SW Engineering CSC 648/848 - FALL 2020

Milestone 2

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 2 v1.1	October 29, 2020

October 30, 2020

Contents

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 any Band.
7. Guest shall be able to register an account with a valid email address.

2.1.2 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 be able to send an invitation to apply to join a Band.
6. Registered User be able to create Band(s).
7. Registered User be able to view all the Bands they are a Band Member of.

2.1.3 Administrator

1. Administrator shall be able to log in using their email and password.
2. Administrator shall be able to delete any Band.
3. Administrator shall be able to delete any content posted by any Band.
4. Administrator shall be able to delete any Event Entry in any Band

2.1.4 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 Event List indicating upcoming events.

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

2.1.6 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 view all Invitations in their Band.
4. Band Admin shall be able to accept an Invitation in their Band.
5. Band Admin shall be able to delete an Invitation in their Band.

2.1.7 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

2.1.8 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 registered User who sent it, to the Band as a Band member.

2.1.9 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 time field indicating when the event will take place.
5. Event Entry must have a field indicating which Band created it.
6. Event Entry shall have a description field.

2.2 Priority 2

2.2.1 Guest

1. Guest shall be able to view all Bands' Set List.
2. Guest shall be able to view all Bands' Repertoire List.
3. Guest shall be able to filter all Bands based on their geographic location.

2.2.2 Registered User

1. Registered User shall be able to change their email in their String Account.
2. Registered User shall be able to change their password in their String Account.
3. Registered User shall be able to edit the field indicating their name in their String Account.
4. Registered User shall be able to edit the field indicating if they are a musician in their String Account.
5. Registered User shall be able to edit the field to specify which instrument(s) they play in their String Account.
6. Registered User shall be able to edit their current geographical location in their String Account.
7. Registered User shall be able to edit their contact information in their String Account.

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.

2.2.4 Band

1. Band shall have a Repertoire List

2.2.5 Band Member

1. Band Member shall be able to leave their Band at any time.
2. Band Member shall be able to add a Set List to an Event Entry of their Band.
3. Band Member shall be able to delete a Set List to an Event Entry of their Band.
4. Band Member shall be able to add a Set Entry to a Set List
5. Band Member shall be able to delete a Set Entry in a Set List
6. Band Member shall be able to add a Repertoire Entry in the Repertoire List of their Band.
7. Band Member shall be able to delete a Repertoire Entry in the Repertoire List of their Band.

2.2.6 Band Admin

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

2.2.7 Event Entry

1. Event Entry shall have a Set List

2.2.8 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.

2.2.9 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.

2.2.10 Repertoire List

1. Repertoire List shall contain a list of Repertoire Entries

2.3 Priority 3

2.3.1 Guest

1. Guest shall be able to filter all Bands based on the genre of music the Bands have selected.
2. 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.

2.3.3 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 to any Band.
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 shall have at least one Band Member.
2. Band which have no Band Members shall be deleted.

2.3.5 Band Member

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

2.3.6 Band Admin

1. Band Admin shall be able to delete their Band.

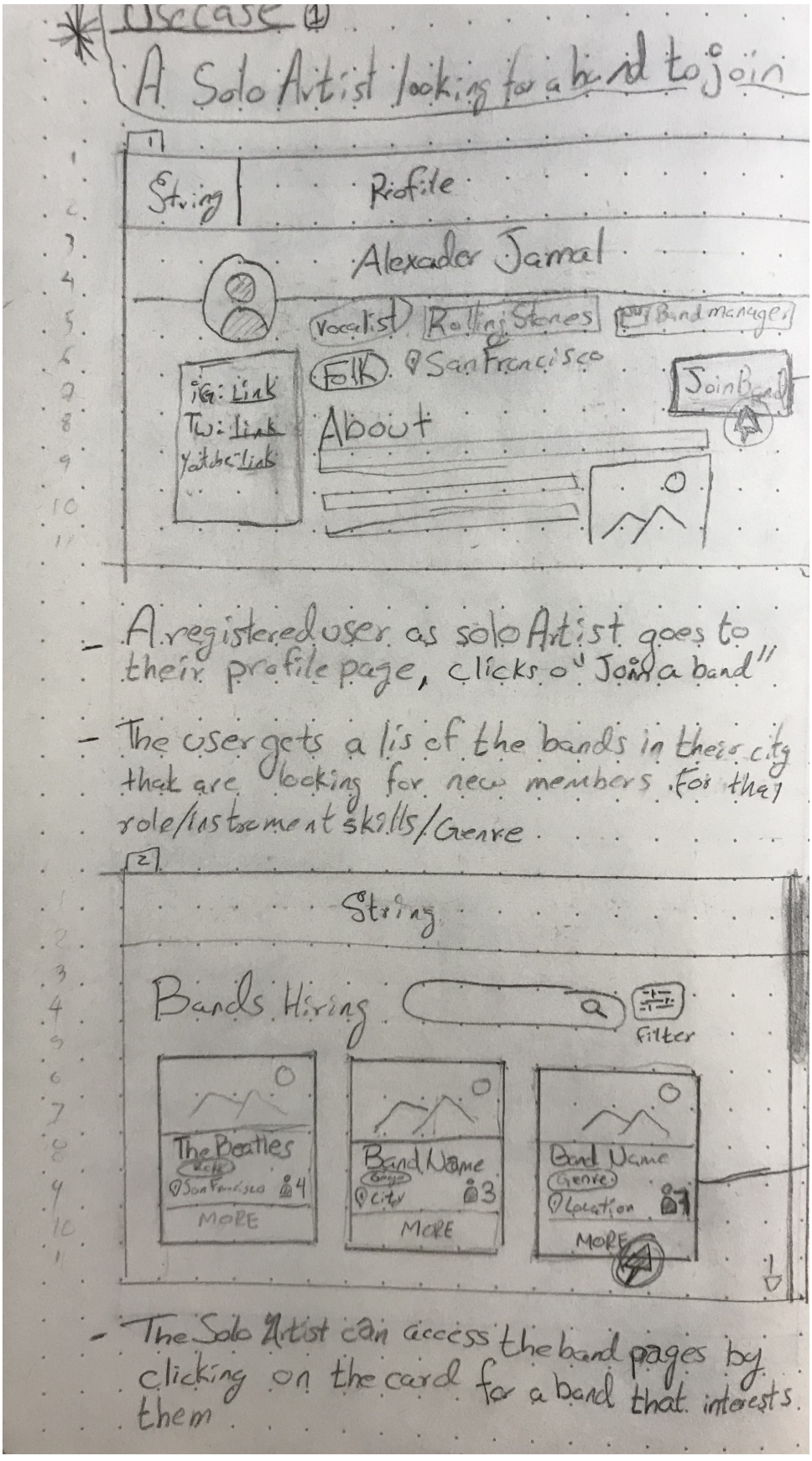
2.3.7 Repertoire Entry

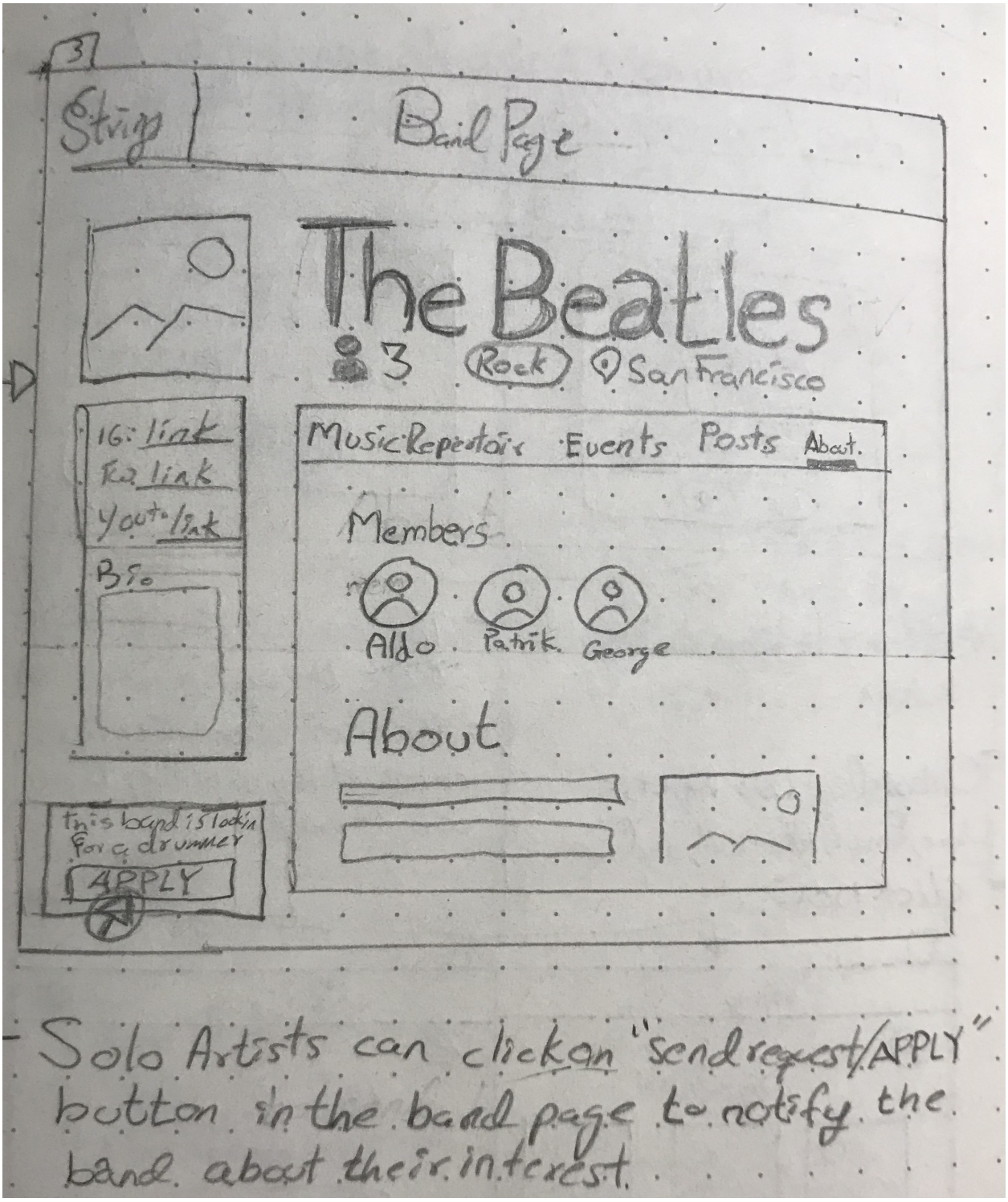
1. Repertoire Entry shall have a list of links.
2. Repertoire Entry shall have a field indicating if the song is in the rehearsal stages.
3. Repertoire Entry shall have notes.

