CodeeBank

Bank Management System

Software Architecture Document

Version 1.0

Revision History

Date	Version	Description	Author
18/03/2020	1.0	Final Draft	An Thị Phương

Table of Contents

1. Introduction

1.1 Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

1.2 Scope

This document applies to the Bank Management System which will be developed by Group ADMP.

1.3 Definitions, Acronyms, and Abbreviations

User – a person who use the system, can be customer.

Bank Staff – a person who is responsible for managing some bank tasks.

Bank Manager – a person who manages some advance tasks in a bank.

Guest – a user who is not logged in the system.

1.4 References

None.

1.5 Overview

In the following section, architectural design of the Bank Management System is provided in detail. Frist, the primary software architecture, the system will be defined. Then, there are further discussion about the goals and constraints that will be imposed upon the quality of the final product, which including but not limited to security, distribution and reuse. In the precedence sections, the key views of the system are demonstrated to depict different aspects of the system. Lastly, criteria concerning with size, performance and quality of the system will be proposed.

2.   Architectural Representation

This document presents the architecture as a series of mandatory views: Use-case View, Logical View, Deployment View and Data View. These view are presented as Visual Paradigm Community Edition Models, Star UML, diagrams.net and use the Unified Modeling Language(UML).

Use-case View

- **Audience:** all the stakeholders of the system, including end-users.

- **Area:** describes the set of the scenarios and/or use cases that represent significant, central functionality to the system.

- **Related artifacts:** Use-case Model, Analysis Model, Use-Case-Realization documents.

Logical View

- **Audience:** designers, programmer.

- **Area:** functional requirements: describes the design's object model.

- **Related artifacts:** Design model.

Development View:

- **Audience:** deployment managers, system administrators.

- **Area:** topology: describes the mapping of the software onto the hardware and shows the system's distributed aspects.

- **Related artifacts:** Deployment Model.

Data View

- **Audience:** data specialists, data administrators.

- **Area:** persistence: describes the architecturally significant persistent elements in the data model.

- **Related artifacts:** Data Model.

3. Architectural Goals and Constraints

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

- The Bank Management System must be designed to fulfill all system requirements specified in requirements definition.

- The Bank Management System design must be structured to be robust, easy to change if and when functional requirements change.

- The Bank Management System must be designed to allow the re-use of business logic across applications;

- therefore, the design is divided into three components: model, view and controller.

- The separation of the three components: model, view and controller are also necessary to provide a convenient cooperation between different development teams.

- The Bank Management System will run on a dedicated platform with access to a database.

The Bank Management System website provides most of the content display. An interface to this system must be capable of handling large traffic volumes.
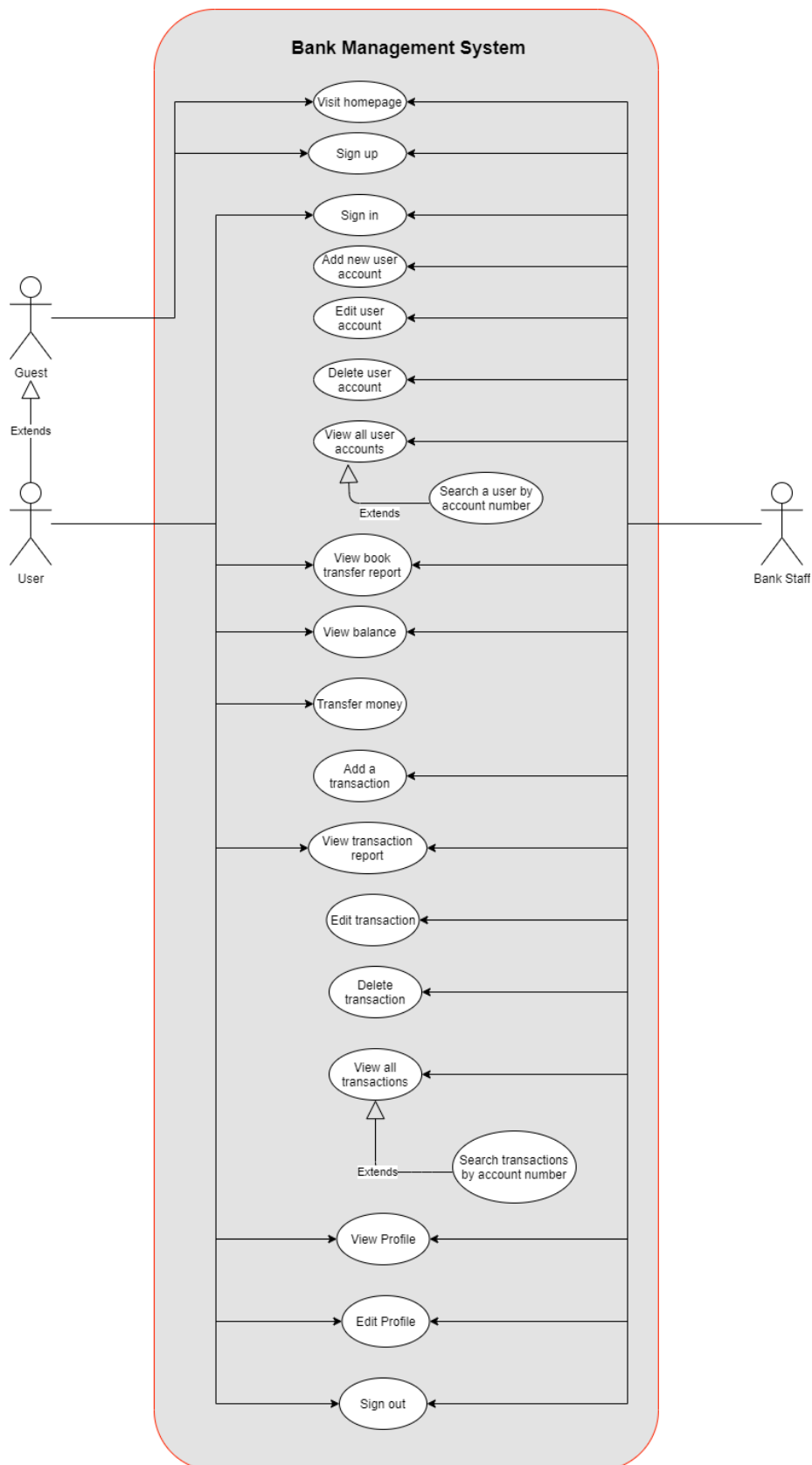
4. Use-Case View

A description of the Use-Case View of the system architecture. The Use Case View is important input to the selection of the set of scenarios and/or use cases that are the focus of an iteration. It describes the set of scenarios and/or use cases that represent some significant, central functionality. It also describes the set of scenarios and/or use cases that have a substantial architectural coverage (that exercise many architectural elements) or that stress or illustrate a specific, delicate point of the architecture.

