CS 30700

**MelodyMunk**

Team 24: Sprint 2 Retrospective

horizontal line



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

Table of Contents

[**What Went Well?**](#_qu59lhbkuh1q) **3**

[User Story #1](#_jolm8ubv7h4) 3

[User Story #2](#_1cei8hkhqsbz) 4

[User Story #3](#_y0qtdda65pua) 5

[User Story #4](#_cess9u1mcgmf) 6

[User Story #5](#_vwj107nobpdd) 7

[User Story #6](#_iggho7ocz254) 8

[User Story #9](#_69b3tmbyn00s) 9

[User Story #11](#_dv1a3fk9y9je) 11

[User Story #12](#_kw0or9xdx17w) 12

[User Story #13](#_a07caox7egp4) 13

[**What Went Wrong?**](#_rxjom57br2rj) **15**

[User Story #7](#_6dum74i4zj1k) 15

[User Story #8](#_ot47fcwxvlso) 16

[User Story #10](#_up8xfjs2ak5l) 17

[User Story #14](#_bgnnx2q2zvpu) 18

[User Story #15](#_wqmntjdtgblj) 19

[**How Should We Improve?**](#_mb8q6yib2uga) **20**

# 

# 

# 

# 

# 

# What Went Well?

## **User Story #1**

As a general user, I would like to upload a profile picture to my profile.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Estimated Time** | **Owner** |
| **1.** | Create AWS S3 Bucket for storing images | 1 Hour | Matt |
| **2.** | Setup permission policies on S3 bucket | 2 Hours | Matt |
| **3.** | Install AWS-PHP API Bundle on project | 2 Hours | Matt |
| **4.** | Add image URL to User entity and DB | 1 Hour | Matt |
| **5.** | Create upload controller to upload files and images to S3 bucket | 3 Hours | Matt |
| **6.** | Create UI for uploading image to profile | 1 Hour | Matt |

Completed: Users can upload profile pictures to their profile and those images are stored using AWS.

## **User Story #2**

As a general user, I would like to submit DJ reviews to hosts of a playlist.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Estimated Time** | **Owner** |
| **1.** | Create DJ Review Entity and DB Table | 2 Hours | Matt |
| **2.** | Create Review Controller to handle submitting a new DJ Review | 3 Hours | Matt |
| **3.** | Create DJ Review Form | 1 Hours | Matt |
| **4.** | Add DJ Reviews to the user profile UI | 3 Hours | Matt |
| **5.** | Create UI for submitting new DJ Reviews | 2 Hours | Matt |
| **6.** | Test to ensure submitting DJ Review is working properly | 1 Hour | Matt |

Complete: Users can submit reviews about the DJ and the DJ reviews can be viewed from a DJ’s profile.

## 

## 

## **User Story #3**

As a general user, I would like to be able to see who is in the room and report guests for to playlist admins.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Estimated Time** | **Owner** |
| **1.** | Create User Report entity and DB table | 2 Hours | Matt |
| **2.** | Create Report Controller in BE to handle submitting reports | 2 Hours | Matt |
| **3.** | Implement feature to allow users to see who is the room | 3 Hours | Matt |
| **4.** | Create UI for reporting a playlist guest | 2 Hours | Matt |
| **5.** | Create UI for playlist admins to see reported guests | 2 Hours | Matt |
| **6.** | Test to ensure flow of reporting user is working properly | 1 Hours | Matt |

Completed: Users can see all of the users that are currently in the room and report the guest that they want to from the UI.

## 

## 

## **User Story #4**

As a general user, I would like to be able to play songs from spotify in my browser

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Estimated Time** | **Owner** |
| **1.** | Enable browser to play audio | 2 hrs | Avnish |
| **2.** | Create User Interface to play songs | 3 hrs | Avnish |
| **3.** | Connect with Spotify API to play songs | 3 hrs | Avnish |
| **4.** | Unit Testing for playing songs in browser | 1 hr | Avnish |
| **5.** | Error handling for playing songs in browser | 2 hrs | Avnish |

Completed: Users can go onto our website and control audio playback from their browser. If spotify sends 400 and 500 response codes, those are handled correctly and don’t affect the user.

## 

## 

## **User Story #5**

As a general user, I would like to be able to pause, fast forward and raise the volume of the songs I am playing from spotify

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Estimated Time** | **Owner** |
| **1.** | Connect pausing with spotify api | 2 hrs | Avnish |
| **2.** | Connect fast forwarding with spotify api | 2 hrs | Avnish |
| **3.** | Connect raising volume with spotify api | 2 hrs | Avnish |
| **4.** | Error handling and unit testing for pausing,fast forward and raising volume | 2 hrs | Avnish |
| **5.** | UI for pausing, fast forwarding and raising volume | 3 hrs | Avnish |

Completed: Users can control the playback of the songs from the MelodyMunk website. Users can control the volume,fast forward through songs and skip to next song using the UI from the MelodyMunk website.

