Software Requirements Specification for Flick Picker: Group Show Finder

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

April 6, 2023

Contents

Revision History

Date	Version	Notes
Oct 1/22	0.0	Copying Volere Template and updating sections
Oct 1/22	0.1	Adding Sections 1, 3, 5
Oct 2/22	0.2	Adding Section 6
Oct 5/22	0.3	Adding Section 2
Oct 5/22	0.4	Adding Section 7
Oct 5/22	0.5	Adding Section 4
Oct 5/22	0.6	Adding Section 8, 10.1, 10.6
April 4/23	1.1	Updating document based on feedback excluding NFRs

1 The Purpose of the Project

1.1 The User Business or Background of the Project Effort

The business aims to make finding shows across groups of friends with differing preferences easier by streamlining the process from start to finish. As a quick summary, users will be able to input their preferences, make groups with their friends, and the group will get recommendations based on all their preferences. In addition, an opportunity arose in the market after COVID passed as large groups can create host gatherings for any events, one such that this application can select.

1.2 Goals of the Project

Goals can shift as development continues, entirely losing scope on the developers' passion for why the application started. Thus there will only be a few immediate goals to capture the passion that developers currently have and will continue to uphold in the future:

- Provide a means of choosing a movie, TVshow, or anime to watch immediately in a large friend group, which they all enjoy
- Users can be recommended a movie, TVshow, or anime individually as well.
- Minimize the number of advertisements while using the application

2 The Stakeholders

2.1 The Client

The clients for Flick Picker are Dr. Spencer Smith and the Teaching Assistants of 4G06, who will be responsible for a few key milestones in the development process. The client will be responsible for project approval, including, but not limited to, the approval of the general idea, scope, and complexity. Additionally, the client will also provide feedback throughout the development on various deliverables.

2.2 The Customer

The customers for Flick Picker are individuals who watch movies, TVshows, or anime as an individual or a group who wish to find catered recommendations.

2.3 Other Stakeholders

The members of 7eam are other stakeholders falling under the categories of:

- Designers and developers

- Testers
- Systems engineers
- Technology experts
- System designers
- Usability experts

2.4 The Hands-On Users of the Product

The hands-on users of the product are the same as listed under the customer section.

Individual Watchers

The individual watcher is a user who will simply input their preference settings and then browse the movies, TVshows, or anime that fit the preferences.

Group Watchers

Group watchers also input their personal preferences. Additionally, they must also create a group and add the other members. With the group created, members can browse the movies, TVshows, or anime that best fit the combined preferences of the group.

2.5 Personas

N/A

2.6 Priorities Assigned to Users

N/A

2.7 User Participation

N/A

2.8 Maintenance Users and Service Technicians

N/A

3 Constraints

3.1 Solution Constraints

There are no specific constraints the stakeholders have asked to be on the product regarding the solution. So instead, 7eam deems it necessary to have a general form of constraint on the development of Flick Picker that is the following:

- Flick Picker must have in-depth tests on any part of the application before releasing it to the users and also follow a strict and healthy software engineering process

There are no specific technological constraints. However, the technology stack must follow industry standards.

3.2 Implementation Environment of the Current System

The application will be deployed on a website, so the restrictions of a browser follow.

3.3 Partner or Collaborative Applications

N/A

3.4 Off-the-Shelf Software

The list of movies, TV shows, and anime will be polled from two separate APIs, which is the following:

- MyAnimeList for data on anime
- OMDB for data on TV shows and movies

3.5 Anticipated Workplace Environment

As Flick Picker is a browser application, 7eam expects users to use the application on their desktops or phones. Thus the application must be optimized for both possible settings, relying on a responsive layout. However, the users' physical workplace has no bearing on the design, as they can use it whenever a computer or phone with internet is available.

