WeListen: Music System

By

Chris Korczyk, Jimmy McDonough, Robert Osley

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

University of Cincinnati
School of Information Technology
College of Education, Criminal Justice, and Human Services

© Copyright 2014 Christopher Korczyk, James McDonough, Robert Osley

The author grants to the School of Information Technology permission
to reproduce and distribute copies of this document in whole or in part.

Christopher S. Korczyk 4/22/2014
Christopher Korczyk Date

Jimmy McDonough 4/22/2014
James McDonough Date

Robert Osley 4/22/2014
Robert Osley Date

Russell E McMahon 4/17/2014
Russell McMahon Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services
April 2014
Dedications

Christopher Korczyk
I would like to give special thanks to my wife, Jessica Noll-Korczyk for always supporting me and giving me the encouragement to embark on this adventure at my age. I would also like to thank my children, Stephanie Korczyk and Trisha Korczyk for being one of my biggest motivations when I was feeling like I couldn’t continue. I would also like to thank my professors, Russell McMahon, Patrick Kumpf and James Scott for all of the guidance they have given our team over the last two semesters. I would like to thank my friends, Jeremy Cook and Doug Post for helping to develop the idea for this project, and last but not least, I would like to thank my partners in this project, James McDonough and Robert Osley for all of the hard work that you put into making this project a huge success.

James McDonough
Senior design has been quite the experience and I definitely could not have done it by myself. First, I want to thank my teammates Chris Korczyk and Robert Osley for all the time and effort they put into this project, it was a pleasure working with you both. Next, I would like to thank my friend and coworker Bjorg Prodan for all the help and knowledge that he provided me during this project. I would also like to thank all the professors in this course, Russell McMahon, Patrick Kumpf and James Scott for helping keep us motivated and focused throughout our time working on the project. Lastly, I want to thank all my friends and family for all the positive words and encouragement they have given me.

Robert Osley
The last two semesters have been an awesome learning experience. I would like to thank my partners in this project, Chris Korczyk and Jimmy McDonough for all of the hard work and dedication to this project. I would like to thank my family and friends for all of the support that they have given me over the years I have spent in school. I would also like to thank the professors Russell McMahon, James Scott and Patrick Kumpf for the time and dedication they have put into guiding us and all of the students in the 2014 Senior Design Class.
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information Collection and Use</td>
<td>29</td>
</tr>
<tr>
<td>Registration</td>
<td>29</td>
</tr>
<tr>
<td>Order</td>
<td>29</td>
</tr>
<tr>
<td>Cookies</td>
<td>30</td>
</tr>
<tr>
<td>Log Files</td>
<td>30</td>
</tr>
<tr>
<td>Links</td>
<td>31</td>
</tr>
<tr>
<td>Surveys and Contests</td>
<td>31</td>
</tr>
<tr>
<td>Special Offers</td>
<td>32</td>
</tr>
<tr>
<td>WeListen End User License Agreement (EULA)</td>
<td>33</td>
</tr>
<tr>
<td>1. PERMITTED USE AND RESTRICTIONS</td>
<td>34</td>
</tr>
<tr>
<td>2. OWNERSHIP</td>
<td>34</td>
</tr>
<tr>
<td>3. COPYRIGHT INFRINGEMENT</td>
<td>35</td>
</tr>
<tr>
<td>4. WARRANTY</td>
<td>35</td>
</tr>
<tr>
<td>5. CUSTOMER REMEDY</td>
<td>35</td>
</tr>
<tr>
<td>6. DISCLAIMER OF DAMAGES</td>
<td>36</td>
</tr>
<tr>
<td>7. GENERAL PROVISIONS</td>
<td>36</td>
</tr>
</tbody>
</table>
Abstract

WeListen is a free-to-use DJ system designed to revolutionize the way DJs manage their music, handle song requests and build their catalog.

The WeListen music system allows the DJ and businesses to maintain a continually updated catalog at no additional charge.

The largest recurring cost to a DJ business is maintaining a fresh catalog of music. The typical jukebox system charges customers over a dollar per song every time a customer plays it.

Through the use of Amazon’s music store, a customer is, on average, charged 99 cents to play a song if it doesn’t exist in the DJ’s catalog.

The one-time purchase adds the song to the DJ’s catalog allowing the requested song to be played by any customer for free after that initial purchase, while removing the distraction of customers inquiring about songs already available.

Problem Need

Small businesses that rely on music to draw customers such as bars, coffee houses and DJ’s have several options for acquiring the most up to date music and methods of playing that music for their customers. Unfortunately, the services available can be costly for memberships that grant access to the latest and greatest music. The alternative, purchasing the music the customers want as the songs are released, can wind up being a constant expense to keep the music fresh and new. There are not any methods that provide little to no cost to the business to maintain access and build a catalog of music for little to no cost.
After interviewing a professional DJ who has managed small bars that promote music and dancing, we learned more about the costs associated with the music that is played. The DJ said, “Once the necessary equipment is purchased by a business, maintaining a diverse catalog of songs becomes the greatest cost associated with developing and maintaining a successful business” (Cook 2013). People do not want to go to clubs where the music is outdated even by a few months. The customers expect to hear the music they like, hear on the radio and see in the music stores and online.

