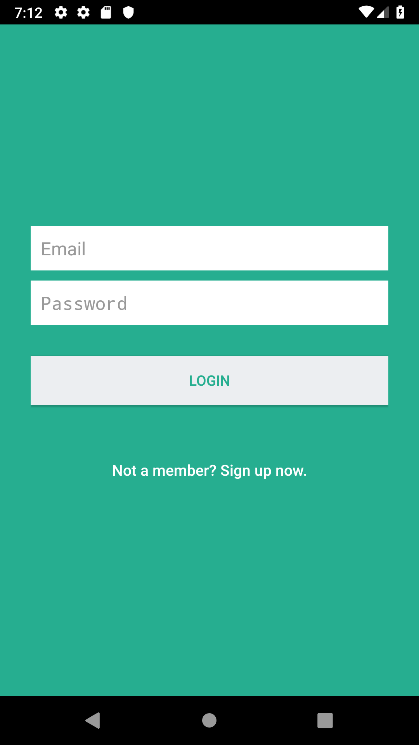
ASSIGNMENT 5

180919

* **Try to build, understand and run the login app**

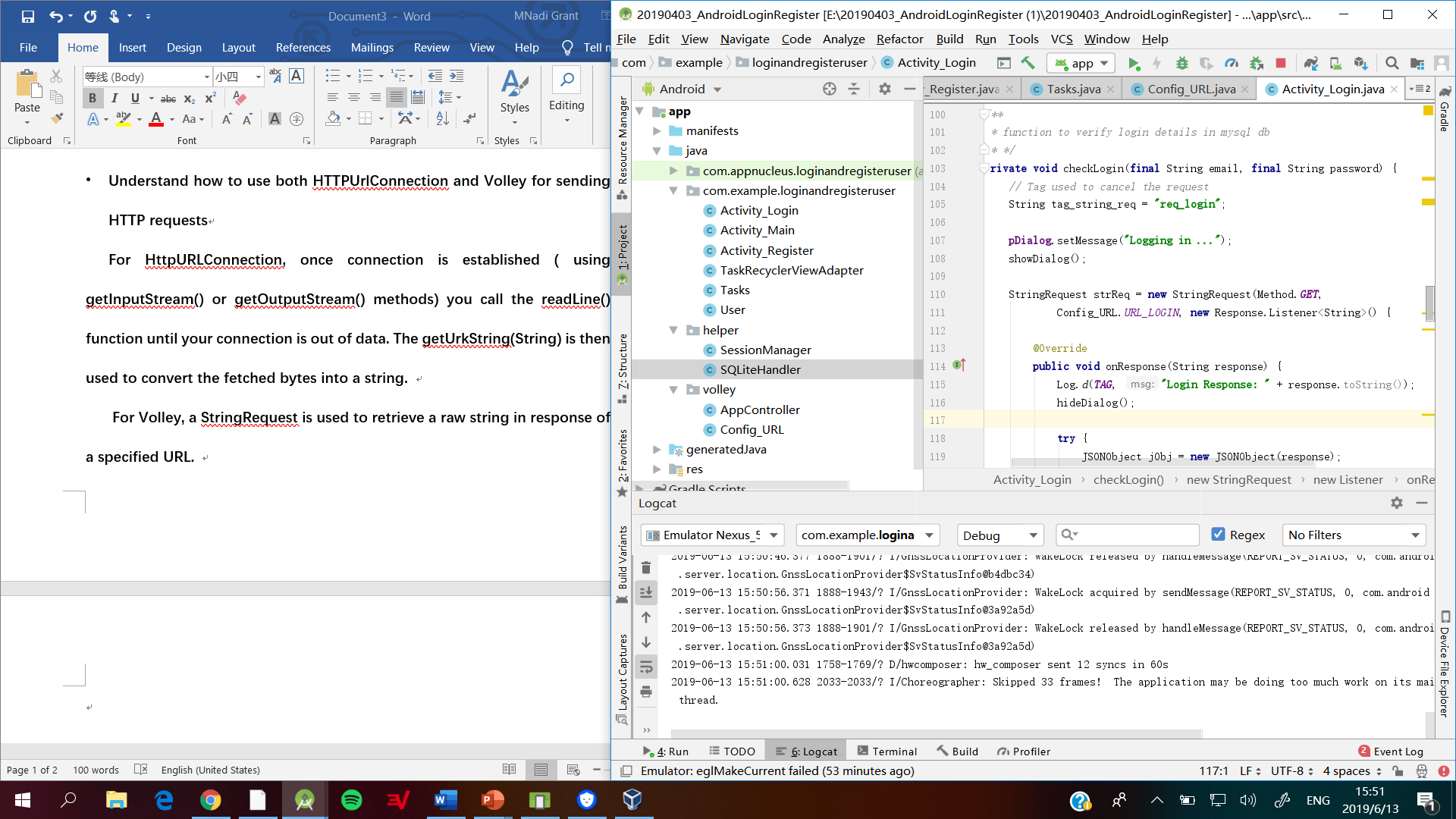
The login application uses Volley and GSON from google’s Java Library to handle the functions of http connection and Java Object conversions.

