|  |  |
| --- | --- |
| **daffodil international university এর চিত্র ফলাফল** | **Daffodil International University (DIU)**  **Department of Computer Science & Engineering** |

LAB PROJECT

Database Management

CSE-312

eCommerce

Project members:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Group #** | **Student ID** | **Name** | **Signature** | **Date** |
| **Group-8** | 162-15-8071 | MD Jewel |  | 19/07/18 |
| 162-15-7779 | MD Moinul Islam |  | 19/07/18 |

**Prof. Dr. Karim Mohammed Rezaul**

**Spring 2018**

**Section - E**

**Year -3, Semester -1**

Contents

[Section 1 Introduction 1](#_Toc520939831)

[1.1 Purpose 1](#_Toc520939832)

[Section 2 System Requirements 2](#_Toc520939833)

[Section 3 Process Model Design 2](#_Toc520939834)

[Section 4 Detailed Design 4](#_Toc520939835)

[4.1 User Interface Design Overview 4](#_Toc520939836)

[Section 5 Technical Architecture 8](#_Toc520939837)

[5.1 System Architecture 8](#_Toc520939838)

[5.2 Technical Architecture Design 8](#_Toc520939839)

[5.3 Performance Testing & Evaluation 16](#_Toc520939840)

[Section 6 References 19](#_Toc520939841)

[Section 7 Glossary 19](#_Toc520939842)

[Section 8 Appendices 20](#_Toc520939843)

Credentials

(It is highly recommended to login as Customer first)

Customer Email: [customernum11@gmail.com](mailto:customernum11@gmail.com)

Password: 123123

Admin Email: admin

Password: 111222

# Section 1 Introduction

## 1.1 Purpose

E-commerce encompasses the activities of purchasing and selling products through online services or over the Internet. This form of commerce leverages technologies like mobile commerce, electronic funds transfer, supply chain management, Internet marketing, online transaction processing, electronic data interchange (EDI), inventory management systems, and automated data collection systems. We've harnessed the power of JavaFx to craft an exceptional interface for both customers and the administrative panel. Our aim is to provide a seamless experience for our customers. Additionally, we've integrated various APIs including Mail, JFonex, controllsJFX, MY-SQL-connector, and more.

The primary objective of this design is to create a relational database structure and implement normalization in a methodical manner, ensuring that the database is well-suited for versatile querying and devoid of certain undesirable characteristics. These include anomalies in insertion, update, and deletion that could potentially compromise data integrity.

In our design, we've employed Entities, which are composed of attributes. An attribute represents a characteristic of an entity. Each attribute falls into one of three categories:

1. Descriptive – An attribute is descriptive if it doesn't serve to identify or relate but is used to illustrate or express a characteristic of an entity occurrence.
2. Identifying – An attribute that identifies serves as a candidate key. If the value of an identifying attribute changes, it should result in the identification of a different entity occurrence. Such attributes should remain unchangeable and immutable.
3. Relational – An attribute that establishes relationships between entities is a foreign key. This attribute refers to the primary key attribute of another (or the same) entity occurrence

# How does it work :

# The process is fairly straightforward. Initially, the customer has the option to either sign up, or if they are already registered, they can simply log in to their account. Following this, all they have to do is select a product and complete the transaction. That's all there is to it for the customer. In contrast, the admin has the capability to log in and gain complete control over the project.

# Section 2 System Requirements

*A list of Software/System Requirements Specifications to the design elements is given below here:*

API:

* 1. Jfoenix 8.0.4.jar
  2. Controlsfx -8.40.jar
  3. Mail.jar
  4. My-sql-connnector-java-8.0.11
  5. JDK-1.8

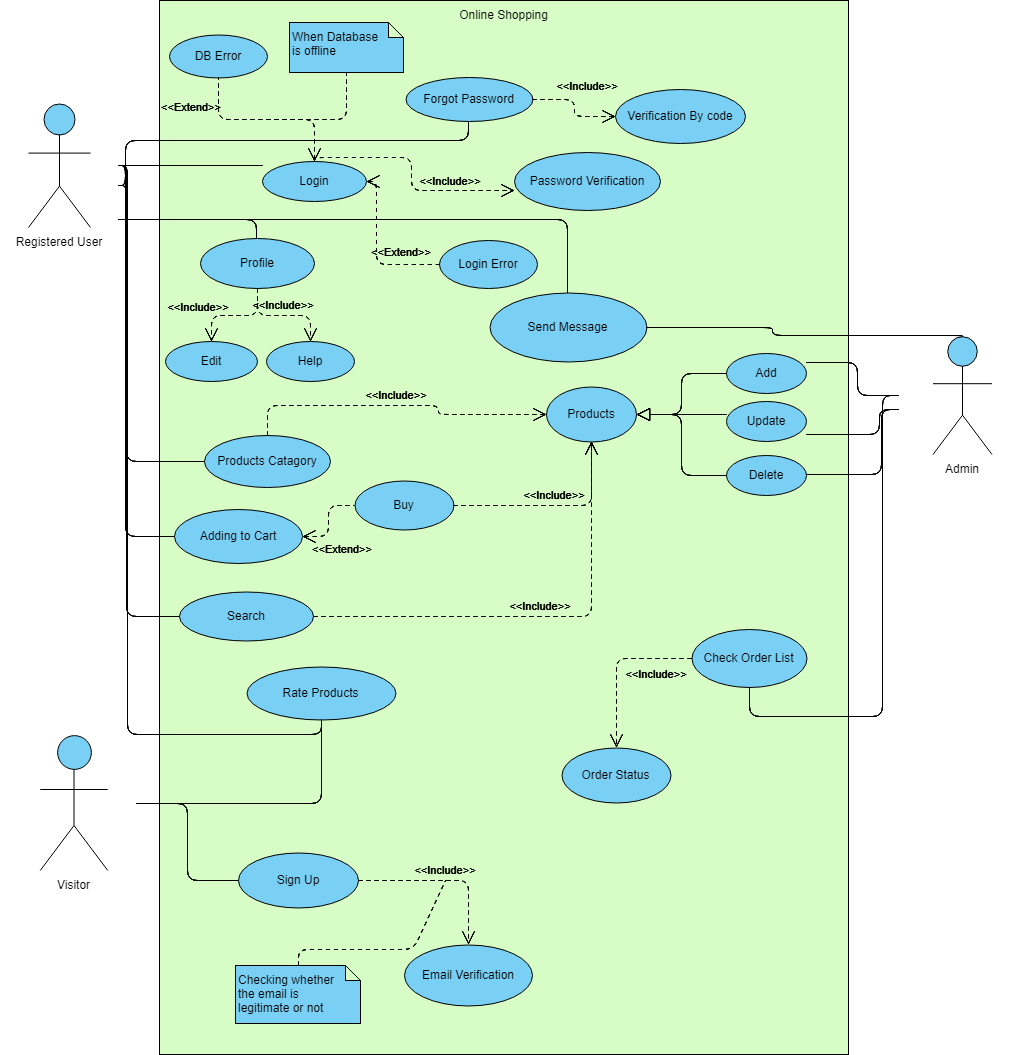
Software:

1. NetBeansIDE with SceneBuilder installed
2. MySql WorkBench
3. Mysql Server

# 

# Section 3 Process Model Design

At its core, a use case diagram serves as a visual representation of a user's interaction with the system, illustrating the connections between the user and the various use cases they engage in. This diagram is instrumental in identifying the diverse user roles within a system and the corresponding use cases. Often, it is complemented by other types of diagrams.

