**POS Source Code & Architecture Details**

**POS System Architecture**

POS is built using the MVC architecture. JavaFX framework is used to implement the MVC architecture conventions.

Front End

Views

(Using .fxml files)

Back End

PostgreSQL DB

Data Access Objects

Controllers use Models & DAOs to

Process requests received from Views

*POS System Architecture*

**POS View Panels**

Side Tab Pane

Calculator

**POS view panels**

**1. Main Panel -**

This is Base of whole POS application. It provides link to other panels using menu bar on the top to layout the application. This panel also does the main sale for the customer by providing payment gateway etc.

View – src.sales.main\_sales\_test.fxml

Controller – src.sales.main\_Controller.java

**2. Main Hamburger Menu –**

This is the main menu for navigation inside the application. It connects to the other parts of the application. It’s attached to every scene of the application

**POS Navigation Mechanism**

**Controller From**

Most of the variables used by all the controllers are either come from database or used from main panel file.

**Target Controller**

Here also the variables are updated from the database or are used from the main pane files.

**Note –** In POS the data used in the application is either come from the database or used from the static variables in the main panel java file. This removes the redundancy of the data and at the same time ensures data integrity and security.

**Kalculate Source Code Major Packages Details**

1. src.sales
2. src.register
3. src.application
4. src.customer
5. src.database
6. src.extraTest
7. src.maintainence

**1. src.sales**

It's the root package of POS application. It contains the main\_Window.java class which contains the main() function of the application. The classes Under this package are following:-

1. **main\_Window.java**

This contains the main() function of POS application.

1. **Main.java**

This file contains the “Stock” scene of the application to serve the Stock functionality.

**2. src.application**

This package contains all the classes used for our testing purposes and XML files that are used for giving permissions and privileges to a user in the application.

**3. src.customers**

This package contain the “Customers” section of the application along with its related functionalities.

Files under this package are:-

1. **C\_controller.java**

It is the controller file of the “Customer” section with functionalities like add customer, view customer and all the customer related reports.

**4. src.database**

This package contains all the functions that are used to connect and query with the database. Using files from this package one can interact with database to connect, query and update with the database. The files used in this are:-

1. **dbfunc\_other.java**

Using the object of this class a user can connect, update and query the database. All the other files are supporting entities for this file.

**5. src.extraTest**

This package contains supporting files for main\_Window.java file for creation of the categories and product panels.

**6. src.maintenance**

This package contains the supporting files for the “Maintenance” functionality of the system.

**7. src.register**

This package contains java controller for three functionalities named “Edit Sale”, “Customer Payment”, “Close Cash” and “Payments”. Classes under this package are:-

* 1. **editcontroller1.java**

This is the file for the edit sale functionality of the system.

* 1. **PayMentRegister.java**

This is the file for the payments functionality of the system.

**3.**  **closeCash.java**

This is the main file for the close cash functionality of the system.

**4.**  **customer\_payment.java**

This is the main file for the customer payment functionality of the system.

**POS Libs Used**

1. jfeonix
2. postgersql-jdbc

**1. jfeonix -**

This library is used for providing material design components to the application like tabs, buttons etc. which are stylized in material design.

**2. Postgresql-JDBC -**

This library provides functionality of accessing postgresql Database from java.

**Back End Database Structure of POS**

List of all the relations and their description can be found in attached text file with the documentation.

**Installation and setup procedure**

To start, first the project should be imported in the eclipse using File->Import. In the next menu select “Import from zip” and provide the zip address there.

Then one has to configure the build path of the project by right click on the project in project explorer and clicking on Build Path-> Configure Build Path.

There in the External Libraries section provide the path of jfeonix and postgresql jar files and the project is up and running.

TO CREATE THE DATABASE:

Using the database dump provided, run command:-

psql database\_name < path/dump;