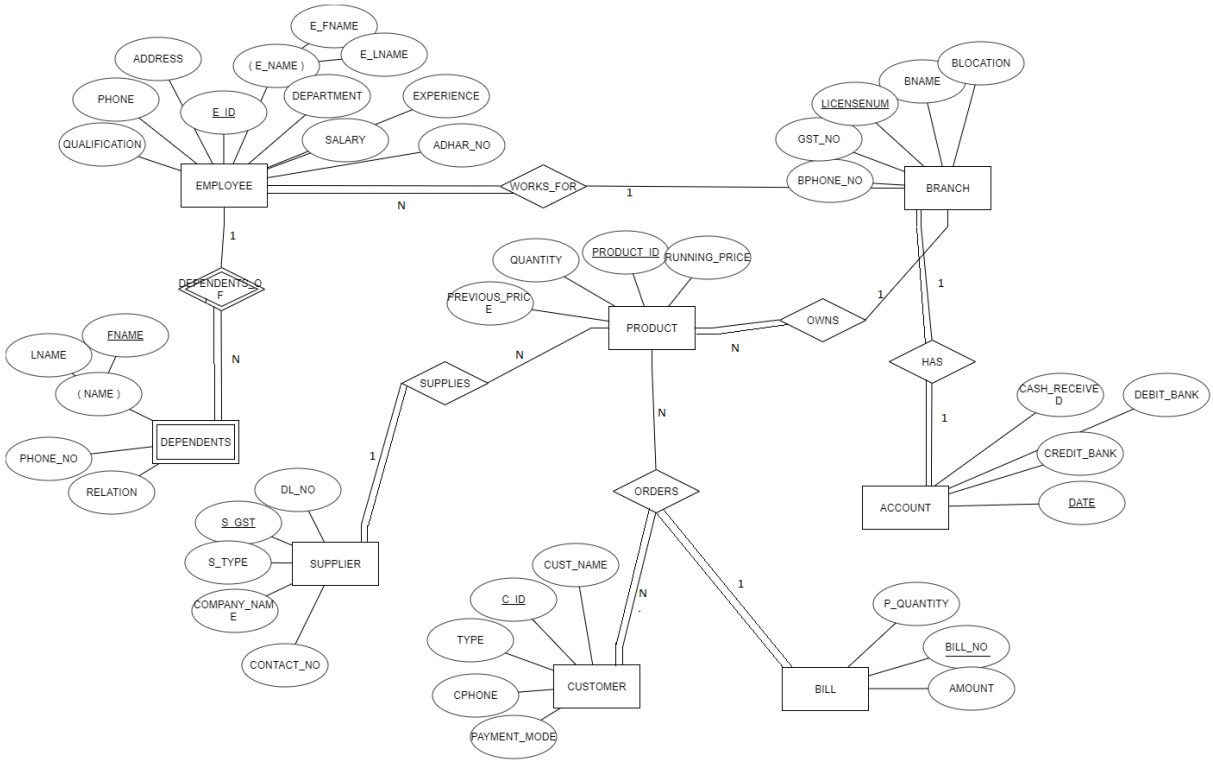
Project Report

Design updating: [Describe the changes (if any) that need to be made to the Entity-Relationship Diagram you have given in Design Phase before it can be implemented as tables in SQL. If you identify any changes, redraw a new E/R diagram.]



Normalization:

No changes in the normalization.

Question1: Give the SQL statement(s) used to create the Oracle/MySQL database tables needed to implement the normalized relational schema.

```
create table Branch (
License_no varchar2(20) primary key,
Branch_name varchar2(30) not null,
Branch_loc  varchar2(40),
Branch_phone varchar2(20),
GST_No varchar2(50));

create table Employee(
Emp_id varchar2(20) primary key,
Emp_fname varchar2(30),
Emp_lname varchar2(30),
Dept varchar2(30),
Salary int,
Experience int,
Qualification varchar2(40),
Emp_Phone varchar2(20),
Address varchar2(50)
License_no varchar2(20);
```

```
create table Customers(
Cust_id varchar2(20) primary key,
Cust_name varchar2(30),
Cust_type varchar2(20),
Cust_phone varchar2(20),
Payment_mode varchar2(20));

create table Supplier(
S_GSTno varchar2(20) primary key,
Company_name varchar2(40),
Supplier_type varchar2(30),
DL_no varchar2(20),
Contact_No varchar2(20));

create table Products(
Product_id varchar2(20) primary key,
Quantity int,
Running_price float,
Previous_price float,
License_no varchar2(20),
foreign key(License_no)references
Branch(License_no),
S_GSTno varchar2(20),
```

```
foreign key(S_GSTno) references
Supplier(S_GSTno));

create table Account(
Trans_date date ,
License_no varchar2(20),
primary key(Trans_date,License_no),
Credit_bank int,
Debit_bank int,
Cash_received float,
foreign key(License_no)references
Branch(License_no));

create table Dependants(
Dep_name varchar2(40),
Emp_id varchar2(20),
primary key(Dep_name,Emp_id),
foreign key (Emp_id) references
Employee(Emp_id),
Dep_Phone varchar2(20),
Relation varchar2(20));

create table bill(
```

cust_id varchar2(20),

product_id varchar2(60),

quantity int,

amount float,

date date,

bill(cust_id) foreign key references customers(cust_id),

bill(product_id) foreign key references product(product_id)

primary key(cust_id,product_id,date));

Question2: Give the actual data stored in each table of the database.

select * from branch;

LICENSE_NO	BRANCH_NAME	BRANCH_LOC	BRANCH_PHONE	GST_NO

confidential

select * from employee;

EMP_ID	EMP_FNAME	EMP_LNAME	DEPT	SALARY_IN_RUPEES	EXPERIENCE	QUALIFICATION	EMP_PHONE	ADDRESS	LICENSE_NO

Confidential;

select * from customers;

CUST_ID	CUST_NAME	CUST_TYPE	CUST_PHONE	PAYMENT_MODE

select * from products;

PRODUCT_ID	RUNNING_PRICE	PREVIOUS_PRICE	LICENSE_NO	S_GSTNO

select * from account;

TRANS_DATE	LICENSE_NO	CREDIT_BANK	DEBIT_BANK	CASH_RECEIVED

select * from supplier;

S_GSTNO	COMPANY_NAME	SUPPLIER_TYPE	DL_NO	CONTACT_NO

select * from dependents;

D_NAME	EMP_ID	DEP_PHONE	RELATION

select * from bill;

CUST_ID	PRODUCT_ID	QUANTITY	AMOUNT	TDATE	BILLNO

Question3: Give the snapshots, description and SQL queries for each of the user interface forms for your application. (Create the front end using python or java and hook it up to the SQL.)

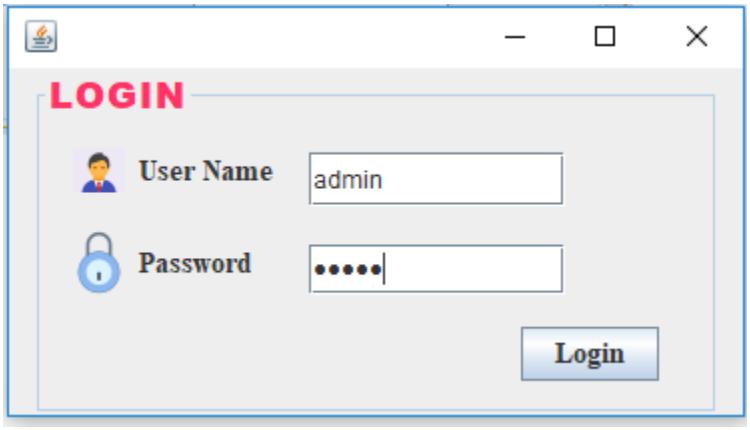


Figure 1: Login Screen to access the database.

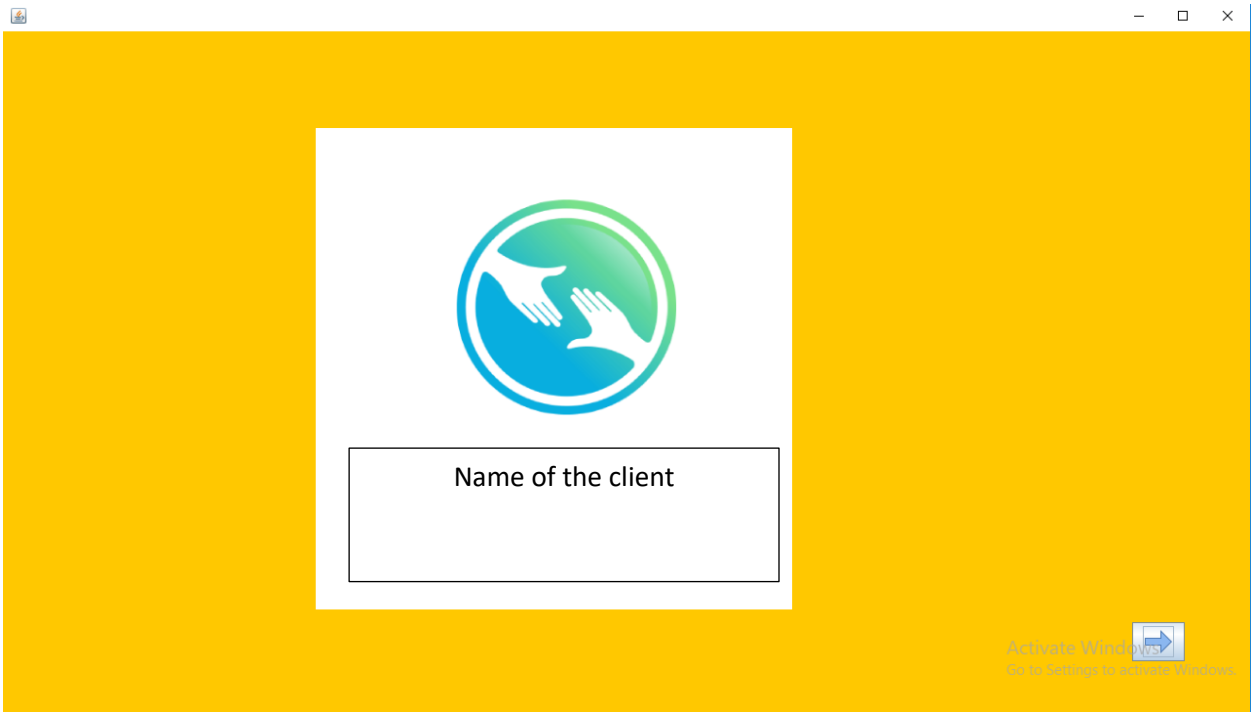


Figure 2: Welcome page of the database.

