**www.heydjplaymysong.com**

CS-670-A Research Project Seminar

Sacred Heart University

5151 Park Avenue, Fairfield, CT 06825-1000

Friday, December 6th, 2013

Ali B. Kaba

Master of Science in Computer Information Science (MSCIS)

Information Technology Track

Francesco Sardo

Table of Contents

[1 Abstract 6](#_Toc374062284)

[2 Introduction 7](#_Toc374062285)

[2.1 Summary of Problems, Opportunities and Directives 7](#_Toc374062286)

[2.2 Scope 8](#_Toc374062287)

[2.3 Intended Audience 8](#_Toc374062288)

[2.4 Computer Applications Used 8](#_Toc374062289)

[2.5 Web Applications Used 8](#_Toc374062290)

[2.6 Languages Used 8](#_Toc374062291)

[2.7 Libraries Used 8](#_Toc374062292)

[3 Analysis Overview 9](#_Toc374062293)

[3.1 System Usage 9](#_Toc374062294)

[3.2 Assumptions, Dependencies and Constraints 9](#_Toc374062295)

[3.3 Development Methods 9](#_Toc374062296)

[4 Requirements 10](#_Toc374062297)

[4.1 Inputs – Data. 10](#_Toc374062298)

[4.2 Outputs – Information. 11](#_Toc374062299)

[4.3 Processes – Manual/Automatic. 11](#_Toc374062300)

[4.4 Storage – Database 11](#_Toc374062301)

[4.5 Control – Interfaces 11](#_Toc374062302)

[4.6 Ease of use 11](#_Toc374062303)

[4.7 Timelines and deadlines 11](#_Toc374062304)

[4.8 Training 12](#_Toc374062305)

[4.9 Quality Management 12](#_Toc374062306)

[4.10 Security and Audits 12](#_Toc374062307)

[4.11 Use Cases 12](#_Toc374062308)

[4.12 Assess Project Worth in Terms of Cost vs. Value 17](#_Toc374062309)

[4.12.1 Estimated Costs 17](#_Toc374062310)

[4.12.2 Value 17](#_Toc374062311)

[4.13 Preliminary Project Plan 17](#_Toc374062312)

[4.13.1 Master Schedule 17](#_Toc374062313)

[4.13.2 Resource Assignment 17](#_Toc374062314)

[4.14 Browser compatibility 18](#_Toc374062315)

[4.15 Security Preventions 18](#_Toc374062316)

[4.15.1 Overview 18](#_Toc374062317)

[4.15.2 Information Gathering 18](#_Toc374062318)

[4.15.3 Configuration Management 18](#_Toc374062319)

[4.15.4 Business Logic, Authentication and Authorization 18](#_Toc374062320)

[4.15.5 Session Management 18](#_Toc374062321)

[4.15.6 Data Validation 18](#_Toc374062322)

[4.15.7 Denial of Service 18](#_Toc374062323)

[4.15.8 Ajax 18](#_Toc374062324)

[5 Design 19](#_Toc374062325)

[5.1 Use Case Diagram 19](#_Toc374062326)

[5.2 Context Data Flow Diagram 20](#_Toc374062327)

[5.3 Entity Relationship Diagram 21](#_Toc374062328)

[5.4 Class Diagram 21](#_Toc374062329)

[5.5 Dataflow Diagram 22](#_Toc374062330)

[5.6 Data Structured & Activity Diagrams 24](#_Toc374062331)

[5.6.1 Register 24](#_Toc374062332)

[5.6.2 Search 26](#_Toc374062333)

[5.6.3 Activate account 27](#_Toc374062334)

[5.6.4 Reset passcode 29](#_Toc374062335)

[5.6.5 Reset password (forgot password) 30](#_Toc374062336)

[5.6.6 Change password 31](#_Toc374062337)

[5.6.7 Login 33](#_Toc374062338)

[5.6.8 Manage playlist 34](#_Toc374062339)

[5.6.9 Create playlist 37](#_Toc374062340)

[5.6.10 Logout 38](#_Toc374062341)

[5.6.11 Upload playlist 39](#_Toc374062342)

[5.8 Conceptual Website Diagram 40](#_Toc374062343)

[6 Database Stored Procedures 40](#_Toc374062344)

[6.1 Account Activation 41](#_Toc374062345)

[6.2 Activation Check 41](#_Toc374062346)

[6.3 Create Account 41](#_Toc374062347)

[6.4 Create Song 41](#_Toc374062348)

[6.5 Delete All 42](#_Toc374062349)

[6.6 Delete One Song 42](#_Toc374062350)

[6.7 Delete Playlist 42](#_Toc374062351)

[6.8 Down Rating 42](#_Toc374062352)

[6.9 Email Duplicate Check 43](#_Toc374062353)

[6.10 Get All Playlist To Delete 43](#_Toc374062354)

[6.11 Get All Songs 43](#_Toc374062355)

[6.12 Get Playlist Name 43](#_Toc374062356)

[6.13 Get Rating 44](#_Toc374062357)

[6.14 Get Top 20 44](#_Toc374062358)

[6.15 Login 44](#_Toc374062359)

[6.16 Manage Playlist Check 44](#_Toc374062360)

[6.17 Reset Activation 45](#_Toc374062361)

[6.18 Reset One Song 45](#_Toc374062362)

[6.19 Reset Password (forgot password) 45](#_Toc374062363)

[6.20 Reset Playlist 45](#_Toc374062364)

[6.21 Status Check 46](#_Toc374062365)

[6.22 Up Rating 46](#_Toc374062366)

[7 Screenshots 47](#_Toc374062367)

[7.1 index.php 47](#_Toc374062368)

[7.2 dashboard.php 47](#_Toc374062369)

[7.3 manageplaylist.php 48](#_Toc374062370)

[7.4 createplaylist.php 48](#_Toc374062371)

[7.5 uploadplaylist.php 49](#_Toc374062372)

[7.6 activate.php 49](#_Toc374062373)

[7.7 changepassword.php 50](#_Toc374062374)

[7.8 resetpassword.php 50](#_Toc374062375)

[7.9 forgotpassword.php 51](#_Toc374062376)

[7.10 music.php 51](#_Toc374062377)

[8 English Instruction Manual 52](#_Toc374062378)

[8.1 Using Hey DJ Play My Song 52](#_Toc374062379)

[8.2 Getting Started 52](#_Toc374062380)

[8.2.1 Getting/creating the CSV document 52](#_Toc374062381)

[8.2.2 Getting the music 52](#_Toc374062382)

[8.2.3 Formatting the CSV for HDJPMS system 53](#_Toc374062383)

[8.3 Setting up an account 54](#_Toc374062384)

[8.3.1 Visiting HeyDJPlayMySong.com 54](#_Toc374062385)

[8.3.2 Creating an account 55](#_Toc374062386)

[8.3.3 Verifying your account 55](#_Toc374062387)

[8.4 Dashboard 57](#_Toc374062388)

[8.4.1 Change password 57](#_Toc374062389)

[8.4.2 Manage playlist 58](#_Toc374062390)

[8.4.3 Create playlist 59](#_Toc374062391)

[8.5 Search 60](#_Toc374062392)

[8.5.1 Music 61](#_Toc374062393)

[8.5.2 Voting 62](#_Toc374062394)

[8.6 Homepage 63](#_Toc374062395)

[9.0 Enhancements 64](#_Toc374062396)

[10 Glossary 67](#_Toc374062397)

[11 Bibliography 68](#_Toc374062398)

[11.1 Books 68](#_Toc374062399)

[11.2 Internet Sites 68](#_Toc374062400)

[12 Conclusion 69](#_Toc374062401)

This page was purposely left blank

# 1 Abstract

Hey DJ Play My Song (HDJPMS) is a music voting system web application. The intended audience of this product is for working professional Disc Jockeys (DJs) however it can be used by friends and family or even someone who plans to hire a DJ. It is a completely separate system from the actual device that is used to use to play the song.

Join the fun, place your playlist online for your friends and family to vote on! Where ever they are!

# Introduction

## 2.1 Summary of Problems, Opportunities and Directives

HDJPMS’s goal is to do the same outcome task as other music voting system on the market and it is as followed:

* Ability to create a playlist with favorite songs.
* Ability to search a song and request it to be played.
* Ability to vote up or down on an already requested song.

However some of the current systems require the following before voting:

* A setup fee.
* Personal information such as name, address.
* Ownership of the songs which can be a legal issue.
* Direct connection to the music system being used to play the song which is a potential threat.

The risk assessment here is that with guest connected to the music systems with users which can be a potential cause the following depending on the value of the playlist to the DJ/host:

* Negligible.
* Slight loss of competitive advantage.
* Significant loss of competitive advantage.
* Significant financial loss.
* Significant business profit.
* Serious loss of life.

