Milestone 2 - Planning Phase (Team 18)

Team Member Name	PID	UCSD Email ID
Anthony Wong	A12773207	anw063@ucsd.edu
Hoi Wa Ng	A14259390	hwn001@ucsd.edu
Nang Chen	A14205066	nac018@ucsd.edu
Shenghao Jiang	A91063762	shj003@ucsd.edu
Ghassan Haddad	A14169330	g1haddad@ucsd.edu

Risk Analysis

1.
Risk: Team did not do much work outside of meetings together a few times during the week and the weekends.
Description: During the last milestone we weren't very productive on our own outside of the times we met.
Severity: High
Resolution: Work more on our own after our meetings as well or divide the group into small teams to work on different parts. Also consistently monitor each other through messenger as well as set stricter deadlines and schedules for specific parts that need to be done on our own time.
Status: Resolved

2.
Risk: Team did not use enough Github and Zenhub through all the projects
Description: During the last milestone, we did not use Github and Zenhub consistently.
Severity: Medium
Resolution: We need to learn how to use Github effectively and consistently. We need to create our own developer branch and merge our progress to the main branch. We need to comment and review each other's merge request regularly. Once we create a merge request, we need to immediately attach it to a Zenhub issue. Also, we did not have a burndown chart showing we developed our step by step. So, we need to plan ahead and adhere to our Zenhub schedule.
Status: In Progress

3.

Risk: While working on our own we have different rates of progress and don't know how each of us are progressing.
Description: During the last time, we weren't working on the project in our own time, so our rates of progress were in consistency. Thus we haven't had this problem last time, but since now we decided to work on project on our own time, this risk would inevitably impede our overall development.
Severity: Medium
Resolution: Have standup meetings more frequently and let each other know when any progress is made through messenger or other means. We need to pull and push our code more frequently to ensure our one stay on the same page.
Status: In Progress

4.
Risk: Multiple bugs when we code that slow down the progress of our project and team still need to resolve issues related to milestone 1.
Description: When we are implementing some features, we might find that there are some bugs that refrain us from moving on to the next features and waste a lot of our time. Even though we are on milestone 2 right now, our team still need to work on features that were supposed to be delivered in milestone 1, so that this could cause us a decent amount of time. It might refrain us from moving on to the new features in Milestone 2 if we cannot fix these issues as soon as possible.
Severity: Medium
Resolution: Perform more pair programming and a teammate can sit by and notice the problem and debug faster than independent programming to facilitate the pace of the project. Also we can allocate more time individually or during the team meetings to resolve the remaining issues.
Status: In progress

5.
Risk: Insufficient testing for our codes
Description: During the delivery phase, we usually just finish coding and run the application right after that. We have not really gotten into the edge cases testing that might affect the behaviors of some of the features.
Severity: Medium
Resolution: Take advantages of unit testing, Espresso, logging to test our codes before we move on to avoid any unnecessary mistakes. We need to write the tests before we implementing the features. We need to test new features and the project as a whole regularly.
Status: In progress

<u>User stories</u>

1. **High Priority - Iteration 1**
Title: Toggling Vibe Mode when it is on or off
Narrative:

As a user I want to turn Vibe mode on or off in the app
So that I can automatically listen to music based on what others around me have been listening to.

Scenario 1: Enabling Vibe mode and playing based on location, date, and friend.
Given the app is open
    And Vibe mode is not enabled
    And “Dreamatorium” was played by user within 300 meters of present location
    And “Dreamatorium” was not played last week
    And “Dreamatorium” was not played by a friend
    And “123 Go” was played in the last week
    And “123 Go” was not played by a friend
    And “Back East” was played by a friend
When the user turns on Vibe mode by pressing a button saying “Vibe On”
Then the player should be on Vibe mode
    And playlist containing “Dreamatorium”, “123 Go”, and “Back East” is created
    And “Dreamatorium” starts playing
    And “123 Go” is played after “Dreamatorium”
    And “Back East” is played after “123 Go”
    And button now says “Vibe Off”

Scenario 2: Disabling Vibe mode
Given the app is open
    And Vibe mode is enabled
    And “Dreamatorium” is currently playing
    And playlist contains “Dreamatorium”, “123 Go”, and “Back East”
When the user turns off Vibe mode by pressing button that says “Vibe Off”
Then the player should be off Vibe mode
    And “Dreamatorium” stops playing
    And Vibe playlist is now empty
    And button now says “Vibe On”

