

Milestone 3

Team 5 - *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 3 v2.0	December 1, 2020
Milestone 3 v1.0	November 19, 2020
Milestone 2 v2.0	November 7, 2020
Milestone 2 v1.0	October 29, 2020
Milestone 1 v2.0	October 14, 2020
Milestone 1 v1.0	October 1, 2020

December 1, 2020

Contents

1 Data Definitions V2	2
2 Functional Requirements V2	5
2.1 Priority 1	5
2.2 Priority 2	8
2.3 Priority 3	10
3 Wireframes	12
3.1 Use Case 1: A solo artist looking for a band to join	12
3.2 Use Case 2: A band manager onboards, searching to have one place manage/display all bands he manages	14
3.3 Use Case 3: Looking to discover music events in their area (guest journey)	16
3.4 Use Case 4: A band looking for musicians	18
3.5 Use Cases 5 & 6: Guest user looking to sell products to local bands (looking for contact info), Guest user looking for bands to play at their event	20
4 High level database architecture and organization	22
4.1 Business Rules	22
4.2 Entities Description	22
4.3 ERD	23
4.4 Database model	24
4.5 Media Storage	24
4.6 Search/filter architecture and implementation	24
5 High Level UML, Application Network and Deployment Diagrams	25
5.1 High Level UML Diagram	25
5.2 Application Networks Diagram	26
5.3 Deployment Diagram	26
6 List of contributions to the three parts of this milestone	27

1 Data Definitions V2

1. String Account: Account of a Registered user on the platform.
 - (a) Email - an email address of the Registered User
 - (b) Password - used to log into the String Account.
 - (c) Name - the name of the Registered User.
 - (d) Phone number - (optional) phone number of the Registered User.
 - (e) Location - (optional) the address of the Registered User.
 - (f) Location Coordinates - (optional) the address coordinates of the Registered User.
 - (g) Role - (optional) list of roles the Registered User can perform.
 - (h) Genre(s) - (optional) list of primary musical interests (i.e. classical, jazz, rock, electronic)
 - (i) Profile picture - (optional) Registered User's picture.
2. Administrator String Account: Account of an Administrator on the platform (i.e. the team building and managing the String Platform).
 - (a) Email: an email address of the Administrator.
 - (b) Password: used to log into the Administrator String Account.
 - (c) Name: the name of the Administrator.
 - (d) Role: role of the Administrator in the team (i.e frontend lead, team lead, content manager, etc.).
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 and 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 Member: A registered user who joins a Band, with a Band Admin's permission.
 - (a) Band Admin: if the Band member has admin access in the band. (a boolean value)
 - (b) Role: Role that the Band Member has in the Band.
 - (c) Date joined: When the Band Member joined the Band.

7. Band: A band can be created by a registered user. It is an entity that other registered users can join if Band Admin permits.

- (a) Band Name: (required) Name of the band. (unique)
- (b) Band logo image: Logo image of the Band.
- (c) Band Description: A short description of what the band is about.
- (d) Band Links: list of links.
 - i. Link: the link pointing to another webpage.
 - ii. Short description: a small description of the link.
- (e) Band Post: Image or video uploaded by bands.
 - i. Media: Link pointing to an image or a video (could be on youtube or Vimeo or Google Drive).
 - A. Type of media: either image or video.
 - B. Link: link to that image or video.
 - ii. Title (required): title of what the Band Post is about.
 - iii. Description (optional): description of the Band Post
- (f) Location (required): Address that the band belongs to and where they are performing.
- (g) Location Coordinates (required): latitude and longitude of the address of the Band
- (h) 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.
 - i. Event title: (required) The title of the event.
 - ii. Date of event: (required) month day and year the event is taking place.
 - iii. Start Time of event: (required) time at which the event will begin.
 - iv. End time of event: (required) time at which event will end.
 - v. Location of event (required): Address and description of where the event will be held.
 - vi. Coordinates: (required) latitude and longitude of the address of the Event
 - vii. Event description: description of the event.
- viii. 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.
 - A. Song: name of the song.
 - B. Run time: length of the song.

- (i) Repertoire Entry: A song which the band can play, including details such as:
 - i. Song - name of the song.
 - ii. Run time - length of the song.
 - iii. Genre - genre of the song.
 - iv. Link - Link to a site with more description/media about the performance of this song.
- 8. Invitation: Sent by a registered user to join a band in response to availability post
 - (a) Message: A short paragraph of why the user would want to join the Band
 - (b) Date: Date on which the Invitation was sent.
 - (c) Sender account: the String Account who sent the invitation
 - (d) Receiver band: the Band that the invitation was sent to.

2 Functional Requirements V2

2.1 Priority 1

2.1.1 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 access contact information of all Bands.
6. Guest shall be able to search for any Band by name.
7. Guest shall be able to filter any Bands by genre.
8. Guest shall be able to filter Bands by location.
9. Guest shall be able to register an account with a valid email address.
10. Guest shall be able to view all Events' Set List.
11. Guest shall be able to view all Bands' Repertoire List.

2.1.2 Registered User

1. Registered User shall be a Guest.
2. Registered User shall have a String Account.
3. Registered User shall be able to view their String Account information.
4. Registered User shall be able to log in using their email and password.
5. Registered User shall be able to send an invitation to apply to join a Band.
6. Registered User shall be able to create Band(s).
7. Registered User shall be able to view all the Bands information they are a Band Member of.
8. Registered User shall be able to change their password in their String Account.

9. Registered User shall be able to edit the field indicating their name in their String Account.
10. Registered User shall be able to edit the field indicating their role.
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.

2.1.3 Band

1. Band can only be created by a Registered User.
2. Band shall have only one Band Admin.
3. The Registered User who creates the Band is set as the Band Admin.
4. Band shall have a field indicating if they are accepting invitations from Registered Users.
5. Band shall have a field indicating their geographic location.
6. Band shall have Band Post(s).
7. Band shall have contact information.
8. Band shall have an Event List indicating upcoming events.
9. Band shall have a Repertoire List.

