**Sprint 3 Report**

*Group Contributions*

*Project Board*

Graphical user interface, text, application, email

Description automatically generatedCompared to Sprint 1 and 2, the project board for this sprint contained more user stories as many of the features had been partially completed but not yet finished. Many of the user stories had their corresponding features completed in Sprint 2 but only in the backend, so most of the project board tasks were frontend related. New GUI pages and their controllers were created, for example for the Staff and Manager to use, and linked to their corresponding functions in those classes.

Since Sprint 3 was the final sprint of this project, all the user stories from the product backlog in addition to any unfinished stories from the previous sprint were added to this sprint’s backlog. Since some of the user stories from Sprint 2 were partially completed, we needed to re-evaluate the story points for these user stories in addition to the newly added ones. Similar to the previous sprint, we utilized PlanitPoker in order to assign user story points to each of the user stories in accordance with what each team member believed with regards to their difficulty and complexity of implementation. This was achieved by having each team member rate the user stories provided by the Product Owner, followed by a discussion of that user story to determine if the points value needed to be adjusted. After all user story points were determined, user stories were allocated to members in roughly equal difficulty and also based on the confidence of the member to complete the story within the Sprint 3 deadline.

Graphical user interface, text, application, email

Description automatically generated

*Git commit history*Text

Description automatically generated

Text

Description automatically generatedText

Description automatically generatedText

Description automatically generated

Graphical user interface, application

Description automatically generatedThe following section will display the commits done to the git repository during the duration of the third sprint. The evidence shown in this section will display approximately 139 git commits which were made and for each collection information will be conveyed illustrating the work done during the commit.

Firstly, multiple tests were added for the Cinema and Booking class. Through writing and running these tests logical mistakes in the code were able to be found and certain code logic was able to be fixed. Further, work was done on the functionality of the staff in the back end being able to modify the list of movies shown on the GUI.

Graphical user interface, text, application, email, Teams

Description automatically generated

Graphical user interface, text, application, email

Description automatically generatedAdditionally, the work was now done on the front end for the staff controller. This was in regard to the staff having their own screen on the GUI with the options they are able to select. Commits were also done which implemented a logout button for any user on the GUI enabling them to logout back to the login page whenever they desired.

Graphical user interface, text, application, email, Teams

Description automatically generatedGraphical user interface, text, application, email

Description automatically generatedOnce the work was completed on the staff controller and logout function the tests were written to ensure there was no logical errors within the code. Following this the staff/manager functionality of getting the upcoming movies was done. This now meant as in the requirements staff/managers now had a button which they can press to fetch the upcoming movies until the following Monday in the database.

Testing was then done on the staff and manager class regarding the previous changes which had just been added. Following this testing simple fixes were made to the logic. One of these fixes included staff/managers being able to edit cast members for each movie. This is able to be done when the program has detected that a staff or manager is logged in.

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email

Description automatically generatedFurthermore, additional tests were written regarding the Cinema class as up to this point in the sprint this class possessed the most amount of untested code. Just like the outcome of the other tests it was highlighted again why testing is crucial as a number of tiny edges cases were able to be discovered in the code and therefore fixed. One of these fixes included the booking function now updating the number of tickets left in each section “front”, “middle” and “rear” so that a movie was able to be fully booked out. Another major develop from this collection of commits is that the functionality of staff/managers being able to edit movie information was now complete. Hence, staff/managers were now able to edit the title, cast, director, synopsis, location, time and number of seats available for each movie.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generatedFollowing on, from the commits made to allow staff/managers to edit movie attributes the further functionality of them being able to add and delete movies was implemented. This enabled one to add/delete movies and for the GUI to update in real time so that the changes can be seen straight away. The work done regarding this was both done on the front and backend of the program. Additionally, the final tests were written for the Movie class so that all lines of logic were now tested in the file.

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email

Description automatically generatedFirstly, for the next collection of commits the logic for gift cards was added. This included staff/managers being able to maintain the gift cards. Hence, this was the functionality that enabled staff/managers to see the list of gift cards available at the moment with the user able to add and remove cards as well as see the balance on each card. The project team also ensured that the user could only add valid cards thus, error checks are done before the gift card is able to be added. Additionally, the user is able to see the redeemed status of the gift card and are able to redeem the gift card themselves to enable it’s use in later transactions done by customers. Once all the backend logic was done for this functionality the GUI pages were written therefore allowing the group to make certain that the requirements were being completed.

Graphical user interface, text, application, email

Description automatically generatedThere was only now a couple sprint backlog items left to complete. One of these was the cancel transaction functionality which was now implemented meaning that if a customer wanted to cancel their booking they were able to. As well, when cancelling the booking on the GUI customers were prompted with two options to select as to why they were cancelling their booking. This was then later able to be seen by the managers. This logic was also implemented whereby a manager had the option on the GUI to see cancelled transactions where a file displaying all the cancel transactions including their reason, date and time were now obtained.

Graphical user interface, text, application, email

Description automatically generatedAdditionally, further fixes were made to the upcoming movies and cancel transactions functionality. These fixes were made upon errors found when running and testing our code. Once these fixes were implemented further tests were written to ensure the coverage was the desired amount.

Lastly, final touches were made to the code for little edge cases that were discovered. All the product backlog ideas regarding sprint 3 were completed at this point. These last commits therefore convey the final polishing of the code that was done getting ready for the demo.

Overall, the final sprint included wrapping the whole project up and finishing all the requirements that were given to the group. Everything was able to be completed in the sprint 3 backlog conveying that it was a successful sprint. Additionally, the team was able to complete the tests that needed to be done to achieve the desired code coverage required.

*Burndown Chart*

*Completed User Stories*

*Jenkins*

*Code Test Coverage*

[add test coverage image]

The final code Test coverage achieved for the project was 86%. Hence, by completing such a high number of tests the group was able to ensure that majority of edge cases were covered by the program produced along with making certain that the main logic was correct. As mentioned previously the importance of writing tests was highlighted in this project as constant minor errors were found and then fixed when writing the tests. By getting group members to consistently test the logic as they were writing it the group was able to work efficiently in the testing component of the project. Overall, the group was delighted with the coverage of the testcases and were confident that the program produced runs correctly due to this.