Group 2: WSUSCC1 2020

19356440 Christian Politis

19352262 Kyle Diamond-Squires

19354718 Sampadanna Heng

Systems Design and Analysis Document

Professional Experience - 300579

# Executive Summary

The following SAD report will focus on the project requirements and design of this Jukebox Application. We have outlined thirteen functional requirements and eight non-functional requirements of the project. These are core to the functionality of the application and how it will be implemented. Within the development, there are three possible risks and three constraints which include quarantine restrictions, limitations with testing and Limited experience with Kotlin. Our team has also found: Six Design considerations, Eighteen Use Cases and Six implemented tests. We have demonstrated three different Sequence Diagrams to provide a functional rundown of several situations. In terms of requirements, there are 13 functional and 6 non-functional requirements identified.

The implementation of an Entity Relationship Diagram is deemed unnecessary due to the scope of the project focusing on the implementation for front end therefore little to no changes to the currently implemented server. Finally, the team has implemented a breakdown visually of the System Architecture focusing on the basic concepts of the entire system, Software Architecture detailing the functionality such as adding or removing songs and, the Network and Communication Architecture which details the physical components of the tablet, computer running the server, speakers and video output device.

**Table of Contents**

[Executive Summary 2](#_Toc38872801)

[Introduction 5](#_Toc38872802)

[Client Statement 6](#_Toc38872803)

[Problem Statement 7](#_Toc38872804)

[System Requirements 8](#_Toc38872805)

[Functional Requirements 8](#_Toc38872806)

[Non-Functional Requirements 9](#_Toc38872807)

[Risks and Constraints 10](#_Toc38872808)

[Risks 10](#_Toc38872809)

[Constraints 10](#_Toc38872810)

[In-Experienced with Specific Languages (Kotlin) 10](#_Toc38872811)

[Physical Interactions 10](#_Toc38872812)

[Internet Accessibility 10](#_Toc38872813)

[Design Considerations 11](#_Toc38872814)

[Screen Layout and Size 11](#_Toc38872815)

[User Friendliness 11](#_Toc38872816)

[Consistency 11](#_Toc38872817)

[Security 11](#_Toc38872818)

[Screen Burn In 11](#_Toc38872819)

[System Architecture 12](#_Toc38872820)

[Software Architecture 13](#_Toc38872821)

[Network and Communication Architecture 14](#_Toc38872822)

[Detailed System Design 15](#_Toc38872823)

[Actors 15](#_Toc38872824)

[A10: Gym Member 15](#_Toc38872825)

[A20: Gym Staff 15](#_Toc38872826)

[A30: Application Admin 16](#_Toc38872827)

[Use Case Diagram 17](#_Toc38872828)

[Expanded Use Cases 18](#_Toc38872829)

[UC10: Search for available songs 18](#_Toc38872830)

[UC20: Adding song to the Queue 19](#_Toc38872831)

[UC30: User required to enter their name 20](#_Toc38872832)

[UC40: Requesting a song for the playlist 21](#_Toc38872833)

[UC50: Accessing the Admin Panel 22](#_Toc38872834)

[UC60: Password Authentication for Admin Panel 23](#_Toc38872835)

[UC70: Changing Announcements and Time Intervals 24](#_Toc38872836)

[UC80: Changing Messages and Interval Times for Marquee Banner 25](#_Toc38872837)

[UC90: Pause/Play/Skip/Reverse and a Song 26](#_Toc38872838)

[UC100: Removing a Song from the Queue 27](#_Toc38872839)

[UC110: Changing the Colour Scheme of the Application 28](#_Toc38872840)

[UC120: Viewing Weekly and Monthly Statistics 29](#_Toc38872841)

[UC130: Changing the Screensaver Time Interval 30](#_Toc38872842)

[UC140: Logging into the IP Configuration Screen 31](#_Toc38872843)

[UC150: Changing the IP Address within the Application 32](#_Toc38872844)

[UC160: Logging into the Settings Screen 33](#_Toc38872845)

[UC170: Changing the Staff Login Password 34](#_Toc38872846)

[UC180: Returning to the Main Screen from the Admin Panel 35](#_Toc38872847)

[Sequence Diagrams 36](#_Toc38872848)

[Sequence Diagram 1: Adding a Song to the Queue 36](#_Toc38872849)

[Sequence Diagram 2: Requesting a Song from the Library 37](#_Toc38872850)

[Sequence Diagram 3: Login to Admin Panel and using Admin Features 38](#_Toc38872851)

[Entity Relationship Diagram (ERD) 39](#_Toc38872852)

[Test Plan 54](#_Toc38872853)

[Features/Use Cases to be Tested 54](#_Toc38872854)

[Candidate Test Cases/Test Data 56](#_Toc38872855)

[Conclusion 58](#_Toc38872856)

# Introduction

Currently, the majority of gyms and fitness centres implement a music system that allows users to decide what music is to play which can be controlled through a Jukebox or music queuing application. The Squashlands Gym & Fitness currently face this issue, as a result, they have been requested several times for music to change that is more suitable for them. Feedback from members is a high priority. If there is no response to feedback, members will take business elsewhere believing they can find better value for their money. This issue should be dealt with as soon as possible.

Our team is developing Squashies Jukebox application and its user interface using the Kotlin programming language. Implementing a rest based service structure allows us to send requests and receive the data needed from song files as well as other features that help achieve full functionality. By designing a jukebox application using a more recent API version, this provides the application with futureproofing for maintenance in the long term which is what has been asked from the client.

The objective of this SAD document is to display an understanding of the project to our client, ensuring we have the correct understanding of the system requirements and design according to the client’s vision and concept. This document will help our team understand any final requirements or adjustments to what will be the final product.

The scope of our project ranges 8 weeks, currently having 4 weeks remaining until project handover. The limitations our team have found within the project include; limited programming knowledge in the programming language Kotlin. The timeframe we have has become a limitation due to the technologies that we are using. Due to the current pandemic happening around the world, another limitation our group has is working remotely and cooperatively

The significance of this project is that it will become a valuable business tool to Squashlands Gym, providing an enjoyable experience to the users of the gym, while the client has access to analytical features, allowing data gained from the member of the gym to help further the experience inside.

# Client Statement

Our client, Angel Georgieff, has requested for an Android jukebox application that will allow gym patrons to play songs within the gym's premises through a pre-selected Squashlands Gym Playlist.

A large group of people spend the majority amount of time exercising and working out at a gym or fitness center. Gyms strive to be the best competitively offering new deals and features that show off and stand out from the rest. The music being played is a considerably demanding feature when it comes to working out. Many Gyms offer this feature and are accessible to members to decide what should be played.

Squashlands Gym and fitness is a well-known established gym in the outer western suburbs who want to implement the current member feedback to improve their whole Gym experience. Members have provided feedback that they would like to have control over the music when they workout rather than an auto queue system in which the music that is playing is not to their taste. Our team will develop and implement a jukebox application to manage the playing of the music in the gym. It will entail the development of an Android based app which will be used on a tablet within the gym that will allow gym management to control and manage playlists of music whilst allowing members to queue up music to play and enjoy the songs they like to work out too.

# Problem Statement

The Application’s Development Idea has evolved from the previous project completed within the Gym, allowing the gym to play music through a server and have a basic functionality whilst exploring the concept of this project through a pre-existing simple application.

Within this project, the client expects the application to be modified with improvements of design whilst fulfilling the following set of criteria within functionality:

* The application will allow gym patrons to queue songs within the application and collect details from them for statistical data.
* The staff users will be able to manage the song queues, volume (if applicable) and music currently playing through the Admin Panel.
* The admin (Angel) will be able to set admin passwords, staff passwords manually within the application as part of their security for accessing the admin panels and configurations.
* The Gym Staff Members will be able to set the time intervals for announcements to play, configure the marquee banner text, screen timeout settings and change the colour theme of the application with a variety of common user-friendly colours.