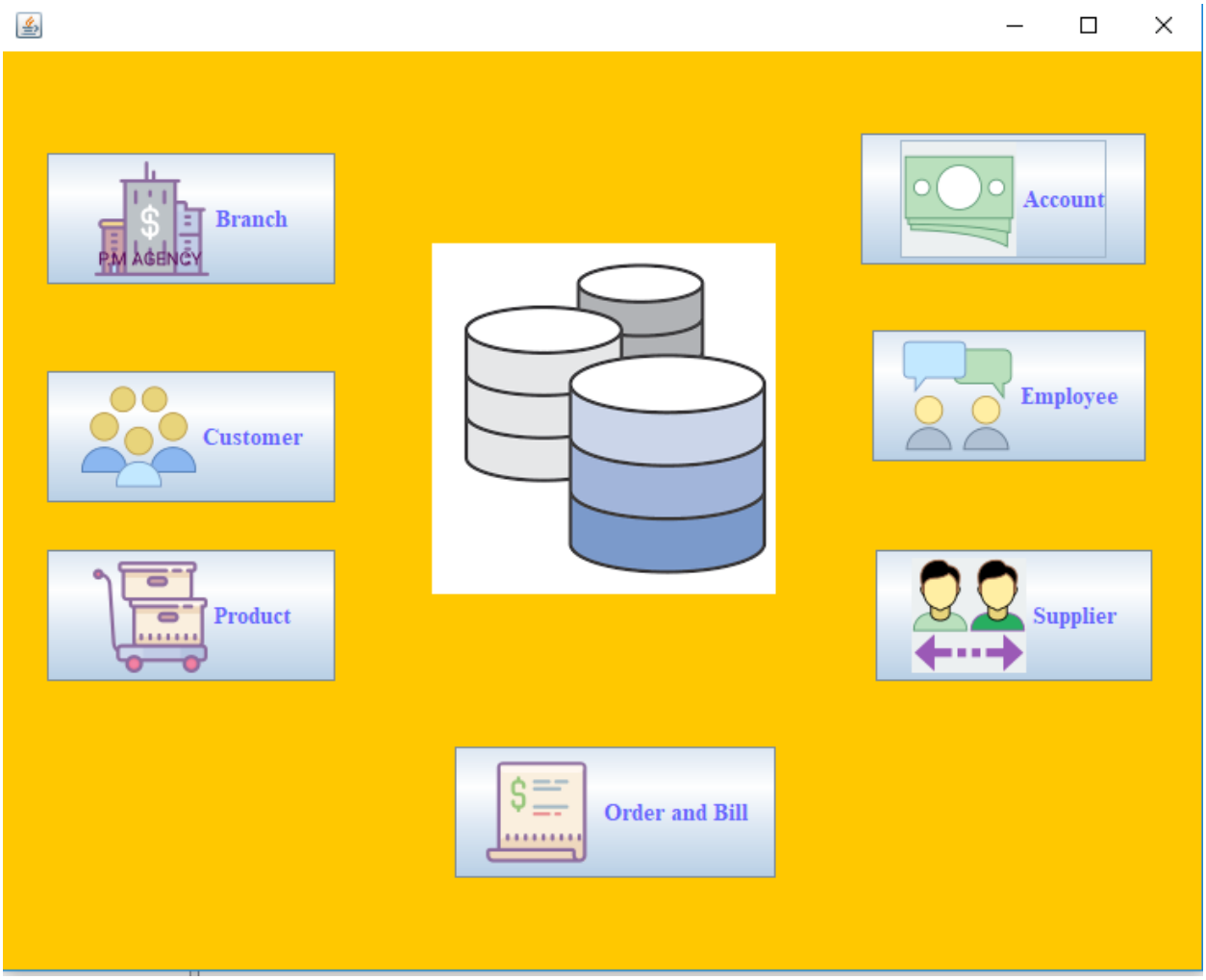


Figure 3: Different functionalities of database.

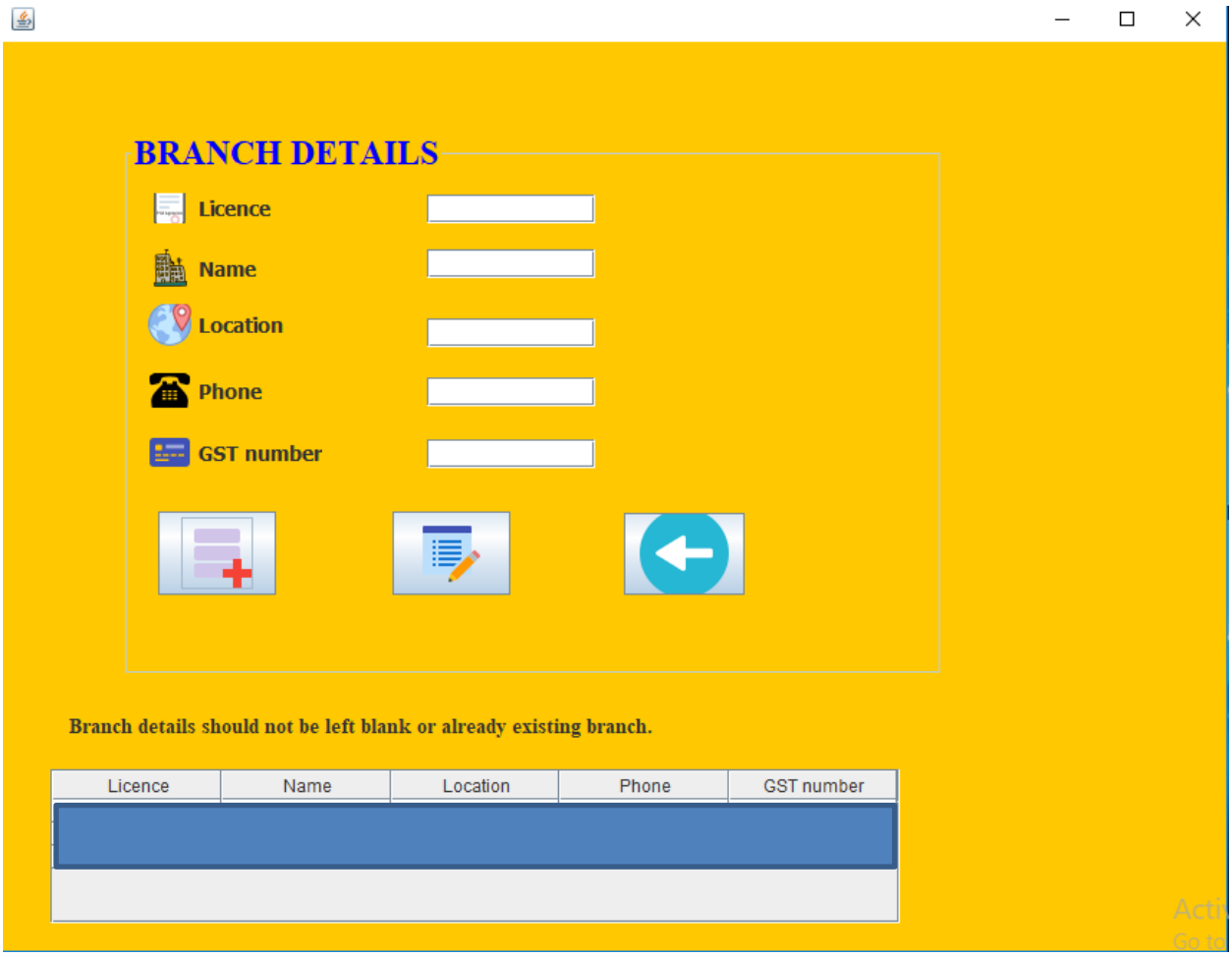


Figure 4: Screen to update or insert the information of branches.

Queries:

Insert query: state.executeQuery("insert into branch values('"+f1+"','"+f2+"','"+f3+"','"+f4+"','"+f5+"')");

Update query: ResultSet rs=s.executeQuery("update branch set branch_name='"+f2+"',branch_loc='"+f3+"',branch_phone='"+f4+"',gst_no='"+f5+"' where license_no='"+f1+"'");

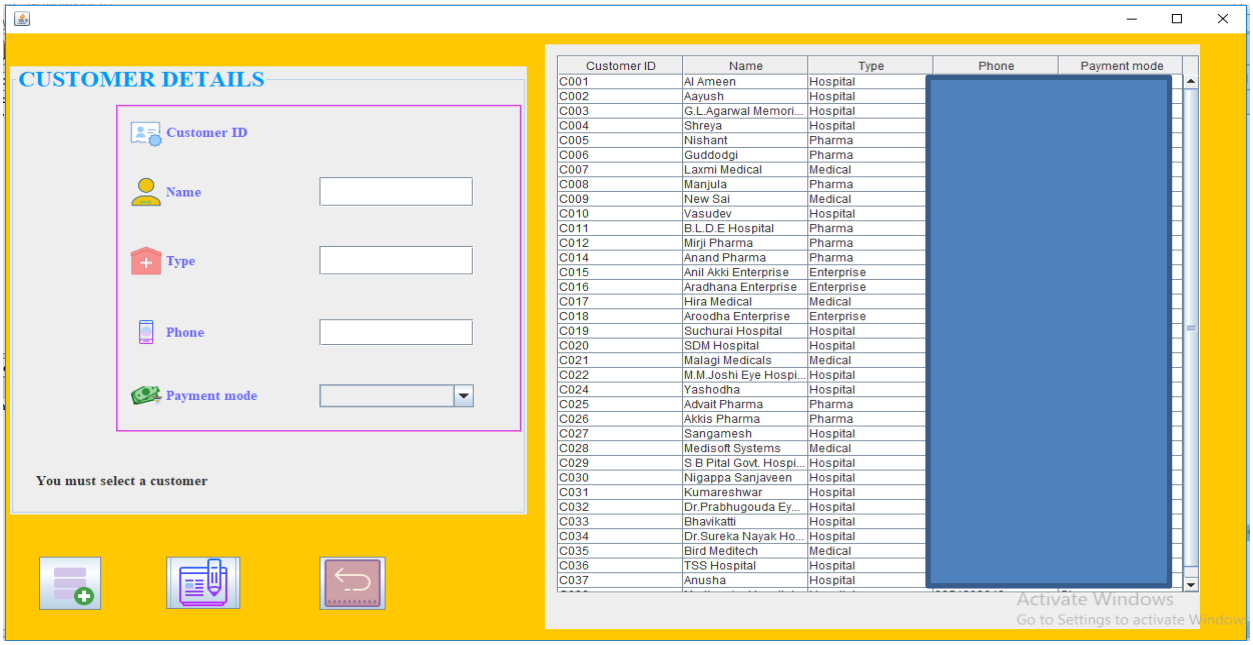


Figure 5: Screen to insert and update customer details.

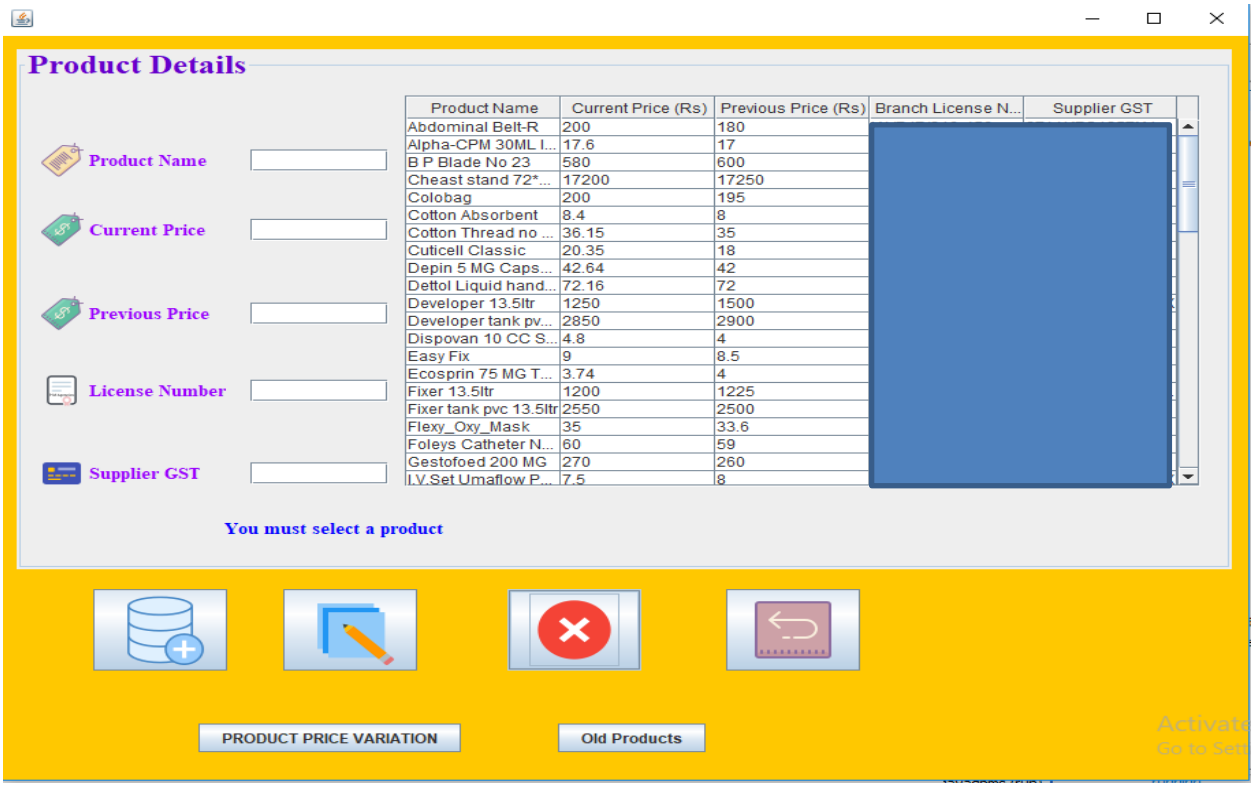


Figure 6: Screen to insert, update and delete product details.

