

Application Development(APD545NDD.12037.2251)

Full Name: Radmehr Behzadfar

Id: 148786221

Seneca email: [rbehzadfar@myseneca.ca](mailto:rbehzadfar@myseneca.ca)

Output: ( there are stage by stage, checking different functionalities)

**Grocery Cart**

Select Item: Bread Quantity: 10 Units: 10

Item Name: Bread Unit Price: 2.25

Add to Cart Remove from Cart Save Cart Load Cart Load Cart from DB Check Out

Available Items:

Item Name	Unit Price
Apple	1.5
Banana	0.75
Bread	2.25
Milk	3.0
Eggs	4.0

Shopping Cart:

Item Name	Unit Price	Quantity	Total Price
No content in table			

Total Amount: 0.00

Grocery Cart

Select Item: 

Rice

Quantity:

Units: 1

Item Name: Rice

Unit Price: 5.00

Add to Cart

Remove from Cart

Save Cart

Load Cart

Load Cart from DB

Check Out

Available Items:

Item Name	Unit Price	
Apple	1.5	
Banana	0.75	
Bread	2.25	
Milk	3.0	
Eggs	4.0	

Shopping Cart:

Item Name	Unit Price	Quantity	Total Price	
Apple	1.5	5	7.5	
Banana	0.75	10	7.5	
Bread	2.25	1	2.25	
Eggs	4.0	7	28.0	
Rice	5.0	1	5.0	

Total Amount: 50.25

Grocery Cart

Select Item: Eggs

Quantity:

Units: 7

Item Name: Eggs

Unit Price: 4.00

Add to Cart

Remove from Cart

Save Cart

Load Cart

Load Cart from DB

Check Out

Available Items:

Item Name	Unit Price	
Bread	2.25	
Milk	3.0	
Eggs	4.0	
Chicken	8.5	
Rice	5.0	

Shopping Cart:

Item Name	Unit Price	Quantity	Total Price	
Apple	1.5	5	7.5	
Banana	0.75	10	7.5	
Bread	2.25	1	2.25	
Eggs	4.0	7	28.0	
Rice	5.0	1	5.0	

Total Amount: 50.25

# Grocery Cart

Select Item: Banana Quantity:  Units: 10

Item Name: Banana Unit Price: 0.75

Add to Cart

Remove from Cart

Save Cart

Load Cart

Load Cart from DB

Check Out

Available Items:

Item Name	Unit Price	
Bread	2.25	
Milk	3.0	
Eggs	4.0	
Chicken	8.5	
Rice	5.0	

Shopping Cart:

Item Name	Unit Price	Quantity	Total Price	
Apple	1.5	5	7.5	
Banana	0.75	10	7.5	
Eggs	4.0	7	28.0	
Rice	5.0	1	5.0	

Total Amount: 48.00

Grocery Cart

Select Item: Banana

Quantity:

Units: 10

Item Name: Banana

Unit Price: 0.75

Add to Cart

Remove from Cart

Save Cart

Load Cart

Load Cart from DB

Check Out

Available Items:

Item Name	Unit Price	
Bread	2.25	
Milk	3.0	
Eggs	4.0	
Chicken	8.5	
Rice	5.0	

Shopping Cart:

Item Name	Unit Price	Quantity	Total Price	
Apple	1.5	5	7.5	
Banana	0.75	10	7.5	
Eggs	4.0	7	28.0	
Rice	5.0	1	5.0	

Total Amount: 48.00

Message

Message

i

Cart saved to DB and file.

OK

Grocery Cart

Select Item:

Banana

Quantity:

Units: 10

Item Name:

Banana

Unit Price:

0.75

Add to Cart

Remove from Cart

Save Cart

Load Cart

Load Cart from DB

Check Out

Available Items:

Item Name	Unit Price	
Bread	2.25	
Milk	3.0	
Eggs	4.0	
Chicken	8.5	
Rice	5.0	

Shopping Cart:

Item Name	Unit Price	Quantity	Total Price	
Apple	1.5	5	7.5	
Banana	0.75	10	7.5	
Eggs	4.0	7	28.0	
Rice	5.0	1	5.0	

Total Amount: 48.00

Confirmation

Confirmation

?

Are you done with your groceries?

OK

Cancel



Grocery Cart



Select Item:

Banana



Quantity:

Units: 10

Item Name: Banana

Unit Price: 0.75

Add to Cart

Remove from Cart

Save Cart

Load Cart

Load Cart from DB

Check Out

Available Items:

Item Name	Unit Price	
Bread	2.25	
Milk	3.0	
Eggs	4.0	
Chicken	8.5	
Rice	5.0	

Shopping Cart:

Item Name	Unit Price	Quantity	Total Price	
No content in table				

Total Amount: 0.00

Grocery Cart

Select Item: Banana

Quantity:

Units: 10

Item Name: Banana

Unit Price: 0.75

Add to Cart

Remove from Cart

Save Cart

Load Cart

Load Cart from DB

Check Out

Available Items:

Item Name	Unit Price	
Bread	2.25	
Milk	3.0	
Eggs	4.0	
Chicken	8.5	
Rice	5.0	

Shopping Cart:

Item Name	Unit Price	Quantity	Total Price	
No content in table				

Total Amount: 0.00

Message

Message

Cart loaded from DB.

