Project Name: TEMPO: Music Band Management
Andrew ID: pchaichi , oonojegh , nfajriya, rushabhs
Sections all member can attend: A B C D

Project Description

Members of the music band are having difficulty manage their events and song's information including the practice session, concert, playlist for each event, list of equipments for each event. With TEMPO, they can manage their band and focus on their music at ease.

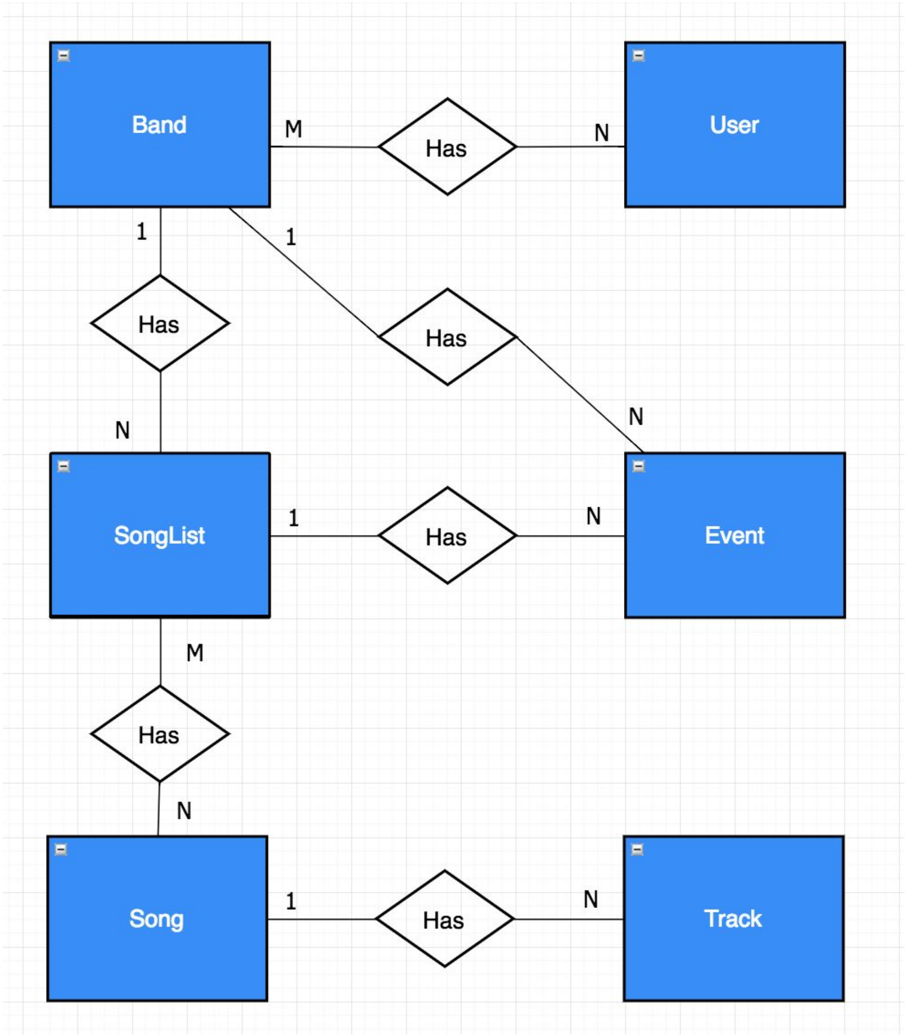
Planned Technologies

- Django, Bootstrap
- Calendar: https://fullcalendar.io/js/fullcalendar-3.5.1/demos/gcal.html
- Upload image: https://docs.djangoproject.com/en/1.11/topics/http/file-uploads/
- Export file (optional): https://docs.djangoproject.com/en/1.11/howto/outputting-pdf/
- Google Calendar API: https://developers.google.com/google-apps/calendar/ https://developers.google.com/google-apps/calendar/quickstart/python
- SoundCloud API : https://developers.soundcloud.com
- Login with facebook: https://simpleisbetterthancomplex.com/tutorial/2016/10/24/how-to-add-social-login-to-django.html
- Audio recorder: https://developers.google.com/web/updates/2016/01/mediarecorder https://www.html5rocks.com/en/tutorials/getusermedia/intro/