2.1.4 Band Member

1. Band Member should be registered users.
2. Band Member shall be able to leave their Band at any time.
3. Band Member shall be able to add Band posts to their Band.
4. Band Member shall be able to delete Band posts to their Band.
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 add a Set Entry to an Event Entry of their Band.
8. Band Member shall be able to delete a Set Entry to an Event Entry of their Band.
9. Band Member shall be able to add a Repertoire Entry to their Band.
10. Band Member shall be able to delete a Repertoire Entry to their Band.

2.1.5 Band Admin

1. Band Admin should be a Band member.
2. Band Admin shall be able to remove a Band member of their Band.
3. Band Admin shall be able to view all outgoing and incoming Invitations in their Band.
4. Band Admin shall be able to issue or accept an outgoing or incoming Invitation in their Band.
5. Band Admin shall be able to delete any outgoing or incoming Invitation in their Band.

2.1.6 Band Post

1. Band Post shall contain a field for an image.
2. Band Post shall contain a field for a link to a video.
3. Band Post shall contain a field for a description.
4. Band Post shall contain a title.

2.1.7 Invitation

1. Invitation shall have a message field.
2. Invitation shall have a date indicating when it was sent.
3. Invitation shall have a field indicating which Registered user sent it.
4. Invitation shall have a field indicating which Band it was sent to.
5. Accepting an incoming Invitation shall result in adding the Registered User to the Band as a Band member.

2.1.8 Event Entry

1. Event Entry shall have a title field.
2. Event Entry shall have a location field indicating where the event will take place.
3. Event Entry shall have a date field indicating which day the event will take place.
4. Event Entry shall have a time field indicating what time the event will take place.
5. Event Entry shall have a field indicating which Band created it.

6. Event Entry shall have a description field.
7. Event Entry shall have a list of Set Entries.

2.1.9 Repertoire Entry

1. Repertoire Entry shall have a field indicating the song's name.
2. Repertoire Entry shall have a field indicating the song's genre.
3. Repertoire Entry shall have a field indicating the song's run time.

2.1.10 Set Entry

1. Set Entry shall have a field indicating song's name.
2. Set Entry shall have a field indicating song's run time.
3. Set Entry shall have a field indicating order in the playlist.

2.2 Priority 2

2.2.1 Guest

1. Guests shall be able to add reviews to a Band.
2. Guests shall be able to play songs linked in Repertoire Entries.

2.2.2 Registered User

1. Registered Users shall be able to change their email details in their String Account.
2. Registered Users shall be able to like a band post.
3. Registered Users shall be able to add a comment on a band post.
4. Registered Users shall be able to RSVP to an event.

2.2.3 Administrator

1. Administrator shall be able to change their email in their Administrator String Account.
2. Administrator shall be able to change their password in their Administrator String Account.
3. Administrator shall be able to delete any Set List in any Event Entry in any Band.

4. Administrator shall be able to delete any Repertoire Entry in any Band.
5. Administrator shall be able to log in using their email and password.
6. Administrator shall be able to delete any Band.
7. Administrator shall be able to delete any content posted by any Band.
8. Administrator shall be able to delete any Event Entry in any Band

2.2.4 Band Member

1. Band Members shall be able to edit an Event Entry in the Event List of their Band.
2. Band Members shall be able to edit a Set List to an Event Entry of their Band.
3. Band Members shall be able to edit a Repertoire Entry in the Repertoire List of their Band.

2.2.5 Band Admin

1. Band Admins shall be able to assign a different Band Member as Band Admin.
2. Band Admins shall be able to change the name of their Band.

2.2.6 Event Entry

1. Event Entry shall have an image
2. Event Entry shall have an RSVP list

2.2.7 Repertoire Entry

1. Repertoire Entry shall have a playable song link.

2.2.8 Set Entry

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.

2.2.9 Repertoire List

1. Repertoire List shall contain an (unordered) list of Repertoire Entries

2.2.10 Band Post

1. Band Post shall have a displayed count of likes.
2. Band Posts shall have a comment section for participation from Registered Users.

2.3 Priority 3

2.3.1 Guest

1. Guest shall be able to contact an Administrator to get help at any time.

2.3.2 Registered User

1. Registered User shall be able to delete their String Account.
2. Registered User shall be able to add comments to an Event.

2.3.3 Administrator

1. Administrator shall be able to create a Band.
2. Administrator shall be able to delete any Registered User.
3. Administrator shall be able to add registered users.
4. Administrator shall be able to add registered users to any Band.
5. Administrator shall be able to remove any Band Member in any Band.
6. Administrator shall be able to change Band Admin in any Band.

2.3.4 Band

1. Band which have less than one Band Member shall be deleted.
2. Band shall have one or more Band Admin(s).

2.3.5 Band Member

1. Band Members shall have access to a text chat viewable only by Band Members.
2. Band Members shall be able to invite Registered Users to an Event.

2.3.6 Band Admin

1. Band Admins shall be able to delete their Band.
2. Band Admins shall be able to make other Band Members to be an additional Band Admin.

2.3.7 Repertoire Entry

1. Repertoire Entry shall have a list of links.
2. Repertoire Entry shall have a field indicating if the song is in the rehearsal stages.
3. Repertoire Entry shall have a field for a short description of or notes on the song.

2.3.8 Event Entry

1. Event Entry shall have a list of registered users invited to the event.
2. Event Entry shall have a section for Registered Users to comment and discuss.

