



Test plan for TODOist project

version 1.0



Revision and Signoff Sheet

Revision History

Version	Date	Changed By	Reason for change
0.1	28.10.2024	Nataliia Shubina	Creation

Approved list

Name	Role	Approved/Reviewer	Approval/Reviewer Date

Reference documents

#	Name	Link
1	Users guide	https://todoist.com/help



CONTENT

1. INTRODUCTION	4
1.1. Purpose	4
1.2. Project Overview	4
2. TEST ITEMS	5
2.1. Features that will be tested.....	5
2.2. Features out scope	5
3. REQUIREMENTS FOR TEST	6
4. TEST STRATEGY	6
4.1. Test Objectives	6
4.2. Test types	6
4.3. Test coverage.....	11
4.4. Activities workflow	11
4.5. Entry criteria.....	11
4.6. Exit criteria.....	12
4.7. Test Deliverables.....	12
5. ROLES AND RESPONSIBILITIES	13
6. RISKS.....	13
7. PROJECT MILESTONES	14
8. RESOURCES	15
8.1. Test tools.....	15
8.2. Test Environment	16



1. INTRODUCTION

1.1. Purpose

The objectives of the test plan are to define the test strategy, test approach and scope, establish input and output criteria, create a schedule, and assess risks. The document aims to identify the necessary resources, tools, roles and provides an estimate of the testing effort.

This document can increase the likelihood that the program will work properly under all circumstances

1.2. Project Overview

TODOist is a task management application designed to help users organize personal and professional tasks efficiently. It features an intuitive interface for adding, editing, and prioritizing tasks. Users can create projects and sub-projects to categorize their workload effectively. The app allows setting due dates, reminders, and recurring tasks for ongoing activities.

In the TODOist app, users can do the following:

- CRUD operations with tasks and sub-tasks;
- Prioritize tasks;
- Use labels and filters;
- Organize projects;
- Search tasks or projects;
- Set Due Dates and Recurring Tasks
- Get notifications and Reminders;
- Integration with applications (Google calendar, Slack, etc.);
- Track-Progress and etc.



2. TEST ITEMS

This paragraph will define the test scope to be performed within the test plan.
Will be tested the TODOist application item.

Testing will be based on the TODOist documentation (user guide).

2.1. Features that will be tested

Testing will cover the following features:

- User authentication (Sign Up, Login, Forgot Password, Log Out);
- CRUD operations with tasks;
- Sub-tasks;
- Task prioritization;
- Due Dates;
- Recurring Tasks;
- Filters tasks;
- Labels;
- Projects;
- Integration with applications.

2.2. Features out scope

- Profile settings;
- Notifications;
- Premium features;
- Settings features;
- Track tasks;
- Activity logs;
- Operation of third-party services.



3. REQUIREMENTS FOR TEST

The application must meet user requirements for activities related to creating new tasks, modifying them, adding labels and priorities, creating subtasks, organizing into projects, and integrating with other applications. The application should be easy to use and intuitive for users.

4. TEST STRATEGY

Test strategy aims to outline the principles that direct test design and oversee the software testing process. The TODOist project will be tested using a "black box" approach, focusing on functionality without access to the internal structure or source code.

4.1. Test Objectives

The objective of the test is to verify that the functionality of TODOist application works according to the requirements, identify and prevent defects, improve user experience.

The test will execute and validate the test cases, identify, resolve, and retest all high and medium severity defects in accordance with the entrance criteria, while prioritizing lower severity defects for future resolution through Change Requests (CR).

4.2. Test types

1. Functional testing

Functional testing is performed to check whether the software meets the specified requirements and whether it performs all the necessary functions. This testing focuses on verifying the behaviour of the system from the user's perspective, ensuring that it is correct and conforms to specifications.



Test Objective	Ensure the proper quality of the target functionality.
Technique	Execute each use case, use-case flow, or function, using valid and invalid data, to verify the following: <ul style="list-style-type: none">- The expected results occur when valid data is used.- The appropriate error or warning messages are displayed when invalid data is used.- Each business rule is properly applied.
Completion Criteria	<ul style="list-style-type: none">- All planned tests have been executed.- All identified defects have been addressed.

