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
Milestone 1 v2.1	10/14/2020

October 15, 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.

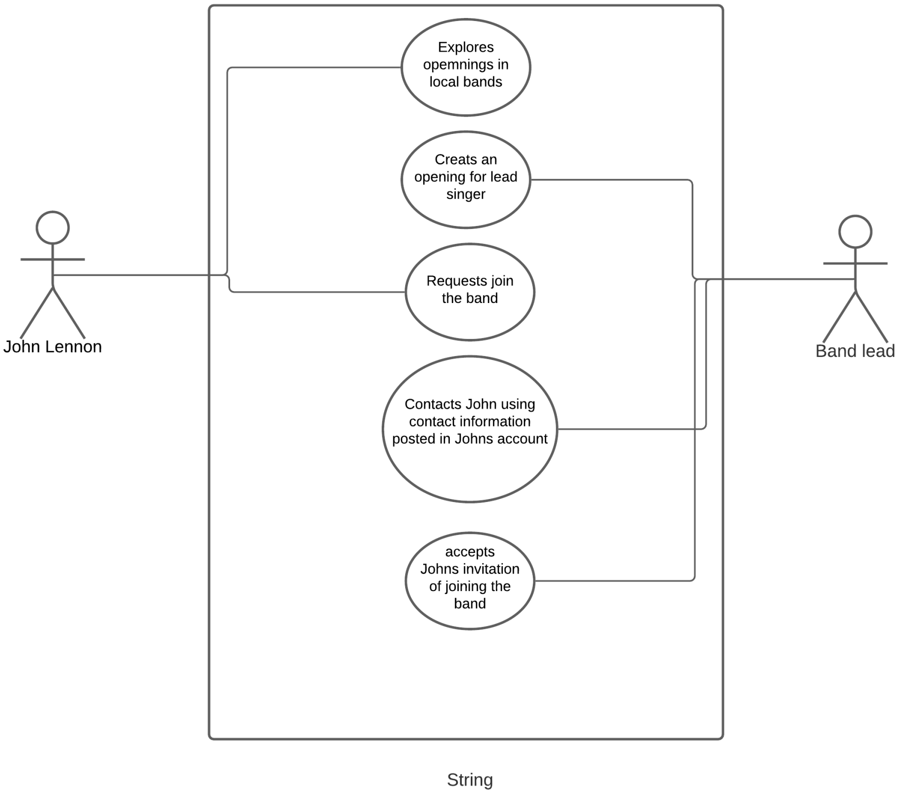
Explores opemnings in local bands
Creats an opening for lead singer
Requests join the band
Contacts John using contact information posted in Johns account
accepts Johns invitation of joining the band
John Lennon
Band lead
String

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.

