

PRODUCT BACKLOG

Meeting Time:

Monday 6:00pm, Hunt library

Overall app/modules:

- User app (optional)
- Music creation modules (related to workspaces, tracks, MIDI creation, music sythethzing, etc.)
- Community modules (related to social platform elements)

Overall set of features:

Music creation -> Workspace Module:

1. Add track: User can click to add another track with a selected instrument.
2. *Velocity Adjustment: For each track the user could adjust velocity/volume. We will plan this function according to the functionality of backend sound library. Will not be a priority.
3. Playback: Play, pause, rewind the synthesized song on the web page. (The availability of adjusting speed might be added if supported by backend sound library).
4. Publish: When satisfied with playback results the user can permanently export the song and publish the work to his portfolio. Other people will be able to listen.
5. Collaboration: Access control (as suggest by TA and professor) will be implemented so that different user can work in the same workspace. Workspace owner can open permission to other users, in this way one user can edit at one time in the workspace but others can playback as well.

Music creation -> Track Module (within workspace):

1. Music note editing: A module that allows users to add musical notes by clicking on the "track timeline". Different notes can be chosen acrosses octaves.
2. Playback: Play, pause, rewind the synthesized song on the web page. (The availability of adjusting speed might be added if supported by backend sound library).

Community:

1. Browse through global stream of published songs
2. Follow other users
3. *Optional: Search user by username (implement later)
4. View a user' portfolio. User portfolio contains current user's picture, description, albums and songs.
5. Like: Liking the songs in portfolio. The songs could optionally be placed under an album according to user needs. Album will have its description and picture.
6. Comment: A song will be able to receive comments underneath.
7. Playback: Even on the portfolio page we still provide a playback function - so that is is more convenient for people to share songs and enjoy melodies.
8. *Fork: User can fork (a.k.a make a copy of the song's workspace and start editing again)

User module:

1. Log in/out.
2. Register.

3. Change basic info like profile picture and description.

DATA MODELS

1. User model: Store info about user accounts, possible social linkage, etc. Fields include username, password, email, picture, description.
2. Collaboration Groups: Will be mapped to a workspace model, users in the group will be getting relative access control. All possible collaborative/access control information could go here. Fields include users and access control info for each respective users (allowed to edit or not, for example).
3. Workspace model: A workspace will be constructed by all the track information. A track is essentially a collection of musical notes across several minutes of time with one instrument assigned. In the end all the tracks composed in different instruments will be synthesized together during playback.
4. User portfolio: User portfolio contains current user's albums and songs. Published musical pieces could be stored in user portfolios. Nested within the portfolio model could be possible albums (another data model), etc. for which we store likes and comments and other related information. Fields could be linked to song or album (there will be a database field to distinguish between them).
5. Song Model: Song in a portfolio will have its name, description, picture and genre and the album it belongs to. It also has a link to a wav file. *If null then it will be put at a "root" location. Additionally it will point to the workspace it was created in.
6. Album model: Album in a portfolio will have name, description, picture.
7. Track model: as mentioned above multiple track will be mapped to a workspace. It will store musical note information such as timing, length and pitch. It will have musical note information that should be translated into MIDI and we will be implementing the wheels by ourselves here. We will work further on the API.

Community app data models:

1. User model
2. User portfolio model
3. Song Model
4. Album Model

User scenario of global stream(community):

1. Display all music work in database.
2. Click on certain music piece can enter its own information page
- 3.

Meeting Time:

Monday morning 2 hours

The first sprint backlog

The name and Andrew ID of the product owner for the first sprint:

Jieying Xu, jieyingx

Backlog specification:

1. Two static web pages to display user workspaces, the first web page is aimed for adding different instruments, following each instrument there will be a bar showing music track and is able to be clicked. If the bar is clicked, the user will be guided to another web page for adding different notes under that instrument to form a melody. (Kejing is responsible for implementing the two HTML pages)
2. Backend functionality (experiments mostly): Find a library that can synthesize sounds out of MIDI files. Demonstrate how that can be done in command line. Try to find a good library written in python that can manipulate and create MIDI files and write a hello-world python script to demonstrate.

*1 by Kejing, 2 by Zeyuan and Jieying

The second Sprint Backlog

The name and Andrew ID of the product owner for the second sprint:

Kejing Meng

Backlog specification:

- **Workspace**
 - Add track to workspace
 - Have a preview of track
 - Select instrument of track
 - Edit track and add note
 - Play, pause, etc.
 - Generate a wav and playback
 - Publish a “song” (with was and lined workspace)
- **Community**
 - View global stream
 - Follower/Followee
 - Playback
 - Organize Albums and songs
 - *Move songs to relocate to different albums
 - Edit pictures and descriptions, etc. of Song and Album
 - Comment, like
 - *Global music player

*Workspace by Kejing and Zeyuan, community by Jieying