3.6 Schedule Constraints

The schedule is the same as the one Dr. Smith has provided us with

Team Formed, Project Selected	September 19
Problem Statement, Development Plan	September 26
Requirements Document Revision 0	October 5
Hazard Analysis 0	October 19
V&V Plan Revision 0	November 2
Proof of Concept Demonstration	November 14–25
Design Document Revision 0	January 18
Revision 0 Demonstration	February 6–February 17
V&V Report Revision 0	March 8
Final Demonstration (Revision 1)	March 20–March 31
EXPO Demonstration	April TBD
Final Documentation (Revision 1) - Problem Statement - Development Plan - Requirements Document - Hazard Analysis - Design Document - V&V Plan - V&V Report - User's Guide - Source Code	April 5

3.7 Budget Constraints

Based on the outline of SFWRENG 4G06, the expenses for this application must not exceed $500. There are five developers working on 7eam.

3.8 Enterprise Constraints

N/A

4 Naming Conventions and Terminology

4.1 Glossary of All Terms, Including Acronyms, Used by Stakeholders Involved in the Project

- BE: Back-End
- FE: Front-End
- API: Application Programming Interface
- BUC: Business Use Case

5 Relevant Facts and Assumptions

5.1 Relevant Facts

There is no existing solution 7eam has found that fills a similar role as selecting a Movie, TV show or anime based on the preferences of a group of users.

5.2 Business Rules

N/A

5.3 Assumptions

There are a few assumptions 7eam deems necessary to see regarding Flick Picker's capabilities and what it will not do:

- Flick Picker will not provide a means to watch the selected show directly through the application.
- Flick Picker will suggest the best movie, TVshow, or anime for the group. This may violate the preferences of a member or multiple members of the group.
- The browser on which the application is used, through any device, is not deprecated

6 The Scope of the Work

6.1 The Current Situation

As Flick Picker is being built from the ground up, no existing business processes exist. However, 7eam has agreed to follow a rigorous development process, which is not changing. The developers must make a pull request against the 'develop' branch to make any changes, which two people must review before merging. If it is a code change, then tests must run on the deployed code, all passing and getting two reviews before merging.

