Detailed Software Technical Design (DSTD)

For

VShare Securities

TABLE OF CONTENTS

Document Revisions

Date	Version	Description	Author	Reviewer	Approver

Information Handling

These documents are strictly for **Virtusa/Client** use only. They shall not be shared with an external party other than the client concerned. These documents should always be kept securely and employees shall use reasonable care protecting these documents from unauthorized use or disclosure to a third party. This category also covers client intellectual property where Virtusa has a non-disclosure agreement with the client.

1 Introduction

1.1 Purpose

This document provides the detailed technical design for the VShare Securities system.

The Virtusa team will use this design as a blueprint when coding and testing the system.

1.2 Scope

Vshare securities is a leading share broker in India. We are planning to reengineer our trading platform. Since it is currently lacking rich UI features and suffering due to heavy load issues.

1.3 Definitions, Acronyms and Abbreviations

[This sub-section must provide definitions for all the terms, acronyms, and abbreviations used in this document.]

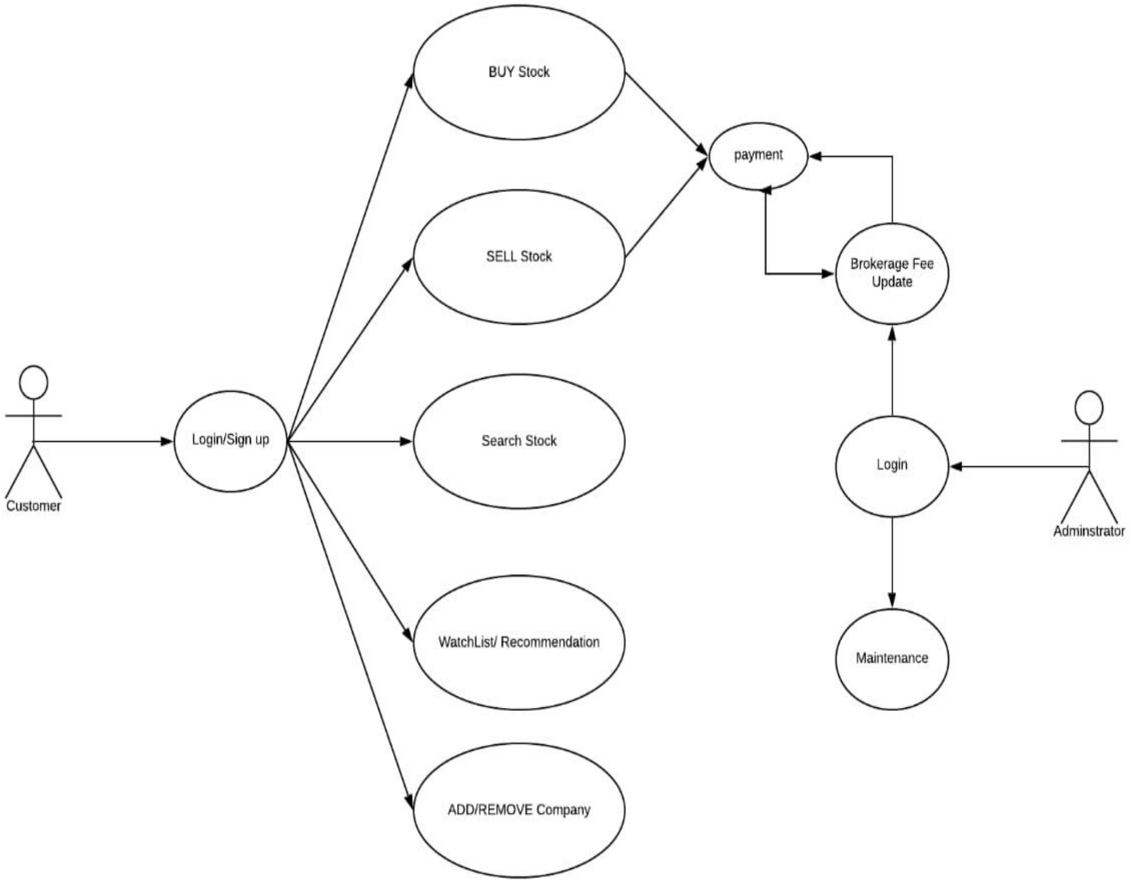
1.4 References

[This sub-section must provide a complete list of all references. Identify each document by title, document number and the version. Specify the sources from which the references can be obtained.

