Write the following phone app. You are expected to use material design concepts, at least as much as taught in class. Lastly, the implementation of the GUI is left for you, but it should be intuitive to use.

This is a two part assignment.   Program #5 will be the backend database and content provider piece. For program 4, you are provided with the content provider (as an apk to install). For this assignment you will write the front end UI for it.

Check/debit card book/expensive application:
You will be writing the front to a content provider (see below). You will be required to use material design for the UI.   Your app will display, add, update, and delete entries in the content provider.   Each entry will also have a category as well.

Content provider tables:
 Transaction table:
   _id: a number assigned by the content provider, but will represent the row in many things.
   Date: listed in any format, it's just a string in the database
   Type: a string, to represent a check number, ATM, credit (purchase), deposit
   Name: name of payee or name for the entry
   Amount: A dollar amount (as a real number)
   Category: an integer id number, where the name is listed in the category table (see below)
       There is also a row id number, assigned by the content provider.

Category table:
   _id: integer number assigned by the content provider, which will match the Category entry
               in the transaction table.
   Name: a string name.

Accounts table:
    _id: which is assigned by the content provider
   Name: string name which is the name of the account for the transactions table.
     Note 4730 students won't have to deal with this table, unless they want to, but it will also have to "used" in the program 4 as well.

The application will need to do the following:
Display entries in the transaction table already listed
Change entries already stored in the transaction table
Add new entries in the transaction table.
Delete entries in the transaction table
Add/update the category table.

Category information

Category names are stored separately so the user can select one when adding/changing an entry and so they don't have to retype it every time. Before a new category can be used, it must be added to the category table and then the row id is used in the transaction. The user should also be able to add/change/delete category names separately as well. A note, you can't delete a category while it is still listed in any transaction table.

For 5730 students: Complete the requirements of the program.
Two additional requirements:
  (1) The application is deal with multiple accounts (multiple transaction tables). So it will be able to open/view/add/change entries for a more than one account.
        a. A note, the table id 1 is created by default and is called checking. It will be listed in the accounts table by default
  (2) If the user wants to "move money" from one account to another account, it should be a category (default one, the user doesn't have to create it) and they only have to make a single entry and your application will add an entry into the other account.
  (3) Creating a second or more account entry. Add an entry to the accounts table, which will create a table in transaction table. The accounts table _id is used to access the associated transaction table via the uri.
        a. To delete a transaction table, you need to remove the entry from the accounts table. It will then delete the associated table.

Accessing the Content provider:
Uri listing:
content://edu.cs4730.prog4db/Category is for the category table of information
content://edu.cs4730.prog4db/Accounts is the account table
  The default first entry will be checking and is for to the first transactions table.
content://edu.cs4730.prog4db/Accounts/transactions/1 is the default checking table. This on created by default for cs4730 to use without the need to use the accounts table.
content://edu.cs4730.prog4db/Accounts/transactions/2 would be the second transaction table to be created, where 2 matches an entry in the accounts table.
As a note
content://edu.cs4730.prog4db/Accounts/transactions/1/1 would be the first entry in the checking table.
content://edu.cs4730.prog4db/Category/1 would be the first entry in the category table.

Also, loaders are provided in the content provider and you are expected to use them in your code.

TURN IN and GRADING:
Hard copy:
  1. A cover page with Name, program #3, cosc 4730 or 5730 depending on which class you enrolled in, a repo name (see github and below for you repo name).

Soft copy:
  1. Use this link to create your repo https://classroom.github.com/assignment-invitations/6f49d96acfc4c2c916e183674ff6b836

2. Upload the project to your repo
3. Create/Edit the readme.md file, add the following:
   o Course number 4730 or 5730
   o Name
   o how to run the program (this is likely very simple for program 1),
   o which phone/emulator to run on including special information like android version (ie v4.4) and screen size.
      ▪ Or if you are using the borrowed, phone, Nexus 5X.
4. Lastly ensure everything has uploaded to the github website and not just the local repo.

Code will be graded on correctness, comments, and coding style.