6.2 The Context of the Work

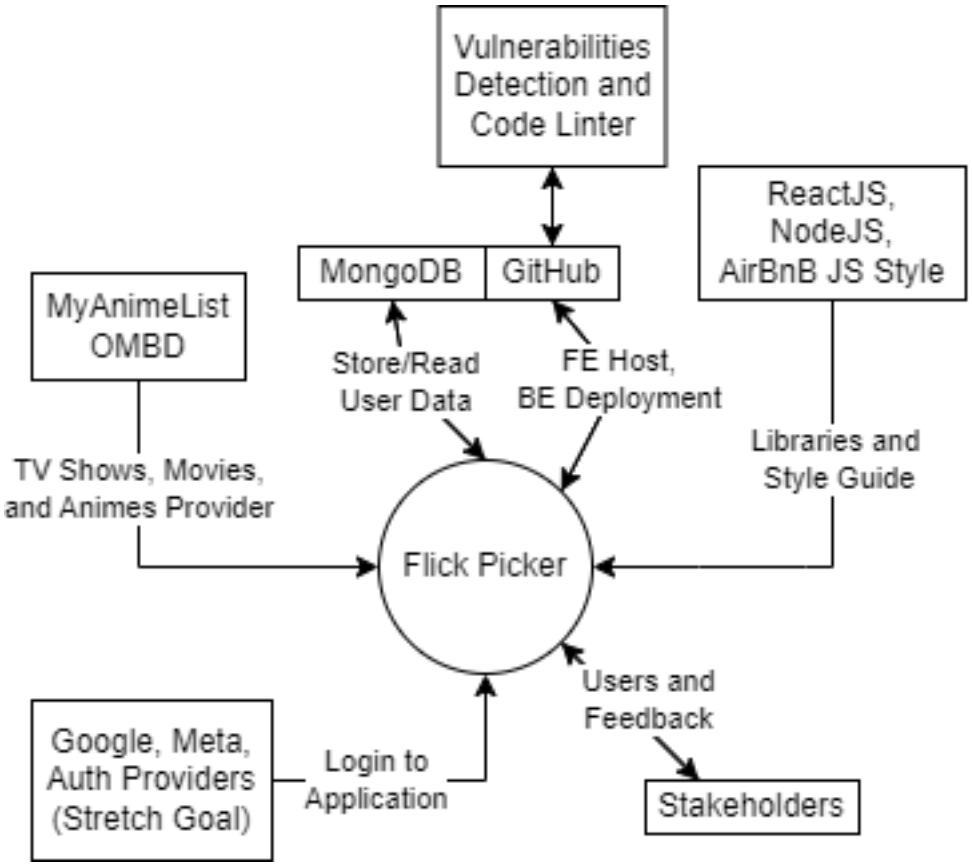

Figure 1: Context of the Work

6.3 Work Partitioning

Business Event List

Event Name	Input and Output	Summary of BUC
MyAnimeList/OMBD are polled	TV show, movie, and anime Data (in)	Collect information about the shows to display to users
MongoDB data is updated	User data updated (out)	Any updates to user data are stored
MongoDB data is queried	User data retrieved (in)	Fetches everything needed for the user after login
Code is updated	GitHub redeploys new changes (in/out)	Updates the FE or BE deployment any time there is a merge to server
Deprecated library is used	Vulnerability detection blocks changes (in)	Deployment protection such that libraries are safe to use
Code does not match style	AirBnB style guide (in), Linter updates code (in/out)	Enforces the style guide on FE/BE development
External login is queried	External authentication validates user (in)	Allows users to log in with industry standard authentication providers

6.4 Specifying a Business Use Case (BUC)

N/A as summary of BUCs are above.

7 Business Data Model and Data Dictionary

7.1 Business Data Model

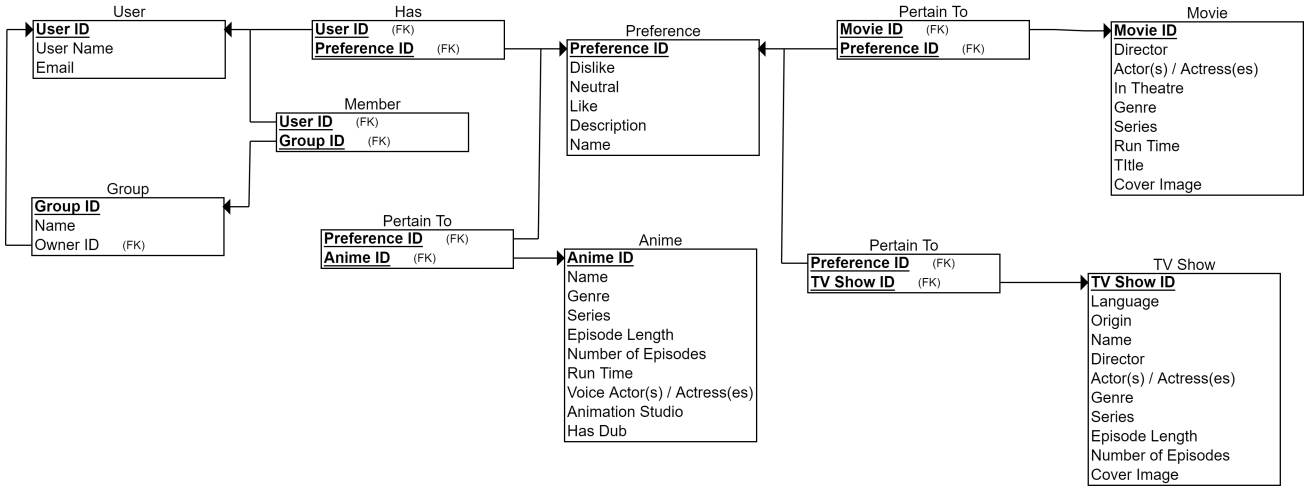

Figure 2: Business Data Model

7.2 Data Dictionary

For content of each element of the data dictionary, see the Business Data Model above. Additionally, attributes that are self explanatory are omitted from the Data Dictionary for document size purposes, they are assumed to be members of the data dictionary.