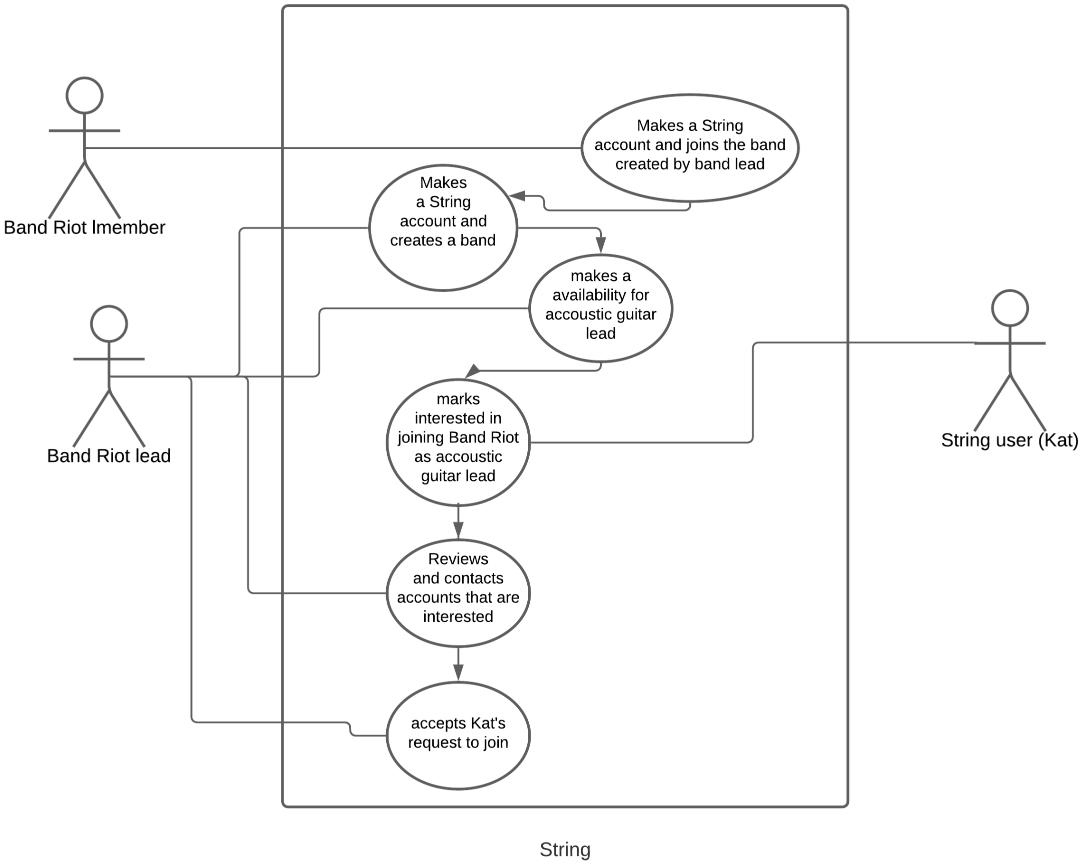
Band Riot lmember
Band Riot lead
Makes a String account and joins the band created by band lead
Makes a String account and creates a band
makes a availability for accoustic guitar lead
marks interested in joining Band Riot as accoustic guitar lead
Reviews and contacts accounts that are interested
accepts Kat's request to join
String user (Kat)
String

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.

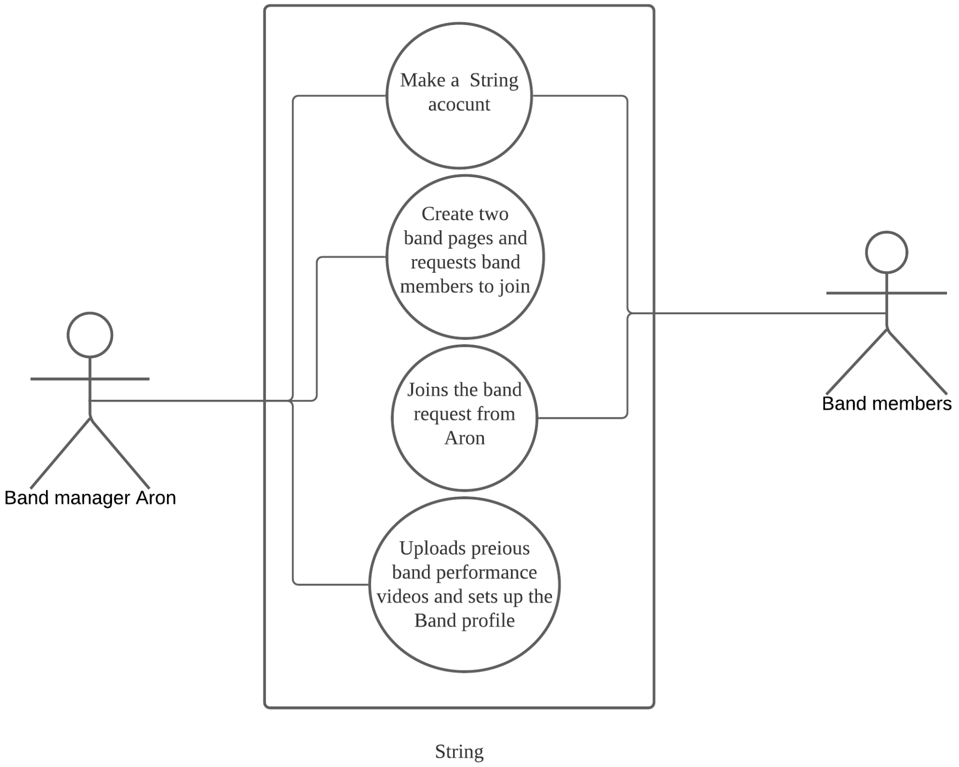
Make a String acocunt
Create two band pages and requests band members to join
Joins the band request from Aron
Uploads preious band performance videos and sets up the Band profile
Band manager Aron
Band members
String

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.

