Assignment 2

Do **NOT** share your solution, source code and/or idea. This is an individual effort assignment.

Cheating and Plagiarism
Each student must be aware of the College's policy regarding cheating and plagiarism. The policy relating to academic dishonesty, as outlined in the Student's Rights and Responsibilities, Code of Conduct and Procedures, and will be fully enforced.

Any student caught cheating in an individual test or assignment is subject to sanctions. See College Academic Policy.

Be sure to read the following general instructions carefully:

Assignment Requirements:
Please submit the following:

1. Pair Programming https://www.youtube.com/watch?v=fQ-x-T34z9w
2. This assignment can be done individually or in a pair. If planning to work with a team, come with a team name, do not use space or underscore in the team's name.
3. Pair means a maximum of two members team.
4. If pair is selected, expected both to contribute, github will be used evaluate each member contributions through their commit and push. You should be familiar with how to contribute and use github to collaborate. Person not contributing will get zero.
5. Collaborations must be done through git hub by pulling and pushing.
6. Please update your github Id, to be YourFullName4Digits, i.e. JohnSmith1111.
7. Replace John with your firstname, Smith with your last name, and 11111 with your studentId.
8. Use Navigation Drawer, or Sliding Window
https://www.tutlane.com/tutorial/android/android-navigation-drawer-sliding-menu

https://github.com/haki11/NavigationDrawer

9. No requirement to clone the above, use a reference.
10. This assignment can be completed individually or by two students in a team (using pair programming technique).
11. Minimum API 28
12. Target API 33.
13. Create a new project, File -> New -> New Project.
14. Project name must be YourFirstNameAssign2, i.e. JohnAssign2 (If team, FirstName1FirstNam2Assign2).
15. Module name must be YourFirstNameAssign22, i.e. JohnAssign22 (If team, FirstName1FirstNam2Assign22).

16. Package name must be yourfirstname.lastname.studentid, i.e. john.smith.s11111 (if team, firstname1.firstname2.teamName).
17. MainActivity name must be YourFirstNameActivity, i.e. JohnActivity (If team, FirstName1FirstNam2Activity).
18. Java classes must start with upper case.
19. On each layout xml file, all object Ids must start with your firstName, i.e. `android:id="@+id/johnTV" (if team, have both names).`
20. All resources in strings.xml, no hardcoding of text.
21. **Must Support English and French** (You can translate just three strings).
22. Full Name, student ID, and section number, must be on ALL JAVA classes as **a comment**. Name and student id of both participants.
23. Full Name, student ID, and section number, must be on both strings.xml as **a comment**. Name and student id of both participants.
24. In strings.xml, app_name value must be your full name and student id (or the team name!).
25. **MUST** use Fragments for the individual screens.
26. Use homescreen icon different than the default Android icon.
27. **Intercept user press** on back key and display Alert Dialog whether user wants to stay on app or exit. Use proper icon for the dialog.
28. Application must not have compilation errors.
29. Application must not crash.

Github Requirements: Follow instructions below (50% will be deducted if you are not following any of the github reqs):

30. Please Update your github Id, to be YourFullName4Digits, i.e. JohnSmith1111.
31. Repository name must be **FullNameAssign2** (i.e. JohnSmithAssign2), or the team name2. No spaces or underscore in the repo name.
32. Make your repository **private** (not public).
33. One of the team members to invite the professor as a collaborator (invite haki11). Take a screenshot of the invitation.
34. Must Commit on your machine and push into github.
35. After each commit, push.
36. Must be a minimum of **20 commits/pushes**. Verify you have meaningful commit messages. (Each member must have a minimum of 10 commits/pushes!). If one person, must have 20 commits as a minimum.
37. Every time you want to commit, select root directory, Commit using Git -> Commit Directory. **The entire project must be committed** and not just the module.
38. Add .gitignore file with the appropriate content into the project folder. Add your name and student id as a comment into the first line of

.gitignore (or the both members names and students ids). Update content from link https://www.toptal.com/developers/gitignore

Please note the file .gitignore must be at project level and not module level (**Delete** the one in your module).

39. Add README.md file into the project folder, with the appropriate content explaining about your app. Use the appropriate tagging format.
40. Add Your name and student id as the first line on the file README.md (or the both members names and students ids). https://www.makeareadme.com/

41. **Include an image in your README as in the example:**
https://github.com/johncodeos-blog/CheckInternetAndroidExample

Please note file must be at project level and not module level.

42. Verify the folder **.git** is included in your submission (located in the project folder), as it has all the git history (**Submission without .git folder will get 0).** It could be hidden on your PC.

https://www.technipages.com/show-hidden-files-windows

43. The entire project should be committed and pushed where it can be cloned.
44. Once you have finished the changes, please make sure you commit and push. Mark the commit as Final, with proper comment.

