Homework 4: Reddit

For this assignment, you will build an app that fetches content from reddit and displays it to the user. The user can refresh the contents, see the details on an individual post, and change to a different subreddit. You will also implement a list of favorite posts for the user.

Up to this point your flipped classrooms and homeworks have been "programming in the small." What I mean by that expression is that the task you have to do is simple, and the tools at your disposal are sufficient for the task. This assignment (and I hope your project) are intended to expose you to "programming in the large," even though by absolute line count this homework is pretty small. But the point is that compromises are necessary: there might be multiple ways of doing something and no clear reasons to prefer one over the other. There is the occasional mysterious parameter which isn't terribly logical, but useful for the implementation.

Let me highlight one quick example before we get mired in homework specifics. The model for this homework is a `RedditPost`, which is a structure with 12 fields. Worse, some of these fields are useful when the `RedditPost` is a post to a subreddit, and other fields are useful when the "`RedditPost`" is actually information about a subreddit. The reddit API (`https://www.reddit.com/dev/api/`) specifies that a request to list subreddits returns what it calls a list of posts, which is identical to what is returned when you ask for the posts in a subreddit. Is that strictly logical? I would argue that it is not. A post and information about a subreddit are logically distinct entities. Even worse, some fields are used when the "post" is a subreddit post and different fields are used when the "post" is information about a subreddit. This is the world we live in and to make progress we must accept reality as it is.

You might say (as I did) that what we need is some sort of inheritance or interface. We have a generic something, `AbstractPost`, that has two subtypes (or interface implementations), `RedditPost` and `SubredditListing`. Well, I started down that road, and Donny, I entered a world of pain. Gson acts like a quiet and omnipotent god when you ask it to decode a class with all primitive types. It just does the right thing. But once you start asking it to create different types of objects, it turns into an angry and vengeful god hell-bent on smiting those who would dare to subtype.

On the other side of the coin, we have been training for this homework all semester. This is a moment to look back (or down?) and see how far you have come. We have coded a lot of these behaviors in isolation, now we get to see them in a single app.

The code as given to you does not build properly. You need to add some code before it will even compile.

Note: This homework requires more programming and is a bit more conceptually difficult than your previous homeworks. It will be worth more points. The code I gave you has about 400 lines cut from my Kotlin implementation and 64 lines of XML.

Note that there is STILL no collaboration for homeworks. This homework is challenging, but every individual should be able to do it for themselves (and you will learn a lot).

A note on **how to get json from reddit**. Looking at the API documentation, it is incredibly poorly described how to get json returned from your reddit API request. I actually found the answer in a reddit post, which is some form of irony, or irony in reverse. Here is an example for getting json that describes a particular subreddit (which isn't needed in your app, its just an example). Look at the URL carefully.

`https://www.reddit.com/r/aww/about.json`

If you are not seeing posts and getting errors from your network requests, you might be requesting HTML, not JSON. Log the URLs and paste them into your browser. If you see HTML, your URL request is not correct. You should see a mess of JSON.

By the way, the reddit documentation claims that appending `raw_json=1` to a request will not only return json, it will not embed those annoying HTML encoded characters. Well, if you can get that to work I'm impressed, because it always fails for me.

Let's start with specifying behaviors. You might get all this information from watching the video demo, but I wanted to give it to you in easy-to-digest written form.

- **Startup.** When your app starts, it should fetch the **hot** 100 items in the `aww` subreddit, displaying them in a list.

  - The title bar has three parts, the subreddit name (also known as a title area), the search area, and a solid heart. Clicking the subreddit name brings up a list of subreddits to choose from. Clicking the search area lets you type in a search term. Clicking the heart brings you to the favorites list.
  - The favorites heart disappears from the title bar whenever you enter: the favorites list, the subreddit list, and OnePost (see below). You can only get to the favorites list from HomeFragment.
  - Display an empty heart if the post is not a favorite, and turn it into a solid heart if the user clicks the empty heart to add the post to their favorite list. Each click toggles the state, and you should see the icon change.
  - When you refresh the posts for a subreddit, that should not change your favorites.
  - When you type in the search bar, the listing is limited to search matches, the first instance of which (in each post) is highlighted. Actually the first instance in the title and the selfText (see below), if they both exist.
  - In the title bar, the subreddit name should be `r/aww` initially, and then r slash subreddit name, depending on your subreddit. Make sure the title bar is correct when you return to the list of posts from favorites or from the subreddit list, or OnePost.

