

**Due Date**

Monday, November 6, 11:59pm

**Submission**

- (1) Zip your project folder and submit the zipped file to Canvas. You MUST include the following grading items.
  - (a) Source folder **src**, which contains the Java packages you developed, including [100 points]
    - All Java files from Project #2, excluding TransactionManager.java and RunProject2.java.
    - 3 JavaFX related files, which replace the TransactionManager.java in Project #2
      - **TransactionManagerMain.java**, the Java class containing the launch() method.
      - **TransactionManagerController.java**, the Java class containing the event handlers.
      - **TransactionManagerView.fxml**, which is the GUI design.
  - (b) Test Specification for System Testing. [15 points]
  - (c) Javadoc folder. [5 points]
- (2) The submission button on Canvas will disappear after **November 6, 11:59pm**. It is your responsibility to ensure your Internet connection is good for the submission. **You get 0 points** if you do not have a submission on Canvas. **Projects sent through the emails will not be accepted.**

**Project Description**

Your team will revamp the user interface for the software you developed in Project 2 and develop a GUI (graphical user interface) with JavaFX. The GUI shall replace the TransactionManager class in Project 2 and provide the same functionalities; that is, you must meet ALL the functional requirements stated in Project 2. In addition, you will add a new feature to load the accounts from a text file to the account database.

**Project Requirement**

1. You MUST follow the Coding Standard posted on Canvas under Week #1 in the “Modules”. **You will lose points** if you are not following the rules.
2. You are responsible for following the Academic Integrity Policy. See the **Additional Note #14** in the syllabus. The consequences of violation of Academic Integrity Policy are: **(i) your group receives 0 (zero) on the project, (ii) the violation is reported, (iii) a record on your file of this violation.**
3. Each source file (.java file) can only include one public Java class, and the file name is the same with the Java class name, or you will lose **-2 points**.
4. You MUST include all the Java classes from Project 2 and use them in this project, EXCEPT the TransactionManager class and RunProject2 class. You will **lose 5 points** for each class in Project 2 not used. You will **lose 10 points** if you use TransactionManager.java. **NOTE**, if you lose points in Project 2, you must fix the problems, or you will lose points again for the same cause!
5. You MUST add a feature to load the accounts from a text file “**bankAccounts.txt**” to the account database, or you will **lose 10 points**.
6. This project uses the Model-View-Controller (MVC) design pattern. You must include only ONE JavaFX fxml file for the “View”, ONE controller class for the “Controller”, and ALL the classes from Project 2 as the “Model”. In addition, there will be ONE additional Java file that contains the main() method to “launch” the GUI. **You will get 0 points** if you don’t follow the MVC design pattern.
7. You should design the GUI for this project, and you can use any JavaFX UI control objects.
8. You MUST set the title of the stage (title for the window), or **-2 points**.

9. You are **NOT ALLOWED** to use **System.out** (write to console) or **System.in** (read from console) ANYWHERE in ALL CLASSES, or you will lose 3 points for each violation, with a maximum of losing 10 points. This means you MUST modify all the print() methods to return a String. All read and write must be done on the GUI.
10. You are required to **generate the Javadoc** after you properly commented your code. Your Javadoc must include the documentations for the constructors, private methods and public methods of all Java classes (\*.java files.)
  - You DO NOT need to comment the **TransactionManagerMain.java**,
  - You MUST comment **TransactionManagerController.java**.
  - DO NOT include the \*.fxml file, which is NOT a java file.
  - Generate the Javadoc in a single folder and include it in your project folder to be submitted to Canvas.
  - You are responsible to **double check** your Javadoc after you generated them. The grader will navigate the Javadoc with the “index.html”. You will **lose 5 points** for not including the Javadoc, OR, the grader cannot navigate your Javadoc through the “index.html”.
11. **System Testing.**
  - Functional testing through the GUI.
  - You MUST create a test specification and design the test cases for testing the GUI. The **Test Specification** is worth **15 points**. You MUST use the table template in the Coding Standard to organize the test cases, or you **will get 0 points** for this part.
  - Use the test cases in **Project2TestCases.txt** as a reference to design your test cases to run the System Testing through your GUI.
  - Use your test cases to manually test your GUI. All invalid data should be rejected by the GUI. Proper error messages must be displayed on the GUI. You will **lose 2 points** for each invalid condition not rejected, or each error message not properly displayed.
  - You are responsible to thoroughly test your software with the test cases you designed. Your software must always run in a sane state and **should not crash in any circumstances**. The grader, as a user of your software, will try to produce exceptions while trying to use your GUI. You must handle all exceptions. Your software shall continue to run until the grader stops the execution or closes the GUI window. **You will lose 2 points** for each exception not caught.