

C O N T E N T S

1. Introduction	2
2. Objective & Scope of the Project	3
3. Theoretical Background	4
4. Problem Definition & Analysis	8
5. System Design & Development	9
5.1 Database Design:	9
5.2 I/O Forms Design & Event Coding:	11
6. References	43

1. INTRODUCTION

This Project is basically created so that the user can easily buy their games at reasonable cost.

The purpose of the software project is to develop the Management Information System (MIS) to automate the record keeping of Account Information, Games They Buy.

A MIS mainly consists of a computerized database, a collection of inter-related tables for a particular subject or purpose, capable to produce different reports relevant to the user. An application program is tied with the database for easy access and interface to the database. Using Application program or front-end, we can store, retrieve and manage all information in proper way.




Project, Java NetBeans IDE, a powerful front-end tool is used for getting Graphical User Interface (GUI) based integrated platform and coding simplicity. As a back-end a powerful, open source RDBMS, My SQL is used as per requirement of the CBSE curriculum of Informatics Practices Course.

2. OBJECTIVE & SCOPE OF THE PROJECT

Basically the objective of this program is that the user doesn't need to search different places or sites for their favorites games, they can get all of them at one place.

This software project is also aimed to enhance the current record keeping system, which will help managers to retrieve the up-to-date information at right time in right shape.

The proposed software system is expected to do the following functionality-

-  The proposed system should maintain all the records and transactions, and should generate the required reports and information when required.
-  To provide graphical and user-friendly interface to interact with a centralized database based on client-server architecture.
-  To identify the critical operation procedure and possibilities of simplification using modern IT tools and practices.

So far as future scope of the project is concerned, firstly it is open to any modular expansion i.e. other modules or functions can be designed and embedded to handle the user need in future. Any part of the software and reports can be modified independently without much effort.

3. THEORETICAL BACKGROUND

3.1 What is Database?

Introduction and Concepts:

A database is a collection of information related to a particular subject or purpose, such as tracking customer orders or maintaining a music collection. Using any RDBMS application software like MS SQL Server, MySQL, Oracle, Sybase etc, you can manage all your information from a single database file. Within the file, divide your data into separate storage containers called tables. You may add and retrieve the data using queries.

A table is a collection of data about a specific topic, such as products or suppliers. Using a separate table for each topic means you can store that data only once, which makes your database more efficient and reduces data-entry errors. Table organises data into columns (called fields) and rows (called records).

A Primary key is one or more fields whose value or values uniquely identify each record in a table. In a relationship, a primary key is used to refer to specific record in one table from another table. A primary key is called foreign key when it is referred to from another table.

To find and retrieve just the data that meets conditions you specify, including data from multiple tables, create a query. A query can also update or delete multiple records at the same time, and perform built-in or custom calculations on your data.

Role of RDBMS Application Program:

A computer database works as a electronic filing system, which has a large number of ways of cross-referencing, and this allows the user many different ways in which to re-organize and retrieve data. A database can handle business inventory, accounting and filing and use the information in its files to prepare summaries, estimates and other reports. The management of data in a database system is done by means of a general-purpose software package called a Database Management System (DBMS). Some commercially available DBMS are MS SQL Server, MS ACCESS, INGRES, ORACLE, and Sybase. A database management system, therefore, is a combination of hardware and software that can be used to set up and monitor a database, and can manage the updating and retrieval of database that has been stored in it.

Most of the database management systems have the following capabilities:

- ◆ Creating of a table, addition, deletion, modification of records.
- ◆ Retrieving data collectively or selectively.
- ◆ The data stored can be sorted or indexed at the user's discretion and direction.
- ◆ Various reports can be produced from the system. These may be either standardized report or that may be specifically generated according to specific user definition.

- ◆ Mathematical functions can be performed and the data stored in the database can be manipulated with these functions to perform the desired calculations.
- ◆ To maintain data integrity and database use.

The DBMS interprets and processes users' requests to retrieve information from a database. In most cases, a query request will have to penetrate several layers of software in the DBMS and operating system before the physical database can be accessed. The DBMS responds to a query by invoking the appropriate subprograms, each of which performs its special function to interpret the query, or to locate the desired data in the database and present it in the desired order.

3.2 What is MySQL ?



The management of data in a database system is done by means of a general-purpose software package called a Database Management System (DBMS). Some commercially available RDBMS are MS SQL Server, MS ACCESS, INGRES, ORACLE, and Sybase.

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. MySQL is named after co-founder Monty Widenius's daughter, My. The name of the MySQL Dolphin (our logo) is "Sakila".

- **MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL is based on SQL.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of "MySQL" stands for "Structured Query Language." SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, "SQL-92" refers to the standard released in 1992, "SQL:1999" refers to the standard released in 1999, and "SQL:2003" refers to the current version of the standard.

- **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License),

- **The MySQL Database Server is very fast, reliable, and easy to use.**

If that is what you are looking for, you should give it a try. MySQL Server also has a practical set of features developed in close cooperation with our users. You can find a performance comparison of MySQL Server with other database managers on our benchmark page. MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under

constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

- **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

The Main Features of MySQL

- Written in C and C++.
- Works on many different platforms.
- Provides the server as a separate program for use in a client/server networked environment, and as a library that can be embedded (linked) into standalone applications. Such applications can be used in isolation or in environments where no network is available.
- Password security by encryption of all password traffic when you connect to a server.
- Support for large databases. We use MySQL Server with databases that contain 50 million records. We also know of users who use MySQL Server with 200,000 tables and about 5,000,000,000 rows.
- MySQL client programs can be written in many languages. A client library written in C is available for clients written in C or C++, or for any language that provides C bindings.
- APIs for C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, and Tcl are available, enabling MySQL clients to be written in many languages.
- The Connector/ODBC (MyODBC) interface provides MySQL support for client programs that use ODBC (Open Database Connectivity) connections.
- The Connector/J interface provides MySQL support for Java client programs that use JDBC connections. Clients can be run on Windows or Unix. Connector/J source is available.

3.3 What is NetBeans IDE ?

NetBeans started as a student project (originally called Xelfi) in the Czech Republic in 1996. The goal was to write a Delphi-like Java IDE in Java. Xelfi was the first Java IDE (Integrated Development Environment) written in Java, with its first pre-releases in 1997. Xelfi was a fun project to work on, especially since Java IDE space was uncharted territory at that time. The project attracted enough interest that these students, once they graduated, decided that they could market it as a commercial product. Soliciting resources from friends and relatives for a web space, they formed a company around it.

Soon after, they were contacted by [Roman Stanek](#), an entrepreneur who had already been involved in several startups in the Czech Republic. He was looking for a good idea to invest in, and discovered Xelfi. He met with the founders; they hit it off, and a business was born.

In the spring of 1999, [NetBeans DeveloperX2](#) was released, supporting Swing. The performance improvements that came in JDK 1.3, released in the fall of 1999, made

NetBeans a viable choice for development tools. By the summer of 1999, the team was hard at work re-architecting DeveloperX2 into the more modular NetBeans that forms the basis of the software today.

Something else was afoot in the summer of 1999: [Sun Microsystems](#) wanted better Java development tools, and had become interested in NetBeans. It was a dream come true for the NetBeans team: NetBeans would become the flagship tool set of the maker of Java itself! By the Fall, with the next generation of NetBeans Developer in beta, a deal was struck. Sun Microsystems had also acquired another tools company, During the acquisition, the young developers who had been involved in open-source projects for most of their programming careers, mentioned the idea of open-sourcing NetBeans. Fast forward to less than six months later, the decision was made that NetBeans would be open sourced. While Sun had contributed considerable amounts of code to open source projects over the years, this was Sun's first *sponsored* open source project, one in which Sun would be paying for the site and handling the infrastructure.

Features of NetBeans

A free, open-source Integrated Development Environment for software developers. You get all the tools you need to create professional desktop, enterprise, web, and mobile applications with the Java platform, as well as C/C++, PHP, JavaScript, Groovy, and Ruby.

NetBeans IDE 6.9 introduces the JavaFX Composer, support for JavaFX SDK 1.3, OSGi interoperability, support for the PHP Zend framework and Ruby on Rails 3.0, and more.

4. PROBLEM DEFINITION & ANALYSIS






The hardest part of building a software system is deciding precisely what to build. No other part of the conceptual work is so difficult as establishing the detailed technical requirement. Defining and applying good, complete requirements are hard to work, and success in this endeavor has eluded many of us. Yet, we continue to make progress.

Problem definition describes the *What* of a system, not *How*. The quality of a software product is only as good as the process that creates it. Problem definition is one of the most crucial steps in this creation process. Without defining a problem, developers do not know what to build, customers do not know what to expect, and there is no way to validate that the built system satisfies the requirement.