OK



Grocery Cart

Select Item: Banana

Quantity:

Units: 10

Item Name: Banana

Unit Price: 0.75

Add to Cart

Remove from Cart

Save Cart

Load Cart

Load Cart from DB

Check Out

Available Items:

Item Name	Unit Price	
Bread	2.25	
Milk	3.0	
Eggs	4.0	
Chicken	8.5	
Rice	5.0	

Shopping Cart:

Item Name	Unit Price	Quantity	Total Price	
Apple	1.5	5	7.5	
Banana	0.75	10	7.5	
Eggs	4.0	7	28.0	
Rice	5.0	1	5.0	

Total Amount: 48.00

Grocery Cart

Select Item:

Banana

Quantity:

Units: 10

Item Name: Banana

Unit Price: 0.75

Add to Cart

Remove from Cart

Save Cart

Load Cart

Load Cart from DB

Check Out

Available Items:

Item Name	Unit Price	
Bread	2.25	
Milk	3.0	
Eggs	4.0	
Chicken	8.5	
Rice	5.0	

Shopping Cart:

Item Name	Unit Price	Quantity	Total Price	
Apple	1.5	5	7.5	
Banana	0.75	10	7.5	
Eggs	4.0	7	28.0	
Rice	5.0	1	5.0	

Total Amount: 48.00

Load Saved Cart (File)

1742182593094.ser

1742346410515.ser

1742346418871.ser

1742347084171.ser

1742347948993.ser

Load

Cancel

Load Saved Cart (File)

1742182593094.ser

1742346410515.ser

1742346418871.ser

1742347084171.ser

1742347948993.ser

Load

Cancel

**Grocery Cart**

Select Item:  Quantity:  Units: 10

Item Name: Banana Unit Price: 0.75

Available Items:

Item Name	Unit Price
Bread	2.25
Milk	3.0
Eggs	4.0
Chicken	8.5
Rice	5.0

Shopping Cart:

Item Name	Unit Price	Quantity	Total Price
Bread	2.25	10	22.5

Total Amount: 22.50

**Code:**

## Cart.java:

\*\*\*\*\*

## Workshop #06

Course: APD545 - Winter

Last Name: BEHZADFAR

First Name: RADMEHR

ID: 148786221

Section:NDD

This assignment represents my own work in accordance with Seneca Academic Policy.