Scenario 3: Disliking a song when Vibe Mode is enabled it skips to next song in playlist
Given the app is open
    And Vibe mode is enabled
    And “Dreamatorium” is currently playing
    And Vibe playlist contains “Dreamatorium”, “123 Go”, and “Back East”
When the user skips “Dreamatorium” by pressing the “>|” button
Then “Dreamatorium” stops playing
    And “123 Go” starts playing
    And Vibe playlist now contains “123 Go” and “Back East”
    And “Back East” is played after “123 Go”

Scenario 4: Vibe mode enabled but no tracks were played
Given the app is open
    And Vibe mode is disabled
    And no song was played by user within 300 meters of present location
    And no song was not played last week

And no song was not played by a friend
When the user turns on Vibe mode by pressing “Vibe On” button
Then no song is played
And Vibe playlist is empty
And button stays the same

Tasks:
1. Create new Vibe class - 2
2. Redo flashback button into a vibe button - 1
3. Create new priority queue based on the new three criterias - 2

2. **High Priority - Iteration 1**
Title: Download tracks from remote sources
Narrative:
As a user I want to download tracks that people around me is/has been listening to when the Vibe mode is on.
So that I can also listen to music that might have not been stored locally in my phone after downloading it.

Scenario 1: User is playing music that are already in library in Vibe Mode
Given the phone has internet access
And the Vibe Mode tracklist (soon to be generated) contains all songs that I have locally
When I click on the Vibe Mode, it starts playing tracklist in order without a need to download.
Then I don’t re-download the same songs again.

Scenario 2: User is playing music that may not be in library in Vibe Mode
Given the phone has internet access
And the Vibe Mode track list contains songs that I do not have locally
When I click Vibe Mode, it will start downloading the song that is not in library from the same remote source as the user nearby me have downloaded from.
Then I can listen to track that i am interested but didn’t have it locally before.

Scenario 3: User is playing music in Vibe Mode with no internet access
Given the phone has no internet access
And the Vibe Mode track list contains songs that I do not have locally
When I click Vibe Mode, it will start playing songs and skipped all songs that needed to be downloaded.
Then I can still listen to songs with no interruption.

Task:
1. Fetch download resources - 2
2. Download songs from fetched link

3. **Medium Priority - Iteration 2**
Title: The complete track list should be viewable when playing an album or in Vibe Mode
Narrative: As a user I want to view the complete track list when playing an album or in Vibe Mode

So that I will be able to know what are the upcoming tracks

Scenario 1: User is playing music in Vibe Mode
Given a song is playing
And Vibe Mode is on
And “Dreamatorium” , “123 Go”, “Back East” were played by users within 300m of my current location (a)
And “Dreamatorium” , “123 Go” were played in last week (b)
And “Dreamatorium” was listened by my friends (c)
When the user clicks on the list button
Then the complete track list would be shown in playing order with the priorities (from (a)-(c)) and user would know what are the upcoming tracks. The order should be “Dreamatorium” -> “123 Go” -> “Back East” according to their total scores in each priorities.

Scenario 2: User is playing music in the album “I Will Not Be Afraid”
Given “America Religious” is playing
And Vibe Mode is off
When the user clicks on the album “I Will Not Be Afraid”
Then all of the songs following “I Will Not Be Afraid” in that album would be shown in playing order and user would know what are the upcoming tracks in that album.

Tasks:
1. Create a view for playlist to show all the upcoming tracks in Vibe Mode
2. Create a view for playlist to show all the upcoming tracks in album
3. Show the currently playing song in the bottom
4. Put a playing button near the currently playing song

4. **Low Priority - Iteration 2**
Title: Sort downloaded tracks by title, album, artist, or favorite status
Narrative:
As a user I want to sort the tracks by track title from A-Z, by album name, by artist name, and by favorite tracks.
So that I can view the tracks from A-Z and select one to play, play tracks in the same album, play tracks from the same artist, and play my favorite tracks.

