IT5007: Software Engineering on Application Architecture

Assignment-4: Issue Tracker System Goes Mobile

Deadline: 26th Nov 2023

Points: 20

The author of the MERN Stack textbook, which we use in IT5007, needs your help. In the revised edition of the textbook intended to be published in 2023, the author has decided to include a chapter on mobile-based front-end development. The author wants to retain the back-end (GraphQL/MongoDB) developed so far but needs you to help him code the mobile-based front-end and the interfacing logic. To save you the trouble of learning a new language, you can code the front-end using ReactNative that has the same principles (e.g., component, state, etc.) as React but uses a different set of React Native libraries (e.g., <TextInput> instead of <input> tag) for rendering the front-end.

You are required to implement all the functionalities currently available on the web-based platform:

1. IssueFilter dummy component [1 point]
2. IssueTable component to display the issues [8 points]
3. IssueAdd component to add a new issue with (title, owner, effort, etc.) [8 points]
4. BlackList component to add an owner to the blacklist. [3 points]
5. Bonus: Navigation bar for the App, Bootstrap/CSS for beautifying the App. [3 extra points, capped at 20 points for the assignment]

Instructions

1) Remember that your mobile app should work with absolutely no modification to the back-end provided to you. The back-end logic is pre-coded and requires no change. However, you could write to the Instructor if you want to make any changes. At the back-end, the graphql server is running at port 3000, path '\graphql'.
2) You will be provided with a sample working code (in ReactNative) of the front-end.
3) The provided sample front-end will have markers/comments (e.g., ***Q1***) indicating where you need to add your logic/code.
4) Make sure that all the dependencies are entered/saved in the package.json of the front-end.
5) We will be testing your front-end code on Android Studio Emulator. We will also be running the back-end provided, as usual using the docker container with port 3000 open. You can test your setup by downloading your own code and trying to run the setup, just as a TA would do.
6) Github Classroom link for this tutorial can be found on Canvas.