HDJPMS tackles the following ideal:

* Free to use.
* Complete anonymous voting system.
* Fast registration with email for verification only, no name or address needed.
* HDJPMS is an entity of its own, doesn’t connect directly to music system.
* This potential works well with legacy systems or closed systems.
* Peace of mind from malicious software.

With the user creating their own playlist, whether they are a DJ or hiring one for an event, this application can be used as a music database which can later on be used during the event for guest to vote on.

Potential directives would be to partner with a paid music record pool and be offered as an incentive for being part of that record pool.

## 2.2 Scope

This document contains a detail description of HDJPMS. This system is includes the integration of PHP, JavaScript, jQuery Mobile on the front end and MySQL on the backend. This document does not describe in detail the 3rd party technologies being used to implement the system.

## 2.3 Intended Audience

The audience for this document is anyone looking to understand this project and how it is designed.

## 2.4 Computer Applications Used

* Agent Ransack
* Apple iTunes
* Fiddler2
* FileZilla
* Gadwin PrintScreen
* Google Chrome Version 31.0.1650.57
* Internet Explorer 11
* jZip
* Microsoft Excel
* Microsoft PowerPoint
* Microsoft Visio
* Microsoft Word
* MySQL Workbench 6.0 CE
* Notepad++
* PuTTY

## 2.5 Web Applications Used

* CPanel
* Dropbox
* GiTHub
* JSFiddle
* JustHost.com

## 2.6 Languages Used

* Ajax
* CSS3
* HTML 5
* JavaScript
* PHP: Hypertext Preprocessor (PHP) 5.4

## 2.7 Libraries Used

* Bootstrap
* Joyride (ZURB University)
* jQuery
* jQuery Mobile
* Uploadifive

# 3 Analysis Overview

## 3.1 System Usage

For a DJ or host to use this application, they first must get a comma-separated value (CSV) document, also freely available on www.heygodjplaymysong.com/about.php. Once the document is downloaded, they must register & activate their account. Once activated, they must create a playlist name and upload the CSV document onto the system. Once that is complete, they must announce to the guests on about the playlist name.

For a guest to use this application, they must first have a playlist name provided to them by the host. With the playlist name, they must visit www.heygodjplaymysong.com and type in the passcode in the search field. Once it is entered, they must hit search to be presented with two list:

* Top 20 Requested Songs
  + Ability to vote up or down songs.
* All Songs
  + Ability to search for songs and vote them up or down.

## 3.2 Assumptions, Dependencies and Constraints

This system is composed of many separate distributed components. The website is hosted with www.JustHost.com and therefore the up-time of the system is on their end. I decided to use PHP because it was the only language that would work with MySQL which was already available. I was also required to use Ajax as a solution of updating MySQL for the voting to be updated live I decided to use jQuery and Bootstrap libraries because they were made designed for mobile devices.

Some security aspects, the website is not secured due to budget, resource or both. See section 4.15 for more details.

## 3.3 Development Methods

The system is developed on a custom machine:

|  |  |
| --- | --- |
| Machine | |
| OS Name | Microsoft Windows 7 Professional |
| Version | 6.1.7601 Service Pack 1 Build 7601 |
| OS Manufacturer | Microsoft Corporation |
| System Manufacturer | INTEL\_ |
| System Model | DZ68BC\_\_ |
| System Type | x64-based PC |
| Processor | Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz, 3401 Mhz, 4 Core(s), 8 Logical Processor(s) |
| BIOS Version/Date | Intel Corp. BCZ6810H.86A.0021.2011.0831.1555, 8/31/2011 |
| SMBIOS Version | 2.6 |
| Total Physical Memory | 16.0 GB |
| Total Virtual Memory | 32.0 GB |
| C Drive | 238 GB |

The software I used to create the database is MySQL Workbench 6.0 CE, and the software I used to write the the webpages is Notepad++. I also used some web applications to test some coding in the applications (see 2.5).

# 4 Requirements

## 4.1 Inputs – Data.

**User:**

* Enter playlist.
* Register.
* Upload playlist.
* Reset playlist.
* Delete playlist
* Reset song.
* Delete song.

## 4.2 Outputs – Information.

* Playlist.

## 4.3 Processes – Manual/Automatic.

**Manual:**

* User search song.
* User builds playlist file.
* User uploads playlist file.

**Automatic:**

* Playlist is collected from Database.
* User inputs are subsequently modifying database.

## 4.4 Storage – Database

* All data will be stored in MySQL database.

## 4.5 Control – Interfaces

* Administrator can modify playlist and its songs.
* User can vote anonymously without registering via the homepage

## 4.6 Ease of use

* Documentation will be downloadable from the website
* Tour is implemented on the website for ease of understanding.

## 4.7 Timelines and deadlines

The design and development of HDJPMS with the scope currently presented would be 4 months or less.

|  |  |  |
| --- | --- | --- |
| Iteration 1 | | |
| Phase | Start Date | End Date |
| Inception | 01/11/13 | 02/22/13 |
| Iteration 2 | | |
| Inception | 02/22/13 | 03/15/13 |
| Elaboration | 03/15/13 | 05/25/13 |
| Iteration 3 | | |
| Inception | 05/25/13 | 06/01/13 |
| Elaboration | 06/01/13 | 07/01/13 |
| Construction | 07/01/13 | 11/14/13 |
| Iteration 4 | | |
| Construction | 11/14/13 | 11/30/13 |
| Transition | 11/30/13 | 11/30/13 |

## 4.8 Training

* Documentation is available on the website for training.
* Tour is also available for basic understanding.

## 4.9 Quality Management

* Data is user derived therefore training will be as details as possible.

## 4.10 Security and Audits

* System has not direct interaction with music system.
* Website doesn’t used HTTPS/SSL however it uses sessions.
* Certain pages include a session page which checks if it is a registered user.

## 4.11 Use Cases

|  |  |
| --- | --- |
| Use Case Name | Create an account |
| Description | Create an account with HDJPMS |
| Actors | User |
| Pre-Conditions | Valid email address and password between 6-12 characters long |
| Basic Flow | 1. Click on Create account button 2. Enter email address 3. Enter password between 6-12 characters long |
| Post Conditions | User is logged on into dashboard page  User is notified account has been created and must activate to use additional features  Activation code emailed to user to verify account |
| Alternate Flows | N/A |
| Notes | N/A |

|  |  |
| --- | --- |
| Use Case Name | Search |
| Description |  |
| Actors | User, registered user |
| Pre-Conditions |  |
| Basic Flow |  |
| Post Conditions |  |
| Alternate Flows | N/A |
| Notes | N/A |

|  |  |
| --- | --- |
| Use Case Name | Activate account |
| Description | Activate account to use additional features |
| Actors | Registered user |
| Pre-Conditions | User must have already created an account and received an email with an activation passcode |
| Basic Flow | 1. Click on Sign in if user not already logged in 2. Select activate account 3. Type in activation passcode into field 4. Hit activate account |
| Post Conditions | User is automatically redirected to dashboard page  User’s account is activated and has access to manage playlist |
| Alternate Flows | 1. Click on email that was sent after account was created 2. User will be notified account is activated and directed to HDJPMS homepage |
| Notes | N/A |

|  |  |
| --- | --- |
| Use Case Name | Reset passcode |
| Description | Reset passcode |
| Actors | Registered user |
| Pre-Conditions | User must have already created an account |
| Basic Flow | 1. Click on Sign in if user not already logged in 2. Select activate account 3. Hit reset passcode |
| Post Conditions | User is automatically redirected to dashboard page  User’s account is still not activated  Email is sent to user with new activation passcode |
| Alternate Flows | N/A |
| Notes | N/A |

|  |  |
| --- | --- |
| Use Case Name | Reset password (forgot password) |
| Description | Reset password (forgot password) |
| Actors | Registered user |
| Pre-Conditions | User must have already created an account |
| Basic Flow | 1. Enter your email address 2. Hit reset button |
| Post Conditions | User will receive a temporary password to their email associated with their account |
| Alternate Flows | Email support@heydjplaymysong.com and request to have password reset |
| Notes | N/A |

|  |  |
| --- | --- |
| Use Case Name | Change password |
| Description | Change password |
| Actors | Registered user |
| Pre-Conditions | User must have already created an account  User must already be logged onto Dashboard page  Password between 6-12 characters long |
| Basic Flow | 1. Double check email is correct otherwise type it in 2. Type in current password in designated field 3. Type in new password in designated field 4. Type in new password again in designed required field 5. Hit the submit button |
| Post Conditions | User is notified of password change  User is redirected to dashboard page |
| Alternate Flows | N/A |
| Notes | N/A |

|  |  |
| --- | --- |
| Use Case Name | Login |
| Description | Login |
| Actors | Registered user |
| Pre-Conditions | User must have already created an account |
| Basic Flow | 1. Enter email address 2. Enter password 3. Hit |
| Post Conditions | User will be redirected to the dashboard |
| Alternate Flows | N/A |
| Notes | N/A |

