Faculty of Engineering and Applied Science
SOFE 3650U Software Design and Architecture
Assignment 2

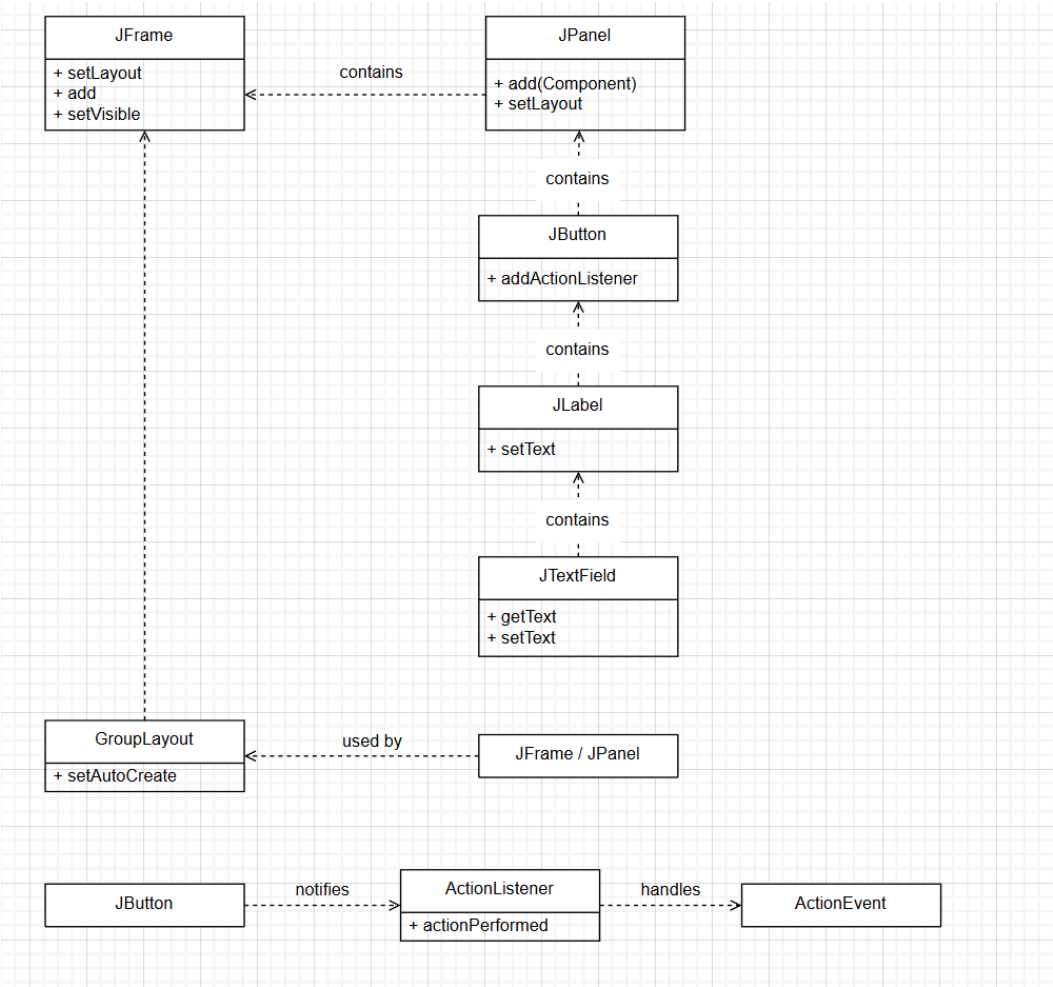
First Name	Last Name	ID
Nithusan	Kandasamy	100869101
Zachary	Matasic	100918415
Mohamed	Tawfik Omar	100912760

Question

1. Do some investigation into the Swing framework and write a short paragraph describing the purpose of the Swing framework. Submit a class diagram of the components of Swing.

In this codebase, the Swing framework serves as the GUI toolkit for implementing a classic MVC architecture pattern in a Java application. The View class demonstrates Swing's purpose by creating a window-based user interface using core Swing components including JFrame for the main window, JLabels for the text display, JTextFields for user input, and JButtons for user interactions. The framework enables separation of concerns where the View handles all GUI presentation logic using Swing's GroupLayout manager for component arrangement, while the Controller class manages user interaction through Swing's event handling system with ActionListeners, and the model maintains the application data state independently of the UI implementation.

Class diagram of the components of Swing:

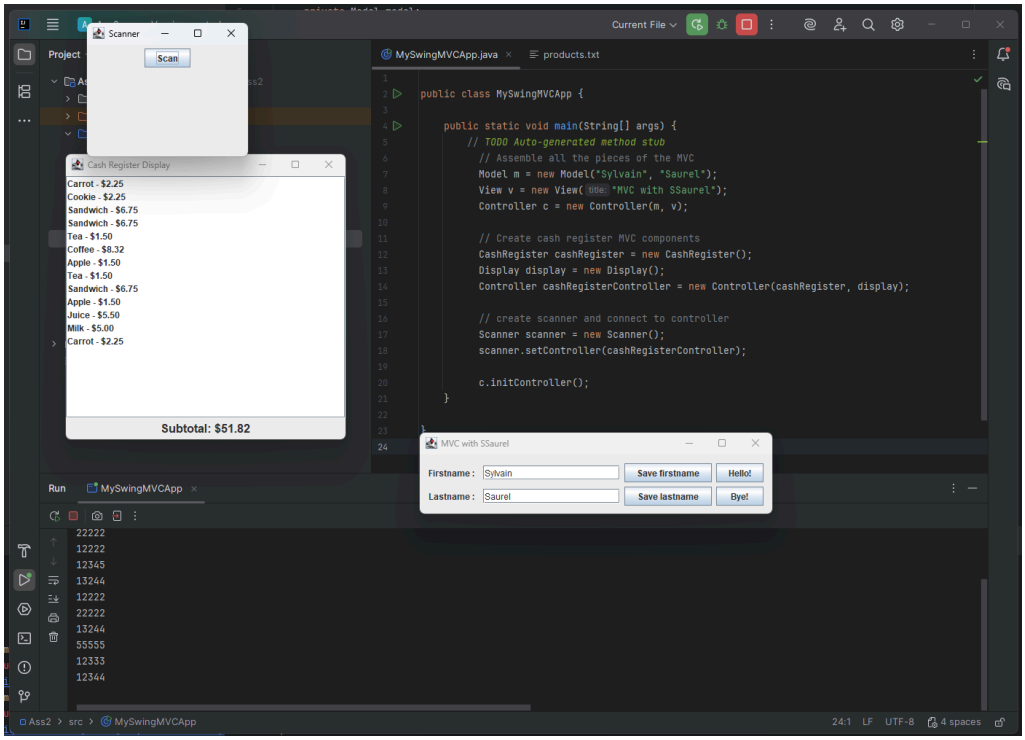


2. Look through the example code in the GitHub repository and explain how this example implements the MVC pattern. How does it differ from the conventional MVC pattern described in the lectures?

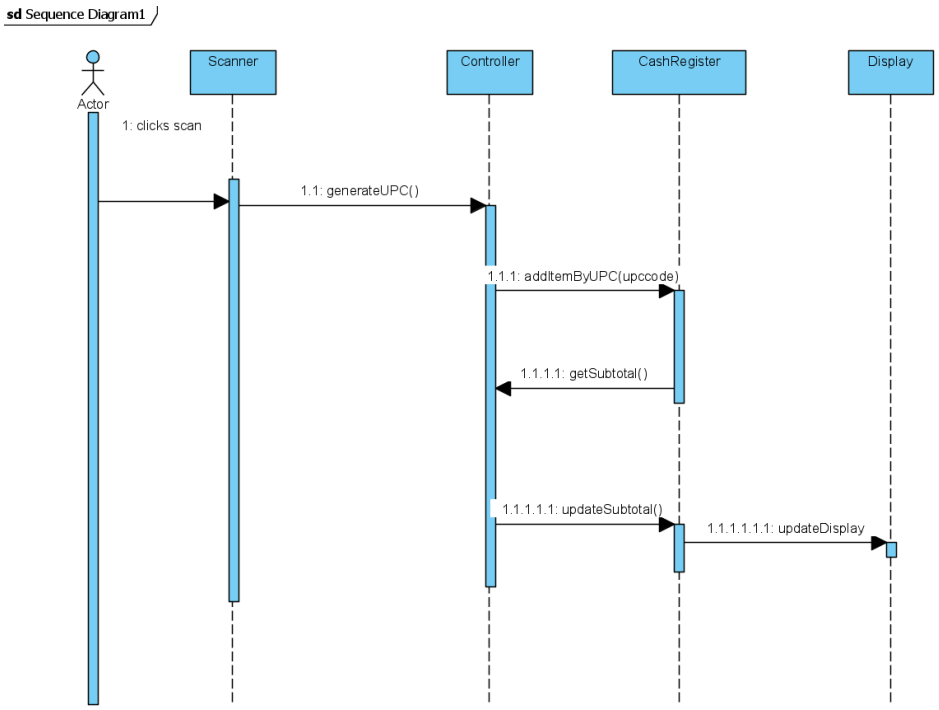
The example code applies a simplified MVC structure where the Model only stores data through basic getter and setter methods, the View manages all Swing interface components and layouts, and the Controller connects them by handling user inputs through ActionListener events. This design differs from the conventional MVC pattern discussed in class because it does not use the Observer approach for automatic updates between model and view. Instead, the controller manually synchronizes changes, creating tight dependencies and a centralized control flow. The model and view are directly coupled without interfaces or abstraction layers, reducing modularity. Finally, user feedback is handled through dialog boxes rather than dynamic UI updates, which further departs from the decoupled and event-driven behavior expected in a full MVC implementation.

3. In the repository there is another Java file that is named Scanner. This Class emulates the scanning of a product by generating a UPC code. If you press on the Scan button you will see that it prints out the “12345” code to the console. For this assignment, modify the Scanner so that each time the Scan button is pressed, it randomly selects a UPC code from a file containing product information with the following format.

UPC Code	Product	Price
12345	Coffee	$8.32
67890	Muffin	2.50



4. Create a sequence diagram of your design for the scenario presented in question 3 representing a single press of the scan button.


sd Sequence Diagram1
Actor
Scanner
Controller
CashRegister
Display
1: clicks scan
1.1: generateUPC()
1.1.1: addItemByUPC(upccode)
1.1.1.1: getSubtotal()
1.1.1.1.1: updateSubtotal()
1.1.1.1.1.1: updateDisplay