Fall 2024

COMP 3504 – Prog IV: Software Engineering

Assignment 1: Planning and Programming

MOUNT ROYAL
UNIVERSITY
1910

Description – In this assignment, your task is to write a program to practice programming and planning within your team. To do this, you will be completing a programming task (in a language of your choice) within two 1-week sprints. The programming task is described in this document.

You can do this by completing a 1-week sprint, e.g., having a planning meeting at the beginning of your week (e.g., on Monday), break down the tasks for your sprint and add them to your sprint backlog (or, your ToDo list), and then do your stand-ups (short meetings or a text updates) during your sprint, and then at the end of the sprint (e.g., on Friday), get together and demo all the code that is developed. At the same meeting, do a retrospective to review your strengths and challenges and devise a plan to improve your challenges for the following week (your second sprint).

Create a Github repository. Make sure your Github repository is private and share your repository with all your team members and with your instructor (Github username: Yasaman-A). Follow the best practices to create and use issue and task boards. It is expected that team members know how to get code from and commit code to the repository. If anyone in your team needs help with these tasks, please help them. If everyone in the team is struggling with using Git, please contact your instructor.

At this assignment, you are not expected to use advanced features such as branches or pull requests. We will discuss the full process later on in the course. Even if some members know these features, please do not use them, if there are members in the team that are not yet comfortable with using them. You will have the opportunity to practice the full set of Git skills in your next assignment.

Report – Complete the following items during your sprint meetings and include them in your submission report.

- After your **sprint planning** meeting and adding issues to the task board, take a screenshot of the task board and include it in your submission.
- During your **retrospective** meetings, reflect on the strength and challenges you had in your sprint. Then, discuss and come up with a plan to improve your challenges in the future. Include a summary of your strength, challenges, and your improvement plan in your report. You should do one retrospective at the end of week 1 and another retrospective at the end of week 2. Please write the strength and challenges identified (or improved) at each retro meeting.

Programming task – The followings are the requirements for the application you will be developing. The application you develop should work correctly, but the code your implement does not need to be the most efficient code you can write. Try to follow the "KIS"- Keep it Simple - strategy while implementing your code. You will have time during your next assignment to review and revise your implemented code.

You can implement the application in any language that your team prefers. Your team may make any assumption for the parts of the requirements that are not clear. Please include your assumptions in your submission report. If your assumptions are not accepted, you will be advised to relax them in your next assignment.

Requirement description - A retail shop that sells tools requires an application to manage the inventory of different types of tools it sells. The store owner wants to be able to modify the store's inventory by adding new tools and deleting tools. The owner also wants to be able to search the inventory for tools by tool name and by tool id. Currently, the information about tools available in the shop and suppliers is stored in two text files which are given: `items.txt, and suppliers.txt.`

The order and type of data given in these files are:

`items.txt:`

```
(id; name of tool; quantity in stock; price; supplier id)
```

`Suppliers.txt:`

```
(id; company name; address; sales person contact)
```

The owner would also like to check the quantity of each item in stock. If the quantity of each item in stock goes below 10 items, then the program should *automatically* generate an order line for that item. The order line will have the supplier information and the required quantity for that item (The default quantity ordered by each item = 30 minus the number of existing item). All items ordered each day should be included in an order with a randomly generated 5-digit id, and the date that was ordered. The order should be written to a text file called `orders.txt.` A sample order file is as follows:

```
===========================================================
ORDER ID.:                 10008
Date Ordered:              September 18, 2022

Item description:          Red Pen
Amount ordered:            39
Supplier:                  GoodStationary Inc.

Item description:          A4 Papers
Amount ordered:            36
Supplier:                  BestPapers

Total cost:                $1200
-----------------------------------------------------------
ORDER ID.:                 29780
Date Ordered:              September 26, 2022

Item description:          Push Pin
Amount ordered:            30
Supplier:                  WinManufacturing Inc.

Total cost:                $1200
===========================================================
```

Submission

- Create a report document with the items requested in the report section (Screenshot of initial planning, Summary of strength, challenges, improvements, and also any assumptions you make).
- Include the **link to your private GitHub repository** in the comment section of your submission. Your repository should have already been shared with your instructor.

Grading

Item	**Grade**
Tasks are clearly defined in the task board using Github issues	15
All team members have contributed almost equally to the development tasks	10
Proper comments are used in Git commits.	10
Clean repository with proper ReadMe	10
The implemented code is working correctly	25
Code is clean and readable	10
Report	20