|  |  |
| --- | --- |
| Use Case Name | Delete playlist |
| Description | Delete playlist |
| Actors | Registered user |
| Pre-Conditions | User must have already created an account  User must be logged in  User must own the playlist which will be display in manage playlist |
| Basic Flow | 1. Type in the playlist name 2. Hit delete |
| Post Conditions | User will be notified the playlist was deleted  User will be redirected to the manage playlist |
| Alternate Flows | N/A |
| Notes | N/A |

|  |  |
| --- | --- |
| Use Case Name | Reset playlist |
| Description | Reset playlist |
| Actors | Registered user |
| Pre-Conditions | User must have already created an account  User must be logged in  User must own the playlist which will be display in manage playlist |
| Basic Flow | 1. Type in the playlist name 2. Hit reset |
| Post Conditions | User will be notified the playlist was reset  User will be redirected to the manage playlist |
| Alternate Flows | N/A |
| Notes | N/A |

|  |  |
| --- | --- |
| Use Case Name | Delete song |
| Description | Delete song |
| Actors | Registered user |
| Pre-Conditions | User must have already created an account  User must be logged in  User must own the playlist which will be display in manage playlist  User must have already searched for the song |
| Basic Flow | 1. Click the trash icon on the song 2. Refresh |
| Post Conditions | N/A |
| Alternate Flows | N/A |
| Notes | N/A |

|  |  |
| --- | --- |
| Use Case Name | Reset song |
| Description | Reset song |
| Actors | Registered user |
| Pre-Conditions | User must have already created an account  User must be logged in  User must own the playlist which will be display in manage playlist |
| Basic Flow | 1. Click the refresh icon on the song 2. Refresh |
| Post Conditions | N/A |
| Alternate Flows | N/A |
| Notes | N/A |

|  |  |
| --- | --- |
| Use Case Name | Create playlist |
| Description | Forgot password |
| Actors | Registered user |
| Pre-Conditions | User must have already created an account  User must be logged in  Playlist name must be available |
| Basic Flow | 1. Type playlist name 2. Hit continue |
| Post Conditions | User will be notified that the playlist is available |
| Alternate Flows | N/A |
| Notes | N/A |

|  |  |
| --- | --- |
| Use Case Name | Upload playlist |
| Description | Forgot password |
| Actors | Registered user |
| Pre-Conditions | User must have already created an account  User must be logged in  User already finished Use Case Name Create playlist |
| Basic Flow | 1. Click browse 2. Select csv file 3. Click upload |
| Post Conditions | User will be notified the upload is completed  User will be redirected to manage playlist |
| Alternate Flows | N/A |
| Notes | N/A |

|  |  |
| --- | --- |
| Use Case Name | Logout |
| Description | Logout |
| Actors | Registered user |
| Pre-Conditions | User must have already created an account  User must be logged in |
| Basic Flow | 1. Hit logout |
| Post Conditions | User will be notified that of being redirected in 3 seconds and if not click here (hyperlink to the homepage) |
| Alternate Flows | N/A |
| Notes | N/A |

## 4.12 Assess Project Worth in Terms of Cost vs. Value

### 4.12.1 Estimated Costs

**Operational Cost = $13,000.00**

* Computer hardware & software = $2,100.00.
  + Desktops, laptops, keyboards, software used to develop or maintain system.
* Hosting = $400.00 (3 years)
  + Database, website at external location.
* Books = $500.00
* Others = $10,000.00
  + Includes utilities, rent, travel, etc.

**Personnel Cost = $0.00**

* Myself = $0.00
  + Programmer, Database Administrator, System Analyst.

**Total Development Cost Estimate: $13,000.00**

### 4.12.2 Value

This application value is the ability is to allow a host to present a music selection to their company in a voting system which can be used with closed systems.

* Hey DJ Play My Song is a free web application.
* Users can save time with this application.

## 4.13 Preliminary Project Plan

### 4.13.1 Master Schedule

This system will consists of approximately ~2,000 lines of codes, the effort and schedule required to complete the system is estimated to be attainable in one year.

### 4.13.2 Resource Assignment

* **Personnel:**
  + 1 Programmer/DBA/System Analysts.

## 4.14 Browser compatibility

|  |  |
| --- | --- |
| Browser | Version |
| Chrome | 1. 🡪 31.0 |
| Firefox | 1.5, 2.0, 3.6 🡪 28.0 |
| Internet Explorer | 7.0 🡪 11.0 |
| Opera | 9.64, 9.8, 11.63, 11.64 |
| Safari | 3.2, 4.0, 5.1 |

## 4.15 Security Preventions

### 4.15.1 Overview

Here we will follow some of the Open Web Application Security Project’s (OWASP) Testing Guide v4.

### 4.15.2 Information Gathering

This is the first line of defense that I am failing since this is hosted on a public IP address from my host.

### 4.15.3 Configuration Management

Due of being hosting by a commercial company and residing on a public IP, it is easy to acquire the system that my application might be running on. The information is given with no effort.

### 4.15.4 Business Logic, Authentication and Authorization

I have implanted a session check for each page that should require a user id. If a user is to be accidently sent to a registered user page, that session check will bounce the user back to the homepage. A second check is placed for every reset and delete in the database.

### 4.15.5 Session Management

Due to budget, I am unable to protect any sessions and cookies from being exploited.

### 4.15.6 Data Validation

I’ve placed PHP’s strip slashes as a first line of defense before placing the values into a stored procedure as a second line of defense. However I haven’t implemented any defenses for Cross-Site Request Forgery for POST which is heavily used in HDJPMS due to limited resources.

### 4.15.7 Denial of Service

My system is vulnerable to Denial of Service.

### 4.15.8 Ajax

Ajax is a big part of my voting system however similar to session, there is no protection due to budget reasons however SQL Injection is prevented to a certain extent.