****

* **Understand how to use both HTTPUrlConnection and Volley for sending HTTP requests**

For HttpURLConnection, once connection is established ( using getInputStream() or getOutputStream() methods) you call the readLine() function until your connection is out of data. The getUrkString(String) is then used to convert the fetched bytes into a string.

For Volley, a StringRequest is used to retrieve a raw string in response of a specified URL. A JsonRequest (JsonObjectRequest or JsonArrayRequest) is used to retrieve a JSON object or array is response of a specified URL.

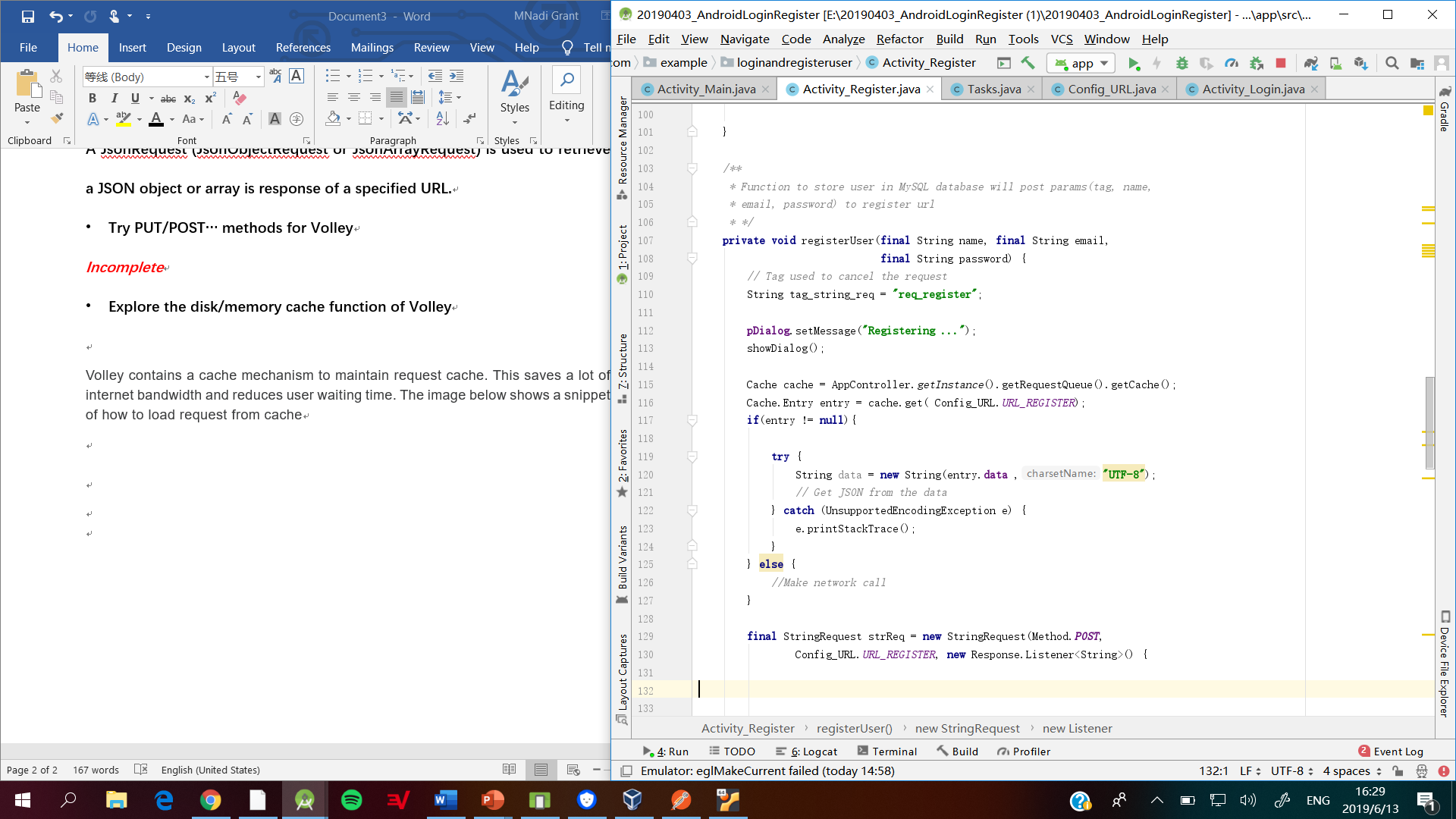


* **Try PUT/POST… methods for Volley**

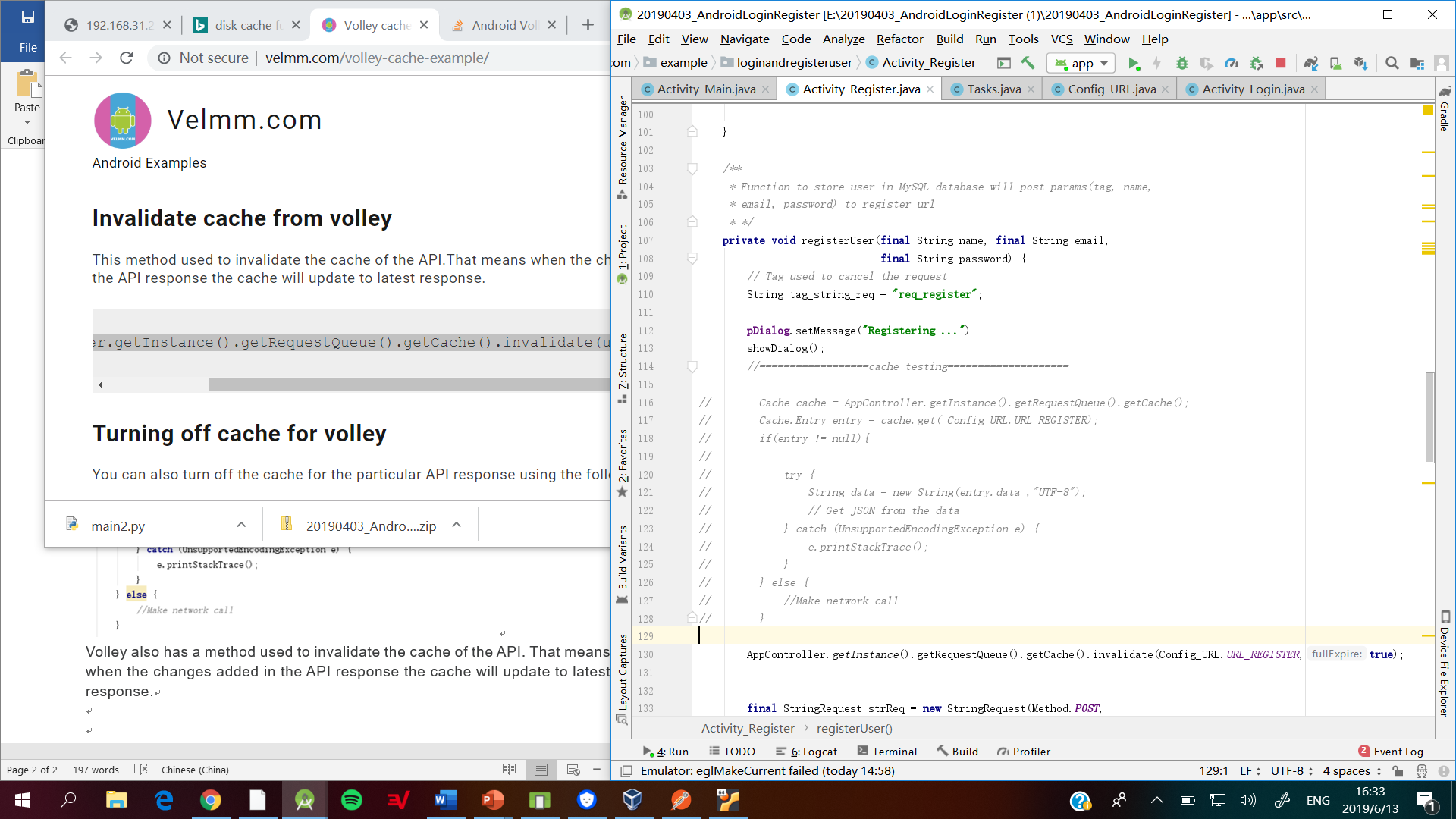
***Incomplete***

* **Explore the disk/memory cache function of Volley**

