

Appendix A: Evidence of Interaction

Note: All conversations were done online through messages because of the pandemic and social distancing

1st conversation with client (through gmail)

Me: Hello, good afternoon!

Client: Hi, I noticed that you are interested in designing a new student timetable program for our school. Are you still interested?

Me: Yes, I am very interested. I have developed many video games in the past but I have never really designed any type of database program yet. This student timetable project will help me develop my first database and help me expand my skills in programming.

Client: That sounds great! I have some specific features that I want you to include in this database. First of all, do you have any questions?

Me: Yes, my only question “why do you need another student timetable program when we already have one in the school for semestered and non-semestered timetables?”

Client: That is a great question, one that I was going to address later on. Basically, the current student timetable program that we have includes a few bugs and lacks some features which I believe are essential. Since the creator of this timetable program, Vance Scott, has left this school, I need new student timetable programs to cover the features that were missing in the old timetable program.

These missing features include:

- Ability to sort the names of the students and teachers in the database
- Including the **proper** prerequisites for all courses offered in our high school this year

- Ability to see the history for which students have been added/removed from the database. This allows us to add a student back into the database if a student was accidentally removed
- Ability to see the contact information of the students and teachers in our school in a more user-friendly manner

Of course, along with these missing features mentioned above, it is also important to cover the features that are necessary in all student timetable programs, such as a login page, a quit button, etc.

Me: Ok, that seems interesting to program. I will try my best to create a student timetable program that incorporates all of the required features. Thanks and have a great afternoon!

1st conversation with advisor (through discord)

Me: Hey, I am currently creating a student timetable program for my ICS IA. I am going to create an Object-Oriented program using Java for my IA. Do you have any ideas of how I should structure my program and what important built-in Java classes I should use?

Advisor: First of all, remember that if you are going to use an advanced data structure (ADT) for your program, it's important to create your own class for it instead of using Java's built-in LinkedList, Queue, Stack, etc. So, if you want to implement a linked list to keep track of all the students which you added, I advise you to create your own LinkedList class for this. In terms of layout, I would suggest a login page which directs you to the main menu page. On this main menu page, there should be many buttons which perform many different functions, such as adding a student. Important built-in Java classes that I would recommend using are JButton, JLabel, JTextField, Choice, and RandomAccessFile.

Me: Alright, thanks for the info! I will start designing a prototype for my program as soon as possible with some of your suggestions.

2nd conversation with client (through gmail)

Me: I have sent you the design document for my student timetable program. Let me know as soon as possible for your opinion on my design document.

Client: Thank you for your effort. I have read your design document and really enjoyed the simplicity in which you portrayed some of the features in your program. However, I do have several suggestions for improvement.

Here are my suggestions for improvement:

- Make a separate page for “History of actions” and a separate page for “Contact information of student & teachers” instead of combining them in one page.
- I do not want the history and contact information of the students and teachers stored in a notepad. I recommend creating a new page that pops up in the program when I want to view the history or contact information.
- Try to make your timetable a little more aesthetically pleasing. For example, you could include formal image backgrounds. As you know, the guidance department will constantly be using timetable programs to generate student timetables, and we do not want to get bored :)
- For the “Add a student” page, instead of providing a text field for the student to insert their grade, include 4 buttons called “Grade 9”, “Grade 10”, “Grade 11”, and “Grade 12.” I should be able to select one of these buttons to assign the student their grade.
- Separate the “Edit current student(s) and teacher(s)” page into two different pages of “Edit current student(s)” and “Edit current teacher(s)”.

Me: Ok, that is very insightful feedback. I am curious about the process in which you want to assign a student’s completed and current courses. Do you want me to include a page with many buttons with each button representing a course and left-clicking the button assigns the course to the student?

Advisor: I was initially thinking about using a scrollbar representing all of the courses, but now that I think about it, I like your idea even more. Sure, go for it. Make sure to refer to myblueprint.ca to view all possible courses offered at our school.

Me: Ok, got it, thanks. Have a great day.

3rd conversation with client (through gmail)

Me: I have now attached the executable jar file for version 1 of my student timetable program. I tried my best to include all of the features which you mentioned before in our previous conversation.

Client: I have played and fiddled around with your student timetable program and I really liked the image choices which you used for the background. However, I do have some feedback that I would like to see in version 2 of your timetable program. The feedback is listed below:

- In the main menu screen, include something that indicates the current # of students in the school, the current # of teachers in the school, and the # of teachers required.
- I want an option to clear the history text from the history page.
- As guidance counsellors, sometimes we forget what each course code stands for because there are many courses in our school. I would like an option to easily view the course description and prerequisites for a particular course code.

Me: Ok, I am confident that I can fulfill the first two requests which you made. However, I would need some time to think about a way to fulfill your third request. Anyways, thanks for the feedback and have a good afternoon.

2nd conversation with advisor (through discord)

Me: Hey, Mr xx, the ideas that you provided me before were pretty successful. I have now attached version 1 of my final solution for my HL ICS IA. Let me know what you think about it.

Advisor: First of all, let me tell you that I really enjoy the simplicity of the program. However, I have spotted a few bugs in your program. The first bug that I discovered was that the `RandomAccessFile` class which you have created does not always save the data which I entered. The second bug which I found was that when I sort the list of students and exit to the main menu, a student is randomly deleted from the list. The last bug which I found was that the deletion of a student/teacher does not always work successfully. For example, if I delete a student, then sort the list of students, then exit to the main menu, and then view the list of current students, it shows that the deleted student was not deleted. Apart from bugs, logic errors

which I found were that it is not possible to insert a student with a middle name and that it is possible to create multiple accounts for the login screen.

Me: Oh, wow, I had much more bugs and errors in my program than I expected. I will try my best to fix these issues later on. In the meantime, do you have any advice on ways I can improve the functionality of my solution?

Advisor: I have a few suggestions. In my personal opinion, there should be a button on the main menu which allows the user to view all of the courses offered at the school. Also, in the page where you insert the student's courses, when you right-click on a specific course code, it should show the course description and prerequisites. By the way, I really liked how you created your own LinkedList class and involved inheritance with it. The hashing algorithm that you used for the login screen usernames and passwords was also intriguing.

Me: Ok, thanks very much, have a great day!

4th conversation with client (through gmail)

Me: Hello, I have attached the final version (version 2) of my student timetable program. I managed to find effective ways of fulfilling the requests which you made in our previous conversation. I hope that you find my program practical.

Client: I can see that you have made some significant changes to your program, changes which I liked and found pretty interesting. Overall, I like this version of your program much better than the initial version because it has much less bugs and it fulfills many of the features of my ideal student timetable program.

Me: Do you have any future extension ideas for my program?

Client: Yes, here they are:

- Allowing the option to assign teachers to teach specific courses
- Allowing the option to assign a teacher as interested in teaching the pre-IB or IB program
- Provide the option to sort the list of students and teachers alphabetically by their last name

- Allowing students to select if they are in specialized programs apart from just academic, pre-IB, and IB programs. For example, a student could be in the applied program or program for international students.

Me: Ok, I will try to include these features when I have the time for it. I have one final question. Did you find my login screen user-friendly and did you find my program simple to use?

Client: To answer your first question, I definitely did find your login screen user-friendly. In terms of simplicity, I definitely found it much simpler to use than previous timetable programs which I used. So, yes, your timetable had some really interesting ideas that allowed it to become simple to use.

Me: Ok, thanks so much for your advice, have a great day!