Problem definition and Analysis is the activity that encompasses learning about the problem to be solved, understanding the needs of customer and users, trying to find out who the user really is, and understanding all the constraints on the solution. It includes all activities related to the following:

- ✓ Identification and documentation of customer's or user's needs.
- ✓ Creation of a document that describes the external behavior and the association constraints that will satisfies those needs.
- ✓ Analysis and validation of the requirements documents to ensure consistency, completeness, and feasibility
- ✓ Evolution of needs.

After the analysis of the functioning of system, the proposed System is expected to do the following: -

-  To provide a user friendly, Graphical User Interface (GUI).
-  The proposed system should maintain all the records and transactions, and should generate the required reports and information when required.
-  To provide efficient and secured Information storage, flow and retrieval system, ensuring the integrity and validity of records.
-  To provide graphical and user-friendly interface to interact with a centralized database based on client-server architecture.
-  To identify the critical operation procedure and possibilities of simplification using modern IT tools and practices.

5. System Design & Development

5.1 Database Design:

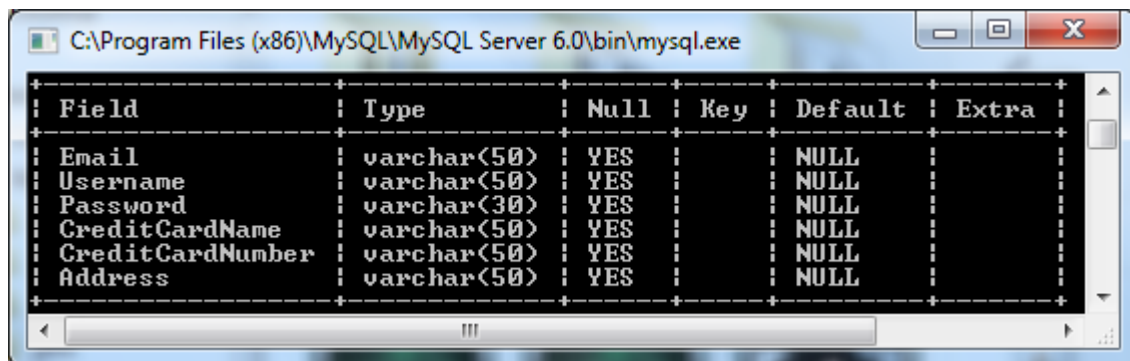
An important aspect of system design is the design of data storage structure. To begin with a logical model of data structure is developed first. A database is a container object which contains tables, queries, reports and data validation policies enforcement rules or constraints etc. A logical data often represented as a records are kept in different tables after reducing anomalies and redundancies. The goodness of data base design lies in the table structure and its relationship.

This software project contains n tables one is account, other is username i.e. more the account more are the tables.

Table Design:

The database of Game System contains 5 tables. Most of the tables are designed to store master records. The tables and their structure are given below.

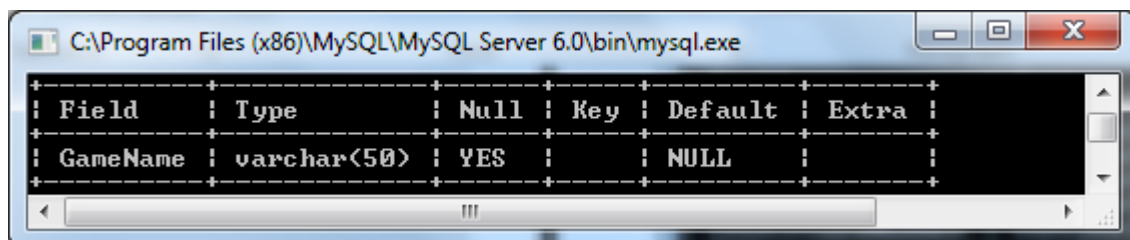
Table: Account



The screenshot shows a MySQL command window with the title 'C:\Program Files (x86)\MySQL\MySQL Server 6.0\bin\mysql.exe'. The output displays the structure of the 'Account' table with the following columns: Field, Type, Null, Key, Default, and Extra.

Field	Type	Null	Key	Default	Extra
Email	varchar(50)	YES		NULL	
Username	varchar(50)	YES		NULL	
Password	varchar(30)	YES		NULL	
CreditCardName	varchar(50)	YES		NULL	
CreditCardNumber	varchar(50)	YES		NULL	
Address	varchar(50)	YES		NULL	

Table:* \ Username * \

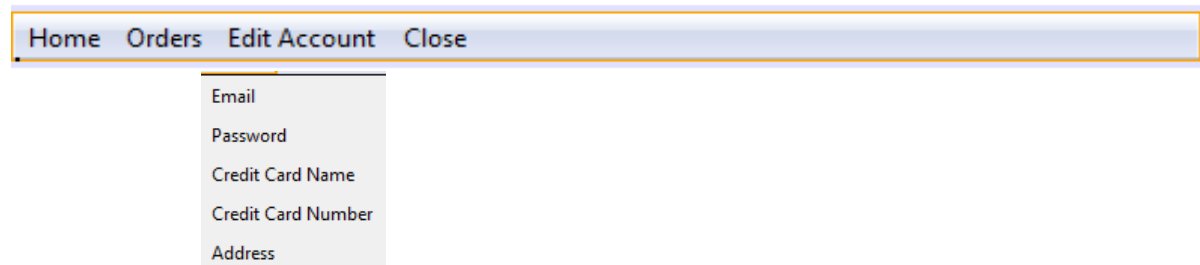


The screenshot shows a MySQL command window with the title 'C:\Program Files (x86)\MySQL\MySQL Server 6.0\bin\mysql.exe'. The output displays the structure of the 'Username' table with the following columns: Field, Type, Null, Key, Default, and Extra.

Field	Type	Null	Key	Default	Extra
GameName	varchar(50)	YES		NULL	

Menu Design:

JSS Info ware gateway comprises the following options, organized in a user friendly way. The menu system divided in Menu Bars, each having a pull down menus containing options for a specific task.



Menu Item	Form	Function
Home	Home.java	It Will Open Home Page
Order (History)	History.java	It Will Show History
Email	Email.java	From Here We Can Change Email
Password	Password.java	From Here We Can Change Password
Credit Card Name	CreditCardName.java	From Here We Can Change Credit Card Name
Credit Card Number	CreditCardNumber.java	From Here We Can Change Credit Card Number
Address	Address.java	From Here We Can Change Address
Close	_____	Close Application

5.2 I/O Forms Design & Event Coding:

Class: Game.java

```
public class Game {  
    public static String email=null;  
    public static String username=null;  
    public static String password=null;  
}
```

Frame: Main.java



```
setSize(569,320);
```

Login Button –

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    Login x=new Login();  
    x.setVisible(true);  
    this.dispose();  
}
```

Sign Up Button –

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
```

```

SignUpEmail x=new SignUpEmail();
x.setVisible(true);
this.dispose(); }

```

Give Up Button –

```

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
System.exit(0); }

```

Frame: SignUpEmail.java



```

setSize(569,320);

```

Next Button –

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
Game.email=jTextField1.getText();
SignUp x=new SignUp();
x.setVisible(true);
this.dispose();
}

```

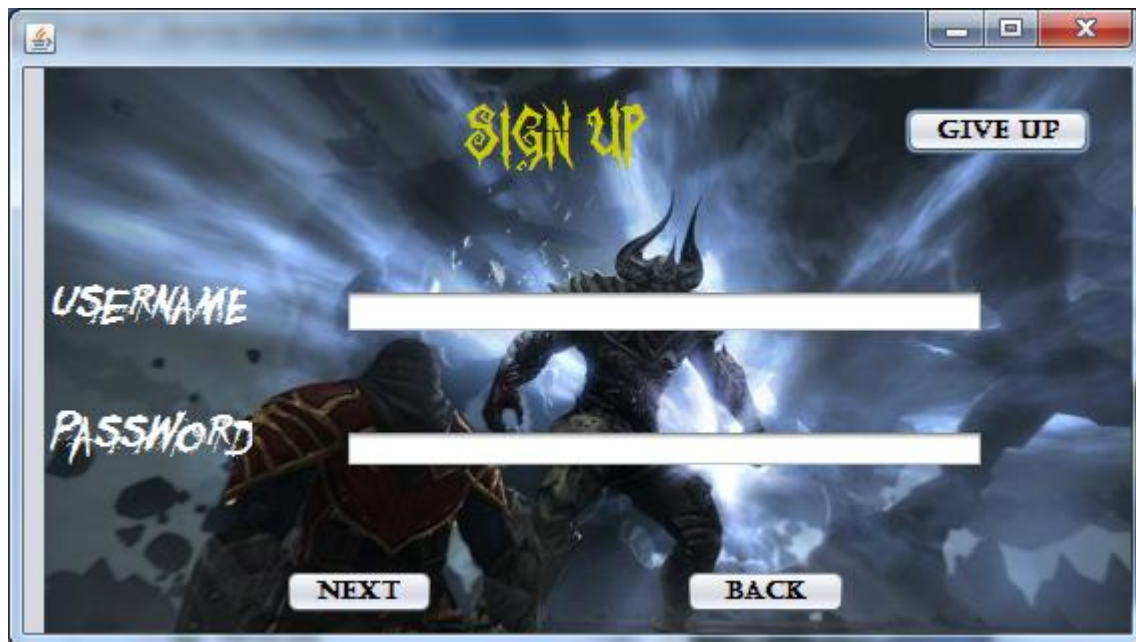
Give Up Button –

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0); }
```