The team will be using Kotlin to develop the application and will be operated on a central tablet that is located within the gym accessible to all gym members that will retrieve and send data back and forth to the server. Python Script will be used to improve the already existing system, with implementing bug fixes and changes that are applicable to allow the application on the tablet to function to its full potential and provide the client with the team's highest quality of work.

# System Requirements

## Functional Requirements

|  |  |
| --- | --- |
| **ID** | **Description** |
| FR01 | Gym members will be able to search for songs in the playlist |
| FR02 | Gym members will be able to add songs to the queue |
| FR03 | Gym members will be able to request a song |
| FR04 | Gym staff will be able to login to the admin panel |
| FR05 | Gym staff will be able to start, stop, skip, and adjust the volume of the music |
| FR06 | Gym staff will be able to remove and add songs to the queue |
| FR07 | Gym staff will be able to edit announcements and set intervals |
| FR08 | Gym staff will be able to edit the message text in marquee banner and set intervals |
| FR09 | Gym staff will be able to change the application colour theme |
| FR10 | Gym staff will be able to view song statistics |
| FR11 | Gym staff will be able to change the screensaver timer |
| FR12 | Admin will be able to change the staff password |
| FR13 | Admin will be able to configure the IP address |

## Non-Functional Requirements

|  |  |
| --- | --- |
| **ID** | **Description** |
| NFR01 | Security |
| NFR02 | Usability |
| NFR03 | Availability |
| NFR04 | Maintainability |
| NFR05 | Performance |
| NFR06 | Compatibility |

**Security**: Security has been accounted for in this application by dividing gym members, staff, and admin. The security has a three-tier system. Clients can access music, staff have limited admin privileges, and the admin has full privileges.

**Usability**: The main action for the users is placed in front of them with instructions on how to perform the task of adding or song or requesting a song. Users can quickly add music to the queue due to how easy the process is.

**Availability**: Because the application will be using the intranet to connect to the server, the update cycle between the application and server will likely occur as fast as the intranet connection will allow, our teams estimate being approximately 3 seconds

**Maintainability**: Maintainability of the application is covered in the back and front-end development. The code is easy to read and is commented throughout to ensure an easy understanding for anyone else who might need to look at the code in the future.

**Performance**: Performance in systems are affected when user/s are doing more than the system can deal with. Performance shouldn’t be an issue for this application as there are only two events that the main actor in this project can perform. The song queue has an enforced limit which can be changed, that will prevent the performance dropping from excessive user requests.

**Compatibility**: The application is being made using Android 10 (API 29) to implement future-proofing for the application. This means the application can be installed on numerous up to date android devices, allowing the choice of which device to use in the gym, as well as being able to change devices if necessary

# Risks and Constraints

## Risks

|  |  |  |
| --- | --- | --- |
| **Risk** | **Resolution** | **Type of Risk (Internal/External)** |
| Team members unable to attend either client, supervisor or team meetings due to other reasons (Work, other commitments etc.) | Time management and planning | Internal |
| Changes and implementation of newer features later on in the development of the application | Revise management of resources and time which will enable us to invest the necessary and required resources to complete the necessary functionality and what’s been requested. | External |
| Client testing the software for feedback with limitations of access due to quarantine | Zoom calls to be arranged which will screen share functionality or find a way where they can test the software from their device. | External |

## Constraints

### In-Experienced with Specific Languages (Kotlin)

Due to the lack of knowledge and inexperience with Kotlin and requiring extra time to research and understand the language and required language. This may restrict the time we have for the actual development.

### Physical Interactions

With the quarantine in place, it has restricted the limited contact we have amongst our team members, supervisor and the client. We will have to solely rely on internet-based communications as we coordinate this project.

### Internet Accessibility

In the case where a member cannot gain access to the internet, it may impact on the level of communication between us as well as connecting to any online repositories to work and backup and code implementation.

# Design Considerations

## Screen Layout and Size

The design of the screen is to be suitable for a tablet, and the application will only be used in a landscape (horizontal) view, as it will allow for the most information to be displayed on the page whilst being compatible to the Squashies current hardware and device.

## User Friendliness

User-friendliness has been heavily focused on the development of this application by designing the interface to be clear and precise. Users are interacting with the application through the use of large buttons that are easily visible allowing the application to create a smooth user experience, and provide an easier learning curve and be usable by a wide range of audience.

## Consistency

Consistency within the application would be portrayed through the style guide of the application, the buttons, colours and shapes would all be uniform in every single page, allowing for flow within the application without confusing the audience through navigation.

Bright colours are used to indicate interactive buttons that users can press on, to perform a function within the application. For example, searching for a song, submitting a song request, or accessing the admin panel would all be highlighted with a bright orange button by default.

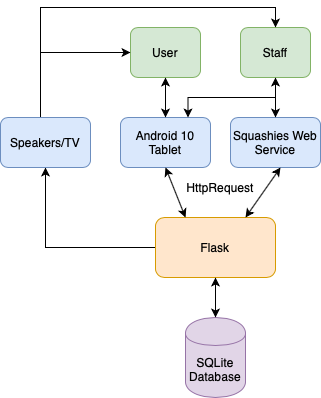
## Security

Security has been implemented through two different methods, the first method of security is the hidden functionality of accessing into the Admin Panel. To initiate this, users will have to enable the Admin button by pressing the logo five times, to make the Admin Button visible. This would be the first layer of security. The second layer of security would be to authenticate with an existing user account within the system, that belongs to a staff member or the administrator.

## Screen Burn-In

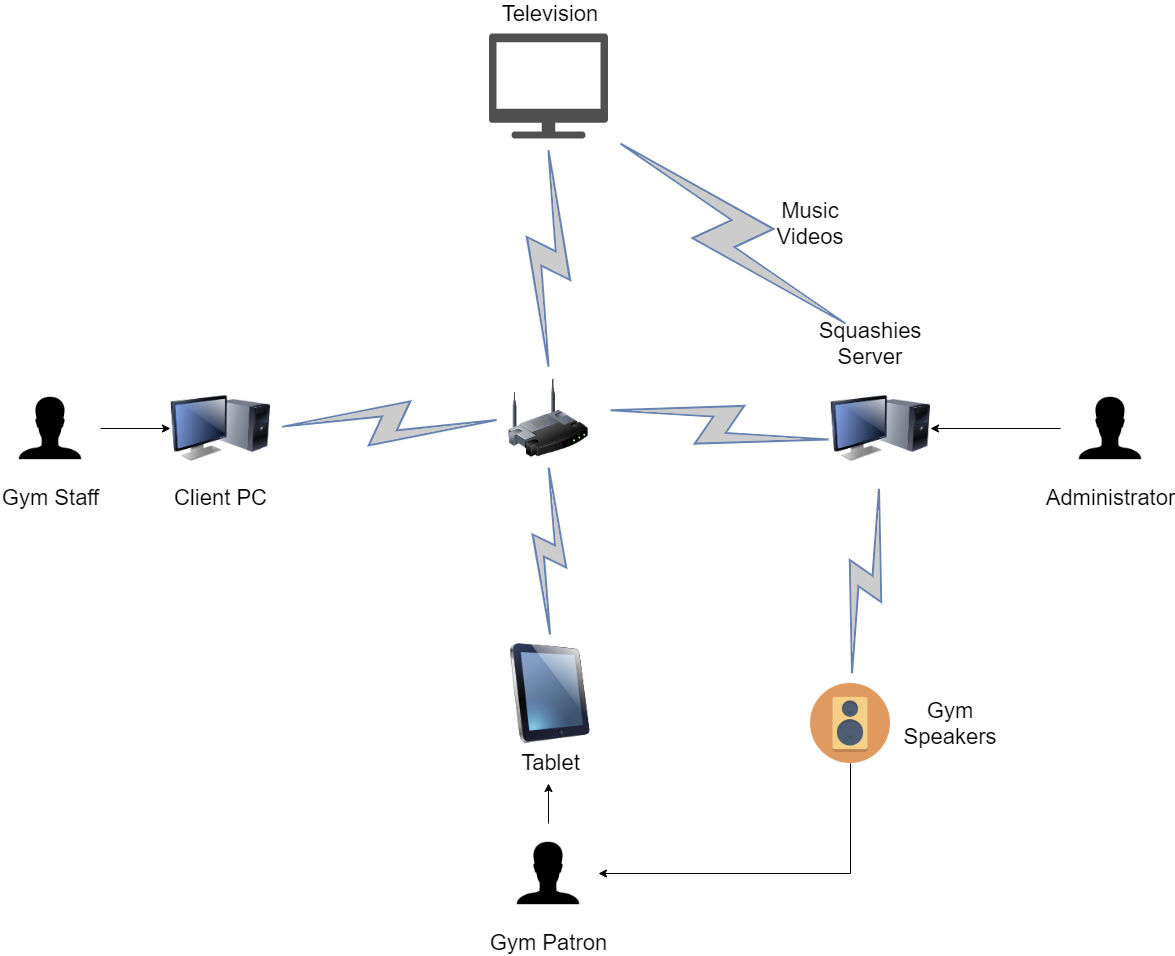
A screen timeout functionality has been implemented to reduce or eliminate the issue of having screen burn-in, with this feature in place, we have created the opportunity to advertise the gyms on-going deals and instructions as well as having a default screen that will have the gym logo bounce around.