# 5 Design

## 5.1 Use Case Diagram



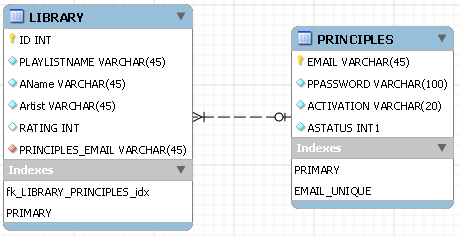
## 5.2 Context Data Flow Diagram



## 5.3 Entity Relationship Diagram



## 5.4 Class Diagram



## 5.5 Dataflow Diagram













## 5.6 Data Structured & Activity Diagrams

### 5.6.1 Register





### 5.6.2 Search





### 5.6.3 Activate account





### 5.6.4 Reset passcode





### 5.6.5 Reset password (forgot password)





### 5.6.6 Change password





### 5.6.7 Login





### 5.6.8 Manage playlist







### 5.6.9 Create playlist





### 5.6.10 Logout





### 5.6.11 Upload playlist



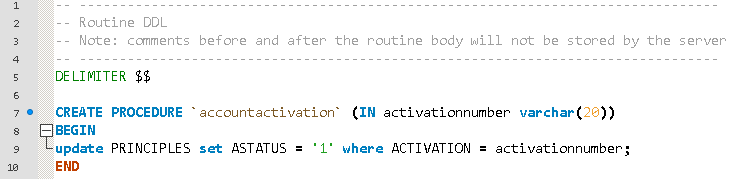


## 5.8 Conceptual Website Diagram

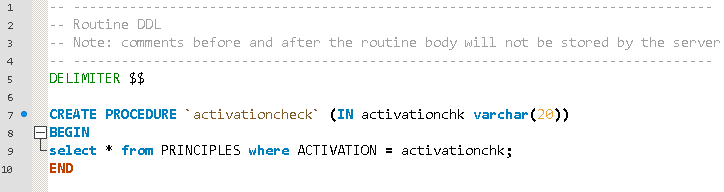


# 6 Database Stored Procedures

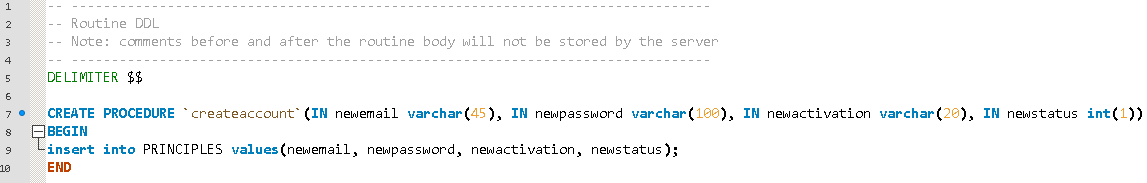
## 6.1 Account Activation



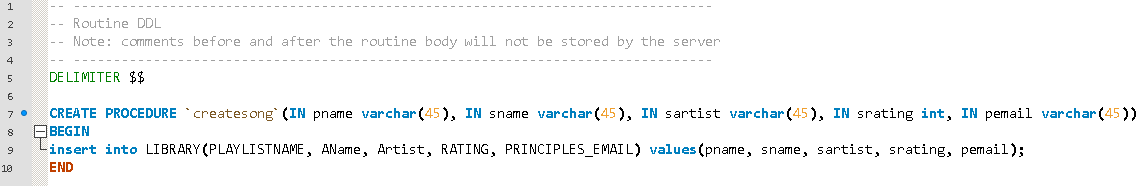
## 6.2 Activation Check



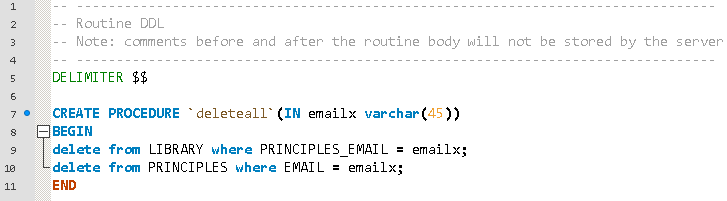
## 6.3 Create Account



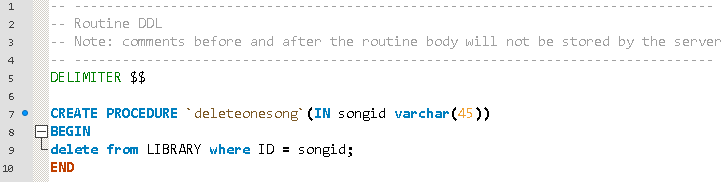
## 6.4 Create Song



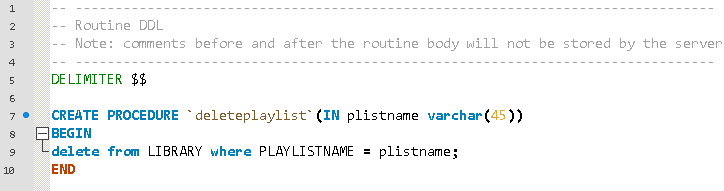
## 6.5 Delete All



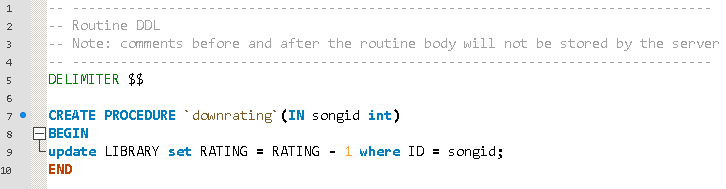
## 6.6 Delete One Song



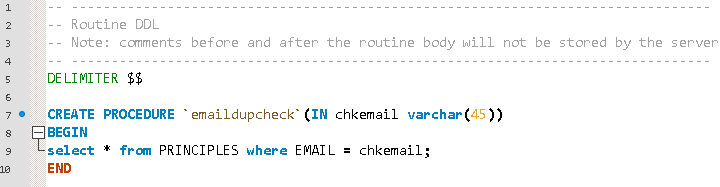
## 6.7 Delete Playlist



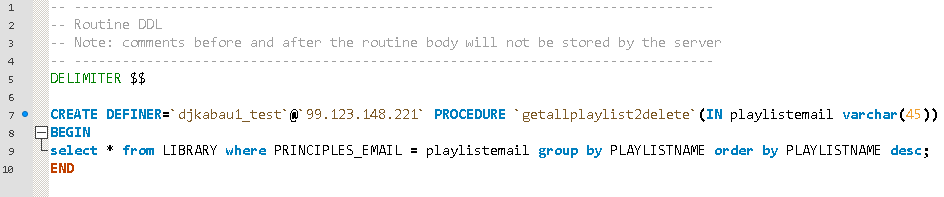
## 6.8 Down Rating



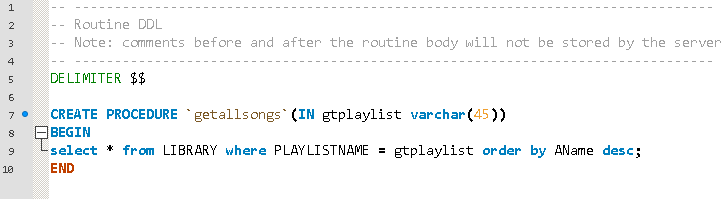
## 6.9 Email Duplicate Check



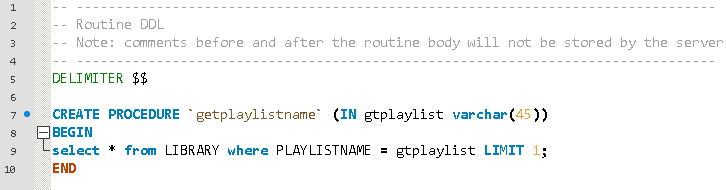
## 6.10 Get All Playlist To Delete



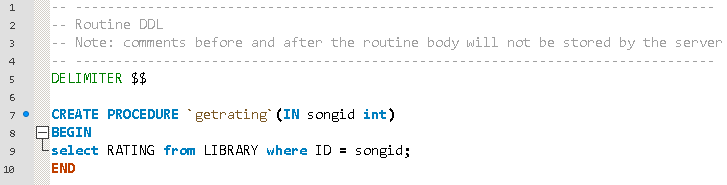
## 6.11 Get All Songs



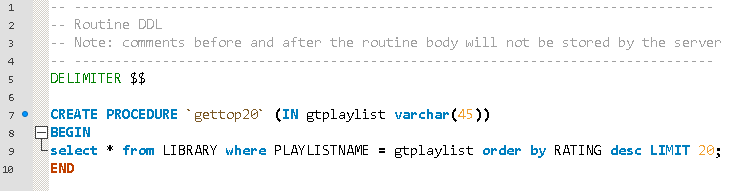
## 6.12 Get Playlist Name



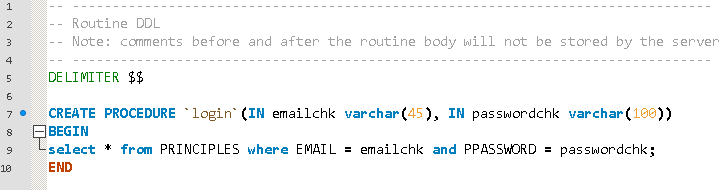
## 6.13 Get Rating



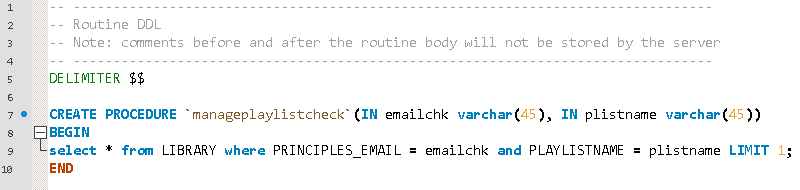
## 6.14 Get Top 20



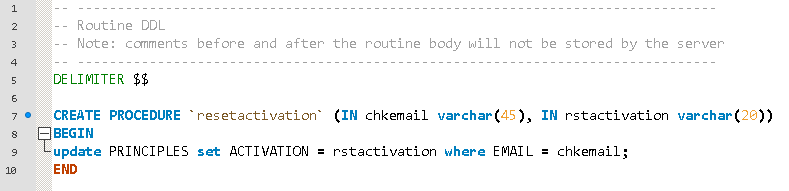
## 6.15 Login



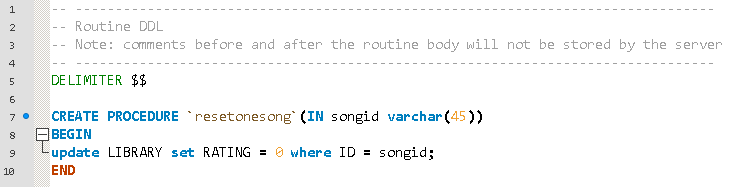
## 6.16 Manage Playlist Check



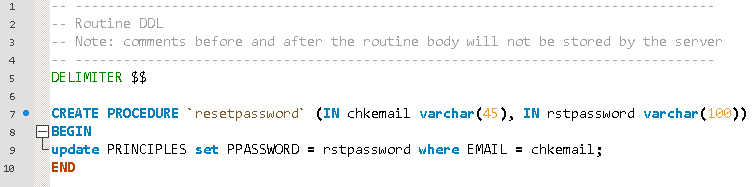
## 6.17 Reset Activation



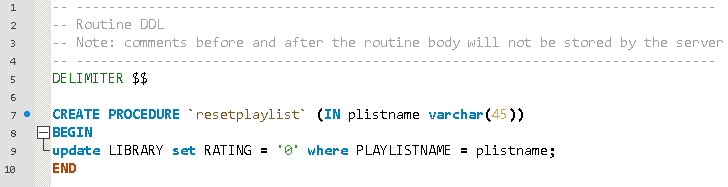
## 6.18 Reset One Song



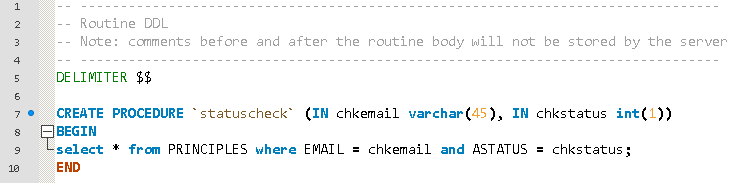
## 6.19 Reset Password (forgot password)



