# ThePeoplesJukebox.com – A Hunter Collage Capstone Project by Megan Williams

## 1.0 Introduction

*OK, so, you're a bartender or perhaps a barista at a cool coffee lounge and you usually plug your iPhone into the public address system to play music.*

*You're the DJ...*

*Even though you have spent hours carefully selecting the play-list from your MP3 or MP4A collection, there's always customers who don't like what you're playing. What if you could let them select the songs from your catalog using their mobile phone or tablet or computer?*

*Now, You're the JUKEBOX!*

This the top level storyboard or "elevator pitch" for ThePeoplesJukebox.com

The purpose of this white paper is to specify the:

- Project Goals
- Technologies Used and
- The Development Strategy

I will speak to the design and technical decisions made and how they relate to the Business Strategy to build the project.

Through my course work at Hunter Collage studying Computer Science and my personal apprenticeship with the former Chief Technology Officer at Interpublic Group, I've learned that through each iterative AGILE SCRUM the three things that must be asked is:
1) What is the business strategy
2) What is the Information Architecture and
3) What is the Data Architecture

As the **Information Architecture** informs **User Experience** as it relates to the **Business Strategy** and the **Data Architecture** informs the **Database Model** and the underlying **Data Structures**, and supporting business rules and logic as it relates to the **Business Strategy**.

This white paper will also speak to the current status of the project, what's missing and where the project is going after I graduate Hunter Collage.

As such; ThePeoplesJukebox.com was not designed to only be a presentation for a class but it's also a PRODUCT.

The Business Strategy came to me because I've been a bartender, waitress and manager in the service industry for over 12 years. I am the perfect customer to use this product and I already have been "Alpha" testing it in my bar on real customers using real phones on busy nights.

I stand behind every decision that I've made in the design of this product and it's implementation.

## 2.0 Project Goals

Here are the high level Use Cases.

One of the major project goals for ThePeoplesJukebox.com is that there must be a **Web-App** that:
1) Attracts potential users
2) informs them about the product
3) allows them to use the product right away and
4) works on mobile phone, tablet or computer without special installation.
   It must be a **mobile first** implementation and
5) has a "back-end" administration facility so that the administrators can manage the site, the site users and site content.

Most users will be people visiting the establishment where the Jukebox is playing and they are much more likely to use the product if
1) it's free
2) they don't have to install anything – they sign-up and start selecting songs on a FREE Jukebox.

Jukebox owners (or as I call them "Jukebox's") are the "stars" in our jukebox ecosystem they are motivated to download the app, install it and activate all of their songs into their on-line catalog for fame and fortune – and happy customers.

**Therefore the Android App is "Jukebox" only** – users will sign-up via the website and manage their profile information, user-names and passwords and profile photo's via the **Web-App**.

Also, since most users will be selecting songs to play using their mobile phone,
**the Web-App must employ a "mobile first" (called Responsive Web design) strategy.**

A Mobile Phone or tablet can be a Jukebox using the **browser only**. (Apple Corp. has unfortunately made this requirement difficult for iPhone **by breaking auto_play** but on every other device it works properly as a jukebox using only a browser and Web-App).

Songs usually reside on the mobile device however ThePeoplesJukebox.com allows for **BOTH LOCAL and UPLOADED** media files. Even in the same catalog.
Future versions of the product will support Spotify as well.

The Server must provide a **public REST API** that can be used by Android or iOS devices (made by third party vendors or myself ). The public API must support all functions related to the Jukebox; "login", "play_next", "currently_playing" ,"catalog", "add_queue", "search_profile", "search_song", "upload_catalog"

Geo-location will be **NOT be used to restrict users** but rather to direct them to Jukebox's near them. (future versions will give the jukebox the ability to reject unwanted users).

Security, User Validation and Robot detection are critical to a functioning product.

## 3.0 Technologies Used

The product currently has several components on **2 platforms**:

1) A Server hosted on an Amazon Elastic Compute Cloud (EC2) running Linux
The Server serves Web-pages to humans and JSON via a REST API to programs.
(I would suggest that the browser is ALSO a platform. Since HTML and JavaScript are SERVED by the server but EXECUTE in the browser. But, this is a matter of semantics. )

2) an Android Client App that runs on Phones and Tablets that run the Android operating system.

### 3.1 Server Technologies

PHP 7 was chosen as the development language. This is because:
1) at the time of this writing, over 80% of the Internet is written in PHP including Facebook and Wordpress
2) PHP basically has everything needed to solve any web problem built-in – not requiring the installation of many packages to do standard things
3) PHP is the best view-template language because it IS a view-template language
4) PHP 7 performance benchmarks are impressive.

#### 3.1.1 Database

MySql is the database – it's fast reliable stable and open source.

The Object-Relational Mapping (ORM) technology being used is
**EntityObjects.com's PHP Model Builder –** it's a simple and easy ORM solution that uses a **database first** strategy where you FIRST design a normalized Data Model, then have the ORM generate classes based on that model (reading the database meta-data). This is how I've been taught to develop applications. Information architecture, data architecture as it relates to the business strategy, ALWAYS first and at the start of every scrum.

This ORM technology writes 80% of your Model for you and I think that is a good thing.