# System Architecture



# https://lh6.googleusercontent.com/sddH71UZJj428d573yHevkmQB3R2pbZABUVqbSPFfiy663nVIBnk6-ySGsaQGRmWihYUIQUTfwbk-6MWBsOl2HVcFhkRT2fHxMnui7E1UHalW6LNdhRX4rcyE6uLE4IAluVxfbjXSoftware Architecture

# Network and Communication Architecture



# Detailed System Design

## Actors

### A10: Gym Member

|  |  |
| --- | --- |
| **Actor** | A10: Gym Member |
| **Type and Stereotype** | This actor represents all gym members who will use the Squashies Jukebox App |
| **Description** | These users will be able to add songs from a playlist to a queue and request new songs to be added to the playlist |
| **Relationships** | None |
| **Use Case Interaction** | UC10, UC20, UC30, UC40 |
| **Interface Specifications** | Refer to Screen Designs |
| **References** | None |

### A20: Gym Staff

|  |  |
| --- | --- |
| **Actor** | A20: Gym Staff |
| **Type and Stereotype** | This actor represents all the staff members who will use the Squashies Jukebox App |
| **Description** | These users can able to pause/play and skip songs, change announcements and messages being displayed, change colour themes, view statistics |
| **Relationships** | None |
| **Use Case Interaction** | UC10, UC20, UC30, UC50, UC60, UC70, UC80, UC90, UC100, UC110, UC120, UC130, UC180 |
| **Interface Specifications** | Refer to Screen Designs |
| **References** | None |

### A30: Application Admin

|  |  |
| --- | --- |
| **Actor** | A30: Application Admin |
| **Type and Stereotype** | This actor represents all the staff members who will have admin access for the application |
| **Description** | These users can able to change the staff password and configure the IP for the application |
| **Relationships** | None |
| **Use Case Interaction** | UC50, UC60, UC70, UC80, UC90, UC100, UC110, UC120, UC130, UC140, UC150, UC160, UC170, UC180 |
| **Interface Specifications** | Refer to Screen Designs |
| **References** | None |

# D:\WSU\SCC1 2020\Professional Experience\systemsAnalysisDesign\UserCaseDiagram_Updated.pngUse Case Diagram

# Expanded Use Cases

## UC10: Search for available songs

|  |  |
| --- | --- |
| **Use Case:** | UC10: Search for available songs |
| **Actors:** | A10: Gym Member, A20: Gym Staff |
| **Description:** | Allow either gym members or Gym staff to decide the music that they would like to be played via the android application connected to the squashies server. |
| **Pre-Condition:** | Server and application must be on and available. |
| **Post-Condition:** | Gym Staff or Member has found the song they were looking for in the search query. |

**Typical Course of Events**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Tap on the search bar | Nothing | 2 | Provide a hint to enter the user's input into the search bar |
| 3 | The user enters their input | Filtered search result | 4 | Provide results based on string input |

**Alternative Course of Events**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Tap on the search bar | Nothing | 2 | Provide a hint to enter the user's input into the search bar |
| 3 | The user enters their input | Filtered search result | 4 | Empty list due to the string value not matching any of the available songs. This can be due to a mismatch/misspelt string input or the song does not exist as in the available library. |

## UC20: Adding a song to the Queue

|  |  |
| --- | --- |
| **Use Case:** | UC20: Adding a song to the Queue |
| **Actors:** | A10: Gym Member, A20: Gym Staff |
| **Description:** | When one of the actors has searched and found the song they want, they will tap on the search result which will queue it into the system. |
| **Pre-Condition:** | Server and application must be on and available. User must make a query in the search bar. |
| **Post-Condition:** | Gym Staff or Members have received a name request. |

**Typical Course of Events**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Tap on the search result which will queue it into the system | Provide a pop up for user input | 2 | A pop up appears asking the user to enter their first name |

**Alternative Course of Events**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | User does not tap on any search result after a timed period | Search is cleared after an idle period. | 2 | Search is reset and ready for the next query |

## UC30: User required to enter their name

|  |  |
| --- | --- |
| **Use Case:** | UC30: User required to enter their name |
| **Actors:** | A10: Gym Member, A20: Gym Staff |
| **Description:** | A pop up appears after the user has entered their name. This will be deposited into the database for analysis purposes |
| **Pre-Condition:** | Server and application must be on and available. User must enter select a song |
| **Post-Condition:** | Gym Staff or Members have successfully added the song to the queue and name into the database. |

Typical Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Enter their name which will queue it into the system | Provide a successful response | 2 | A pop up appears letting the users know their song was successfully queued. |

Alternative Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | User does not enter their name | Pop up disappears | 2 | After a small period of idle time, the pop up will disappear and the search query will be reset. |

## UC40: Requesting a song for the playlist

|  |  |
| --- | --- |
| **Use Case:** | UC40: Requesting a song to be added to the playlist |
| **Actors:** | A10: Gym Member |
| **Description:** | If the user can't find the song they are looking for, they can fill in a form that requires name, email, and song request. |
| **Pre-Condition:** | Server and application must be on and available. |
| **Post-Condition:** | The form is submitted to be later reviewed by the admin. Once the song is added to the database, the admin will then send an email letting the user know their song has been added to the database. |

Typical Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Click “**Don’t see what you want?**” button | Song Request | 2 | Slide the screen to the left to display the Song Request Form |
| 3 | Fills out questions that are both required and optional then presses submit. | Request Form Completed | 4 | Send data to the server, the server writes data into a CSV file. |

Alternative Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Click “**Don’t see what you want?**” button | Song Request | 2 | Slide the screen to the left to display the Song Request Form |
| 3 | User doesn’t fill out the required questions within the form | Screen Times out | 4 | Returns the application to the Main Menu |

## UC50: Accessing the Admin Panel

|  |  |
| --- | --- |
| **Use Case:** | UC50: Accessing the Admin Panel |
| **Actors:** | A20: Gym Staff, A30: Application Admin |
| **Description:** | A navigational button called “Admin Panel” appears after the logo has been clicked on five times. |
| **Pre-Condition:** | Server and application must be on and connected. |
| **Post-Condition:** | Accessibility to the Admin Panel Login Page. |

Typical Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| **1** | Click on the Logo five times. | Displays Admin Panel Button | **2** | Admin Panel Button appears after three seconds |
| **3** | Click on “**Admin** **Panel**” Button | Change Screen | **4** | Re-directs to Admin Panel Login Page |

Alternative Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| **1** | Click on the Logo five times. | Displays Admin Panel Button | **2** | Admin Panel Button appears after three seconds |
| **3** | User does not click on Admin Panel Button | Click Timer Timeouts | **4** | The Admin Panel button disappears |

