Software Requirements Specification

for

UDj

Version 1.0 approved

The Pokey Little Puppies

9/9/2013

Prepared by:

|  |  |  |
| --- | --- | --- |
| **Michael Rothberg** | **Ryan Paul** | **Hyejin Kang** |
| **Matteo Zancanella** | **Kristin McKenzie** | **Matthew Brock** |
| **Rainer Ledesma** | **Horacio Delgado** | **Raul Parras** |

Table of Contents

[Introduction 3](#_Toc366516051)

[Purpose 3](#_Toc366516052)

[Intended Audience and Reading Suggestions 3](#_Toc366516053)

[Project Scope 3](#_Toc366516054)

[References 3](#_Toc366516055)

[Overall Description 4](#_Toc366516056)

[Product Perspective 4](#_Toc366516057)

[Product Features 4](#_Toc366516058)

[User Classes and Characteristics 4](#_Toc366516059)

[Operating Environment 5](#_Toc366516060)

[Design and Implementation Constraints 5](#_Toc366516061)

[User Documentation 5](#_Toc366516062)

[Assumptions and Dependencies 5](#_Toc366516063)

[System Features 6](#_Toc366516064)

[Main Menu Screen 6](#_Toc366516065)

[Session Menu Screen 7](#_Toc366516066)

[Create Session Menu 8](#_Toc366516067)

[8](#_Toc366516068)

[List of Songs (Queue) 8](#_Toc366516069)

[Add Song Menu 9](#_Toc366516070)

[External Interface Requirements 10](#_Toc366516071)

[User Interfaces 10](#_Toc366516072)

[Hardware Interfaces 10](#_Toc366516073)

[Software Interfaces 10](#_Toc366516074)

[Communications Interfaces 10](#_Toc366516075)

[Other Nonfunctional Requirements 12](#_Toc366516076)

[Performance Requirements 12](#_Toc366516077)

[Safety Requirements 12](#_Toc366516078)

[Security Requirements 12](#_Toc366516079)

[Software Quality Attributes 12](#_Toc366516080)

[Business Rules 12](#_Toc366516081)

[Key Milestones 13](#_Toc366516082)

[Restrictions, Limitations, and Constraints 14](#_Toc366516083)

# Introduction

## Purpose

The purpose of this document is to describe the functionalities of UDj including functional and nonfunctional features. The Software Requirements Specification or SRS is designed to offer a view of the final product which includes possible User Interface (UI), Tools that will be used to develop the project, dependencies, and constraints.

## Intended Audience and Reading Suggestions

The document is targeted to those individuals participating and monitoring the UDj project. The document starts with a brief overview, a big picture, of the entire project. After a brief introduction the SRS will be highly technical. Users that are interested in a brief product overview should limit reading to section 2.x.

More technical users are encouraged to read the entire document, specifically section 3 and 4. Section 3 covers System Features which goes into the project details, while section 4 focuses to Interface Requirements including user interface, and software platforms

## Project Scope

UDj is a shared music playlist management web application. A single host can create a session on the UDj website. This host has a computer connected to some sort of audio system. Each session is unique, and contains a queue of YouTube videos (music) to be played (initially empty). Anonymous guest users are able to connect to the session and add videos to the queue. The host will have permission to delete, skip, and reorganize the queue. Administrators who enter the password chosen by the host will also have these permissions.

Potential uses of the application target social gathering activities where media playing could be selected by its participants.

## References

YouTube API: <https://developers.google.com/youtube/>

NodeJs: <http://nodejs.org/>

# Overall Description

## Product Perspective

This application is intended to improve guests’ experience at social gatherings by allowing personalization and sharing of live audio playlists. Every participant should be able to submit requests for preferred YouTube videos to be streamed from their mobile devices. This could be relevant in both recreational and professional settings and can be extended virtually to any other scenario, for example: restaurants, weddings, birthday parties and business networking events. To our knowledge, there is no complete web-based service that allows mobile users to quickly and easily add to a dynamic playlist.