Name	Description	Type
User	A single user of the product	Class
Group	A collection of users with one being the owner	Class
Movie	North American or European movie	Class
TV show	North American or European television shows	Class
Anime	Japanese, Chinese, or Korean animated shows	Class
Group ID	Unique identifier for a group	
Movie ID	Unique identifier for a movie	Attribute
TV show ID	Unique identifier for a TV show	Attribute
Anime ID	Unique identifier for an anime	Attribute

8 The Scope of the Product

8.1 Product Boundary

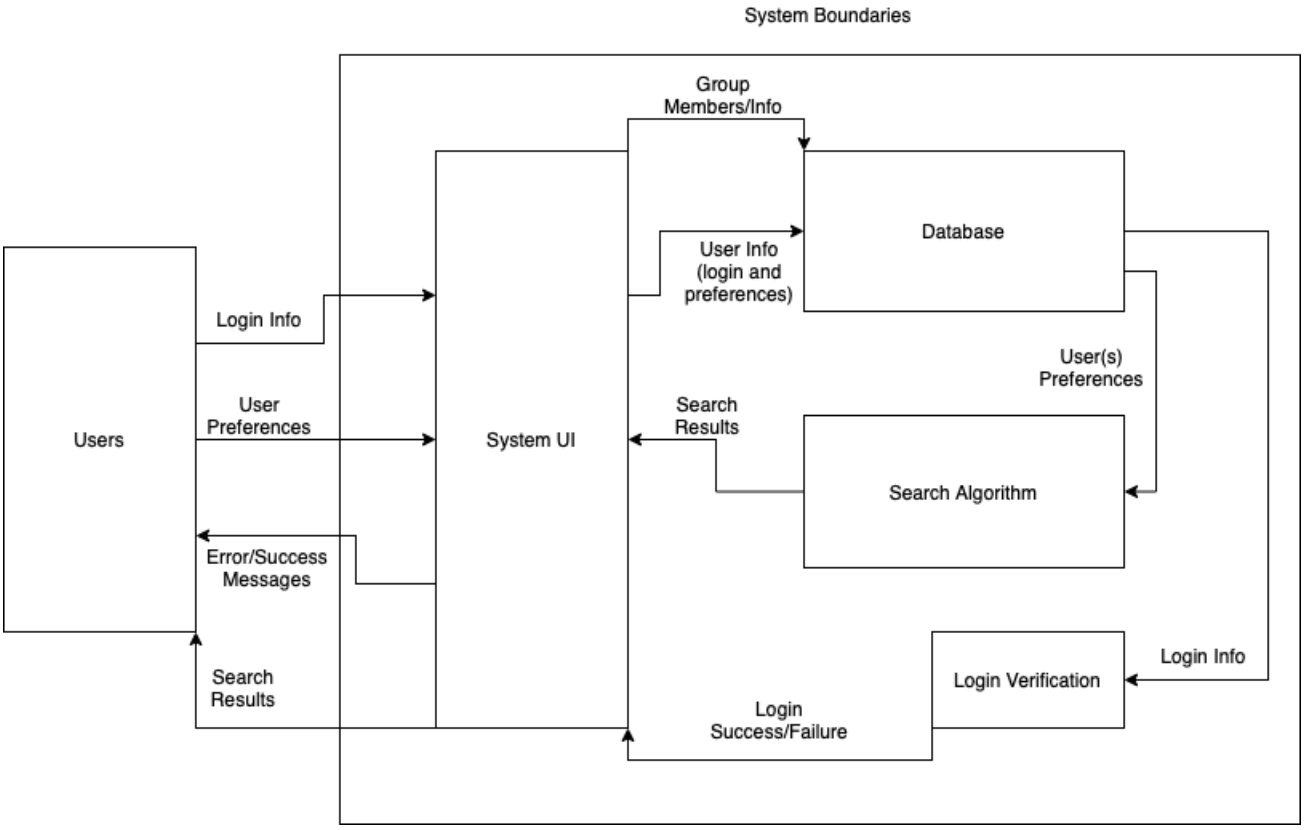

Figure 3: Product Boundaries

8.2 Product Use Case Table

Use Case	Actors	Description
1.0 Create Account	Primary: User	Prompts User for login and password, then stores both in the Database
1.1 Verify Login Details (Included in Use Case 1.0)	None	Searches Database for the username and password provided and verifies whether both correspond to the same account
1.2 Invalid Login (Extends Use Case 1.0)	None	If the login or password are invalid, shows an error message to the user indicating that the login info was incorrect
2.0 Create Preferences	Primary: User	User inputs their movie/show preferences, which the Database stores and connects to their account
3.0 Movie Selection	Primary: User	User search for movie/show recommendations based on the input preferences, and the system outputs a list of recommended media based on those preferences
3.1 Provide Feedback	Primary: User	Allows User to label a result with "like", "neutral", or "dislike" which will impact what kinds of media the system shows in its results for that User
4.0 Create Group	Primary: User/Group Owner Secondary: Other Users	Allows User to create a group and send invites to Other Users
5.0 Group Search	Primary: User/Group Owner Secondary: Other Group Users	The Group Owner (user who created the group) will initialte the search for a movie/show, to which the system will respond with a result that takes into account the individual users' preferences

8.3 Individual Product Use Cases

Please refer to Section 8.2 for individual Use Case descriptions.

9 Functional Requirements

9.1 Authentication Requirements

Requirement #: 1 **Requirement Type:** 1 **Event/Use Case: 1.0**

Description: The application shall allow the user to signup/login with their email and password
Rationale: Allows users to securely sign-in to our application
Fit Criterion: Unauthorized users should not be able to access and use the application

Requirement #: 2 **Requirement Type:** 1 **Event/Use Case: 1.0**

Description: The application shall allow the user to signup/login with an existing Google/Facebook account
Rationale: Allows users a convenient method to securely sign-in to the application
Fit Criterion: Unauthorized users should not be able to access and use the application

