**Introduction**

This is a software which can be used by a automobile centre for keeping the records of items which is to be transacted for purchase and sell. It helps to keep a record of all the transaction and a password login is created for security.

In proposed system we do not have to maintain everything manually. Throughout this system if any transaction occurs it is corresponding entries is done automatically because database management system gives facility of having relationship between the table. In proposed system we do not have to maintain different accounting books.

**ABOUT PROJECT**

The project entitled “BAJAJ SPARE PARTS MANAGEMENT SYSTEM WITH GST BILL” for a software program. This project is developed for “DADDI AUTOMOBILES” the information for this project is collected from **”DADDI AUTOMOBILES FROM GADHINGLAJ”**.

The information stored Database is used for report generation. The System is specially designed to maintain storage item, product detail, customer detail and bill.

This system can be used to manage all records and database. The project will help the store keeper fast billing. Essay to maintain in future prospect. The project cannot be used for other in other managements.

ORGANIZATION PROFILE

|  |  |
| --- | --- |
| **Organization Name :** | **Daaddi Motors** |
| **Owner :** | **Rajshekar Daddi** |
| **Mobile Number :** | **7038801124** |
| **Year of Establishment :** | **2012** |
| **Area of Work :** | **Gadhinglaj** |

**Existing System**

1. The present System is maintained manually and involves many registers if some previous records have to be found then all the registers have to be searched until the required is found.
2. Existing manual system partly work on registers and ledgers work. Here each record is written manually and consumes lot of time.
3. All the calculations are done manually so the error rate is high.
4. Validation, deletion, updating of any record is done manually**.**

**Need of System**

1. Existing system is completely depending on manual work. This is information store in registers.
2. In existing system more man power is required.
3. This information store in register is not accurate as well as there is various mistakes in data fill process of system.
4. To storage large amount of data, more stationary is required which increases the cost of stationary.
5. In existing system lots of time is spend on information in register and managing mistakes and not accuracy of the records.
6. The existing system, any authorized system can access the register so, security is providing in existing system.
7. In existing system, we cannot modify the record also we cannot keep backup of document.
8. The management cannot be effectively followed by existing system.

**Scope of System**

1. This software plays important role in work of spares.
2. It maintains information of customer and invoice records .
3. It maintains information of patient when patient first comes to optician.
4. It maintains check up information of patient after check up by opticians.
5. It can provide facility to maintain appointments of optician.
6. It maintains sales records and dispense records.

**Proposed System**

There are several aspects where computerization scores over the manual system. Some of them are: -

1. This application software plays important role in work of automobile centre.
2. It maintains information of spare part.
3. It maintains information of customer.
4. It maintains distributor information.
5. It maintains stock information.
6. It maintains bill report.

**Objective of System**

1. Easy to generate report for any transaction
2. It is very much faster than manual system.
3. Easy and fastest record finding technique.
4. It is very much flexible to work
5. Data can be stored for a longer record.
6. To provide security and to save valuable time.
7. To get sufficient information of each customer.

**Requirement Engineering**

**Requirement gathering**

Information of this software is collecting by visiting to automobile centre. Software requirements are specified directly as per user demands and requirements.

And by observing of system and working.

**SRS (Software Requirements Specification)**

A] Functional Requirements

1. Software created should be able to store information of every customer and distributor and should return information whenever needed.
2. It should provide facility to find customer and distributor information from stored information.
3. To reduce the paperwork.
4. It’s easy to maintain all date wise report.
5. It will allow only authorized person to operate the system.
6. It makes billing procedure very faster and accurate.

B] Non-Functional Requirements

1. Software should be reliable and should work correctly.
2. It should provide proper user interface to user.
3. Software should provide proper security and authorization to user.

**System Development Life Cycle**

**THE PROCESS MODEL USED FOR THE SYSTEM:**

The process model used for this system “Classic Life Cycle” as this is simple and is best for small scale project.

The “Classic Life Cycle” is also called System Development Life Cycle (SDLC). It is defined “The growth of an information system is through various identifiable stages. These stages are grouped together and referred as SDLC.” The structure of its stages which we used in our project is as follows:

Waterfall model

**System**

**Analysis**

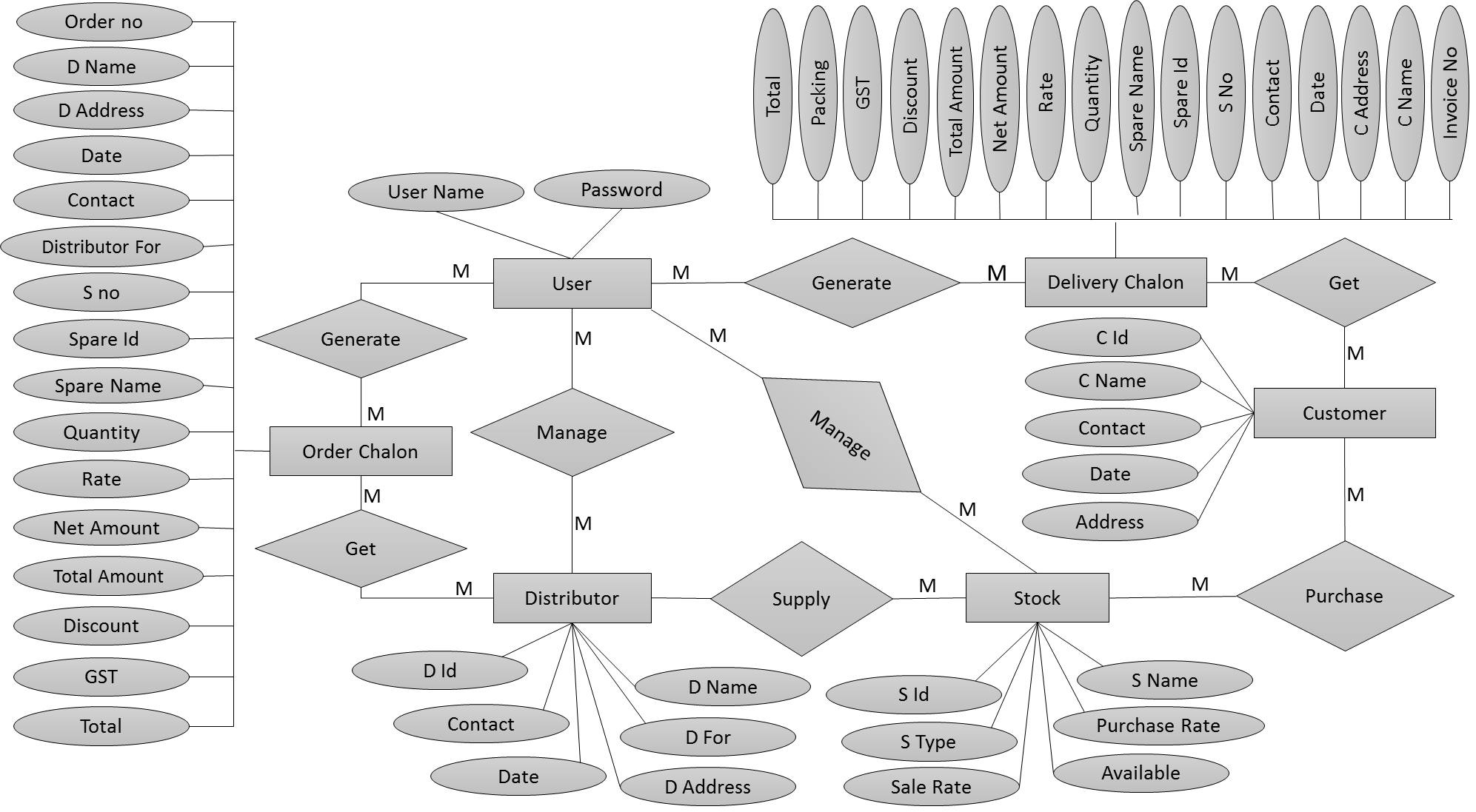
**Design**

**Code**

**Testing**

**Maintenance**

**Entity Relationship Diagram**



**Data Flow Diagram**

**0th level DFD**



0.0

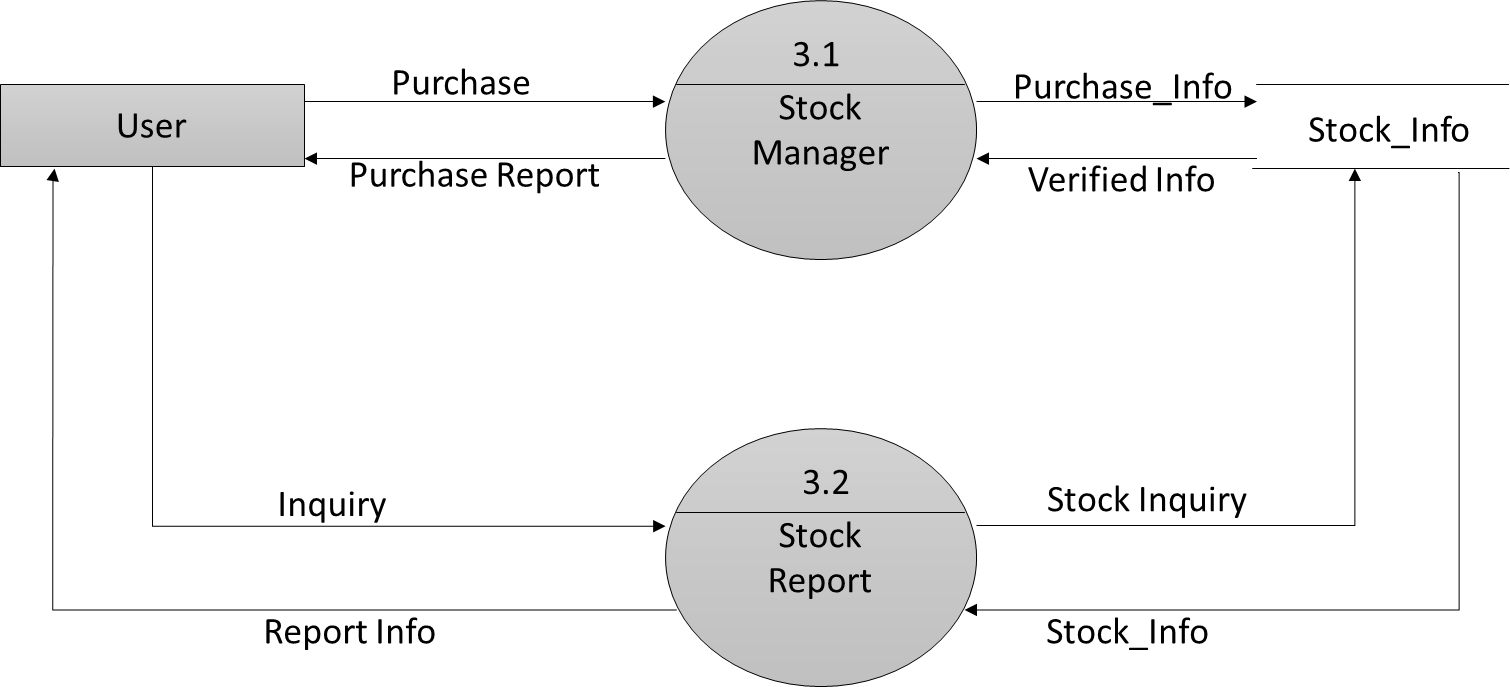
**1st level DFD**



**2nd level DFD for 2nd process**



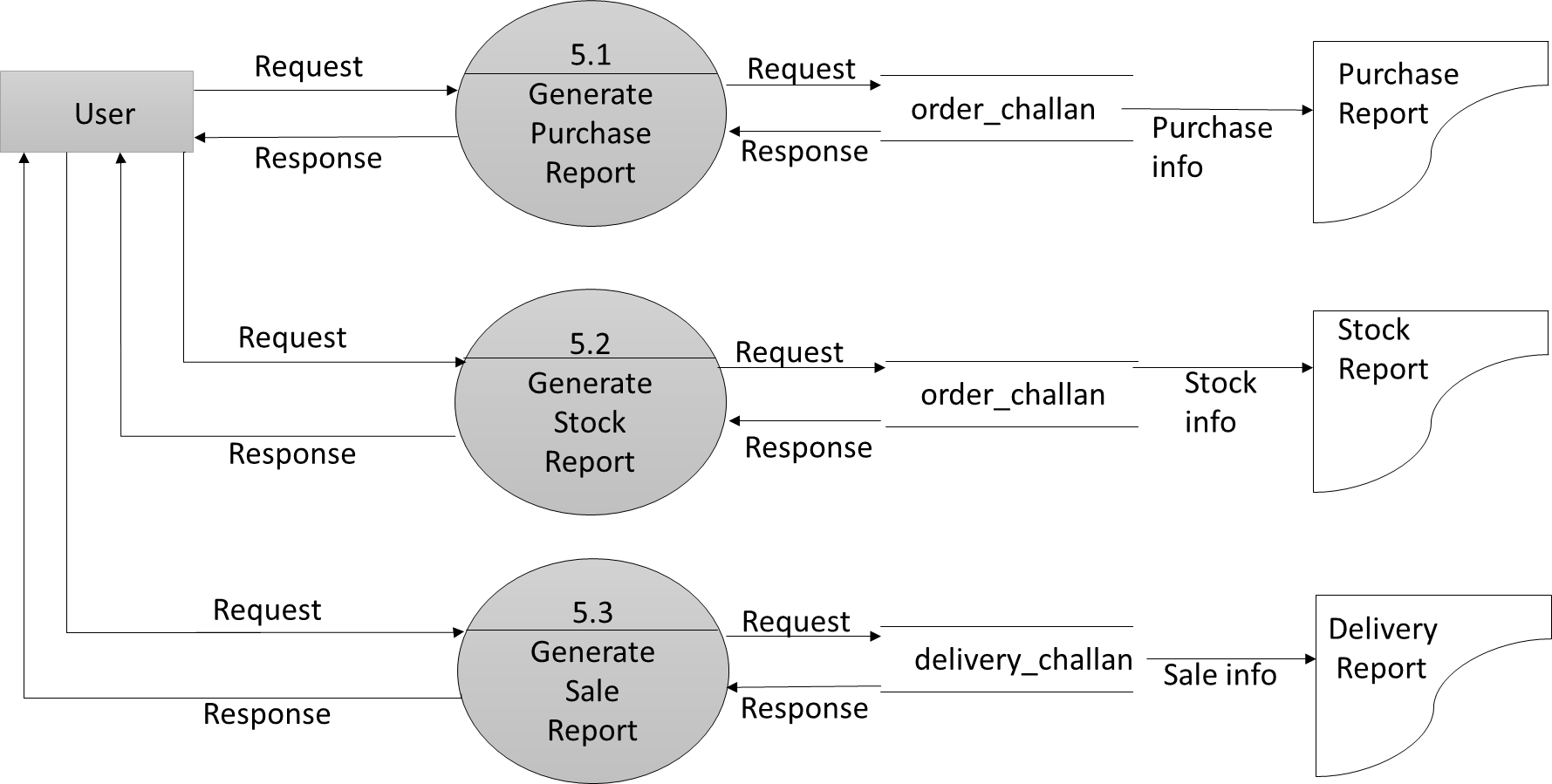
**2nd level DFD for 3rd process**



**2nd level DFD for 4th process**



**2nd level DFD for 5th process**



**Software Requirements: -**

* NetBeans 8.1
* SQL Server

**Hardware Requirements: -**

* Intel dual core or above
* Ram 1.0GB or Above
* Minimum 20Gb or Above

**Front End: -**

* Advance Java (Swing technology)

**Back End: -**

* SQL Server.

**Reporting**: -

* Jasper report.

Java Is an [object-oriented programming](http://www.freejavaguide.com/object_oriented_programming.html) language developed by James Gosling and colleagues at Sun Microsystems in the early 1990s. Unlike conventional languages which are generally designed either to be compiled to native (machine) code, or to be interpreted from source code at runtime, Java is intended to be compiled to a byte code, which is then run (generally using JIT compilation) by a [Java Virtual Machine](http://www.freejavaguide.com/virtual_machine.html).

Java was started as a project called "Oak" by James Gosling in June 1991. Gosling's goals were to implement a virtual machine and a language that had a familiar C-like notation but with greater uniformity and simplicity than C or C++. The first public implementation was Java 1.0 in 1995. It made the promise of "Write Once, Run Anywhere", with free runtimes on popular platforms. It was fairly secure and its security was configurable, allowing for network and file access to be limited. The major web browsers soon incorporated it into their standard configurations in a secure "[applet](http://www.freejavaguide.com/java_applet.html)" configuration popular quickly. New versions for large and small platforms (J2EE and J2ME) soon were designed with the advent of "Java 2". Sun has not announced any plans for a "Java 3".

Goals of Java Language -

1. It should use the [object-oriented programming](http://www.freejavaguide.com/object_oriented_programming.html) methodology.
2. It should allow the same program to be executed on multiple operating systems.
3. It should contain built-in support for using computer networks.
4. It should be designed to execute code from remote sources securely.
5. It should be easy to use by selecting what was considered the good parts of other object-oriented languages.

**Advantages of Java –**

* Java is easy to learn.   
  Java was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages.
* Java is object-oriented.   
  This allows you to create modular programs and reusable code.
* Java is platform-independent.

One of the most significant advantages of Java is its ability to move easily from one computer system to another. The ability to run the same program on many different systems is crucial to World Wide Web software, and Java succeeds at this by being platform-independent at both the source and binary levels.

Java is distributed.  
Java is designed to make distributed computing easy with the networking capability that is inherently integrated into it. Writing network programs in Java is like sending and receiving data to and from a file.

* Java is secure.  
  Java considers security as part of its design. The Java language, compiler, interpreter, and runtime environment were each developed with security in mind.
* Java is robust.  
  Robust means reliability. Java puts a lot of emphasis on early checking for possible errors, as Java compilers are able to detect many problems that would first show up during execution time in other languages.

Java is Multithreaded is the capability for a program to perform several tasks simultaneously within a program. In Java, multithreaded programming has been smoothly integrated into it, while in other languages, operating system-specific procedures have to be called in order to enable multithreading. 

**ABOUT JAVA NET BEANS IDE 8.2**

The Net Beans Platform is a framework for simplifying the development of Java Swing desktop applications. The Net Beans IDE bundle for Java SE contains what is needed to start developing Net Beans plugins and Net Beans Platform based applications no additional JDK is required.

Applications can install modules dynamically. Any application can include the Update Centre module to allow users of the application to download digitally signed upgrades and new features directly into the running application. Reinstalling an upgrade or a new release does not force users to download the entire application again.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:

Net Beans IDE is an open-source integrated development environment. Net Beans IDE supports development of all Java application types (Java SE (including Java FX), Java ME, web, EJB and mobile applications) out of the box. Among other features are an Ant-based project system, Maven support, refactoring, version control (supporting CVS, Subversion, Mercurial and Clear case).

