

Team #28

Team Member Name	PID	UCSD Email
Abinesh Sarvepalli	A13971976	asarvepa@ucsd.edu
Aditya Mullick	A13675049	amullick@ucsd.edu
Ankur Gupta	A13877512	ang121@ucsd.edu
Youyuan Lu	A92022633	yol093@ucsd.edu
Chengyou Ju	A9209645	c1ju@ucsd.edu
Shaoze Wang	A92103590	shw154@ucsd.edu

Milestone 2 - Planning Phase

Risk Analysis

1. Risk: Having no idea how to download tracks from remote.
 - a. Description: All team members have no experience of server-based programming and therefore don't know how to stream a track in Android.
 - b. Severity: High.
 - c. Resolution: Go asking the TAs in the lab, plus there might be a coming lab assignment that covers how to do that. Also look up APIs that could help support this feature.
 - d. Status: Solved.

2. Risk: Refactoring risks
 - a. Description: The Vibe Mode is different from Flashback Mode and if we change the code for Flashback mode, then functionalities in other classes might be affected.
 - b. Severity: Medium.
 - c. Resolution: Do the refactoring thoroughly before any actual development.
 - d. Status: Solved.

3. Risk: Testing difficulties.
 - a. Description: The functionality of Vibe Mode depends on other friends so in order to test it, we have to make mock users which makes testing harder.
 - b. Severity: Medium.
 - c. Resolution: By applying Android location and time mocking mechanisms, team members will be able to obtain test data at ease.
 - d. Status: Solved.

4. Risk: Members live both on-campus and off-campus, making in-person coordination difficult.
 - a. Description: On-campus team members have limited access to transportation, forcing most meetings to have to take place on campus. This also limits the ability to conduct stand-ups.
 - b. Severity: Medium.
 - c. Resolution: Conduct stand-ups after class on Wednesdays and conduct Skype meetings for general questions. Integrative development (putting the app together) will also take place on weekends—namely Saturday afternoons.
 - d. Status: Solved.
5. Risk: Use of SRP and DRY
 - a. Description: There are more functionalities than ms1 and it will be harder to separate different functionalities into different classes.
 - b. Severity: Medium.
 - c. Resolution: Refactor SongTableActivity to be MediaViewActivity, and have different views for all 4 sorting characteristics (Title, Album, Favorite, Artist).
 - d. Status: Solved.
6. Risk: Demo difficulties.
 - a. Description: We will require other users to demo our app.
 - b. Severity: Low.
 - c. Resolution: Create mock Google Accounts on two different Android devices and friend the accounts together. That way, when running the applications separately, we are able to mock two different users.
 - d. Status: Solved.

Velocity: Our previous velocity was approximately 0.2. This is reasonable since the velocity for an average software engineer working full time is 0.7, but due to being college students and having other responsibilities, we cannot have as many person-hours. With 6 people working on the project, 0.2 velocity should be able to finish the project, just like in milestone 1.

Planning Poker

Story #	Story Title	Hands	Assumptions
1	Applications Stores User Credentials Upon First Launch	5 1 1 2 3 3	<ul style="list-style-type: none"> Not familiar with Google Sign-In API Implement User class and Database for this User Story
1	Applications Stores User Credentials Upon First Launch	2 2 2 2 2 2	(None)
2	Curating Basic Track List for all Local Tracks	2 2 2 3 4 2	<ul style="list-style-type: none"> We put all our old res songs into the new download folder for simplicity
2	Curating Basic Track List for all Local Tracks	2 2 2 2 2 2	(None)
3	Viewing Tracks by Artist	1 2 1 1 1 2	<ul style="list-style-type: none"> Refactoring code within this activity Similar to AlbumView from milestone #1
3	Viewing Tracks by Artist	2 2 2 2 2 2	(None)

4	Viewing Tracks by Favorite	1 1 1 1 1 1	<ul style="list-style-type: none"> • Similar to AlbumView from milestone #1 • Follow refactoring from the previous user story
4	Viewing Tracks by Favorite	1 1 1 1 1 1	(None)
5	Switching Between Modes	0 1 1 1 1 1	<ul style="list-style-type: none"> • Implementing the Toggle Switch (already implemented from previous milestone)
5	Switching Between Modes	1 1 1 1 1 1	(None)
6	Curating Playlists for Vibe Mode	4 4 4 3 3 3	<ul style="list-style-type: none"> • Refactor our flashback class into a vibe mode class, except use different priorities • Extracting friends metadata
6	Curating Playlists for Vibe Mode	4 4 4 4 4 4	(None)
7	Playing Music in Vibe Mode	2 2 2 2 2 3	<ul style="list-style-type: none"> • Similar to playing music from flashback mode • Will play music from noisetrade.com • Not familiar with noisetrade
7	Playing Music in Vibe Mode	2 2 2 2 2 2	(None)
8	Download Tracks from Remote Sources when in Vibe Mode	3 4 4 3 4 4	<ul style="list-style-type: none"> • Not familiar with DownloadManager API and storing URLs

8	Download Tracks from Remote Sources when in Vibe Mode	4 4 4 4 4 4	(None)
9	Generating Friends Report for Songs in Vibe Mode	1 3 2 3 1 2	<ul style="list-style-type: none"> • Cross checking two APIs (Google Friends and Firebase) • Familiarity with using APIs by this User Story should allow for higher efficiency
9	Generating Friends Report for Songs in Vibe Mode	2 2 2 2 2 2	(None)



Zenhub

Zenhub Board

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/boards?repos=119616844>

Burndown Chart

Iteration 1. <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/reports?report=burndown&milestoneId=3146013&showPRs=false>

Iteration 2. <https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/reports?report=burndown&milestoneId=3146069&showPRs=false>

User Stories

US #1. Application Stores User Credentials Upon First Launch

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/100>

US #2. Curating Basic Track List for All Local Tracks

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/101>

US #3. Viewing Tracks by Artist

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/102>

US #4. View Tracks by Favorite

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/103>

US #5. Switching Between Modes

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/104>

US #6. Curating Playlist for Vibe Mode

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/105>

US #7. Playing Music in Vibe Mode

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/106>

US #8. Download Tracks from Remote Sources when in Vibe Mode

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/107>

US #9. Generating Friends Report for Songs in Vibe Mode

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/108>

Scenario-Based System Tests

US #1. Application Stores User Credentials Upon First Launch

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/133>

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/134>

US #2. Curating Basic Track List for All Local Tracks

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/135>

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/136>

US #3. Viewing Tracks by Artist

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/137>

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/138>

US #4. View Tracks by Favorite

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/139>

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/140>

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/141>

US #5. Switching Between Modes

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/142>

US #6. Curating Playlist for Vibe Mode

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/143>

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/144>

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/145>

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/146>

US #7. Playing Music in Vibe Mode

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/147>

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/148>

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/149>

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/150>

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/151>

US #8. Download Tracks from Remote Sources when in Vibe Mode

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/152>

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/153>

US #9. Generating Friends Report for Songs in Vibe Mode

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/154>

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/155>

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-28/issues/156>

User Interface Progressions/Screens (Wireframes)

