SW Engineering CSC 648/848 - FALL 2020 - Team 5 - Milestone 1

Project String

Team Lead & Github Master: Jainam Shah - jshah3@mail.sfsu.edu

Frontend lead: Alfredo Diaz

Backend lead: Warren Singh

Frontend Developer: Xuanjun (Eric) Chen

Frontend Developer: Leonid Novoselov

Backend Developer: Ritesh Panta

Milestone/Version	Date
Milestone 1 v1.1	10/01/2020

October 2, 2020

Contents

1 Executive Summary

String is a one stop platform for local bands, individual musicians, and listeners. Bands which are ready to perform and have upcoming performances can list upcoming events on their profile page and be found by a bigger audience. String is also aiming to change the landscape for listeners who like live music and how they spend their leisure time. Our platform being specific to local bands lets tourists and locals discover music and genres directly from the platform. They won't have to go to the bar and hear a random musician or a band anymore. With String they can pick and choose who and when they want to listen to in person and discover new bars, restaurants or cuisines while listening to the music they enjoy.

Our biggest competitors are reverbnation and Bandlab. These platforms offer lots of features but are generalised, and built for remote collaboration and exposure. We are also able to offer more specialized services to musicians aiming for in-person rehearsals, collaborations, and performances, aiming to give them the ability to do everything at a singular place. Local artists gain huge value from String.

According to IFPI(https://www.ifpi.org/our-industry/industry-data/) the total value of the recording industry in 2019 was US$20.2bn and reported an 8.2% global growth (that's +18.9% for Latin america alone) - As we explore and experiment with monetization we look at paid discovery options for bands, and brands as possible consumption based options that bands, music equipment brands and even restaurant joints could access. We are also exploring subscription based options where musicians and bands would pay a recurrent amount each month to keep track of their audience, this initially can be simple metrics on the people looking at their profiles but also later can be grown to integrate to the rest of their audiences through streaming platforms hosting their music.

There is no beating the benefit of in person connection, nor the value that a specfic location can bring. String capitalizes on that, and offers a unique new value as a result.

2 Main Use Cases

Title	A musician looking for a band to join
Actor	John Lennon
Action	John is a guitarist and singer from San Francisco just arrived in LA and is looking for a band who is open to playing the same type of music with him. Due to corona there are no live social gatherings to meet other musicians in the area. John signs up on String, he is then able to upload some of his content and post a request to join a band. Within minutes he has bands messaging him to send him samples and test out team dynamics. Within a week he already made new friends and signed into a band. Bands add him to their band page and they start playing music together.

