Software Engineering Project 1

(Comp 10050)

Assignment 3 – Implementing a Kanban-style board with a series of lists on a command line with a text interface

Deadline 6 pm April 21st 2023

Aim: to create a program that generates a Kanban-style board with lists that contain items and an interfaces that allows you to edit it, load it and save it

You are expected to perform this assignment in a group with just one fellow student who has been randomly assigned with you.

A. Detailed Specification

The objective for this assignment is to create a program that:

1) Display a menu with 6 options allowing for Display, Loading, Editing an item on a list, Editing a list on the board, Saving the Board to a file, and Quitting . The program should have a default board already saved.

This is an example of how the program could function when the display command is called, which will use a default board setup (this example is for my VR lab)

Menu: 1. Display board 2. Load board from a file 3. Edit list 4. Edit Board 5. Save board to a file 6. Quit Enter your choice (1-5): 1	Nick: 3070 RTX Tim: Oculus Quest 2 Dante: Oculus Quest 1 3070 RTX Abey: Oculus Pro Oculus Quest 1

2) The program should load a file , stored in the same directory as the executable.

Menu: 1. Display board 2. Load board from a file 3. Edit list 4. Edit Board 5. Save board to a file 6. Quit Enter your choice (1-5): 2	Enter filename: **VRlab.dat** Or Enter filename: NotThere.dat Error: could not open file NotThere.Dat

3) The program allows a user to Edit an item on a list

Menu: 1. Display board 2. Load board from a file 3. Edit list 4. Edit Board 5. Save board to a file 6. Quit Enter your choice (1-5): 3 Enter the name of the list to edit: Abey: Or Can't find list Options: 1. Edit an item 2. Add a new item 3. Delete an item 4. Return to main menu Enter your option: 1 Enter the name of the item to edit: Oculus Pro Enter new name for item 'Oculus Pro': Oculus Pro A	Enter the name of the list to edit: Abey: Options: 1. Edit an item 2. Add a new item 3. Delete an item 4. Return to main menu Enter your option: 2 Enter the name of the new item: Oculus Pro B Options: 1. Edit an item 2. Add a new item 3. Delete an item 4. Return to main menu Enter your option: 3 Enter the name of the item to delete: Oculus Quest 1

State of “Abey:” list after these changes :

BEFORE	**AFTER**
Abey: **Oculus Pro** **Oculus Quest 1**	**Abey:** **Oculus Pro B** **Oculus Pro A**

4) Editing a list on the board

Menu: 1. Display board 2. Load board from a file 3. Edit list 4. Edit Board 5. Save board to a file 6. Quit Enter your choice (1-5): 4 Options: 1. Edit the name of a list 2. Add a new list 3. Delete a list 4. Return to main menu Enter your option: 1 Enter the name of the list to edit: Abey: Enter new name for list 'Abey:': Abraham: Or Can't find list	Options: 1. Edit the name of a list 2. Add a new list 3. Delete a list 4. Return to main menu Enter your option: 2 Enter the name of the new list: Star: Options: 1. Edit the name of a list 2. Add a new list 3. Delete a list 4. Return to main menu Enter your option: 3 Enter the name of the list to delete: Dante:

State of Board list before and after these changes :

BEFORE	AFTER
Nick: **3070 RTX** **Tim:** **Oculus Quest 2** **Dante:** **Oculus Quest 1** **3070 RTX** **Abey:** **Oculus Pro** **Oculus Quest 1**	**Star:** **Nick:** **3070 RTX** **Tim:** **Oculus Quest 2** **Abraham:** **Oculus Pro B** **Oculus Pro A**

5) Saving the Board to a file so you can open up it up to save

Menu: 1. Display board 2. Load board from a file 3. Edit list 4. Edit Board 5. Save board to a file 6. Quit Enter your choice (1-5): 5	Enter filename: **VRlab2.dat**

6) Quit: simply exit the program / "return 0;"

B. Requirements:

1. In writing the program the students are require to upload their progress to a git repository. They must be use https://csgitlab.ucd.ie/ if not submission are made I may either just give a 0 to the overall grade and then invite students to an interview. **Week 6 Notes .**
2. The student should add "Abey-Campbell" / abey.campbell@ucd.ie to their cs gitlab repository so we can check their Git commitment when grading to see the work was fairly distributed. If the student pair program then that should be listed in the comments.
3. The board should be made of lists, and inside of them should be items. Both of these should use a struct. **Week 5 notes**
4. The program needs to load and save to a file, this needs to be entered by the user can save to an existing file but the student must supply the file **Week 8 notes**
6. You should use a linked list for the Board. **Week 9 notes**
7. The program should be able to deal with incorrect input and state that it could not find the file, list or item . Some examples have been listed above but to keep the assignment 3 outline short I have not included every possible not found error .

C. Code Design Requirements:

- Comment your code,
- Use functions where you can but you will not be marked against you if you do not use separate files, but it is recommended to separate your code into independent modules, ideally to help with co-development
- The output does not have to match exactly what has been shown in this document. In this example, I use ":" to help me enter the lists and kept it in the names for ease. Alternatively, are possible.
- String manipulation is difficult in C ,so in this example, I assume just single line input and output in the data file. Ideally, a CSV file would be used but that would be too difficult at 1st year in C, most high-level languages though it would probably be easier to take in input that way. A student will not be marked against them if they use a CSV file or JSON but it does not result in extra marks. It's a good challenge if you find this assignment too easy, however, it will possibly be a topic of conversation when interviewed.

D. Design Hints:

1. Start by just getting a test example working , Sample test data is in Week 8 notes
2. The text does not have to match exactly, just the functionality.
3. Remember, a double link list is need here, why will become apparent when you load and save data to a file.
4. Remember, overall this is a tricky assignment to get full marks, each function is worth roughly 10%, but some are more difficult to implement than others. This is deliberate. The idea is to make it easy to pass but harder at each stage. Do not cheat on the last 10% and lose everything.

Google strcpy, strcmp, strncmp, and strstr. Get used to one of the standard online resources, e.g.
https://en.wikibooks.org/wiki/C_Programming/C_Reference/string.h/strcpy

E. Submission:

- Submit two items through Moodle,
    - an archive file (e.g., .zip or .tar.gz) containing your source module
    - a text file (.doc or .pdf) providing implementation details and comments on the design decisions you have made.
        - How do you load a board
        - How did you edit lists and items
        - How do you save a board
        - What's your Ascii art about?
        - List Git Repo , use your "group name" + " Assignment 3"

F. Evaluation Criteria

A mark [0-100] will be given according to the following criteria

- You have created an interface to interact with the program that can deal with bad inputs, and Quit (Function 6) . **(15 points)**
- Function 1 : Display test data : **(15 points)**
- Git Repo is listed within the text file **(5 points)**
- The code is well commented **(10 points)**
- The submitted text file describes your design choices appropriately **(10 points)**
- Function 2: Load a board **(10 points)**
- Function 3: Edit/Add/Delete an item on a list **(10 points)**
- Function 4: Edit/Add/Delete a list **(10 points)**
- Function 5: Save a Board **(10 points)**
- Adding a personalised piece of ascii art to the main c file **(5 points)**