Software Requirements Specification

for

Shop Inventory Management System

Version 1.2

Prepared by

Group #: 20

Kshitij Kabeer	180366	kshitijkabeer@gmail.com
Rishabh Kothary	180608	rishabhkothary76@gmail.com
Kartavya	180343	kartavya4301@gmail.com
Pravar Deep Singh	160508	pravardeepsingh@gmail.com

Course:	CS253
Mentor TA:	*Swastik Maiti*
Date:	27th April 2022

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.1 Initial Draft	Kshitij Kabeer Rishabh Kothary Kartavya	First draft.	30.01.2022
1.2 Second Major Draft	Kshitij Kabeer, Rishabh Kothary	Revision which reflects the major changes in dependencies and software frameworks used	27.04.2022

1 Introduction

1.1 Product Scope

This product is meant for small retailers and shop owners and will help them effectively manage their inventory, calculate profit and loss, and also suggest optimal restocking amounts. The software will also get sales data directly from the cash registers without the need for manual entering the amount of each item sold. The software will also suggest optimal restocking amounts for each product, on the basis of how much capital the shop owner has, for restocking. The software can be separately accessed by both the administrators and the employees manning the cash registers.

1.2 Intended Audience and Document Overview

Shopkeepers and small retailers who need a software to efficiently manage their inventory and want to use a software that will seamlessly update the database as the items are sold or restocked.

1.3 Definitions, Acronyms and Abbreviations

1.4 Document Conventions

No specific conventions employed so far

1.5 References and Acknowledgments

1) *https://www.aleksandrhovhannisyan.com/blog/finite-state-machine-fsm-tutorial-implementing-an-fsm-in-c/*
2) *https://www.tutorialspoint.com/struts_2/basic_mvc_architecture.htm*
3) Colin Lewis. Demand forecasting and inventory control. Routledge, 2012.

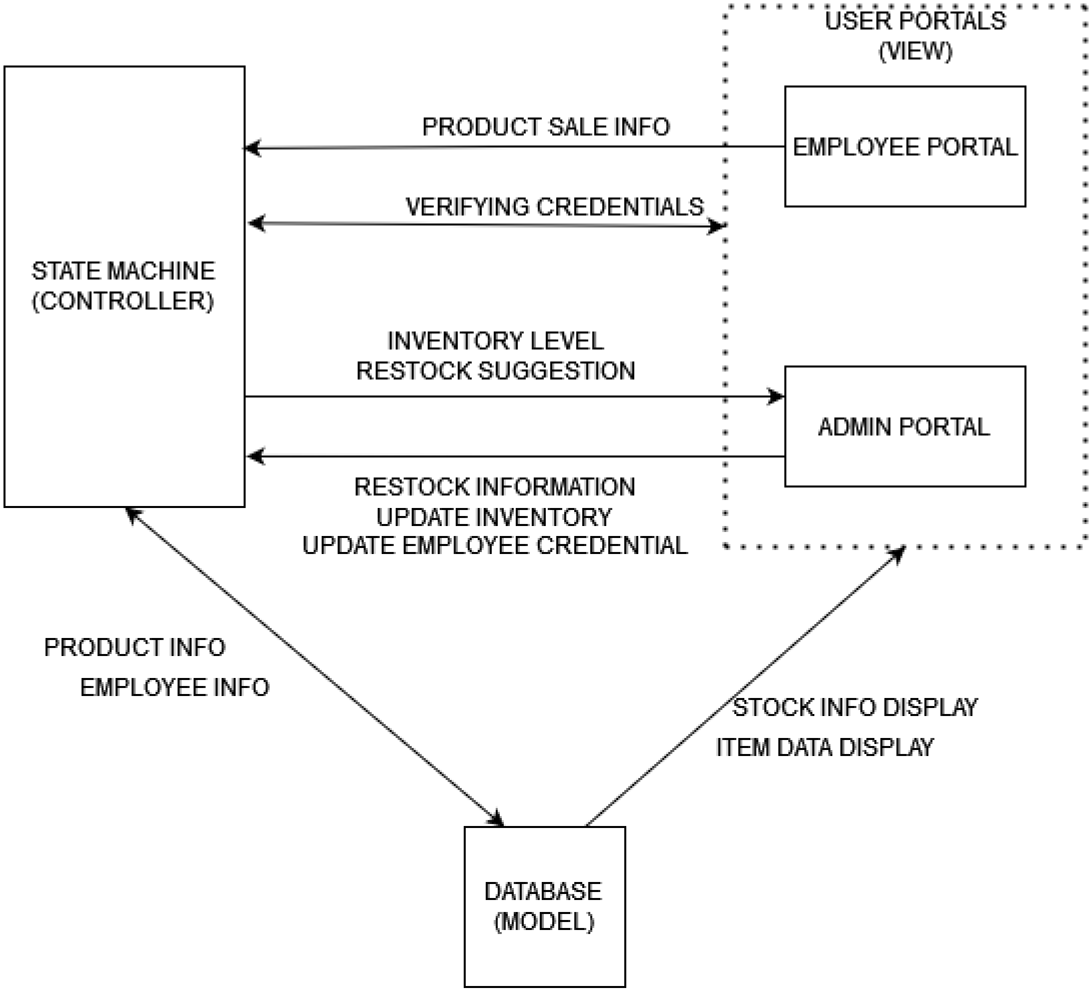
4) Gerrit K Janssens and Katrien M Ramaekers. A linear programming formulation for an inven-tory management decision problem with a service constraint. Expert Systems with Applications, 38(7):7929–7934, 2011.

