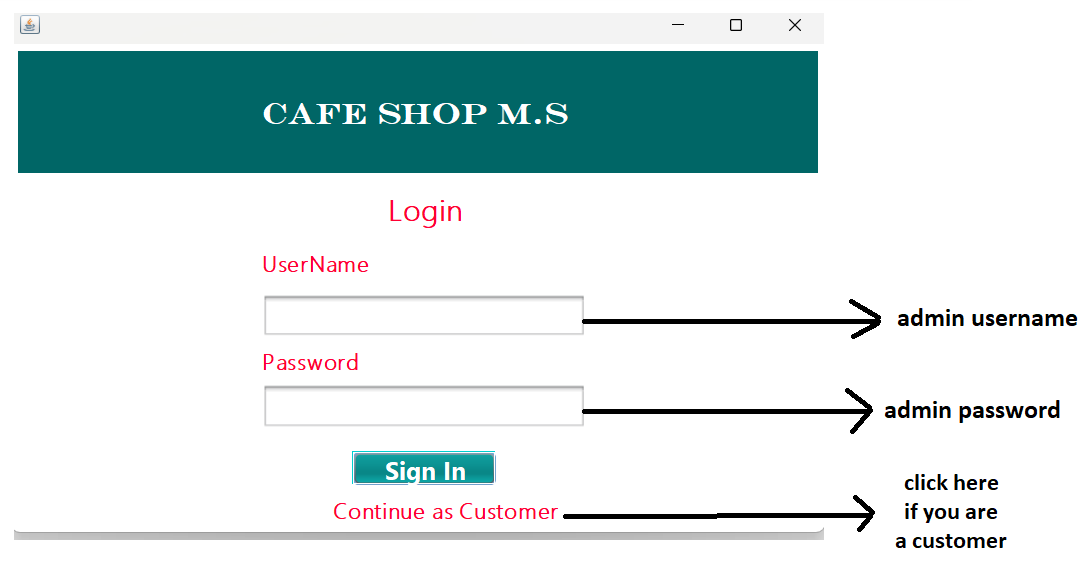
6. MySQL server: It provides a database management system with querying and connectivity capabilities, as well as the ability to have excellent data structure and integration with many different platforms.

7. MySQL workbench: It is a unified visual tool for database architects, developers, and DBAs. Provides data modelling, SQL development, and comprehensive administration tools for server configuration.

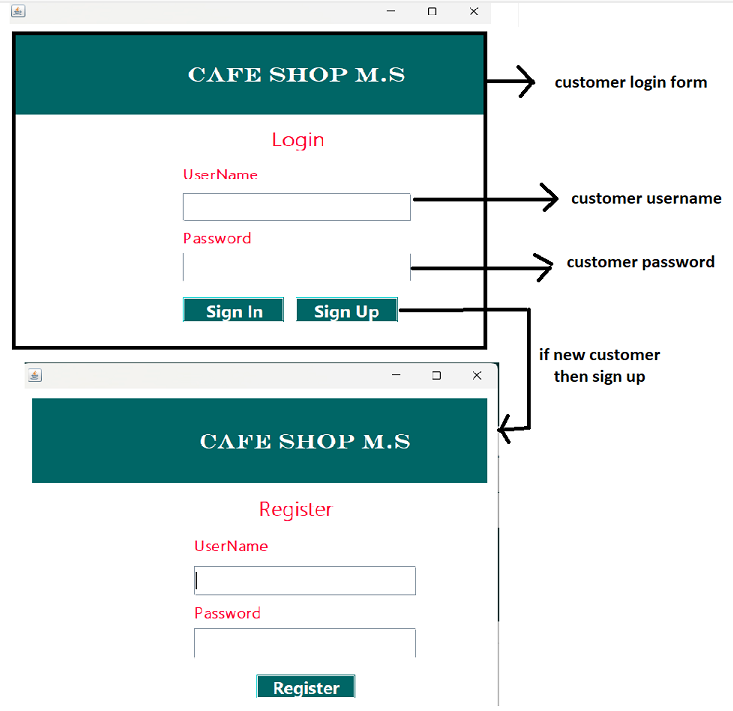
8. JDBC: JDBC is the JavaSoft specification of a standard application programming interface (API) that allows Java programs to access database management systems. The JDBC API consists of a set of interfaces and classes written in the Java programming language.

