

Team #<34>

Team Member Name	PID	UCSD Email ID
Yidong Luo	A14009868	yil623@ucsd.edu
Yixin Chen	A13824695	yic286@ucsd.edu
Samprith Reddy Kalakata	A92126278	skalakat@ucsd.edu
Xiangyi Gong	A14181245	xig074@ucsd.edu
Jay Gandhi	A12500586	jgandhi@ucsd.edu
Yitong Liu	A12961067	yil445@ucsd.edu
Rajit Dang	A92067342	r3dang@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

Note: we named our branch milestone1_delivery_final

link:https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/tree/milestone1_delivery_final

2. Provide an example or two from your code (best if you insert relevant links to code snippets) where SRP & DRY were demonstrated.

Example: You can provide the link to a code snippet in Github as

https://github.com/CSE-110-Winter-2018/Default-Repository/blob/milestone1_delivery/MyApplication/app/src/main/java/com/andriod/helloworld/MainActivity.java#L23-L27

Notice how specific line numbers in the concerned source code file have been mentioned in the URL above.

Example of Single Responsibility Principle (SRP) and Don't Repeat Yourself (DRY):

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/main/java/com/example/jaygandhi/flashbackmusicteam34/Track.java#L22-L189>

Track.java stores the information pertaining to a single song (the album it belongs to, artist name, last played location, last played time and day etc) it only allows users to get (using getters) information pertaining to the track and update it (using setters). It also allows us not to duplicate code as our implementation the same Track objects are used to build flashbacks queue, album queues and playback queues which are then passed into the PlaySongActivity.java file where the songs in the queues are played.

Example of Single Responsibility Principle (SRP) and Don't Repeat Yourself (DRY):

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/main/java/com/example/jaygandhi/flashbackmusicteam34/Library.java#L14-L68>

Library.java stores a list of all songs (Track objects) and all albums (Album objects). This class is only responsible for loading the songs in the user's library and loading all the albums in the user's library. It provides functionality to load tracks and albums for the Playback queue, Flashback queue and Album Queue.

Example of Don't Repeat Yourself (DRY):

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/main/java/com/example/jaygandhi/flashbackmusicteam34/PlaySongActivity.java#L103-L106>

PlaySongActivity.java handles playing songs or a list of songs regardless of whether in flashback or regular playback mode. We pass in a Queue (with objects that contain songs) into the PlaySongActivity.java which handles playing

