

Part 4: Communication & Documentation

Overview of the Testing Process

The testing process was conducted in three key stages: manual testing, automated testing, and CI/CD pipeline integration.

1. Manual Testing: Initial bug identification and UI/UX evaluation were performed manually across different platforms (Android, iOS, Web).
 2. Automated Testing: Tests were written using the Flutter Integration Testing framework to cover two key scenarios:
 - Adding a new item to the list.
 - Deleting an existing item from the list.
 3. CI/CD Pipeline Setup: A CI/CD pipeline was implemented using GitHub Actions to automate linting, testing, and build verification for Android and iOS.
-

Bugs Found and Their Severity

1. Item Persistence (Severity: High)
 - Description: Items added or deleted do not persist across sessions.
 - Reproduction Steps:
 1. Open the app.
 2. Add an item.
 3. Restart the app and observe that the item is not present.
 - Platforms Affected: Android, iOS.
2. Title Field Error Handling (Severity: Medium)
 - Description: No user feedback (e.g., toast or Snackbar) is provided if the title field is left empty when adding a new item.
 - Reproduction Steps:
 1. Navigate to the "Add Item" screen.
 2. Leave the title field empty and tap "Add Item."
 3. No visual feedback is given to the user.
 - Platforms Affected: iOS, Web, Android.
3. Outdated Flutter Version (Severity: Medium)
 - Description: The Flutter version initially used was outdated, leading to issues when integrating the testing framework.
 - Resolution: Updated to Flutter 3.24.3 to resolve dependency issues.
4. iOS Compatibility Issues (Severity: High)

- Description: The app was initially only configured for Android and was not functional on iOS.
 - Resolution: iOS configuration was added.
-

Suggestions for Performance Improvements and UI/UX Enhancements

1. Performance Enhancements:
 - Caching Dependencies: In the CI/CD pipeline, consider caching the `pubspec` dependencies to speed up build times.
 2. UI/UX Enhancements:
 - User Feedback: Implement a Snackbar or Toast to inform the user when items are successfully added or deleted.
-

Overall Test Coverage and Areas Not Covered

- Test Coverage:
 - The automated tests covered the critical functionalities of adding and deleting items.
 - Static analysis and linting checks were executed successfully.
 - Builds for both Android and iOS were verified through GitHub Actions.
 - Areas Not Covered:
 - Persistence tests (e.g., saving data across app sessions) were not implemented due to time constraints.
 - Testing on a broader range of device sizes and web compatibility was not extensively covered due to limited resources.
-

Deliverables:

1. GitHub Actions Configuration:
 - The full `.yaml` configuration for setting up the CI/CD pipeline is included (refer to Part 3: GitHub CI/CD Pipeline Setup).
2. Test Execution:
 - The pipeline successfully ran linting, testing, and build verification.
 - Output logs from linting, automated tests, and build steps can be seen in the GitHub Actions workflow logs (refer to Part 3: GitHub CI/CD Pipeline Setup).