2 Overall Description

2.1 Product Overview

The product is a self-contained product to help small retailers manage their inventory, and help the admin keep track of employee performance. For a better understanding refer to the diagram below . The architecture used is Model-View-Controller Architecture

(Context Diagram changed from initial draft)



2.2 Product Functionality

1) *Store stock information for all products in the shop in the database.*
2) *Generate invoices for customers and update the database accordingly*
3) *Recommend what stock of each item to get , provided the capital available for restocking*

2.3 Design and Implementation Constraints

Only SQL databases can be used, No-SQL databases would require significant rewrite of the code.

Qt 6 is being currently used for the frontend. However, it is possible to work with Qt5 and earlier versions as well, with minimal changes in the CMakeLists.txt. For getting restocking suggestions you need google or-tools library,

2.4 Assumptions and Dependencies

1) The recommendations made by the software are based only on the previous demand of the product in the shop. It does not take into account any external factors.
2) For smooth working of the software, we assume that the admin diligently notifies the software of any restocks.
3) The software will suggest how much of each product to restock, by solving a constrained optimization problem taking into account the capital available for restocking, the buying price, selling price and storage costs.
4) We are ignoring secondary costs like transportation cost, electricity cost, etc. while calculating the profit or loss.
6) We are also assuming that there isn't going to be any abrupt shutdowns of the system due to uncontrollable circumstances like power cuts. We assume the shopkeeper has already put measures to prevent this.

Dependencies:

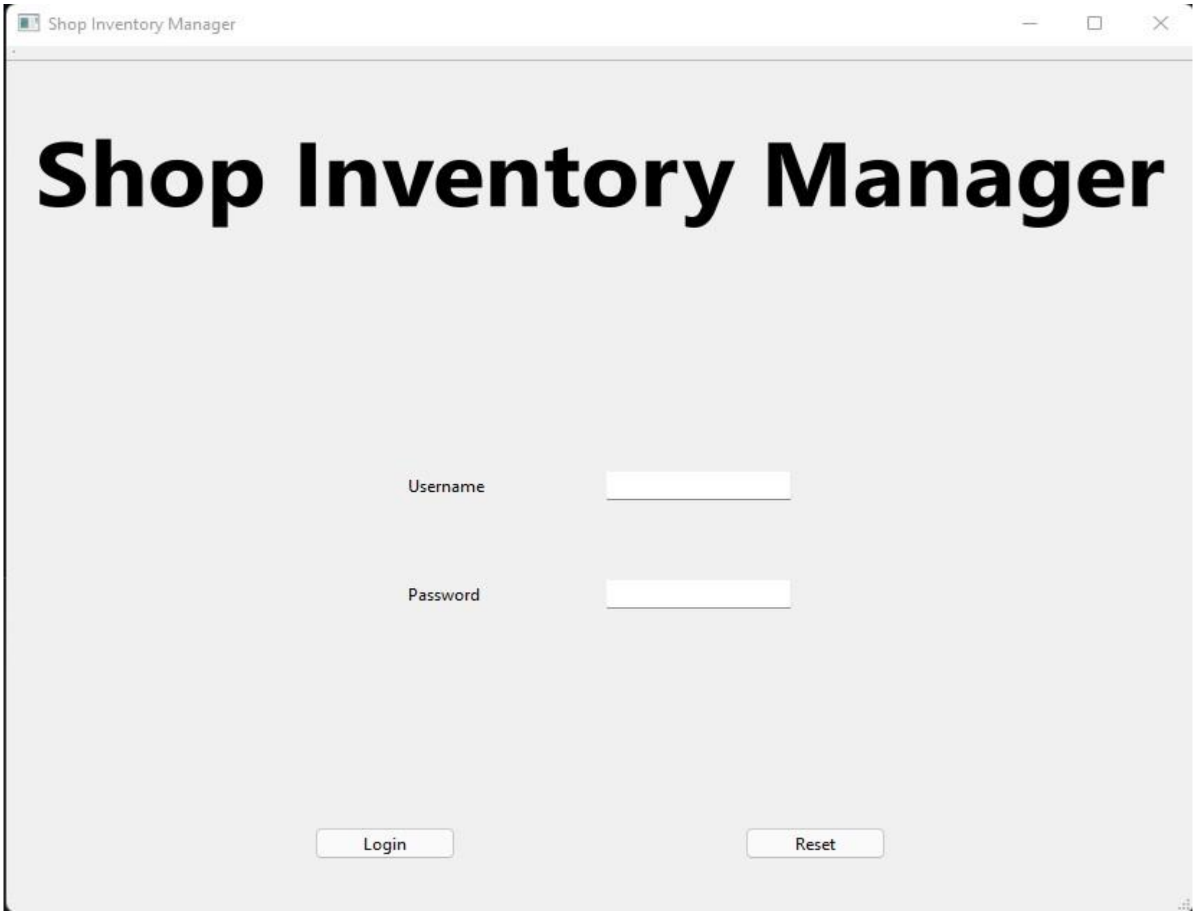
1) *End User has Windows operating system*
2) *End User has Qt installed on their computer.*

