Test task Documentation

Documentation

This document provides an overview of the external APIs and libraries used, along with a guide to the code structure and primary modules in the project.

External APIs and Libraries

Kingfisher
- **Purpose**: Downloads and caches images for iOS.
- **Why Used**: Kingfisher efficiently handles image caching and downloading, which is particularly useful in list views or galleries where images are loaded frequently.

Code Structure Overview

The project is organized as follows:

- **AppDelegate.swift**: Configures the app's lifecycle and manages essential setup.
- **Cells/**: Contains custom table view cells for displaying user and position information.
  - **UserTableViewCell**: Displays individual user information in a table view.
  - **PositionTableViewCell**: Displays position information in a table view.

- **Fonts/**: Contains custom fonts (e.g., Nunito-Regular and Nunito-SemiBold) used in the app's UI.

- **Model/**: Contains data models used across the app.
  - **GetTokenResponse**: Represents the response for the `getToken` API call.
  - **Position**: Model representing a user position.
  - **RegisterUserResponse**: Represents the response for the `registerUser` API call.
  - **UserModel**: Model representing user data.

- **View/**:
  - **NoInternetView**: A custom view displayed when there is no internet connection. It includes retry functionality.

- **ViewController/**:
  - **CustomViewController**: A view controller as a base view controller in app
  - **ResultViewController**: Displays user registration result.

  - **SignUpViewController**: Manages user registration, allowing users to input their details and submit a signup request.
  - **UsersViewController**: Displays a list of users, handling pagination and data fetching for the user list.

- **NetworkMonitor.swift**: Manages network status monitoring and notifies views when the connection status changes.

- **DataManager.swift**: Contains API request functions, including `getPositions`, `fetchUsers`,`performRegistration`,`performRegistration`, `registerUser`, and `getToken`. This is the primary class for coordinating data interactions between the app and external APIs.

- **Assets.xcassets**: Holds images, colors, and other assets used in the app.

- **Info.plist**: Stores metadata about the app, such as permissions and configuration details.

- **LaunchScreen.storyboard**: Defines the initial launch screen of the app.

- **Main.storyboard**: Contains the main UI layout and navigation setup for the app.

- **SceneDelegate.swift**: Manages scene lifecycle for multi-window support.

Code Usage Examples

- **Fetching User Data**:
  ```swift
  DataManager.shared.getPositions { result in
      switch result {
      case .success(let positions):
          // Update UI with positions
      case .failure(let error):
          // Handle error
      }
  }
  ```