Test Report: Flick Picker

Team 7, 7eam
Talha Asif - asift
Jarrod Colwell - colwellj
Madhi Nagarajan - nagarajm
Andrew Carvalino - carvalia
Ali Tabar - sahraeia

April 6, 2023

1 Revision History

Date	Version	Notes
March 7	1.0	Added FR requirements - Talha
March 8	1.1	Added unit test table - Ali
March 8	1.2	Added some unit tests - Ali
March 8	1.3	Reformatted unit test tables - Jarrod
March 8	1.4	Added all API unit test information - Jarrod
March 8	1.5	Added missing UI unit test information - Ali, Jarrod
March 8	1.6	Added unit tests from Look and Feel T1 to Security T2 - Andrew
March 8	1.7	Added information in the changes due to testing, automated testing, and code coverage metrics sections - Jarrod
March 8	1.8	Added trace to modules - Madhi

2 Symbols, Abbreviations and Acronyms

symbol	**description**
T	Test
FR	Functional Requirement

Contents

1 Revision History i

2 Symbols, Abbreviations and Acronyms ii

3 Functional Requirements Evaluation 1
3.1 Authentication Requirements 1
3.1.1 FR 1 1
3.1.2 FR 2 1
3.1.3 FR 3 1
3.2 Profile/Group Requirements 2
3.2.1 FR 4 2
3.2.2 FR 5 2
3.2.3 FR 6 2
3.2.4 FR 7 2
3.2.5 FR 8 2
3.3 Recommendation Requirements 2
3.3.1 FR 9 2
3.3.2 FR 10 3
3.3.3 FR 11 3

4 Nonfunctional Requirements Evaluation 3
4.1 Look and Feel Requirements 3
4.1.1 Appearance Requirements (10.1.1) 3
4.1.2 Style Requirements (10.1.2) 3
4.2 Usability and Humanity Requirements 4
4.2.1 Ease of Use Requirements (10.2.1) 4
4.2.2 Learning Requirements (10.2.3) 4
4.3 Performance Requirements 4
4.3.1 Speed and Latency Requirements (10.3.1) 4
4.3.2 Safety-Critical Requirements (10.3.2) 4
4.3.3 Precision or Accuracy Requirements (10.3.3) 4
4.3.4 Reliability and Availability Requirements (10.3.4) . . . 4
4.3.5 Robustness or Fault-Tolerance Requirements (10.3.5) . 4
4.3.6 Capacity Requirements (10.3.6) 5
4.3.7 Scalability or Extensibility Requirements (10.3.7) . . . 5
4.4 Operational and Environmental Requirements 5

4.4.1 Requirements for Interfacing with Adjacent Systems (10.4.3) . . . 5
4.5 Maintainability and Support Requirements . . . 5
4.5.1 Adaptability Requirements (10.5.3) . . . 5
4.6 Security Requirements . . . 5
4.6.1 Access Requirements (10.6.1) . . . 5
4.6.2 Integrity Requirements (10.6.2) . . . 5
4.6.3 Privacy Requirements (10.6.3) . . . 5

5 Comparison to Existing Implementation **6**

6 Unit Testing **6**

7 Changes Due to Testing **13**

8 Automated Testing **13**

9 Trace to Requirements **13**

10 Trace to Modules **13**

11 Code Coverage Metrics **13**

List of Tables

1 Unit Tests Pt. 1 . . . 7
2 Unit Tests Pt. 2 . . . 8
3 Unit Tests Pt. 3 . . . 9
4 Unit Tests Pt. 4 . . . 10
5 Unit Tests Pt. 5 . . . 11
6 Unit Tests Pt. 6 . . . 12
7 Nonfunctional Requirements Traceability Matrix . . . 14
8 Functional Requirements Traceability Matrix . . . 15
9 Trace Between Modules and Tests . . . 16

List of Figures

1 Example of Code Coverage Metrics . . . 13

This document contains the report for Flick Picker, where the details were documented in the VnV Plan. It covers the evaluations for Functional and Nonfunctional requirements, as well as the testing results for each test outlined in the VnV Plan.

