

Team #31

Team Member Name	PID	UCSD Email ID
Gaurav Parmar	A13947003	gparmar@ucsd.edu
Daniel Gordee	A13954957	dgordee@ucsd.edu
Aaron Apte	A13420131	aapte@ucsd.edu
Veronica Lin	A13671255	velin@ucsd.edu
Alberto Montilla Ochoa	A13573544	amontill@ucsd.edu
Yue Zhu	A92020065	yuz345@ucsd.edu

Milestone 2 - Planning Phase

Risk Analysis

Risk: Unstable base app

Description: The FBP we had from the first milestone does not meet all the requirement and contains bugs. We cannot implement new features on top of a buggy app.

Severity: High

Resolution: All hands on deck to fix it asap

Status: Solved

Risk: Lack of skill to establish central server

Description: Vibe mode requires us to be able to store and retrieve information from different users. No one on the team is familiar with the technique to do that.

Severity: High

Resolution: Do research and learn how to implement a server.

Lab will teach us this on Monday!

Status: Solved

Risk: How do users find their songs?

Description: The description is not clear in terms of how users are going to get their songs. Do we provide a full list for them to choose from? Do we provide search function?

Severity: High

Resolution: Ask the marketing team about this and get clarification. Piazza response: @522

Provided URL with songs, need not support login and password.

Status: Solved

Risk: How do we get friend list for a user?

Description: How do we want user to make friends? Do we want a brand new log-in system and platform where people can add each other as friends, or do we want to reply on third-party APIs?

Severity: High

Resolution: Ask the marketing team about this and get clarification. Piazza response: @453

Use Google+ friends the 'android way', aka use API provided by Google+

Status: Solved

Risk: What is "around you"?

Description: User would like to listen to the tracks played by those around him recently. How do we define the distance around the user? Do we use a gradient? Do we use threshold?

Severity: Medium

Resolution: Ask the marketing team about this and get clarification. Piazza response @453

The distance around the user remains the same as for milestone 1 (1000ft)

Status: Solved

Risk: How much control does user have over the Vibe playlist?

Description: Can user modify the list in anyway (add, remove, skip, like, dislike songs)?

Severity: Medium

Resolution: Ask the marketing team about this and get clarification.

If the user dislikes a song, the song won't be played in vibe mode. If we dislike it in vibe mode, it will skip to the next song. User can pause, resume and skip just like in normal mode.

Status: Solved.

Risk: Whose name takes priority?

Description: When a track is played by more than one our of user's friends, user himself, and a stranger, whose name should be displayed?

Severity: Medium

Resolution: Ask the marketing team about this and get clarification.

When a friend played the song, the person among your friends who most recently played it will be tracked.

When a stranger closer by plays the song, his proxy name displays.

Status: Solved.

Risk: How are we tracking who played this song?

Description: Do we keep a separate record of who played this song and where and when for each individual user based on locality and friend circle, or do we keep a unified record of who played this song before, and pick what to display based on who the user is?

Severity: Medium

Resolution: We need to keep track of all the songs played near the user's location for each user using Firebase.

Status: Solved.

Risk: When to remember a song?

Description: If a user has never played a song before, and hears it in Vibe mode, does the play remember that song?

Severity: Medium

Resolution: Ask the marketing team about this and get clarification.

Write-up update. Only downloaded songs are remembered.

Status: Solved

Risk: Ambiguity about storing an album

Description: When a user only played one song from one album, and chooses to browse and play tracks by album, will the entire album be played or only just that song?

Severity: Medium

Resolution: Ask the marketing team about this and get clarification.

Write-up updated. The downloaded song will be displayed, which means only downloaded song will be displayed.

Status: Solved

Risk: Weekdays Vs. weekends

Description: How exactly do we decide the similarity of time the song is played?

Severity: Medium

Resolution: Ask the marketing team about this and get clarification.

Same as MS1, we are just doing days.

Status: Solved.

Risk: If you dislike a song, does it still show up in Vibe mode?

Description: When a user dislikes a song in normal mode, does the song still play in Vibe mode if it is played by a nearby friend.

Severity: Low

Resolution: Ask the marketing team about this and get clarification.

No.

Status: Solved.

Risk: What if there is no access to Google+ account?

Description: Without a Google+ account, user cannot find out what is trending nearby. What will happen in Vibe Mode then?

Severity: Medium

Resolution: Ask the marketing team about this and get clarification.

App should function just like a normal, media player.

Status: Solved.

Risk: Where is music being streamed from? Spotify? Pandora?

Description: Where will the user get all the stream music from in Vibe Mode. There are many streaming mediums. Should our app support all streaming sites?

Severity: High

Resolution: Ask the marketing team about this and get clarification.

Music can be streamed from noisetrade.com.

Status: Solved

Risk: What songs do we include in normal mode?

Description: Do we include all songs that have been played? Only local songs? Songs that have been favorited?

Severity: High

Resolution: Ask the marketing team about this and get clarification.

Locally downloaded songs via normal mode or vibe mode.

