Chapter 1

INTRODUCTION

1.1 OVERVIEW OF THE SYSTEM

An automated module for the Post Office will help in automating functions of the administration department. It helps in reducing the time spent in record keeping and the work can be carried out effectively. The searching of records in future will also become easy. The redundancy in the data due to manual data will also be tackled. The Post office will be able to access the personal information of each customer easily. The administration department will also be able to add new branches of post office, staffs, customers, set the rates for the courier or any other parcel based on distance and type of delivery.

This document describes the design for the Post Office will help in automating functions of the administration department. It helps in reducing the time spent in record keeping and the work can be carried out effectively. The searching of records in future will also become easy. The redundancy in the data due to manual data will also be tackled. The Post office will be able to access the personal information of each customer easily. The administration department will also be able to add new branches of post office, staffs, set the rates for the courier or any other parcel based on distance and type of delivery. The e-Post Office is expanded permanently through new products and services in order to offer a product portfolio corresponding to the market.

Private customer and business customers can order the selected products of the postal service online quickly and comfortably. Besides this, the e-Services offer new flexibility through e-Packet, the PICKUP order for packages over the Internet as well as the online forwarding order and storage order. For the case of the absence or the move, one can let delegate here the after shipment of the postal service at another address or store the letter shipments. It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in JAVA and some extent Windows Application and Oracle 10g, but also about all handling procedure related with **"E-POST OFFICE MANAGEMENT SYSTEM".** It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

1.2 NEED OF E-POST OFFICE MANAGEMENT SYSTEM:

The Post office will be able to access the personal information of each customer easily. The administration department will also be able to add new branches of post office, staffs, customers, set the rates for the courier or any other parcel based on distance and type of delivery. The manual based and need many efforts and consume enough time. In the existing system, we can cannot check details of the post office and price. It may lead to corruptions in the all process. The existing system does not deals with automatic price calculation.

Disadvantages:

- More human power
- More strength and strain of manual
- Labor needed Repetition of same procedure.
- Low security.
- Data redundancy.
- Difficulty to handle.
- Difficulty to update data.
- Record keeping is difficult.

Hence, to eradicate all these disadvantages we need E-Post Office management system.

1.3 APPLICATIONS, OBJECTIVE AND FEATURES:

- A Single application through which offices at different locations are connected.
- The administrator will be able to view all kinds of details like customer details, delivery details, factors to set charges, list of post offices and their addresses and ids etc… and he will also be able to update, edit, and delete them
- The staff members of the post office will be able to view all the contact details of the customers, rates, delivery details, charges for various factors, update their own profile etc.
- Customer is the user of the system. An administrator of the application is the super user.
- There is no risk of data mismanagement at any level while the project development is under process.

- The system makes the overall project management much easier and flexible. User friendliness is provided in the application with various controls.
- This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the desired information through so much simplicity.
- The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stages of any new creation, data entry or updating so that the user cannot enter the invalid data, which can create problems at later date.
- Sometimes the staff finds in the later stages of using project that he needs to update some of the information that he entered earlier. There are options for him by which he can update the records. This keeps the validity of the data to longer extent.
- User is provided the option of monitoring the records he entered earlier. He can see the desired records with the variety of options provided by him.
- From every part of the project, the user is provided with the links through framing so that he can go from one option of the project to other as per the requirement. This is bound to be simple and very friendly as per the user is concerned. That is, we can sat that the project is user friendly, which is one of the primary concerns of any good project.
- Data storage and retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database.
- Decision making process would be greatly enhanced because of faster processing of information since data collection from information available on computer takes much less time than manual system.
- Easier and faster data transfer through latest technology associated with the computer and communication.
- Through these features, it will increase the efficiency, accuracy and transparency.

Chapter 2

REQUIREMENT SPECIFICATION

2.1 HARDWARE CONFIGURATION:

The section of hardware configuration is an important task related to the software development insufficient random access memory may affect adversely on the speed and efficiency of the entire system. The process should be powerful to handle the entire operations. The hard disk should have sufficient capacity to store the file and application.

- Processor :Pentium Dual core and above
- Processor speed : 1.4 GHz Onwards
- System memory : Minimum 10 Gb recommended
- Cache size : 1024 KB
- RAM : 1 GB(Minimum)
- Hard disk : 100 Gb
- Monitor : SVGA Color 15”
- Mouse : 104 keys US Key Serial, USB or PS/2

2.2 SOFTWARE CONFIGURATION:

A major element in building a system is the section of compatible software since the software in the market is experiencing in geometric progression. Selected software should be acceptable by the firm and one user as well as it should be feasible for the system. This document gives a detailed description of the software requirement specification. The study of requirement specification is focused specially on the functioning of the system.It allow the developer or analyst to understand the system, function to be carried out the performance level to be obtained and corresponding interfaces to be established.

- Front end tool : JAVA with NET BEANS
- Backend : Oracle 10g
- Operating system : Windows 2007/2008/2010

2.3 RESOURCES USED FOR THE PROJECT:

NETBEANS- NetBeans is a software development platform written in Java. The NetBeans Platform allows applications to be developed from a set of modular software components called *modules*. Applications based on the NetBeans Platform, including the NetBeans integrated development environment (IDE), can be extended by third party developers. The NetBeans IDE is primarily intended for development in Java, but also supports other languages, in particular PHP, C/C++and HTML5.

NetBeans is cross-platform and runs on Microsoft Windows, mac OS, Linux, Solaris and other platforms supporting a compatible JVM. The editor supports many languages from Java, C/C++, XML and HTML, to PHP, Groovy, Javadoc, JavaScript and JSP. Because the editor is extensible, you can plug in support for many other languages.

ORACLE 10G- An Oracle **database** is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information. A database server is the key to solving the problems of information management. In general, a **server** reliably manages a large amount of data in a multiuser environment so that many users can concurrently access the same data. All this is accomplished while delivering high performance. A database server also prevents unauthorized access and provides efficient solutions for failure recovery.

Oracle Database is the first database designed for enterprise grid computing, the most flexible and cost effective way to manage information and applications. Enterprise grid computing creates large pools of industry-standard, modular storage and servers. With this architecture, each new system can be rapidly provisioned from the pool of components. There is no need for peak workloads, because capacity can be easily added or reallocated from the resource pools as needed. The database has **logical structures** and **physical structures**. Because the physical and logical structures are separate, the physical storage of data can be managed without affecting the access to logical storage structures.

Chapter 3

SYSTEM DESIGN AND ANALYSIS

3.1 DATA:

The three-schema approach is software engineering concept that enables the database user to separate the user application and physical database. In this architecture, schemas can be defined at 3 levels:

- internal level or internal schema: It describes the physical storage structure of the database, the internal schema uses physical data model and describes the complete details of data storage and access paths of the database
- conceptual schema or conceptual level: describes the structure of the whole database for a community of users it hides the details of the physical storage structures and concentrates on describing the entities, data types, relationships and constraints
- external level or external schema: it includes a number of user views each view describes the part of the database that the database user is interested in and hides the rest of the database from user, implementation data model can be used at this level

This system use windows platform, java net beans as front-end technology and Oracle 10g as backend technology. Thus, E-POST-OFFICE MANAGEMENT SYSTEM is technically feasible. Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as benefit analysis using java net beans and Oracle 10g easily available in internet. The section of hardware configuration is an important task related to the software development insufficient random access memory may affect adversely on the speed and efficiency of the entire system.