As far back as anyone in the business can remember the cost of maintaining their music has been an issue for businesses and DJs. The alternative to spending large portions of their profits on maintaining their music library has recently become signing up for a membership with a service. The services that have become the modern day jukeboxes have expensive memberships for the bars to install their equipment in the establishment. On top of the cost incurred by the business, the services charge the customers a fee to select the music to be played. Ultimately both the business and the patrons wind up spending money to play and listen to the music.

If a business chooses to maintain a catalog of music without a service, the issue of organization becomes evident quickly. Every song that is purchased needs to be cataloged and recorded in a way that allows the patrons and the DJ or business owner to locate the music quickly. Many DJ’s still rely on binders with the music printed out and placed around the bar for patrons to look up the songs that they want to hear. The books many times have the songs organized by many different categories such as title, artist and genre. Therefore, every song needs to be printed multiple times for the selection books. Every time the business or DJ adds music to the catalog those catalogs need to be printed again with the new material.
Finally, while most DJ’s are willing to talk to customers, when they are working a show, every time a customer has a request they have to interrupt the DJ and ask for a song to be played. If the DJ is working with a large number of customer requests and does not take the time to write down the request there is a good chance that the requested song will be forgotten which many times results in confrontations with customers and makes many customers less likely to return to the establishment.

The WeListen System is our solution to the problems listed above. The system has been developed to be the best and least expensive option for businesses and DJ’s to build and maintain a current and up to date catalog of music using customer participation to purchase songs for the catalog for less than the cost of purchasing one play of a song on the modern jukebox service systems.

**Solution**

The WeListen Music system addresses the problems that most businesses that rely on music face during day to day operations. The WeListen Music System has two basic elements to address these problems. Each is designed with a specific purpose and therefore specifically targets two types of users: the business/DJ/owner and the user/patron. In addition the WeListen Music System uses several different tools and software libraries to enable to software to function properly.

**Branding**

Several logo prototypes were developed and refined by the WeListen team. Below are the two logo styles that the team decided would best fit within the theme of the WeListen Music System.
Through the evolution of the project the branding has changed. The original idea of the project was to include a DJ system and a Jukebox system. Each of these systems had their own logos. See figures 1 and 2.

![Figure 1. WeListen - DJital DJ logo](image)

![Figure 2 WeListen – Jukebox Logo](image)

As the project evolved and the DJ and Jukebox versions of the software were combined into one application, these logos were scrapped and the WeListen team chose to just use the WeListen Logos. By eliminating the extra logos we were able to streamline the WeListen brand.

Two logos were developed for the WeListen Music System. The standard banner style Logo (Figure 3) featuring musical notes and a microphone and the complete title spelled out, and the WL small logo (Figure 4) featuring simply the letters “W” and “L” to represent the WeListen brand. The WL small logo is to be used for the favicon and icons for the software shortcuts. This logo is also to be used as a profile image for any and all social media associated with the WeListen Music System.
The WeListen Team believes this branding brings a look and feel to the WeListen Music System that we feel will become recognizable as the use of this system grows throughout the DJ industry. The WeListen Team will apply for trademarks on the logos in the months to come as we move toward the initial release of the WeListen Music System software.

**Development**

The WeListen Team had a great number of obstacles throughout production, the biggest of which being the Amazon Product Advertising API. During the early stages of development, the TagLib library was utilized to pull and parse meta-data from mp3 files including Album, Artist, Title, etc. Those variables were then passed along to the LastFM API which would then match the Title and Artist to the correct Song Object in LastFM’s database, thus returning a complete
Song Object with every matched variable including time, release date, etc. The biggest concern was how the WeListen Team could incorporate the new Song Objects into the Amazon API. After a great deal of trial and error, it was discovered that the LastFM API was extremely limited. The API would only accept the title and artist variables to perform a song lookup, and often times, duplicate entries existed in the LastFM Database, each with different spelling and punctuation.

Ultimately, the LastFM API was discarded and the Amazon API took over as the Song Object manager. Now, not only were the artist and title search limitations lifted, but the Amazon API acted just as the search-box on Amazon.com does. Taking any of the meta-data variables from a TagLib result and passing them through the Amazon API would return the most relevant and closely related Song Object to the search variable.

Primarily utilizing the Amazon API made searching for music more accurate, allowed for more returned Song Object variables, and streamlined the Amazon Purchasing for WeListen by returning a direct Amazon ASIN (ID Number) for each returned song making future object references much easier. Album artwork became reference-able from the API through direct URL returns, and the Song Object purchase price became directly accessible.

