Team #15 Milestone 2 Planning Phase

Team Member Name	PID	UCSD Email ID
Carlos Eduardo Matos Ribeiro	A14032489	cribeiro@ucsd.edu
Tong Wang	A91117834	tow014@ucsd.edu
Tosshaan Thapar	A14015364	tthapar@ucsd.edu
Cory Liang	A12770408	chl550@ucsd.edu
Wei Cheng Liu	A13627310	wcl011@ucsd.edu
Graham Mosbrucker	A12768355	gmosbruc@ucsd.edu

Risk Analysis

Risk: Scheduling conflicts within our team
Description: Our schedule is quite different. We do not have so much free time in common during workdays
Severity: High
Resolution: We will try to have short stand-up meetings on class days. And we'll have late night meetings for the planning phase. After the planning phase, we will divide our groups into pairs who have similar schedule so that they can meet regularly. Finally, we'll have group meetings resolving major issues during the week(late night) or on weekends.
Status: resolved

Risk: New requirements involve a lot of new technologies that we are not familiar with.
Description: The new requirements of the app add a lot of complexity to the project. None of us have experience with technologies such as keeping track of multiple users and downloading songs dynamically through the app and we have not yet had labs to help us learn them.
Severity: High
Resolution: We will try our best to develop the project. In order to support each other we will do pair programming and help teach each other on APIs and etc if one of the teammates is experienced with it.For technologies in which no one is experienced, the team will make sure to prioritize them, start early and have tasks for researching the technology. Additionally, we will use conservative velocity calculations and under-pack iterations to account for our inexperience and still have realistic goals. Also, when in doubt, we'll consult with TAs or Bill to get help and to see if our task goal is achievable.
Status: In Progress

Risk: Time Management due to increasing workloads from other classes/work/clubs
Description: We all have many different classes along with other extracurriculars that take up time off of class. And with the increasing workload that will arrive in the second round of midterms, it might be hard for teammates to find time to work on the project. If teammates fall behind, it would take some time to bring them to the current project status.
Severity: Medium
Resolution: Each teammate will be expected to keep up and get ahead with all of their work. Also, each team member will hold the others accountable for the work assigned to them and use the burn-down chart for reference to monitor productivity. Additionally, we have gained more experience with this from Milestone 1 and expect to handle it better this time. Finally, just as in the previous risk, being conservative with velocity and iterations will help us obtain realistic goals to tackle the project.
Status: Resolved

Risk: Dependencies halts production
Description: Throughout milestone 1, we couldn't work fully on our assigned part because the functionality for each assigned part are dependent on the parts that another group is assigned to. It was very hard to move forward without a working part from another pair group. Ideally, we want to work in parallel to get maximum production but it was difficult to achieve.
Severity: Medium
Resolution: Everyone in the teammate should aim to work together as a group more often. This allows each member to understand each part that is currently in progress. If one group was stuck, another group can immediately give help and resolve the problem as soon as possible to keep production moving.
Status: Resolved

Risk: Misunderstanding the requirements of the project
Description: We might misunderstand the project requirements, which could lead us going to wrong directions during development. Since the description of the project is minimal, we may misinterpret the product that the customer is asking for.
Severity: Low
Resolution: We will look on piazza and keep an updated list of the clarifications of the customer. We will also ask questions about our assumptions on piazza for the customer to clarify.
Status: Resolved

Velocity: 0.30
Justification with calculation: For Milestone 1, we estimated that we would spend about 31 hours in total to complete the app. But each pair groups (3) spent about 34 hours each working on the app. Dividing the total time by the estimated time, we get a velocity of about 0.30. Even though this part of the project is objectively more complicated, we were more conservative with our estimates this time, so we believe these facts cancel each other out and it is safe to assume that our velocity will be similar.

Planning Poker

Following guidelines on Moodle (Link: https://csemoodle3.ucsd.edu/mod/page/view.php?id=1321)
Insert your work here

User Story	Hand played	Assumptions Uncovered
Vibe Mode	30 30 30 30 40+ 40+	The greatest difficulty of Vibe mode is incorporating other users into the app and being able to store information from all of them. Even though we predict that will be extremely complicated given our current skill-set, after this is solved, implementing Vibe mode will be equivalent to slightly modifying the previous Flashback mode, and should be somewhat straightforward.
Vibe Mode	**30 30 30 30 30 30**	None
Downloading New Music	10 10 10 20 20 10	The selected source to download songs might not have a very intuitive and easy to use API. Or, we might have to provide our own source of songs. We do not know how to use the media player with an ID that is not known beforehand; it might be very tricky.
Downloading New Music	20 20 20 10 10 10	At this point it is quite unclear whether the app is supposed to download music in vibe mode automatically (will look to clarify) and that doubt is complicating our estimate.
Downloading New Music	**20 20 20 20 20** 10	None
New Information About Song	3 5 5 10 10 20	Most of the functionality regarding information about each user will already have been covered in Vibe Mode; this is more a visual issue. However, there will still be some complication regarding how to name proxy users, how to effectively pull data