Draft Design

https://drive.google.com/open?id=0B5GA_qIhnBCaZWNlNXVKa0RKR0E

Data Models



- User (Django User model)
- Band : Group of user (id, name, list of member)
- SongList (id, name, list of song)
- Song (id, name, tempo, list of track, Lyric-Chord file)
- Track (id, name, type, audio file, version_number)
- Event (id, name, location, time, SongList)

Features (Notes: ★ is the main new features)

- Homepage
  - See the list of the bands
  - User can request to join the band
  - User can access the band's information that they have permission to
- Profile page
  - List of user's events
  - List of user's bands
  - User has the ability to leave the band
- Each Band has ..
  - Team Member Page
    - User can see the list of team members
    - Admin user can add new members to join the band

  - Home Page : Calendar
    - User sees all practice sessions and live sessions schedule on the calendar
    - User can add the schedule to their Google calendar
    - Each event has the link to the related model e.g. song list, live session
  - Live Session (Event)
    - User can add/edit/delete the live session
    - Live Session data displays on 'Home Page: Calendar' page
    - Live Session Data
      - Practice date and time (shown in Calendar)
      - Session place
      - Type: Concert, Practice, Others
      - Participants (Link to user profile page)
      - Song list (Link to song list)
      - Equipment needed
      - Notes
  - Song List
    - User can manage their songs which will be used in the live session
      - Add/Delete songs to a particular list
      - Edit the name of the list and option to change order of songs in the list
    - One Live Session can have one song list
    - One song list can have many song tracks

        - Song
            - Song name
            - Option to go to previous or next song.
            - Lyric & Chord: Uploaded photo
            - Audio Recorder for each track (for different instrument & version)★
            - Audio Player for each track and the master ★
            - SoundCloud or other apps Widget (optional)
- Other features
    - Login/Registration
    - User Permission
        - Grouping, 1 user can belong to multiple band
        - Permissions - (Read-only, Edit)
    - Sharing (optional)
        - Generate a shareable like for the song list, live session, of practice session
    - Export file
        - Download the song data (chord & lyric) in form of jpeg or pdf
    - Deploying to Heroku

Tasks Assignment

- **Nurlaili** : Live Session(Event), Calendar (fullcalendar library) , Google Calendar API, Base Template, Deploying to Heroku for testing with user
- **Mook** : Song Management: Song List(album), Song, Recording, Image Uploader, SoundCloud Widget, UX/UI design, Audio player
- **Rushabh**: Homepage, Profile Page, Sharable pages for song list and event (ref: Dropbox share like), Export/Download files
- **Omo**: Team Member VS Band Permission: Band request, Admin vs Regular user, LogIn, Registration, Log in with Facebook/Google account, Notification

Additional Details for Large Team: Member's experiences

Rushabh(rushabhs) : Worked for a MNC and helped develop functionalities for finance module using SQL, PL/SQL; Worked on python projects and libraries like pandas to analyze structured/ unstructured data; experience with numerous Java projects at Heinz in courses like Java and Distributed Systems; experience with docker and spring framework during the summer internship; Non-Technical: Been Business developer intern for 2 start-ups and handled the business strategy and growth for those start-ups

Nurlaili (nfajriya) : Worked in a school project with 4 members to build a web app using PHP and CI framework (2013), analyzed and designed a transactional system in an online ticketing company; including payment, refund, and reschedule processes (2016), had some school projects on Java and Distributed System courses in Heinz College.

Mook (pchaichi) : Worked with iOS applications/Unity using objective C, C++, Implemented various projects with Java, Teaching Assistant for Distributed System(Heinz College), (non technical experience) Leading development team building webapp using Django, Python

Omoefe (oonojegh): Worked on a Distributed System project to build a simple Andriod terminal which receives data from a webserver hosted on the cloud (2017). Non Technical experience with Project and Product Management, and an intended responsibility to identify user requirements, feasibility and easy of use.