Status: Solved

Velocity

Milestone 1 Iteration 1 velocity:

$$31/(2*6*5)$$

$$= 0.5167$$

Milestone 1 Iteration 2 velocity:

$$45.5/(2*6*5)$$

$$= 0.7583$$

We calculated the velocity based on the time it took to complete iteration 2 of milestone 1. We did not use the iteration 1 of that milestone, because all of the team members were initially new to Android Studios and GitHub collaborative workflow; hence, the low velocity for that iteration. However, in iteration 2, our velocity was 0.7583 because we did not have such issues anymore.

Therefore we decided to use 0.75 as our initial velocity for the second milestone.

Planning Poker



Story #	Name	Hand (hours)	False Assumptions Uncovered
1	Story 1: Connect Vibe Mode with friend lists from Google+	5, 6, 3, 5, 5, 2.5	<ul style="list-style-type: none"> Need to create new user login system
1	Story 1: Connect Vibe Mode with friend lists from Google+	3, 2.5, 2, 3, 2.5, 2	<ul style="list-style-type: none"> Ambiguities about Google+ API
1	Story 1: Connect Vibe Mode with friend lists from Google+	2, 2, 2, 2, 2, 2	(None)
2	Story 2: Vibe mode functionality	11, 14, 9, 13, 10, 12	<ul style="list-style-type: none"> Ambiguity on how friend's data is stored Unsure how to implement central database (prediction made prior to Firebase lab)
2	Story 2: Vibe mode functionality	11, 12, 10, 11, 10, 10	<ul style="list-style-type: none"> Ambiguity on whether a separate record of who played what songs at a particular location is needed <ul style="list-style-type: none"> Ambiguity on where music is being streamed from <ul style="list-style-type: none"> Ambiguity on how to stream songs
2	Story 2: Vibe mode functionality	10, 10, 10, 10, 10, 10	<ul style="list-style-type: none"> None
3	Story 3: Liked and dislike a song	1, 2, 4, 4, 2, 3	<ul style="list-style-type: none"> Ambiguity on whether disliked songs need to show up in Vibe Mode Ambiguity on what changed in the toggle button from MS1 version (need to store status in SharedPreferences file)
3	Story 3: Liked and dislike a son	4, 4, 4, 4, 4, 4	(None)
4	Story 4: Download songs from Vibe Mode	2.5, 3, 2.5, 5, 4, 4	<ul style="list-style-type: none"> Ambiguity on how songs can be downloaded from a streaming website Ambiguity on whether albums need to be downloadable? Ambiguity on whether streamed songs are downloaded permanently (kept in storage)
4	Story 4: Download songs from Vibe Mode	2.5, 2.5, 2.5, 3, 3, 2	<ul style="list-style-type: none"> Ambiguity on what songs and albums are included in Normal Mode Ambiguity on difficulty of downloading songs through URL link and saving to local storage
4	Story 4: Download songs from Vibe Mode	2.5, 2.5, 2.5, 2.5, 2.5, 2.5	(None)
5	Story 5: Browse the songs differently	6.5, 7, 5, 6, 5, 6	<ul style="list-style-type: none"> Ambiguity on the sorting behavior for artist, title, author, favorite status Ambiguity on whether to fix MS1 version of sorting tracks and albums Certain members have ambiguity on difficulty/how to create new tab for new categories

5	Story 5: Browse the songs differently	5.5, 5.5, 5.5, 5.5, 5.5, 5.5	(None)
6	Story 6: See where and who else has played this track	6, 8, 6, 6, 5, 6	<ul style="list-style-type: none"> Ambiguity on how to access and save this kind of data Ambiguity on everyone's skill level in using databases. Song completion may be different than MS1 considered "completed"
6	Story 6: See where and who else has played this track	5, 5, 5, 5, 5, 5	(None)
7	Story 7: See the current playing list	2, 2, 3, 1.5, 2, 3	<ul style="list-style-type: none"> Ambiguity on whether a queue implementation already exists (it does) Ambiguity on difficulty of showing a queue of songs
7	Story 7: See the current playing list	2, 2, 1.5, 1.5, 2, 2	<ul style="list-style-type: none"> Ambiguity on difference in implementation when showing tracklist for album versus tracklist in Vibe Mode (relatively similar) Ambiguity on whether a new screen needs to be created
7	Story 7: See the current playing list	1.5, 1.5, 1.5, 1.5, 1.5, 1.5	(None)

Zenhub

- *Zenhub Board*: <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/boards?repos=119226356>
- *Burndown Chart*: <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/reports?report=burndown&milestoneId=3146170&showPRs=false>
- *User Stories*:
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/159>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/160>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/162>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/161>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/163>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/164>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/165>

- *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-31/issues/190>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/191>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/192>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/193>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/194>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/195>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/196>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/197>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/198>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/199>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/200>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/201>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/202>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/203>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/204>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/205>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/206>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/207>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/208>
 - <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/issues/209>

User Interface Progressions/Screens (Wireframes)

