**Senior Android Developer Case Study**

**Part 1 – Android Case Study:**

Create a native Android application that can receive and parse data from the Google Play Store broadcasted intent parameter “referrer”. It should combine that data with what data it can get from the device, display it in a simple UI, post it to an API endpoint, and display the API response.

The relevant information from the referrer parameter will be stored in the querystring of the URL we pass in. You need to parse this data and store it in the app in a database alongside the device data (ID, manufacturer, model, etc.).

The app should display the combined information from the database (keys and values) when it is open in a simple UI.

Configure the app to also support a deep link URL. Take the parameters in the deep link URL and add them to the database, overwriting non-unique keys, if necessary. This can be validated using a custom run configuration.

Include a button that formats all the data as JSON and posts it to our API endpoint. It should then display the response from the API via a custom dialog. This dialog should allow the user to copy the response to the clipboard.

**Play Store Referrer:**

referrer=https%3a%2f%2fm.alltheapps.org%2fget%2fapp%3fuserId%3dB1C92850-8202-44AC-B514-1849569F37B6%26implementationid%3dcl-and-erp%26trafficSource%3derp%26userClass%3d20200101

**Deep Link URL:**

spigot://eng.dev/cs/product\_info?id=12345&trafficSource=deeplink

**Example Device Info:**- Device ID  
- Manufacturer  
- Model  
- [Anything else Interesting]

**Endpoint URL:**

POST JSON tohttps://casestudy.alltheapps.org/mobile/install

**Part 2 – Written Prompt**

In a text file included in your project, add a brief written answer to the following prompt:

You need to mirror and link the local database from part 1 to a database on an external server. Data added directly to the remote database must reflect on the devices UI immediately, and any data added to the local device must be synced to the remote database immediately. How would you architect this to ensure that the UI is always showing the latest information and can detect changes to the remote database?

Requirements:

* Must be able to handle network loss / drops
* Device must be notified / query database quickly and efficiently