

1. Automating Code Quality and Security Checks
  - a. For our project Savar, ensuring high code quality and robust security is important. The selected success criterion for automation within the CI pipeline will be to maintain and enhance these two aspects: code quality and security.
2. Why These Criteria?
  - a. Maintaining high code quality is essential for ensuring that the app is maintainable, scalable, and bug-free. Given that the app deals with personal data in diary entries, security is equally critical to protect user information from vulnerabilities, especially when integrating with third-party services for recipe recommendations.
3. Travis CI
  - a. To integrate these criteria into our Travis CI pipeline, we'll configure the pipeline to run static code analysis and security scans every time changes are pushed to the repository.
4. Adding a Test to the Repository
  - a. Given the functionality of the app, a simple yet effective test to add would be an integration test for the diary entry feature. This test will ensure that the app can successfully create, retrieve, and store diary entries as expected.
5. Integration Test Details
  - a. The test will simulate user interaction with the diary feature from inputting a new diary entry to saving and retrieving it from the backend. This will not only test the UI components but also the integration with the backend services, ensuring the end-to-end functionality works seamlessly.
6. Setting Up a Software Quality Assurance Tool
  - a. We are considering integrating SonarQube into our development pipeline due to its comprehensive capabilities to detect bugs, vulnerabilities, and code smells. SonarQube also offers detailed code quality histories and statistics, making it an invaluable tool for ongoing improvement in our project. SonarQube supports multiple languages and integrates well with Travis CI, making it an ideal choice for our React Native application. It provides both the overview and the depth of analysis required to maintain high standards for our codebase. Its security scanning capabilities will also be pivotal in identifying and stopping potential security issues early in the development cycle.

By integrating these elements—automated code quality and security checks, an integration test for critical functionality, and a robust tool like SonarQube for continuous quality and security assessment—we aim to enhance the reliability and security of the Savar app. These steps will ensure that the application not only meets its functional requirements but also adheres to high standards of quality and security, crucial for user trust and satisfaction.