Team 7

Team Member Name	PID	UCSD Email ID
Mikhail Boulgakov	A13543506	mboulgak@ucsd.edu
Simar Chhabra	A13891316	sichhabr@ucsd.edu
Eldon Tay	A12963052	ektay@ucsd.edu
Amritansh Gupta	A13831681	amg080@ucsd.edu
Nirha Patel	A13695370	nrp001@ucsd.edu
Val Gonzalez	A13188505	vgd001@ucsd.edu

Milestone 1 - Delivery Phase

Refer the grading rubric (link: https://csemoodle3.ucsd.edu/mod/page/view.php?id=1326) for more details.

Here we are following a sample repository (github.com/CSE-110-Winter-2018/Default-Repository) to provide sample links for code snippets, branch etc. as per requirements.

Remember to push this document in the *docs* folder of your project in the branch "milestone1_delivery", like: github.com/CSE-110-Winter-2018/Default-Repository/blob/milestone1_delivery/docs/Milestone1_DeliveryPhase.pdf

Software design

1. Checkout a new branch "milestone1_delivery" , push the final code to it and provide a link to the code folder of your team project repository on Github
   https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-7/tree/milestone1_delivery/app/src/main/java/com/cse110/flashbackmusicplayer

2. Provide an example or two from your code (best if you insert relevant links to code snippets) where SRP & DRY were demonstrated.
   1. The whole songdatabase class, but especially the following functions are a demonstration of SRP:
   https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-

7/blob/milestone1_delivery/app/src/main/java/com/cse110/flashbackmusicplayer/SongDatabase.java#L94-L185
These functions have only one job, which is described by the name, so you don't have to worry about them manipulating data or something unexpected.

2. The following 4 classes follow the strategy pattern. UserState is the interface that UserStateImpl, UserStateSnapshot, and MockUserState implement. UserStateImpl uses calendar and system time to constantly update itself. UserStateSnapshot is an immutable object that stores the data in a UserState at a point in time. MockUserState is set with the current date and time by the developer.
https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-7/blob/milestone1_delivery/app/src/androidTest/java/com/cse110/flashbackmusicplayer/MockUserState.java
https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-7/blob/milestone1_delivery/app/src/main/java/com/cse110/flashbackmusicplayer/UserState.java
https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-7/blob/milestone1_delivery/app/src/main/java/com/cse110/flashbackmusicplayer/UserStateImpl.java
https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-7/blob/milestone1_delivery/app/src/main/java/com/cse110/flashbackmusicplayer/UserStateSnapshot.java

3. We noticed that the album ui, track ui, and flashback ui were very similar and behaved much the same. To achieve that functionality we had a lot of duplicate code. To fix that we gave each of those three activities a UI object that draws the UI.
https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-7/blob/milestone1_delivery/app/src/main/java/com/cse110/flashbackmusicplayer/SongCallbackUI.java#L125-L187

Testing

Link(s) to all the JUnit and/or Espresso tests and logs of untestable non-trivial methods.
UnitTests: For every single public method of helper classes (song.java, songdatabase.java, timesegment.java)
https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-7/blob/milestone1_delivery/app/src/test/java/com/cse110/flashbackmusicplayer/TimeSegmentTest.java#L1-L47

https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-7/blob/milestone1_delivery/app/src/androidTest/java/com/cse110/flashbackmusicplayer/SongClassTest.java#L1-L200
https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-7/blob/milestone1_delivery/app/src/androidTest/java/com/cse110/flashbackmusicplayer/SongDatabaseTest.java#L1-L366

Scenario Tests: For every single scenario we had in our user stories (around 2-3) we have a test. For every single scenario based system test we also have a file. There are two many to link the snippets here, so I'm just going to link the folder.
https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-7/tree/milestone1_delivery/app/src/androidTest/java/com/cse110/flashbackmusicplayer
All of these scenarios use espresso to run.

Logging: Because we couldn't do unit tests on some classes (such as the activities), we used graybox testing:
https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-7/tree/milestone1_delivery/logs
We first created a log.txt file for every single class that we logged. We made sure to test every single possible path of execution and save the output of the tests. Next, we created logs for our scenario based system tests (we also have espresso unit tests for sbst).

ZenHub

1. Insert a valid link to your Zenhub board covering all the required points from the rubric
   https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-7/boards?repos=119468919

2. Insert a valid link to the burndown chart
   https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-7/reports?report=burndown&milestoneId=3082427&showPRs=false

GitHub

1. Insert a valid link to the contribution chart of all the contributors,
   https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-7/graphs/contributors?from=2018-01-28&to=2018-02-18&type=a

2. Checkout a new branch "milestone1_delivery" , push the final code to it and provide a link to it for final code review
   https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-7/tree/milestone1_delivery