

Criterion E: Evaluation

After the development of the product, it was tested against the test plan by the developer (me). This was followed by a final interview with the client where the product was evaluated using the success criteria and the test plan was implemented. This interview also served as a means to receive feedback from the client. A transcript of the second half of this interview is documented in the **Appendix**.

1. The product complies with the success criteria as follows:

Success Criteria	Result
Program will display a greeting and menu upon start	Fully met
Program will display data-entries in a nicely formatted table	Fully met
Program asks for member info and adds members if add member is selected	Fully met
Program asks for info to edit, then makes edits to member if edit member is selected	Fully met
Program will allow the user to remove a member if remove member is selected	Fully met
Program will display a submenu allow the client to take attendance	Fully met
Program displays members individually and asks for attendance then updates attendance table	Fully met
If any data entry errors occur, the program will output an error message and redirect the user as needed	Fully met
Program restarts without the user having to restart it manually	Fully met

2. Feedback from the client

After the product was fully tested by the client, he was able to give me feedback, and we then dived into a conversation about how the product could be improved. The client seemed to value the efficiency of the program and the user interface. He also pointed out a few things that increase efficiency, such as "username to email conversion."

The client pointed out a few things which could make the program better. He found it a "little odd" that the button to exit the program was called "Exit". He says it would be better if it was changed to something like "Save". Something else the client found weird was the fact there is no way to exit a function once it's started. For example, if we start the "add member" command by accident, we can't exit it without either doing as told by the program or exiting by stopping the program. This goes towards a lack of efficiency where the client would have to input random information, then delete that "member" to fix his mistake of clicking the "add member" option. These were the few nitpicks pointed out by the client. Overall, the client agrees that this program solves his problems in an efficient way as he requested.

3. Recommendations for future improvements

- As the client said when giving feedback, there should be a way to exit a certain function or command such as "take attendance" if accidentally started.
- There should be a function to search for a member and output his/her information as needed. The client should be able to input either ID or name to search for a member. The member can be searched from the member database using the "fetchone" SQLite3 function.
- The client said that people who run the club should be able to "access" and use the program. This is not possible while the program is used in the terminal or an IDE, with the database file being saved locally. The program should be uploaded to a website using "flask" or some other tool, with the database file saved to the cloud so it's accessible by others.
- Regarding the previous recommendation. Once the program is on a website it means that anyone can access it. To ensure member information stays confidential and important information regarding the club is not lost/changed, the program must be made secure. A password (one which can only be changed by the presidents) must be entered to use the program and access the data within. This ensures that only trusted people have access to the program.

Word Count: 515