Back Button –

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    Main x=new Main();  
    x.setVisible(true);  
    this.dispose(); }
```

Frame: SignUp.java



```
setSize(569,320);  
jLabel5.setText(""+Game.email);
```

Next Button –


```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    Game.username=jTextField1.getText();
    Game.password=String.valueOf(jPasswordField1.getPassword());
    SignUpCredit x=new SignUpCredit();
        x.setVisible(true);
        this.dispose();}
```

Give Up Button –

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);}
```

Back Button –

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    SignUpEmail x=new SignUpEmail();
    x.setVisible(true);
    this.dispose();}
```

Frame: SignUpCredit.java



```
import java.sql.*;
import javax.swing.*;
```

```
setSize(569,320);
```

Login Button –

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    String a,b,c;  
    a=jTextField1.getText();  
    b=jTextField2.getText();  
    c=jTextField3.getText();  
  
    JOptionPane.showMessageDialog(this,"Register Successfull");  
    Home z=new Home();  
    z.setVisible(true);  
    this.dispose();  
    try  
    {  
        Class.forName("com.mysql.jdbc.Driver");  
        Connection x;  
        String s1,s2,s3;  
        s1="jdbc:mysql://localhost:3306/Game";  
        s2="root";  
        s3="admin";  
        x = DriverManager.getConnection(s1,s2,s3);  
        String q="INSERT INTO Account  
VALUES('"+Game.email+"','"+Game.username+"','"+Game.password+"','"+a+"','"+b+"','"+c+"');";  
        String l="Create Table '"+Game.username+"' (GameName varchar(50));";  
        Statement y;  
        y=x.createStatement();  
        y.executeUpdate(q);  
        y.executeUpdate(l);  
  
        x.close();  
        y.close();  
    }  
    catch(Exception ex)  
    {}  
}
```

Give Up Button –

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0);}
```

Back Button –

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    SignUp x=new SignUp();
```

```
x.setVisible(true);  
this.dispose();
```

Frame: Login.java



```
import java.sql.*;  
import javax.swing.*;  
  
setSize(569,320);
```

Login Button –

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    String a,b;  
  
    Game.username=a=jTextField1.getText();  
    Game.password=b=String.valueOf(jPasswordField1.getPassword());  
    try  
    {  
        Class.forName("com.mysql.jdbc.Driver");  
        Connection x;  
        String s1,s2,s3;  
        s1="jdbc:mysql://localhost:3306/Game";
```



```

s2="root";
s3="admin";
x = DriverManager.getConnection(s1,s2,s3);
Statement y;
y=x.createStatement();
String q="SELECT username , password FROM Account WHERE username='"+a+"' AND
password='"+b+"'";
ResultSet r;
r=y.executeQuery(q);
if(r.next())
    { if((a.compareTo(r.getString(1))==0) && (b.compareTo(r.getString(2))==0))
      { Home z=new Home();
        z.setVisible(true);
        this.dispose();

      }

    }
else
{jOptionPane l.showMessageDialog(this,"Login Fail");}

    x.close();
    y.close();
}
catch(Exception ex){
    }}

```

Give Up Button –

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);}

```

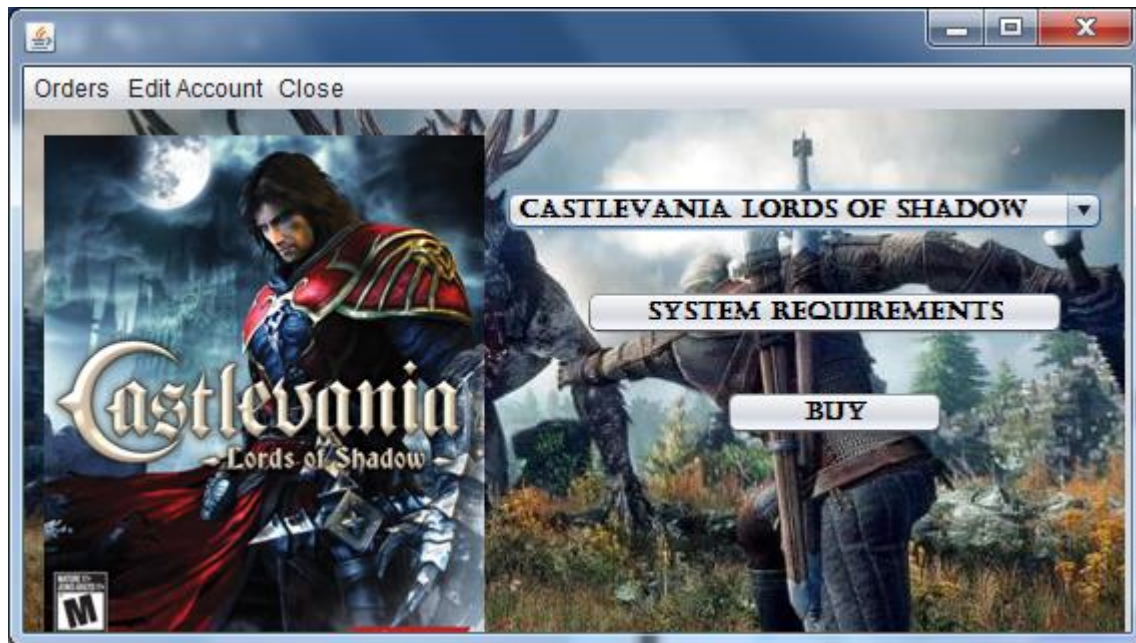
Back Button –

```

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    SignUp x=new SignUp();
    x.setVisible(true);
    this.dispose();
}

```

Frame: Home.java



```
import javax.swing.ImageIcon;
import java.sql.*;
import javax.swing.*;
setSize(569,320);
```

History (Orders) Menu –

```
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    History x=new History();
    x.setVisible(true);
    this.dispose();
}
```

Email (Edit Account) Menu –

```
private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
    Email x=new Email();
    x.setVisible(true);
    this.dispose(); }
```

Password (Edit Account) Menu –

```
private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
    Password x=new Password();
    x.setVisible(true);
    this.dispose(); }
```

Credit Card Name (Edit Account) Menu –

```
private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {
    CreditCardName x=new CreditCardName();
    x.setVisible(true);
    this.dispose();}
```

Credit Card Number (Edit Account) Menu –

```
private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {
    CreditCardNumber x=new CreditCardNumber();
    x.setVisible(true);
    this.dispose();}
```

Address(Edit Account) Menu –

```
private void jMenuItem7ActionPerformed(java.awt.event.ActionEvent evt) {
    Address x=new Address();
    x.setVisible(true);
    this.dispose();}
```

Close Menu –

```
private void jMenuItem4MenuSelected(javax.swing.event.MenuEvent evt) {
    System.exit(0);}
```

Game List –

```
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    int game=jComboBox1.getSelectedIndex();
    if(game==0)
    {
        jLabel2.setIcon(new
        ImageIcon("F:\\Project2\\src\\CastlevaniaLordsOfShadows.png"));
    }
    if(game==1)
    {
        jLabel2.setIcon(new ImageIcon("F:\\Project2\\src\\CastlevaniaMirrorOfFate.jpg"));
    }
    if(game==2)
    {
        jLabel2.setIcon(new
        ImageIcon("F:\\Project2\\src\\CastlevaniaLordsOfShadows2.jpg"));
    }
    if(game==3)
    {
        jLabel2.setIcon(new ImageIcon("F:\\Project2\\src\\ShadowOfColossus.jpg"));
    }
    if(game==4)
    {
```

```

jLabel2.setIcon(new
ImageIcon("F:\\Project2\\src\\SphinxAndTheCursedMummy.png"));
}
if(game==5)
{
jLabel2.setIcon(new ImageIcon("F:\\Project2\\src\\AssasienCreedOdyssey.png"));
}
if(game==6)
{
jLabel2.setIcon(new ImageIcon("F:\\Project2\\src\\AssasienCreedOrigins.png"));
}
if(game==7)
{
jLabel2.setIcon(new ImageIcon("F:\\Project2\\src\\ResidentEvil7.jpg"));
}
if(game==8)
{
jLabel2.setIcon(new ImageIcon("F:\\Project2\\src\\ClockTower3.jpg"));
}
if(game==9)
{
jLabel2.setIcon(new ImageIcon("F:\\Project2\\src\\FarCry5.jpg"));
} }

