1. **News App: Article Details Screen**

**Scenario:** In a news app, a user can browse headlines and select an article. Once the article is selected, the app navigates to a screen that provides detailed information about the article, including full text, images, and author information.

https://newsapi.org/v2/top-headlines/sources?apiKey=<YOUR\_APIKEY>&category=business

**Navigation & Arguments:**

* **Navigation**: The user taps on a news headline, which navigates to a detailed article screen.
* **Arguments**: The article ID is passed through the Navigation component to ensure that the correct article data is fetched and displayed.

**Retrofit Usage:**

* **API Call**: Retrofit is used to request the article content from the server.
  + **API Endpoint**: GET /articles/{articleId}
  + The app uses Retrofit to fetch article content.

**Steps:**

1. The user taps on a news headline.
2. The app navigates to the article details screen, passing the articleId as an argument.
3. Retrofit makes a call to fetch the article content using the article ID.
4. The detailed article is displayed, including text, images, and author information.
5. **Social Media App: User Profile Screen**

**Scenario:** In a social media app, a user can click on another user's profile picture or name to navigate to that user's profile page. The profile screen displays user data such as name, bio, followers, and posts.

https://jsonplaceholder.typicode.com/users

**Navigation & Arguments:**

* **Navigation**: A click on a user profile triggers navigation to the user’s detailed profile page.
* **Arguments**: The user’s unique ID (userId) is passed through the navigation component to ensure that the correct profile data is displayed.

**Retrofit Usage:**

* **API Call**: Retrofit is used to make a GET request to fetch the user profile information from the server.
  + **API Endpoint**: GET /users/{userId}
  + The app calls the Retrofit service to retrieve data related to the user.

**Steps:**

1. A user taps on a friend's profile from the list or feed.
2. Navigation to the profile screen occurs, passing the userId as an argument.
3. The Retrofit API call is made to fetch data like name, bio, follower count, and posts.
4. The user profile is displayed, with the fetched information.
5. **Fitness App: Exercise History and Details**

* **Scenario**: A fitness app displays a list of exercises performed by the user. Each exercise entry shows the type, duration, and calories burned. When a user taps on an exercise entry, they are taken to a detailed screen showing stats for that specific exercise.
* **Navigation with Arguments**: The app passes the exercise ID as an argument to navigate to the details screen.
* **Retrofit**: Retrofit is used to fetch the detailed exercise data using the exercise ID, including breakdowns of calories burned, heart rate, and duration.
  + **Use case**: GET /exercise/{exerciseId}/details

**4. Travel App: Booking Details**

* **Scenario**: A travel app allows users to search and book flights, hotels, and car rentals. After the user selects a flight, the app navigates to a booking confirmation screen.
* **Navigation with Arguments**: The app navigates to the booking confirmation screen, passing booking details (flight ID, user info, etc.) as navigation arguments.
* **Retrofit**: Retrofit makes network requests to fetch the latest flight status, seating availability, and booking details. These details are then displayed to the user.
  + **Use case**: GET /flight/{flightId}/status

**5. Online Banking App: Transaction History**

* **Scenario**: An online banking app shows a list of recent transactions. The user taps on a transaction to view more details about it (such as date, amount, transaction type).
* **Navigation with Arguments**: The transaction ID is passed as an argument to a new screen, where detailed information about that specific transaction is fetched and displayed.
* **Retrofit**: Retrofit is used to make a GET request to fetch the transaction details based on the transaction ID.
  + **Use case**: GET /transaction/{transactionId}