The significant use cases in this system are listed below:

- Sign –up

- Sig-in

- Sign-out

- Manage user account (Create, Retrieve, Update, Delete user accounts)

- Make a transaction

- Manage transaction (Create, Retrieve, Update, Delete transaction logs)

- View/Edit Profile

Bank Management System

Visit homepage

Sign up

Sign in

Add new user account

Edit user account

Delete user account

View all user accounts

Search a user by account number

Extends

View book transfer report

View balance

Transfer money

Add a transaction

View transaction report

Edit transaction

Delete transaction

View all transactions

Search transactions by account number

Extends

View Profile

Edit Profile

Sign out

Guest

Extends

User

Bank Staff

4.1 Use-Case Realizations

4.1.1 Sign-up
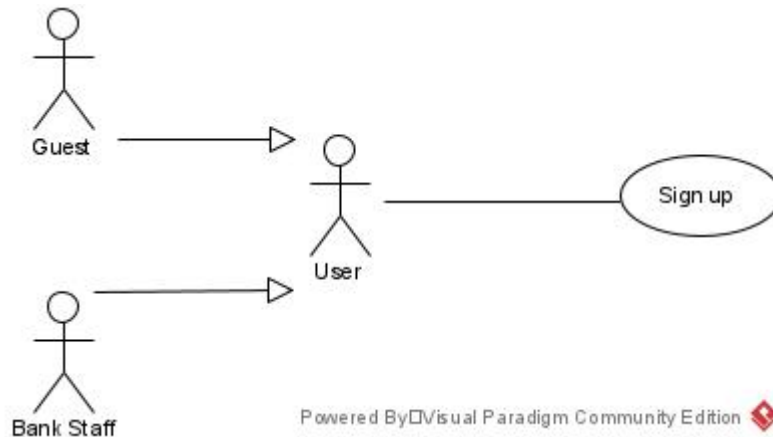
- **Brief Description:** A user creates an account.

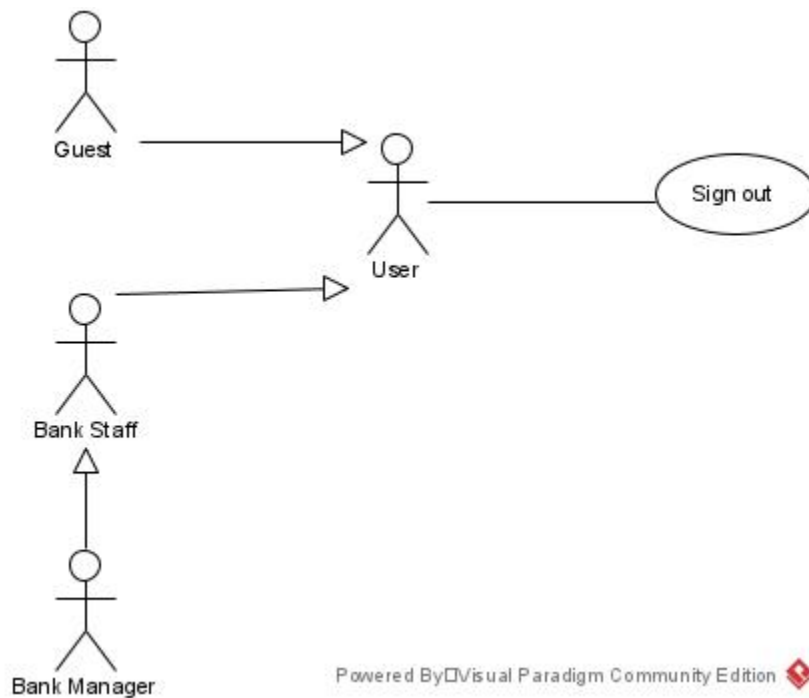- **Specification:** See Use-Case-Realization Specification: Sign-up.

