Software Requirement Specification for inTune (Version 1.0)

Lance Wahlert, Jovanny Frias, Rory Spralls

Table of contents

1) Introduction

This document is a Software Requirement Specification (SRS) for the web based social media application inTune. This document is prepared following IEEE conventions for software requirement specification.

The purpose of this project is to provide a platform to assist musicians in connecting, meeting, and sharing original work with other like-minded musicians. It does this by streamlining the process of meeting other musicians through a few simple steps. The inTune web based app will make it easier for musicians to reach their musical goals.

1.1. Purpose

The purpose of this document is to specify a complete description of the web based application inTune. This document will provide an overall description as well as describe the functionality, external interfaces, functional and performance requirements, and constraints of the application being developed. Therefore, intended reader groups are possible users and investors.

1.2. Scope

This project is intended to make use of today's popular social media platform combined with most musician's inherent need to make new music with new people. There are tons of ways to meet people on the internet, but they are mostly restricted to dating services. This project is an attempt to make an app similar to a dating service, that relies on people's hobbies instead of their love interests.

2) Overall Description

2.1. Product Perspective

This project is intended to bridge the gap between the different members of the music industry. The layout is designed to link the fans to the best local artists, and subsequently, the artists to the best possible local venues. The aim is to create a new platform for up and coming artists to increase their opportunities and grow their fame through our application. Our functionality on top of the social media aspect combines the best of the social media phenomenon and the music industry into one.

2.2. Product Functions

Application features include:

1) Creation of Musician accounts in which users can post video clips, sound clips, and status posts.
2) Creation of Fan accounts in which non-musicians can create an account to follow all of their favorite artists, share all of their favorite posts, post statuses, and have the ability to discover musicians, or other fans that share the same qualities.
3) Creation of "Manager" accounts that can be anyone from music labels, venues, or music festivals. These accounts sole purpose are to promote artists and publicize music events going on locally.
4) Our main function is in our explore page where users can click like or dislike on local musicians. If the user is a fan it will pop up new artists the user might like. If the user is a musician, it will display other musicians that the user might like or potentially want to jam with- maybe to start a new collaboration or band. Finally, if the user is a manager, the explore page will help them find artists based on search criteria to fill venues and promote shows.

2.3. User Characteristics

Musician Features

1) Ability to post up to 20 second video clips for status updates
2) Normal functionality of posting, liking, and re-sharing other fan/musician posts
3) Ability to personal message other users
4) Special function of looking for other musicians in your area based on a filter the user provides, examples could be: those looking to start a band, just jamming out, looking for a feature on a track, etc.

Fan Features

1) Functionality of being able to post(non videos), like, and repost other fans/musician posts.
2) Ability to personally message other fan/musicians (messages to musicians will be labeled as “fan mail” in musician accounts)
3) An explore page in which they can find other fans with the same genre interests, or mutual favorite artists

Manager Features

1) Ability to post promotional videos, and statuses
2) Ability to personal message musicians (will appear as “gig offers,” or something similar in musicians inboxes.)
3) Explore page to find musicians based on a filter provided, examples could be: musicians of this genre, musicians willing to play for free, etc.

2.4. Constraints

Currently our biggest constraint is that this will solely be a web based application. One should be able to access all the features from a browser on mobile, but the exact performance on these devices are unknown.

Secondly, because of the time constraints for this class we will have to implement security devices as they become obvious. Because of this, inTune will start as a beta application as user's data will not be entirely secure at initial release.

2.5. Assumptions and Dependencies

We assume user's understand the mechanics of the explore page based on past successful applications like tinder, or bumble. Left swipe means dislike, right swipe means like.

We assume if a user creates an account upon completion of this stage of inTune that they understand personal data may not be entirely secure based on the constraints of our project.

We assume if a user signs up as a fan they will be using the explore page to find musicians to listen to, and other fans who like similar musicians.

We assume if a user signs up as a musician they will be using the explore page to promote their work to fans, and managers, or they will be looking for other musicians to jam with.

We assume if a user signs up as a manager they will be promoting a venue or artists, and will be using our explore page to find musicians to either promote or help them promote their brand.

Our project is dependent upon musicians wanting to promote their work and/or find other musicians to jam with. Our application is made for and by musicians, so it is up to their accounts to provide good enough content to attract fans and manager accounts.