## 3.2 User Interface Prototyping

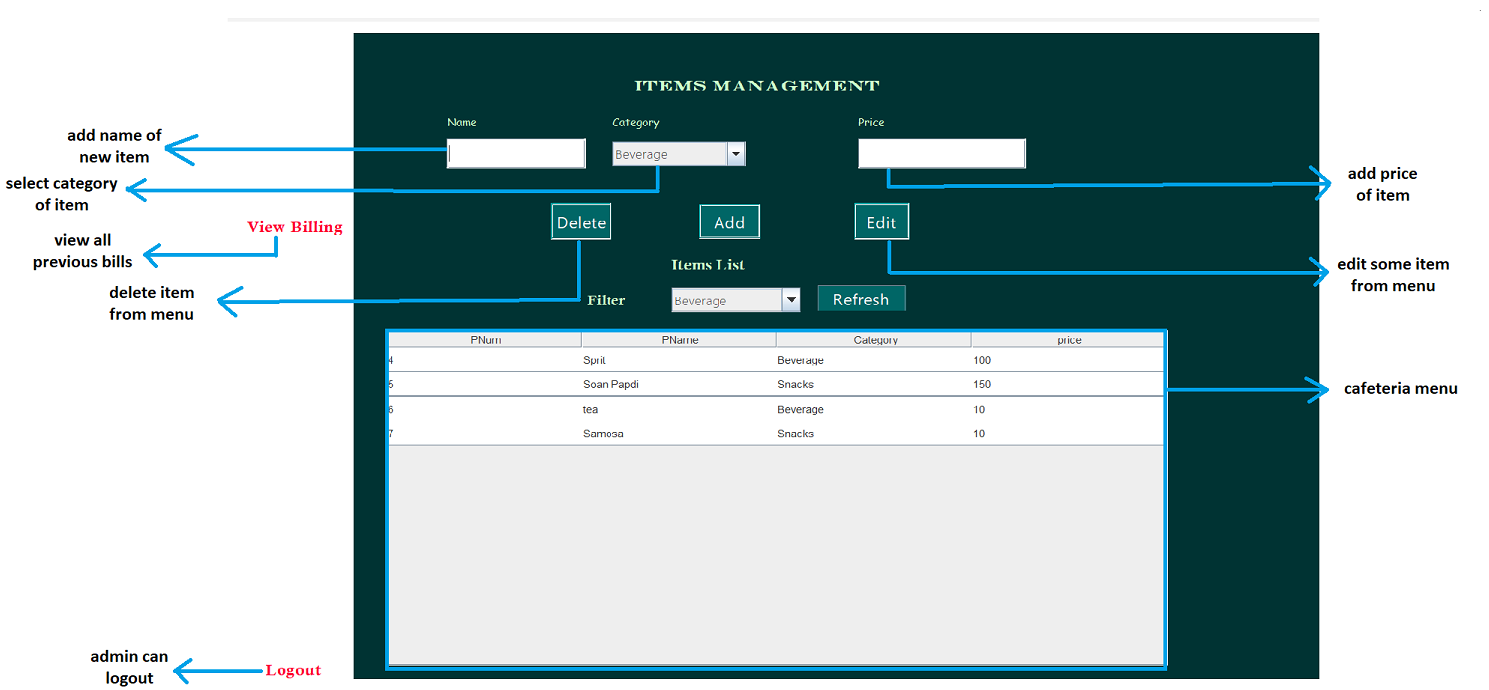
**Admin Login**



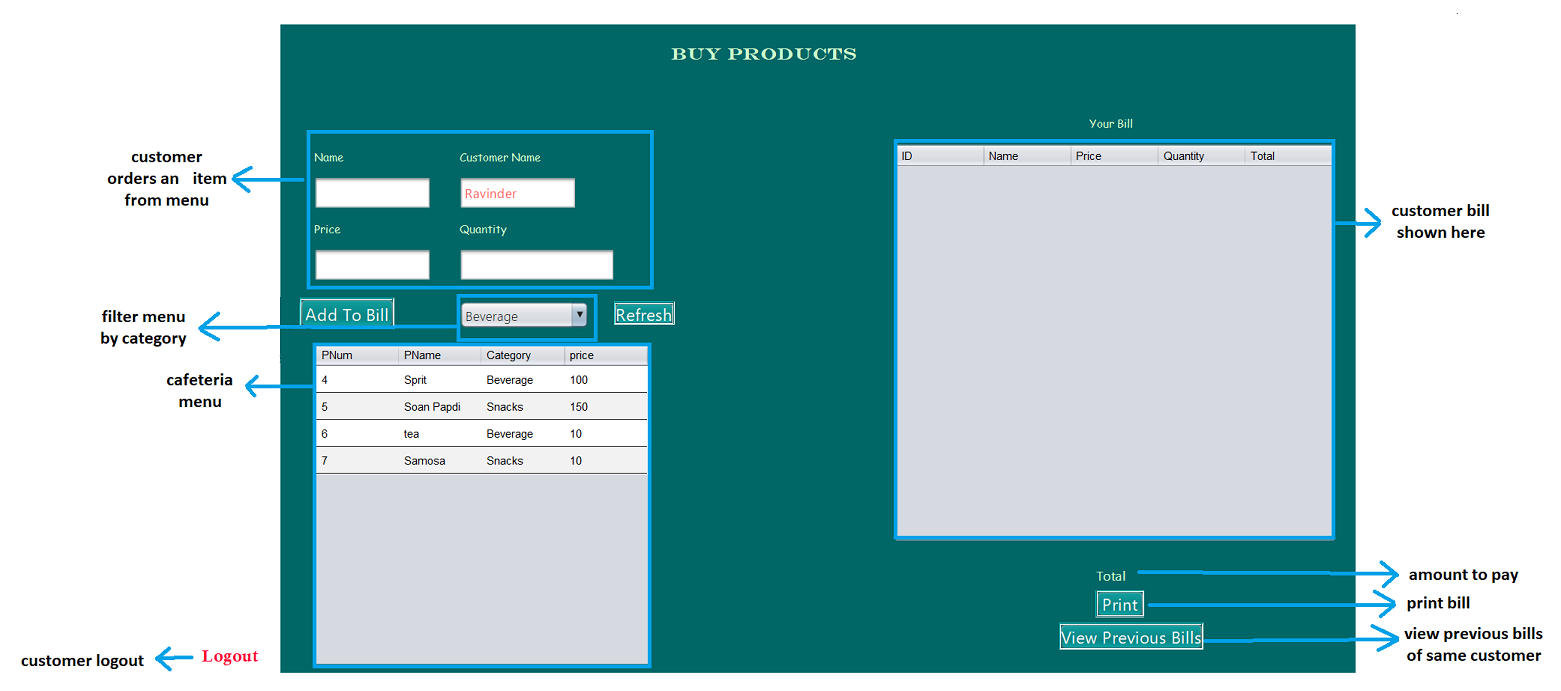
**Customer Login/SignUp**



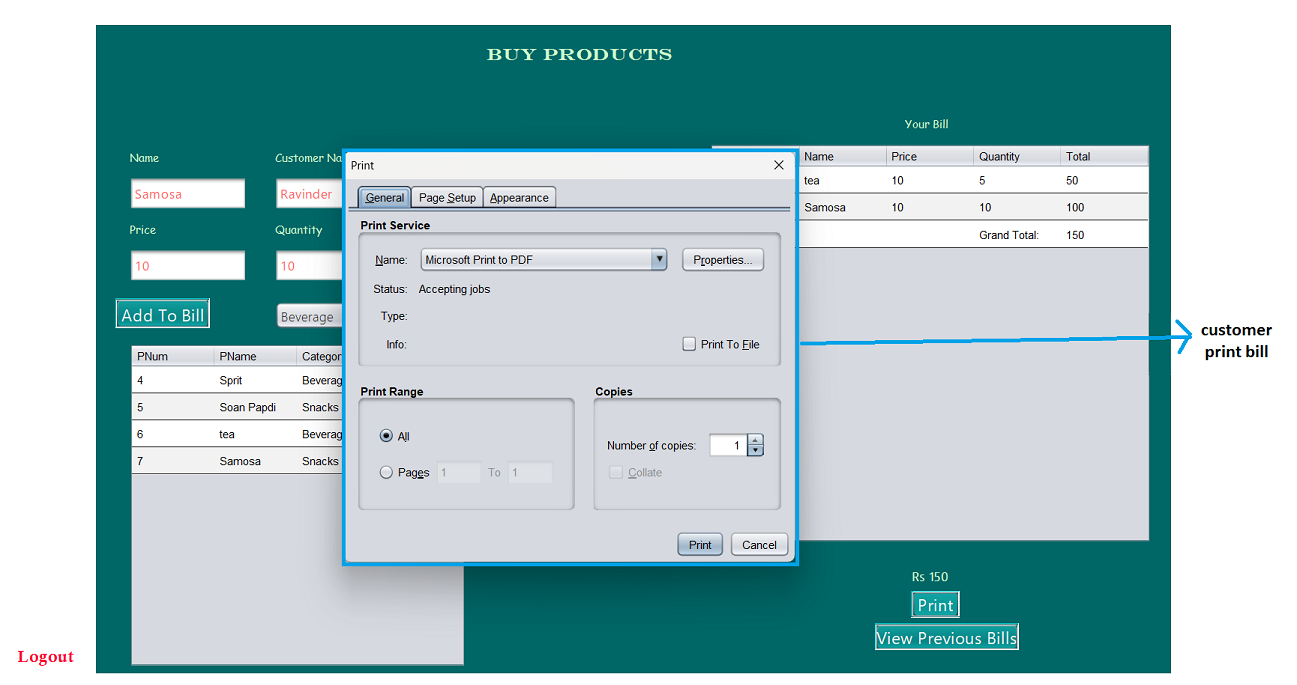
