

Team #29

Team Member Name	PID	UCSD Email ID
Matthias Smyrl	A13014876	msmyrl@ucsd.edu
Stephanie Mitchener	A12861347	smitchen@ucsd.edu
Joel Loo	A12623303	jcloo@ucsd.edu
Amanda Moffitt	A14210917	amoffitt@ucsd.edu
Inga Klassy	A13534486	iklassy@ucsd.edu
Muyao Wu	A11735974	muwo06@ucsd.edu

Milestone 1 - Planning Phase**Risk Analysis**

<u>Risk</u>	<u>Description</u>	<u>Severity</u>	<u>Resolution</u>	<u>Status</u>
Not meeting enough	<i>Our schedules hardly ever mesh for more than hour a few times a week.</i>	<i>High</i>	<i>Meet during the weekends. Meet on campus to maximise the time we can spend together.</i>	<i>Resolved</i>
Falling behind schedule	<i>Not being able to finish assignments on time due to being disorganized.</i>	<i>High</i>	<i>Pick a leader, i.e "project manager" who will keep everyone on track. Rotate leaders each week.</i>	<i>Resolved</i>
Music access	<i>We don't know how we will have to access music in the app and we don't know much about using API's if that end up being the way we need to access music.</i>	<i>Medium</i>	<i>none yet</i>	<i>Unresolved</i>

Lack of Android experience	<i>Any members of our team who had any previous practice developing and/or coding android apps had shallow experiences. Even with the aid of the labs we are looking at a nice learning curve.</i>	<i>Medium</i>	<i>We all expected to learn this material by taking this class, so this learning curve was sort of expected prior to this project, and we all take it serious.</i>	<i>Resolved</i>
Multiple teams doing GUI	<i>With multiple people working on different parts of the project's GUI. There's a chance the end result will end up awkward.</i>	<i>Low</i>	<i>The individuals all working on elements of the GUI will use the completed portions as a reference.</i>	<i>Resolved</i>
One of the teammates drops the class	<i>We cannot be sure that we can follow our plan precisely because one of us can drop the class unexpectedly</i>	<i>Low</i>	<i>Continue our frequent communication so we can assess people's attitudes</i>	<i>Unresolved</i>
Missed clarifications by the customer	<i>We are responsible for being aware of clarifications our team may not have asked, through the Piazza boards</i>	<i>Medium</i>	<i>Continue to stay up to date on Piazza clarifications of assumptions, ask Bill more questions</i>	<i>Unresolved</i>

Initial Estimated Velocity

Since we have never worked with each other before and very few of us have experience working in a team of this size, our initial estimated team velocity is 0.5 so that we stay on the safe side. We also chose a lower velocity to hopefully counteract the fact that we are human and even with all of our clarifications and planning, we are still probably missing things we haven't thought of due to the fact that we have never done a project like this before.

Planning Poker



<u>User Story</u>	<u>Hand</u>	<u>Uncovered Assumptions</u>
US-1 Task 1	4,4,2,2,2,2	“it’ll probably be like 30 lines of code”, first stuff like song title, file, etc might seem like not much but may become more
	3,3,3,3,2,4	need to revote again, discovered we needed new methods
	5,5,5,5,4,4	<u>chose 5</u>
US-1 Task 2	2,2,2,3,3,1	“when the song class is done, the album is basically done as well”, assuming the song class is functional, album just needs name + songs in album
	1,1,1,1,1,2	<u>chose 1</u>
US-1 Task 3	4,4,5,3,2, not sure	assumed we would know how to use Google API
	4,4,4,6,5, 10	still not sure how many days it will take to use the Google API, conflicted on wanting to be safe on amount of days needed
	5,5,5,4,4,6	<u>chose 5</u>
US-1 Task 4	6,4,4,3,2,3	thought back to 8A and sound file project, had methods already implemented in their library so assuming Google API