3 Specific Requirements

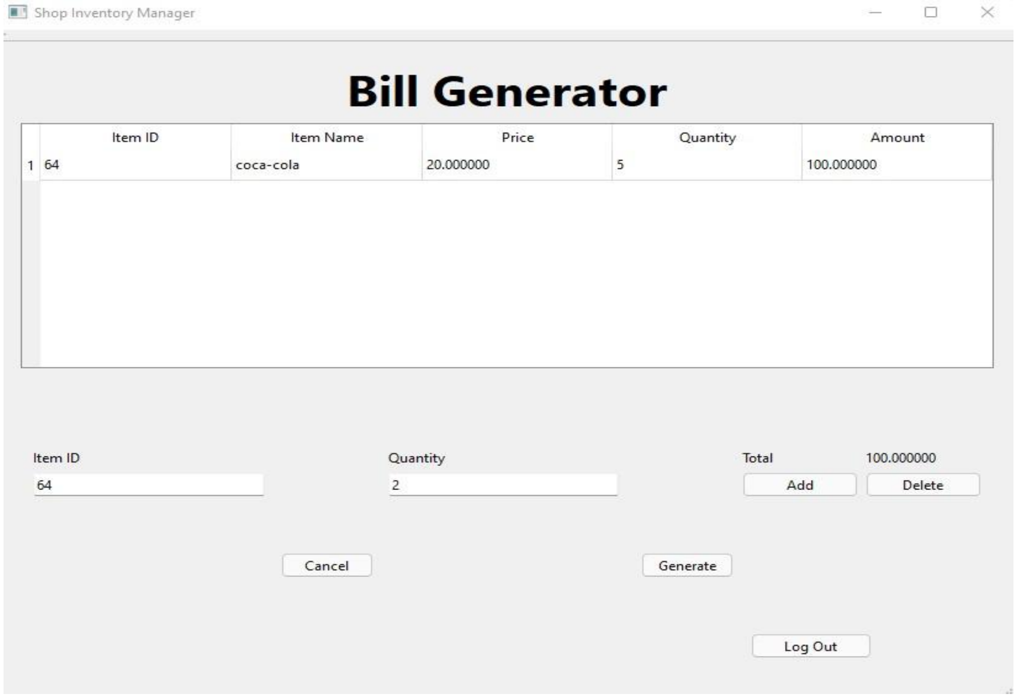
3.1 External Interface Requirements

3.1.1 User Interfaces

(Diagrams Changed from initial draft)