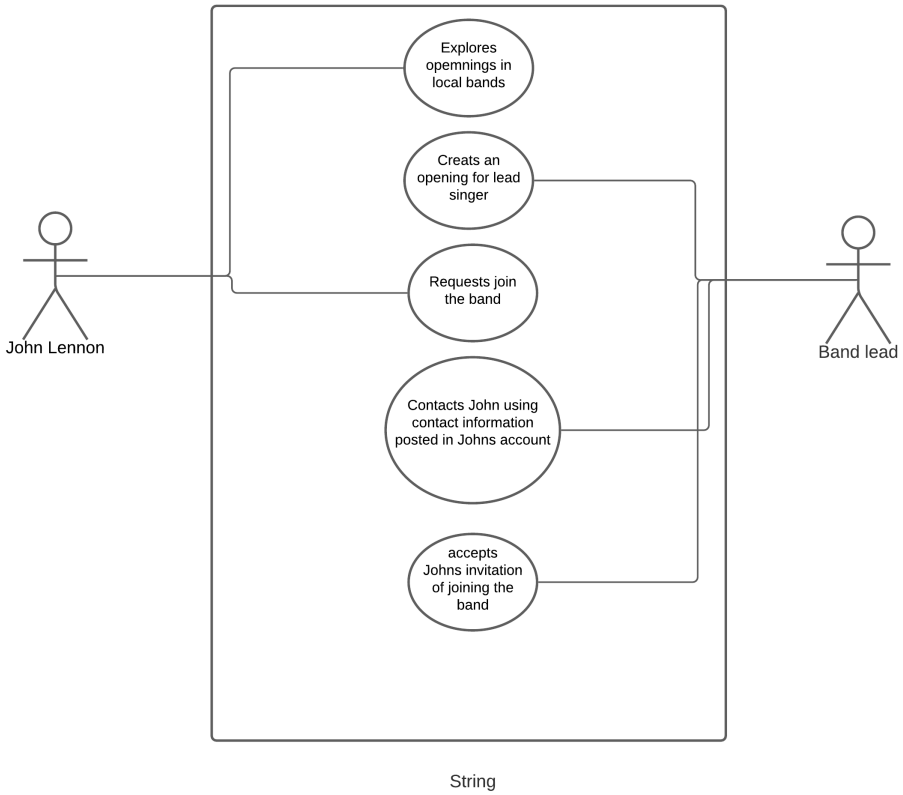

Figure 1: A musician looking for a band to join

Title	A band looking for musicians
Actor	Riot (a local band)
Action	Riot is one of the local bands in the city. Recently this band has been getting a lot of suggestions about adding an acoustic guitarist to their group. They are confused on how they should post this opening for free. The band lead discovers String. He makes an account and adds the band in his account and asks his other band members to join using their String accounts. Then they create an availability which is visible to other members on the platform and can request to join. The band reviews the applicants and contacts them with the contact information provided and if they see the applicant fit, then they add them to their band. They also see the feature of adding videos to their profile and so they add their previous recordings to the platform.

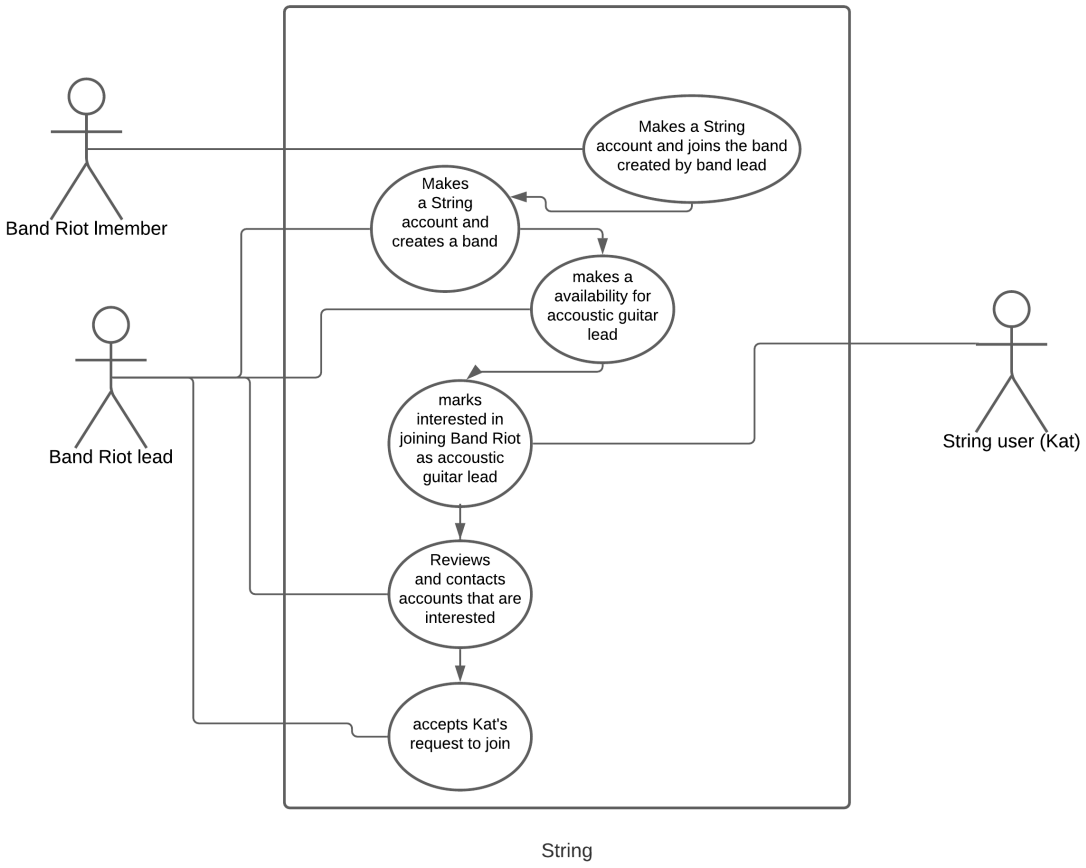

Figure 2: A band looking for musicians

Title	A band manager looking to have one place to manage/display all bands he manages
Actor	Aron
Action	Aron is a band manager who manages two local bands. He is doing this as a passion and not as his occupation. He wants a place where he can easily manage these bands and upload their materials for publicity. Aron comes to String and registers as a user. On the website he creates two band pages and asks all the players to register as users on String and join his request. He then uploads all the previous band performance videos on the platform and uses the band profile link to share via social media.