2. Usability testing

This type used to check whether the GUI corresponds to a project UI standards and user experience. The purpose of Graphical User Interface (GUI) Testing is to ensure the functionalities of software application work as per specifications by checking screens and controls like menus, buttons, icons, etc.

Test Objective	Verify the following: <ul style="list-style-type: none">- Adaptability. The website should correct adapts to different screen sizes;- Window objects and characteristics, such as menus, size, position, and focus conform to standards;- Website is consistent when it comes to colors, font types, and other visual elements;- Spelling errors throughout the application;- Behavior of interactive elements;- Functional validation.
----------------	--



Technique	Create or modify tests for each window to verify proper navigation, spelling, validation objects, adaptability and behaviour of interactive elements for each application window and object.
Completion Criteria	Each window successfully verified to remain consistent with benchmark version or within acceptable standard

3. Stress Testing

Stress testing is defined as the process of testing the hardware or software for its stability under a heavy load condition. This testing is done to find the numerical point when the system will break (in terms of a number of the users and server requests etc.) and the related error handling for the same.

Test Objective	Verify that the website functions properly and without error under the following stress conditions: <ul style="list-style-type: none">- Multiple users performing the same transactions against the same data- Maximum actual or physically capable number of clients connected or simulated
Technique	For stress tests, multiple clients should be used, either running the same tests or complementary tests to produce the worst case transaction volume or mix.
Completion Criteria	All planned tests are executed and specified system limits are reached or exceeded without the software failing or conditions under which system failure occurs outside of the specified conditions.



4. Installation testing

Performed for check is performed to check the correctness of the software installation process.

Test Objective	Verify that the software application installs correctly and functions as intended across various environments and configurations.
Technique	Checking the installation process for completeness, ensuring that all components are properly installed, validating that the application runs without errors post-installation, and confirming that uninstallation procedures work effectively.
Completion Criteria	Successful installation across all targeted environments without errors, ensuring all application components function as intended. Additionally, updates and rollbacks must work correctly, and uninstallation should leave no residual files, with accurate documentation provided throughout the process..

5. Smoke testing

Smoke testing is a confirmation for the QA team to proceed with further software testing. It ensures that all critical functionalities are working correctly or not.

Test Objective	To quickly check that the main features of the application are working properly after a new build, ensuring it's stable enough for more detailed testing.
Technique	Using basic functionality checks: Verifying that key features work as expected without going into detailed testing



Completion Criteria	All critical functionalities pass the basic checks, indicating that the application is stable enough for further testing without any major issue
---------------------	--

6. Regression testing

Performed to verify that new components, defect fixes integrated into existing systems are working as planned and already existing functionality wasn't broken

Test Objective	Ensure that recent code changes have not adversely affected existing functionalities and that the application continues to perform as expected across a
Technique	Focus on the most critical features and areas of the application that are likely to be impacted by recent changes, ensuring that any defects are identified quickly.
Completion Criteria	All selected test cases are executed successfully, with no critical or high-severity defects found in existing functionalities. Additionally, the application must perform as expected across all relevant features, demonstrating that recent changes have not introduced new issu

7. Confirmation testing

Performed to verify that failed tests from previous runs are passed successfully

Test Objective	Verify that previously identified defects have been successfully fixed and that the related functionality now works as intended.
Technique	Use re-testing to verify that previously identified defects



	have been fixed by running the same test cases that uncovered them.
Completion Criteria	All identified defects have been successfully resolved, and the corresponding test cases pass without errors.

4.3. Test coverage

All new functionality and improvements to existing functionality included in the scope of integration should be covered by Smoke, UI, Functional, and regression tests.

4.4. Activities workflow

The testing process will consist of the following activities united into stages:

Stage 1: Analysis of all existing software requirements and creating a test plan.

Stage 2. Develop test cases and test scripts based on requirements and specifications.

Stage 3. Prepare the testing environment, including hardware, software, and network configurations.

Stage 4. Detailed run of tests with the detection and description of defects.

Stage 5. Checking the bugs solved by the developers and conducting regression testing.