		probably has something similar
	5,3,3,4,4,6	never know with different environments how retrieving works, don't know much about each other's abilities,
		<u>chose 5 after conversation</u>
US-1 Task 5	8,7,6,4,2,2	don't need to do paper prototypes, may have trouble with implementing functions depending on how we want our UI to look, need to talk more about the design in general
	6,6,5,4,2,3	search doesn't have to be a search bar (can be just a list of albums), more debate about how our design wants to be built
		<u>chose 5 after conversation</u>
US-2 Task 1	7,7,6,5,5,4	dependent on other classes like song/album, worried about debugging location/Google API
	7,7,7,6,6,5	<u>chose 7</u>
		<u>REWROTE TASK</u>
US-2 Task 1 (New)	4,4,5,5,3,4	Can just override the comparison operator, use already-written priority queue
		<u>chose 4</u>
US-2 Task 1.5	not sure, 9, 7,7,6,6	thinking it could be easy to implement a "point system" which takes all of these features into account, feel like it should be easier if there is already methods/priority queues implemented in the location class, still thinking debugging will be hard for this
	7,7,7,7,8,8	<u>chose 8</u>
US-2 Task 2	1,2,2,4,4,5	thinking a toggle between display changes wouldn't be that difficult
	3,3,3,2,2,2	<u>chose 3</u>
US-3 Task 1	4,4,3,3,2,2	the 4s think that pulling from each object might take a lot of time to debug
	3,3,3,3,4,4	<u>chose 3</u>
US-4 Task 1	2,2,2,3,3,4	not sure about things like transitions, etc that will go into the UI design
	2,2,2,3,3,3	<u>chose 3</u>
US-4 Task 2	1,1,2,2,2,3	<u>chose 2</u>

URL of ZenHub Project:

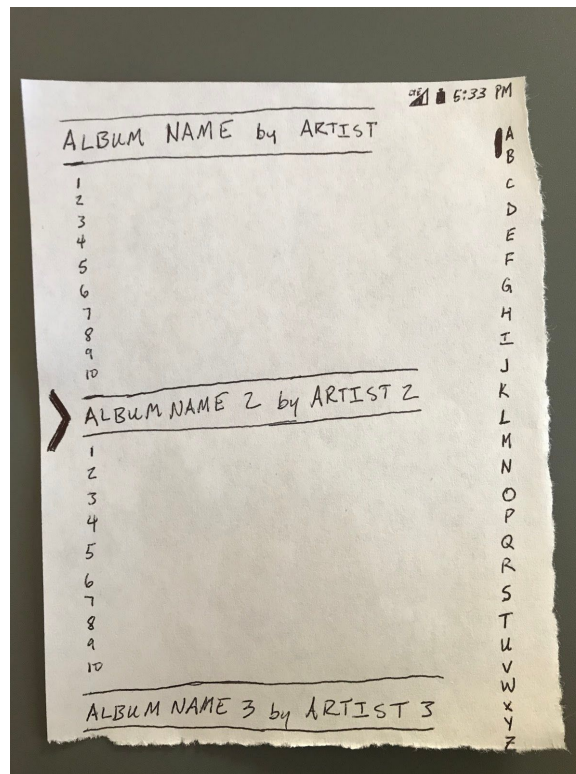
<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-29/boards?repos=119205445>

Note: Make sure to cover the below 4 items **in** your ZenHub project

- User Stories (including UI wireframes, if not included below)
- Tasks
- Iterations
- Scenario-Based System Tests (We recommend a “Developer Story” at the end of the Iteration to hold these, one Task for System Test.)