## UC60: Password Authentication for Admin Panel

|  |  |
| --- | --- |
| **Use Case:** | UC60: Password Authentication for Admin Panel |
| **Actors:** | A20: Gym Staff, A30: Application Admin |
| **Description:** | User is required to enter a Staff or Admin password to authenticate into the Admin Panel. |
| **Pre-Condition:** | Server and application must be on and available. |
| **Post-Condition:** | Gym Staff and Application Admin can view the Admin Panel Page. |

Typical Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Enters in Staff Password and clicks “Login”. | Validates User Credentials with Local Data – CORRECT CREDENTIALS | 2 | Re-directs user to Admin Panel Page |

Alternative Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Enters in Staff Password and clicks “Login”. | Validates User Credentials with Local Data – INCORRECT CREDENTIALS | 2 | A pop up will appear notifying that the password was incorrect. |

## UC70: Changing Announcements and Time Intervals

|  |  |
| --- | --- |
| **Use Case:** | UC70: Changing Announcements and the Intervals |
| **Actors:** | A20: Gym Staff, A30: Application Admin |
| **Description:** | Process of how users change announcements selected or time interval between each announcement. |
| **Pre-Condition:** | Server and application must be on and available. Staff or Admin is logged into the Admin Panel and clicked on Announcements Tile. |
| **Post-Condition:** | Time Interval or Announcement Selected is changed. |

Typical Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Changes Announcement Selected | Provide a successful response | 2 | A pop up appears letting the user know that a different announcement has been selected. |

Alternative Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Changes Time Interval for Announcements | Provide a successful response | 2 | A pop up appears letting the user know that a different time interval has been selected. |

## UC80: Changing Messages and Interval Times for Marquee Banner

|  |  |
| --- | --- |
| **Use Case:** | UC80: Changing Messages and Interval Times for Marquee Banner |
| **Actors:** | A20: Gym Staff, A30: Application Admin |
| **Description:** | Process of how staff users change the message on the marquee banner or time interval between each announcement. |
| **Pre-Condition:** | Server and application must be on and available. Staff or Admin is logged into the Admin Panel clicked on Message Scroller Tile. |
| **Post-Condition:** | Marquee Scroller or Time Interval for Marquee Scroller is changed. |

Typical Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Enters in a new string of text in Marquee Text Input Box | Checks Local Saved Messages and Appends new message. | 2 | Checks if the system has 3 saved messages if so, delete the last message, and add a new message to the end. |
| 3 | Selects on the Message within the Message List | Highlight Message | 5 | The message will be indicated that it has been selected. |
| 4 | Selects “**Set Active**” | Changes Current Active Message | 6 | Changes the current text to the one that is selected. |

Alternative Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Selects on the Message within the Message List | Highlight Message | 2 | The message will be indicated that it has been selected. |
| 3 | Selects “**Remove**” | Pop Up Appears | 4 | Deletes a message from System and displays a pop up that the message has been deleted. |

## UC90: Pause/Play/Skip/Reverse and a Song

|  |  |
| --- | --- |
| **Use Case:** | UC90: Pause/Play/Skip/Reverse and a Song |
| **Actors:** | A20: Gym Staff, A30: Application Admin |
| **Description:** | Process of how staff users can change the music through the music player within the Admin Panel. |
| **Pre-Condition:** | Server and application must be on and available. Staff or Admin is logged into the Admin Panel clicked on Music Player. |
| **Post-Condition:** | Setting on the current music playing is changed. |

Typical Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Selects “**Pause**” | Pause Song | 2 | Pauses the currently playing song. |

Alternative Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Selects “**Skip**” | Skip Song | 2 | Skips the currently playing song to the next song in queue. |

## UC100: Removing a Song from the Queue

|  |  |
| --- | --- |
| **Use Case:** | UC100: Removing a Song from the Queue |
| **Actors:** | A20: Gym Staff, A30: Application Admin |
| **Description:** | Process of how staff users can remove a song through the music player within the Admin Panel. |
| **Pre-Condition:** | Server and application must be on and available. Staff or Admin is logged into the Admin Panel clicked on Music Player button. |
| **Post-Condition:** | A song is removed from the queue. |

Typical Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Remove “**Song**” | Provide a confirmation pop-up | 2 | A pop up appears asking the user to confirm the decision. |
| 3 | Accepts Pop-up message | Remove Song | 4 | The song is removed from the queue. |

Alternative Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Remove “**Song**” | Provide a confirmation pop-up | 2 | A pop up appears asking the user to confirm the decision. |
| 3 | Decline Pop-up message | Don’t Remove Song | 4 | The song is not removed from the queue. |

## UC110: Changing the Colour Scheme of the Application

|  |  |
| --- | --- |
| **Use Case:** | UC110: Changing the Colour Scheme of the Application |
| **Actors:** | A20: Gym Staff, A30: Application Admin |
| **Description:** | Process of how staff users can change the colour theme within the Admin Panel. |
| **Pre-Condition:** | Server and application must be on and available. Staff or Admin is logged into the Admin Panel clicked on Colour Theme button. |
| **Post-Condition:** | Colour within the application has been changed. |

Typical Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Selects “**Dark Theme**” | Change Colour Theme | 2 | The System will change the colour theme of the application from current to Dark. |

Alternative Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Selects “**Preview**” | Pop Up Appears | 2 | After a small period of idle time, the pop up will disappear. |

## UC120: Viewing Weekly and Monthly Statistics

|  |  |
| --- | --- |
| **Use Case:** | UC120: Viewing Weekly and Monthly Statistics |
| **Actors:** | A20: Gym Staff, A30: Application Admin |
| **Description:** | Process of how staff users can view the statistics of songs played through the within the Admin Panel. |
| **Pre-Condition:** | Server and application must be on and available. Staff or Admin is logged into the Admin Panel clicked on Statistics button. |
| **Post-Condition:** | Can view total statistics of songs within a certain time period. |

Typical Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Selects “**Time Period**” | Sets Date | 2 | Changes the time period date to the time user has set the time and displays the data for the user during that time period. |

Alternative Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Selects “**View All**” | Changes view | 2 | Displays all records of each song within the system that has been played. |

## UC130: Changing the Screensaver Time Interval

|  |  |
| --- | --- |
| **Use Case:** | UC130: Changing the Screensaver Time Interval |
| **Actors:** | A20: Gym Staff, A30: Application Admin |
| **Description:** | Process of how staff users change the time interval of the screen saver through the within the Admin Panel. |
| **Pre-Condition:** | Server and application must be on and available. Staff or Admin is logged into the Admin Panel clicked on Screensaver button. |
| **Post-Condition:** | Application Screen Timeout setting is changed. |

Typical Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Selects “**30 Minutes**” | Pop Up Appears | 2 | After a small period of idle time, the pop up will disappear. |

Alternative Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Select “**Never**” timeout option. | Pop up appears | 2 | The pop appears in the centre and darkens the surrounding areas to change attention. |
| 3 | Selects “**Confirm**” | Pop up disappears | 4 | System changes settings within the application. |

## UC140: Logging into the IP Configuration Screen

|  |  |
| --- | --- |
| **Use Case:** | UC140: Logging into the IP Configuration Screen |
| **Actors:** | A30: Application Admin |
| **Description:** | A login page displays asking for the user to authenticate to access the IP Configuration screen. |
| **Pre-Condition:** | Server and application must be on and available. Staff is logged into the Admin Panel. |
| **Post-Condition:** | Authenticates user to IP Configuration Screen. |

Typical Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Enters in “**Admin**” Password | Provide a successful response | 2 | A pop up appears letting the users know that they have been authenticated and changes screen. |

Alternative Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Enters in “**Admin**” Password | Provide an unsuccessful response | 2 | A pop up appears letting the users know that that their password is incorrect and the pop-up disappears after. |

## UC150: Changing the IP Address within the Application

|  |  |
| --- | --- |
| **Use Case:** | UC150: Changing the IP Address within the Application |
| **Actors:** | A30: Application Admin |
| **Description:** | The process of how a user changes the IP Address to connect to the server. |
| **Pre-Condition:** | Server and application must be on and available. Staff is logged into the Admin Panel and IP Configuration Panel. |
| **Post-Condition:** | Changed IP Address within Application and reconnects to the server. |