3 Functional Requirements Evaluation

Every developer conducted an ad-hoc review of the functional requirements to ensure every single one of them had been met. Failing to fulfil any of the functional requirements will be implemented by the final demonstration. The ordering for these functional requirements is the same as the order in the SRS and used in the VnV Plan's traceability matrix.

3.1 Authentication Requirements

3.1.1 FR 1

Users are presented with a sign-up/login screen on launch if they have not previously signed into the application before. From there they can sign-up or login with the respective button and by entering their credentials. If a user has previously logged into the application, not signed out, but instead just closed the website, their authentication is stored in the browser's cookie so it immediately redirects into the user's home page. This functional requirement is fully met.

3.1.2 FR 2

From the sign-up/login page, it does not have the functionality to sign-up/login through Google or Facebook. This functional requirement is not met and needs to be implemented.

3.1.3 FR 3

The user can click "Log Out" from anywhere in the application in the header to successfully log out. This functional requirement is fully met.

3.2 Profile/Group Requirements

3.2.1 FR 4

The user can navigate to the Account page by clicking the respective button in the header, from which they can change their username, email, or password. This functional requirement has fully been met, however might need to be revisited as the application allowed emails to be changed but it is not detailed in the functional requirement itself.

3.2.2 FR 5

The user can navigate to the Preference page by clicking the respective button in the header, from which they can change their preferences. This functional requirement is fully met.

3.2.3 FR 6

The user can navigate to the Create Group page by clicking the respective button on the home page, from where they can create a group with a name for the group. This functional requirement is fully met.

3.2.4 FR 7

The user can invite friends into a group by navigating to it's Group page, and a user can accept invites into a group by navigating to the Social page in the header. This functional requirement is fully met.

3.2.5 FR 8

The user can navigate to the Social page by clicking the respective button in the header, from which they can send friend invitations to other users by email or username. This functional requirement is fully met.

3.3 Recommendation Requirements

3.3.1 FR 9

The user can see an ongoing list of shows by navigating to a Group or the Just Me Page. This functional requirement is fully met.

3.3.2 FR 10

The user is given a list of shows in a group, where they can vote with "Like", "Neutral", or "Dislike" to voice their preference in the group. This functional requirement is fully met, however the wording might need to be revisited as the same phrasing has not been used.

3.3.3 FR 11

The user and group can see the current recommendation based on the votes by clicking the view result button in the Group Page. This functional requirement is fully met.

4 Nonfunctional Requirements Evaluation

Every developer also conducted an ad-hoc review of the nonfunctional requirements to ensure every single one of them had been met. Failing to fulfil any of the nonfunctional requirements will be implemented by the final demonstration, or otherwise rectified. The ordering for these nonfunctional requirements is the same as the order in the SRS and used in the VnV Plan's traceability matrix.

4.1 Look and Feel Requirements

4.1.1 Appearance Requirements (10.1.1)

The application starts on a login or sign-up screen, which is better than what is currently described in this nonfunctional requirement. It will need to be updated to reflect such. The rest of the details are entirely met.

4.1.2 Style Requirements (10.1.2)

There is a colour scheme and formatting for the front-end which has been adhered too and this this nonfunctional requirement is met. However, the layout has to be revisited and further refined.

4.2 Usability and Humanity Requirements

4.2.1 Ease of Use Requirements (10.2.1)

This nonfunctional requirement is met, but can be revisited from feedback by users as “simple” is not universally defined for individuals.

4.2.2 Learning Requirements (10.2.3)

This nonfunctional requirement is met, but can be revisited from feedback by users.

4.3 Performance Requirements

4.3.1 Speed and Latency Requirements (10.3.1)

This nonfunctional requirement is met. However, further development has to be aware of this requirement and not cause immense slowdown.

4.3.2 Safety-Critical Requirements (10.3.2)

User’s private data is safely stored with Firebase and thus this nonfunctional requirement is met.

4.3.3 Precision or Accuracy Requirements (10.3.3)

The recommendation provided is based on user votes in the group, thus this nonfunctional requirement is met.