```

Buy Button –

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
int game=jComboBox1.getSelectedIndex();
String a="";
if(game==0)
{
a="Castlevania Lords Of Shadows";
}
if(game==1)
{
a="Castlevania Mirror Of Fate";
}
if(game==2)
{
a="Castlevania Lords Of Shadows 2";
}
if(game==3)
{
a="Shadow Of Colossus";
}
if(game==4)
{
a="Sphinx And The Cursed Mummy";
}
}

```

```

if(game==5)
{
    a="Assasien Creed Odyssey";
}
if(game==6)
{
    a="Assasien Creed Origins";
}
if(game==7)
{
    a="Resident Evil 7";
}
if(game==8)
{
    a="Clock Tower 3";
}
if(game==9)
{
    a="Far Cry 5";
}
jOptionPane1.showMessageDialog(this,"Added To Cart");
try
{
    Class.forName("com.mysql.jdbc.Driver");
    Connection x;
    String s1,s2,s3;
    s1="jdbc:mysql://localhost:3306/Game";
    s2="root";
    s3="admin";
    x = DriverManager.getConnection(s1,s2,s3);

    String q="INSERT INTO "+Game.username+" VALUES('"+a+"');";
    Statement y;
    y=x.createStatement();
    y.executeUpdate(q);
    x.close();
    y.close();

}
catch(Exception ex)
{} }

```

System Requirements Button –

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
int game=jComboBox1.getSelectedIndex();

if(game==0)
{

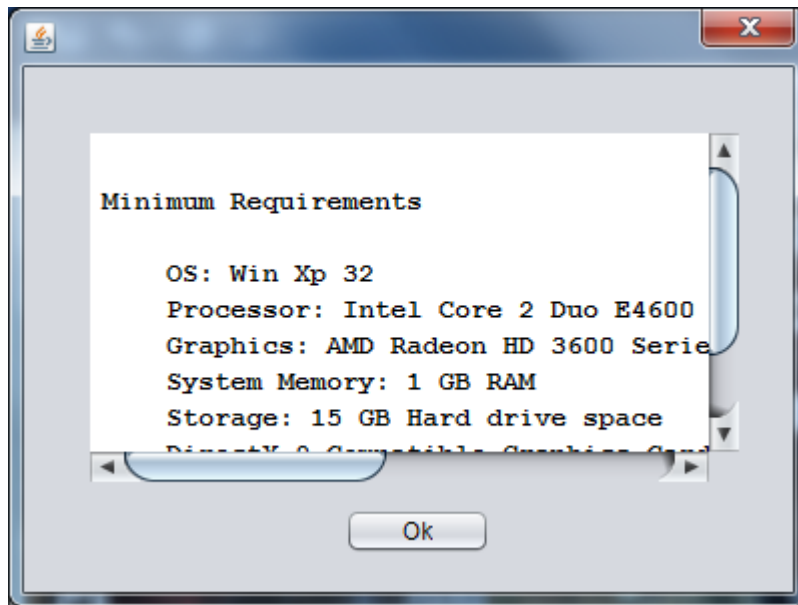
```

```

        jDialog1.setVisible (true);
    }
    if(game==1)
    {
        jDialog2.setVisible (true);
    }
    if(game==2)
    {
        jDialog3.setVisible (true);
    }
    if(game==3)
    {
        jDialog4.setVisible (true);
    }
    if(game==4)
    {
        jDialog5.setVisible (true);
    }
    if(game==5)
    {
        jDialog6.setVisible (true);
    }
    if(game==6)
    {
        jDialog7.setVisible (true);
    }
    if(game==7)
    {
        jDialog8.setVisible (true);
    }
    if(game==8)
    {
        jDialog9.setVisible (true);
    }
    if(game==9)
    {
        jDialog10.setVisible (true);
    }
}

```

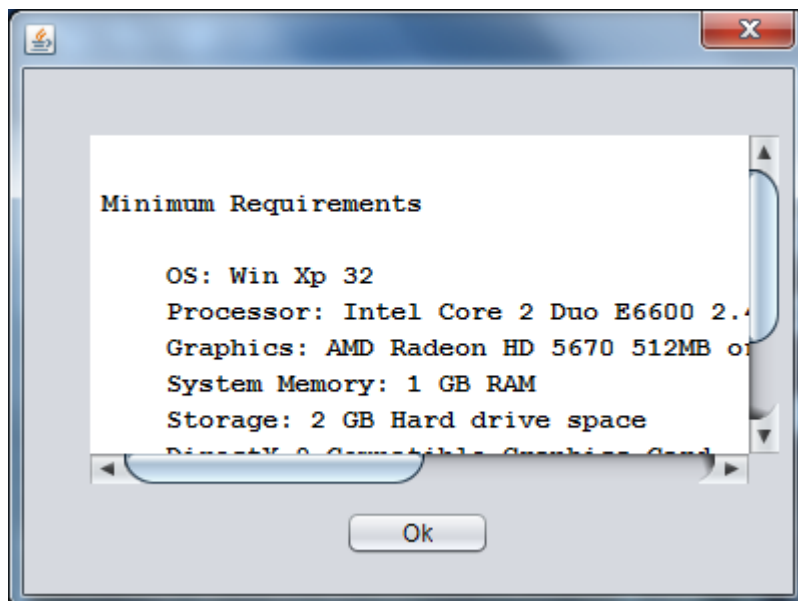
Dialog Box 1 –



Ok Button-

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    jDialog1.setVisible (false); }  
}
```

Dialog Box 2 –

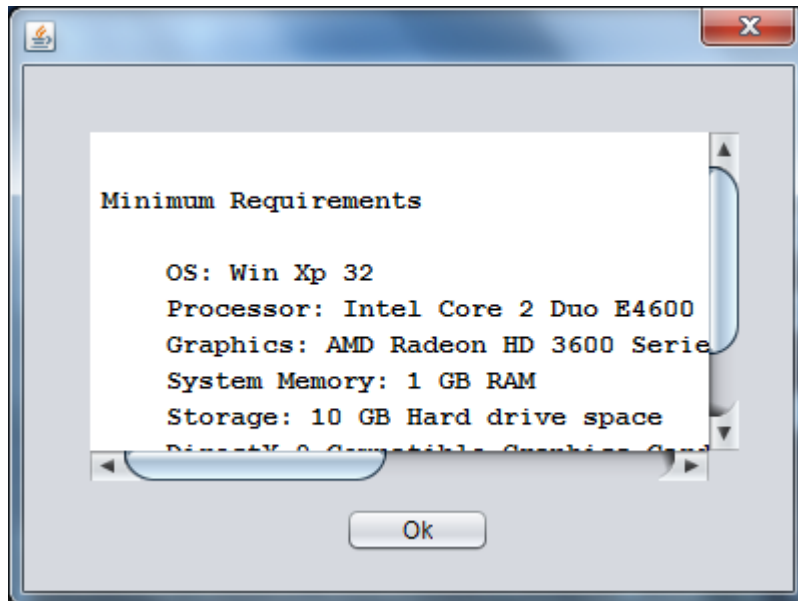


Ok Button-

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
    jDialog1.setVisible (false); }  
}
```

```
jDialog2.setVisible (false); }
```