Submission Criterion:

1. **Take screenshots of the different screens.**
2. **Take screenshots of the Firebase DB showing clearly the data stored in the table.**
3. **Screenshot of github invitation.**
4. **Put all screenshots into a document, with proper comments.**
5. **Put team members and Ids at the top, with proper formatting and save as PDF.**
6. **Create a folder with yourfullName (or team name), include the below prior to zipping and uploading**

a. **Android Studio Project. Verify .git folder is included in the Android Studio project, at the project level.**
b. **PDF file.**

7. **Github link as a comment, when you submit. I should be able to click on the link, without the need to download and unzip your project.**
8. **I will use the github link to clone your project and mark it.**
9. **I will only attempt to download your project if can NOT clone your project, or did not invite me (50% will be deducted!)**
10. **Full Name and student Ids, as a comment for both members.**
11. **Coordination between the two members, and who will submit. No submission, will get zero. Only one member to submit!.**
12. **Team member who is not contributing, will get 0, github will be used to validate each member contribution.**

Functionalities: • Correct implementation of functionalities • All activities, fragments working, proper naming of activities, variables, and methods. Provide comments. Provide explanation when asked during the demonstration of the app.	40% 40% 25%
UI friendliness (proper layout, controls, styles, themes, images)	25%
Declaring resources in proper resource files	20%
Innovative features	10%
GitHub Requirements	50%

Business Requirements;

NavigationDrawer:

:NavigationLayout (DanielAssign3)

1- Use styles and themes to create a nice look and feel of the screens.
2- Drawer must include the following options: please see screenshot below.
3- All options in the drawer must have appropriate icons and text. Use **different images** than the ones shown in the github repo.

4- **Select different background colors for the fragments**. Each fragment must have a different background color.

5- Implement Menu on actionbar (tool bar):

a. **Location**: always appear, icon of your choice, and title location, when clicked, display current longitude and latitude on a **snackbar**, with an action. Handle **runtime programmatically**.

Implement runtime permission to get the user location: See how to retrieve location from example below, and how **onRequestPermissionsResult** is implemented:

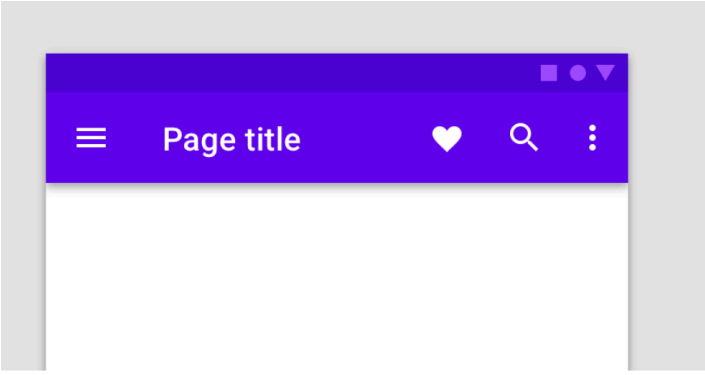
https://github.com/haki11/GPSLocation

When user **clicks on the action**, waiting **indefinitely**, display Google map passing the long and lat received from the location.
https://material.io/develop/android/components/snackbars

b. **Home:** appear if enough space, when click, display the App home screen, with an image, and title "Home".

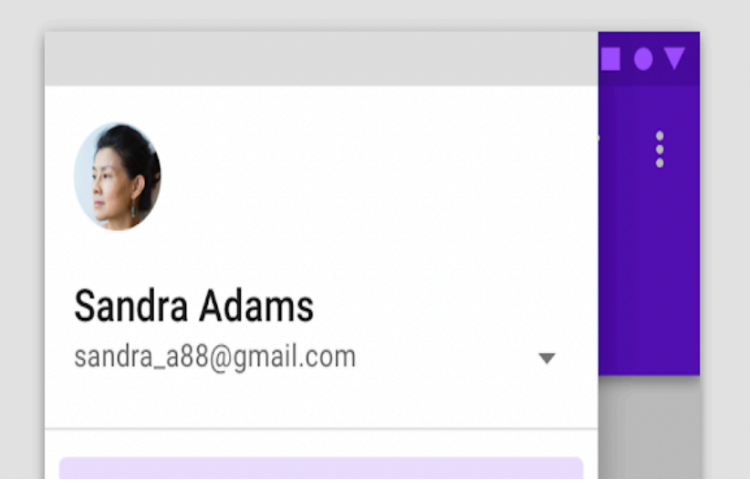
6- Implement the following fragments in the Navigation Drawer:

i. https://material.io/components/navigation-drawer
ii. Should show the three lines when not sliding to indicate the Drawers.
iii. User clicks on the 3 lines to have the drawer sliding
https://developer.android.com/guide/navigation/navigation-ui



top app bar.

