

Test Plan – "To-Do List" Application

1. Objective

Ensure that the "To-Do List" application meets functional and non-functional requirements, including correct task management (create, read, update, delete), usability, performance, and compatibility.

2. Test Scope

Main functionalities of the application:

- Create tasks.
- Edit tasks. (the application includes this functionality)
- Delete tasks.
- Mark tasks as completed.
- List all tasks.
- Filters (pending/completed).
- Compatibility check on different browsers or devices (if applicable).
- Basic performance evaluation under light load.

3. Types of Tests to Perform

Functional Testing: Validate all user actions related to task management available in the application.

Regression Testing: Confirm that functionalities existing prior to modifications remain operational after changes.

Usability Testing: Evaluate ease of use and navigation.

Performance Testing: Measure response times when creating, listing, and deleting multiple tasks.

Compatibility Testing: Verify behavior across different browsers/devices.

4. Deliverables

- Documented test cases.
- Execution evidence (screenshots, reports).
- Incident reports.

5. Testing Strategy

a. Functional Tests:

- - Create a task with valid data.
- Attempt to create a task without data (validation of possible errors).
- Edit the description of an existing task (if the application supports it).
- Delete a specific task.
- Mark a pending task as completed.
- List completed/pending tasks.

b. Regression Tests:

- - Re-execute creation, editing, and deletion tests after updates or fixes.

c. Usability Tests:

- - Is adding a new task intuitive?
 - Are the action buttons (create, edit, delete) visible and understandable?
 - Do the displayed titles match the specified typography?
 - Are the objects presented according to the provided mockups?

d. Performance Tests:

- - Create 100 tasks and measure the list load time.
 - Mark 50 tasks as completed and measure the response time.

e. Compatibility Tests:

- - Execute tests on different devices since it is a mobile app.
 - Verify maximum and minimum version support on main operating systems.
 - Validate minimum coverage across all devices between the minimum and maximum supported versions.
 - Verify installation and download of the application (validation of access channels to the app).

6. Test Environment

- Device: PC, Laptop, Android/iOS Mobile (as applicable).
- Network: Standard Wi-Fi.

7. Risks

- Lack of access to repositories for downloading the respective applications.
- Lack of a Mac for automation execution in Xcode.
- Possible compilation errors when simulating in Xcode and Android Studio.
- No official documentation available for validating business rules and basic functionalities.
- Lack of physical devices for executing manual and automated tests.
- No defined flows for the main functionalities of the application.
- Extended time needed for test planning and case creation due to the QA team not being involved from the early development stages.
- Insufficient basic onboarding for the QA team by the DEV team, causing lack of knowledge of business rules and main functionalities.

8. Tentative Schedule

Activity	Estimated Time
-----	-----
Test case design	4 hours

Test execution	8 hours
Results reporting	2 hours