3 Wireframes

3.1 Use Case 1: A solo artist looking for a band to join

1. A registered user as solo artist goes to the explore page and taps the filter button
2. In the filter modal user selects to browse Bands by “Bands that are hiring”.
3. Scroll down to find more
4. The user gets a list of all the bands in their city that are looking for new members
5. Solo artists can access the band pages
6. Inside the band pages registered users can click the “apply” apply button to send a notice of interest on the role to that band
7. Alternatively, they can reach out directly to the bands that have their contact information set as public

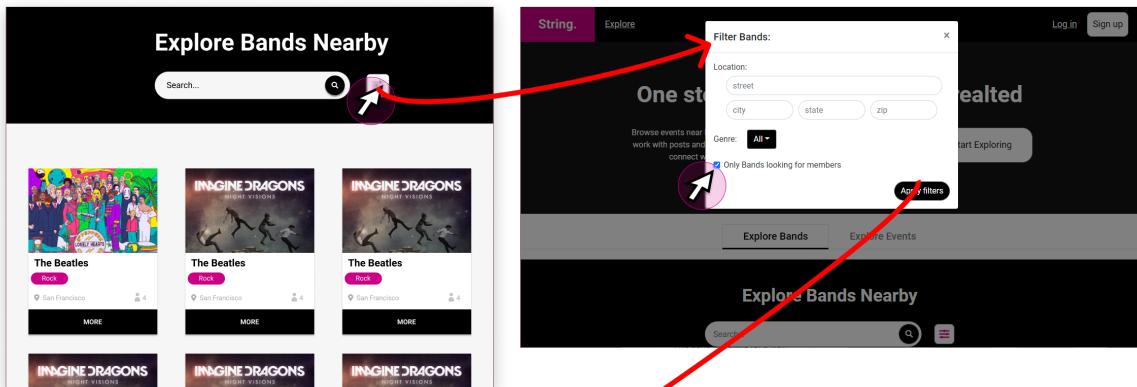
Usecase 1:

A solo artist looking to join a band

In explore pg filter bands by “bands that are hiring”

