|  |  |
| --- | --- |
|  | **Stock Management System**  **Technical Design Document** |
| |  |  |  |  | | --- | --- | --- | --- | |  | **Prepared By / Last Updated By** | **Reviewed By** | **Approved By** | | **Name** | Hemadharshini S |  |  | | **Role** | Program Analyst Trainee |  |  | | **Signature** |  |  |  | | **Date** |  |  |  | |
|  |

Table of Contents

[1.0 Introduction 3](#_Toc14171023)

[1.1 Purpose of this document 3](#_Toc14171024)

[1.2 Project overview 3](#_Toc14171025)

[2.0 Solution Summary 3](#_Toc14171026)

[2.1 Scope 3](#_Toc14171027)

[2.2 Risks 4](#_Toc14171030)

3.0 Use Case 4

[4.0 System Design 8](#_Toc14171032)

[4.1 Proposed design 8](#_Toc14171033)

[5.0 Database Design 8](#_Toc14171035)

[5.1 Data Model 8](#_Toc14171036)

5.2 Tables Structure 9

6.0 Appendices 9

6.1 Glossary 9

6.2 Other 9

7.0 Terms & Conditions 9

# Introduction

## 1.1 Purpose of this document

This document defines the stock management system that efficiently manages inventory, optimizes stock levels, control costs, prevents stockouts, tracks goods movement, and minimizes product wastage. It contributes to overall business efficiency.

## Project overview

The stock management system is a web-based application that allows- users to add, show, update, and delete stocks. Stock class has all the getters and setters for stock name, quantity, price, and availability. It also has a method for removing stock with condition of quantity to remove is less than available quantity.

The system is developed using Java as the programming language and MySQL as the database management system. The system uses JDBC to connect to the database and execute queries.

The system has the feature of adding, updating, deleting, and viewing product details in the database.

The system is designed to improve the efficiency and effectiveness of store, as well as to enhance the user experience and satisfaction of the customer and administrator. The system also aims to reduce manual errors, paper-work, and operational costs. The system is scalable, secure, and user-friendly.

# Solution Summary

## Scope

Following are the key features that needs to be implemented for Customer and Admin:

**Customer**

1. Add Stock products
2. View Stock products
3. Delete the Stock products

**Admin**

1. View Stock products
2. Update Stock products

## Risks

* **Technical Risks**: These include risks related to software development such as potential bugs, system crashes, or data corruption. There could also be risks associated with integrating the new system with existing systems.
* **Data Integrity and Accuracy**: Inaccurate or corrupted data can lead to incorrect stock levels, order fulfillment issues, and financial losses.
* **Integration Challenges**: Integrating the stock management system with existing systems (e.g., accounting, sales, and e-commerce platforms) can be complex.
* **Security Vulnerabilities**: Unauthorized access, data breaches, or cyberattacks can compromise sensitive stock information.
* **Performance Bottlenecks**: Slow database queries or inefficient indexing can impact overall system performance.
* **User Acceptance Risks**: There is a risk that end users (admin and users) may not accept or adapt to the new system.
* **Training Risks**: There could be risks associated with training staff to use the new system effectively.
* **Maintenance Risks**: Over time, the system will require updates and maintenance. There is a risk that these could be more costly or time-consuming than anticipated.

# Use Case

There will be two types of users who will be using this application:

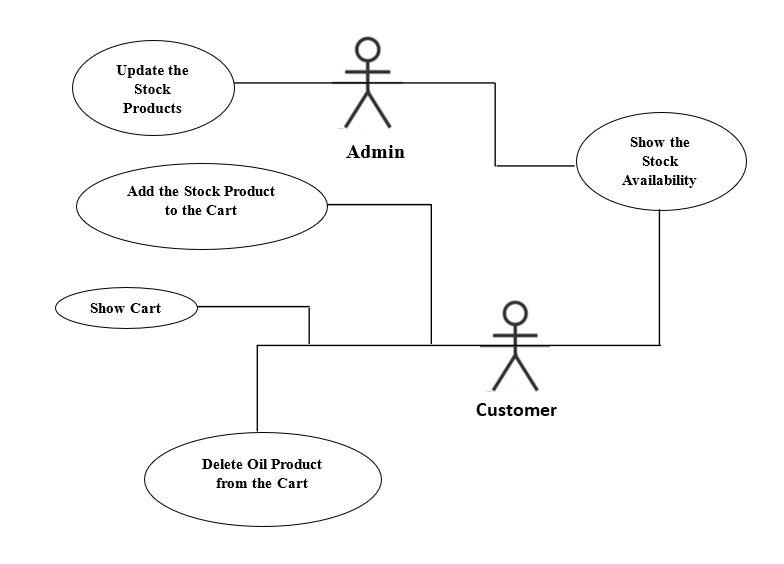
1. Admin

2. Customer

Admin – Responsible for managing the stock products available in store.

Customer – Find stock products available in the store and add them to the cart. The customer also has the option to delete products which were already added to the cart.

Find below the list of use case diagram that represents the list of Use Cases that needs to be incorporated.



## Add Menu Item List

Both Admin and Customer can add the stock products into the stock products available in the platform. For Admin all the stock products will be listed. For Customer, only stock products that are available in the store.

|  |  |
| --- | --- |
| Stock ID | 101 |
| Stock Name | Add Stock products (Admin) |
| Actor(s) | Admin |
| Description | Add the stock products available in the system |
| Pre-condition | Admin should have privilege to add stock products List |
| Post-Condition | Stock products are listed in the screen. Each stock product will have option to edit the stock product. |
| Basic Path | 1. User opens the stock product list for admin in the browser. 2. Online stock store system fetches the stock product list from the database. 3. List of menu items shown to the user with following details about each menu item:    1. ID    2. Name    3. Quantity    4. Price    5. Availability |

## Menu Item

Admin uses this option to view the availability details of stock products.

|  |  |
| --- | --- |
| Stock ID | 101 |
| Stock Name | View the Stock Product |
| Actor(s) | Admin |
| Description | Admin is allowed to view the availability of the stock product details |
| Pre-condition | Admin can view the stock product privilege |
| Post-Condition | Availability stock product details are saved in the database |
| Basic Path | 1. Admin uses the view option on stock products from stock products List. 2. Fetched stock product details from database and populate a form with following fields:    1. ID    2. Name    3. Quantity    4. Price    5. Availability |
| Pre-condition | Customer should have Add to Cart privilege |
| Post-Condition | An electronic product is added to customer’s Cart and the added is listed when viewing the Cart |
| Basic Path | 1. Customer uses the Add to Cart option on an electronic product from electronic product List. 2. The electronic product is added in the Cart. 3. Confirmation message displayed to the user after adding the product to Cart. |

## Update Cart

Admin can update the list of stock products in the Cart.

|  |  |
| --- | --- |
| Stock ID | 101 |
| Stock Name | Update the Stock products |
| Actor(s) | Admin |
| Description | Admin can update the stock products from the Cart |
| Pre-condition | Admin have the privilege to can change the name of the product. |
| Post-Condition | Admin can update into a system of the changed stock name. |
| Basic Path | 1. Admin uses the update option on stock products from stock products List. 2. Fetched stock product details from database and populate a form with following fields:    1. ID    2. Name    3. Quantity    4. Price    5. Availability |

## Delete item from Cart

A customer can remove a stock product from the Cart.

|  |  |
| --- | --- |
| Stock ID | 101 |
| Stock Name | Remove stock product from Cart |
| Actor(s) | Customer |
| Description | Customer can Remove Cart Product privilege |
| Pre-condition | The user can Remove Cart privilege |
| Post-Condition | Selected stock products removed from the Cart |
| Basic Path | 1. User opens the Cart. 2. User selects the Remove option on any one of the stock products from the Cart. 3. Online stock store removes the stock product from the user’s Cart. |

