



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).

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 users 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).
- REQUIREMENT 4.** Own code logic should be commented on.

Generic Requirements

- REQUIREMENT 1.** Code must be done in Kotlin 1.4.x. using AndroidStudio.
- REQUIREMENT 2.** Data must be persisted using **Room**.
- REQUIREMENT 3.** UI must be done using **ConstraintLayout** where appropriate.
- 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.

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, **show** appropriate UI indicators.

REQUIREMENT 3. The app must **automatically** retry **loading** data once the **connection** is **available**.

REQUIREMENT 4. When there is **data available** (saved in the database) from previous launches, that data should be **displayed first**, then (in parallel) new data should be fetched from the backend.

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** (*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.

REQUIREMENT 4. The list has to display a **spinner** while loading data as the **last list item**.

REQUIREMENT 5. Every **fourth** avatar's (the image - not the background!) colour should have its colours **inverted**.

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.

Profile

- REQUIREMENT 1.** Profile info can be obtained from [https://api.github.com/users/\[username\]](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).

Submit your work

- REQUIREMENT 1.** Make sure you have met **all** the **requirements** (from above)
- REQUIREMENT 2.** Make sure code is running on both - **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.

BONUS

1. Empty views such as list items (while data is still loading) should have Loading Shimmer aka **Skeletons** ~ https://miro.medium.com/max/4000/0*s7uxK77a0FY43NLe.png **resembling** final views.
2. **Exponential backoff** must be used when trying to reload the data.
3. MVVM pattern should be used.
4. Users list **UI** must be done in **code** and Profile - with **Layout Editor**.
5. The app has to support **dark mode**.

Wireframe

[wireframe.png](#)