2 Assumptions

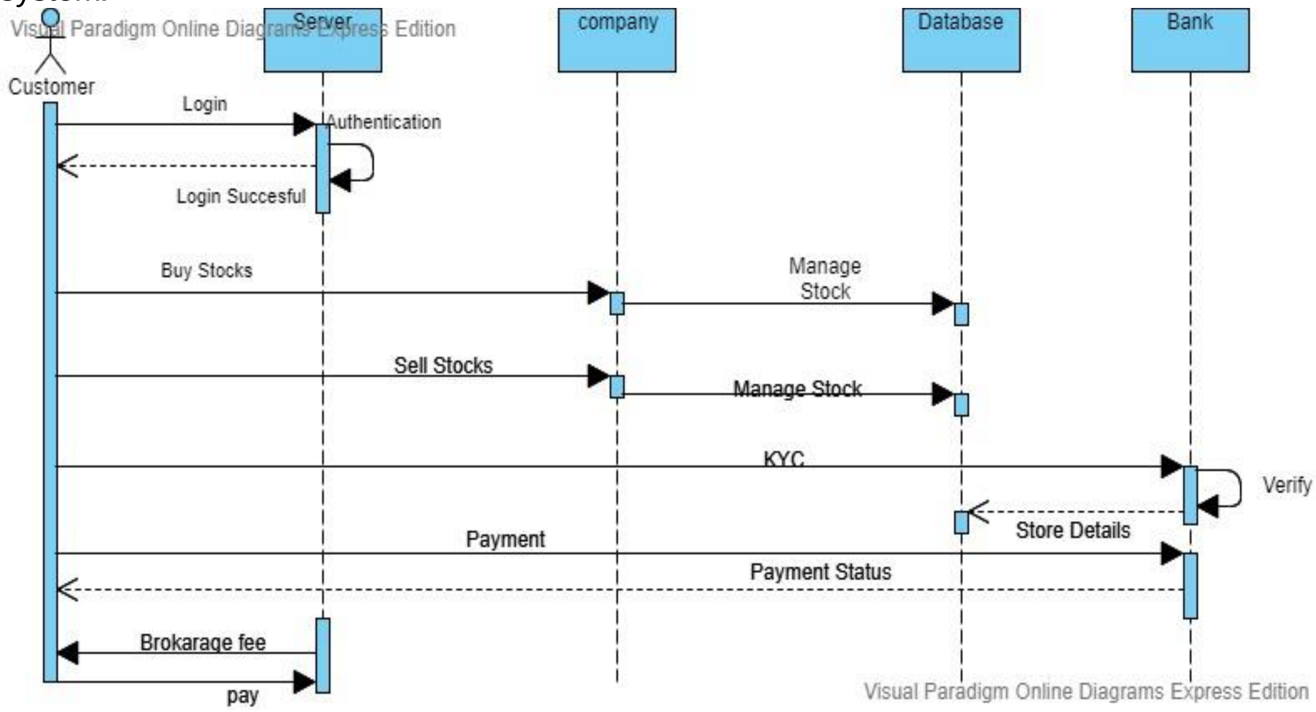
-The timely changing stock price will be based on the variable values that are given by the developers.

3 PROJECT OVERVIEW AND USECASE DIAGRAMS



4 Detailed Design for Module: <<Module Name>>

This section describes the detailed design for the ***<<Module Name>>*** module/sub-system.