 Fig: Use Case diagram

# Section 4 Detailed Design

## 4.1 Overview of User Interface Design

## The system prototype for the ecommerce project prioritizes a highly functional interface. In this section, we have integrated numerous distinctive features, which will be highlighted. The prototype for the eCommerce project is provided below:

## The initial component pertains to the login page within the project:

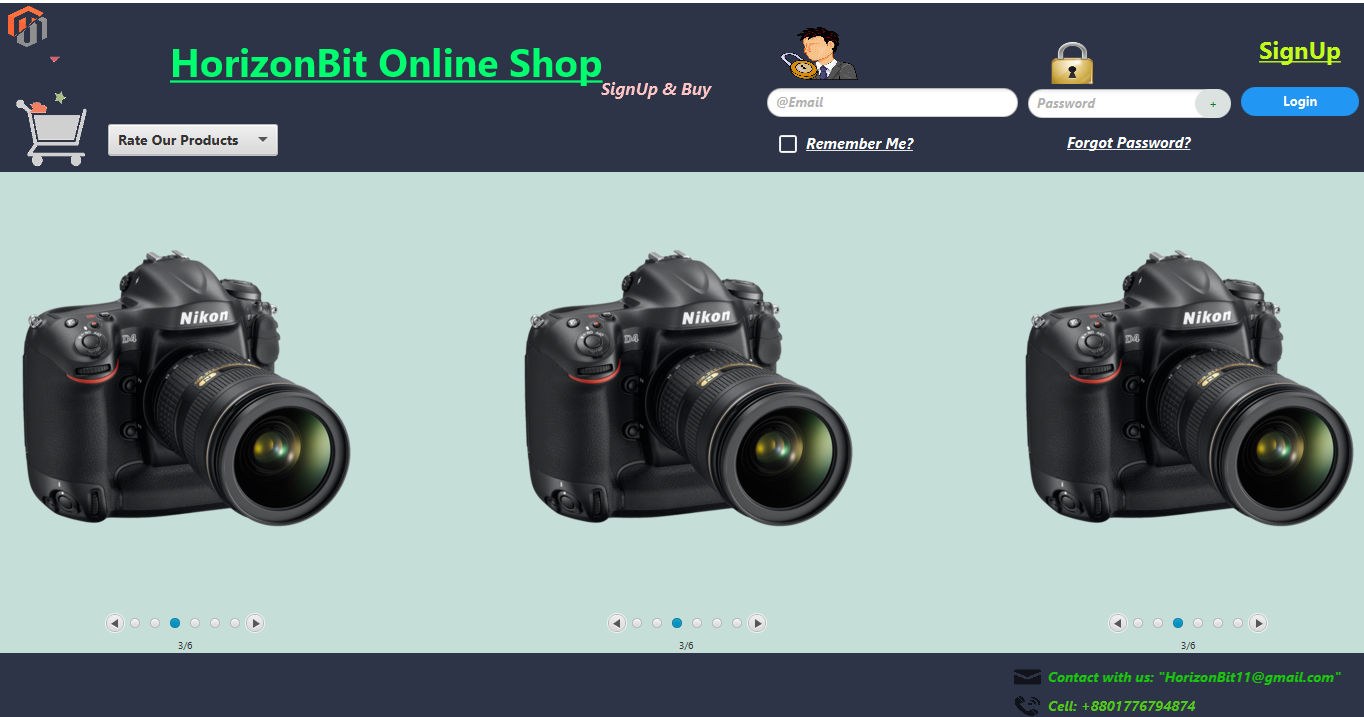
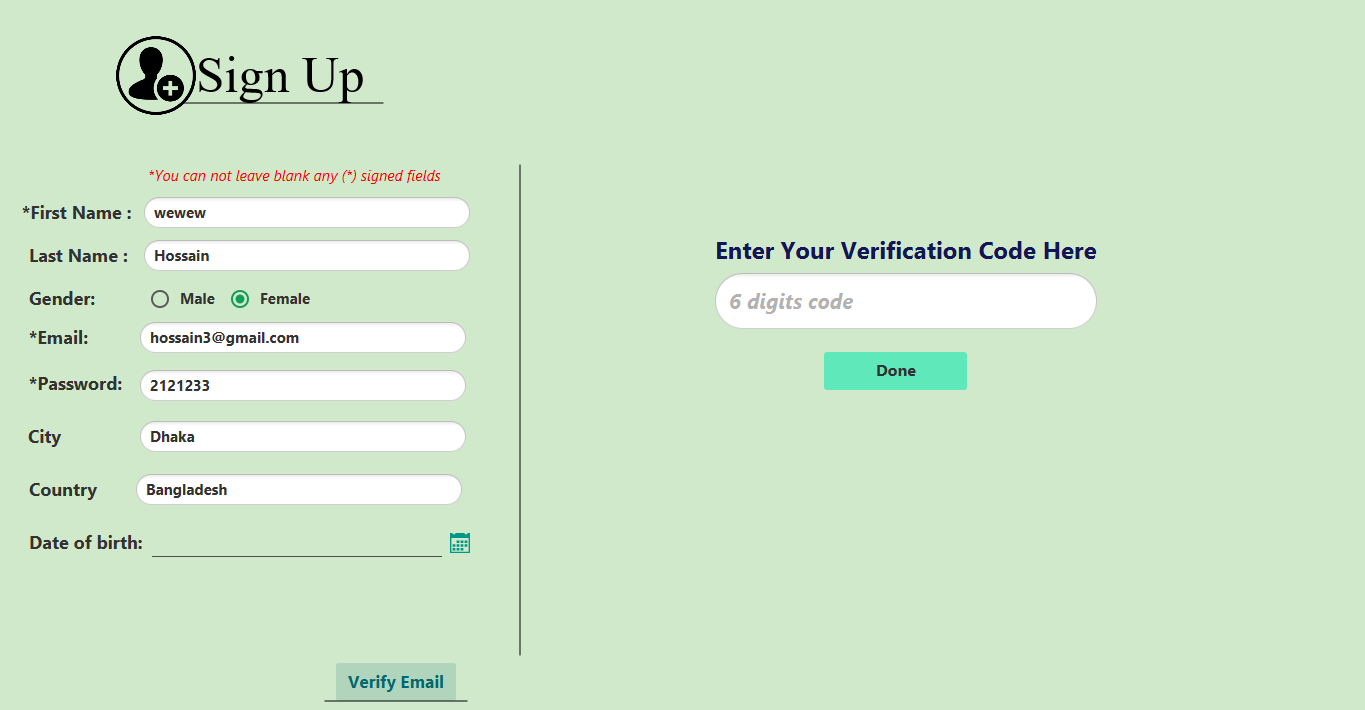


Fig: Login Page

 FIG: SignUp Page

Verification Process:

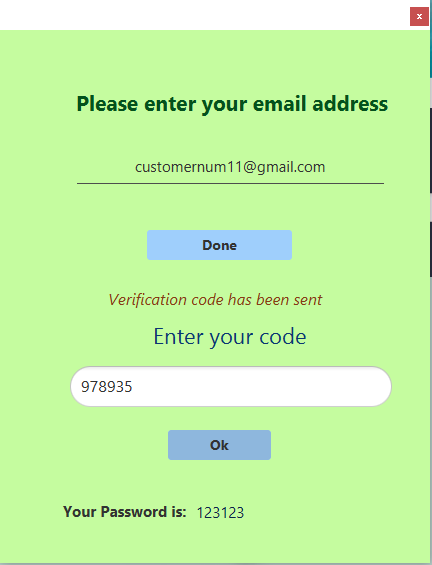


Fig: Forgot Password

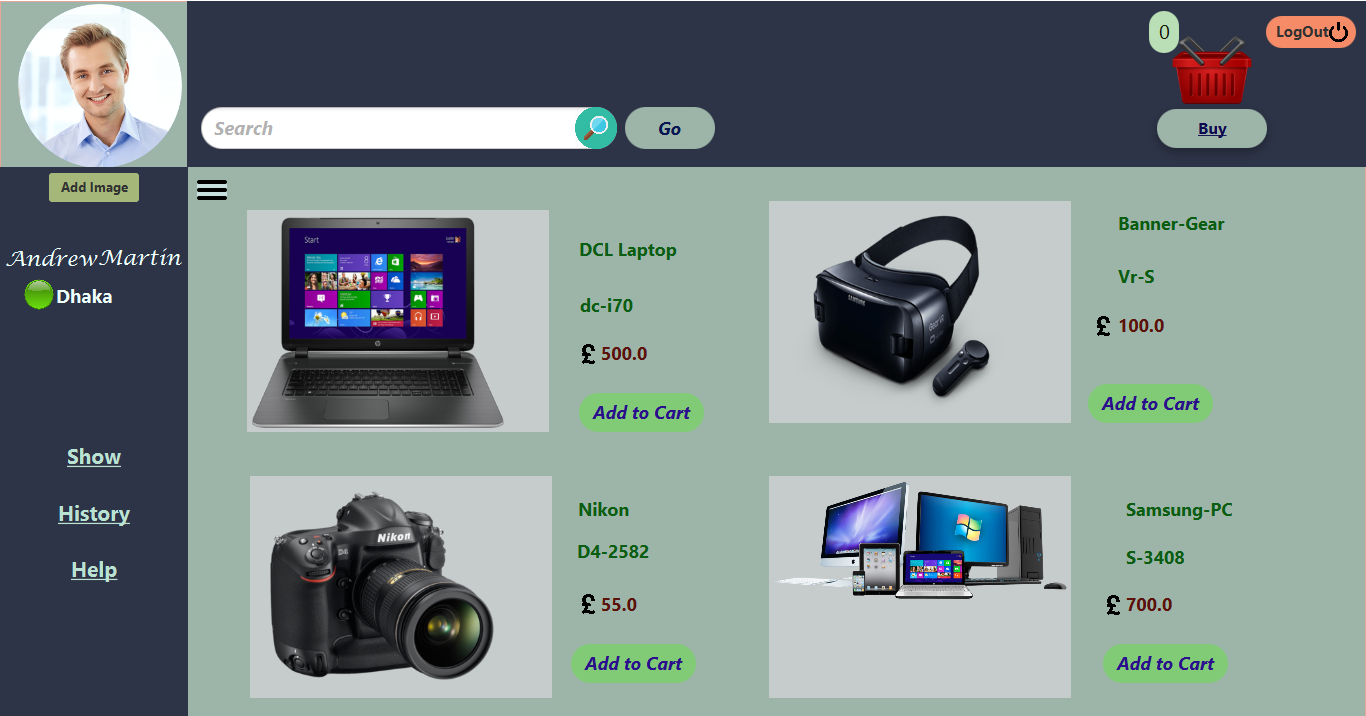


Fig: Customer Home Page

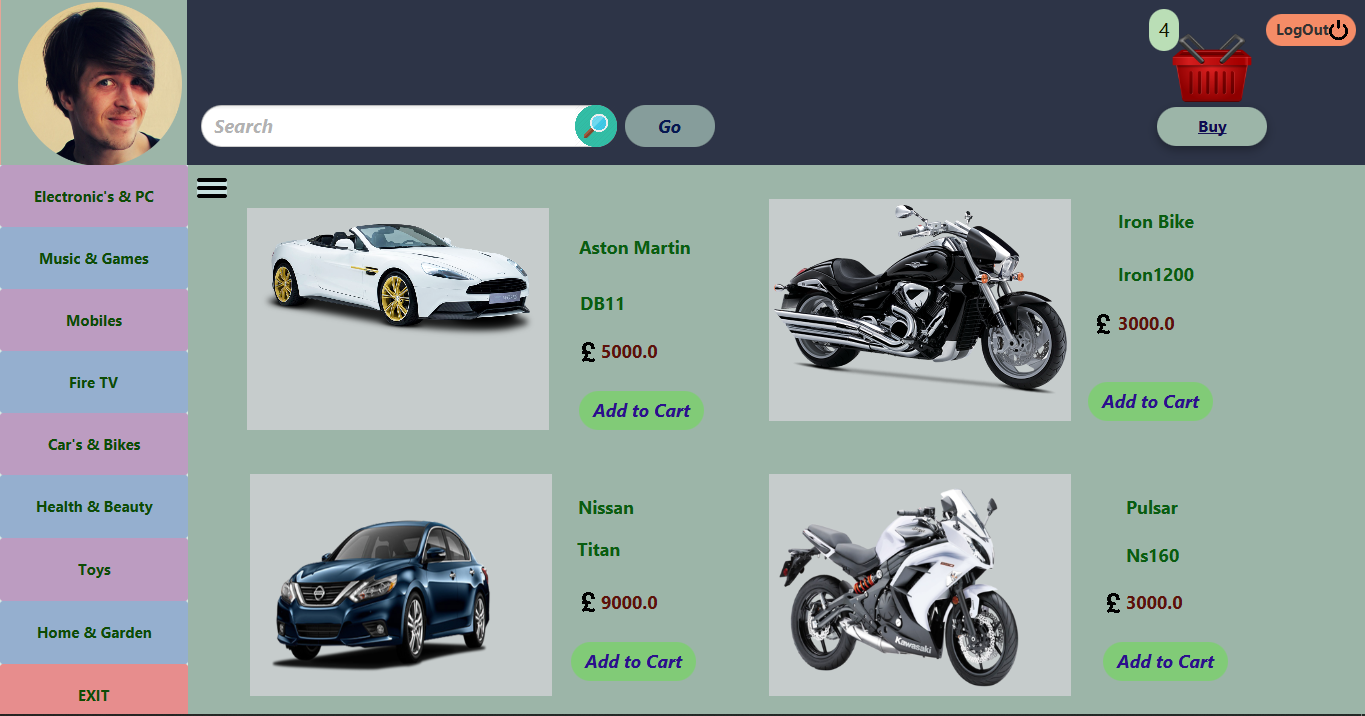


FIG: Customer’s selection from category

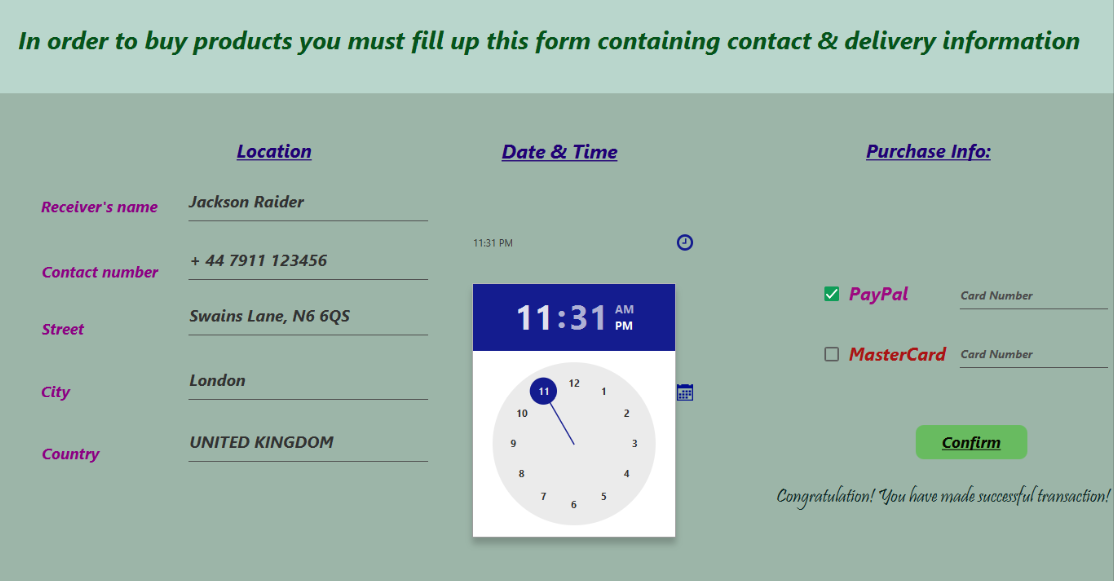


Fig: Order Form

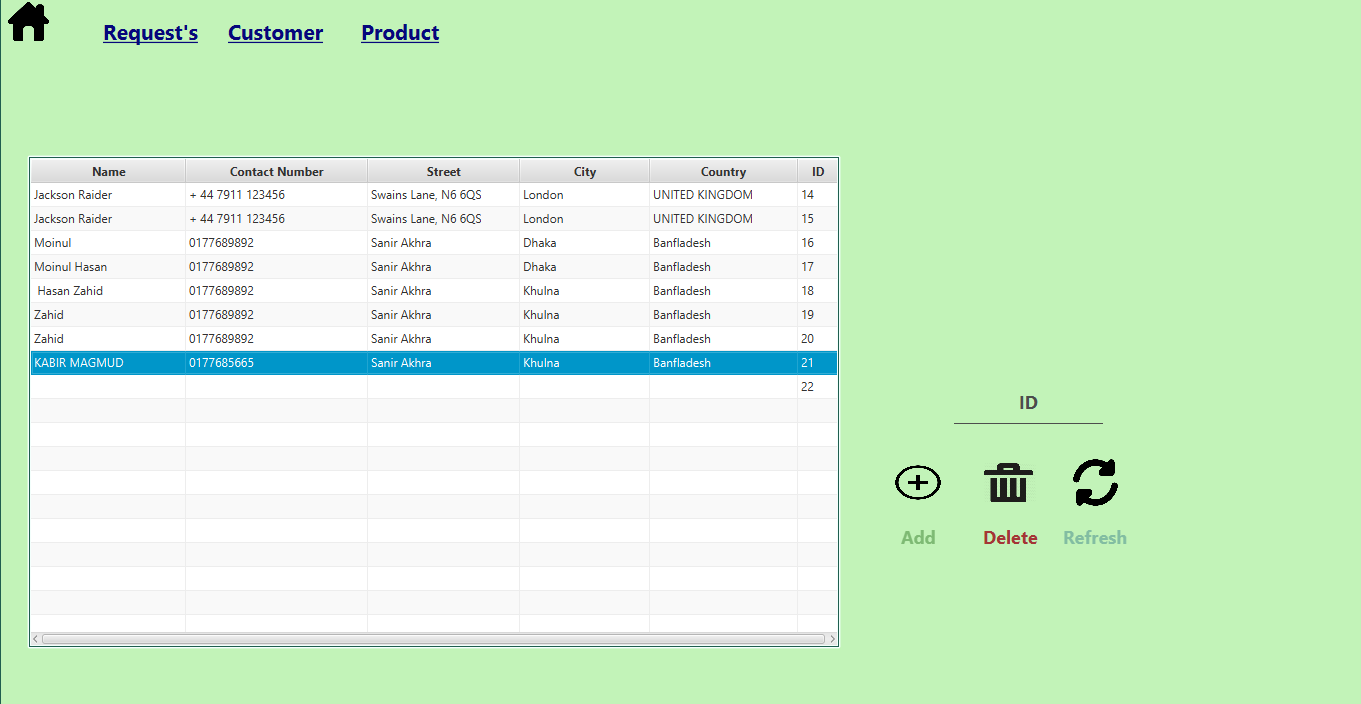


FIG: Admin home

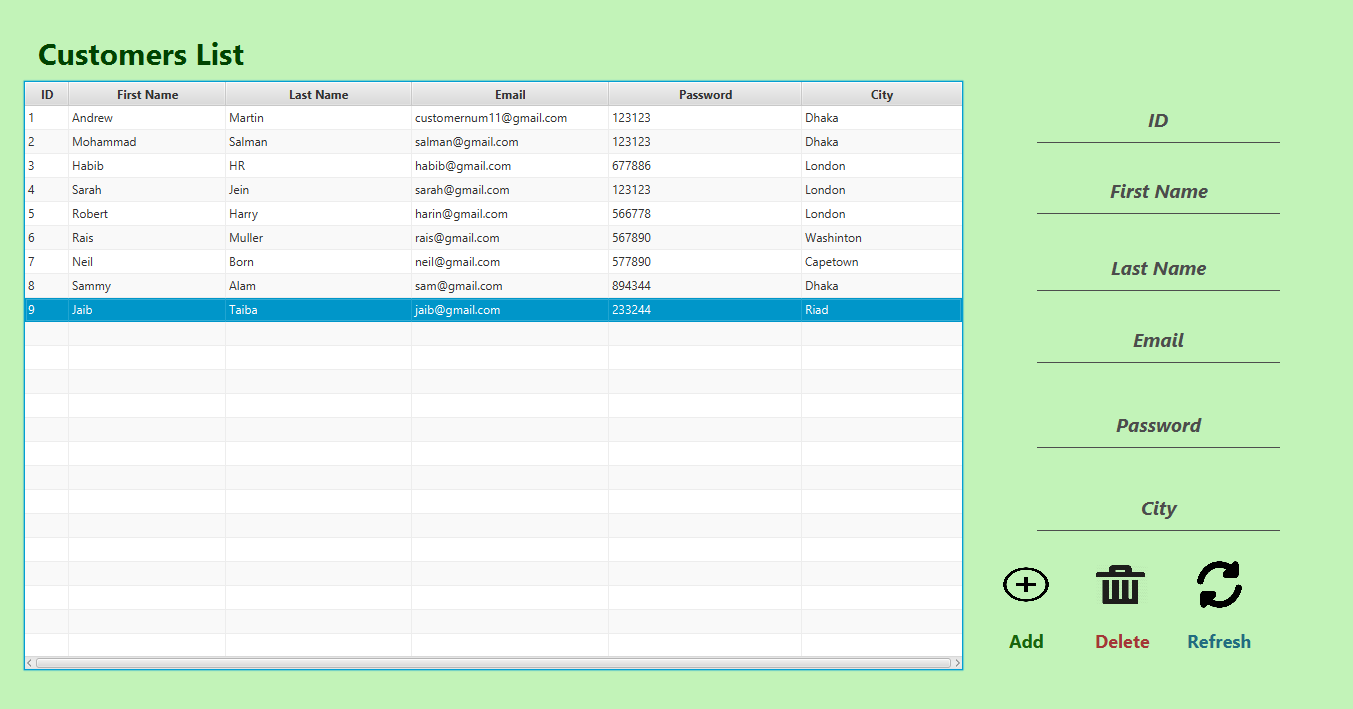


Fig: Admin as customer controller

4.2 User Interface Navigation Hierarchy

⇒

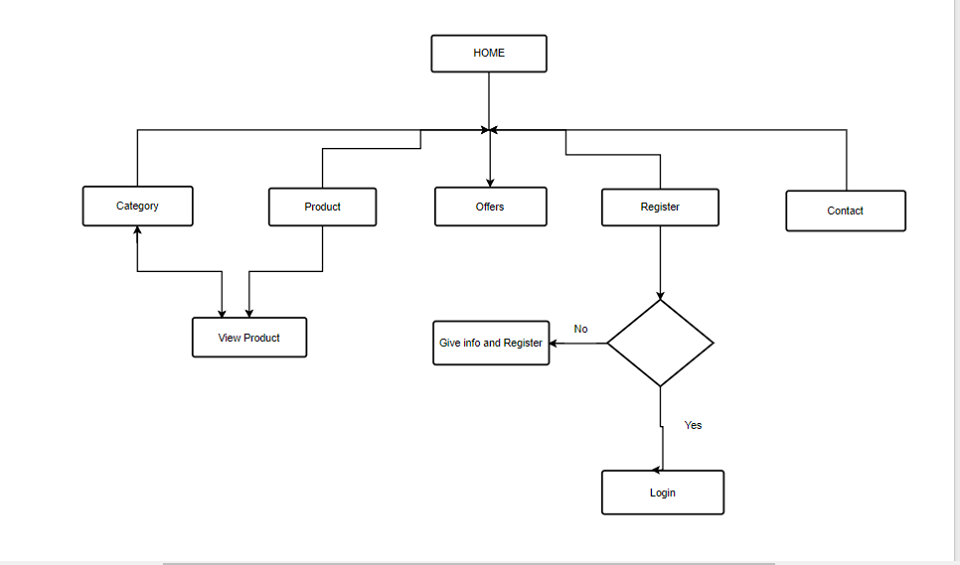


Fig: Customer Navigation

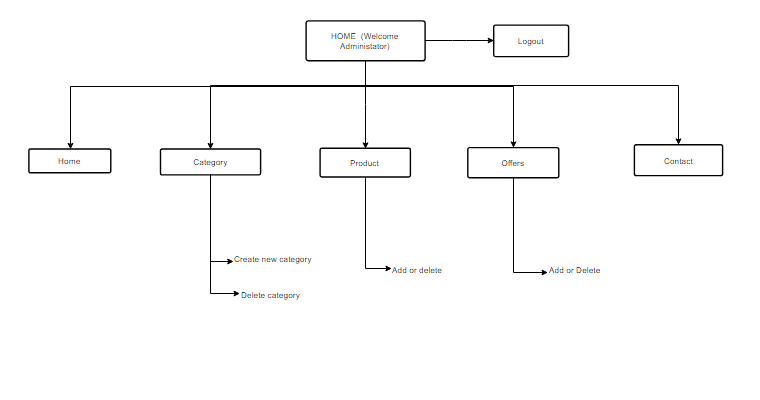


Fig: Admin Navigation

# Section 5 Technical Architecture

## 5.1 System Architecture

OS: Windows 10

Installed Memory: 4gb

Processor: Intel Core i3

CPU 2.40GHz

This project has been built in windows 10 environment.

Language:

English language has been used in this project.

Software:

MySql workbench

MySql Server

NetBeans IDE

## 5.2 Technical Architecture Design

### 5.2.1 Class Diagram

In the Unified Modeling Language (UML), a class diagram is a form of static structure diagram that delineates a system's structure. It showcases the system's classes, including their attributes, operations (or methods), and the relationships among objects.

Class diagrams serve as the foundational element in object-oriented modeling. They are employed for both broad conceptual modeling of the application's systematics and for granular modeling, translating these models into programming code. Additionally, class diagrams can be utilized for data modeling. The classes within a class diagram encompass the primary elements and interactions within the application, as well as the classes slated for programming.

