|  |  |
| --- | --- |
|  | **WALL STREET STOCK**  **Technical Design Document** |
|  |

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 4](#_Toc14171026)

[2.1 Scope 4](#_Toc14171027)

[2.2 Assumptions 5](#_Toc14171028)

[2.3 Dependencies 5](#_Toc14171029)

[2.4 Risks 5](#_Toc14171030)

[3.0 Schematic Diagram 5](#_Toc14171031)

[4.0 System Design 6](#_Toc14171032)

[4.1 Proposed design 6](#_Toc14171033)

[4.2 Component inventory 7](#_Toc14171034)

[5.0 Database Design 7](#_Toc14171035)

[5.1 Data Model 10](#_Toc14171036)

[5.2 Tables Structure 11](#_Toc14171037)

[6.0 Appendices 11](#_Toc14171038)

[6.1 Glossary 14](#_Toc14171039)

# Introduction

## Purpose of this document

## About this document

### Purpose of the document

The purpose of the software requirements document is to systematically capture requirements for the project and the system “Wall Street Stock” to be developed. Both functional and non-functional requirements are captured in this document. It also serves as the input for the project scoping.

## Project overview

Here you will find the flow and procedure of the project. This overview is to guide how the contract management will flow.

* **Role Selection** – User will enter into a web page to find two roles to select. One is Portfolio Admin and the other is Investor. Once the user selects any role, the page will guide the user to any of the login pages and if the user has not yet registered, he can go with the option “click here to register” which will guide him to the respective registration page.
* **Registration and Login** – will be used by the Portfolio admin and Investor to register and login into the system and also to edit their respective registration details in the system.
* **Requirements** – The Portfolio Admin can create requirements for registration on the portfolio platform. The investor can view the requirements created by the admin and can choose the requirements he/she is willing to provide for.
* **Proposal** – The portfolio manager can add the stock or remove the unsold stock in the portfolio from where investor can brought or sold the stocks. The admin can view the proposal of stock that they want to buy or sold if they put more stock then available then admin will reject the proposal, or if they want to sold more stock than they hold that time also admin will reject the proposal.
* **Investment** – If the investor has given correct number of stocks, then it will be accepted by Admin. After that they can trade freely
* **Terms and Conditions of a Trading** - will be used by Portfolio Admin and Investor to read terms and conditions before editing a Trading or saving a Trading.

––

* **Setting Rule for Trading** - will be used by Admin to add rules under a Contract of Trading.
* **Manage Delivery –** The Portfolio manager can manage deliveriesofstocksby updating the field Successful added and Put correct number.
* **Help –** The investor can ask for help if find any difficulty in trading by clicking “Help” button.
* **Dashboard -** This will help investor to redirect to home page from any pages by clicking “Dashboard” bottom.

# Solution Summary

## Scope

* **Role Selection** – User will enter into a web page to find two roles to select. One is Portfolio Admin and the other is Investor. Once the user selects any role, the page will guide the user to any of the login pages and if the user has not yet registered, he can go with the option “click here to register” which will guide him to the respective registration page.
* **Registration and Login** – will be used by the Portfolio admin and Investor to register and login into the system and also to edit their respective registration details in the system.
* **Requirements** – The Portfolio Admin can create requirements for registration on the portfolio platform. The investor can view the requirements created by the admin and can choose the requirements he/she is willing to provide for.
* **Proposal** – The portfolio manager can add the stock or remove the unsold stock in the portfolio from where investor can bring or sold the stocks. The admin can view the proposal of stock that they want to buy or sold if they put more stock then available then admin will reject the proposal, or if they want to sold more stock than they hold that time also admin will reject the proposal.
* **Investment** – If the investor has given correct number of stocks, then it will be accepted by Admin. After that they can trade freely
* **Terms and Conditions of a Trading** - will be used by Portfolio Admin and Investor to read terms and conditions before editing a Trading or saving a Trading.

––

* **Setting Rule for Trading** - will be used by Admin to add rules under a Contract of Trading.
* **Manage Delivery –** The Portfolio manager can manage deliveriesofstocksby updating the field Successful added and Put correct number.
* **Help –** The investor can ask for help if find any difficulty in trading by clicking “Help” button.
* **Dashboard -** This will help investor to redirect to home page from any pages by clicking “Dashboard” bottom.

## Assumptions

*Not applicable.*

## Dependencies

If there is any similar contract creation process required for other applications of the same client, this system should be able to accommodate such a reusability dependency.

The proposal has to be accepted by the Admin.

The contract has to be accepted by the Admin.

The Delivery status has to be managed by the Supplier.

The admin can close or extend the contract.

## Risks

If there is any failure in contract creation process, the system should be able to handle such a financial risk.