**Modularity**: All the functions of the IDE are provided by modules. Each module provides a well-defined function, such as support for the Java language, editing, or support for the CVS versioning system, and SVN. Net Beans contains all the modules needed for Java development in a single download, allowing the user to start working immediately. Modules also allow Net Beans to be extended. New features, such as support for other programming languages, can be added by installing additional modules. For instance, Sun Studio, Sun Java Studio Enterprise, and Sun Java Studio Creator from Sun Microsystems are all based on the Net Beans IDE.

**License:** From July 2006 through 2007, Net Beans IDE was licensed under Sun's Common Development and Distribution License (CDDL), a license based on the Mozilla Public License (MPL). In October 2007, Sun announced that Net Beans would henceforth be offered under a dual license of the CDDL and the GPL version 2 licenses, with the GPL linking exception for GNU Class path

* User Interface Management (e.g. Menus and Toolbars)
* User Settings Management
* Storage Management (Saving and Loading any Kind of Data)
* Window Management
* Wizard Framework (Supports Step-by-Step Dialogs)
* Net Beans Visual Library
* Integrated Development Tools

**WHAT IS DATABSASE**

A database is a separate application that stores a collection of data. Each database has one or more distinct Application for creating accessing managing searching and replicating the data it holds.

Other kinds of data stores can be used such as files on the file system or large hash tables in memory but data fetching and writing would not be fast and easy with those types of systems.

So now a day we use relational database management systems (RDBMS) TO store and manage huge volume of data. This is called relational database because all the data is stored into different bales and relations are established using primary keys or other keys known as foreign keys.

A RELATIONAL DATABASE MANAGEMENT SYSTEM(RDMS)

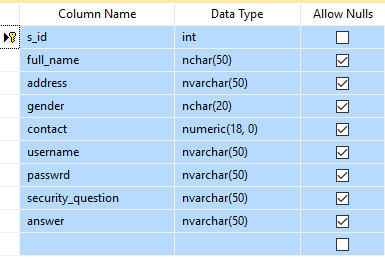
* Enables you to implement a database with tables, columns and indexes.
* Guarantees the Referential Integrity between rows of various tables.
* Updates the indexes automatically.
* Interprets my SQL query and combines information from various tables.

# SQL Server Management Studio

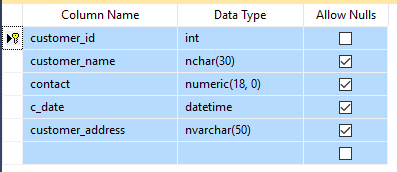
SQL is a fast, easy-to-use RDBMS being used for many small and big businesses.SQL was initially developed at IBM by DONALD D.Chamberlin and Raymond F.Boyce after learning about the relation model from Ted Codd in the early 1970 s.:

* SQL is released under an open-source license. So, you have nothing to pay to use it.
* SQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* SQL uses a standard form of the well-known SQL data language.
* SQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* SQL works very quickly and works well even with large data sets.
* SQL is very friendly to all language, the most appreciated language for web development.
* SQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