Signature: RadmehrBehzadfar

Date:2025-03-29

\*\*\*\*\*/

```
package grocerystore;
```

```
import javafx.beans.binding.Bindings;
```

```
import javafx.beans.binding.DoubleBinding;
```

```
import javafx.collections.FXCollections;
```

```
import javafx.collections.ObservableList;
```

```
import javafx.fxml.FXML;
```

```
import javafx.fxml.FXMLLoader;
```

```
import javafx.scene.Scene;
```

```
import javafx.scene.control.Alert;
```

```
import javafx.scene.control.Button;
```

```
import javafx.scene.control.ButtonType;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.Slider;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.Parent;

import javafx.stage.Modality;

import javafx.stage.Stage;

import java.util.List;


public class Cart {

    @FXML private ComboBox<Item> itemsComboBox;

    @FXML private Slider quantitySlider;

    @FXML private Label purchaseUnitsLabel;

    @FXML private Label unitValueLabel;

    @FXML private Label unitPriceValueLabel;
```

```
@FXML private TableView<Item> itemsTableView;
```

```
@FXML private TableColumn<Item, String> itemNameColumn;
```

```
@FXML private TableColumn<Item, Number> unitPriceColumn;
```

```
@FXML private TableView<ItemInCart> cartTableView;
```

```
@FXML private TableColumn<ItemInCart, String> cartItemNameColumn;
```

```
@FXML private TableColumn<ItemInCart, Number> cartUnitPriceColumn;
```

```
@FXML private TableColumn<ItemInCart, Number> cartQuantityColumn;
```

```
@FXML private TableColumn<ItemInCart, Number> cartTotalPriceColumn;
```

```
@FXML private Button addButton;
```

```
@FXML private Button removeButton;
```

```
@FXML private Button saveCartButton;
```

```
@FXML private Button loadCartButton;
```

```
@FXML private Button loadCartFromDBButton;
```

```
@FXML private Button checkOutButton;
```

```
@FXML private Label totalValueLabel;
```

```
private Model model;
```

```
private ObservableList<ItemInCart> cartObservableList =  
FXCollections.observableArrayList();
```

```
private User loggedInUser;
```

```
@FXML
```

```
private void initialize() {
```

```
    model = new Model();
```

```
    String csvPath = getClass().getResource("/ItemsMaster.csv").getPath();
```

```
    model.loadData(csvPath);
```

```
    setupAction();
```

```
}
```

```
public void setupAction() {
```

```
    itemsComboBox.setItems(model.getItemsObservableList());
```

```
    itemsComboBox.getSelectionModel().clearSelection();
```

```
    purchaseUnitsLabel.textProperty().bind(Bindings.format("%.0f",  
quantitySlider.valueProperty()));
```

```
unitValueLabel.textProperty().bind(Bindings.createStringBinding(() -> {
```

```
    Item selectedItem = itemsComboBox.getValue();
```

```
    return selectedItem != null ? selectedItem.getName() : "";
```

```
}, itemsComboBox.valueProperty()));
```

```
unitPriceValueLabel.textProperty().bind(Bindings.createStringBinding(() -> {
```

```
    Item selectedItem = itemsComboBox.getValue();
```

```
    return selectedItem != null ? String.format("%.2f", selectedItem.getUnitPrice()) : "";
```

```
}, itemsComboBox.valueProperty()));
```

```
    itemNameColumn.setCellValueFactory(cellData ->  
cellData.getValue().nameProperty());
```

```
    unitPriceColumn.setCellValueFactory(cellData ->  
cellData.getValue().unitPriceProperty());
```

```
itemsTableView.setItems(model.getItemsObservableList());
```

```
cartTableView.setItems(cartObservableList);
```

```
    cartItemNameColumn.setCellValueFactory(data ->  
data.getValue().itemNameProperty());
```

```
    cartUnitPriceColumn.setCellValueFactory(data -> data.getValue().unitPriceProperty());
```



```
cartQuantityColumn.setCellValueFactory(data -> data.getValue().quantityProperty());
```

```
cartTotalPriceColumn.setCellValueFactory(data ->  
data.getValue().totalPriceProperty());
```

```
addButton.setOnAction(event -> {
```

```
    Item selectedItem = itemsComboBox.getSelectionModel().getSelectedItem();
```

```
    if (selectedItem != null) {
```

```
        int quantity = (int) quantitySlider.getValue();
```

```
        ItemInCart itemInCart = new ItemInCart(selectedItem.getName(),  
selectedItem.getUnitPrice(), quantity);
```

```
        cartObservableList.add(itemInCart);
```

```
    }
```

```
});
```

```
DoubleBinding totalBinding = Bindings.createDoubleBinding(() ->
```

```
    cartObservableList.stream().mapToDouble(ItemInCart::getTotalPrice).sum(),
```

```
    cartObservableList
```

```
);
```

```
totalValueLabel.textProperty().bind(Bindings.format("%.2f", totalBinding));
```

```
cartTableView.getSelectionModel().selectedItemProperty().addListener((obs,  
oldSelection, newSelection) -> {
```

```
    if (newSelection != null) {
```

```
        for (int i = 0; i < model.getItemsObservableList().size(); i++) {
```

```
            if  
(model.getItemsObservableList().get(i).getName().equals(newSelection.getItemName())) {
```

```
                itemsComboBox.getSelectionModel().select(i);
```

```
                quantitySlider.setValue(newSelection.getQuantity());
```

```
                break;
```

```
            }
```

```
        }
```

```
    }
```

```
});
```

```
removeButton.setOnAction(event -> {
```

```
    int selectedIndex = cartTableView.getSelectionModel().getSelectedIndex();
```

```
        if (selectedIndex >= 0) {

            cartObservableList.remove(selectedIndex);

        }

    });

    saveCartButton.setOnAction(event -> {

        boolean successDB = DBHelper.saveCart(new
        java.util.ArrayList<>(cartObservableList), loggedInUser.getId());

        CartFileHandler.saveCart(new java.util.ArrayList<>(cartObservableList));

        if(successDB){

            new Alert(Alert.AlertType.INFORMATION, "Cart saved to DB and
            file.").showAndWait();

        } else {

            new Alert(Alert.AlertType.ERROR, "Failed to save cart to DB.").showAndWait();

        }

    });

    loadCartButton.setOnAction(event -> {
```

```
try {

    FXMLLoader loader = new
FXMLLoader(getClass().getResource("/SavedCart.fxml"));

    Parent root = loader.load();

    SavedCartController controller = loader.getController();

    controller.setMainController(this);

    Stage stage = new Stage();

    stage.initModality(Modality.APPLICATION_MODAL);

    stage.setScene(new Scene(root));

    stage.setTitle("Load Saved Cart (File)");

    stage.showAndWait();

} catch (Exception e) {

    e.printStackTrace();

}

});

loadCartFromDBButton.setOnAction(event -> {

    List<ItemInCart> loadedCart = DBHelper.loadCart(loggedInUser.getId());
```

```
        if (loadedCart != null) {

            loadCartData(loadedCart);

            new Alert(Alert.AlertType.INFORMATION, "Cart loaded from DB.").showAndWait();

        } else {

            new Alert(Alert.AlertType.ERROR, "No saved cart found in DB.").showAndWait();

        }

    });

    checkoutButton.setOnAction(event -> {

        Alert alert = new Alert(Alert.AlertType.CONFIRMATION, "Are you done with your groceries?");

        alert.showAndWait().ifPresent(response -> {

            if (response == ButtonType.OK) {

                cartObservableList.clear();

            }

        });

    });

}
```

```
public void loadCartData(List<ItemInCart> loadedItems) {  
  
    cartObservableList.clear();  
  
    cartObservableList.addAll(loadedItems);  
  
}
```

```
public void setLoggedInUser(User user) {  
  
    loggedInUser = user;  
  
}  
  
}
```

CartFileHandler.java:

/\*\*\*\*\*\*

Workshop #06

Course: APD545 - Winter

Last Name: BEHZADFAR

First Name: RADMEHR

ID: 148786221

Section:NDD

This assignment represents my own work in accordance with [Seneca Academic Policy](#).

Signature: RadmehrBehzadfar

Date:2025-03-29

\*\*\*\*\*/

```
package grocerystore;
```

```
import java.io.File;
```

```
import java.io.FileOutputStream;
```

```
import java.io.FileInputStream;
```

```
import java.io.ObjectOutputStream;
```

```
import java.io.ObjectInputStream;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
public class CartFileHandler {
```

```
    private static final String CARTS_DIR = "saved_carts";
```

```
    public static void saveCart(List<ItemInCart> cart) {
```

```

try {

    File dir = new File(CARTS_DIR);

    if (!dir.exists()) {

        dir.mkdir();

    }

    String filename = CARTS_DIR + File.separator + System.currentTimeMillis() + ".ser";

    ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream(filename));

    oos.writeObject(cart);

    oos.close();

} catch (Exception e) {

    e.printStackTrace();

}

}

```

```

@SuppressWarnings("unchecked")

```

```

public static List<ItemInCart> loadCart(File file) {

```

```

    try {

```



```
        ObjectInputStream ois = new ObjectInputStream(new FileInputStream(file));

        List<ItemInCart> loaded = (List<ItemInCart>) ois.readObject();

        ois.close();

        return loaded;

    } catch (Exception e) {

        e.printStackTrace();

        return null;

    }

}
```

```
public static List<File> listSavedCarts() {

    List<File> result = new ArrayList<>();

    File dir = new File(CARTS_DIR);

    if (dir.exists() && dir.isDirectory()) {

        File[] files = dir.listFiles((d, name) -> name.endsWith(".ser"));

        if (files != null) {

            for (File f : files) {
```

```
        result.add(f);

    }

}

}

return result;

}

}
```

DBHelper.java:

/\*\*\*\*\*\*

Workshop #06

Course: APD545 - Winter

Last Name: BEHZADFAR

First Name: RADMEHR

ID: 148786221

Section:NDD

This assignment represents my own work in accordance with Seneca Academic Policy.

