**Technical questions**

How long did you spend on the coding test? What would you add to your solution if you had more time? If you didn't spend much time on the coding test then use this as an opportunity to explain what you would add.

**I spent 2 days for the coding test because I am about to go out for a holiday. Even though I had 3 days in total for this test, I was able to only spare 2 days. Even so, I was able to complete the user stories and add some extra features to the app. If I could use more time, I would,**

**- Add unit tests and flowcharts for user stories.**

**- Add user behavior analytics tools like Mixpanel, Customer.IO or Firebase Analytics to keep track of user events ad it was mentioned at task description to understand user behavior**

What was the most useful feature that was added to the latest version of your chosen language?

Please include a snippet of code that shows how you've used it.

**StateFlow of Kotlin coroutines has been recently added to the Kotlin coroutines features. It is basically a Flow but it has an internal state value just like LiveData. Since StateFlow extends from Flows, it has advanced filtering and mapping capabilities. In code test, I used it for the state of the user.**

**User state object**

sealed class UserState {

object Loading: UserState()

object Unauthenticated: UserState()

class FailedSignIn(val message:String): UserState()

class Authenticated(val customer: CustomerEntity): UserState()

}

**Definition in UserRepository**

private val userState :MutableStateFlow<UserState> = *MutableStateFlow*(UserState.Loading)

**Conditional navigation at LauncherActivity**

*observe*(viewModel.userState)**{**

when(**it**){

is UserState.Unauthenticated -> navigateToActivity(AuthActivity::class.*java*)

is UserState.Authenticated -> navigateToActivity(HomeActivity::class.*java*)

}

**}**

How would you track down a performance issue in production? Have you ever had to do this?

**For this purpose I would use Firebase Analytics or legacy Fabric Crashlytics tools. For ANR and slow startup issues, google play console has a decent setup and I find it very useful to work with.**

How would you debug issues related to API usage? Can you give us an example?

**For me, if this is happening on my local machine, first step is to check the API endpoint with an external tool like PostMan or Insomnia. This step mainly tells me if the issue is backend related. If it works at the external tool, I compare my implementation with the tool and make a lot of use of breakpoints and debug logging.**

**If issue debugging happens on production, then something like below works fine.**

**API usage is highly known with the repository classes, which extends BaseScope. BaseScope has a global error handler. It gets triggered whenever exception occurs in suspend blocks of the repository. At this point** *CoroutineExceptionHandler* **will be able to receive the exception. Then it can be shared with any remote server to be viewed by the developers.**

open class BaseScope:CoroutineScope {

private val errorHandler = *CoroutineExceptionHandler* **{** coroutineContext, throwable **->**

Timber.e(throwable)

**}**

override val coroutineContext: CoroutineContext

get() = Dispatchers.IO + errorHandler

}

How would you improve the Node server’s API that you just used?

**One issue that I noticed is the case sensitivity of backend. It requires exact email address to be inputted with lower and upper cases to let a user log in. For example there is a user with email [Nadiah.Spoel@example.com](mailto:nadiah.Spoel@example.com) in the database. If customer inputs her email as lowercase it will return 404 error message.**

**Another issue is, after creating a profile with sign up, api needs to restart at least once to be able to sign in with the new user.**

**Also there could be improvements in the db schema, for example there can be a separate categories table to have proper naming. Right now there are categories with the same name with upper and lowercase letters.**

Please describe yourself using JSON.

{

"name":"Kemal Atlı",

"dateOfBirth":"1989-07-25T07:30:00+03:00",

"nationality":"Turkey",

"numberOfChildren":1,

"martialStatus":"Married"

"city":"İstanbul",

"loves":[

"Programming in Kotlin",

"Reading History Books",

"Playing Chess",

"Playing Video Games"

],

"lookingFor":[

"Space Exploration",

"Peace"

]

}