

FCS 30700

MelodyMunk

Team 24: Sprint 1 Planning Document



MelodyMunk

Matt Twardowski, Brooks Van Buren, Elizabeth Stallter, Tristan Pickett, Shrikar Peri, Avnish Bablani

Table of Contents

Sprint Overview	3
Risks and Challenges	4
Current Sprint Details	5
User Story #1	5
Acceptance Criteria:	5
User Story #2	6
Acceptance Criteria:	6
User Story #3	7
Acceptance Criteria	7
User Story #4	8
Acceptance Criteria	8
User Story #5	9
Acceptance Criteria	9
User Story #6	10
Acceptance Criteria	10
User Story #7	11
Acceptance Criteria	11
User Story #8	13
Acceptance Criteria	13
User Story #9	14
Acceptance Criteria	14
User Story #10	16
Acceptance Criteria	16
User Story #11	17
Acceptance Criteria	17
User Story #12	18
Acceptance Criteria	18
User Story #13	19
Acceptance Criteria:	19
User Story #14	20
Acceptance Criteria	20

User Story #15	21
Acceptance Criteria	21
Remaining Backlog	22
Functional Requirements	22
Non-functional Requirements	25

Sprint Overview

This sprint will be focused on setting up the backbone system of our webapp. This will include making an AWS database, setting up AWS hosting, and setting up a website interface. In addition to the minimum setup, we will be looking to get basic functionality of making rooms, joining rooms, maintaining a playlist, and having a login system that holds basic user preferences.

Team Leader: Brooks Van Buren

Meeting Schedule: Tuesday/Thursday 9pm, Sunday 4pm

Risks and Challenges

It might be difficult to set up some of our systems without knowing how we might implement things further down the line. If we decide to change how something works or our plan for how we implement something, it might change some code that we are writing now.

Another issue can be that since this is our first sprint, we will be learning each other's coding styles and habits, and will have to learn how to work with each other and divide work well.

Current Sprint Details

User Story #1

As a user, I would like to see a landing page when I visit MelodyMunk, that gives me information about the webapp and see links to login and registration pages.

#	Task Description	Estimated Time	Owner
1.	Create static homepage	3 hours	Shrikar
2.	Link to other webpages	2 hour	Shrikar
4.	UI for Error Handling	2 hour	Shrikar

Acceptance Criteria:

- Given that I'm not logged in, when I visit the homepage, I will see the homepage with the MelodyMunk Logo and see an overview explanation of what the website does.
- Given that I'm on the homepage, when I skim the page, I can easily find links to login and register on the webapp.
- Given that I'm on the homepage, when I view the page on desktop or mobile, the design and layout of the webpage is easy to read and find information on.
- Given that I'm not logged in, when I visit the homepage, I can follow a link that allows me to create a room without signing in.

User Story #2

As a developer, I would like to set up an AWS instance and MySQL database to use in future development.

#	Task Description	Estimated Time	Owner
1.	Setup development EC2 instance on AWS	3 Hours	Matt
2.	Setup production EC2 instance on AWS	3 Hours	Matt
3.	Install MySQL database on dev and prod instances	2 Hours	Matt
4.	Set up SSH access for dev and prod instances	2 Hours	Matt
5.	Allow outside connection to dev and prod MySQL databases	3 Hours	Matt

Acceptance Criteria:

- Given that I log into our AWS account, when I view our dashboard, I can see our development and production instances, configure them, and connect to a terminal on the instances.
- Given that I connect to the development or production instance, when I try to connect to the MySQL database, I can login as root
- Given that I am on any machine, when I try to SSH into the dev or prod server, I can log in as a generic team account
- Given that I am on any machine, when I try to connect to a MySQL server on dev or prod, I can log in as root

User Story #3

As a user, I would like to create an account on MelodyMunk.

#	Task Description	Estimated Time	Owner
1.	Create User Entity	3 Hours	Matt
2.	Create RegistrationController to handle user registration	5 Hours	Matt
3.	Create Registration Page UI	3 Hours	Elizabeth

Acceptance Criteria

- Given that the RegistrationController is implemented successfully, when a user submits the registration form with valid input, then it will create a row in the users DB table and an account will be created.
- Given that the RegistrationController is implemented successfully, when a user tries to register with an email address already in use, then upon submission the user is notified that email is already taken.
- Given that the registration page is designed well, when a user attempts to create an account, the page will be easy to use.

User Story #4

As a user, I would like to log into an account on MelodyMunk.