Signature: RadmehrBehzadfar

Date:2025-03-29

```
*****/
```

```
package grocerystore;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
import java.sql.Statement;
```

```
import java.io.ByteArrayOutputStream;
```

```
import java.io.ByteArrayInputStream;
```

```
import java.io.ObjectOutputStream;
```

```
import java.io.ObjectInputStream;
```

```
import java.util.List;
```

```
public class DBHelper {
```

```
    private static final String DB_URL = "jdbc:sqlite:shopcart.db";
```

```
    static {
```

```
        try (Connection conn = getConnection(); Statement stmt = conn.createStatement()) {
```

```
String createUsers = "CREATE TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY  
AUTOINCREMENT, username TEXT UNIQUE, password TEXT)";
```

```
String createCarts = "CREATE TABLE IF NOT EXISTS carts (id INTEGER PRIMARY KEY  
AUTOINCREMENT, user_id INTEGER, cart_data BLOB, timestamp DATETIME DEFAULT  
CURRENT_TIMESTAMP)";
```

```
stmt.execute(createUsers);
```

```
stmt.execute(createCarts);
```

```
} catch(Exception e){
```

```
e.printStackTrace();
```

```
}
```

```
}
```

```
public static Connection getConnection() throws Exception {
```

```
return DriverManager.getConnection(DB_URL);
```

```
}
```

```
public static boolean registerUser(String username, String password) {
```

```
try (Connection conn = getConnection(); PreparedStatement pstmt =  
conn.prepareStatement("INSERT INTO users(username, password) VALUES(?,?)")) {
```

```
pstmt.setString(1, username);
```

```
pstmt.setString(2, password);
```

```
        pstmt.executeUpdate();

        return true;

    } catch (Exception e) {

        return false;

    }

}

public static User loginUser(String username, String password) {

    try (Connection conn = getConnection(); PreparedStatement pstmt =
conn.prepareStatement("SELECT id, username FROM users WHERE username=? AND
password=?")) {

        pstmt.setString(1, username);

        pstmt.setString(2, password);

        ResultSet rs = pstmt.executeQuery();

        if(rs.next()){

            return new User(rs.getInt("id"), rs.getString("username"));

        }

    } catch (Exception e) {

        e.printStackTrace();

    }

}
```

```
}

return null;

}

public static boolean saveCart(List<ItemInCart> cart, int userId) {

    try (Connection conn = getConnection(); PreparedStatement pstmt =
conn.prepareStatement("INSERT INTO carts(user_id, cart_data) VALUES(?,?)")) {

        ByteArrayOutputStream baos = new ByteArrayOutputStream();

        ObjectOutputStream oos = new ObjectOutputStream(baos);

        oos.writeObject(cart);

        oos.close();

        byte[] cartBytes = baos.toByteArray();

        pstmt.setInt(1, userId);

        pstmt.setBytes(2, cartBytes);

        pstmt.executeUpdate();

        return true;

    } catch (Exception e){

        e.printStackTrace();

        return false;

    }

}
```

```
}
```

```
}
```

```
@SuppressWarnings("unchecked")
```

```
public static List<ItemInCart> loadCart(int userId) {
```

```
    try (Connection conn = getConnection(); PreparedStatement pstmt =  
conn.prepareStatement("SELECT cart_data FROM carts WHERE user_id=? ORDER BY  
timestamp DESC LIMIT 1")) {
```

```
        pstmt.setInt(1, userId);
```

```
        ResultSet rs = pstmt.executeQuery();
```

```
        if(rs.next()){
```

```
            byte[] cartBytes = rs.getBytes("cart_data");
```

```
            ByteArrayInputStream bais = new ByteArrayInputStream(cartBytes);
```

```
            ObjectInputStream ois = new ObjectInputStream(bais);
```

```
            List<ItemInCart> cart = (List<ItemInCart>) ois.readObject();
```

```
            ois.close();
```

```
            return cart;
```

```
        }
```

```
    } catch (Exception e){
```

```
        e.printStackTrace();

    }

    return null;

}

}
```

Item.java:

```
/******
```

Workshop #06

Course: APD545 - Winter

Last Name: BEHZADFAR

First Name: RADMEHR

ID: 148786221

Section:NDD

This assignment represents my own work in accordance with Seneca Academic Policy.

