SCC31901 SYSTEM HANDOVER AND COMPLETION REPORT

Squashie’s Jukebox

Music Control App

Joshua Incollingo - 19049437

Themis Giaras - 19034124

Dharmjit Kailay - 17879659

# Executive Summary

The project required by Squashlands Gym was to create a jukebox-like app/service to allow gym patrons to queue up music to play as well as request for songs that are not on the current playlist. Alongside this, the gym management will also aim to control and manage the playlist of music and use statistics to continually update their playlist.

This document discusses how the handover will operate and what the client will need to do in order to get the system up and running when all the data is transferred to their desktop

Known bugs show issues that were faced in this project and it will tie in to any potential further improvements and any outstanding development work.

Throughout the implementation, there have been some variation with the functional use cases, screen designs and data designs from either the client and academic supervisor, which have been highlighted in this report and whether they have been implemented or not.

# Table of Contents

[**Executive Summary**](#_paj353druv4p)2

[**Table of Contents**](#_rkzejzqilv3f)3

[**Introduction**](#_9fs10fyef4ht)3

[**Handover and Running**](#_geyak8hvca0)4

[**Implemented DB Schema**](#_i7qbqyljjtw)5

[**Test Results**](#_otf7wkx3vhji)6

[Use Case - Register into Squashie’s Webpage](#_udd6kz4uly07) 6

[Use Case - Search for Song in the Queue](#_oxkmottimyip) 6

[Use Case - Add Song to the Queue](#_spu8ujp7pow0) 7

[Use Case - Remove Song from Queue](#_zbk8e39l503j) 8

[Use Case - Entering the IP Address on the App](#_f21cqtinhxjd) 9

[Use Case - View Statistics in Staff Webpage](#_1aoofnycxxlh) 10

[Use Case - Requesting Song Not On Playlist](#_b06xno3rditf) 11

[**Variations Report**](#_1jsxqfmzk6u6)12

[**Known Bugs**](#_e3ya4erxeyyr)13

[**Outstanding Development Work or Further Improvements**](#_2qxhy1kdjvno)13

[**Conclusion**](#_o7txxqq49dp2)14

Introduction

In the final stages of this project, it’s time to hand over the final system to the client. Throughout this report will include:

* The process of how the system will work when first run on the client’s hardware
* How all the processes are connected throughout the system with the use of the database schema
* Test results of the use cases that were explored in the system analysis and design report
* Any variations that were made throughout the duration of the project and whether it has been successfully implemented or not
* Any bugs/further development and outstanding issues existing in the system

# Handover and Running

The completed project will be copied onto a USB drive in a zip folder.

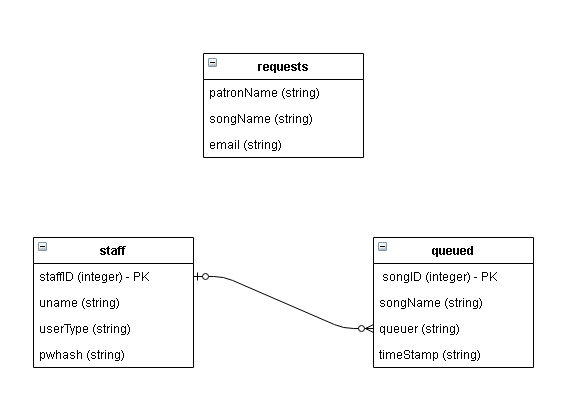
To run the server, python needs to be installed on the server computer and added to Windows PATH, as well as the many dependencies imported in the server (using pip install), which include:

* Flask
* SQLite3
* DateTime
* Json
* Time
* Atexit
* VLCCont
* Flask\_Bcrypt
* CSV
* OS
* Threading
* Random

Once that’s all installed, the client will go through a command prompt in the current working directory and then run the server script. Once it’s started running, it’ll trigger VLC with the media library and current queue inside. The client will then be able to open a browser, which will be spawned locally, displaying the staff webpage and asking for a log-in. The Android app will also be ready to be opened with the IP address to be confirmed, where it will connect to the server and display the playlist and current queue.

An MIT Licence Agreement has been included inside the code documentation, which allows the program to be open-sourced to anyone that uses the system.

# Implemented DB Schema



Both the staff/admin and gym patron can add songs, but the gym patrons are the only ones able to request for songs that are not on the playlist.

The only requirement from the gym patron is their name when they are adding a song to the queue.

When admin add staff to the system, they will be able to enter a username, password and a user type (staff or admin). Only admin will be able to add staff members and change passwords as well as other settings.

In “Queued Songs”, the “queuer” is either the “patronName” (Gym Patron) or the “staffUserName” (Staff) depending on who adds the song to the queue.