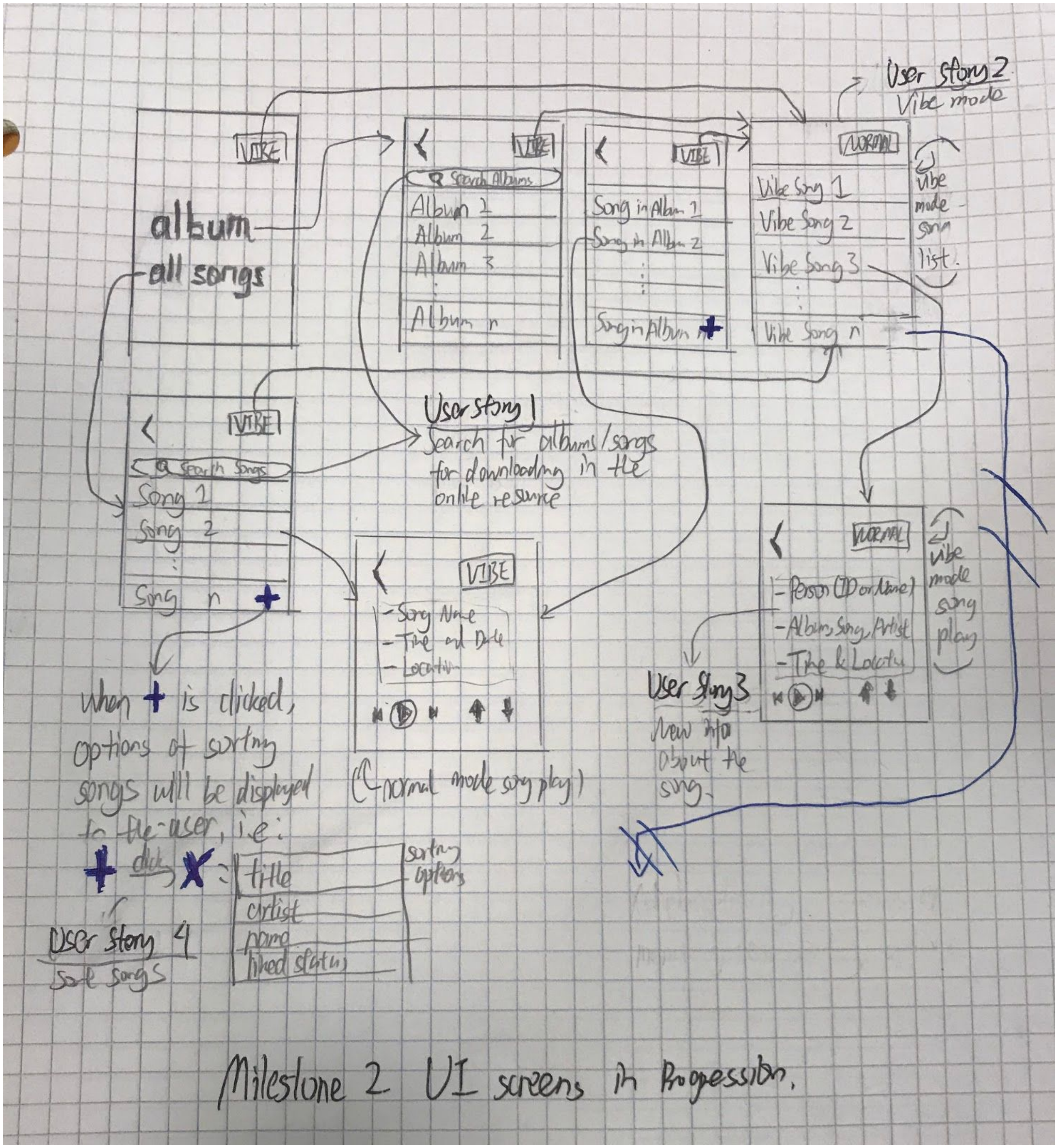
		from database and what to do if multiple friends have listened to the song.
New Information About Song	**10 10 10 10 10** 5	None
Sorting Songs Based On Different Parameters	5 5 10 10 10 20	As of right now we are not sure how to write out own comparator class, which will be essential for this part of the app. There were also concerns about the efficiency of the implementation depending on the number of songs, but realistically no given user will have downloaded enough songs for this to be a concern.
Sorting Songs Based On Different Parameters	5 **10 10 10 10 10**	None
Disliking/deleting songs	5 5 5 5 10 10	This user story is somewhat complicated by the fact that there is a need to keep track of songs that were deleted to ensure that they do not show up again in Vibe mode. There is also need to test this story extensively. However, keeping the list is not necessarily complicated, and a lot of this functionality will build on methods already implemented in milestone 1, reducing the need to write a lot of code.
Disliking/deleting songs	5 5 5 10 10 10	Espresso tests can be very time consuming to be written. However, this story does not involve a lot of UI and therefore there is no need to write espresso tests for it. Additionally, most of us are assuming that the actual deleting of a song will be trivial, but that is not necessarily true since none of us know how it is done. In that case, it might be better to be in the side of caution.
Disliking/deleting songs	**10 10 10 10 10** 5	None

Photo of us doing planning poker:
We used our laptop instead of cards. We have a list of time estimates and highlighting a time to indicate the estimate each member believes is the time estimate. We show our laptop to each other at the same time each round of the planning poker.

URL of ZenHub Project:

https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-15-1/boards?repos=119942516

User Interface Progressions/Screens (Wireframes)



Milestone 2 UI screens in Progression.