Media Wiz

Software Requirements

Ryan Gross
Krasimir Hristov
David Williams

February 23, 2012
## Index

### 1.0 Introduction
- 1.1 Goals and Objectives ................................................................. 4
- 1.2 Statement of Scope ................................................................. 4
- 1.3 Software Context ........................................................................ 4
- 1.4 Major Constraints ...................................................................... 4

### 2.0 Usage Scenario
- 2.1 User Profiles ............................................................................. 5
- 2.2 Use-Cases .................................................................................. 5
- 2.2.2 Retrieve Music File Information ............................................. 6
- 2.2.3 Play Music Files ...................................................................... 6
- 2.2.4 Profile View ........................................................................... 6
- 2.3 Special Usage Considerations .................................................. 6

### 3.0 Data Model and Description
- 3.1 Data Description ......................................................................... 7
- 3.1.1 Data Objects ............................................................................ 7
- 3.1.2 Relationships .......................................................................... 8
- 3.1.3 Complete Data Model ............................................................. 10
- 3.1.4 Data Dictionary ....................................................................... 11

### 4.0 Functional Model and Description
- 4.1 Description of Functions ............................................................ 12
  - 4.1.1 Function narratives .............................................................. 12
  - 4.1.2 Functions flow diagram ...................................................... 12
  - 4.1.3 Function interface descriptions ........................................... 13
  - 4.1.4 Performance Issues ............................................................ 13
- 4.2 Software Interface Description .................................................. 13
  - 4.2.2 External system interfaces ................................................... 13
  - 4.2.3 Human interface ................................................................. 14
- 4.3 Control Flow Description .......................................................... 14

### 5.0 Behavioral Model and Description
- 5.1 Description for Software Behavior ............................................ 14
  - 5.1.1 Events .................................................................................. 14
  - 5.1.2 States ................................................................................... 14

### 6.0 Restrictions, Limitations, and Constraints

### 7.0 Validation Criteria
- 7.1 Classes of Tests ........................................................................ 15
  - 7.1.1 Database querying performance ......................................... 15
  - 7.1.2 Echoprint music identification accuracy .............................. 15
  - 7.1.3 Application response time ................................................... 16
  - 7.1.4 Portability ............................................................................ 16
  - 7.1.5 Ease of use ........................................................................... 16
- 7.2 Expected Software Response .................................................... 16
  - 7.2.1 Database querying performance expectations .................... 16
  - 7.2.2 Echoprint music identification accuracy expectations .......... 16
  - 7.2.3 Application response time expectations ............................. 16
7.2.4 Portability expectations........................................................................................................16
7.2.5 Ease of use expectations ......................................................................................................16

7.3 PERFORMANCE BOUNDS ....................................................................................................16

8.0 APPENDICES .......................................................................................................................17
  8.1 PRODUCT STRATEGIES .....................................................................................................17
  8.2 CODE LICENSE ..................................................................................................................17
1.0 Introduction

1.1 Goals and objectives
The overall goals and software objectives are:

- Identify and fix missing song information/Artwork
- Sort music collection
- Play music collection
- Display top listened to songs
- Links to music store

1.2 Statement of scope
The Media Wiz application will aid in the users music listening experience. Users will be able to use the track record fixer to find and replace missing or incorrect track records for one music file or the entire collection. The user will also be able to manage the music collection by moving all the music on located on the android device to a default folder or a folder specified by the user. Users will also be able to listen to the music collection through the application. The Media Wiz application will also display the current top listened to songs and provide links to the songs page in the following music stores: Google, Android, and Amazon.

1.3 Software context
Media Wiz is an Android application that will be able to modify track records by identifying the songs through a unique audio identifier.

The application will also manage the music collection by moving the songs on the device to a default or user specified folder. The application will create sub folders, with in the specified folder for the collection, for each artist found in the music collection. Albums matching that artist will have sub folders created with in the specific artist folder. Songs will be placed in the correct album folder.

The application will be able to function as simple media player for the user to listen to their music collection. To help the user continue to grow their music collection the application will a list of top listened to songs and links to the different music stores available in the android market.

1.4 Major constraints
Any business or product line constraints that will impact the manner in which the software is to be specified, designed, implemented or tested are noted here.
2.0 Usage scenario

2.1 User profiles

Users of this system are individuals with access to the android market. This application will work for any mobile device including tablets, and devices with emulators.

2.2 Use-cases

2.2.1 Manage music files/folders
Here the user can create, edit, and delete music folders within the music directory of the device. The user can also copy, cut and paste music files into music folders.

