INTRODUCTION

1.1 ABOUT THE PROJECT

The project titled “RETAIL SHOP” is shop management software for managing purchase, sales, and stocks of a shop. This software mainly focuses on basic operations in a Shop like adding new vendors, new items, and updating new information, searching items and vendors and facility to purchase and return items.

It features a familiar and well thought –out, an attractive user interface, combined with strong searching, insertion and reporting capabilities.

1.2 OBJECTIVE & SCOPE OF THE PROJECT

The objective and goals of the proposed system are:

- To maintain Transaction of shop in digital form (computer system).
- The project is totally built at admin end and thus, only admin is guaranteed the access, Easy to use and work on it.
- The Admin itself is an owner of the shop this software provides a better facilities.
- To save the time and resources.
- The new system will take less time in entering the data, processing it, storing it and getting its output.
- Security is essential part to secure our data here security is provided at each level.

1.3 DEFINITION OF PROBLEM:

- The main aim of the project entitled “RETAIL SHOP” is to provide an application for keeping and managing track of details of purchasing, selling, returns, vendors, stock of items for a shop in its database. Everything is managed from the database.

This project is developed with following layers, which are:

1. Admin (Shop Details)
2. Purchase
3. Sales
4. Stock Availability
5. Vendors
6. Viewing Details of items, vendors and customers.
7. Store Details of items, vendors and customers.

SYSTEM ANALYSIS

2 SYSTEM ANALYSIS

System analysis is connected with studying the existing system, identifying the problem and the creation of the requirement specification. To launch the system analysis, we need a master plan detailing the steps to be taken. Major steps are defining requirement, studying the present system to verify the problem, defining the performance by the proposed system.

The objective of our study is to design and implement a Retail Shop System where purchase, sales, stock of item, vendor details are added, edited, searched, item availability in stock is checked.

2.1 EXISTING SYSTEM

In our existing system, all the record keeping processes are done manually, so that it was taking more time and calculation. Another thing is maintaining details of customers, availability of items in stock and after purchasing or selling, items need be counted so this was done manually. So after conducting the feasibility study I decided to make computerized software that does these stuffs automatically.

DISADVANTAGES OF EXISTING SYSTEM

In our existing system possibility of occurrence of error while entering the data is more. Formal knowledge about the system is need for the user to use the system. This system is unreliable, insecure, slow and more prone to errors. User interface is not interactive.

2.2 PROPOSED SYSTEM

Proposed system is an automated retail shop system. Through our software user can view, search available item details, can add, view, search, and delete details of items and vendors. Staff

Our proposed system has the following advantages:-

- User friendly
- Less error
- Search facility
- Look and feel environment
- Fast access to Database
- Report Generation

ADMINISTRATOR

Admin has access to the software only when the correct user id and password is known to him. Software User can change the contents of a items in stock and vendors from database. It is in Software User hand to manage and manipulate data of shop.

2.3 FEASIBILITY STUDY

The feasibility study is major factor, which contributes to the analysis and development of the system. The decision of the system analyst whether to design a particular system or not depends on its feasibility study.

1. Operational Feasibility:-

- The introduction to this system is not going to hamper any user of the system.
- The proposed system is very flexible and user friendly
- The proposed system produces best results and gives high performance. It can be implemented easily .So this project is operationally feasible

2. Technical Feasibility:-

- This feasibility deals with technicality of the system. Instead of storing details related to banking in manual way, this system is fully automated.
- No efficient manpower is required to handle the system.

3. Economic Feasibility:

Economic Feasibility deals about the economic impact faced by the organization to implement a new system.

- Economic Feasibility in this project.
- The cost to conduct a full system investigation is possible.
- There is no additional manpower requirement.
- There is no additional cost involved in maintaining the proposed system.

SYSTEM REQUIREMENT

AND

SPECIFICATIONS

3.1 HARDWARE CONFIGURATIONS

The selection of hardware is very important in the existence and proper working of any of the software. When selecting hardware, the size and capacity requirements are also important. This software is able to run with following hardware configuration.

Processor	: Intel Duo core
Processor Speed	: 1GHz to 2GHz
RAM	: 512MB to 1GB
Hard Disk	: 1GB and above
Keyboard	: 108 keys
Clock speed	: 500 MHz
System bus	: 32 bits

3.2 SOFTWARE CONFIGURATIONS

One of the most difficult tasks is selecting software, once the system requirement is find out then we have to determine whether a particular software package fits for those system requirements. This section summarizes the application requirement.

Operating System	**:** Windows XP/7/8/8.1/10, Linux, etc.
Language	**:** JAVA, SQL
IDE	**:** NetBeans
Database	**:**MySQL

3.3 TECHNOLOGY USED

- Java Programming Language
- NetBeans IDE to write and execute the code
- MySQL Database for storing data

3.4 PLATFORM USED

- **JDK SE 8.0**
- **JVM for Byte Code Execution**
- **WAMP Server for database server (localhost)**
- **Windows and Linux.**

SYSTEM DESIGN

4. SYSTEM DESIGN

System design is the process of developing specification for a candidate system that meet the criteria established in the system analysis. Major step in system design is the preparation of the input forms and output report in a form application to the user.

The main objectives of the system design are to the use of the package easily by any computer operator. System design is the creative act of invention, developing new inputs, a database, offline files, method, procedures and output for processing business to meet an organization objective. System design builds information gathered during the system analysis.

4.1 INPUT DESIGN

Input design is the process of converting user-oriented input to a computer based format. Input design is a part of overall system design, which requires very careful attention .Often the collection of input data is the most expensive part of the system. The main objectives of the input design are

1. Produce cost effective method of input

2. Achieve highest possible level of accuracy

3. Ensure that the input is acceptable to and understood by the staff.

4.2 OUTPUT DESIGN

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of these results for latter consultation. Computer output is the most important and direct source of information to the users. Designing computer output should proceed in an organized well throughout the manner. The right output must be available for the people who find the system easy use. The outputs have been defined during the logical design stage. If not, they should define at the beginning of the output designing terms of output connect, format, response etc.

4.3 DATABASE DESIGN

The general theme behind a database is to handle information as an integrated whole. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and effectively. After designing input and output, the analyst must concentrate on database design or how data should be organized around user requirements. The general objective is to make information access, easy quick, inexpensive and flexible for other users.