Signature: RadmehrBehzadfar

Date:2025-03-29

```
*****/
```

```
package grocerystore;
```



```
import javafx.beans.property.SimpleDoubleProperty;
```

```
import javafx.beans.property.SimpleStringProperty;
```

```
import java.io.Serializable;
```

```
public class Item implements Serializable {
```

```
    private static final long serialVersionUID = 1L;
```

```
    private transient SimpleStringProperty name;
```

```
    private transient SimpleDoubleProperty unitPrice;
```

```
    private String nameCache;
```

```
    private double priceCache;
```

```
    public Item(String name, double unitPrice) {
```

```
        this.name = new SimpleStringProperty(name);
```

```
        this.unitPrice = new SimpleDoubleProperty(unitPrice);
```

```
        this.nameCache = name;
```

```
        this.priceCache = unitPrice;
```

```
    }
```

```
public String getName() {  
  
    return name.get();  
  
}
```

```
public void setName(String name) {  
  
    this.name.set(name);  
  
    this.nameCache = name;  
  
}
```

```
public double getUnitPrice() {  
  
    return unitPrice.get();  
  
}
```

```
public void setUnitPrice(double unitPrice) {  
  
    this.unitPrice.set(unitPrice);  
  
    this.priceCache = unitPrice;
```

```
}
```

```
public SimpleStringProperty nameProperty() {
```

```
    return name;
```

```
}
```

```
public SimpleDoubleProperty unitPriceProperty() {
```

```
    return unitPrice;
```

```
}
```

```
private void writeObject(java.io.ObjectOutputStream out) throws java.io.IOException {
```

```
    out.defaultWriteObject();
```

```
}
```

```
private void readObject(java.io.ObjectInputStream in) throws java.io.IOException,  
ClassNotFoundException {
```

```
    in.defaultReadObject();
```

```
    name = new SimpleStringProperty(nameCache);
```

```
        unitPrice = new SimpleDoubleProperty(priceCache);  
    }  
}
```

*@Override*

```
public String toString() {  
  
    return getName();  
}  
}
```

ItemInCart.java:

/\*\*\*\*\*\*

Workshop #06

Course: APD545 - Winter

Last Name: BEHZADFAR

First Name: RADMEHR

ID: 148786221

Section:NDD

This assignment represents my own work in accordance with [Seneca Academic Policy](#).

Signature: RadmehrBehzadfar

Date:2025-03-29

\*\*\*\*\*/

```
package grocerystore;
```

```
import javafx.beans.property.SimpleDoubleProperty;
```

```
import javafx.beans.property.SimpleIntegerProperty;
```

```
import javafx.beans.property.SimpleStringProperty;
```

```
import java.io.Serializable;
```

```
public class ItemInCart implements Serializable {
```

```
    private static final long serialVersionUID = 1L;
```

```
    private transient SimpleStringProperty itemName;
```

```
    private transient SimpleDoubleProperty unitPrice;
```

```
    private transient SimpleIntegerProperty quantity;
```

```
    private transient SimpleDoubleProperty totalPrice;
```

```
    private String itemNameCache;
```

```
    private double unitPriceCache;
```

```
    private int quantityCache;
```

```
private double totalPriceCache;
```

```
public ItemInCart(String itemName, double unitPrice, int quantity) {
```

```
    this.itemName = new SimpleStringProperty(itemName);
```

```
    this.unitPrice = new SimpleDoubleProperty(unitPrice);
```

```
    this.quantity = new SimpleIntegerProperty(quantity);
```

```
    this.totalPrice = new SimpleDoubleProperty(unitPrice * quantity);
```

```
    this.itemNameCache = itemName;
```

```
    this.unitPriceCache = unitPrice;
```

```
    this.quantityCache = quantity;
```

```
    this.totalPriceCache = unitPrice * quantity;
```

```
}
```

```
public String getItemName() {
```

```
    return itemName.get();
```

```
}
```

```
public void setItemName(String itemName) {  
  
    this.itemName.set(itemName);  
  
    this.itemNameCache = itemName;  
  
    updateTotalPrice();  
  
}
```

```
public double getUnitPrice() {  
  
    return unitPrice.get();  
  
}
```

```
public void setUnitPrice(double unitPrice) {  
  
    this.unitPrice.set(unitPrice);  
  
    this.unitPriceCache = unitPrice;  
  
    updateTotalPrice();  
  
}
```

```
public int getQuantity() {
```

```
        return quantity.get();  
    }  
}
```

```
public void setQuantity(int quantity) {  
  
    this.quantity.set(quantity);  
  
    this.quantityCache = quantity;  
  
    updateTotalPrice();  
}
```

```
public double getTotalPrice() {  
  
    return totalPrice.get();  
}
```

```
private void updateTotalPrice() {  
  
    totalPrice.set(getUnitPrice() * getQuantity());  
  
    totalPriceCache = getUnitPrice() * getQuantity();  
}
```



```
public SimpleStringProperty itemNameProperty() {  
  
    return itemName;  
  
}
```

```
public SimpleDoubleProperty unitPriceProperty() {  
  
    return unitPrice;  
  
}
```

```
public SimpleIntegerProperty quantityProperty() {  
  
    return quantity;  
  
}
```

```
public SimpleDoubleProperty totalPriceProperty() {  
  
    return totalPrice;  
  
}
```

```

private void writeObject(java.io.ObjectOutputStream out) throws java.io.IOException {

    updateTotalPrice();

    out.defaultWriteObject();

}

private void readObject(java.io.ObjectInputStream in) throws java.io.IOException,
ClassNotFoundException {

    in.defaultReadObject();

    itemName = new SimpleStringProperty(itemNameCache);

    unitPrice = new SimpleDoubleProperty(unitPriceCache);

    quantity = new SimpleIntegerProperty(quantityCache);

    totalPrice = new SimpleDoubleProperty(totalPriceCache);

}

}

```

LoginController.java:

/\*\*\*\*\*\*

Workshop #06

Course: APD545 - Winter

Last Name: BEHZADFAR

First Name: RADMEHR

ID: 148786221

Section:NDD

This assignment represents my own work in accordance with Seneca Academic Policy.

