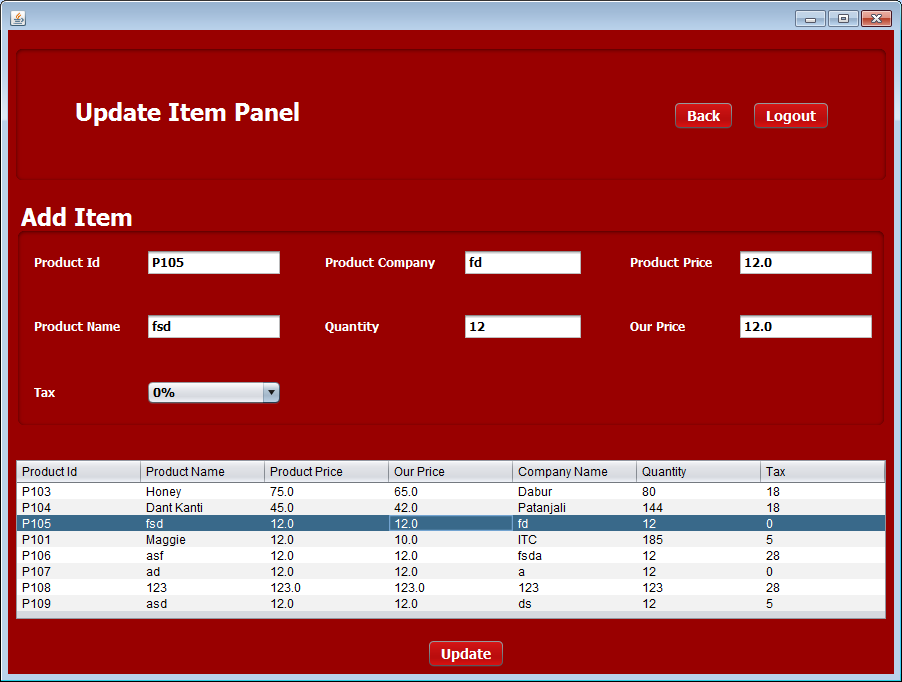
**Designing The UpdateStocksFrame**



**STEPS TO BE DONE IN** **UpdateStocksFrame**

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

1. Allow the user to **logout**
2. Handle the constructor so that it loads all the Product data from the database. To do this we need to call the method **getAllProductDetails ( )** of the class **ProductDAO**.
3. Handle the "**Back**" Button

**THE POJO CLASSES USED IN UpdateStocksFrame**

1. The **Product** POJO

**THE DAO CLASSES USED IN UpdateStocksFrame**

1. The **ProductDAO**

**HOW TO LOAD AND SHOW ALL STUDENT EXAM DETAILS:**

To do this we need to create a method called **getAllProductDetails( )** in the class **ProductDAO**. This method should fetch and return details of all the Products from the table **PRODUCT.** This return value is a list of **Product** POJO objects . So the prototype of this method will be:

***public static ArrayList<Product> getAllproductDetails()throws SQLException***

**DISPLAYING PRODUCT DETAILS IN JTABLE:**

To display details of all the Products we have to use another UI control called **JTable**. The **JTable** control allows us to display multiple rows of data. So we need to create a method called **showDataInTable()** in the **UpdateStocksFrame** which does the following:

**a. Calls the method getAllData( ) of the class ProductDAO.**

**b. If no id is returned , it displays the message "No Product Has Been Added"**

**c. Otherwise it iterates over the Product list and displays it in the JTable**

**d. It also handles SQLException which the method getAllProductDetails( ) of the class ProductDAO can throw**.

**DISPLAYING PRODUCT DETAILS IN TEXTFIELDS:**

1. We have to handle the row click of the JTable. By which we can display the data in jTextFields
2. For doing this we need to iterate the row which has been clicked by the user and fetch the data of each column and display it in the required fields.

**HANDLING THE CONSTRUCTOR:**

From the body of constructor we would call the above method so as to show the students details as soon as the frame opens

**WRITING THE CODE FOR BUTTON UPDATE IN UpdateStocksFrame**

1. **On the click of the update button we have to call the method updateProduct() in class ProductDAO.**
2. **It will take the updated product object and save it to the Product Table.**