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 favorites list for the user.

Up to this point your flipped classrooms and homeworks have been “programming 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 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. Is that strictly logical? I would argue that it is not. But it is the world we live in and we must code to that world.

You might say (as I did) that what we need is some sort of inheritance or interface. We have a generic something, RedditPost, that has two subtypes (or interface implementers), ActuallyAPost and Subreddit. 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 a winy, needy baby. And surely we can all agree on how distasteful babies are. Right?

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.

Yes, 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. My ui directory has 613 lines of code, yours has 202.

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 documented how to get json returned from your reddit api request. Here is an example for getting json that describes a particular subreddit. You never need to make this particular call in your app, but if you look at the example, it should make it clear how to get a json response from reddit when you do need it (getting posts and getting the list of subreddits). https://www.reddit.com/r/aww/about.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.
- **HomeFragment.** When your app starts, it should fetch the **hot** 100 items in the **aww** subreddit, displaying them in a list.
  - `row_post.xml` has the layout. The FrameLayout that contains both an ImageView and a TextView is a little wacky. Some posts have a big image, some have text. You just fill both in and don’t worry about it.
  - You need to fill in the score and the comment count.
  - Load the `imageUrl` and as a backup, the smaller `thumbnailURL` using glide (see below).
  - 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.
  - The title bar has three parts, the subreddit name, 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.
  - When you are in the favorites list or the subreddit list, clicking the favorite icon or the titleBar title should do nothing.

- **OnePost.** When the user clicks on the title, they should see a new activity that just displays the single post. I will call this the OnePost view, and your activity should be named OnePost.
  - You can get back from the OnePost view by the system back button and by the “back” arrow in the upper left.
  - When you go back from the OnePost view, do NOT refresh the listing.
  - The listing view should display the text (`selfText`) in a scrollable environment, because it can be long
  - If the post’s title is less than or equal to 30 characters, then make it the title to the OnePost view, otherwise take the first 30 characters and add “...” for the title.

- **Favorite’s list.**
  - Change the title to “Favorites”.
  - 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. You might find the key field in a RedditPost useful to keep track of your favorites.
  - The user can click the heart of an item in the favorites list and that item is removed from the list.
  - When you go back to the PostRow view using the system back button, do not refetch the list, but do update the list because the favorite status of posts can change in the favorites list view.
• **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](https://www.reddit.com/dev/api/#GET_subreddits_popular).

  - This list will look a bit different from the list of posts, but it is still a recyclerview of RedditPosts. It will have its own adapter (and its own fragment/view).
  - The title should say, “Pick.”
  - If you exit the subreddit list with the system back button, do NOT refresh your listing and restore the current subreddit title.
  - Use the iconURL for the subRowPic, the displayName for the subRowHeading, and the publicDescription for the subRowDetails.
  - When the user clicks a heading, this list disappears, we are back to the PostRow view, the title is updated with the new subreddit name and we fetch the new subreddit contents.

• **Search.** In both the PostRow view of a subreddit, the favorite’s list, and in the subreddit listing view, the search box provides incremental, ignoreCase filtering for the title/subtitle text.

  - As you type characters, the list is restricted to the entries with text that matches. You don’t need to highlight the matching text.
  - 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.
  - When the search string is empty, you must dismiss the soft keyboard.
  - Search works in all lists.
  - Searching does not create new network requests. Reddit provides a search API that we are not using.

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. You have done all of these steps in class and in the flipped classroom.

• **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 shows this syntax on piazza, but you might need to google around a bit.

  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](https://www.reddit.com/dev/api/#listings), which is what reddit returns. Also read [https://www.reddit.com/dev/api/#GET_hot](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 the setSearchText prototype, but you need to figure out how to use it and why. One hint is that you use it to help implement 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.

Now that I look at it, there is an argument to be made to make Favorites and Subreddits activities. But they are fragments, so let’s go with that for this assignment.

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

- **AndroidManifest.xml** This does not have the right permissions and it does not list all activities. We add two permissions for Glide. Read the documentation for `android:parentActivityName`.