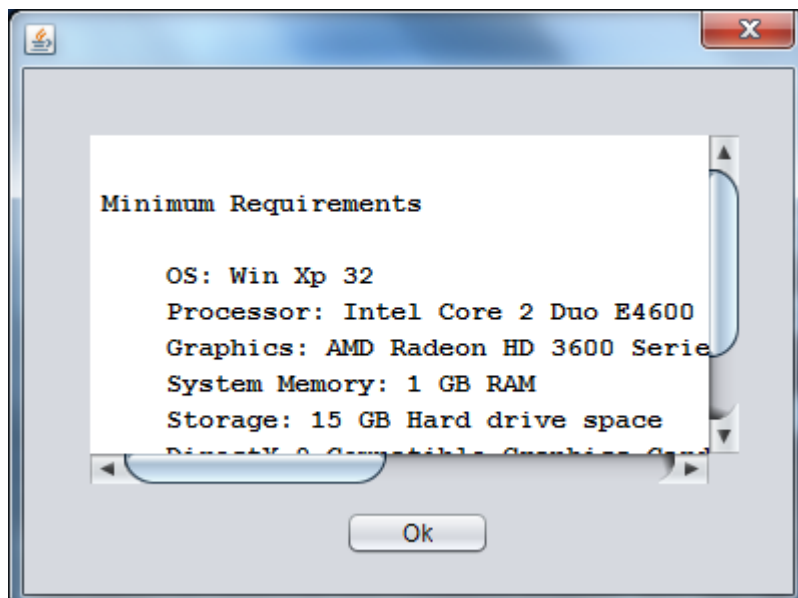
Dialog Box 3 –



Ok Button-

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {  
jDialog3.setVisible (false); }
```

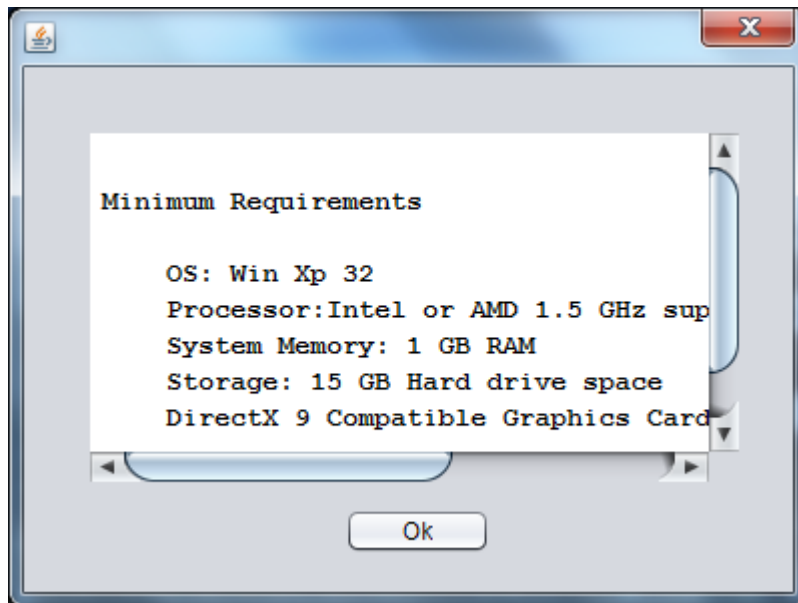
Dialog Box 4 –



Ok Button-

```
private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {  
    jDialog4.setVisible (false); }  
}
```

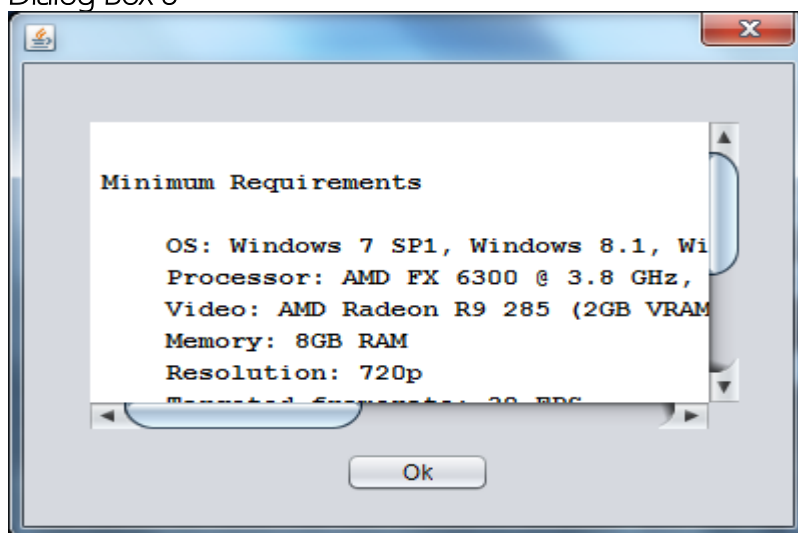
Dialog Box 5 –



Ok Button-

```
private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {  
    jDialog5.setVisible (false); }  
}
```

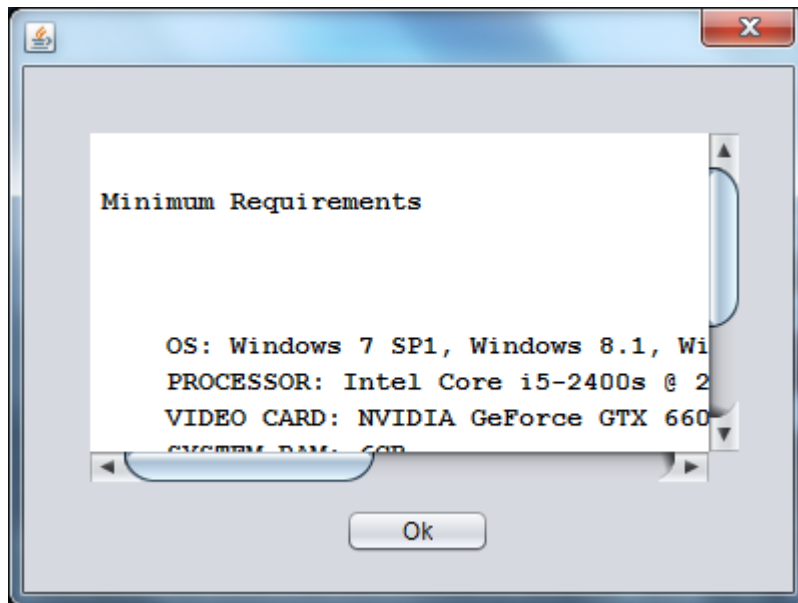
Dialog Box 6 –



Ok Button-

```
private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {  
    jDialog6.setVisible (false); }  
}
```

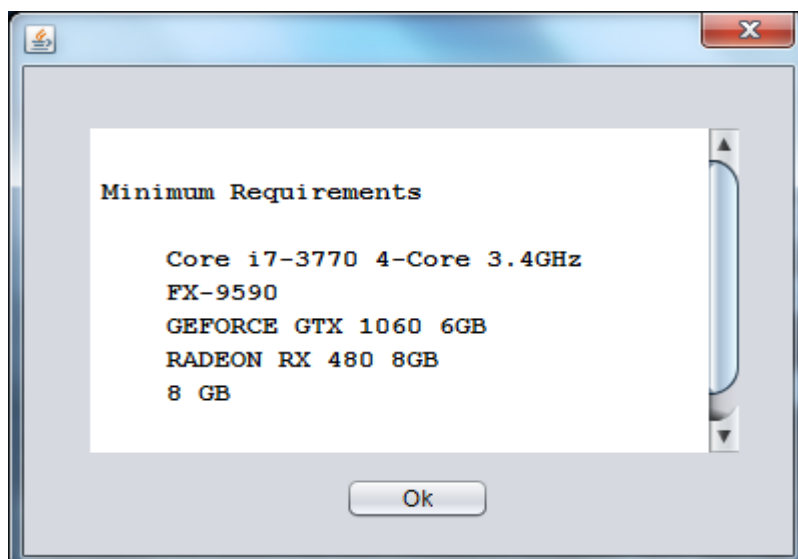
Dialog Box 7 –



Ok Button-

```
private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {  
    jDialog7.setVisible (false); }  
}
```

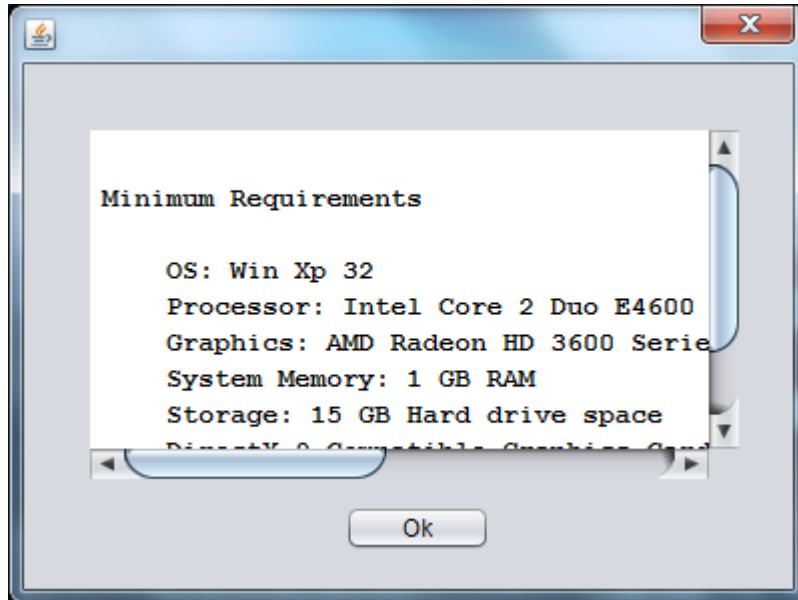
Dialog Box 8 –



Ok Button-

```
private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {  
    jDialog8.setVisible (false); }  
}
```

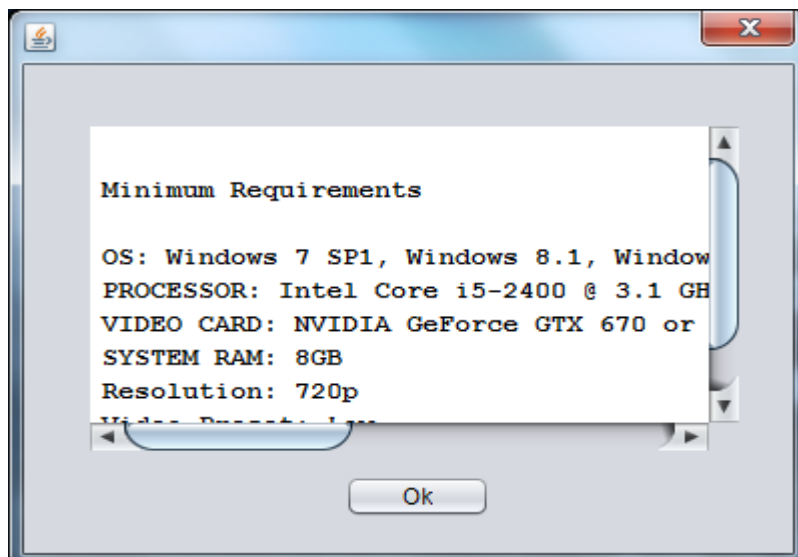
Dialog Box 9 –



Ok Button-