3 UI Mockups and Storyboards

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

1. A registered user as solo artist goes to their profile page
2. User clicks on “Join a band”
3. The user gets a list of all the bands in their city that are looking for new members pre-filtered by the solo artist’s instruments/roles.
4. Solo artists can access the band pages and click on send request/apply on the ones that are interested
5. Alternatively, they can reach out directly to the bands that have their contact information set as public.

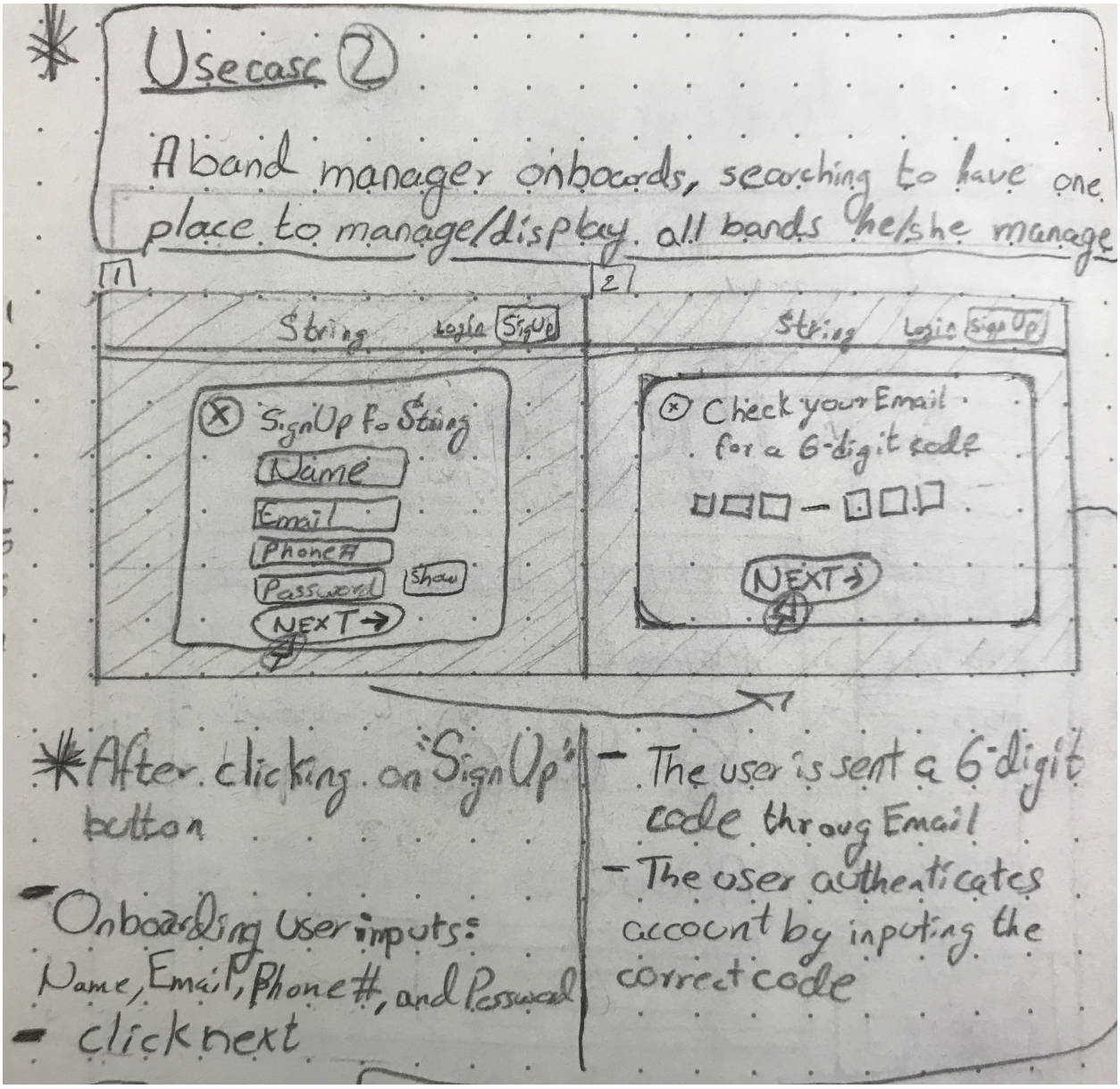

Usecase 1
A Solo Artist looking for a band to join
1
String
Profile
Alexander Jamal
Vocalist
Rolling Stones
Band manager
Folk
San Francisco
IG: Link
TW: Link
Youtube Link
About
Join Band
- A registered user as solo Artist goes to their profile page, clicks on "Join a band"
- The user gets a list of the bands in their city that are looking for new members for their role/instrument skills/Genre
2
String
Bands Hiring
Filter
The Beatles
San Francisco
4
MORE
Band Name
City
3
MORE
Band Name
Genre
Location
MORE
- The Solo Artist can access the band pages by clicking on the card for a band that interests them

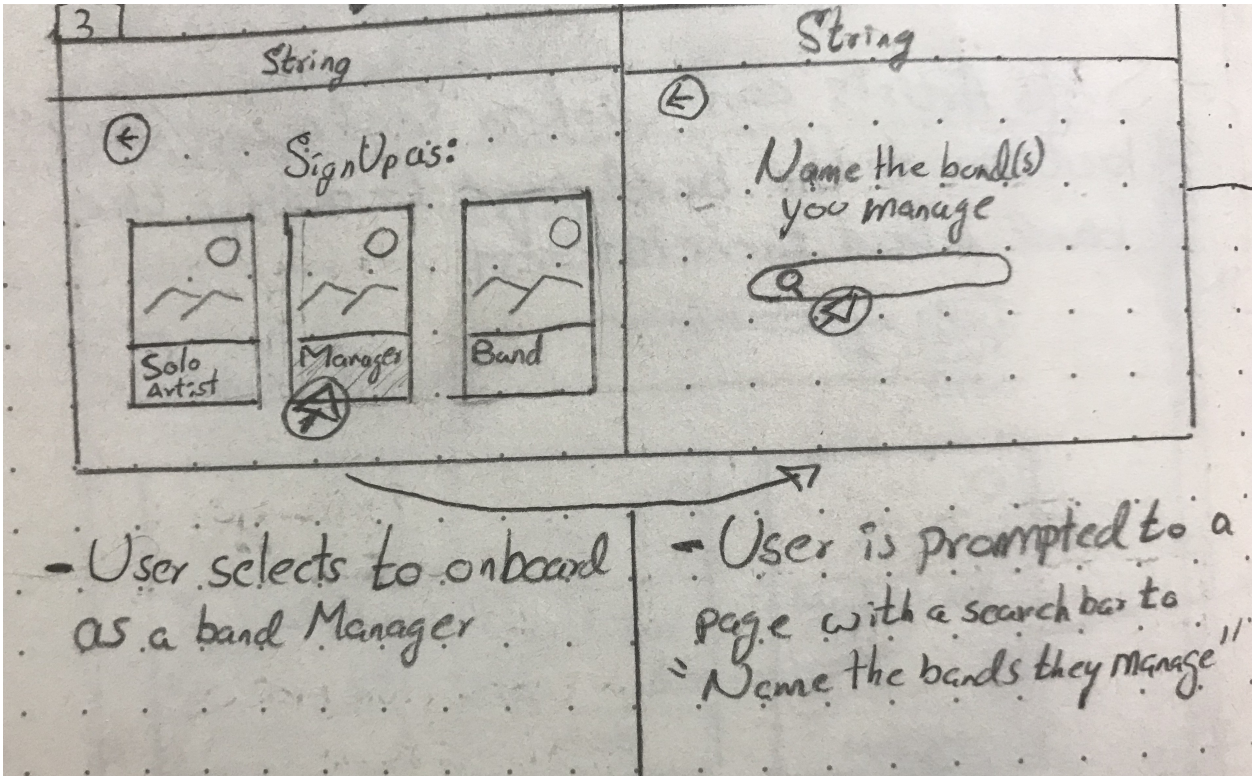

3
String
Band Page
The Beatles
3
Rock
San Francisco
IG: link
Fa link
Yout. link
Bio
Music Repertoire
Events
Posts
About
Members
Aldo
Patrik
George
About
this band is looking for a drummer
APPLY
- Solo Artists can click on "send request/APPLY" button in the band page to notify the band about their interest