(optional) Screencast - demo of all milestone requirements

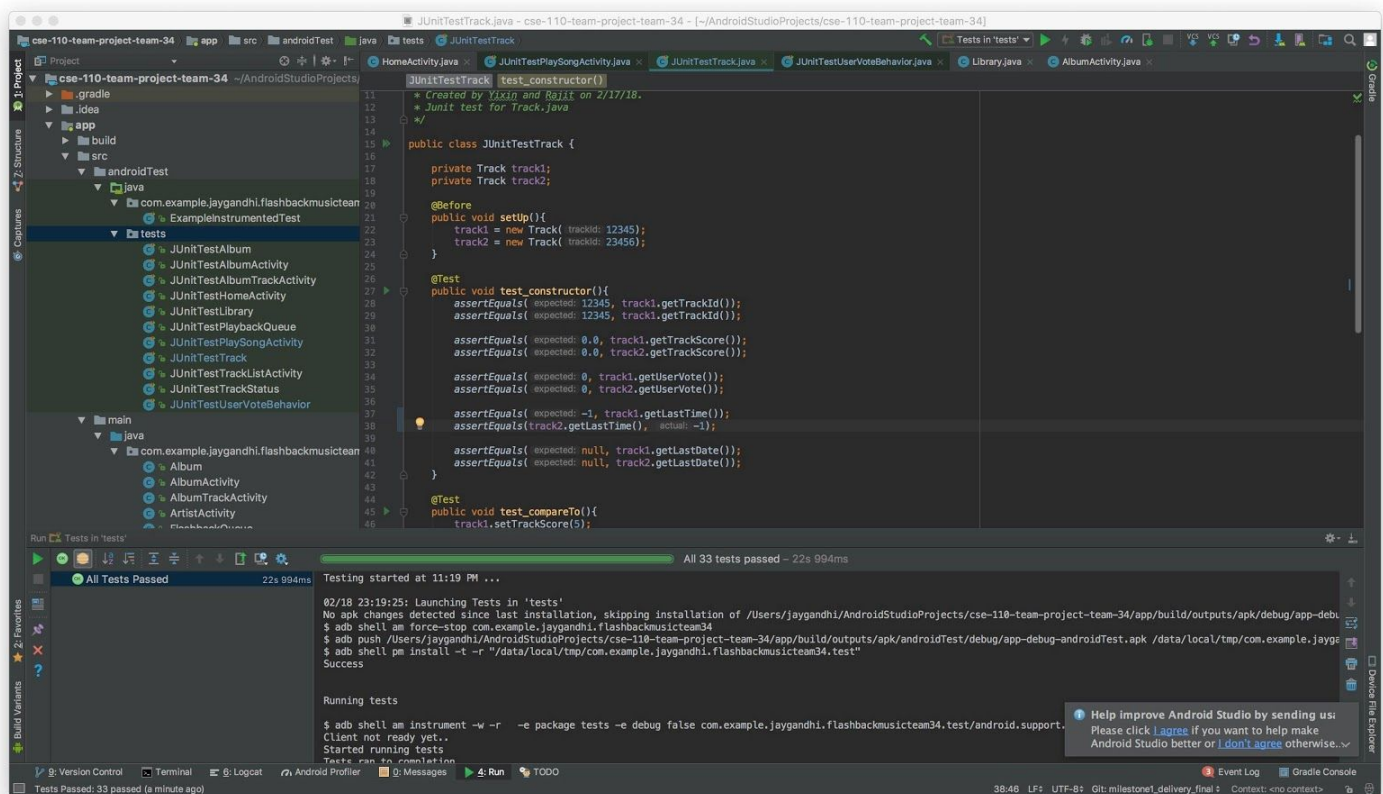
Screencast of your app either on a real Android device or an emulator covering all the required points in the “demo” section of the grading rubric (link: <https://csemoodle3.ucsd.edu/mod/page/view.php?id=1326>).

Note: The screencast is optional, the actual demo will be in-person, details about which shall be posted soon.

Helpful link to know more about screencasts and ways to capture it:

<http://edutechwiki.unige.ch/en/Screencast>

Testing



Link(s) to all the JUnit and/or Espresso tests and logs of untestable non-trivial methods.

1. JUnitTestAlbum.java:

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitTestAlbum.java#L32-L37> Testing Album class constructor

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitTestAlbum.java#L39-L45> Testing Album class addTrack() method

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitTestAlbum.java#L47-L53> Testing Album class removeTrack() method

2. JUnitTestAlbumActivity.java:

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitAlbumActivity.java#L57-L61> Testing AlbumActivity class header

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitAlbumActivity.java#L63-L69> Testing AlbumActivity class list view

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitAlbumActivity.java#L71-L85> Testing AlbumActivity class item click

3. JUnitAlbumTrackActivity.java:

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitAlbumTrackActivity.java#L65-L70> Testing AlbumTrackActivity class header

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitAlbumTrackActivity.java#L72-L78> Testing AlbumTrackActivity class list view

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitAlbumTrackActivity.java#L80-L93> Testing AlbumTrackActivity class item click

4. JUnitHomeActivity.java:

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitHomeActivity.java#L30-L48> Testing Button text displays correctly HomeActivity

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitHomeActivity.java#L53-L68> Testing whether Playback activity opens on clicking flashback button.

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitHomeActivity.java#L70-L81> Tests whether album activity launches when albums button clicked.

Untestable non-trivial behavior: Flashback Mode Button. It cannot be tested because PlaySongActivity will appear only when there are songs in flashback queue.