4.1.2 Sign in
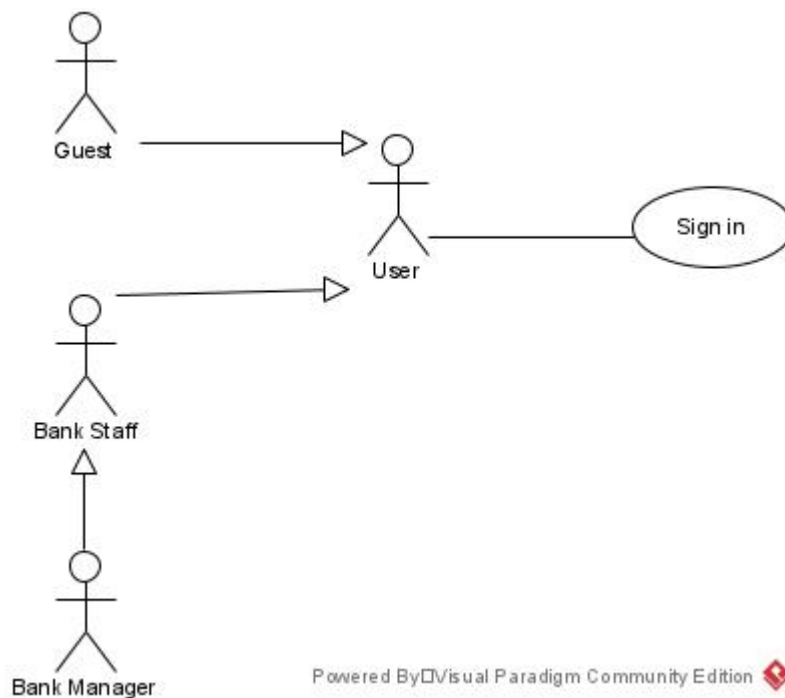
- **Brief Description:** A user logs in to the system.

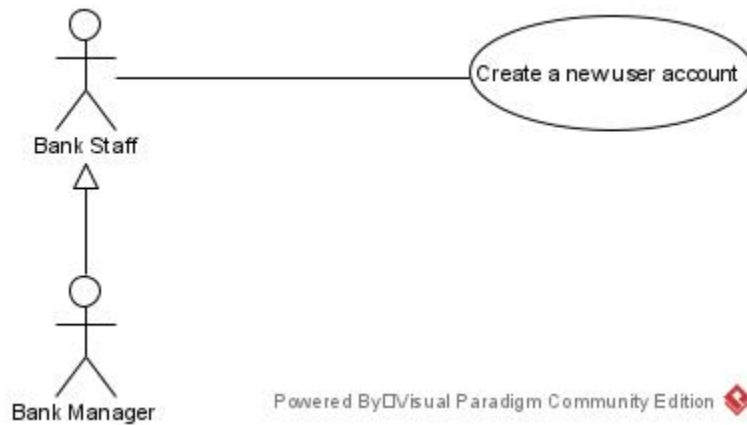- **Specification:** See Use-Case-Realization Specification: Sign-in.

4.1.3 Sign out

- **Brief Description:** A user logs out to the system.
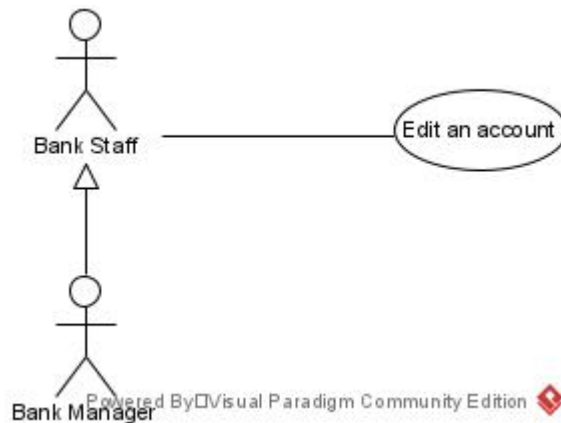- **Specification:** See Use-Case-Realization Specification: Sign-out.

4.1.4 Create a new account

- **Brief Description:** A bank staff creates a new account**.**

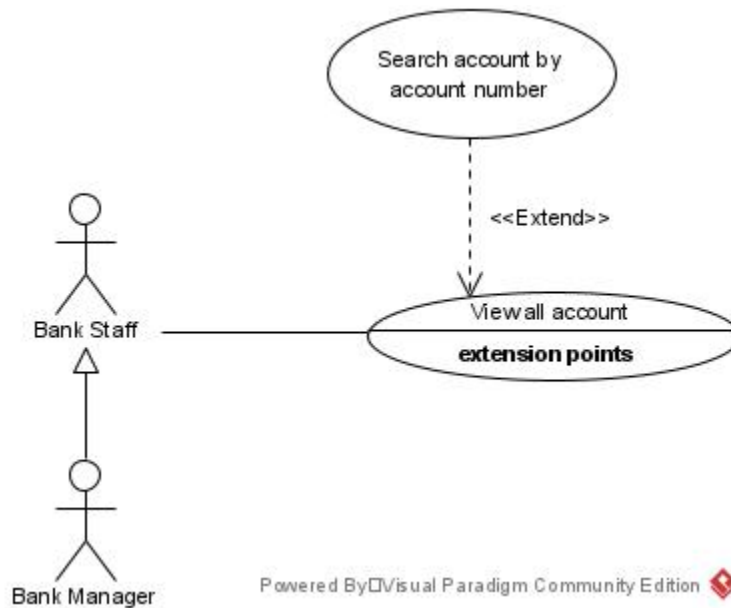- **Specification:** See Use-Case-Realization Specification: Create a new account

4.1.5 Edit an account

- **Brief Description:** A bank staff edits an account**.**

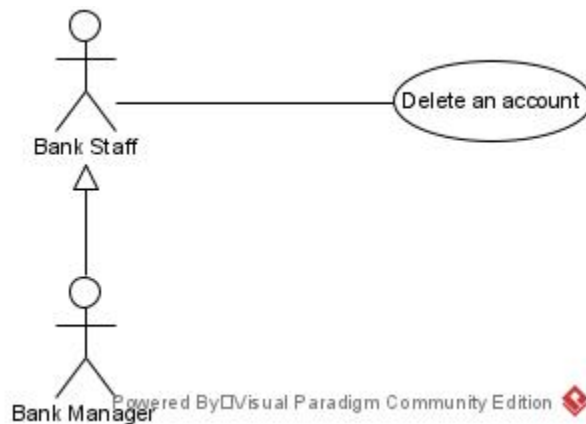- **Specification:** See Use-Case-Realization Specification: Edit an account

4.1.6 View all accounts

- **Brief Description:** A bank staff view all accounts.

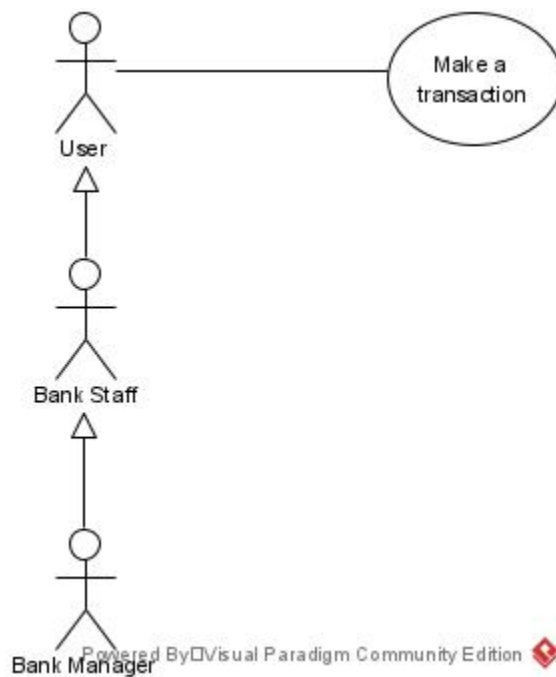- **Specification:** See Use-Case-Realization Specification: View all Accounts

4.1.7 Delete an account

- **Brief Description:** A bank staff deletes an account.

