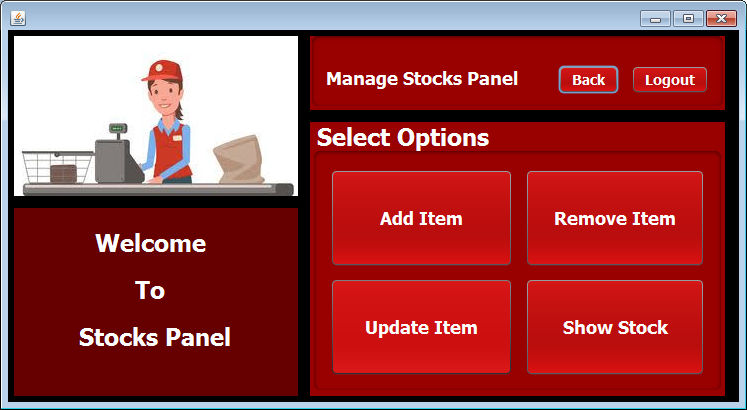
**Designing The AddStocksFrame**

****

**STEPS TO BE DONE IN** **AddStocksFrame**

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

1. Allow the user to **logout**
2. Generate a product id
3. Generate the barcode for that product id
4. And store to product in the database
5. Write code for the "**Generate Product-id**":
   1. Same as generating employees id
6. Code for generating **Barcode jpg. :**
   1. We will use an **external library** for doing this
      1. **org.krysalis.barcode4j.impl.code128.Code128Bean**
      2. **org.krysalis.barcode4j.output.bitmap.BitmapCanvasProvider**
   2. Above two classes will be used to generate a barcode image.
   3. We will create a method called **createImage().** It will take two arguments
      1. The **image name** which is the file name of the image
      2. And the **string** whose image has to be created
   4. We will pass our **product id** as image name as well as our image string so that the image of our product id will be created.

**THE TABLES USED IN AddStocksFrame**

**1. PRODUCT**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| **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** |

**THE POJO CLASSES USED IN AddStocksFrame**

1. The **Product** POJO

**THE DAO CLASSES USED IN AddStocksFrame**

1. The **ProductDAO**

We will call the method **getProductId( )** of **ProductDAO** to get the next Product-Id.

Same as in **AddEmployeesFrame**

Now we have to create the **addProduct( )** method in the **ProductDAO** to add Product in the **PRODUCT** table.

The prototype of the method is:

***public static boolean addproduct(ProductPojo p)throws SQLException***