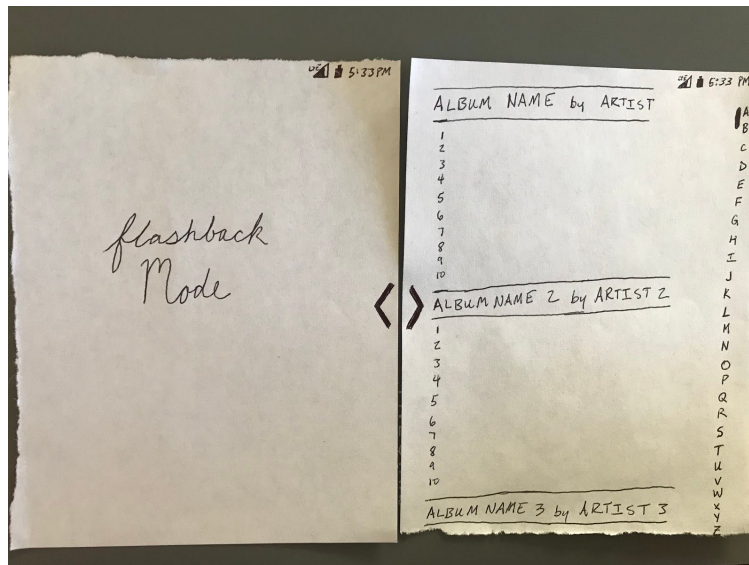
User Interface Progressions/Screens (Wireframes)

Only if you don't store User Stories in ZenHub, insert here, ordered and labelled by User Story



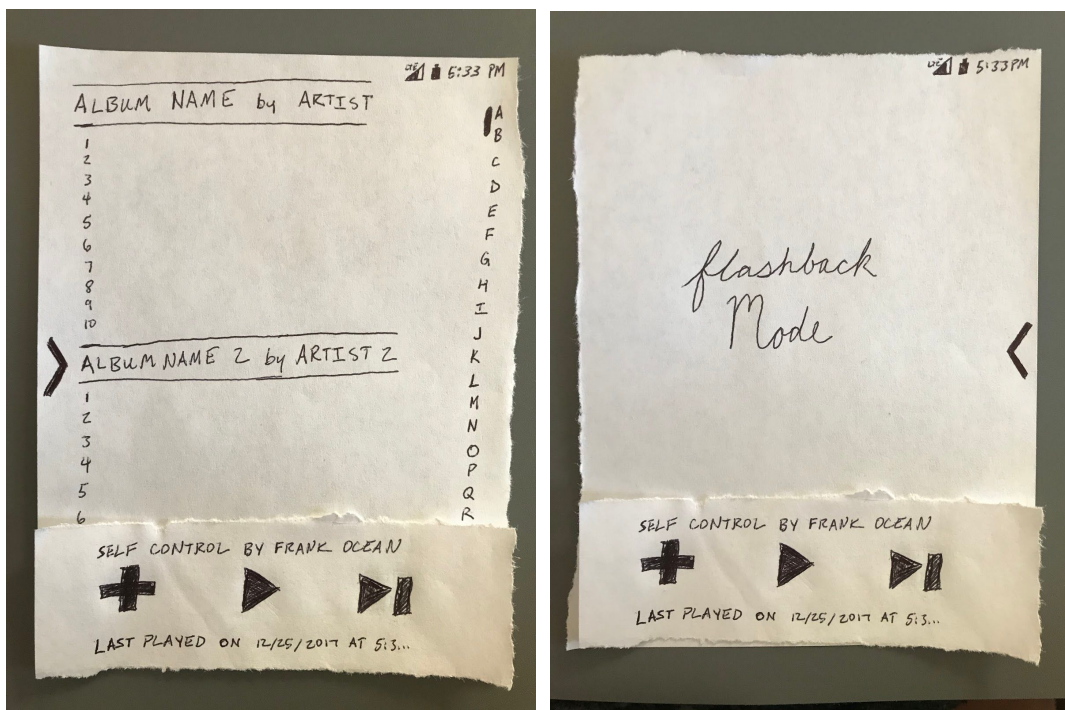
US-1 Task 5: Design UI to display available songs to user (which a user can select)

In this interface, users can scroll by either swiping on the list of albums/songs, or by using the scrollbar on the right side of the screen.