4.3.4 Reliability and Availability Requirements (10.3.4)

The application is not deployed and thus is not currently available to users, and this nonfunctional requirement is not met. However, once it is deployed it will meet the requirement.

4.3.5 Robustness or Fault-Tolerance Requirements (10.3.5)

User’s in a Group are constantly fed show recommendations, this nonfunctional requirement is met.

4.3.6 Capacity Requirements (10.3.6)

Every user session is different and thus this nonfunctional requirement is met.

4.3.7 Scalability or Extensibility Requirements (10.3.7)

Firebase allows easy scalability and thus this nonfunctional requirement is met.

4.4 Operational and Environmental Requirements

4.4.1 Requirements for Interfacing with Adjacent Systems (10.4.3)

This nonfunctional requirement is met as Flick Pickeris adhering to standard web development principals.

4.5 Maintainability and Support Requirements

4.5.1 Adaptability Requirements (10.5.3)

This nonfunctional requirement is met as Flick Pickeris adhering to standard web development principals.

4.6 Security Requirements

4.6.1 Access Requirements (10.6.1)

The user can only access their own data from the database thus this nonfunctional requirement is met.

4.6.2 Integrity Requirements (10.6.2)

User data is safely and properly stored in the cloud thus this nonfunctional requirement is met.

4.6.3 Privacy Requirements (10.6.3)

This nonfunctional requirement is met.

5 Comparison to Existing Implementation

Not Applicable for Flick Picker.

6 Unit Testing

Table 1: Unit Tests Pt. 1

Test Number	**Referenced Requirement**	**Input Given**	**Expected Output**	**Actual Output**	**Result**
Look and Feel T1	Appearance Requirements 10.1.1	User signs up with email, logs in, and navigates through all possible button/link paths	Each page that corresponds to a button clicked will be successfully displayed with all possible changes to the UI	All pages and UI changes were displayed when the user attempted to navigate with little to no confusion	Pass
Look and Feel T 2	Style Requirements 10.1.2	User logs in and explores the different screens and services	User will provide feedback on style choices for the UI	The colour pallete was consistent, though the UI is slightly empty and could have more, such as a background image	Pass
Usability and Humanity T 1	Ease of Use Requirements 10.2.1	User logs in and explores the different screens and services	User is able to go through the system without the need of assistence or misunderstanding the affordances of different screens and their buttons	The user was able to navigate through the UI and the functionality offered by different screens without difficulty	Pass
Usability and Humanity T 2	Learning Requirements 10.2.2	Users will fill out a survey of how easy or complicated their experience with the system was	The surveys are returned with the responses filled out	The system was seen as being easy to use, with little confusions on affordances and functionalities	Pass

Table 2: Unit Tests Pt. 2

Test Number	**Referenced Requirement**	**Input Given**	**Expected Output**	**Actual Output**	**Result**
Performance T 1	User Input Responsiveness 10.3.1	User will enter inputs by typing into textboxes or clicking buttons	The system will give the appropriate response to the given input and respond in a timely manner that is below 1 second	Text boxes and buttons work as intended with a timely responsiveness below 1 second	Pass
Performance T 2	Safety-Critical Requirements 10.3.2	Inputing user data to the signup and login and then navigating the pages	Verification that user info is not being improperly used or displayed	User info is not needlessly output on any page where it ought not be	Pass
Performance T 3	Precision or Accuracy Requirements 10.3.3	Clicking button to generate suggested media	User assesses accuracy of the suggested shows/movies, with the media matching the input requirements	The suggested media matched the requirements entered by individual users and groups of users	Pass
Performance T 4	Reliability and Availability Requirements 10.3.4	Visiting the webpage	Connection to the server is a success or failure	User was able to consistently connect to the server when loading the webpage	Pass
Performance T 5	Robustness or Fault-Tolerance Requirements 10.3.5	Saving a set of unique or few preferences	Appropriate suggestions will be returned for individual and group preferences for limit cases	The system returned appropriate suggestions for limit cases of preferences for individuals and groups	Pass
Performance T 6	Capacity Requirements 10.3.6	Several users login at the same time	Verification of the system's ability or inability to accommodate the simultaneous sign-ins	The system was able to sign-in and serve the multiple sign-ins	Pass