- **ui/OnePost.kt** This is an activity that shows a single post. Specification is above and is pretty complete. The one wrinkle is getting the back button to work. You might need Mr. Google and stack overflow to help with that one.

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

  Even better, we don’t give you the layout file. The layout file isn’t totally trivial, but it doesn’t have any FrameLayouts or anything like that. Just print the title, a scrollable section for selfText, and an ImageView for an image, if there is one.

- **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.

  Bear in mind that the callbacks you set on the title bar will remain in effect even if you swap in another fragment.

  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.

  I will say something about search because the way I do search uses livedata in a way you haven’t seen. I don’t think you have to do it this way, but it makes it clean. The `postRowAdapter` supports `submitList`, and you can call `submitList` both when you fetch new rows from the network, and also when your search term changes and you need to filter the output. But remember, the filtering has to be non-destructive, so that when the user erases their search string, all of the posts come back.

  Note that `observeSearchPosts` takes an input parameter. We haven’t seen that before. Instead of storing every livedata object as a member of our viewmodel, we can just create them and return them. They can be very useful, especially `MediatorLiveData` objects.

  You can implement the search filtering functionality in the viewmodel, but when I tried that, I had to have a bunch of duplicate code for searching the postrows and for the subreddit list. There will be some duplicate (or near-duplicate code), but I think it is better to express the filtering as you call the observe methods. That is all I’ll say for now. Please think about it. I’m sure I’ll give more hints, but let’s all try to allow each student to have their own ah-ha moment for this one, as opposed to just saying what you do.

  fragment_home.xml is the layout, which should be obvious.

  Your file has 67 lines, mine has 158 lines.
• **ui/PostRowListAdapter.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. We didn’t go over this version of a listadapter in class, but there is an example in your ListViewFrag project.

All of what you need to implement in his file is the standard adapter stuff, namely binding and a view holder.

The one oddity is the boolean being passed in (**unfavoriteIsRemove**). This PostRow adapter is used for two different lists: the list of posts to a subreddit, and the favorites list. When looking at the posts to a subreddit, the user can change the favorited state of a post by clicking the heart. That changes how a single item looks. But in the favorites list view, if we unfavorite an item it disappears. We need to call different adapter functions when an item changes and when an item is removed. Hint: when the item is removed call `notifyDataSetChanged`.

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

Yours 38, mine 109.

• **ui/Favorites.kt** This fragment can use a PostRowAdapter. It is pretty simple, but has to support search. You can figure out which layout to use.

Yours 23, mine 51.

• **ui/subreddits/Subreddits.kt** This fragment is almost identical to Favorites.kt. They didn’t start out quite so similar, but I might merge them the next time I do this lab. You can figure out which layout to use.

• **ui/subreddits/SubredditListAdapter.kt** I think you can handle this one. The only wrinkle is the `subredditChosen` callback. I include that so I know to exit the subreddit fragment when the subreddit is chosen.

`row_post.xml` has the layout. The FrameLayout that contains both an ImageView and a TextView is a little wacky. Some posts have a big image, some have text. You just fill both in and don’t worry about it.

• **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`.

As we have seen, `MediatorLiveData` is your friend. You can use `Transform` or a few other tricks, but `MediatorLiveData` is sufficient.

Also, recall that our favorites list persists across nework refreshes, so be sure to test that and you might want to use the key field in a RedditPost.

Finally, you might wonder why I have a `doOnePost` function in my ViewModel companion object. This is pure convenience. When it comes time to launch a OnePost activity, I’m in some code that makes it difficult. Rather can pass in callbacks, I just call MainViewModel.doOnePost. Is it strictly logical? Probably not, but it makes the code easier to write.

Yours 32, mine 123.

Now we describe code given to you that you do not need to change.
• **MainActivity.kt** This file has the `hideKeyboard` method, which you will need in the search functionality. We set up the custom action bar so you don’t have to. It also starts the HomeFragment and attaches it to the layout, without adding to the backstack.

The one somewhat mysterious bit of functionality is the calling of `HomeFragment.myRestore` when the back stack has zero entries. You can think about why you might want that.

• **glide/AppGlideModule.kt** 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 again 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.

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