Updated documentation is nearly non-existent since the last Amazon Product Advertising API re-build. The WeListen Team researched the mechanics of the API and came up with a working solution that allowed calls to the API, and workarounds which would return mp3 download objects most closely related to the search parameters passed from the WeListen Client. The WeListen team did a fantastic job constructing the Amazon Product Marketing API and has
hopes in sharing this information to the many others who may be struggling in making calls to this API.

Software

The first element is the web application for patron access and song request. This web application allows the patron to log on and search for songs. The songs are listed as available or unavailable for the business to play.

For unavailable songs, the patron can use their account to purchase the song for play. The song will then be available for free play as part of the businesses catalog of music.

Songs that are available can be selected and sent to the music queue for play. The music is played automatically by the software in the order in which it is received unless the business administrator overrides the order.

This section of the software was be written as a web-based application using ASP.NET MVC 4 Entity Framework. In addition, it will be coded using Microsoft Visual Studio 2012.

Software – Player

The player software receives the song requests from the patrons via the WeListen Web Application Programming Interface (API). The songs that are received through the API are placed into the play list. The songs are played from the playlist in the order in which they are received, however; the business owner or DJ has the option of removing songs or moving the play order of the songs in the play list by using the playlist controls in the WeListen Player Software.
The WeListen Player software has been written as a stand-alone C# application that will be ran on the business’s computer. The WeListen player was coded using Microsoft Visual Studio 2013.

**Database**

WeListen requires the use of a SQL Server database. This database is used to hold all sorts of information including, but not limited to: information associated with the user, song, location and business.

For development of our database, we used SQL Server Management Studio 2012. Both software elements will interact with the same database providing synchronized data between the two features.

**API and Library**

The WeListen Music System makes use of several APIs and software libraries to allow the software to function properly and more efficiently. Each of the libraries and APIs bring specific tools and functions to the WeListen Music System to improve the look, feel and operation of the software to create a pleasant user interface (UI).

**Amazon API**

The WeListen Music System makes use of the Amazon Web Services API. This API is used in several places throughout the WeListen Music System to maintain an up to date list of available music as well as provide a vendor to provide the ability of users to purchase MP3 music files. The Amazon API is used as the sole source of new music at this stage of development. The option to incorporate more sources of music in the future will be reviewed.
during future development. There are two structural pieces offered by Amazon that will help us with our database features.

“The Amazon API comes as a simple web Service. Here are the first web services that they have exposed:

- Product Display: The ability to search for and retrieve product information from the Amazon.com catalog.

- Shopping Cart: The ability to add Amazon.com products to shopping carts, wish lists, and registries from third party locations” (http://www.phpclasses.org/package/642-PHP-A-PHP-XML-class-to-access-Amazon-API-they-released.html).

These two features help present content to the users in an organized fashion along with the ability to easily purchase music from the application through the Amazon web service.

In order to access the data that amazon has available, the WeListen team developed the AmazonAccesser class (See Appendix B). The AmazonAccesser class was created on both the desktop and web applications and is one of the most essential classes in the WeListen Music System for properly handling music. Every song that is uploaded or purchased will be handled through this class.

**NAudio**

The WeListen Music System makes use of the NAudio Library. This library provides all of the tools that the WeListen Music System uses to play the music on the player portion of the software.
NAudio describes itself as follows, “NAudio is an open source .NET audio and MIDI library, containing dozens of useful audio related classes intended to speed development of audio related utilities in .NET. It has been in development since 2002 and has grown to include a wide variety of features. While some parts of the library are relatively new and incomplete, the more mature features have undergone extensive testing and can be quickly used to add audio capabilities to an existing .NET application (NAudio).”

Through the use of NAudio the software was developed to be able to play any MP3 file that is available on the local machine weather it was sent via the WeListnen Web API or selected by the DJ or business owner locally.

**Windows Presentation Foundation Sound Visualization Library (WPFSVL)**

The WeListen Music System makes use of the Windows Presentation Foundation Sound Visualization Library (WPFSVL). This library was particularly useful in expediting the process of visualizing the UI built into the WeListen Player desktop application.

The description provided on the WPFSVL website says the following, “The WPF Sound Visualization Library is a collection of WPF Controls for graphically displaying data related to sound processing. Current controls include a Spectrum Analyzer, Waveform Timeline, and an Album Art Display Panel (WPFSVL).”

**WeListen Web API**

In order to create a line of communication between the WeListen Music System desktop application and the Web application. This API allows the DJ to upload the meta-data from each song that is stored on their local machine to the database server. The API also takes the requests from the users on the Web site and send the request to the appropriate DJ at the appropriate
location. The WeListen Web API makes use of both Extensible Markup Language (XML) and JavaScript Object Notation (JSON) to send the data between the web and the local machines.

