

City University of New York (CUNY)

CUNY Academic Works

Open Educational Resources

John Jay College of Criminal Justice

2019

CSCI 380-04 Final Project Requirements

Bhargava Chinthirla
CUNY John Jay College

Eric Spector
CUNY John Jay College

NYC Tech-in-Residence Corps
rdomanski@sbs.nyc.gov

[How does access to this work benefit you? Let us know!](#)

More information about this work at: https://academicworks.cuny.edu/jj_oers/23

Discover additional works at: <https://academicworks.cuny.edu>

This work is made publicly available by the City University of New York (CUNY).
Contact: AcademicWorks@cuny.edu

CSCI 380-04 Final Project Requirements

- **Technical Requirements:**
 - Must use some sort of API (similar to how you used Spotify's API for assignment 3)
 - Here is a list of public APIs: <https://github.com/toddmotto/public-apis>
 - Must use SharedPreferences in some sort of manner to persist user data
 - Make sure to follow the Mobile Application Layers that we've been talking about, it'll help with unit testing the domain layer
 - Send me a link to your project's public GitHub repo
 - Domain layer logic MUST be unit tested
- **Presentation Requirements:**
 - Each group member must present
 - Presentation should contain the following elements
 - Introduction
 - Timeline and milestones
 - Group roles and responsibilities
 - Agile user stories
 - Lessons learned
 - Demo
 - Presentation shouldn't be longer than 20 minutes per group

Here's an example application with Spotify's API:

- Data layer:
 - Spotify's APIs to search for an artist and get their top tracks:
 - <https://developer.spotify.com/console/get-search-item/>
 - <https://developer.spotify.com/console/get-artist-top-tracks/>
 - SharedPreferences to store an artist id
- Domain layer:
 - Utility class to convert an artist model from the data layer to an artist model in the presentation layer
 - Utility class to convert a track model from the data layer to a track model for the presentation layer
- Presentation layer:
 - Activity 1: Search screen for users to search for an artist. On successful search, user will see a list of artists matching their search query. Tapping on an artist will persist the artist's id in SharedPreferences and take the user to Activity 2
 - Activity 2: reads artist's id from SharedPreferences and loads their top tracks in a list

Grading Rubric:

- Application (50%):
 - Provide a **readme.md** file explaining what the application does for each mobile application layer
 - Provide a link to the API, and which endpoints the application uses
 - Your grade for the application portion will be split as follows:
 - 33%: Presentation layer (with at least 2 activities)
 - 33%: Domain layer, fully unit tested
 - 33%: Data layer using your API's endpoints and SharedPreferences
 - Your project **must** compile. If you submit a final project that does not compile, you will receive a 0% for this portion of your grade. Make sure to fully run your app on an emulator and test out different features on it to make sure they work properly.

- Presentation (50%):
 - Each group member must present
 - Introduction
 - Timeline and milestones
 - Group roles and responsibilities
 - Agile user stories
 - Lessons learned
 - Demo
 - Presentation shouldn't be longer than 20 minutes per group

- Extra credit (20%):
 - If you write an espresso test going through a happy path for your application, you will get an extra 20%. There will be no support given for espresso testing, so this work must be researched on your own. As an intro, here is a youtube link which may help: <https://www.youtube.com/watch?v=kL3MCQV2M2s>