1

2

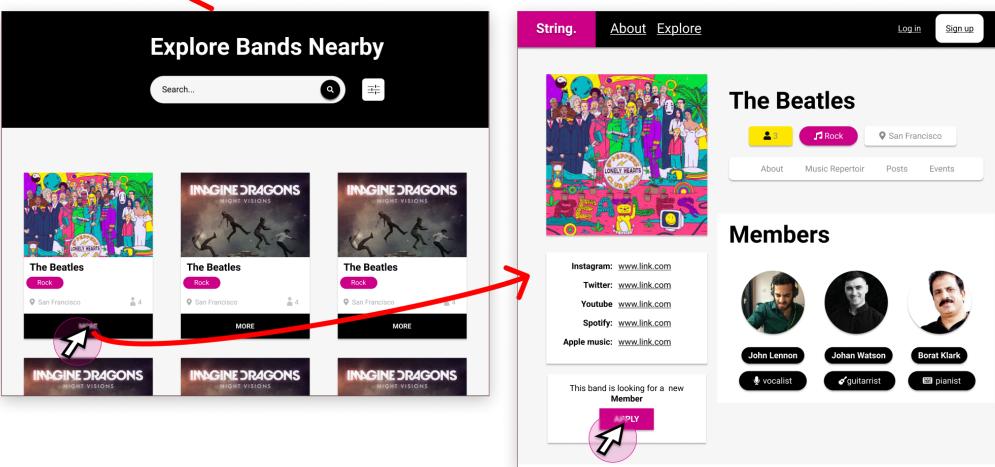


Click on the band that interests you

Apply in the band page

3

4

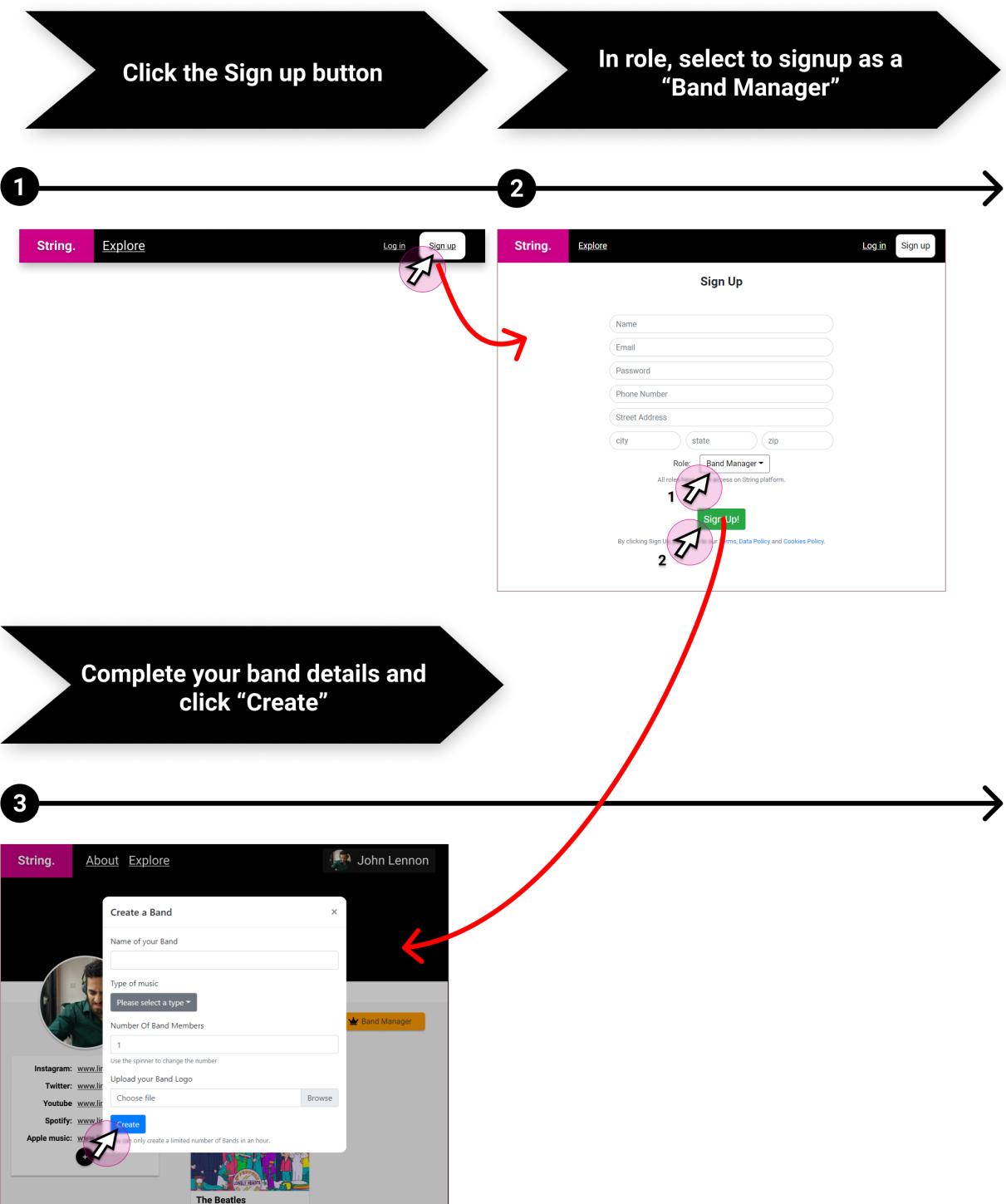


3.2 Use Case 2: A band manager onboards, searching to have one place manage/display all bands he manages

1. User clicks on the sign up button in the explore page or any other public page.
2. Onboarding user inputs: name, last name, email, phone number, and password, address, role, instrument and genre
3. Users select onboard as either: a solo artist, a manager, or a band.
4. Users that select to sign ups as “Band Manager” are prompted to create a band after signup
5. The user may complete the details for the band page or finish later (Band name, logo/image, street address, genre, description, links, events, posts, music repertoire)

Usecase 2:

A band manager onboards and creates a band on String



3.3 Use Case 3: Looking to discover music events in their area (guest journey)

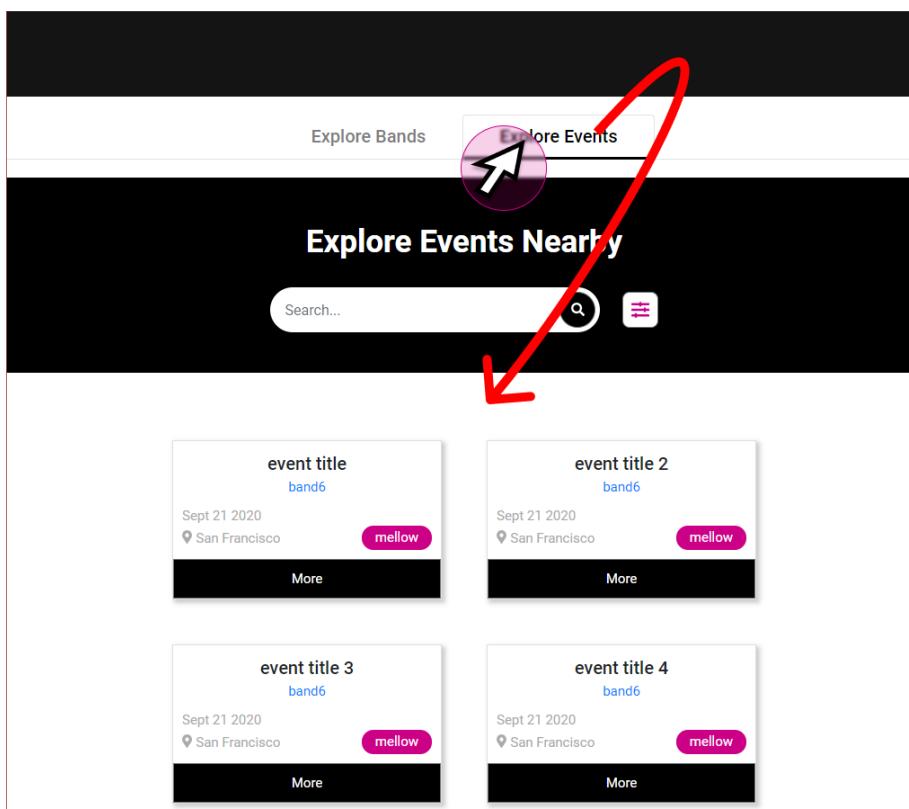
1. In the explore page tap the Events button to browse events by city.
2. Scroll down to find more.
3. For filtering results of any kind the filter button located on the top right of the screen opens an overlay with filter preferences (i.e. zip, genre, city, size).

Usecase 3:

Looking to discover events in the area (guest journey)

Click on “Explore events” in the explore page

1



3.4 Use Case 4: A band looking for musicians

1. A band manager enables "looking for new members" checkbox on the band page
2. An envelope icon appears next to the checked checkbox and would display a red bubble with the amount of requests from users interested shall anyone click on the "apply" button on their page

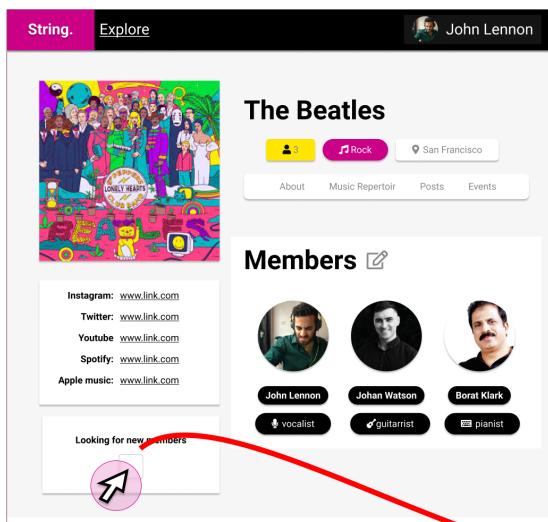
Usecase 4:

A band looking for musicians

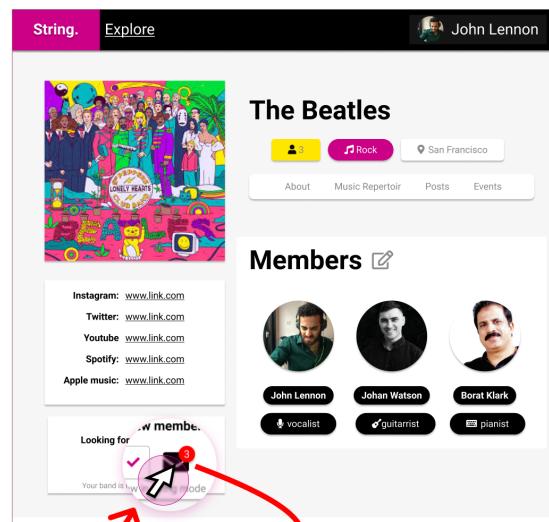
In the band page click the "looking for new members" checkbox

Red bubble on the envelope will indicate people applying to join your band

1

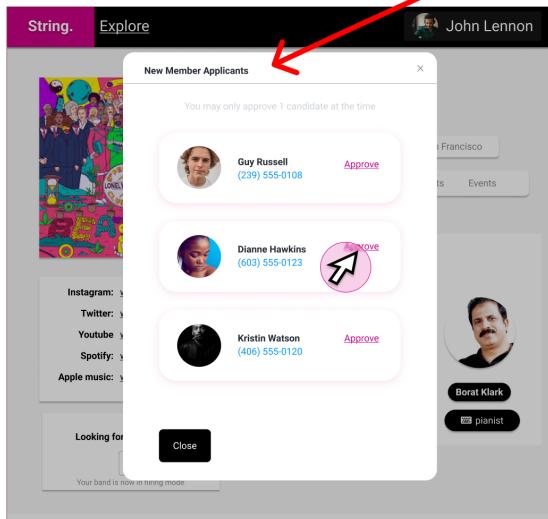


2



You may find applicants contact information and approve the desired candidate

3



3.5 Use Cases 5 & 6: Guest user looking to sell products to local bands (looking for contact info), Guest user looking for bands to play at their event

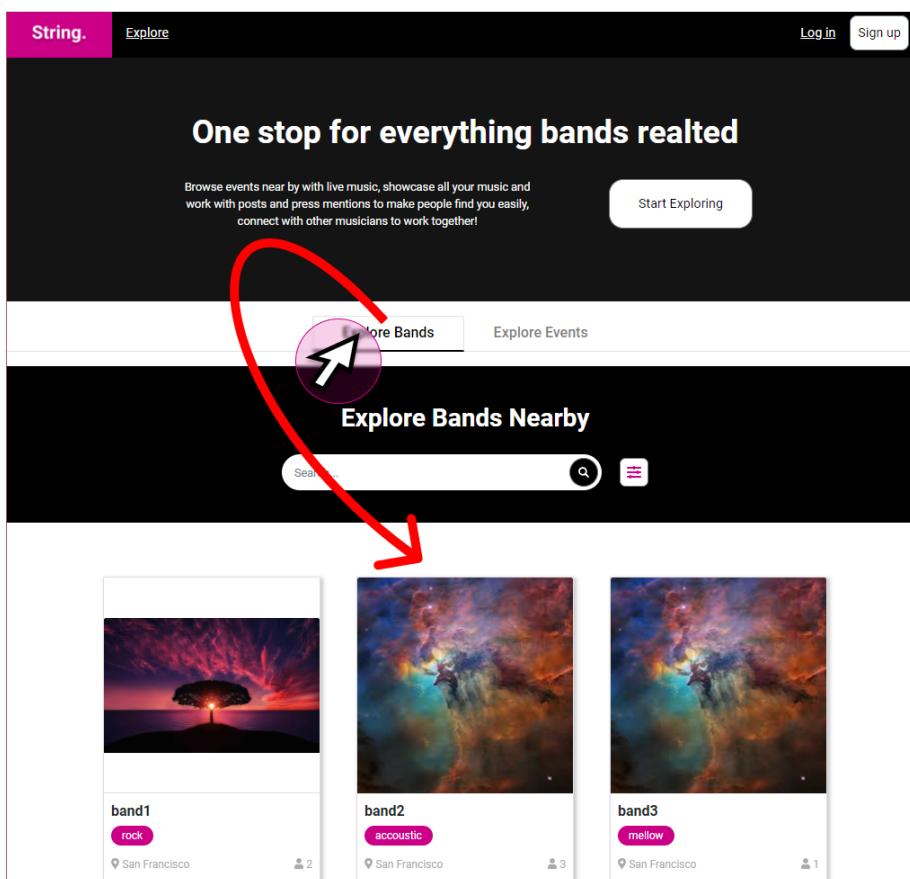
1. In the explore page tap the Bands button to browse Bands by city.
2. Scroll down to find more.
3. For filtering results of any kind the filter button located on the top right of the screen opens an overlay with filter preferences (i.e. zip, genre, city, size).
4. By tapping on either one result the guest user is funneled to the respective bands page.
5. Guests can reach out directly to the bands that have their contact information set as public.

Usecase 5 & 6:

- Guests looking for bands to play at their event/venue
- Guests looking to sell products to local bands

