

MILESTONE 4

Overview

You are going to be adding finishing touches on your project in this milestone. It is time to make a GUI. You will be using [Swing](#) to make it! The Swing package is already available in IntelliJ, and you just need to import it. This milestone may require that you investigate on-line (e.g., how to add images to your GUI)

*Keep in mind that most of the tasks you will be required to do are relatively **open-ended**. You must justify any decisions you made that were not obvious in your Change Log. Refer to the examples in Milestone 1.*

GUI

All the functionality you implemented in the *StoreView* class from Milestone 2 should be available to the user through the GUI. Now, the user can perform actions by the click of a button instead of typing in a command. Your GUI can be arranged and styled to your preference. However, it should be intuitive. For example, it might not make sense to have an “Add to Cart” button on the order summary page. It is entirely possible for you to have everything on a single page, provided everything works correctly and there are no bugs! A GUI inspiration is provided in the Appendix, but feel free to make your own design.

Basic Functionality

To reiterate, your program must contain the following functionality, the rest is up to you:

- The GUI must contain the Basic operations you would expect to be able to perform in an online store: browse, add items to a cart, remove items from a cart, checkout. These were completed in Milestone 2.
- Products should be displayed for viewing. How you do this is up to you.
 - o There should be a visual element included with the displayed Products. This can be an image, GIF, icon, or html (Hint: [hint](#)). Determine what makes the most sense for your store. If your store sells varieties of apples, a product image would suffice. If your store sells digital marketing packages, an icon with a link to some example work might make more sense. *Note: For obvious reasons, no embedding videos!*
- The user should be able to examine the details of each Product. Once again, it is up to you how this is presented.
- Once the user goes through the checkout, an order summary (receipt) should be displayed. It should then wait for some feedback from the user (an “OK” button) and exit the program.
- It will be difficult to test if the store functions with multiple clients open in this milestone. It is not advisable to create multiple Swing JFrames. However, if you verified it worked in Milestone 2 + 3, it would most likely work for multiple GUIs as well. Note that in this milestone you should really only be changing the *StoreView*. Working with multiple GUI is beyond the scope of this course, so you can focus on having a single client.

SYSC 2004 – Course Project

Question

When refactoring the *StoreView* class into a GUI, did you take a composition or an inheritance approach? Why?

Milestone 4 Deliverables

1. The following classes completed according to Milestone 4 specifications: *StoreView.java*, *ShoppingCart.java*, *Inventory.java*, *StoreManager.java*, and *Product.java*. You need to include any auxiliary class you use in your application. Include any resources required by your application. All the project files should be zipped into an archive called **myStore.zip**. Make sure that the structure of your zip file allows the TAs to import your project to IntelliJ.
2. Everything applicable to Milestone 4 from General Submission Requirements. Do not forget the updated UML diagram, to document all your classes, and the report with the Change Log and the answers to the question.

APPENDIX