The process should be powerful to handle the entire operations. The hard disk should have sufficient capacity to store the file and application. A major element in building a system is the section of compatible software since the software in the market is experiencing in geometric progression. Selected software should be acceptable by the firm and one user as well as it should be feasible for the system

3.2 E-R DIAGRAM: An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects, or concepts relate to each other within a system.

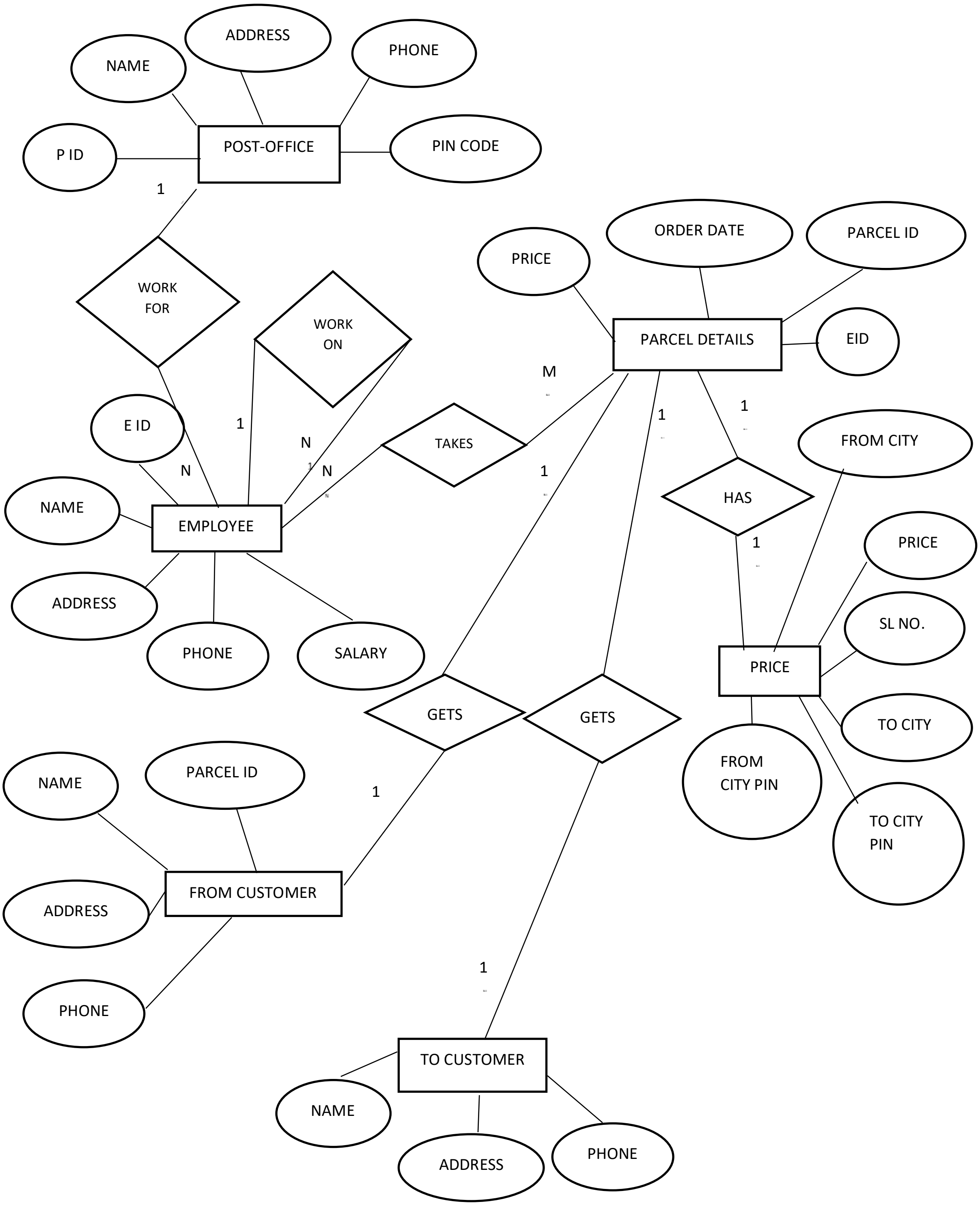


Figure 3.1: E-R Diagram of E post office

3.3 E-R MAPPING:

ENTITY RELATION MAPPING ALGORITHIM:

STEP 1: For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E. Include only the simple component attributes of a composite attribute. Choose one of the key attributes of E as primary key for R.

STEP 2: For each weak entity type W in the ER schema with owner entity type E, create a relation R, and include all simple attributes (or simple components of composite attributes) of was attributes of R.

STEP 3: For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R. Choose one of the relations—S, say—and include as foreign key in S the primary key of T.

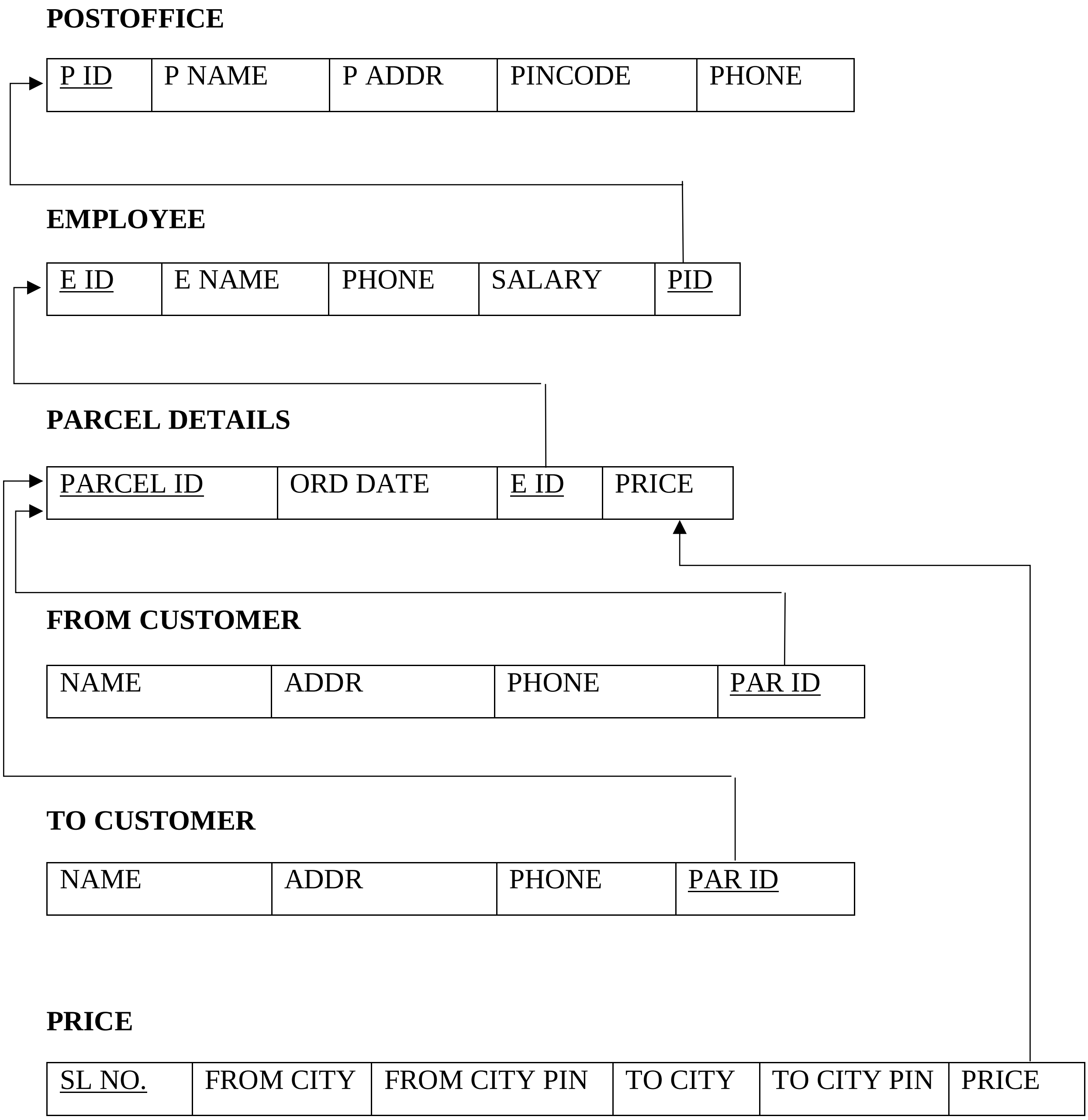
STEP 4: For each regular binary 1: N relationship type R, identify the relation S that represents the participating entity type at the N-side of the relationship type.