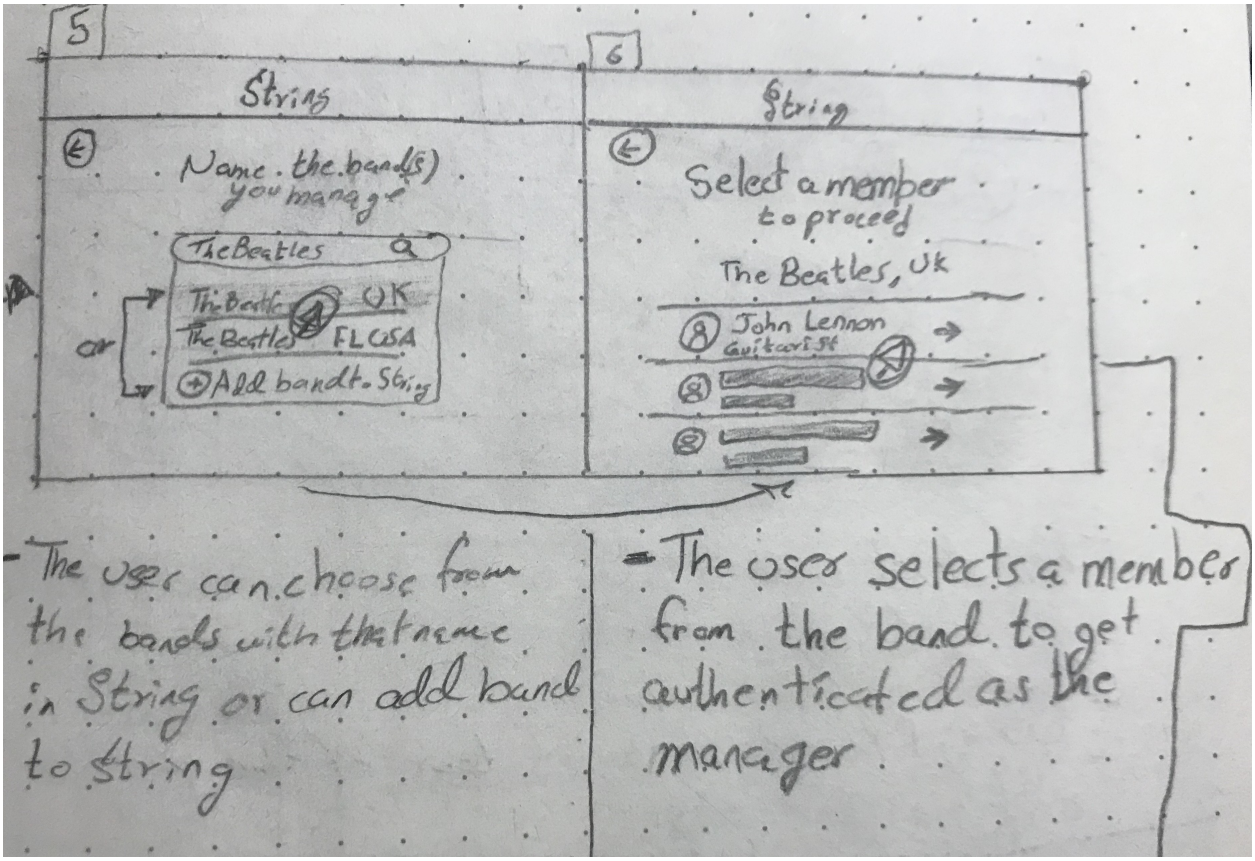
3.2 Use Case 2: A band manager onboards, searching to have one place manage/display all bands he manages

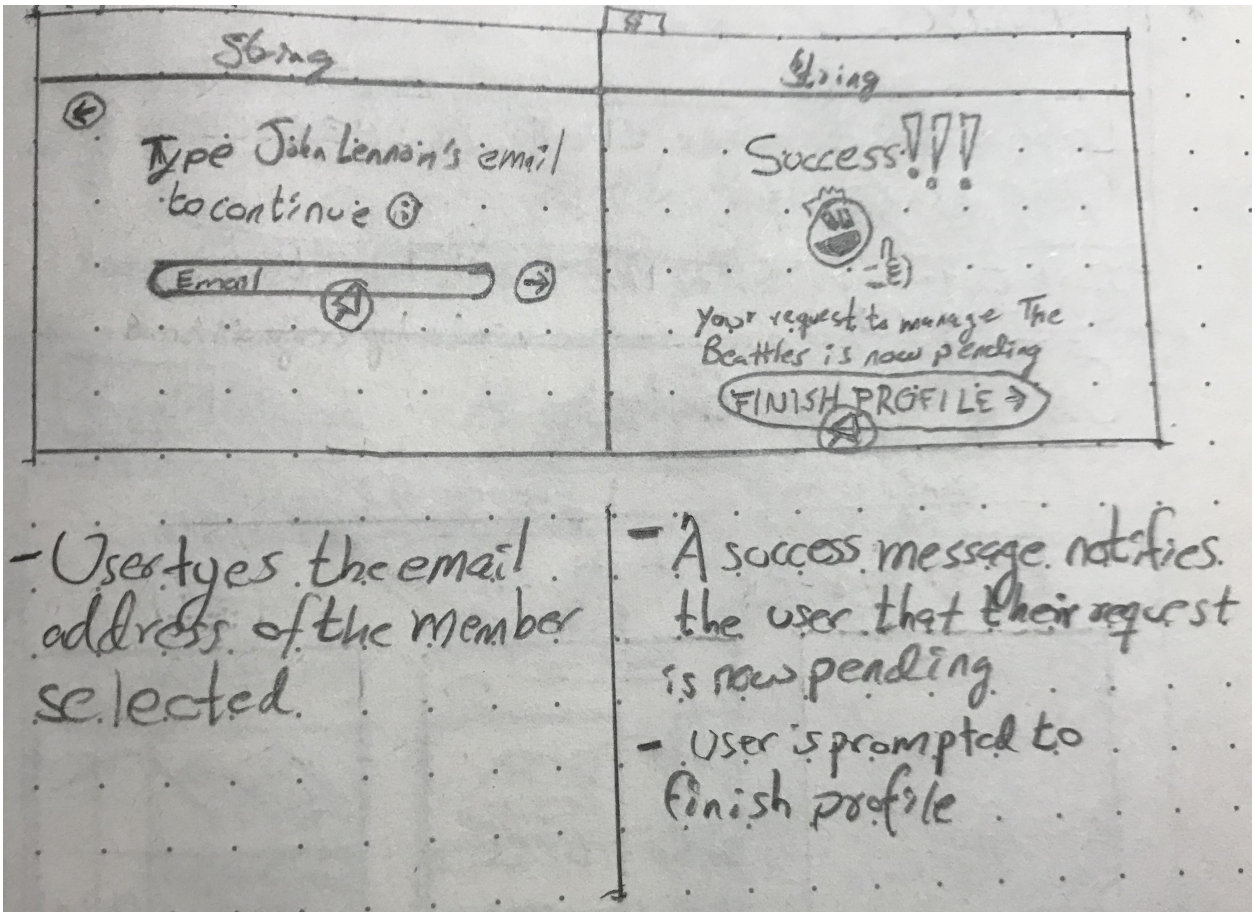
1. User clicks on the sign up button in the explore page or any other public page.
2. Onboarding user inputs: name, last name, email, phone number, and password.
3. User is sent a random 6-digit code through either email or phone
4. Then user authenticates the account by typing, replying, or opening link with matching code
5. Users select onboard as either: a solo artist, a manager, or a band.
6. Users that select to sign ups as "Band Manager" are prompted to a page/overlay with a searchbar.
7. "Name the band(s) you manage"
8. The user taps the name of the band and suggestions should appear right bellow it for bands already in String with that name or similar, relevant results should show a bit of added information to differentiate one from the other, for instance the city where they're from
9. There's also an option with a plus sign to "add the band to String"
10. On the next step the user will choose a member form that band (if the user selected to "add band to string" then manager shall add the email address, name and phone of at least 1 member to add the band to pending accounts to authenticate)
11. By tapping on a member profile of an existing account the user should be able to type that member's email to continue
12. there should be an button with with an "i" icon that reads "why we ask for this step"
13. By tapping on it a message should read: "As a band manager you'll get administrator access and tools, as such we require the band to authenticate you"
14. After imputing email a Success message should read: "your request to manage [name of band here] is now pending"
15. The user may tap on "Finish profile"
16. The registered user is taken to their profile, to edit.
17. String prompts a brief tour of the profile and main features.