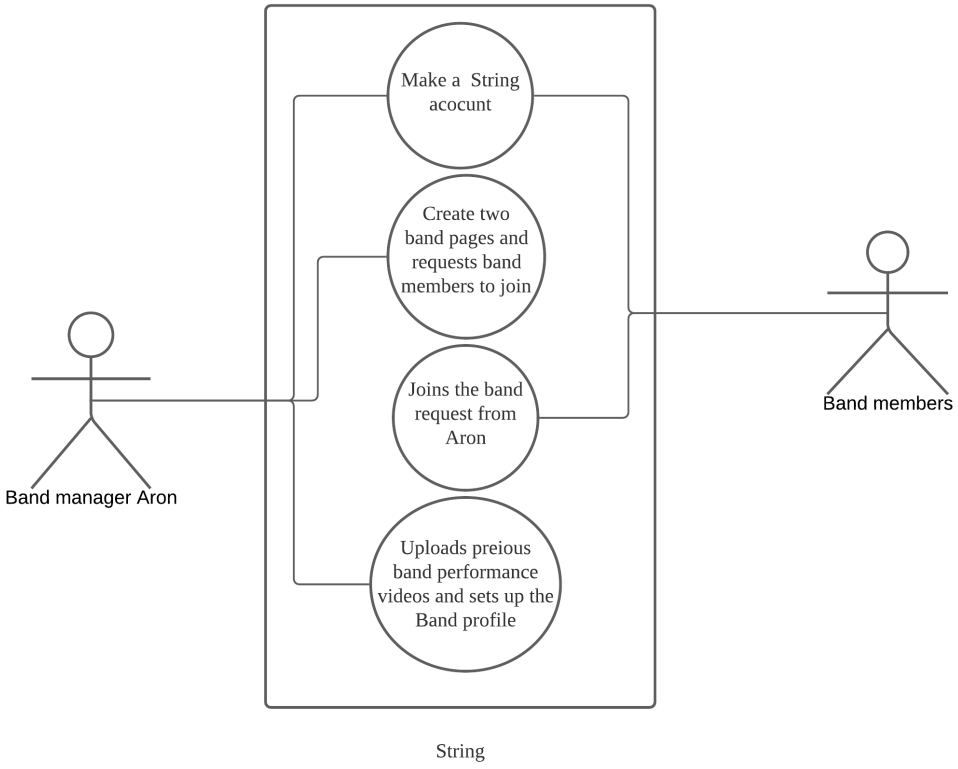

Figure 3: One place to manage and display

Title	Looking to discover more music in their area (as a listener)
Actor	Alan
Action	Alan is a local to the area, and enjoys discovering and listening to live music. However, he does not always enjoy staying out late at bars where local music performances usually happen. He searches online for local bands and visits String platform. He is able to see a list of local bands and their previous performances. He enjoys looking for local bands that have open rehearsals during the day (as indicated on their band profile description), showing up, meeting local musicians, listening to new songs, and helping to give the band feedback on new material.