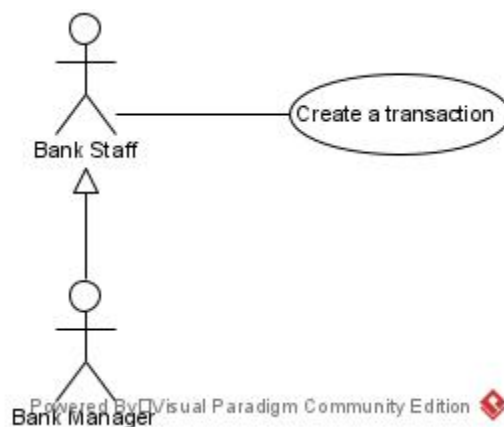- **Specification:** See Use-Case-Realization Specification: Delete an account

4.1.8 Make a transaction

- **Brief Description:** A user makes a transaction**.**

- **Specification:** See Use-Case-Realization Specification: Make a transaction
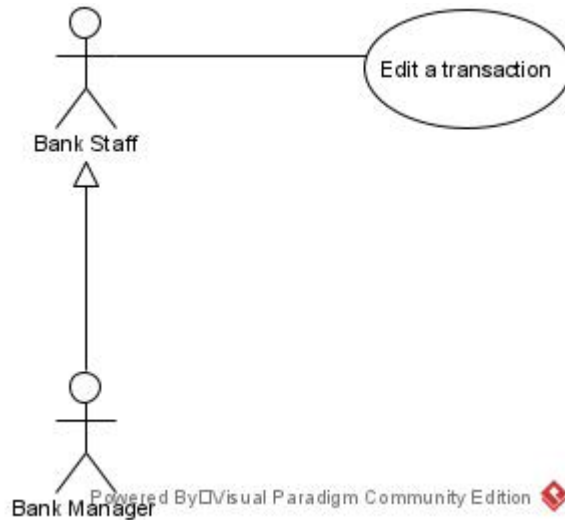
4.1.9 Create a transaction

- **Brief Description:** A bank staff creates a new transaction**.**

- **Specification:** See Use-Case-Realization Specification: Create a transaction.
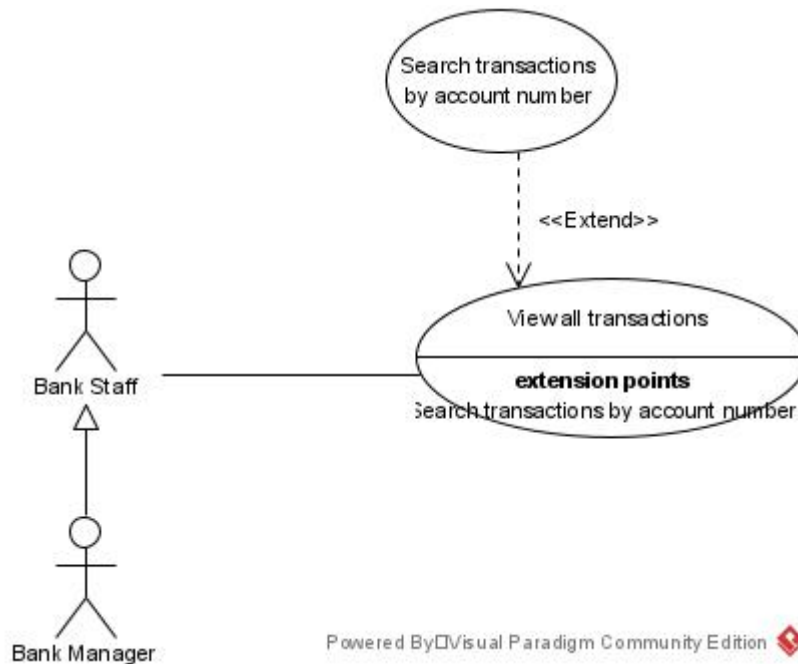
4.1.10 Edit a transaction

- **Brief Description:** A bank staff edits a transaction**.**

- **Specification:** See Use-Case-Realization Specification: Edit a transaction.
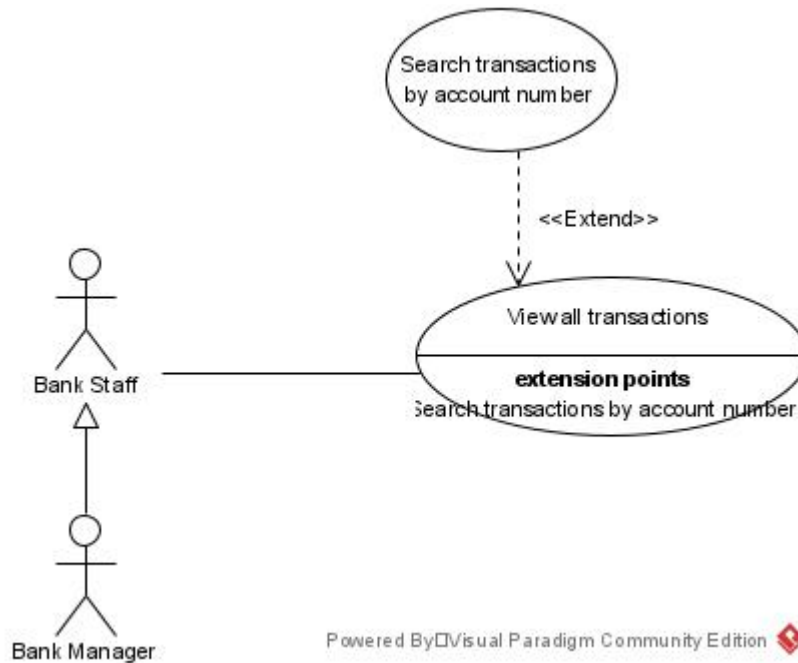
4.1.11 View all transactions