TABLE DESIGN:

- **Purchase Table**

➢ PRIMARY KEY: pid (purchase id)

Field	Type	Null	Default
pid	int(6)	No	
vendor	varchar(50)	Yes	NULL
item	varchar(50)	No	
quantity	int(11)	No	
price	float	No	
date	varchar(12)	Yes	NULL

- **Sale Table**

➢ Primary Key : sid (Sales Id)

➢ Foreign Key : item (Item name)

Field	Type	Null	Default
sid	int(11)	No	
item	varchar(50)	No	
quantity	int(11)	No	
price	float	No	
customer	varchar(40)	No	
address	varchar(100)	Yes	NULL
mobile	varchar(10)	Yes	NULL
date	varchar(12)	Yes	NULL

- **Stock Table**

➢ PRIMARY KEY: itemname (name of the item)

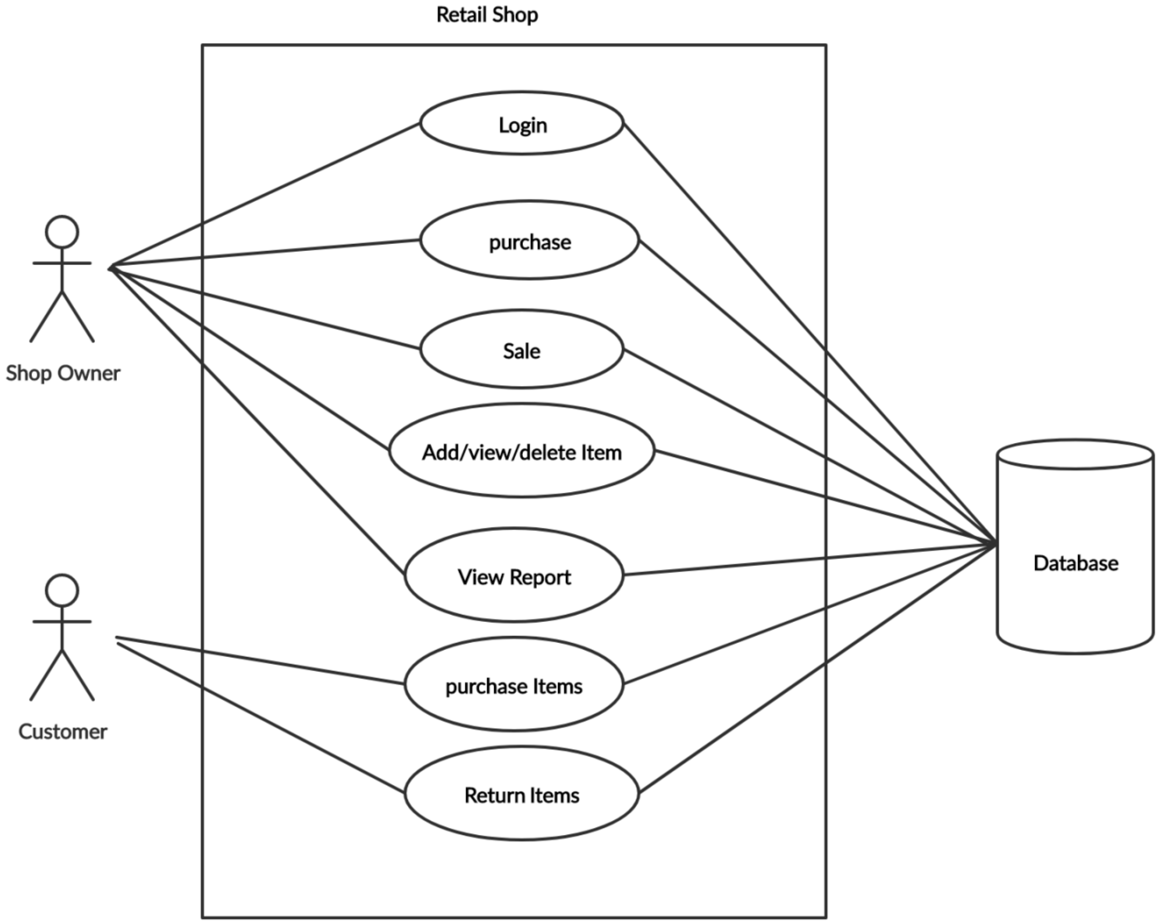
Field	Type	Null	Default
itemname	varchar(50)	No	
quantity	int(11)	No	
price	float	No	

- **Vendor Table**

➢ PRIMARY KEY: mobile (mobile number of item)

