Investigating the application of agile methodologies in the development of a social networking application implementing Spotify api and other technologies

by

Alicia Smith

A thesis submitted in partial fulfillment of the requirements for the degree of

BSc Software Design

Technological University of the Shannon

2022

Approved by

Chairperson of Supervisory Committee

Program Authorized   
to Offer Degree

Date

Technalogical university of the shannon

Abstract

**Investigating the application of agile methodologies in the development of a social networking application implementing Spotify API and other technologies.**

by Alicia Smith

Chairperson of the Supervisory Committee: Professor xxxx xxxxxx  
 Department of Engineering

A thesis presented on the use of agile methodologies and the development of a social networking application, implementing the Spotify API, Python, Android SDK, and Database technologies.

# Table of Contents

[Table of Contents i](#_Toc99611964)

[List of figures ii](#_Toc99611965)

[Acknowledgments iii](#_Toc99611966)

[Glossary iv](#_Toc99611967)

[Chapter 1 1](#_Toc99611968)

[**Introduction** 1](#_Toc99611969)

[Chapter 2 2](#_Toc99611970)

[**Background** **Research** 2](#_Toc99611971)

[Chapter 3 3](#_Toc99611972)

[**System Design** 3](#_Toc99611973)

[Chapter 4 13](#_Toc99611974)

[**Testing & Evaluation** 13](#_Toc99611975)

[Chapter 5 14](#_Toc99611976)

[Conclusions 14](#_Toc99611977)

[Bibliography 15](#_Toc99611978)

# List of figures

[Fig. 1. Top Songs CSV (user: Alicia Smith) 7](#_Toc99611941)

[Fig. 2. Data Visualisation of *foreverfifteen* Audio Features 7](#_Toc99611942)

[Fig. 3. Data Visualisation of *Rät* Audio Features 8](#_Toc99611943)

[Fig. 4. Top Artists CSV (user: Alicia Smith) 9](#_Toc99611944)

[Fig. 5. Temp Playlist with Seed Tracks (user's top 5) 9](#_Toc99611945)

[Fig. 6. Playlist Created Containing Recommendations Based on Seed Tracks Playlist 10](#_Toc99611946)

[Fig. 7. Architecture Diagram 11](#_Toc99611947)

[Fig. 8. System Flow Diagram 12](#_Toc99611948)

# Acknowledgments

The author wishes to express sincere appreciation to Professors Smith and Jones for their assistance in the preparation of this manuscript. In addition, special thanks to Dr. Elsa Leavitt whose familiarity with the needs and ideas of the class was helpful during the early programming phase of this undertaking. Thanks also to the members of the school council for their valuable input.

# Glossary

API – Application Programming Interface, can be described as “*a software intermediary that allows two applications to talk to each other*” (MuleSoft, n.d.).

**Short-term (short\_term)** – last 4 weeks

**Medium-term** – last 6 months

**Long-term** – all-time

## Chapter 1

### **Introduction**

* Provide a brief context and rationale for the project. References are very important.
* Present your research question(s).
* ~500 Words

As someone who listens to approximately 100,000 minutes of spotify content on a yearly basis, I do have a vested interested in the development and success of this project.

## Chapter 2

### **Background** **Research**

* Introduce the seminal work in the subject area of the project.
* Discuss current work in the domain of your project.
* Identify the problem/gap in the subject area that your project is going to
* address.
* References are extremely important in this section.
* ~1400 Words

## Chapter 3

### **System Design**

#### 3.1 Introduction

In this chapter I will discuss the requirements for *Spotifriend,* which were curated through my investigation into existing systems and applications with similar use cases, as well as my further exploration of the gap in the market for this particular type of application. This chapter will also provide an in-depth analysis of architectural and design decisions I made throughout the process, the explanations of which will be supported through visualisations of certain aspects of the project using UML diagrams.

#### 3.1.1 System Overview

*Spotifriend* is intended for use by existing Spotify users, in order to enhance their own listening experiences and also to develop music as a platform which allows users to socialise seamlessly with others; *Spotifriend* takes the hesitancy and indecision out of the search for new favourite tracks. As well as the objective of having recommendations based of friends’ top tracks, another goal was for the user to have the choice of entering their own seed tracks, or playlist, to the recommendation system so as to create a more specific set of recommendations – this being of particular benefit to those of us with more eclectic listening habits.

Due to the lack of insight into personal listening habits that Spotify provides, as well as the general vagueness of Spotify’s own recommendation algorithm, this application presents a unique opportunity for users to gain a deeper understanding of their own music tastes and to expand their musical repertoire without jumping outside the bounds of familiarity. In its current state this application is primarily self-driven, requiring only the user’s username and authentication agreement to retrieve top track and artist statistics, generate recommendations based on these, create playlists, and visualise audio features.