- **Brief Description:** A bank staff views all transaction logs**.**

- **Specification:** See Use-Case-Realization Specification: View all transactions.
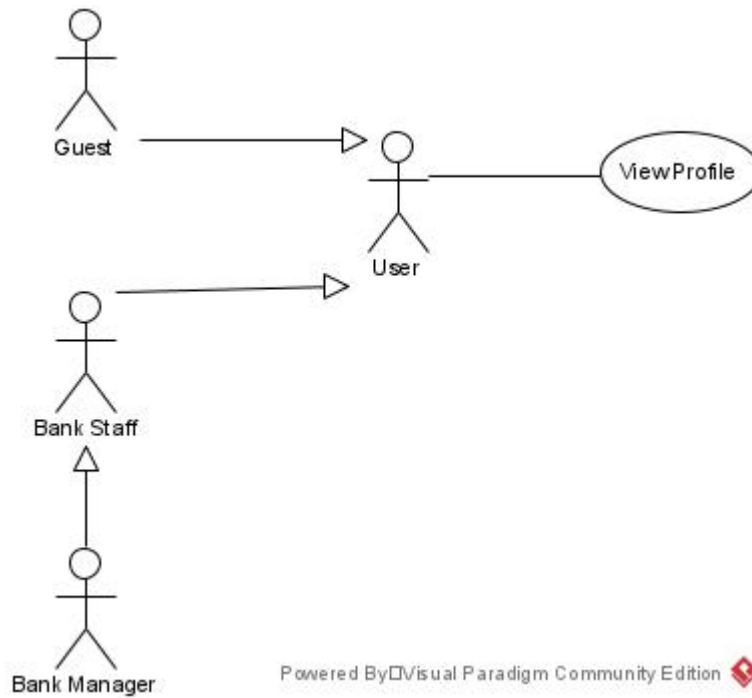
4.1.12 Delete a transaction

- **Brief Description:** A bank staff views all transactions**.**

- **Specification:** See Use-Case-Realization Specification: View all transactions

4.1.13 View Profile

Powered By☐Visual Paradigm Community Edition

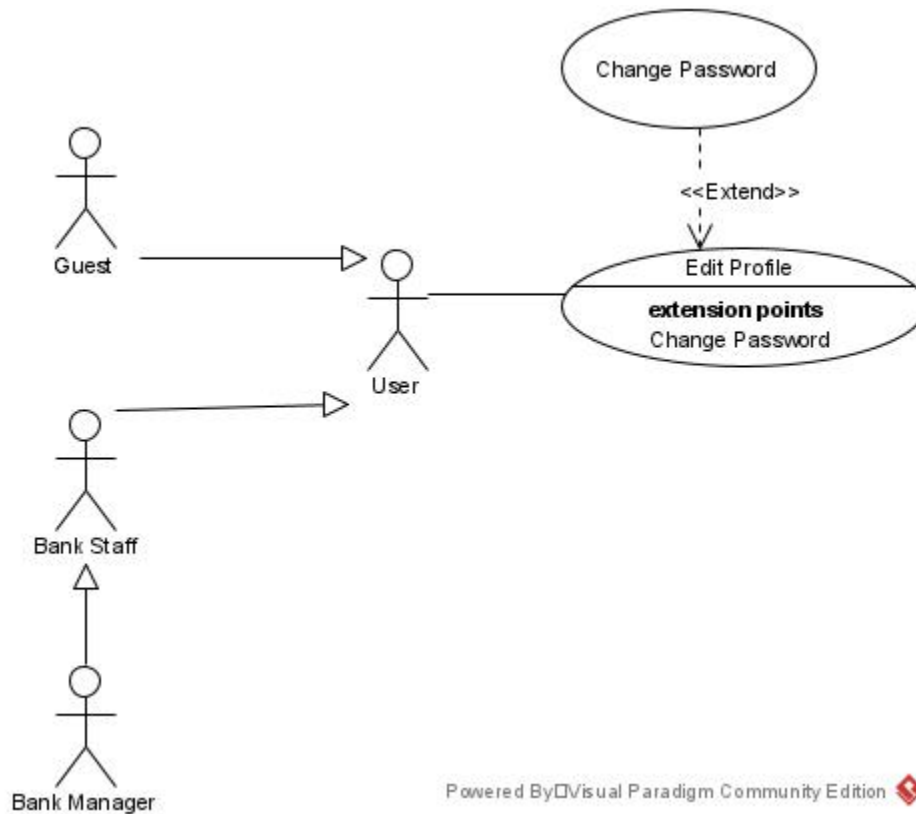- **Brief Description:** A bank staff views profile.

- **Specification:** See Use-Case-Realization Specification: View profile

4.1.14 Edit Profile

- **Brief Description:** A bank staff edits profile**.**

- **Specification:** See Use-Case-Realization Specification: Edit profile.