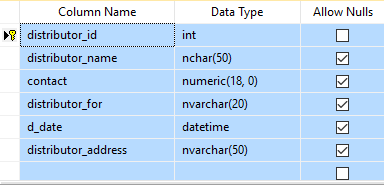
**Sing In: -**



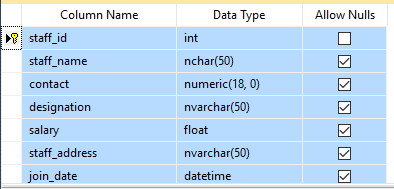
**Customer Info: -**



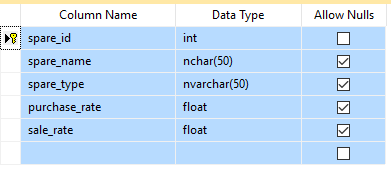
**Distributor Info: -**



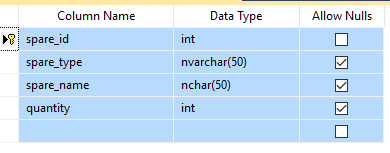
**Staff Info: -**



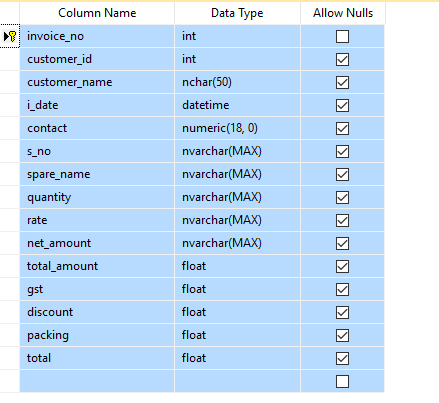
**Spare Info: -**



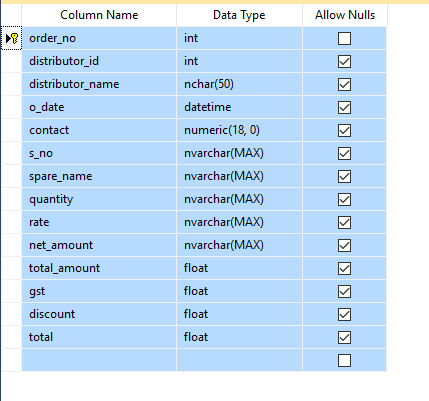
**Stock Info: -**



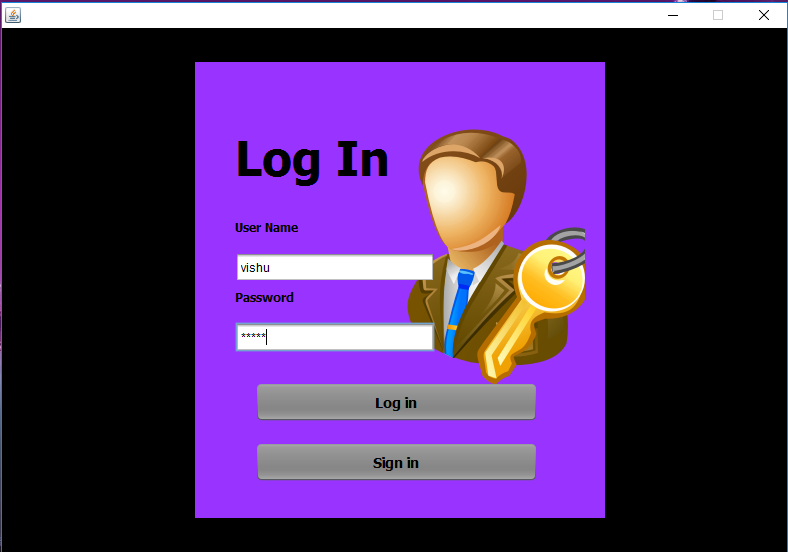
**Invoice Challan: -**



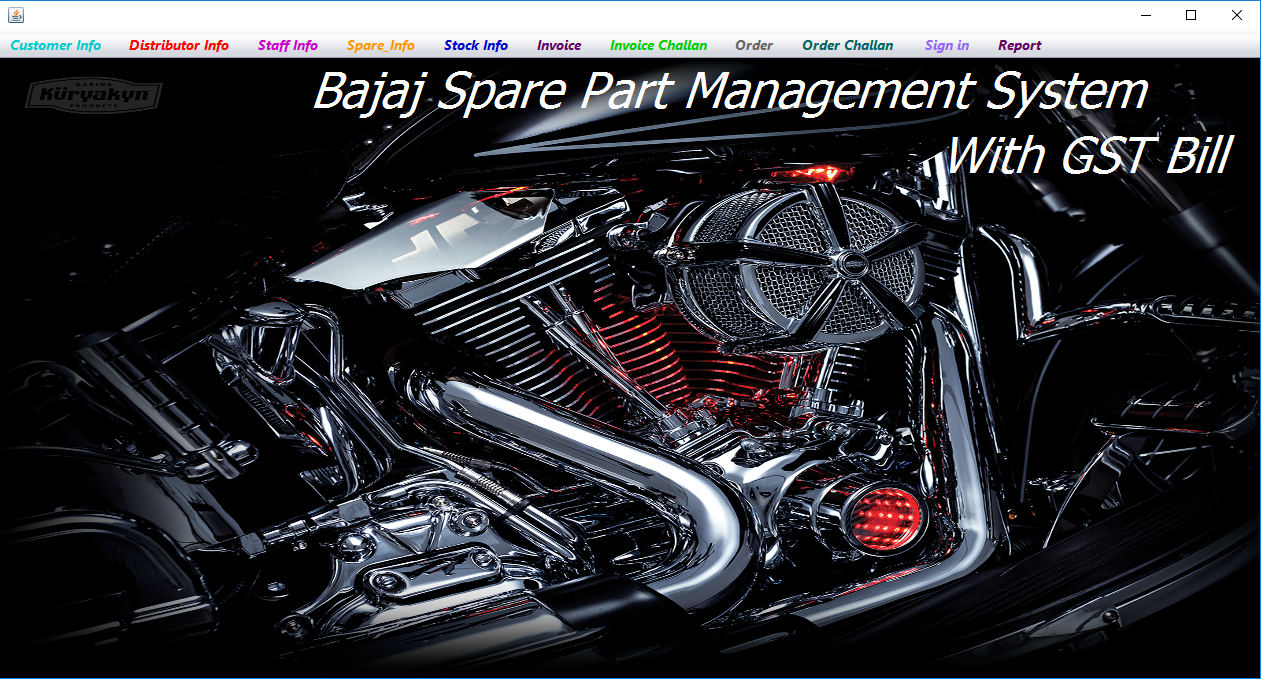
**Order Challan: -**



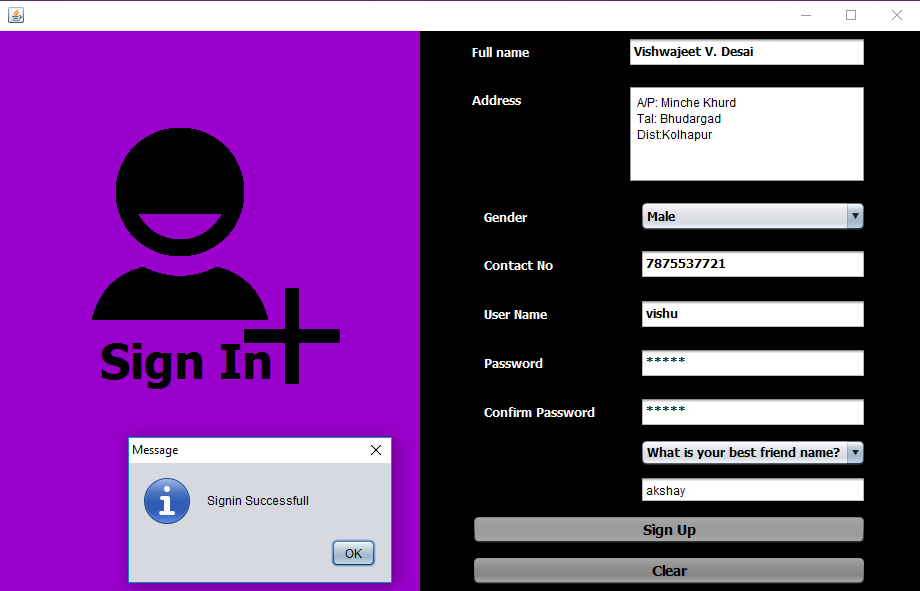
**Login: -**



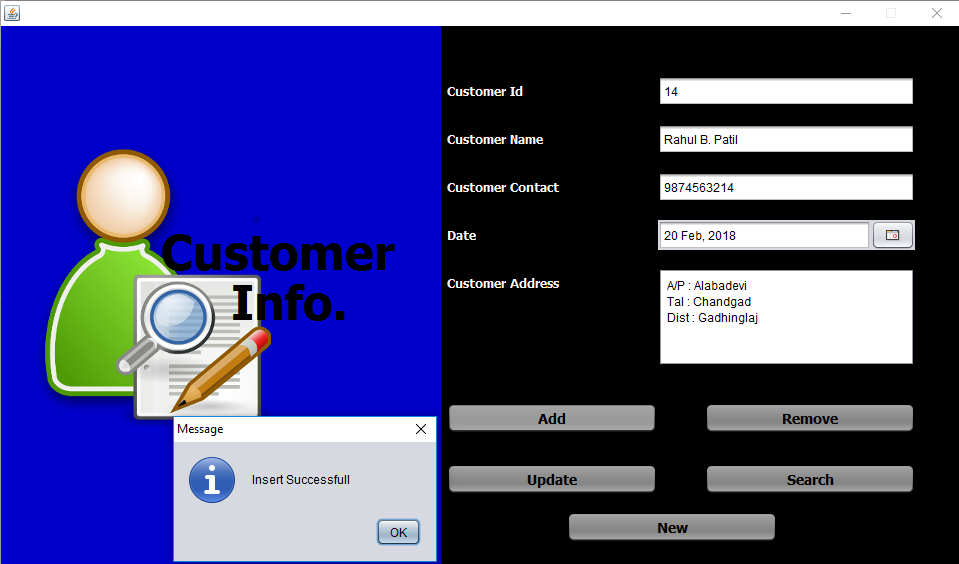
**Home Page: -**



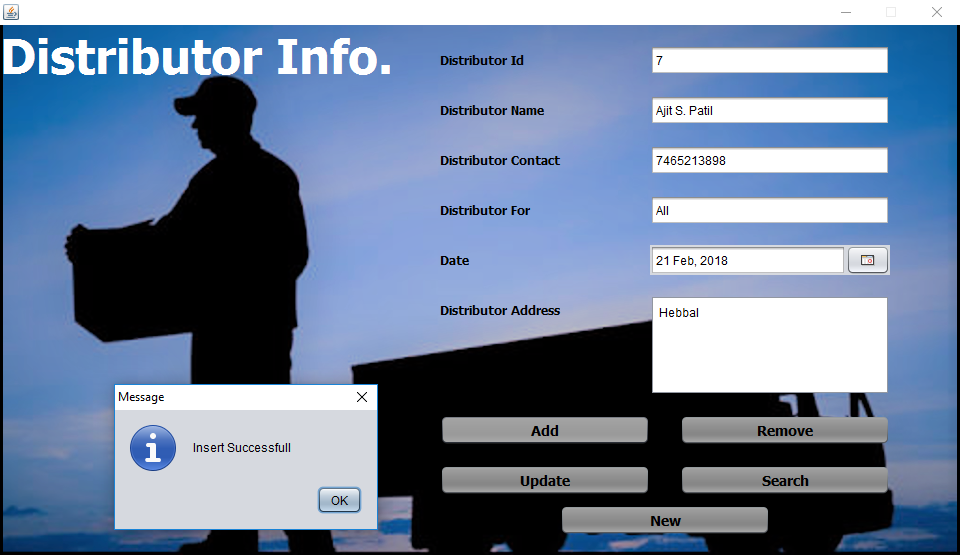