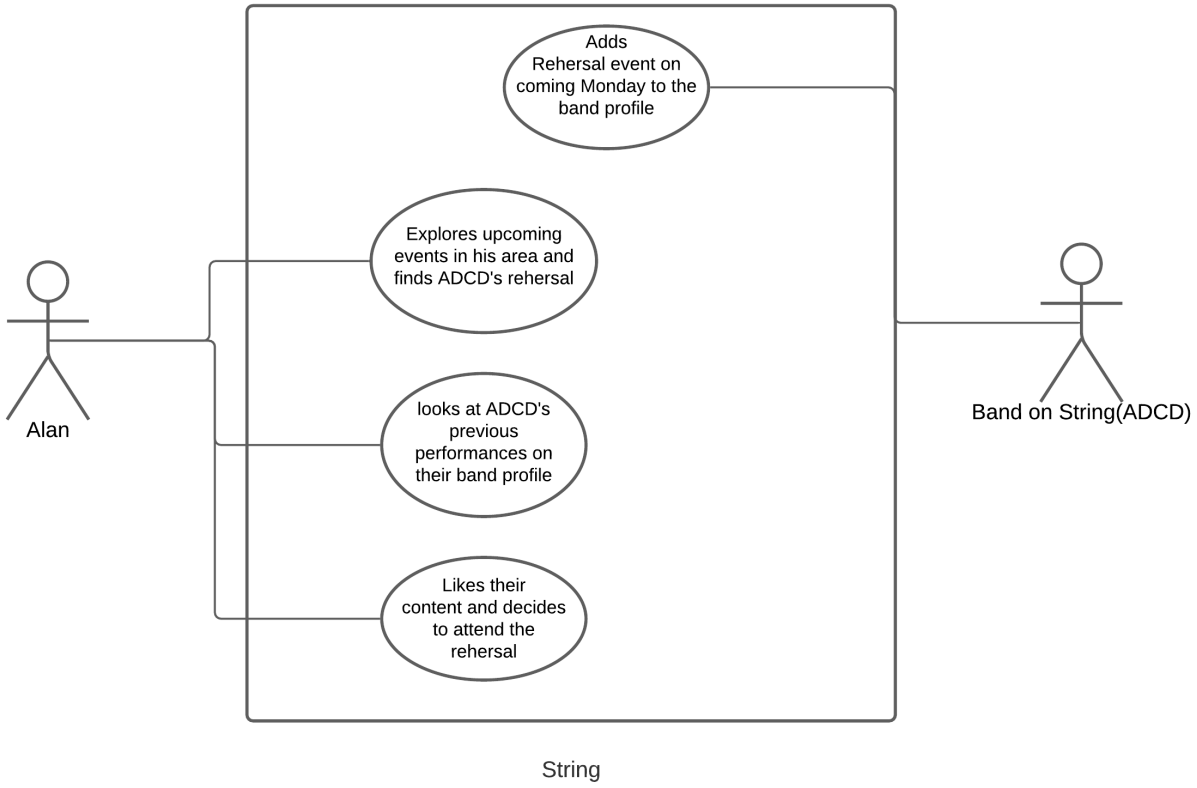

Figure 4: Looking to discover more music in their area

Title	Looking to sell products to local bands (looking for contact info)
Actor	RockyRoad
Action	A small store named RockyRoad is getting shipments of musical instruments and they are struggling with selling because of COVID-19, so they decide to expand into personal outreaching. They want to contact bands in their local area that would be interested in buying instruments from them. The manager looks at the bands in the local area via String and collects their contact information from the band profile page and contacts them about the store's offers. He is able to find some new customers looking to upgrade their instruments and he is able to sell his products to them.