#	Task Description	Estimated Time	Owner
1.	Create AuthController to handle the creation of new users	3 Hours	Matt
2.	Create Login Page	3 Hours	Matt
3.	Unit testing to make sure new user is created	1 Hour	Elizabeth
4.	Implement login throttling for each user to prevent brute force	3 Hours	Brooks

Acceptance Criteria

- Given that the AuthController is implemented successfully, when a user enters valid inputs for login, then the user will be retrieved from the database and logged in.
- Given that the AuthController is implemented successfully, when a user enters invalid inputs for login, then the user will receive an error message describing what to fix, and they will not be logged in.
- Given that the AuthController and database are designed correctly, when a malicious user attempts to brute force another user's login password, the webapp will throttle their login attempts.
- Given that the AuthController is implemented securely, when a user attempts to login with a non-existent username or email, then they will receive an error that is indistinguishable from when the username or email exists.
- Given that the AuthController is implemented securely, when a user attempts to login with a non-existent username or email or incorrect password, then the error displayed will always say that the "username or password" is incorrect.

- Given that the user account is being throttled, when a user attempts to log into that user account, then a message will display telling the user that they are making too many login attempts and they need to wait before trying again.

User Story #5

As a user, I would like to customize my account and link my Spotify account to it.

#	Task Description	Estimated Time	Owner
1.	Create UserController to handle features that relate to the user	5 Hours	Matt
2.	Create SpotifyController to handle requests to Spotify's Third-Party API	5 Hours	Matt
3.	Create UserAPIToken Entity	3 Hours	Avnish
4.	Create User settings page UI	4 Hours	Shrikar
5.	Implement ability to link Spotify account to a user	4 Hours	Avnish

Acceptance Criteria

- Given the UserController is implemented successfully, when valid data is sent to controller, then the user's account will be updated with the new data.
- Given the SpotifyController is implemented successfully, when a user enters in correct spotify credentials, then it will return the access and refresh tokens and save them in the database.
- Given the user settings form is implemented successfully, when a user submits the form with valid input, then a request to the UserController will be sent with the input provided.

User Story #6

As a party guest I want to be able to join a room.

#	Task Description	Estimated Time	Owner
1.	Designing a room module that handles creating a room and updating attributes	6 hours	Elizabeth
2.	Create module to find room identified by URL	5 hours	Elizabeth
3.	Create UI to join and create rooms	4 hours	Elizabeth
4.	Create UI for URL input. Design error message dialog box if input is invalid.	4 hours	Shrikar

Acceptance Criteria

- Given that a user is on our webapp, when they input a valid URL to a room, they will be added into that room.
- Given that a user is on our webapp, when they input an invalid URL to a room, they will be brought to the homepage and shown an error message.
- Given that a user is on our webapp, when they join a room, that room will update the UI to show the new user and the host will be notified.

User Story #7

As a user I would like to be able to search for songs.

#	Task Description	Estimated Time	Owner
1.	Connect with Spotify API to be able to search for songs and make queries to search songs.	5 hours	Avnish
2.	Create a UI to search through songs and allow users to select the correct songs.	3 hours	Shrikar
3.	Play proper song from Spotify API by calling it to get the song.	2 hours	Avnish
4.	Unit testing to verify song requested is correctly called and with the right song	1 hour	Avnish
5.	Error handling for Spotify API	3 hours	Avnish

Acceptance Criteria

- Given that the Spotify API is connected, when a user types a search term in the search box for songs, users will be shown matching songs that are available on Spotify.

- Given that there is sufficient error handling, when an API call fails, a user will be notified that an error has occurred, and developers will be sent information regarding the error that has happened if it is uncommon.
- Given that song selection UI is done, when users have a list of Spotify songs they searched for, users can select songs to recommend, add into the queue, or vote on.

User Story #8

As a host I would like to generate a QR code that can be displayed on a screen or printed to share with guests

#	Task Description	Estimated Time	Owner
1.	Create module to generate QR code for given URL	3 hours	Brooks
2.	Allow party host to view QR code in web page	2 hours	Brooks
3.	Make QR code easily printable	1 hour	Brooks
4.	Add UI button to get sharable link or QR code	3 hours	Brooks

Acceptance Criteria

- Given that I own a room, when I click a share link in the UI, I will be given a sharable link to the room.
- Given that I own a room, when I click a share link in the UI, I will be given the option to generate a QR code.
- Given that the QR code is easily printable, when I generate a QR code, I can print the QR code which is readable by smartphone QR readers.