Typical Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Enter new IP Address in IP Input Box and submits | Provide a successful response | 2 | A pop up appears letting the user know that the server was able to be connected. |

Alternative Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Enter new IP Address in IP Input Box and submits | Provide an unsuccessful response | 2 | A pop up appears letting the user know that the server was unable to be connected due to the wrong IP address being submitted. |

## UC160: Logging into the Settings Screen

|  |  |
| --- | --- |
| **Use Case:** | UC160: Logging into the Settings Screen |
| **Actors:** | A30: Application Admin |
| **Description:** | A login page displays asking for the user to authenticate to access the Settings screen. |
| **Pre-Condition:** | Server and application must be on and available. Staff is logged into the Admin Panel. |
| **Post-Condition:** | Authenticates user to Settings Screen. |

Typical Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Enters in “**Admin**” Password | Provide a successful response | 2 | A pop up appears letting the users know that they have been authenticated and changes screen. |

Alternative Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Enters in “**Admin**” Password | Provide an unsuccessful response | 2 | A pop up appears letting the users know that that their password is incorrect and the pop-up disappears after. |

## UC170: Changing the Staff Login Password

|  |  |
| --- | --- |
| **Use Case:** | UC170: Changing the Staff Login Password |
| **Actors:** | A30: Application Admin |
| **Description:** | The process of how an admin can change a Staff Login password to authenticate to the Admin Panel. |
| **Pre-Condition:** | Server and application must be on and available. Admin is logged into the Admin Panel and Settings Screen. |
| **Post-Condition:** | Changed Admin Password within the Application. |

Typical Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Enter new Staff Password and Submits | Provide a successful response | 2 | A pop up appears letting the user know that the Staff Password has been changed. |

Alternative Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Clicks on “**How to change Admin Password**” tooltip | Pop Up Appears | 2 | After a small period of idle time, the pop up will disappear and the search query will be reset. |

## UC180: Returning to the Main Screen from the Admin Panel

|  |  |
| --- | --- |
| **Use Case:** | UC180: Returning to the Main Screen from the Admin Panel |
| **Actors:** | A20: Gym Staff, A30: Application Admin |
| **Description:** | Method of how users re-direct back to the main screen from the admin panel. |
| **Pre-Condition:** | Server and application must be on and available. Staff is logged into the Admin Panel. |
| **Post-Condition:** | The application is displaying the main screen. |

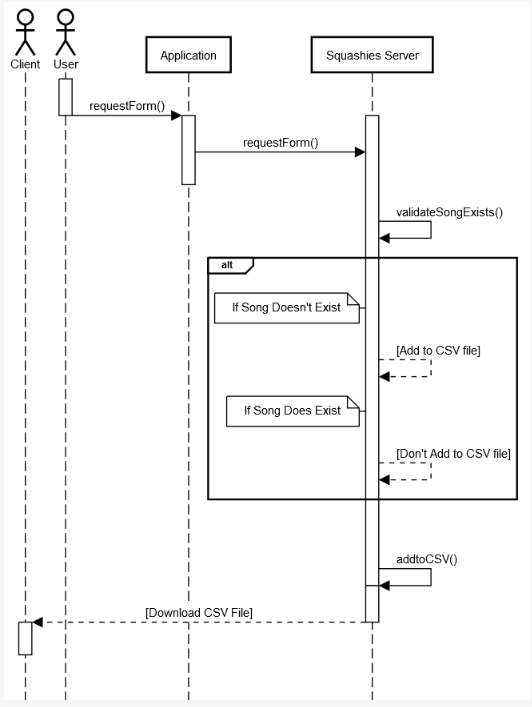
Typical Course of Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Actor Action** | **System Event** | **#** | **System Response** |
| 1 | Selects “**Log Out**” | Screen Changes | 2 | Changes the displayed screen back to the main screen |

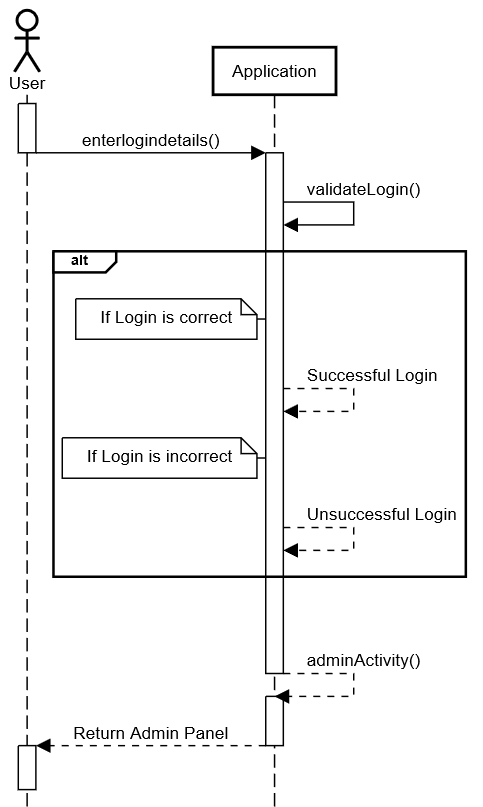
# Sequence Diagrams

## https://lh6.googleusercontent.com/EuI_4ysgM3un88t4bKI2B5hbo86tBbtsb4hXVeUeGDr6u2rbSR_6-gXUVyiFJEEn5hKDX2s096K44QP5_tzQ2ued_TT8b8FYEmnxMo2024h_lmZA_89IGVKxOeGOcowgakehZ7kFSequence Diagram 1: Adding a Song to the Queue

## Sequence Diagram 2: Requesting a Song for the Library

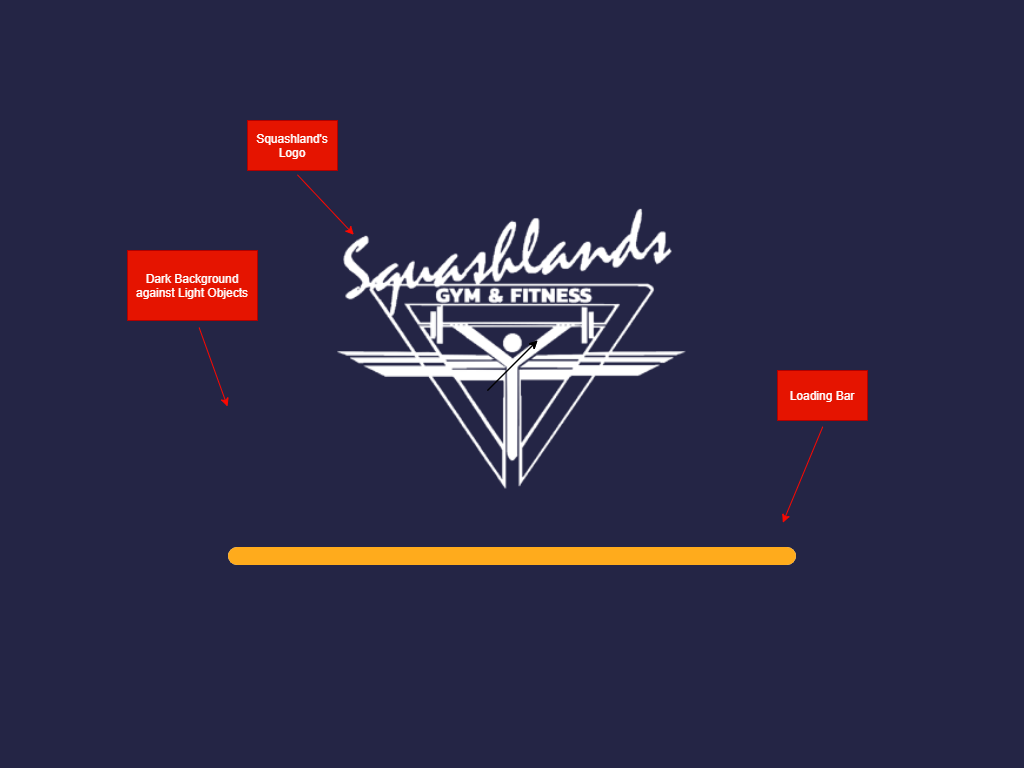


## Sequence Diagram 3: Log in to Admin Panel and using Admin Features



# Entity Relationship Diagram (ERD)

Our team will not have the need for an ERD as the databases within our system does not communicate with each other.Screen Designs



