Criterion E Evaluation

After finishing my program, it has been tested numerous times in accordance with my success criterias, test plans, and finally my client, Sunny Li. Small issues in syntax were consulted by my advisor, Jason Wang. I have interviewed Sunny for one last final time for important feedback and potential improvements and additions I can implement into my program to elevate its ability. The final interview can be found in the **Appendix.**

The Following Shows the Success Criterias that are Satisfied:

Program will display a working login screen	Fully Met
Program will be able to update a new password	Fully Met
Program will display a menu for the choice of revenue, spendings, and revenue vs spendings	Fully Met
Program is able to create a submenu after selecting one the choices from the main menu	Fully Met
Program is able to add new data to each respective table	Fully Met
Program is able to delete data to each respective table	Fully Met
Program is able to edit/update data to each respective table	Fully Met
Program is able to query & search for existing data	Fully Met
Program is able to state “Data is non-existent” if inputted data is not available or does not exist within the relational database	Fully Met
Program is able to error check non-applicable inputs, and redirect the user back to the interface	Fully Met
Program is able restart automatically and loop continuously without the user having the run the program	Fully Met

multiple times	
Program is able to backtrack from the submenu to the main menu	Fully Met
Program is able to exit	Fully Met

Feedback from Client

After my client has tested and researched my client, valuable and insightful feedback was given to me. The points of improvement were the structure of the inputs, exiting a process, and inclusion of loans and mortgages.

For the structure of the inputs, my client has pointed out an inefficiency where when invalid input is given, the system requires the user to input every single input again, instead of inputting the single invalid value. To add on to this, it would also provide a more user-friendly experience if the user was able to exit out of a process anytime. For example, if the user has started “update spendings”, they will not be able to exit out of the process unless all the inputs are filled in. This makes the user-interface relatively weak, as the user would be forced to input random values, and then delete those random values again. The solution to this problem would be to implement an exit option under the circumstance that a user has unintentionally chosen a process, and also create an input checking function that allows for the system to check which input the user has inputted incorrectly, and ask for only that specific input again. This would allow for much improvement in terms of user-friendliness and experience, and would give my client more satisfaction.

An extension that was recommended by my client was the ability to track and calculate loans and mortgages, which would help business as almost all businesses require mortgages and loans to start, as well as external investors.

Overall, my client was satisfied with my program and was happy with its functionality and abilities.

Recommendations for Future Improvements

- A return/back function should be implemented for accidental starts on certain processes. For example, if “Add revenue” was accidentally started, there should be a return/back option bringing the user back to the submenu.

- My client has also hinted towards the inefficiency of the input system implemented in my project, as if a single input was invalid, the entire input system would run again, and force the user to input all the needed values again. A better and more efficient solution towards this problem would be to allow the user to input only the invalid input again, saving both time and allowing for a better user experience.

- This program should be uploaded to the web using flask to allow not only my client Sunny to edit the database, but also her co-owner and co-workers, as the database is stored locally, and any changes made to the database would only apply for a single person using the program. If the database was connected to a website, and saved to the cloud, all data would be shared among her co-workers, making the program more accessible and interconnected.

Word Count: 539