Shop Inventory Manager
Shop Inventory Manager
Username
Password
Login
Reset

S

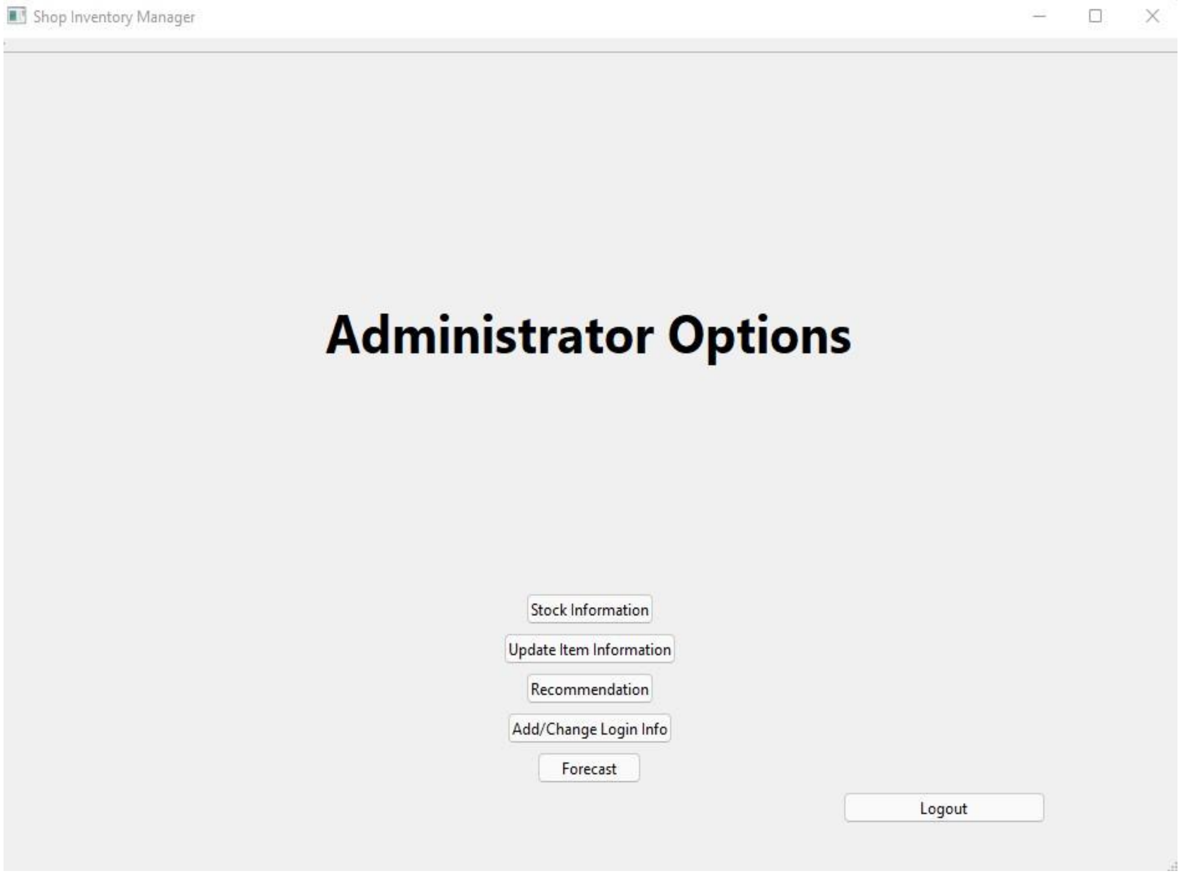


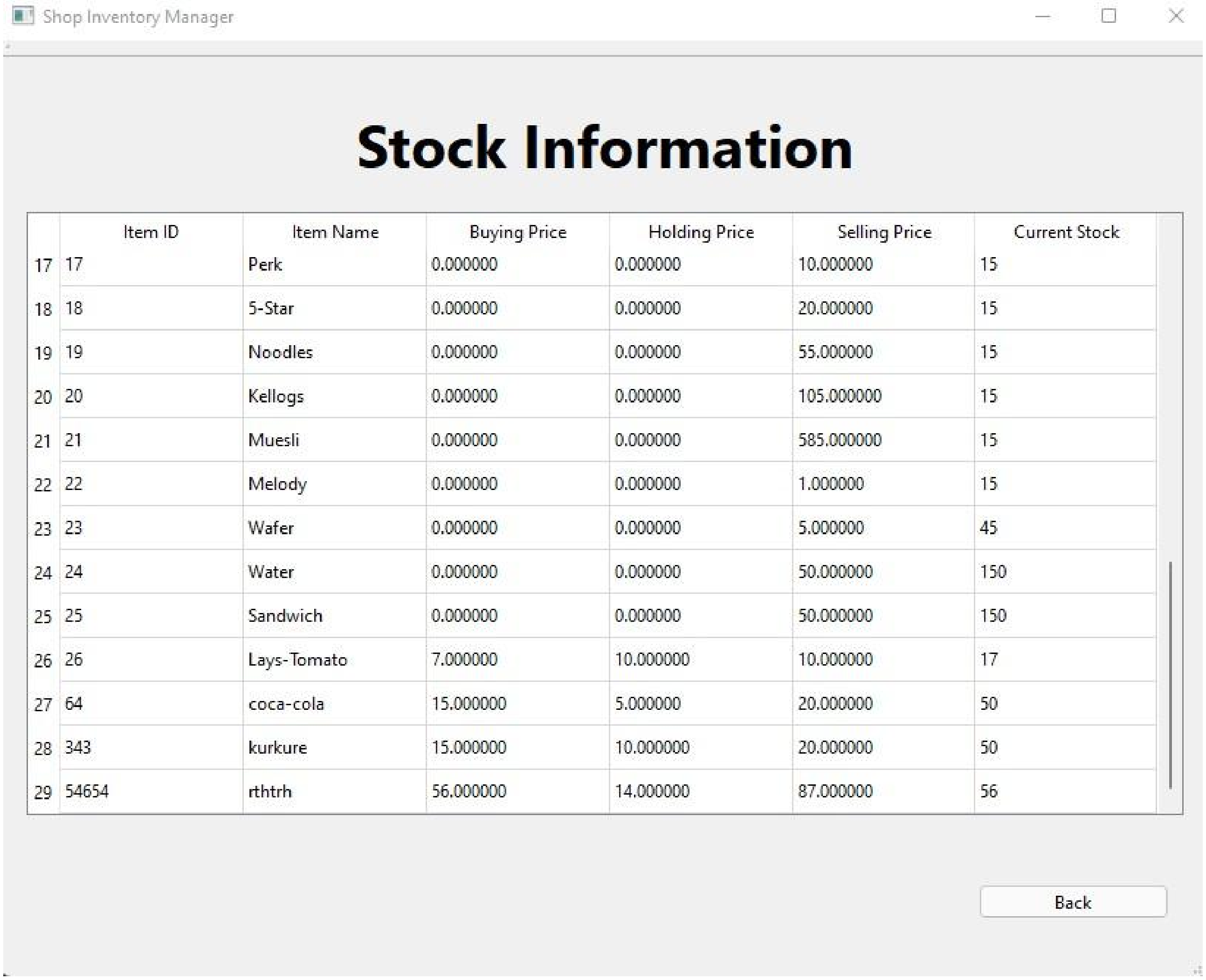
	Item ID	Item Name	Price	Quantity	Amount
1	64	coca-cola	20.000000	5	100.000000