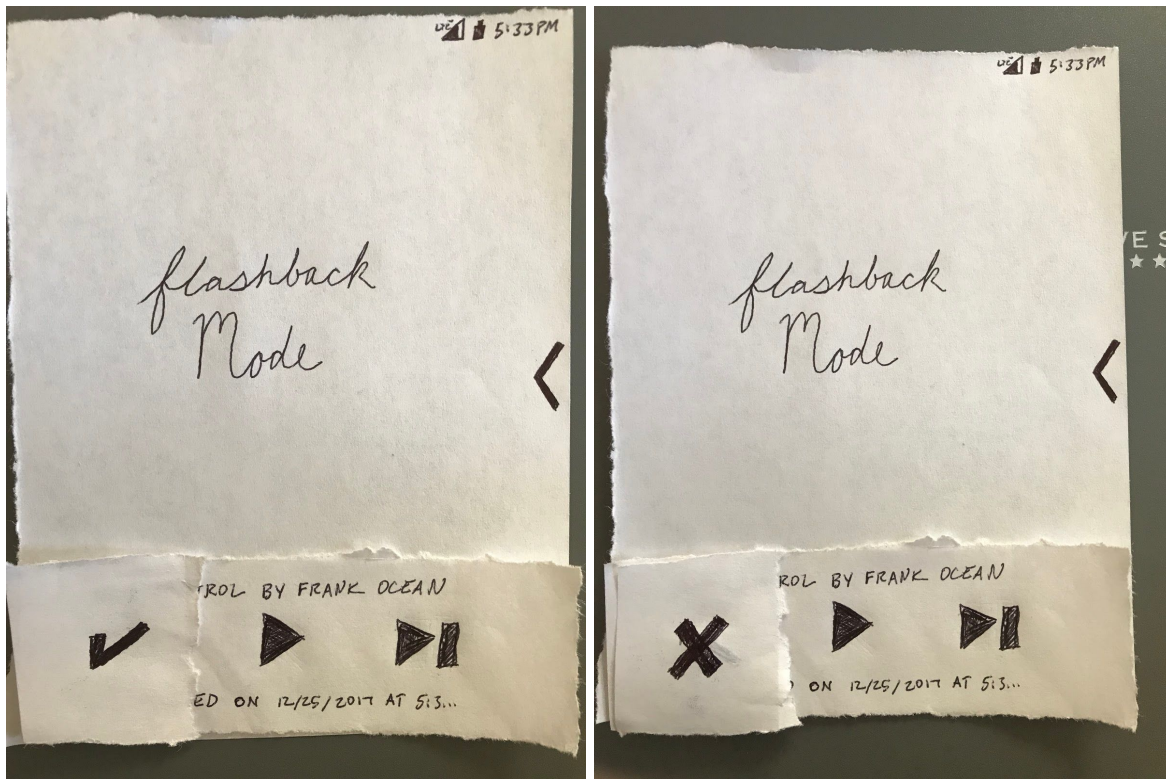
US-2 Task 2: Design a UI to make it clear that the user is in Flashback Mode

Flashback Mode has its own screen. Indicated by the arrows on the right side and the left side of the screen (for Flashback Mode and music searching, respectively), users will intuitively swipe between Flashback Mode and the regular music searching mode.



US-3 Task 1: Design UI to display current song (and the data associated with the song: date, time, place last played)

When a song is playing, a play bar appears at the bottom of the screen, displaying three buttons as well as the song title/artist and associated data. Inspired by Snapchat, the buttons and data are shown as an overlay on the screen and have a slight blur in their backgrounds, as to make the words/buttons pop out without taking away from the overall aesthetic of the application. The play bar stays visible for both Flashback Mode and the regular music searching mode and does not move after a song is chosen.



US-4 Task-1: toggle the button to display a checkmark, once pressed; toggle the button to display a “X”, once pressed; toggle the button to display a “+”, once pressed

As shown in previous photos, the left button on the Play Bar starts as a “+” sign. When a user wants to Favorite a track, the “+” button changes to a check mark, signifying that it is going to be favorited. When a user wants to dislike a song, the button is tapped again and changes to an “X” to signify that it is disliked. Finally, to make a song neutral again, the button is tapped once more and it changes back to the “+” sign. The functionality works in both Flashback Mode and in regular music searching.