18. A banner will remain at the top of the page with a % figure for profile completion. This allows users to pick up where they left off and have a walkthrough through all main features.



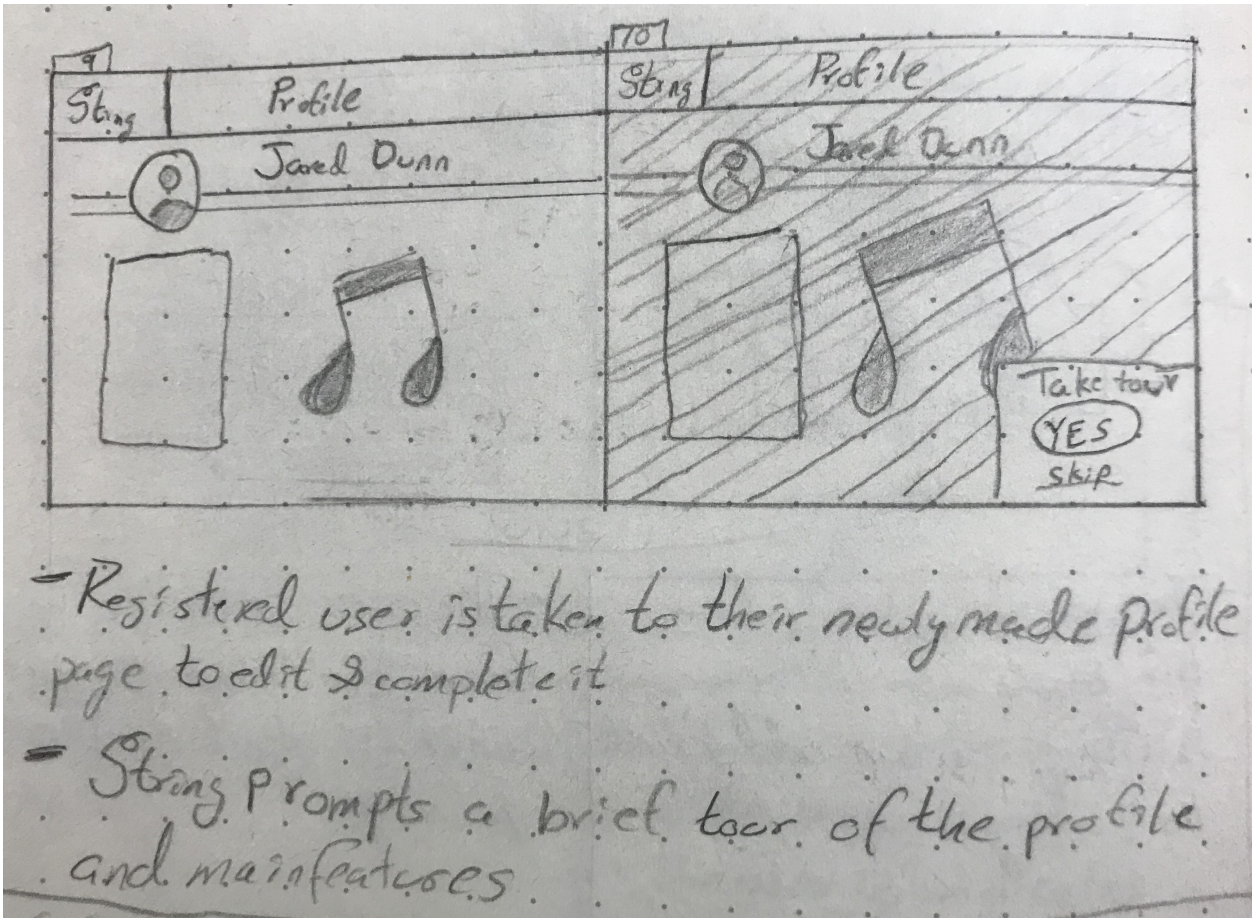

3
String
SignUp as:
Solo Artist
Manager
Band
String
Name the band(s) you manage
- User selects to onboard as a band Manager
- User is prompted to a page with a search bar to "Name the bands they manage"


5
String
Name the band(s) you manage
TheBeatles
TheBeatles OK
TheBeatles FLUSA
or
Add band to String
6
String
Select a member to proceed
The Beatles, Uk
John Lennon
guitarist
- The user can choose from the bands with that name in String or can add band to String
- The user selects a member from the band to get authenticated as the manager



- User types the email address of the member selected.

- A success message notifies the user that their request is now pending
- User is prompted to finish profile



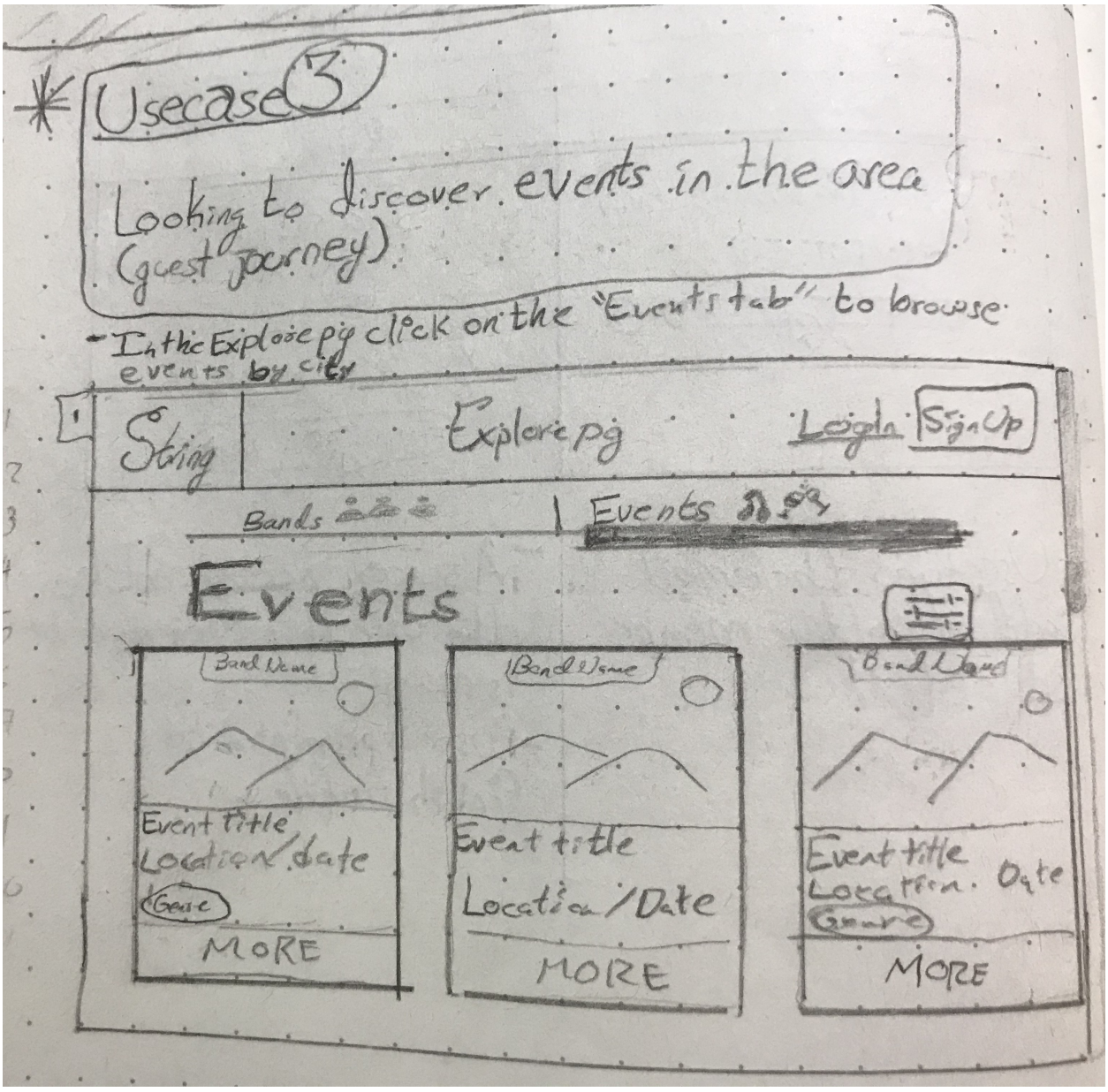
- Registered user is taken to their newly made profile page to edit & complete it
- String prompts a brief tour of the profile and main features

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.