**Sign In: -**



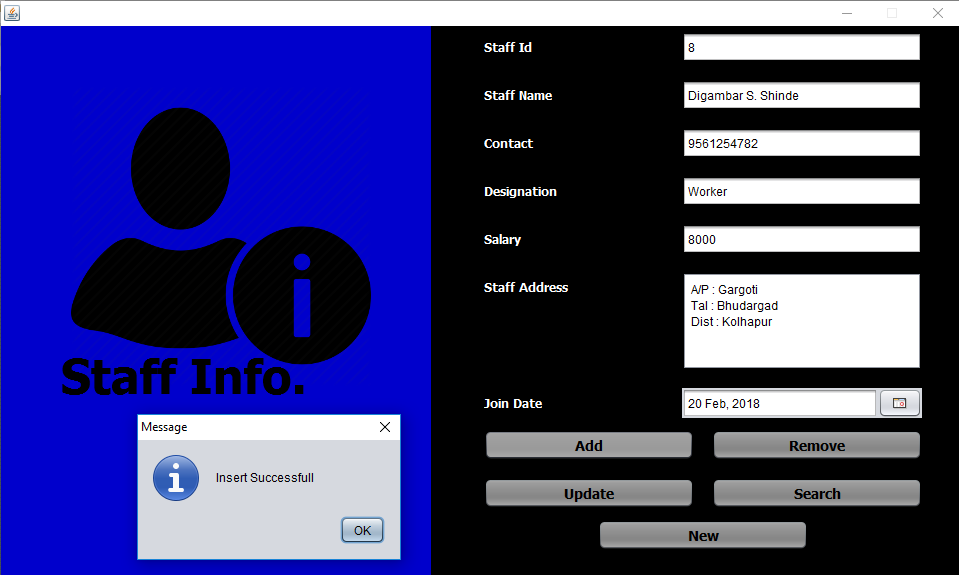
**Customer Info: -**



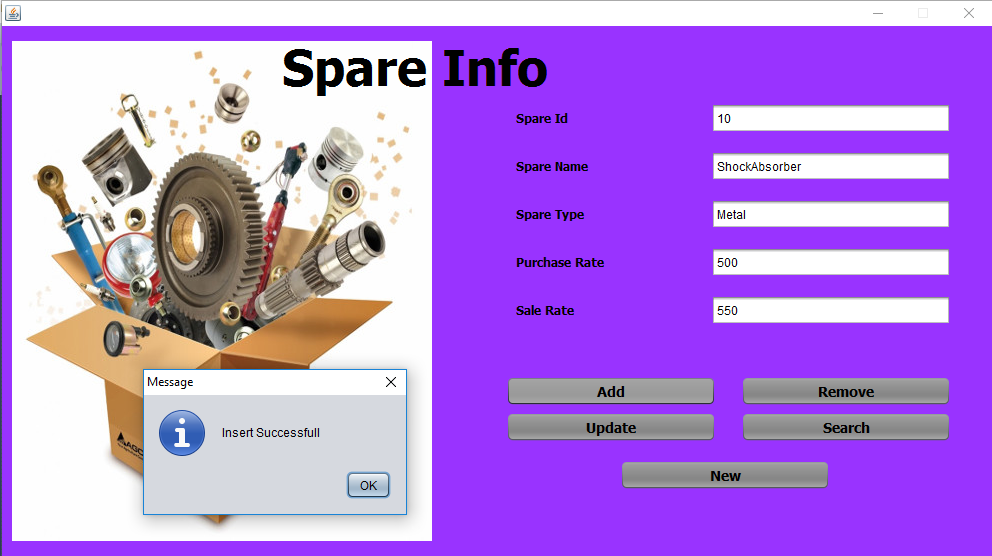
**Distributor Info :-**



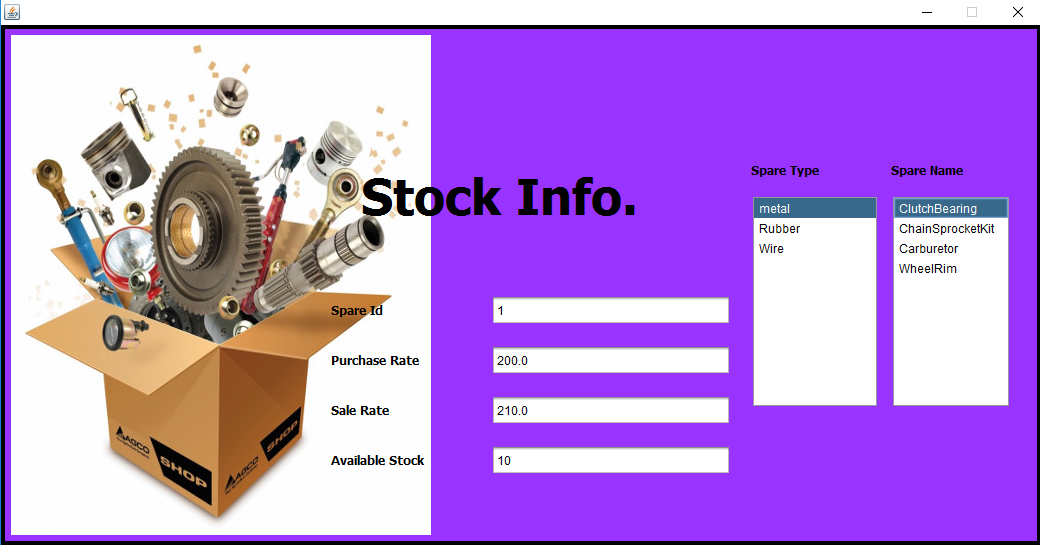
**Staff Info :-**



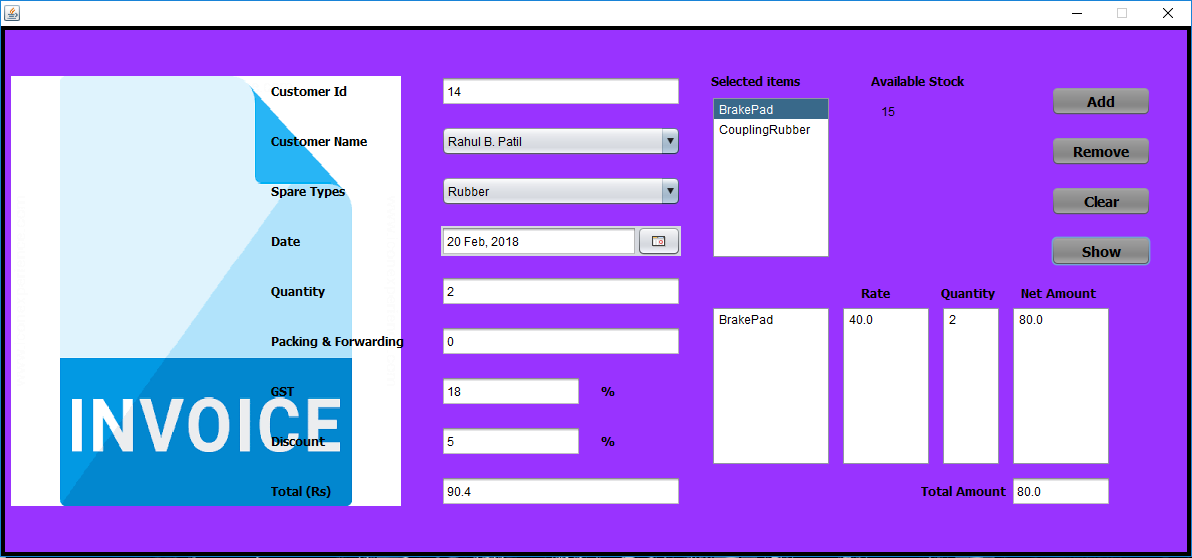
**Spare Info :-**

****

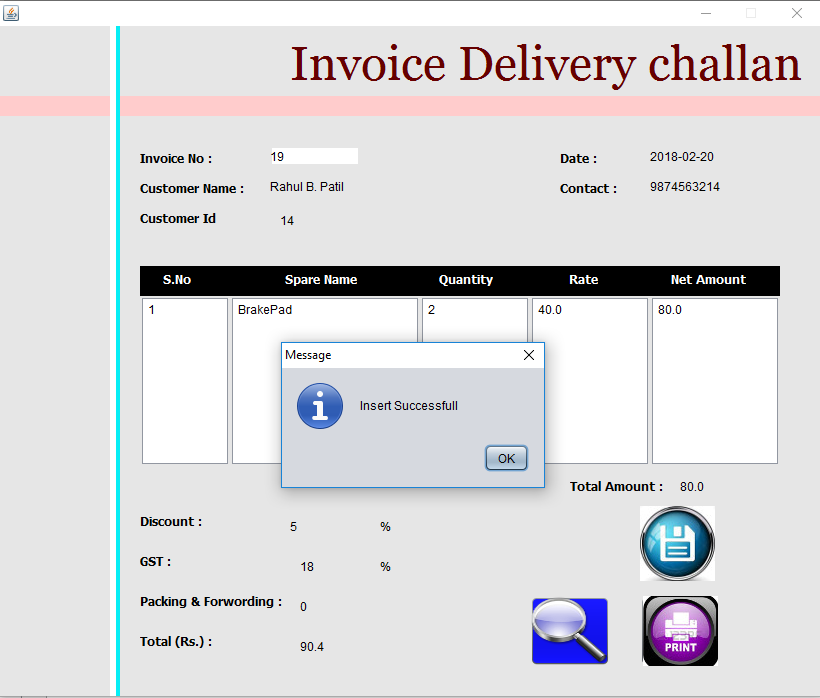
**Stock Info :-**



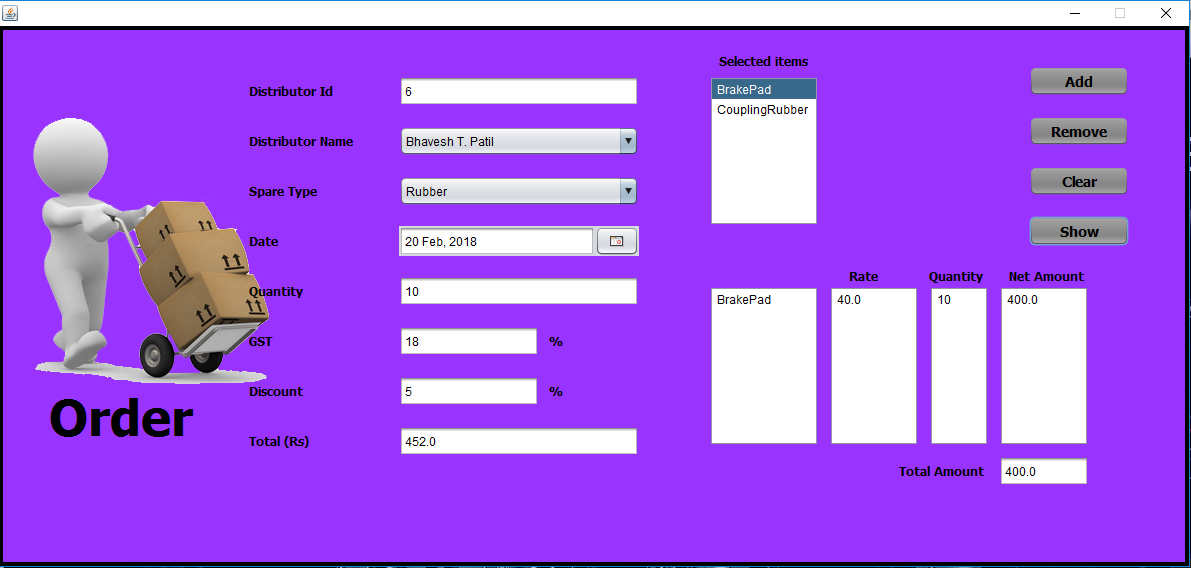
**Invoice :-**

****

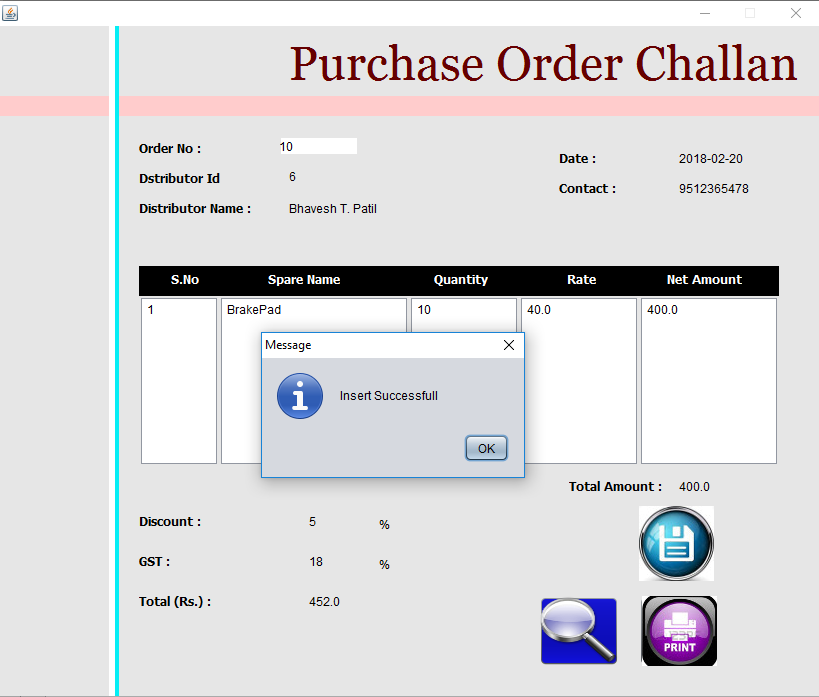
**Invoice Challan :-**



**Order:-**



**Order Challan :-**



.

import java.sql.Connection;

import java.sql.Statement;

import javax.swing.JPanel;

import java.sql.\*;

import javax.swing.JOptionPane;

public class log\_in extends javax.swing.JFrame {

Connection cn=null;