Signature: RadmehrBehzadfar

Date:2025-03-29

\*\*\*\*\*/

```
package grocerystore;
```

```
import javafx.fxml.FXML;
```

```
import javafx.scene.control.TextField;
```

```
import javafx.scene.control.PasswordField;
```

```
import javafx.scene.control.Alert;
```

```
import javafx.scene.control.Alert.AlertType;
```

```
import javafx.fxml.FXMLLoader;
```

```
import javafx.scene.Parent;
```

```
import javafx.scene.Scene;
```

```
import javafx.stage.Stage;

import java.util.*;

public class LoginController {

    @FXML private TextField usernameField;

    @FXML private PasswordField passwordField;

    @FXML

    private void handleLogin() {

        String username = usernameField.getText();

        String password = passwordField.getText();

        if(username.isEmpty() || password.isEmpty()){

            new Alert(AlertType.ERROR, "Please enter username and password.").showAndWait();

            return;

        }

        User user = DBHelper.loginUser(username, password);

        if(user != null){
```

```
        new Alert(AlertType.INFORMATION, "Welcome " + user.getUsername() +
"!").showAndWait();

    try {

        FXMLLoader loader = new FXMLLoader(getClass().getResource("/layout.fxml"));

        Parent root = loader.load();

        Cart cartController = loader.getController();

        cartController.setLoggedInUser(user);

        Stage stage = new Stage();

        stage.setTitle("Grocery Cart");

        stage.setScene(new Scene(root));

        stage.show();

        ((Stage)usernameField.getScene().getWindow()).close();

    } catch (Exception e){

        e.printStackTrace();

    }

} else {

    new Alert(AlertType.ERROR, "Invalid credentials.").showAndWait();

}
```

```
}
```

```
@FXML
```

```
private void openRegister(){
```

```
    try {
```

```
        FXMLLoader loader = new FXMLLoader(getClass().getResource("/register.fxml"));
```

```
        Parent root = loader.load();
```

```
        Stage stage = new Stage();
```

```
        stage.setTitle("Register");
```

```
        stage.setScene(new Scene(root));
```

```
        stage.show();
```

```
    } catch (Exception e){
```

```
        e.printStackTrace();
```

```
    }
```

```
}
```

```
}
```

MainApp.java:

```
/******
```

Workshop #06

Course: APD545 - Winter

Last Name: BEHZADFAR

First Name: RADMEHR

ID: 148786221

Section:NDD

This assignment represents my own work in accordance with Seneca Academic Policy.

Signature: RadmehrBehzadfar

Date:2025-03-29

\*\*\*\*\*/

```
package grocerystore;
```

```
import javafx.application.Application;
```

```
import javafx.fxml.FXMLLoader;
```

```
import javafx.scene.Scene;
```

```
import javafx.scene.Parent;
```

```
import javafx.stage.Stage;
```

```
public class MainApp extends Application {

    @Override

    public void start(Stage primaryStage) throws Exception {

        Parent root = FXMLLoader.load(getClass().getResource("/login.fxml"));

        primaryStage.setTitle("Login");

        primaryStage.setScene(new Scene(root));

        primaryStage.show();

    }

    public static void main(String[] args) {

        launch(args);

    }

}
```

Model.java:

/\*\*\*\*\*\*

Workshop #06

Course: APD545 - Winter

Last Name: BEHZADFAR

First Name: RADMEHR



ID: 148786221

Section:NDD

This assignment represents my own work in accordance with Seneca Academic Policy.

Signature: RadmehrBehzadfar

Date:2025-03-29

\*\*\*\*\*/

```
package grocerystore;
```

```
import javafx.collections.FXCollections;
```

```
import javafx.collections.ObservableList;
```

```
import java.io.BufferedReader;
```

```
import java.io.FileReader;
```

```
import java.io.IOException;
```

```
public class Model {
```

```
    private ObservableList<Item> itemsObservableList;
```

```
    public Model() {
```

```
        itemsObservableList = FXCollections.observableArrayList();  
  
    }  
}
```

```
public ObservableList<Item> getItemsObservableList() {  
  
    return itemsObservableList;  
  
}
```

```
public void loadData(String csvFilePath) {  
  
    try (BufferedReader br = new BufferedReader(new FileReader(csvFilePath))) {  
  
        String line;  
  
        while ((line = br.readLine()) != null) {  
  
            String[] values = line.split(",");  
  
            if (values.length > 0) {  
  
                String name = values[0].trim();  
  
                double unitPrice = 10.0;  
  
                if (values.length > 1) {  
  
                    try {
```

```
        unitPrice = Double.parseDouble(values[1].trim());

    } catch (NumberFormatException e) {}

    }

    itemsObservableList.add(new Item(name, unitPrice));

    }

    }

    } catch (IOException e) {

        e.printStackTrace();

    }

    }

    }
```

RegisterController.java:

/\*\*\*\*\*\*

Workshop #06

Course: APD545 - Winter

Last Name: BEHZADFAR

First Name: RADMEHR

ID: 148786221

Section:NDD

This assignment represents my own work in accordance with [Seneca Academic Policy](#).

Signature: RadmehrBehzadfar

Date:2025-03-29

\*\*\*\*\*/

```
package grocerystore;
```

```
import javafx.fxml.FXML;
```

```
import javafx.scene.control.TextField;
```

```
import javafx.scene.control.PasswordField;
```

```
import javafx.scene.control.Alert;
```

```
import javafx.scene.control.Alert.AlertType;
```

```
import javafx.stage.Stage;
```

```
public class RegisterController {
```

```
    @FXML private TextField usernameField;
```

```
    @FXML private PasswordField passwordField;
```

```
    @FXML private PasswordField confirmPasswordField;
```

*@FXML*

```
private void handleRegister(){

    String username = usernameField.getText();

    String password = passwordField.getText();

    String confirm = confirmPasswordField.getText();

    if(username.isEmpty() || password.isEmpty() || confirm.isEmpty()){

        new Alert(AlertType.ERROR, "All fields are required.").showAndWait();

        return;

    }

    if(!password.equals(confirm)){

        new Alert(AlertType.ERROR, "Passwords do not match.").showAndWait();

        return;

    }

    boolean registered = DBHelper.registerUser(username, password);

    if(registered){

        new Alert(AlertType.INFORMATION, "Registration successful.").showAndWait();

    }

}
```

```
((Stage)usernameField.getScene().getWindow()).close();

    } else {

        new Alert(AlertType.ERROR, "Registration failed. Username may already exist.").showAndWait();

    }

}

}
```

SavedCartController.java:

/\*\*\*\*\*

Workshop #06

Course: APD545 - Winter

Last Name: BEHZADFAR

First Name: RADMEHR

ID: 148786221

Section:NDD

This assignment represents my own work in accordance with Seneca Academic Policy.