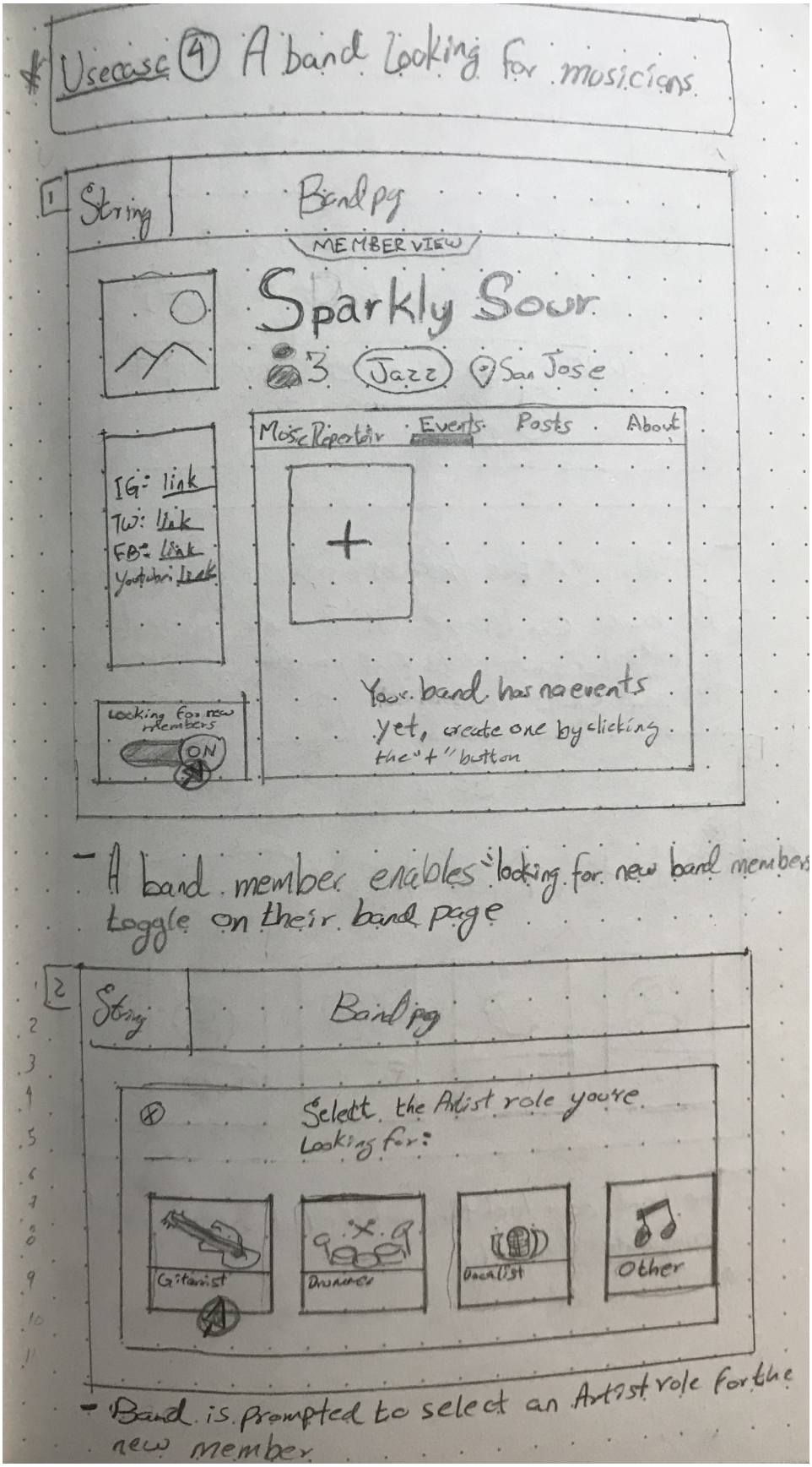
2. Scroll down to find more.

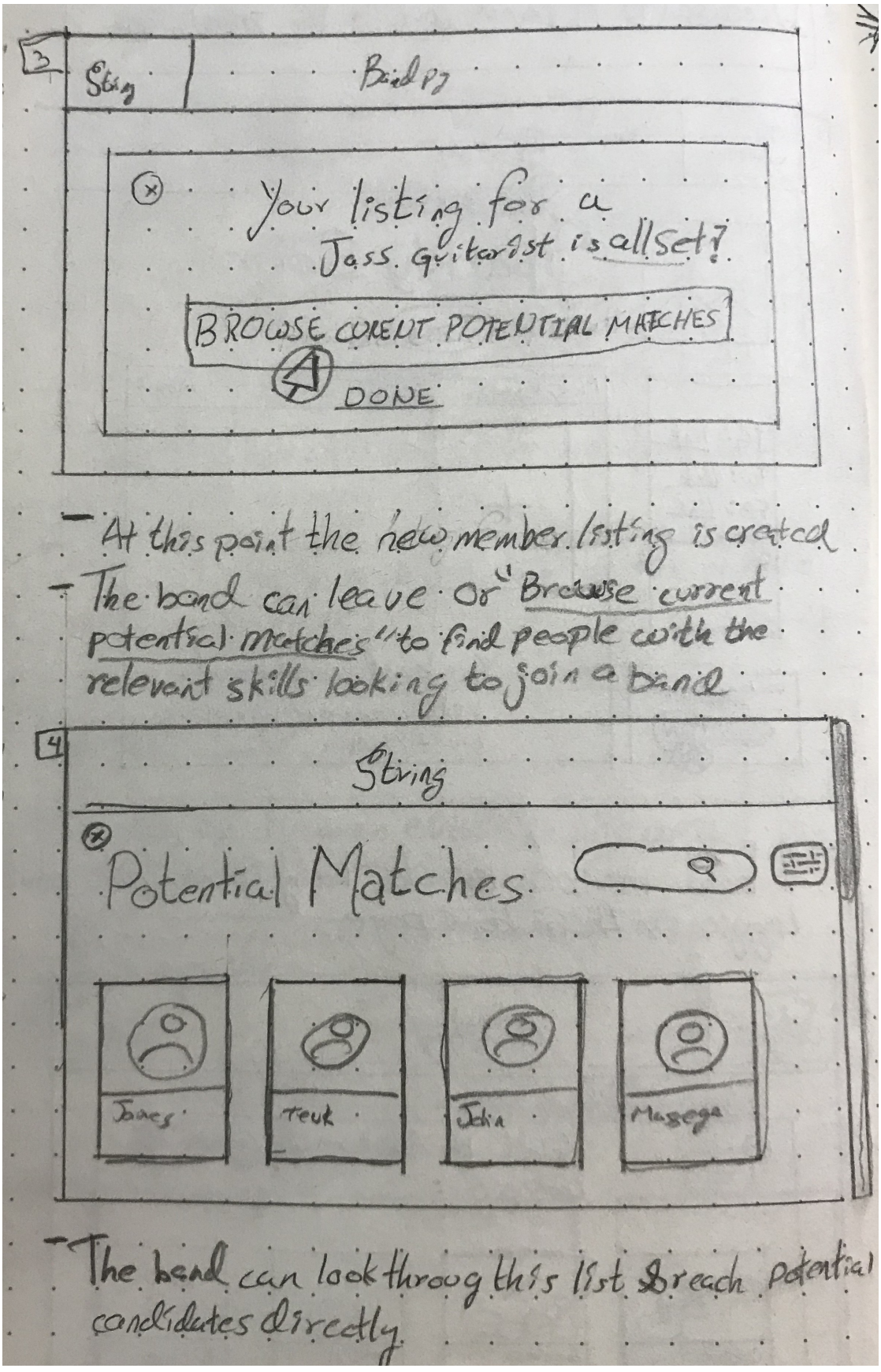
3. For filtering results of any kind the filter button located on the top right of the screen opens an overlay with filter preferences (i.e. zip, genre, city, size).



3.4 Use Case 4: A band looking for musicians

1. A band member enables "looking for new members" toggle on the band page.
2. Users are prompted to select the type of musician the band is looking for (i.e. guitarist, drummer, vocalist, etc.) They may also choose to make their own type. (This will enable the banner on band page and also makes it easy for the pool of potential candidates to filter bands out by the roles that apply to them)
3. At this point the user may also skip and go back to do what they were doing since the band will become visible to people looking to join a band.
4. Should the user that enabled the toggle decide to continue, they'd arrive to a list of all the people looking to join a band filtered by the role type the band is looking for.
5. They can then tap on any of the people there and look at their profile/reach out.


