Team #15

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

Milestone 1 - Planning Phase

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: Not familiar with app development
Description: Compare to full-time app developers, we are just students with no industry and little Android development experience. Consequently, we don't have much experience with different technologies.
Severity: Medium
Resolution: We will try our best to develop the project. However, to avoid situations that involve developing something out of our capability, we will constantly analyse team risks for our

tasks. We will do pair programming and help teach each other on APIs and etc if a teammate is experienced with it. If no one is experienced, the team will research 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 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 from midterm week and onwards, it will be hard for teammates to find time to work on the project, or keep up with other coursework. If teammates fall behind, it would take some time to bring them to the current project status.
Severity: High
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. 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: Misunderstanding the requirement 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: 0.5
Justification with calculation: Based on our two meetings, we reasoned that our velocity should start at 0.5. Since the book suggests that we should start at 0.7, but we are also students with conflicting schedules and less experience than professionals. Also to account for the risks, we chose a very low velocity. Furthermore, a conservative velocity will help us obtain realistic goals for the first iteration.

Planning Poker

Story C	Story Name	Hand	Assumptions Uncovered
1	User selects tracks to play (normal mode)	3 5 5 5 5 Long(10+)	The finalized UI does not have to be ready to get this feature working. It can just be as rudimentary as possible to view the functionality. However, since this will be the first story, the overhead with getting started will be larger than usual.
1	User selects tracks to play (normal mode)	**5 5 5 5 5 5**	(None)

2	Playing music on Flashback Mode	5 5 10 10 10 Long (10+)	Location functionality is the most complicated part of this user story. However, the Google location API should facilitate considerably. Still, none of us know how to operate it or exactly how it works, so we should still be cautious in our estimates. Furthermore, breaking ties might prove to be complicated.
2	Playing music on Flashback Mode	**10 10 10 10 10 10**	(None)
3	Switching between Flashback and Normal Mode	3 3 **5 5 5 5**	We are going to design separate UI screens for each possible mode. However, they will already have been created on previous stories; switching is not a lot of code. Yet, there is a lot of detailed testing to be performed for specific situations (switch during song, etc). Additionally, we do not know how to make the app "remember" the state it was before it was shut down.
3	Switching between Flashback and Normal Mode	**5 5 5 5 5** 3	(None)
4	Last Played Information	**3 3 3 3 3 3**	Story is in fact quite simple, but there are a few edge cases to test (end of song vs beginning, etc)

5	Mark a song as favorite	1 3 3 3 5 10	The “ranking system” used to determine the order of songs in flashback mode will already have been mostly implemented during Story 2; adding a favorite component will just be a small addition. The GUI component of this
5	Mark a song as favorite	1 3 3 3 5 5	Even though they are some edge cases, they can be tested reasonably quickly, and the actual programming component is fast enough to compensate for that anyway
5	Mark a song as favorite	**3 3 3 3 3**	(none)
6	Dislike a song	3 3 3 3 5 5	There are some tricky edge cases to consider, especially when a song is disliked while playing (song is to be skipped and that means different behaviors in other situations). However, maybe there would already a song queue in either situation and skipping would be as simple as going to the next song in the queue.
6	Dislike a song	3 **5 5 5 5**	(none)

URL of ZenHub Project:

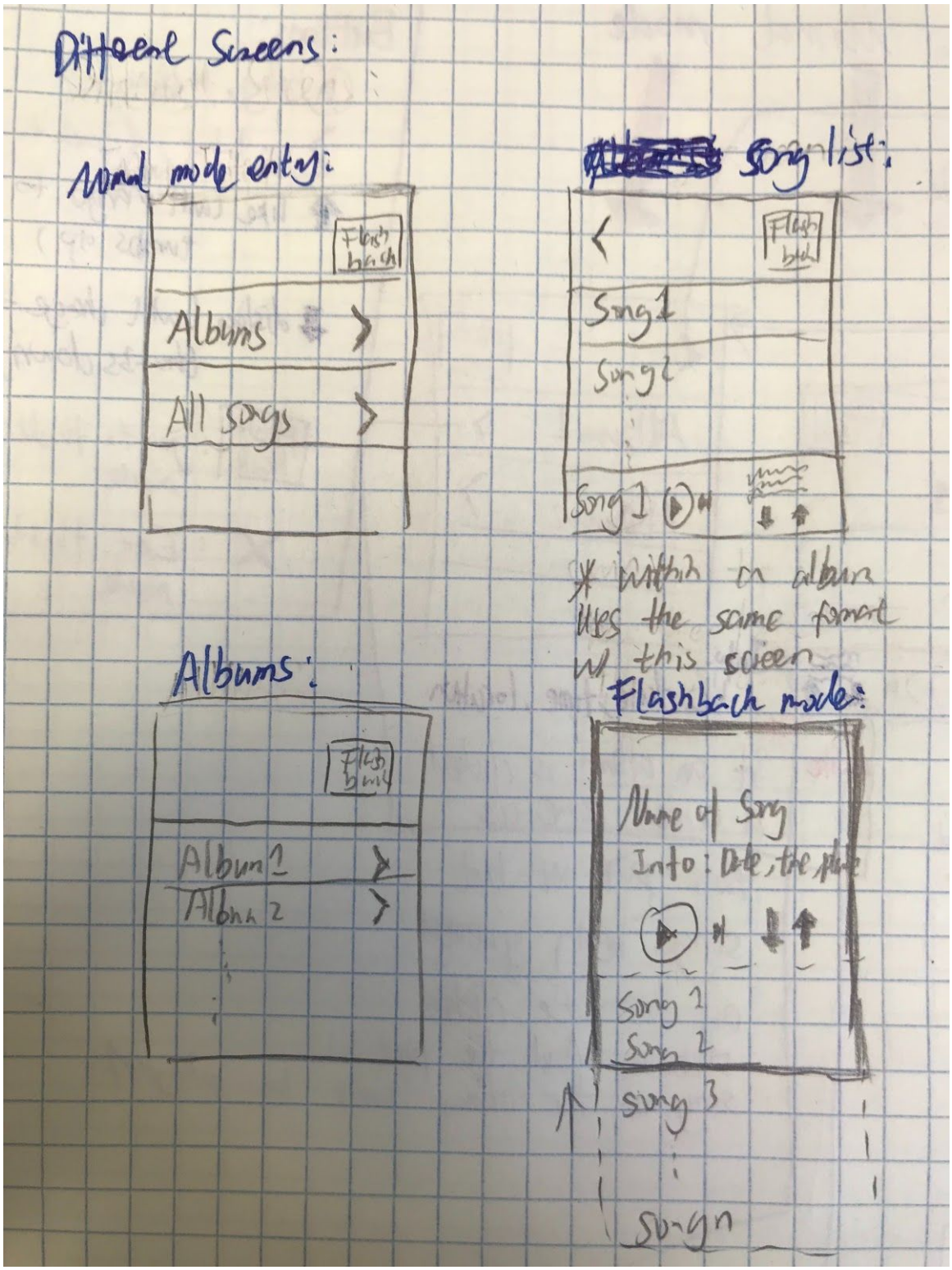
Insert ZenHub URL here

https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-15-1

User Interface Progressions/Screens (Wireframes)

UI Progressions:

1. **Different Screens Types**



2. UI screens in progressions