Click on “Explore Bands” in the explore page

1



4 High level database architecture and organization

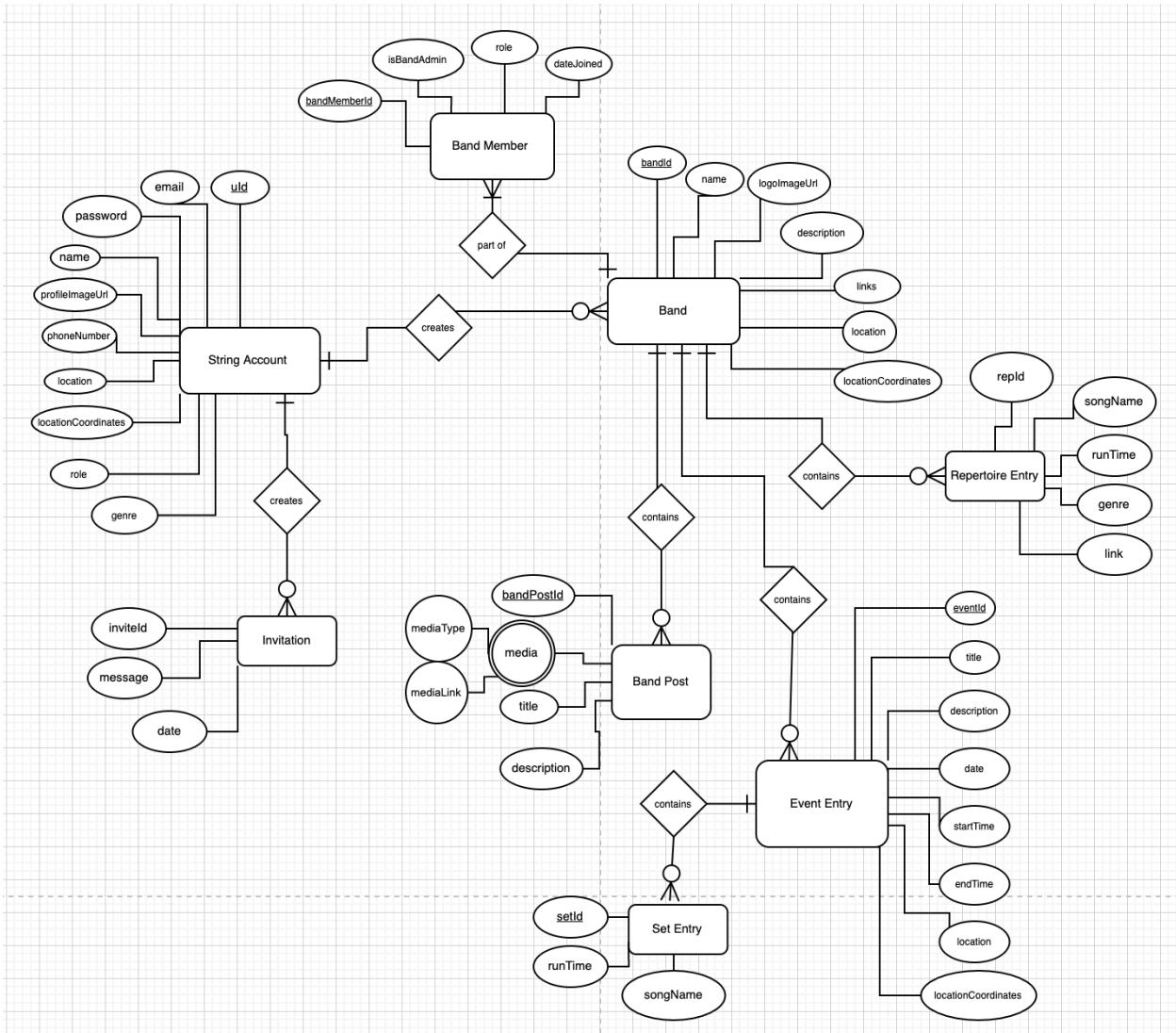
4.1 Business Rules

1. One Guest(with a email) can have one String Account.
2. Each String Account can be a part of zero or more Band(s).
3. Each Band can have one or more Band member(s)
4. Each Band can have zero or more Event Entry.
5. Each Event can have zero or more Event Entry.
6. Each Band can have zero or more Repertoire Entry.
7. Each Band can have zero or more Band Post.
8. Each String Account can send zero or more invitation(s)

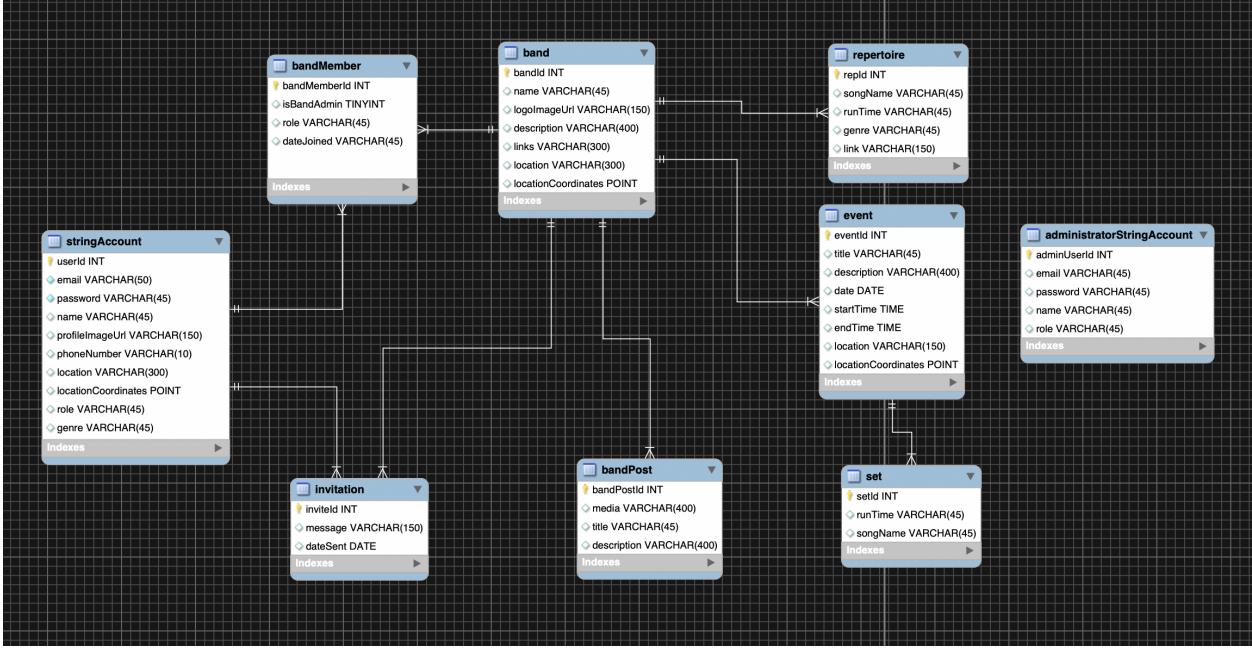
4.2 Entities Description

Same as described in Data Definitions V2 (section 1 of this document)

4.3 ERD



4.4 Database model



We chose to use MySQL database because most of our team members were familiar with it and AWS RDS offered MySQL server.