**Item Management Page**



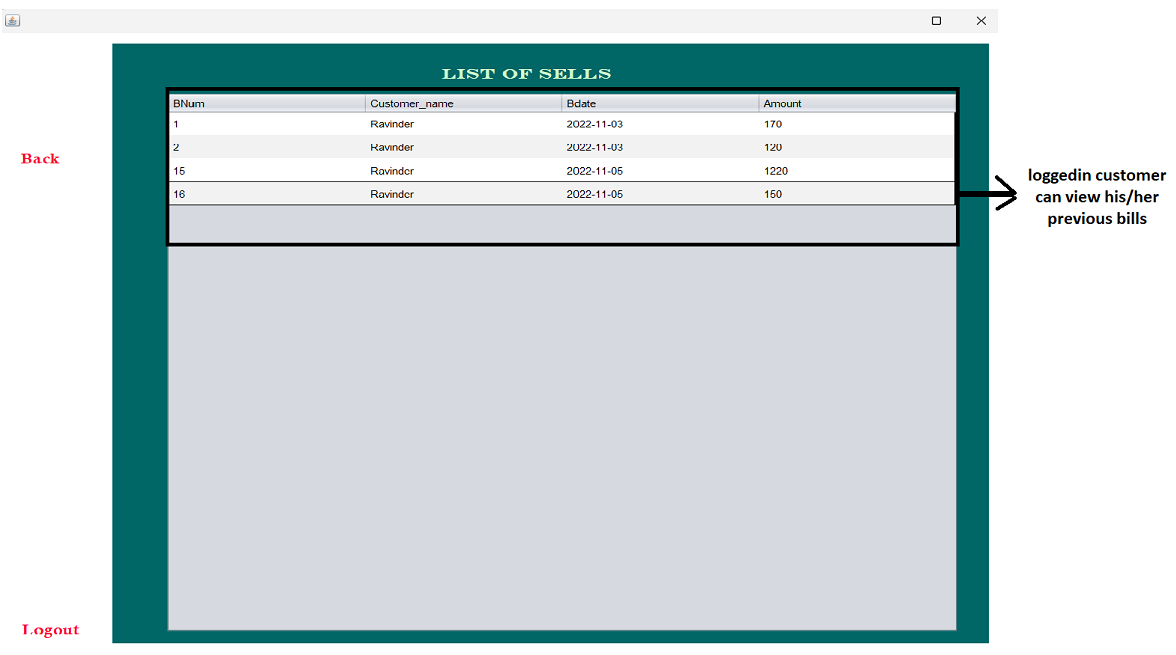
**Order Place Page**



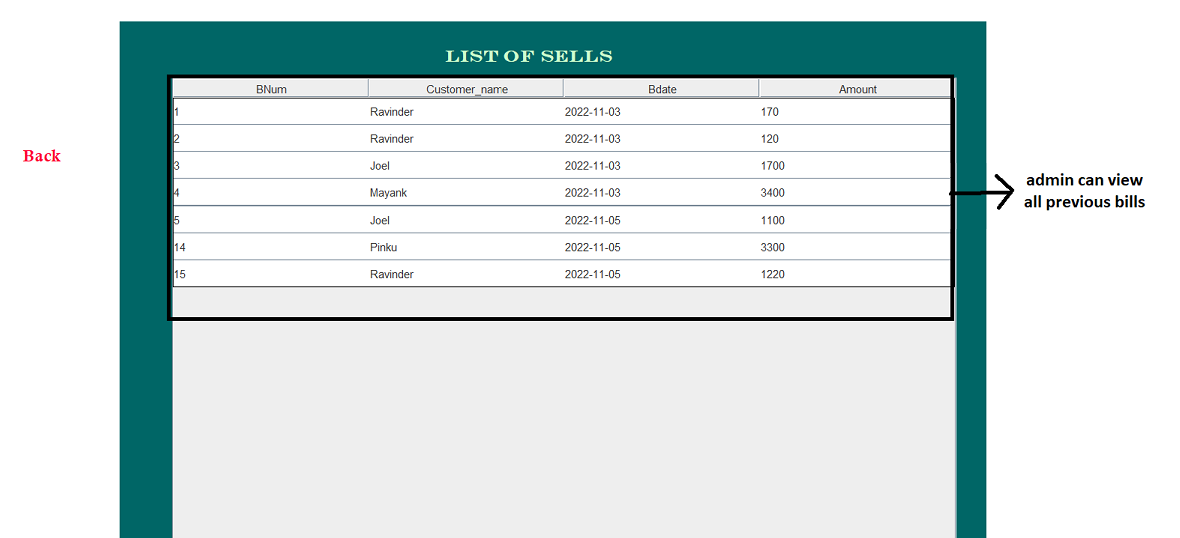
**Bill Printout Page**



**Customer Previous Bills Page**

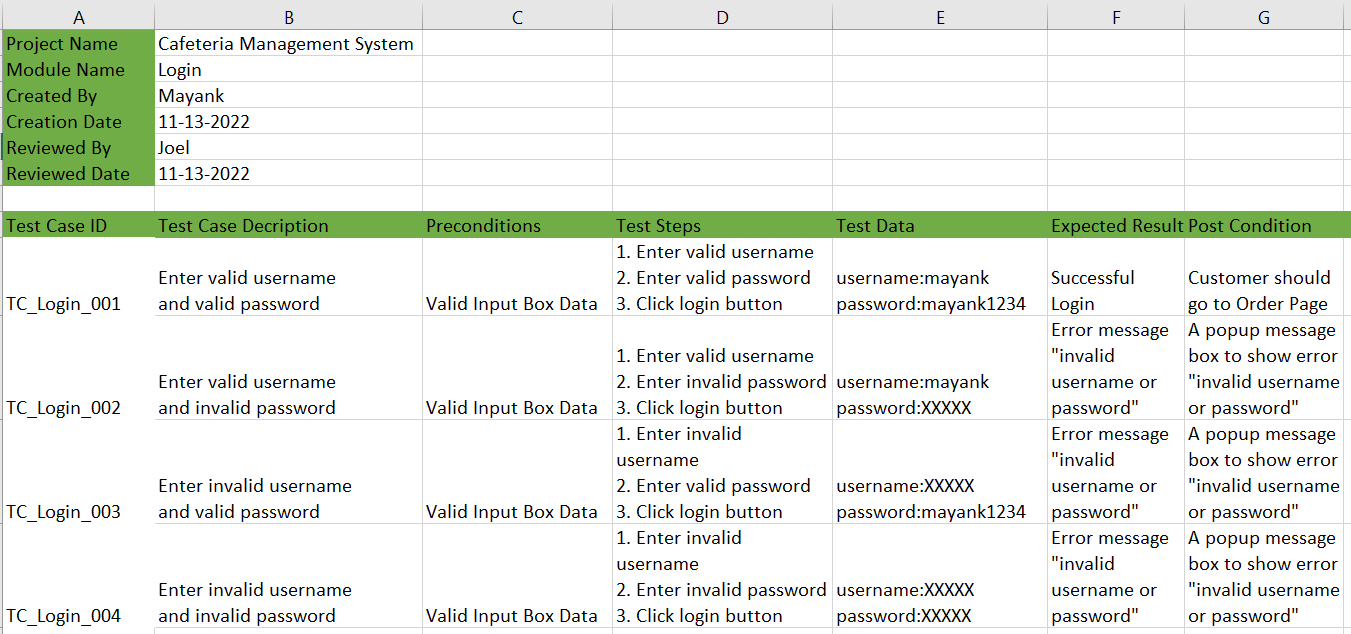