```
private void jButton11ActionPerformed(java.awt.event.ActionEvent evt) {  
    jDialog9.setVisible (false); }  
}
```

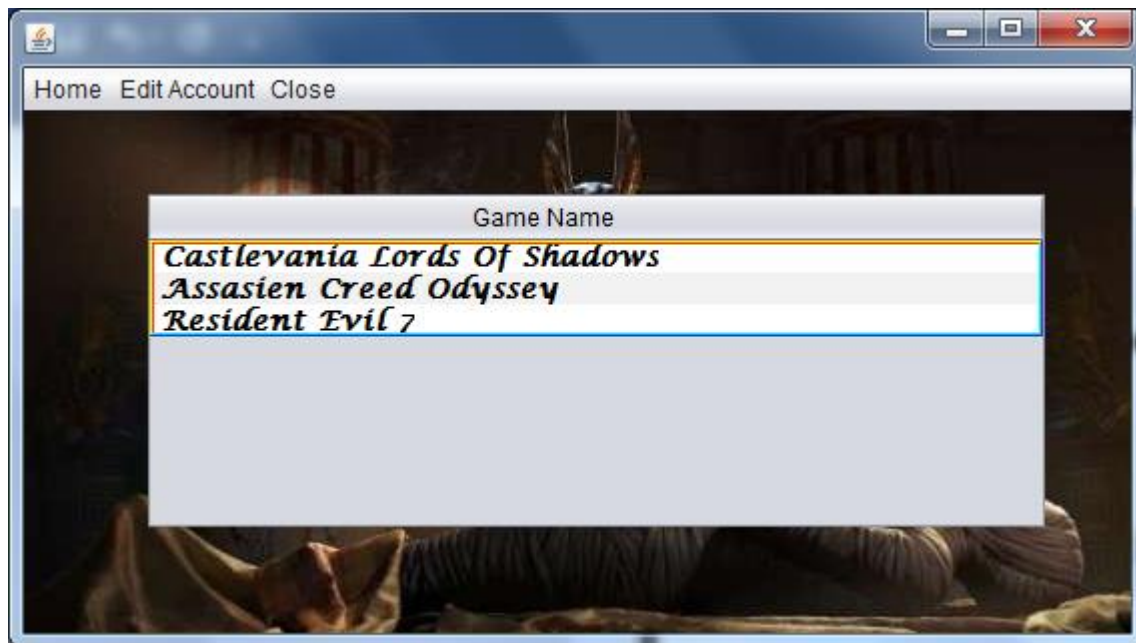
Dialog Box 10 –



Ok Button-

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    jDialog10.setVisible (false); }  
}
```

Frame: History.java



```
import java.sql.*;  
import javax.swing.table.DefaultTableModel;  
  
setSize(569,320);  
DefaultTableModel m =(DefaultTableModel) jTable1.getModel();  
try{  
    Class.forName("com.mysql.jdbc.Driver");  
    String s1,s2,s3;  
    s1="jdbc:mysql://localhost:3306/Game";  
    s2="root";  
    s3="admin";  
    Connection x=DriverManager.getConnection(s1,s2,s3);  
    Statement y=x.createStatement();  
    String q="Select * from "+Game.username+";";  
    ResultSet r=y.executeQuery(q);  
    while(r.next()){  
        String a = r.getString(1);  
        m.addRow(new Object[] {a});  
    }  
    catch (Exception ex) {}  
}
```

Home Menu –

```
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {  
    Home x=new Home();  
    x.setVisible(true);  
    this.dispose();  
}
```

Email (Edit Account) Menu –

```
private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {  
    Email x=new Email();  
    x.setVisible(true);  
    this.dispose(); }  
}
```

Password (Edit Account) Menu –

```
private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {  
    Password x=new Password();  
    x.setVisible(true);  
    this.dispose(); }  
}
```

Credit Card Name (Edit Account) Menu –

```
private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {  
    CreditCardName x=new CreditCardName();  
    x.setVisible(true);  
    this.dispose();}
```

Credit Card Number (Edit Account) Menu –

```
private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {  
    CreditCardNumber x=new CreditCardNumber();  
    x.setVisible(true);  
    this.dispose();}
```

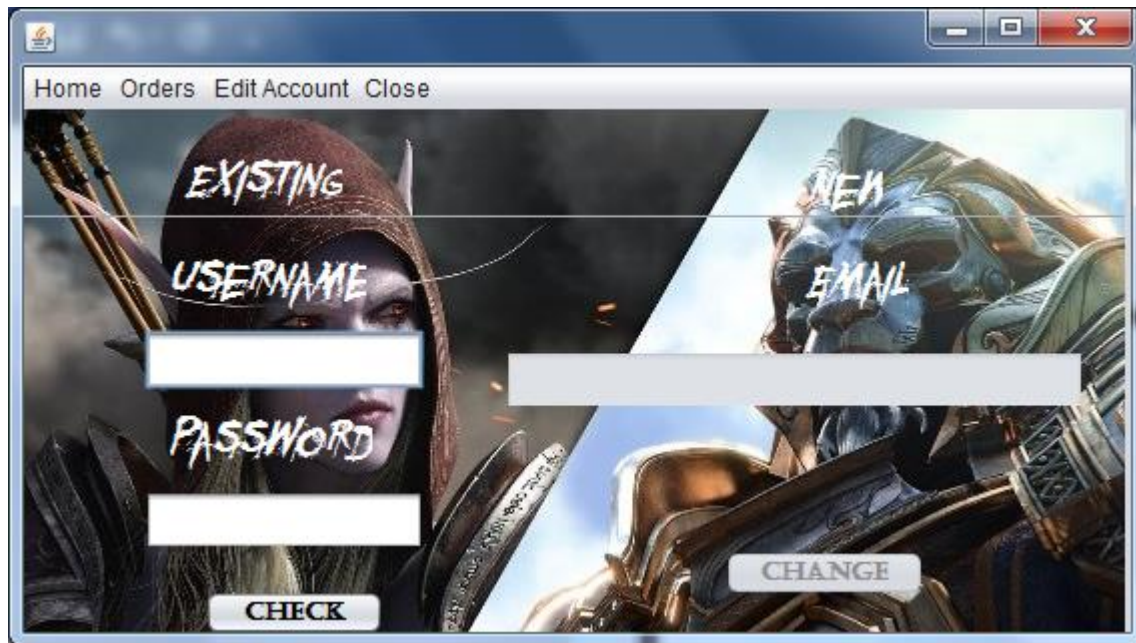
Address(Edit Account) Menu –

```
private void jMenuItem7ActionPerformed(java.awt.event.ActionEvent evt) {  
    Address x=new Address();  
    x.setVisible(true);  
    this.dispose();}
```

Close Menu –

```
private void jMenuItem4MenuSelected(javax.swing.event.MenuEvent evt) {  
    System.exit(0);  
}
```

Frame: Email.java



```
import javax.swing.*.*;
import java.sql.*;
```

```
setSize(569,320);
```

Home Menu –

```
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    Home x=new Home();
    x.setVisible(true);
    this.dispose();
}
```

History (Orders) Menu –

```
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    History x=new History();
    x.setVisible(true);
    this.dispose(); }
```

Password (Edit Account) Menu –

```
private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
    Password x=new Password();
    x.setVisible(true);
    this.dispose(); }
```

Credit Card Name (Edit Account) Menu –

```
private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {
    CreditCardName x=new CreditCardName();
```

```
x.setVisible(true);  
this.dispose();}
```

Credit Card Number (Edit Account) Menu –

```
private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {  
    CreditCardNumber x=new CreditCardNumber();  
    x.setVisible(true);  
    this.dispose();}
```

Address(Edit Account) Menu –

```
private void jMenuItem7ActionPerformed(java.awt.event.ActionEvent evt) {  
    Address x=new Address();  
    x.setVisible(true);  
    this.dispose();}
```

Close Menu –

```
private void jMenuItem4MenuSelected(javax.swing.event.MenuEvent evt) {  
    System.exit(0);}
```