STEP 5: For each binary M:N relationship type R, create a new relation S to represent R. Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S.

STEP 6: For each multivalued attribute A, create a new relation R. This relation R will include an attribute corresponding to A, plus the primary key attribute K—as a foreign key in R—of the relation that represents the entity type or relationship type that has A as an attribute.

STEP 7: For each binary relationship type R, where n > 2, create a new relation S to represent R. Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types.

3.4 SCHEMA DIAGRAM: A database schema is the skeleton structure that in which represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.



POSTOFFICE

P ID	P NAME	P ADDR	PINCODE	PHONE

EMPLOYEE

E ID	E NAME	PHONE	SALARY	PID

PARCEL DETAILS

PARCEL ID	ORD DATE	E ID	PRICE

FROM CUSTOMER

NAME	ADDR	PHONE	PAR ID

TO CUSTOMER

NAME	ADDR	PHONE	PAR ID

PRICE

SL NO.	FROM CITY	FROM CITY PIN	TO CITY	TO CITY PIN	PRICE

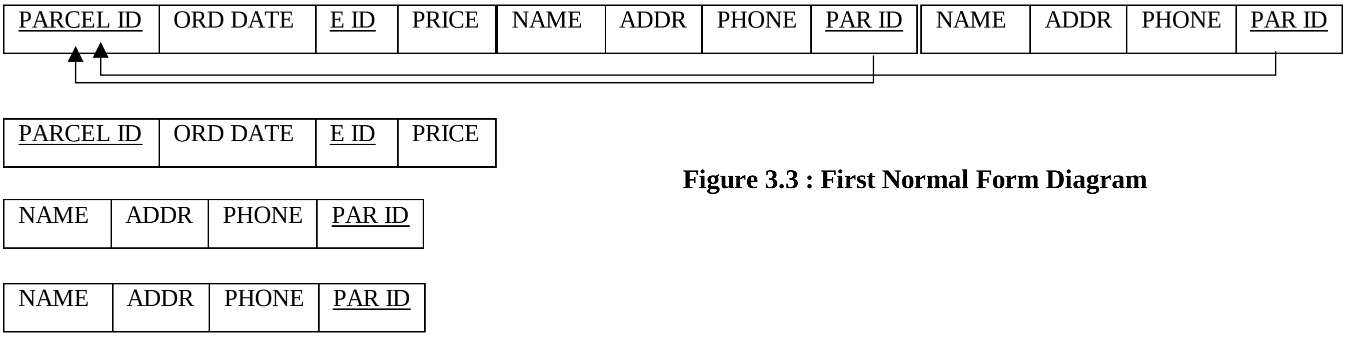
Figure 3.2: Schema Diagram of E Post Office

3.5 NORMALIZATION

Database normalization, or simply normalization, is the process of organizing the columns (attributes) and tables (relations) of a relational database to reduce data redundancy and improve data integrity. Normalization is also the process of simplifying the design of a database so that it achieves the optimal structure composed of atomic elements. Normalization involves arranging attributes in relations based on dependencies between attributes, ensuring that the dependencies are properly enforced by database integrity constraints. Normalization is accomplished by applying some formal rules either by a process of synthesis or decomposition. Synthesis creates a normalized database design based on a known set of dependencies. Decomposition takes an existing (insufficiently normalized) database design and improves it based on the known set of dependencies.

FIRST NORMAL FORM (1 NF):

For a table to be in 1NF you need to ensure that the data is atomic, having no repeating groups. A concatenated pkey characterizes a 1NF table. Atomic data is a form of minimalism for data items. A data item is atomic if only one item is in each cell of a table.



PARCEL ID	ORD DATE	E ID	PRICE	NAME	ADDR	PHONE	PAR ID	NAME	ADDR	PHONE	PAR ID

PARCEL ID	ORD DATE	E ID	PRICE

NAME	ADDR	PHONE	PAR ID

NAME	ADDR	PHONE	PAR ID

Figure 3.3 : First Normal Form Diagram

The I NF says there should be atomic attributes in a table. The above table satisfy this 1 NF.

SECOND NORMAL FORM (2 NF):

A table is said to be in 2NF if both the following conditions hold:

- Table is in 1NF (First normal form) No non-prime attribute is dependent on the proper subset of any candidate key of table.
- An attribute that is not part of any candidate key is known as non-prime attribute. Item is in each cell of a table.

THIRD NORMAL FORM (3 NF):

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

- Table must be in 2NF
- Transitive functional dependency of non-prime attribute on any super key should be removed.