#### 3.1.2 Web-App (Front end)

**Bootsrap** is an open source toolkit for developing with HTML, CSS, and JavaScript.
Bootstrap framework provides out of the box most of the design elements needed to make clean professional Web-Apps using Jquery. Technologies such as **React and Vue are excellent** but if you're not a UX Engineer with at least 5 years experience in UX design and implementation your Web-App will still look like a high-school science project. Bootstrap only requires the developer to define the various div classes and you get professional design elements for free.

**Perhaps, most importantly, bootstrap provides a mobile first strategy right out of the box.**

### 3.2 Client Technologies
The client "jukebox" app was developed using the Android Studio IDE using **Java 8**
because Android devices currently can only run up to Java 8

## 4.0 Development Strategy and Implementation

### 4.1 The Server

ThePeoplesJukebox.com is built on a LAMP stack (Linux Apache MySql and PHP) implementing a MVC (Model View Controller) architectural pattern.

#### 4.1.1 Model

The Data Model or ER Diagram for the database is **here: ER Diagram**
Data is normalized to 3'rd normal form.

The Model objects are first generated using EntityObjects PHP Model Builder.
Which reads the Mysql database meta-data and then creates the Data Access Object (DAO), value objects and the models for each table. I then sub-class any table I choose to write an AD-Hoc Query for.

The Model Objects generated and sub-classed are **here: models**
The Model Object Diagram is located **here: Object Diagram**

#### 4.1.2 View

Views do not have any database references or controlling functions they are simply HTML templates, implemented as PHP functions.
Views are organized by page/site-section in the includes section.
Please review **here: views**
Views are split into functions that are called by controllers or other views
the following are the view functions for the home page:
global header ==> head.php
global navigation==> nav.php
content area ==> about.php
content area ==> why.php
content area ==> faq.php
global footer ==> foot.php

Where appropriate views use JavaScript and Jquery functions to call API calls so that page doesn't have to go back to server an example is the Jukebox Profile page which uses JavaScript/Jquery and API to search songs and play songs in a completely **decoupled** way

#### 4.1.3 Controller

Controllers are the entry points to Web functions and they instantiate Model Objects and call view functions.

An example controller can be found here: /admin/users.php – it's an admin function to manage users.

#### 4.1.4 REST API

REST API functions are located here: REST API

### 4.1.5 /etc

The Server /include/etc directory contains general purpose functions and objects and third party implementations that are useful for development.

**session.php** has functions for session management.
**email/email.php** is a class that sends email
**email/smtp.php** is used by the email class
**config.php** has configuration functions
**notify.php** sends notifications to system administrators
**json.php** has JSON functions
**sql.php** has SQL helper functions
**random.php** has function to generate random string
**PHPMailer** is a popular open source SMTP implementation
**captcha** is a popular open source CAPTCHA implementation used for robot detection.

### 4.1.6 Other Cool Functions

The Server also serves the Jukebox profile photos – so, the Client App (be it Android, iOS, or the Web-App) does not have to keep jukebox profile photos locally. This is because the profile photos are NOT just "links" to an image in the web document root (in fact the photos are NOT stored in document root) photos are served by a PHP function called photoviewer.php

As mentioned before, Music files (be they MP3 or MP4A files) can reside LOCALLY in the client app OR be uploaded to the server.

Once again the media files are NOT stored in the web server document root and as such CANNOT be stolen but must be rendered via our server to the client app via a function called mp3player.php
(By the way,mp3player will serve mp4a or any media files in the same manor)

## 4.2 Android Jukebox

The Android Jukebox is written in **Java 8** and was developed using **Android Studio**.
ThePeoplesJukebox.com was built using **APK 19 (kitKat)** so that 95% of all existing Android devices can use it.

### 4.2.1 User Interface (UI)

The UI has two parts:
1) a resource directory that defines the layout and string constants and menu items **located HERE**
2) the Java source file located **here: JavaHome.java**

### 4.2.2 com.thepeoplesjukebox packages

**common** – These classes are include things such as DixieHash which is an enhanced HashTable class.
**Config.java** – is a hash table that stores itself in the Android SQLite database
Rest.java – implements the REST protocol methods for GET and POST
**Api.java** – is a Wrapper class for the servers API
**model** – this package contains the structured data returned by the API

## 5.0 What's Missing

1) editing the user profile is not yet complete – this does not prevent users who select music on Jukebox's from signing up and using a jukebox. However, I currently have to make a regular user a jukebox user using our site administration tools.

2) Geo-Location is not complete – the plan is to use Google Maps to provide directions giving your longitude and latitude – Google even has examples in PHP
Provision has already bee made in the code so I know where to implement it and Google also provided the query to use to find a Jukebox nearest you.

3) I need to develop an IOS Jukebox App

4) find a Jukebox is not complete – the API function is there but the page with search form (which will be very similar to search songs that is working)

## 6.0 Conclusion

ThePeoplesJukebox.com is currently working as a Pre-Beta implementation.

For the Capstone which started in September of this year, I've developed over 40,000 lines of code and implemented:

1) A Web_App
2) a REST API
3) a MySql Database and Data Model and
4) an Android Jukebox App.

It is built on a solid platform using tried true technologies and I will continue to work on it after I graduate. I estimate sometime late January 2019 , I will launch the BETA 1.0 release of the product.

Currently, I have been "Alpha" testing it in my bar on real customers using real phones on busy nights with excellent results as well as great user feedback.