Criterion A

Word Count: 436

Problem Statement:

My client needs a way to optimize the management of the various tasks he has to complete on a daily basis.

Description of Scenario:

Mr. Benmoussa is an independent entrepreneur who works from home[1]. He invests in private companies, and advises financial institutions[2]. As he works with different groups based in different time zones[3], his tasks are miscellaneously spread at times and structurally organized[4]. He additionally faces difficulty with different deadlines, priorities and overall does not have an organized system to determine the progress, completion and importance of each of his tasks. This wastes his time as he has to double check the completion of his tasks in different places, rather than having a schedule and completion organizer in place.

Rationale of Proposed Product:

The reason for a computerized program that uses graphical user interface (GUI) is to increase efficiency and the ease of organization of my client's tasks through visual depiction. By creating a visual task organizer with a 'to-do' checklist that my client will work with, he will save time[5].

This is best exemplified in a database, because it enhances the client's ease of tracking his tasks through **visual depiction** via an organized checklist, manifested on a desktop 'all-in-one' desktop app, allowing him to structure and prioritize all of his different tasks in one place.

For this program, Netbeans will be used as the development environment, due to its integrated GUI features which allows an easier, visual creation of the program as opposed to coding

[1] Hamza Benmoussa, interview by author, Bangkok, January 10 2022, transcript line A1, Appendix 2 IA

[2] Hamza Benmoussa, interview by author, Bangkok, January 10 2022, transcript line A2, Appendix 2 IA

[3] Hamza Benmoussa, interview by author, Bangkok, January 10 2022, transcript line A3, Appendix 2 IA

[4] Hamza Benmoussa, interview by author, Bangkok, January 10 2022, transcript line A6, Appendix 2 IA

[5] Hamza Benmoussa, interview by author, Bangkok, January 10 2022, transcript line A7, Appendix 2 IA

exclusively. The program is written using Java. I chose Java as it is the only programming language I am fluent in, and I am comfortable identifying and understanding errors in Java.

Furthermore, Java allows me to use the Object Oriented Programming (OOP) principles of polymorphism, encapsulation and inheritance. **Encapsulation will increase organization as I am working with many variables of similar categories. Inheritance will allow me to create a superclass to input tasks, which can extend its properties onto other subclasses such as updating task completion. Finally, polymorphism will allow me to extend properties and use a superclass, improving convenience.** These OOP techniques will allow me to create a template class of tasks, which can later be called upon or accessed in other classes, and added to a Linked List.

Success Criteria:

This program will allow the client to input his miscellaneous tasks, and display them in a table which can then be categorized in terms of urgency, how much time he has to prepare, or how many days he has remaining for a task. Additionally, he can search for information associated with a task such as its location or urgency by inputting its name into the 'Search' tab.

Having discussed with my client, I have come up with a criterion that will allow the success of this program[6].

- System must let the client input his list of tasks and remove them when finished in a continuous manner.
- Allow the user to remove/delete tasks.
- Retrieve inputted data and display tasks in a table sorted by either urgency, days remaining or time to prepare.
- Allow the creation of new tasks, and the ability to refresh his organizer after completion of tasks.
- Save tasks that are not yet finished (Ready to be outputted).
- Connect to a database to permanently save data
- Create a drop-down menu for the user to decide if a task is urgent or not
- Validate data when inputted and let the user know that their desired outcome has successfully occurred
- Calculate a task's completion based on initial preparation time divided by updated preparation time.
- Allow the user to search for a task based on any attribute inputted - such as name or location
- Allow the user to update a task/specific attributes of a task.
- Allow the user to completely refresh his organizer, to create a new list of tasks every day.

[6] Hamza Benmoussa, interview by author, Bangkok, January 10 2022, transcript line A14, Appendix 2 IA

- Easily visually represent a task if completed/how much of it is completed.
- Be friendly to the human eye, being simple, evenly spread out, functioning smoothly and communicating with the user well.