

Team #31

Team Member Name	PID	UCSD Email ID
Gaurav Parmar	A13947003	gparmar@ucsd.edu
Aaron Apte	A13420131	aapte@ucsd.edu
Veronica Lin	A13671255	velin@ucsd.edu
Daniel Gordee	A13954957	dgordee@ucsd.edu
Alberto Montilla Ochoa	A13573544	amontill@ucsd.edu
Yue Zhu	A92020065	yuz435@ucsd.edu

Milestone 1 – Planning Phase

Risk Analysis

Risk #1: Not enough meeting Time

Description: Due to schedule conflicts, we don't have slots during the day where we are all available.

Severity: High

Resolution: We decided to meet at odd hours, find whatever time we can.

Status: Resolved

Risk #2: Lack of a proper communication channel

Description: One of the group member does not have a Facebook account, which we use as our main channel of communication.

Severity: High

Resolution: The member made a Facebook account and joined the conversation

Status: Resolved

Risk #3: Poor Productivity/ Project Management

Description: Given that deadline is not in the immediate future, the sense of urgency to work on the project diminishes.

Severity: Low

Resolution: Plan ahead. We can allocate specific time to work on the project no matter when the deadline is. If we are falling behind, we can add more hours.

Status: Resolved.

Risk #4: Low Android skill level

Description: Everyone on the team has limited experience in making Android applications.

Severity: Medium

Resolution: We will look through online tutorials and review labs, as well as using online resources such as StackOverflow in order to learn the appropriate skills for the tasks we are performing. Pay attention to the weekly labs.

Status: In Progress

Risk #5: What constitutes as a location?

Description: The description is unclear as to how a location is denoted (i.e. whether it records a point and uses a specific radius around that point as a location, or determining the closest landmark to that location).

Severity: Medium

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

Response: Hmm. Good question. When I started asking around the MVC offices here, one of our resident developers said that Android is “event driven”, and that this means that location changes are generated by the phone’s operating system, and cannot be specified by the developer (you). She said that it was her impression that location updates are generated according to how much you move (e.g., more than 20 feet), not according to how much time has passed. Anyway, apparently you have no control over it.

Status: Resolved

Risk #6: Where do we get music from?

Description: Do we play the music from what’s saved on their phone, or an external app such as Spotify or Pandora?

Severity: Medium

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

Status: Pending

Risk #7: What takes priority when calculating the chance of a song being played?

Description: The song being played is determined by location, time, and day of when it was last played. However, is there an equal weightage to these values, or is one value more “important” than the others?

Severity: Medium

Resolution: Location = day = time. Ties will be broken by likes

Status: Resolved

Risk #8: Default values for a new song?

Description: When a song has not been played before, what should be the default value for location, date, and time for when it was last played?

Severity: Low

Resolution: Ask the marketing team and get clarification

Response: We ran a little scenario, and figured out that this is pretty unlikely. Unless you’re totally new to the app, you’ll always have a song that was played in the morning, afternoon, and evening, resulting in at least a time match, and you’d pick a song that was played in the appropriate time range. Still, since a no-match is possible when you first start using the app, let’s say the app should be any previously played song. Don’t forget about Favorites, though. The app should favor favorite songs.

Status: Resolved

Risk #9: Are songs automatically played when entering in and out of FlashBack Mode?

Description: When a song is still playing, should a song be immediately played when entering flashback mode, or should the song be played after song completion.

Severity: Low

Resolution: Ask the marketing team and get clarification

Response: The starts in the same mode where you left off. If you’re in regular mode, nothing plays. If you’re in Flashback Mode, it automatically plays.

Status: Resolved

Risk #10: No UI/UX experience

Description: Nobody on this team can design an interface that’s pretty and useful.

Severity: Low

Resolution: Learn along the way, study other successful apps and learn from what they did right and wrong.

Status: In Progress

Risk #11: Manually selecting music Vs Flashback Mode

Description: When user selects a track manually during flashback mode, does that interrupt the flashback mode, or does it do something else?

Severity: Medium

Resolution: When you are in flashback mode, you cannot choose a track.

Status: Resolved

Risk #12: Ambiguity in chance of a song being played.

Description: When clicking the + button, playing a song, approaching a new/old location, by how much does the chance the song being played again increase/decrease?

Severity: Medium

Resolution: Everything has an equal weightage and is given a point in the probability calculation.

Status: Resolved

Risk #13: No access to location information.

Description: What happens when location is disabled or location services are unavailable due to low signal, etc.?

Severity: Medium

Resolution: Location information is ignored and the list is based on the remaining factors.