4.5 Media Storage

We will be using S3 buckets to store profile pictures of Individuals and Bands.

For image files we will accept files with extension jpg, jpeg, png and gif.

The size of the file should be less than 10mb.

We will not be storing videos but if Bands want to upload video on their Band Post they will upload a link to the video uploaded on other websites like youtube.

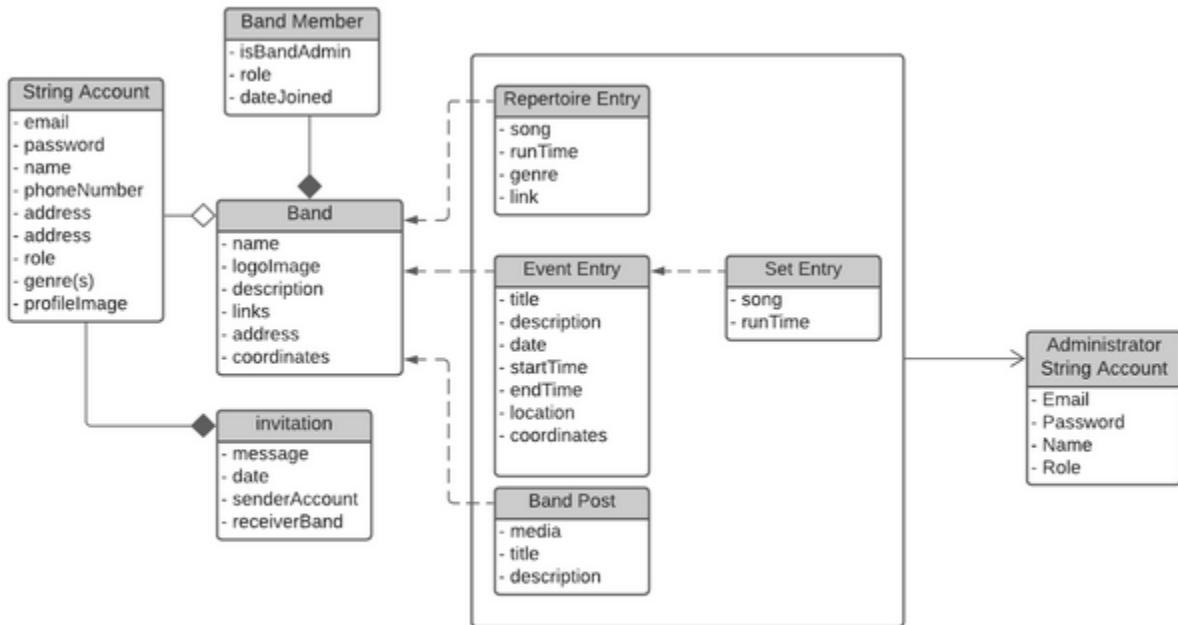
4.6 Search/filter architecture and implementation

We will have a search feature for Bands and for events. The searching will be through band name and event title. The search function will use SQL like and % functions to get results matching search query.

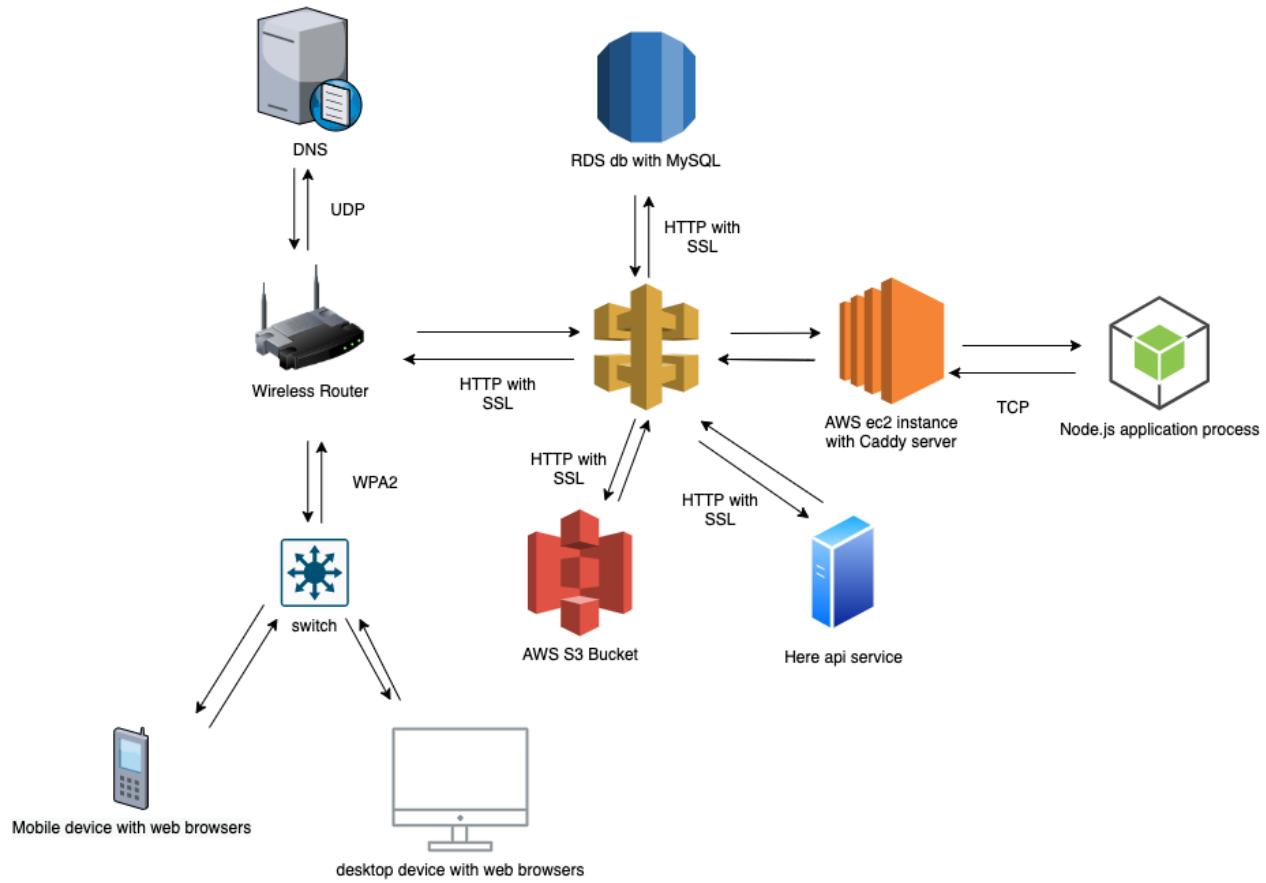
We will have some filters out of which location filter will be the most important. We take the location of all Bands and of registered users which will be converted to coordinates using Here map's api which will be stored in the database as a POINT element. We would then use MySQL geospatial functions to sort using this point and the user's location coordinates.