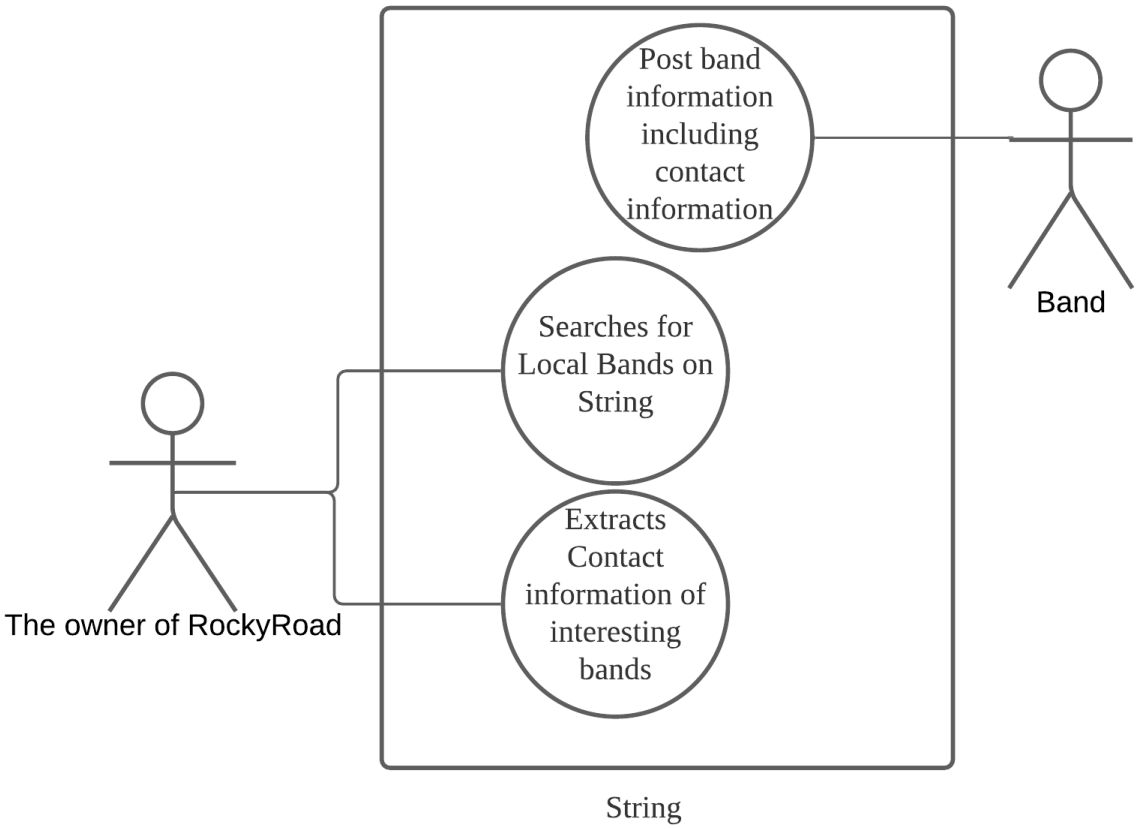

Figure 5: Looking to sell products to local bands

Title	Looking for bands to play
Actor	Angello
Action	Angello has a commercial property and he wants to rent it out . He also has a little dinner restaurant "At Angello's" that he wants to have live Jazz music at. He looks through local groups in the area. After careful consideration he gets in contact with 10 bands through contact information that he found on String. Lucky for him, local band Riot was looking for a place where they could store their equipment and practice so they rented Angello's commercial property. Later that day 3 Jazz bands got back to Angello and agreed to play in his restaurant. Angello schedules each band for Friday, Saturday and Sunday - all different bands and musicians. He is very happy with the result.