Status: Resolved

Risk #14: When app is closed, what mode or song are we in and played.

Description: When app is closed, is the last played song saved such that when the next time the app is reopened, what is played. Do we stay in last mode, or default to normal mode.

Severity: Low

Resolution: It picks up where you left off.

Status: Resolved

Risk #15: How does the app keep track of user's information?

Description: How do we track the user? Do we ask the user to register?

Severity: High

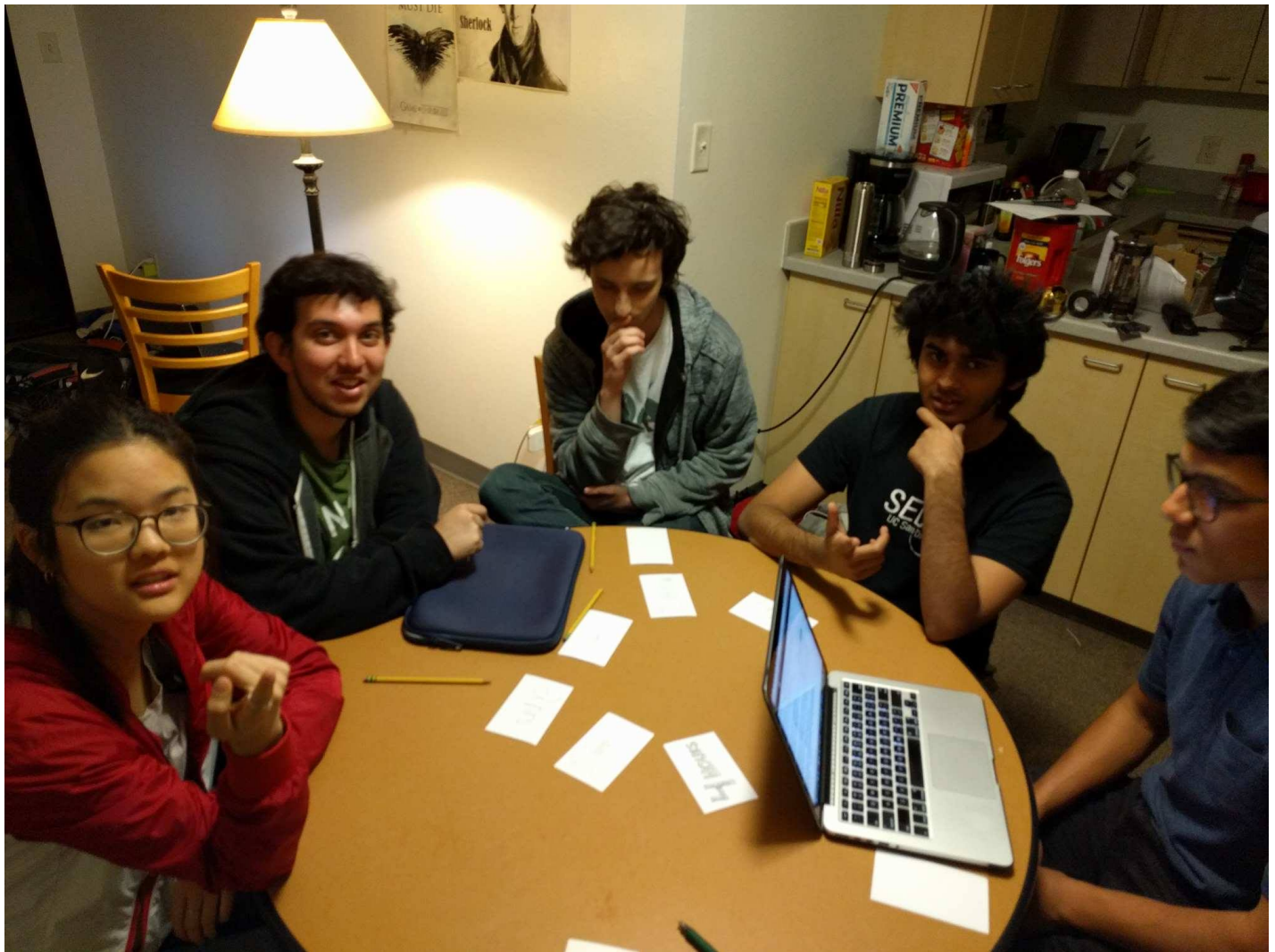
Resolution: We will store all information inside the phone's internal storage.

Status: Resolved

Velocity

Based off our readings in lectures, we are using 0.7 as our benchmark velocity.

