Term Project Assignment

Code Version 0.8

Object-Oriented Design

Prof: Ahmad Yazdankhah

"A good developer always reads the requirements at least 11 times! Ahmad Y"

Objective

In this assignment, **your team** is going to **implement version 0.8** of your product.

Code Requirements

1. At least **80%** of the product is operational.
2. It's a **standard JavaFX project**.
3. It can be compiled and run with **Java 8** without any errors (warnings are OK).
4. The **name of the project and its folder is team-xx-yy** while **xx is your section number** and **yy is your team number**. Both prefixed with '0' if the number is one digit. (e.g. **team-03-05** for team five in section 3)
5. The class containing the **main method** is called "**Main**" and is located in the "**application**" **package**. Your other classes could be located in any packages that you design.
6. The code is **reasonably documented** by comments.
7. The code should run without any other requirement from your client's side.
8. The project should **NOT contain extra hidden files**, **for example version control files**.

A sample project named team-00-12 is uploaded in:

Canvas → Files/Term Project/Samples/team-00-12.zip

What You Submit

1. **Clean up the project** and remove all unnecessary files and folders.
2. Zip the project folder that's named as team-xx-yy as mentioned in item 4 of the "Code Requirements".
3. Upload it in the Canvas before the due date.
4. **One submission per team is enough.**

Note: if you resubmit your file several times, Canvas adds a number at the end of your file name. **I won't consider that number as part of the file name**.

Product Requirements (Repeated and Modified)

1. The software should be a **desktop application**, not a web-based application **but it's OK to use cloud for your database, and webservices for getting data**.
2. Required technologies: **Java, JavaFX**
3. The **data should be persisted** in the flat files, or database.
4. In the case of using database, it should be a simple in-memory database that does not need installation from your client (your professor) side. It means, when your client runs the application, all configurations should be done automatically.

Rubrics

- (90 points) if your code runs without any issue and stratifies the requirements
- (10 points) Naming the folder and project
- **You'll get 0 if there is any issue.** In this case, you'll need to resubmit and I'll follow the late policy.

General Hints

1. Always read the requirements at least 10 times before and during implementation!
2. Read the requirement one more time right before submitting your work to make sure that nothing is missed. (Overall, 11 times!)
3. **Always set the due date for yourself at least 2-3 days before the official due date**.
4. After submitting your work, always download it back and re-open it to make sure that the process of submission was fine.
5. Always make sure that you have the latest version of this document. Sometimes, based on your questions and feedbacks, I need to add some **clarifications**. If there is a new version, it will be announced via Canvas.
6. For the **late submission policy**, please refer to the Greensheet. I don't expect a good software engineer submit the assignments late. Therefore, absolutely no excuse will be accepted.
7. This is a team-based project. Therefore, **the whole team get one grade** regardless of who did what. It's the **teams' leaders' responsibility** to fairly distribute the tasks between the teammates.

This will be **an unacceptable and non-professional behavior** if at the end, a student tells me he/she did all tasks and **would expect extra credit**!

8. If there is any ambiguity, question, or concern, **please open a discussion in Canvas**.