5. Logical View

5.1 Overview

A description of the logical view of the architecture. Describes the overall decomposition of the design
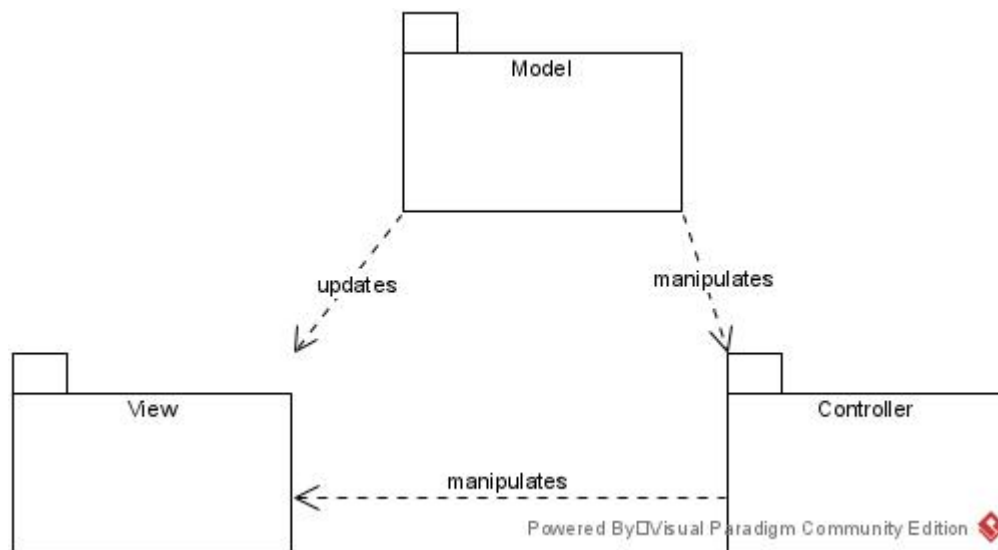model in terms of package hierarchy and layers.

The logical view of the Computer Shop Management System is comprised of 3
significant packages:

- **Model**: contains classes that directly manages the data, logic and rules of the
  Computer Shop

Management System and displayed in the view.

- **View:** contains classes that generates output representation of information to the user based on changes in the model.

- **Controller:** contains classes that can send commands to the model to update the model's state (e.g. add a new computer); it can also send commands to its associated view to change the view's presentation of the model (e.g., scrolling through computer's reviews)

5.2   Architecturally Significant Design Packages

Package Model:

Model

Account
-_id : ObjectId()
-role : String
-cardNo : String
-username : String
-password : String
-address : String
-dob : Date
-phone : String
-email : String
-balance : Double
+get()
+post()
+delete()

Transaction
-_id : ObjectId()
-receiver_number : String
-receiver_name : String
-bankName : String
-giver_name : String
-giver_number : String
-date : Date
-amount : Double
-status : String
+get()
+post()
+delete()

Powered By□Visual Paradigm Community Edition

Name	model
Brief Description	Contains classes that directly manages data, logic and rule of the Bank Management System and displayed in the view.
Classes	Account, Transaction

Package view:

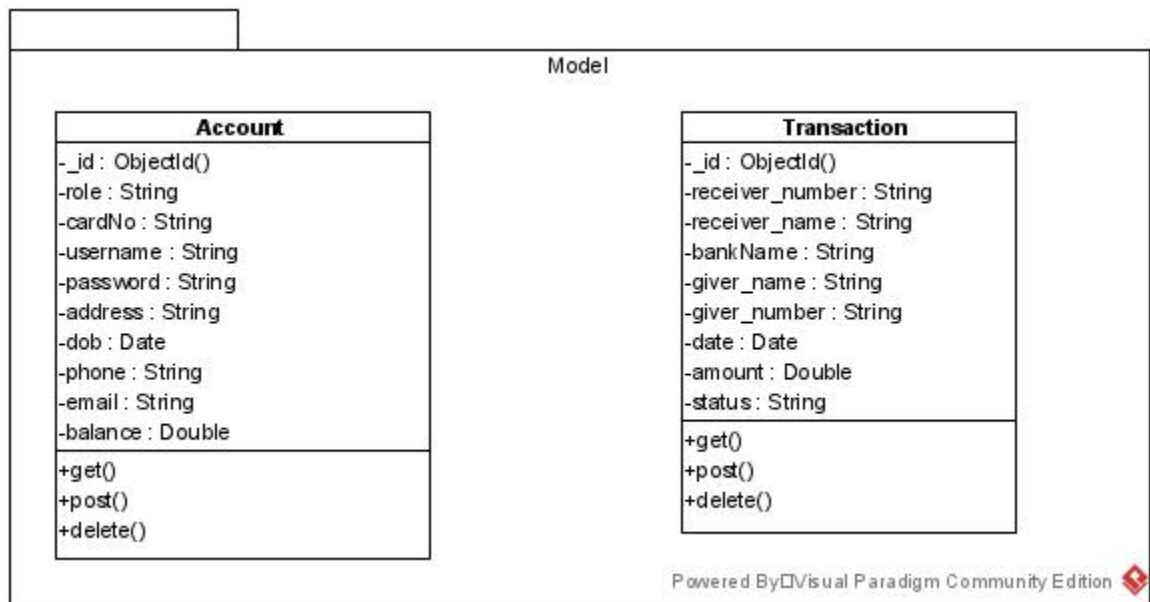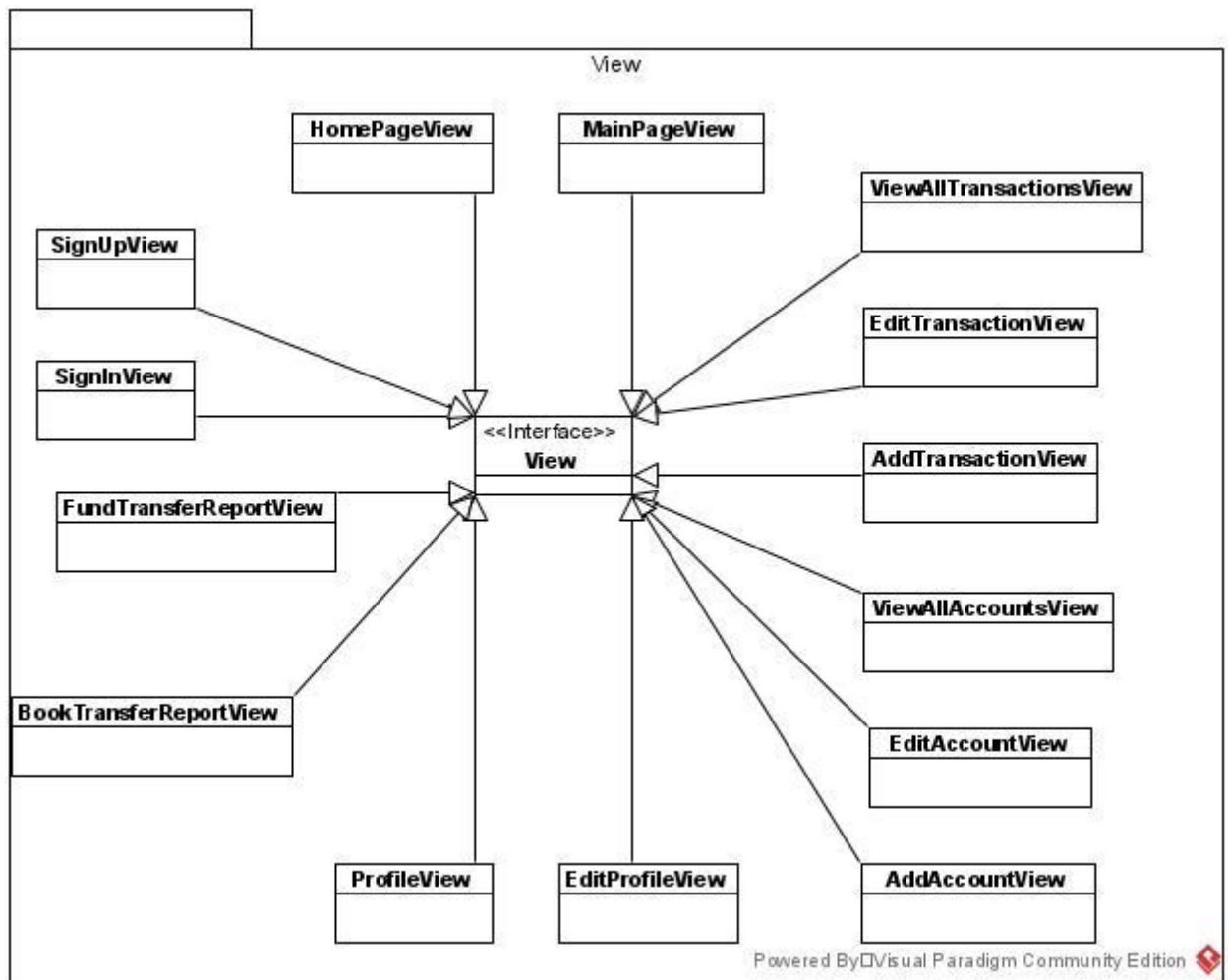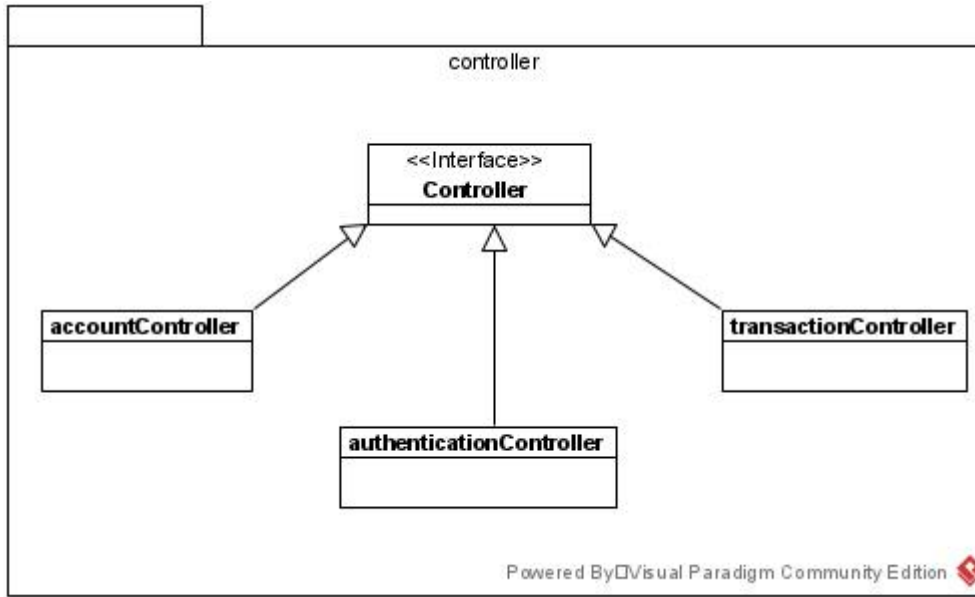| Name	view
Brief Description	Contains classes that generates output representation of information to the user based on changes in the models.
Interfaces	View
Classes	HomePageView, MainPageView, SignUpView, SignInView, FundTransferReportView, BookTransferReportView, ProfileView, EditProfileView, AddAccountView, EditAccountView, ViewAllAccountsView,

	EditTransactionView, AddTransactionView, ViewAllTransactionsView.

Package Controller:

| Name	controller
Brief Description	Contains classes that directly manages the data, logic and rules of the Bank Management System and displayed in the view.
Interfaces	Controller
Classes	accountController, transactionController, authenticationController

6. Process View

The Bank Management System is designed to be implemented on Node.js server which support single-threaded asynchronous event handling (even loop); therefore, concurrency issues will not be considered in this document.