**Screen Design 1:** The Loading Screen – Loads for three seconds before proceeding to the Initial Configuration page.



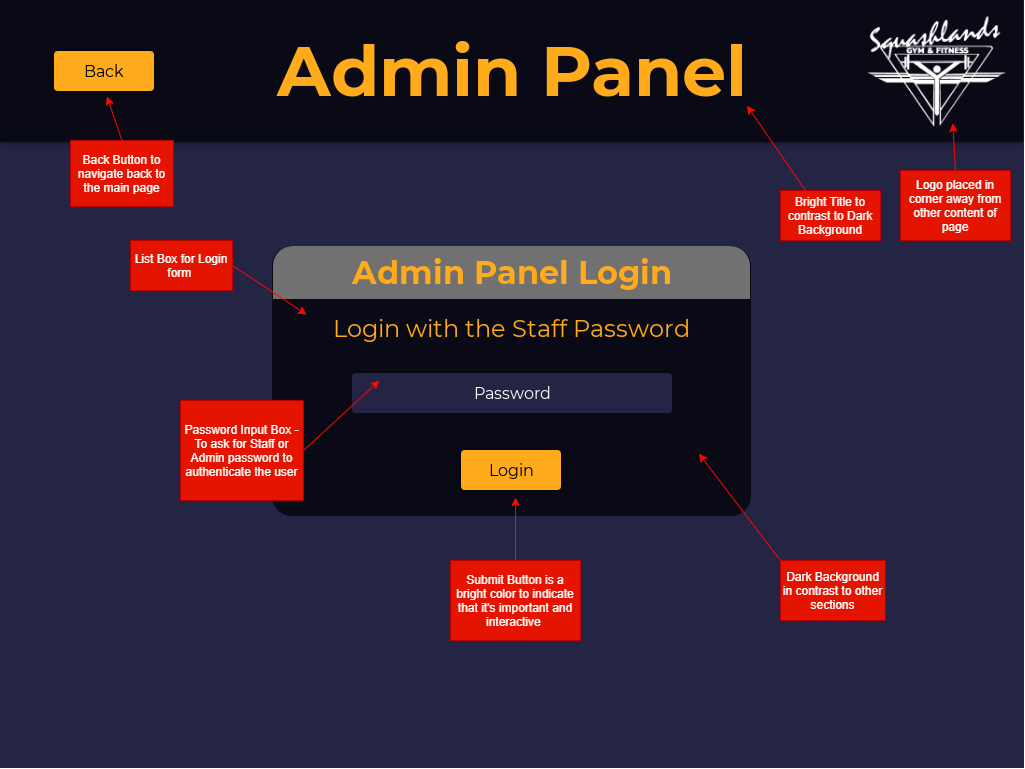
**Screen Design 2:** The Initial Configuration Page – Connects the application to the server and retrieves admin and staff accounts from server and saves locally.



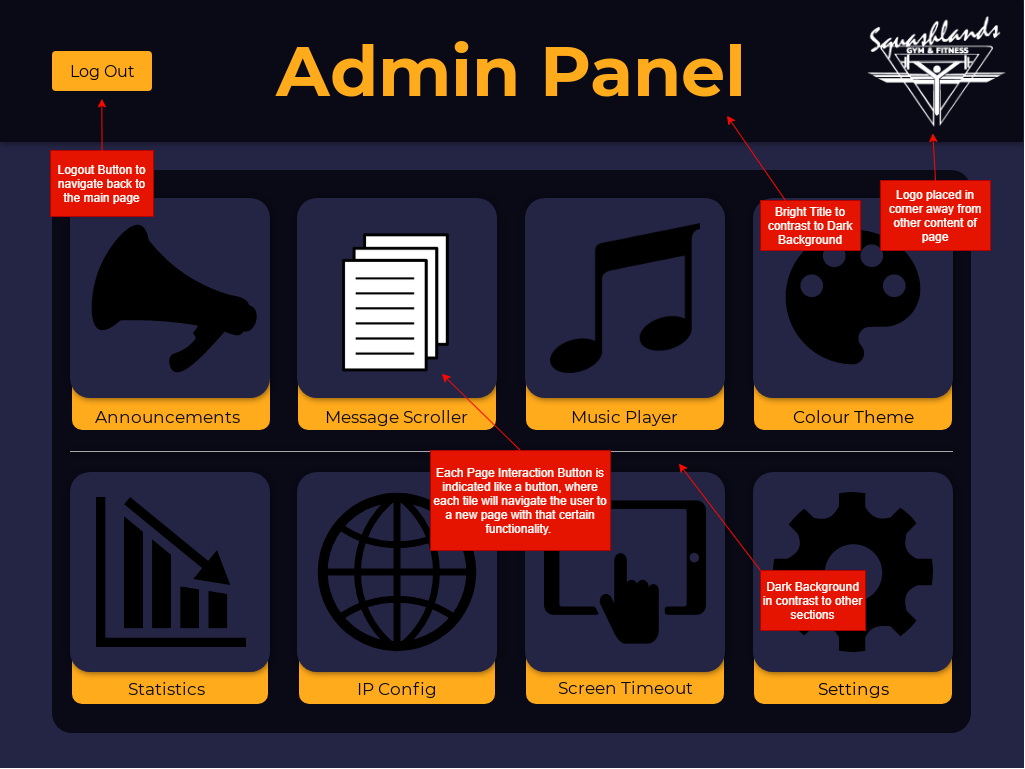
**Screen Design 3:** Application Main Page – Queue List, Song Search, Current Song Playing, Accessibility to Admin Panel and Song Request Form.