Requirement #: 3 **Requirement Type:** 1 **Event/Use Case: N/A**

Description: The application shall allow the user to logout of the application
Rationale: Allows users securely logout of the application
Fit Criterion: User must already be authorized and is logged-in

9.2 Profile/Group Requirements

Requirement #: 4 **Requirement Type:** 2 **Event/Use Case: 1.0**

Description: The application shall allow the user to modify their profile settings (username and password)
Rationale: Ensures that users can change their settings if needed
Fit Criterion: Username must be unique. Password is required to be longer than 6 characters.

Requirement #: 5 **Requirement Type:** 2 **Event/Use Case: 2.0**

Description: The application shall allow the user to set their profile preferences
Rationale: Required since the application needs these preferences to efficiently recommend movies/TV shows to the user. The preferences pertain to their favourite genre, movies, TV shows, anime, actors etc.
Fit Criterion: The user's input (eg. movie name) should be valid

Requirement #: 6 **Requirement Type:** 2 **Event/Use Case: 4.0**

Description: The application shall allow the user to create a group, and specify a group name
Rationale: Allows users to create groups so that they can find a recommendation match
Fit Criterion: The group name must be unique.

Requirement #: 7 **Requirement Type:** 2 **Event/Use Case: N/A**

Description: The application shall allow the user to join a group, either through request or invite
Rationale: Allows users to join groups so that they can find a recommendation match
Fit Criterion: If joining a group through invite, the invite must be valid and specific to the user.

Requirement #: 8 **Requirement Type:** 2 **Event/Use Case: 4.0**

Description: The application shall allow the user to invite other users based on their username or email
Rationale: Allows users to join groups so that they can find a recommendation match
Fit Criterion: The username/email invited must be registered as a valid user in our system

9.3 Recommendation Requirements

Requirement #: 9 **Requirement Type:** 3 **Event/Use Case: 3.0**

Description: The application shall provide an ongoing list of movie/TV show recommendations to the user
Rationale: Ensures that the user is consistently getting recommendations to find a possible match
Fit Criterion: The recommendation provided must be an existing movie/TV show

Requirement #: 10 **Requirement Type:** 3 **Event/Use Case: 3.1**

Description: The application shall allow the user to "like", "neutral" or "dislike" each movie/TV show recommendation
Rationale: Allows application to continually provide better recommendations and to find a recommendation match
Fit Criterion: The user must respond with one of the options: "like", "neutral" or "dislike".

