**Criterion E**

# Evaluation of the product

## Have the success criteria and test plans been met?

1. Allowed my client and his employees to make their own account and login to the system through that. This created a multi-user interface where multiple users are able to enter the client details but isn't fully featured,
2. There is error checker in the signup and login window so that only the correct data is entered into the staff table,
3. There is Help section answering the possible question that the staff may have,
4. There is a Team section, allowing for staffs to see other staffs,
5. The CEO is able to delete staffs, giving access rights,
6. Staffs are able to enter client data using the forms provided,
7. Staffs are also able to update or delete existing client data,
8. Able to print receipt of the client,
9. The user is type which will allow to search for a particular client which matches with the given conditions,
10. The user can email the selected client directly from the software,
11. Changelog unfortunately could not be met,
12. User-friendly client data entry forms ensuring easy and efficient insertion of data to the database,
13. This success criterion assesses the whole program. My client is able to use this software efficiently than his previous methods and is saving time and effort.

**Word Count:** 213

# Recommendations for further development

## Client Feedback

I handed in a prototype after fulfilling my client's initial demands and meeting the original success criteria. After using it, he definitely saw some room for improvements and he followed up the previous email. The final product implements the suggested features.[1]

He suggested to add:

1. FAQs section so that he and his employees would not be confused,
2. Being able to directly email the client from the program.

## Future Recommendations and Extensibility areas

With the pseudo-multiuser system, I could have added a feature where the admin (my client in this case) can check what each of the users have inputted. If User 'A' inputs Tourist 'A' and User 'B' inputs Tourist 'B', then the admin will be able to back track who inputted Tourist 'A' and who inputted Tourist 'B'. This way, it would allow my client to solve a problem, like a mistake in the client information by finding the base of the problem. In short, having a **changelog.**

There are more features in the NetBeans that could have been utilized to increase complexity of the program. For e.g. more libraries. I could have used separate frames for each user whether they were admins or just users. This would make my pseudo-multiuser system to a fully-functional multi-user system. This way, I would have been able to add more depth to the authorization and access rights.

[1] The followed-up email is attached in Appendix 2.0

**Word Count:** 222