5 High Level UML, Application Network and Deployment Diagrams

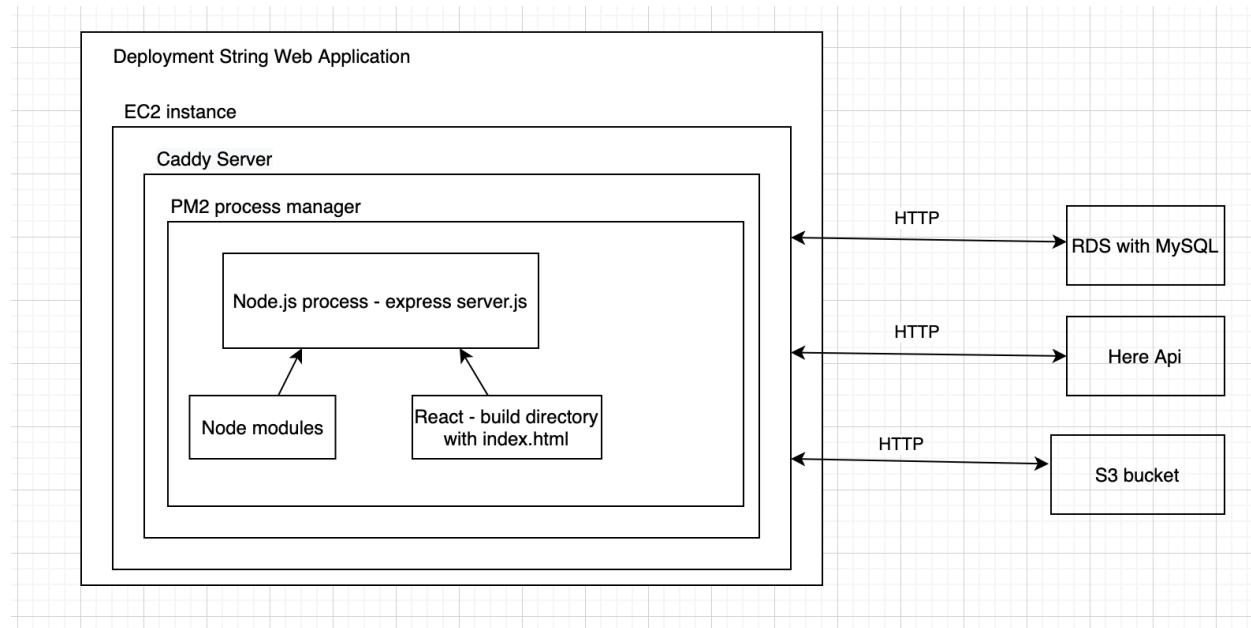
5.1 High Level UML Diagram



5.2 Application Networks Diagram



5.3 Deployment Diagram



6 List of contributions to the three parts of this milestone

- Jainam Shah: (Team Lead)
 1. Conducted meetings and paired programming with team showing and explaining how to make routes, controllers, middlewares and helpers.
 2. Helped provide timely feedback and help on UI design.
 3. Build the auth routes - login, register and logout and band routes - createBand and searchBand.
 4. managed multer, S3 for file handling and passport for auth.
 5. Build the frontend of the application, connected redux and integrated the routes from backend.
 6. Made query for sorting bands nearest to a given location(based on coordinates) in MySQL.
- Alfredo Diaz: (Frontend lead)
 1. Took lead on designing the UI using Figma and completed all designs by themselves.
 2. Communicated with frontend team the requirements for horizontal prototype.
 3. Build story boards using figma to describe user interactions in usecases.
 4. Helped building the milestone document, regularly attended the meeting and communicated ideas.
- Leonid Novoselov: (Frontend developer)
 1. Build the Navbar and user profile page frontend from scratch.
 2. Attended meetings regularly, shared their opinions when they could and helped with populating the prototype.
- Eric Chen: (Frontend developer)
 1. Was assigned to complete band profile page but got sick and wasn't able to communicate it.
- Warren Singh: (Backend lead)
 1. Led backend team meetings and did peer programming.
 2. Coordinated meetings with frontend lead to coordinate M2 revisions and M3 layout and changes
 3. Helped build createBand, searchBand route
 4. scaffolded createEvent, searchEvents, getBands routes.
 5. Handled formatting and typesetting for M3 and M2 v2 documents, regularly attended the meeting and communicated ideas.

- Ritesh Panta: (Backend developer)
 1. Scaffolded and worked on getBandInfo, getEvent, getBandEvent routes.
 2. Helped backend team with building queries in MySQL.
 3. Attended meetings regularly, shared their opinions when they could and helped with populating the prototype.