Check Button –

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    String a,b;  
    a=jTextField2.getText();  
    b=String.valueOf(jPasswordField1.getPassword());  
  
    if((a.equals(Game.username) &&(b.equals(Game.password)))  
        {jTextField1.setEnabled(true);  
        jButton2.setEnabled(true);  
    }  
  
    else  
        {jOptionPane1.showMessageDialog(this,"Wrong Username Or Password");}  
}
```

Change Button –

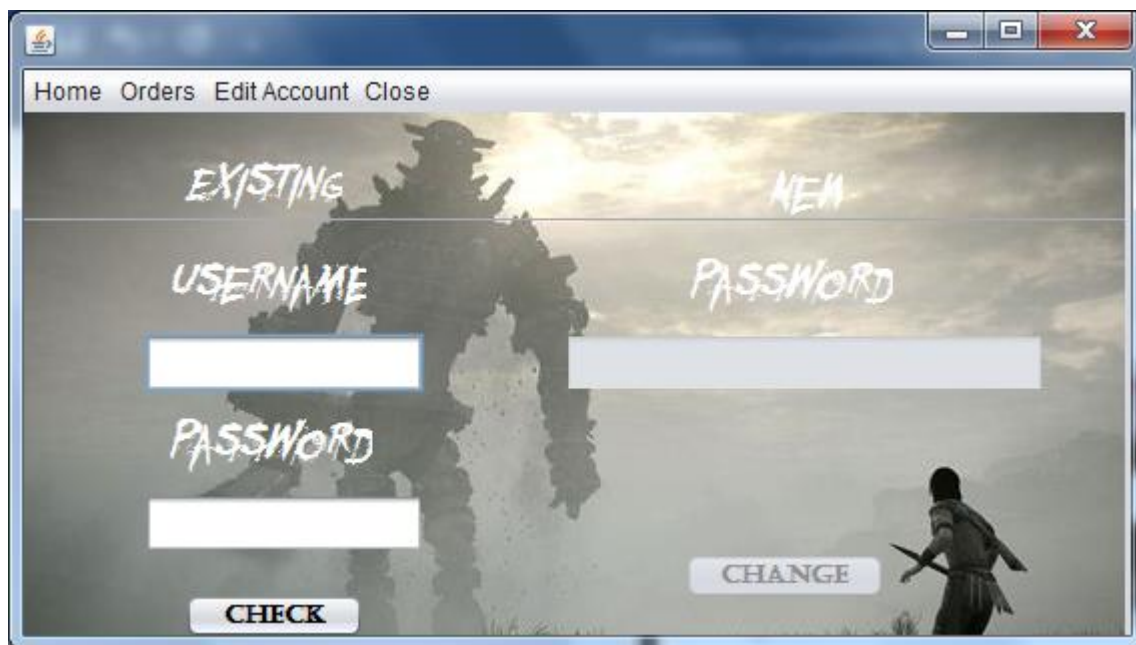
```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    String c;  
  
    c=jTextField1.getText();  
    try  
    {  
        Class.forName("com.mysql.jdbc.Driver");
```

```

Connection x;
String s1,s2,s3;
s1="jdbc:mysql://localhost:3306/Game";
s2="root";
s3="admin";
x = DriverManager.getConnection(s1,s2,s3);
Statement y;
y=x.createStatement();
String q="Update Account SET Email = '"+c+"' where Username = 
"+Game.username+"'";
OptionPane1.showMessageDialog(this,"Change Successful");
y.executeUpdate(q);
x.close();
y.close();
}
catch(Exception ex){
}}

```

Frame: Password.java



```

import javax.swing.*;
import java.sql.*;

```

```

setSize(569,320);

```

Home Menu –

```

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
Home x=new Home();

```



```
x.setVisible(true);
this.dispose();
}
```

History (Orders) Menu –

```
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    History x=new History();
    x.setVisible(true);
    this.dispose(); } }
```

Email (Edit Account) Menu –

```
private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
    Email x=new Email();
    x.setVisible(true);
    this.dispose(); } }
```

Credit Card Name (Edit Account) Menu –

```
private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {
    CreditCardName x=new CreditCardName();
    x.setVisible(true);
    this.dispose();}
```

Credit Card Number (Edit Account) Menu –

```
private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {
    CreditCardNumber x=new CreditCardNumber();
    x.setVisible(true);
    this.dispose();}
```

Address(Edit Account) Menu –

```
private void jMenuItem7ActionPerformed(java.awt.event.ActionEvent evt) {
    Address x=new Address();
    x.setVisible(true);
    this.dispose();}
```

Close Menu –

```
private void jMenuItem4MenuSelected(javax.swing.event.MenuEvent evt) {
    System.exit(0);}
```

Check Button –

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    String a,b;
```

```

a=jTextField2.getText();
b=String.valueOf(jPasswordField1.getPassword());

if((a.equals(Game.username) &&(b.equals(Game.password))
{jTextField1.setEnabled(true);
  jButton2.setEnabled(true);
}

else
{jOptionPane1.showMessageDialog(this,"Wrong Username Or Password");}

}

```

Change Button –

```

String a,b;
a=jTextField1.getText();
b=String.valueOf(jPasswordField1.getPassword());

if((a.equals(Game.username) &&(b.equals(Game.password))
{jPasswordField2.setEnabled(true);
  jButton2.setEnabled(true);
}

else
{jOptionPane1.showMessageDialog(this,"Wrong Username Or Password");}

// TODO add your handling code here:
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    String c;

    Game.password= c=String.valueOf(jPasswordField2.getPassword());
    try
    {
        Class.forName("com.mysql.jdbc.Driver");
        Connection x;
        String s1,s2,s3;
        s1="jdbc:mysql://localhost:3306/Game";
        s2="root";
        s3="admin";
        x = DriverManager.getConnection(s1,s2,s3);
        Statement y;
        y=x.createStatement();
        String q="Update Account SET Password = '"+c+"' where Username =
"+Game.username+"";
        JOptionPane1.showMessageDialog(this,"Change Successful");
        y.executeUpdate(q);
        x.close();
        y.close();
    }
}

```

```

}
catch(Exception ex){
}}

```

Frame: CreditCardName.java



```

import javax.swing.*;
import java.sql.*;

```

```

setSize(569,320);

```

Home Menu –

```

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    Home x=new Home();
    x.setVisible(true);
    this.dispose();
}

```

History (Orders) Menu –

```

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    History x=new History();
    x.setVisible(true);
    this.dispose(); }

```

Password (Edit Account) Menu –

```

private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
    Password x=new Password();
}

```

```
x.setVisible(true);  
this.dispose(); }
```

Email (Edit Account) Menu –

```
private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {  
    Email x=new Email();  
    x.setVisible(true);  
    this.dispose();}
```

Credit Card Number (Edit Account) Menu –

```
private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {  
    CreditCardNumber x=new CreditCardNumber();  
    x.setVisible(true);  
    this.dispose();}
```

Address(Edit Account) Menu –

```
private void jMenuItem7ActionPerformed(java.awt.event.ActionEvent evt) {  
    Address x=new Address();  
    x.setVisible(true);  
    this.dispose();}
```

Close Menu –

```
private void jMenuItem4MenuSelected(javax.swing.event.MenuEvent evt) {  
    System.exit(0);}
```

Check Button –

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    String a,b;  
    a=jTextField2.getText();  
    b=String.valueOf(jPasswordField1.getPassword());  
  
    if((a.equals(Game.username) &&(b.equals(Game.password))  
    {jTextField1.setEnabled(true);  
      jButton2.setEnabled(true);  
    }  
  
    else  
    {jOptionPane1.showMessageDialog(this,"Wrong Username Or Password");}  
  
}
```

Change Button –

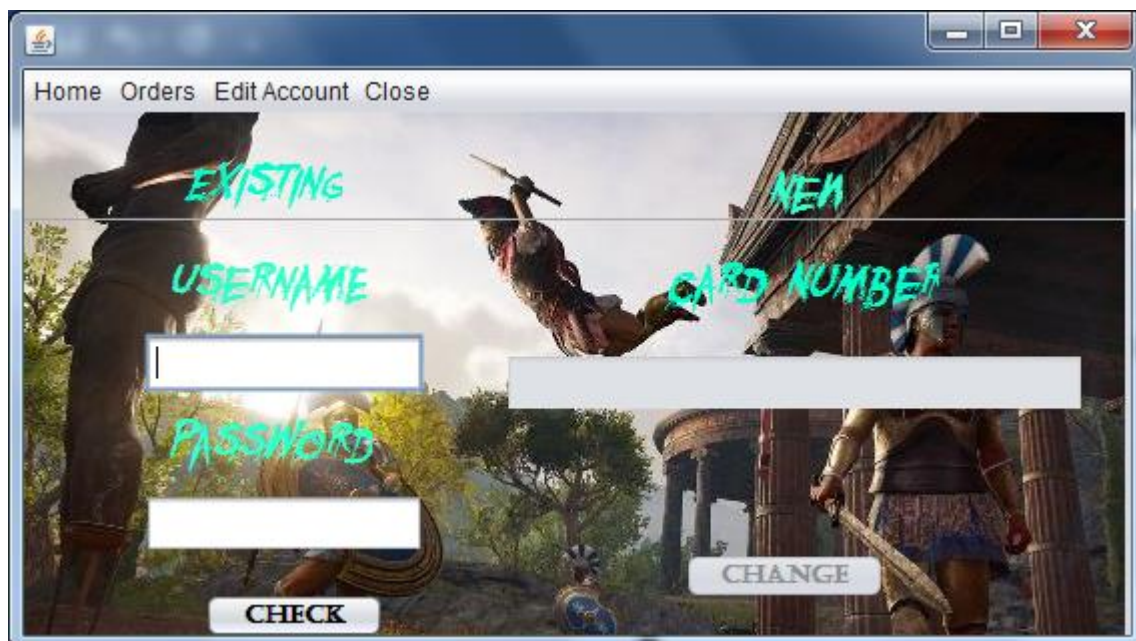
```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    String c;