Due to the time constraints associated with this project, it would have been unviable to fulfil all requirements initially delineated during the planning phase; despite this, five main features have been accomplished to a satisfactory level. Regardless of the limited scope of the current application, *Spotifriend* as it exists now serves as a prototype for what could be further accomplished with increased time and resources. These main features are outlined below:

***Current User’s Top 5 Tracks*** *–* This feature involves the retrieval and display of a given users top 5 tracks over a chosen period of time.

***Current User’s Top 5 Artists*** *–* This feature involves the retrieval and display of a given users top 5 artists over a chosen period of time.

***Playlist Creation for Current User*** *–* This feature refers to the run-time creation of multiple Spotify playlists, one which is created for temporary use as seed tracks – which recommendations are based on - and subsequently deleted, the next containing the recommended tracks.

***Visualisation of Audio Feature Data*** *–* This feature encompasses the tasks of retrieving audio features from selected songs and graphing them in a way that is easily interpretable by the user.

***Visualisation of Artist Genres*** *– (check with MD about plotting clusters) This feature refers to the aspects of the project that retrieve details of artist genres and plot them accordingly into cluster graphs.*

#### 3.3 Requirements

During the emergence of this project, there were many systems requirements outlined and planned, many of which could be described as non-functional and were primarily concerned with user interaction and the GUI. As previously mentioned, many of these requirements had to be pared in order to prioritise the successful completion of a proof of concept.

Throughout my development of these requirements, I followed the iterative system analysis process (requirement analysis); this is made up of three symbiotic phases:

1. Understand the existing system or solutions
2. Identify improvements that can be made
3. Outline requirements and develop user stories for new system functionality

During this process it became evident that while there were some existing applications using the Spotify API which would fulfill some of my goals for this project, it was clear that the sort of integration I wanted to achieve was not already available. In addition to this, none of the existing applications seemed to have investments in the social potential of the Spotify API.

Below are a selection of user stories pertaining to the development of a prototype application comprising the main features outlined in section 3.1.1 above:

|  |  |  |
| --- | --- | --- |
|  | **As a** music listener  **I want to** see my top played tracks  **So I can** gain insight into my listening history | 1. **Verify** user can specify timeframe to query listening history 2. **Verify** user can view their top played tracks 3. **Verify** user can view details regarding their top tracks (artist, album, etc.) |
|  | **As a** music listener  **I want to** see my top played artists  **So I can** gain insight into my listening history | 1. **Verify** user can specify timeframe to query listening history 2. **Verify** user can view their top played artists 3. **Verify** user can view details regarding their top artists (genres, top album, etc.) |
|  | **As a** music listener  **I want to** get recommendations from friends account  **So I can** listen to similar music  **\*** | 1. **Verify** user can request song recommendations from friends account 2. **Verify** user can request artist recommendations from friends account 3. **Verify** track recommendation query returns 5 songs that are similar to particular user's top 5 tracks |
|  | **As a** user  **I want to** view breakdowns about my listening history  **So I can** get insights into my listening habits  **\*** | 1. **Verify** top 5 tracks data is visualised 2. **Verify** top 5 artist data is visualised 3. **Verify** data can be grouped by various criteria (genre, album, year) 4. **Verify** data visualisations have meaningful drilldowns |
|  | **As a** music listener  **I want to** add my recommended tracks to a playlist  **So I can** locate and listen to them easily | 1. **Verify** playlist ‘Spotifriend Recommendations’ is created in user’s Spotify account 2. **Verify** recommendations based on seed tracks is created 3. **Verify** playlist cover image is set to Spotifriend logo |

**US1**

While the *Spotifriend* application does not have an interactive user interface in its current state, it is possible to alter the query to the server in order to specify the time frame the user wishes to explore. There are three possible time spans that can be specified: short\_term (past 4 weeks), medium\_term (past 6 months), long\_term (all-time).

The user can view their top tracks in multiple formats at present, when the query data is requested from the server it is returned as a convoluted JSON file, for readability purposes this is then converted to a CSV file which has been pared down to only show relevant data to the user – omitting data such as market availability in various countries. Additionally, the user can view their top five tracks through the visualization of each tracks audio features, with the title and artist labeled on the graph.

Fig. 1 below shows the CSV file containing information on the given users top 5 tracks with the selected range of long-term. Though a song’s popularity and explicit status may not be of most importance to a user it may help to give a depth of further insight into a user’s listening habits. Fig. 2 anf Fig. 3 each show the data visualization graphs created from the audio features (*danceability, energy, acousticness, speechiness, liveness, valence*) of two of the user’s top tracks. These visualisations fulfill some acceptance criteria outlined in US1, and also that of US4.