Item ID: 64

Quantity: 2

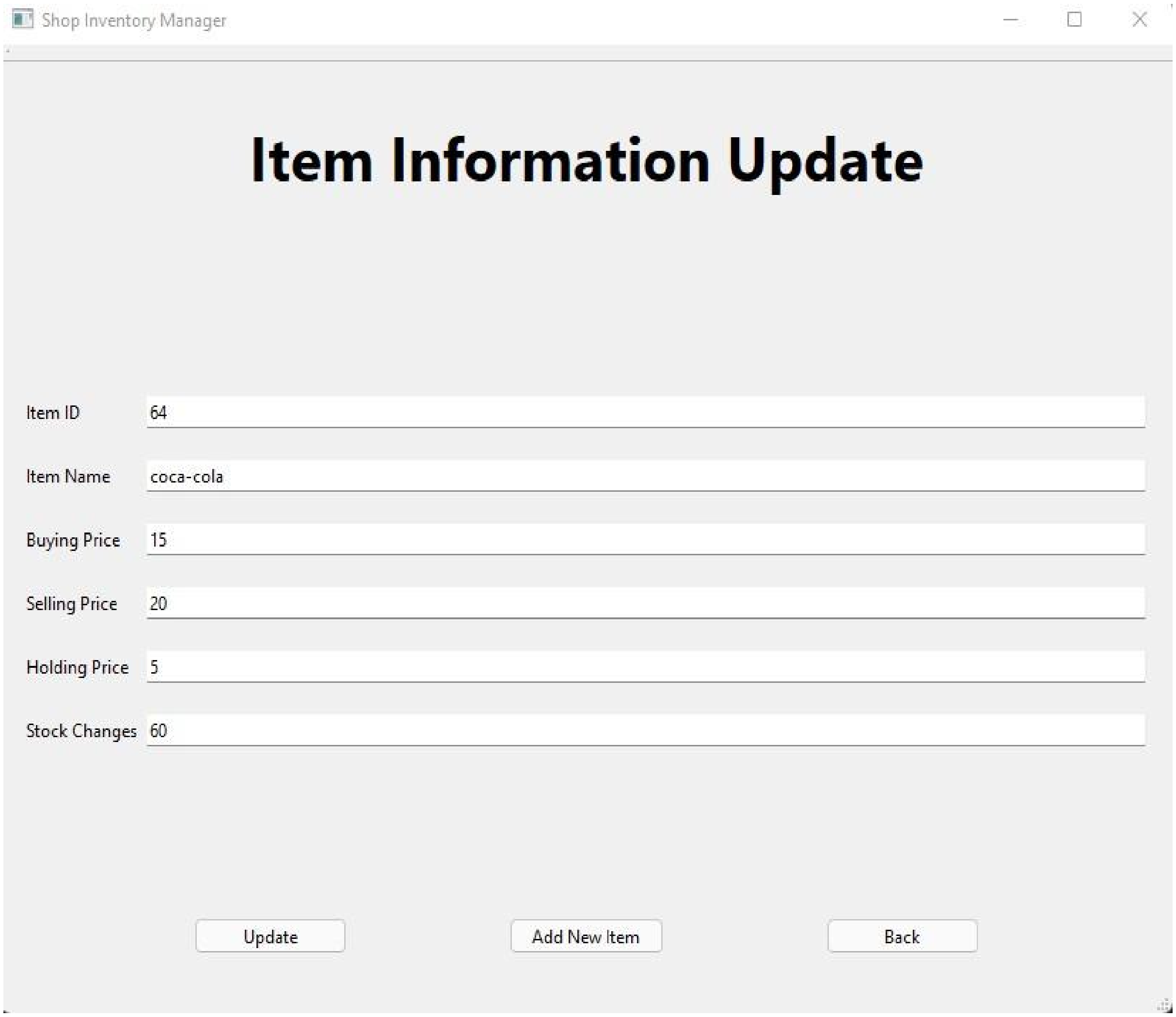
Total 100.000000





Stock Information

	Item ID	Item Name	Buying Price	Holding Price	Selling Price	Current Stock
17	17	Perk	0.000000	0.000000	10.000000	15
18	18	5-Star	0.000000	0.000000	20.000000	15
19	19	Noodles	0.000000	0.000000	55.000000	15
20	20	Kellogs	0.000000	0.000000	105.000000	15
21	21	Muesli	0.000000	0.000000	585.000000	15
22	22	Melody	0.000000	0.000000	1.000000	15
23	23	Wafer	0.000000	0.000000	5.000000	45
24	24	Water	0.000000	0.000000	50.000000	150
25	25	Sandwich	0.000000	0.000000	50.000000	150
26	26	Lays-Tomato	7.000000	10.000000	10.000000	17
27	64	coca-cola	15.000000	5.000000	20.000000	50
28	343	kurkure	15.000000	10.000000	20.000000	50
29	54654	rthtrh	56.000000	14.000000	87.000000	56