Usecase (4) A band looking for musicians
1
String
Band pg
MEMBER VIEW
Sparkly Sour
3
Jazz
San Jose
Music Repertoir
Events
Posts
About
IG: link
TW: link
FB: link
Youtube: link
+
Your band has no events yet, create one by clicking the "+" button
Looking for new members
ON
A band member enables "looking for new band members" toggle on their band page
2
String
Band pg
Select the Artist role you're looking for:
Guitarist
Drummer
Vocalist
Other
Band is prompted to select an Artist role for the new member

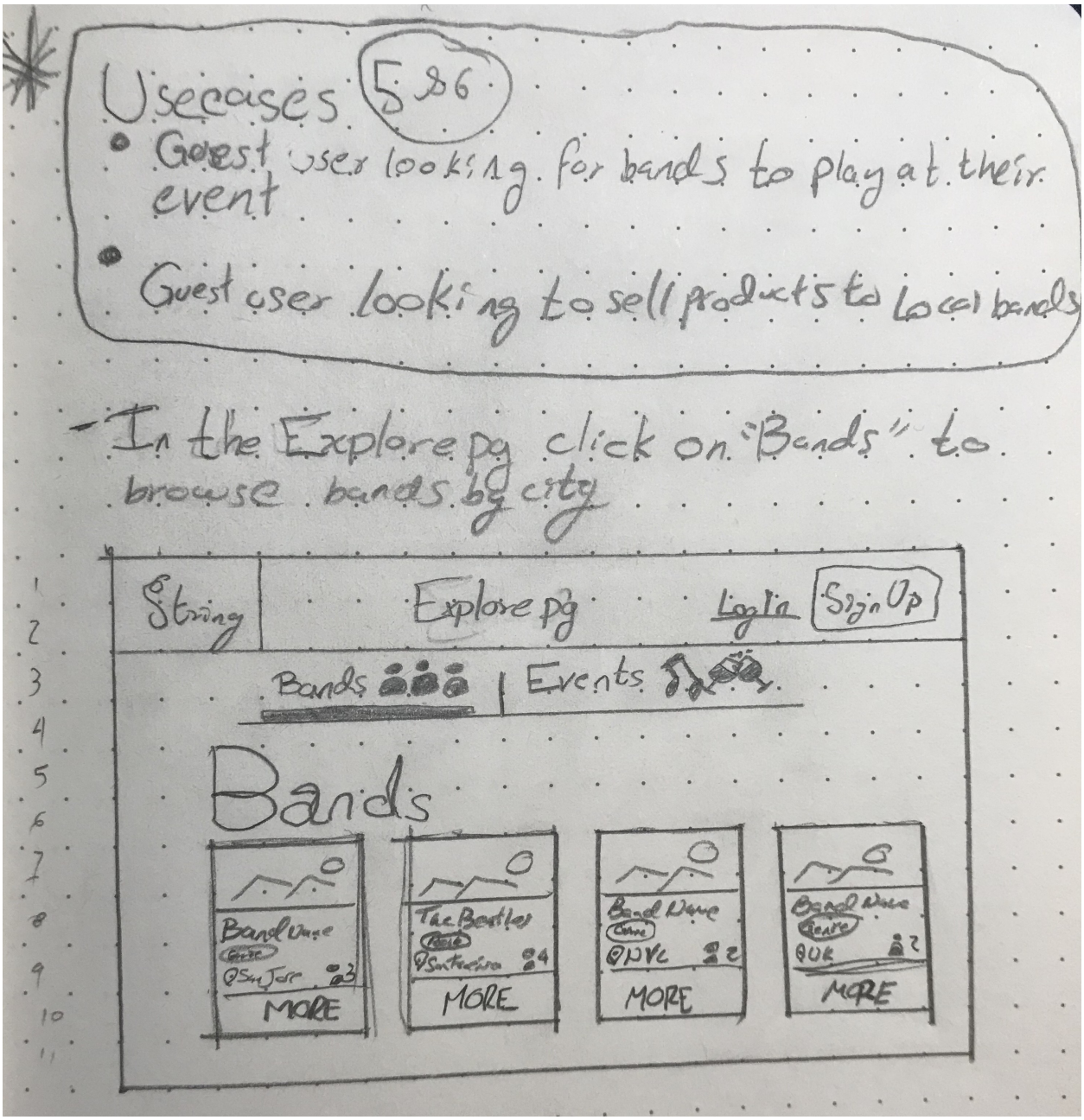


- At this point the new member listing is created
- The band can leave or "Browse current potential matches" to find people with the relevant skills looking to join a band

- The band can look throug this list & reach potential candidates directly

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.


Usecases
5 & 6
Guest user looking for bands to play at their event
Guest user looking to sell products to local bands
- In the Explore pg click on "Bands" to browse bands by city
String
Explore pg
Log In
Sign Up
Bands
Events
Bands
MORE
MORE
MORE
MORE

4 High level database architecture and organization

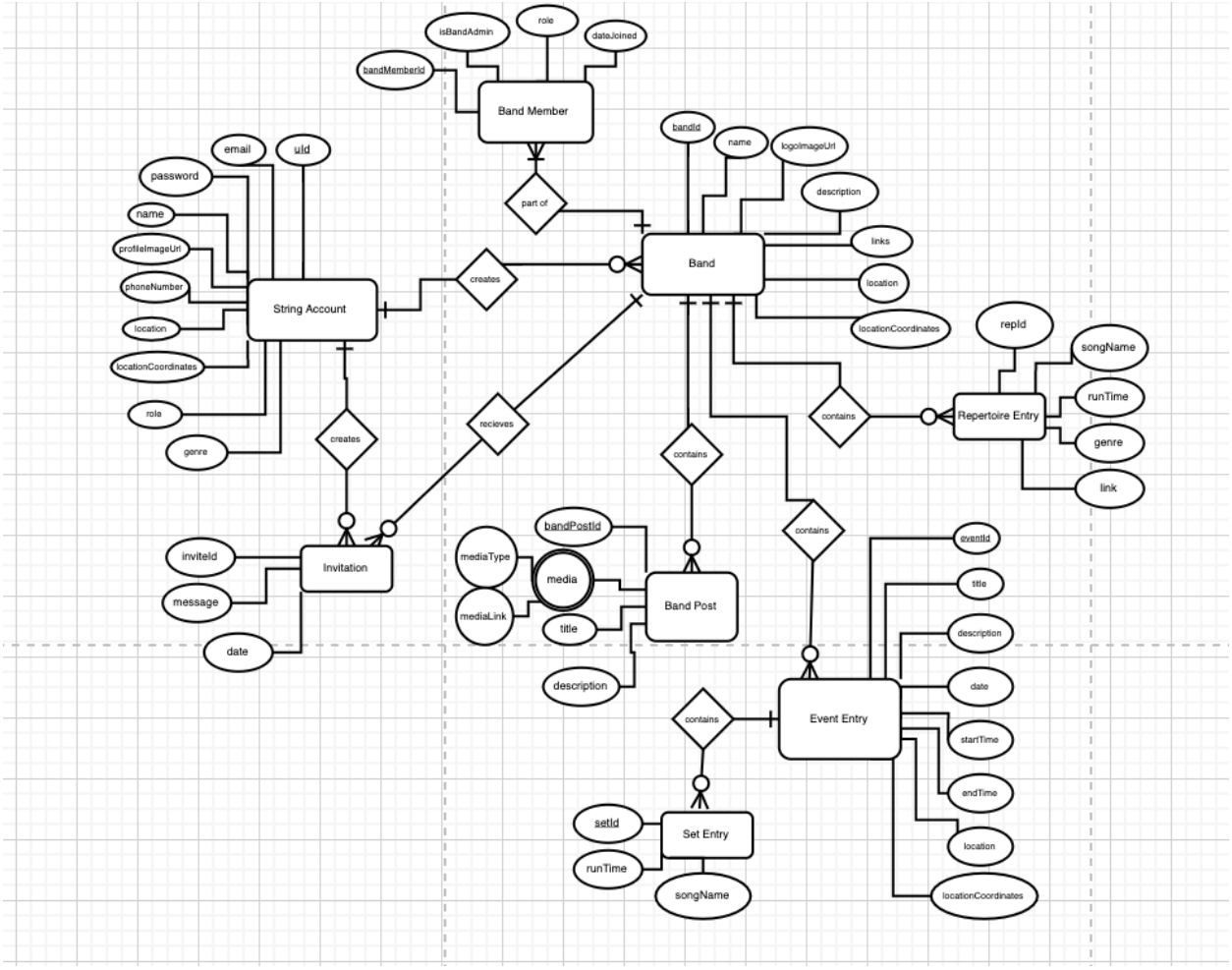
4.1 Business Rules

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)

