

Criterion A: Planning

The Problem

My client Emily Shao, is a grade 12 IB chemistry student. While Emily has now completed her IA for her course, she noticed challenges with the planning process of the IA for both herself and her teacher. She found that planning her IA without a guided structure was overwhelming and made the process much more difficult, causing her to procrastinate the task. She also noticed that the process for her teacher in checking the IA plans for several students proved to be much more time consuming and resulted in delays between starting and completing the IA process. Knowing these issues, Emily and I had a conversation and discussed how we could improve this situation. I chose to create a chemistry IA planner that could be used by both students and teachers to simplify the IA planning process as well as streamline the teacher's accessibility to the student's work. Since my application is set to help both students and teachers, both types of users will have access to a specific planning page that is customized with the appropriate features that are necessary for efficient planning, usability, and functionality. Students will be able to enter their personal student and IA information while teachers can simply search for their students' IAs by entering the student's student number; simplifying the process and ensuring the teacher can easily track the progress and engagement of their students.

Product Rationale

I chose Java as my main coding language due to its benefits specifically related to its object-oriented structure. Java's OOP structure includes the main features of encapsulation, inheritance, and polymorphism which allows myself as a programmer to efficiently code a project of this magnitude while ensuring that my users are given a high quality application that suits their needs. Encapsulation allows variables and methods in my code to be hidden to my users, providing an extra layer of security. Inheritance via my sub-classes for the various users and static polymorphism through my GUI methods allows for code reusability and efficiency especially considering that my program involves developing links between the two types of users present on my application. Java is also beneficial for locating runtime and logic errors present in my code through my IDE's console; helping eliminate code inconsistencies and developing a smooth-running program. My client also mentioned that they would like an app that contains simple and clear graphics as users that are less technologically advanced may run the application. This prompted me to choose Java, as its graphical user interface (GUI) is both customizable and user-friendly, improving both accessibility and usability with various features such as fields, scroll panes, and checklists. These features help keep my client organized as well as simplify the planning process, which my client has mentioned can be overwhelming and complicated when there is no clear structure or design present. Additionally, method and object creation in Java ensures code efficiency when editing and customizing containers or objects within the class, implementing both code reuse and efficiency.

Success Criteria

1. Both student and teacher users will be able to create their own accounts as well as login into their profiles to access the data pertaining to their account type
2. Both student and teacher users will be able to successfully logout of the application
3. Student users will be able to add their personal information (i.e. student email, number, level) as well as the necessary components of the IA (i.e. topic, research question, variables, materials)
4. Student users will also have access to a complete material list (already sorted with most frequently accessed near the top)
5. Teacher users can access their students' IAs by entering the student's username and student number
6. Teacher users will have a clear display of the student's IA information and can also head back to their planner and check the work of other students as well
7. The application will maintain simplicity while also ensuring the use of design customizations and appropriate graphics (i.e. fields, scroll panes, checklists)

Words: 494