- **OnePost.** When the user clicks on the title of a post (or the selfText or the image), they should see a new fragment that just displays the single post (with the source subreddit above in blue). I will call this the OnePost view, and your fragment should be named OnePostFragment.kt. You will write it from scratch because you are experienced Android software engineers.

  - You can get back from the OnePost view to the list of posts using the phone's back button and by the "up" arrow in the upper left in the title bar.
  - When you go back from the OnePost view, do NOT refresh the listing (so do NOT make a network request, though Glide image loads are fine). If you let navigation do its thing, this happens naturally.
  - The OnePost view should display the text (`selfText`) in a scrollable environment, because it can be long. See notes about the layout below.
  - If there is no selfText (if it is null or the empty string), then set the visibility of the `onePostSelfText` to GONE. This keeps the layout tight. But it is a detail.
  - The title in the title bar is "One Post".
  - The favorites heart disappears from the title bar. See ViewModel.hideActionBarFavorites.
  - Be sure to test cases with long selfText (e.g., r/AITA), images.
  - Don't stretch images (i.e., force their horizontal dimension to match_parent).

- **Favorites list.** When the user clicks the heart in the action bar, they are taken to the favorites list.

  – Change the title to "Favorites".

  – Hide the favorites icon in the title bar.

  – Favorites do NOT have to last between sessions, you never need to write them to persistent storage. But they do persist when you refetch over the network. They also stay the same if you change subreddits. Note how `equals` is defined for a RedditPost. That will make implementing favorites easier.

  – The user can click the heart of an item in the favorites list and then that item should disappear from the list. This behavior should happen even if the user is in the middle of searching the favorites list.

  – When you go back to the PostRow view (HomeFragment), do not refetch the list from the network, but do redraw the list because the favorite status of posts can change in the favorites list view. This should happen naturally with navigation. You should be able to return to the PostRow view using the phone's back button, or using the "Up" left arrow.

- **Subreddit list.** When you click the current subreddit, you get a list of subreddits. These should be the **popular** subreddits
  https://www.reddit.com/dev/api/#GET_subreddits_popular.
  Get 100 of them.

  – The title should say, "Subreddits".

  – Pull down (swipe) to refresh is disabled here, there is no need to refresh.

  – This list will look a bit different from the list of posts, but it is still displayed as a recycler view of RedditPosts. It will have its own adapter (and its own fragment/view).

  – When you exit the subreddit list with the phone's back button or the up arrow, do NOT refresh your listing from the network (though Glide image loads are fine). But please restore the current subreddit title. Navigation makes this natural.

  – When the user clicks a heading or the pic, this list disappears, we are back to the PostRow view, the title is updated with the new subreddit name. We fetch the new subreddit contents from the network, and it often takes a beat for the new contents to arrive. That delay is fine.

- **Search.** In every view/fragment, the search box provides incremental, ignoreCase filtering for both the title and the selfText. The first match found is colored cyan. If there is a match in the title and in the selfText, then there are two cyan highlights. Search is incremental, it changes every time you type (or remove) characters.

  – In the PostRow, Subreddits, and Favorites view, only the matching entries in the list stay in the list. The non-matching entries are filtered out. There is no filtering in OnePost.

  – As you delete characters, the list expands, returning to its original state when the search box is empty. In this sense search is non-destructive. You always "have" the entire list that you fetched over the network.

  – When the search string is empty, you must dismiss the soft keyboard.

  – Searching does not create new network requests. Reddit provides a search API that we are not using.

- For a regular post search text in `title` and `selfText`.
- For a subreddit listing search test in `displayName` and `publicDescription`.

Let's start by describing parts of files given to you that you do not need to change.

- **MainActivity.kt** This file has the `hideKeyboard` method, which you will call for part of the search functionality. We set up the custom action bar so you don't have to. The default navigation location is HomeFragment.

  Check out `safeNavigate`. It is a Kotlin extension function, meaning you can call it on a navController even though we wrote it. If you have an action (say from the HomeFragment to the Subreddits fragment), and you navigate to that action, but you are already in the Subreddits fragment, you will crash. But if you safeNavigate, then nothing happens. Our action bar title is controlled by main activity. We want to click the title to get to the subreddits list. But if we click the title in the subreddits list (or the favorites list or OnePost), then we don't want anything to happen. safeNavigate does that.

  Note, we don't really need safeNavigate for favorites, since we remove the action bar favorites icon in all views except PostRow. But we can use it!

  `initDebug` is for us, not you.

- **glide/AppGlideModule.kt** You don't have to change this file.