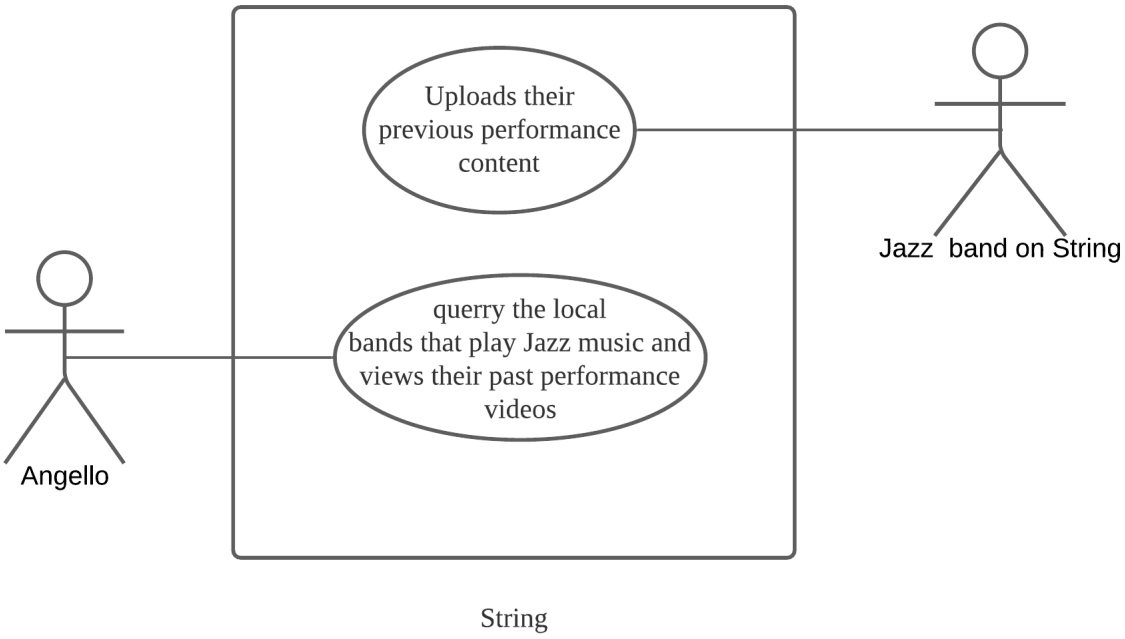

Figure 6: Looking for bands to play

3 List of main data items and entities

1. Registered user: A person who is a registered user on our platform and is logged into their account.
2. Guest: A person who has not made an account but is just viewing pages on our platform.
3. Administrator: A person who has access over all information, items and entities on the platform. They are the ones who will manage inappropriate content.
4. Band: A band can be created by a registered user. It is an entity that other registered users can join if Band Admin permits it.
5. Band Admin: The person who created the band using their registered account, or anyone promoted to that role by an existing Band Admin
6. Band Member: A registered user who joins a Band, with a Band Admin's permission.
7. Band post: Image or video uploaded by bands
8. Invitation: Sent by a registered user to join a band in response to availability post
9. Event Entry: A band can create an entity known as an event in order to provide information on an upcoming rehearsal, or performance - with sections for public and private facing information and Set List.
10. Set List: A list that a band can create, tied to an event, detailing the song list (and possibly order) that the group will be performing for said event
11. Event List: A registered user who is a part of many bands can create an event list, which will list in one place all upcoming events for all the bands that they belong to
12. Repertoire (aka 'Rep') Entry: A song which the band can play, including details such as run time, genre, links to past video, etc.
13. Repertoire (aka 'Rep') List: A list that a band can create which holds the names, notes on, etc. of the current songs the band is rehearsing and can play.

4 Functional Requirements

4.1 Registered Users

1. Registered Users shall be able to contact the administrator to get help.
2. Registered Users shall be able to access and modify their registered user profiles.
3. Registered User profiles shall have an optionally guest-viewable field indicating if the user is a musician or not.
4. Registered User profiles shall have an optionally guest-viewable field to specify which instrument(s) they play.
5. Registered User profiles shall have an optionally guest-viewable field displaying a primary method of contact.
6. Registered User profiles shall have an optionally guest-viewable field for linking a user-provided website(s).
7. Registered users shall be able to request to join a band.
8. Registered users shall be able to create band(s).
9. Registered users shall be able to accept invitations to be added to a band.
10. Registered User profiles shall have an optionally guest-viewable field indicating the current geographical location of the user.
11. Registered Users shall be able to see all content viewable by guests.
12. Registered Users shall be able to view other member information which like showing time, showing place update if it's public.
13. Registered users shall be able to see all and any content provided by them on the String platform.
14. Registered users shall be able to edit all and any content provided by them on the String platform.
15. Registered users shall be able to delete all and any content provided by them on the String platform.
16. Registered users shall be able to see all guest-viewable fields and contents.

4.2 Guest Users (Guests)

1. Guests shall be able to see all guest-viewable fields.
2. Guests shall be able to see all guest-viewable contents.
3. When viewing the website, Guests shall be able to access a landing page which displays a succinct overview of what services String offers.
4. Guests shall be able to register an account with a valid email address.
5. Guests shall be able to register an account with a valid password.
6. Guests shall verify their email address when registering.
7. Guests shall be able to contact the administrator to get help at any time.
8. Guests shall be able to preview specific music previews from bands and individual musicians on their respective pages.
9. Guests shall be prompted to input email address when accessing certain features
10. Guests shall be able to exit the email prompt and continue without inputting address if they choose to do so.
11. Guests shall be able to access contact information of any given account that has made that information public.
12. Guests shall be able to search for bands in a specified geographic area.

4.3 Bands

1. Band's creator will be the Band administrator by default.
2. Bands shall have an optionally public field for indicating if they are adding any members.
3. Bands shall have a minimum of one member.
4. Bands which have no member shall be deleted.
5. Bands shall be able to report other bands or members to the administrator in case of copyright issues.
6. All content uploaded by Bands shall be public.
7. Band shall have a compulsory public field indicating the geographical location that the band is located in

4.4 Band Members

1. Band members should be registered users.
2. Band members shall be able to be a part of more than one band.
3. Band members shall have an optionality to display band related posts in band's guest viewable field.
4. Band members shall have an optionality to display their vacancy status.
5. Band members shall have the option to leave the band.
6. Band members of one band shall be able to invite new members which will be sent to be reviewed by Band Administrator.
7. Band members shall be able to send a text-based message attached to any invitation they send.
8. Band members of a band shall be able to toggle their status within a band between the following states: active, reserve, as available.
9. Band members shall have access to view all the posts related to the band.
10. Band members shall have access to comment on all the posts related to the band.
11. Band members shall be able to change their field or position in the band.

4.5 Band Administrators

1. Band administrators shall be able to recruit member(s) for their band.
2. Band administrators shall be able to add details of their next rehearsal or performance.
3. Band administrators shall ban band members in case of violating band policy.
4. Band administrators may add other members from another band.
5. Band administrators shall have access to assign any band members as the leader role.
6. Band administrators shall have the opetion to reject invitations from other users.
7. Band administrators shall have access to view all the band posts.
8. Band administrators shall have access to edit all the band posts.
9. Band administrators shall have access to delete all the band posts.

