Deliverable #1 - Software Requirement Specification

SE 3A04: Software Design II – T2 Group 10
Jemar Jones, Rendavid Dimen, Samraj Nalwa,
Samuel Scargall, Spencer Park, Stephan Arulthasan

February 7th, 2016

Contents

List of Tables

List of Figures

1 Introduction

1.1 Purpose

This app is for searching and discovering songs. It gives the user a chance to find a song based on very minimal and natural inputs (e.g. beat of the song). This app is designed for the majority of music listeners with an interest in identifying tunes.

1.2 Scope

The app is called *Merlin* and will identify songs based on artist, lyric and beat. It will use these 3 criteria to try to identify the potential song using a pre-existing database. It also will utilize Google Maps to identify and display the artists home town. It will not give the user a song file or a link to a song file. The app interface will accept finger tapping for the beat, voice for lyrics and as well as keyboard input for the artist name. These 3 inputs will be used to identify a single song to the user. The user will not see the individual searches of the inputs only the final guess of the *Merlin* system.

The wide range of inputs of the *Merlin* system allows users to approximate their songs in a quick and easy manner without even having to know the language. The main goals of the application include having a robust voice recognition system and a very low latency tap analyser in order to provide the best possible service. If the user knows atleast one of the inputs as certain, it is very likely that the list returned to that user will include the desired song.

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

Figure 1: `https://musicmachinery.files.wordpress.com/2014/02/with_streaming_and_sharing__teens_find_ways_around_paying_for_music_-_emarketer-2.png?w=620`

1.5 Overview

The organization of this document follows the template for an SRS for scientific computing software outlined by a combination of the Volere Requirements Specification Template and the IEEE Software Requirements Specification Template. The presentation follows the standard pattern of presenting goals, theories, definitions, and assumptions.

Term	Definition
BPM	Beats per Minute. A measure of tempo.
speech-to-text	An in-app functionality translating sound to the text represented by the words spoken in the sound.
friend	Another user of the application with whom one user of the application can share information with.
social media	the use of dedicated websites and applications to interact with other users, or to find people with similar interests to oneself.
database	a collection of information that is organized so that it can easily be accessed, managed, and updated.
beat	a main accent or rhythmic unit in music or poetry.
lyrics	the words of a song.
voice recognition	the ability of a machine or program to identify words and phrases in spoken language and convert them to a machine-readable format. (In this scope, converts to text that can be parsed by the application.)
latency	the delay between the receipt of a stimulus and the response to it.
tempo	the speed at which a passage of music is played.
API	API (application program interface) is a set of routines, protocols, and tools for building software applications. The API specifies how software components should interact.
ambient noise	noise, unintentionally created, existing in a space.

Table 1: Definitions, Acronyms, and Abbreviations

2 Overall Description

2.1 Product Perspective

The *Merlin* application is an Android application and as such is a component in the Android system. Merlin must therefore abide by all of the interfaces dictated for an Android application. *Merlin* must be written using a language and set of APIs that are accepted by the specification for an Android application. *Merlin* will also make use of both the microphone and touch screen peripherals of the device it is run on.

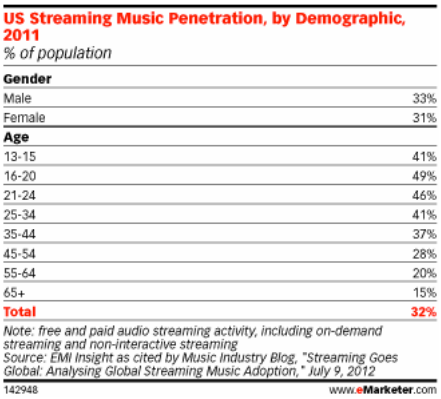
2.2 Product Functions

1. The product will feature an innovative input method. *Merlin* shall be able to receive user input in the form of speech or text for artist names and/or song lyrics as well as tapping of the screen for the desired tempo (*BPM*) of a song.
2. *Merlin* shall use user input to search a database for songs relating to the information given.
3. *Merlin* will display all relevant songs and artists to the user.
4. The selected song's artists biography shall be displayed when one is selected. Home town information will also be included and shown via the Google Maps API.