# Test Results

## Use Case – Admin Creating Staff Account

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case/Feature** | Register into Squashies Webpage | | |
| **Test Purpose** | To test if the **gym admin** can successfully create an account for their staff | | |
| **Expected Results** | See individual test data sets | | |
| **Success/Failure** | **Success** | | |
| **Test Results** | | | |
|  | **Test 1** | **Test 2** | **Test 3** |
| **User ID/Username** | admin | test | default |
| **Password** | admin1 | test123 | default |
| **Re-enter Password** | admin1 | test12 | default |
| **User/Admin** | admin | user | user |
| **Expected Results** | Pass | Fail | Pass |
| **Success/Failure** | Success | Success | Success |
| **Testing Date** | 29/01/2020 | 29/01/2020 | 29/01/2020 |
| **Actions to be Taken** | None | None | None |

## 

## Use Case - Search for Song in the Queue

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case/Feature** | Search for Song in the Queue – **(Android App)** | | |
| **Test Purpose** | When there are thousands of songs in the media library, there needs to be a way for the users to filter through the playlist so they can find what song they want quicker | | |
| **Expected Results** | See individual test data sets | | |
| **Success/Failure** | **Success** | | |
| **Test Results** | | | |
|  | **Test 1** | **Test 2** | **Test 3** |
| **Song Typed** | Crying in the Club | Greyhound | Greyhound |
| **Success/Failure** | Failure | Failure | Success |
| **Testing Date** | 29/01/2020 | 29/01/2020 | 29/01/2020 |
| **Actions to be Taken** | Need to create a temp variable to store this filtered output and compare it to the array list | Works but when I type in any other letter other than “G” it doesn’t pick up the rest of the song, so need to edit the statement | Song is successfully found and if “hound” is just typed it can find the song “Greyhound”  No further action needs to be taken |

## 

## Use Case - Add Song to the Queue

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case/Feature** | Adding a Song to the Queue **– (Android App)** | | |
| **Test Purpose** | This is one of the major features of the application where the gym patron will approach the tablet and pick a song from the playlist and add it to the queue. | | |
| **Expected Results** | See individual test data sets | | |
| **Success/Failure** | **Success** | | |
| **Test Results** | | | |
|  | **Test 1** | **Test 2** | **Test 3** |
| **Song Selected** | Crying in the Club | Greyhound | Greyhound |
| **Song ID** | 3 | 8 | 8 |
| **Patron Name** | Ben | Joshua | Rhys |
| **Success/Failure** | Success | Failure | Success |
| **Testing Date** | 29/01/2020 | 29/01/2020 | 29/01/2020 |
| **Actions to be Taken** | Test other songs to see if we get the same outcome | From debugging the issue, we saw that instead of the song ID being 8, it was 3 which belongs to another song that is already in the queue. | None |

## 

## Use Case - Remove Song from Queue

|  |  |  |
| --- | --- | --- |
| **Use Case/Feature** | Remove Song from Queue | |
| **Test Purpose** | Staff will be able to remove a song in the queue | |
| **Expected Results** | See individual test data sets | |
| **Success/Failure** | Success | |
| **Test Results** | | |
|  | **Test 1** | **Test 2** |
| **Song Title** | Mi Gente | Crying in the Club |
| **“Remove Button” Clicked** | Yes | Yes |
| **Is Song Currently Playing?** | No | Yes |
| **Expected Results** | Pass | Fail - Error Message displays |
| **Success/Failure** | Success | Fail |
| **Testing Date** | 04/01/2020 | 04/01/2020 |
| **Actions to be Taken** | None | None |

## 

## Use Case - Entering the IP Address on the App

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case/Feature** | Enter the IP Address on the App | | |
| **Test Purpose** | This was a new feature in the app implemented after the SAD report, for when the app first opens up and to establish a connection with the local server | | |
| **Expected Results** | See individual test data sets | | |
| **Success/Failure** | **Success** | | |
| **Test Results** | | | |
|  | **Test 1** | **Test 2** | **Test 3** |
| **IP Address** | 192.168.1.35 | 192.143 | 192.156.32.23 |
| **Valid/Invalid** | Valid | Invalid | Valid |
| **Redirects to Home** | No | No | Yes |
| **Expected Results** | Valid | Invalid | Valid |
| **Success/Failure** | Failure | Success | Success |
| **Testing Date** | 29/01/2020 | 29/01/2020 | 29/01/2020 |
| **Actions to be Taken** | Put the text input into a string and trim it based on the pattern (of a typical IP address) | None | None |