10. Band administrators shall be able to create an availability post which can mark down schedules for collaboration.

4.6 Administrators

1. Administrators shall have access to add registered users in the band.
2. Administrators shall have access to delete registered users.
3. Administrators shall have access to edit any content posted by a band.
4. Administrators shall have access to delete any content posted by a band.
5. Administrators shall have access to edit any content posted by a registered user.
6. Administrators shall have access to delete any content posted by a registered user.
7. Administrators shall have access to remove a band.
8. Administrators shall have access to edit any content posted by a guest user.
9. Administrators shall have access to delete any content posted by a guest user.
10. Administrators shall have access to edit any content posted by a band member.
11. Administrators shall have access to delete any content posted by a band member.
12. Administrator shall have access to modify any band posts.
13. Administrator shall have access to modify any event lists.
14. Administrator shall have access to modify any set lists.

5 Non-functional Requirements

- Compatibility

  1. Website must be compatible with Safari 12.0+, Chrome 81+, Firefox 70+
  2. Website should be compatible on mobile with a minimum width of 300px and desktop with any dimensions.

- Security

  1. All passwords should be hashed using bcrypt algorithm before storing it in the database.
  2. Passwords set by users should be at least 8 characters long.
  3. Passwords set by users should be at most 21 characters long.
  4. Passwords should contain at least one uppercase letter.
  5. Passwords should contain at least one number.
  6. Passwords should not be logged in the production build.
  7. Any private keys including but not limited to Mysql database password should be pushed to github.
  8. Website should always have SSL encryption enabled.
  9. Login api calls shall be limited to 20 calls per ip each hour.
  10. Sign up api calls shall be limited to 20 calls per ip each hour.
  11. Application shall be secured from sql injection attacks.
  12. Application shall be secured against HTTP Parameter Pollution attacks.

- Deployment

  1. Application should be deployed on AWS EC2 instance.
  2. Only the main branch should be deployed on the AWS server instance.
  3. Any and all deployment should be only handled by the deployment master.
  4. The website should be able to host at least 300 users in the same hosted region as the EC2 instance.

- Legal

1. Privacy policy and terms and conditions of use should be accepted by the user before creating an account on the platform.
2. The user generated data collected by the website shall not be shared with any third party without user consent.

- Documentation

1. All api shall be well documented on a high level usage view.
2. All api shall be well documented in detail with implementation details.

- UI/UX

1. All pages on the website should have consistent design and color.
2. Color combination for the website should be calming and should not cause strain on eyes.
3. All text in the page should be readable with a minimum font size of 8px and should have contrasting font than the background color.
4. All fonts used shall be web safe fonts.
5. Users should be able to navigate between pages.

- Coding standards

1. VS code version 10.8+ should be used as the code editor for the project.
2. VS code extension Prettier version 2.2+ should be enabled ands used to format all documents before pushing any commits to github.
3. Code shall have proper indentation and spacing.
4. Code should have error handling to prevent failure.
5. Functions should have at least one comment explaining its functionality at the beginning of the function.

- Environment

1. Node version 12-13 and npm version 6.14.8 should be used to develop the application.

- Internalization

1. English should be the language used in the site.
2. All locations on the website should be validated to be in the United States.

3. Dollar should be the only currency accepted in all tractions on the website.
4. Distance should be measured in Miles.
5. Time should be measured only in HST, AKDT, PDT, MST, MDT, CDT and EDT time zones.

- Content limits/Data Integrity

1. Videos uploaded by users should be less than 20 frames per second.
2. Videos uploaded by users should have length less than 120 seconds.
3. Videos uploaded by users should have file size less than 200mb.
4. Videos should have .mp4 or .mpv extension.
5. Images uploaded by users should have file size less than 8mb.
6. Images with .jpg, .jpeg or .png extensions only are accepted

6 Competitive Analysis

Company Feature	AMY	Bandcamp	BandLab	Drooble	Reverbnation	String
Multiple group/event management	-	-	-	-	-	+
Repertoire list	-	-	-	-	-	+
Event public/private details	+	-	-	-	-	++