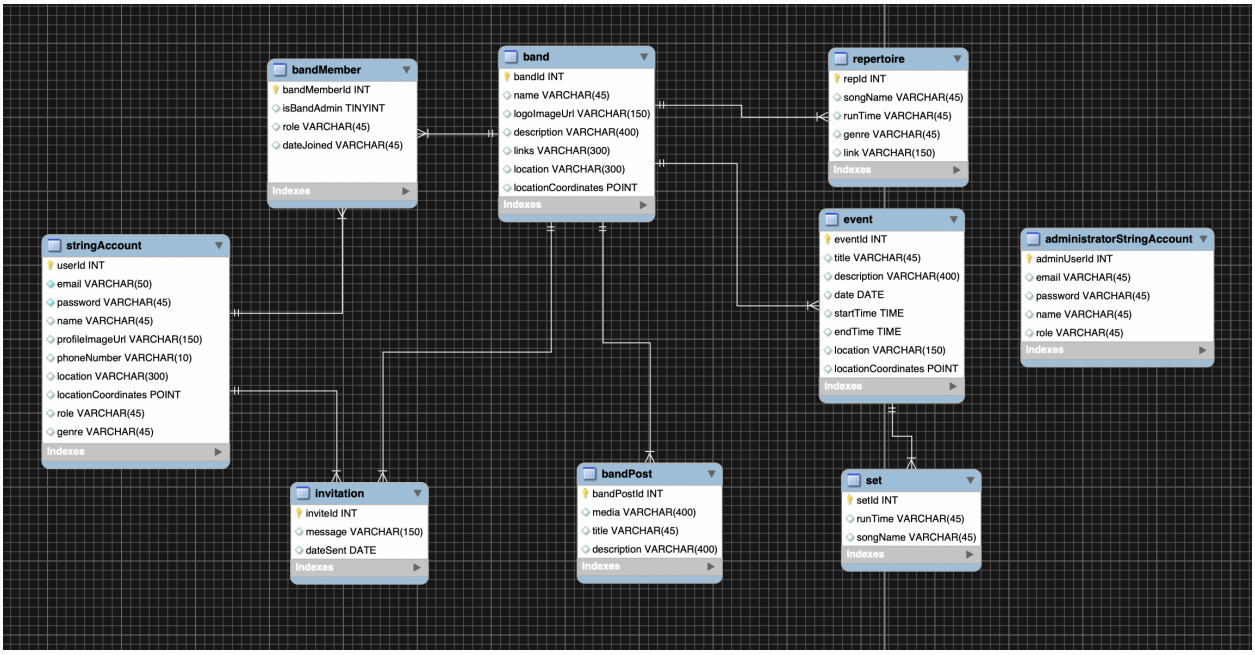
4.2 Entities Description

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

4.3 ERD


isBandAdmin
role
dateJoined
bandMemberId
Band Member
email
uId
password
name
profileImageUrl
phoneNumber
location
locationCoordinates
role
genre
String Account
part of
creates
bandId
name
logoImageUrl
description
links
location
locationCoordinates
Band
repId
songName
runTime
genre
link
Repertoire Entry
contains
creates
recieves
contains
contains
inviteId
message
date
Invitation
bandPostId
mediaType
media
mediaLink
title
description
Band Post
eventId
title
description
date
startTime
endTime
location
locationCoordinates
Event Entry
contains
setId
runTime
songName
Set Entry

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 geospacial functions to sort using this point and the user's location coordinates.

5 High Level APIs and Main Algorithms

- /api/register
  - POST request
  - Will be used to create a new String account
- /api/login
  - POST request
  - Will be used to log into an existing String account
- /api/getBands
  - POST request
  - Will be used to get Bands according to search value and filters applied
- /api/getEvents
  - POST request
  - Will be used to get Events according to search value and filters applied
- /api/getBand?bandId
  - GET request
  - Will be used to get Band corresponding to bandId
- /api/getEvent?eventId
  - GET request
  - Will be used to get Event corresponding to eventId

The following api are only accessible to a user logged into their string account:

- /api/user/changeLocation
  - POST request
  - Will be used to change the location of the registered user
- /api/band/create

  - POST request
  - Will be used to create a band

- /api/band/invite/send
  - POST request
  - Will be used to create an invitation to join a band

The following api are only accessible to a band member of a band:

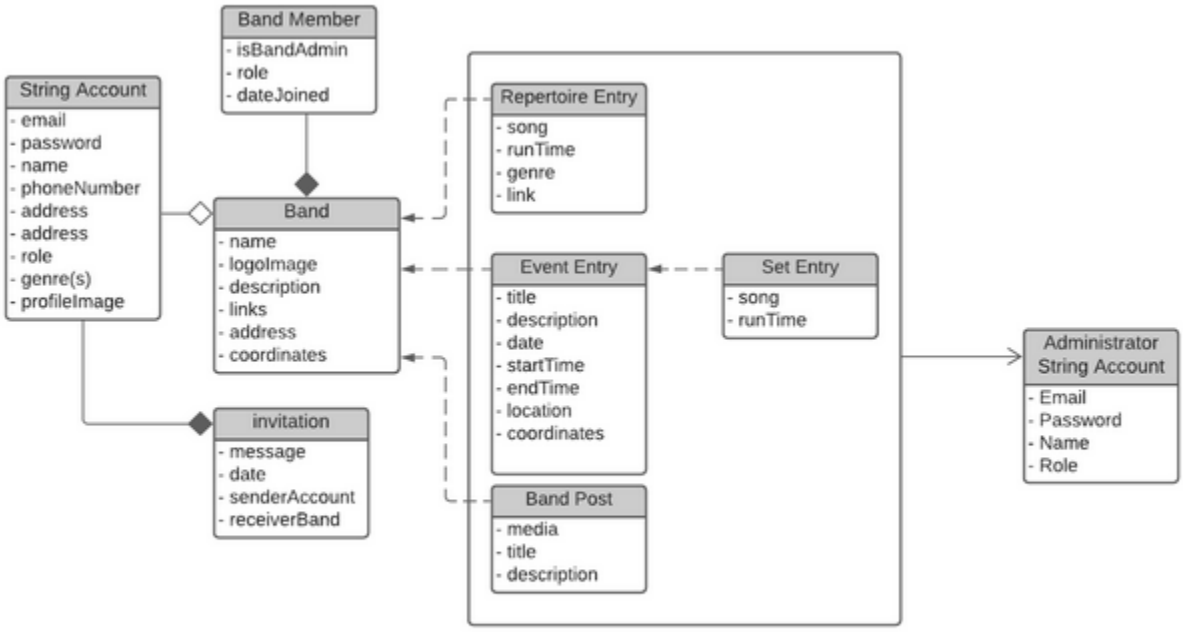
- /api/band/rep/create
  - POST request
  - Will be used to create a repertoire entry for a band
- /api/band/event/create
  - POST request
  - Will be used to create an event entry for a band
- /api/band/post/create
  - POST request
  - Will be used to create a band post for a band
- /api/band/rep/delete
  - POST request
  - Will be used to delete a repertoire entry for a band
- /api/band/event/delete
  - POST request
  - Will be used to delete an event entry for a band
- /api/band/post/delete
  - POST request
  - Will be used to delete a band post for a band

The following api are only accessible to a Band Admin of a band:

- /api/band/delete
  - POST request
  - Will be used to delete a band
- /api/band/invite/accept
  - POST request
  - Will be used to accept an invite to join a band from a registered user

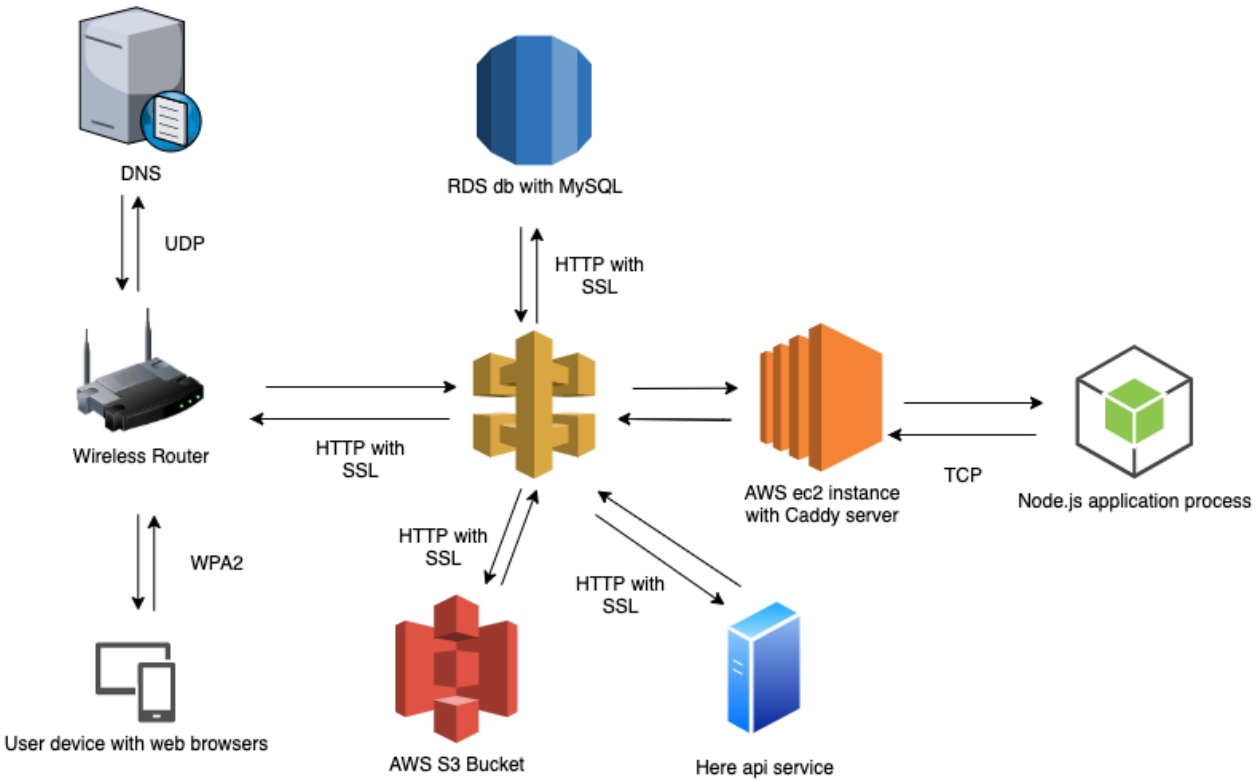
We have not changed any of our existing frameworks but have added AWS S3 to store image assets and will be using Here api to convert a location to its coordinates.

