**G52GRP Individual Report**

|  |  |
| --- | --- |
| Project title | A decision-making system for engineers to synthesize a wastewater treatment plant |
| Date | 13/03/2022 |
| Group ID | 2A |
| Name | Tan Zhun Xian |
| Student ID | 20313854 |
| Username | hcyzt1 |
| Other Group Members | * Lee Pei Yi * Lim Zi Qin * Ng Kar Yan * Liew Zi Xian * Kaori Choi Wai Teng * Khoa Vu Nguyen Anh |
| Supervisor | Mr KR Selvaraj |

**Introduction**

This report is based on a project given by the University of Nottingham Malaysia Campus in our Software Engineering Group Project (SEGP) module to be completed over the course of a year. My team and I were given this assignment to work on with a client to design a Wastewater Decision-Making System. The project's purpose was for students to have a better understanding of team management, coding skills, research skills, presentation skills, report writing, and, most significantly, how work is done in the real world with actual clients. This is a report on my contribution to the project and my reflections on it. It includes a description of the work we've completed, as well as my input and thoughts on the project.

**Summary of the work**

**What we did**

Our team managed to create a wastewater decision-making system with the following functionalities:

* Add new wastewater technology to the system
* Modify and delete wastewater technology from the system
* Select only certain wastewater technology to generate results
* Generate all possible combinations of user-selected wastewater technology in a five-step system
* Show users the best combination of user-selected wastewater technologies in each category such as cost, cleaning efficiency and area and comparing it with the current selected combination of technologies

**Why we did it**

In our current day and age, water pollution has become a very serious issue. This is contributed by the fact that wastewater treatment is often neglected by factories and other industries as it is often very expensive. Couple this fact with the lack of government surveillance, people would just dump their wastewater into the nearest water source such as lakes and rivers as it is much cheaper and there are no consequences to their actions. So, we decided to design a wastewater treatment decision system to take the tedious task of designing a wastewater treatment plant out of their hands and automate the process. We hope that with this system in place, people would be more incentivised to actively treat their water waste as it is now more convenient.

**Summary of Individual Contribution**

* **Initial Code**
  + After the meeting with the client, I was quite free. This was due to the fact that it was the start of the semester and thus our workload was still quite light. Having a lot of free time on my hands, I decided to try creating the initial code for the software. With my limited Java skills from last semester, I managed to cobble together a prototype for the software in a few short days. The prototype was text-based and has no GUI so it was not visually appealing. Users also had to enter numbers using the keyboard to select choices so it was quite bothersome. However, it performed adequately as a proof of concept for the software. Its influence can still be seen and felt in the later versions of the software.
* **Reworked Code**
  + After the holidays, the second semester of the school year started. At that point, I had completed my Software Maintenance module and had learned some valuable concepts such as code maintainability. So, I decided to look at the code with my more experienced eyes. What I saw in my code shocked me to the core. From a maintainability standpoint, my code was horrendous and would definitely fail industrial standards. So, I decided to rework my code to adhere to the principles of maintainability. It took a few days of hard work to restructure the code. I also decided to add a simple AI search method to the software. The search method in question is the uniform cost search and is used to find the best combination of technologies in the wastewater decision-making system. The addition of this AI search method was to satisfy the requirement of the project to incorporate AI into our software. In the end, what I managed to create was enough to satisfy me. I showed my new and improved code to my group members and they were also impressed with it. Thus, we decided to use this new code as the base of our software.
* **Maintenance**
  + After completing the demo for the software, we now had to rigorously test the software to root out any bugs or issues so that it can be fixed before presenting it to the clients and examiners. It was a group effort and we could not have done it without the help and cooperation of the team. At first the bugs were quite numerous, needing the combined efforts of both the frontend and backend members to cooperate and exchange notes to even understand why the bug was occurring in the first place. There was one infamous bug where we could modify then delete a wastewater technology but not the other way around as the program would crash. In the end, we managed to pin down the issue to forgetting to refresh the page after each operation and finally fixed the bug. Through our perseverance and also some sleepless nights, we were able to fix all the bugs that we uncovered before presentation day.
* **Refactoring**
  + After finishing the GUI for the software, we decided to refactor the code to improve readability and maintainability. So, I took it upon myself to start refactoring the code since I was most familiar with it. I first started by adding comments to explain the code and changing method names to be self-explanatory. Then, I started to remove redundant pieces of code which were introduced during code and GUI integration. After that, I combined different classes together with similar functionalities to reduce any dependencies. Finally, after a few long months, we had created our first working demo.

