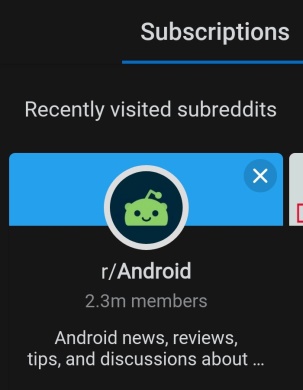
**Lab #2**

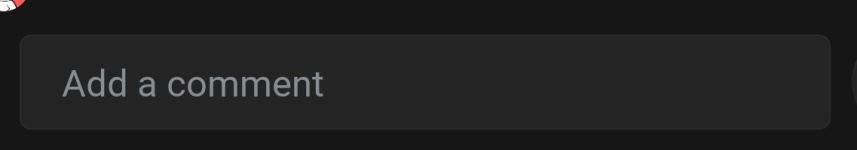
For this lab I will be talking about the **Reddit App**. One task I perform:

* **Fetching “hot posts”** for a topic of interest like r/Android for example.
* **Adding a comment** to a post
* **Making an account**

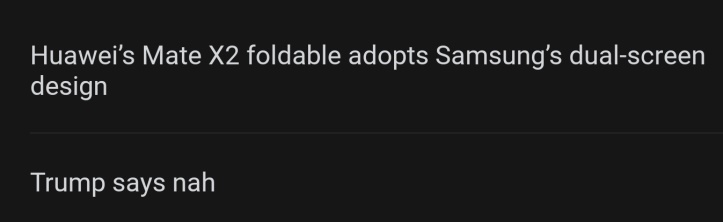
Non-technical:



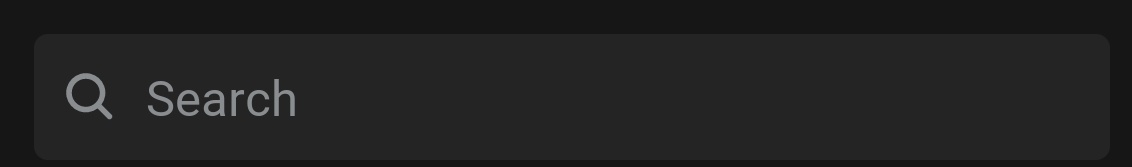
*List of Subscriptions*



*Add a comment option*



*Adding a comment*

****

*Search bar option*

****

*Subscription’s icon*

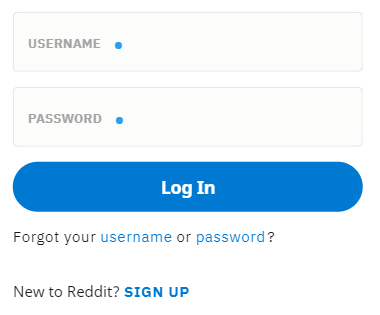
There are basically **3 ways** to fetch hot posts for r/Android. **First** is to search for the Android thread and selecting it from the results then it will load up r/Android. The default feed is ordered by “hot posts” i.e the most popular posts. **Second** is if you have visited the thread before, you can check your subscriptions to see the recent threads you visited and select r/Android from there and once again the default feed is ordered by “hot posts”. And the **third** way is to just scroll through the trending feed which also consists of your subscriptions/recent threads upon loading the app.

To add a comment, you perform whats above in addition to selecting a post and choose add a comment. You type your comment and click post.

For the technical explanations, I will be using the concepts of JSP/Servlets.

Technical:

* With JSP you can compose a form which associates with a servlet class that has logic. Each class usually performs one function displayed from a UI with an icon like a check box or clickable option. So in this case, say we have 2 forms, one for a search bar and another for subscriptions.
* Inside the form we can write some inputs to accept like read a string, say, “Android” and send the String as a parameter to our servlet class to perform some actions like query threads titled with “Android”. It will redirect or dispatch to another servlet class which can be associated with another JSP/HTML so it will load a new page in this case say with our search results. After selecting r/Android, it will again redirect or dispatch to another JSP/HTML which will query up “hot posts” which I guess can be done via a “dofilter” method of the servlet class. I mean its not very powerful. It’s probably a lot better to set up a get request in a controller and JPA repo via spring framework.
* Adding a comment is pretty similar process to the previous JSP form, but the input will accept more characters and instead of fetching it will be posting i.e sending data/comment to the server and persisting it. Of course this is granted you made an account.



*Registration page*

***What can go wrong?***

* So technically it’s a terrible idea to actually use JSP/HTML to make accounts and web apps in general. Thus far, everything described didn’t assume a domain and encryption. The big problem with JSP is if a user were to hit the back button it will simply redirect/dispatch them back to the previous page. In this case, the page before the user logged out (possibly user profile) can enable another user to simply hit the back button and “hack” the previously logged user. JSP does have methods to redirect a new user after log out to the login page however this would require developers to add a redirect for every page that is domain specific. So I would imagine developers usually opt for a solution with a guaranteed SLA and security standard such as Microsoft Azure to secure the data that passes through reddit. Reddit I guess is ultimately one app loosely coupled with microservices inside some form of public cloud separated by defined security/domain boundaries using something like an application gateway within a platform like Azure.
* So in this case, when a user makes an account, their inputs will be associated with a subscription under a “user” active directory which will have access to specific resource groups composed of resources normally accessible to all users. There maybe a few resources that is not available till the user accumulates some karma/internet points.
* One important thing to note is account creation will enable reddit to queue notifications to the user’s email’s inbox. So as the user consumes services on reddit. Actions such as creating a post, receiving a PM or receiving a reply to a comment will release a notification/message from a queue to consumer/subscribers. In this case it will be performing message queue style transactions (As opposed to pub/sub or fanout) since it’s just one consumer/user being targeted.

So all in all, for technical I wanted to describe what the fundamental functionalities are like from the back end from my perspective given my experience which is incredibly miniscule. I hope this was articulate and I can improve my technical writing and perspective moving forward