Planning Poker



Story #	Name	Hand (hours)	False Assumptions Uncovered
1	Story 1: User picks what track to play	8, 10, 14, 10, 11, 9	<ul style="list-style-type: none">Using outside music source like Spotify to retrieve musicPredicted hours do not include JUnit and Espresso testing
1	Story 1: User picks what track to play	10, 11, 9.5, 11, 10, 11	<ul style="list-style-type: none">Ambiguities on how songs are stored (Using a List, arrayList, or array)
1	Story 1: User picks what track to play	10.5, 10.5, 10.5, 10.5, 10.5 , 10.5	(None)

2	Story 2: Album is implicitly queued when selected	6, 7, 7, 5, 4, 5.5	<ul style="list-style-type: none"> • There is a queuing feature for tracks • Data structure to queue tracks for albums (queue and stacks) • Song and album class have already been created • There is album selection in flashback mode
2	Story 2: Album is implicitly queued when selected	5, 5, 5, 5, 5, 5	(None)
3	Story 3: Flashback mode takes into account a song's last known location	4, 5, 4, 6, 5, 8	<ul style="list-style-type: none"> • No experience with Android Location • Ambiguity on what constitutes as a location (proximity to building, coordinates, etc) • Location services will always be available
3	Story 3: Flashback mode takes into account a song's last known location	5, 6, 6, 6.5, 6, 5.5	<ul style="list-style-type: none"> • Location saved is at the location when the song is in real-time (when song ends or exact location when it was last interrupted) • Location saves even when song is skipped • Android has an API to calculate proximity of location (Concluded that we should use distance formula)
3	Story 3: Flashback mode takes into account a song's last known location	6.5, 6.5, 6.5, 6.5, 6.5, 6.5	(None)
4	Story 4: User toggles between like/neutral/dislike	2.5, 3, 2.5, 5, 4, 4	<ul style="list-style-type: none"> • Clicking the "+" button twice is a quick double tap (instead of toggling) • Whether to create 3 separate buttons for like/neutral/dislike • Do not need to update these attributes in Song class
4	Story 4: User toggles between like/neutral/dislike	4.5, 4.5, 4.5, 4.5, 4.5, 4.5	(None)
5	Story 5: Seeing song information	7, 6, 6, 5, 6, 7	<ul style="list-style-type: none"> • Android does not have an API for retrieving date and time. (Unfamiliar) • Usage of database
5	Story 5: Seeing song information	6.5, 6.5, 6.5, 6.5, 6.5, 6.5	(None)
6	Story 6: Toggle Between Flashback mode and Regular Mode	11, 10, 7, 5, 10, 6	<ul style="list-style-type: none"> • Multiple screens are used between the two modes • Song selection is available in Flashback mode • Ambiguities on whether UI has been finished
6	Story 6: Toggle Between Flashback mode and Regular Mode	8, 8, 8, 8, 8, 8	(None)
7	Story 7: User can skip songs (next button) and play previous song (previous button)	2, 4, 3, 2, 3, 5	<ul style="list-style-type: none"> • Ambiguity on how to retrieve previous and next song • Ambiguity of using queue, linked list, arrayList, array, etc.

7	Story 7: User can skip songs (next button) and play previous song (previous button)	3, 3, 3, 3, 3, 3	(None)
8	Story 8: User can hear songs he/she likes more often	10, 10, 9, 7, 8, 9	<ul style="list-style-type: none"> • Great complexity of how to calculate probability • Probability of song being played in flashback mode only depends on location, day of the week, and time.
8	Story 8: User can hear songs he/she likes more often	6.5, 6.5, 6.5, 6.5, 6.5, 6.5	(None)
9	Story 9: The app should be able to remember previous data prior to being closed	6, 7, 5, 6, 4, 5	<ul style="list-style-type: none"> • Android does not have a built-in function to remember the last state before closing
9	Story 9: The app should be able to remember previous data prior to being closed	4.5, 4.5, 4.5, 4.5, 4.5, 4.5	(None)
10	Story 10: Be able to see list of albums and tracks.	6, 8, 7, 15, 9, 5	<ul style="list-style-type: none"> • Ambiguities on whether Song and Album class has been created (Majority believe Song and Album class has been created) • Screen with all albums and tracks have already been built • Screen with albums and its tracks have been created
10	Story 10: Be able to see list of albums and tracks.	15, 17, 19, 16, 19, 18	<ul style="list-style-type: none"> • Ambiguities on how songs are retrieved • Ambiguities on how to show all tracks within an Album • Android has built-in feature to show songs within an Album
10	Story 10: Be able to see list of albums and tracks.	19, 19, 19, 19, 19, 19	(None)

URL of ZenHub Project:

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-31/boards?repos=119226356>

Wireframes