Side-by-side Comparison

Company	Strengths	Weaknesses
Amy	• Can find or list related services • Multiple profile types available • Can organize jam sessions	• Tiny social media reach • Complicated interface • Does not come up on search engines consistently for related searches • Not prominent (no discussion on internet) • Limits videos to 20 seconds(!)
Bandcamp	• Easy to contact and get together w/ other bands • Bands actually get paid • Vinyl Purchases and direct digital downloads	• Cannot download songs directly to mobile app • Most users are artists, not fans
BandLab	• Great music creation tools • Strong community of DIY members	• Prominent latency issues
Drooble	• Incentivizes participation by offering money for reviews of music, enabling feedback for other users • Rewards social currency gathering by letting musicians purchase promotion tool access • Provides a goal for participants by running a platform specific popular music chart	• Payment rises with services used (no clear cost) • No networking except for online only • Users mostly in Eastern Europe
Reverbnation	• The most features, including music hubs, vlogs, collabs, and promotions • Large fan reach and sales potential • Easy distribution of music to major digital platforms	• Contests are just a time sink for artists

Table 1: Competitor Strengths and Weaknesses

Company	Pricing	Social Media	Onboarding Experience
Amy	• Free	• Facebook, Twitter, Instagram, Youtube	Facebook/Google Integration
Bandcamp	• Free (revenue share on sales, 10-15%), optional paid tier (10 - 50 USD/mo.)	• Facebook, Twitter, Instagram	• Allows for own domain
BandLab	• Free	• Facebook, Twitter, Instagram	• A bit involved (user categories for role, activity, and music genres)
Drooble	• Variable according to services purchased	• Facebook, Instagram	• Email or Google sign in
Reverbnation	• Free, or 13 USD/mo., or 20 USD/mo.	• Facebook, Instagram, Twitter, Youtube	• Intro tour, Upload prev. work, complete user profile

Table 2: Competitor Pricing, Social Media, and Onboarding Experience Comparisons

The main focus of String is on musicians in a specific local geography who already (or who want to) play in groups together, either for fun, or for paid gigs. Therefore, our feature list is rich in management tools toward that end. None of the other competitors that we examined had the feature of easy multiple group/event management, or a repertoire list; they focus on networking and music making over the internet instead. Thus, our focus on live music and local artists yields a tool which is superior in these ends, and we believe that our tools will help our intended users have a better experience than our competitors.

7 High-level System Architecture and Technologies Used

- Server Host: AWS EC2 t2.micro 1 vCPU 1GB RAM
- Operating System: Ubuntu 16.04 Server
- Database: MySQL
- Web Server: Caddy v1
- Server-Side Language: JS (Node.js)
- Additional Technologies: Web Framework: Express.js, React.js, Redux
- IDE: VS Code
- Web Analytics: Google Analytics
- SSL Cert: Lets Encrypt

8 Team

- Team Lead and Github Master: Jainam Shah
  - Worked on setting up team meetings and making the agenda for the meeting.
  - Maintained communication with team members to execute appointed tasks.
  - Worked to create the content of the M1 document
    - Formed the non functional requirements.
    - Helped create use case diagrams.
    - Helped introduce data items and entities
    - Helped developing the use cases
- Frontend Lead: Alfredo Diaz
  - Attended all team meetings on time and actively responded on slack.
  - Worked on setting up Trello dashboard and getting team members added to the board.
  - Worked to create the content of the M1 document
    - Researched the competitors and took responsibility for the competitor analysis section.
    - Helped create use case diagrams.
    - Worked on the executive summary to make it more compelling
    - Helped developing the use cases
- Frontend Developer: Leonid Novoselov
  - Attended all team meetings on time and actively responded on slack.
  - Worked to create the content of the M1 document
    - Helped create the functional requirements.
    - Worked on the executive summary to make it more compelling
    - Helped developing the use cases
- Frontend Developer: Eric Chen
  - Attended all team meetings and actively responded on slack.
  - Worked to create the content of the M1 document.
    - Helped create use case diagrams.

        * Helped developing the use cases

- Backend Lead and Documentation Master: Warren Singh
    - Attended all team meetings on time and actively responded on slack.
    - Worked on compiling and updating the M1 document using LaTeX.
    - Worked to create the content of the M1 document
        * Helped create the functional requirements.
        * Helped with the competitive analysis section
        * Helped introduce new data items and entities
        * Helped developing the use cases
- Backend Developer and Database Architect: Ritesh Panta
    - Attended all team meetings on time and actively responded on slack.
    - Worked to create the content of the M1 document
        * Helped create the functional requirements.
        * Helped introduce new data items and entities
        * Helped developing the use cases

9 Checklist

- Team found a time slot to meet outside of the class: **DONE**
- Github master chosen: **DONE**
- Team decided and agreed together on using the listed SW tools and deployment server: **DONE**
- Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing: **DONE**
- Team lead ensured that all team members read the final M1 and agree/ understand it before submission: **DONE**
- Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.): **DONE**