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/main/java/com/example/jaygandhi/flashbackmusicteam34/HomeActivity.java#L95-L96>

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitHomeActivity.java#L88-L103> Tests whether SongActivity launches when Songs Button is clicked.

5. JUnitLibrary.java:

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitLibrary.java#L13-L19> Tests whether songs are loaded

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitLibrary.java#L21-L25> Tests whether albums are loaded

6. Untestable non-trivial method: MusicListActivity.java is an abstract class. Its functions have already been tested in the subclasses. The messages are logged in MusicListActivity.java.

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/main/java/com/example/jaygandhi/flashbackmusicteam34/MusicListActivity.java#L28>

7. JUnitTrack.java:

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/tests/JUnitTrack.java#L26-L42> Testing track Constructor

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/tests/JUnitTrack.java#L26-L42> Testing track CompareTo method

8. JUnitTrackListActivity.java:

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/tests/JUnitTrackListActivity.java#L59-L63> tests whether header displays correctly

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/tests/JUnitTrackListActivity.java#L65-L71> Testing whether track list view is generated correctly

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/tests/JUnitTrackListActivity.java#L73-L86> Tests whether clicking song launches play song activity

9. JUnitTrackStatus.java:

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/tests/JUnitTrackStatus.java#L16-L19> tests whether value of neutral_status is 0

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/tests/JUnitTrackStatus.java#L21-L24> tests whether value of like_status is 1

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/tests/JUnitTrackStatus.java#L26-L29> tests whether value of dislike_status is -1

10. JUnitUserVoteBehavior.java:

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/tests/JUnitUserVoteBehavior.java#L73-L76> Tests UserVoteBehavior constructor is correct

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/tests/JUnitUserVoteBehavior.java#L78-L111> Tests whether buttons update user status correctly

11. JUnitPlaySongActivity.java:

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/tests/JUnitPlaySongActivity.java#L83-L90> Test the displaying metadata

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitTestPlaySongActivity.java#L93-L104> Test the displaying historical data(time and location that the song was played last time)

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitTestPlaySongActivity.java#L107-L121> Test the play/pause button

Untestable trivial behavior: update the gps location. Log when set up the location manager and whenever get location update callback.

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/main/java/com/example/jaygandhi/flashbackmusicteam34/PlaySongActivity.java#L437>

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/main/java/com/example/jaygandhi/flashbackmusicteam34/PlaySongActivity.java#L444>

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/main/java/com/example/jaygandhi/flashbackmusicteam34/PlaySongActivity.java#L76-L77>

12. JUnitTestPlaybackQueue.java

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitTestPlaybackQueue.java#L48-L64> Tests whether queue is sorted correctly

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitTestPlaybackQueue.java#L67-L100> Tests whether likeing and dislikes affects the flashback queue correctly

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitTestPlaybackQueue.java#L103-L159> Tests whether different time of day affects flashback mode correctly

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/blob/master/app/src/androidTest/java/ests/JUnitTestPlaybackQueue.java#L162-L218> Tests whether Location updates alone affect flashback queue correctly.

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-34/boards?repos=119413615>

2. Insert a valid link to the burndown chart
-

Burndown chart for milestone 1

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-34/reports?report=burndown&milestoneId=3080199&showPRs=false>

Burndown chart for milestone 2

<https://app.zenhub.com/workspace/o/cse-110-winter-2018/cse-110-team-project-team-34/reports?report=burndown&milestoneId=3080201&showPRs=false>

Note: Someone on our team decided to move to all our Zenhub issues from closed to done, which made our burndown chart look like it dropped in one day. If you look at the history of the issues and tasks, you will see that they were closed appropriately and were interspersed throughout development. If needed we would be happy to provide further explanation during our demo.

GitHub

1. Insert a valid link to the contribution chart of all the contributors,

Our Link:

<https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/graphs/contributors>

2. Checkout a new branch “milestone1_delivery”, push the final code to it and provide a link to it for final code review

Our link:

https://github.com/CSE-110-Winter-2018/cse-110-team-project-team-34/tree/milestone1_delivery_final