4.1 *Detailed Design for Feature:* <<Feature Name>>

Login – Helps to authorize the user and secures the transactions done by the user.

Buy – Allows users to buy stocks of their favourable price and quantity.

Sell – Allows users to sell stocks that they no longer require.

Predict – Feature helps to e-connect with a broker to buy/sell stocks, a feature user can enable at a small price.

Sort – Helps to sort the stocks available based on the price, quantity available etc...

4.1.1 User Profile

Vendor – Have access to adding and removing stocks based on the availability and can fix price.

Buyer/Seller – This type of user can look into the available stocks and have access to buying/selling the same.

4.1.2 Navigation Map

[Describe the navigation map along with use case traceability of all screens for this feature.]

4.1.3 UI Screen Design

Page Summary	
Name	Search for Customer File
Description	The user retrieves the customer information based on the search criteria such as File ID, last name, first name, middle name, business name, SSN/TIN, address, city, state and/or zip.
Use Case Number	UC-002

4.1.3.1 *Screen Layout*

[Include a screen shot and describe each element/button in the following table.]

4.1.3.2 *Validations*

Error Trigger	Action	Description
If the user group is not selected.	Displays the message “Please select a User Group from the list”.	Prompts the user to select an entry from the User Group list.

User enters an invalid date	Displays the message “Enter a valid date. The date must be in <mm-dd-yyyy> format”.	Displays invalid date message.

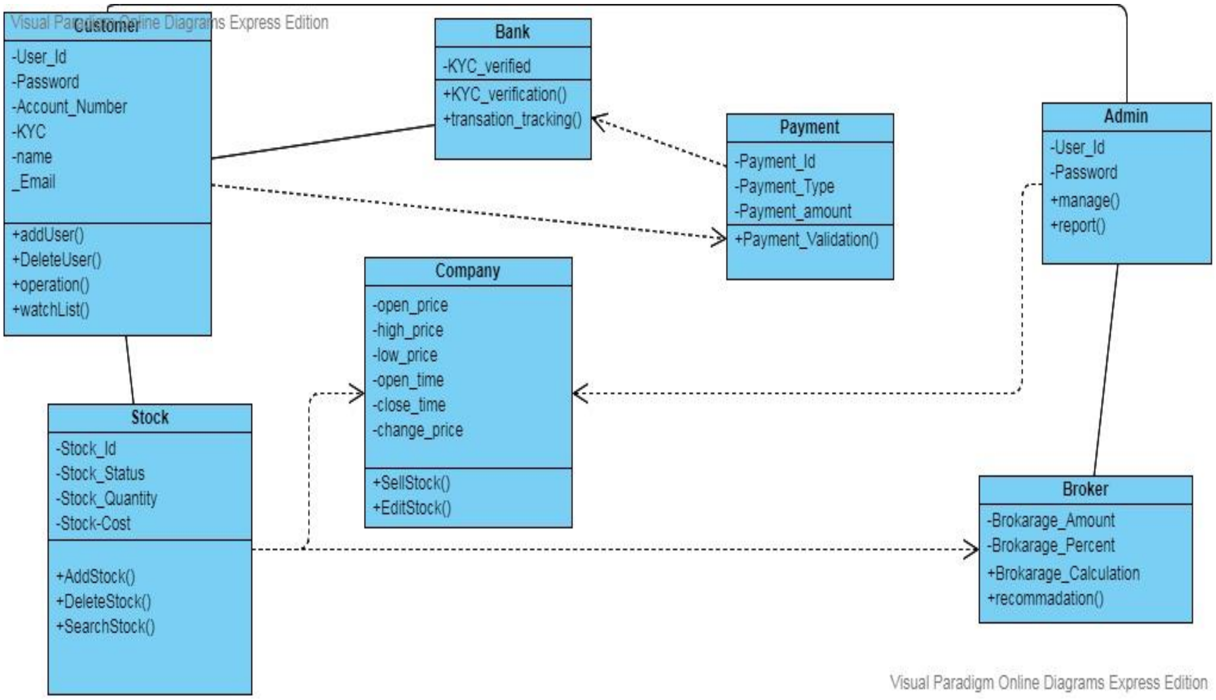
4.1.4 **Object Model/Data Flow Design**

*[This sub-section describes the detailed design for the system pertaining to the <<Feature Name>> feature**.** Interaction diagrams showing the details of the component structure, behavior, or information/control flow may be included in the sub-section devoted to that particular component.]*