## Use Case - View Statistics in Staff Webpage

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case/Feature** | View Statistics in Staff Webpage | | |
| **Test Purpose** | The client wants to be able to check what songs have been queued during certain days and certain times to help keep track if they need to change the playlist (removing songs and adding new ones in) | | |
| **Expected Results** | See individual test data sets | | |
| **Success/Failure** | **Success** | | |
| **Test Results** | | | |
|  | **Test 1** | **Test 2** | **Test 3** |
| **Start Date** | 23/01/2020 | 27/01/2020 | 19/01/2020 |
| **Start Time** | 0900 | 1300 | 1400 |
| **End Date** | 28/01/2020 | 25/01/2020 | 22/01/2020 |
| **End Time** | 1300 | 0600 | 2300 |
| **List/Count** | List | List | Count |
| **Expected Results** | Table format of song names, queuer and the timestamp | Error – Start date greater than end date | Table format of song names, queuer and the timestamp and the amount of records in that table |
| **Success/Failure** | Success | Success | Success |
| **Testing Date** | 29/01/2020 | 29/01/2020 | 29/01/2020 |
| **Actions to be Taken** | None | None | None |

## Use Case - Requesting Song Not on Playlist

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case/Feature** | Requesting Song Not on Playlist | | |
| **Test Purpose** | This feature will be available on the Android app and it will allow gym patrons to enter a song that is not currently on the playlist and send it through to the Squashland server/database, in the hope that the gym admin will add it in the near future | | |
| **Expected Results** | See individual test data sets | | |
| **Success/Failure** | **Success** | | |
| **Test Results** | | | |
|  | **Test 1** | **Test 2** | **Test 3** |
| **Patron Name** | Joshua | Themis | Dharmjit |
| **Song Name** | Hey Ya | I Want It That Way |  |
| **Email Address** | ji.4@hotmail.com | jd.com | dha@gmail.com |
| **Expected Results** | Request has been sent | Enter a valid email address | Field cannot be empty |
| **Success/Failure** | Success | Success | Success |
| **Testing Date** | 03/04/2020 | 03/04/2020 | 03/04/2020 |
| **Actions to be Taken** | None | None | None |

# Variations Report

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Variation** | **Initiated By** | **Implemented or Not** |
| Functional/Use Case Design | Admin Control | Client | Initially it was intended that the gym staff would be able to register themselves an account  Has been implemented and completed - admin now has control of adding staff accounts and removing them when necessary |
| Saving Marquee Text Announcements into Text File/Webpage | Client | Has not been implemented - the visual marquee text feature was unable to be implemented as the group had run out of time to work on this feature |
| Data Design | Searchbar for Staff Webpage to Filter Media Library - particularly when there will be thousands of songs in the media library | Group | Has been implemented and completed |
| Integrate Timepicker and Date Picker | Academic Supervisor | Has been implemented and completed |
| Ask Client for IP Address when App First Runs | Academic Supervisor | Has been implemented and completed - input will be saved in the app and will stay the same even when app is closed and reopened and will only change if the user enters in another IP address - validation method included |
| “Favourite Songs” Feature for When Queue is Started Up | Client | Not implemented - due to time constraints since there were still other features discussed before this one that were still needed to be implemented |
| Screen Designs | Announcements and Statistics Page Content went Horizontal Rather than Vertical | Group | Design has been implemented and functionalities for each page are completed |
| Textbox Added in Splash Screen of Android App for the IP Address | Academic Supervisor/ Group | Has been implemented and completed - initially input was to be taken from an alert box (recommended by supervisor), however when implemented, there was an opinion that it needed to look cleaner and professional, therefore the change to a textbox. |

# Known Bugs

|  |  |
| --- | --- |
| **Known Bugs** | **Description** |
| Refreshing Current Queue on Android App | * In the Android app there is a set timer running in the background every 10 seconds, the current queue is refreshed and updated. * The function works fine but results are only shown when any button is pressed or when a user clicks onto the search bar to type in a song name. |

# Outstanding Development Work or Further Improvements

|  |  |
| --- | --- |
| **Development Work/Further Improvement** | **Notes/Actions** |
| Making the Marquee Text in the App Display the Current Song that is Playing | * Currently, the marquee text is set to just “Welcome to Squashlands Jukebox App. Feel free to search for a song from the playlist” * Write displaying marquee text function that would grab the first item of the array list and set the marquee text of that song name |
| “Favourite Songs” Feature for Current Queue | * Currently, the songs in the playlist are randomised into the song queue when the server first runs * Write favourite songs function that grabs the top 10 or so songs that have been requested and have them randomised and being among the first songs on the current queue |
| Marquee Text on VLC | * One of the requests from the client is that they want to be able to display text (visual announcements) scrolling across VLC during songs being played * After some research and discussions with the academic supervisor, we weren’t able to find a method where we can call the marquee text over the HTTP interface and can only be done during startup |

# Conclusion

To summarise, the system is ready to be officially handed over to the client, with all the testing, any variations and any known bugs/outstanding issues/further development being discussed along the way. By doing all this, we allow for the app to be used for the client’s company and help them with any components that could be improved for future development.