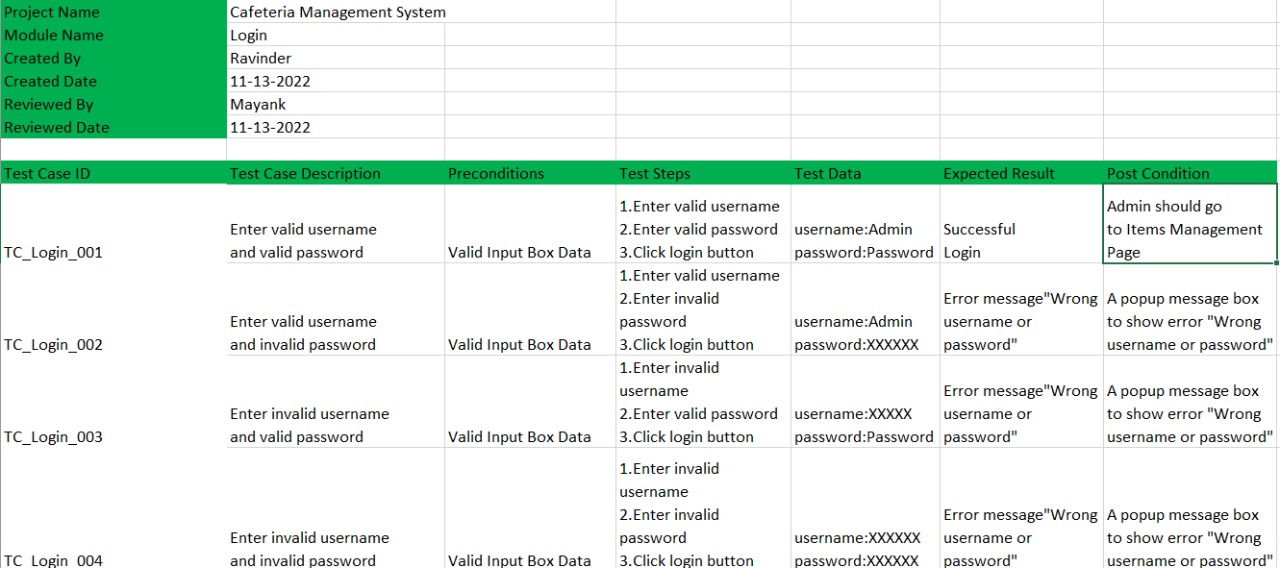
**All Previous Bill Page**



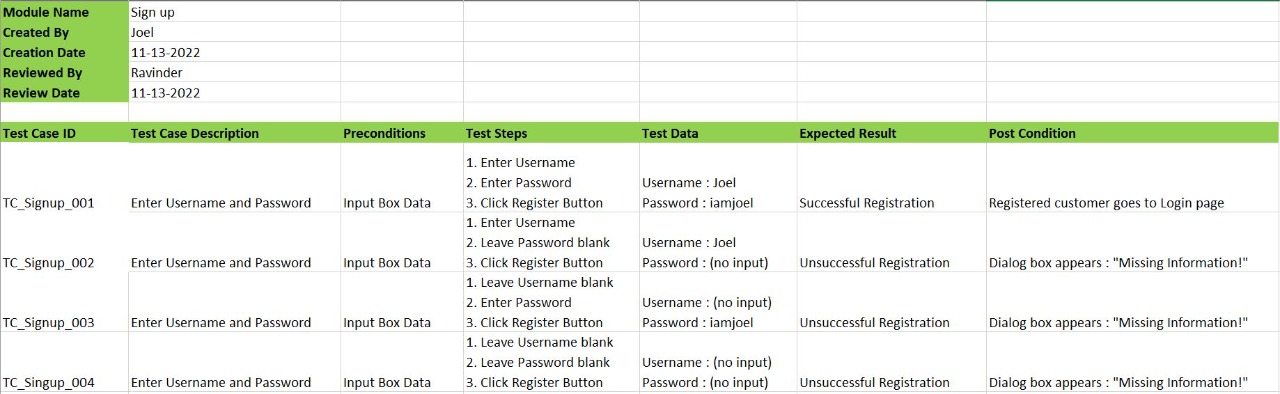
1. Test Cases

### **4.1 Test Case #1**





### **4.2 Test Case #2**



# References

1. [Use Case Diagrams: Tips and FAQs](https://www.andrew.cmu.edu/course/90-754/umlucdfaq.html)

2. [Class Diagrams](https://sparxsystems.com/resources/tutorials/uml2/class-diagram.html)

3. [What is Object Diagram?](https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-object-diagram/)

4. [What is Sequence Diagrams?](https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/)

5. [How to write Test Cases?](https://www.youtube.com/watch?v=g0PrXoWKM2Y)