## Product Features

Media streaming at social gatherings often comes from static, pre-selected playlists biased on the playlist manager’s preferences. Guests have little to no choice when selecting audio/media files to be streamed. Sometimes the overall quality of an event can be diminished by a poor or inappropriate sound environment. UDj aims to solve such problems by implementing the following core functionality:

* Dynamic, shared playlist
* Host control over playlist
* Searching of YouTube videos on the UDj web site, and ability to add those results to the queue
* Multi-platform session that allows guests to add audio/media files to the playlist
* Easily join a current session by scanning a QR Code
* Host can add IP addresses to a ban list to remove trouble-causing users
* Granting user administrator privileges using a password
* Host addition of filters that will prevent videos from being added that contain specific words
* Users can “upvote” a queued song, and if majority upvote the song, it will be queued to play next

Further features will be implemented if main project deadlines are met, among them:

* Playlist history for registered users
* Playlist order/priorities based on track rankings.
* Track suggestions based on guests submissions.
* Adding music from host’s personal computer
* Adding music from services other than YouTube, i.e. Grooveshark

## User Classes and Characteristics

This project targets as possible users anyone with a device capable of accessing internet. Users would fall into three main categories; hosts, managers and guests. The host is the singular administrator who streams the audio files and has full access to all features and control capabilities of the session created. They are the user connected to the speakers. Administrators are users who have entered a password created by the host, and are granted control capabilities of the specified playlist. Guest users can only have access to basic capabilities of the application, ie the viewing of and addition to the queue.

Further implementation would allow user registration to be able to setup a predefined playlist for their account allowing them to log in into their account and starting to play music from its predefined playlist and also keep track of playlist history and other basic information. Unregistered users would still have the same privileges as before allowing them to access the app and be able to add music to the queue without an account.

## Operating Environment

The application will initially run on a small VPS (Virtual Private Server). The VPS will be running a distribution of Linux i.e. Ubuntu. The front-end will be designed using Twitter Bootstrap, which will allow different layouts for PC, phones, and tablets. The web server will be written using Nodejs. MySQL will be used as a database back-end.

## Design and Implementation Constraints

* Only one session can be created per IP address
* Connection to YouTube is necessary
* The session will be automatically terminated after 30 minutes of an empty queue to kill idle/unused sessions
* Max 100 songs in the queue
* The number of sessions that can be hosted at a time will be limited by the hardware, however the hardware can easily upgraded

## User Documentation

The application would provide users with a help section that contains simple tutorials on basic software functionalities and settings. A brief text description will be placed next to inputs to portray to users how to use the application. A separate help/about web page will display more detailed information about the application.

## Assumptions and Dependencies

UDj heavily relies on YouTube Services to stream audio/media files selected on the playlists.

In order for the application to work YouTube must be up and running. The UDj frontend website will be built using bootstrap and its libraries. The back-end will use Nodejs, so its installation is required. A MySQL installation is also necessary for the database back-end.

The functionalities of UDj are divided into core and additional features. Core features are essentials because they have the highest priority. Core features will be delivered with the final package of the product.

Due to time constraints we cannot guarantee that additional features will be delivered with the final product package. However, the team will try to integrate those features after product release.

# System Features

## Main Menu Screen

The menu screen will allow the user to either create session or to join a session.

* The create session button (Create Session) allows the user to go to the log in menu.
  + The log in menu prompts the user to enter his/her account information such as username and password and will enable the user to access its account. The join session button (Join Session) allows the user to go to the session list.
* The join session button (Join Session) allows the user to go to the session list.
  + The session list will allow the user to join a specific session by entering the session code.
  + Users will also be able to join a session by scanning a QR code that will be shown in each session, and will contain a direct link to the session



## 

## Session Menu Screen

The session menu allows the user to interact with the interface depending on the type of user (host, administrator, or guest).

The host will have all control features, he/she can choose from the following options:

* Add song
* Skip song
* Delete song
* Ban user
* Close session
* List of songs
* Reorder the playlist

Administrators will have the control features only of the playlist:

* Add song
* Skip song
* Delete song
* List of songs
* Reorder the playlist

The User will only be able to search for and add music to the queue:

* Add song



## Create Session Menu

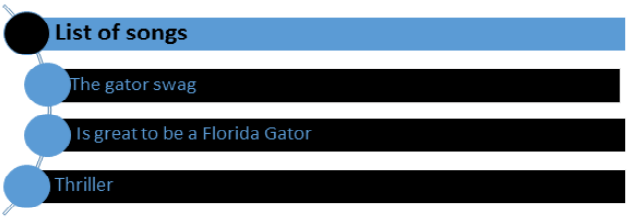
The create session menu will allow the host to name the session and create it. After creating the session a session code and a QR code will be giving to allow access to the session.

