

NIT3213 Final Assignment Specification

Android Application Development Project

Objective:

Develop an Android application that demonstrates proficiency in API integration, user interface design, and Android development best practices.

Project Overview:

Create an Android app with three main screens: Login, Dashboard, and Details. The app will interact with the 'vu-nit3213-api' to authenticate users and retrieve data.

API Details:

Base URL: <https://nit3213api.onrender.com/>

1. Login Endpoint:

- URL: /footscray/auth or /sydney/auth or /ort/auth (depending on your class location)
- Method: POST
- Request Body:

```
{
  "username": "YourFirstName",
  "password": "sYourStudentID"
}
```

- Successful Response (200 OK):

```
{
  "keypass": "topicName"
}
```

2. Dashboard Endpoint:

- URL: /dashboard/{keypass}
- Method: GET
- Successful Response (200 OK):

```
{
  "entities": [
    {
      "property1": "value1",
      "property2": "value2",
      "description": "Detailed description"
    },
    // More objects...
  ],
  "entityTotal": 7
}
```

Requirements:

1. Login Screen:

- Implement a user interface for login with username and password fields.
- Use the student's first name as the username and student ID (format: s12345678) as the password.
- Make a POST request to the appropriate auth endpoint based on your class location.
- Handle and display appropriate error messages for unsuccessful login attempts.
- Upon successful login, navigate to the Dashboard screen.

2. Dashboard Screen:

- Implement a RecyclerView to display the list of entities received from the dashboard endpoint.
- Use the 'keypass' received from the login response to make a GET request to the dashboard endpoint.
- Each item in the RecyclerView should display a summary of the entity (excluding the description).
- Implement click functionality on RecyclerView items to navigate to the Details screen.

3. Details Screen:

- Display all information about the selected entity, including the description.
- Implement a user-friendly layout to present the information clearly.

Technical Requirements:

1. Implement dependency injection using a framework like Hilt or Koin.
2. Follow clean code principles and maintain a well-organized project structure.
3. Implement unit tests for critical components of the application.
4. Use Git for version control, with meaningful commit messages and a clear commit history.

Assessment Criteria:

1. Project Completion (40%): All required features are implemented and functional.

2. Code Organization and Cleanliness (15%): Well-structured, readable, and maintainable code.
3. Dependency Injection Implementation (25%): Proper use of a DI framework.
4. Unit Testing (10%): Presence and quality of unit tests for critical components like ViewModels.
5. Git Usage (5%): Meaningful commit messages and a clear project history.
6. ReadMe file (5%).

Submission:

- Submit your project as a Git repository link.
- Include a README.md file with instructions on how to build and run your application.
- Ensure all necessary dependencies and setup instructions are clearly documented.

Good luck with your project!