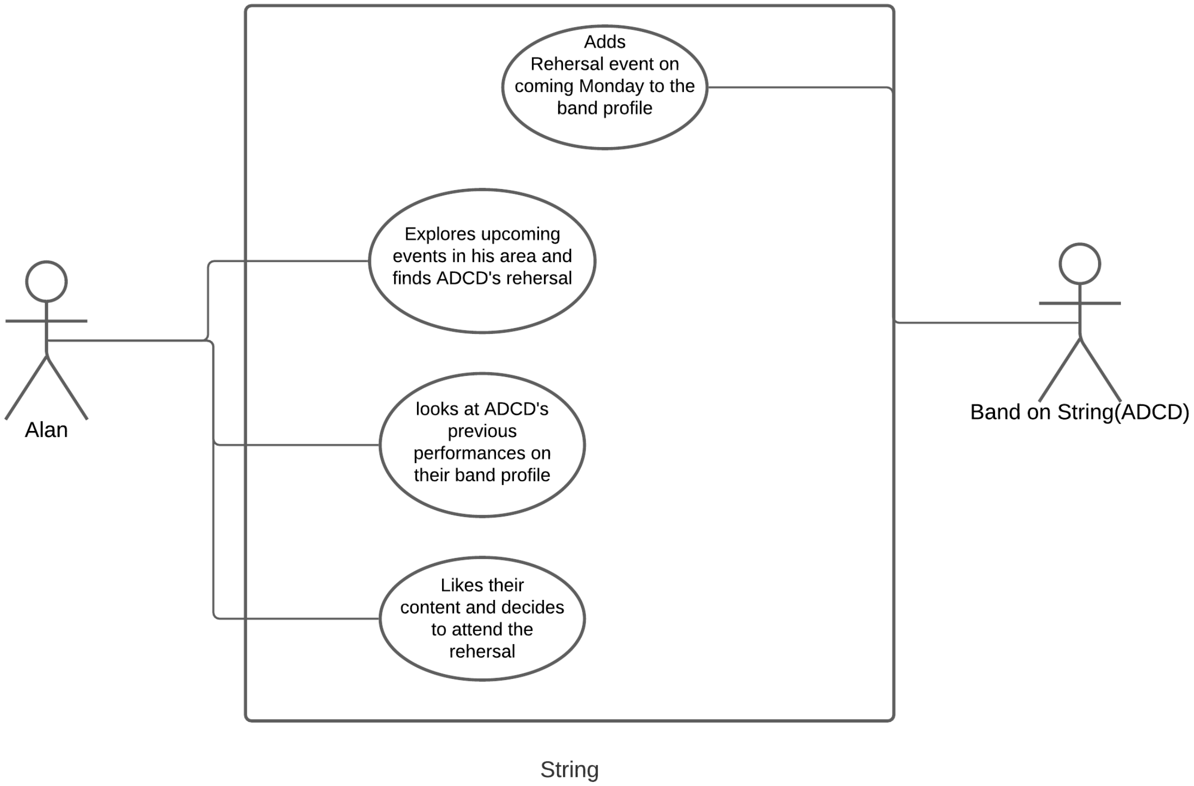
Adds Rehersal event on coming Monday to the band profile
Explores upcoming events in his area and finds ADCD's rehersal
looks at ADCD's previous performances on their band profile
Likes their content and decides to attend the rehersal
Alan
Band on String(ADCD)
String

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.

