```
import javafx.application.Application;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.stage.Stage;

public class FoodBeverageManagementApp extends Application {

    private ObservableList<FoodItem> foodItems;
    private ObservableList<BeverageItem> beverageItems;
    private ObservableList<Order> orders;

    public static void main(String[] args) {
        launch(args);
    }

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("Food and Beverage Management");

        // Initialize data
        foodItems = FXCollections.observableArrayList();
        beverageItems = FXCollections.observableArrayList();
        orders = FXCollections.observableArrayList();

        // Create UI components
        TabPane tabPane = new TabPane();
        tabPane.setTabClosingPolicy(TabPane.TabClosingPolicy.UNAVAILABLE);

        // Food Item Tab
        Tab foodItemTab = new Tab("Food Items");
        foodItemTab.setContent(createFoodItemContent());
        tabPane.getTabs().add(foodItemTab);

        // Beverage Item Tab
        Tab beverageItemTab = new Tab("Beverage Items");
        beverageItemTab.setContent(createBeverageItemContent());
        tabPane.getTabs().add(beverageItemTab);

        // Order Tab
        Tab orderTab = new Tab("Orders");
        orderTab.setContent(createOrderContent());
        tabPane.getTabs().add(orderTab);

        // Create main scene
        Scene scene = new Scene(tabPane, 800, 600);
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    private VBox createFoodItemContent() {
        TableView<FoodItem> foodItemTable = new TableView<>();
        TableColumn<FoodItem, String> nameColumn = new TableColumn<>("Name");
        nameColumn.setCellValueFactory(data -> data.getValue().nameProperty());
        TableColumn<FoodItem, String> descriptionColumn = new
TableColumn<>("Description");
        descriptionColumn.setCellValueFactory(data -> data.getValue().descriptionProperty());
        TableColumn<FoodItem, Double> priceColumn = new TableColumn<>("Price");
        priceColumn.setCellValueFactory(data -> data.getValue().priceProperty().asObject());
        TableColumn<FoodItem, Integer> quantityColumn = new TableColumn<>("Quantity");
        quantityColumn.setCellValueFactory(data ->
data.getValue().quantityProperty().asObject());

        foodItemTable.getColumns().addAll(nameColumn, descriptionColumn, priceColumn,
quantityColumn);
        foodItemTable.setItems(foodItems);

        TextField nameTextField = new TextField();
        TextField descriptionTextField = new TextField();
        TextField priceTextField = new TextField();
        TextField quantityTextField = new TextField();

        Button addButton = new Button("Add");
        addButton.setOnAction(event -> {
          String name = nameTextField.getText();
          String description = descriptionTextField.getText();
          double price = Double.parseDouble(priceTextField.getText());
          int quantity = Integer.parseInt(quantityTextField.getText());

          FoodItem foodItem = new FoodItem(name, description, price, quantity);
          foodItems.add(foodItem);

          clearFields(nameTextField, descriptionTextField, priceTextField, quantityTextField);
        });

        VBox content = new VBox(10);
        content.setPadding(new Insets(10));
        content.getChildren().addAll(foodItemTable, new Separator(),
createFoodItemForm(nameTextField, descriptionTextField, priceTextField, quantityTextField,
addButton));
        return content;
    }
```

```
    private VBox createBeverageItemContent() {
        // Similar to createFoodItemContent(), but for beverage items
    }

    private VBox createOrderContent() {
        // Create UI components for managing orders
    }

    private GridPane createFoodItemForm(TextField nameTextField, TextField
descriptionTextField, TextField priceTextField, TextField quantityTextField, Button addButton)
{
        GridPane form = new GridPane();
        form.setHgap(10);
        form.setVgap(5);

        form.addRow(0, new Label("Name:"), nameTextField);
        form.addRow(1, new Label("Description:"), descriptionTextField);
        form.addRow(2, new Label("Price:"), priceTextField);
        form.addRow(3, new Label("Quantity:"), quantityTextField);
        form.addRow(4, addButton);

        return form;
    }

    private void clearFields(TextField... fields) {
        for (TextField field : fields) {
            field.clear();
        }
    }
}
```