48024 Programming 2 Assignment 2

Topics:

OO design, GUIs, MVC, tables, lists

Learning Outcomes:

This assignment supports objectives 3 - 5.

Due date:

28 October 2024 - 11:59PM (Monday Week 12)

Weight:

25%

Individual Work

All work is individual. You must write every line of code yourself except for code copied from study module sample code, lecture sample code, tutor demos or lab code.

In most cases, you may discuss ideas, approaches and problems. However, if an assignment task is labeled as “Advanced”, you must not discuss ideas, approaches and problems. Advanced tasks are designed to test your ability to think on your own.

You **MUST NOT** let another student see your solution code, and you **MUST NOT** look at another student’s solution code. Sharing your code on public forums such as the discussion board, or Internet forums such as stackoverflow.com is not permitted. More information about Academic Misconduct can be found at:

http://www.gsu.uts.edu.au/rules/student/section-16.html

Working Language

You can choose either Java or Python to complete assignment 2. The higher mark between your Java solution and Python solution will be counted into your final grade. However, you are only credited with one of your solutions, either Java or Python, not both of them or the mixture.

The specification is illustrated based on Java. You can simply translate the Java syntax to Python for your Python solution. Detailed explanations about Python criteria will be posted on the FAQ page.

Expected workload

The time to do the assignment to a credit level (i.e. a mark between 65% to 75%) has been estimated at 25 hours for a student of average ability who has completed all the tutorial and lab exercises.

Skeleton Code

You must use the skeleton code provided as a starting point for your assignment. The following restrictions apply to the scaffold:

Java:

- **You are not allowed** to modify the model classes in any way
- **You cannot** add extra controllers or views; you can only write your implementation in the files provided
- **You cannot** alter the folder structure
- **You may add** to the style.css file if you like, but do not modify any of its existing content.

Python:

- **You are not allowed** to modify the model classes in any way
- **You will need** to add extra classes/views/controllers to complete your implementation; how you structure this is up to you.
- **You may alter** the folder structure. If you cannot understand a reason to do this, then don't; but certain implementations may run into import problems that can be easily resolved with moving folders around; you are allowed to do this.
- **You may add** to the Utils.py file if you like, but do not modify any of its existing content.

Model Changes

Some alterations to the models for assignment 1 have been made to better facilitate a GUI approach. It is highly recommended that you look through the skeleton first to try and understand it before beginning your implementation. Some notable changes are as follows:

- Organisation now has a static field that keeps track of the logged in user.
- Cart now has a catalogue. This acts as a temporary duplicate of the suppliers products that can be altered as the user is ordering. (The actual suppliers list of products should not be altered until the user has clicked 'checkout')
- Order has a reference to the Cart it belongs to.

Specification

The specification is presented in several parts. This document outlines the rules of the assignment, as well as all relevant information.

A demonstration video is also provided on Canvas. This demonstration video is considered part of the specification and is intended to demonstrate exactly what your program should do in both functionality and looks.

The video is considered your intended solution for both Java and Python with the sole exception that Javas colour scheme is yell scheme is blue. However, this will be done for you if you use the provided classes

A rubric is also provi this is mostly just a written version of what the demonstration video presents. How ing as this is the exact rubric that your tutor will mark against during the final demo

Requirements

Windows

For both Java and Python, your windows should not be resizable, and the X button (at the top right) should not be capable of closing the window. The only way to close the program should be using the Close button you have implemented.

For Java, you can use the FixedStage class provided for you in the scaffold to achieve this. This class is not provided in the python scaffold; however, lecture 9 discusses a method to achieve it.

Layout

To get full marks, you should layout your windows to look as close as possible to the demonstration video. This includes the spacing between elements vertically and horizontally. An example window is shown below:

The exact values used in the window above were 600 for the width of the window, and a vertical spacing of 20 between the elements. However, note that there is no requirement that you use the same exact values, as long as the elements proportionally are the same.

Code

Your solution must satisfy the following code requirements:

- Your solution must follow the MVC architecture and utilize multiple window displays.
- Your solution must keep the package structure and class names that were provided in the skeleton code.

- For Java, the views must be laid out in FXML.

Submission

Regardless of the lar ill need to submit your assignment in 2 places:

Java

- **Canvas:** JAR file
- **Ed:** Complete source code

Python

- **Canvas:** .zip file
- **Ed:** Complete source code

You will need to submit the source code to ED. The marker will check the source code to make sure all the requirements have been met. Missing the source code will result in 10 marks deduction of assignment 2.

The Ed submission is used for record-keeping and will not provide any test cases or feedback on your submission. Tutors will mark the functionality of your solution in the week 12 demonstration by opening your .jar/.zip from Canvas. It is your responsibility to rigorously test this and make sure it works.