Item Information Update

Item ID 64

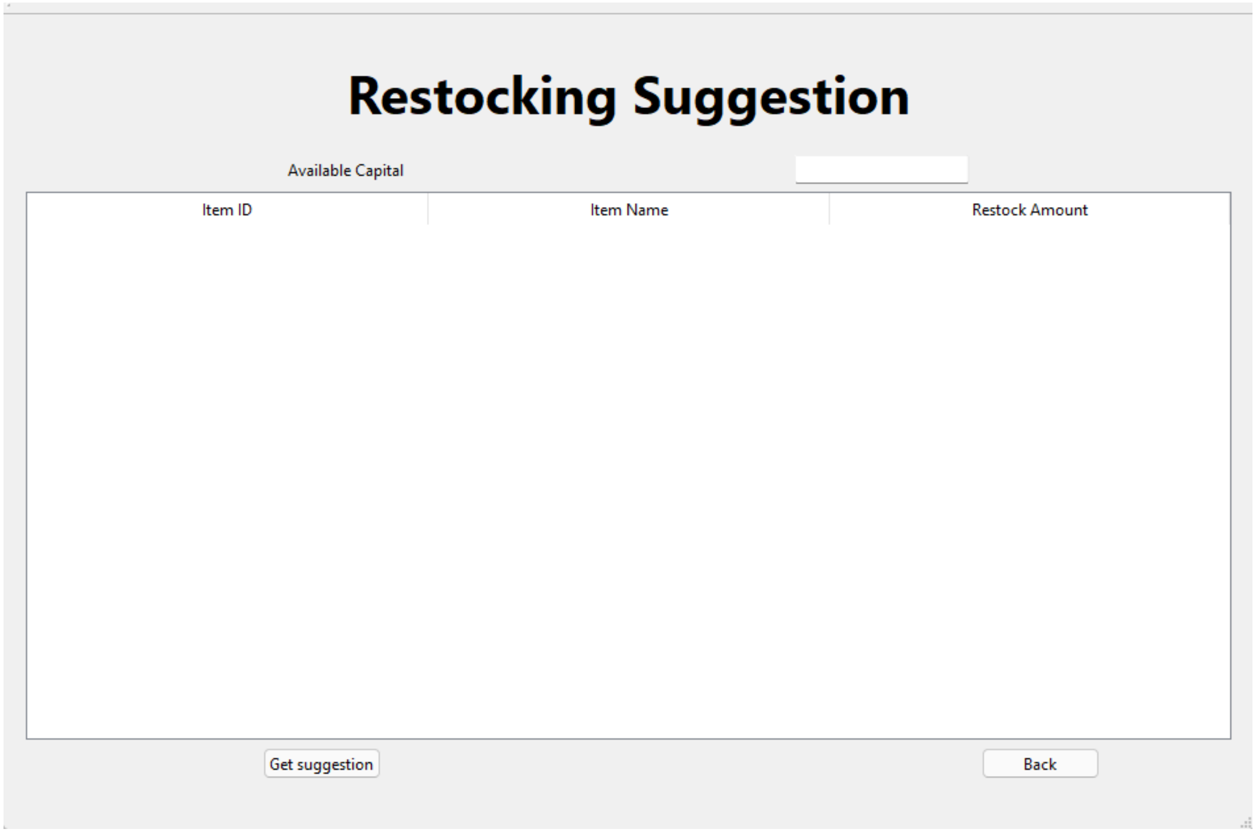
Item Name coca-cola

Buying Price 15

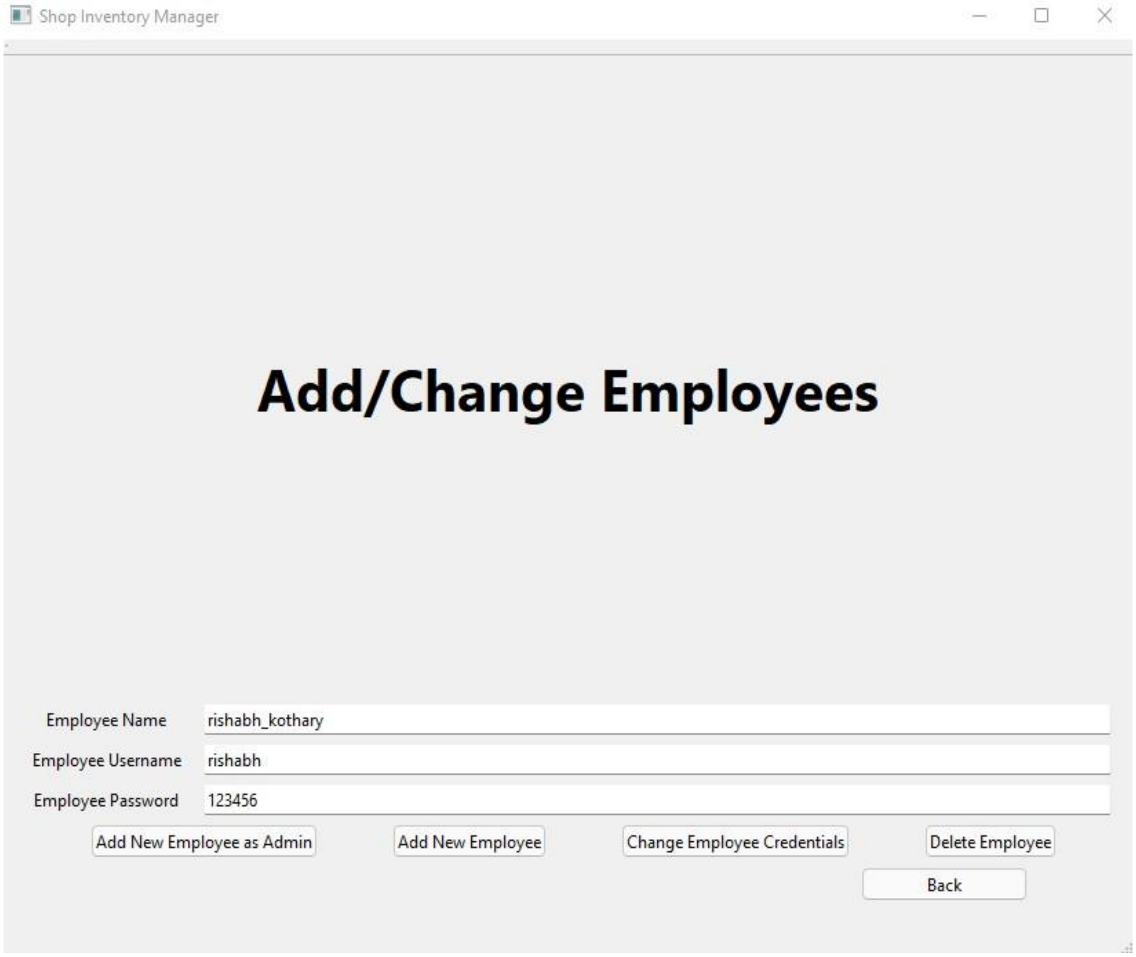
Selling Price 20

Holding Price 5

Stock Changes 60



Item ID	Item Name	Restock Amount



Shop Inventory Manager

Add/Change Employees

Employee Name rishabh_kothary

Employee Username rishabh

Employee Password 123456

3.1.2 Hardware Interfaces

No specific hardware interface is required.

