

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI-590014



A DBMS Mini Project Report On

“PROVISION STORE MANAGEMENT SYSTEM”

Submitted in the partial fulfilment of the requirements for the award of the Degree of

Bachelor of Engineering in Information Science and Engineering

Submitted by,

CHANDRASHEKAR V

Under the guidance of

Mrs. SANDYA RANI V

(Asst Prof, Dept. of ISE)

Project Guide



Department of Information Science and Engineering

The Oxford College of Engineering

Bommanahalli, Bangalore-68

2020-2021

THE OXFORD COLLEGE OF ENGINEERING

Bommanahalli, Hosur Road, Bangalore – 560068

(Affiliated to Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering



CERTIFICATE

Certified that the project work entitled “**PROVISION STORE MANAGEMENT SYSTEM**” carried out by **CHANDRASHEKAR V** Bonifide students of The Oxford College of Engineering, Bangalore in partial fulfilment for the award of the Degree of **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belgaum** during the year 2020-2021. The Mini project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Mrs.Sandya Rani V

Project Guide

Dr.Kanagavalli R

H.O.D,Dept. of ISE

Dr. G.T RAJU

Principal, TOCE

External Viva

Name of the Examiners

Signature with Date

1. _____

2. _____

ACKNOWLEDGEMENT

A project is a job of great enormity and it can't be accomplished by an individual all by them. Eventually, we are grateful to a number of individuals whose professional guidance, assistance and encouragement have made it a pleasant endeavour to undertake this project.

It gives us great pleasure in expressing our deep sense of gratitude to our respected Founder Chairman **Shri S. Narasa Raju** and to the respected Chairman **Shri S.N.V.L Narasimha Raju** for having provided us with great infrastructure and well-furnished labs.

We take this opportunity to express our profound gratitude to our respected Principal **Dr. G.T RAJU** for his support.

We are graceful to the Vice Principal and Head of the Department **Dr. R Kanagavalli** for her unfailing encouragement and suggestion given to us in the course of our project work.

Guidance and deadlines play a very important role in successful completion of the project on time. We convey our gratitude to **Mrs Sandya Rani V**, Project Guide for having constantly guided and monitored the development of the project.

A note of thanks to the Department of Computer Science Engineering, both teaching and non-teaching staff for their co-operation extended to us.

We thank our parents for their constant support and encouragement. Last, but not the least, we would like to thank our peers and friends.

CHANDRASHEKAR V

ABSTRACT

The project is a complete management based software. The purpose of Provision store management system is to automate the existing manual system by the help of computerized equipment's and full-fledged computer software, fulfilling their requirements, so that their data/information can be stored for a longer period with easy accessing and manipulation of the same. It can assist the workers to concentrate on their activities rather than to concentrate on the record keeping. Thus, it will help the store in better utilization of its items. The store can maintain computerized records without redundant entries. That means one need not be distracted by information that is not relevant, while being able to reach the information. Basically, the project describes how to manage for good performance and better services for the customers. Stored procedures are used to categorize items based on the product name and billing them. The project introduces a point relation to the customer ,whose role is to add points to the customers shopping list,for purchased items costing 10,000 and above .The project also includes discount section which is handled by trigger. The project is very useful for those small and large scale business people without costing scanners and a very fast processing. The project provides facilities like storing managers,employees and customers details, the sales and employee report and items price details.

Table of Contents

Chapter no	Name	Page.no
1	Introduction	1
1.1	Preamble	1
1.2	Problem statement	1
1.3	Proposed solution	1
2	Analysis and system requirement	2
2.1	Analysis survey	2
2.2	System requirement	3
3	System design and modelling	4
3.1	Preliminary design	4
3.1.1	ER diagram	4
3.1.2	Schema diagram	5
3.2	Normalization	6
4	Implementation	7
4.1	Module 1 description	7
4.2	Module 2 description	8
5	Testing	25
5.1	Introduction on testing	25
5.2	Test cases	26
6	Conclusion	27
7	Appendix A:Snapshots	28
8	References	31

1.INTRODUCTION

1.1 Preamble

The project provision store management system is a management-based application that allows the store manager to handle all store activities. The ability to manage various item entries and billing make this system very flexible and convenient.

The store manager is a very busy person and does not have the time to sit and manage the entire activities manually on paper. This application gives him the power and flexibility to manage the entire system from a single online system.

Provision store management project provides secret owner key, workers hiring details and other necessary store management functions. The system allows the manager to entry the imported product details and the employees to bill the purchased item. Customers can purchase and bill their selected items in the store. The employees has the option of either bill the purchased item or deleting the purchased items from the customer's shopping list. The system is hence useful in the point where bar code scanners are no longer required during the billing activity.

1.2 Problem Statement

Customers always spend a lot of their time visiting grocery websites to check the prices of each product and purchase the order. But the problem is the item wont be sent immediately and sometimes damage may also come, and a huge problem in replacement Also, many smaller stores suffer from not being able to record their availability to customers needs.

Hence there was a need for more efficient system to route product details to local owners with availability while saving valuable time that could be inverted in other community services.

1.3 Proposed Solution

Provision store application is a complete admin based application software. This software is proposed for customers where immediate items should be available in there near by stores. In online application , it may fail to deliver huge items in one day, and as soon as possible. This near by store helps customers to get near by items immediately, and for customers priority we give a billing ,points and item details too.

- The use of automated provision store management system in place of papers and scanners.
- Obtaining the sales report which allows the admin to keep track of various store processes with ease.
- This reduces the time consumption in assessing of data.
- The system keeps track on products ,thus preventing human errors.
- It generates and discount deduction and billing report automatically.
- One system operator will be enough for deploying and maintaining data thus reduces the number of workers in the store.

2. Analysis and System Requirements

2.1 Analysis Survey

- The feasibility study is a 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.
- Study of requirement analysis is done through different feasibility studies.
- Feasibility studies are undertaken whenever a requirement to improve the existing system or designing new system is necessary. Feasibility study helps to meet user requirements.
- It enables us to determine the potential of existing systems and helps in improving the existing system. It helps to develop a technically feasible system. It helps to know what should be embedded in the new system and helps develop a cost-effective system.