3) Specific Requirements

3.1. External interface

3.1.1. User Interfaces

The client interface will be a web interfaced designed for use with keyboard and mouse. It will follow the Material design guidelines.

All three types of accounts will have a home, profile, and explore page along with a drop down for notifications on their homepage. The homepage will look similar for all accounts. It will display posts and statuses from all accounts the user follows based on timestamps. The profile page will also look similar across the different types of accounts. It will display information input by the user including: pictures, personal info, statuses and videos.

There will be a messages column on the right side of the page on every page that can be collapsed. It will be the same across all types of profiles except musicians will have 2 extra inboxes for “fan mail”, and “gig offers”.

The explore page will look the same as well except the mechanics will be different. Musicians will not be able to see fan accounts or managers in the explore page, they will only have the opportunity to swipe on other musicians. Fans will be able to see both other fans and musician accounts. Managers will only be able to see musicians.

3.1.2. Hardware Interfaces

The client’s hardware interface will consists of a keyboard and mouse in which all users will use. Communication between client and server will be achieved through wireless internet.

3.1.3. Software Interfaces

- Database- MongoDB
- Client Language - React.js
- Server Application Language - Node.js
- Admin Website - TBD

3.1.4. Communication Interfaces

Data transfer between the client and server will be over wifi connection on the device. All data will be in JSON format.

3.2. Functional Requirements

Communication will happen from the server to the client within a browser from either a wifi connection or cellular data. Since it is a web application, and some data submitted by the user could be considered private, we will be seeking to encrypt with https. Data will be JSON format.

3.2.1 Login Page

- The user shall access the login page from the home page
- The login page will consist of two tabs, one for registration to website and another for login
- The user shall create an account using the a register tab section
- The user will be prompted to enter username, email, password, and whether they are a fan, musician , or a venue on register tab
- The server shall notify user whether account username is already taken or not and prompt to try again
- The user will be prompted for username and password on the login tab
- The server will store the all users in the database regardless of the type of user

3.2.2 User Dashboard

- The user shall be redirected to a starter page on the first sign in into brand new account
- The user shall be prompted with their feed once if its not their first sign in
- The dashboard will consist of a feed , explore , and a profile all navigated through buttons on the navigation bar

3.2.3 Feed

- The user will be shown a feed of all the accounts the user follows in order by date
- The user has the ability to like, share, and comment under each post on the feed.
- The user can make own post of their own that allows for text of 140 char and an image
- The server will save each posts posted in a database for the user

3.2.4 Explore

- The users shown in explore page will be filtered by music style for fans
- The users shown in explore page will be filtered by location and or music style for musicians and venues
- The musicians shown will have a brief 20 second video clip of their choice as an introduction
- The user will have the ability to follow someone or just move on
- The user can direct message any other user they find

3.2.5 Profile

- The user has the ability to change profile picture
- The user has the ability to change their settings such as name, username, password, email, etc.
- The user has the ability to add a banner on profile page
- The profile page will show the name, bio, music genres they like, friends list, and feed of posts by user.
- Only musicians can post 20 second video clips.

3.2.6 Navigation Bar

- The navigation bar will have the logo on the left that always takes you back to user dashboard
- The navigation bar will have 3 routes on the right that take you to the explore page, profile, or direct messages.

3.2.7 Messages

- Each user has the ability to message any other user
- The server will store all direct messages of the user in the database
- The user shall be able to delete chats with other users
- Musicians will have separate chats between musicians and fans.
- The user can send images and text to other user
- The user shall be able to create new chats with any other user

3.3. Performance Requirements-TBD

The following requirements detail the user's performance needs:

- The UI must have consistent theme
- Each page must load in under two seconds
- Server-Client interactions must happen in under five seconds

3.4. Design Constraints

- Database- MongoDB
- Client Language - React.js
- Server Application Language - Node.js
- Admin Website - TBD

3.5. Other Requirements

The beta for inTune will have limited security and availability as it will be a web based application. We require that our first user's don't input any information they consider sensitive, and that they only try to access our application through a browser on a desktop computer.

4) Appendix

MongoDB- stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time

Node.js- is an open-source, cross-platform, JavaScript run-time environment that executes JavaScript code outside of a browser.

React.js- is a JavaScript library for building user interfaces.