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

Chapter 4

IMPLEMENTATION

4.1 TABLES:

1. POST OFFICE

PID	NOT NULL VARCHAR2(20)
PNAME	VARCHAR2(20)
PADDR	VARCHAR2(20)
PHNO	NUMBER(20)
PINCODE	NUMBER(20)

2. EMPLOYEE

EID	NOT NULL VARCHAR2(20)
ENAME	VARCHAR2(20)
EPHNO	NUMBER(20)
SALARY	NUMBER(10)
PID	VARCHAR2(20)

3. PARCEL_DETAILS

PARCELID	NOT NULL VARCHAR2(20)
ORDDATE	DATE
EID	VARCHAR2(20)
PRICE	NUMBER(20)

4. FCUST

FNAME	VARCHAR2(20)
FADDR	VARCHAR2(50)
PHNO	NUMBER(10)
PARCELID	VARCHAR2(20)

5. TCUST

TNAME	VARCHAR2(20)
TADDR	VARCHAR2(50)
TPHNO	NUMBER(10)
PARCELID	VARCHAR2(20)

6. PRICE

SLNO	NOT NULL NUMBER(10)
FROMCITY	VARCHAR2(20)
FROMPIN	NUMBER(10)
TOCITY	VARCHAR2(20)
TOPIN	NUMBER(10)
PRICE	NUMBER(10)

4.2 QUERIES:

- **Query for searching the price:**

```
SELECT PRICE
FROM PRICE
WHERE FROMPIN=”+FROMPIN+”
AND TOPIN=”+TOPIN+”
```

- **Query for searching the sum from dates:**

```
SELECT SUM(PRICE)
FROM PARCEL_DETAILS
WHERE ORDDATE BETWEEN '"+FROMDATE+"'
 AND '"+TODATE+"'
```

- **Query for calculating total sum:**

```
SELECT SUM(PRICE)
FROM PARCEL_DETAILS
```

- **Query for inserting the values:**

```
1.   INSERT INTO POSTOFFICE
VALUES(' "+PID+" ',' "+NAME+" ',' "+ADDR+" ',"+PHONE+", "+PIN+" )
2.   INSERT INTO PRICE
VALUES( "+SLNO+" ,' "+FCITY+" ',"+FPIN+",' "+TCITY+" ',"+TPIN+",
"+PRICE+" )
3.   INSERT INTO EMPLOYEE
VALUES(' "+EID+" ',' "+NAME+" ', "+PHONE+" , "+SALARY+" ,' "+VALUE+" ')
4.   INSERT INTO PARCEL_DETAILS
VALUES(' "+PARCAL_ID+" ',' "+DATE+" ',' "+EID+" ',"+PRICE+")
5.   INSERT INTO FCUST
VALUES(' "+NAME+" ',' "+ADDR+" ', "+PHONE+" ,' "+PID+" ')
6.   INSERT INTO TCUST VALUES(' "+TCNAME+" ',' "+ADDR+" ',"+PHONE+")
```

- **Query for update:**

```
1.   UPDATE POSTOFFICE SET PHNO="+PHONE+" WHERE PID="+POID+"
2.   UPDATE PRICE SET PRICE="+PRICE+" WHERE SLNO="+SLNO+"
3.   UPDATE EMPLOYEE SET SALARY="+SALARY+" WHERE EID="+EID+"
```

- **Query for deleting:**

```
1.   DELETE FROM POSTOFFICE WHERE PID="+PID+"
2.   DELETE FROM EMPLOYEE WHERE EID="+EID+"
3.   DELETE FROM PRICE WHERE SLNO="+SLNO+"
```

- **Query for displaying:**

1. SELECT * FROM POSTOFFICE.EMPLOYEE WHERE ROWNUM <= 100;
2. SELECT * FROM POSTOFFICE.TCUST WHERE ROWNUM <= 100;
3. SELECT * FROM POSTOFFICE.PRICE WHERE ROWNUM <= 100;
4. SELECT * FROM POSTOFFICE.POSTOFFICE WHERE ROWNUM <= 100;
5. SELECT * FROM POSTOFFICE.PARCEL_DETAIL WHERE ROWNUM<= 100;
6. SELECT * FROM POSTOFFICE.FCUST WHERE ROWNUM <= 100;

4.3 PROCEDURES AND TRIGGERS:

- **Procedure for update:**

Post-Office:

```
DELIMITER @@
CREATE OR REPLACE PROCEDURE POSTOFFICE.PUPDATE2(POID IN VARCHAR,PHONE IN NUMBER) AS
BEGIN
UPDATE POSTOFFICE SET PHNO=PHONE WHERE PID=POID;
END; @@
DELIMITER ;
```

Employee:

```
DELIMITER @@
CREATE OR REPLACE PROCEDURE POSTOFFICE.EMPLOYEE1(EID IN NUMBER,SALARY IN NUMBER) AS
BEGIN
UPDATE EMPLOYEE SET SALARY=SALARY WHERE EID=EID;
END; @@
DELIMITER ;
```

Price:

```
DELIMITER @@
CREATE OR REPLACE PROCEDURE POSTOFFICE.EMPLOYEE1(EID IN
NUMBER,SALARY IN NUMBER) AS
BEGIN
UPDATE EMPLOYEE SET SALARY=SALARY WHERE EID=EID;
END; @@
DELIMITER ;
```

- **Trigger:**

```
CREATE OR REPLACE
TRIGGER EMPLOYEE
INSTEAD OF INSERT ON EMPLOYEE
FOR EACH ROW
BEGIN
DBMS_OUTPUT.PUT_LINE('INSERTING: ' :NEW.NAME);
INSERT INTO EMPLOYEE(EID,ENAME) VALUES (N,:NEW.NAME);
END;
/
```

4.4 FRONT-END SOURCE CODE:

Login form code:

```
if(username.equals("root") && password.equals("root"))
{
        new main().setVisible(true);
        new home().setVisible(false);
        dispose();
}
else
{
        JOptionPane.showMessageDialog(null,"User Name or Password invalid");
}
```

For displaying date automatically :

```
public void date(){
String txtdate = new SimpleDateFormat("d-MMM-yyy").format(new Date());
t3.setText(txtdate);
}
```

For getting random unique number:

```
private void random(){
t1.setText(Integer.toString( ThreadLocalRandom.current().nextInt(501,5000)));
}
```

Retreving data from database

```
private void fillcombo(){
    try{
        Class.forName("oracle.jdbc.OracleDriver");
        con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","postoffice","postoffice");
        String sql="select * from employee";
        pst=con.prepareStatement(sql);
        ResultSet rs=pst.executeQuery();
        while(rs.next()){
            String eid=rs.getString("eid");
            t2.addItem(eid);
        }
    }catch(Exception e){

            JOptionPane.showMessageDialog(null,e);
            }
}
```

Connection for data base :

```
Class.forName("oracle.jdbc.OracleDriver");
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","postoffice","
postoffice");
```

Inserting into database

```
 String value = (String) t2.getSelectedItem();
 String parcal_id =t1.getText().toString();
 String date = t3.getText().toString();
 String eid = value;
 String price = t4.getText().toString();
 try{
    Class.forName("oracle.jdbc.OracleDriver")
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","postoffice","post
office");
    Statement stmt=con.createStatement();
    String query="insert into parcel_details values(' "+parcal_id+" ',' "+date+" ','
"+eid+"”',"+price+")";
    stmt.executeUpdate(query);

    JOptionPane.showMessageDialog(null,"INSERTED SUCCESSFULLY ");
    con.close();
    jPanelSlider1.nextPanel(5,P2,jPanelSlider1.left);
 }
 catch(ClassNotFoundException | SQLException ex)
 {
    JOptionPane.showMessageDialog(null,"ERROR : --  "+ex);
 }
```