Text

Description automatically generated

Fig. 1. Top Songs CSV (user: Alicia Smith)

Chart, line chart

Description automatically generated

Fig. 2. Data Visualisation of *foreverfifteen* Audio Features

Chart, line chart

Description automatically generated

Fig. 3. Data Visualisation of *Rät* Audio Features

**US2**

Similar to US1, users can alter the request query the timeframe in which they want to see their top artists, as well as viewing relevant imagine within a CSV file.

The user can also see a breakdown of which genres are frequent amongst their top artists through cluster graphs. While the accuracy of these graphs may seem skewed, it is simply due to the fact that Spotify has created innumerable oddly specific genres that many only relate to a small few artists. With further development I would aim to categorise these genres into more general terminology and create drilldowns within each genre. Examples of these unique genres can be seen below (Fig.2.).

Text

Description automatically generated

Fig. 4. Top Artists CSV (user: Alicia Smith)

**US3**

Desc

A screenshot of a video game

Description automatically generated

Fig. 5. Temp Playlist with Seed Tracks (user's top 5)

**US4**

Desc

**US5**

Desc

A screenshot of a computer

Description automatically generated with medium confidence

Fig. 6. Playlist Created Containing Recommendations Based on Seed Tracks Playlist

#### 3.3 Architecture

Diagram

Description automatically generated

Fig. 7. Architecture Diagram

#### 3.4 Design

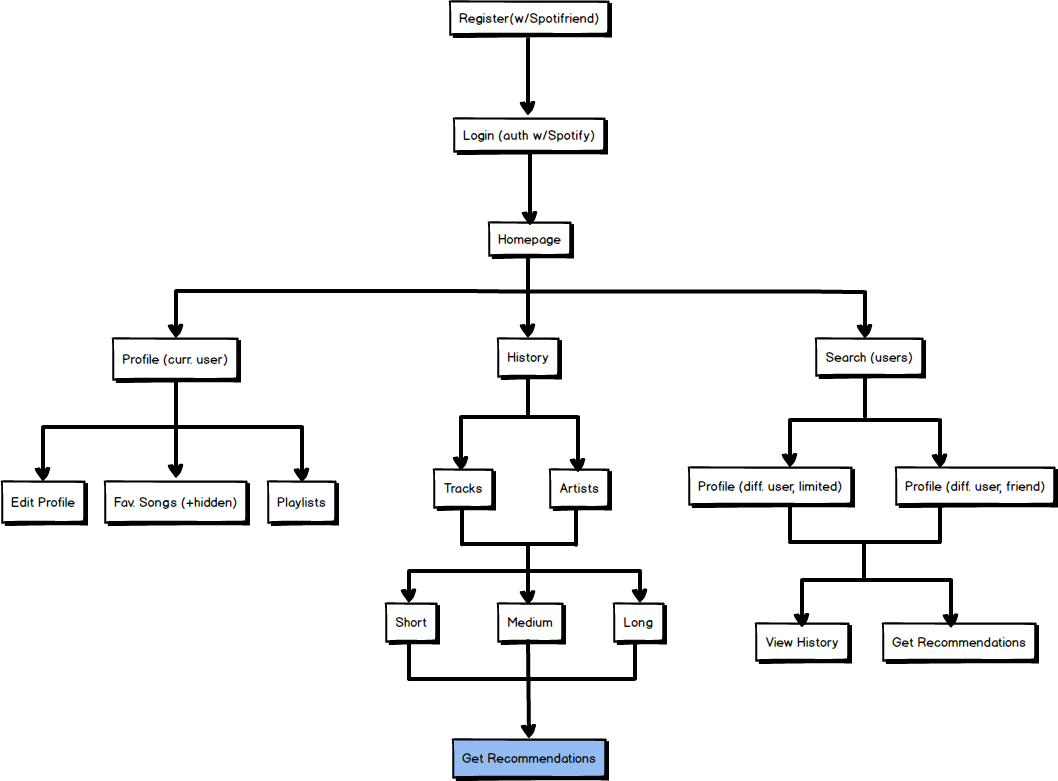


Fig. 8. System Flow Diagram

#### 3.5 Implementation

#### System flow

#### Auth flow

## Chapter 4

### **Testing & Evaluation**

(The total word count between Chapters 3 & 4 should be around 3500. How this the distributed between the chapters depends on your type of project. For example, if your project has quantitative results to discuss Chapter 4 will be longer than if your project is solely implementation based. You should discuss this with your supervisor.)

## Chapter 5

### Conclusions

* The chapter is key in demonstrating your understanding of what you’ve done.
* Provide a summary of the thesis.
* What are the implications of your project?
* Provide a set of recommendations.
* ~1500 Words

# Bibliography