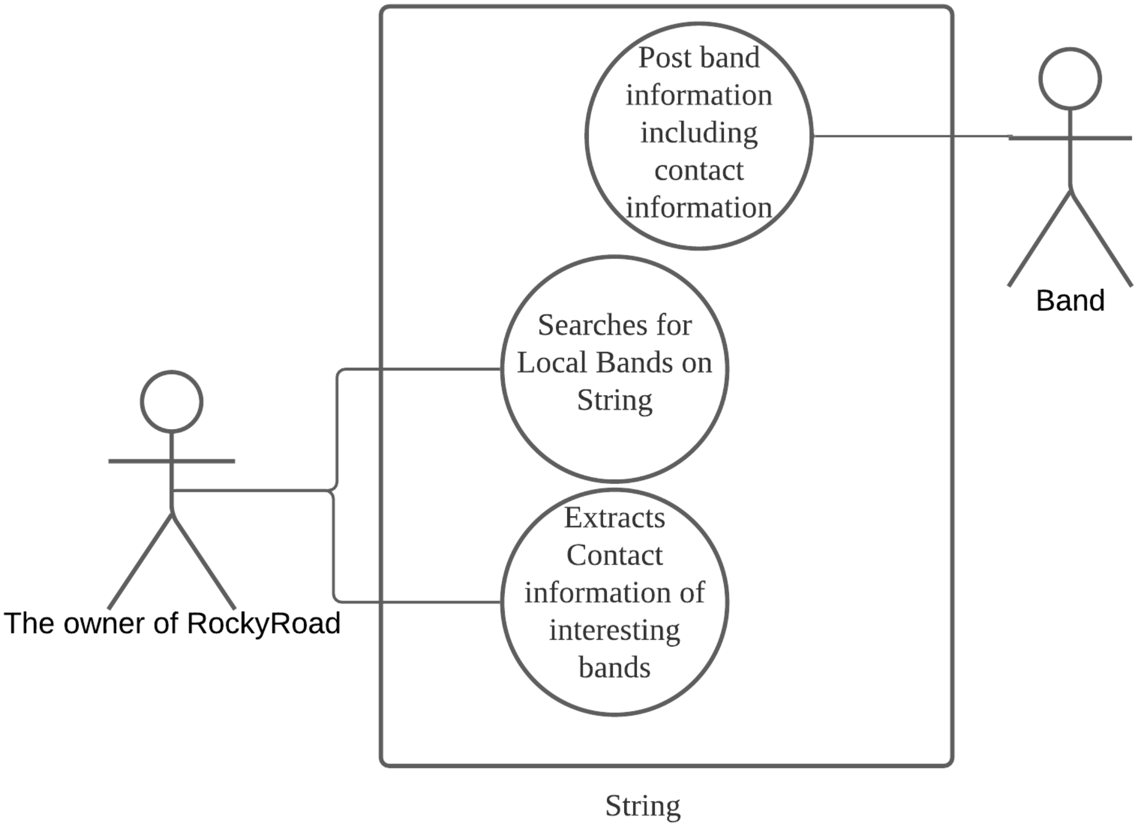
Post band information including contact information
Band
Searches for Local Bands on String
Extracts Contact information of interesting bands
The owner of RockyRoad
String

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.

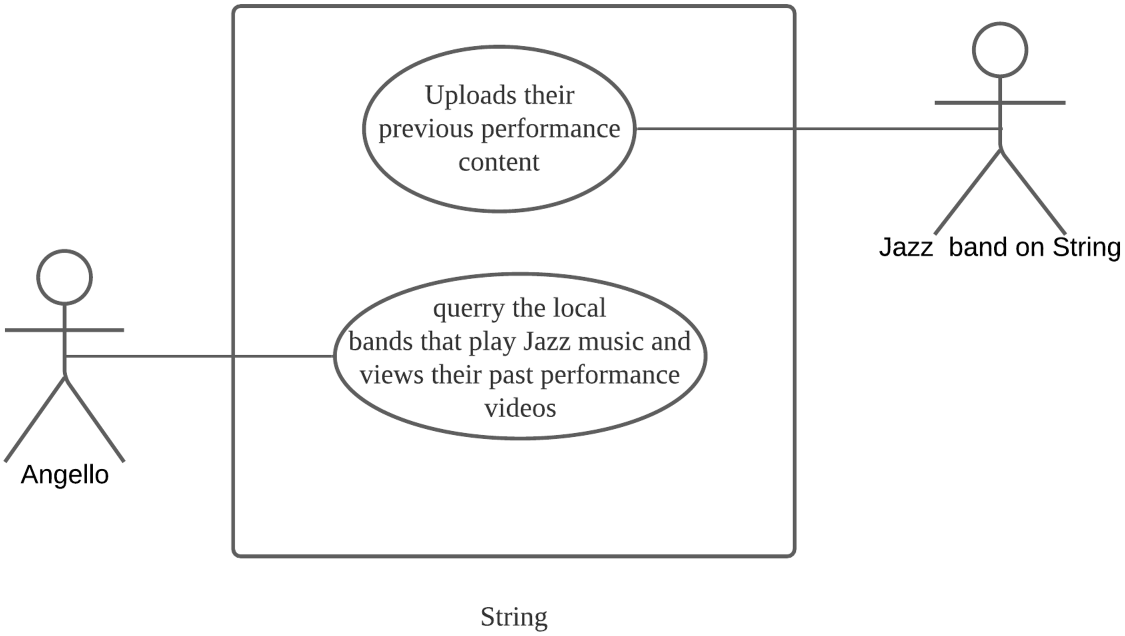
Uploads their previous performance content
Jazz band on String
querry the local bands that play Jazz music and views their past performance videos
Angello
String