Display:

```
try{
         Class.forName("oracle.jdbc.OracleDriver");
         con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","postoffice","postoffice");
         PreparedStatement pst;
         String query ="select * from price";
         pst=con.prepareStatement(query);
         ResultSet rs;
         rs=pst.executeQuery();
         q1.setModel(DbUtils.resultSetToTableModel(rs));
        }
        catch(Exception e){
            JOptionPane.showMessageDialog(null,e);
        }
    }
```

Delete:

```
String slno= t3.getText().toString();


       try{
        Class.forName("oracle.jdbc.OracleDriver");
        con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","postoffice","postoffice");
        String sql="Delete from price where slno="+slno+"";
        pst=con.prepareStatement(sql);
        pst.executeUpdate();
        JOptionPane.showMessageDialog(null,"DELETED SUCCESSFULLY ");
       }catch(Exception e){
       JOptionPane.showMessageDialog(null,e);
       }
```

Calling a procedure:

```
String eid = r1.getText().toString();
        String salary = r2.getText().toString();
        try{
        Class.forName("oracle.jdbc.OracleDriver");
        con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","postoffice","postoffic
e");
        Statement stmt=con.createStatement();
        CallableStatement cs= con.prepareCall("{call price1(?,?)}");
        cs.setString(1,eid);
        cs.setString(2,salary);
        cs.execute();
            JOptionPane.showMessageDialog(null,"UPDATED SUCCESSFULLY ");
            con.close();
        }
        catch(ClassNotFoundException | SQLException ex)
            {
                JOptionPane.showMessageDialog(null,"ERROR : --  "+ex);
            }
```

Chapter 5

TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every convincible fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies, and/or finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner.

5.1 UNIT TESTING:

Unity testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures and operating procedures For unit testing first we adopted the code testing strategy, which examined the logic of program. During the development process, itself all the syntax errors are rooted out.For this developed test case that result in executing every instruction in the program.

Output testing:

After performing validation testing, the next step is output testing of the proposed system. Since the system cannot be useful if it does not procedure the required output. Asking the user about the user about tis required format in which the system is required tests the output displayed or generated by the system under consideration.

GUI Testing:

GUI testing is use to ensure the visual clarity of the system, flexibility of the system, user friendliness of the system. The various component that are to be tested are:

- Relative layout
- Various Buttons

5.2 VALIDATION TESTING:

At the culmination of black box testing, software is completely assembled is a package. Interfacing errors have uncovered and the correct and final states of tests i.e. validation is defined with a simple definition that validation succeeds when the software function in a manner that can be reasonably accepted by the customer.

5.3 SNAPSHOTS:

The first page of the project is the Introducing page where the actual process starts from here by clicking the CUSTOMER and EMPLOYEE button. As shown in the fig 5.3.1.

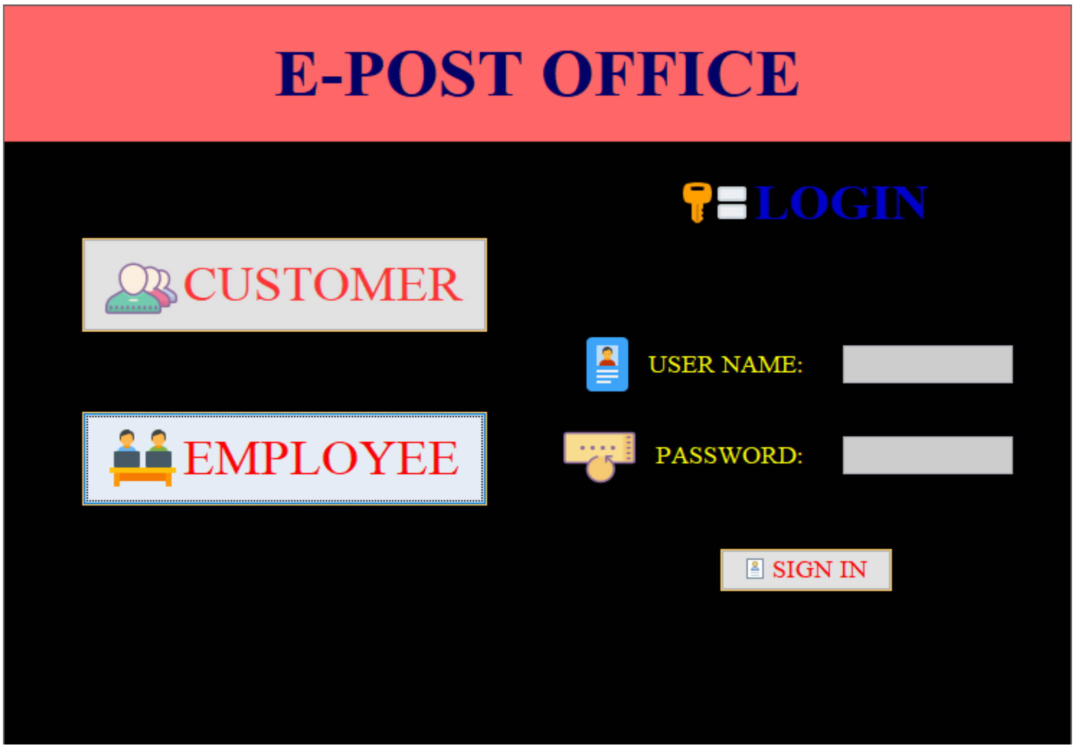
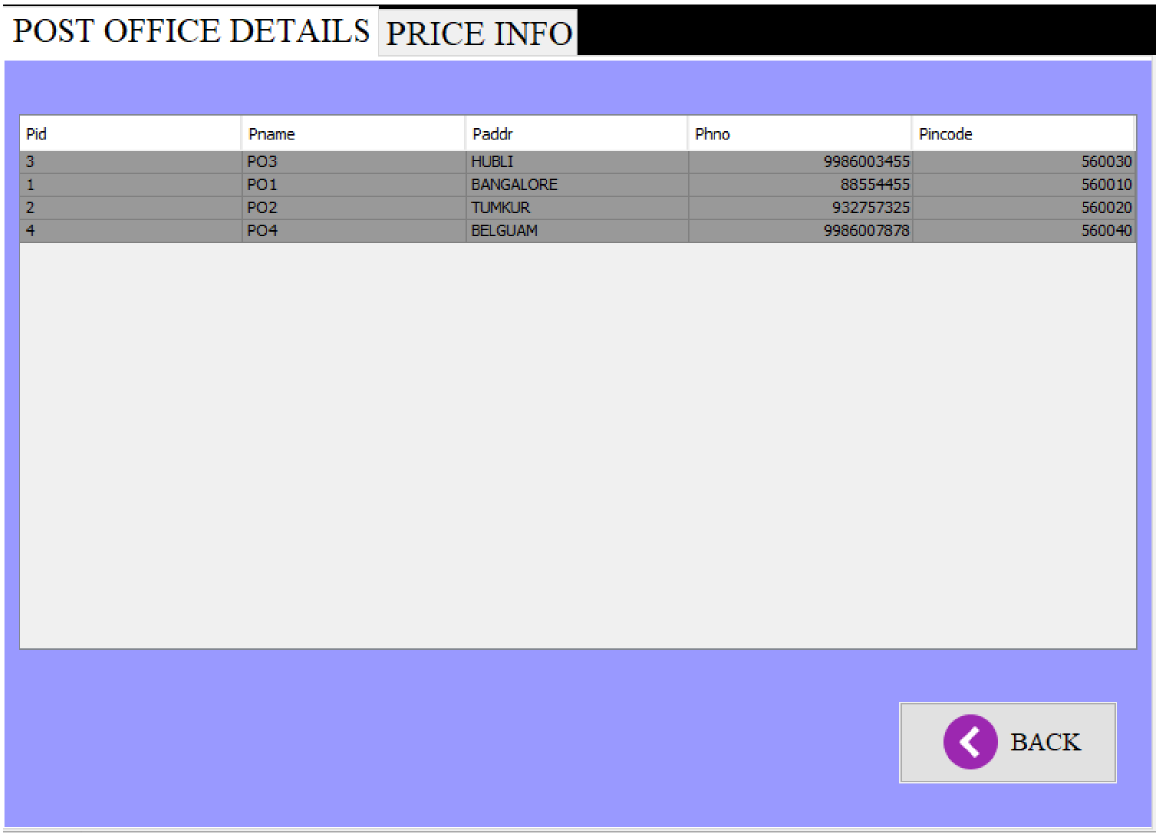