# Schematic Diagram

* A schematic, or schematic diagram, is a representation of the elements of a [system](https://en.wikipedia.org/wiki/System) using abstract, graphic [symbols](https://en.wikipedia.org/wiki/Symbol) rather than realistic pictures. It gives an overview of overall system

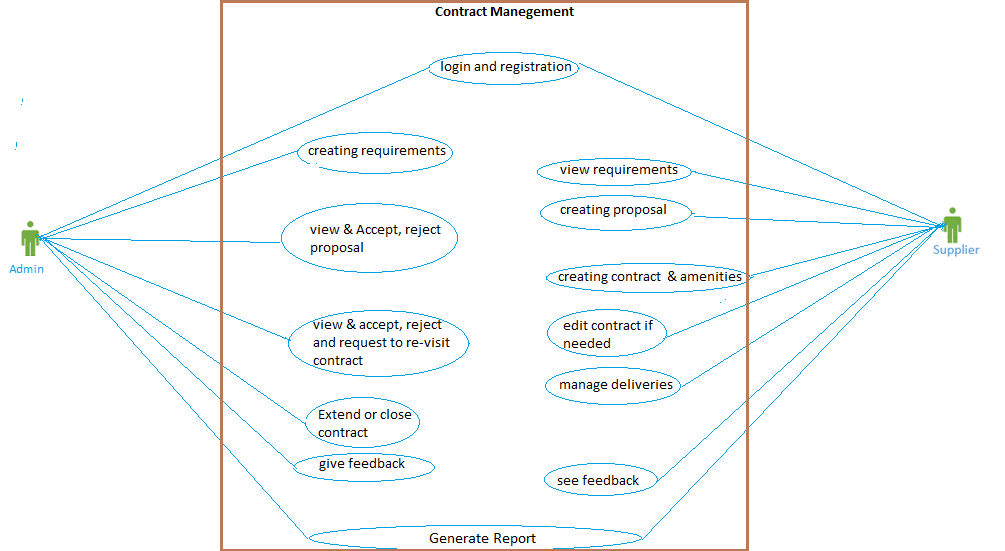


Figure: Use Case Diagram

System Design

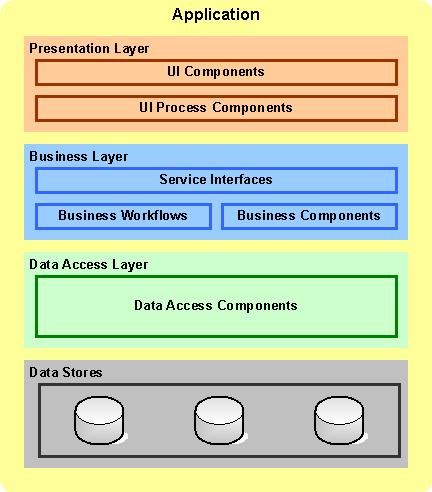
## Proposed design

### Architecture diagram

#### Physical Architecture:

A physical architecture is an arrangement of physical elements, (system elements and physical interfaces) that provides the designed solution for a product, service, or enterprise. It is intended to satisfy logical architecture elements and system requirements. Contract Management System follows a three-layered architecture namely presentation layer, business logic layer and data access layer.

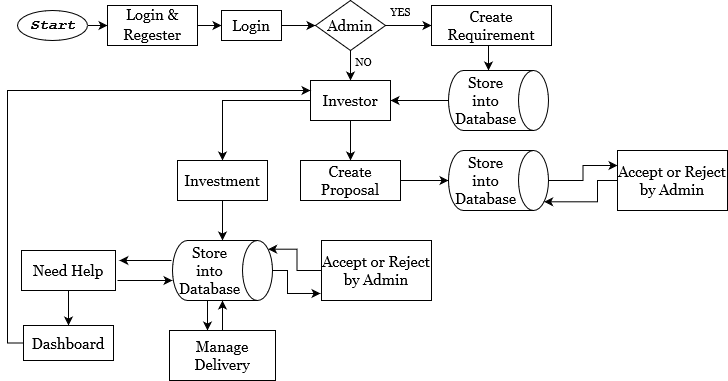
* **Presentation Tier** is the tier in which the users interact with an application. Presentation Tier contents Shared UI code, Code Behind and Designers used to represent information to user.
* **Business Tier** is mainly working as the bridge between Data Tier and Presentation Tier. All the Data passes through the Business Tier before passing to the presentation Tier. Business Tier is the sum of Business Logic Layer, Data Access Layer and Value Object and other components used to add business logic.
* **Data Tier** is basically the server which stores all the application’s data. Data tier contents Database Tables, XML Files and other means of storing Application Data.