**Web Development**

The main focus of WeListenMusic’s webpage is to provide a clean, simple and easy to use interface which allows clients and customers to queue the music that they want to hear at their convenience. WeListenMusic uses Microsoft’s ASP.NET MVC4 and Twitter’s Bootstrap to help with the look and feel of the website. Leveraging the use of Bootstrap, making this site mobile friendly was much easier for the WeListen team. Therefore allowing users to have the ultimate accessibility in being able to access the site right from their phone as well as a desktop computer, laptop or a tablet. The use of Bootstrap made it possible for the WeListen team to create a website with a responsive design. According to web design company’s website, “A responsive website changes its appearance and layout based on the size of the screen the website is displayed on (Thridda Design Studios).” By using Bootstrap to create a responsive design in the WeListen Music System website, the WeListen team was able to make the website’s user interface easy to access regardless of the type of device the customer is using to access the website.

**Commercial**

The WeListen Music System is free to use for both the business and the patrons. By using the Amazon Web API and being registered with Amazon Web Services, the WeListen Music team receives a portion of every MP3 that is sold through the WeListen website. The WeListen Team believes that as the use of the system grows that there is a large potential for profit through the use of this product.
Future Development

A plan has been developed to continue to improve the WeListen Music System. Ideas that were not possible to complete due to the time constrains of this project have been documented and will be implemented in future releases of the software. Below are descriptions of some of the future tools that will be developed into the WeListen Music System.

Genre Filtering

In future releases of the WeListen Player, the plans to incorporate setting to filter the types of music that the business will allow to be played will be incorporated. These filters will allow the DJ to limit what types of music will be able to be played at different types of business establishments. By enabling the filters, it would avoid the customers being able to search and request hip-hop music at a country themed club, or classical music at a rock and roll establishment.

DJ Controls

Future versions of the player software will look to incorporate some of the more advanced features that DJs use to enhance their shows. Some examples of these tools are described below.

Sampling controls are used to set up a button to loop and repeat sections of audio or recordings at the push of a button. The NAudio Music Library has controls to allow for recording samples and playing them back.

Duo-player controls are used by DJ’s to seamlessly transition between songs. Future editions of the WeListen Player will incorporate a second set of controls for playing music, a cross-fade slider to fade between the two players, speed controls to allow the DJ to match the speed of the songs on the players as well as individual volume and equalizer controls.
The goal of this software is to eventually provide software that will be usable even to the most advanced DJ.

**Music Library Management System**

Future versions of the player software will include a music library management system that will allow the DJ or business owner to view, search, upload and edit the music that is uploaded to the database. This will improve the way that the DJ or business owner will be able to keep track of and manage their music.

**Offline Operations**

Future versions of the player software will look to develop a local database that will allow the software and request system to operate via a Local Area Network (LAN) system. This would allow the DJ to set up an additional laptop or tablet to be used as a request station for customers to be able to search the available music in the catalog and send requests to the DJ. This offline operation will not allow music to be purchased like it would be if the system is online.

**Budget**

The proposed budget was created prior to the development process and was estimated based off the intention that there will be low usage of the program at beginning stages. Therefore, as the amount of usage increases, costs will need to be adjusted accordingly. Therefore, this budget is subject to change.
Table 1 Budget

Design Elements

User Controls

See the User Profile Table (Table 2) to get an in-depth description of the types of users that were kept in mind as we progressed with the development of the WeListen Music System.
## User Profile Table

<table>
<thead>
<tr>
<th>Application:</th>
</tr>
</thead>
<tbody>
<tr>
<td>WeListen – Djital DJ - Jukebox</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Potential Users:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. DJ’s</td>
</tr>
<tr>
<td>2. Business owners</td>
</tr>
<tr>
<td>3. music listenersaqqaa</td>
</tr>
<tr>
<td>4. Business patrons</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Software and Interface Experience:</th>
</tr>
</thead>
<tbody>
<tr>
<td>The business owners and DJs should have a basic knowledge of MP3 based music players and/or DJ software. The interface that these users will be on their local system. The web portion of this program will have a clean and smooth user interface and should be suitable for the average computer user.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experience with Similar Applications:</th>
</tr>
</thead>
<tbody>
<tr>
<td>I have sought out and used some of the expensive existing services on the market. While the interface is usually familiar and easy to use, there is a high cost associated with using the software incurred by both the customer and the business. As a customer the existing jukebox systems charge over $1.00 per song to be played each time</td>
</tr>
</tbody>
</table>
you play it. The business also pays a service fee to keep the system in their establishment.

<table>
<thead>
<tr>
<th>Task Experience:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Business Patron</td>
</tr>
<tr>
<td>- Locate songs by typical meta data. (Title, Artist, Genre, Album, etc.)</td>
</tr>
<tr>
<td>- See if the song is available for free play or purchase.</td>
</tr>
<tr>
<td>- Login to account to view favorites to select or purchase songs not in catalog</td>
</tr>
<tr>
<td>- Select songs and send to be played.</td>
</tr>
<tr>
<td>2. Business owner and DJ</td>
</tr>
<tr>
<td>- Songs appear in a request list for the DJ or at the end of the play list for the jukebox.</td>
</tr>
<tr>
<td>- DJ selects song from the request list and sends them to the play list.</td>
</tr>
<tr>
<td>- DJ has control of typical DJ features like speed, fade, track levels and others.</td>
</tr>
<tr>
<td>- Jukebox shows the songs to be played in the order they are received.</td>
</tr>
</tbody>
</table>
- The Jukebox has checks to assure that songs cannot be played more than one time within a set amount of time.