Figure 5.1: Home page

After clicking on CUSTUMER, the page takes you to the home page. Where it takes you the different options shown in fig 5.3.1. After clicking on EMPLOYEE the login field will be seen after giving the input, it will take you to the main page as shown in fig 5.3.2 and fig 5.3.3

Once the administrator is logged in, he can access different options, which are displayed in below figures. In fig 5.3.5 post office details can be managed like inserting, displaying and update. In fig 5.3.7 employee details can be managed like inserting, displaying and update. In fig 5.3.6 price details can be managed like inserting, displaying and update. The above form allows admin to login into the database successfully.

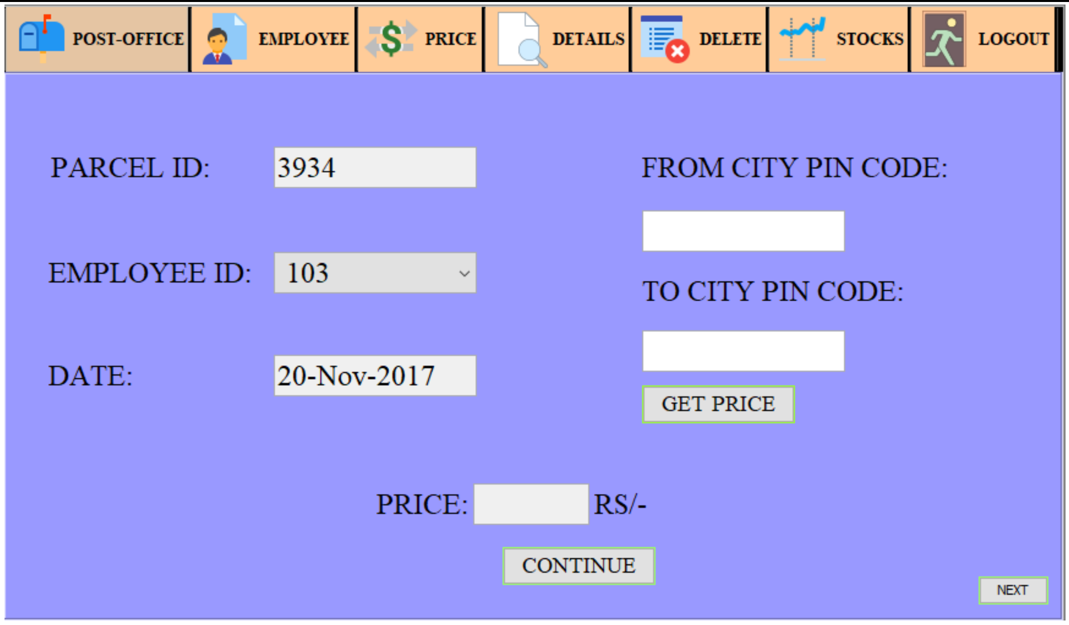


Pid	Pname	Paddr	Phno	Pincode
3	PO3	HUBLI	9986003455	560030
1	PO1	BANGALORE	88554455	560010
2	PO2	TUMKUR	932757325	560020
4	PO4	BELGUAM	9986007878	560040

Figure 5.2 : Customer page

The first tab Post Office Details will display the list of post-office. The second tab Price Details will display the list of price. The entries made by the administrator will be displayed for the customers for reference. These tabs contain the information about the post office and the price of parcel charges from one city to another. The back button is placed for returning to the home page. In further enhancement, we will provide the new about the post offices and updates in the home page.

The tables are directly connected to the database so whenever the details modifie by the admin will be updated in this. Hence, the application works dynamically.



PARCEL ID: 3934

EMPLOYEE ID: 103

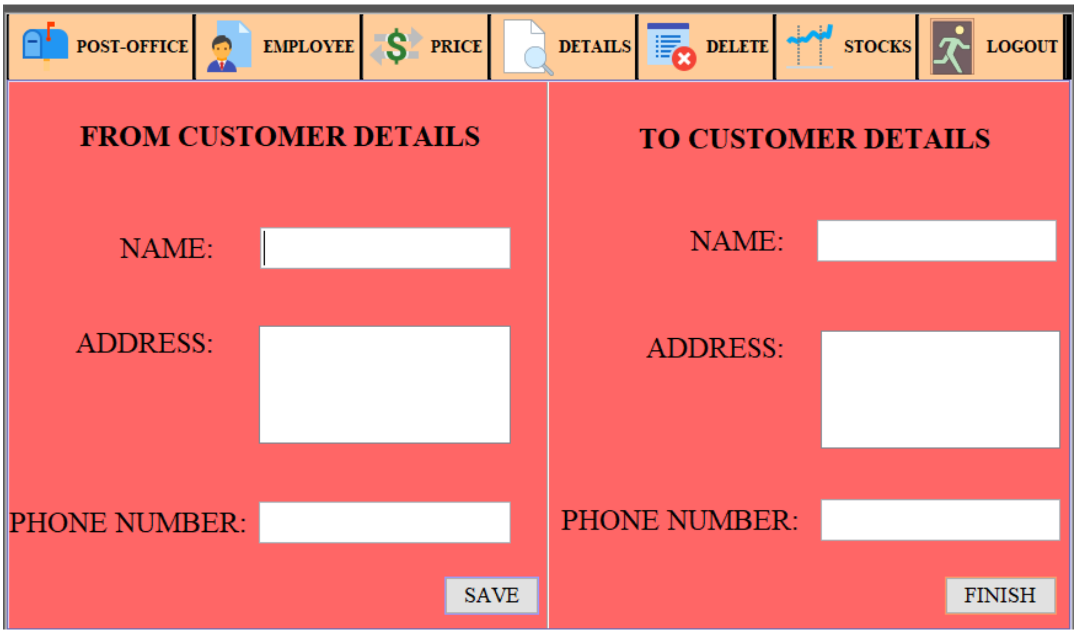
DATE: 20-Nov-2017

FROM CITY PIN CODE:

TO CITY PIN CODE:

PRICE: RS/-

Figure 5.3: Parcel Details (1) page



FROM CUSTOMER DETAILS

NAME:

ADDRESS:

PHONE NUMBER:

TO CUSTOMER DETAILS

NAME:

ADDRESS:

PHONE NUMBER:

Figure 5.4: Parcel Details (2) page

We can add the parcel details in the shown text box and click Add button. The message will be shown as INSERTED SUCCESSFULLY. In the parcel id is automatically generated. It uses unique id to avoid collision. It also automatically shows today’s date. Once the employee enters pin codes of two cities the price will be automatically displayed. Then its goes to the next part where employee has to enter the names of the customes.

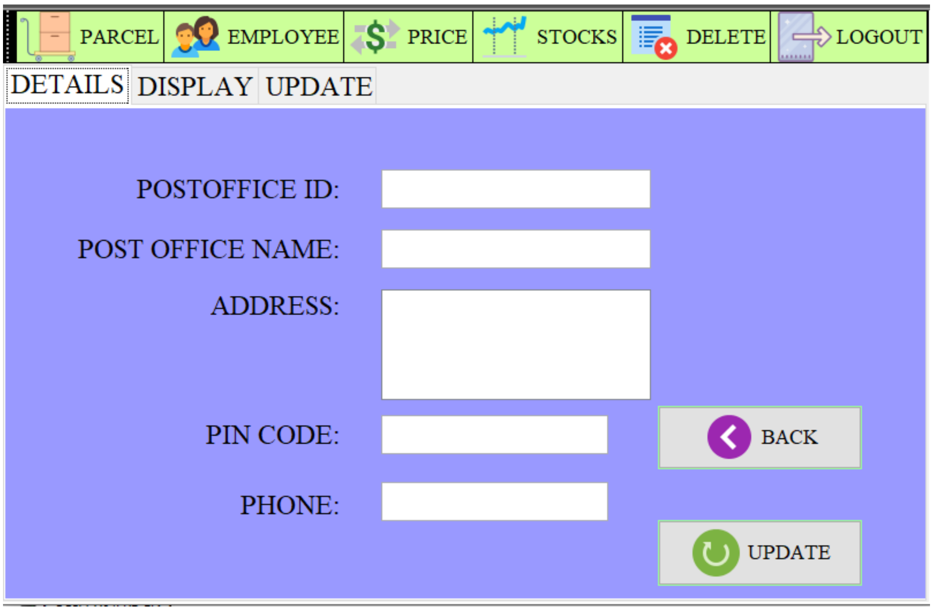


Figure 5.5: Post Office Page

Takes the input for the post office details and it will be displayed. And it also have the update function for update the existing details

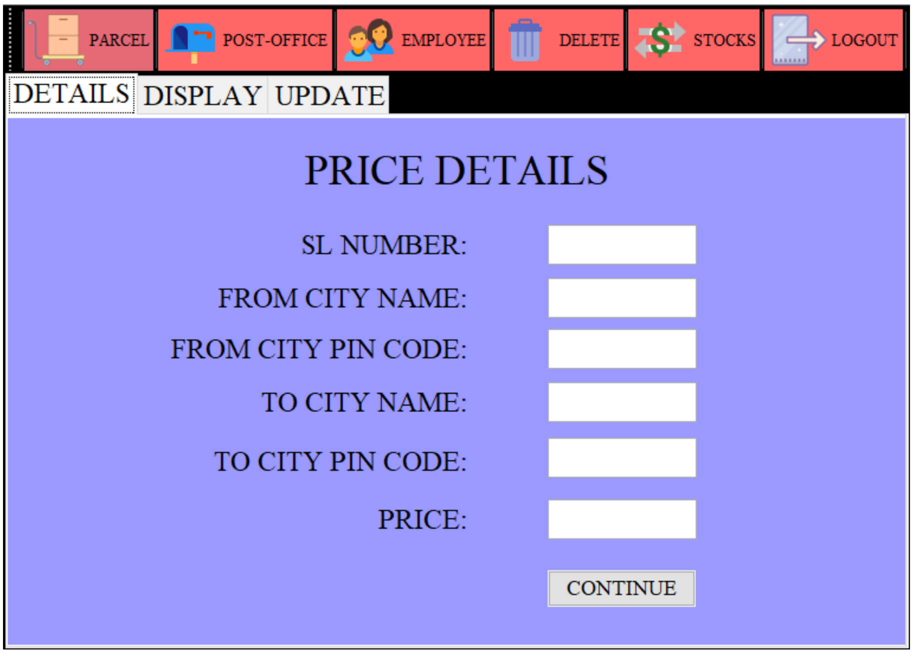
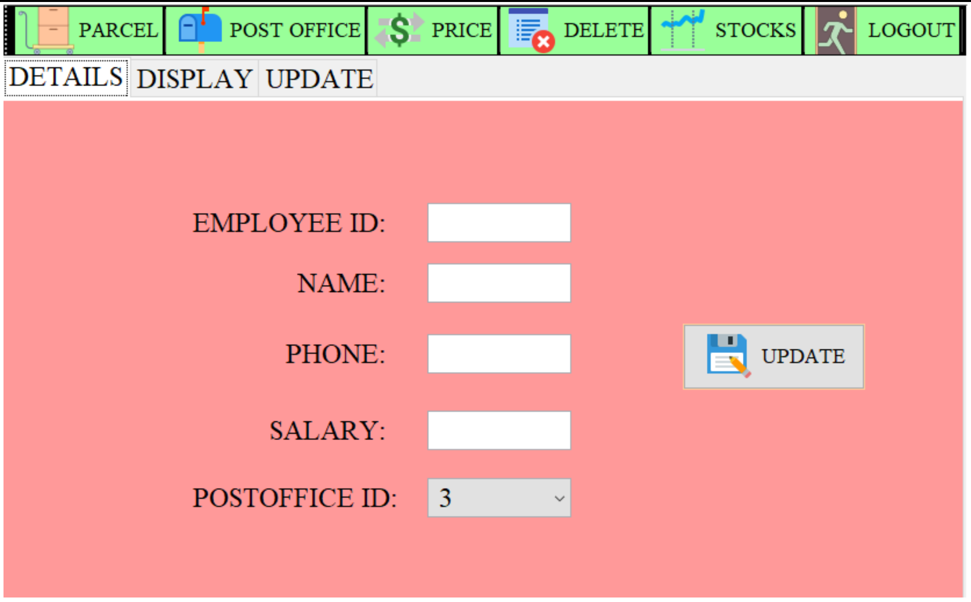


Figure 5.6: Price Page

Takes the input for the price details and it will be displayed. And it also have the update function for update the existing details



