

Flutter Take-Home Project

Requirements

Develop a small app where users can:

1. **View Available LPG Gas Cylinders:** Display a list of available LPG gas cylinders (e.g. 3kg, 6kg, 13kg) with images and prices.
2. **Purchase a Cylinder:** Users can select a cylinder and add it to the cart.
3. **Manage Checkout States:** Implement logic to handle different outcomes of the checkout process, including success and error scenarios, providing clear feedback and appropriate user actions.

Technical Stack

- **Flutter & Dart:** Build a simple UI with a list of cylinders and order history using Flutter.
- **State Management:** Use a suitable state management approach (e.g. Redux, Provider, Bloc) for managing the app's state. Note that leveraging **Redux** for state management could give you an advantage, as it demonstrates experience with a more complex and scalable architecture.
- **Http/Dio Package:** Set up the http or Dio package to make network requests to a mocked API.
- **Async Programming:** Handle asynchronous tasks using appropriate Flutter mechanisms such as `Future`, `async/await`, and `Stream` to ensure smooth and efficient data fetching and user interactions.
- **Architecture:** Implement the app using the MVVM or DDD pattern.
- **Offline Support (Optional):** Consider implementing offline capabilities, such as caching cylinder data, so the app can function without an active internet connection.
- **Performance Optimization:** Optimize the app for performance by minimizing unnecessary rebuilds, using efficient data structures, and handling images and animations efficiently.
- **Security:** Implement security best practices to protect sensitive user data and ensure secure communication.
- **Testing:** Write unit tests, widget tests, and integration tests to ensure the app's functionality and maintain code quality. Use testing frameworks like `flutter_test` and `mockito` for mocking dependencies.

Project Expectations

- **Code Quality:** Write clean, maintainable code, and utilize Dart and Flutter best practices.
- **Commit Messages:** Write meaningful commit messages and push changes to the appropriate branches (i.e., feature, refactor, fix).
- **Version Control (GitLab):** Host your project on GitLab, using a well-organized repository structure. Ensure the repository is public or provide access as necessary.
- **GitLab Link:** Share the GitLab repository link before the interview for review.
- **CI/CD:** Demonstrate how to set up and use a CI/CD pipeline to automate testing and deployment based on branches.

Extras

Below is a sample cylinder object you can use to mock your APIs:

```
{
  "id": "123",
  "name": "13",
  "price": 2500.00,
  "currency": "KES",
  "image_url": "https://example.com/images/13kg-cylinder.png"
}
```