Criterion E

Client Assessment

Interviewing the client provided valuable feedback about the strengths and weaknesses of the system. Mr Wailes was surprised with how much needs to be written to produce a fairly simple CLI menu (+ GUI menu). He appreciated the FAQ menu that will not only immensely help (Grade-11) students coming into the university process, but also clarify information amongst teachers and AAs. The university rankings with IB, SAT and ACT percentiles were a useful addition, because the previous product (and others online) don't relate rankings directly with these important scores, mainly IB, since most US schools do not do the IB Diploma. He mentioned that the menu allows for clean navigation within the program. He appreciated the messaging feature due to its simplicity and access. Rather than opening his mail and always composing an email, this built-in method in the program allows for quick and easy messaging. It also doesn't require any salutations, formalities, etc. He liked the validation and verification in the product itself, like the unique ID. The client also recognized the importance of confirming details (y -- yes; n -- no), when he was trying the product, he had wrongly inputted a name. Selecting 'n' allowed him to reenter this information without actually storing the false one. [1]

There were a couple criticisms of the product by my client. Primary criticism includes the functionality that Bridge-U provides, mainly being external document sending. Bridge-U, and Parchment, allow for quick, official and secure sending of confidential and important documents to university institutions. As expected, the GUI is much superior for the previous product.[2]

Though, overall he was very impressed with my work and would work as an effective standalone product for the school.

[1]Wailes, Gareth, 2021
[2]Wailes, Gareth, 2021

Evaluation of System against Success Criteria

Success Criteria	Evaluation
System should accomplish main objectives of: ● Storing grades, university choices and applications ● Displaying information about universities and processes ● Inputting/Editing information about universities -- location/country, name, application date, method, etc	Client was able to successfully complete the following main objectives with no issues. As mentioned previously, I did inform him about the commenting of loadAAObjects when first time starting. Client could see the effect of Serialization and GUI and its importance, and all main objectives were met.
System should feature a secure log in page/menu which will redirect to a separate page depending whether user is AA or student	Client was easily able to create and login account, through the menu of receiving user input of first & last name, ID, username and password.
System's login page should store/save login details	Through the use of Serialization, the client was thoroughly satisfied with the storing of login details.
System should allow AA to create and assign student accounts	After setting up AA's profile, he was easily able to add student through the add student method.
System should prompt users for a profile setup	After AA creates student accounts, student receive a mandatory prompt to setup their profile. This ensures all students have accounts.
System should allow inputting values, notices/messages and documentation	Client reassures that all inputs are correct with the correct validation. After setting up profiles, users are free to explore the menu to input values or send messages.

System should allow deleting, editing and changing grades, documentation (AAs only), choices, profile (students only), etc	See above. AA actions are differentiated from Student actions due to different classes. This allows for a smoother and cleaner program.
System should have data verification and validation and prohibit input of invalid data (for appropriate fields). E.g, only numerical data for grades in specific range (range check, type check)	For all necessary inputs, verification and validation allow for valid and correct data to pass such that it makes the program function better, faster and makes it easier to sort. It prevents unknown errors from disrupting the program.
System should have GUI(s) to support easier navigation, readability and functionality	GUI allow for aesthetically pleasing interfaces that help (non-tech) users to function and navigate much better. Client was very pleased with GUI and encouraged for more.
System should have a section where students can input self-reported testing.	Along with inputting personal information like application methods, pursuing major, there is input for self reporting tests. This has a crucial caution of "do not enter false information as universities will check this, and your application may be rescinded!", which is true.
System should have a 'Review' page where every component can be viewed, for AAs and students.	A review page is helpful in several ways. It allows for clarification of data entered, and if wrong, user can go back and edit data. It also allows a big-picture view of students to quickly see their attributes rather than having to go through the menu again.[3]

[3]Wailes, Gareth, 2021

System should contain error messages, warning and exceptions.	Exception and error handling are crucial in a large project like this. The client appreciated the planning of exceptions that prohibit the invalid data, random bugs and glitches in the code.
System should allow AAs to search students by ID number and/or by name	Searching through the ArrayList provided extremely useful when display student's profile. The client also stated "this feature is valuable because it would be handy when actually submitting/reviewing an application from a student"[4]
System can allow comments or communication between teachers/AAs and/or student	This extra feature was taken with a positive response. As stated above, it allowed for quick messages increasing efficiency and functionality
System should be less than 5 MB	This makes it easy to transfer files to others if they need it. System was 1.6MB

Recommendations for future improvements

Client Recommendations
Though difficult, Mr Wailes recommended me to try and predict grades. This would be done with machine learning and historical data, which can be obtained directly from AAs since they have every student's predicted grades and final grades. A place to store both grades -- teacher-given and machine learning, can be beneficial, as if there is a discrepancy of 2 or more grades, then the AAs will definitely look into it.[5]

[4]Wailes, Gareth, 2021
[5]Wailes, Gareth, 2021

Universities like University of Toronto, Durham & Oxbridge(Oxford & Cambridge) University have colleges which students apply to directly, as opposed to just the university. Thus, for universities that have colleges, the client has suggested an additional menu option of applying to a specific college.[6]
Passwords are visible when typed in, Through a GUI, this could be avoided and would increase security if entering password in public.[7]
Sorting students by homebase would be useful because AAs are assigned homebases which contain students. This way AAs can easily view, edit, etc.[8]
Automatic account creations rather than manual creation. My Client recommended to implement a .csv to create student accounts instead of manually inputting them.[9]

Self Recommendations
Add functionality of adding grades along with prediction using machine learning, from language R. I could also take historical data from AAs and teacher and analyse which students got admitted in which universities with which grades (predicted and final).
Use more data in FAQ menu and university rankings for a full and thorough display of information for all skill levels. Not everyone can make in top universities. To reduce code and redundancy, I limited to few universities in uniRank.
Creates and stores a .csv or .pdf file of the overview/summary page. This can help externally while storing files, instead of always running program to view. Creating a file per student may also be feasible for organisation.

[6] Wailes, Gareth, 2021
[7] Wailes, Gareth, 2021
[8] Wailes, Gareth, 2021
[9] Wailes, Gareth, 2021

Even though, there may not be a need to reset passwords. This functionality may be useful if users forget passwords. Having a secure way to reset user passwords increases the security of the system. This could be done using a GUI and JPasswordField from Java Swing.
A minor change could be to successfully implement text wrapping in the GUI, since it becomes complex when tried before. This would allow users to have a more favourable experience and not confuse them with broken words.

Client and Self:

Implement more GUI throughout to help many "not-technically gifted humans[10]", that will allow easier navigation, and "less training to use the command-line menu[11]" for clients. GUI may be implemented along with other techniques like a GUI Login page with Serialization and Encryption.

Incorrect values may be entered, especially Strings like University names, application method, countries, etc. This may mislead AAs due to human error. So, a searchable dropdown menu, using Java Swing's JComboBox, would solve this problem. The list would have different university names, and can be expandable by adding more names. This could again work with outside elements like .csv which is "predominant[12]" within the AA department.

Word Count: 402

[10] Wailes, Gareth, 2021
[11] Wailes, Gareth, 2021
[12] Wailes, Gareth, 2021