Signature: RadmehrBehzadfar

Date:2025-03-29

```
*****/
```

```
package grocerystore;
```

```
import javafx.fxml.FXML;
```

```
import javafx.scene.control.ListView;
```

```
import javafx.scene.control.Button;
```

```
import javafx.stage.Stage;
```

```
import java.io.File;
```

```
import java.util.List;
```

```
public class SavedCartController {
```

```
    @FXML private ListView<String> savedCartList;
```

```
    @FXML private Button loadButton;
```

```
    @FXML private Button cancelButton;
```

```
    private Cart mainController;
```

```
    private List<File> cartFiles;
```

```
    @FXML
```

```
private void initialize() {  
  
    cartFiles = CartFileHandler.listSavedCarts();  
  
    for (File f : cartFiles) {  
  
        savedCartList.getItems().add(f.getName());  
  
    }  
  
}
```

*@FXML*

```
private void handleLoad() {  
  
    int index = savedCartList.getSelectionModel().getSelectedIndex();  
  
    if (index >= 0) {  
  
        File file = cartFiles.get(index);  
  
        List<ItemInCart> loadedCart = CartFileHandler.loadCart(file);  
  
        if (loadedCart != null) {  
  
            mainController.loadCartData(loadedCart);  
  
        }  
  
        Stage stage = (Stage) loadButton.getScene().getWindow();
```



```
        stage.close();

    }

}
```

*@FXML*

```
private void handleCancel() {

    Stage stage = (Stage) cancelButton.getScene().getWindow();

    stage.close();

}
```

```
public void setMainController(Cart c) {

    mainController = c;

}

}
```

User.java:

```
/******
```

Workshop #06

Course: APD545 - Winter

Last Name: BEHZADFAR

First Name: RADMEHR

ID: 148786221

Section:NDD

This assignment represents my own work in accordance with [Seneca](#) Academic Policy.

Signature: RadmehrBehzadfar

Date:2025-03-29

\*\*\*\*\*/

```
package grocerystore;
```

```
import java.io.Serializable;
```

```
public class User implements Serializable {
```

```
    private int id;
```

```
    private String username;
```

```
    public User(int id, String username) {
```

```
        this.id = id;
```

```
        this.username = username;

    }

    public int getId() {

        return id;

    }

    public String getUsername() {

        return username;

    }

}
```

ItemMaster.csv:

Apple,1.50

Banana,0.75

Bread,2.25

Milk,3.00

Eggs,4.00

Chicken,8.50

Rice,5.00

layout.fxml:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--
```

```
/******
```

Workshop #06

Course: APD545 - Winter

Last Name: BEHZADFAR

First Name: RADMEHR

ID: 148786221

Section:NDD

This assignment represents my own work in accordance with [Seneca Academic Policy](#).

Signature: RadmehrBehzadfar

Date:2025-03-29

```
*****/
```

```
-->
```

```
<?import javafx.geometry.Insets?>
```

```
<?import javafx.scene.control.Button?>
```

`<?import javafx.scene.control.ComboBox?>`

`<?import javafx.scene.control.Label?>`

`<?import javafx.scene.control.Slider?>`

`<?import javafx.scene.control.TableColumn?>`

`<?import javafx.scene.control.TableView?>`

`<?import javafx.scene.layout.BorderPane?>`

`<?import javafx.scene.layout.HBox?>`

`<?import javafx.scene.layout.VBox?>`

`<BorderPane xmlns:fx="http://javafx.com/fxml"`

`xmlns="http://javafx.com/javafx"`

`fx:controller="grocerystore.Cart">`

`<top>`

`<VBox spacing="10">`

`<padding>`

`<Insets top="10" right="10" bottom="10" left="10"/>`

`</padding>`

```
<HBox spacing="10">
```

```
    <Label text="Select Item:"/>
```

```
    <ComboBox fx:id="itemsComboBox" prefWidth="150"/>
```

```
    <Label text="Quantity:"/>
```

```
    <Slider fx:id="quantitySlider" min="1" max="10" value="1" prefWidth="150"/>
```