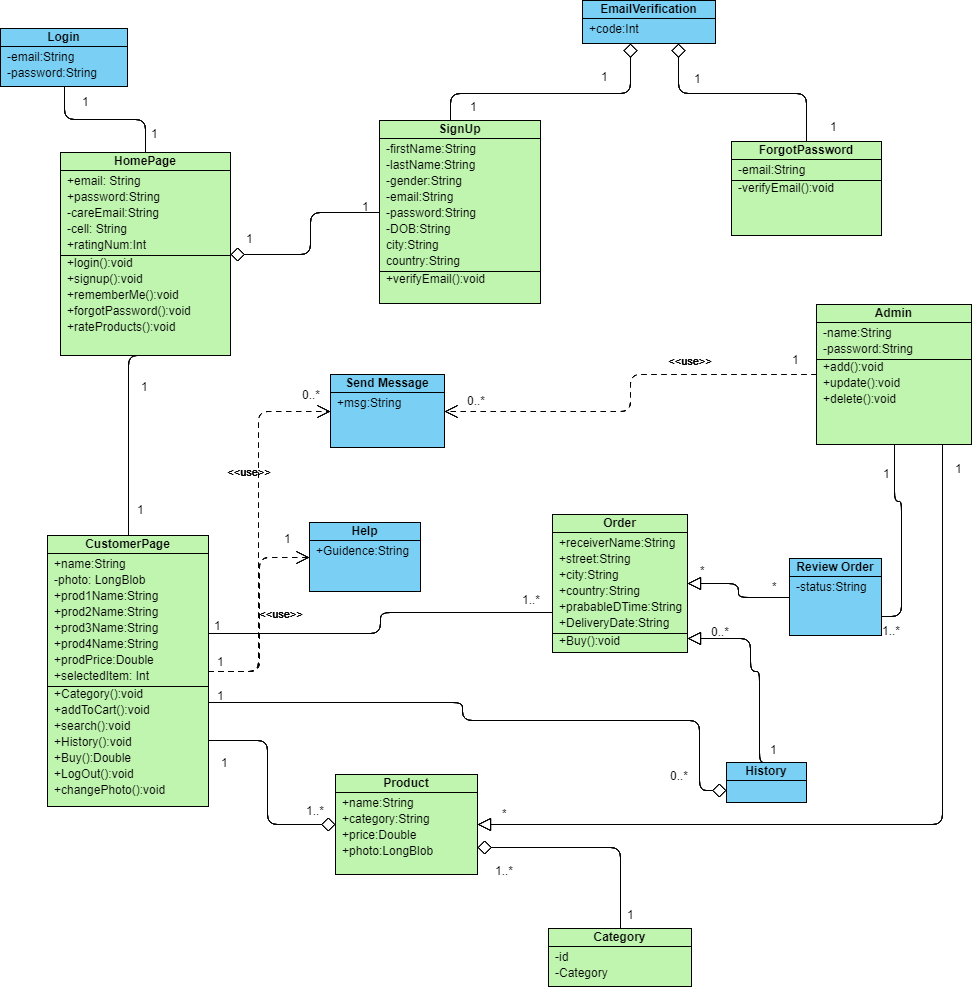


Fig: Class Diagram

### 5.2.2 Object Mapping

An Entity-Relationship model, abbreviated as ER model, delineates the interconnected entities of significance within a defined domain of knowledge. This fundamental model comprises entity types, which categorize the pertinent elements, and defines relationships that may exist between instances of these entity types.

**Schema:**

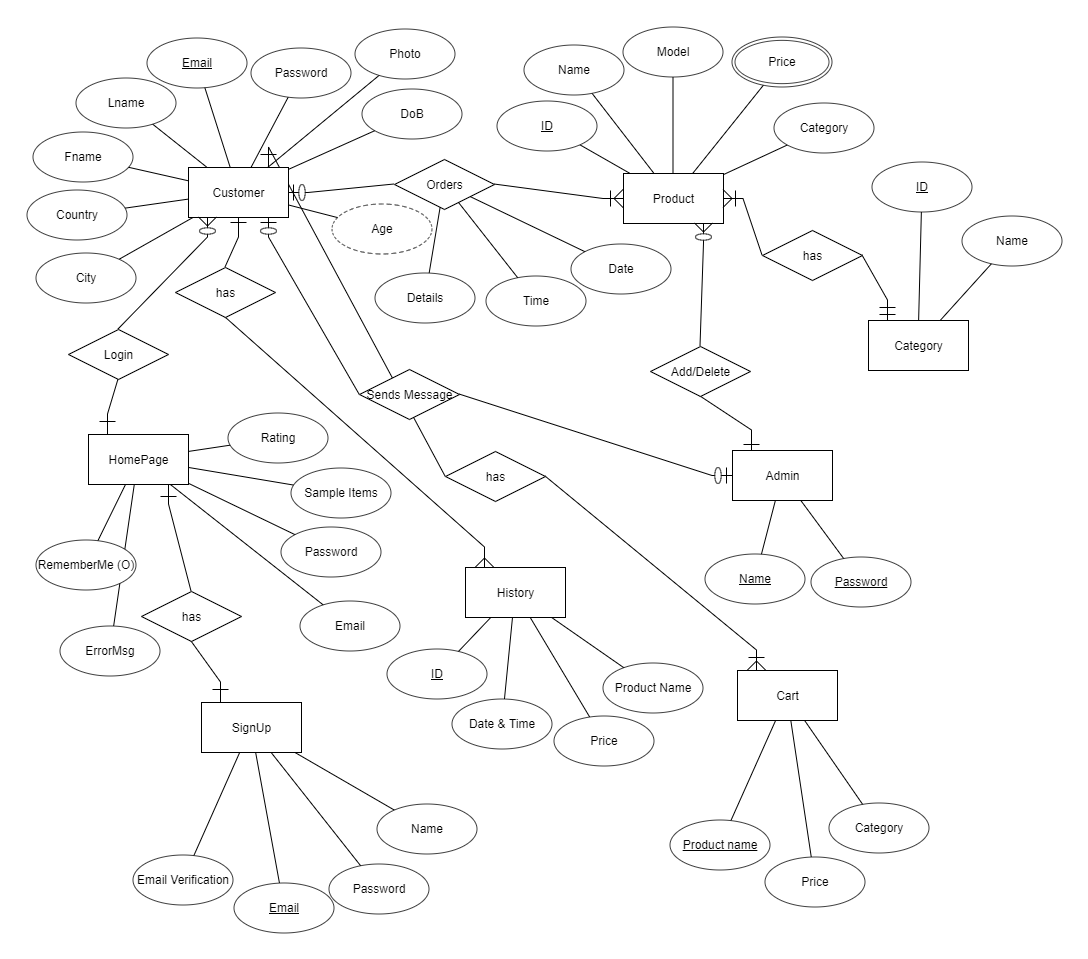
The database schema in a database system represents its formalized structure, articulated in a language supported by the database management system (DBMS). The term 'schema' denotes the arrangement of data, essentially serving as a blueprint outlining how the database is organized, particularly evident in the division into database tables, as is the case with relational databases.