5. *Merlin* shall store songs and artists previously searched for by the user.

2.3 User Characteristics

The desired characteristics of a user of the *Merlin* application are as large as the domain in which the application resides. *Merlin* seeks to serve users who listen to music and actively try to expand their repertoire of musical choices. Users may be of any age, but *Merlin* shall primarily be designed for users from 13 to 40 years of age, as research (Figure 1) shows that these age groups listen to the most music via modern platforms such as streaming services. The same research shows that males and females both stream a similar amount of content, so *Merlin* will target both genders.



US Streaming Music Penetration, by Demographic, 2011

% of population

Gender	
Male	33%
Female	31%
Age	
13-15	41%
16-20	49%
21-24	46%
25-34	41%
35-44	37%
45-54	28%
55-64	20%
65+	15%
Total	**32%**

Note: free and paid audio streaming activity, including on-demand streaming and non-interactive streaming
Source: EMI Insight as cited by Music Industry Blog, "Streaming Goes Global: Analysing Global Streaming Music Adoption," July 9, 2012

142948 www.eMarketer.com

Figure 1: Streaming By Demographic

2.4 Constraints

Merlin is subject to all constraints imposed by the Android operating system, as described in section 2.1. The application is also subject to the physical constraints of the device it runs on, these include: device screen size, device battery life, device memory, device storage, and device processor speed. The application will also be constrained by the strength of the network connectivity used to communicate its data. Merlin will also be constrained by the environment in which the device it is run on is being used. Excessive or ambient noise near the device's microphone may impede *Merlin*'s voice recognition accuracy. Weather conditions that make it impractical for a user to use their touch screen will similarly affect the application.

2.5 Assumptions and Dependencies

Assumptions:

- The user has a working understanding of how to use a smartphone running the Android operating system.
- Users of *Merlin* should have reasonable knowledge of the song they are searching for.
- When using the BPM search option, the user should have a sense of rhythm that should be similar to the beat of the song to be searched.
- Songs and artists being searched are relatively well known to the general populous.

Dependencies:

- *Merlin* is run using the Android operating system. Should any changes in Android's operating system affect the application, the necessary updates to ensure the functionality of the application shall take place.

- The application uses the Google Maps API and should therefore be adjusted, if necessary, should any changes the the API occur

2.6 Apportioning of Requirements

The following functions may be delayed until future versions of the system:

- Being able to connect with new *friends*
- Sharing song and artist searches with your *friends* on social media.
- Playing the song that has been searched or the most popular song of an artist tat has been searched via an external streaming service.

3 Functional Requirements

BE1. The user wants to find a song.

VP1.1 Developer

i. *Merlin* must accept 3 search inputs: lyrics, artist and tempo.

VP1.2 User

i. The user will have the option to use the app's *speech-to-text* functionality to enter lyrics into the search field or type the lyrics with the devices keyboard.

ii. The user will have the option to use the app's *speech-to-text* functionality to enter the artist's name into the search field or type the artist's name with the devices keyboard.

iii. The user will have the option to tap their touch screen to the beat of the song they are searching for in the tempo search field.

iv. The system must return a collection of songs and artists that *Merlin* finds to match the input criteria.

BE2. The user wants to view their previous searches.

VP2.1 Developer

i. *Merlin* must track and store all searches.

VP2.2 User

i. The user must be able to view and remove searches from their saved history.

ii. The user must be able to re-execute a previous search.

BE3. The user wants more information about a song result.

VP3.1 Developer

i. *Merlin* must be able to retrieve and display an artist's home town on Google Maps.

ii. *Merlin* must be able to retrieve and display information about the entry's artist.

iii. *Merlin* must be able to retrieve and display song information including title and tempo.

VP3.2 User

i. Detailed information about song results can be requested by the user.

BE4. The administrator wants to change the experts.

VP4.1 Developer

i. The experts included in *Merlin* must be disjoint.

VP4.2 Administrator

i. *Merlin* must support the addition or removal of an expert as requested by an administrator.

BE5. An external entity wants to view a user's searches.

VP5.1 Security

i. *Merlin* must encrypt all traffic being sent to an external endpoint.