2.2.2 Retrieve music file information
In this use-case the user selects a file, or a folder of files that they want to send to the database. After the database has matched the audio files, the application automatically updates the files in the user's device.

2.2.3 Play music files
When selecting the media player tab the user can select a song to play and pause, as well as seek through other music files in that folder.

2.2.4 Profile view
Here the user can view his or her top ten most played music files. In addition the user can select links to various online music stores.

2.3 Special usage considerations
This is an Android based application and will not be available to Apple devices. This application will be formatted for compatibility with any sized device, including emulators. While the application does require Internet access to acquire song information, the other uses of this system can be utilized without an active Internet connection.
3.0 Data Model and Description
This section describes information domain for the software

3.1 Data Description
The main data object that the software is going to manage and play are digital audio files, especially audio files in the MP3 format. These file are going to have different attributes as ID3 metadata.

The playlist, that contains the, at the moment played, songs is a data object.

The user is an important data object, since the user is going to play the music, manage it and rate it. He is going to be able to interact with a user friendly interface. For every user (phone) an individual music profile is going to be created on the echonest server.

There are several external data objects as echonest.com API & allcdcovers.com API. This service will provide data that will be used for the ID3 metadata.

However, echonest.com API provides a profile creation service, with that service it is possible to generate a profile of each user, that will be stored on their server, so recommendations can be provided for each user.

3.1.1 Data objects
MP3 format – ID3 metadata contain the following attributes:

- Header
- Title
- Artist
- Album
- Year
- Comment
- Zero byte
- Track
- Genre

Furthermore as in ID3v2 supported there will be an image attribute.

- Cover

The current playlist has only one attribute

Music songs

The user’s attributes are going to be
• music rating – favorites songs
• music profile – echonest based taste profile
• saved playlists – custom playlist of MP3 songs
• Music library – contains all MP3 songs

Echonest.com API has a lot of attributes, for now no specific attributes will be listened besides this ones

• API developer key – provides access to the API
• Taste profiles
• Identifying a MP3 by metadata
• Identifying a MP3 by upload

allcdcovers.com API has following attributes:
Image – front/back/CD/inside cover of a song or album

3.1.2 Relationships
3.1.3 Complete data model

- Music Library
- Copy/delete/move
- Edit tags
- User
- User profile
- Music playlist
- Music rating
- Echonest Profile
- Get song information
- MP3 Music
- Song 1
- Song 2
- Song n
- Echonest data
- alldcover
- • Header
  • Title
  • Artist
  • Album
  • Year
  • Comment
  • Zero byte
  • Track
  • Genre
  • Cover
- Play
- Play list
- Music profile
- Rate
3.1.4 Data dictionary
The software itself will maintain a SQLite database in the background, where the user's rating is going to be saved. To maintain a music profile online on echonest.com, certain data can be saved in the database when the user is offline and uploaded when he is again online. Therefore it will be used as a buffer for user data.
4.0 Functional Model and Description

4.1 Description of Functions

4.1.1 Function narratives

4.1.1.1 Music information retrieval narrative
The user selects a file or folder of files to send to the database to be matched. When the song is recognized and matched in the database the information regarding that file or files is sent back to the user. The application automatically updates the files in the hard drive of the device.

4.1.1.2 Music player narrative
The user selects a music file to play and the player plays that file. The user can then pause and play the song as well as move back and forth through the folder to other songs.

4.1.1.3 File/Folder management narrative
Users can move music files to different folders by cutting or copying them. In addition to moving files users can also create, edit, or move folders within the music directory for desired file organization.

4.1.1.4 Profile view narrative
In the profile view users can see the top ten most played files, as well as click on links to online music stores provided by Google, Android, and Amazon.

4.1.2 Functions flow diagram
4.1.3 Function interface descriptions

4.1.3.1 Music information retrieval interface description
A sample of the selected music file is taken and sent via the internet to the Echoprint music recognition software. Here the music file is matched based on the sample to corresponding information in the database. The Echoprint software then sends the file information back to the application.

4.1.3.2 Music store links interface description
When a link to a music store is selected by the user the application will automatically open the default browser of the device and enter the url of the desired music store into the browser's address bar.

4.1.4 Performance Issues
The Echoprint software requires that a sample be at least twenty seconds in length, but for better accuracy it has been decided that a thirty second sample will be used.

4.2 Software Interface Description

4.2.2 External system interfaces
The Echoprint software that the system interfaces with consists of three major components: the code generator, the server, and the database. The code generator uses advanced signal processing techniques to translate the audio file into code that can be compared and matched. The server then indexes the code for look-up, and finds the corresponding song in the database. A more detailed image of this process is shown below.
4.2.3 Human interface
The user interface is going to be a simple graphical user interface that will represent file item with traditional icons. The three main views, file management, music player, and profile view, will be selected by a tab system at the top of the screen.