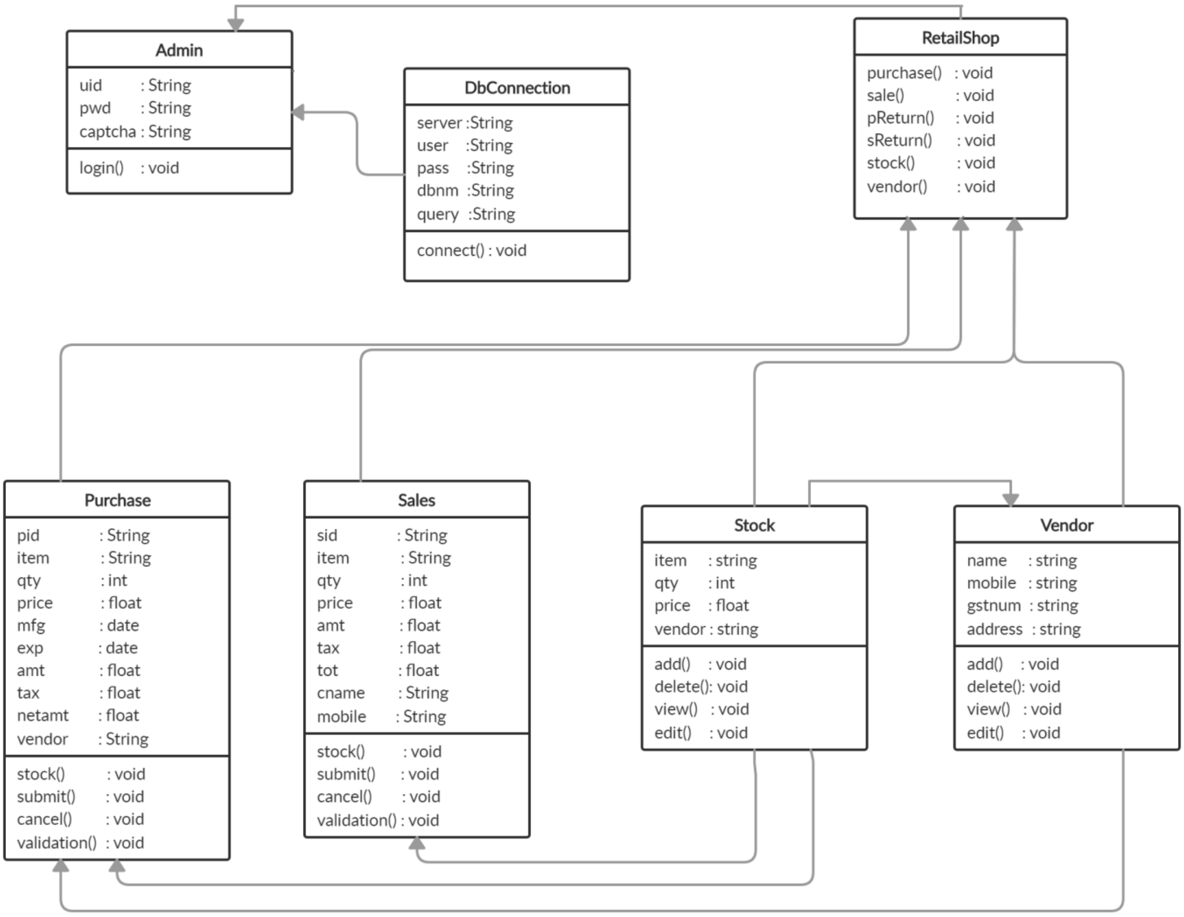
Field	Type	Null	Default
mobile	varchar(10)	No	
name	varchar(50)	No	
gstnumber	varchar(15)	Yes	NULL
address	varchar(100)	Yes	NULL

4.4 USE CASE DIAGRAM



4.5 UMLS

4.5.1 Class Diagram



SYSTEM IMPLEMENTATION, TESTING

&

MAINTENANCE

5.1 SYSTEM IMPLEMENTATION

Places the emphasis on how the data are represented in the database or on how the data structures are implemented to represent what is modelled. Implementation models include the hierarchical database model, the network database model, the relational database model, and the object oriented database model. In this project the module used is the relational database model, because this model provides the following basic Structure:

- The relational database model is implemented through a very sophisticated relational database management system (RDMBS). The RDMBS performs the same basic function provided by the hierarchical and network DBMS system plus a host of other function that make the relational database module easier to understand and implemented.

- The data and relationship are represented by a collection of tables. • Relational model does not use pointer or links but relates records by the value that contains the value.

- Each table in matrix consisting of a series of row/column intersection. Tables, also called relations, are related to each other by sharing a common entry characteristic.

- Although the tables are completely independent of one another, we can easily connect the data between tables. The relational model thus provides a minimum level of controlled redundancy to eliminate most of the redundancies commonly found in file systems.

5.2 SYSTEM TESTING

Software Testing Fundamentals (STF) is a platform to gain (or refresh) basic knowledge in the field of Software Testing. If we are to ‘cliché’ it, the site is of the testers, by the testers, and for the testers. Our goal is to build a resourceful repository of Quality Content on Quality.

The box approach:

Software testing methods are traditionally divided into white- and black-box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

5.2.1 WHITE BOX TESTING

White-box testing (also known as clear box testing, glass box testing, transparent box testing and structural testing, by seeing the source code) tests internal structures or workings of a program, as opposed to the functionality exposed to the end-user. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT).

While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

Techniques used in white-box testing include:

API testing – testing of the application using public and private APIs (application programming interfaces)

Code coverage – creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)

Fault injection methods – intentionally introducing faults to gauge the efficacy of testing strategies

5.2.2 BLACK BOX TESTING

Black-box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation, without seeing the source code. The testers are only aware of what the software is supposed to do, not how it does it. Black-box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, state transition tables, decision table testing, fuzz testing, model-based testing, use case testing, exploratory testing and specification-based testing.

Specification-based testing aims to test the functionality of software according to the applicable requirements. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behaviour), either "is" or "is not" the same as the expected value specified in the test case. Test cases are built around specifications and requirements, i.e., what the application is supposed to do. It uses external descriptions of the software, including specifications, requirements, and designs to derive test cases. These tests can be functional or non-functional, though usually functional.

Specification-based testing may be necessary to assure correct functionality, but it is insufficient to guard against complex or high-risk situations.[26]

One advantage of the black box technique is that no programming knowledge is required. Whatever biases the programmers may have had, the tester likely has a different set and may emphasize different areas of functionality. On the other hand, black-box testing has been said to be "like a walk in a dark labyrinth without a flashlight. “Because they do not examine the source code, there are situations when a tester writes many test cases to check something that could have been tested by only one test case, or leaves some parts of the program untested.

5.2.3 GREY BOX TESTING

Grey-box testing involves having knowledge of internal data structures and algorithms for purposes of designing tests, while executing those tests at the user, or black-box level. The tester is not required to have full access to the software's source code.[30][not in citation given] Manipulating input data and formatting output do not qualify as grey-box, because the input and output are clearly outside of the "black box" that we are calling the system under test. This distinction is particularly important when conducting integration testing between two modules of code written by two different developers, where only the interfaces are exposed for test.

However, tests that require modifying a back-end data repository such as a database or a log file does qualify as grey-box, as the user would not normally be able to change the data repository in normal production operations. Grey-box testing may also include reverse engineering to determine, for instance, boundary values or error messages.

By knowing the underlying concepts of how the software works, the tester makes better-informed testing choices while testing the software from outside. Typically, a grey-box tester will be permitted to set up an isolated testing environment with activities such as seeding a database. The tester can observe the state of the product being tested after performing certain actions such as executing SQL statements against the database and then executing queries to ensure that the expected changes have been reflected. Grey-box testing implements intelligent test scenarios, based on limited information. This will particularly apply to data type handling, exception handling, and so on.