Fig : ERD diagram

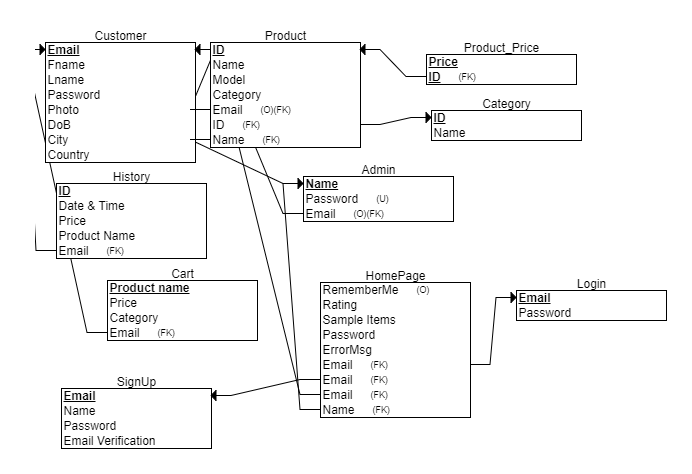


Fig: ERD schema

### 5.2.3 Interface Control Document-

⇒

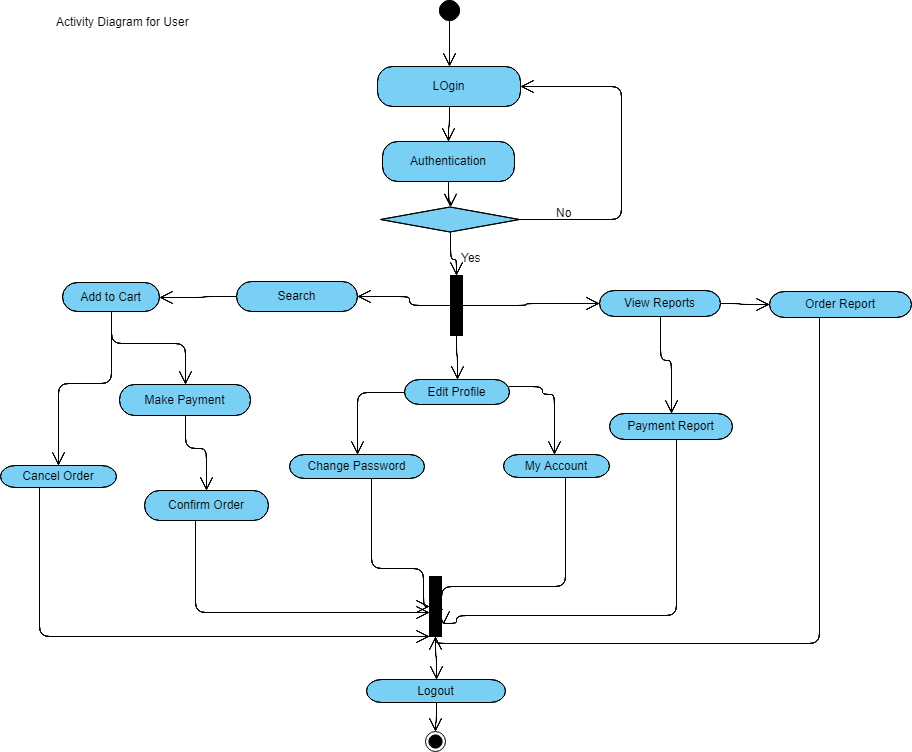


Fig : User activity

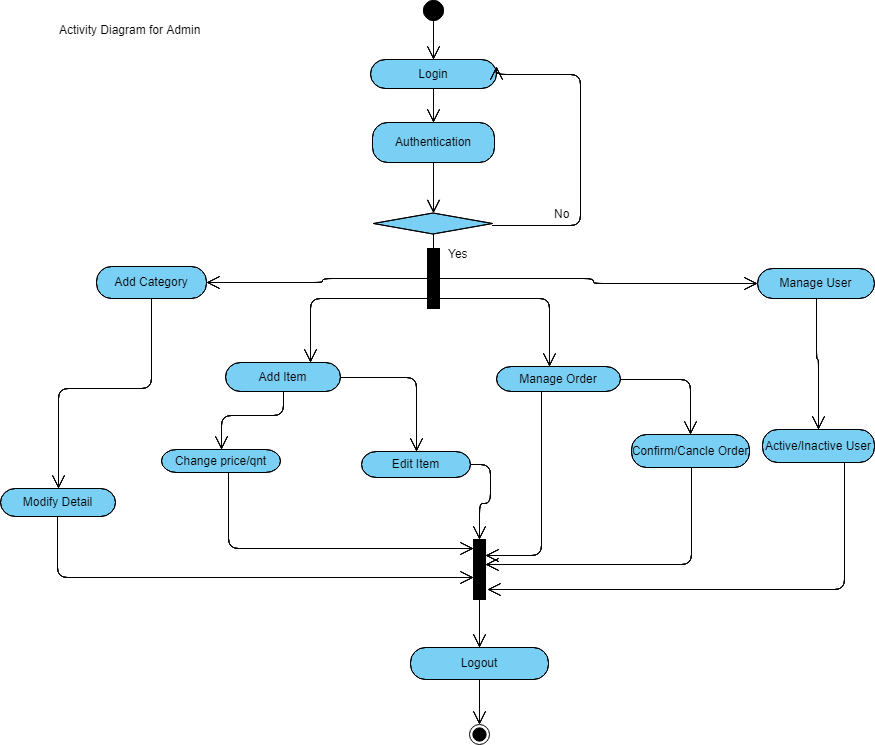


Fig: Admin activity

### 5.2.4 Sequence Diagrams,

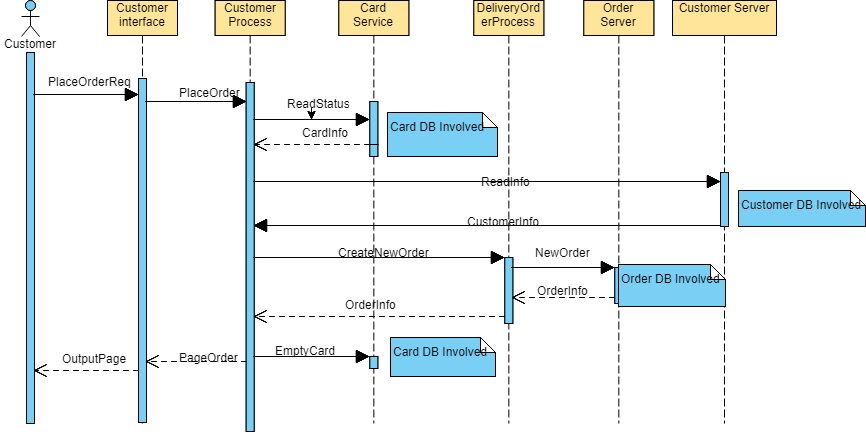
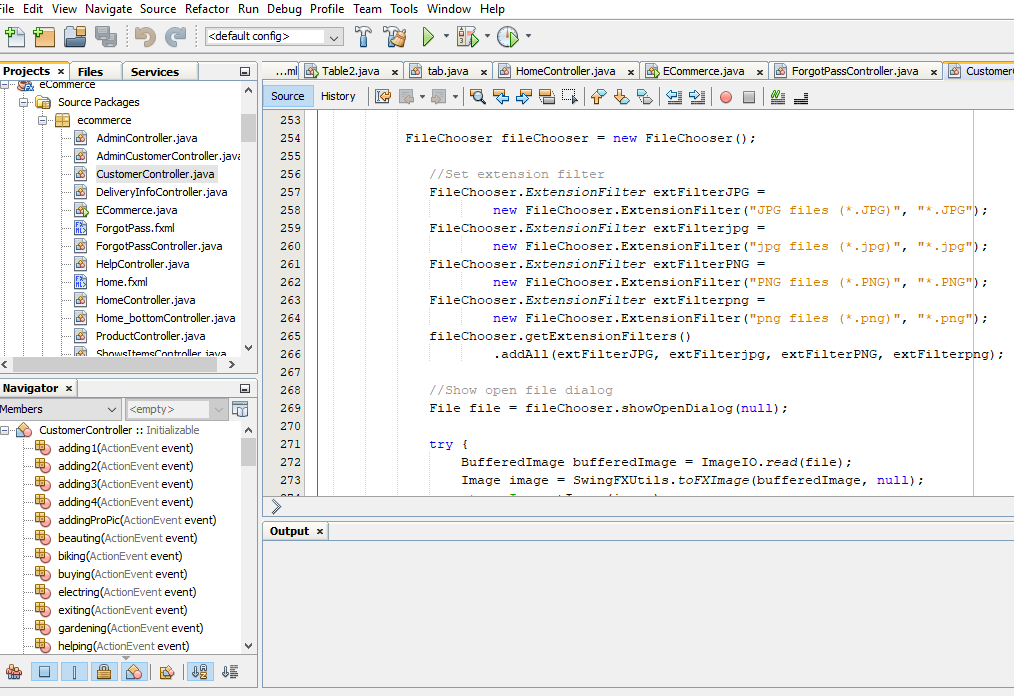
Object Behavior Model: A sequence diagram illustrates interactions between objects in a chronological order, presenting the objects and classes participating in a given scenario. It outlines the sequential flow of messages exchanged among these objects, essential for executing the functionality outlined in the scenario⇒