User Story #9

As a host, I would like to be able to specify a list of actions guests and admins have permission to take

#	Task Description	Estimated Time	Owner
1.	Create a module to specify a list of permissions associated with a room	5 hours	Brooks
2.	Allow module to be queried against in order to check if an action is permitted	2 hours	Brooks
3.	Interface module with database to save and retrieve preferences across sessions	3 hours	Brooks
4.	Interface module with UI to allow host to specify permissions	3 hours	Elizabeth
5.	Unit testing to ensure that permissions work properly	1 hour	Avnish

Acceptance Criteria

- Given that the permissions are set for a user, when the module is called from another part of the code, then the action can be verified as valid for a specific user.
- Given that the module is designed correctly, when permissions are added or updated, the corresponding rows in the database are updated so that the permissions persist.

- Given that the UI is designed successfully, when a host goes to update the permissions set for a room, they will be met with a checklist for each role's possible permissions.

User Story #10

As a host, I would like to specify users as admins.

#	Task Description	Estimated Time	Owner
1.	Create module that contains info for users in a room and which role they are	5 hours	Brooks
2.	Interface module with UI to allow host (or admin) to specify a user as an admin	3 hours	Tristan
3.	Interface module with database to save and retrieve preferences across sessions	3 hours	Tristan/Avnish
4.	Unit test to verify promotion to admin	1 hour	Avnish

Acceptance Criteria

- Given that I am a host, when I wish to allow other users advanced privileges, then I can promote them to admin status.
- Given that I am a host, when I wish to define what permissions roles have, I can use a check-list for the roles Admin, Guest, and Anonymous Guest.
- Given that I am calling this module as a developer, I can check whether or not a particular user have permission to perform an action in a specific room.
- Given that I am a host, when I reuse a room for another session, my previous permission specifications will persist from the last session.

User Story #11

As a general user I would like to suggest songs.

#	Task Description	Estimated Time	Owner
1.	Take song from Spotify API	4 hours	Avnish
2.	Add song to suggested queue	3 hours	Avnish
3.	Unit Test to verify that songs are properly suggested	1 hours	Avnish

Acceptance Criteria

- Given that a user is in a room, when they search for songs, they can suggest them to the queue.
- Given that a song has been suggested, when a host looks at the queue, they can see suggested songs from users.
- Given that a song has been suggested, when a guest looks at the queue, they can see suggested songs from other users and vote on them. (thumbs up)

User Story #12

As an admin, I would like to be able to restrict room access to specific users

#	Task Description	Estimated Time	Owner
1.	Adding to the module that manages users in a room, create a whitelist and blacklist of users	5 hours	Brooks
2.	Allow the room to be switched between blacklist mode, whitelist mode, or neither	4 hours	Tristan
3.	When attempting to add a user into a room, verify the user is allowed based on the current restriction mode	3 hours	Tristan
4.	Unit testing to ensure restrictions hold	1 hour	Avnish

Acceptance Criteria

- Given that I am a host, when I desire to restrict access, then I can switch between various user restriction modes to my liking.
- Given that a user attempts to join a room, when they connect to the URL, then it is determined if they are a valid user.
- Given that a user is blacklisted in a room in blacklist mode, when they try to join, then they are shown an error message.
- Given that a user is not whitelisted in a room in whitelist mode, when they try to join, then they are shown an error message.

User Story #13

As a host I would like to be able to create a room for others to join

#	Task Description	Estimated Time	Owner
1.	Creation of distinct URLs when created for requests for a new room	3 hours	Tristan
2.	Keeping track of which URLs are currently taken for other rooms	3 hours	Tristan
3.	After rooms are no longer in use allow URL to eventually be recycled back into potential URLs for new rooms	4 hours	Tristan
4.	Ensure distinct URLs and recyclable URLs through unit testing	1 hour	Avnish

Acceptance Criteria:

- Given that a user is on our webapp, when they attempt to create a new room, they will be given a succinct creation wizard that allows them to set options.
- Given that a user requests a new room, when the room is created, then a new URL is assigned for that room
- Given that a room is no longer in use, when a long enough time period has passed to ensure no remnants from the room (previous URLs handed out to friends), then the URL can be allowed to be used again

User Story #14

As a user I would like a ranking system for voted on suggested songs in a room.