5.2.4 UNIT TESTING

The software units is a system are module and routines that are assembled and integrated to perform a specific function. Unit testing focuses first on modules, independently of one another to locate errors. This enables to detect errors in coding and logic that are contained within each modules, this testing includes entering data and ascertaining if the value matches to the type and size. The various control and tested to ensure that each performs its action as required.

5.3 SYSTEM MAINTENANCE

An integral part of software is the maintenance one, which requires an accurate maintenance plan to be prepared during the software development. It should specify how users will request modifications or report problems. The budget should include resource and cost estimates. A new decision should be addressed for the developing of every new system feature and its quality objectives. The software maintenance, which can last for 5–6 years (or even decades) after the development process, calls for an effective plan which can address the scope of software maintenance, the tailoring of the post-delivery/deployment process, the designation of who will provide maintenance, and an estimate of the life-cycle costs. The selection of proper enforcement of standards is the challenging task right from early stage of software engineering which has not got definite importance by the concerned stakeholders.

CONCLUSION

6.1 CONCLUSION

- The RETAIL SHOP has been computed successfully and was also tested successfully by taking “test cases”. It is user friendly, and has required options which can be utilized by the user to perform the desired operations.
- Optimum utilization of resources.
- Less processing time and getting required information.
- User friendly.
- Portable and flexible for further enhancement.

6.2 FUTURE SCOPE

This project is developed to fulfil user requirements however there is lots of scope to improve the performance of this project, in the area of user interface, database performance and query process time etc.

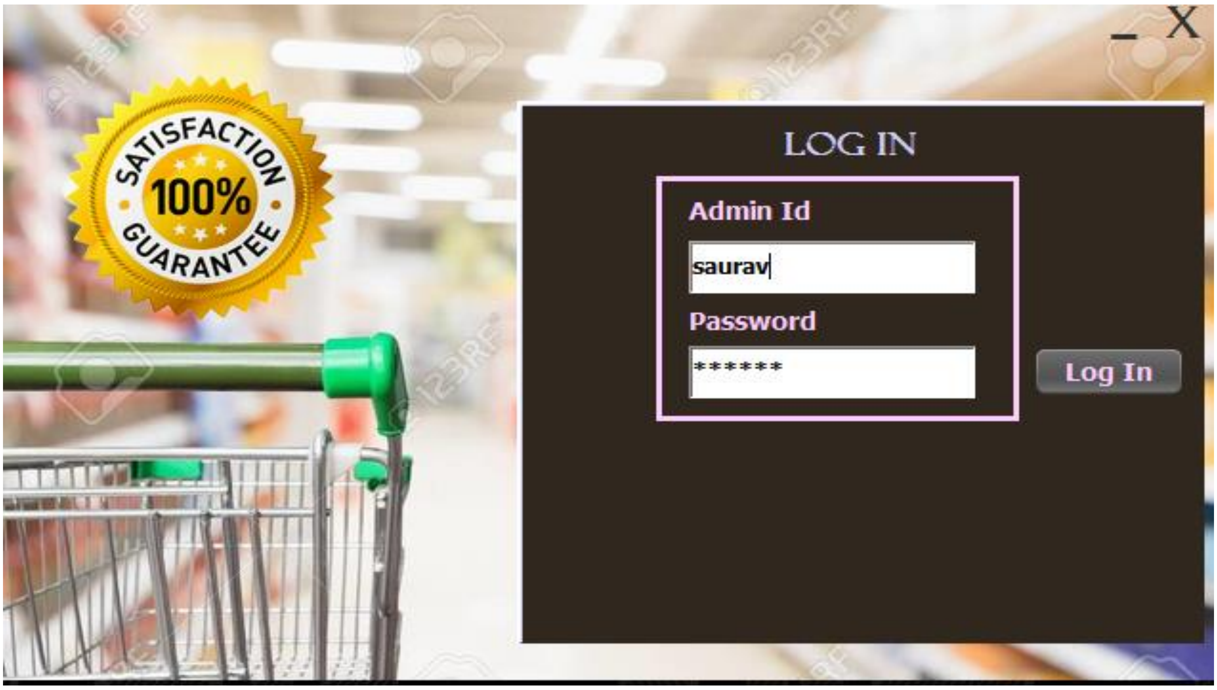
So there are many things for future enhancement of this project. The future enhancements that are possible in the project are as follows:-

- In future shopping will be based on online.
- We can book or return product online.
- It will be a web based system.
- It should be more interactive.
- Update features will be included.
- Searching will be enhanced.
- Payment mode will be provided.
- Customer will also be interact with the system.

SCREENSHOTS

7. SCREENSHOTS(FORMS)

Log In Page Form:



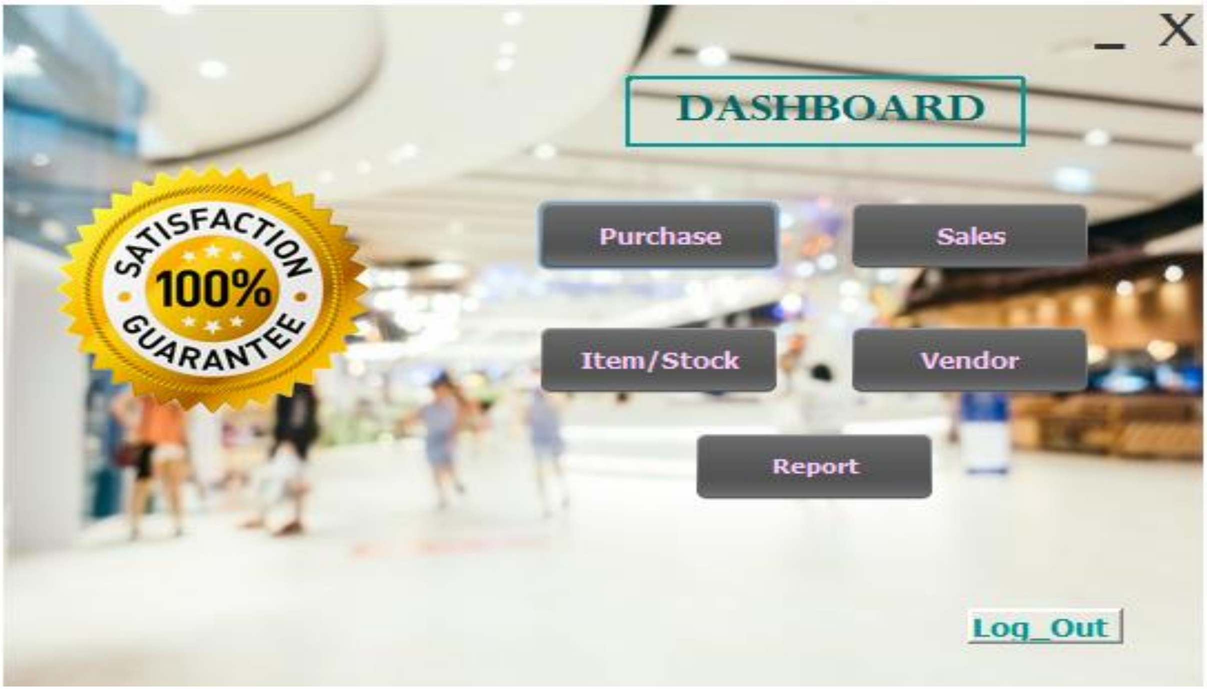
Admin Id

saurav

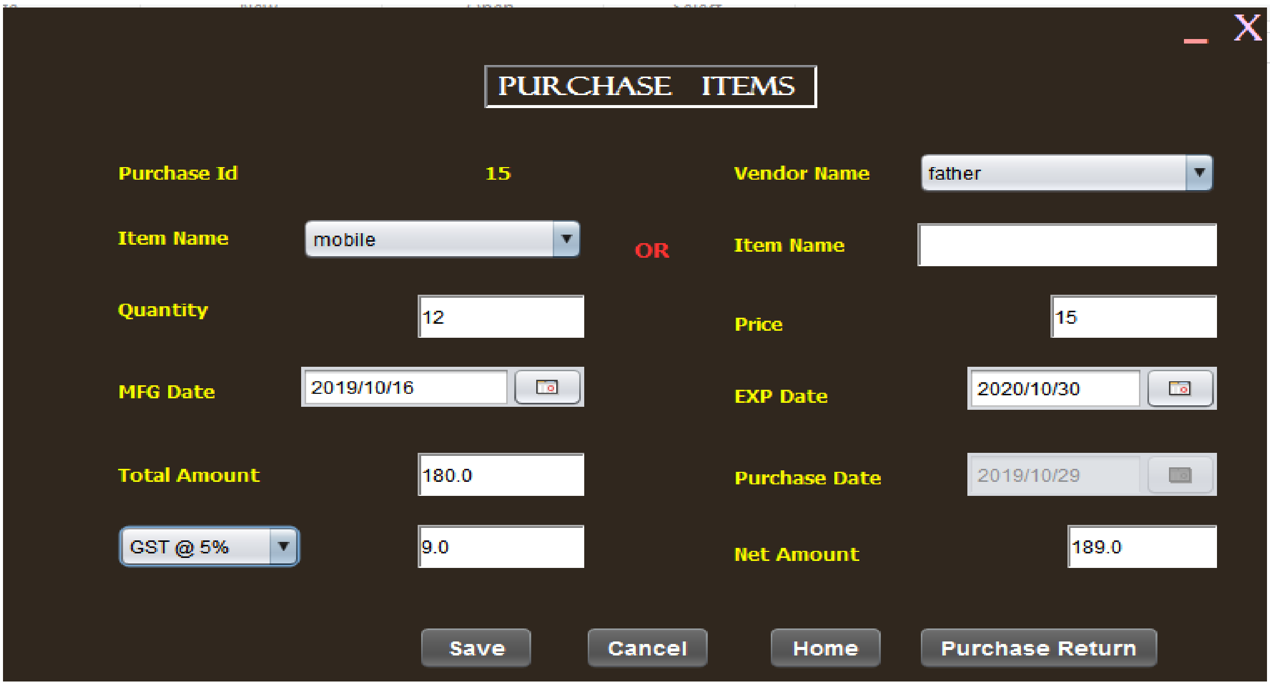
Password

******

Home Page Form:



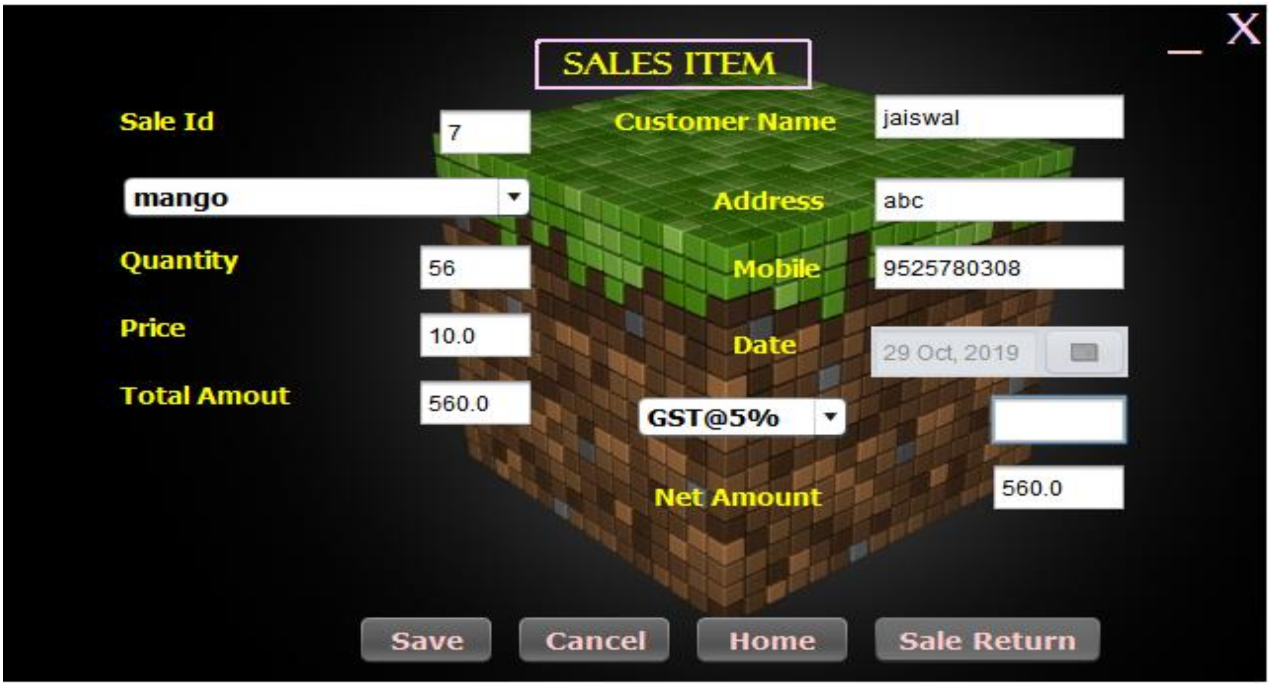
Purchase Form:



PURCHASE ITEMS

Purchase Id	15		Vendor Name	father
Item Name	mobile	OR	Item Name	
Quantity	12		Price	15
MFG Date	2019/10/16		EXP Date	2020/10/30
Total Amount	180.0		Purchase Date	2019/10/29
GST @ 5%	9.0		Net Amount	189.0

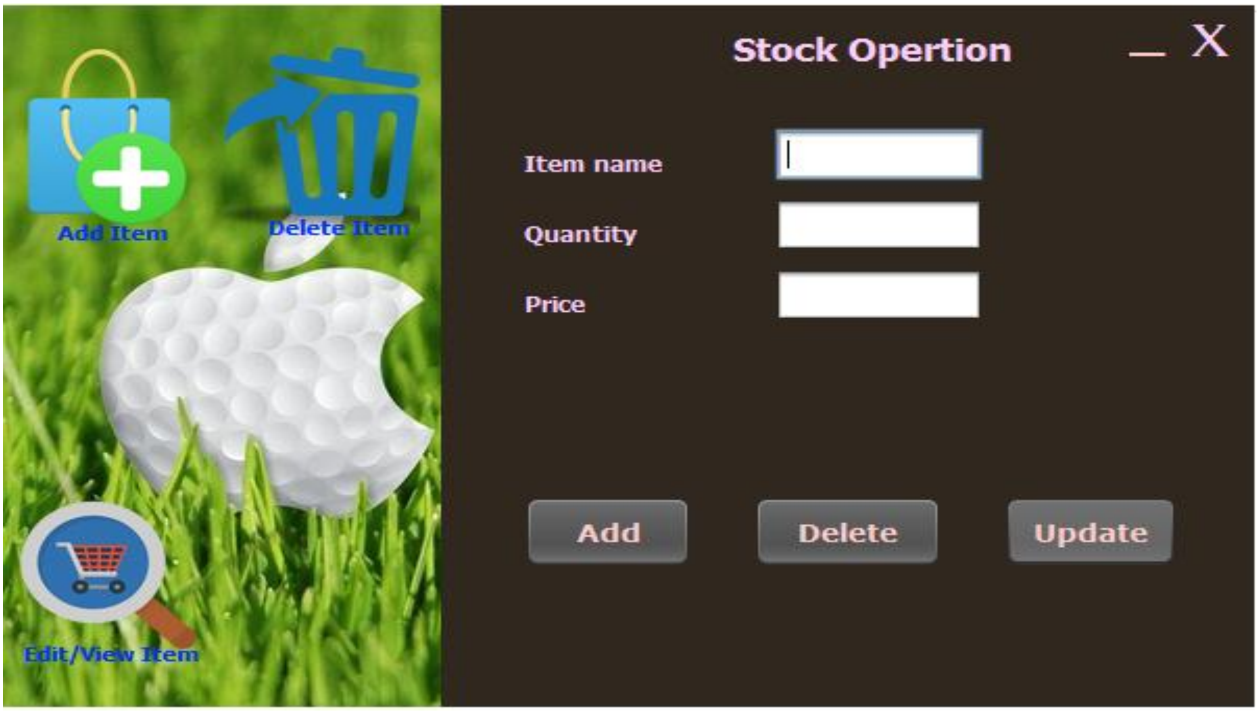
Sales Form:



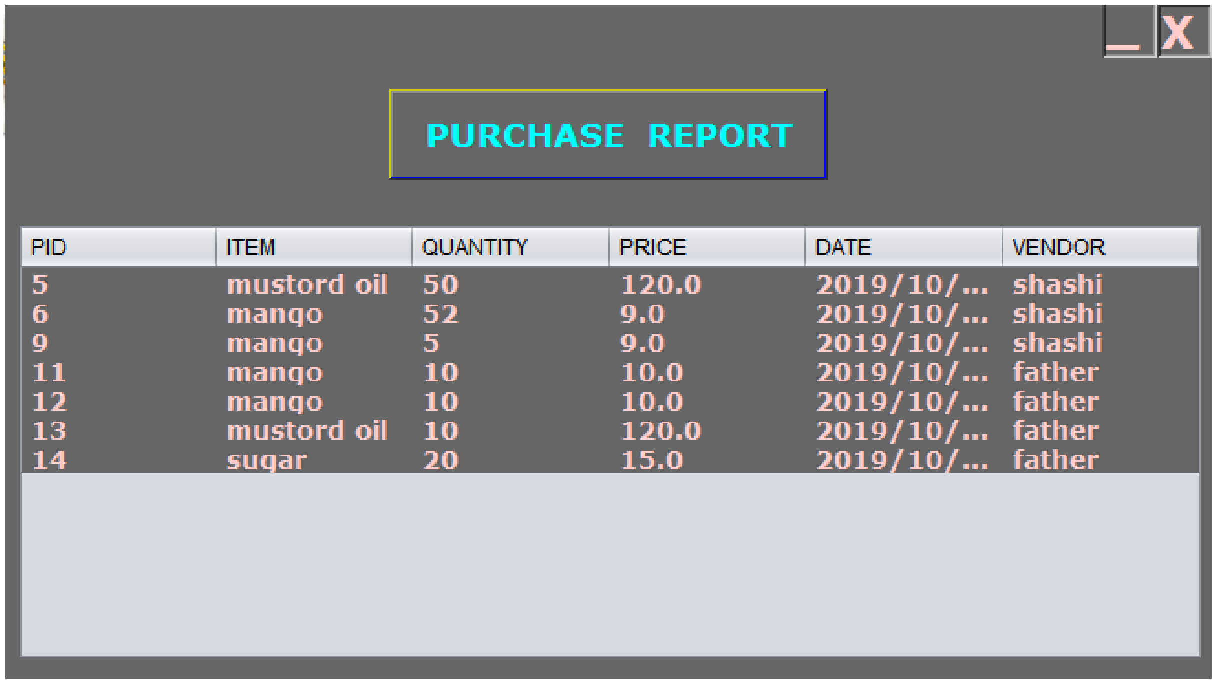
SALES ITEM

Sale Id	7	Customer Name	jaiswal
mango		Address	abc
Quantity	56	Mobile	9525780308
Price	10.0	Date	29 Oct, 2019
Total Amout	560.0	GST@5%	
		Net Amount	560.0

Stock Page Form:



Purchase Table Form:



PURCHASE REPORT

PID	ITEM	QUANTITY	PRICE	DATE	VENDOR
5	mustord oil	50	120.0	2019/10/...	shashi
6	mango	52	9.0	2019/10/...	shashi
9	mango	5	9.0	2019/10/...	shashi
11	mango	10	10.0	2019/10/...	father
12	mango	10	10.0	2019/10/...	father
13	mustord oil	10	120.0	2019/10/...	father
14	sugar	20	15.0	2019/10/...	father