The project concept is feasible because of the following:

1)Technical Feasibility:

Technical feasibility is the study of hardware and software requirements on which the application is to be used upon. It is important to consider the budget of the system and overall cost. This system is technically feasible as it has been developed with the help of available technology. The proposed system requires Net Beans and backend tool as MySQL for storing and maintain the database. No expert man power is required for this purpose.

2)Economic Feasibility:

This area is concerned with cost for the development and implementation of the system. We have looked into the maintenance of the system post development. So, the actual cost of the system is important before designing a new system which was perfectly estimated by us. All this is studied in economic feasibility study. The designed system will provide tangible as well as intangible benefits to the organization.

3)Operation Feasibility:

Automation makes lives easy. The proposed system is very friendly and the user is easily able to interact with the system. Therefore, the users will readily accept the system. Data entry and making queries can be done easily.

2.2 System Requirements

Definition of Software:

Computer software or just software is any set of machine-readable instructions that directs a computer's processor to perform specific operations. The term is used to construct with computer hardware, the physical objects (processor and related devices) that carry out the instructions. Computer hardware and software require each other and neither can be realistically used without the other.

Software Specification:

Operating System: Windows 10

Front End: Java Net Beans

Rear End: MYSQL

Definition of Hardware:

Computer hardware is the collection of physical elements that constitute a computer system. Computer hardware refers to the physical components of a computer such as the monitor, mouse, keyboard, computer data storage, hard drive disk, system unit (graphic cards, memory, motherboard and chips) etc. all of which are physical objects that can be touched. In contrast, software is instructions that can be stored and run by hardware.

Hardware Specification:

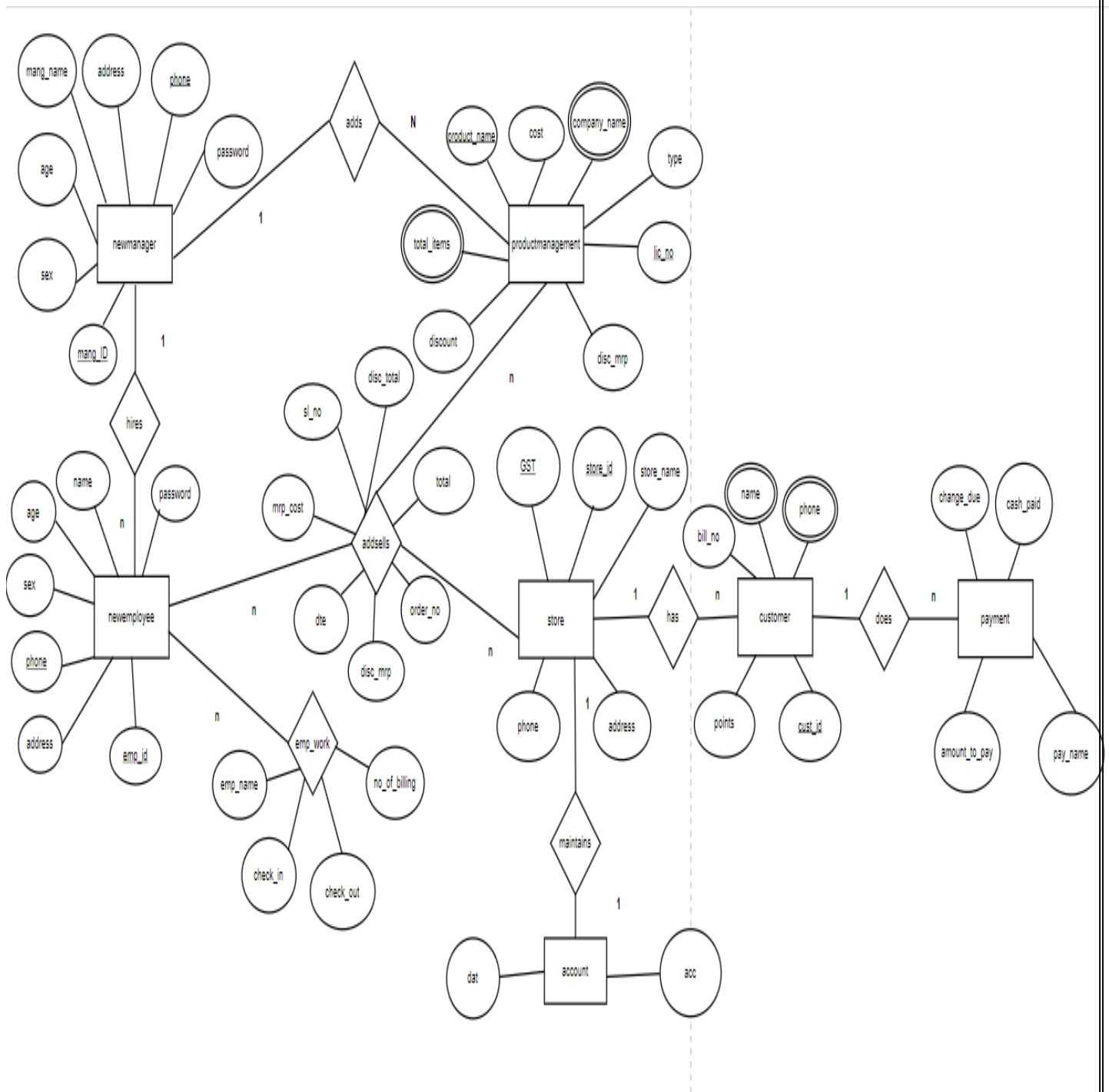
<u>Processor</u>	:	CORE i3 7 th Gen
<u>Processor speed</u>	:	2.0 Ghr
<u>RAM</u>	:	4GB RAM
<u>Storage space</u>	:	1TB
<u>Resolution</u>	:	1024*768 or 1336*768 or 1280*1024 2.2