Scenario 1:
Given: the current downloaded song list is empty
When: the user presses sort by title, album, artist, or favorite status
Then: the current downloaded song list should remain empty

Scenario 2:
Given: the current downloaded song list have three songs, “apple”,”banana”, and “cherry”
When: the user presses sort by title
Then: “apple” should appear to be the first song

“Banana” should appear to be the second song
“Cherry” should appear to be the third song

Scenario 3:
Given: the current downloaded song list have three songs
“apple”,”banana”, and “cat”
“Apple ”and “banana” belongs to the “fruit” album
“Cat” belongs to the “animal” album
When: the user presses sort by album
Then: “Cat” should appear as the first song
“Apple” and “banana” should appear to the second and third song

Scenario 4:
Given: the current downloaded song list have two songs
“Apple” by Rebecca and “Banana” by Allen
When: the user presses sort by artist
Then: “banana” should appear to be the first song
“Apple” should appear to be the second song

Scenario 5:
Given: the current downloaded song list have three songs
“Apple” with status liked,
“Banana” with status dislikes,
“Cherry” with status neutral
When: the user presses sort by status
Then: “Apple” should appear to be the first song
“Cheery” should appear to be the second song
“Banana” should appear to be the third song

Tasks:
1. Create a priority queue to store the song objects.
2. Implement the comparator class to support sort by title, album, artist, or favorite status
3. Create a Sort class storing the priority queue and each sort methods
4. Create four buttons in the song list activity with name sort by title, album, artist,and status and each calls the corresponding method in the Sort class.

5. **Low Priority - Iteration 2**

Title: Displaying the name of the user who lastly played the song which is currently playing.
Narrative:
As a user, I want to see the name of the person who played the song most recently
So that I can determine if the song was played by me or by a friend in the last time.

Scenario 1: the song was lastly played by me.
Given: the app is open
And the player is in Vibe mode or normal mode

And song Z is playing
And song Z was lastly played by me.
When I click the details button
Then the player should display a popup window showing a message that song Z was lastly played by "you" and "you" is displayed in italics.

Scenario 2: the song was lastly played by a friend or an anonymous user.
Given: the app is open
And the player is in Vibe mode or normal mode
And song Z is playing
And song Z was lastly played by Bill or an anonymous user.
When the user click the details button
Then the player should display a popup window showing a message that song Z was lastly played by Bill if song Z was lastly played by Bill or the proxy name if song Z was lastly played by an anonymous user.

Tasks:
1- add the lastListener field to the song class, and the required setters and getters to access this field.
2- retrieve the relevant information from each track to get the name of the user who was the last one to play the track.
3- modify the popup window activity so it shows the value of the lastListener field in additions to the details it was supposed to show in milestone 1.

<u>**Velocity:**</u>

0.65
We're calculating the velocity for each iteration based on the following formula
Velocity = days of work / days required to get work done = 9*5 / 5 *14 = 0.65
Where 5 is the number of people in our team and 9 is the number of days to get the work done in an iteration and 14 is the total work days available.

<u>**Planning Poker**</u>

User story 1: Toggling Vibe Mode when it is on or off

User story 2: Download tracks from remote sources
User story 3: The complete track list should be viewable when playing an album or in Vibe Mode
User story 4: Sort downloaded tracks by title, album, artist, or favorite status
User story 5: Displaying the name of the user who lastly played the song which is currently playing.

