INTRODUCTION

This project is aimed at developing an expense tracking software with the final goal of being able to determine various statistics about student expenditures across various schools. Through those statistics, we were hoping to be able to determine meaningful things such which school could potentially be more affordable.

GROUP COLLABORATION

Collaboration:

Regarding collaboration, it was a bit difficult at first since we did not know each other and had a few clashing schedules. We started off meeting through zoom but it was not very productive so we eventually started meeting in person to work on the project. I contributed several things to the project. Firstly regarding the topic on which we would base our project, then with the design of data structures which were used in the first demo (arraylists/hashmaps , …). I contributed in the implementation of data storage and also designed it in a way that we could access that data from main for the statistics calculation without having illogical relation between different data such as students/schools.

Challenges:

One of the main challenges faced when working as a group was deciding on how to split the work and still complete your part without knowing how your implementation could conflict with the logic used by other group members. To overcome this, we realized that regularly committing work on Git would keep all of us updated on how the others are progressing and complete our tasks in a synced manner.

TECHNICAL REFLECTION

Development Phases:

During the procedural programming stage, It helped me understand a lot more on how useful Hashmaps are as they were not used in assignment 1. At first it was challenging to make sure I can link expenses to a specific student and then be able to retrieve them all logically to get stats for students individually or include them in calculations for schools. Hashmaps for lookup with id made this relatively easier and then the use of methods to do calculations for individual stats and then use them in a loop for overall school statistics made it simpler. When implementing the OOP design, I learnt that it is extremely easier to do calculations for students if you have them as objects rather than storing them as an array object. With OOP we were able to encapsulate the attributes of students and add/retrieve expenses just by instantiating a student object and using the getters/setters. When implementing the user interface I learnt about how to link my different stages (for adding students/schools) to the main GUI stage. Although it was challenging to make sure that the various controls and containers were well organized on the screen because at first everything was scattered. However the use of vertical and horizontal boxes was extremely helpful in order to keep the appearance well aligned.

Application of Concepts:

The use of an abstract class of expenses was highly inspired by the lectures on abstraction and the use of an abstract spaceship in assignment 2. Enums were highly useful for the expense classes in order to always distinguish between groceries, rent and tuition. The assignment 3 user interface was extremely useful as it shared certain features such as loading or saving. Concepts such as enhanced for loops and file reading/writing which we learnt in class were also indispensable to implementing a functional program.

Reflection on Tools and Technologies

Git:

Git facilitated collaboration as I could keep up to the advancement in my teammates' tasks which helped us design our tasks so as to avoid logically clashing methods. There were a few design decisions I disagreed on with my group members so GIT was also helpful for us to redesign certain functions and then just merge. However, we faced several merge conflicts for example as we made packages for our demo 2, there were conflicts which could only be fixed locally through intelliJ. We were eventually able to merge through intelliJ by merging the current branch on the ide into the remote main branch.

Junit:

As for Junit unlike Git (which is really hard to fix conflicts), it is a tool that I really enjoyed using as it gives a satisfying feeling of verifying that what you created works. At first, although running tests individually passed, running them at once was failing, so this brought about the intuition to instantiate a new Data class before each test/Run to avoid unwanted data to remain. To get the average spending by school, we first tried to do all calculations within that method but through testing we realized that it would be more useful to implement a total spending by school method which could then be used for both getting the average spending and also getting the most affordable school in the menu.

Lessons Learned

Project Outcome:

Although it met the goal we set for ourselves, one thing I would have done differently would be starting it earlier. This could have helped avoid facing several challenges we did just because we were under pressure.

Skills Development:

This project helped me improve my problem solving skills as at some steps the code would return a few illogical outputs which always pushed me to find design strategies which would make it run correctly but also efficiently. As someone who is not used to teamwork, this was an experience that helped improve that and it also helped me get the hang of Git a bit as I barely understood what it was during the lecture.

CONCLUSION

Personal Growth:

In conclusion, this project was an exciting introduction to what development is. It was an eye opener that collaboration can be extremely difficult as people disagree on design ideas. Getting stuck at the first stage which was finding a topic was already where we had to problem solve but the problem-solving involved in coding and fixed errors while maintaining a good logic of the code was more involving. It helped realize that problem-solving and good logic intuition are essential for coding and that once you have those you could implement a code in various ways and still find a way of fixing any error.

Final thoughts:

I find the project extremely meaningful and that the course is well designed as a lot of the assignments paved the path for the project which in turn for me makes me look forward to future courses that involve software development. It showed me that software development is not only about writing code and that there is more exciting and complex tools/tasks that are involved in it.