3 System Design & Modelling

3.1 Preliminary Design

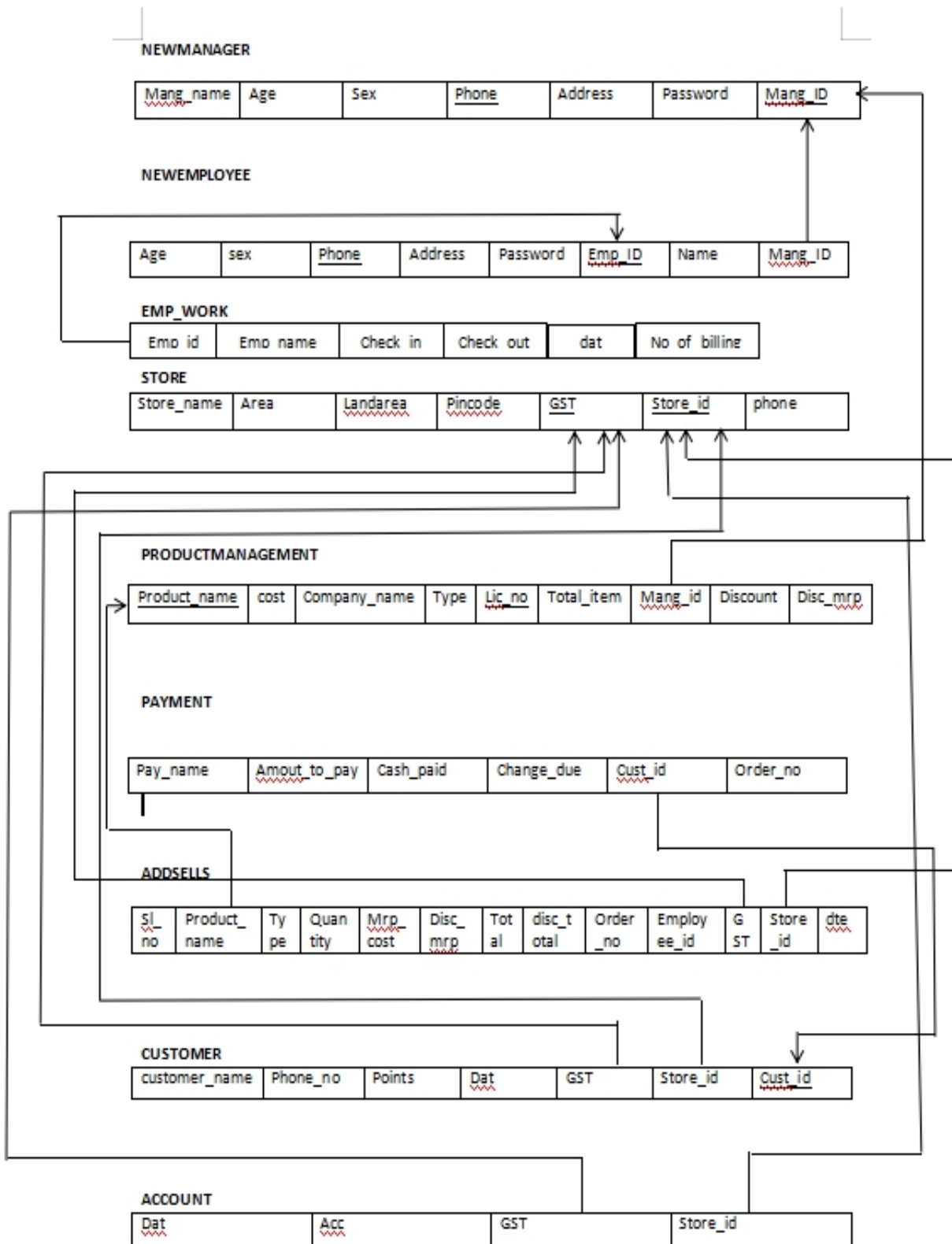
3.1.1 ENTITY RELATIONSHIP DIAGRAM

An *entity relationship model*, also called an *entity-relationship(ER) diagram*, is a graphical representation of entities and their relationships to each other.



3.1.2 Schema Diagram

A database **schema** defines its entities and the relationship among them. It contains a description detail of the database, which can be depicted by means of **schema diagrams**. Figure 3.2: SCHEMA DIAGRAM



3.2 Normalization

First normal form: each table cell should contain a single value.

Each records needs to be unique

1NF

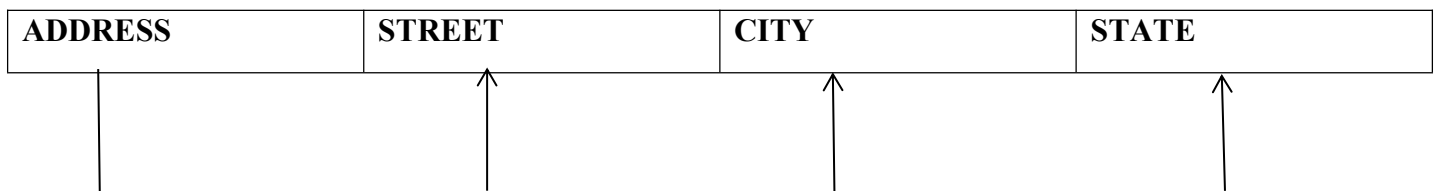
<u>Emp_id</u>	EMP_NAME	AGE	ADDRESS	PHONE_NO	SEX	PASSWORD
---------------	----------	-----	---------	----------	-----	----------

- Second normal form: a table is said to be in 2NF if both the following conditions holds

-table is in 1NF

-No non-prime attribute is dependent on the proper subset of any candidate key of table

2NF



- Third normal form(3NF)