## 

## 

## **User Story #6**

As a general user, I would like to be able to manage my spotify account from rooms

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Estimated Time** | **Owner** |
| **1.** | Rewrite authentication code from backend javascript to php | 5 hrs | Avnish |
| **2.** | Make api calls from ajax | 1 hrs | Avnish |
| **3..** | Populate Song entity with calls from spotify api | 2 hrs | Avnish |
| **4..** | Make get song api call from php backend | 1 hrs | Avnish |

Completed: The spotify integration has know been moved to php so it can interact with other parts of the project.

## User Story #9

As a developer, I would like better performance and usability for my local testing environment (virtual machine).

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Estimated Time** | **Owner** |
| **1.** | Research optimizations for Vagrant | 3 hr | Brooks |
| **2.** | Test some of the possible improvements | 3 hr | Brooks |
| **3.** | Incorporate optimizations into Vagrantfile so that other team members can get them | 2 hr | Brooks |
| **4.** | Add some useful bash customizations | 2 hr | Brooks |
| **5.** | Test Vagrantfile with fresh virtual machine to ensure changes are working correctly | 1 hr | Brooks |

Complete: There are significant improvements to the VM performance and page load times. Previously, page load times were from 15-60 seconds. After optimizations, load times are usually less than 5 seconds.

There are also helpful aliases as alternatives to typing long and repetitive commands for starting and stopping the server. It is also easy to update these aliases through a single file in the project, and share the changes through Git with other team members.

## 

## 

## **User Story #11**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Estimated Time** | **Owner** |
| **1.** | Design room info page | 4 hours | Shrikar |
| **2.** | Design playlist page | 8 hours | Shrikar |
| **3.** | Design chat page | 5 hours | Elizabeth |
| **4.** | Create Notification Settings (Melody Munk news/updates) | 3 hours | Shrikar |

Completed: The UI for room, playlist,chat and settings have been completed. Users can access pages for room info and playlists. Users also have the ability to chat with one another. Notification banners are pushed to the website when new updates/ news is issued. Fully functional.

## User Story #12

As a host I would like UI for room specific actions and functions (cleanup of some UI related tasks from previous sprints).

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Estimated Time** | **Owner** |
| **1.** | Create UI for seeing QR code for a room | 3 hours | Brooks |
| **2.** | Create UI for printing a QR code for a room | 3 hours | Brooks |
| **3.** | UI for specifying list of guest and admin actions allowable (likely under room settings->advanced settings) | 3 hours | Tristan |
| **4.** | UI for promoting users to admins | 3 hours | Tristan |

Partially Complete: The QR display button and print page are done and functioning very well; they only need to be inserted into the other UI pages, which is a trivial task once those pages are ready. UI for promotion of users to admin level privileges has been fully implemented along with a specific settings UI for specifying list of allowable actions.

## **User Story #13**

As a user, I would like easy access to settings, and be able to set the mood (i.e. an audio visualizer) depending on the theme/genre of the music being played. The ability to access my profile information and edit information about myself as and when required.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Estimated Time** | **Owner** |
| **1.** | Create User Settings UI | 4 hours | Shrikar |
| **2.** | Create a music visualizer | 6 hours | Shrikar |
| **3.** | UI for Account Overview (i.e. Profile information:- Username, email, dob, country) | 3 hours | Shrikar |
| **4.** | UI for Edit Profile / Change Password | 2 hours | Shrikar |

Completed: UI for Settings, visualizer, account overview, and edit profile are completed. Users have the ability to access settings, make changes to their account info and edit passwords. They also have the ability to visualize the songs that they are playing via the visualizer.

**User Story #16**

As a user, I would like a text based chat for each room.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Estimated Time** | **Owner** |
| **1.** | Create database entities for chat entries | 2 hours | Brooks |
| **2.** | Create functions to entering and retrieving chat messages | 3 hours | Brooks |
| **3.** | Create controller for exchanging chat messages with front end asynchronously | 3 hours | Brooks |
| **4.** | Create basic UI structure for chat testing | 3 hours | Brooks |
| **5.** | Create frontend functionality for displaying messages in a room and exchanging with server | 3 hours | Brooks |

Complete: A fully functional chat is working with a test page. It should be very simple task to integrate this into other UI pages once they are ready. There should be some extra work done for more robust error handling, reconnecting to chat if connection is lost, etc., but all criteria and tasks were complete.

