

Project #3 (45 points)

Due Date

- Monday, November 2, by 11:59pm.

Submission

1. Group Submission (one copy per team)
 - (a) You must designate a submitter (one of the team members) and **submit the zipped project folder** to Canvas.
 - (b) You must include both team members' names in the comment block on top of EVERY Java file.
 - (c) Your project folder must include the following **subfolders/files for grading**.
 - Source folder, including all Java files [30 points]
 - Test design document [10 points]
2. Individual Submission (everyone must submit a copy)
 - Personal time log, using the template posted on Canvas. [5 points]

Project Description

In this project, you will develop a GUI with JavaFX for the Transaction Management System you created for Project 2. The GUI shall behave exactly like the TransactionManager class in Project 2. In addition, you must add the text file import and export functionality to the GUI.

Requirements

1. This is a **group assignment**. You **MUST** work in pair in order to get the credit for this program. You **MUST** follow the software development ground rules, or **you will lose points** for not having a good programming style.
2. You are required to log your times working on this project with the template provided on Canvas. **The time log is an individual assignment. You will lose 5 points** if the log is not submitted. If the times and comments are not properly logged, you will only get partial credits. You must type, handwriting is not acceptable.
3. Each Java class must go in a separate file. **-2 points** if you put more than one Java class into a file.
4. You **MUST** import all the classes from Project 2 and use them in this project, **EXCEPT** the TransactionManager class. **-2 points** for each class not used.
5. Project 3 will use the Model-View-Controller (MVC) design pattern. You must use a JavaFX fxml file for the “view”, a Controller class for the “control”, and the entity classes from Project 2 for the “model”. The Main.java, fxml and controller files constitute the GUI version of Transaction Manager. **You will get 0 points** if you don't.
6. Your GUI must include the following JavaFX components.
 - (a) Use at least 3 different Layout Panes, such as BorderPane, GridPane, VBox, Hbox,, or **-5 points**.
 - (b) Use a TextArea to display output, or **-5 points**. All output **MUST** be appended to the TextArea. **You are NOT ALLOWED to use System.out** in ALL CLASSES, or you **will lose 10 points**. This means you must modify all the print methods in the AccountDatabase class to return a string.
 - (c) No more than 4 buttons on any GUI page display, or **-2 points for each additional button**.
 - (d) Use a RadioButton group that allows the user to choose between Checking, Savings or MoneyMarket. In other words, the user can only select one RadioButton at a time. **-5 points** if this is not done.
 - (e) You must use the CheckBox for the options such as direct deposit for the Checking, or loyal customer for the Savings. You are required to disable the options that are not applicable. For example, disable loyal customer option if Checking or MoneyMarket is selected, and disable direct deposit if Savings or MoneyMarket is selected. **-2 points** for each violation.
 - (f) You **MUST** set the title of the primaryStage (title for the window.) **-2 points if you don't**.

- (g) You must provide the functionality on the GUI to import and export a text file. A text file “database.txt” is provided for you to test your program. Data tokens in the text files must be delimited by the commas (.). **You will lose 10 points** if this is not done.

Program Testing

1. You **MUST** create a test document and design the test cases for the GUI program. In addition to the test cases from Project 2, you must design the test cases to test all possible invalid conditions. For example, type mismatch or not entering data on GUI, etc. **The test document is worth 10 points.**
2. You must use your test cases to manually test your GUI. All invalid data should be rejected by the GUI. Proper error messages must be displayed in the TextArea. **You will lose 2 points** for each invalid condition not rejected, or error message not properly displayed in the TextArea.
3. Your program must always run in a sane state and **should not crash in any circumstances**. In other words, the graders will try to produce exceptions in any way they can while running your GUI. You must catch all Java Exceptions and your program will continue to run until the user stops the program execution or closes the window. **You will lose 2 points** for each exception not caught.