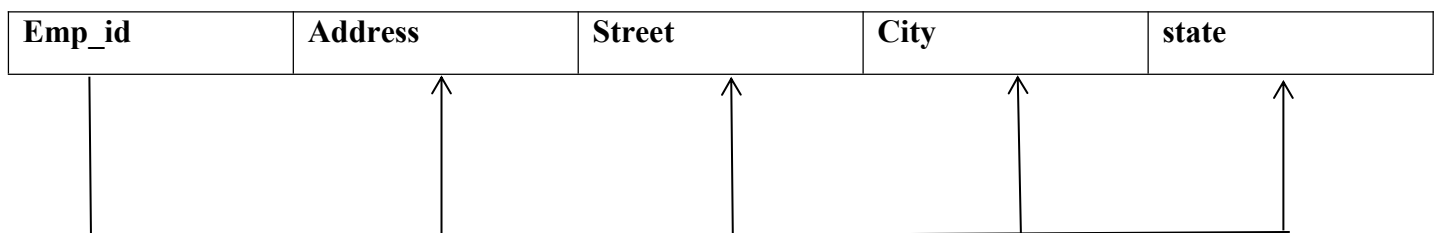
A table design is said to be in 3NF if both the following conditions hold

-Table must be in 2NF

-Transitive function dependency of non-prime attribute on may super key should be removed

-An attribute that is not part of any candidate key is known as non-prime attribute.

3NF



4.Implementation

4.1 MODULE 1 DESCRIPTION(FRONT END)

A Basic introduction to Java

Java is a simple and yet powerful object-oriented programming language and it is in many respects similar to C++. Java originated at Sun Microsystems, Inc. in 1991.

Java was designed with a concept of 'write once and run everywhere'. Java Virtual Machine plays the central role in this concept. The JVM is the environment in which Java programs execute. It is a software that is implemented on top of real hardware and operating system. When the source code (.java files) is compiled, it is translated into byte codes and then placed into (.class) files. The JVM executes these bytecodes. So, Java byte codes can be thought of as the machine language of the JVM. A JVM can either interpret the bytecode one instruction at a time or the bytecode can be compiled further for the real microprocessor using what is called a just-in-time compiler. The JVM must be implemented on a particular platform before compiled programs can run on that platform.

Types of Java Applications:

- **Web Application** - Java is used to create server-side web applications. Currently, Servlet, JSP, Struts, JSF, etc. technologies are used.
- **Standalone Application** - It is also known as the desktop application or window-based application. An application that we need to install on every machine or server such as media player, antivirus, etc. AWT and Swing are used in java for creating standalone applications.
- **Enterprise Application** - An application that is distributed in nature, such as banking applications, etc. It has the advantage of the high-level security, load balancing, and clustering. In Java, EJB is used for creating enterprise applications.
- **Mobile Application** - Java is used to create application software for mobile devices. Currently, Java ME is used for building applications for small devices, and also Java is a programming language for Google Android application development.

Features of Java:

- **Object-Oriented** - Java supports the features of object-oriented programming. Its object model is simple and easy to expand.
- **Platform independent** - C and C++ are platform dependency languages hence the application programs written in one Operating system cannot run in any other Operating system, but in platform independence language like Java application programs written in one Operating system can able to run on any Operating system.
- **Simple** - Java has included many features of C / C ++, which makes it easy to understand.
- **Secure** - Java provides a wide range of protection from viruses and malicious programs. It ensures that there will be no damage and no security will be broken.
- **Portable** - Java provides us the concept of portability. Running the same program with Java on different platforms is possible.
- **Robust** - During the development of the program, it helps us to find possible mistakes as soon as possible.
- **Multi-threaded** - The multithreading programming feature in Java allows you to write a program that performs several different tasks simultaneously.
- **Distributed** - Java is designed for distributed Internet environments as it manages the TCP/IP protocol.

4.2 MODULE 2 DESCRIPTION(BACK END)

A Basic introduction to Microsoft Access

Ms Access is a database management tool that enables one to have good command of data collected. The programme enables one to retrieve, sort, summarize and report results speedily and effectively. It can combine data from various files through creating relationships, and can make data entry more efficient and accurate.

Microsoft Access (MS Access) enables one to manage all important information from a single database file. Within the file, one can use:

- Tables to store your data.
- Queries to find and retrieve specific data of interest.
- Forms to view, add, and update data in tables.
- Reports to analyse or print data in a specific layout.
- Data access pages to view or update, the data.

In MS Access, data is stored once in one table, but can be viewed from multiple locations. When the data is updated in a Table, Query or Form, it is automatically updated everywhere it appears.

Establishment of Ms Access database

All Ms Access databases files are saved with extension **.mdb**

A database should have a separate table for every major subject, such as *pedigree records*, *Production data* or *Treatment information*.

Data should not be duplicated in multiple tables.

Microsoft Access provides three methods to create a database

- **Database Wizard** (though easy, the wizard offers limited options to customize the database)
- Using a **template** (This method works best if one can find and use a template that closely matches the specific requirements)
- **Creating a database directly** (This is the most flexible method, but it requires one to define each database element separately).

4.2.1 Trigger:

TRIGGER specifies an event, a condition and an action. It specifies type of the action to be taken when certain events occur and when certain condition are satisfied.

1. Start provision store database system.
2. Create trigger trigger_name for table product management with action to be taken before update.

3. Update the discount for the mentioned item in the where clause of the respected product's mrp price.
4. stop

4.2 Pseudocode

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

import java.awt.Color;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author cs_loneranger_
 */
public class LOGIN_PAGE1 extends javax.swing.JFrame {
    Connection connect;
    Statement stmt;
    ResultSet rs;

    /**
     * Creates new form LOGIN_PAGE1

```

```

*/

public LOGIN_PAGE1() {

    initComponents();

    Connectivity();

    jPanel4.setBackground(new Color(0,0,0,50));

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */

public void Connectivity()
{
    try{
        connect=DriverManager.getConnection("jdbc:mysql://localhost:3306/provisionstore","root","tiger");
        stmt=connect.createStatement();
    }
    catch(SQLException e){
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
    jPanel4.setVisible(false);
}

@SuppressWarnings("unchecked")
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:
    LOGIN_PAGE lp=new LOGIN_PAGE();
    lp.setVisible(true);
    dispose();
}

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

    jPanel4.setVisible(true);

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

```

```

// TODO add your handling code here:

new owner(null,true).show()

