



B5 - Application Development

B-DEV-501

Redditech

Browsing app for Reddit

1.0



Redditech

language: Java/Kotlin (Android), React Native, C#/.NET (UWP)



- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.

GENERAL CONSIDERATIONS

As part of this project, you will take on the role of a Software Architect.

Your main goal is neither reinventing the wheel nor writing numerous lines of code. On the contrary, your main action is to understand, select and integrate a wide range of existing libraries.

Your code will be composed of two main parts: the first can be assimilated to *glue* in between of existing functional components, the second part will be the so-called *business* logic. These two things (and a bit more) should compose the requested project.

However, before embarking on such a project, we suggest you take the time to:

- Analyze each framework (and available libraries for each) at your disposal (See the Development Environments section)
- Understand the pros and cons of each framework
- Analyze the API you'll have to implement
- Choose the framework that suits you and the project best



There are several codeless tools to request APIs and check the proper formatting of requests, responses status and data, ... such as Postman, try it, it's worth it!



PROJECT DESCRIPTION

For this project, you are asked to create an application (mobile or desktop) using the Reddit API.

Your application must:

- Comply with the project's build constraints
- Offer the features detailed in the next section
- Offer a high-quality, polished UX (User Experience) and UI (User Interface) while respecting the chosen framework best practices

Your application should:

- Come with a clear test strategy and be tested
- Come with a technical documentation (Class Diagram, Component Life Cycle, ... depending on your framework)

Your application could (AKA bonuses):

- Come with a user documentation in the form of a PDF (or equivalent) file
- Come with a user documentation in the form of an integrated tutorial (carousel at first launch, ...)
- Have any other feature of your choosing not listed in the next section



Be careful, the Reddit API doesn't like JSON much! Request bodies shouldn't be passed as JSON, or at least, not always :p



These are a few links concerning the Reddit API that you might find useful. You're welcome ^^

- [API base url](#)
- [Register a new App](#)
- [Endpoints documentation](#)
- [How to OAuth2 and more](#)



MANDATORY FEATURES

You must create a browsing application with these features:

- Authenticate to the *Reddit* API with the OAuth2 protocol
- Display the user's profile. The bare minimum is to display the profile picture, username, and description
- Display posts from the subreddits the user is subbed to
- Filter the posts (ie. best, hot, new, top, random, ...). At least 3 different filter must be proposed
- Search for subreddits
- Display a subreddit basic information. The bare minimum is to display the name, title, number of subscribers, description, and header image
- Display posts from a subreddit (You must also be able to filter the posts with the same restrictions as mentioned above)
- Subscribe/Unsubscribe to a subreddit
- Display at least 6 of the user's settings
- Be able to update at least 6 of the user's settings



The order of the features listed above has its importance.



You must handle the pagination, called **listings** in the Reddit API. It will be checked during the project review!



Be careful what subreddit you use during your tests, your app must be able to properly display and play videos, gifs, images, ... contained in a post!!

USER INTERFACE AND EXPERIENCE (UI/UX)

The user experience that your application offers is an aspect that is just as important as its functionalities. You should consider some ideas for the experience that you would like to offer to your users, and how your - to be created - app interface will connect with this experience.

Each application is built according to its own set of rules and expectations. Creation of a visual identity, navigation ergonomics, readability, intuitiveness are all non-exhaustive points that should be taken into account.



This reflection is an integral part of your application's design process.



Consult the guidelines provided by the technology you chose. For instance, if you develop an Android app, we recommend you to look at [Material Design/UI](#)



Be attentive to details. For instance, you can easily create [Dark Patterns](#) by mistake, which isn't ideal

Your application's visual identity and user interface will be evaluated, just like the quality of the experience it offers to the user.

As such, we will check if you respect the chosen platform's guidelines (ie. Material Design/UI for Android) such as:

- The choice and coherence of style (colors, icons, typography, etc.)
- The usage of space (size, placement, density, etc.)
- The choice of elements (buttons, lists, menus, etc.)
- Your project's identity and the cohesion of it all

APPLICATION TEST

As you know, any software project must be tested in order to make sure that it functions correctly and complies with its scope statement.

You should set up a test strategy for your application. The completeness of your strategy will be taken into account during the evaluation of your project.



You're free to incorporate any test framework you want in order to make your test strategy more complete.

DOCUMENTATION

It is important to take the time to define a simple - yet scalable - architecture without code duplication. You should provide a clear and comprehensive documentation for your project, containing class diagrams, sequence diagrams, components Life Cycle, ... depending of the framework and libraries you chose

The goal is to have a document that helps newcomers to easily understand the project architecture and more in order to reduce the project's learning curve and ease communication between team members. Thus, there is no need to make diagrams/schemas of the entire project, but rather to choose the important parts that needs to be documented.



DEVELOPMENT ENVIRONMENTS

For this project, you have the choice to use one of three languages/frameworks detailed below.



It's up to you to choose the libraries you want in order to meet the project's needs (network, data etc.)

JAVA/KOTLIN: SPECIFIC DETAILS

If you choose to develop the project in the JVM environment on Android, you must respect the following details:

- Your application must be developed in Java or Kotlin (Webviews and NDK Android usage prohibited).
- Your application must be targeted at the Android SDK API 29.

Your project must use *Gradle* as its build automation system.

The builds must be launched by using the *Gradle Wrapper* tool in the command line.



That means that a `build.gradle` file should be in the root of your repository.

REACT NATIVE: SPECIFIC DETAILS

- The project must be, at least, able to build for Android

Your project must use *Gradle* as its build automation system (to be built for Android).

The builds must be launched by using the *Gradle Wrapper* tool in the command line.



That means that an `build.gradle` file should be in the `android` directory at the root of your repository.

C#/.NET: SPECIFIC DETAILS

- The project must be developed in C#
- The project must be built using a Visual Studio 2019 solution (available [here](#) in the Education section)



That means that an `redditech.sln` file should be in the root of your repository.