Luis Enriquez
Week 4 Test:

1. **What is a content provider?**
   Is a framework that manage access to data stored by itself, stored by other apps, and provide a way to share data with other apps.

2. **Can a database have multiple content providers?**
   No, every content provider should have it's own instance of SQLiteOpenHelper.

3. **How would your access a content provider?**
   You use the "ContentResolver" object in your application's "Context" to communicate with the provider as a client.

4. **What is a system content provider?**
   The Android System content providers store common data such as contact informations, calendar information, and media files. These classes provide simplified methods of adding and retrieving data from these content providers.

5. **Explain the flow of grabbing content from a content provider? What other classes?**

6. **How to query data from a content provider?**

7. **What are projection, selection arguments, selection?**

8. **How to get a phone number from a device? Explain the flow?**

9. **How can you make your own content provider?**

10. **Does a custom content provider needs to be declared in the manifest?**

11. **What is a web service? Name some.**
    A web service is a collection of open protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer. Some examples of web Services are like the Github service, that returns us user data like name, profile picture, repositories created by the user, etc. Another one would be the OpenWeatherMap, that returns us the current weather and forecast of a given location.

12. **What are the verbs used in REST service?**
    Post, Get, Put, Patch and Delete.

13. **How can you make a REST call in Android without third party libs? Name the classes.**
    - You need to create an IntentService that will handle the requests.
    - On the Intent you need to get the url of the call, and use it to stablish a connection with the HttpURLConnection method with the provided URL.
    - After that you need to declare an InputStream with the getInputStream Method.
    - Now that you have the stream, you must create a scanner to read the data, and use a stringbuilder to take the data from the scanner.
    - Now that we have the info, we create a new intent and pass the data and send a broadcast.

14. What happens when you make a network call on the main thread?

If the call takes too long, you could get an ANR (Application Not Responding).

15. Explain how you can make a REST call using okhttp3?

16. How can you make synchronous/asynchronous call using okhttp3?

17. How to add headers/query parameters to a request in okhttp?

Using an Interface with VERBS for the REST service, Like this:

```
@GET("/users/{username}")
Single<UserData> getUserData(@Path("username") String useName);
```

18. What is Retrofit? How do you setup RetrofitHelper class?

Retrofit is a library that makes parsing an API response easy and handled better for consumption in the app.

For setting up the RetrofitHelper Class:

- First you need to create the interface that is going to make the queries to the URL.
- And in the Helper Class, you need to create an Interceptor, and set its level to the BODY.
- Then you need to create the HttpClient, preferably with OkHttpClient, and add the interceptor to it.
- Finally, you create the Retrofit with Builder(), you add the baseURL, and the client, also, if you are going to use it with RXJava, and Gson, you need to add the convertorFactory, and the CallAdapterFactory.

19. How can you use different verbs in Retrofit?

With the Interface that was mentioned on the previous question.

20. What are the ways to transform JSON response to DataBean classes?

You can use POJOS.

21. How can we serialize a REST JSON response to the data bean classes?

Using Retrofit with a POJO.

22. Can you add the serialization library to the Retrofit instance? How?

Yes, First, you need to create the Retrofit, and create a service using the interface that is going to get the response from the API, and this method is also going to return us the POJO.

23. How would you serialize a XML response to Data bean objects using Retrofit?

Using the retrofit converter.

24. What is MVP? How do you setup MVP?

Is an architecture pattern that stands for Model View Presenter. For setting up, as it's name says, you need to divide the code in three layers, The Model, which will hold all the data structure, it is recommended to use a Repository pattern, where we can store all the databases and network calls. In the View layer, we will have the views, the ui part of the application, this layer, will only be in charge of displaying the UI, it won't be having any logic, it will only be receiving and sending the data for the user to the Presenter, which is the last layer. Here on the Presenter, which is the middle part, will be getting the data from the view, and will be applying the logic for the different tasks that are needed, and will be communicating with the model, getting the data from it, or updating it. All this tasks are done using Interfaces between each layer.

25. Is MVP better than MVC? Why? Explain the benefits of MVP.

In some aspects MVP is better than MVC, for example, in MVC The data- and event-flow is circular. And the view will often contain significant logic (like event handlers for user actions). Together, these properties makes the system difficult to test and hard to maintain. Whereas in MVP the logic, (like event handlers and user interface state) can be moved from the view to the presenter. Also The user interface can be unit tested in terms of the presenter, since it describes the user interface state. Inside the unit test, we replace the view with a test driver that makes calls to the presenter.Since the user interface is isolated from the application logic, both can be developed independently.

26. What is Dagger? What are the four components of Dagger?

Dagger is a fully static, compile-time dependency injection framework for both Java and Android. The components of Dagger are:

- Scopes.
- Providing Objects.
- Component Dependencies.
- Subcomponents.

27. How do you setup Dagger? Explain the classes created.

- First you need to add it in the Dependencies in the gradle file.
- Then you need to create the Component Interface and the Module Class.
- The Component is declared as a Singleton and a component, and we declare a the Component class with the "Component" Annotation.
- Here in the interface, we are going to declare the activities and ViewModels that we want to Inject using Dagger.
- Back in the Module Class, we create a Constructor, that is going to recive the context, and also create the getter method, that will return the context.
- Inside the module, we will declare the providers that will return the repositories.

28. How do you create the object graph in Dagger?

29. What are Runtime permissions? Which Android version introduced it? Why was this introduced?

The permissions are the ones that are asked for the user during the runtime of the application, these were introduced on Android 6.0 these were added for dangerous operations, these are asked every time the user is going to do an specific task, so the user can revoke these permissions at any time.

30. What are normal and dangerous permissions? Name some in each.

The normal permissions, are permissions that we give to our application for simple tasks, like for example, access to the internet, set an alarm, change the wallpaper, change the Time Zone, etc. As for dangerous permissions, these are permissions that require the approval of the user at runtime, so each time we want to do a "dangerous" task, we must ask for permission to the user, some dangerous permissions are: read the calendar, record audio, make a phonecall, get access to the camera, etc.

31. How would you ask for permissions at runtime?

First we need to check if we have permission to do the task in hand with the checkSelfPermission() method, if we don't have it, we need to use the method requestPermission(), and this will prompt an Android dialog asking for permission from the user, if we are granted the permission, we can continue with the given task.

32. What is the callback after the user responds to a permission?

The callback is the same request code that was passed to the “requestPermission()” method.

33. Why do we need to send a request code while asking for a permission?

We need to send the request code so we can ask for the appropriate permission.

34. What does ActivityCompat.shouldShowRequestPermissionRationale do?

The shouldShowRequestPermissionRationale() function returns true if the app has requested this permission previously and the user denied the request. If the user turned down the permission request in the past and chose the Don't ask again option, this method returns false.

35. How can you ask for multiple permissions at once?

You add the desired permissions to a String Array that is supplied as the first parameter of the “requestPermissions()” method, after this, you need to handle the acceptance and rejections of each permission.

36. How to use Google Play Services lib in your app?

37.What are the permissions you need to access a device location? What is the difference in both of them?
38.Which permission group does the above permissions belong to?
39.Which class is used to get the location on the device once?
40.How to get the location using the above class?
41.How can you get frequent location requests?
42.What are the methods implemented in LocationListener interface?
43.Is the Location class serializable or parcelable?
44.How to setup a Google Maps in your app?
45.What is the callback that updates the Map? How do you call that callback in the activity?
46.How to add a marker in the map?
47.How to animate a map to a different location?
48.How to see current location (blue dot) in the map?
49.What does the onStatusChanged used for? What are the arguments?