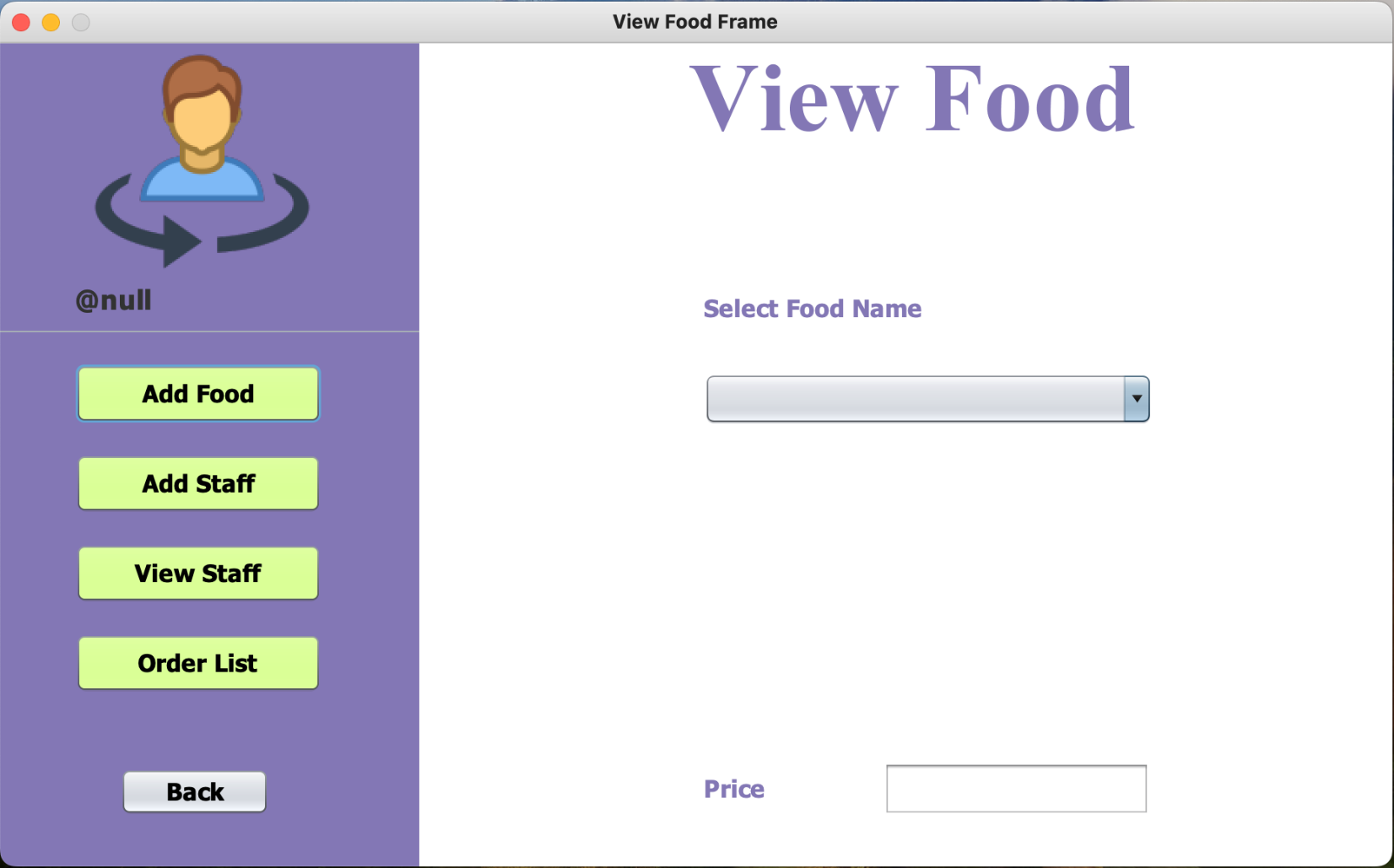
**Java SE Project**

YumXpress Application

***ViewFood Frame Documentation***



**2. STEPS TO BE DONE IN** **ViewFoodFrame**

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

1. **Adding a method called getProductDetailsByCompany() in the ProductDAO class This method will accept a companyId as argument and return the details of all the Products of that company as key-value pair.**
2. **Show all the product/food names in the JComboBox as soon as the ViewFoodFrame loads**
3. **Also show the price and Image of the selected food on the Frame by setting event handler for JComboBox**
4. **Coding of getProductDetailsByCompany() method:**