### Frequency of Use:

This program has the intention and potential to be used daily by businesses and DJ’s. DJ’s can bring the system with them to any establishment that they work at, while businesses would have the system set up to run on its own at all times. A durable customer interface kiosk is being considered for the patron interface.
Key Interface Design Requirements that the Profile Suggests:

Patron Interface:

1. Simple interface similar to any MP3 player/purchase software (I.E. ITunes)
2. Prominent and functional search
3. Select button associated with each search result
4. Purchase button/screen for songs not already available in the catalog
5. Send button to send selections to the DJ or the Jukebox.
6. Simple and usable forms
7. Login screen to access patron account
8. Favorites area to store patrons favorite songs to be played on a regular basis

Business Interface - Jukebox:

1. Typical MP3 software controls (I.E. Play, Pause, Back, forward, etc.)
2. Play list view to see the upcoming, selected songs.
3. Override controls to skip songs or allow multiple plays of the same song.
4. Random play for light business times when selections are not coming in at regular intervals.

Patron Interface:
1. Interface similar to any MP3 based DJ software (I.E. Virtual DJ)
2. Request list with song information and requester’s name
3. Duo-play song control interface with fade for mixing.
4. Search feature to find catalog music.
5. DJ controls for track levels, speed, stop, play, forward, back, etc.

Table 2 User Profile Table

Entity-Relationship Diagram

Prior to development of WeListen, an Entity-Relationship (ER) diagram was created to help plan out the design of the database. Creating an ER diagram was important because it helps ensure that the design is consistent and accurate for future reference.

The below ER Diagram illustrates all of the entities and their connections to other entities through relationships. This ER Diagram is a good visual representation of the database.

Figure 5 WeListen – ER Diagram
Prior to development of WeListen, a Unified Modeling Language (UML) diagram was also created to help plan out the design of the database. A professional Database Administrator (DBA) was consulted to guide us while creating this diagram to be sure that the tables and data that we were keeping track of would best suit our project. Creating an UML diagram was important because it helps ensure that the design is consistent and accurate for future reference.

The below UML Diagram illustrates all of the tables and entities and how they relate to each other. This UML Diagram is another useful visual representation of the database.
Use Case Diagram

The idea is for the Establishment Manager to control media access by genre and title while the DJ, if applicable, controls the user requests and playlist of music. During an example scenario, the user will approach the kiosk and either select a track to play, or purchase a new track for the establishment. The requested track will either be sent into the establishment Playlist for playback or to the DJ, if applicable, to be sorted and managed in the playlist. The user requests and song selections will be managed through Amazon Music Store API for title and categorical sorting while the user transactions during purchases will be managed securely through the integrated Amazon Web Services API.

Figure 6 Use Case Diagram
**Legal Issues**

When working with music has to make sure that the use of intellectual property is properly addressed. Legal methods of using music have been established and most businesses will have the licensing in place before this software is used.

When asked about these issues, a professional DJ said, “I rely on the business to have an established BMI license. This is a question that I always ask prior to accepting a gig to do shows at their establishment (Cook, 2013).”

According to the BMI licensing website, “Copyright laws require music users to get permission from songwriters and composers who can charge a fee before their music is played publicly, which then allows them to continue to create music. Purchasing the appropriate license for your business allows you to play the music you need for your business without fear of breaching the copyright laws (BMI).”

By maintaining the required license for the type of business the use of music will be allowed with the appropriate restrictions. WeListen has incorporated the licensing issues in the terms and conditions section of the software and will accept no responsibility for businesses that attempt to use the software without the proper licensing.

In addition to the music licensing, the users will be required to agree to the Terms and Conditions of the WeListen Music System (See Appendix A). The first part of the Terms and Conditions includes the End User License Agreement (EULA). The EULA has been written to cover the legal issues that many programs and applications deal with.
The WeListen Music System also includes a privacy statement (See Appendix A). The privacy statement includes all of the details about how the WeListen Music Team will handle and protect the personal information of the system users.

Conclusion

Over the 2013-2014 school year the WeListen team has worked exceptionally hard to develop the WeListen Music System. We have learned many developing techniques that we had not experienced in our time at University Of Cincinnati. The WeListen Team not only learned about using different programming methods, but also about working together to complete the project in the time that was available.

While working on a team didn’t always go smoothly, in the end the WeListen team was able to resolve differences and in the end it became an valuable educational experience for all involved with the project.
Works Cited


Apendix A - Terms and conditions

WeListen Privacy Statement

Information Collection and Use

WeListen is the sole owner of the information collected on this site. We will not sell, share, or rent this information to others in ways different from what is disclosed in this statement.

