COMP8860 Project Questionnaire

This questionnaire lets us assess how you've done during the week — and gives you an opportunity to reflect on what you've been up to. You can use the questions as a guideline to write your report.

A. Requirements

1. What approaches did you consider to improve the accessibility of the HEAT system?

Adding more font size choices and options to change the colours of the screen. Also adding more accessibility with some audio files and the possibility of adding a magnifier. We discussed adding text to speech too, but this would be difficult within the time frame. Sound warnings were an option we considered, along with making HEAT completely keyboard accessible.

2. How did you decide which approaches to work on, and which were unimportant or impractical to attempt in the time given to you?

We looked into adding text to speech however while we did a test implementation it ended up slowing HEAT down significantly and we decided a week was not long enough to address the issues. After considering our user stories, we also decided to discard the sound warnings and instead focused on a small amount of text to speech, the magnifier, the theme selector and adding more customisable font sizes.

3. Have you documented what and how should be modified in HEAT?

We had meetings where we used a shared google doc to share our ideas and add to each other’s where we had any suggestions. During our morning meetings every day, we updated each other on what we completed and planned to work on and if anything didn’t work or we found major problems we discussed as a team how we could address them and if we had the time available to fix it before working on the other tasks. When addressing how to modify the code for HEAT, after identifying which class we had to edit, we would use a find function to locate where exactly needs to be edited.

B. Design

4. How the design of the new features was generated from the requirements? Was the design documentation created before or after you have started coding?

5. How have you reconcile the design of the user interface with the software design?

6. How have you documented the design of the new features?

C. Implementation

7. Was there always an obviously correct way to implement each feature you chose to add, or did you need to consider multiple options? If the latter, how did you choose?

When it came to merging, we faced a lot of code conflicts as originally we used the main method to house our additional code. We had to consider multiple options, and concluded on creating new classes to clean up the main method. We decided on which features were more important and discarded the code causing us the most problems and we chose methods with the simplest codes so they were easier to understand and merge. It was also important to consider how the code would affect the run time of the application as when we had tested text to speech it had caused considerable lag within HEAT.

8. How did you locate the relevant parts of the HEAT code to modify? Did you discover any other issues with HEAT while you were working on it? If so, what did you do

We inspected the code closely to understand the functionality of each class and how they link to other relevant classes. From there, we utilised Eclipse’s built-in search function to assist us in locating where certain methods and instances are. While we were working on the HEAT source code, we had initially wanted to utilise JavaFx’s media player to provide a short text-to-speech function at the beginning when HEAT launches. However, after facing difficulties implementing the JavaFx libraries, we switched to try with Javax audio system. Yet, we were unable to implement the feature due to audio format incapability and decided to put aside that feature due to time constraint.

10. Which features have you successfully implemented?

We added a theme selector with a pop up, to choose the theme on opening, and additional options for changing font size.

D. Development Process

11. What methodology did you follow (i.e., what were the steps in the development process you used)? Did you make any deviation from the methodology? How did the particular requirements of this project influence your choice of development process?

We chose agile and scrum, deciding to allocate each member a task to implement a feature each. We reviewed our user stories and found we had to focus more on specific visual impairments. Regarding changes, we stayed using agile and scrum, adapting and changing tasks where necessary which felt the best way to take on the tasks we chose. Due to the short time frame this felt more suitable. We decided which features were most important to implement and made sure it was functional by the deadline. We looked at other methods such as waterfall however this would not fit into the project constraints as it's more reliant on the completion of one task's completion before the other, which would hold up our project at a few points as we wait for each task to be completed.

12. How was the customer involved in your development process?

We asked questions on their requirements and how the features we implemented worked with them. This made us revisit our user stories to make them more suitable for the customer. We also checked in again to make sure we were headed in the right direction and asked for their opinions again if there was anything else that might have needed to be added. For the functional testing, their input was very helpful as it helped us find any potential bugs the user may face.

13. How did you distribute the work between the members of your team? Was everybody happy with the amount of work they had to do?

We had a group meeting on teams before the week started to allocate the roles, this gave us time to think over what we had been allocated to bring any concerns up first thing Monday morning. After deciding on all of the tasks we wanted to complete, we divided them up between us making sure we picked tasks we were most comfortable with completing and knew we could finish within the week. We were all happy with the spread of work as it meant if some of us had more work to finish, the others could start working on the documentation, which has saved us a lot of time.

E. Tools

14. How did you use issue tracking tools and GitLab during the week?

We used GitLab for version control and merging. Each member had their own branch to keep the code separate before the merge which allowed for changes to be made beforehand. Due to our separate branches, we tracked issues individually, working on them as they came up if we could, trying to fix most of the major problems before any of our merges.

16. Did you use other tools (excepting GitLab)?

We used Lucidchart to make our UML diagrams.

F. Quality Assurance

17. How did you draw up testing procedures? Did you make any use of automated testing (e.g. unit tests)? Was there anything that was particularly difficult to test?

We leveraged JUnit testing after we'd completed the coding for each features. We performed our JUnit tests on features that were more significant to the accessibility we wanted to add, as opposed to the other functional features we had introduced to improve on our previous implementations.

18. How did you make use of code inspections? Did they help to spot any problems with the changes you'd made?

We organised our code inspections in a way where the individuals who did not participate in the coding of a particular feature were tasked to check the codes manually. They were instrumental in identifying minor errors in indentation and commenting which assisted the respective coders to correct their codes. Besides that, the inspectors also played an important role in providing feedback on user experience and recommendations on further improvements which could be implemented. This has assisted with building on top of existing ideas to introduce more cohesive features.

19. How confident are you that your changes to HEAT meet the customer's requirements? Why?

We had regular meetings with the customers to check on our progress and made any necessary changes to the best of our abilities based on the customers' feedback and satisfaction. The customers' feedback has proved very valuable as they assisted us to ensure that our development process and ideas were aligned with the objectives of this project. This has allowed us to implement our features with confidence that they'd be fitting to the customers' visual impairment requirements.

G. Summing Up

20. Were you happy with your modifications to HEAT — both in terms of design, and in terms of code quality? Was there anything you'd do differently given more time?

We are happy with the modifications we managed to add, however some of them don't have full functionality. If we had more time we would have tried to fix the major issues we had to discard, such as the magnifier merging problems and the issues we had with the JAR from the audio. With regards to code quality, our final product has had good results with regards to maintainability as it is easy to understand the implementations we have added.

21. What went *wrong* during the week? What went *right*? What seemed particularly useful to you?

We had problems with the magnifier and audio not working. The magnifier did not merge correctly and cause several issues so had to be discarded. The JAR file for the audio stopped working a few days in and became unable to run the mp3 files so they had to be commented out in the code. The theme selector also doesn't change the console window theme. However, the font size options work completely and the theme selector works in the other windows, like help and option. Doing a 1 week sprint will be something that can definitely be mentioned during interviews and job applications as even though most sprints are longer than a week, working with tight deadlines and having to prioritise what tasks are most important are both good skills to have.

22. Do you have any other comments on how the project went?

One week is too tight of a deadline and there is a lack of clear instructions. We haven't learnt all the relevant content yet either, making project planning hard so we wasted a lot of time at the beginning. This was especially a problem because of the one week deadline.