**public static Map<String,ProductPojo> getProductDetailsByCompany(String companyId) throws SQLException, IOException {**

**Connection conn = DBConnection.getConnection();**

**PreparedStatement ps = conn.prepareStatement("select \* from products where company\_id=?");**

**ps.setString(1, companyId);**

**ResultSet rs = ps.executeQuery();**

**HashMap<String, ProductPojo> productList = new HashMap<>();**

**while (rs.next()) {**

**ProductPojo product = new ProductPojo();**

**product.setCompanyId(rs.getString("company\_id"));**

**product.setProductId(rs.getString("product\_id"));**

**product.setProductName(rs.getString("product\_name"));**

**product.setProductPrice(rs.getDouble("product\_price"));**

**//convert the blob to image :**

**// Fetch the image data as InputStream**

**InputStream inputStream = rs.getBinaryStream("product\_image");**

**// Convert InputStream to BufferedImage**

**BufferedImage bufferedImage = ImageIO.read(inputStream);**

**// Convert BufferedImage to Image**

**Image image = bufferedImage;**

**product.setProductImage(image);**

**productList.put(product.getProductName(), product);**

**}**

**return productList;**

**}**

**B. Coding for ViewFoodFrame Constructor :**

**public class ViewFoodFrame extends javax.swing.JFrame {**

**private JFrame showFrame;**

**private Map<String, ProductPojo> productDetails;**

**private ProductPojo product;**

**public ViewFoodFrame() {**

**initComponents();**

**this.setLocationRelativeTo(null);**

**lblOwnerProfile.setText("@" + OwnerProfile.getOwnerName());**

**loadProductDetails();**

**}**

**Coding for loadProductDetails() method :**

**private void loadProductDetails() {**

**try {**

**System.out.println("company id : " + OwnerProfile.getCompanyId());**

**productDetails = ProductDao.getProductDetails(OwnerProfile.getCompanyId());**

**if(productDetails.isEmpty()){**

**JOptionPane.showMessageDialog(this, "No foods added by "+OwnerProfile.getCompanyName());**

**return;**

**}**

**loadProductNames();**

**} catch (IOException | SQLException ex) {**

**ex.printStackTrace();**

**Logger.getLogger(ViewFoodFrame.class.getName()).log(Level.SEVERE, null, ex);**

**}**

**}**

**Coding for loadProductNames() method :**

**private void loadProductNames() {**

**Set<String> productNames = productDetails.keySet();**

**Iterator<String> it = productNames.iterator();**

**while (it.hasNext()) {**

**jcProductNames.addItem(it.next());**

**}**

**System.out.println("names :" + productNames);**

**}**

**Coding for actionPerformed() of JComboBox() method :**

**private void jcProductNamesActionPerformed(java.awt.event.ActionEvent evt) {**

**String selectFoodName = jcProductNames.getSelectedItem().toString();**

**showProductDetails(selectFoodName);**

**}**

**Coding for showProductDetails() of JComboBox() method :**

**private void showProductDetails(String productName) {**

**product = productDetails.get(productName);**

**txtProductPrice.setText(String.valueOf(product.getProductPrice()));**

**ImageIcon imageIcon = new ImageIcon(product.getProductImage()**

**.getScaledInstance(lblProductImage.getWidth(), lblProductImage.getHeight(), Image.SCALE\_SMOOTH));**

**lblProductImage.setIcon(imageIcon);**

**}**

**Coding for handling all the options :**

**private void btnViewStaffActionPerformed(java.awt.event.ActionEvent evt) {**

**showFrame = new ViewDeliveryStaffFrame();**

**showFrame.setVisible(Boolean.TRUE);**

**this.dispose();**

**}**

**private void btnOrderListActionPerformed(java.awt.event.ActionEvent evt) {**

**// TODO add your handling code here:**

**showFrame = new OrderListFrame();**

**showFrame.setVisible(Boolean.TRUE);**

**this.dispose();**

**}**

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {**

**showFrame=new SellerOptionFrame();**

**showFrame.setVisible(true);**

**this.dispose();**

**}**