4.1.4.1 *Interface Description*

[Describe the interfaces including component interfaces and interfaces to other systems, products, or networks.]

4.1.4.2 *Class Design*



4.1.4.2.1 Algorithmic/Business Logic Description (Optional)

4.1.4.2.2 Local Data Structure (Optional)

[Describe key data structure algorithms used (if any).]

4.1.4.3 *Data Transfer/Value objects*

[Based on the design pattern being used, identify and describe the value/transfer objects.]

4.1.4.4 *Related Database Tables*

[Give the list of related database tables.]

4.1.4.5 *Dependencies with Other Sub-systems/Components*

[Describe the dependencies with other sub-system/components, if any.]

4.2 ***Detailed Design for Feature:*** <<Feature Name>>

Repeat all sub-sections given in 3.1 for this feature.

5 Detailed Design for Module: <<Module Name>>

[Detailed design for Module 2.]

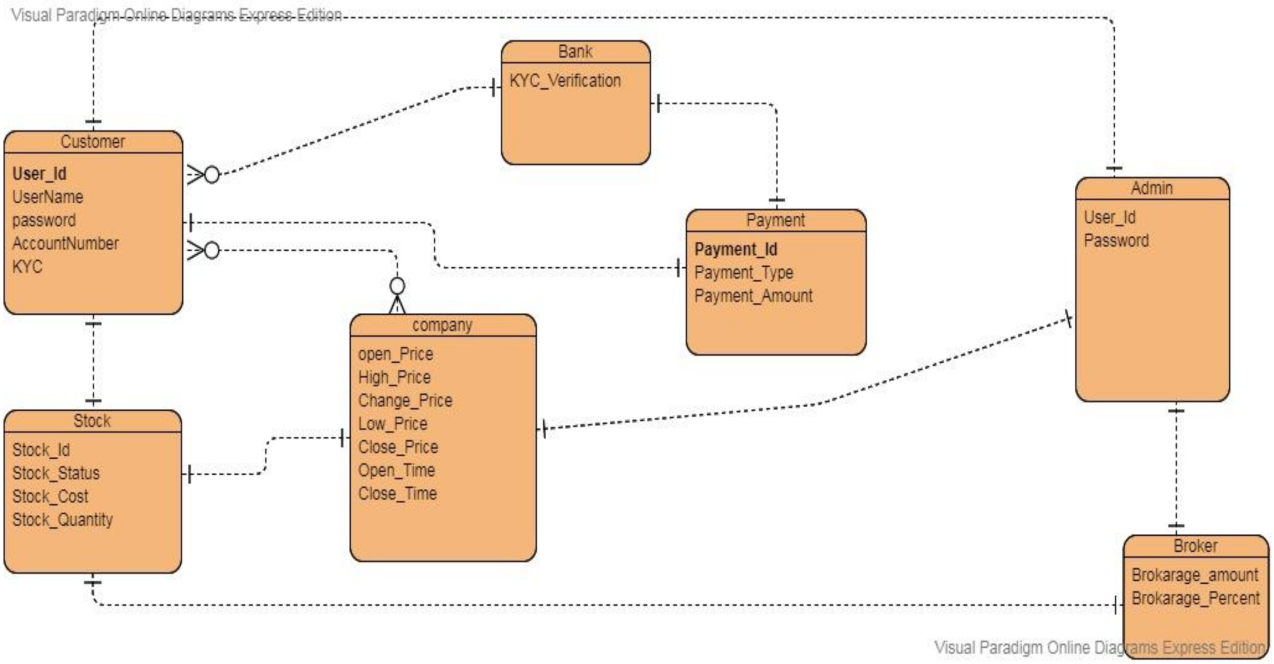
Repeat all sub-sections given in section 3 for this module.