Requirement #: 11 **Requirement Type:** 3 **Event/Use Case: 5.0**

Description: The application shall notify the group once a recommendation match is found
Rationale: The end goal is to provide the group with a recommendation match, so that they can watch that movie/show together.
Fit Criterion: The recommendation provided must be an existing movie/TV show and has been approved ("like", "neutral") by the majority of the group.

10 Non-Functional Requirements

10.1 Look and Feel Requirements

10.1.1 Appearance Requirements

The application will start up with a login screen with the options for the username and password of the account, with an "enter" button that does not become available until both the username and password are entered. The application will notify the user if an invalid login attempt occured, or if it was successful, bring them into the main page where there will be buttons for logging out, viewing/chaning profile settings, viewing/changing media preferences, and creating/joining a group.

10.1.2 Style Requirements

The UI will be styled with a consistent and predefined colour pallet, wherein buttons, banners, and other UI elements will have a limited set of different possible colours for the developers to select from. The buttons will be placed appropriately, with the size and spacing being optimal for user experience.

10.2 Usability and Humanity Requirements

10.2.1 Ease of Use Requirements

The application will be used by anyone who is able to operate a desktop or mobile device, and will have a simple graphic user interface that will be easy to navigate and use.

10.2.2 Personalization and Internationalization Requirements

N/A

10.2.3 Learning Requirements

Anyone with familiarity on how to use a phone or computer will be able to operate the application.

10.2.4 Understandability and Politeness Requirements

N/A

10.2.5 Accessibility Requirements

N/A

10.2.6 Convenience Requirements

N/A

10.3 Performance Requirements

10.3.1 Speed and Latency Requirements

This application shall immediately (within 1s), by human perception, respond to user input (within 1s after input) The system shall quickly login and logout the user (within 10s).

10.3.2 Safety-Critical Requirements

The application shall handle the user's private data with care.

10.3.3 Precision or Accuracy Requirements

The recommendation match provided to the group shall always be the recommendation that is the most approved by the group. The match should have almost zero "dislikes" within the group (cannot always be zero due to certain circumstances).

10.3.4 Reliability and Availability Requirements

The web application shall be available for at least 90% of the day, given that the servers running the application (on Firebase) are stable.

10.3.5 Robustness or Fault-Tolerance Requirements

The application shall always provide users with continual recommendations. The application shall always determine a recommendation match.

10.3.6 Capacity Requirements

Each user session within our application is independent of another. Therefore, there is no limit to the number of users who can use our app at once.

10.3.7 Scalability or Extensibility Requirements

While scalability is not a major concern, the application should still be scalable in the event that it is needed to do so.

10.3.8 Longevity Requirements

N/A

10.4 Operational and Environmental Requirements

10.4.1 Expected Physical Environment

N/A

10.4.2 Wider Environment Requirements

N/A

10.4.3 Requirements for Interfacing with Adjacent Systems

The product will work on all and any web browsers, whether on desktop or mobile, that support embedded JavaScript.

10.4.4 Productization Requirements

N/A

10.4.5 Release Requirements

N/A

10.4.6 Backwards Compatibility Requirements

N/A

10.5 Maintainability and Support Requirements

10.5.1 Maintenance Requirements

N/A

10.5.2 Supportability Requirements

N/A

10.5.3 Adaptability Requirements

This product is expected to run on common web browsers (Chrome, Firefox, Edge). It is also expected to run on mobile browsers.

10.6 Security Requirements

10.6.1 Access Requirements

The application will allow users to access only their own profile settings, while being able to view some of the information of other users.

10.6.2 Integrity Requirements

User data will be properly stored and have sufficient protection from corruption.