# System Design

## Proposed design

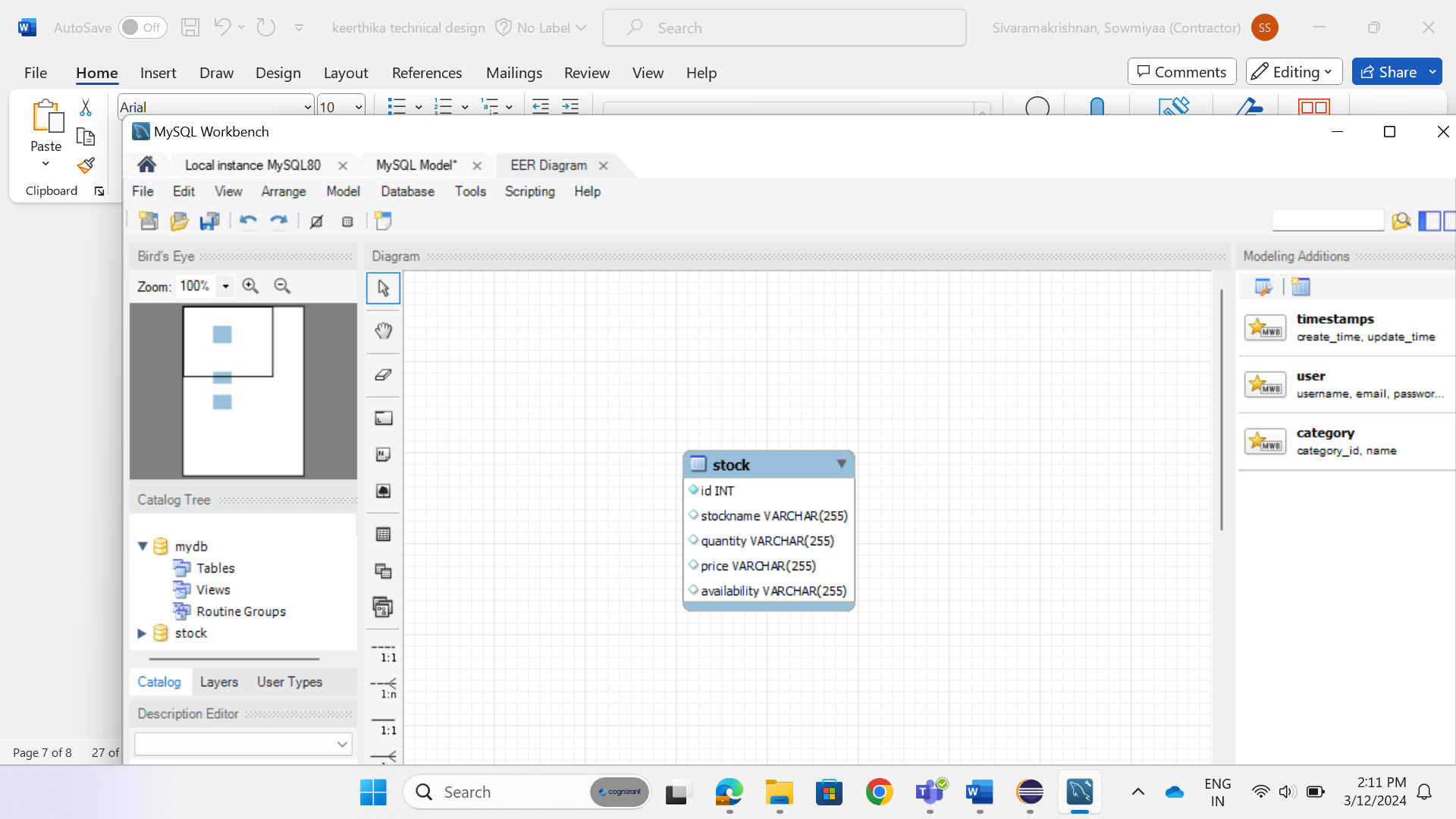
The aim of the proposed system is to develop a system of improved facilities. The proposed system can overcome the overall limitations of the existing system. The system provides proper security and reduces manual work. This overall increase management productivity, eliminates paperwork, reduce manpower and prove to be very economical in the long run.

# Database Design

## Data Model

*The data model consists of tables for each class (Id, Stock Name, Quantity, Price, and Availability), each with fields relevant to that entity.*

## Tables Structure



# Appendices

## Glossary

|  |  |
| --- | --- |
| **Acronyms** | **Definitions** |
| SMS | Stock Management System |

## Other

# Terms & Conditions

***Disclaimer: Please do not circulate or distribute this document outside of Cognizant Network, We have a Zero Tolerance Policy. Kindly adhere to 100% Compliance at all times.***