}

private void jPasswordField1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

try{

String sql1="select * from newmanager where mang_ID='"+jTextField1.getText()+" and
password='"+jPasswordField1.getText()+"''";

rs=stmt.executeQuery(sql1);

boolean s=rs.next();

if(s){

store_page sp=new store_page();

sp.setVisible(true);

jButton1.setBackground(Color.yellow);

sp.connect(rs.getString("mang_name"),rs.getString("mang_ID"));

dispose();

jTextField1.setText("");

jPasswordField1.setText("");

}

else{

JOptionPane.showMessageDialog(this, "not found\n create a new ID");

jTextField1.setText("");

jPasswordField1.setText("");

}

}

catch(SQLException s){

JOptionPane.showMessageDialog(this, "enter create a new employee");

jTextField1.setText("");

jPasswordField1.setText("");

}

}

private void jLabel7MouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

jLabel7.setForeground(Color.yellow);

```



```

}

private void jLabel7MouseExited(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:

jLabel7.setForeground(Color.black);
}

private void jLabel7MouseClicked(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:

jPanel4.setVisible(false);
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

try{

String sql1="select * from newemployee where emp_ID='"+jTextField1.getText()+" and
password='"+jPasswordField1.getText()+"'";

rs=stmt.executeQuery(sql1);

boolean s=rs.next();

if(s){

add_sells sp=new add_sells();

sp.setVisible(true);

jButton1.setBackground(Color.yellow);

DateTimeFormatter dtf2=DateTimeFormatter.ofPattern("HH:mm:ss");

DateTimeFormatter dtf=DateTimeFormatter.ofPattern("yyyy-MM-dd");

DateTimeFormatter dtf1=DateTimeFormatter.ofPattern("yyyyMMddHHmmss");

LocalDateTime now=LocalDateTime.now();

sp.empis(rs.getString("emp_ID"),
"abc"+dtf1.format(now),dtf.format(now),dtf2.format(now),rs.getString("name"));

dispose();

jTextField1.setText("");

jPasswordField1.setText("");

}

else{

JOptionPane.showMessageDialog(this, "not found\n create a new ID");

jTextField1.setText("");

jPasswordField1.setText("");

```

```

    }
    }

    catch(SQLException s){
        JOptionPane.showMessageDialog(this, "enter create a new employee");
        jTextField1.setText("");
        jPasswordField1.setText("");
    }
}

private void jPasswordField1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        String sql1="select * from newemployee where emp_ID='"+jTextField1.getText()+"' and
password='"+jPasswordField1.getText()+"'";
        rs=stmt.executeQuery(sql1);
        boolean s=rs.next();
        if(s){
            add_sells sp=new add_sells();
            sp.setVisible(true);
            jButton1.setBackground(Color.yellow);
            DateTimeFormatter dtf2=DateTimeFormatter.ofPattern("HH:mm:ss");
            DateTimeFormatter dtf=DateTimeFormatter.ofPattern("yyyy-MM-dd");
            DateTimeFormatter dtf1=DateTimeFormatter.ofPattern("yyyyMMddHHmmss");
            LocalDateTime now=LocalDateTime.now();
            System.out.println(dtf.format(now));
            sp.empis(rs.getString("emp_ID"),
"abc"+dtf1.format(now),dtf.format(now),dtf2.format(now),rs.getString("name"));
            dispose();
            jTextField1.setText("");
            jPasswordField1.setText("");
        }
        else{
            JOptionPane.showMessageDialog(this, "not found\n create a new ID");
            jTextField1.setText("");
            jPasswordField1.setText("");
        }
    }
}

```

```

    }

    catch(SQLException s){

    JOptionPane.showMessageDialog(this, "enter create a new employee");

    jTextField1.setText("");

    jPasswordField1.setText("");

    }

    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

jButton2.setBackground(Color.yellow);

LOGIN_PAGE1 ep=new LOGIN_PAGE1();

ep.setVisible(true);

dispose();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

String gender;


String empname=jTextField1.getText();

String age=jTextField2.getText();

if(jRadioButton1.isSelected()){

gender=jRadioButton1.getText();

}

else if(jRadioButton2.isSelected()){

gender=jRadioButton2.getText();

}

else{

gender=jRadioButton3.getText();

System.out.println();

}

System.out.println(gender);

String phone_number=jTextField4.getText();

String address=jTextField5.getText();

```

```

String emp_id=jTextField6.getText();
String pass=jPasswordField1.getText();

try{
if(jPasswordField1.getText().trim().isEmpty()){
JOptionPane.showMessageDialog(this, "enter the correct password");
jPasswordField1.grabFocus();
return;
}
else{

String sql="insert into newemployee
values("+age+", '"+gender+"', '"+phone_number+"', '"+address+"', '"+pass+"', '"+emp_id+"', '"+empname+"', '"+
name+"')";

int n= stmt.executeUpdate(sql);//executeUpdate saves the data into the table
if(n==1){
JOptionPane.showMessageDialog(this, n + " records saved successfully");
jTextField1.setText("");
jTextField2.setText("");
jLabel12.setVisible(false);
jTextField4.setText("");
jTextField5.setText("");
jTextField6.setText("");
jPasswordField1.setText("");
}
}

}

catch(SQLException s){
JOptionPane.showMessageDialog(this, "employee ID:"+jTextField6.getText()+"\n already exist\n OR enter
the correct age");
}
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

```

```

//String name=jTextField1.getText();
//String age=jTextField2.getText();
//String sex=jTextField3.getText();
String emp_ID=jTextField6.getText();
//String address=jTextField5.getText();

try{
String sql="delete from newemployee where emp_ID='"+emp_ID+"'";
int n= stmt.executeUpdate(sql);//executeUpdate saves the data into the table
System.out.println(n);
if(n==1){
JOptionPane.showMessageDialog(this, n + " records deleted successfully");
jTextField1.setText("");
jTextField2.setText("");
//jTextField3.setText("");
jTextField4.setText("");
jTextField5.setText("");
jTextField6.setText("");
jPasswordField1.setText("");
}
else{
JOptionPane.showMessageDialog(this, "employee ID:"+jTextField6.getText()+"\n not found");
}
}

catch(SQLException s){
JOptionPane.showConfirmDialog(this,"employee ID:"+jTextField6.getText()+"\n not found");
}
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here
new employee_table(null,true).show();
}

private void jToggleButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

```