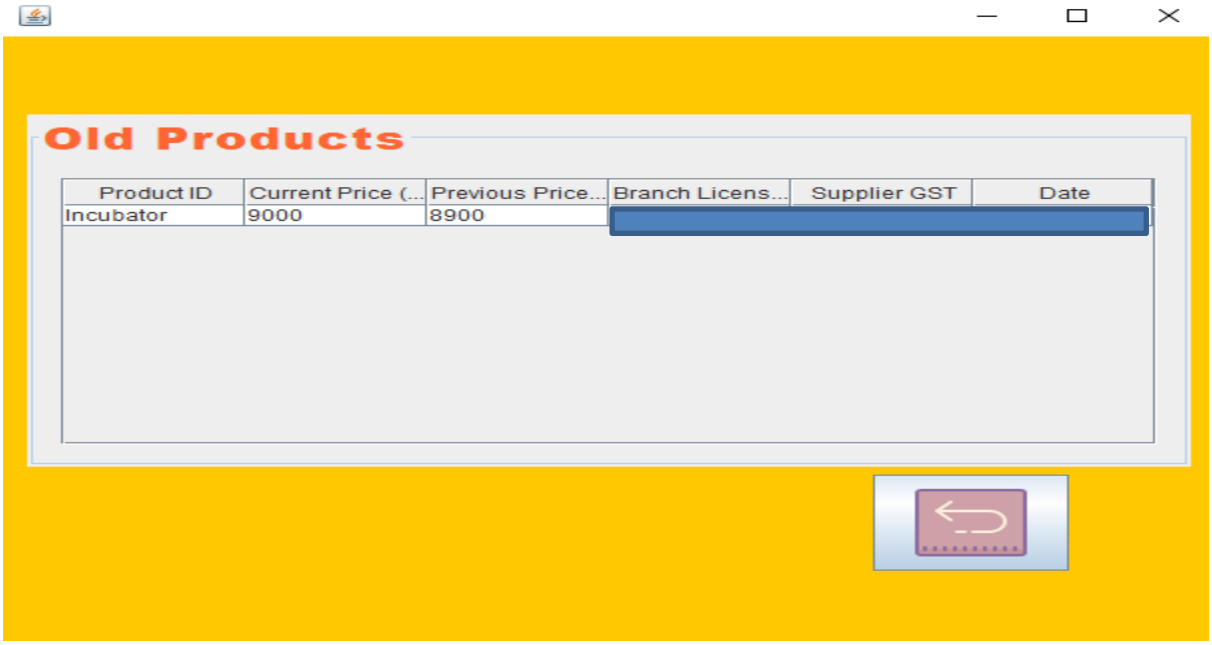


Figure 7: Old products back up.

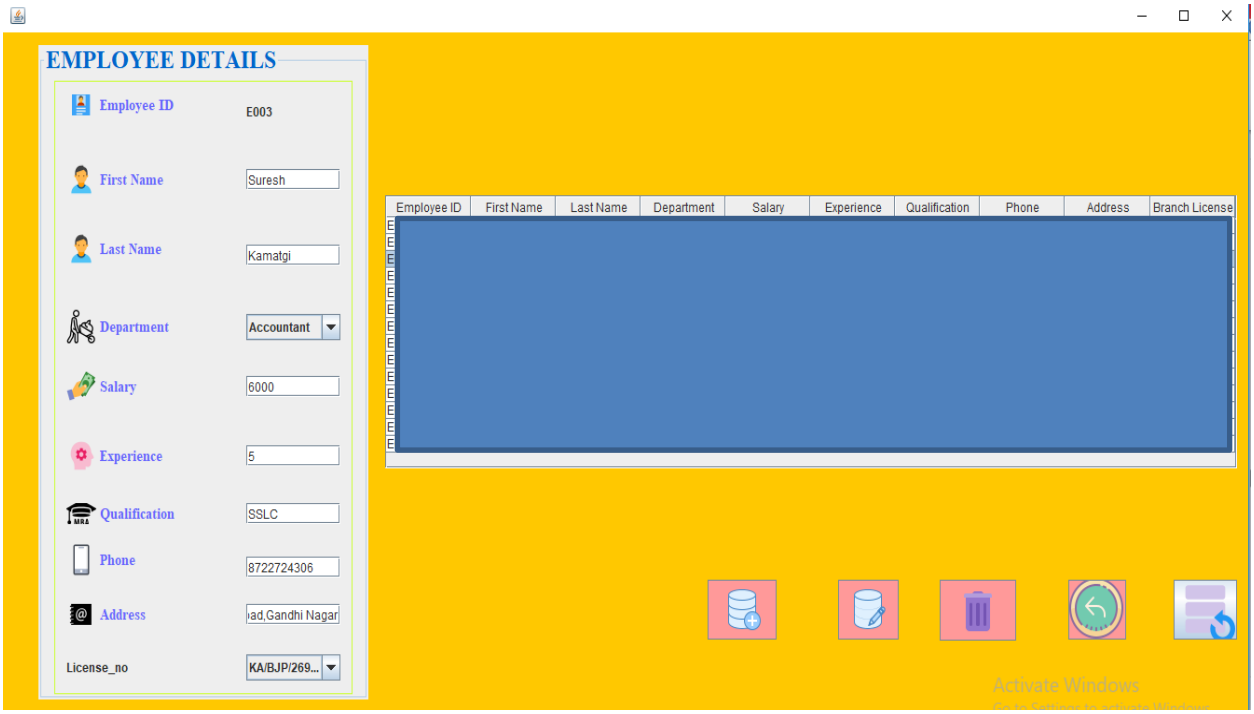


Figure 8: Screen to insert, update and delete the employee details.