Figure 6: Looking for bands to play

3 List of main data items and entities

1. String Account: Account of a Registered user on the platform.
2. Administrator String Account: Account of an Administrator on the platform.
3. Registered user: A person who has a String account and is logged into that account.
4. Guest: A person who does not have a String account but is just viewing pages on our platform.
5. Administrator: A person who has an Administrator String account and is logged into that account. They have access over all information, items and entities on the platform. They are the ones who will manage inappropriate content.
6. 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.
7. Band Admin: The person who created the band using their registered account, or anyone promoted to that role by an existing Band Admin
8. Band Member: A registered user who joins a Band, with a Band Admin's permission.
9. Band post: Image or video uploaded by bands
10. Invitation: Sent by a registered user to join a band in response to availability post
11. 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 information and Set List.
12. 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.
13. Set Entry: A song which the band is planning to play for a specific event/performance. Includes details such as run time, tempo, and order in performance.
14. 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.
15. Repertoire Entry: A song which the band can play, including details such as run time, genre, links to past video, etc.
16. Repertoire 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 Guest

1. Guest shall be able to view all Bands.
2. Guest shall be able to view all Band Members.
3. Guest shall be able to view all Bands' Posts.
4. Guest shall be able to view all Bands' Event List.
5. Guest shall be able to view all Bands' Set List.
6. Guest shall be able to view all Bands' Repertoire List.
7. Guest shall be able to access contact information of all Bands.
8. Guest shall be able to search any Band.
9. Guest shall be able to filter all Bands based on their geographic location.
10. Guest shall be able to filter all Bands based on the genre of music the Bands have selected.
11. Guest shall be able to register an account with a valid email address.
12. Guest shall be able to contact an Administrator to get help at any time.

4.2 Registered User

1. Registered User should be a Guest.
2. Registered User should have a String Account.
3. Registered User shall be able to view their String Account.
4. Registered User shall be able to log in using their email and password.
5. Registered User shall be able to change their email in their String Account.
6. Registered User shall be able to change their password in their String Account.
7. Registered User shall be able to delete their String Account.
8. Registered User shall be able to edit the field indicating their name in their String Account.