Statement st=null;

public log\_in() {

initComponents();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jPanel2 = new javax.swing.JPanel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

txt\_username = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

txt\_password = new javax.swing.JPasswordField();

jLabel4 = new javax.swing.JLabel();

jLabel1 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setResizable(false);

addWindowListener(new java.awt.event.WindowAdapter() {

public void windowOpened(java.awt.event.WindowEvent evt) {

formWindowOpened(evt);

}

});

jPanel1.setBackground(new java.awt.Color(0, 0, 0));

jPanel2.setBackground(new java.awt.Color(153, 51, 255));

jPanel2.setLayout(null);

jLabel2.setBackground(new java.awt.Color(0, 0, 0));

jLabel2.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N

jLabel2.setText("User Name");

jPanel2.add(jLabel2);

jLabel2.setBounds(40, 150, 81, 30);

jLabel3.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N

jLabel3.setText("Password");

jPanel2.add(jLabel3);

jLabel3.setBounds(40, 220, 81, 30);

txt\_username.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

txt\_usernameActionPerformed(evt);

}

});

jPanel2.add(txt\_username);

txt\_username.setBounds(40, 190, 200, 30);

jButton1.setBackground(new java.awt.Color(102, 102, 102));

jButton1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

jButton1.setText("Log in");

jButton1.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jPanel2.add(jButton1);

jButton1.setBounds(60, 320, 283, 40);

jButton2.setBackground(new java.awt.Color(102, 102, 102));

jButton2.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

jButton2.setText("Sign in");

jButton2.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jPanel2.add(jButton2);

jButton2.setBounds(60, 380, 283, 40);

txt\_password.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

txt\_passwordActionPerformed(evt);

}

});

jPanel2.add(txt\_password);

txt\_password.setBounds(40, 260, 200, 30);

jPanel2.add(jLabel4);

jLabel4.setBounds(99, 11, 35, 14);

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/keylogin.png"))); // NOI18N

jPanel2.add(jLabel1);

jLabel1.setBounds(180, 60, 210, 270);

jLabel5.setFont(new java.awt.Font("Tahoma", 1, 48)); // NOI18N

jLabel5.setText("Log In");

jPanel2.add(jLabel5);

jLabel5.setBounds(40, 60, 230, 70);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addContainerGap(193, Short.MAX\_VALUE)

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 410, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(182, 182, 182))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(34, 34, 34)

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 456, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(34, Short.MAX\_VALUE))

);

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.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

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>

public void setjPanel2(JPanel jPanel2) {

this.jPanel2 = jPanel2;

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try

{

st=cn.createStatement();

String sql="select \* from sign\_in where username='"+txt\_username.getText().toString()+"' and passwrd='"+txt\_password.getText().toString()+"'";

ResultSet rs=st.executeQuery(sql);

if(rs.next())

{

new home\_page().setVisible(true);

this.dispose();

}

else

{

JOptionPane.showMessageDialog(null,"Password is not correct");

}

}catch(Exception ex)

{

System.out.println(ex);

}

}

private void txt\_usernameActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

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

new sign\_in().setVisible(true);

this.setVisible(false);

}

private void formWindowOpened(java.awt.event.WindowEvent evt) {

// TODO add your handling code here:

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

System.out.println("Driver Registred");

cn=DriverManager.getConnection("jdbc:odbc:spare\_parts");

System.out.println("Connection Success");

}catch(Exception ex)

{

System.out.println(ex);

}

}

private void txt\_passwordActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

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(log\_in.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(log\_in.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(log\_in.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(log\_in.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new log\_in().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

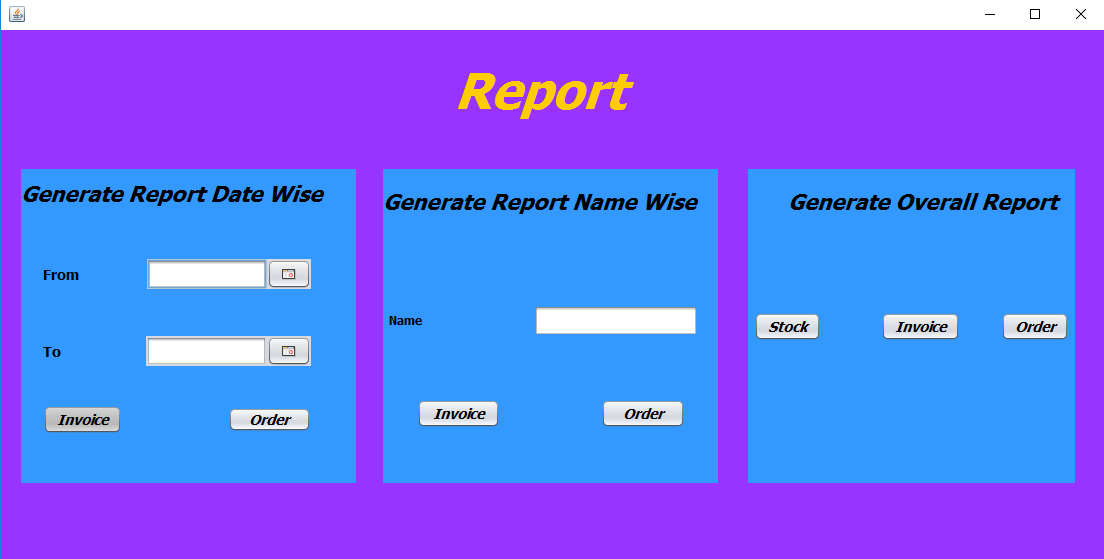
private javax.swing.JPasswordField txt\_password;

private javax.swing.JTextField txt\_username;