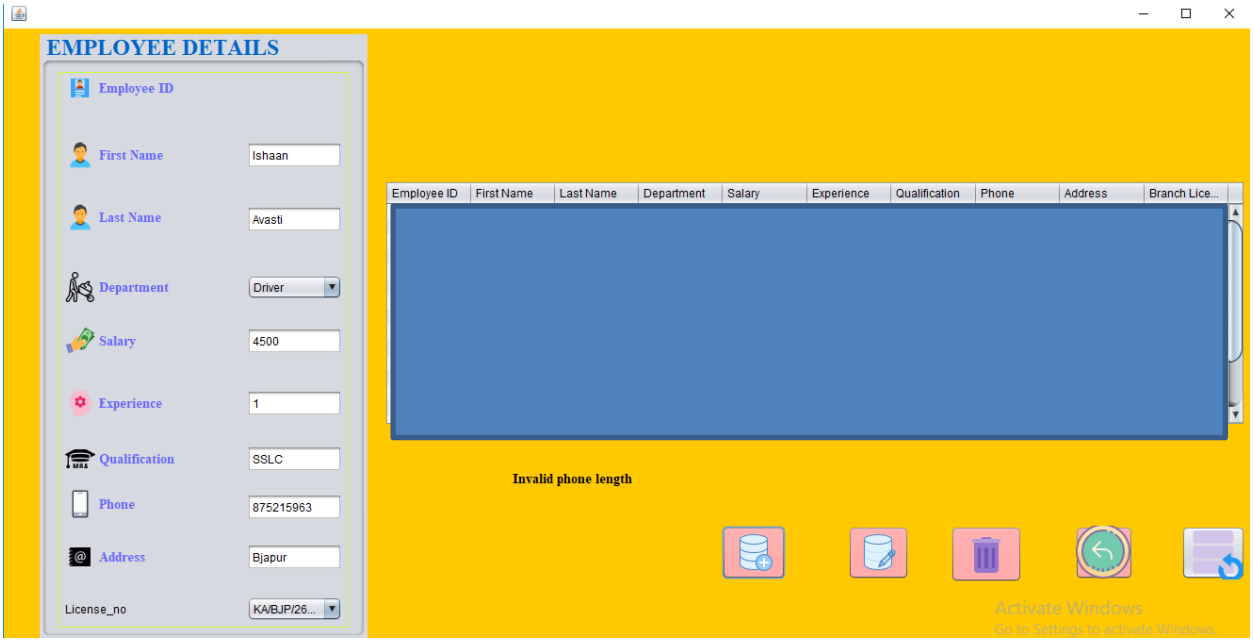


Figure 9 :Invalid Phone length

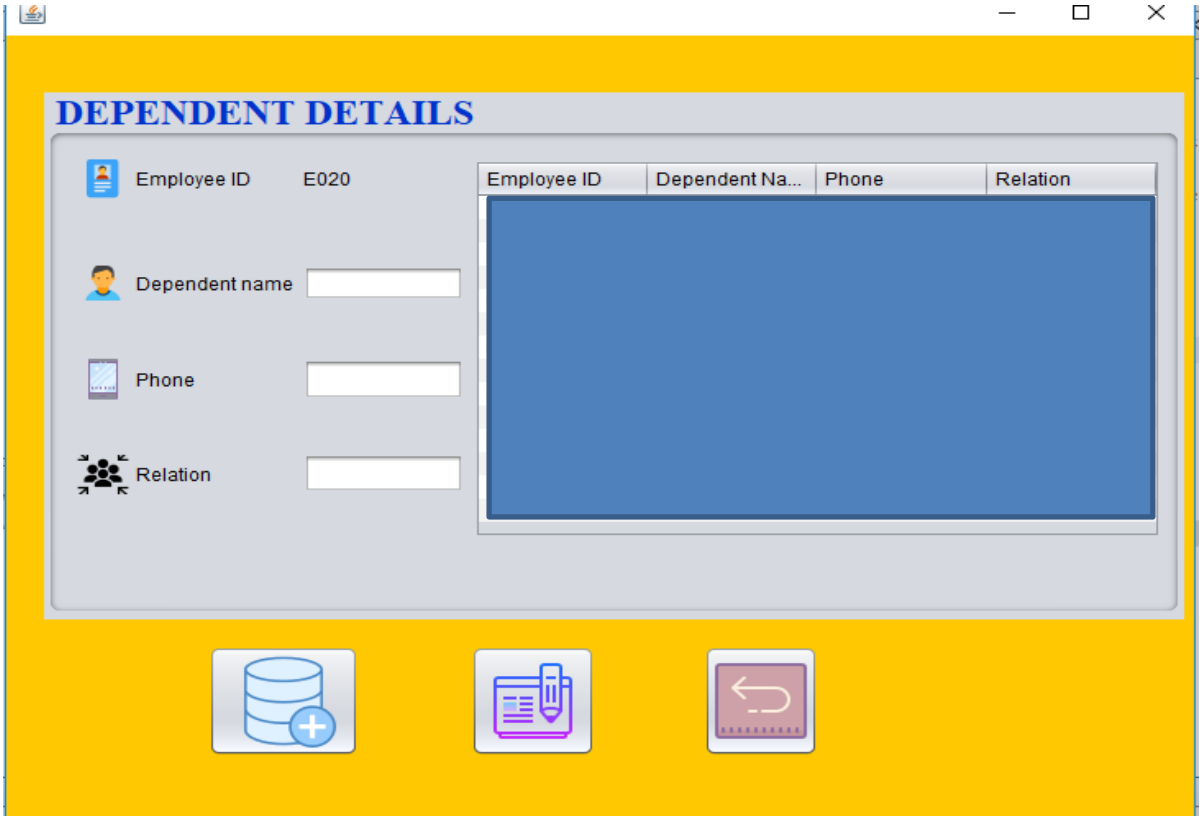


Figure 10: Dependent Screen

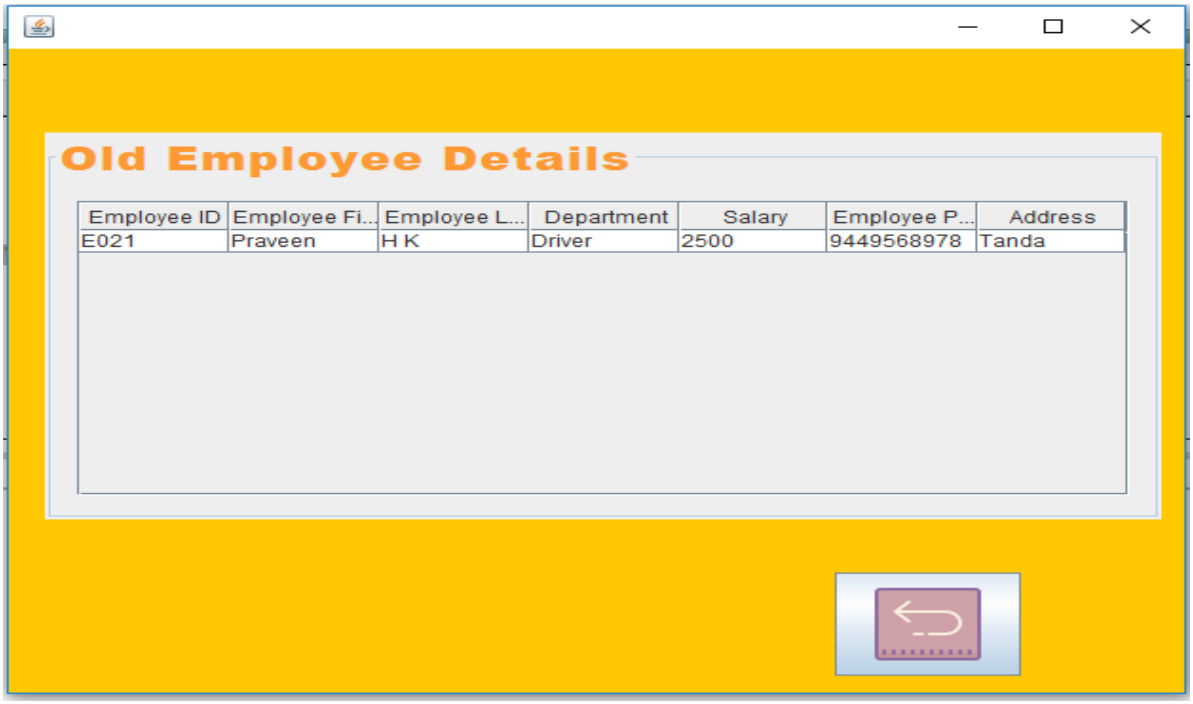


Figure 11: Table to store deleted employee details.

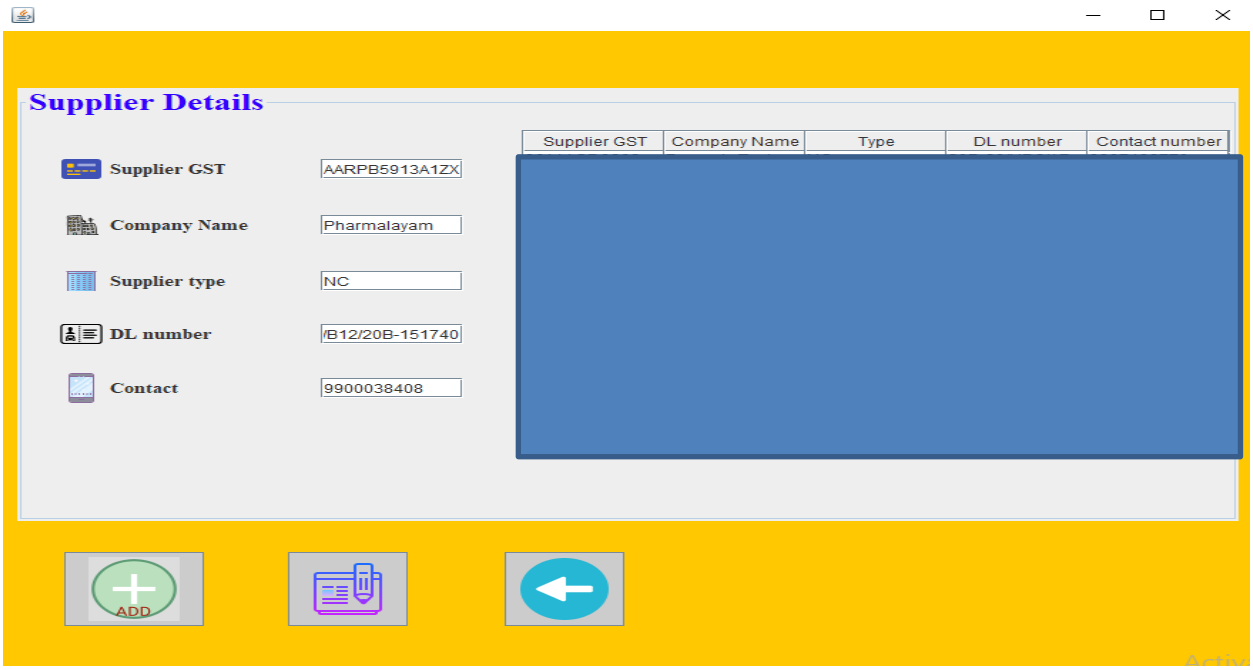


Figure 12: Screen to insert or update the information of the supplier.

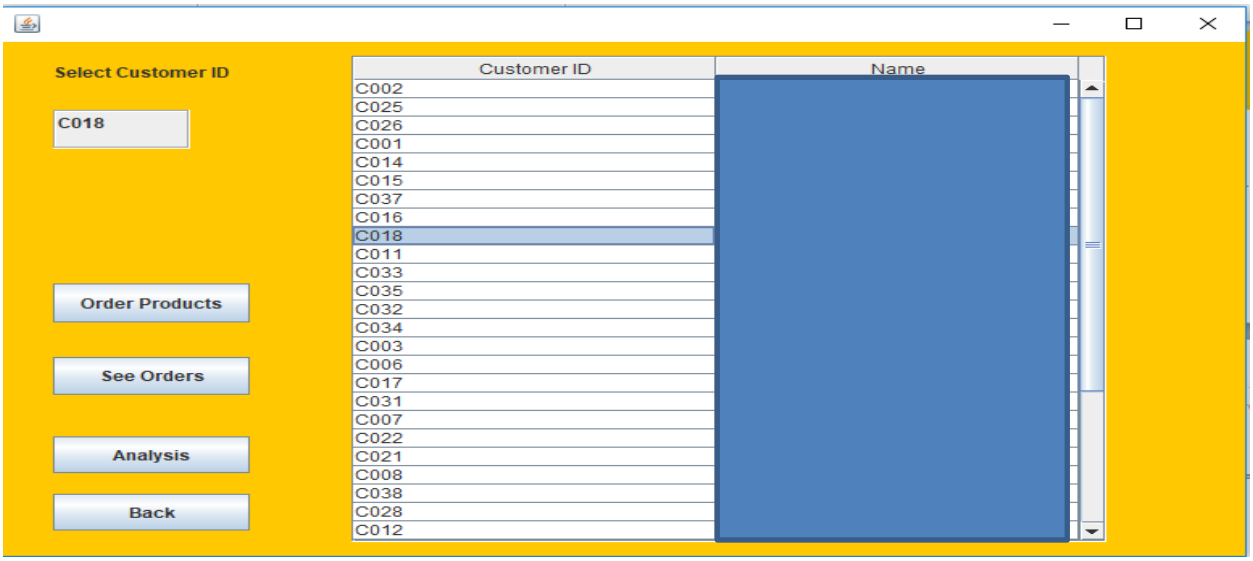


Figure 13: Screen to select the customer to order products or see their previous order.

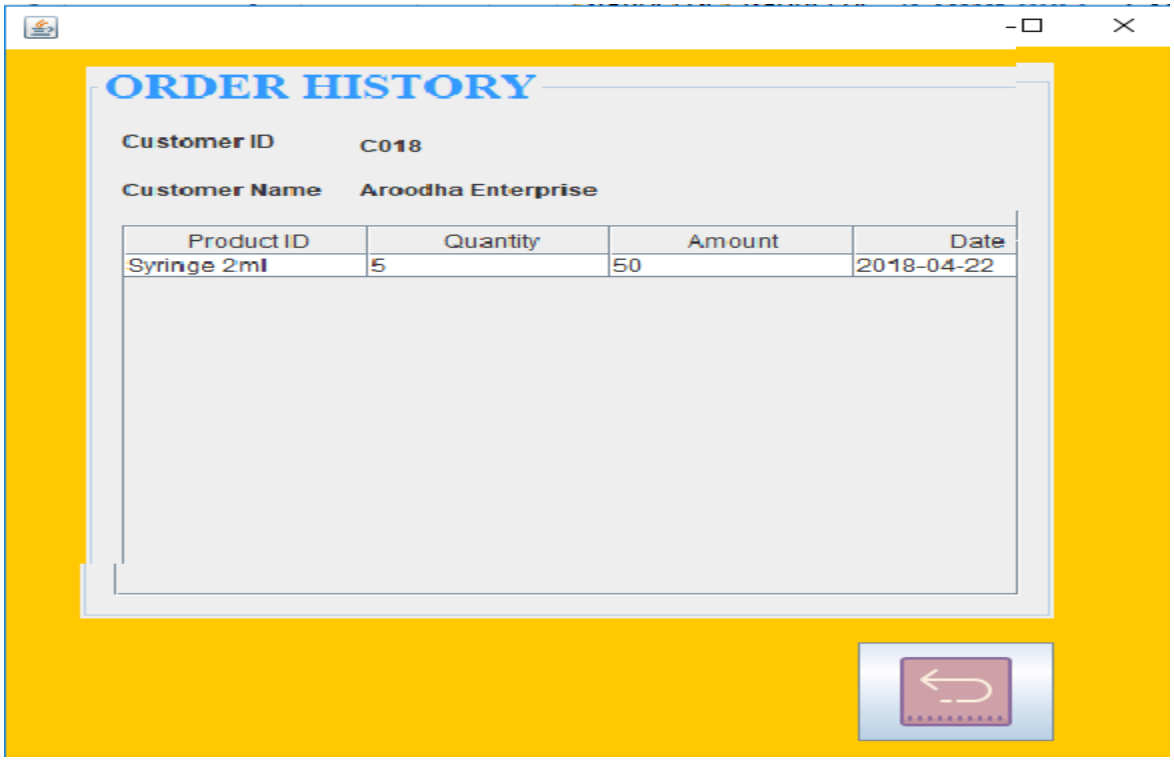


Figure 14: Screen to display the order history of selected customer.

Figure 15: Screen to display the customer analysis along with the number of products bought.

Query:

```
ResultSet rs= state.executeQuery("select c.cust_name as name,count(*) as products_count from customers c,bill b where c.cust_id=b.cust_id group by c.cust_name");
```

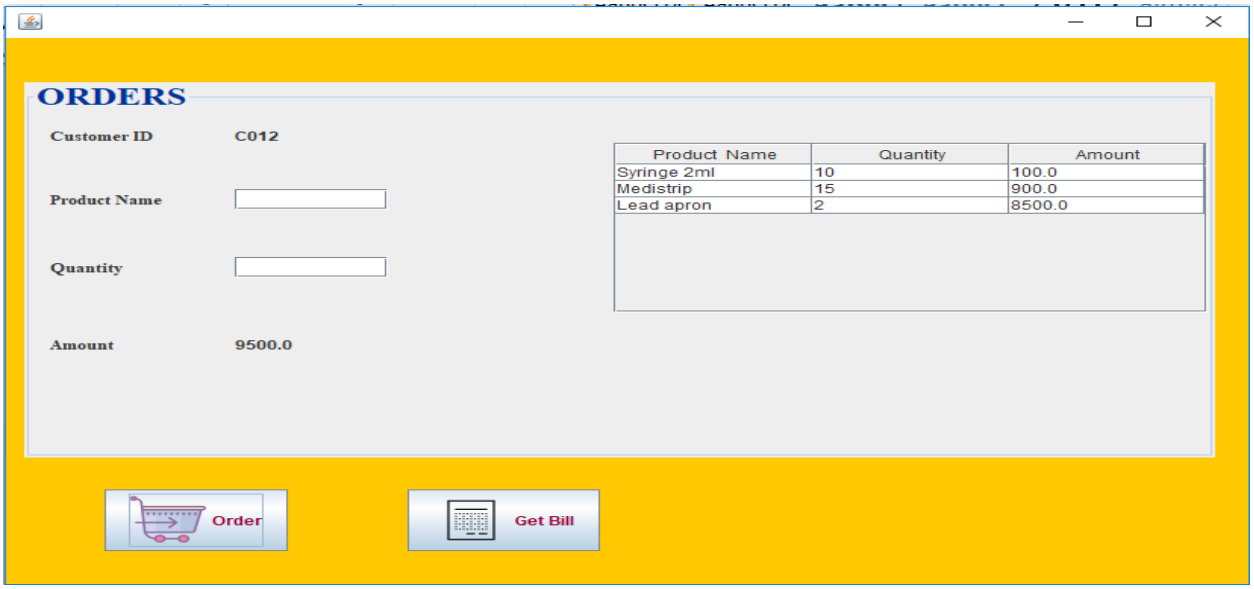


Figure 16: Screen to store the orders given by the customers and generate bill.

