

Test Report: Flick Picker

Team 7, 7eam

Talha Asif - asift

Jarrold Colwell - colwellj

Madhi Nagarajan - nagarajm

Andrew Carvalino - carvalia

Ali Tabar - sahraeia

March 8, 2023

1 Revision History

Date	Version	Notes
March 7	1.0	Added FR Tests - Talha

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	6
8	Automated Testing	6
9	Trace to Requirements	6
10	Trace to Modules	6
11	Code Coverage Metrics	6

List of Tables

1	Nonfunctional Requirements Traceability Matrix	7
2	Functional Requirements Traceability Matrix	8

List of Figures

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

7 Changes Due to Testing

8 Automated Testing

9 Trace to Requirements

Refer to the Appendix.

10 Trace to Modules

11 Code Coverage Metrics

Appendix

Traceability Matrices

Table 1: 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 2: 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

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.