3.1.3 Software Interfaces

The front end interface visible to the users will run on any up-to-date web browser. The data will be sent through LAN connections to the server managing the database and recommendation system

3.2 Functional Requirements

3.2.1 F1: *Employee portal for scanning items and storing information in the database: This portal should allow the employee to scan items by entering the item ID. The software will maintain a list of items being scanned, and report the total amount that has to be paid by the customer. The software automates the process of updating the database if items sold or restocked directly from the employee portal or admin portal.*

3.2.2 F2: *Giving restocking reminders for products whose stock is low: As the stock level reaches reorder point, the system gives a recommendation on how much to restock for every item. This is done through a constrained optimization depending on the capital available to restock, buying price, selling price and storage cost.*

3.3 Use Case Model

3.3.1 U1: Items Invoice

Author - Kshitij Kabeer

Purpose - Calculating invoice of items sold to each customer and calculate payable amount

Requirements Traceability –

1) *Employee portal for scanning items*

2) Update the database of items as products are sold or restocked

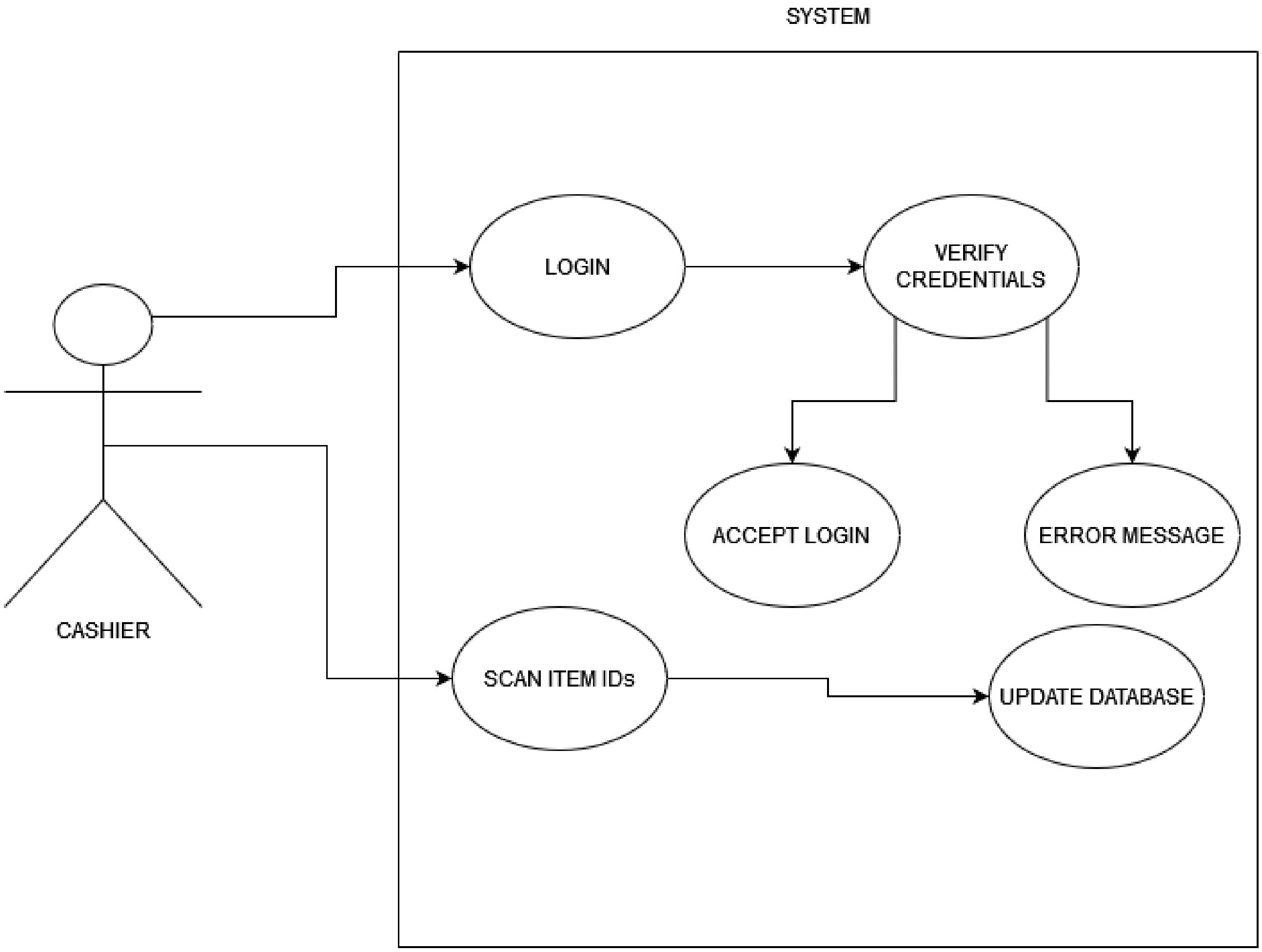
Priority - High

Preconditions - Proper cost price and sale price for each item is given, also

Post conditions - The stock of the items sold to the customer is updated in the database.

Actors – Employee manning the cash register

Exceptions - If the stock of the item reduces to 0, then exception will occur (only possible if there is a human error in entering the amount restocked or sold)



3.3.2 U2: Restocking Function

i) **Author** – Rishabh Kothary

Purpose - Restocking reminder and updates

Requirements Traceability –

1) *Update the database of items as products are sold or restocked*

2) Giving restocking reminders for products whose stock is low

Priority - Medium

Preconditions - For suggesting the optimal restocking amounts, the software should have the previous sale data for at least 1 day. Admin has to press the forecast button everyday at the end of day for the optimization tool to work properly.