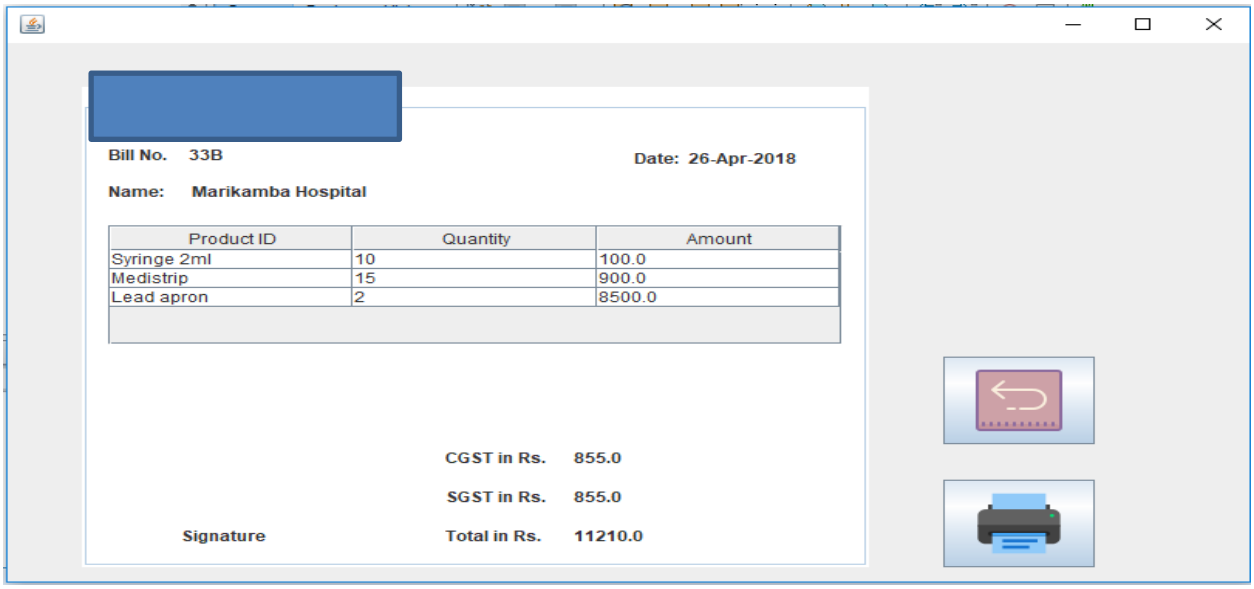


Figure 17: Screen to print the bill.

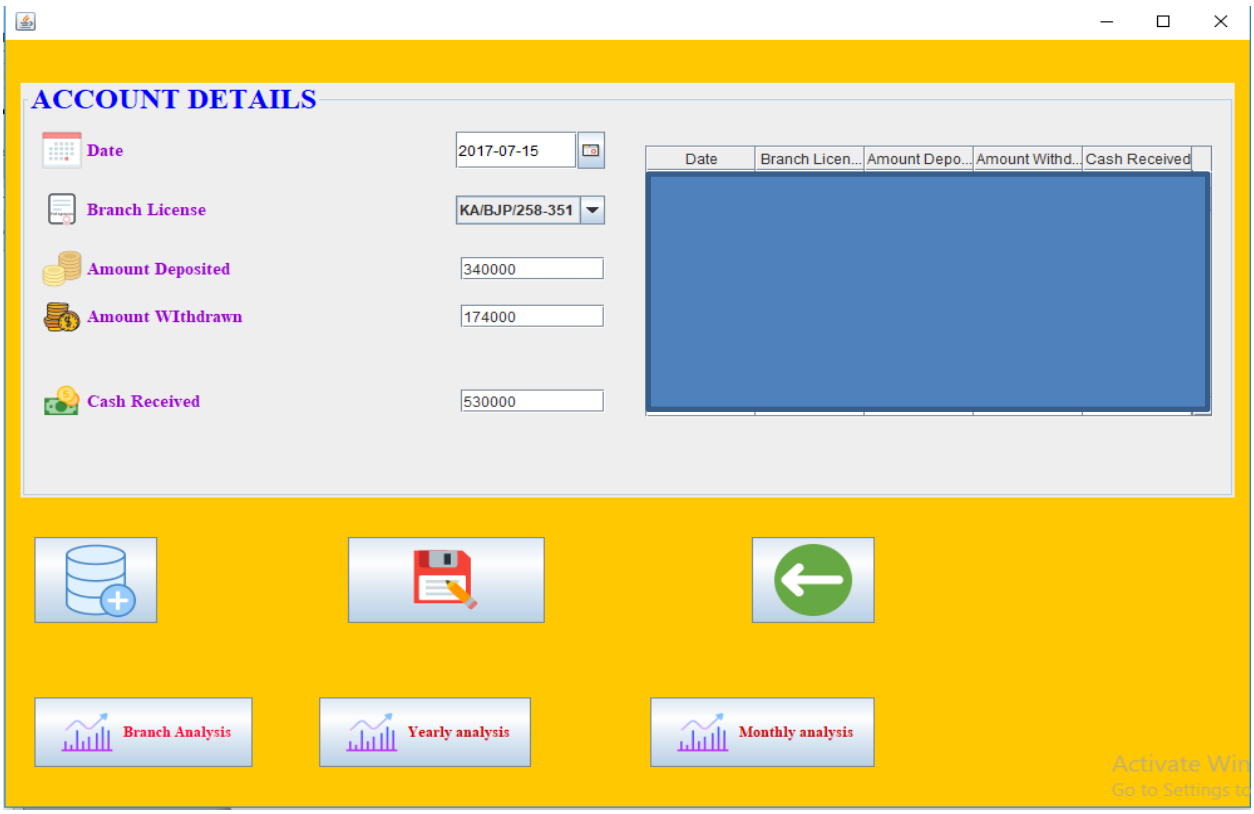


Figure 18: Screen to insert and update the account details of the branch and analyze income each branch.

Query:

ResultSet r=s.executeQuery("select sum(cash_received),license_no from account group by license_no");

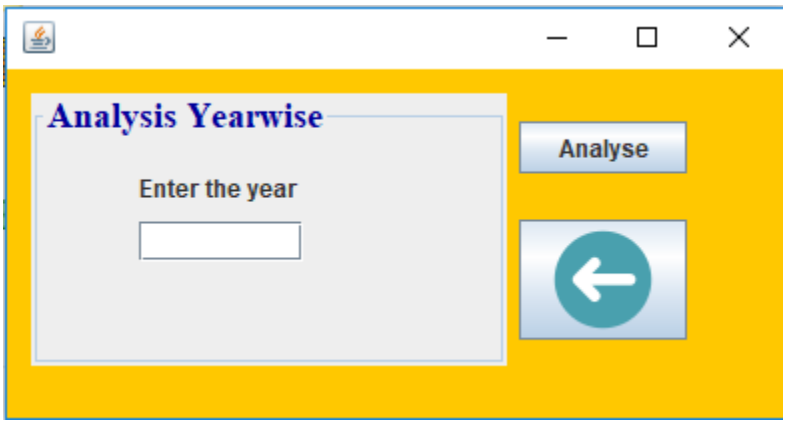


Figure 19: Screen to enter the year to plot the graph year wise.

Query:

ResultSet r=s.executeQuery("select license_no,sum(cash_received)from account where ltrim(to_char(trans_date,'yyyy'),'0') ='"+d+"'group by license_no");

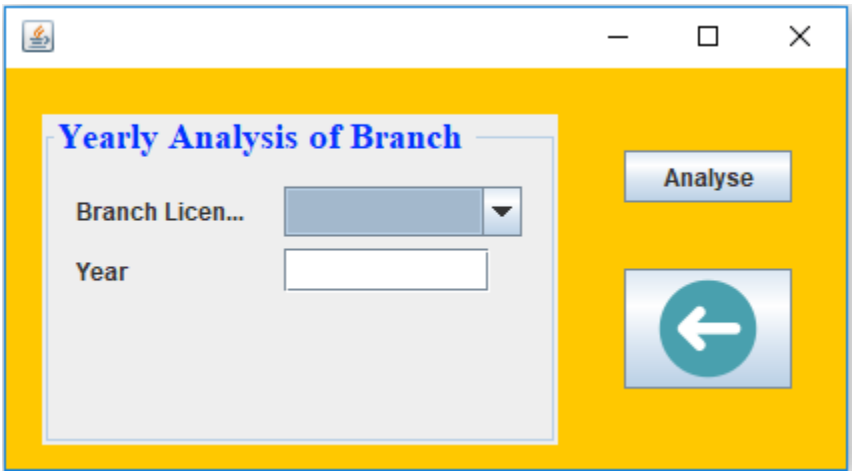


Figure 20: Screen to enter the year and branch to get the graph month wise.

Query:

ResultSet r=s.executeQuery("select sum(cash_received),ltrim(to_char(trans_date,'MM'),'0') as month from account where license_no='"+branch+"' and ltrim(to_char(trans_date,'yyyy'),'0')='"+year+"' group by ltrim(to_char(trans_date,'MM'),'0')");

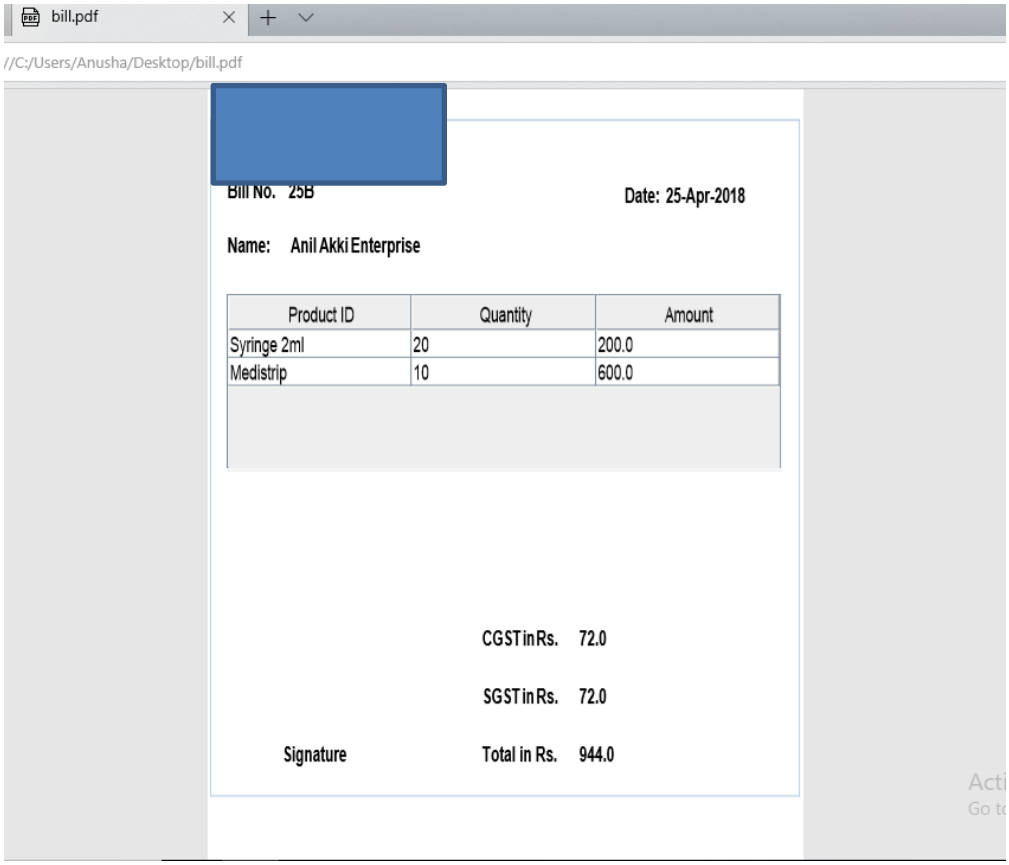


Figure 21: Bill

Question4: Give all possible final reports and graphs obtained by your application.