Table 3: Unit Tests Pt. 3

Test Number	**Referenced Requirement**	**Input Given**	**Expected Output**	**Actual Output**	**Result**
Operational And Environmental T 1	Interfacing with Adjacent Systems 10.4.1	Developer will use the system as a regular user would on different web browsers	The system will have the same fundamental functionalities on all browsers, with leniency for minor differences in visual presentation	The system worked on various browsers such as safari, chrome, and firefox	Pass
Maintainability and Support T 1	Adaptability Requirements 10.5.1	Developer will use the system as a regular user on a mobile device	Verification of the system's adaptability to the mobile screen and size	System is not optimized for mobile use and is more difficult to navigate	Fail
Security T 1	Access Requirements 10.6.1	Developer will click on their profile and will view and change settings	Verification of the ability to change user's email, password, and username while being unable to view other users' private info	Developer was able to view and change their profile settings while being unable to view the settins and info of others	Pass
Security T 2	Integrity Requirements 10.6.2	Developer will change profile settings or preferences	After different incrememnts of time after the changes are made, developer will verify that the changes persist	The changes to the account and preferences persisted, though the username required a page refresh to display the change	Pass
UI Auth T 1	FR 1	Typing in a registered username and password and clicking the log in button	Home screen is pulled up on main menu, giving user access to program features	Expected Output	Pass
API Auth T 1	FR 1	Backend does not deal with user login	N/A	N/A	N/A

Table 4: Unit Tests Pt. 4

Test Number	**Referenced Requirement**	**Input Given**	**Expected Output**	**Actual Output**	**Result**
UI Auth T 2	FR 1	Typing in an incorrect username and password and clicking the Log In button	Message displaying failure to log in	Nothing happens	Fail
API Auth T 2	FR 1	Backend does not deal with user login failure	N/A	N/A	N/A
UI Auth T 3	FR 1	Clicking button to register after entering an email and password	Message displaying confirmation of successful registration	Nothing happens	Fail
API Auth T 3	FR 1	User account information is submitted to the addUser API	A user with the same information now exists	Expected Output	Pass
UI Auth T 4	FR 1	Clicking button to register after entering an invalid email and /or password	Message displaying a failure of registration	Nothing happens	Fail
API Auth T 4	FR 1	Backend does not deal with user sign-up failure	N/A	N/A	N/A
UI Auth T 5	FR 3	Clicking Log Out button	Taken back to login page	Expected Output	Pass
API Auth T 5	FR 1	Backend does not deal with logout	N/A	N/A	N/A
UI Auth T 6	FR 1	Clicking OAuth button on login page	Login screen with OAuth providers	N/A	Fail -Yet to be implemented

Table 5: Unit Tests Pt. 5

Test Number	**Referenced Requirement**	**Input Given**	**Expected Output**	**Actual Output**	**Result**
UI PG T 1	FR 4	Clicking Account button and modifying account credentials	Profile page loads, account credentials can be successfully altered	Expected Output	Pass
API PG T 1	FR 4	Call the user service to change a user's username	User's username matches the new name	Expected Output	Pass
UI PG T 2	FR 5	Clicking Preferences button and modifying preferences	Preferences page loads, prefences of Genres, Type, and Minimum Rating can be successfully altered	Expected Output	Pass
API PG T 2	FR 5	Call the update preference service with a new set of preferences for a user	User's preferences now match tnew set of preferences	Expected Output	Pass
UI PG T 3	FR 6	Creating a group with the Create Group button	Group appears in Home screen beside other groups with its given name	Expected Output	Pass
API PG T 3	FR 6	Group creation is sent to backend with group name and owner ID	Owner ID and Name matches input and group ID is created	Expected Output	Pass
UI PG T 4	FR 7	Joining a group with the Join Group button	Clicking the Join Group button shows the list of invites. Clicking the checkmark on an invite adds the user to the group	Expected Output	Pass