7- Navigation Header should include the appropriate image and text, example below:



8- Names of Java Fragment classes and XML layout must be as requested below: (**4 Fragments / 4 Screens**)

9- Display a snackbar when menu sliding, with your name (team name) + Menu open.

10- Display a toast when menu closing, with your name (team name) + Menu closing.

11- Follow as requested below from left to right.

12- Home Fragment: Must be the **one selected by default and displayed when launch the app** and include the following requirements:

i. Class name FirstNameHome.
ii. Layout xml firstName_home
iii. If team, replace FirstName with teamName2.
iv. Background color: your choice.
v. **TextView** with full name, bold, blue, italic, and 25 sp, at top centered.
vi. **Three Radio buttons**:
    1. Random Number (selected by default)
    2. Random Text.
    3. Firebase.

vii. **Button**, with a label Options, when selected:

1. If Random Number selected, generate random number, between -50 and +50, and display on TextView below, replace previous data.
   https://stackoverflow.com/questions/5887709/getting-random-numbers-in-java

2. If Random Text selected, generate random String, check link below on how to generate random text

   https://www.baeldung.com/java-random-string#java8-alphabetic

   and display on TextView below, replace any previous data.

3. If Firebase selected, store the following info into **Firebase** DB.
   a. Last generated Random Number, initial 0.
   b. Last generated Random Text, initial "Hello World".
   c. Your Name.
   d. Your Student ID.
4. If Firebase selected, update TextView with "Firebase Data".

viii. **TextView:** with some initial text of your choice, set in strings.xml.

13- Download Fragment:

Must include the following requirements:

a. Class name LastNameDownload.
b. Layout xml lastname_download
c. If team, replace lastName with teamName22.

i. When user selects this option, display the following:
ii. Background color: your choice, different than above.
iii. ListView with 4 different **text** options (i.e. flower, car, planet and urban).
iv. When user selects one of the items in the ListView, download a picture from the **Internet**, based on the option selected from the ListView and display on the screen.
v. Highlight the item which is selected in the ListView, with a color of your choice.
vi. Display ProgressBar while the image is downloaded. Display the appropriate progress bar while the image gets downloaded.
vii. Simulate some delay when downloading, i.e. set delay for 5 secs.

14- Weather Fragment,

a. Class name nStudentIdWeather.
b. Layout xml nStudentId_weather
c. If team, replace nStudentId with nStudentIdOne_nStudentTwo_weatehr.

d. Register with the site https://openweathermap.org/ and get the API key
e. Information about the API:
https://openweathermap.org/current

When you get the key, wait for 30 mins before using it.

f. Background color: your choice. Different than above.
g. TextView: Display your student ID, i.e. N1343242.
h. Radio button with two options Celsius vs Fahrenheit. Default Celsius.
i. Spinner with 5 different cities (5 cities in 5 different countries):
j. Declare the cities as Array in strings.xml.
k. When user selects an item from the spinner, call the web service and pass the latitude and longitudes that correspond to that city. i.e. if user selects Toronto, pass: lat=43.777702&lon=-79.233238
l. https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={API key}
https://api.openweathermap.org/data/2.5/weather?lat=43.777702&lon=-79.233238&appid=XXXXXXX
m. Parse the data and display in TextViews; lon, lat, Country, humidity, name, and description.
n. Temperature (temp), convert from Kelvin to user preference as selected in the radio button: Celsius or Fahrenheit

2. Example response:


```
{"coord":{"lon":75,"lat":43},"weather":[{"id":804,"main":"Clouds","description":"overcast
clouds","icon":"04n"}],"base":"stations","main":{"temp":276.2,"feels_like":272.36,"temp_min":274.81,"temp_max":276.98,"pressure":1000,"humidity":92,"sea_level":1000,"grnd_level":978},"visibility":10000,"wind":{"speed":4.42,"deg":270,"gust":10.11},"clouds":{"all":100},"dt":1648270375,
"sys":{"type":2,"id":2022886,"country":"US","sunrise":1648291953,"sunset":1648336714},"timezone":-
14400,"id":5127233,"name":"Mohawk","cod":200}
```


14- Database Fragment:

i. Class name is DBScreen.
ii. Layout name: db_screen.
iii. Button with a label “Read from DB”.
iv. TextView: with initial value “Your full name”.
v. When user clicks on the button, read the data from the **Firebase** database and display on the TextView:
    a. If DB does not exist, or an error, display the error returned, and update the TextView and set the color to red, bold and italic.

b. If data retrieved from the DB, set the color of the TextView to black and bold, and update with the content from the DB.