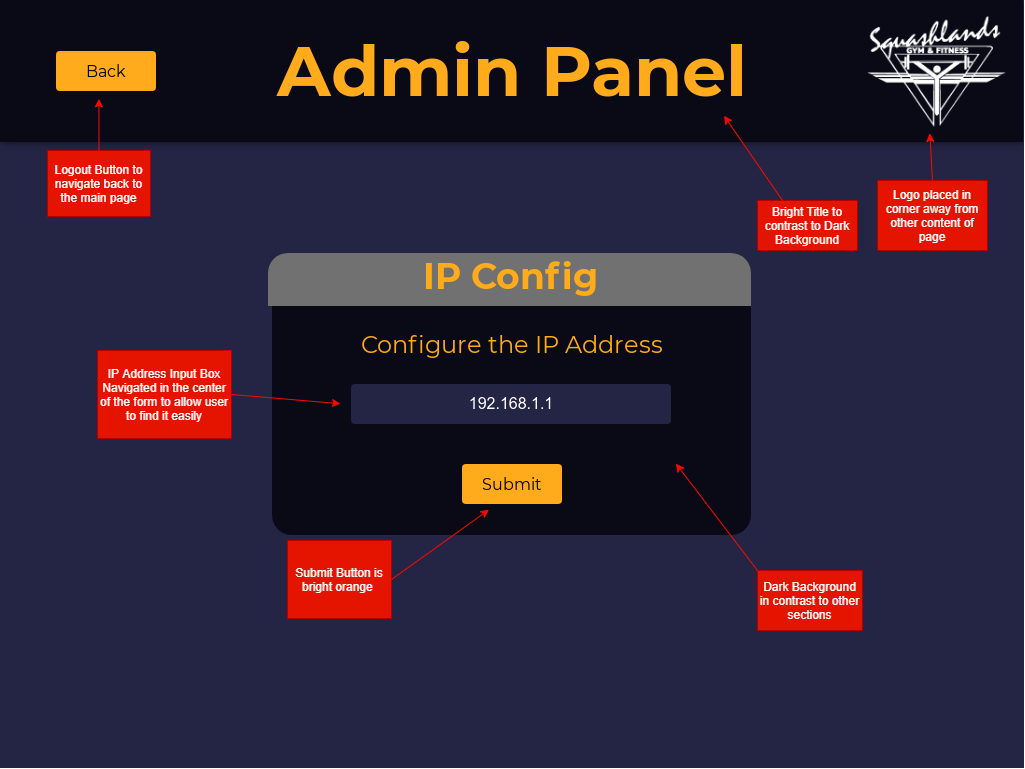
**Screen Design 4:** Song Request Form



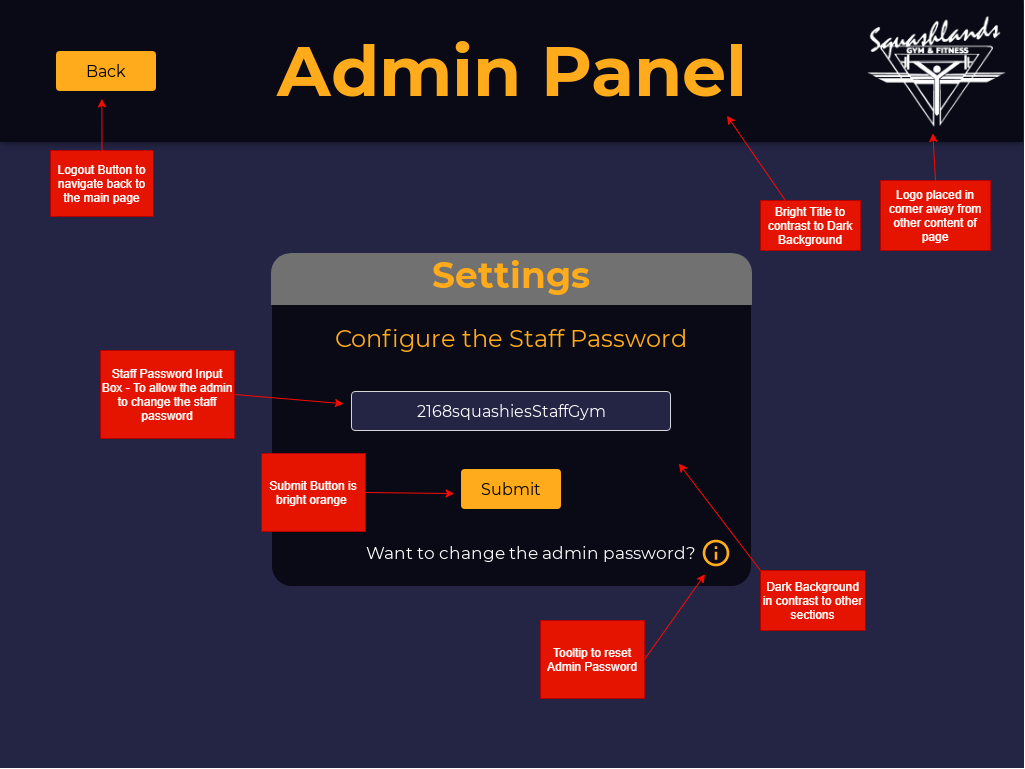
**Screen Design 5:** Admin Panel Login Page



**Screen Design 6:** Admin Panel Page



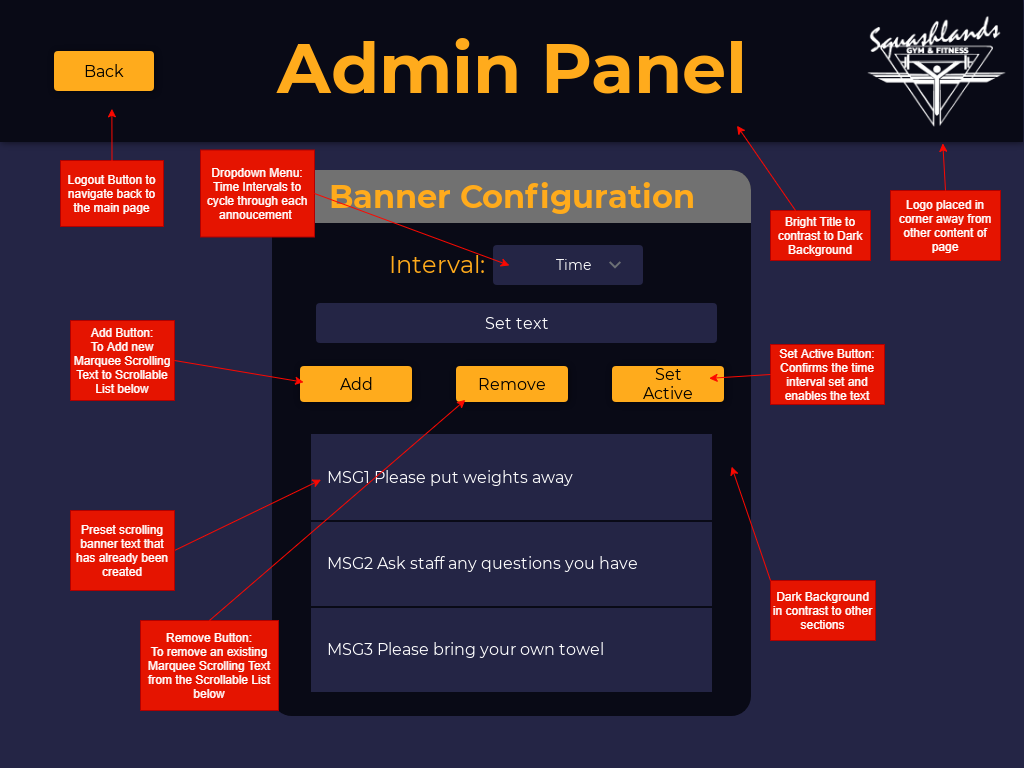
**Screen Design 7:** IP Configuration Page



**Screen Design 8:** Admin Settings Page



**Screen Design 9:** Announcements Page



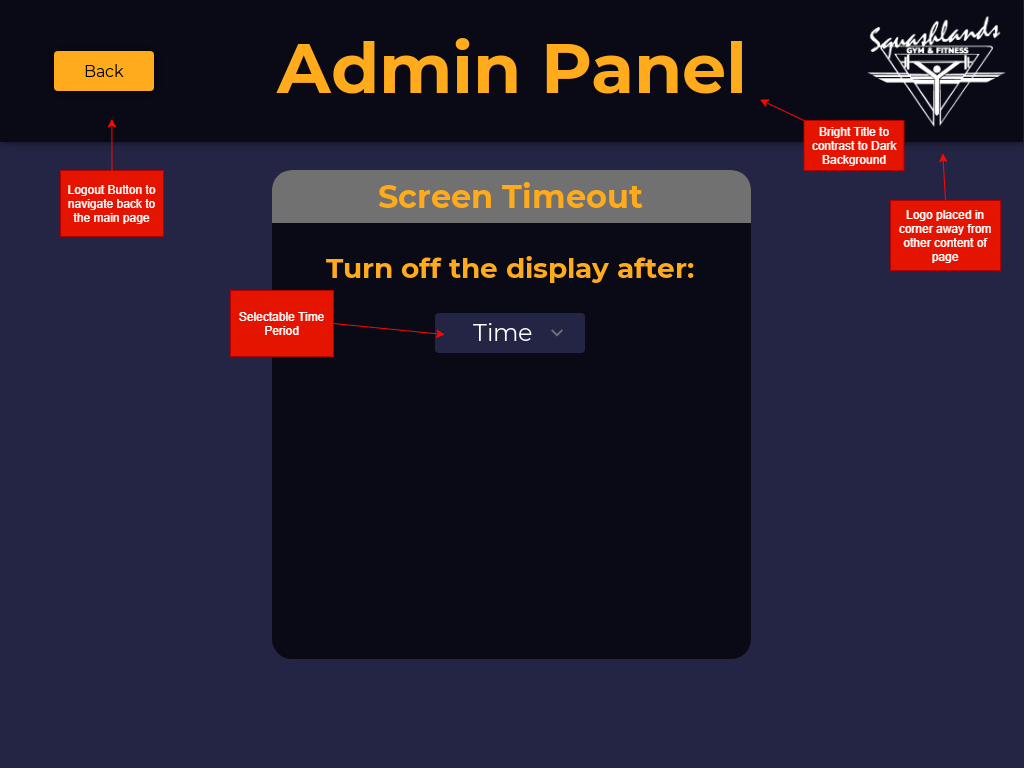
**Screen Design 10:** Marquee Scrolling Text Page