Node JS Application/Node JS Server
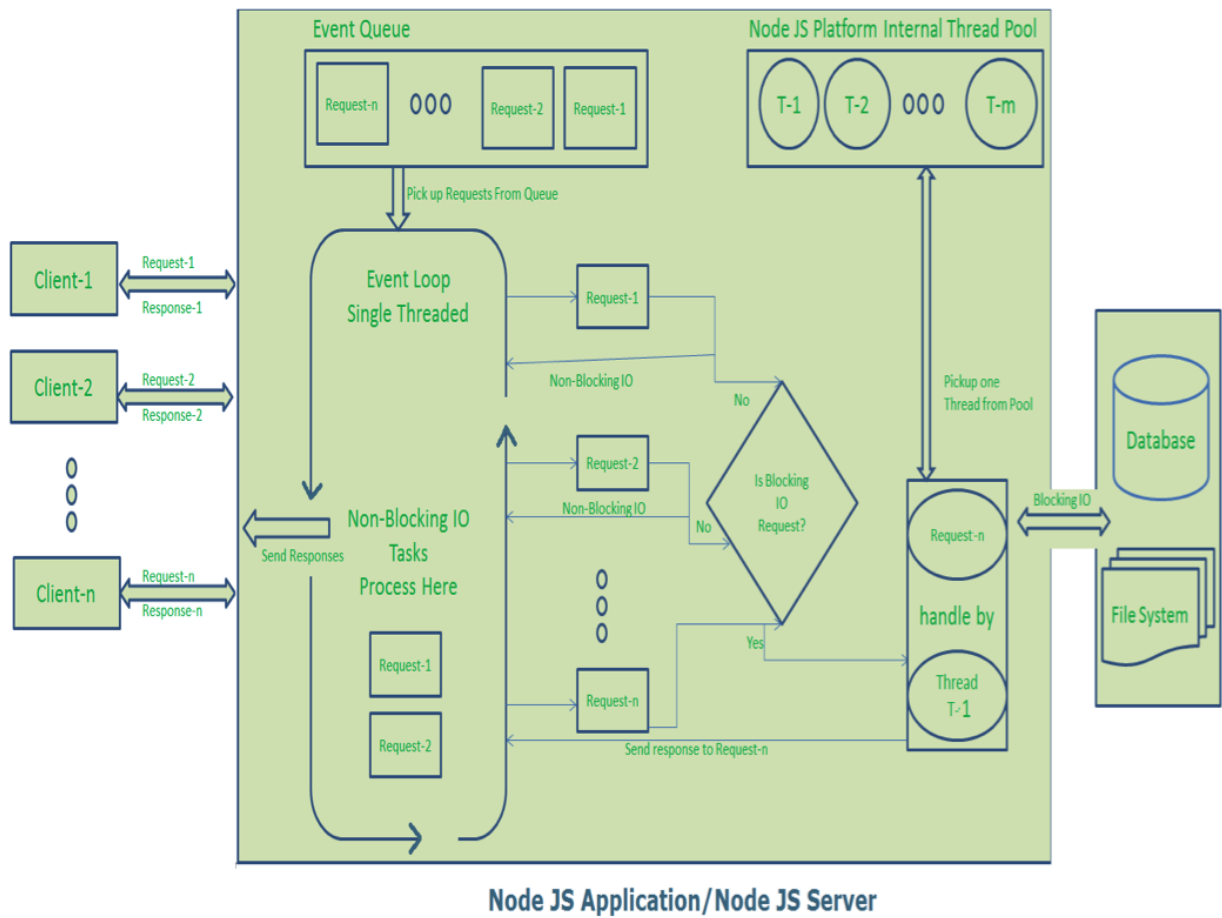
7. Deployment View

This section describes one or more physical network (hardware) configurations on which the Bank Management System is deployed and run. The system is comprised of these mandatory physical nodes: two firewalls (internal and external), a web server, a database server and a backup database server.
The diagram below is the simplicity version of the Bank Management System development view.
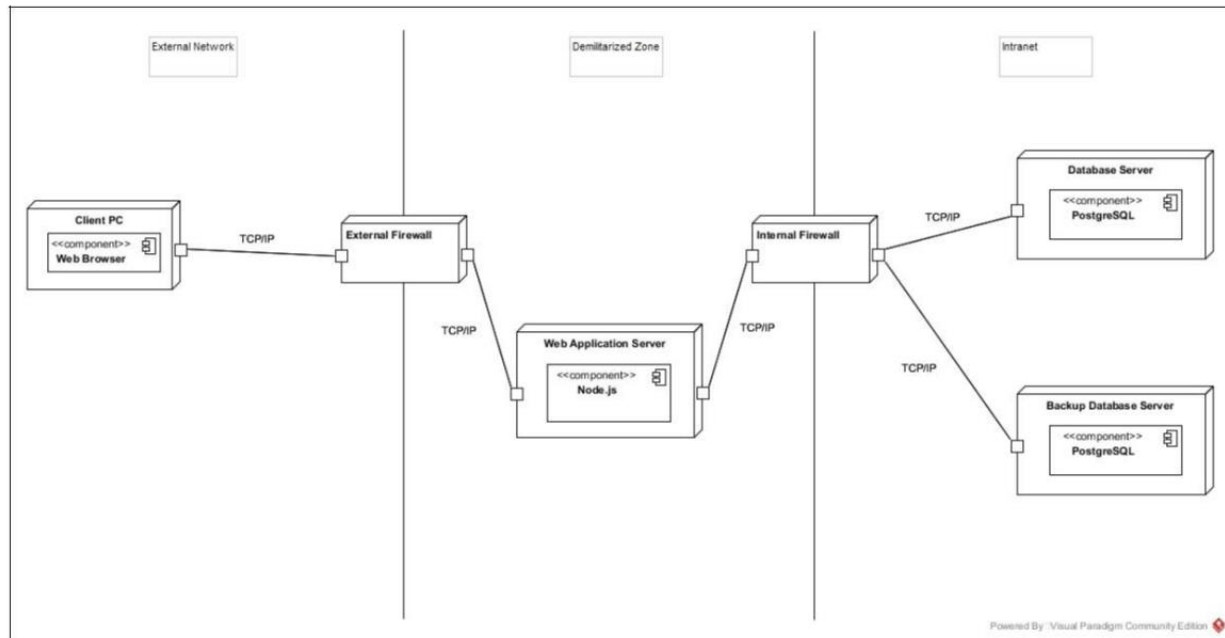
8. Implementation View

The implementation of the system is strictly driven from the design; therefore, the implementation view

will not be considered in this document.

9. Data View (optional)

10. Size and Performance

11. Quality

The system architecture supports the quality requirements:

● In order to maintain the highest degree of system integrity, the system is capable of ensuring that all information transitions are saved.

● Databases will be backed up on a daily basis in concern with safety implications.

● The system website is capable of display correctly on different devices web browser of any screen size (i.e. responsive design).

● All system website functions are available through popular web browsers; for instance, Google Chrome, Mozilla Firefox, Opera, Safari, Microsoft Edge, Internet Explorer.