6 High Level UML Diagrams

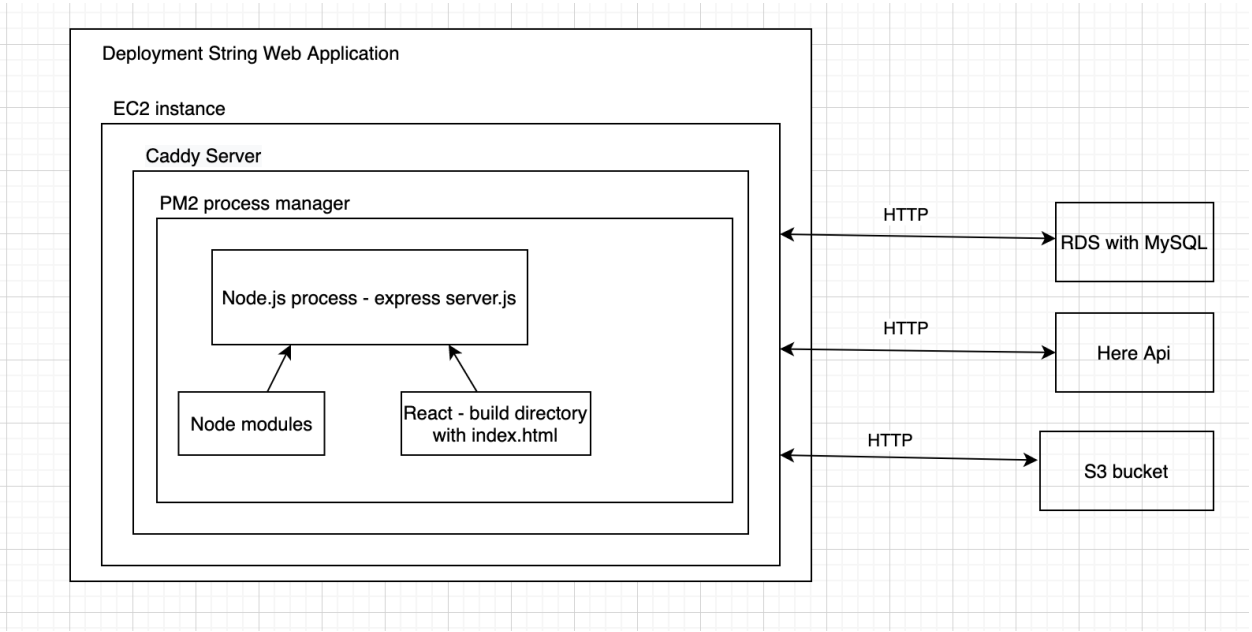


7 High Level Application Network and Deployment Diagrams

7.1 Application Networks Diagram



7.2 Deployment Diagram



8 Key Risks at time of writing

8.1 Skills risks

- No risks as currently assessed

8.2 Schedule risks

- Time shortage: Our project has a complex UI with different user views and can take longer than a month to complete
  - Complete priority 1 and basic functioning of the app before moving to priority 2 or optimising the app UI.

8.3 Technical risks

- Working with multer and s3 buckets could be challenging to manage images and assets.
  - Use youtube tutorials to follow along how to upload images using aws sdk and multer, such as this.
- Using redux to store the state persistently between screens has a steep learning curve.
  - Use boilerplate redux and change the functionality and methods to work with our requirements.

8.4 Teamwork risks

- No risks as currently assessed.

8.5 Legal/content risks

- Band uploading inappropriate content on the String platform.
  - Legal document with behavioral guidelines that only content that is legally permissible and age appropriate should be uploaded or used on the platform.
  - Use Administrator String account to monitor the postings and remove them if they are not in accordance with the terms & conditions of the platform.
- Band copying other band's work(images/videos) or Fraudulent bands

  - Our solution to this would be a set of legally binding terms users agree to when they register stating all work uploaded is their own and have rights to, and String is not responsible for any content that is not such.

- Safety of guests in the case of a fraudulent event (or other issues with the information posted on the platform)
  - Legal document that String is not responsible for user safety, that the user acknowledges that they are ultimately responsible for themselves, and that String's responsibility ends with hosting the information as received from the originator(s).

- Setting up the Terms and Condition documents.
  - Use boilerplate Terms and Conditions document and change it according to our needs and specifications.
  - In the event of further application development, legal consulting is a possibility.

9 Project Management

- We are using Trello as our project management tool:
  - We create cards that multiple team members are assigned to and notified of as they change from "current week - TO DO" to "IN PROGRESS" to "Ready for review - MERGE" (on applicable tasks) and finally to "SHIPPED - DONE."
  - On the board we also have a section for "References & Resources" where we keep things like the links to the shared google drive, the zoom room, and the milestone documents.
  - We also create further modifications to cards as applicable, for instance to do lists within cards this makes sure everyone is aware of what other members are doing at any given point.
- We distributed the parts of the milestone amongst the frontend and backend teams:
  - The Frontend team worked on part 3 and frontend of the vertical prototype.
  - The Backend team worked on part 4,5,6,7 and API development of the vertical prototype.
- We did updates before every meeting in order to come to an understanding of what both teams had worked on and were planning on doing until the next meeting
- We completed some parts: 1,2,8,9 together during the meeting time because they concern all team members.
- From next milestone on:
  - We would follow a similar management pattern.
  - Tasks that concern the team will be done during team meetings.
  - Tasks that concern a specific sub team will be done on their time and a report will be presented in the beginning of the meeting.
  - All the tasks will be uploaded on Trello which will be a central place to know who is working on which part of the application and its progress.

10 Detailed list of contributions

- Jainam Shah: (Team Lead)

    1. Organised two weekly meetings and built agenda and plans to complete the milestone in time.
    2. Managed and distributed tasks amongst team members.
    3. Worked on the defining data entities, prioritising functional requirements and indentifying key risks.
    4. Verified and completed the ERD and UML diagrams and database structure.
    5. Created the S3 and file management part of the backend and built a home screen as per design for the vertical prototype.
    6. Deployed the vertical prototype on the EC2 instance.

- Alfredo Diaz: (Frontend lead)

    1. Timely attendance and participation in discussions during meetings.
    2. Worked on the defining data entities, prioritising functional requirements and indentifying key risks.
    3. Created user journeys and clarified functionality of components on main screens and created the paper UI Mockups.
    4. Coordinated with the fronted team what components we could use from existing libraries assigned tasks to build basic components for vertical prototype.
    5. Shared card component vision by handing off high fidelity wireframes for vertical prototype.

- Leonid Novoselov: (Frontend developer)

    1. Participation in discussions during meetings.
    2. Worked on the defining data entities, prioritising functional requirements and indentifying key risks.
    3. Worked on creating paper UI mockups and storyboards.
    4. Worked on the display card component for the vertical prototype.

- Eric Chen: (Frontend developer)

    1. Attended weekly meetings and participated in discussions.

2. Worked on the defining data entities, prioritising functional requirements and indentifying key risks.

- Warren Singh: (Backend lead)

1. Timely attendance and participation in discussions during meetings.
2. Worked on the defining data entities, prioritising functional requirements and indentifying key risks.
3. Helped create Business Rules and database design.
4. Edited and Compiled M2 document
5. Coordinated with back end team on workflow for vertical prototype.
6. Contributed in defining routes for the vertical prototype.

- Ritesh Panta: (Backend developer)

1. Timely attendance and participation in discussions during meetings.
2. Worked on the defining data entities, prioritising functional requirements and indentifying key risks.
3. Contributed in defining business rules and creating network and deployment diagrams.
4. Contributed in building the MYSQL queries for the vertical prototype.