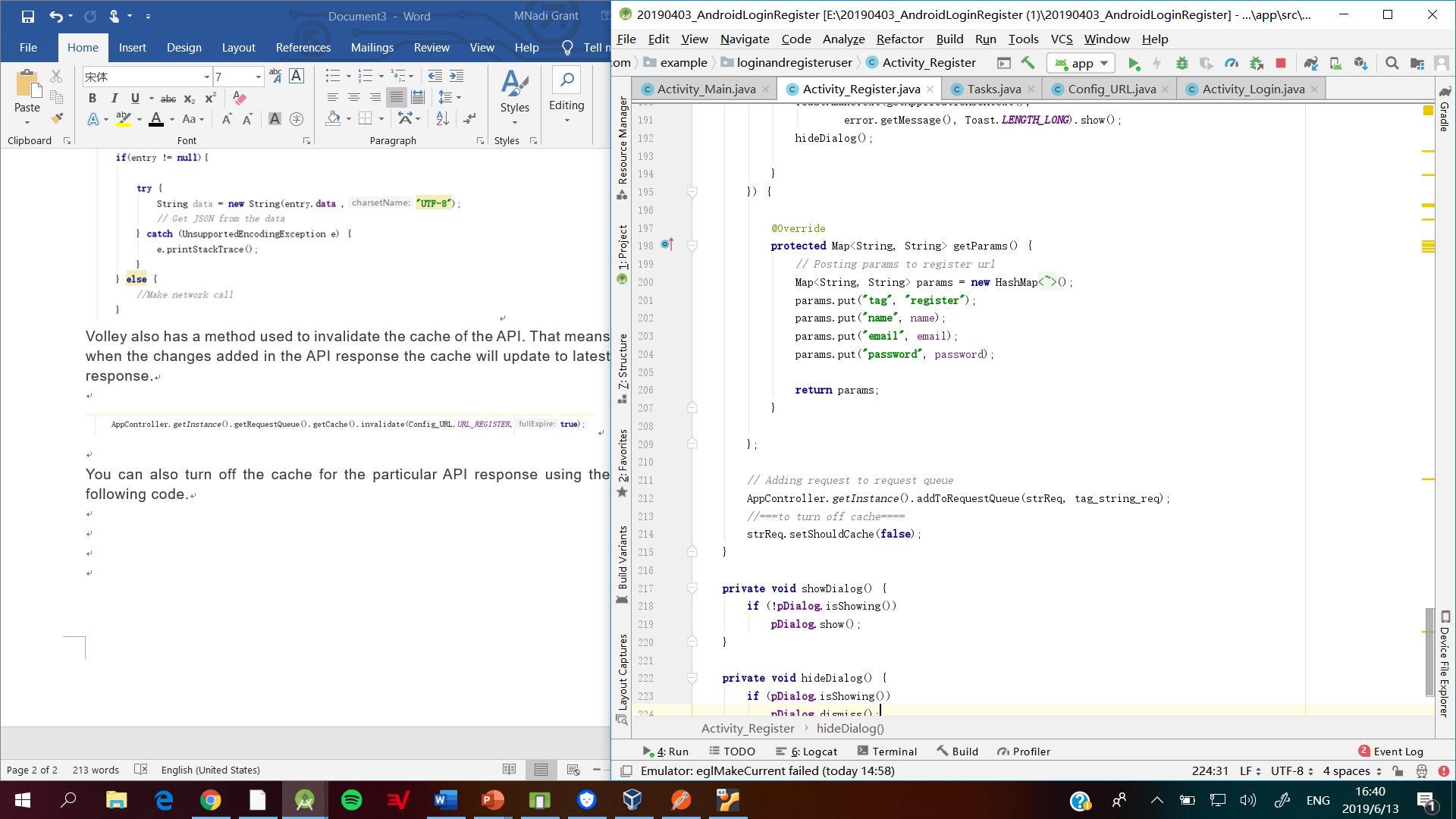
Volley contains a cache mechanism to maintain request cache. This saves a lot of internet bandwidth and reduces user waiting time. The image below shows a snippet of how to load request from cache.



Volley also has a method used to invalidate the cache of the API. That means when the changes added in the API response the cache will update to latest response.

