SE 3XA3: Test Report
Title of Project

Team 35, PGH
Hamid Ghasemi and ghasemih
Pratyush Bhandari, bhandarp
Gazenfar Syed, syedg1

December 4, 2018

Contents

List of Tables

List of Figures

This document ...

1 Functional Requirements Evaluation

The aim of these tests is to make sure that user is able to use the software according to the given requirements. These tests will include sorting testing, searching teasting, accessing the summary, rate and date released, and trailer testing.

Test Name: FRSR
Results: The user is able to sort movies based on rate, popularity, and released date.

Test Name: FRSE
Results: The user is able to search a movie by using movie name.

Test Name: FRAC
Results: The user is able to access movie's summary, rate and released date.

Test Name: FRTR
Results: The user is able to watch a movie trailer by simply selecting movie's trailer.

Table 1: **Revision History**

Date	**Version**	**Notes**
Dec3	1.0	Notes
Dec4	1.1	Notes

2 Nonfunctional Requirements Evaluation

2.1 Usability

2.1.1 GUI Testing

The Graphical User Interface (GUI) was tested by 7 students from Mac who are not in Software Engineering but they were interested to see our apps (From other programs) to reflect the technological experience of the potential users for this program. All the participants observed the timing it took them to perform the requested tasks, and they see the difficulty of the softwar, and how long would take them to get their hands around the app. At the end of the section they all gave us feedbacks about our apps and evaluate the difficlty and performance of our application.

Test Name: SS-1
Results: All participants were able to successfully complete installing the program to their personal phone (they all had android phone) within 2-3 minutes.

Test Name: SS-2
Results: All participants were able to successfully complete the task of searching a favorite movie by inputing the movie's name .

Test Name: SS-3
Results: All participants were able to successfully the task of sorting movies based on rate, released date and popularity.

Test Name: SS-4
Results: All participants mentioned that all the tasks which they have done above were really easy and so simple at the same time. Their feedback was that our app is easy to install, easy to use, understandable and in overal simple. They were all satisfied but some of them suggested that the software can be improved. For instance one of participants said that software can have an option to sort the movies which will come out within 6 months. This way people can realize which movies are coming out. Moreover, they enjoyed using the app since it is touchscreen so it makes a connection with the user.

2.1.2 Media Output Testing

The program was installed into phone with an android system operator. The results of attempting to launch the program were noted.

Test Name: SS-5
Results: The program installed on phones that has android OS and there wasn't any problem.

2.2 Performance

2.2.1 Screen Speed Performance

The preformance was calculated based on how long the app takes to perform user requests and if the app run smoothly.

Test Name: SS-6
Results: The time to perform user requests was less than 2 second from all participants.

2.3 Output

2.3.1 Media Output Testing

The time between finding a movie and retrieving the data from dataset was calculated. In addition to that the aim of this test was to check if the output is the expected output. And each time different movie requested by participants to ensure that data, and output file is consistent. Plus, the calculation between sorting movies were noted down in this section.

Test Name: SS-7
Results: Movies were seatrched by participants retrieved within approximately 3 seconds.

Test Name: SS-8
Results: Sorting movies based on rate, popularity, and date released took about 5 seconds and it outputed list of movies based on those categories.

3 Comparison to Existing Implementation

In our app, we used retrofit which is one of the library of the android studio. It is a powerful framework for authenticating and interacting with api and sending network requests. Retrofit made our app easier to code since it can be used to create json object and then we used that in our implementation, using it in our java implementation. This is one of the main difference between our code and existing implementation.

4 Unit Testing

5 Changes Due to Testing

6 Automated Testing

7 Trace to Requirements

8 Trace to Modules

9 Code Coverage Metrics

References