10.6.3 Privacy Requirements

The Developers and those with access to the database will not use or sell user info.

10.6.4 Audit Requirements

N/A

10.6.5 Immunity Requirements

N/A

10.7 Cultural Requirements

10.7.1 Cultural Requirements

N/A

10.8 Compliance Requirements

10.8.1 Legal Compliance Requirements

N/A

10.8.2 Standards Compliance Requirements

N/A

11 Project Issues

11.1 Open Issues

The algorithm to be used that finds appropriate media to recommend to user groups is yet to be determined. While we know the algorithm will be making use of each group members'

data as variables, we are deliberating on whether to create one ourselves, or research and use something that already exists.

If we choose to create our own algorithm, we need to have sound reasoning for its logic and structure, and how it makes use of the inputted data.

11.2 Off-the-Shelf Solutions

11.2.1 Ready-Made Products

N/A

11.2.2 Reusable Components

N/A

11.2.3 Products That Can Be Copied

There are publicly-known algorithms and methods we can research and implement in our project. We can also consult any open-source applications that make use of recommendation algorithms. It is likely that we will use one or more algorithms that are proven to be successful, and tailor them to our needs, using a combination of pieces as necessary. Creating an algorithm from scratch is a very complex task that is far beyond the scope of this project.

11.3 New Problems

12 Non-Functional Requirements

12.1 Look and Feel Requirements

12.1.1 Appearance Requirements

The application will start up with a login screen with the options for the username and password of the account, with an "enter" button that does not become available until both the username and password are entered. The application will notify the user if an invalid login attempt occurred, or if it was successful, bring them into the main page where there will be buttons for logging out, viewing/changing profile settings, viewing/changing media preferences, and creating/joining a group. It will be verified visually by a developer of the application.

12.1.2 Style Requirements

The UI will be styled with a consistent and predefined colour pallet, wherein buttons, banners, and other UI elements will have a limited set of different possible colours for the

developers to select from. The buttons will be placed appropriately, with the size and spacing being optimal for user experience. The UI will be verified visually by a developer of the application.

12.2 Usability and Humanity Requirements

12.2.1 Ease of Use Requirements

The application will be used by anyone who is able to operate a desktop or mobile device, and will have a simple graphic user interface that will be easy to navigate and use. This will be verified by developers testing the application.

12.2.2 Personalization and Internationalization Requirements

N/A

12.2.3 Learning Requirements

Anyone with familiarity on how to use a phone or computer will be able to operate the application. This will be verified by users testing the application, being both developers and volunteers, with a survey given to volunteers afterwards.

12.2.4 Understandability and Politeness Requirements

N/A

12.2.5 Accessibility Requirements

N/A

12.2.6 Convenience Requirements

N/A

12.3 Performance Requirements

12.3.1 Speed and Latency Requirements

This application shall immediately (within 1s), by human perception, respond to user input (within 1s after input) The system shall quickly login and logout the user (within 10s). This will be verified with automated Selenium testing.

12.3.2 Safety-Critical Requirements

The application shall handle the user's private data with care. This will be verified with automated Selenium testing.

12.3.3 Precision or Accuracy Requirements

The recommendation match provided to the group shall always be the recommendation that is the most approved by the group. The match should have almost zero "dislikes" within the group (cannot always be zero due to certain circumstances). This will be verified by users testing the application, being both developers and volunteers, with a survey given to volunteers afterwards.

12.3.4 Reliability and Availability Requirements

The web application shall be available for at least 90% of the day, given that the servers running the application (on Firebase) are stable. It will be verified with automated Selenium testing.

12.3.5 Robustness or Fault-Tolerance Requirements

The application shall always provide users with continual recommendations. The application shall always determine a recommendation match. It will be verified with automated Selenium testing.

12.3.6 Capacity Requirements

Each user session within our application is independent of another. Therefore, there is no limit to the number of users who can use our app at once. It will be verified with automated Selenium testing.

12.3.7 Scalability or Extensibility Requirements

While scalability is not a major concern, the application should still be scalable in the event that it is needed to do so.