WeListen collects information from our users at several different points on our website.

Registration

In order to use the Service, and be part of the WeListen online community, the users have the opportunity to complete the registration form. During registration a user is required to give his email, and, optionally, other contact informations. This information is used to contact the user about the services on our site for which they have expressed interest, and to allow a more intense involvement in the WeListen community. It is optional for the user to provide demographic information, but encouraged so we can provide a more personalized experience on our site.

Order

The ordering of WeListen is processed by third party companies which are responsible for the use of the informations they may gather. Please consult their privacy statements on their respective websites:

Amazon.com

http://www.amazon.com

All credit card informations are requested and transmitted securely by these third party companies. We never store any credit card information in our databases.
Cookies

A cookie is a piece of data stored on the user’s hard drive containing information about the user. Usage of a cookie is in no way linked to any personally identifiable information while on our site.

WeListen does not currently use cookies, however; WeListen reserves the right to do so in future releases of our software. In the even cookies are used, once the user closes their browser, the cookie simply terminates. Cookies are optionally used on our website to the convenience of the user. If a user rejects the cookie, they may still use our site. We don't use to track and target the interests of our users, neither to transmit passwords.

Some of our business partners use cookies on our site. However, we have no access to or control over these cookies.

Log Files

WeListen currently does not keep log files, however, WeListen reserves the right to do so in future releases of the WeListen software. WeListen will use IP addresses to analyze trends, administer the site, track user’s movement, and gather broad demographic information for aggregate use. IP addresses are not linked to personally identifiable information

Sharing

In the event that WeListen starts keeping log files, WeListen will share aggregated demographic information with our partners and advertisers. This is not linked to any personal information that can identify any individual person. We will use an outside shipping company to ship orders, and a credit card processing company to bill users for goods and services. These companies do not retain, share, store or use personally identifiable information for any secondary purposes.
We partner with another party to provide specific services. When the user signs up for these services, we will share names, or other contact information that is necessary for the third party to provide these services. These parties are not allowed to use personally identifiable information except for the purpose of providing these services.

Links
This web site contains links to other sites. Please be aware that we, WeListen, are not responsible for the privacy practices of such other sites. We encourage our users to be aware when they leave our site and to read the privacy statements of each and every web site that collects personally identifiable information. This privacy statement applies solely to information collected by this Web site.

Surveys and Contests
From time-to-time our site may request information from users via surveys or contests. Participation in these surveys or contests is completely voluntary and the user therefore has a choice whether or not to disclose this information. Information requested may include contact information, and demographic information. Contact information will be used to notify the winners and award prizes. Survey information will be used for purposes of monitoring or improving the use and satisfaction of this site Security.

This website takes every precaution to protect our users’ information. When users submit sensitive information via the website, your information is protected both online and off-line. Only employees who need the information to perform a specific job are granted access to personally identifiable information. Our employees must use password-protected screen-savers when they leave their desk. When they return, they must re-enter their password to re-gain access to your information. Furthermore, ALL employees are kept up-to-date on our security and
privacy practices. Every quarter, as well as any time new policies are added, our employees are notified and/or reminded about the importance we place on privacy, and what they can do to ensure our customers’ information is protected.

If you have any questions about the security at our website, you can contact us at WeListenMusic.com.

**Special Offers**

We send all new members a welcoming email to verify password and username. Established members will occasionally receive information on products, services, special deals, and a newsletter. Out of respect for the privacy of our users we present the option to not receive these types of communications. Please see our choice and opt-out below.

**Site and Service Updates**

We also send the user site and service announcement updates. Members are not able to unsubscribe from service announcements, which contain important information about the service. We communicate with the user to provide requested services and in regards to issues relating to their account via email or phone.

**Correction/Updating Personal Information**

If a user’s personally identifiable information changes, or if a user no longer desires our service, we will endeavor to provide a way to correct, update or remove that user’s personal data provided to us. This can usually be done at the member information page or by emailing our Customer Support.
Notification of Changes

If we decide to change our privacy policy, we will post those changes on our Homepage so our users are always aware of what information we collect, how we use it, and under circumstances, if any, we disclose it.

Privacy Contact

For any further information about the privacy statement of Virtual DJ, please contact us at WeListenMusic.com.

WeListen End User License Agreement (EULA)

- This license gives you the right to install and use the software on a maximum of two computers, by one person.

- It does not give you the right to rent or sell computer systems on which the software is installed, or to rent or sell the software or part of it.

- The software is distributed "as-is" and no warranty of any kind is expressed.

- WeListen and its suppliers retain all intellectual property rights in the software.

- Once activated, the software cannot be refunded, unless found to be defective by WeListen Technical Support.