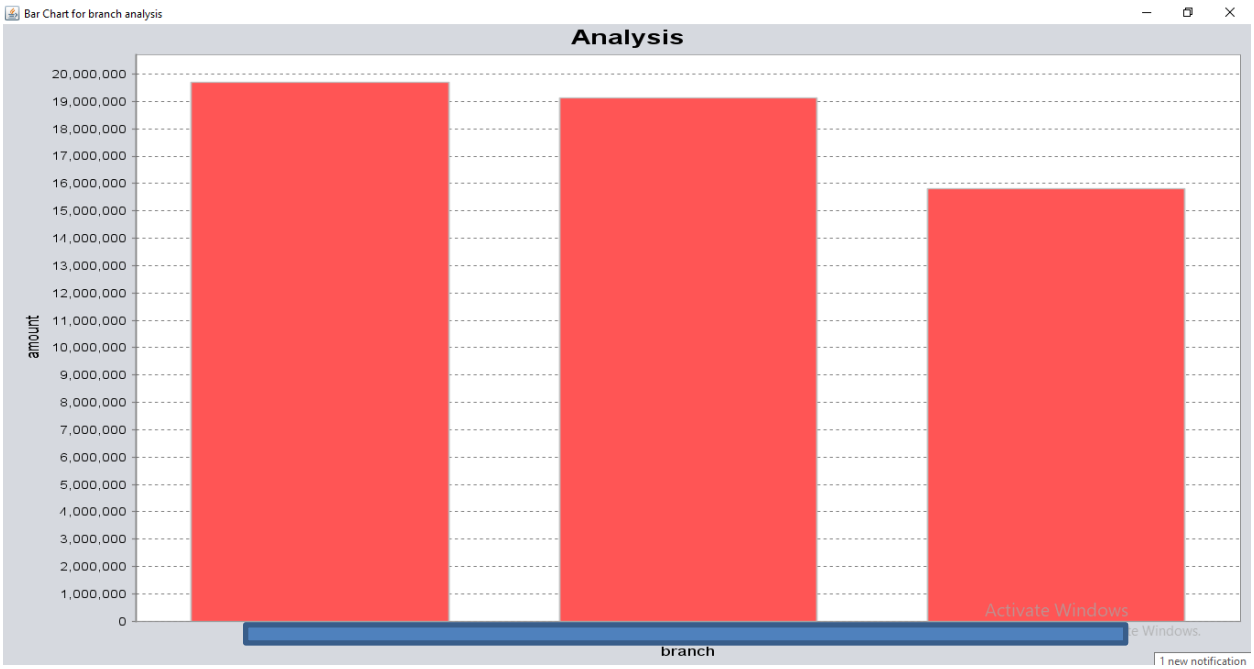


Figure 22: Branch Income Analysis

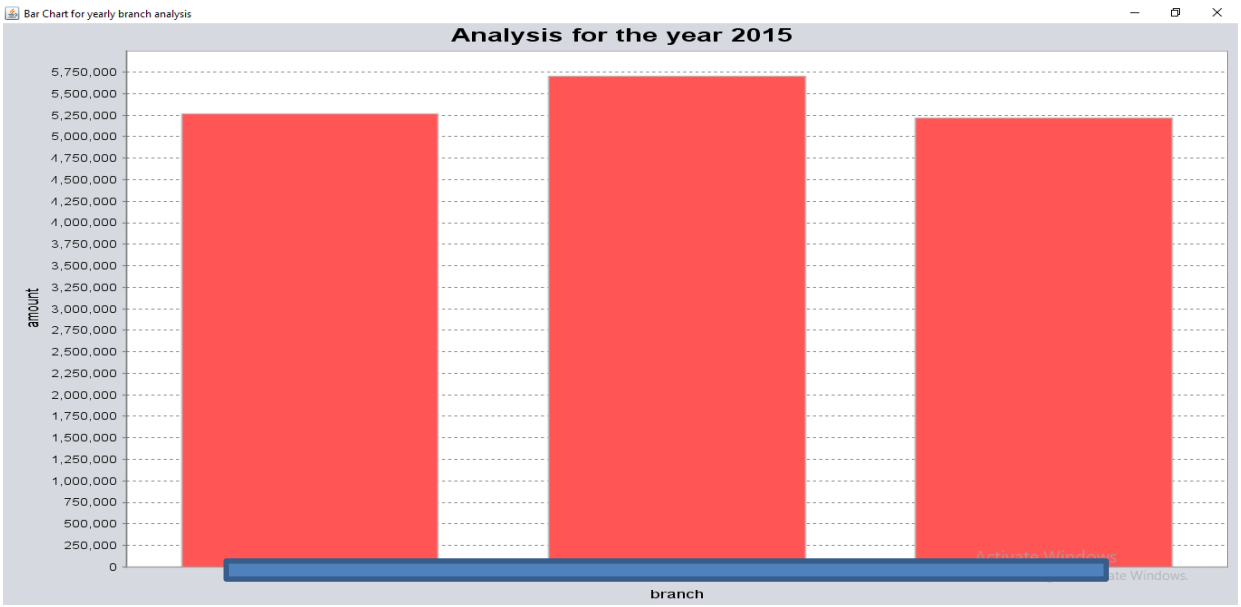


Figure 23: Branch Income Analysis for the year 2015

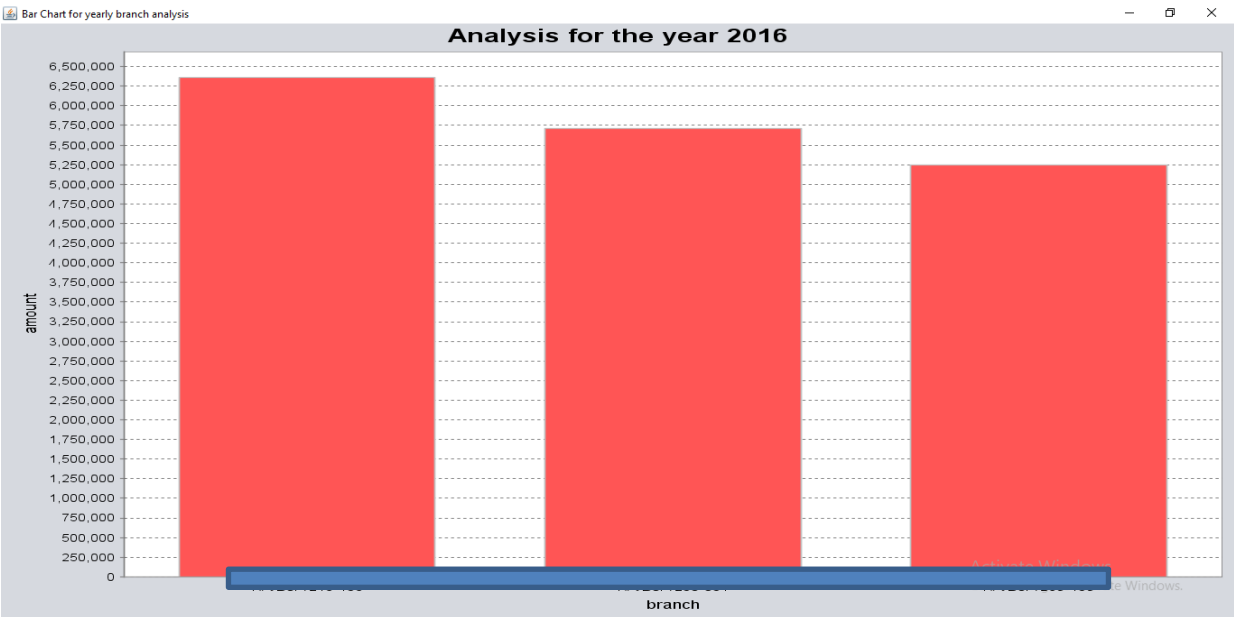


Figure 24: Branch Analysis for the year 2016

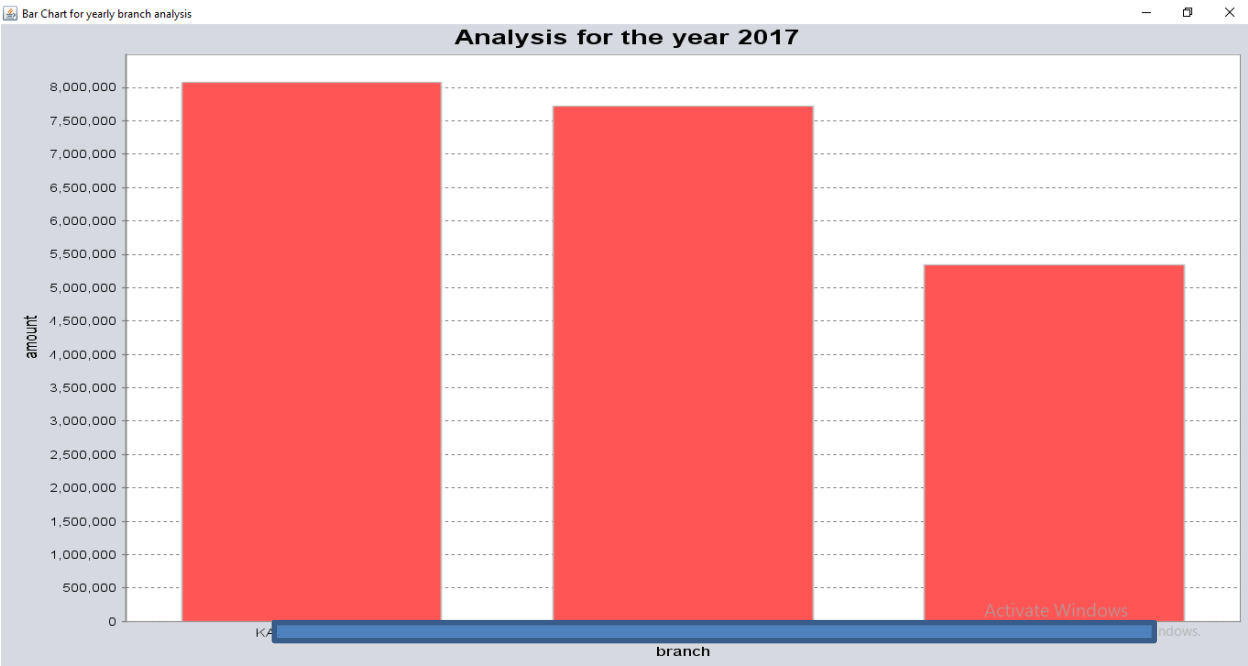


Figure 25: Branch Income Analysis for the year 2017

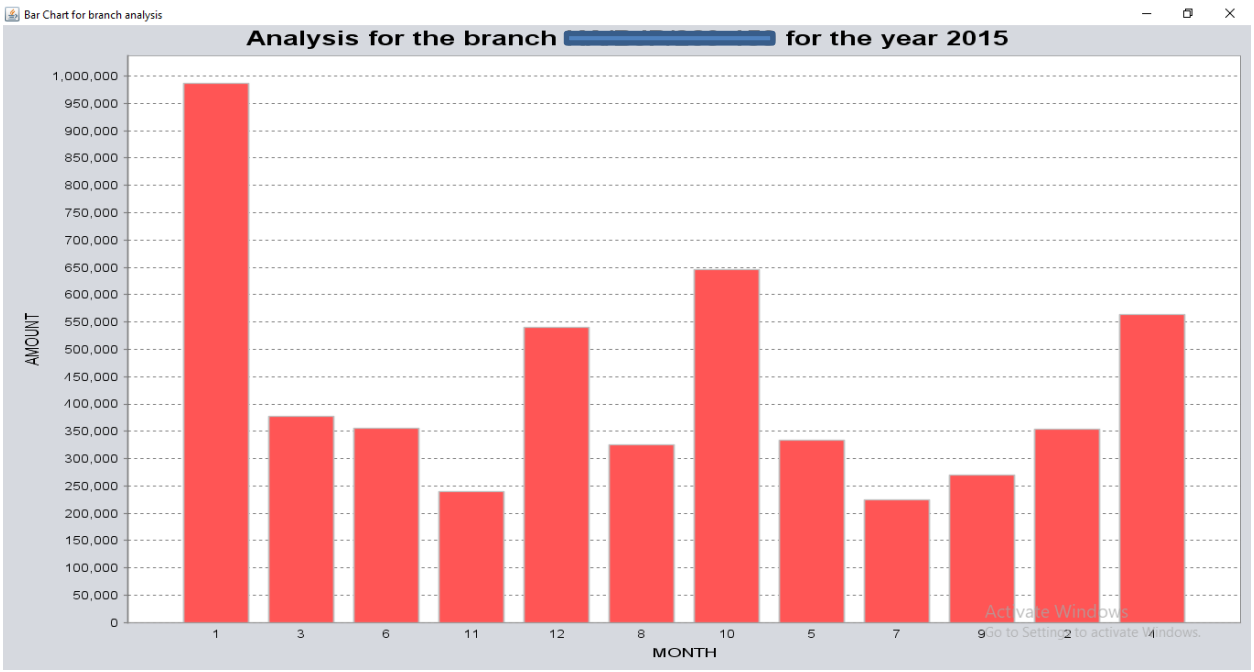


Figure 26: Monthly Income Analysis for the branch ***** for the year 2015