6 Database Design

[This section describes the persistent data and metadata used and generated by the module/sub-system. This will include database schemas (if the database schema is large, use a separate document for database design), registry structures, property file structures, etc.

Models	Tools
Database Entity Relationship Diagrams	ERWin
Object Models	Rational Rose

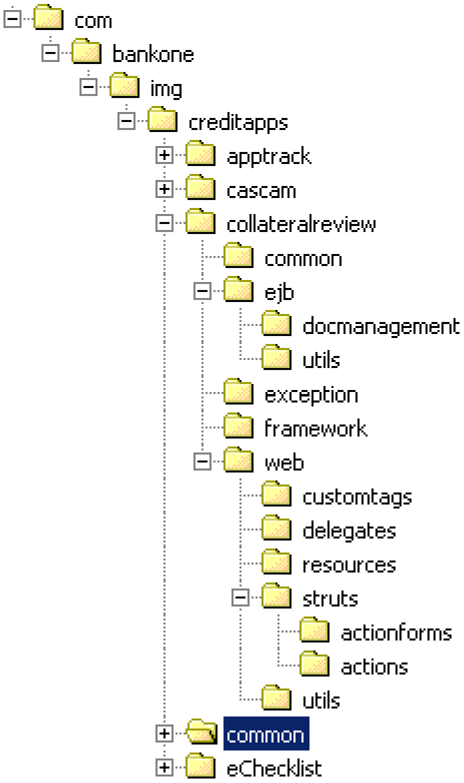
6.1 Entity Relationship Diagrams



7 Packaging/Folder Structure and System Artifacts

[This section describes the file structure for the <Project Name> project, which will indicate where the code, unit tests, executables, libraries, etc. are to be placed during implementation. Optionally the file structure can be described in a separate GA chapter and a reference given here.]

[Sample structure



```
com
  bankone
    img
      creditapps
        apptrack
        cascam
        collateralreview
          common
          ejb
            docmanagement
            utils
          exception
          framework
          web
            customtags
            delegates
            resources
            struts
              actionforms
              actions
            utils
        common
        eChecklist
```

End Sample Structure]

7.1 System Artifacts

[Provide information on the Registry structure, INI files, property files, configuration files, etc.]

8 Core Technical Services Design

[This section describes the detailed design and usage pattern for the core technical services of the system.]

8.1 Persistence

[This sub-section captures the detailed design and usage patterns of the persistence service for the application.]

8.2 Inter-process Communication

[This sub-section captures the detailed design and usage patterns of the inter-process communication framework for the application.]

8.3 Authentication and Authorization

[This sub-section captures the detailed design and usage patterns of the security services for the application.]

8.4 Error Handling

This sub-section captures the detailed design and usage patterns of the error handling services for the application.]

8.5 Logging

[This sub-section captures the detailed design and usage patterns of the logging framework for the application.]

8.6 Transaction Management

[This sub-section captures the detailed design and usage patterns of the Transaction Management service for the application.]

8.7 Other Applicable Technical Services

[This sub-section captures the detailed design and usage patterns of the other core technical services to be handled by the system such as the installation mechanism, failure prevention, fault tolerance, caching design, internationalization, validation framework, client and server initialization, error handling, etc.]

9 Non Functional Requirements

Appendix

[Presents information that supplements the design specification.]

1. *Design Metrics to be Used*

[A description of all design metrics to be used during the design activity is listed here.]

2. *Supplementary Information*

[Provide as required.]

3. *DQI – Design Quality Index Checklist*

This checklist shall be used to perform self-review as a reference while preparing software design.

DQI-Design Quality
Index Checklist-1.00.