```

try{
    if(jToggleButton1.isSelected()){
        if(jTextField6.getText().isEmpty()){
            JOptionPane.showMessageDialog(this, "enter the employee id!!");
            jToggleButton1.setSelected(false);
        }
        else{String sql1="select * from newemployee where emp_ID='"+jTextField6.getText()+"'";
            rs=stmt.executeQuery(sql1);
            if(rs.next()){
                jTextField1.setText(rs.getString("name"));
                jTextField2.setText(rs.getString("age"));
                jTextField4.setText(rs.getString("phone"));
                jTextField5.setText(rs.getString("address"));
                jPasswordField1.setText(rs.getString("password"));
                jTextField6.setText(rs.getString("emp_ID"));
                jToggleButton1.setText("UPDATE");
            }
            else{
                JOptionPane.showMessageDialog(this, "ID not found!!");
            }
        }
    }
    else{
        if(jRadioButton1.isSelected()){
            String sql2="update newemployee set
            address='"+jTextField5.getText()+"',mang_ID='"+name+"',name='"+jTextField1.getText()+"',age="+Integer.
            parseInt(jTextField2.getText())+",sex='"+jRadioButton1.getText()+"',phone='"+jTextField4.getText()+"',pas
            sword='"+jPasswordField1.getText()+"' where emp_ID='"+jTextField6.getText()+"'";

            n=stmt.executeUpdate(sql2);
        }
        else if(jRadioButton2.isSelected()){
            String sql2="update newemployee set
            address='"+jTextField5.getText()+"',mang_ID='"+name+"',name='"+jTextField1.getText()+"',age="+Integer.
            parseInt(jTextField2.getText())+",sex='"+jRadioButton2.getText()+"',phone='"+jTextField4.getText()+"',pas
            sword='"+jPasswordField1.getText()+"'where emp_ID='"+jTextField6.getText()+"'";

            n=stmt.executeUpdate(sql2);
        }
    }
}

```

```

    }

    else{

        String sql2="update newemployee set
address='"+jTextField5.getText()+"',mang_ID='"+name+"',name='"+jTextField1.getText()+"',age="+Integer.
parseInt(jTextField2.getText()+"',sex='"+jRadioButton3.getText()+"',phone='"+jTextField4.getText()+"',pas
sword='"+jPasswordField1.getText()+"' where emp_ID='"+jTextField6.getText()+"'";

        n=stmt.executeUpdate(sql2);

    }

    if(n==1){

        JOptionPane.showMessageDialog(this,"DETAILS UPDATED SUCCESSFULLY");

        jTextField1.setText("");
        jTextField2.setText("");
        jTextField4.setText("");
        jTextField5.setText("");
        jTextField6.setText("");
        jPasswordField1.setText("");
        jToggleButton1.setText("MODIFY");

    }

    }

    }

    catch(SQLException s){

    }

    } private void jTextField1KeyPressed(java.awt.event.KeyEvent evt) {

// TODO add your handling code here:

boolean mat= Pattern.matches("[a-z]*", jTextField1.getText());

if(mat){

jLabel9.setVisible(false);

jtext1=mat;

}

else{

jLabel9.setVisible(true);

jtext1=mat;

}

}

private void jTextField1KeyReleased(java.awt.event.KeyEvent evt) {

```

```

// TODO add your handling code here:

boolean mat= Pattern.matches("[a-z]*", jTextField1.getText());

if(mat){

jLabel9.setVisible(false);

jtext1=mat;

}

else{

jLabel9.setVisible(true);

jtext1=mat;

}

}

private void jTextField1KeyTyped(java.awt.event.KeyEvent evt) {

// TODO add your handling code here:

boolean mat= Pattern.matches("[a-z]*", jTextField1.getText());

if(mat){

jLabel9.setVisible(false);

jtext1=mat;

}

else{

jLabel9.setVisible(true);

jtext1=mat;

}

} private void jTextField4KeyPressed(java.awt.event.KeyEvent evt) {

// TODO add your handling code here:

boolean phone1=Pattern.matches("[6789][0-9]{9}",jTextField4.getText().trim());

if(phone1){

jLabel10.setVisible(false);

}

else{

jLabel10.setVisible(true);

}

} private void jTextField4KeyReleased(java.awt.event.KeyEvent evt) {

// TODO add your handling code here:

boolean phone1=Pattern.matches("[6789][0-9]{9}",jTextField4.getText().trim());

```



```

if(phone1){
jLabel10.setVisible(false);
}
else{
jLabel10.setVisible(true);
}
}

private void jTextField4KeyTyped(java.awt.event.KeyEvent evt) {
// TODO add your handling code here:
boolean phone1=Pattern.matches("[6789][0-9]{9}",jTextField4.getText().trim());
if(phone1){
jLabel10.setVisible(false);
}
else{
jLabel10.setVisible(true);
}
} private void jTextField6KeyPressed(java.awt.event.KeyEvent evt) {
// TODO add your handling code here:
boolean id=Pattern.matches("([a-z A-Z]*)([0-9]*)", jTextField6.getText());
if(id){
jLabel11.setVisible(false);
}
else{
jLabel11.setVisible(true);
}
}

private void jTextField6KeyReleased(java.awt.event.KeyEvent evt) {
// TODO add your handling code here:
boolean id=Pattern.matches("([a-z A-Z]*)([0-9]*)", jTextField6.getText());
if(id){
jLabel11.setVisible(false);
}
else{
jLabel11.setVisible(true);
}
}

```