  This file lets you fetch images via glide. Check out `applyOptions` where you can set the log level if you want more output from Glide. I find Glide very chatty, so I want less output by default.

  Also note that Glide fails to load many images, especially when given a video. In general, reddit returns some sketchy URLs for images and it isn't "wrong" for Glide to fail to load them. We call glide with a regular URL and a backup thumbnail precisely because the main URL (`imageURL`) is often busted (it can be a video or a web page, neither of which Glide handles).

  When we are fetching the subreddit icon, you can use the icon as both the `urlString` and the `thumbnailURL`.

  Note that we find the size of the user's display and pass that information to Glide, so it can more efficiently manage memory. Large pictures take up a lot of memory and storage and we don't need images that have greater resolution than our display.

  Also note that I have to "adjust" the HTML escape characters that reddit puts in these URLs by calling `fromHtml`.

  Don't change this file because we are going to replace it when we test your code.

- **navigation/nav_graph.xml** Please provide fragments using these IDs: "@+id/favorites" and "@+id/onePostFragment". Make an action from Favorites to OnePost. onePostFragment takes an argument called "post". That argument is of type "edu.cs371m.reddit.api.RedditPost", which is a "Custom Serializable" according to the attributes editor.

Let's talk about the networking code. Much of this is given to you, but just like the flipped classroom, you need to fill in the key parts. You should start here because it will be tough to do anything in your project without data from the network. All of these steps should be familiar from flipped classroom activities.

- **api/RedditApi.kt** Just like for CatNet and TriviaGame, you need to tell Retrofit how to fetch posts for a particular subreddit and how to fetch the list of subreddits. For posts, you can hardcode the limit of 100 posts, but you do have to pass a parameter that identifies the current subreddit. I talk about it a bit in the code comments, but don't be afraid to google around a bit.

  Note that pinned posts don't count against your limit, so if you ask for 5 items and there are 2 pinned posts, you will get 7. Don't worry about it. Ask for 100, deal with what you get.

  I have given you the types you need to "decode" Reddit's responses. Go and read this documentation
  `https://www.reddit.com/dev/api/#listings`
  which is what reddit returns. Also read
  `https://www.reddit.com/dev/api/#GET_hot`.

- **api/RedditPost.kt** This is your model. I talked about it in the intro. It has a bunch of fields, some of which you won't use, some of which are only relevant for the subreddit list, etc.

  We provide most of the search/text highlighting functionality, but leave out the implementation of `searchFor`. There are helper functions in the file. Highlight the first match in each string you search.

- **api/RedditPostRepository.kt** This is pretty simple, it provides the bridge between your application code and the web API. We include debugging code that is important to us and you should not change. You only need to fill in the `// XXX Write me` lines in this file.

  The `unpackPosts` function takes a listing (what is returned from the Reddit web service) and does some work to turn it into a `List<RedditPost>`.

These files you have to write all or most of. Figure out the quickest way to start, were you connect to reddit and you can display something. Build it up from there.

I have some `// XXX` in the code, but there are no guarantees for these files.

- **AndroidManifest.xml** This needs you to add a permission.

- **ui/OnePostFragment.kt** This fragment a single post. Do the normal things we usually do in a fragment, like get a view model. You should also gets the Args object set up by Navigation.

  We give you nothing. Create the file (called OnePostFragment.kt) and write it!

  Here are a few subtleties. If the reddit post's selfText is empty or null, make it `View.GONE`. It helps the layout.

  You can find the current subreddit by `observeSubreddit`.

  You can call `observeSearch` to implement highlighting in this view. This view does not filter.

- **fragment_one_post.xml** We do not provide this layout file. You must create it. The layout file isn't totally trivial, but it is not complicated. Use these settings.
  TextView ``@+id/onePostSubreddit`` 14sp
  TextView ``@+id/onePostTitle`` 16sp
  TextView (scrollable, so nested in a ScrollView) ``@+id/onePostSelfText`` 14sp
  ImageView ``@+id/onePostImage`` layout_gravity="center_horizontal"

- **MainActivity.kt** You need to set up onClickListeners on the title bar. These will remain in effect even if you swap in another fragment, which is why we use safeNavigate and why we hide the favorites icon.

  No menu.

- **content_main.xml** contains the navigation controller in `main_frame`.

- **ui/HomeFragment.kt** Home base for your app. I left in a bunch of auxiliary function names that I used. You don't need to use them, but they are evocative of the kind of functionality you have to implement.

