**Designing The BillingFrame**

****

**STEPS TO BE DONE IN** **BillingFrame**

In the **BillingFrame** we need to do following steps:

1. Allow the user to **logout**
2. Write code for the "**Generate Bill**" Button . When this button is clicked it should:
   1. **Create a new object of Order POJO and initialize it.**
   2. **Add the Order details to the database by calling the method addOrder( ) of OrderDAO**
   3. **If record is successfully added it should display the message "Order Created Successfully" .**
   4. **It should handle all the exceptions also**

4. Write code for the "**Back**" Button . When this button is clicked it should:

a**. Dispose the current frame and open the ReceptionistOptionsFrame**

**THE TABLES USED IN BillingFrame**

1. **Orders**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| **ORDER\_ID** | **Varchar2(10)** | **Contains id of the order** |
| **P\_ID** | **Varchar2(10)** | **Contains id of the product** |
| **P\_NAME** | **Varchar2(10)** | **Contains the product name** |
| **P\_COMPANYNAME** | **Varchar2(10)** | **Contains the product company name** |
| **P\_PRICE** | **NUMBER(6,2)** | **Contains Product price** |
| **OUR\_PRICE** | **NUMBER(6,2)** | **Contains our price** |
| **P\_TAX** | **NUMBER(3)** | **Contains tax on the product** |
| **QUANTITY** | **NUMBER(6)** | **Contains quantity of the product** |
| **USERID** | **Varchar2(20)** | **Contains userid by which order is generated** |

**THE POJO CLASSES USED IN BillingFrame**

1. The **Order** POJO
2. The **Product** POJO

**THE DAO CLASSES USED IN BillingFrame**

1. The **OrderDAO**
2. The **ProductDAO**

**HOW TO ADD ORDERS TO THE DATABASE:**

To Add a Order in the database we need to insert record in the Order table.

To do this we will create a method called **addOrder( )** in the **OrderDAO**. The prototype of the method is:

***public static boolean addOrder(ArrayList<ProductPojo> al, String uid)throws SQLException***

This method will take 2 arguments:

**1. The Userid by which the order has been generated and the arraylist of Product**

**WRITING THE CODE FOR BUTTON GENERATE BILL IN BillingFrame**

1. **First it will store product objects in the arraylist**
2. **After it it will call method addOrder and it will take two arguments ArrayList<ProductPojo>, And order-id.**

***Prototype-***

***public static boolean addOrder(ArrayList<ProductPojo> al, String s)throws SQLException***

1. **After successfully bill created it will open the ShowOrdersFrame and dispose the current frame**

**WRITING THE CODE FOR BUTTON BACK IN BillingFrame**

1. **It will dispose the current frame and open the ReceptionistOptionsFrame**