**

***Figure*** *– High Level Design (HLD)*

## Component inventory

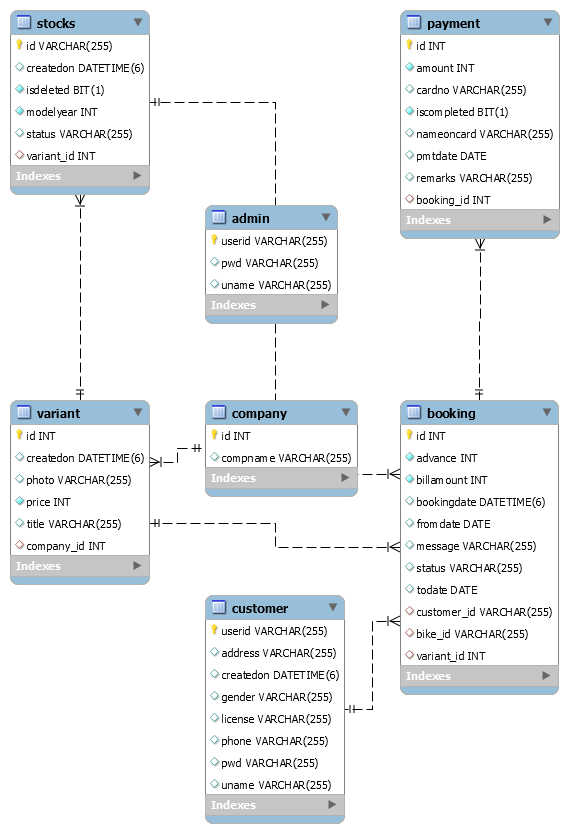
* **Role Selection** – User will enter into a web page to find two roles to select. One is Portfolio Admin and the other is Investor. Once the user selects any role, the page will guide the user to any of the login pages and if the user has not yet registered, he can go with the option “click here to register” which will guide him to the respective registration page.
* **Registration and Login** – will be used by the Portfolio admin and Investor to register and login into the system and also to edit their respective registration details in the system.
* **Requirements** – The Portfolio Admin can create requirements for registration on the portfolio platform. The investor can view the requirements created by the admin and can choose the requirements he/she is willing to provide for.
* **Proposal** – The portfolio manager can add the stock or remove the unsold stock in the portfolio from where investor can bring or sold the stocks. The admin can view the proposal of stock that they want to buy or sold if they put more stock then available then admin will reject the proposal, or if they want to sold more stock than they hold that time also admin will reject the proposal.
* **Investment** – If the investor has given correct number of stocks, then it will be accepted by Admin. After that they can trade freely
* **Terms and Conditions of a Trading** - will be used by Portfolio Admin and Investor to read terms and conditions before editing a Trading or saving a Trading.
* **Setting Rule for Trading** - will be used by Admin to add rules under a Contract of Trading.
* **Manage Delivery –** The Portfolio manager can manage deliveriesofstocksby updating the field Successful added and Put correct number.
* **Help –** The investor can ask for help if find any difficulty in trading by clicking “Help” button.
* **Dashboard -** This will help investor to redirect to home page from any pages by clicking “Dashboard” button.



**Figure:** Component Inventory

# Database Design

## Data Model



***Figure:*** *Data Model*

## Tables Structure

***Portfolio\_manager/Admin***

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| userid | VARCHAR | 255 | No |
| pwd | VARCHAR | 255 | Yes |
| uname | VARCHAR | 255 | Yes |

***Investor***

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| userid | VARCHAR | 255 | No |
| address | VARCHAR | 255 | Yes |
| createdon | DATETIME | 6 | Yes |
| gender | VARCHAR | 255 | Yes |
| license | VARCHAR | 255 | Yes |
| phone | VARCHAR | 255 | Yes |
| pwd | VARCHAR | 255 | Yes |
| uname | VARCHAR | 255 | Yes |

***company***

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| id | INT |  | No |
| compname | VARCHAR | 255 | Yes |

***booking***

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| id | INT |  | No |
| advance | INT |  | No |
| billamount | INT |  | No |
| bookingdate | DATETIME | 6 | Yes |
| fromdate | DATE |  | Yes |
| message | VARCHAR | 255 | Yes |
| status | VARCHAR | 255 | Yes |
| todate | DATE |  | Yes |
| customer\_id | VARCHAR | 255 | Yes |
| stock\_id | VARCHAR | 255 | Yes |
| category\_id | INT |  | Yes |

***payment***

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| id | INT |  | No |
| amount | INT |  | No |
| cardno | VARCHAR | 255 | Yes |
| iscompleted | BIT | 1 | No |
| nameoncard | VARCHAR | 255 | Yes |
| pmtdate | DATE |  | Yes |
| remarks | VARCHAR | 255 | Yes |
| booking\_id | INT |  | Yes |

***stocks***

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| Id | VARCHAR | 255 | No |
| createdon | DATETIME | 6 | Yes |
| isdeleted | BIT | 1 | No |
| enlistedyear | INT |  | No |
| status | VARCHAR | 255 | Yes |
| category\_id | INT |  | Yes |

***category***

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| id | INT |  | No |
| createdon | DATETIME | 6 | Yes |
| photo | VARCHAR | 255 | Yes |
| price | INT |  | No |
| title | VARCHAR | 255 | Yes |
| company\_id | INT |  | Yes |

# Appendices

## Glossary

|  |  |
| --- | --- |
| **Acronyms** | **Definitions** |
| N. A | N. A |