



tawk.to Android Developer Test

Your task is to make an app that can fetch GitHub users list and then display the selected user's profile.

PLEASE NOTE - **All requirements** in all the sections **must be met** (except the BONUS section - that is optional, you can do all, some or none of the BONUS tasks).

1. General Requirements

- REQUIREMENT 1** In the first screen, the app has to fetch GitHub users list, parse it and display in the list.
- REQUIREMENT 2** Selecting a user has to fetch user's profile data and open a profile view displaying the user's profile data.
- REQUIREMENT 3** The design must loosely follow the wireframe (at the bottom of this document) but you must demonstrate a high level of knowledge about best practises of Android UX and UI principles (e.g. Material design). The app must look good and work smoothly and the design must follow platform defaults.
- REQUIREMENT 4** Code must be commented.

2. Generic Requirements

- REQUIREMENT 1** Code must be done in **Kotlin 1.6.x.** using **AndroidStudio**.
- REQUIREMENT 2** Data must be persisted using **Room**.
- REQUIREMENT 3** UI must be done using **ConstraintLayout** and/or using **Jetpack Compose**. As there are 2 views of the app - the list and the details view, one of these must be done in **Jetpack Compose**, it is up to your choice which one.
- REQUIREMENT 4** All **network calls** must be **queued** and **limited to 1** request at a time.

- REQUIREMENT 5 All **media** has to be **cached** on disk.
- REQUIREMENT 6 Write **Unit tests** for data processing logic & models, Room models (validate creation & update).
- REQUIREMENT 7 Screen **rotation** change must be supported.
- REQUIREMENT 8 **MVVM** pattern must be used utilising **Kotlin Coroutines**.

3. GitHub users

- REQUIREMENT 1 The app has to be able to **work offline** if data has been previously loaded.
- REQUIREMENT 2 The app must handle **no internet** scenario and **display** appropriate UI indicators.
- REQUIREMENT 3 The app must **automatically** retry **loading** data once the **connection** is **available**.
- REQUIREMENT 4 When there is persisted **data available** from previous sessions, that data should be **displayed first** and then (in parallel) new data should be fetched from the backend.

4. Users list

- REQUIREMENT 1 Github users list can be obtained from <https://api.github.com/users?since=0> in JSON format.
- REQUIREMENT 2 The list must support **pagination** using **Android Paging Library** (scroll to load more) utilizing `since` parameter as the integer **ID** of the **last User** loaded.
- REQUIREMENT 3 **Page size** has to be **dynamically** determined after the first batch is loaded based on the number of items returned.
- REQUIREMENT 4 The list has to display a **spinner** while loading data as the **last list item**.
- REQUIREMENT 5 Every **fourth** list item's avatar (the **image** - not the background!) should have its **colours inverted** ([example](#)).
- REQUIREMENT 6 List item view should have a **note icon** if there is note information saved for the given user.
- REQUIREMENT 7 Users list has to be **searchable** - local search only; in search mode, there is no pagination; **username** and **note** (see Profile section) fields should be used

when searching, precise match as well as `contains` should be used also the search should be **case insensitive**.

5. Profile

- REQUIREMENT 1** Profile info can be obtained using Github user's `username` from <https://api.github.com/users/username> in JSON format (e.g. <https://api.github.com/users/tawk>).
- REQUIREMENT 2** The view should have the user's avatar as a header view followed by information fields (UIX is up to you).
- REQUIREMENT 3** The section must have the possibility to retrieve and save back to the database the **note** data (not available in GitHub api; local database only).

6. Submit your work

- REQUIREMENT 1** Make sure you have met **all requirements** (from above)
- REQUIREMENT 2** Make sure the code is running on both - the **simulator** and **real** device.
- REQUIREMENT 3** Push up to your **repo**. The repository should be **publicly accessible** - must not require any invitation to be accepted or registrations.
- REQUIREMENT 4** Reply back to the initial email the link to your repository.
- REQUIREMENT 5** Any **3rd party code** that is being used has to be **bundled** with the code even if the project is using any code dependency manager.
- REQUIREMENT 6** Please **indicate** which **bonus tasks** (see next section) have you completed.

7. BONUS

- BONUS 1** Empty views such as list items (while data is still loading) should have Loading Shimmer aka **Skeletons** ([example1](#), [example2](#), [example3](#)) resembling final views.
- BONUS 2** **Exponential backoff** must be used when trying to reload the data.
- BONUS 3** The app has to support **dark** (night) **mode**.

Wireframe

[wireframe.png](#)