    c=jTextField1.getText();
    try
    {
        Class.forName("com.mysql.jdbc.Driver");
        Connection x;
        String s1,s2,s3;
        s1="jdbc:mysql://localhost:3306/Game";
        s2="root";
        s3="admin";
        x = DriverManager.getConnection(s1,s2,s3);
        Statement y;
        y=x.createStatement();
        String q="Update Account SET CreditCardName = '"+c+"' where Username =
"+Game.username+"";
        JOptionPane.showMessageDialog(this,"Change Successful");
        y.executeUpdate(q);
        x.close();
        y.close();
    }
    catch(Exception ex){
    }}

```

Frame: CreditCardNumber.java



```

import javax.swing.*;
import java.sql.*;

```

```
setSize(569,320);
```

Home Menu –

```
private void jMenuItem10ActionPerformed(java.awt.event.ActionEvent evt) {  
    Home x=new Home();  
    x.setVisible(true);  
    this.dispose();  
}
```

History (Orders) Menu –

```
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {  
    History x=new History();  
    x.setVisible(true);  
    this.dispose(); }  
}
```

Password (Edit Account) Menu –

```
private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {  
    Password x=new Password();  
    x.setVisible(true);  
    this.dispose(); }  
}
```

Credit Card Name (Edit Account) Menu –

```
private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {  
    CreditCardName x=new CreditCardName();  
    x.setVisible(true);  
    this.dispose();}  
}
```

Email (Edit Account) Menu –

```
private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {  
    Email x=new Email();  
    x.setVisible(true);  
    this.dispose();}  
}
```

Address(Edit Account) Menu –

```
private void jMenuItem7ActionPerformed(java.awt.event.ActionEvent evt) {  
    Address x=new Address();  
    x.setVisible(true);  
    this.dispose();}  
}
```

Close Menu –

```
private void jMenuItem4MenuSelected(javax.swing.event.MenuEvent evt) {  
    System.exit(0);  
}
```

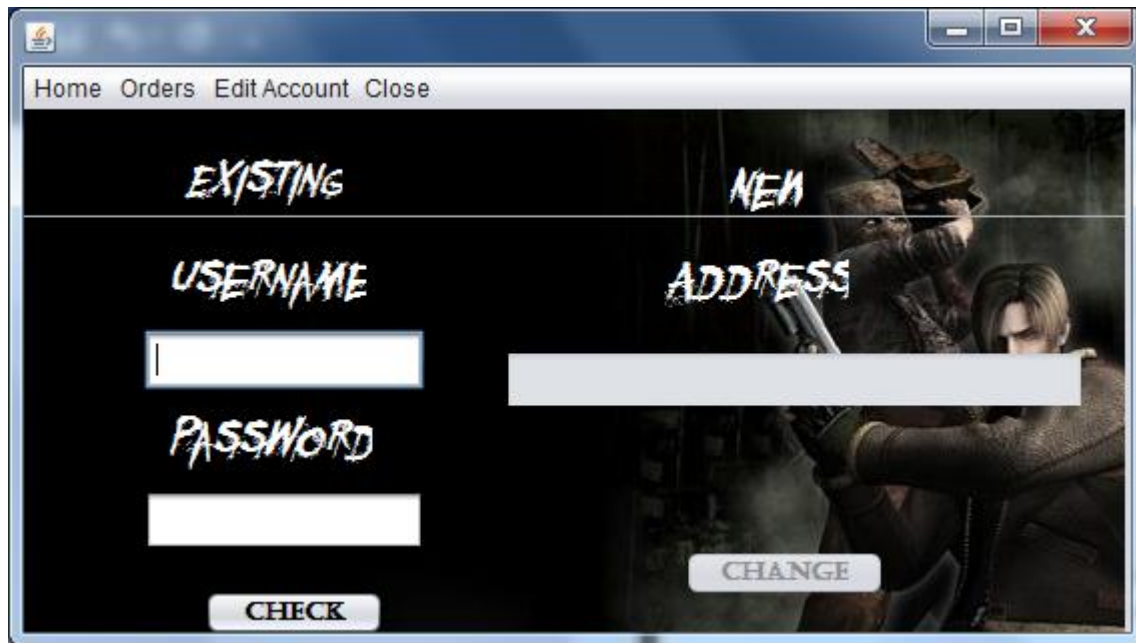
Check Button –

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    String a,b;  
    a=jTextField2.getText();  
    b=String.valueOf(jPasswordField1.getPassword());  
  
    if((a.equals(Game.username) &&(b.equals(Game.password))  
    {jTextField1.setEnabled(true);  
      jButton2.setEnabled(true);  
    }  
  
    else  
    {jOptionPane1.showMessageDialog(this,"Wrong Username Or Password");}  
  
}
```

Change Button –

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    String c;  
  
    c=jTextField1.getText();  
    try  
    {  
        Class.forName("com.mysql.jdbc.Driver");  
        Connection x;  
        String s1,s2,s3;  
        s1="jdbc:mysql://localhost:3306/Game";  
        s2="root";  
        s3="admin";  
        x = DriverManager.getConnection(s1,s2,s3);  
        Statement y;  
        y=x.createStatement();  
        String q="Update Account SET CreditCardNumber = '"+c+"' where Username =  
        '"+Game.username+"';";  
        JOptionPane1.showMessageDialog(this,"Change Successful");  
        y.executeUpdate(q);  
        x.close();  
        y.close();  
    }  
    catch(Exception ex){  
    }  
}
```

Frame: Email.java



```
import javax.swing.*.*;
import java.sql.*;
```

```
setSize(569,320);
```

Home Menu –

```
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    Home x=new Home();
    x.setVisible(true);
    this.dispose();
}
```

History (Orders) Menu –

```
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    History x=new History();
    x.setVisible(true);
    this.dispose(); }
```

Password (Edit Account) Menu –

```
private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
    Password x=new Password();
    x.setVisible(true);
    this.dispose(); }
```

Credit Card Name (Edit Account) Menu –

```
private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {
    CreditCardName x=new CreditCardName();
```



```
x.setVisible(true);
this.dispose();}
```

Credit Card Number (Edit Account) Menu –

```
private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {
CreditCardNumber x=new CreditCardNumber();
x.setVisible(true);
this.dispose();}
```

Email (Edit Account) Menu –

```
private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
Email x=new Email();
x.setVisible(true);
this.dispose();}
```

Close Menu –

```
private void jMenuItem4MenuSelected(javax.swing.event.MenuEvent evt) {
System.exit(0);}
```

Check Button –

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
String a,b;
a=jTextField2.getText();
b=String.valueOf(jPasswordField1.getPassword());

if((a.equals(Game.username) &&(b.equals(Game.password)))
{jTextField1.setEnabled(true);
jButton2.setEnabled(true);
}

else
{jOptionPane1.showMessageDialog(this,"Wrong Username Or Password");}

}
```

Change Button –

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
String c;

c=jTextField1.getText();
try
{
Class.forName("com.mysql.jdbc.Driver");
```

```

Connection x;
String s1,s2,s3;
s1="jdbc:mysql://localhost:3306/Game";
s2="root";
s3="admin";
x = DriverManager.getConnection(s1,s2,s3);
Statement y;
y=x.createStatement();
String q="Update Account SET Address = '"+c+"' where Username =
"+Game.username+"";
jOptionPane1.showMessageDialog(this,"Change Successful");
y.executeUpdate(q);
x.close();
y.close();
}
catch(Exception ex){
}}

```

6. REFERENCES

In order to work on this project, the following books and literature are referred by me during the various phases of development of the project.

- (1) Informatics Practices for class XII
-by Sumita Arora
- (2) <http://www.mysql.org/>
- (3) <http://www.netbeans.org/>
- (4) On-line Help of NetBeans ®
- (5) Various Websites of Discussion Forum and software development activities.

Other than the above-mentioned books, the suggestions and supervision of my teacher and my class experience also helped me to develop this software project.