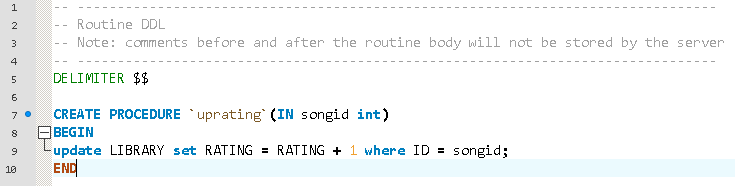
## 6.20 Reset Playlist



## 6.21 Status Check

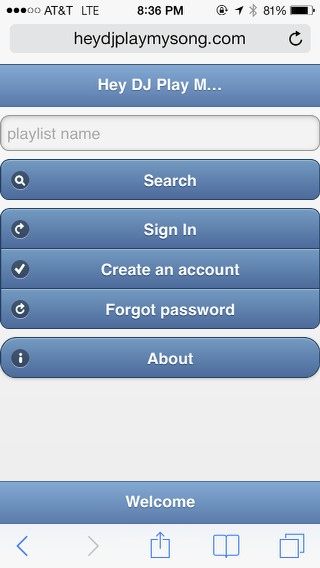


## 6.22 Up Rating

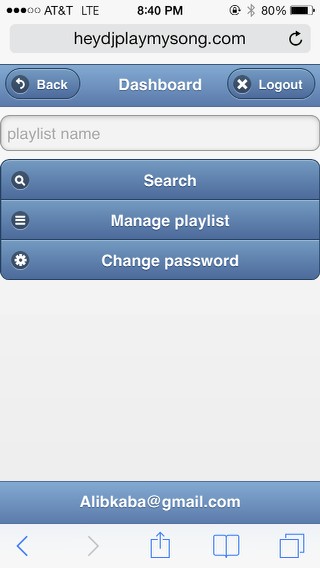


# 7 Screenshots

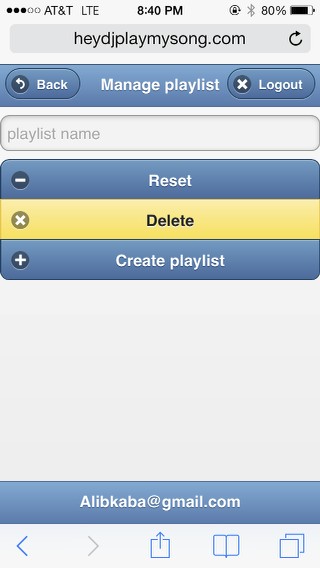
## 7.1 index.php



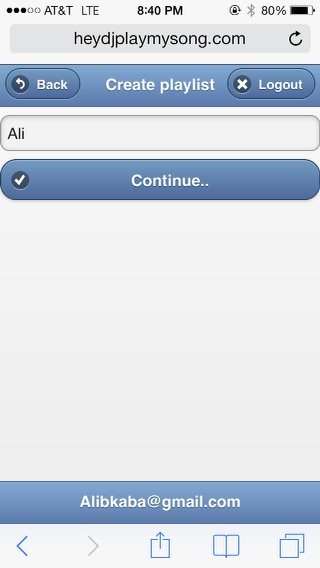
## 7.2 dashboard.php



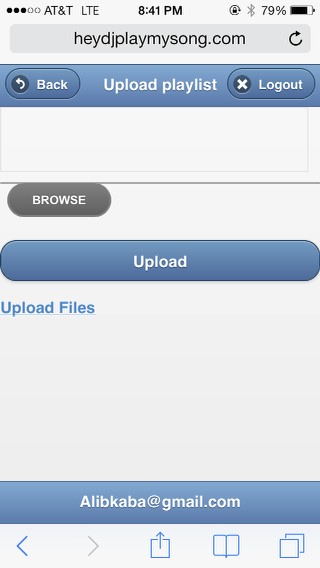
## 7.3 manageplaylist.php



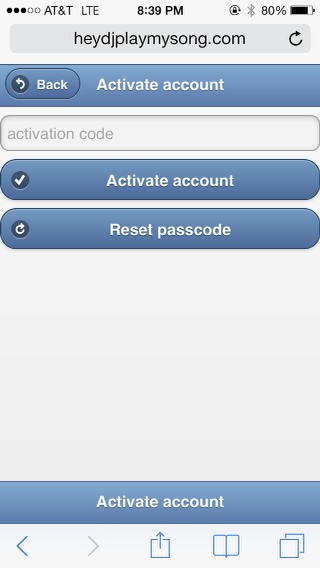
## 7.4 createplaylist.php



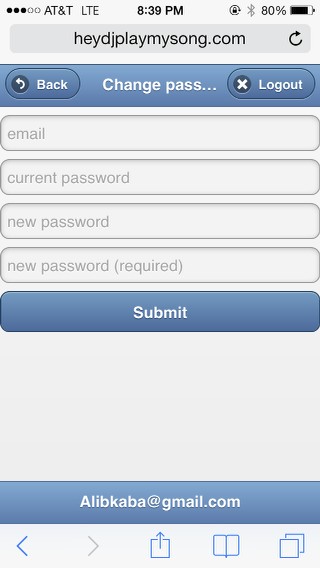
## 7.5 uploadplaylist.php



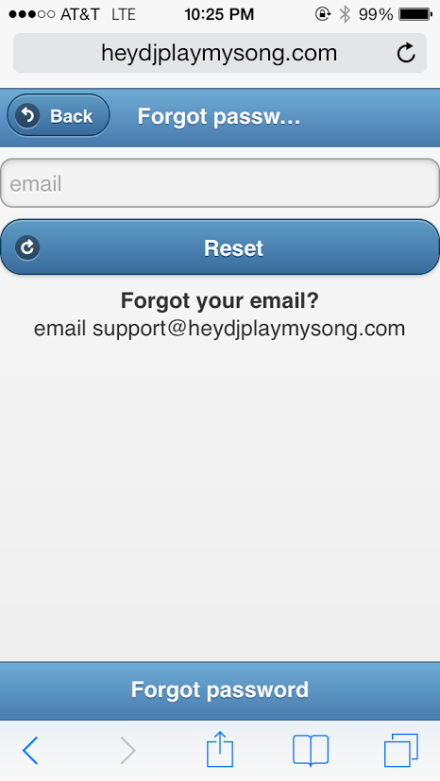
## 7.6 activate.php



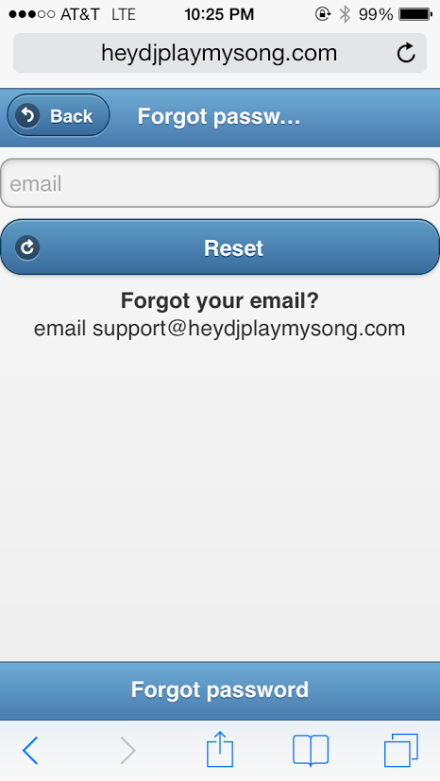
## 7.7 changepassword.php



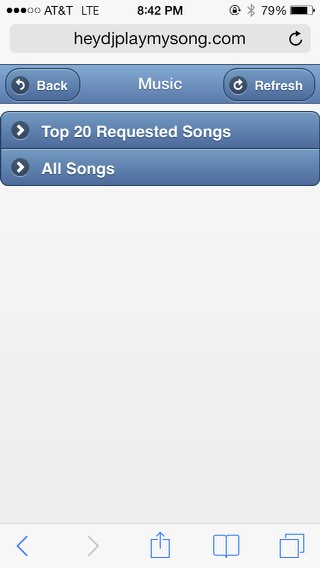
## 7.8 resetpassword.php



## 7.9 forgotpassword.php



## 7.10 music.php



# 8 English Instruction Manual

## 8.1 Using Hey DJ Play My Song

Dear Customer,

Thank you for using Hey DJ Play My Song system.