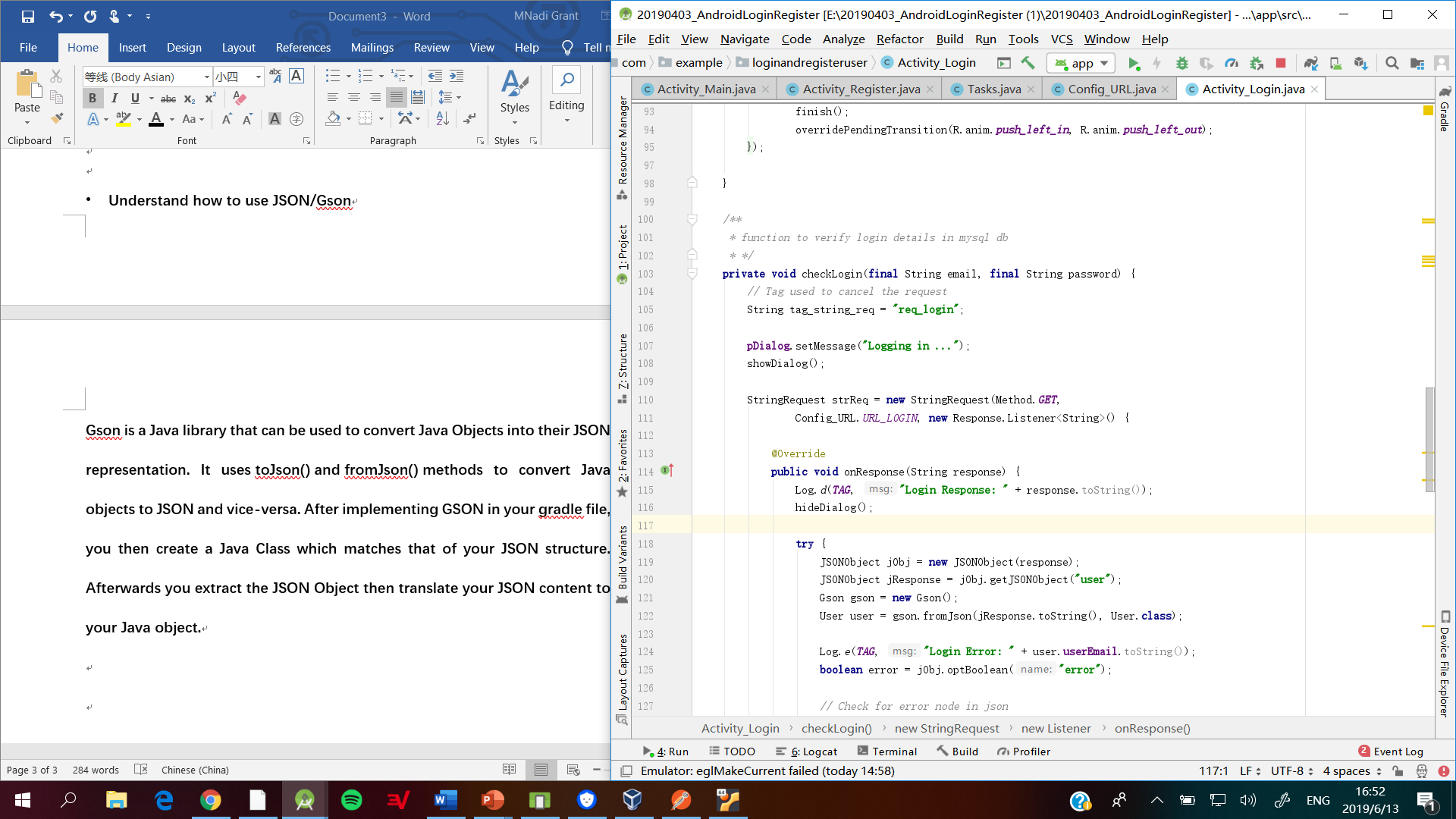


**You can also turn off the cache for the particular API response using the following code.**

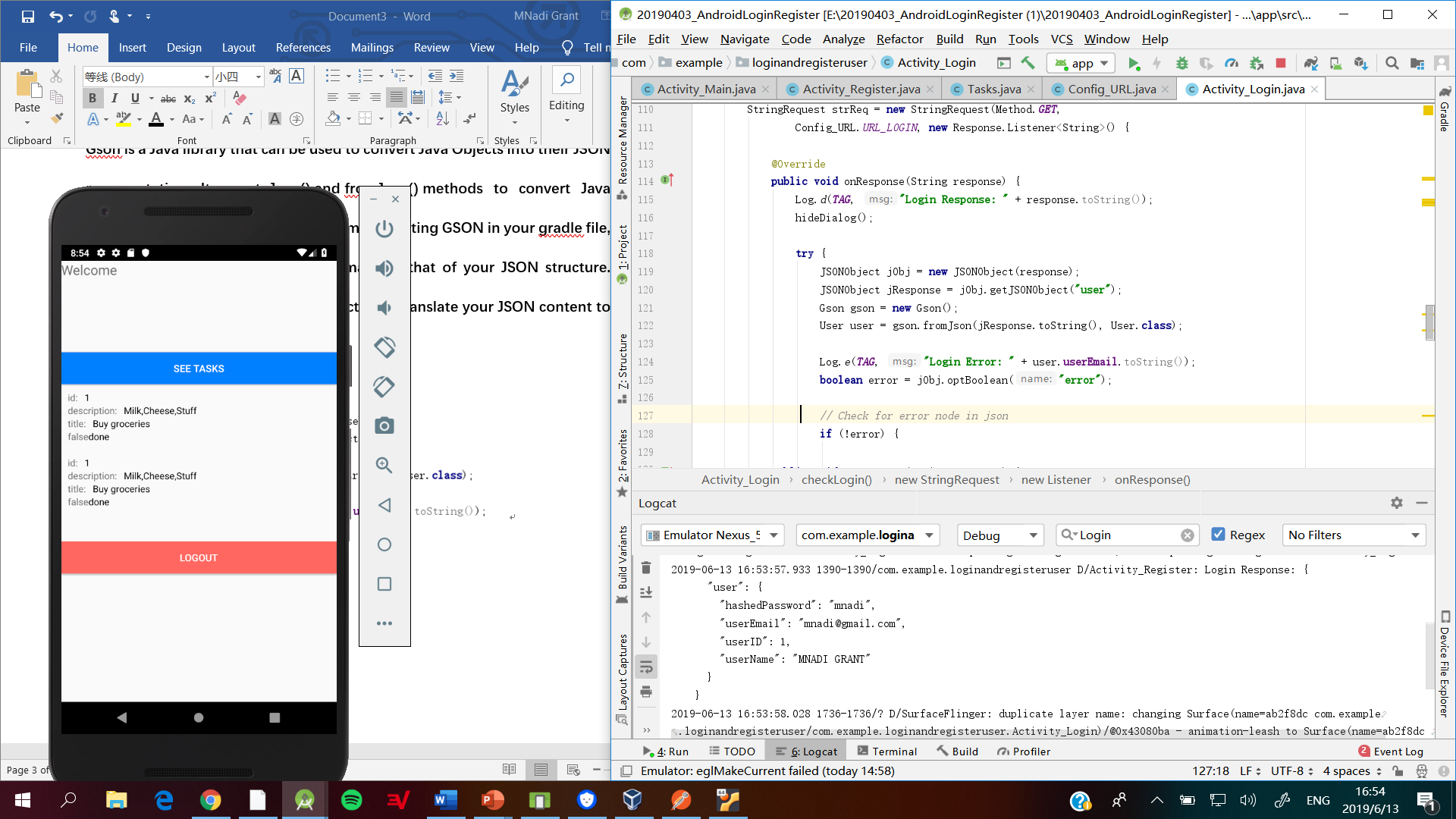


* **Understand how to use JSON/Gson**

Gson is a Java library that can be used to convert Java Objects into their JSON representation. It uses toJson() and fromJson() methods to convert Java objects to