EMPLOYEE ID:

NAME:

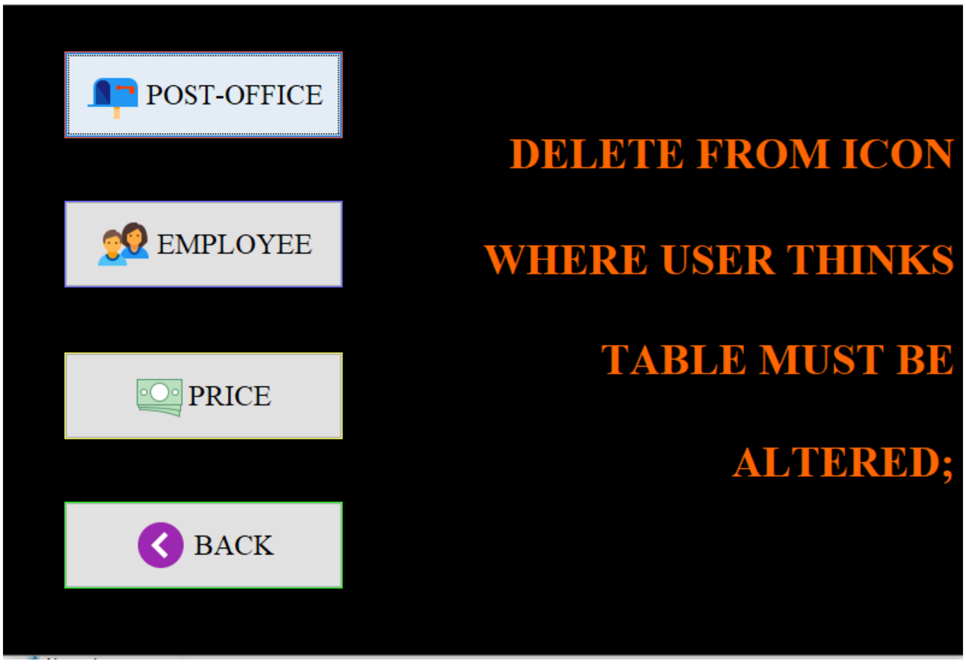
PHONE:

SALARY:

POSTOFFICE ID: 3

Figure 5.7: Employee Page

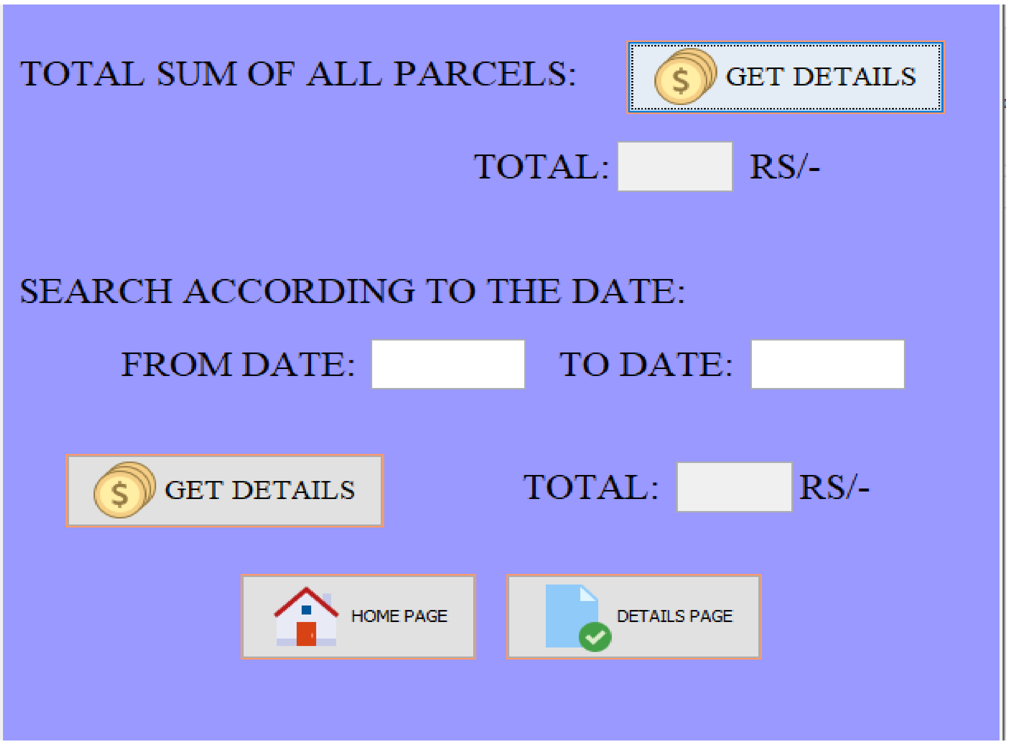
Takes the input for the price details and it will be displayed. And it also have the update function for update the existing details



DELETE FROM ICON WHERE USER THINKS TABLE MUST BE ALTERED;

Figure 5.8 : Delete Page

In the fig 5.3.8 the values inserted in the post office, employee and price tables can be deleted easily.



TOTAL SUM OF ALL PARCELS:

TOTAL: RS/-

SEARCH ACCORDING TO THE DATE:

FROM DATE: TO DATE:

TOTAL: RS/-

Figure 5.9 : Stock Page

In fig 5.3.9 the calculation of total profit amount of that month and the search function according to date is present. Here we can view the total sales made in the shop till date.From which the administrator will easily calculate the profit.

Hence, all the frames are working will all the operations without any errors and with basic valaditation. It will enhanced in further builds.

Chapter 6

CONCLUSION AND FURTHER ENHANCEMENT

6.1 CONCLUSSION:

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in JAVA and some extent Windows Application and Oracle 10g, but also about all handling procedure related with **"E-POST OFFICE MANAGEMENT SYSTEM".** It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

LIMITATIONS:

- The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity.
- Training for simple computer operations is necessary for the users working on the system.

6.2 FUTURE ENHANCEMENT:

- This System should be converted into web-based application and an undertaking of Cyber Security Division, needs to be thoroughly tested to find out any security gaps.
- A console for the data center may be made available to allow the personnel to monitor on the sites, which were cleared for hosting during a particular period.
- Moreover, it is just a beginning; further, the system may be utilized in various other types of auditing operation viz. Network auditing or similar process/workflow based applications...
- A number of directions can be followed as an extension of this project. Some challenges and open questions still to be explored in the project work.
- New methods to improve the performance of Post office Management System.

REFERENCES:

[1] Fundamentals of Database Systems, 7thEdition, 2017, Pearson.

[2] Database management systems, 3rd Edition, 2014, McGraw Hill.

[3] JAVA the complete Reference, 7th Edition, Tata McGraw Hill, 2007.

[4] J2EE-The Complete Reference, McGraw Hill, 2007.

[5] https://netbeans.org/

[6] https://stackoverflow.com/

[7] https://www.oracle.com/index.html

[8] https://docs.oracle.com

[9] https://stackoverflow.com/

[10] https://hackr.io/

[11] https://w3schools.com/