#	Task Description	Estimated Time	Owner
1.	Allow users to vote on suggested songs	5 hours	Tristan
2.	Store votes on songs in the room	4 hours	Tristan
3.	Rank songs based off of number of upvotes, time passed since suggestions, and current rate of upvotes	3 hours	Tristan
4.	Unit tests to make sure songs are properly rated and ranked	1 hours	Tristan

Acceptance Criteria

- Given a user sees a song on the suggested list, when they find they like the suggestion, then they have the ability to upvote that song
- Given that a user upvotes a song, when the upvote is processed, then the rankings will immediately shift to reflect the updated standings
- Given that a user does not realize they already upvoted a song, when they attempt to upvote it again, that action will be barred
- Given that a user causes an error, when they view the UI, then they will see a UI error message

User Story #15

As a user I would like to have a user-friendly website interface.

#	Task Description	Estimated Time	Owner
1.	Design login page	3 hours	Elizabeth
2.	Design room info page	4 hours	Shrikar
3.	Design playlist page	8 hours	Shrikar
4.	Design chat page	5 hours	Elizabeth

Acceptance Criteria

- Given the requirement of a login page to access our web application, when I want to log in, I would like to implement two separate text fields, one for username and one for password, with a button to initiate authentication.
- Given access to a room, when view the room I would like to see a list of users, a room photo, and the room title.
- Given access to a room, when I select a playlist, then I would expect a banner showing the current song, songs in the queue, a pop-up search option, and the ability to upvote songs in the queue.

- Given access to a room, when I would like to communicate with other users, then I would expect a chat interface where users are easily distinguished through the use of usernames and color.

Remaining Backlog

Functional Requirements

- As a party host
 - I would like to play songs my guests like
 - I would like the option to restrict songs to non-explicit versions
 - I would like the option to restrict songs to certain genres, artists, or Youtube channels (e.g. VEVO channels)
 - I would like the option to show the video feed from the current song if there is one (youtube videos)
 - I would like to be able to log in with a Facebook account, and have preferences accounts)for rooms saved
 - ⊖ ~~I would like to be able to restrict room access to specific users~~
 - ⊖ ~~I would like the option to enable a text chat for the room~~
 - I would like to be able to restrict chat to only admins (for announcements)
 - ⊖ ~~I would like to be able to specify a list of actions guests and admins have~~

~~permission to take~~

- I would like a constant flow of music
- I would like to set a suggested genre of music to be played (classical, pop, kids, dance, etc.)
- ~~I would like to be able to easily share a link to join a created room (QR code, URL)~~
- I would like the option to display a visualizer for the currently playing music
- I would like for songs to autoplay if the playlist is empty
- ~~I would like to specify users as admins~~
- I would like to notify people that I have started a party via Facebook, Twitter etc
- I would like to set a mood for the room
- ~~I would like to generate a QR code that can be displayed on a screen or printed to share with guests~~
- As an admin
 - I would like to be able to ban specific users from chatting
 - I would like to be able to ban specific guests from modifying the playlist or suggesting songs
 - I would like to remove undesired songs (Think... What's New Pussycat playing 21 times in a row, Rickrolls, etc.)
- As a professional DJ (all sub-bullets if time allows)
 - I would like to advertize my skills
 - I would like to share clips of my work
 - I would like to have aggregate reviews available from past performances

- I would like live reviews to come in to gauge audience reactions
- As a general user
 - ⊖ ~~I would like to create and login to an account on MelodyMunk.~~
 - ⊖ ~~I would like to customize my account and link my Spotify account to it.~~
 - I would like to have niche music available to play (if time allows) (Use Spotify, Soundcloud, Youtube)
 - ⊖ ~~I would like to have a mobile friendly website interface~~
 - ⊖ ~~I would like to be able to search for songs~~
 - ⊖ ~~I would like to be able to vote on songs~~
 - I would like to be able to write reviews on the party host or DJ
 - I would like to be able to report other party guests to the admins
- As a party guest
 - I would like to sign a guestbook to leave a note when I'm leaving the party (if time allows)
 - ⊖ ~~I would like to suggest songs~~
 - ⊖ ~~I would like to join a room quickly and easily (QR code or URL)~~
 - I would like my song to be played in a reasonable amount of time
 - I would like suggestions for songs that I may be interested in adding
 - I would like to be alerted when my song is going to be played next (If time allows)
- As a developer
 - I would like to account for host throttling to ensure users aren't blocked from Spotify

Non-functional Requirements

- ~~Connection to Spotify music streaming service~~
- Expand to connecting with Soundcloud or Vevo Youtube (if time allows)
- Run smoothly without delay from the website in accordance to limitations set by 4g cell service
- ~~Secure user profiles and user data~~
- Banning specific users from rooms can utilize IP bans or website profile bans
- Website must be able to handle at least 250 users at any given time
- We would like to have high quality music playing (minimum 128 kbps bitrate)
- A way to test whether or not the service is functioning properly