**Reflection of the Software Project**

**What was achieved**

In the end, we managed to implement all the functions specified by the client in out wastewater decision making system:

* Add new wastewater technology to the system
  + By entering all the relevant information about the new wastewater technology, users can easily enter the new technology into the system
  + Sliders were provided to the users to enter the TSS, COD and BOD cleaning efficiencies as they only run from 0 to 1 to improve user experience
* Modify and delete wastewater technology from the system
  + Users can modify wastewater technology by double-clicking the value, changing it and then clicking the modify button
  + Deletion is much easier as the user needs to only select the technology to be deleted and click the delete button
* Select only certain wastewater technology to generate results
  + Users can select only certain wastewater technologies to focus on in the results. The unselected technologies are not taken into consideration when calculating the results
  + Users only need to double-click on the technology to select or deselect it while Select All and Delete All buttons are also provided for convenience
* Generate all possible combinations of user-selected wastewater technology in a five-step system
  + Users must first enter the initial pollution levels of the wastewater by manually typing or using one of the predefined values provided by selecting some predefined areas
  + Users can see all the possible combinations of their selected wastewater technology in action and see their results
* Show users the best combination of user-selected wastewater technologies in each category such as cost, cleaning efficiency and area and comparing it with the current selected combination of technologies
  + Users can compare the current selected combination of wastewater technologies with the best combination in each category by double-clicking on it
  + Users can select to see the best combination of wastewater technology in a specific category and compare the results with the current combination

**What was not achieved and why**

Although we managed to create a software which the client was satisfied with, we did not manage to link our software with a remote database, more specifically Jupiter and CPanel. We tried to link the database for months, asking for help from our seniors, lecturers and also making visits to the IT Department of the university. In the end, we had to make the choice to pull the plug on the remote database due to time constraints to focus on other aspects of the project. We settled for using a text file as our database. It was a poor substitute for the remote database but it worked so we were satisfied. We suspect that the issue might be due to the changing IP address of the university WiFi which was implemented as a security measure after a serious data breach. In the end, it can be summarised that the reason our database was not implemented was due to a shortage of time to fix the issue.

**Critical Discussion on the Running of the Project**

Our project was quite rushed as we had other coursework to complete simultaneously with our SEGP project. So, time management was absolutely critical to the success of our project. By implementing the scrum methodology in our project, members were able to choose to complete the parts of the project where they were more proficient. This however created a situation where some members did a lot more of the work while some members were relatively free. However, we as a group have discuss this issue and have since made sure that job allocations were much fairer.

**What you have learnt from the project**

The most important thing that I have learned from this project is time management. As our SEGP project runs in parallel with other modules, we inadvertently have to make time for other modules to study the material and complete their coursework. If time management is poor, then we may fall behind in some of the modules and do poorly in their respective coursework. In the worst case, we might do poorly in all our modules and get held back. However, the worst part of poor time management is actually the burden we place on our fellow group members. As we fail to meet our deadlines, the other members have to pick up our slack and do our part of the work and this is very unfair to them.

**What you would do differently if you were starting over again**

If I could start over again, I would make sure I had better time management. I did not face any issues in time allocation and thus none of my modules were negatively affected by all the increased workload of the second year. Nevertheless, I believe that with better time management, I could have put in more work into the project and contribute just a little more. In the end, we were able to create a software that we as a team and our client were satisfied with. However, with more time, I believe we could have created something more impressive.

If I could start over again, I would also make sure that more time is allocated to solving the database issue. Even though a database change may seem minor in the grand scheme of things, with a remote database in place, we could handle large sets of data much easily. The risk of data loss due to corruption can also be mitigated as the data is saved online and not in a easily corrupted text file.

**What I could do better**

Towards the end of the project, my Java skills as a programmer have improved exponentially. This is due to the fact that I gained a lot of experience during the coding of the software. I also attribute this to the fact that I took the Software Maintenance module last semester. With a more experienced eye, I confess that my coding of the software could have been better. Principals such as code readability, maintainability and MVC could have been implemented much better in the code. The AI method implemented in the code was Uniform Cost Search. With a better grasp of Java, I could have implemented a more advanced AI method and this would have improved our software immensely. However, with the deadline fast approaching, I could not rework the entire code and expect my team members to suddenly understand it in a short amount of time. So, no drastic improvements could be made to the code. In the end, I believe that I could have done better in the coding aspect of the project.

**Appendix**

**Peer Assessment**