## Rythmix

Group 2: Pranav Gupta | Atharva Tendle | Ibra Kurti | Kaustubh Gupta

**Software Requirements Specification Document**

### Version: (2.0) Date: (09/15/2017)

**Table of Contents**

1. Introduction
   1. Purpose
   2. Scope
   3. Definitions, Acronyms, and Abbreviations
   4. Overview
2. The Overall Description
   1. Product Perspective
      1. System Interfaces
      2. Interfaces
      3. Hardware Interfaces
      4. Software Interfaces
      5. Communications Interfaces
      6. Memory Constraints
      7. Site Adaptation Requirements
   2. Product Functions
   3. User Characteristics
   4. Constraints
   5. Assumptions and Dependencies
   6. Apportioning of Requirements
3. Specific Requirements
   1. External interfaces
   2. Functions
   3. Performance Requirements
   4. Logical Database Requirements
   5. Design Constraints
      1. Standards Compliance
   6. Software System Attributes
      1. Reliability
      2. Availability
      3. Security
      4. Maintainability
      5. Portability

# Introduction

## Purpose

The purpose of this document is to delineate the functionality, software requirements, design, and features of the Rythmix Application. The intent of this document is to provide a clear understanding of the App’s functionality to the customers. The document also aims to give an unambiguous holistic description of the application which will ameliorate the development process for the developers.

## Scope

The goal of this application is to make Karaoke sessions less of a struggle and more of an enjoyable experience which is accomplished via efficient Karaoke management. The application will allow users to make an account which they can use to invite their friends and host personalized group sessions. The session creation will consist of the user choosing a mode, followed by a theme and then queueing a song of his/her choice.

## Definitions, Acronyms, and Abbreviations

**API (Application Programming Interface)-** API’s are a set of useful tools that aid the software development process. The YouTube API and the Google Maps API are few of the many examples.

**Activity-** An Android activity is an interface that the user can interact with to navigate or use the functions of the app.

**RAM (Random Access Memory)-** Is the temporary memory allocated to the application by the device to run the different processes in the application.

**OS (Operating System)-** It is the main software which binds the hardware and the applications on the device and makes sure that everything runs smoothly.

**Firebase-** Is a technology created by Google which supplements the application development process for developers by providing functionalities like storing and syncing data in real time.

**Android Studio-** Is an IDE (Integrated Development Environment) based on IntelliJ IDEA that makes development process efficient.

**SDK (Software Development Kit)-** It is a collection of software used for developing applications for a specific device or operating system.

**UI (User Interface)-** It is the space where interactions between humans and machines occurs.

## Overview

The rest of this document will contain detailed information on the functionalities of the application. It will be divided into two main sections, the overall description and specific requirements. In the overall description, the main functionalities of the app and the interfaces used will be described whilst in the specific requirements, the focus will be on describing how these functionalities will be implemented in a technical language including database implementation and technologies used to build the application.

# The Overall Description

## Product Perspective

#### System Interfaces

Rythmix uses just one system interface, that is the YouTube API, for displaying a lyric video of the song requested by the user.

#### Interfaces

The application, just like other mobile applications, will use the touchscreen to interact with its users.

#### Hardware Interfaces

The application can be installed and run on any device with an Android operating system 5.0 (Lollipop) or higher.

#### Software Interfaces

This application can be downloaded from the Google Play Store.

#### Communications Interfaces

This application requires the implementation of push notifications via services provided by either the phone carrier or via the web service provider.

#### Memory Constraints

The application will be based on a simple design and will not have high graphic requirements thus, would not require a huge amount of RAM.

#### Site Adaptation Requirements

The application does not require any adaptation to run.

## Product Functions

The application basically creates a karaoke session for a group of friends to use and enjoy. The application creates user profiles and saves them by integrating them into the database (Firebase). The application consists of a phone verification and profile registration. It will consist of a user profile page which will allow them to send invites to friends and manage their information. The karaoke activity will allow users to play their songs via the integrated YouTube API. And finally, there will be a ratings activity which will allow everyone to rate someone's performance.

## User Characteristics

The users of the application need to own a device that runs on Android OS (Lollipop or higher) in order to download it. Users do not need any technical expertise, they just need to love singing. However, to be able to create an event and sing in the Drunk Mode, the user should comply with the legal drinking age of the country they are in. Even though users worldwide will be able to download the application, it will only support the English language, so the user will need to be proficient in it to be able to understand how to operate the app.

## Constraints

The application can be installed and will run on almost all modern devices with an android operating system 5.0 (lollipop) or higher. For running the application, the user also has to have a reliable internet connection to stream lyric videos from the YouTube API and for in application interaction between the users.

## Assumptions and Dependencies

The application will be built using Android Studio, so most of its development depends on it. The rest depends on Firebase, an online platform used as a database to store user data. As long as the application is being run on Lollipop or a higher version, it should work as expected. The application will be available in any country around the world.

## Apportioning of Requirements

Phase 1 should include basic functionalities and Phase 2 will add new functionality like themes, song queueing and push notification.

# Specific Requirements

* 1. **External Interfaces**

This application will not use any other external interfaces.