```
    <Label text="Units:"/>
```

```
    <Label fx:id="purchaseUnitsLabel" prefWidth="50"/>
```

```
</HBox>
```

```
<HBox spacing="10">
```

```
    <Label text="Item Name:"/>
```

```
    <Label fx:id="unitValueLabel" prefWidth="150"/>
```

```
    <Label text="Unit Price:"/>
```

```
    <Label fx:id="unitPriceValueLabel" prefWidth="100"/>
```

```
</HBox>
```

```
<HBox spacing="10">
```

```
    <Button fx:id="addButton" text="Add to Cart"/>
```

```
    <Button fx:id="removeButton" text="Remove from Cart"/>
```

```
<Button fx:id="saveCartButton" text="Save Cart"/>
```

```
<Button fx:id="loadCartButton" text="Load Cart"/>
```

```
<Button fx:id="loadCartFromDBButton" text="Load Cart from DB"/>
```

```
<Button fx:id="checkoutButton" text="Check Out"/>
```

```
</HBox>
```

```
</VBox>
```

```
</top>
```

```
<center>
```

```
<VBox spacing="10">
```

```
<padding>
```

```
<Insets top="10" right="10" bottom="10" left="10"/>
```

```
</padding>
```

```
<Label text="Available Items:"/>
```

```
<TableView fx:id="itemsTableView" prefHeight="150">
```

```
<columns>
```

```
<TableColumn fx:id="itemNameColumn" text="Item Name" prefWidth="150"/>
```

```
<TableColumn fx:id="unitPriceColumn" text="Unit Price" prefWidth="100"/>
```

**</columns>**

**</TableView>**

**<Label** text="Shopping Cart:"/>

**<TableView** fx:id="cartTableView" prefHeight="150">

**<columns>**

**<TableColumn** fx:id="cartItemNameColumn" text="Item Name"  
prefWidth="150"/>

**<TableColumn** fx:id="cartUnitPriceColumn" text="Unit Price" prefWidth="100"/>

**<TableColumn** fx:id="cartQuantityColumn" text="Quantity" prefWidth="80"/>

**<TableColumn** fx:id="cartTotalPriceColumn" text="Total Price"  
prefWidth="100"/>

**</columns>**

**</TableView>**

**</VBox>**

**</center>**

**<bottom>**

**<HBox** spacing="10">

**<padding>**



```
<Insets top="10" right="10" bottom="10" left="10"/>
```

```
</padding>
```

```
<Label text="Total Amount:"/>
```

```
<Label fx:id="totalValueLabel" prefWidth="100"/>
```

```
</HBox>
```

```
</bottom>
```

```
</BorderPane>
```

login.fxml:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--
```

```
/*****
```

Workshop #06

Course: APD545 - Winter

Last Name: BEHZADFAR

First Name: RADMEHR

ID: 148786221

Section:NDD

This assignment represents my own work in accordance with Seneca Academic Policy.

Signature: RadmehrBehzadfar

Date:2025-03-29

\*\*\*\*\*/

-->

<?import javafx.geometry.Insets?>

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.Label?>

<?import javafx.scene.control.PasswordField?>

<?import javafx.scene.control.TextField?>

<?import javafx.scene.layout.GridPane?>

<GridPane xmlns:fx="http://javafx.com/fxml"

fx:controller="grocerystore.LoginController"

alignment="center" hgap="10" vgap="10">

<padding>

<Insets top="20" right="20" bottom="20" left="20"/>

</padding>

```
<Label text="Username:" GridPane.rowIndex="0" GridPane.columnIndex="0"/>

<TextField fx:id="usernameField" GridPane.rowIndex="0" GridPane.columnIndex="1"/>

<Label text="Password:" GridPane.rowIndex="1" GridPane.columnIndex="0"/>

<PasswordField fx:id="passwordField" GridPane.rowIndex="1"
GridPane.columnIndex="1"/>

<Button text="Login" onAction="#handleLogin" GridPane.rowIndex="2"
GridPane.columnIndex="0"/>

<Button text="Register" onAction="#openRegister" GridPane.rowIndex="2"
GridPane.columnIndex="1"/>

</GridPane>
```

register.fxml:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--
```

```
/*****
```

Workshop #06

Course: APD545 - Winter

Last Name: BEHZADFAR

First Name: RADMEHR

ID: 148786221

Section:NDD

This assignment represents my own work in accordance with [Seneca Academic Policy](#).

Signature: RadmehrBehzadfar

Date:2025-03-29

\*\*\*\*\*/

-->

<?import javafx.geometry.Insets?>

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.Label?>

<?import javafx.scene.control.PasswordField?>

<?import javafx.scene.control.TextField?>

<?import javafx.scene.layout.GridPane?>

<**GridPane** xmlns:fx="http://javafx.com/fxml"

fx:controller="grocerystore.RegisterController"

alignment="center" hgap="10" vgap="10">

<padding>

```

        <Insets top="20" right="20" bottom="20" left="20"/>

</padding>

<Label text="Username:" GridPane.rowIndex="0" GridPane.columnIndex="0"/>

<TextField fx:id="usernameField" GridPane.rowIndex="0" GridPane.columnIndex="1"/>

<Label text="Password:" GridPane.rowIndex="1" GridPane.columnIndex="0"/>

<PasswordField fx:id="passwordField" GridPane.rowIndex="1"
GridPane.columnIndex="1"/>

<Label text="Confirm Password:" GridPane.rowIndex="2" GridPane.columnIndex="0"/>

<PasswordField fx:id="confirmPasswordField" GridPane.rowIndex="2"
GridPane.columnIndex="1"/>

<Button text="Register" onAction="#handleRegister" GridPane.rowIndex="3"
GridPane.columnIndex="1"/>

</GridPane>

```

SavedCart.fxml:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--
```

```
/*****
```

Workshop #06

Course: APD545 - Winter

Last Name: BEHZADFAR

First Name: RADMEHR

ID: 148786221

Section:NDD

This assignment represents my own work in accordance with Seneca Academic Policy.

Signature: RadmehrBehzadfar

Date:2025-03-29

\*\*\*\*\*/

-->

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.ListView?>

<?import javafx.scene.layout.HBox?>

<?import javafx.scene.layout.VBox?>

<VBox xmlns="http://javafx.com/javafx" xmlns:fx="http://javafx.com/fxml"

fx:controller="grocerystore.SavedCartController" spacing="10">

<children>

```
<ListView fx:id="savedCartList" prefHeight="200" prefWidth="300"/>
```

```
<HBox spacing="10">
```

```
<children>
```

```
<Button fx:id="loadButton" text="Load" onAction="#handleLoad"/>
```

```
<Button fx:id="cancelButton" text="Cancel" onAction="#handleCancel"/>
```

```
</children>
```

```
</HBox>
```

```
</children>
```

```
</VBox>
```