**Sprint 1 Report**

Group Contributions

The evidence of commit history seen below (Figure 1) shows the contributions of group members in their respective roles and demonstrates our implementation of Agile practices. Branches were used where necessary, for example to separate testing code from application code, and regular communication regarding what was to be done and estimated times of completion was helpful to keep everyone up to date.

Outside of these commit logs, tasks were managed in the Project Board (see below – Figure 2) by the Project Owner, in order to keep on track for the overall goal of delivering working software over multiple sprints. Organising the tasks to be done in this way broke down the application into manageable chunks that could be evaluated in terms of how effectively they met individual requirements of the client.

Text

Description automatically generated

Text

Description automatically generated

Figure 1: Log of the commit history over Sprint 1, showing the work done by group members.

Project Board

The project board (Figure 2) held the user stories corresponding to required features of the application. After discussion on which were the highest priority tasks, some were moved to the Sprint 1 Backlog to be completed in the first sprint. By writing these in the voice of the customer, guest user, staff or manager we ensured that all user needs would be covered.

From a client’s point of view, the first point of contact when running the application would be the initial screens for logging in as a member or guest, followed by the screen to view available movies. These were prioritized most highly for the first Sprint in order to deliver a working application.

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Figure 2: The Git project board showing the product backlog of tasks for the 3 sprints, as well as tasks chosen for Sprint 1 which were moved along the board to track progress.

Next Steps

The group decided to make explicit assignments of tasks between members, in order to divide the workload more evenly and better plan for the amount of work that could be completed within the sprint. Reflecting on the contributions of Sprint 1, adjustments will be made to make sure that all group members are completing an equal amount of work in Sprint 2.

**Git commit history**

Application

Description automatically generated with medium confidenceThroughout the duration of sprint 1 the following git commit history reflects the work done on the project by the group. There was a total of 66 commits submitted across the week of development.

The following aspects of the project were worked on during this sprint. Since this is the first sprint for the assignment task initially the repository was created and tested to ensure the initialisation was successful. Following this the project backlog was added to as well as the sprint 1 backlog. Hence, the tasks for sprint 1 were identified and the develop of the code began.

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated with low confidence

Firstly, the basic structures of the Java classes we expected were needed were written. This included the creating of Java files with each class having its own variables, constructor along with getter and setter methods. This foundation would therefore be able to be built upon for the completion of each task within our sprint 1 backlog. The rest of the git commits for this day were done for the login and book function of our sprint one backlog. By adding functions within the classes, the back end for a user logging along with a basic book function was completed just pending the implementation on the front end of the GUI.

Graphical user interface, text, application, email

Description automatically generated

Majority of these git commits above were for implementing the how movies are read into the cinema class from JSON files. This contribution would aid in any tasks requiring the customer to be able to view the movies listed for each cinema as well as when the customer wanted to make a booking.

Graphical user interface, text, application, email, Teams

Description automatically generatedGraphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Additionally, all this code written required tests hence, over this day unit testing scripts were developed so that the group could ensure that the code written so far was working correctly. This brought to our attention some bugs in the code including logic and syntax errors which were easily able to be fixed therefore, ensuring the tests were all successful. Following this more code was written to begin implementing the GUI so that we had a display that a customer was able to interact with. Initially the login page displaying login, register, and login as guest were developed.

A picture containing application

Description automatically generated

Lastly, for the development of the code for sprint one the front-end GUI was created. This was the final step in displaying that all the backend develop that had taken place over the first sprint could be displayed. The GUI which was created enabled the customer to login or register an account, or login as a guest if they wanted to. All options lead to a listing screen where the user can see all the movies being displayed. Further, the customer could sort the categories of the movies alphabetically if they desired. Unfortunately, not all categories for a movie were able to be filled in and the filtering of categories is a step which will be

completed in the following sprints.