SAMPLE CODE

```
import java.awt.event.WindowStateListener;
import java.sql.*;
import java.util.Random;
import javax.swing.JOptionPane;
public class HomePage extends javax.swing.JFrame {
    public HomePage() {
        initComponents();
    }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {
        jFrame1 = new javax.swing.JFrame();
        jFrame2 = new javax.swing.JFrame();
        jPanel1 = new javax.swing.JPanel();
        jPanel2 = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
        loginid = new javax.swing.JTextField();
        jLabel4 = new javax.swing.JLabel();
        loginpassword = new javax.swing.JPasswordField();
        otp = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        loginbtn = new javax.swing.JButton();
        otplabel = new javax.swing.JLabel();
        submitbtn = new javax.swing.JButton();
        filler1 = new javax.swing.Box.Filler(new java.awt.Dimension(250, 0), new java.awt.Dimension(250, 0),
new java.awt.Dimension(250, 32767));
        otptext = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        min = new javax.swing.JLabel();
        closed = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        javax.swing.GroupLayout jFrame1Layout = new javax.swing.GroupLayout(jFrame1.getContentPane());
        jFrame1.getContentPane().setLayout(jFrame1Layout);
        jFrame1Layout.setHorizontalGroup(
            jFrame1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 400, Short.MAX_VALUE)
        );
        jFrame1Layout.setVerticalGroup(
            jFrame1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 300, Short.MAX_VALUE)
        );

        javax.swing.GroupLayout jFrame2Layout = new javax.swing.GroupLayout(jFrame2.getContentPane());
        jFrame2.getContentPane().setLayout(jFrame2Layout);
        jFrame2Layout.setHorizontalGroup(
            jFrame2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 400, Short.MAX_VALUE)
        );
        jFrame2Layout.setVerticalGroup(
            jFrame2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 300, Short.MAX_VALUE)
        );
        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setLocation(new java.awt.Point(250, 150));
        setUndecorated(true);
        setResizable(false);
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowActivated(java.awt.event.WindowEvent evt) {
                formWindowActivated(evt);
            }
        });
        jPanel1.setBackground(new java.awt.Color(255, 255, 255));
        jPanel1.setToolTipText("");
        jPanel1.setAlignmentX(10.0F);
        jPanel1.setAlignmentY(10.0F);
        jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
        jPanel2.setBackground(new java.awt.Color(204, 204, 255));
        jPanel2.setBorder(new javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
        jPanel2.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
        jLabel2.setFont(new java.awt.Font("Felix Titling", 1, 18)); // NOI18N
        jLabel2.setForeground(new java.awt.Color(204, 204, 255));
        jLabel2.setText("LOG IN");
        jPanel2.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 10, 80, 31));
       loginid.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N

loginid.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.LOWERE
D));
        loginid.setOpaque(true);
```

```
    loginid.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        loginidActionPerformed(evt);
      }
    });
    jPanel2.add(loginid, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 80, 150, 30));
    jLabel4.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
    jLabel4.setForeground(new java.awt.Color(255, 204, 255));
    jLabel4.setText("Password");
    jPanel2.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 110, 80, 30));
    loginpassword.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N

loginpassword.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.LO
WERED));
    loginpassword.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        loginpasswordActionPerformed(evt);
      }
    });
    jPanel2.add(loginpassword, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 140, 150, 30));
    otp.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N

otp.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.LOWERED));
    otp.addFocusListener(new java.awt.event.FocusAdapter() {
      public void focusGained(java.awt.event.FocusEvent evt) {
        otpFocusGained(evt);
      }
      public void focusLost(java.awt.event.FocusEvent evt) {
        otpFocusLost(evt);
      }
    });
   jPanel2.add(otp, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 220, 140, 30));
    jLabel3.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
    jLabel3.setForeground(new java.awt.Color(255, 204, 255));
    jLabel3.setText("Admin Id");
    jPanel2.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 47, 74, 30));
    loginbtn.setBackground(new java.awt.Color(51, 51, 51));
    loginbtn.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
    loginbtn.setForeground(new java.awt.Color(255, 204, 255));
    loginbtn.setText("Log In");
```

```
        loginbtn.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                loginbtnMouseClicked(evt);
            }
        });
        loginbtn.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                loginbtnActionPerformed(evt);
            }
        });
        jPanel2.add(loginbtn, new org.netbeans.lib.awtextra.AbsoluteConstraints(270, 140, 80, -1));
        otplabel.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
        otplabel.setForeground(new java.awt.Color(255, 204, 255));
        otplabel.setText("Captcha code");
        jPanel2.add(otplabel, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 200, 100, -1));
        submitbtn.setBackground(new java.awt.Color(51, 51, 51));
        submitbtn.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        submitbtn.setForeground(new java.awt.Color(255, 204, 255));
        submitbtn.setText("Submit");
        submitbtn.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                submitbtnMouseClicked(evt);
            }
            public void mousePressed(java.awt.event.MouseEvent evt) {
                submitbtnMousePressed(evt);
            }
        });
        submitbtn.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                submitbtnActionPerformed(evt);
            }
        });
        jPanel2.add(submitbtn, new org.netbeans.lib.awtextra.AbsoluteConstraints(93, 263, 130, -1));
        filler1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(255, 204, 255), 3));
        jPanel2.add(filler1, new org.netbeans.lib.awtextra.AbsoluteConstraints(73, 43, 190, 140));
        otptext.setFont(new java.awt.Font("Verdana", 1, 14)); // NOI18N
        otptext.setForeground(new java.awt.Color(255, 255, 255));
        jPanel2.add(otptext, new org.netbeans.lib.awtextra.AbsoluteConstraints(260, 220, 90, 30));
        jLabel6.setIcon(new javax.swing.ImageIcon("C:\\java\\netbeans\\MedicalShopManagement\\img\\back
brown3d.png")); // NOI18N
```

```
        jLabel6.setText("jLabel6");
        jPanel2.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(3, 3, 360, 310));
        jPanel1.add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(270, 55, 360, 310));
        jLabel8.setIcon(new
javax.swing.ImageIcon("C:\\java\\netbeans\\MedicalShopManagement\\img\\rsz_qualityguarantee_icon.png")
); // NOI18N
        jPanel1.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 30, 150, 160));
        min.setFont(new java.awt.Font("Felix Titling", 1, 56)); // NOI18N
        min.setForeground(new java.awt.Color(51, 51, 51));
        min.setText("-");
        min.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                minMouseClicked(evt);
            }
        });
        jPanel1.add(min, new org.netbeans.lib.awtextra.AbsoluteConstraints(580, 10, 20, 20));
        closed.setFont(new java.awt.Font("Felix Titling", 1, 24)); // NOI18N
        closed.setForeground(new java.awt.Color(51, 51, 51));
        closed.setText("X");
        closed.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                closedMouseClicked(evt);
            }
        });
        jPanel1.add(closed, new org.netbeans.lib.awtextra.AbsoluteConstraints(610, 0, 20, 20));

        jLabel7.setIcon(new
javax.swing.ImageIcon("C:\\wamp\\www\\ShopManagement\\img\\shopback6.jpg")); // NOI18N
        jPanel1.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 636, 386));
        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
    );
    pack();
}// </editor-fold>
static String otpGenerate()
{
    String numbers="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
            Random r=new Random();
            char[] otptxt = new char[6];
            for(int p=0;p<6;p++)
                otptxt[p]=numbers.charAt(r.nextInt(numbers.length()));
                //otptxt[i]=numbers.charAt(r.nextInt(numbers.length()));
            String otp=new String(otptxt);
            return otp;
}
private void minMouseClicked(java.awt.event.MouseEvent evt) {
   this.setState(ICONIFIED);
}

private void closedMouseClicked(java.awt.event.MouseEvent evt) {
    System.exit(0);
}

private void formWindowActivated(java.awt.event.WindowEvent evt) {
    otplabel.setVisible(false);
    otp.setVisible(false);
    submitbtn.setVisible(false);
   otptext.setVisible(false);
}
private void loginbtnMouseClicked(java.awt.event.MouseEvent evt) {
 String userid=loginid.getText();
 String password=loginpassword.getText();

 if(userid.equals("saurav") && password.equals("saurav")){
     otplabel.setVisible(true);
     otp.setVisible(true);
     otptext.setVisible(true);
     otp.setText("Enter Captcha code");
     submitbtn.setVisible(true);

     String otp=otpGenerate();
```

```
        otptext.setText(otp);
      }
     else{
        JOptionPane.showMessageDialog(null, "User Id or Password is Incorrect...");
        loginid.setText("");
        loginpassword.setText("");
      }
    }
    private void loginbtnActionPerformed(java.awt.event.ActionEvent evt) {
             String id=loginid.getText();
      String pass=loginpassword.getText();
    }
    private void otpFocusGained(java.awt.event.FocusEvent evt) {
        otp.setText("");
    }
    private void submitbtnActionPerformed(java.awt.event.ActionEvent evt) {
        String otptxt=otptext.getText();
        if(otp.getText().equals(otptxt)){
        JOptionPane.showMessageDialog(null, "Login Successfull...");
        ShopHome s=new ShopHome();
        s.setVisible(true);
        s.setLocationRelativeTo(null);
        this.dispose();
             }else{
            JOptionPane.showMessageDialog(null,"Invalid Captcha","",JOptionPane.WARNING_MESSAGE);
            loginid.setText("");
            loginpassword.setText("");
            otp.setText("");
        }
    }
    public static void main(String args[]) {
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new HomePage().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JLabel closed;
    private javax.swing.Box.Filler filler1;
    private javax.swing.JFrame jFrame1;
    private javax.swing.JFrame jFrame2;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JButton loginbtn;
    private javax.swing.JTextField loginid;
    private javax.swing.JPasswordField loginpassword;
    private javax.swing.JLabel min;
    private javax.swing.JTextField otp;
    private javax.swing.JLabel otplabel;
    private javax.swing.JLabel otptext;
    private javax.swing.JButton submitbtn;
}
```

BIBLIOGRAPHY

[1] Joseph S. Valacich, *Modern Systems Analysis and Design*, 4/E, Pearson Prentice Hall, 2004

[2] S Skidmore, R Framer and G Mills, *SSADM Models and Methods* Version 4.

[3] M Goodland and C Slater, *SSADM VERSION 4:A Practical Approach*, McGraw-Hill, 1995.

[4] Korth,Henry F- *Database System Concepts* , Hill Book,Year:2014 Edition 4

[5] Siberschatz- *Database System Concepts* , Year:2014

[6] http://www.javatpoint.com/

[7] http://www.javaworld.com/javaworld/jw-01-1998/jw-01-bookreview.html

[8] http://java.sun.com/javase/technologies/desktop/