4.5. Entry criteria

Testing can begin only when the specified entry criteria are met. This ensures that all the necessary preparations are in place for an effective testing process.

List of entry criteria:

- Scope of work defined and described;



- Scope is prioritized;
- Availability of necessary test data and resources (environments, platforms, versions, systems);
- Finalized and approved test plans and test cases.
- Acceptance criteria defined.

4.6. Exit criteria

Testing can conclude only when the defined exit criteria are satisfied. This ensures that all necessary evaluations have been completed and that the product is ready for release. List of exit criteria.

- All functional and non-functional requirement have been tested.
- All planned tests have been executed.
- All critical and major issues have been resolved.
- There are no significant problems remaining.
- All found issues are reported in Jira.

4.7. Test Deliverables

The project has the following test documentation:

1. Test plan;
2. Checklists;
3. Test cases;
4. Bug reports;
5. Test result reports.



5. ROLES AND RESPONSIBILITIES

Human Resources	
Worker	Responsibilities
Project Manager	<ul style="list-style-type: none">- Monitor the progress of the project;- Identify and resolve issues that arise.
Business Analyst	<ul style="list-style-type: none">- Gathers requirements;- Ensures alignment between stakeholders and testing efforts.
Full Stack developer	<ul style="list-style-type: none">- Implement user logic.- Assure that all user input is validated before submitting to back-end- Building and maintaining web applications;- Writing high-quality code for project functional;- Managing database;- Troubleshooting and debugging
Test Lead	<ul style="list-style-type: none">- Coordinates testing activities;- Manages the test team, and ensures quality.
Test Engineer	<ul style="list-style-type: none">- Create test plan;- Review software requirements;- Preparing test scenarios;- Design test cases- Execute tests;- Analyzing test results;- Evaluates results- Preparing reports on all aspects related to the software testing

6. RISKS

Risk 1. The customer can change the basic requirements for one of the testing features. **Actions:** Add buffer time to the schedule for contingencies.

Risk 2: The customer has reduced costs for developing the project. **Actions:** Review all tasks, remove some functionalities, and change priorities for tasks.



Risk 3: More time spent on review than expected. **Actions:** Add buffer time to the schedule for contingencies.

Risk 4: Hardware failure during testing. **Actions:** Prepare backup resources.

Risk 5: Incomplete or unclear requirements. **Actions:** Conduct regular meetings with stakeholders to clarify requirements.

Risk 6: High defect rate found during testing. **Actions:** Increase collaboration with development teams for early issue resolution.

7. PROJECT MILESTONES

Milestone task	Effort	Start Date	End Date
Review Requirements documents	8h	31.10.2024	31.10.2024
Creating test plan	12h	01.10.2024	02.10.2024
Design Tests	40h	-	-
Test Environment Setup	5h		
Executing Tests	30h	-	-
Create bug reports	20h		
Retesting	15h		
Regression testing	40h	-	-
Performance testing	8h	-	-
Test Closure	15h	-	-
Post-Launch Support	ongoing		



8. RESOURCES

8.1. Test tools

The following tools will be employed for this project:

Title	Tools
Test-management tool	Jira
Defect Tracking System	Jira
Test documentations	Google Docs, Google Table
Collaboration tools	Discord
Performance testing	JMeter
Screen Recorder and Capture	ShareX
GUI testing	Browser DevTools; Page Ruler (extension)
API testing	Postman

The testing will be conducted on a variety of devices to ensure compatibility and performance across different environments. This includes:

Name	Operation System
iPhone 7, 7+, to latest	from iOS 10 to latest
Samsung, Xiaomi, and others with required OS	Android, from v.8 to latest
Tablets	iOS from 10 to latest; Android, from v.8 to latest



8.2. Test Environment

Testing will be conducted in the latest two versions of Google Chrome, Safari, Mozilla Firefox, Opera, and Microsoft Edge.

The environment for Functional Testing and Regression testing is ***Staging***: <url>.

The environment for Acceptance Testing is ***Production***: <url>

If any additional configurations are required to support specific functionalities or iterations, they will be communicated prior to iteration planning.