## Functions

### Phase 1:

* + - 1. Splash Screen:
         1. When the users start the application, they will see a splash screen for about 3 seconds which will display the logo of the application and will automatically switch to the next page after that time has passed. The users can also tap the logo to instantly proceed to the next page and skip the waiting time.
      2. Phone Verification:
         1. The next activity is going to be the phone verification activity which will consist of the logo of the application and two buttons for send code and verify.
         2. The send code button sends a randomly generated code to the phone number entered by the user if the number is valid. The verify button verifies the code and registers the user and takes them to the next activity.
      3. Registration:
         1. If the user is registering for the first time they will be asked to enter a username and full name they wish to use.
         2. If the user enters an username which already exists, the system shall display a message to the user regarding the issue.
      4. User profile:
         1. Once the user registers or logs back in to the system they shall be sent to the user profile page. The user profile page will display an account picture along with the details (username, full name, phone number).
         2. This page will include an edit profile button which allows them to make changes to the username and full name.
         3. This page will also include a create session button which will take them to the next activity.
      5. Create session activity:
         1. The user profile will have a create session button through which the users shall have the option to create a session, name the session and pick the mode.
         2. Mode Selection:

The user will be given a choice of two modes while creating a session

Sober mode:

Sober mode is a competitive mode.

In Phase 2, users that sign up for it, will choose their songs and each user will sing once per round. At the end of each song, the users in the same session will have an option to rate the current singer, and everything will be accumulated at the end, choosing one winner. A victory message will be displayed with the winner’s name at the end of the session.

Drunk mode:

Users legally able to drink can choose to have a session in the Drunk Mode. The application has no way of legally confirming each user’s age, however, a disclaimer will be prompted before starting the session. This mode will in no way be competitive

but rather a casual way for the users to have fun. In Phase 2, there will be a few UI changes to go along with the same.

* + - 1. Final Activity:

### Phase 2:

**3.2.1.6.1** After all the users are done singing their respective songs in their modes, they will be given an option to either 1). Call it a night! Or 2). Create another session activity.

Will include all these changes along with the changes to the drunk and sober mode.

* + - 1. Themes:

The theme changes would consist of one major change which is an UI

Overhaul for example: Halloween theme would have a dark UI with carved pumpkins and small ghosts or Christmas would have a lighter UI with snow and candies or similar design.

* + - * 1. After creating a session activity, the users will be directed to the Themes Selection page where they will be able to select a base theme for the entire session.
        2. The Themes available to the users will be ‘Genres’, ‘Festive’, and ‘No Theme’. The Genres and Festive themes will offer sub-themes that the users will be able to choose from.
        3. The Genres theme will have the sub-themes ‘Rock’, ‘Pop’, ‘Metal’, and ‘Hip-Hop’.
        4. The Festive theme will have the sub-themes ‘Halloween’, ‘Christmas’, ‘New Year’.
        5. If the users do not wish to have a theme for their session, they can choose the ‘No Theme’ option.
      1. Song Selection:
         1. The users will now be able to search for a song that they wish to sing and the app will provide them with a lyric karaoke video of that song. This will be achieved through the use of the YouTube API.
      2. Song queuing:
         1. The users will now be able to queue the songs in the order the users wish to sing and they will be played accordingly.
      3. Rating functionality:
         1. The application will now provide the users with a rating feature which will give them the option to rate other users/friends on their performances.

## Performance Requirements

The animations and transitions within the application should work with no visible lag and the YouTube API used for the karaoke event should have no lag from the apps end. The only possibility of lag or delay should be due to the internet strength of the user’s device.

## Logical Database Requirements

The database shall store and encrypt user data and will be able to retrieve it.

In the Phase 1 there will be 1 predefined song for the users to play with, these will be stored in the database. The Phase 2 however, will allow everyone to queue the songs they wish to play and these songs will be stored under the user profile in the database for later reference and retrieval.

## Design Constraints

The app will be made in Android Studio and will only be supported by Android mobile devices.

## Software System Attributes

#### Reliability

The app should seamlessly be able to play a song through the YouTube API without any bugs. It should also be able to store user data and retrieve it whenever required without any issues.

#### Availability

The users should be able to use the application at their own whim and fancy. In the event that the application crashes due to memory or internet issues, the users should be able restart the app and it should function as expected.

#### Security

All the user data will be stored on Firebase and encrypted so that it cannot be accessed by anyone except the owners and/or the developers.

#### Maintainability

Every activity is different and is linked to the others via methods and thus, adding future functionality like queuing songs and rating users should be easy to implement in Phase 2.

#### Portability

The application will be built on Android Studio and should be available for use on most of the newer android mobile devices. The android version will be Lollipop and above (minimum SDK: 21). It can be installed from the Google Play Store.

## References

1. <https://en.wikipedia.org/wiki/Operating_system>
2. <https://en.wikipedia.org/wiki/Random-access_memory>
3. <https://developer.android.com/guide/components/activities/index.html>
4. <https://en.wikipedia.org/wiki/Application_programming_interface>
5. <https://firebase.google.com/products/database/>
6. <https://developer.android.com/studio/intro/index.html>
7. <https://techterms.com/definition/sdk>
8. <https://en.wikipedia.org/wiki/User_interface>