Hey DJ Play My Song (HDJPMS) is a 21st century solution of requesting songs. The system is an online tool that consists of playlist created by your host, DJs and anyone looking to give their company a great time.

## 8.2 Getting Started

Every playlist available on HDJPMS was imported using comma-separated values (CSV) document. A CSV is document format that can be made from a numerous applications.

### 8.2.1 Getting/creating the CSV document

A template is available for download on the swebsite **About** page and should be followed accordingly for the songs to be uploaded correctly.

Figure 1 below is how the document should look like.

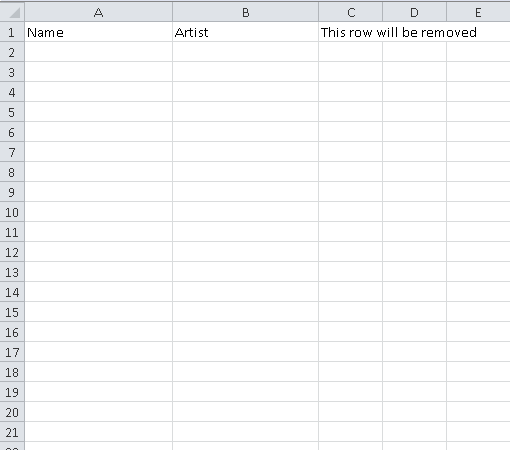


Figure 1

### 8.2.2 Getting the music

Populate the field with data as it is visually shown on Figure 2.



Figure 2

### 8.2.3 Formatting the CSV for HDJPMS system

Remove the first row and save the file as it is visually shown on Figure 3 & 4.

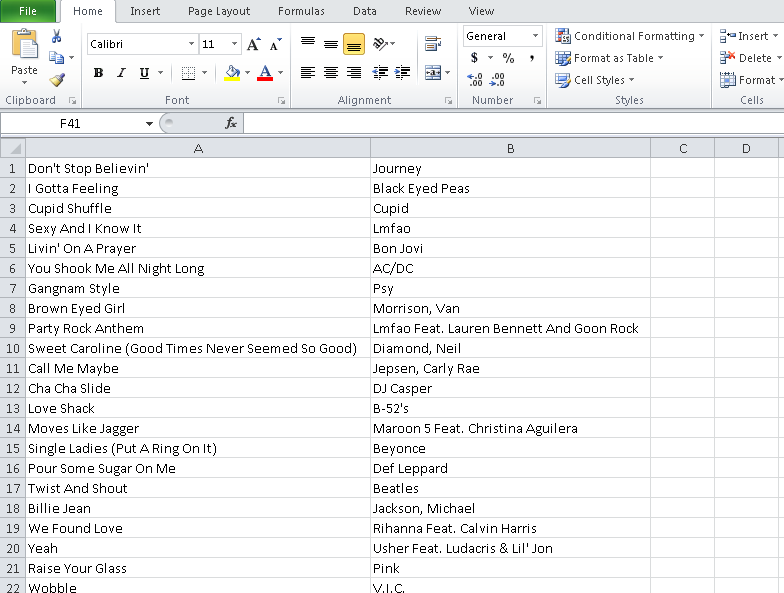


Figure 3

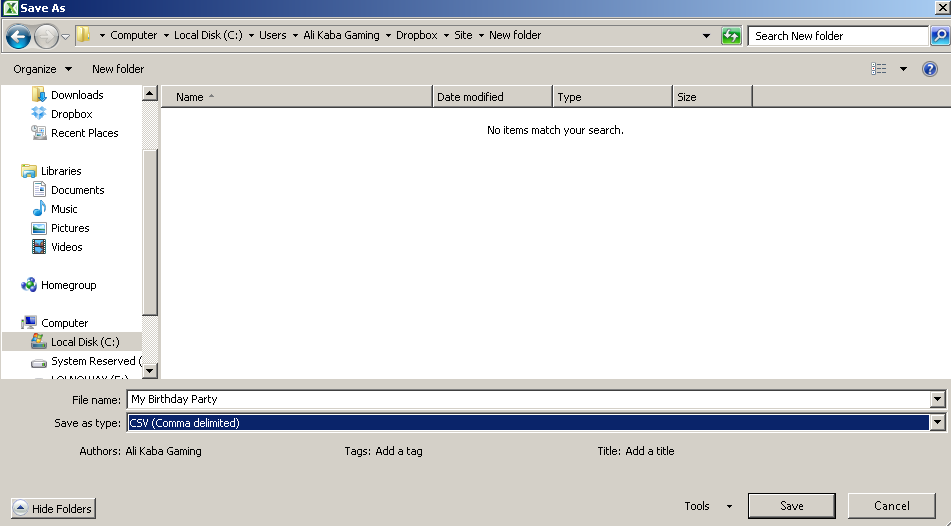


Figure 4

## 8.3 Setting up an account

An account is required to be able to create a playlist.

### 8.3.1 Visiting HeyDJPlayMySong.com

HDJPMS website as it is visually shown on Figure 5.

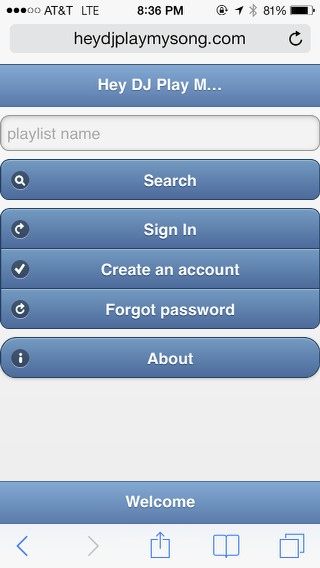


Figure 5

### 8.3.2 Creating an account

Click on **Create an account** and enter your email and password as it is visually shown on Figure 8. Password must be between 6 to 12 characters long.

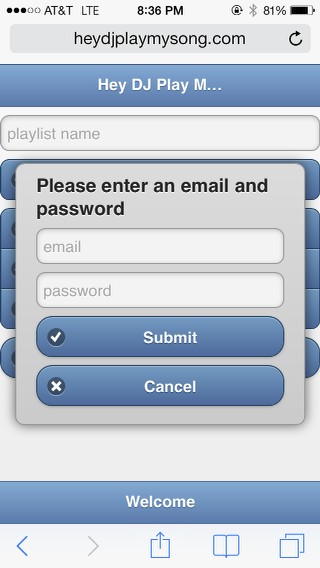


Figure 6

### 8.3.3 Verifying your account

After creating your account, you will be prompt a message to activate your account. Not activating your account will not permit you of using the manage playlist features which allows you to create a playlist as it is visually shown on Figure 7. To activate your account you either have to click on the link that was emailed to the email used to create the account or use the code listed on that same email and go to Activate account and enter it there as it is visually shown on Figure 8. Hit reset passcode to have a new one emailed again. Once your account is verified, you will access Manage playlist as it is visually shown on Figure 9.

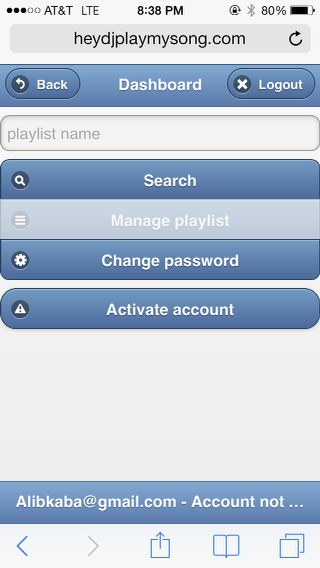


Figure 7

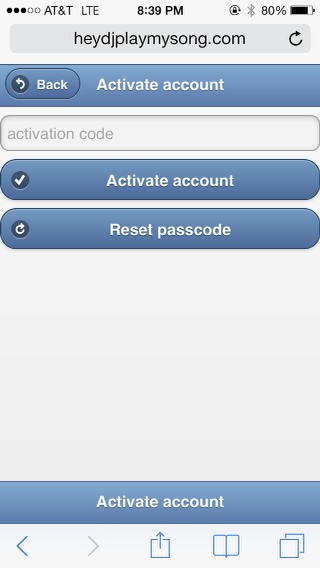


Figure 8

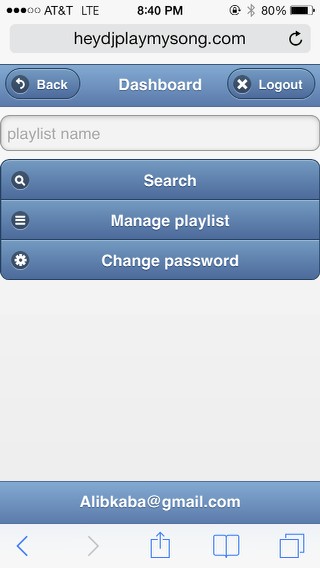


Figure 9

## 8.4 Dashboard

Here you can change your password, manage your playlist and search a playlist to vote on.