Table 6: Unit Tests Pt. 6

Test Number	**Referenced Requirement**	**Input Given**	**Expected Output**	**Actual Output**	**Result**
API PG T 4	FR 7	A test user is added to a group	Group contains the test user	Expected Output	Pass
UI PG T 5	FR 8	Inviting another user with the Invite to Group button	Upon clicking on the Social button, the user faces a menu where they can select a group and friend to invite to the group	Expected output	Pass
API PG T 5	FR 8	Two test users, an invite is sent from one to the other	An invite exists with the first test user as the sender and the second test user as the recipient	Expected Output	Pass
UI R T 1	FR 9	Entering the Home page, via logging in or clicking the Home button	The home page should be filled with a rotation of suggested movies	None	Fail
API R T 1	FR 9	A group with test members is created and a voting session is started	A set of recommendations is received	Expected Output	Pass
UI R T 2	FR 10	Hovering the mouse cursor over a suggested movie	A hidden widget of a 5-star bar should appear, allowing the user to input a remembered rating for the movie	None	Fail -Yet to be implemented
API R T 2	FR 10	A mock group voting session is started and two test users submit votes for the first recommendation	The voting session has a record of these votes	Expected Output	Pass
UI R T 3	FR 11	User clicking view best match	Seeing an appropriate resultant movie	Expected output	Pass
API R T 3	FR 11	A mock group voting session is completed after voting	The media with the highest vote rating is selected as the best match	Expected Output	Pass

7 Changes Due to Testing

Selenium tests were replaced with manual testing due to the inefficient testing available with Selenium and the sufficiency of manual testing. Authentication testing for the API was removed due to the UI covering all test cases and the API covering none.

8 Automated Testing

Some unit tests are automated, refer to V&V Plan.

9 Trace to Requirements

Refer to the Appendix.

10 Trace to Modules

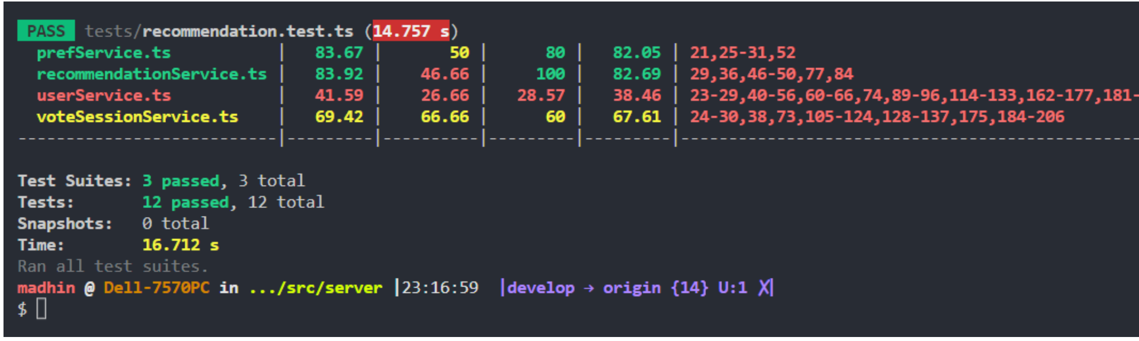
Refer to Table 9 in the appendix.

11 Code Coverage Metrics

Code coverage for frontend and backend tests is covered through Jest. Test suites will be run for each pull request which will cover the success or failure of tests along with code coverage.

Figure 1: Example of Code Coverage Metrics

Coverage Report BE.png



```
 PASS  tests/recommendation.test.ts (14.757 s)
  prefService.ts              |   83.67 |       50 |      80 |   82.05 | 21,25-31,52
  recommendationService.ts    |   83.92 |    46.66 |     100 |   82.69 | 29,36,46-50,77,84
  userService.ts              |   41.59 |    26.66 |   28.57 |   38.46 | 23-29,40-56,60-66,74,89-96,114-133,162-177,181-
  voteSessionService.ts       |   69.42 |    66.66 |      60 |   67.61 | 24-30,38,73,105-124,128-137,175,184-206
 -----------------------------|---------|----------|---------|---------|-----------------------------------------------

Test Suites: 3 passed, 3 total
Tests:       12 passed, 12 total
Snapshots:   0 total
Time:        16.712 s
Ran all test suites.
madhin @ Dell-7570PC in .../src/server |23:16:59  |develop → origin {14} U:1 ✗|
$ 
```