# 

# What Went Wrong?

## **User Story #7**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Estimated Time** | **Owner** |
| **1.** | Create database relations to track a room and users inside | 3 hours | Elizabeth |
| **2.** | Create helper functions to update room attributes and add/remove users | 3 hours | Elizabeth |
| **3.** | Create module to find room identified by URL | 5 hours | Elizabeth |
| **4.** | Create UI to join and create rooms | 3 hours | Elizabeth |

Partially Complete: The database, helper functions, and URL room identification are finished. The join UI is created, the create room is functioning but needs polishing.

## 

## 

## **User Story #8**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Estimated Time** | **Owner** |
| **1.** | Interface module with UI to allow host (or admin) to specify a user as an admin | 3 hours | Tristan |
| **2.** | Interface module with database to save and retrieve preferences across sessions | 3 hours | Tristan |
| **3.** | Unit test to verify promotion to admin | 1 hour | Tristan |

Partially Complete: UI has proper implementation for specification of user roles. Saving and retrieving preferences is not complete nor do unit tests exist to verify user roles were correctly updated. Backend connection on these needs further work.

## 

## 

## **User Story #10**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Estimated Time** | **Owner** |
| **1.** | Add database relationships to track a whitelist and blacklist of users per room | 4 hours | Brooks |
| **2.** | Create functions to add and remove users from white/blacklist | 3 hours | Brooks |
| **3.** | Allow the room to be switched between blacklist mode, whitelist mode, or neither | 2 hours | Tristan |
| **4.** | When attempting to add a user into a room, verify the user is allowed based on the current restriction mode | 1 hour | Tristan |
| **5.** | Unit testing to ensure restrictions hold | 1 hour | Tristan |

Partially Complete: Database relations exist for proper tracking, room modes exist for whatever mode the host/admins decide is needed, and helper functions exist for all tasks up to 3. There is no unit testing and no user permissible to join verification.

## **User Story #14**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Estimated Time** | **Owner** |
| **1.** | Make UI for mobile friendly room page with slide out tabs | 2 | Elizabeth |
| **2.** | Make UI for mobile friendly chat tab in the room page | 3 | Elizabeth |
| **3.** | Make UI for mobile friendly playlist tab in the room page | 2 | Elizabeth |
| **4.** | Make UI for mobile friendly room info tab in the room page | 2 | Elizabeth |

Partially Complete: Again the website looks good on mobile but could use more polishing to be a better user experience.

## 

## 

## **User Story #15**

As an admin, I would like a playlist feature that continues playing songs even when the playlist empties.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Estimated Time** | **Owner** |
| **1.** | Create playlist data structure | 4 hours | Tristan |
| **2.** | Check if playlist has finished all songs in the playlist | 2 hours | Tristan |
| **3.** | If playlist is empty start playing radio based off last song to be played. | 3 hours | Tristan |
| **4.** | Privileged users given the ability to remove song from the playlist queue | 4 hours | Tristan |

Partially complete: Playlist data structure exists along with ability to check if a playlist has finished music in queue. Playing radio when queue is empty is not complete nor is privileged users having the ability to remove a song from the queue.

# 

# 

# How Should We Improve?

* Prioritize stories that other members depend on for their work
  + Issues wherein work that others needed would be delayed at times making it difficult for others to get work done. Committing more often can help to mitigate this issue, especially if we emphasize getting some commits done within meetings.
* Commit more time to creating proper unit testing
  + Unit testing remained a secondary thought in Sprint 2 even after addressing it after Sprint 1. It needs to be heavily prioritized for our final sprint.
* Be more strict with weekday meeting attendance
  + These meetings are usually important and are when members are trying to resolve issues that are holding them back. Sometimes those issues could be solved by a member who had skipped out on that meeting.
* Have ways to demonstrate error handling
  + When we have error handling as a task, we need a reliable way to trigger an error to demonstrate that it works
* Accountability
  + We should have higher expectations of how much work each member should have done by the end of each week. It seems like a lot of work gets pushed to the very end of the sprint.
* Code comments
  + There has been next to no system for proper commenting of the code so far beyond annotations. As the project has progressed this has created confusion when connecting new items and wasted time attempting to figure out what a given file or functions within the file are for.
* Better Communication
  + Better response time in GroupMe and Slack. Make sure everyone knows what tasks they have to do and by when.
  + Continual reminders of group meetings and clarifying who will not be able to attend for whatever reason at minimum the night before so that meetings can run smoothly in preparation of the absence.