Fig: Sequence Diagram

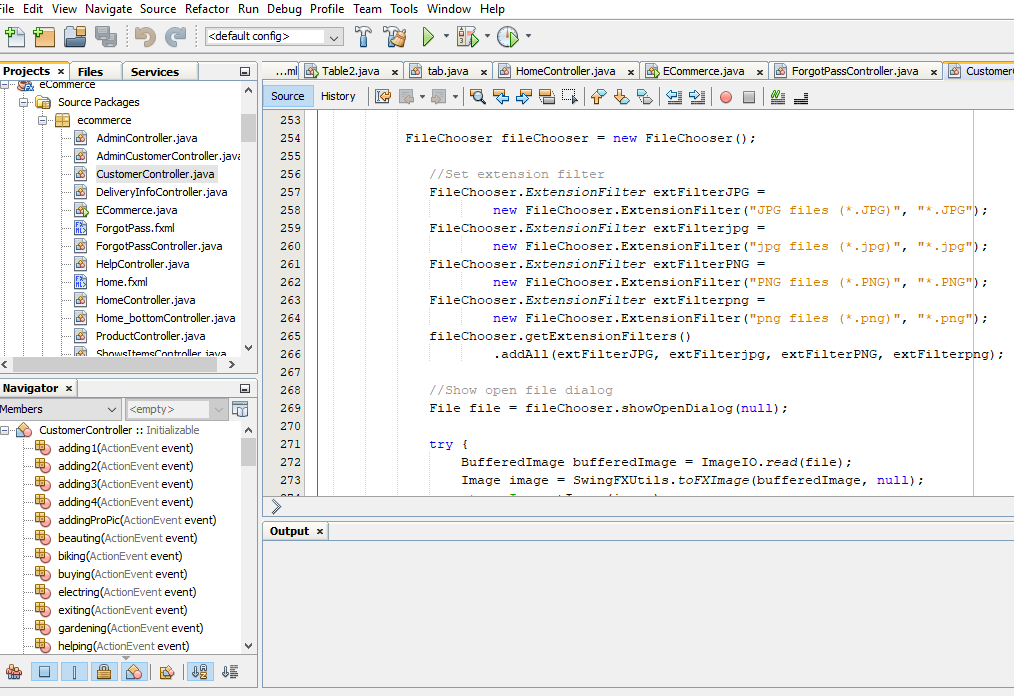
## SceneBuilder5.3 Performance Testing & Evaluation

*Some basic screen shot of work environment has been given below to provide a better understanding of the whole project in a nustle.*

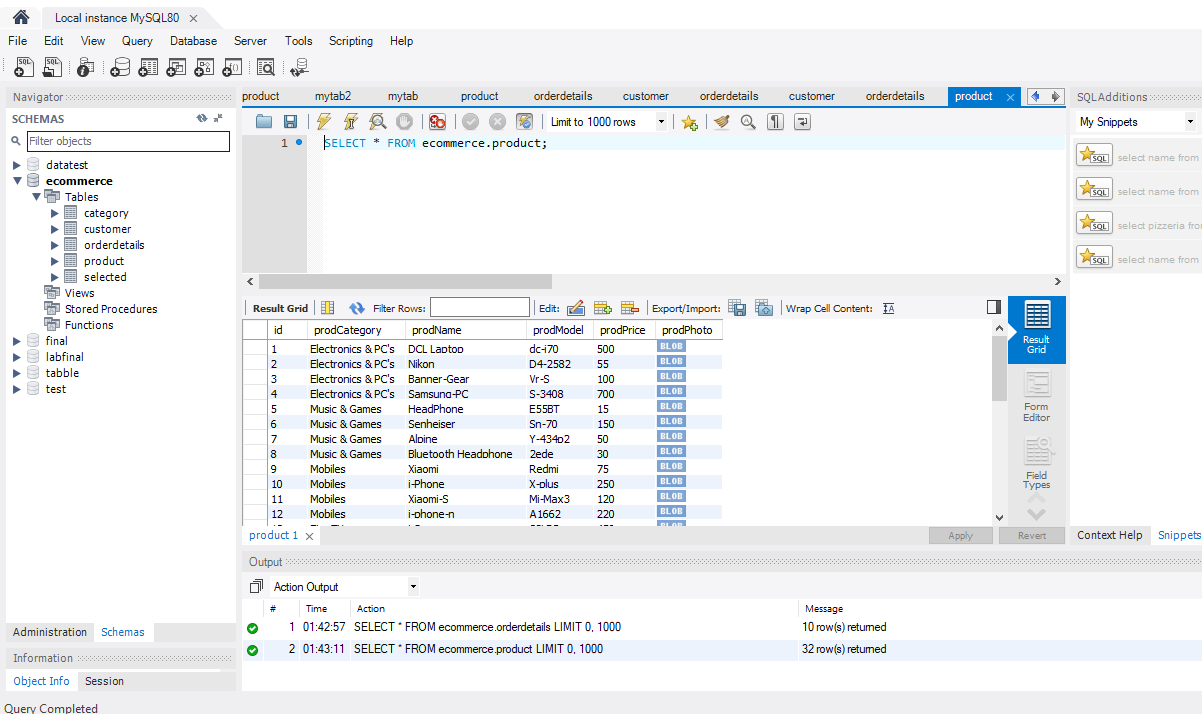
***Scene Builder***



***Fig: Scene Builder***

******

***Fig: NetBeans Environment***

******

***FIG: Mysql Workbench***

# Section 6 References

| Document No. | Document Title | Date | Author |
| --- | --- | --- | --- |
| 1 | Ecommerce infrastructure | 19/7/18 | McGraw-Hill |
| 2 | Ecommerce introduction | 19/7/18 | Craig Mullins  (Consultant at Mullins Consulting, Inc) |
| 3 | Database Schema | 19/7/18 | Rybinski, H. (1987). "On First-Order-Logic Databases" |
| 4 | Google search bar | 1/8/18 |  |
| 5 | Google Image | 1/8/18 |  |
| 6 | Wikipedia | 1/8/18 |  |

# Section 7 Glossary

⇒

**API:** A collection of functions and procedures enabling the development of applications to access the functionalities or data of an operating system, application, or other service.

**Workbench:** MySQL Workbench serves as a comprehensive visual instrument designed for database architects, developers, and DBAs. It offers features for data modeling and SQL development.

# 

# Section 8 Appendices

⇒

To execute the project, ensure that the database is appropriately configured, and NetBeans is preinstalled. Activate the MySQL server while running this program. Additionally, ensure all APIs are correctly linked to the project's libraries.