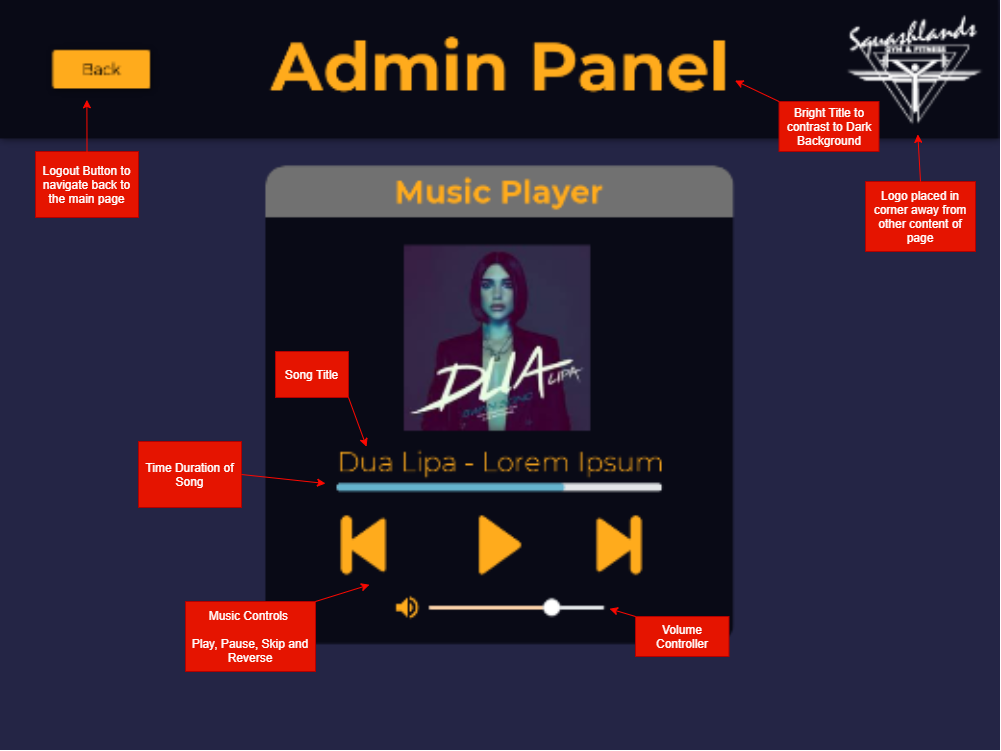
**Screen Design 11:** Theme Changer



**Screen Design 12:** Statistics Page



**Screen Design 13:** Screen Timeout Page



**Screen Design 14:** Admin – Music Player Page

# Test Plan

## Features/Use Cases to be Tested

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Features/Use cases to be tested** | **Types of Testing** | **Pass-Fail Criteria** | **Personnel** | **When and Where** | **Training** | **Risk** | **Contingency** |
| Add Song to Queue | Design Requirement Testing | If a song can be added through the search function from the available songs. | Client: Angel  OR  Team Member: Christian | Week 11: Zoom | Not Needed | Present search results and select item | The song can be selected via tapping on it through a list format from responsive string results |
| Removing Song | Acceptance Testing | If the admin can successfully remove the song from the song list | Team Member: Christian | Week 11: Zoom | Needed | Trying to remove a song that is currently playing | Provide a toast or pop up message saying the action is not allowed and prevent it |
| Request a New Song | Acceptance Testing | User can fill in a request form which the data is sent back to the client for review | Team Member: Sam | Week 11: Zoom | Not Needed | Data presented in a specific format | Data presented in a spreadsheet form that could be more visible |
| Access Admin Panel | Acceptance Testing | If the user can successfully access the admin panel and work with all features available. | Team Member: Kyle | Week 11: Zoom | Not Needed | Incorrect password or forgotten | Perhaps provide a way to change or reset the password |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Changing Banner Message | Acceptance Testing | If the user can successfully change the text on the marquee scrolling banner. | Team Member: Christian | Week 11: Zoom | Not Needed | Adding and removing Banner Message | When adding a message, a text field appears allowing them to change the banner text |
| Changing Announcement | Acceptance Testing | If the user can select an announcement and it. | Team Member: Christian and Sam | Week 11: Zoom | Not Needed | Adding and remove announcements | Announcements can be accessed through a directory |

## Candidate Test Cases/Test Data

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case/Feature** | Add Song to Queue | | |
| **Interface Ref:** | Screen Design 3: Main Screen - Song Search | | |
| **Test Purpose** | To Test if the Gym Patrons can successfully select a song from the song search and enter it into the queue within the Squashies Jukebox Application. | | |
| **Expected Results** | See individual test data sets | | |
| **Success/Failure** | Success | | |
| **Test Results** | | | |
|  | **Test 1** | **Test 2** | **Test 3** |
| **Song Name** | David Guetta - Little Bad Girl | Flo Rida – Good Feeling | Ed Sheeran – Cross Me |
| “**Queue Button**” Clicked | Yes | Yes | Yes |
| Entered in “**First Name**” | Yes | Yes | No |
| **Expected Results** | Pass | Pass | Fail |
| **Success/Failure** | Success | Success | Success |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case/Feature** | Request a New Song | | |
| **Interface Ref:** | Screen Design 4: Song Request Form | | |
| **Test Purpose** | To Test if the Gym Patrons can request for a new song to be added to the song list within the Squashies Jukebox Application. | | |
| **Expected Results** | See individual test data sets | | |
| **Success/Failure** | Success | | |
| **Test Results** | | | |
|  | **Test 1** | **Test 2** | **Test 3** |
| Clicked on “**Don’t see what you want?”** | Yes | Yes | Yes |
| **Full Name (\*required)** | Sam Heng | Christian Politis | Kyle Diamond-Squires |
| **Email Address (\*required)** | samheng@gmail.com |  | kyleds@gmail.com |
| **Song Name (\*required)** | Blue | When I’m Gone | Rap God |
| **Artist (\*required)** | Eiffel 65 | Eminem | Eminem |
| **Year (optional)** | 2010 |  |  |
| **Expected Results** | Pass | Fail | Pass |
| **Success/Failure** | Success | Success | Success |

# Conclusion

This systems analysis and design document highlight our understanding of the project. After discussing the background information on the project. We list the required system functionalities, outlining the functional and nonfunctional requirements of the system. Our team lays out the risks and constraints of the project before discussing design considerations and showing architecture diagrams of the application.

Our team's detailed system design highlights the actors involved in the applications use and delegated actors to each of our listed use cases. Then displaying sequence diagrams for activities and an entity-relationship diagram of the system. Screen designs from our application prototype are presented before identifying features and test cases that will be tested.

Our team believes we understand the project and have a clear layout of the path ahead. We are aware of the risks and constraints involved with the undertaking of the project but believe we will be able to combat them. With this document being a proof concept and allowing the team to make further progress in development, our team is capable of creating a fully functional jukebox application.

The Squashies Jukebox Application will be a valuable business tool for Squashlands Gym. After the submission of this Systems Analysis and Design document, our focus moves towards coding. We will be coding and testing functionalities individually and also collaboratively. Working consistently on this project over the next 4 weeks will ensure a fully-fledged mobile application for Squashlands Gym.