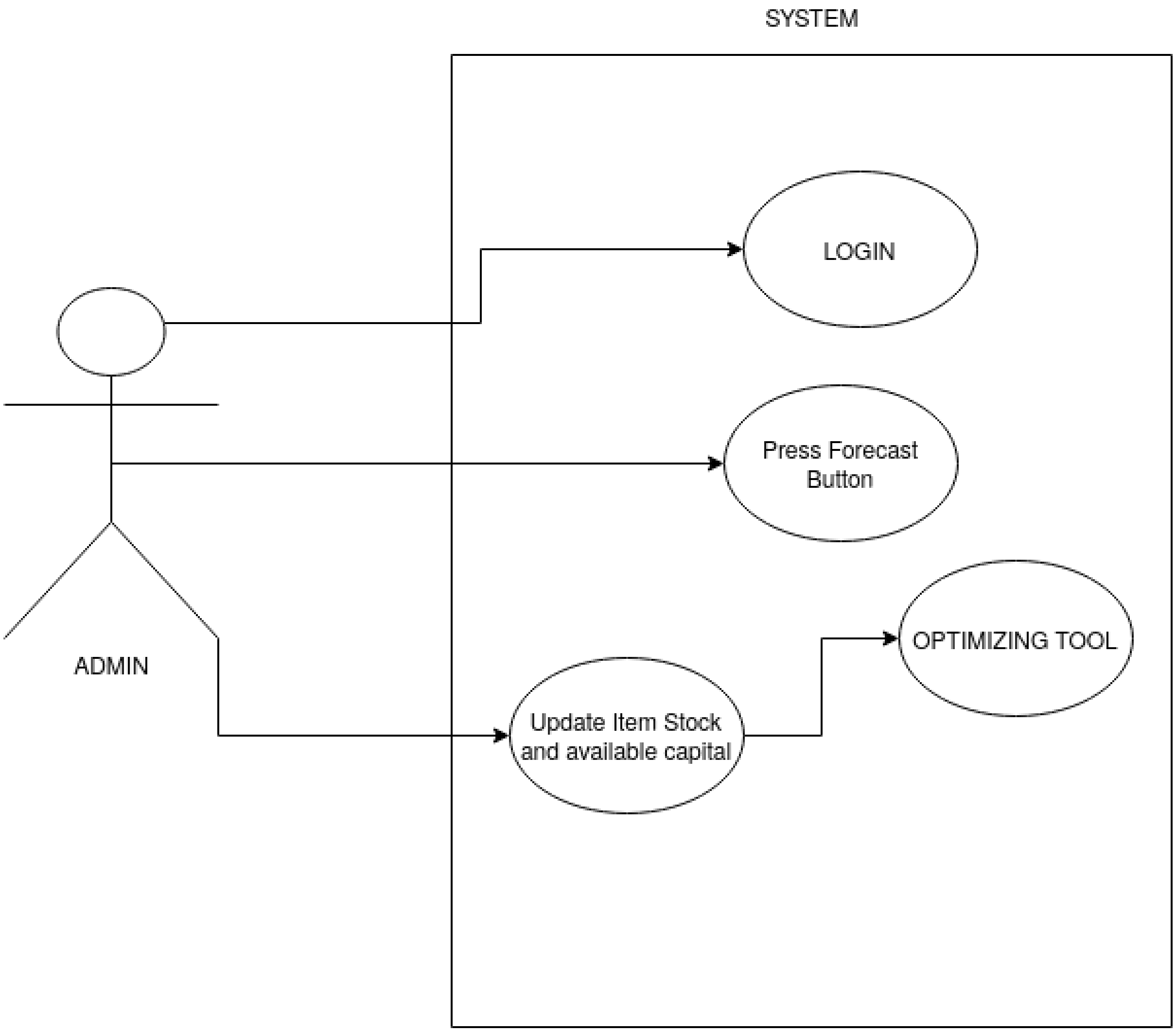
Post conditions - The software will suggest when to restock and also the optimal restocking amounts for each product when the stock gets low. Additionally, it will update the database when new stock arrives.

Actors – The admin has to enter the restock amounts as and when they arrive.

Exceptions - If the amount restock is entered incorrectly, it might cause issues when the product is sold, and also mess up the restocking reminder.

(Updated Use Case Diagram)

Use Case : Restocking Recommendations



4 Other Non-functional Requirements

4.1 Performance Requirements

No performance requirements

4.2 Safety and Security Requirements

For login security we have provided a unique username and password for every employee which is stored in the database. We have provided different login interfaces for admin and employee so that private information about the retailer isn't released out. We have no safety requirements.

4.3 Software Quality Attributes

4.3.1 Usability

The software has a simple and easy interface although can be significantly improved over time. The software manual is clearly and concisely written explaining each functionality in depth.

4.3.2 Reusability

The software well tested stable library Google OR-Tools for optimization, Qt for frontend and CMake to build the software.

Appendix A – Data Dictionary

No constant for state variables needed.

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to produce this document>

24 Jan 2022 Discussions

Optimization criterions:
1. Stocking inventory
2. Setting Price
3. Infrastructure Management

Functional requirements
1) Manage transactions
2) Inventory restocking reminders
3) Suggest optimal inventory, price and employees
4) Calculate profit/loss

Language Requirement - MERN Stack

Design -
1. It will be a website with role based authorization(admin, employee)
2. Plotting graphs

Assumptions -
1) Loss function depends on price and helps evaluate demands
2) Re-stocking reminder does not take season into factor

User Interface -
1) Interface for cash counter to register the transactions
2) Interface for admin to see profit /loss, restocking reminder, optimization results and employee performance.

Safety and Security -
1) Make it password protected so that information(employee performance, profit/loss) is not leaked

27th Jan 2022 Discussion

1) Optimization criteria is now simplified, and will only be used to suggest optimal stocking
2) Assumptions finalized
3) Separate portals for Employee and Administrator
4) Use cases finalized

29th Jan 2022 Discussion

1) Language requirements: Can use C++ and a GUI (developed using Visual Studio for Windows or QT for linux) instead of a web application using MERN stack
2) Distributed Database Management systems might be required if more than one employee portals are interacting at the same time

10th April 2022 Discussion

Removed use-cases which we could not implement.

27th April 2022

Went through the entire document and revised all the fields and changed the use-case diagrams.