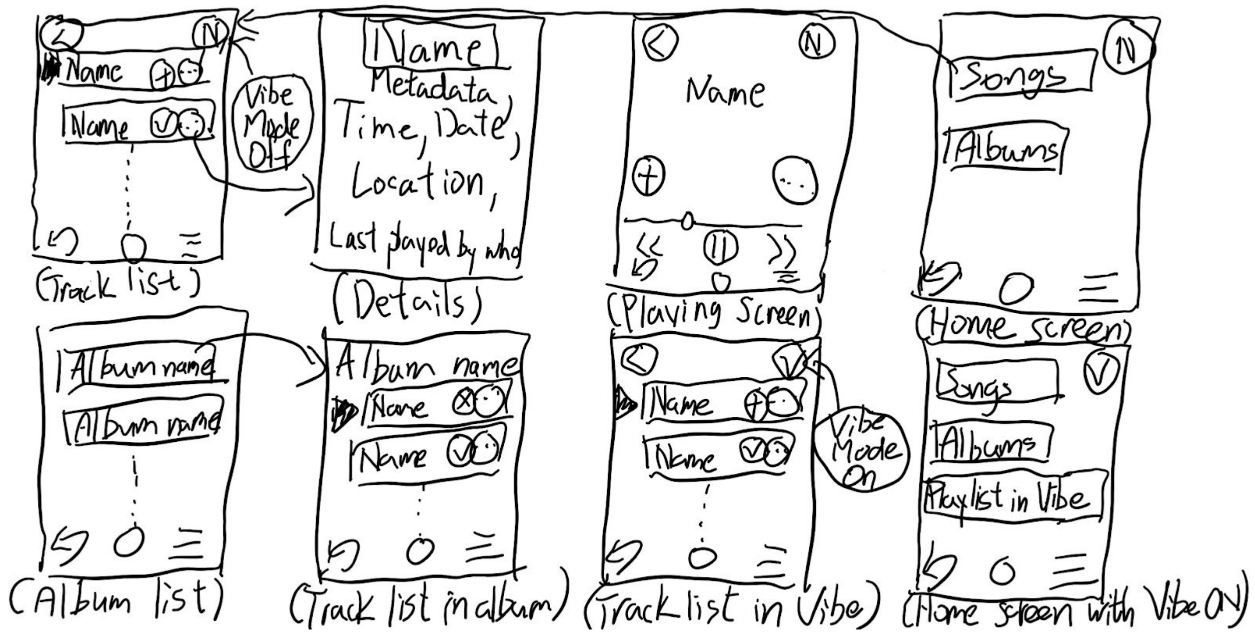
S#	**Name**	**Hand**	**False Assumptions Uncovered**
1	Vibe mode	2 5 5 5 4	Assumed that it is an extension of Flashback, but it is a completely new system based on friends around user. We think this will take 5 days because it is the bulk of the work for our iteration and the main functionality the client has asked for from review of the first delivery. It will take time to learn how to access friend locations and things about friends.
1	Vibe mode	5 5 5 5 5	(None)
2	Download Songs	4 5 4 5 4	Assume the user needs to manually download songs
2	Download Songs	4 4 4 4 4	(None)
3	Display tracklist in album/Vibe Mode	2 2 3 3 3	Assumed that we have already implemented the tracklist functionality for album in last milestone. However, it's not implemented yet to show all the upcoming tracks and also the current playing song.This takes us 3 days in total since we need to implement tracklists for both album and Vibe Mode. We use 2 days to figure out how to create the playlist with the priorities in the Vibe Mode.
3	Display tracklist in album/Vibe Mode	3 3 3 3 3	(None)
4	Sort track	5 2 2 3 8	Assumed that we need to create methods for each criteria separately. However, we can implement four comparator to support sort by title, artist, album, and favorite status.Someone thinks we can use the comparator from flashback mode. However, this assumption is not true because we have other criteria to sort.
4	Sort track	4 4 4 4 4	(None)
5	Display details	2 3 2 2 3	Assumed that we need to recreate a popup window to show details but we can use the one we already have.
5	Display details	2 2 2 2 2	(None)

Zenhub

- *Zenhub Board:* Insert link to ZenHub board
https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-18/boards?repos=119287955
- *Burndown Chart:* Insert link to burndown chart
https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-18/reports?report=burndown&milestoneId=3141176&showPRs=false
- *User Stories:* Insert links to user stories, make sure they are also linked to their respective tasks
https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-18/boards?epics:settings=epicsOnly&repos=119287955

- *Scenario-Based System Tests:* Insert links to Scenario-Based System Tests, make sure they are also linked to their respective user stories and iterations
https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-18/boards?labels=good%20first%20issue&repos=119287955

User Interface Progressions/Screens (Wireframes)



Name
Name
Vibe Mode Off
(Track list)
Name
Metadata,
Time, Date,
Location,
Last played by who
(Details)
Name
(Playing Screen)
Songs
Albums
(Home screen)
Album name
Album name
(Album list)
Album name
Name
Name
(Track list in album)
Name
Name
Vibe Mode On
(Track list in Vibe)
Songs
Albums
Playlist in Vibe
(Home screen with Vibe ON)