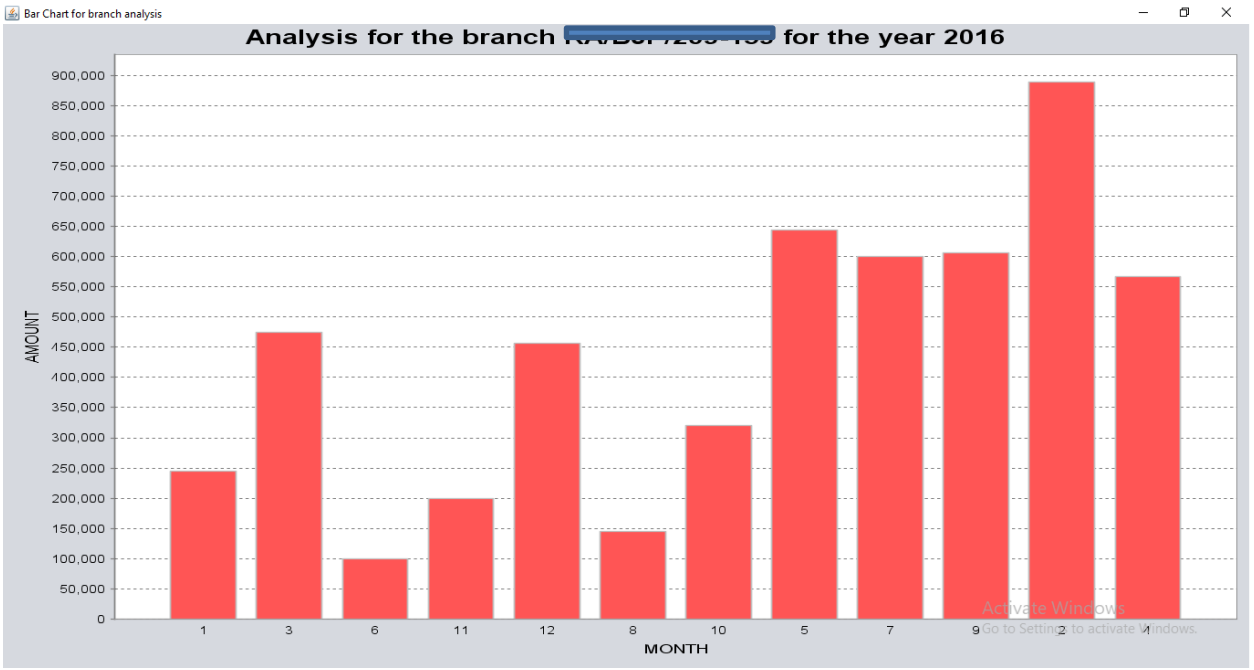


Figure 27: Monthly Income Analysis for the branch ***** for the year 2016

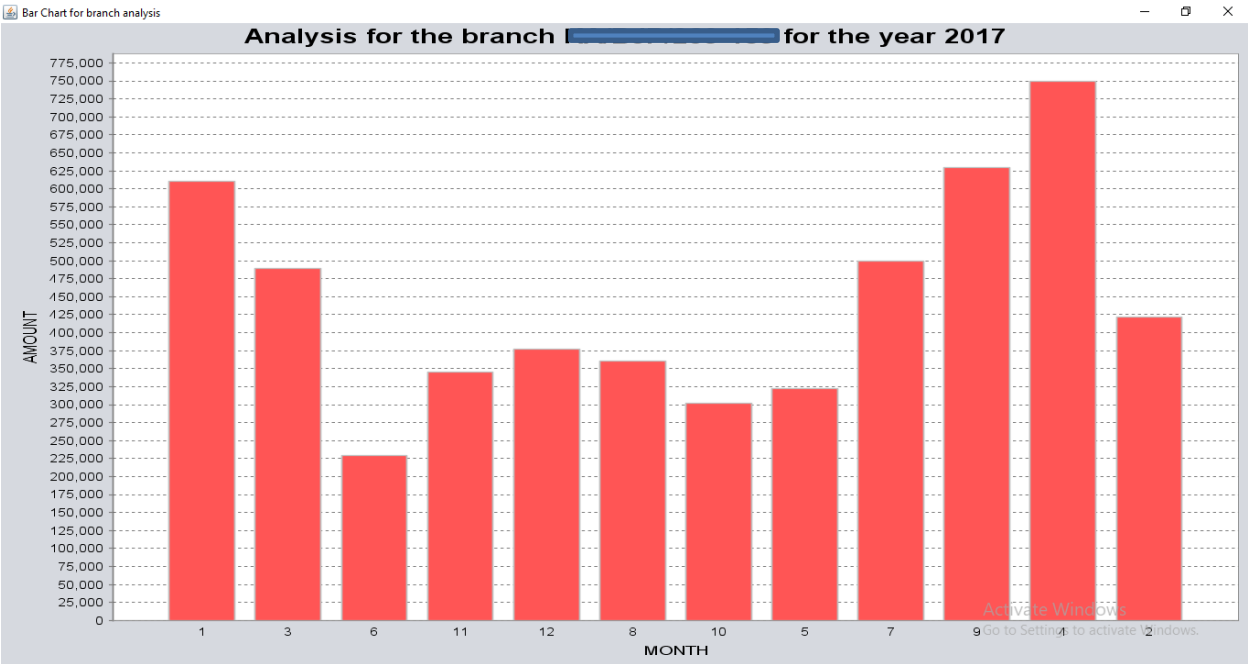


Figure 28: Monthly Income Analysis for the branch ****** for the year 2017

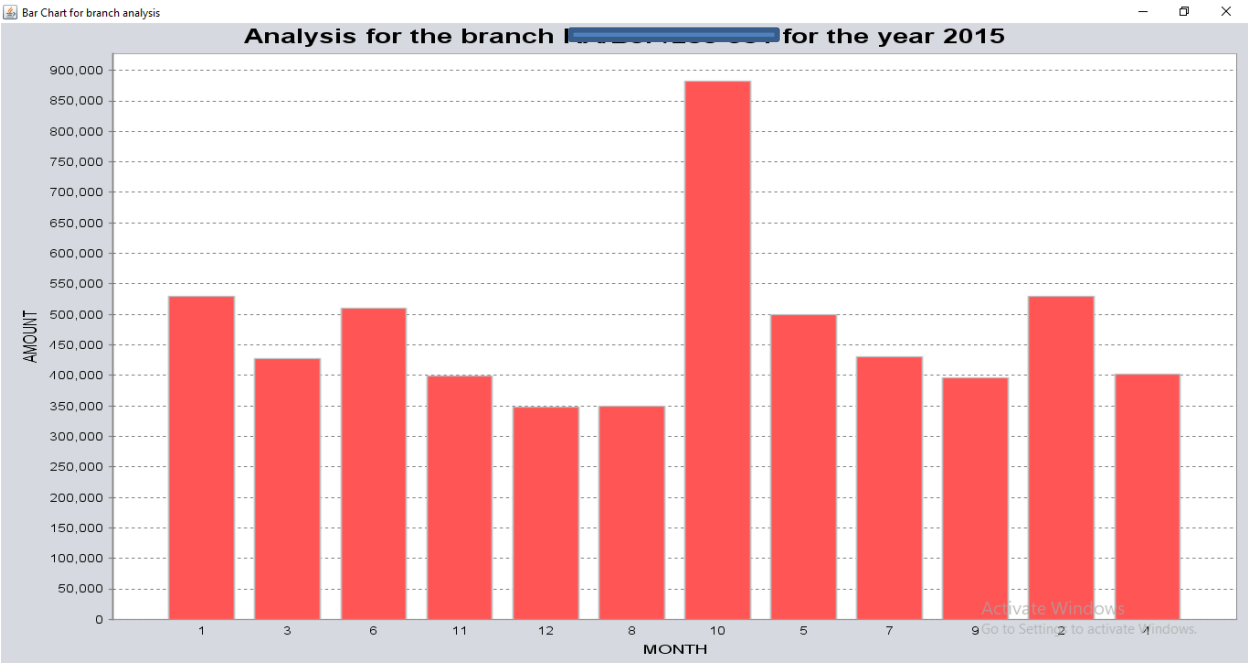


Figure 29: Monthly Income Analysis for the branch ****** for the year 2015

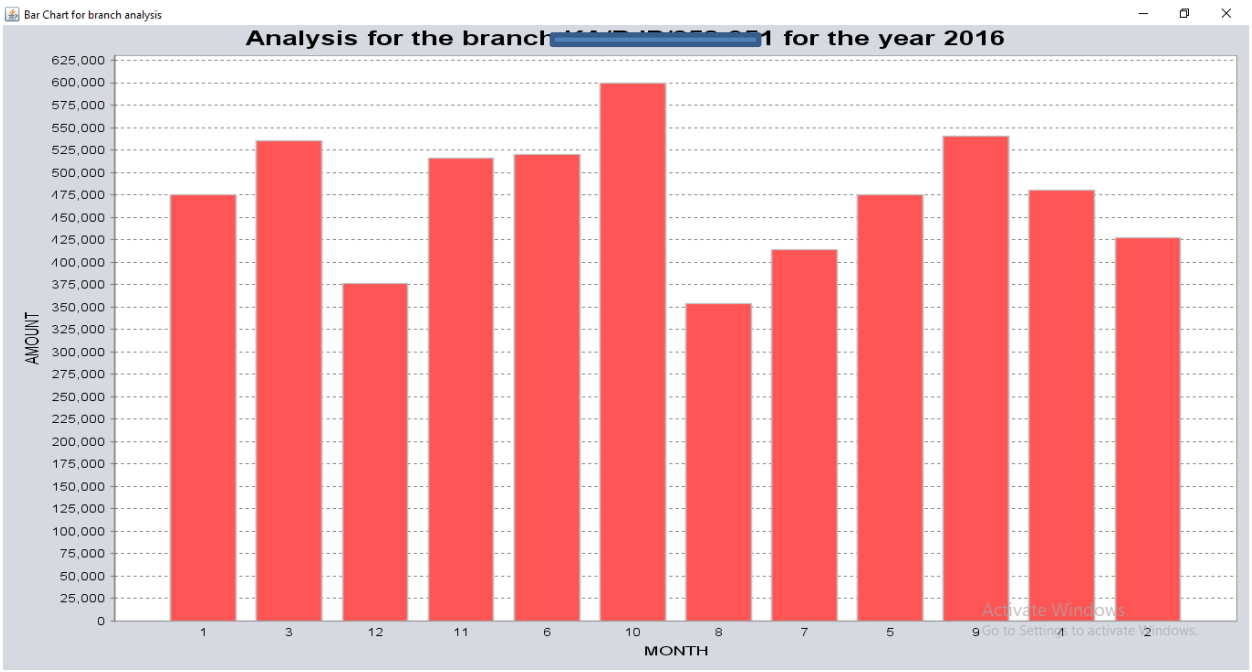


Figure 30: Monthly Income Analysis for the branch ***** for the year 2016