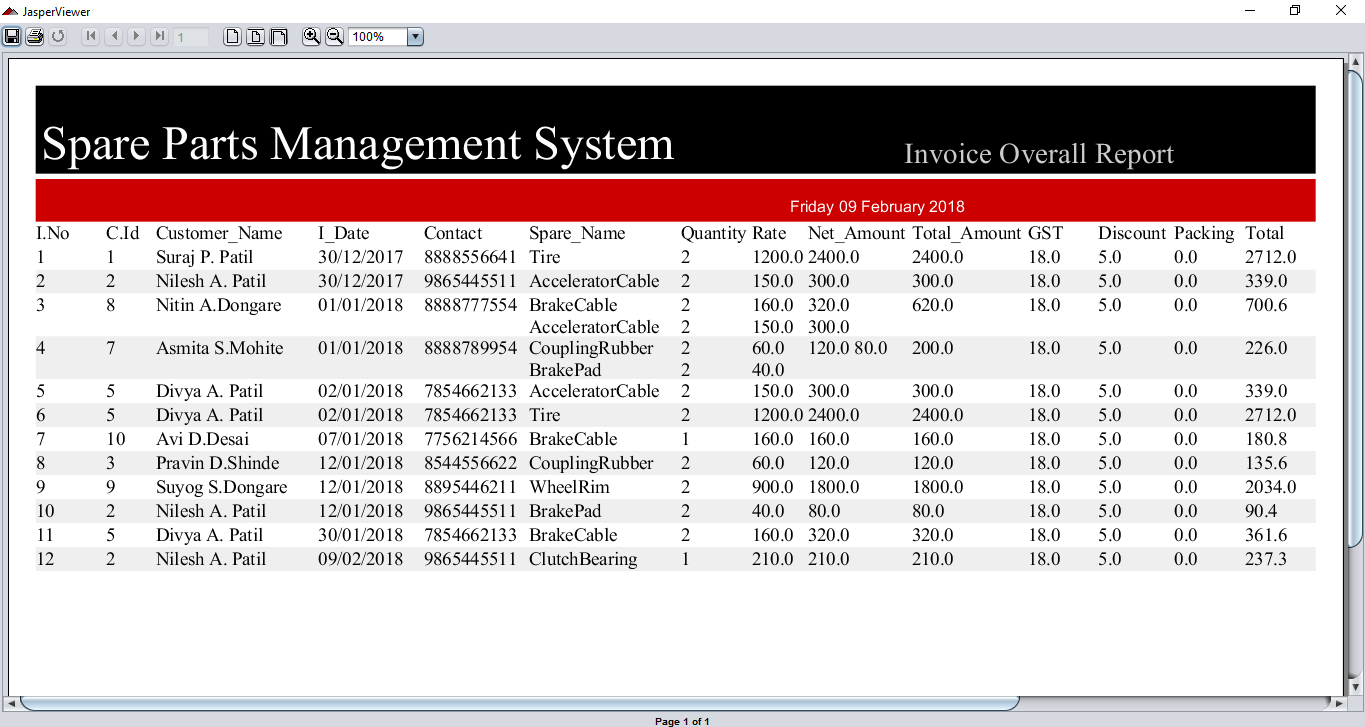
// End of variables declaration

}

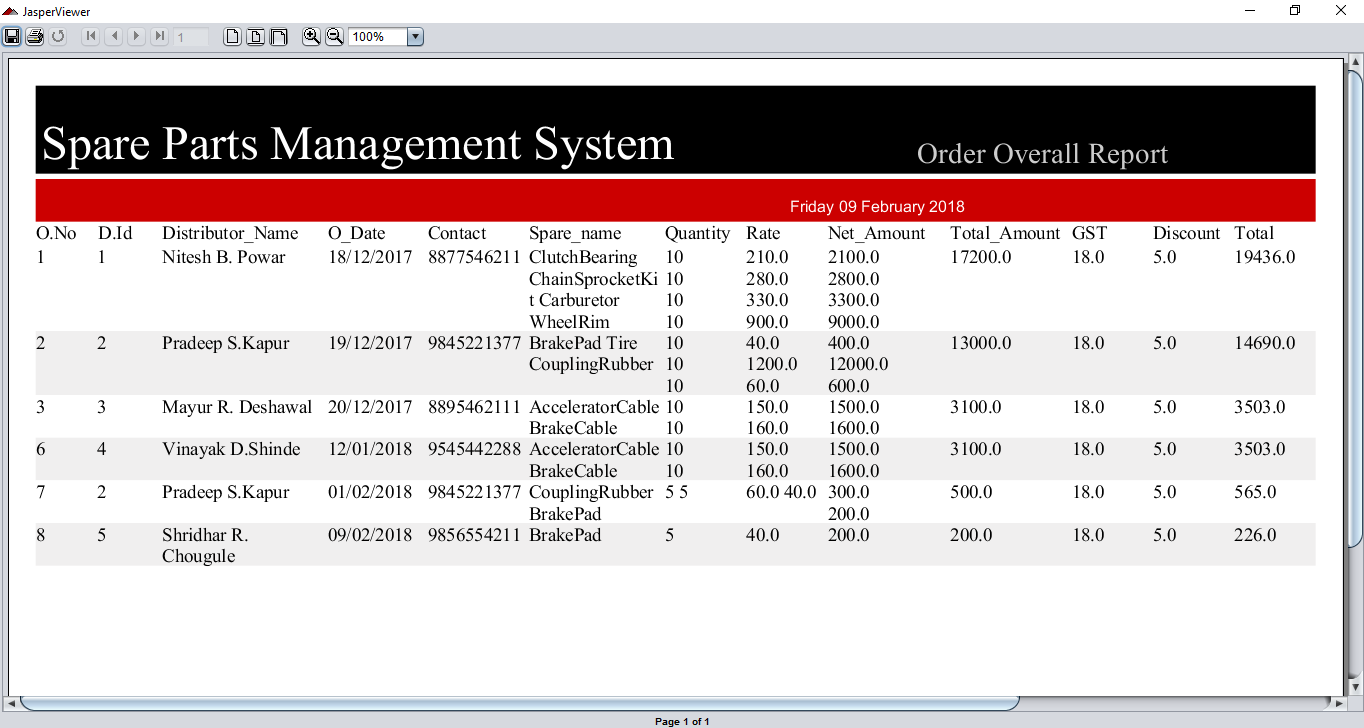
**Report :-**



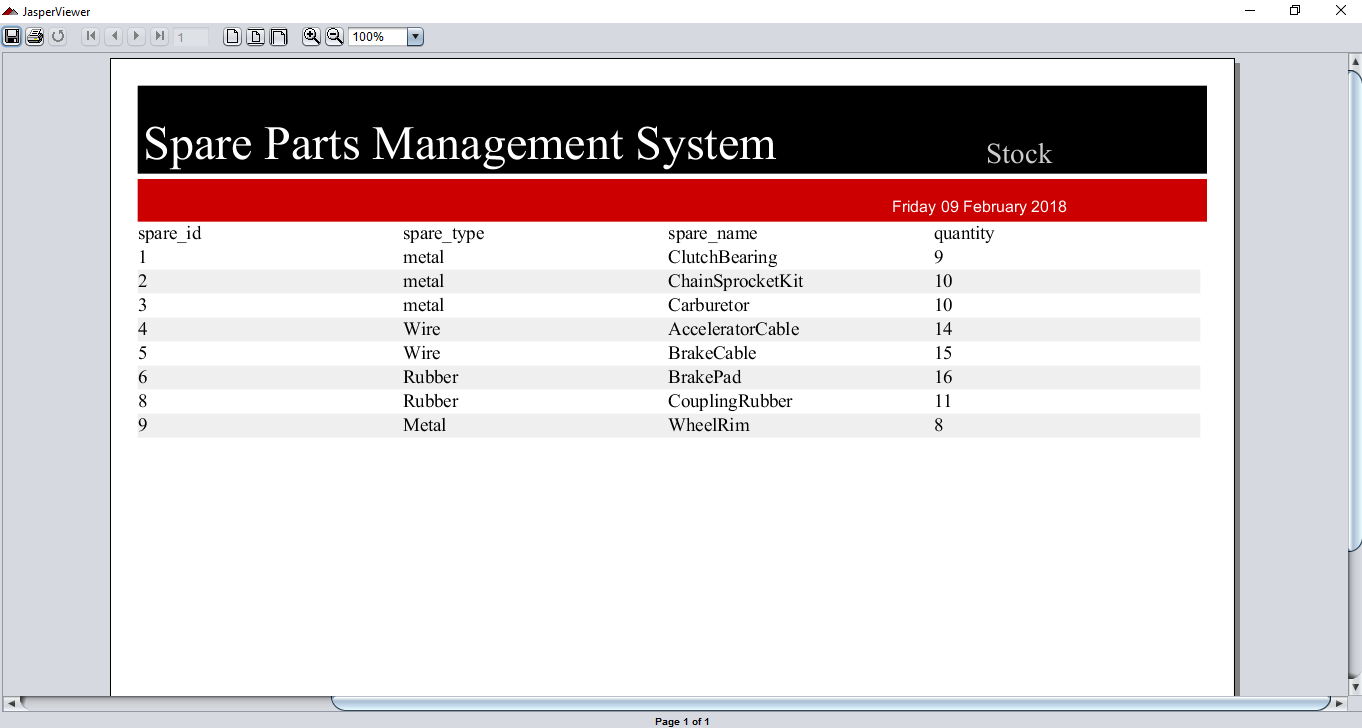
**Invoice Overall:** -



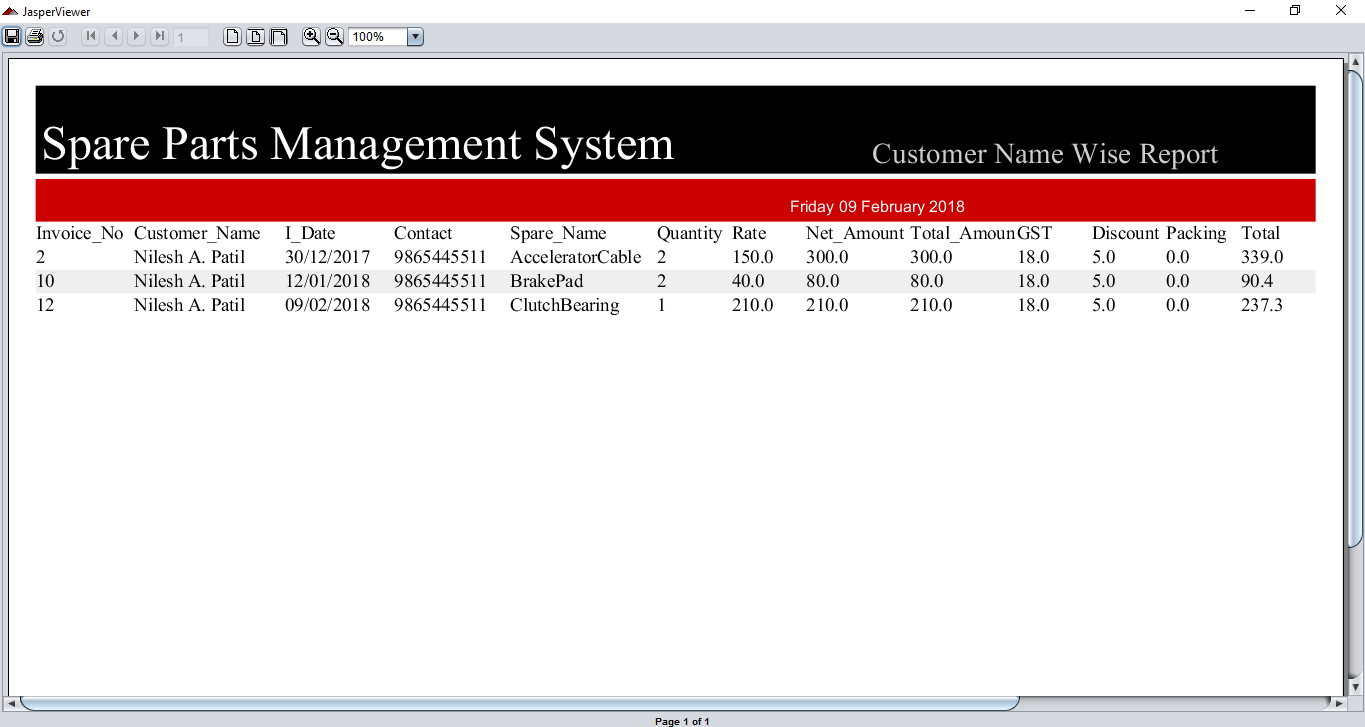
**Order Overall: -**



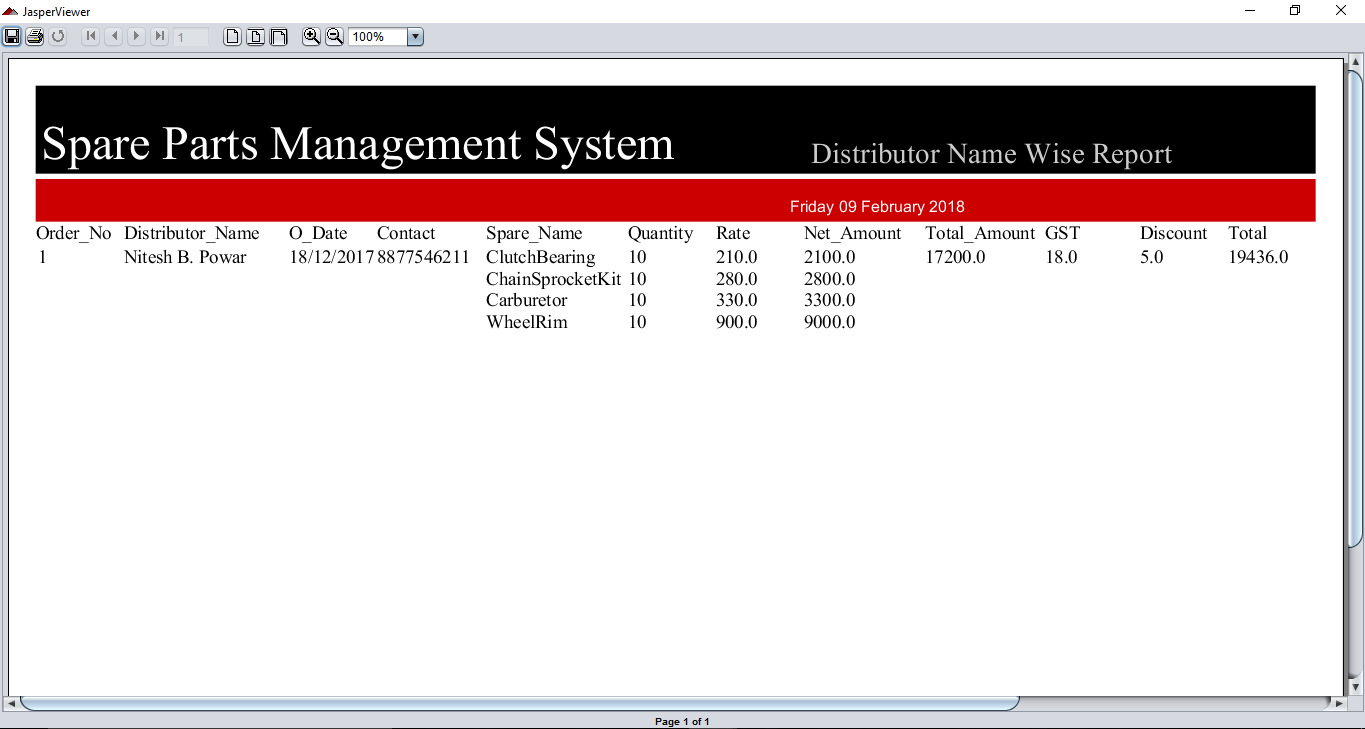
**Stock: -**



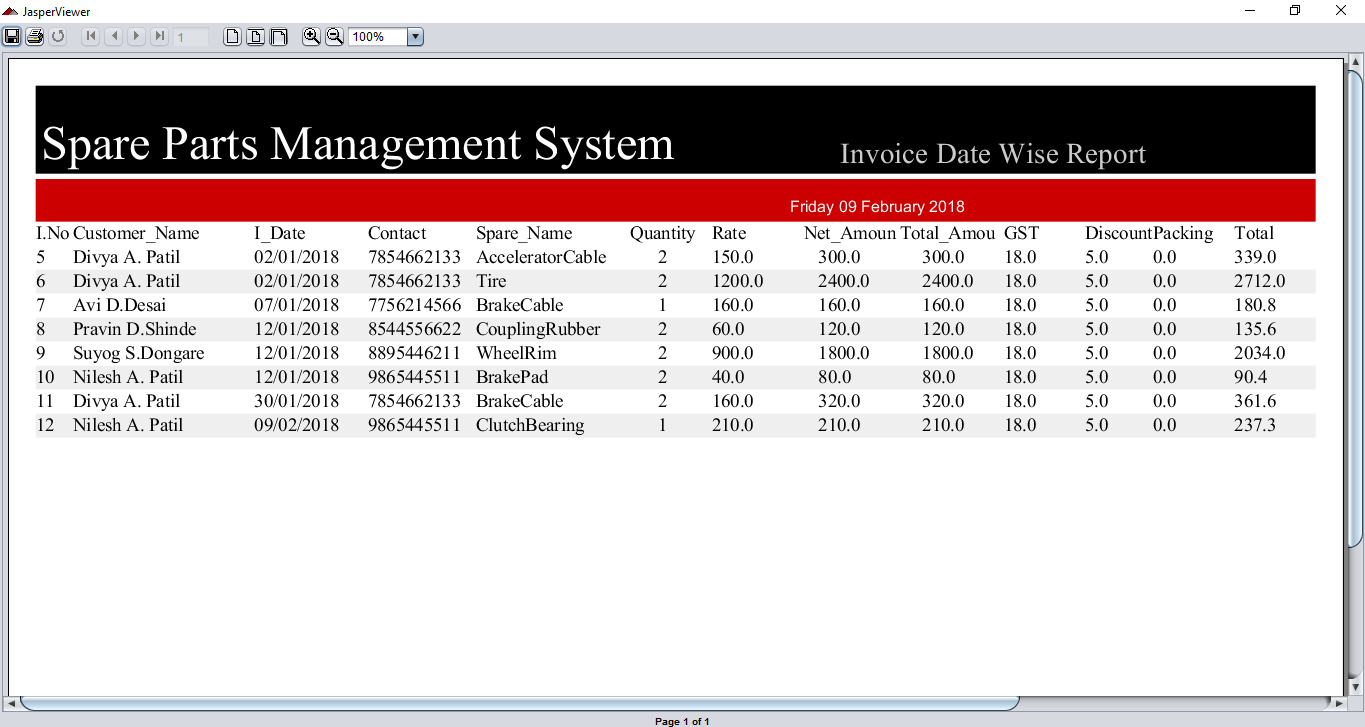
**Customer Name Wise: -**



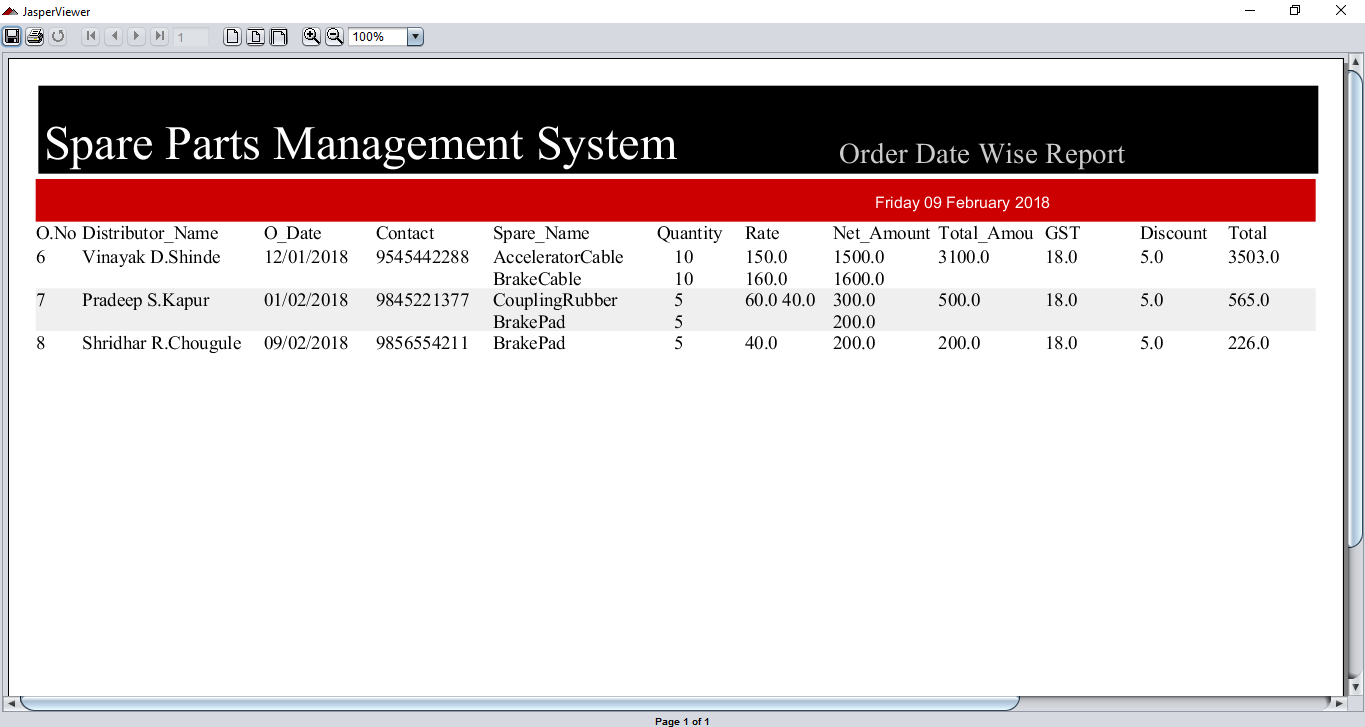
**Distributor Name Wise: -**



**Invoice Date wise: -**



**Order Date Wise: -**



**Limitation**

1. User should have minimum (basic) knowledge about computer to handle the software.
2. User should be trained for handling the system and interact with the various facilities provided in it.
3. This system is only made for spare parts management purpose.
4. Maintenance cost of expensive.
5. Only a trained user can handle the system.

# Conclusion

After designing and implementation of this system we have come to the following conclusion.

1. Due to computerization we can easily update, deletes or insert the data of customer and hence retrieval of any record that is stored becomes easier.

2. Due to computerization, a lot of time is saved because all the paper work can be done on computer with greater accuracy.

3. Changes can be made immediately and efficiently as we require.

4. print- out of updated records can be taken.

5. Due to computerization department can maintain information of customer organization easily in the computer itself

**Bibliography**

* + JAVA the complete reference

-By Herbert Schildt.

* Google
* Tutorials point
* Introduction of Software Engineering