12.3.8 Longevity Requirements

N/A

12.4 Operational and Environmental Requirements

The application will be usable on most common web browsers, such as Google Chrome and Mozilla Firefox. This will be verified with a developer conducting exploratory testing.

12.4.1 Expected Physical Environment

N/A

12.4.2 Wider Environment Requirements

N/A

12.4.3 Requirements for Interfacing with Adjacent Systems

The product will work on all and any web browsers, whether on desktop or mobile, that support embedded JavaScript. This will be verified with a developer conducting exploratory testing.

12.4.4 Productization Requirements

N/A

12.4.5 Release Requirements

N/A

12.4.6 Backwards Compatibility Requirements

N/A

12.5 Maintainability and Support Requirements

12.5.1 Maintenance Requirements

N/A

12.5.2 Supportability Requirements

N/A

12.5.3 Adaptability Requirements

This product is expected to run on common web browsers (Chrome, Firefox, Edge). It is also expected to run on mobile browsers. This will be verified with a developer conducting exploratory testing.

12.6 Security Requirements

12.6.1 Access Requirements

The application will allow users to access only their own profile settings, while being able to view some of the information of other users. This will be verified with a developer conducting exploratory testing.

12.6.2 Integrity Requirements

User data will be properly stored and have sufficient protection from corruption. This will be verified with a developer conducting exploratory testing.

12.6.3 Privacy Requirements

The Developers and those with access to the database will not use or sell user info.

12.6.4 Audit Requirements

N/A

12.6.5 Immunity Requirements

N/A

12.7 Cultural Requirements

12.7.1 Cultural Requirements

N/A

12.8 Compliance Requirements

12.8.1 Legal Compliance Requirements

N/A

12.8.2 Standards Compliance Requirements

N/A

13 Project Issues

13.1 Open Issues

The algorithm to be used that finds appropriate media to recommend to user groups is yet to be determined. While we know the algorithm will be making use of each group members' data as variables, we are deliberating on whether to create one ourselves, or research and use something that already exists.

If we choose to create our own algorithm, we need to have sound reasoning for its logic and structure, and how it makes use of the inputted data.

13.2 Off-the-Shelf Solutions

13.2.1 Ready-Made Products

N/A

13.2.2 Reusable Components

N/A

13.2.3 Products That Can Be Copied

There are publicly-known algorithms and methods we can research and implement in our project. We can also consult any open-source applications that make use of recommendation algorithms. It is likely that we will use one or more algorithms that are proven to be successful, and tailor them to our needs, using a combination of pieces as necessary. Creating an algorithm from scratch is a very complex task that is far beyond the scope of this project.

13.3 New Problems

N/A

13.4 Tasks

N/A

13.5 Risks

13.5.1 Ineffective algorithm

The main and biggest risk of the project is the algorithm recommending undesirable choices to user groups, which would make our application ineffective at its sole purpose. The probability of this risk actually becoming a problem is very low, as we plan on doing thorough research and analysis of different algorithms to create our customized algorithm using principles that are proven to be successful. Additionally, we will conduct testing using mock data, and make use of the appropriate metrics to determine the effectiveness. Then, we can consider these results to determine if changes should be made - so if our algorithm isn't effective at first, we will eventually reach a final product that is successful.

13.5.2 Insecure user login

With any service that requires users to create an account that has a password for their login, there is the risk of their stored data being compromised. The probability of this risk becoming a problem is low, as we plan on using proper encryption methods for passwords, and implementing Google and Facebook account connection with user authentication.

13.6 Costs

Minimal costs are associated with Flick Picker. Current costs are associated with the server and total between $0.01 and $1.00 per active month.

13.7 User Documentation and Training

13.7.1 User Documentation Requirements

N/A

13.7.2 Training Requirements

N/A

13.8 Waiting Room

N/A

13.9 Ideas for Solutions

N/A

14 References

Created From: Volere, Requirements Specification Template, Edition 18
AirBnB Style Guide

15 Traceability Matrix