Appendix

Traceability Matrices

Table 7: Nonfunctional Requirements Traceability Matrix

Test Number	**Requirement**
Look and Feel T 1	10.1.1
Look and Feel T 2	10.1.2
Usability and Humanity T 1	10.2.1
Usability and Humanity T 2	10.2.3
Performance T 1	10.3.1
Performance T 2	10.3.2
Performance T 3	10.3.3
Performance T 4	10.3.4
Performance T 5	10.3.5
Performance T 6	10.3.6
Operational and Environmental T 1	10.4.3
Maintainability and Support T 1	10.5.3
Security T 1	10.6.1
Security T 2	10.6.2

Table 8: Functional Requirements Traceability Matrix

Test Number	**Requirement**
UI Auth T 1	FR 1
API Auth T 1	FR 1
UI Auth T 2	FR 1
API Auth T 2	FR 1
UI Auth T 3	FR 1
API Auth T 3	FR 1
UI Auth T 4	FR 1
API Auth T 4	FR 1
UI Auth T 5	FR 3
API Auth T 5	FR 3
UI Auth T 6	FR 2
UI PG T 1	FR 4
API PG T 1	FR 4
UI PG T 2	FR 5
API PG T 2	FR 5
UI PG T 3	FR 6
API PG T 3	FR 6
UI PG T 4	FR 7
API PG T 4	FR 7
UI PG T 5	FR 8
API PG T 5	FR 8
UI R T 1	FR 9
API R T 1	FR 9
UI R T 2	FR 10
API R T 2	FR 10
UI R T 3	FR 11
API R T 3	FR 11

M1: Hardware-Hiding Module

M2: Behaviour-Hiding Module

- M3: Native Login Module
- M4: Friends Module
- M5: Groups Module
- M6: Profile Module

M7: Software Decision Module

- M8: Matching Algorithm Module
- M9 OAuth Login Module
- M10: API Module

Table 9: Trace Between Modules and Tests

Module	**Tests**
M1	
M2	
M3	UI Auth (T 1-6), API Auth (T 1-6)
M4	
M5	UI PG (T 4, 5), API PG (T 4, 5)
M6	UI PG (T 1-3), API PG (T 1-3)
M7	
M8	UI R T 4, API R T 4
M9	
M10	UI R (T 1-3), API R (T 1-3)

Reflection

Completing the VnV Plan and Report highlighted the effectiveness of the VnV Process. By first completing verification on the system, we got to clearly document the requirements that were thoroughly implemented, the requirements that were missed, and the requirements which were not defined properly and needed to be revisited. Doing so, the SRS and other documents stay fully updated and all the developers stay on the same page regarding the goals of Flick Picker.

The other aspect of the VnV Plan were the tests planned for the system, which the report implements. However, a handful of them were misclassified and do not belong in automation testing or duplicates of other tests that could just be a single integration test. While it is important to test each component individually, which unit tests do, having multiple integration tests which do the same thing is a waste of time and resources. By going through the Plan to fill out the Report we found cases which the latter applied, and were subsequently changed to ensure full coverage in our system with developer time allocated appropriately. What's left for verification is to get the CI/CD Pipeline running after the tests have all been merged.

References

7eam. Module guide. https://github.com/TAsif/Team7Project/blob/develop/docs/Design/MG/MG.tex, 2022a.

7eam. Module interface specification. https://github.com/TAsif/Team7Project/blob/develop/docs/Design/MIS/MIS.tex, 2022b.

7eam. System requirements specification. https://github.com/TAsif/Team7Project/blob/develop/docs/SRS/SRS.tex, 2022c.

?

7eam (2022c)
7eam (2022a)
7eam (2022b)