- Software purchased on-line for instant download through our website is considered opened at the time of purchase, and therefore no refund can be given.
1. PERMITTED USE AND RESTRICTIONS

You are granted one license to:

- install and use the Software on one computer.
- install and use the Software on a second computer, if this computer is normally used by the same person and the Software will not be used on both computers at the same time.
- make back-up copies of the Software for archival purposes.

This license does NOT allow you to:

- rent, lease, sell, lend, distribute, publish, bundle or sublicense the Software or any part of it.
- rent, lease, sell, lend or distribute hardware on which the Software is installed, unless specifically authorized by WeListen.
- Electronically transfer the Software from one computer to another or over a network.
- decompile, reverse engineer, disassemble, or otherwise reduce the Software to a human-perceivable form.
- modify the Software or create derivative works based upon the Software.

2. OWNERSHIP

The foregoing license gives you limited rights to use the Software.

WeListen and its suppliers retain all right, title and interest, including all copyrights, in and to the Software and all copies thereof. All rights not specifically granted in this EULA, including International Copyrights, are reserved by WeListen and its suppliers.
3. COPYRIGHT INFRINGEMENT

Unauthorized copying, distribution, modification, public display, or public performance of copyrighted works is an infringement of the copyright holders’ rights. As a condition to your agreement with WeListen, you agree that you will not use the Software to infringe the intellectual property rights of others in any way.

All licenses required to publicly show or play intellectual property is the responsibility of the individual or the establishment in which the WeListen software is being used, including but not limited to audio files and video files.

4. WARRANTY

WeListen warrants that the media on which the Software is furnished, if any, will be free from defects in materials and workmanship.

WeListen offers no warranty of performance, express or implied, with regard to the licensed software and all accompanying materials. WeListen further disclaims all other warranties, express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose, with regard to the licensed software and all accompanying materials.

5. CUSTOMER REMEDY

Your exclusive remedy under the above limited warranty shall be, at WeListen option, either a full refund of the purchase price or correction of the defective software or media. Notice is hereby given however, that absent any defects, there will be no refunds issued on activated software.
6. DISCLAIMER OF DAMAGES

You assume responsibility for, among other things, (i) the selection of the Licensed Software to achieve your intended results, (ii) the acquisition of other software (including any programming or operating system software) and/or equipment compatible with the Licensed Software, and (iii) the installation, use and results obtained from the Licensed Software. Further, for the express purpose of limiting the liability against Atomix, you agree that, to the maximum extent permitted by law, WELISTEN SHALL IN NO EVENT BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OR INABILITY TO USE THE LICENSED SOFTWARE, WHETHER DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL OR OTHERWISE, REGARDLESS OF THE FORM OF ACTION, EVEN IF ATOMIX HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL PROVISIONS

The software is protected by the copyright laws of the U.S. and other countries.

If any part of this agreement is found to be invalid or unenforceable, the remaining terms will stay in effect.

This agreement shall be governed by the laws of the state of Delaware, USA.

Legal Notice (WeListenMusic)

http://www.WeListenMusic.com

Copyright Pending

Trademarks:
The Logo trademark of WeListen is pending

Windows and DirectX are registered trademarks of the Microsoft Corporation.

License and restrictions:

According to the EULA (End User License Agreement) accepted by every user upon installation of WeListen:

You MAY NOT make or distribute copies of WeListen, or electronically transfer WeListen from one computer to another over a network.

You MAY NOT decompile, reverse engineer, disassemble, or otherwise reduce WeListen to less than human perceivable form.

You MAY NOT modify or create derivate works from WeListen.

You MAY NOT sell, to rent, to transfer or sublicense the software. You may not sell, to rent, to transfer or sublicense hardware on which the software is installed, without written prior agreement of WeListen.

Reproduction of this manual in full or in part without written permission from WeListen Productions is prohibited.
namespace WeListenPlayer
{
    class AmazonAccesser
    {
        // Amazon Keys
        // Provided with Amazon Product Advertising API account
        private const string accessKeyId = "KEY HERE";
        private const string secretKey = "SECRET HERE";

        // Song Data Object is passed to getAmazonInfo
        public async Task<SongData> getAmazonInfo(SongData song)
        {
            // Parses out title, artist, etc from SongData object
            var fullRequest = parseInfo(song);

            // getData sends fullRequest to amazon and returns
            // new SongData object from Amazon API
            var newSong = await getData(song, fullRequest);

            return newSong;
        }

        public string parseInfo(SongData song)
        {
            var fullRequest = "";
            var artist = song.Artist;
            var album = song.Album;
            var title = song.Title;

            // Remove anything in parenthesis/brackets and all special characters
            string regEx = @"(?<=\()((.\*)|(?=\))|(?<=\[)(.\*)|(?=\]|(?<=\{)(.\*)|(?=\})|[^\\w ]")";