For JavaFX solution, your assignment should be submitted as a JAR file that includes:

- All FXML, CSS and image files required to run your assignment.
- The binary files(.class) of the runnable jar file.

You can check the JAR file on a lab PC to make sure it works. Kindly note that jdk1.8 is the baseline for the assignment. If your work is based on other JDK versions, it's your own responsibility to generate a compatible jar that can run on jdk1.8. The incompatible jar file will be marked 0 directly.

The corresponding coversheet stating the reference of the codes and the use of GenAIs should be submitted to Canvas for the assessment. Detailed specification about GenAI clarification refers to the document "GenAI-Intro-Slides-for-students-48024.pdf" in GenAI overview on Canvas.

Demonstration and Extension

You will be marked by the Canvas submission for the features that can be demonstrated to work on a lab PC. Your tutor will ask questions based on your demonstration in the last lab class. Aside from marks for the functionality, the marker will also check your code to ensure that all code requirements have been met. Your final mark will be a combination of marks for functionality, marks for code and moderation marks for answering questions.

Marking the code and analyzing spoofing, cheating and plagiarism is done in the two weeks following the due date. If you are suspected of Academic Misconduct, your case will be forwarded to the Misconduct Committee, and you will receive a "high similarity" note on ED submission and your mark will be pending.

The due date won't be extended as assignment 2 requires a 3-minute demonstration in the lab class of Week 12. Extension approval CANNOT be given after the due date. Extension request must be submitted before due through forms: Extension request for 48024_2024_SPR. For any extension beyond the demonstration class, you will need to submit a Special Consideration following the Special Consideration process for special consideration for reasons including unexpected health, family, work problems, or other extenuating circumstances.

Online support

If you have a questio first, it may already be answered there. You should read the FAQ at least once be ution, but to be safe check it every couple of days. Anything posted on the FAQ is the assignment specification.

If anything about the r inconsistent, check the FAQ first, then contact the subject coordinator who will t plying to you directly and posting the common questions and answers to the FAQ. on the job, where you ask your client if you are unsure what has to be done, but then you write all the code to do the task.

The preferred way to ask assignment-related questions is to participate in the lectures, lab activities, UPASS, and consultation sessions. Students are expected to seek help through the following steps:

Step 1: Check the FAQ
Step 2: Ask peers or via the group discussion board
Step 3: Ask your tutor in the lab class
Step 4: Forward to the subject coordinator

Requests for appointments outside the given consultation hours may be arranged where circumstances require, and to do so please book time to meet with us.

Marking Scheme

The marking scheme for the assignment provides more details to the Rubric Table on Canvas. Note that individual tests may test several functionality components, and a functionality component may be tested by several tests.

Login	Invalid credentials open the error window	0.5
	Valid credentials go to applicable dashboard	0.5
	Stage disappears on valid credentials	0.5
User Dashboard	Shop button goes to appropriate window Order History button goes to appropriate window	0.5
	Close button exits entire application	0.5
Supplier List	All suppliers in the organisation shown in ListView	0.5
	ListView is Single Select only	0.25
	Shop button goes to appropriate window	0.5
	Stage disappears on valid selection	0.5
Supplier Dashboard	Available products shown in TableView	0.5
	TableView doesn't allow selection	0.5
	Order button goes to appropriate window	0.5
Cart Dashboard	Available products shown in TableView	0.5
	TableView is Multi Select	0.25
Add to Cart	Appropriate window opens for all selected items	1
	On add, selected products are removed from catalogue	1
	On add, selected products are not removed from the supplier	1
	Error window opens when an invalid stock is entered	0.5
View Cart	All products in the catalogue shown in TableView	0.5

	...oves product from cart	0.5
	...product back to catalogue	1
Cancel Cart	...makes no changes to the supplier list	1
Cart Checkout	...the order is reflected in the supplier list	1
Order History	...in a ListView	0.5
User Dashboard (Manager)	...reflect a Manager is logged in	0.5
	Manage button opens supplier list with only the suppliers that the user manages	1
Supplier Dashboard (Manager)	Filter checkbox filters TableView by availability	1
Remove/Delist (Manager)	Remove button removes product from supplier	0.5
	Delist delists product	0.5
	Delist button stays inactive on a product that has been previously delisted	1
Error Window	Exception and appropriate custom message are displayed	0.5
Layout	Heading content consistently matches demo video	0.5
	All elements stretch width of window	0.5
	Buttons have no horizontal padding	0.5
	Elements are spaced out vertically	0.5
Styling	Colour, font and size of elements/window consistently match demo video	1
Buttons	Buttons activate and deactivate as expected consistently	1.5
Icons/Title	Window icon and title consistently match demo video	0.5
Window	Windows are unresizable and cannot be closed using the X button	0.5
Total		**25**