4 Non-Functional Requirements

4.1 Look and Feel Requirements

4.1.1 Appearance Requirements

LF1. The product shall have a simple and easy to understand layout with 2 input methods.

4.1.2 Style Requirements

LF2. The product shall be styled with easily readable font and no contrasting colors.

4.2 Usability and Humanity Requirements

4.2.1 Ease of Use Requirements

UH1. The product shall be easy for a high school student to use.

4.2.2 Personalization and Internationalization Requirements

UH2. The product shall keep track of the user's past searches for data analysis and improved searching.

4.2.3 Learning Requirements

UH3. The product shall be easy for a teenager with Android experience to learn.

4.2.4 Understandability and Politeness Requirements

UH4. The product shall not use an offensive vocabulary or derogatory words unless contained in the proper title of the searched song.

4.2.5 Accessibility Requirements

UH5. The product shall be usable by anyone that has access to an android phone running an API level of 16+ (Jelly Bean).

4.3 Performance Requirements

4.3.1 Speed and Latency Requirements

PR1. The product shall have a response time of maximum 2 seconds.

4.3.2 Safety-Critical Requirements

PR2. The product shall not heat up the phone battery.

4.3.3 Precision or Accuracy Requirements

PR3. The product shall have 90% similar song results.

4.3.4 Reliability and Availability Requirements

PR4. The product shall have the same availability as Google Maps API and database availability.

4.3.5 Robustness or Fault-Tolerance Requirements

PR5. The product shall remain available but lacking search functionality if the database connection is lost.

4.3.6 Capacity Requirements

PR6. The product must be able to cater for 1 simultaneous search.

PR7. *Merlin* should support 100 simultaneous users searching at a given time.

4.3.7 Scalability or Extensibility Requirements

N/A

4.3.8 Longevity Requirements

N/A

4.4 Operational and Environmental Requirements

4.4.1 Expected Physical Environment

OE1. The product shall be used by a person indoors under normal room temperatures and ambient noise levels.

4.4.2 Requirements for Interfacing with Adjacent Systems

N/A

4.4.3 Productization Requirements

N/A

4.4.4 Release Requirements

OE2. The product shall have a new update released with every set of bug fixes. These updates will be deployed via Google Play's update service.

4.5 Maintainability and Support Requirements

4.5.1 Maintenance Requirements

MS1. The product shall be keeps in proper running condition at all times.

4.5.2 Supportability Requirements

MS2. The product shall have a unique email address for issues/concerns by users.

4.5.3 Adaptability Requirements

MS3. The product shall be able to adapt to the user's voice.

4.6 Security Requirements

4.6.1 Access Requirements

SR1. The product shall be protected from all unauthorized attempts to read or manipulate data.

4.6.2 Integrity Requirements

SR2. The product shall maintain integrity even during a data hack.

4.6.3 Privacy Requirements

SR3. The product shall not disclose previous user history to anyone other than the user themself.

4.6.4 Audit Requirements

N/A

4.6.5 Immunity Requirements

N/A

4.7 Cultural and Political Requirements

4.7.1 Cultural Requirements

CP1. The product shall not use any words, images or videos that will offend anyone in any part of the world excluding title's of song search responses.

4.7.2 Political Requirements

CP2. The product shall not be linked to any political standing or party.

4.8 Legal Requirements

4.8.1 Compliance Requirements

LR1. The product shall obey and comply with the laws of the Data Protection Act.

4.8.2 Standards Requirements

LR2. The product shall comply with all data and privacy standards and specifications.

A Division of Labour

Jemar Jones	Completion of sub sections 2.1, 2.3 and 2.4.
Rendavid Dimen	Completion of sub sections 2.2, 2.5 and 2.6. Supporting work on 3.
Samraj Nalwa	Completion of section 4.
Samuel Scargall	Completion of sub sections 1.1 and 1.2.
Spencer Park	Worked on section 3, reviewed, edited and formatted the document.
Stephan Arulthasan	Completion of sub sections 1.3, 1.4 and 1.5. Supporting work on 3.

Agreement with the Division of Labour outline above:

Jemar Jones	
Rendavid Dimen	
Samraj Nalwa	
Samuel Scargall	
Spencer Park	
Stephan Arulthasan	