Gini Assignment

* Run the application

I’m using Android Studio Ladybug Feature Drop | 2024.2.2

* What I have implemented
  + **EnterCityScreen** in the example app
  + **ForecastScreen** in the WeatherSdk
  + Both screens are built-in compose and adapted to support XML in **EnterCityScreenFragment** and **ForecastScreenFragment**
  + The solution Contains **MainActivity** for Compose sample and **MainActivityWithXml** where the default one is **MainActivityWithXml**
  + I’m using MVVM architecture where I have the following layers
    - WeatherApiService (networking)
    - Weather and local (repositories)
    - Use cases for current weather and forecast (Forecast)
    - Weather (ViewModel)
    - I’m using Compose **NavHost** for navigation and routing.
  + I’m using Object WeatherGiniSDKBuilder as a singleton class
  + Within WeatherGiniSDKBuilder, I’ve added sdkStatus as MutableLiveData to post and observe the status globally within the app.
  + I’m using the BaseState Sealed class to handle the data and error cases
  + I’ve created StatesLayoutCompose to handle the data and errors or exceptions in a friendly way (show a refresh button if there is a connectivity issue, unauthorized message …etc).
  + I’ve Implemented Unit tests using **Mockk**, and **KoinTest** libraries within the below (.
    - **WeatherViewModelCurrentWeatherTest** has the following tests
      * ApiKey not initialized
      * ApiKey initialized
      * Data loaded successfully
      * No internet connection
      * Not Authorized
      * Internal server error
    - **WeatherViewModelForecastTest** has the following tests
      * ApiKey not initialized
      * ApiKey initialized
      * Data loaded successfully
      * No internet connection
      * Not Authorized
      * Internal server error
  + I’ve added the Kdoc comments to the code and integrated Dokka into the solution.
  + I’ve added Integration Guide **IntegrationGuide.md** within **WeatherGiniSdk** directly and **readme.md** within the root project is updated as well.
* What could be enhanced?
  + Within WeatherGiniSDKBuilder, I’ve added sdkStatus as MutableLiveData but we can also update the status within the application in other ways depending on the case itself.
    - Broadcast receiver
    - EventBus
    - SharedViewModel
  + We can wrap the use cases for the current weather and forecast in one use case in a sync or async way but I’ve kept them separated to manage multiple states in one screen.
  + Dokka is integrated but it still generates an empty file.
  + We can implement UI Tests to compose and generate more complex scenarios within the existing ones.
  + I’ve used **MVVM** but if we consider compose I would love to move to other **MVI** libraries like **Orbit**.
  + The design absolutely could be much better.
  + The **ApiKey** could be encrypted and stored more securely.
  + Enhance the Gradle by moving from Groovy to Kotlin and storing the dependencies better (store them in the object class and include them in Gradles).
* Reference
  + IntegrationGuide.md  
    <https://github.com/adhamkhwaldeh/WeatherSdk/blob/main/WeatherGiniSDK/IntegrationGuide.md>
  + Record for the application  
    <https://github.com/adhamkhwaldeh/WeatherSdk/blob/main/ApplicationRecord.mp4>  
    <https://drive.google.com/file/d/1644pRm5F3Ro1CJbb8O_n5Kw7SRC96q2d/view?usp=drive_link>