9. Registered User shall be able to edit the field indicating if they are a musician in their String Account.

10. Registered User shall be able to edit the field to specify which instrument(s) they play in their String Account.

11. Registered User shall be able to edit their current geographical location in their String Account.

12. Registered User shall be able to edit their contact information in their String Account.

13. Registered User be able to send an invitation to join a Band.

14. Registered User be able to create Band(s).

15. Registered User be able to view all the Bands they are a Band Member of.

4.3 Administrator

1. Administrator shall be able to log in using their email and password.

2. Administrator shall be able to change their email in their Administrator String Account.

3. Administrator shall be able to change their password in their Administrator String Account.

4. Administrator shall be able to create a Band.

5. Administrator shall be able to delete any Band.

6. Administrator shall be able to delete any Registered User.

7. Administrator shall be able to add registered users to any Band.

8. Administrator shall be able to delete any content posted by any Band.

9. Administrator shall be able to delete any Event Entry in any Band.

10. Administrator shall be able to delete any Set List in any Event Entry in any Band.

11. Administrator shall be able to delete any Repertoire Entry in any Band.

12. Administrator shall be able to remove any Band Member in any Band.

13. Administrator shall be able to change Band Admin in any Band.

4.4 Band

1. Band can only be created by a Registered User.

2. Band shall have only one Band Admin.

3. Band shall have at least one Band Member.

4. The Registered User who creates the Band is set as the Band Admin.

5. Band shall have a field indicating if they are accepting invitations from Registered Users.

6. Band shall have a minimum of one Band Member.

7. Band which have no Band Members shall be deleted.

8. Band shall have a geographic location.

9. Band shall have Event List indicating upcoming events.

10. Band shall have Band Post(s).

11. Band shall have contact information.

12. Band shall have a Repertoire List.

4.5 Band Member

1. Band Member should be registered users.

2. Band Member shall be able to add Band posts to their Band.

3. Band Member shall be able to delete Band posts to their Band.

4. Band Member shall be able to leave their Band at any time.

5. Band Member shall be able to add an Event Entry in the Event List of their Band.

6. Band Member shall be able to delete an Event Entry in the Event List of their Band.

7. Band Member shall be able to edit an Event Entry in the Event List of their Band.

8. Band Member shall be able to add a Set List to an Event Entry of their Band.

9. Band Member shall be able to delete a Set List to an Event Entry of their Band.

10. Band Member shall be able to edit a Set List to an Event Entry of their Band.

11. Band Member shall be able to add a Repertoire Entry in the Repertoire List of their Band.
12. Band Member shall be able to delete a Repertoire Entry in the Repertoire List of their Band.
13. Band Member shall be able to edit a Repertoire Entry in the Repertoire List of their Band.

4.6 Band Admin

1. Band Admin should be a Band member.
2. Band Admin shall remove a Band member of their Band.
3. Band Admin shall be able to assign other Band Member as Band Admin.
4. Band Admin shall be able to view all Invitations in their Band.
5. Band Admin shall be able to accept an Invitation in their Band.
6. Band Admin shall be able to delete an Invitation in their Band.
7. Band Admin shall be able to delete their Band.
8. Band Admin shall be able to change the name of their Band.

4.7 Band Post

1. Band Post can contain an image.
2. Band Post can contain a video.
3. Band Post must contain a description.
4. Band Post must contain a title.

4.8 Invitation

1. Invitation can have a message field.
2. Invitation must have a date indicating when it was sent.
3. Invitation must have a field indicating which Registered user sent it.
4. Invitation must have a field indicating which Band was it sent to.
5. Accepting an Invitation results in adding the sender to the Band as a Band member.

4.9 Event Entry

1. Event Entry must have a title field.
2. Event Entry must have a location field indicating where the event will take place.
3. Event Entry must have a date field indicating when the event will take place.
4. Event Entry must have a field indicating which Band created it.
5. Event Entry shall have a description field.
6. Event Entry shall have a Set List.

4.10 Repertoire Entry

1. Repertoire Entry must have a field indicating song's name.
2. Repertoire Entry shall have a field indicating song's tempo.
3. Repertoire Entry shall have a field indicating song's mood.
4. Repertoire Entry shall have a field indicating song's run time.
5. Repertoire Entry shall have a list of links.
6. Repertoire Entry shall have a field indicating ift the song is in the rehearsal stages.

4.11 Set Entry

1. Set Entry shall have a field indicating song's name.
2. Set Entry shall have a field indicating song's mood.
3. Set Entry shall have a field indicating song's tempo.
4. Set Entry shall have a field indicating song's approximate run time.

4.12 Event List

1. Event List shall contain a list of Event Entries.

4.13 Repertoire List

1. Repertoire List shall contain a list of Repertoire Entries.

4.14 Set List

1. Set List shall contain a list of Set Entries.

2. Set List must contain at least one Set Entry

3. Set List shall have Set Entries in projected performance order.

5 Non-functional Requirements

- Scalability

  1. The website should be able to host at least 300 users in the same hosted region as the EC2 instance.
  2. A load balancer shall not be used while deployment.

- 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.

- Network and Deployment capabilities

  1. Application should be deployed on AWS EC2 instance.
  2. Only the master branch of github should be deployed on the AWS server instance.
  3. Any and all deployment should be handled only by the DevOps Lead.

- 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.

- Storage and data integtity

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

None of the other competitors that we examined had any substantial focus on group membership management or in person musical collaboration; therefore, our feature list is rich in management tools toward that end, in order to fully take advantage of that underserved population. The Repertoire and Set Lists allow for easy management tools for musical groups to prepare for events, and Event Lists allow for easy performance, rehearsal, and casual ‘jam session’ management. Users can be a part of many Bands at once on our platform, which means that String can be a one-stop hub for musicians to find, join, view, manage, and ultimately make music with differing groups of people in their area. All these features make String the best platform for musicians in a specific local geography who want to (or already do) play in groups together, either for fun, or for paid gigs.

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**