  Your HomeFragment will interact with your viewmodel, and I'm leaving that up to you. Conceptually, the interaction with the viewmodel and the views is the most difficult. Think about what livedata your viewmodel should manage and who should be observing it.

  Just like in the flipped classroom, you need to manage the swiperefreshlayout.

- **fragment_rv.xml** is the layout for several views, HomeFragment, Favorites, and Subreddits. All of these views are simple recycler views with a swipe refresh layout.

  Each of these views has their own adapter, but it is kind of cool that they all share a layout.

- **ui/PostRowAdapter.kt** A good old fashioned recycler list adapter. I use the **ListAdapter** base class, which exports the `submitList` function, which is easy to use when you fetch across the network. I left in the RedditDiff. You need to implement the standard adapter stuff, namely binding and a view holder.

  The adapter takes in a view model. We have seen that before. Those are convenient in case one needs to manipulate a favorites list or something.

  The adapter also takes a function called navigateToOnePost that it should call if the user clicks the title, selfText or image of a post. It should take you to the OnePost view and pass it the RedditPost object.

  You can call `bindingAdapterPosition` to get the position for an onclick listener.

  One detail: in the XML for selfText, we specify elipsize, which will automatically truncate text that is too long and add an ellipsis.

  If selfText is null or empty, set the visibility to View.GONE. Otherwise it leaves too much space.

  Load the `imageURL` and as a backup, the smaller `thumbnailURL` using glide (see below).

  row_post.xml has the layout.

- **ui/Favorites.kt** This fragment should use a PostRowAdapter. I give you some hints in this file that should be useful for Subreddits and OnePostFragment.

- **ui/subreddits/Subreddits.kt** This fragment is almost identical to Favorites.kt. It is more trouble than it is worth to merge them and use subtyping.

- **ui/subreddits/SubredditListAdapter.kt** Use the iconURL for the `subRowPic`, the `displayName` for the `subRowHeading`, and the `publicDescription` for the `subRowDetails`.

  Pass in a view model and a navController.

  Exit the current fragment with navController.popBackStack() (navController.navigateUp() also works).

  row_subreddit.xml has the layout.

- **ui/MainViewModel.kt** Well, I saved the best for last. As you might be picking up, the viewmodels simplify your life in the fragment, but they are a bit tricky to design and write.

  Remember how to fetch over the network. Use `viewModelScope` and see the comments in the code.

  We have seen `MediatorLiveData` in FilterList. It is your friend. Read the documentation.

  We change the title in the action bar via the view model. The MainActivity better observe the title and do something in `initTitleObservers`.

  The favorites list is pretty straightforward though make sure you reassign to the value to trigger any live data that is observing.

  I'd like to talk about search, because implementing search is a bit tricky. I would advise getting pretty far into your implementation before adding it.

  `netSubreddits` is some MutableLiveData that holds the entire list of subreddits (where each subreddit is described by a redditpost, even though it isn't a post) returned by the server. We only fetch this list once.

  Having the full list is useful because as we search we will filter the list, but as we reduce the search string, we want to restore items. We keep netSubreddits as the full list returned by the server.

  I create a MediatorLiveData called `searchSubreddits` that has netSubreddits as one source and `searchTerm` as another. That way, if either changes, the list of filtered subreddits change. Ah, the magic of MediatorLiveData.

  `netPosts` is like `netSubreddits`, it is the list of posts returned over the network. Note that it already is a MediatorLiveData, with the current subreddit as a source. Whenever we change subreddits, we fetch new netPosts.

  We do the same trick for searching with netPosts that we do with subreddits. `searchPosts` is a MediatorLiveData that I create with sources `netPost` and `searchTerm`. How do we filter these lists? Check out the FilterList demo!

This assignment should be submitted through github according to standard proceedures. Please fill in the README.

Possible plan. This plan is really just a suggestion. If you have already blown past some of these tasks, good for you.

1. Start with retrieving 100 posts from aww.

2. Once you have a listing, do `OnePost`. I hope that will be "easy" to at least display something. Look at Favorites.kt for some ideas.

3. Then tackle subreddit listings and choosing a new subreddit.

Only after you have completed these three tasks should you even start to think about favorites, search, and details like proper handling of the action bar. I think I would do favorites before search.

I am a big believer in trying to get something in your code working as soon as possible. Then build from that piece of functionality to the next. Don't get distracted by more complex tasks that wait for you in the future. Once you get experience with the basic functionality, it will be easier to understand how to implement the more complex functionality.