```

}

}

private void jTextField6KeyTyped(java.awt.event.KeyEvent evt) {
// TODO add your handling code here:

boolean id=Pattern.matches("([a-z A-Z]*)([0-9]*)", jTextField6.getText());

if(id){
jLabel11.setVisible(false);
}

else{
jLabel11.setVisible(true);
}

}

private void jPasswordField1KeyPressed(java.awt.event.KeyEvent evt) {
// TODO add your handling code here:

boolean bad=Pattern.matches("[a-z A-Z]*|[0-9]*", jPasswordField1.getText());

if(bad){
jLabel12.setVisible(true);jLabel12.setText("atleast 1 special character");
}

else{
jLabel12.setVisible(true);jLabel12.setText("strong");
}

}

private void jPasswordField1KeyReleased(java.awt.event.KeyEvent evt) {
// TODO add your handling code here:

boolean bad=Pattern.matches("[a-z A-Z]*|[0-9]*", jPasswordField1.getText());

if(bad){
jLabel12.setVisible(true);jLabel12.setText("atleast 1 special character");
}

else{
jLabel12.setVisible(true);jLabel12.setText("strong");
}

}

private void jPasswordField1KeyTyped(java.awt.event.KeyEvent evt) {
// TODO add your handling code here:

boolean bad=Pattern.matches("[a-z A-Z]*|[0-9]*", jPasswordField1.getText());

```

```
if(bad){
jLabel12.setVisible(true);jLabel12.setText("atleast 1 special character");
}
else{
jLabel12.setVisible(true);jLabel12.setText("strong");
}
}

private void jLabel13MouseClicked(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
jPanel2.setVisible(true);
}

private void jLabel13MouseEntered(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
jLabel13.setForeground(Color.yellow);
}

private void jLabel13MouseExited(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
jLabel13.setForeground(Color.white);
}

private void jLabel14MouseClicked(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
jPanel2.setVisible(false);
}

private void jLabel14MouseEntered(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
jLabel14.setForeground(Color.yellow);
}

private void jLabel14MouseExited(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
jLabel14.setForeground(Color.black);
}

private void jLabel15MouseClicked(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
store_page sp=new store_page();
```

```

sp.setVisible(true);
dispose();
}

private void jLabel15MouseEntered(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
jLabel15.setForeground(Color.yellow);
}

private void jLabel15MouseExited(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
jLabel15.setForeground(Color.white);
}

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(LOGIN_PAGE1.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(LOGIN_PAGE1.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    } catch (IllegalAccessException ex) {

```