JSON and vice-versa. After implementing GSON in your gradle file, you then create a Java Class which matches that of your JSON structure. Afterwards you extract the JSON Object then translate your JSON content to your Java object.



**Logcat**



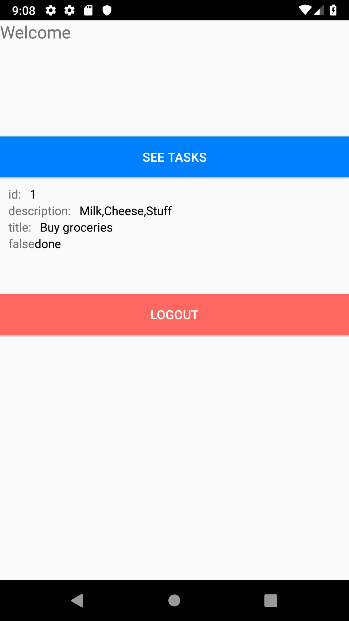
* **Add “Task” Class, and related logic to the login app, check and make sure the app can communicate with the web service.**

**Accomplishments**

* **The Task Class was added.**
* **The app communicates with the web service.**
* **User can only enter the application if their email and password matches any of the permitted users provided by the web server.**

**Implemented Functions**

* **User can check the list of task they have created by clicking “SEE TASKS” button.**

****