* Session name
* Create it

## 

## List of Songs (Queue)

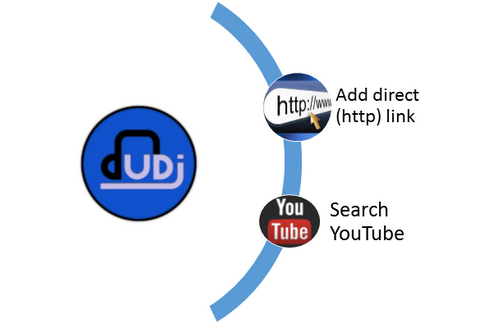
The queue screen will display all the songs that are in the queue. It will also display the current song and location in the track.



## Add Song Menu

This screen will allow the user to search YouTube directly on the UDj website, and add results from the search to the queue. To prevent abuse, only videos with a category of Music will be returned. This will be accomplished through the use of YouTube’s API. Users will also be able to directly paste in YouTube links in case they want to use the YouTube site to search instead of UDj.

* Add direct (http) link
* Search YouTube



# External Interface Requirements

## User Interfaces

Users will be ushered to create, join, and manage a session from the main page of our application. Created rooms will grant the session leader with a QR Code that will grant access to the specific session. Users within a session will see the queue, selected songs, and other users within the session, and the search area for music selections.

## Hardware Interfaces

UDj is a web application that will be used through an internet browser. Any device with internet access that can run an internet browser (smart phone, tablet, personal computer, etc.) can use and/or participate in the functions of UDj.

* Smartphones will use the touchscreen interface and any gestures standard to using the mobile internet browser.
* Tablets will use the touchscreen interface (with or without stylus) and any gestures standard to using the mobile internet browser.
* Personal computers will use any and all input/output devices necessary to browse the internet including, but not limited to, a monitor, keyboard, and mouse.

## Software Interfaces

The back-end of UDj will be running a Nodejs server that contains a series of Javascript files to control parts of the application. The back-end functionality will be exposed through an API that the front-end bootstrap site can access using JQuery getJson requests. UDj will also interact with YouTube’s API through getJson requests. The entire application will be running on a Linux (Ubuntu) VPS (Virtual Private Server).

## Communications Interfaces

The application communicates with a database server that contains the created room and the songs that were requested in each room. It also utilizes JQuery getJson requests for API calls. As UDj is a web application, the front-end will be written in HTML5 using bootstrap.

# Other Nonfunctional Requirements

## Performance Requirements

Performance of the web application shall be directly correlated to strength of the network used and the number of users adding to the queue. Depending on average number of users and on a good network connection, adding YouTube link to the queue should not take longer than few minutes to update it to the server. The web application should be able to be accessed using any device that can surf the internet and depending on the corresponding screen size, the web application should be able to adjust to the screen size of the device.

## Safety Requirements

Using the web application should not cause any hardware or software damage to the device. It should not destroy any data in your device nor access any data from your device for malicious usage. In addition, the user should follow the safety requirement of using the device.

## Security Requirements

The web application does not require the host, administrator, or users to create an account; therefore it does not require any identification of the user to authenticate. In order to get in the room, the user is required to either scan a QR code or input the identification of the room; furthermore, the host has the option to require a password to finalize entering the room for users. Therefore, a malicious user will have to access both the identification of the room and the password.

## Software Quality Attributes

The application shall implement sorting algorithms so that each user's selected songs will be added with equal priority to the songs that already exist in the queue. The session manager will be able to skip, remove, and manage the queue in any way he or she chooses. Users may also be removed and/or banned from the session at the session manager's discretion. In addition, the application should run on all browsers.

## Business Rules

The user, who is not the manager, has limited access of the application. The user cannot change the queue order or delete any songs he or she has not requested to the queue.

# Key Milestones

|  |  |  |
| --- | --- | --- |
| **#** | **Milestone** | **Target Completion Date** |
| 1. | Software Requirements | September 7 |
| 2. | Design/Scrum Sprint 1 | September 11 |
| 3. | Scrum Sprint 2 | October 2 |
| 4. | Scrum Sprint 3 | October 23 |
| 5. | Scrum Sprint 4 | November 13 |
| 6. | Finished | November 17 |
| 7. | Try adding additional Features | November 20 |

# Restrictions, Limitations, and Constraints

The application relies on several applications. First YouTube site must be up, Bootstrap must work, and browsers must support JavaScript.