

Flutter Developer

Technology Assessment Case Study

Exercise 1

- Low Level Design
- Software Development Practices
- Hands-on Programming

Media Probe Mobile App

Introduction

Media Probe is a revolutionizing online media. They aim to be the one stop shop for all relevant media articles, which can be personalised to users. They aim to reach all corners of the globe and be easily accessible. They have requested for a mobile application.

Build a simple app to hit the NY Times Most Popular Articles API and show a list of articles, that shows details when items on the list are tapped (a typical master/detail app). You should use **Flutter**.

Business Requirements

User Story 1: As a customer, I want to view a list of most popular news articles for the last 7 Days

Acceptance Criteria

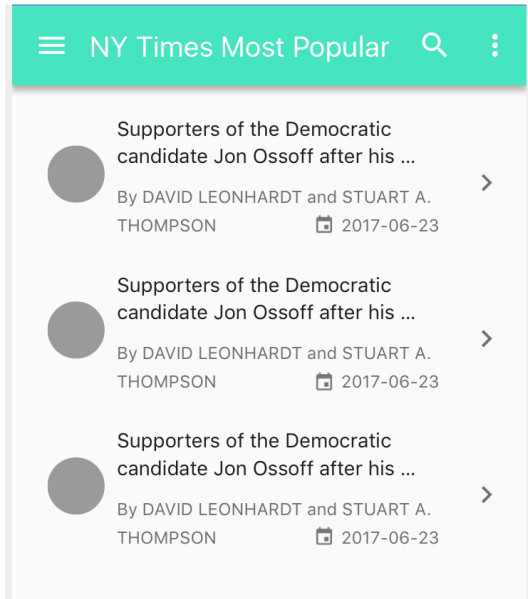
- The list should include the title of the article, a short snippet of the article and an image related to the article
- The list should show only the first page of results
- The list should be sorted based on published dates

User Story 2: As a customer, I want to select a news article and view more details

Acceptance Criteria

- The article detail page should include the article title, the publish date of the article, an image related to the article, the full text of the article
- The article should open within the application and not on an external link

Screen Design Sample



Technical Requirements

We'll be using the most viewed section of this API. Note: you need to signup for an API key at: <https://developer.nytimes.com/get-started>, then replace the 'sample-key' below with the API key assigned to your account.

<http://api.nytimes.com/svc/mostpopular/v2/viewed/{period}.json?api-key=sample-key>

To test this API, you can change the period section of the path (available period values are 1, 7 and 30, which represents how far back, in days, the API returns results for).

<https://api.nytimes.com/svc/mostpopular/v2/viewed/7.json?api-key=sample-key>

WebViews are not allowed in this exercise. Create a **GitHub repository**, ensure the name is generic and doesn't have any company names. Commit your code to the GitHub repository and share the link with us. Only share a link, do not send the actual code files

Required Activities

- Object oriented programming approach
- Good UI approach e.g. MVC, MVVM, MVP, etc. Please specify the pattern used in the README.md
- Unit tests using Junit, XCTest, to achieve good code coverage
- Some UI tests using espresso, XCUITest, etc.
- Code to be generic and simple
- Leverage today's best coding practices e.g. Swift, Android coding standards
- Clear README.md that explains how the code and the test can be run and how the coverage reports can be generated
- Developed code should be pushed to **GitHub** share the link with us

Bonus Activities

- Create scripts, e.g. Gradle, Fastlane, etc. to:
 - o Build the project from the command-line
 - o Run static code analysis such as linting
 - o Run unit tests and code coverage
- SonarQube report for the code showing its quality summary