4.3 Control flow description
The control flow for the system is presented with reference to Section 5.0 of this document.

5.0 Behavioral Model and Description
A description of the behavior of the software is presented.

5.1 Description for software behavior
A detailed description of major events and states is presented in this section.

5.1.1 Events
- **Copy/move/delete music file**
The user marks one or more music files and completes one of the operations by pressing one of this buttons.
- **Edit ID3 Tags**
The user presses the edit button and edits ID3 Tags of a MP3 file by himself.
- **Auto refill ID3 Tags**
The user presses the auto refill button and based on the title of the MP3 file, the rest of the ID3 tags is looked up online on echonest.com
- **Play/Pause/Stop Song**
User performs operation by pressing one of this buttons.
- **Previous/Forward Song**
User performs operation by pressing one of this buttons.
- **Shuffle Option Play Song**
User performs operation by pressing the shuffle button.
- **Rate Song**
The user can rate a song by giving the song a certain amount of stars.
- **View Player View**
User can switch by pressing the player button to the player view.
- **View My Library**
User can switch by pressing the my library button to the my library view.

5.1.2 States
- **Saving ID3 Tags**
After the editing either by the user or echonest, the ID3 Tags are been saved to the chosen MP3 file.
• **Playing Music**
  The current playlist is played.

• **Getting Information from echonest.**
  A load bar is shown while ID3 Tags are gathered from echonest.

• **saving taste profile**
  After a specific amount of played songs, the echonest user profile is been updated. If app is closed before update could be completed or user is offline, data will be saved in the database and performed next time the app is running and the user is online.

• **executing the app the first time**
  Disclaimer and short descriptions is shown. User needs to choose directories with his music. If user does not choose anything a quick scan for MP3 will be performed. After scan is finished, echonest profile will be created in the background.

### 6.0 Restrictions, Limitations, and Constraints

Android API needs to be first explored and see which limitations it has.

Echonest API needs to be test properly; servers could be too slow or not reachable.

The project will contain 3 big milestones.

1. Create interface and implement my library. Music manager functionality as ID3 editing will be provided.
2. Implement a Music Player.
3. Customize the app by adding rating system, taste profile creation and recommendations.

### 7.0 Validation Criteria

#### 7.1 Classes of tests

7.1.1 **Database querying performance**
  This refers to the speed at which the application can retrieve information from the Echoprint database and return that information back to the application.

7.1.2 **Echoprint music identification accuracy**
  The Echoprint music identification software should return correct song identification results when accessed.
7.1.3 Application response time
The application response time is the time that the user has to wait during a transition in the application. A transition can occur when a new page is selected, a music file has been selected to be played, or a link to another website has been selected.

7.1.4 Portability
The application should look and interact as expected on all devices regardless of aspect ratio and size.

7.1.5 Ease of use
The user interface should be designed to accommodate simplicity to all ages of users.

7.2 Expected software response

7.2.1 Database querying performance expectations
The querying process should take under twelve seconds per song, given that the user has excellent signal strength on a 3G network or better.

7.2.2 Echoprint music identification accuracy expectations
The Echoprint music identification software should return the correct information one hundred percent of the time.

7.2.3 Application response time expectations
The application response time should be less than half a second for any action the user selects and navigation through the application should appear seamless. The only exception to this is playing a song which may take one or two seconds, and the retrieval of data from the Echoprint database as mentioned in section 7.2.1.

7.2.4 Portability expectations
All devices running the application should experience the same view of this product via dynamic formatting layouts dependent on the device screen’s dimensions. This will be tested on different emulator screen sizes as well as several devices that the project has access to.

7.2.5 Ease of use expectations
To test how easy it is to use this application several public beta tests will be conducted with surveys providing user feedback.

7.3 Performance bounds
In order to meet the desired performance requirements for retrieving information from the database, a thirty second sample will be used to for song identification.
This will decrease the upload time of sending music files to the Econprint music identification software. The upload and download speeds will be dependent on the signal strength of the device, as well as the user's service provider. Updating music files on the user's device will be limited to the hardware inside that individual device.

**8.0 Appendices**

Presents information that supplements the Requirements Specification

**8.1 Product Strategies**

If the app is implemented & tested successfully, it will be released to the android market as a free app.

**8.2 Code license**

GNU GENERAL PUBLIC LICENSE v3 http://www.gnu.org/licenses/gpl.html