### 8.4.1 Change password

Requirements are the email, current password and typing your new password in both fields as it is visually shown on Figure 10.

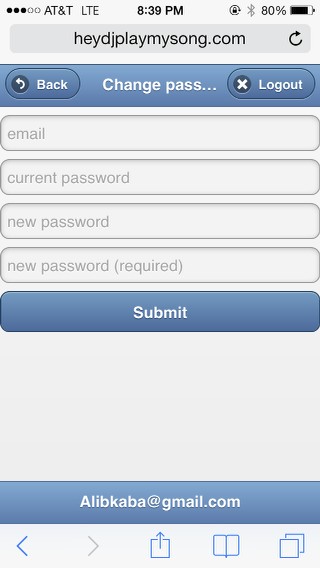


Figure 10

### 8.4.2 Manage playlist

HDJPMS values playlist and understands the pain of accidents. In order to prevent an accidental reset or delete, we require that you type in your playlist name. Figure 11 doesn’t have any playlist listed however skip to Figure 14 to see one playlist that is listed.

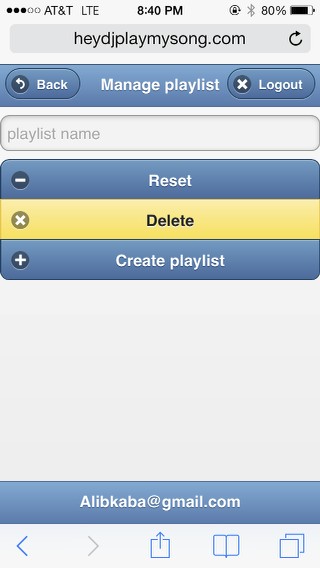


Figure 11

### 8.4.3 Create playlist

Once Create playlist is selected on Figure 11, you will be required to enter a playlist name as it is visually shown on Figure 12. If the playlist name is available, you will proceed to Figure 13 to finalize the upload process where you will click on browse, select your .CSV file and click upload where you will be re-directed back Figure 14.

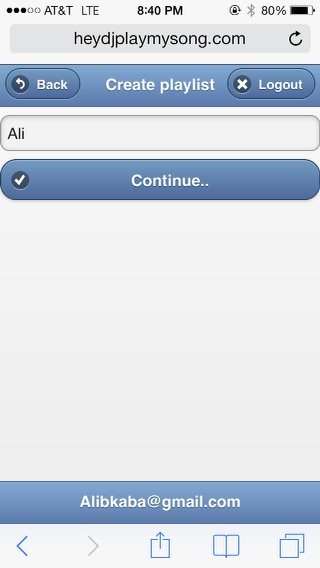


Figure 12

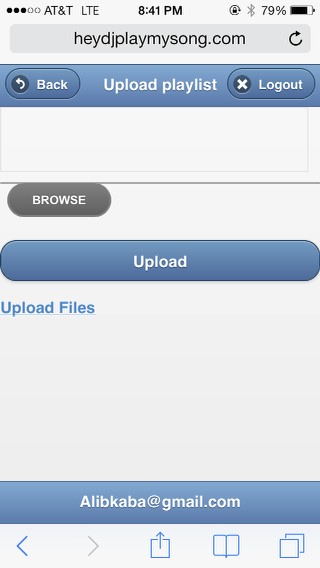


Figure 13

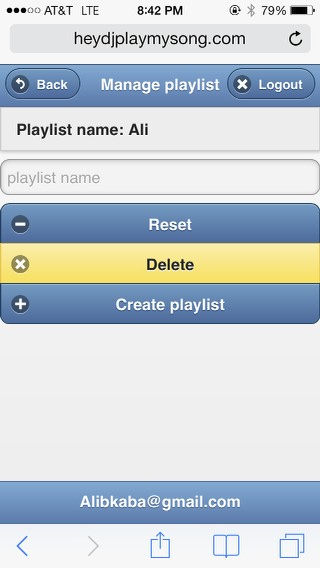


Figure 14

## 8.5 Search

Here you can perform a search from Figure 1 or Figure 15.

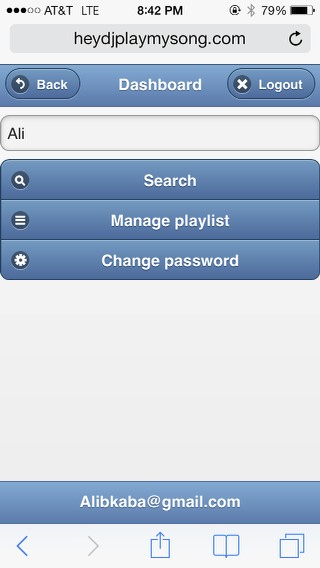


Figure 15

### 8.5.1 Music

The music page consists of Top Requested 20 Songs and All songs as it is visually shown on Figure 18.

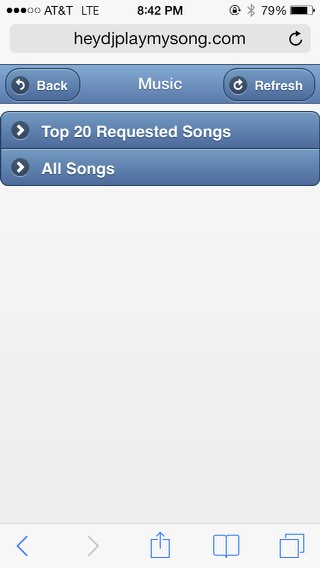


Figure 16

### 8.5.2 Voting

On Figure 17 and 18 you can vote for the song with the up or down arrow. If you are the owner of the playlist, you will be able to reset or delete individual songs as it is visually shown on Figure 19.

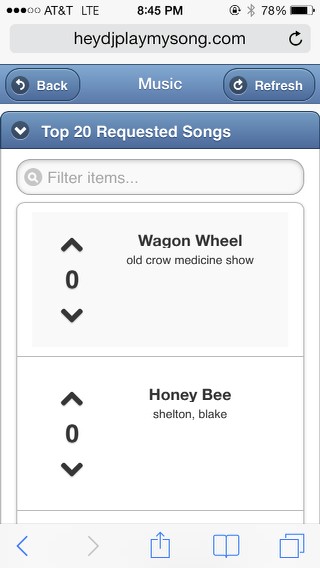


Figure 17

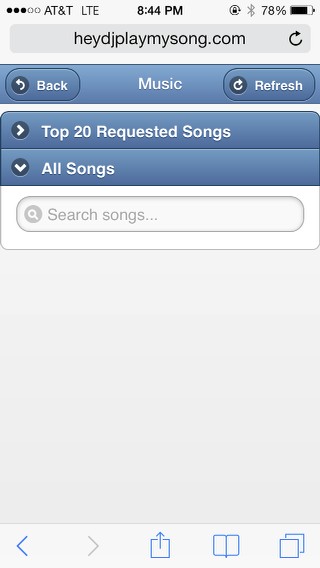


Figure 18

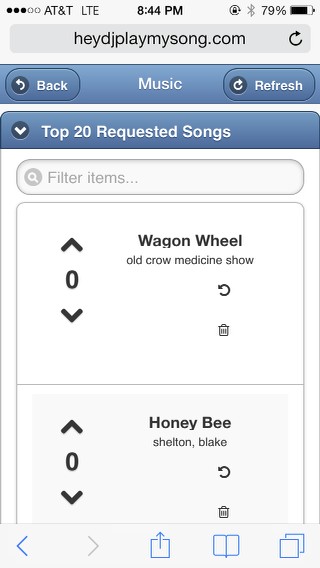


Figure 19

## 8.6 Homepage

One you are logged, the homepage will change from Figure 1 to Figure 20.

