

**Mobile Application Developer Technical Question**

Hi, first of all, we are pleased to see you among us. Before moving on to the next step, we would like you to build an application so that you can present your application writing capabilities to us and see what will be your adventure with us. **You have 72 hours to build that application. You can send your answers to** [**talent@evreka.co**](mailto:talent@evreka.co)**.**

**Information About The Application**

* Our mainstream applications are written in native Android SDK, you can use Android SDK or Flutter to give your application cross-platform support. Flutter is a bonus.
* With the application you will build, we want you to show the fullness rate of containers in a city and to change the location of those containers. The application should consist of 2 main screens: User Screen and Operation Screen. User screen is not required but is a huge plus. Therefore, please make sure to complete the Operation Screen first.
* **The correct implementation of the design is required.** You will be able to see component ratios, color, and font size types in **XD File.**  
  <https://xd.adobe.com/view/34e57fb3-210a-428e-42b1-a5d8edbb1ecb-42d4/grid>  
  **(Please try again if it does not open.)**

**Essentials**

* The application must check connectivity first and warn users about connectivity issues. The issues that connectivity related or user-related ones must be presented to the user correctly.
* If you will implement the Login screen as a bonus, after passing the login screen the user will be passed directly to the operation page. If not, the operation page must be opened after the app is opened.
* We expect from the operation screen to be built with Google Maps. Users should be able to see the fullness rates of the containers on the map.
* After the user clicks on any container on the map, s/he must be able to see the last known fullness rate of the container and the temperature that is measured with a sensor, and also provide the user up-to-date container data, sensor id and the last data from the sensor.
* You must be able to update the container's position on the operation page, as the position of the containers may change physically over time. We expect you to make a container update considering the user experience.
* All of those operations must be held on a remote server. Here you can use Firebase. You can make a request to the database on the Firebase and keep all updates alive. There might be different users using the same app in the same field, so all updates must be received by other users. At that point, we are expecting from you the most applicable approach of updates and explanation of why to use that approach when updating users.
* You can produce the position of the containers in the city by yourself. The data measured from the sensor can also be the data you have generated. This is a bonus if you can get this data ready from the outside.
* We expect you to display at least 1000 containers on the map. Our measurement criterion is the application's performance and user experience under 1000 containers. Also, we would like to hear what performance issues can be experienced and what can be applied to solve those issues.
* We recommend that you use the REST structure and follow the Android life-cycle while writing the application.
* Your application must be able to present several telemetries to the developer, so that developer will be able to identify issues before users get high. At that point, your key point indicator (KPI) suggestions will be precious for us.
* We like to run the application under odd testing, so we expect the application to be able to work for users with different behavior under all circumstances.
* Finally, after finishing the application, share the application codes with us in a git repository and share the released .apk file.

Summary of the app features will be like the following:

**Required:**

* Operation screen

○ Google Maps

* Display of containers on the map

○ 1000 containers

○ Container occupancy rates are visible on the map

* Container information

○ Occupancy rate measured by the sensor

○ The container temperature measured by the sensor

○ Date of data received from the sensor

○ The id of the sensor

○ Container ID

● Update container position

● All data must be kept on the server, can be updated

**Bonus:**

● User login screen:

○ Internet check

■ Non-availability

■ The guidance of user in such cases (bonus)

○ User name - password entry

○ Server authentication

■ The case of non-connectivity

■ The cases of incorrect user name, password entry