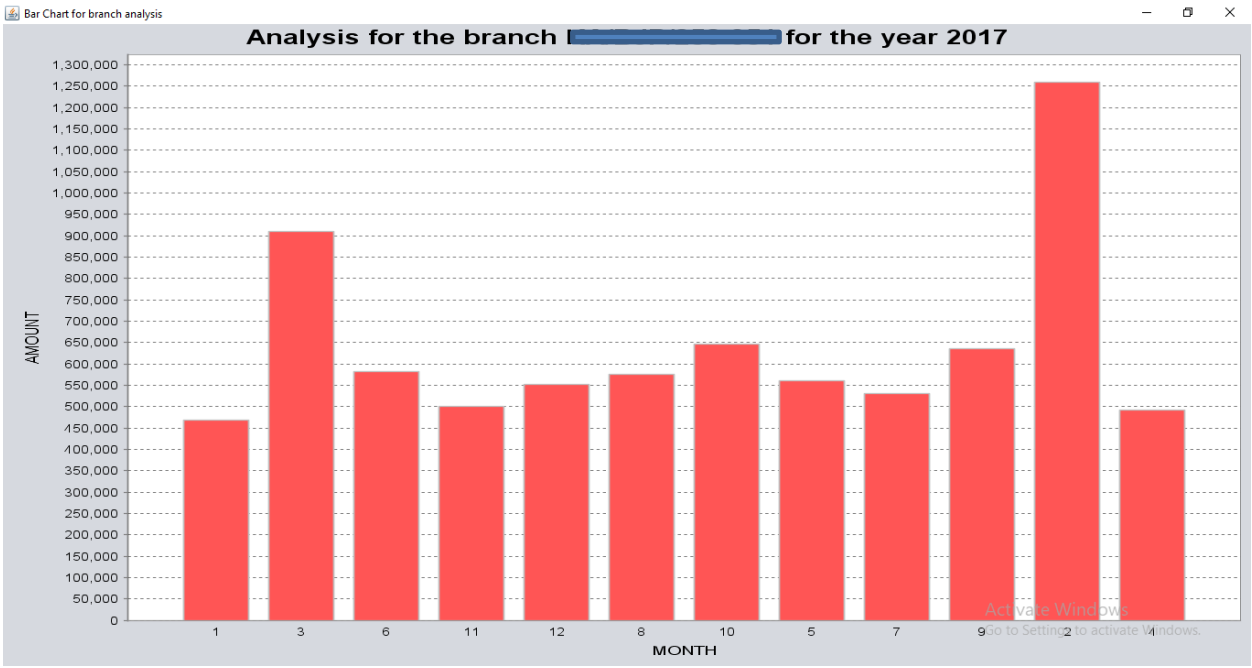


Figure 31: Monthly Income Analysis for the branch ***** for the year 2017

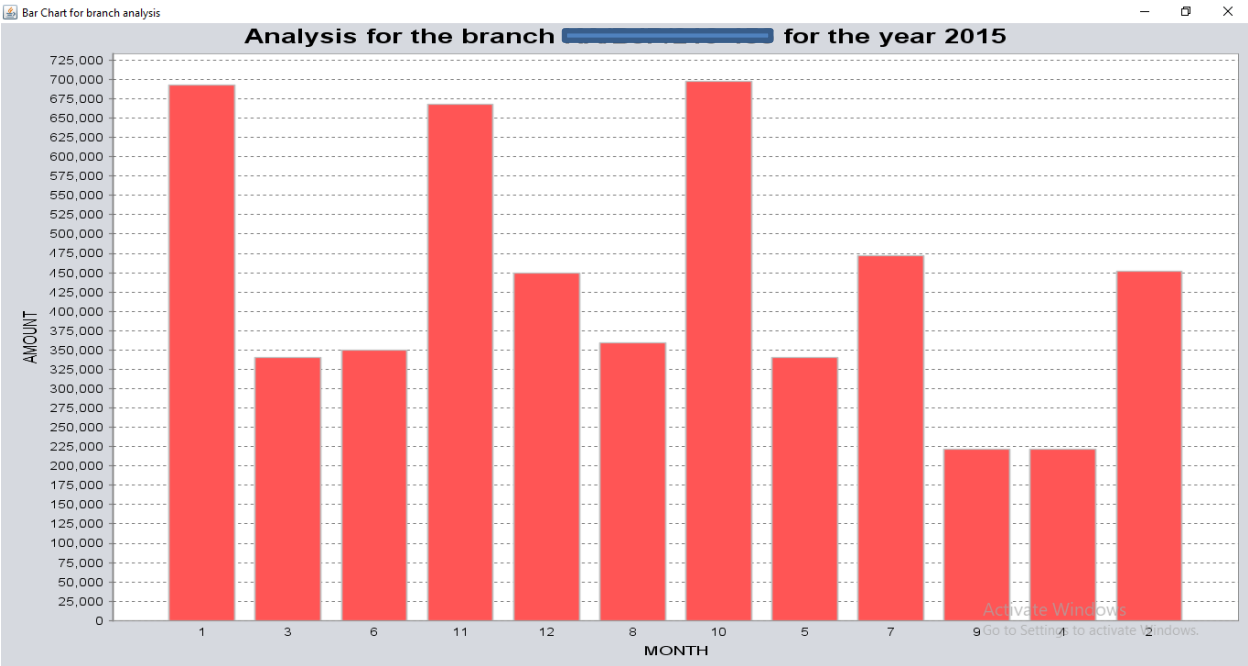


Figure 32: Monthly Income Analysis for the branch ***** for the year 2015

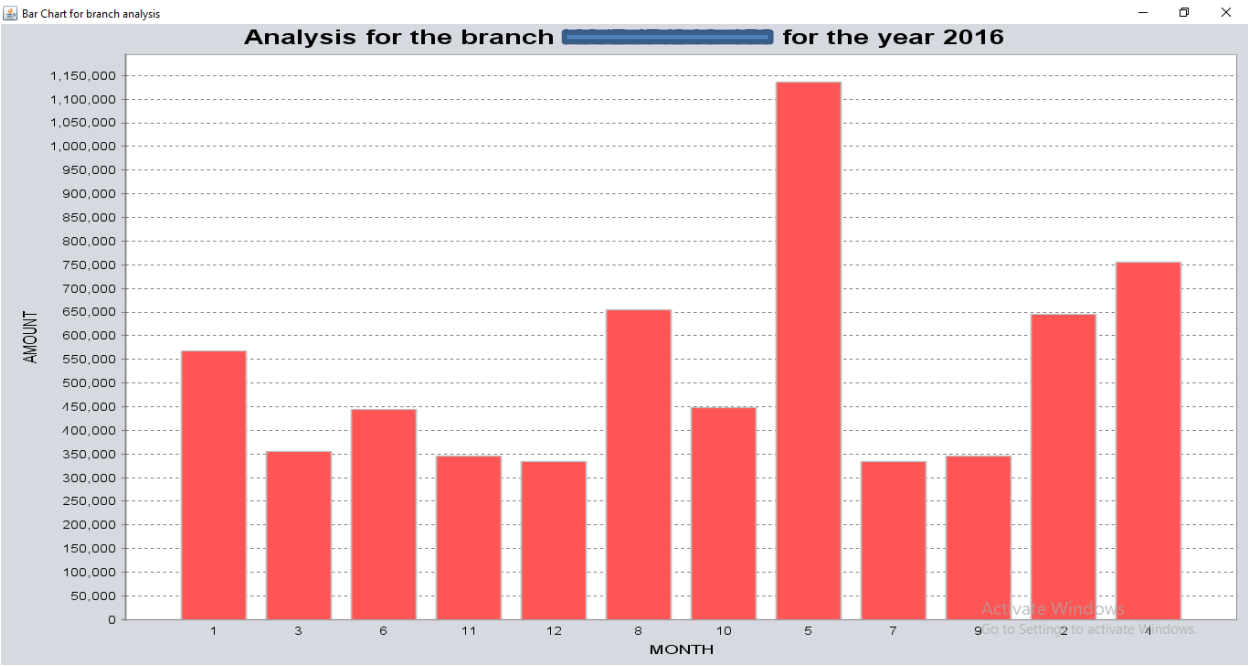


Figure 33: Monthly Income Analysis for the branch ****** for the year 2016

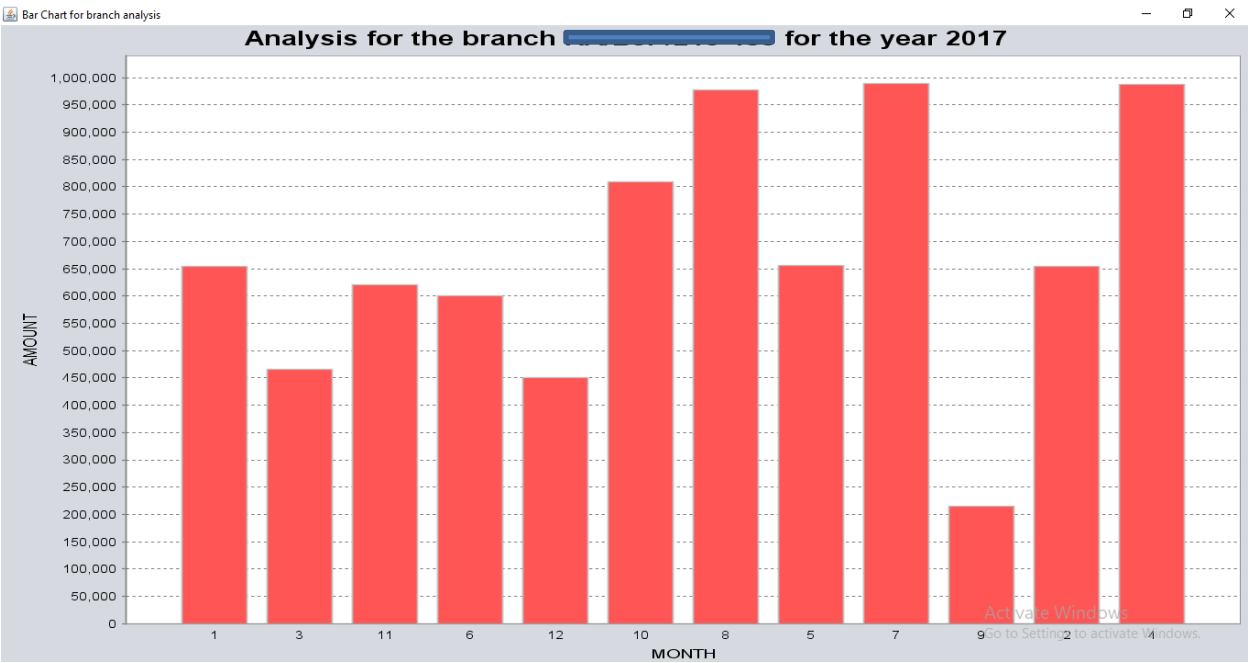


Figure 34 :Monthly Income Analysis for the branch ***** for the year 2017

Submission Date:28/04/2018