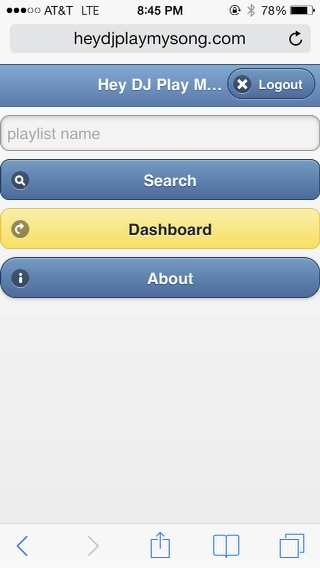


Figure 20

# 9.0 Enhancements

|  |  |
| --- | --- |
| Enhancement Number | 1 |
| Entered by | Ali Kaba |
| Status | New |
| Business Priority | 2 - Medium |
| Version Planned | Future Release |
| Description | Consolidate stored procedure which will improve MySQL database performance |
| Comments |  |
| Outcome | Improve system performance |

|  |  |
| --- | --- |
| Enhancement Number | 2 |
| Entered by | Ali Kaba |
| Status | New |
| Business Priority | 3 - Low |
| Version Planned | Future Release |
| Description | Incorporate social login such as Open-ID |
| Comments |  |
| Outcome | Increase traffic to system |

|  |  |
| --- | --- |
| Enhancement Number | 3 |
| Entered by | Ali Kaba |
| Status | New |
| Business Priority | 2 - Medium |
| Version Planned | Future Release |
| Description | Place a login attempt lock out |
| Comments | Account will lock out after 5 failed attempt |
| Outcome | Prevents brute force login |

|  |  |
| --- | --- |
| Enhancement Number | 4 |
| Entered by | Ali Kaba |
| Status | New |
| Business Priority | 1 - High |
| Version Planned | Future Release |
| Description | Revamp the upload playlist process |
| Comments |  |
| Outcome | Improve loading process and ease of use |

|  |  |
| --- | --- |
| Enhancement Number | 5 |
| Entered by | Ali Kaba |
| Status | New |
| Business Priority | 2 - Medium |
| Version Planned | Future Release |
| Description | Switch all sqli queries to PHP Data Object (PDO) |
| Comments |  |
| Outcome | Improve MySQL performance and website |

|  |  |
| --- | --- |
| Enhancement Number | 6 |
| Entered by | Ali Kaba |
| Status | Suspended |
| Business Priority | 1 - High |
| Version Planned | Future Release |
| Description | Place an Secured Socket Layer (SSL) |
| Comments | No budget |
| Outcome | Prevent session hijacking |

|  |  |
| --- | --- |
| Enhancement Number | 7 |
| Entered by | Ali Kaba |
| Status | Suspended |
| Business Priority | 2 - Medium |
| Version Planned | Future Release |
| Description | Crate a table to keep track of which SQL is sent to the database and by what user, the time it was sent and it if it was successful |
| Comments | No budget |
| Outcome | Prevent session hijacking |

|  |  |
| --- | --- |
| Enhancement Number | 8 |
| Entered by | Ali Kaba |
| Status | New |
| Business Priority | 2 - Medium |
| Version Planned | Future Release |
| Description | Pop-up error message when a user runs into issues in the application |
| Comments | This will required a different PHP to display specific errors and most likely require to wrap try and catch when things are being executed which will also be stored on a database table with the user (see enhancement Number 7) |
| Outcome | Prevent session hijacking |

|  |  |
| --- | --- |
| Enhancement Number | 9 |
| Entered by | Ali Kaba |
| Status | New |
| Business Priority | 1 - High |
| Version Planned | Future Release |
| Description | Activation check needs to compare with the userid, not just search for the activation number. Will be worked on along with enhancement #1 |
| Comments | N/A |
| Outcome | Prevent account from not getting activated and prevent activating random accounts |

|  |  |
| --- | --- |
| Enhancement Number | 10 |
| Entered by | Ali Kaba |
| Status | New |
| Business Priority | 2 - Medium |
| Version Planned | Future Release |
| Description | Check if the fields are empty before posting therefore nothing well be sent to the database |
| Comments | N/A |
| Outcome | Improve DB performance |

|  |  |
| --- | --- |
| Enhancement Number | 12 |
| Entered by | Ali Kaba |
| Status | New |
| Business Priority | 3 - Low |
| Version Planned | Future Release |
| Description | Increase the varchar length from 45 to a ~100 |
| Comments | N/A |
| Outcome | Song name and various fields won’t be cut off due to the limit |

|  |  |
| --- | --- |
| Enhancement Number | 13 |
| Entered by | Ali Kaba |
| Status | Closed |
| Business Priority | 2 - Medium |
| Version Planned | 1.1 |
| Description | Ability to reset and delete songs |
| Comments | N/A |
| Outcome | N/A |

# 10 Glossary

|  |  |
| --- | --- |
| Activation code | An auto-generated code used to activate a registered account |
| Comma-Separated Values (CSV) | A file format used to populate the music and later on uploaded into HDJPMS |
| Denial of Service (DoD) | An interruption in an authorized user's access to a computer network, typically one caused with malicious intent. |
| Disc Jockey (DJ) | The musician who is in charge of playing the music for his/her host/client and guest. This user can create the playlist or request the host/client for a type of playlist |
| Encryption | In [cryptography](http://en.wikipedia.org/wiki/Cryptography), encryption is the process of encoding messages (or information) in such a way that third parties cannot read it, but only authorized parties can (wikipedia.com) |
| Guests | Friends and family of the host |
| Host/Client | The person who is hosting an event and has guests |
| Open Web Application Security project (OWASP) | The Open Web Application Security Project (OWASP) is a worldwide not-for-profit charitable organization focused on improving the security of software. (owasp.org) |
| Playlist name | The name of the playlist used to upload the CSV file and also used to access the music page to vote on the songs loaded with that name |
| SQL Injection | SQL injection is a [code injection](http://en.wikipedia.org/wiki/Code_injection) technique, used to [attack](http://en.wikipedia.org/wiki/Attack_(computing)) data driven applications, in which malicious SQL statements are inserted into an entry field for execution. (Wikipedia) |
| Stripslashes | Returns a string with backslashes stripped off (PHP.net) |

# 11 Bibliography

## 11.1 Books

Meloni, Julie C. *Sams Teach Yourself PHP, MySQL and Apache: All in One*. Indianapolis, IN: Sams Pub.,

2012. Print.

Meloni, Julie C., and Matthew A. Telles. *PHP 6 Fast & Easy Web Development*. Australia: Course

Technology PTR, Cengage Learning, 2007. Print.

Murach, Joel, and Ray Harris. *Murach's PHP and MySQL: Training & Reference*. [Fresno, Calif.]: Mike

Murach & Associates, 2010. Print.

Ullman, Larry E. *PHP and MySQL for Dynamic Web Sites*. Berkeley, CA: Peachpit, 2012. Print.

## 11.2 Internet Sites

"Alias.io." *— How to Store Passwords Safely with PHP and MySQL*. N.p., n.d. Web. 19 Nov. 2013.

<http://alias.io/2010/01/store-passwords-safely-with-php-and-mysql/>.

"Developing PHP/MySQL Webapps." *Developing PHP/MySQL Webapps*. N.p., n.d. Web. 21 Nov. 2013.

<http://www.ntu.edu.sg/home/ehchua/programming/webprogramming/php2\_webapps.html>.

"Dynamic Voting System with JQuery, PHP and MySQL." *Programming and Designing Blog*. N.p., n.d.

Web. 01 Dec. 2013. <http://www.w3bees.com/2013/09/voting-system-with-jquery-php-and-mysql.html>.

"HTML5 or Flash Multiple File Upload JQuery Plugin Script." *Uploadify HTML5 or Flash Multiple File*

*Upload JQuery Plugin Script Comments*. N.p., n.d. Web. 05 Dec. 2013.

<http://www.uploadify.com/>.

"Ircmaxell's Blog." *Seven Ways To Screw Up BCrypt*. N.p., n.d. Web. 21 Nov. 2013.

<http://blog.ircmaxell.com/2012/12/seven-ways-to-screw-up-bcrypt.html>.

"MySQL :: The World's Most Popular Open Source Database." *MySQL :: The World's Most Popular*

*Open Source Database*. N.p., n.d. Web. 19 Nov. 2013. <http://www.mysql.com/>.

"MySQL Tutorial." *MySQL Tutorial*. N.p., n.d. Web. 19 Nov. 2013. <http://www.mysqltutorial.org/>.

"PDO Tutorial for MySQL Developers." *Hashphp.org*. N.p., n.d. Web. 21 Nov. 2013.

<http://wiki.hashphp.org/PDO\_Tutorial\_for\_MySQL\_Developers>.

"PHP Freaks." *PHP Freaks*. N.p., n.d. Web. 19 Nov. 2013. <http://forums.phpfreaks.com/>

"PHP: Hypertext Preprocessor." *PHP: Hypertext Preprocessor*. N.p., n.d. Web. 19 Nov. 2013.

<http://www.php.net/>.

"PHP Login Page Example." *PHP Login Page Example.* N.p., n.d. Web. 19 Nov. 2013.

<http://www.9lessons.info/2009/09/php-login-page-example.html>

"Top Questions." *Stack Overflow*. N.p., n.d. Web. 19 Nov. 2013. <http://stackoverflow.com/>

"ZURB University." *JQuery Joyride Plugin*. N.p., n.d. Web. 02 Dec. 2013.

<http://zurb.com/playground/jquery-joyride-feature-tour-plugin>.

# 12 Conclusion

What I have learned in the whole process is how complicated and amazing web sites are built. It isn’t just one language, rather multiple languages either built on top of each other or completely different.

It is an evolving innovation in both good and bad ways depend on how you look at it, people will use it for good things and others will exploit its unknown weakness. It is a fun realm if you accept both sides and move along and the best way to start on the web is to first determine the value of the data you are working with. If the data needs to be online, construct your web around it like a spider and protect it.