            // Handle passed parameters
            if (artist != "UNKNOWN")
            {
                artist = Regex.Replace(artist, regEx, "");
                fullRequest += artist;
            }

            if (album != "UNKNOWN" && album != artist)
            {
                album = Regex.Replace(album, regEx, "");
                fullRequest += " " + album;
            }
        }
    }
}
if (title != "UNKNOWN" && title != album)
{
    title = Regex.Replace(title, regEx, "");
    fullRequest += " " + title;
}

return fullRequest;

// Primary method to make a request from Amazon
public async Task<SongData> getData(SongData newSong, string fullRequest)
{
    try
    {
        // Instantiate Amazon ProductAdvertisingAPI client
        BasicHttpBinding binding = new
            BasicHttpBinding(BasicHttpSecurityMode.Transport);
        binding.MaxReceivedMessageSize = int.MaxValue;
        AWSECommerceServicePortTypeClient amazonClient = new
            AWSECommerceServicePortTypeClient(binding, new
                EndpointAddress("https://webservices.amazon.com/onca/soap?Service=AWSECommerceService"));

        // add authentication to the ECS client
            AmazonSigningEndpointBehavior(accessKeyId, secretKey));

        // prepare an ItemSearch request
        ItemSearchRequest request = new ItemSearchRequest();
        request.SearchIndex = "MP3Downloads";
        request.RelationshipType = new string[] { "Tracks" };
        request.ResponseGroup = new string[] { "ItemAttributes", "Images", "Offers", "RelatedItems" };

        request.Keywords = fullRequest;

        ItemSearch itemSearch = new ItemSearch();
        itemSearch.Request = new ItemSearchRequest[] { request };
        itemSearch.AWSAccessKeyId = accessKeyId;
        itemSearch.AssociateTag = "1330-3170-0573";

        // send the ItemSearch request
        ItemSearchResponse response = amazonClient.ItemSearch(itemSearch);

        var item = response.Items[0].Item[0];

        //<ProductTypeName>DOWNLOADABLE_MUSIC_TRACK</ProductTypeName>
        if (response.Items[0].Item[0].ItemAttributes.ProductTypeName == "DOWNLOADABLE_MUSIC_ALBUM")
        {
            item = response.Items[0].Item[1];
        }

        // if no response to search
        if (item == null)
        {

try
{
    // Try new search and remove the album
    newSong.Album = "UNKNOWN";
    // Re-iterate over the search method
    await getData(newSong, fullRequest);
}
catch
{
    // Removing the album produced no results
    // Continue forward...
}
)

// Get year from full Release Date
var formatYear = DateTime.Parse(item.ItemAttributes.ReleaseDate).Year;

// Pull all other variables from request and return as SongData obj
newSong.UserID = 1;
newSong.LocationID = 1;
newSong.Album = item.RelatedItems[0].RelatedItem[0].Item.ItemAttributes.Title;
newSong.Artist = item.ItemAttributes.Creator[0].Value;
newSong.Title = item.ItemAttributes.Title;
newSong.Year = (int)formatYear;
newSong.Genre = item.ItemAttributes.Genre;
newSong.FilePath = "";
newSong.Duration = (int)item.ItemAttributes.RunningTime.Value;
newSong.Price = item.Offers.Offer[0].OfferListing[0].Price.FormattedPrice;
newSong.ASIN = item.ASIN;
newSong.Artwork = item.LargeImage.URL;

return newSong;
}
catch
{
    return newSong;
}
}
using System;
using System.Configuration;
using System.ServiceModel;
using System.Text.RegularExpressions;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;
using WeListenPlayer.Amazon.PAAPI;

namespace WeListenPlayer
{
    class AmazonAccesser
    {
        // Amazon Keys
        // Provided with Amazon Product Advertising API account
        private const string accessKeyId = "KEY HERE";
        private const string secretKey = "SECRET HERE";

        // Song Data Object is passed to getAmazonInfo
        public async Task<SongData> getAmazonInfo(SongData song)
        {
            // Parses out title, artist, etc from SongData object
            var fullRequest = parseInfo(song);

            // getData sends fullRequest to amazon and returns
            // new SongData object from Amazon API
            var newSong = await getData(song, fullRequest);
            return newSong;
        }

        public string parseInfo(SongData song)
        {
            var fullRequest = "";
            var artist = song.Artist;
            var album = song.Album;
            var title = song.Title;

            // Remove anything in parenthesis/brackets and all special characters
            string regEx = @"(?<="\()\(.*)\(?=\))\|(?<=[\[])(.*?)(?=\])\|(?<={})(.*?)(?=\})\|[^\w ]";

            // Handle passed parameters
            if (artist != "UNKNOWN")
            {
                artist = Regex.Replace(artist, regEx, "");
                fullRequest += artist;
            }

            if (album != "UNKNOWN" && album != artist)
            {
                album = Regex.Replace(album, regEx, "");
                fullRequest += " " + album;
            }

            if (title != "UNKNOWN" && title != album)
            {
                title = Regex.Replace(title, regEx, "");
            }
        }
    }
}