```
java.util.logging.Logger.getLogger(LOGIN_PAGE1.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(LOGIN_PAGE1.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 LOGIN_PAGE1().setVisible(true);
    }
});
}
```

5. Testing

SOFTWARE TESTING

Software Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding, Testing presents an interesting anomaly for the software engineer.

Testing Objectives include:

- Testing is a process of executing a program with intent of finding an error.
- A good test case is one that has a probability of finding a yet undiscovered error.
- A successful test is one that uncovers an undiscovered error.

Testing Principles:

- All tests should be traceable to end user requirements Test should be planned long before testing begins
- Testing should begin on a small scale and progress towards testing in large Exhaustive testing is not possible
- To be most effective testing should be conducted by a independent third party.

5.1 Introduction to testing

5.1.1 Unit Testing:

Unit testing focuses verification effort on the smallest unit of software design that is the module. Using procedural design description as a guide, important control paths are tested to uncover errors within the boundaries of the module. The unit test is normally white box testing oriented and the step can be conducted in parallel for multiple modules.

5.1.2 Validation Testing:

At the end of integration testing software is completely assembled as a package. Validation testing is the next stage, which can be defined as successful when the software functions in the manner reasonably expected by the customer. Reasonably expectations are those defined in the software requirements specifications. Information contained in those sections form a basis for validation testing approach.

5.1.3 System testing:

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer - based system. Although each test has a different purpose, all work to verify that all system elements have been properly integrated to perform allocated functions.

5.2 Test Cases

Serial No	Test Objective	Expected Output	Actual Output	Status
1	To check whether the program runs or not	A webpage is opened	A webpage is opened	Pass
2	Register to the page	A pop up window opens	A pop up window opens	Pass
3	Login to the page	A corresponding window appears	A corresponding window appears	Pass
4	Add product details	A corresponding window appears	A corresponding window appears	Pass
5	The product details is created	A pop up window opens	A pop up window opens	Pass

6. Conclusion

The project entitled “**PROVISION STORE MANAGEMENT SYSTEM**” was completed on time with total satisfaction after testing with possible sample data. The performance was found to be efficient and error free. This is a user-friendly packaged application which is very easy to access and understand. Anyone with Knowledge of Computers will find it very easy to use this software and perform various operations on it.

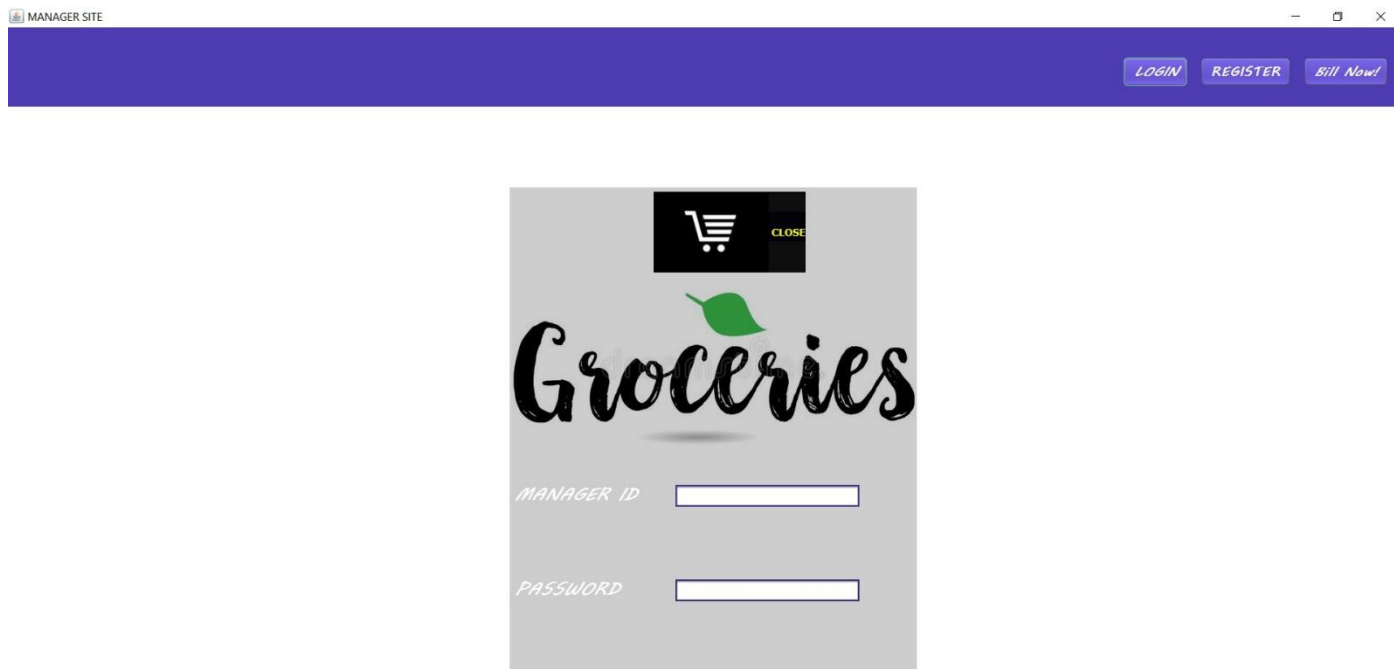
In this project, first an attempt has been made to find the need of the system. To fulfil the needs, a detailed study had been conducted to find the various requirements of the system. This particular system has been designed in an attractive manner, so that even a user with minimum knowledge can be able to operate the system easily.

This software combines the best of both the world i.e.; programming language (JAVA), and database (MYSQL) providing easy accessibility and security. It was developed to benefit the organizations and the customers. Finally the system was tested with real data and everything worked successfully. Thus the system has fulfilled all the objectives identified and is able to replace the existing system.

7. APPENDIX A:SNAPSHOTS

1) Manager site:

The manager/owner login page



MANAGER SITE

LOGIN REGISTER Bill Now!

CLOSE

Groceries

MANAGER ID

PASSWORD

2) Cashier site:

The new employee uses this software page to continue the billing actions



ADMIN

GROCERY STORES

EMPLOYEE ID

PASSWORD

LOGIN

© CanStockPhoto.com - csp52904870

3) Billing site:

The employee of the shop enters the items purchased by the customers in this page.

BILLING SITE

LOGOUT
Sales Dashboard
ITEM DETAILS
CUSTOMERS POINTS

Sales Dashboard

Sales Product
lays
Quantity
1
Add to cart

Product Name	Type	Cost Per Unit	Quantity	Total
Jim Jam	Bakery	10.0	1.0	0.0
Good Day	Bakery	10.0	1.0	0.0
Kurkure Green Chilli	Bakery	10.0	1.0	0.0
Tomato	Produce	20.0	1.5	10.0
Lays	Bakery	10.0	1.0	0.0

Customer Name
Calvin
Phone Number
9888888888

CASH
70
CARD
NET BANKING
POINTS

CART
No of Items: 5
Total Cost: 67.0
Save Order
POINTS

4) product management site:

The owner/manager of the store can enter the item details using this page.

PRODUCT MANAGEMENT

SEARCH
Product Management
*0*SETTING

Product Management

Product Name
Cost Per Unit
Company Name
Type
--SELECT--
ITEM ID
TOTAL ITEM
SAVE

CLOSE
DELETE
UPDATE
Discount

ID	Product Name	MRP	Company	Type	Total Items	DISCOUNT	DISCOUNT MRP	ADDED BY
britan789oop	good day	10.0	britania	Bakery	39.0	0.0	0.0	lakshmi001
capcsi	capsicum	15.0	null	Produce	20.0	0.0	0.0	madhu001
cucum77op	cucumber	6.0	null	Produce	30.0	0.0	0.0	madhu001
kurkure789oop	kurkure green chilli	10.0	kurkure	Bakery	74.0	0.0	0.0	sand789
lays44oop	lays yellow	10.0	lays	Bakery	39.0	0.0	0.0	sand789
lays789oop	lays	10.0	lays	Bakery	34.0	0.0	0.0	lakshmi001
onions	onions	48.0	null	Produce	298.0	0.0	0.0	sand789
tomato789oop	tomato	20.0	tomato	Produce	443.0	10.0	18.0	lakshmi001

5) bill sample:

An automatic bill is generated showing the amount, total no of items and change due to be given.

BILL

GROCERY STORE
KORAMANGALA
#75 80 FEET ROAD 1ST BLOCK KORAMANGALA
Phone:8025525489
GSTIN:29AAACL4299H1Z4
Invoice/Bill of Supply

BILL NO:pic20210106142648
YOU HAVE SAVED RS: 7.0 OUR PRICE IS INCLUSIVE OF ALL GST

ITEMS ARE LESSER THEN THE MRP PRICE

POINTS:0.0

PARTICULARS	TYPE	MRP	RATE	QTY	AMOUNT
lays yellow	Bakery	10.0	0.0	1.0	10.0
tomato	Produce	20.0	18.0	3.5	63.0

No Of Items:2
TOTAL:73.0
PAYMENT METHOD:CASH
CASH PAID:100.0
CHANGE DUE:27.0

6) new employee site:

This software page is used by the manager/owner who hires new employees and to enter their details

NEW EMPLOYEE SITE

<-

*O*SETTINGS

NAME

AGE

GENDER ☐ MALE ☐ FEMALE ☐ OTHERS

MOBILE NUMBER

ADDRESS

CREATE ID

NEW PASSWORD

SAVE

8. References

BOOKS:

Database Management System (5th edition) -Ramakrishna and Gehrke.

WEBSITE:

www.google.co.in

www.w3schools.com

www.stackoverflow.com

www.slideshare.net

www.indianrail.gov.in

www.irctc.gov.in

www.wikipedia.org.in

www.schoolprojects.com

www.m.etrain.info.com

www.indiamike.com