

Testing types for **todoist** desktop app

Created by
solomiia pryshliak

softserve

Agenda

- Application overview
- Smoke testing
- Functional testing
- Non-functional testing
- Confirmation testing
- Regression testing

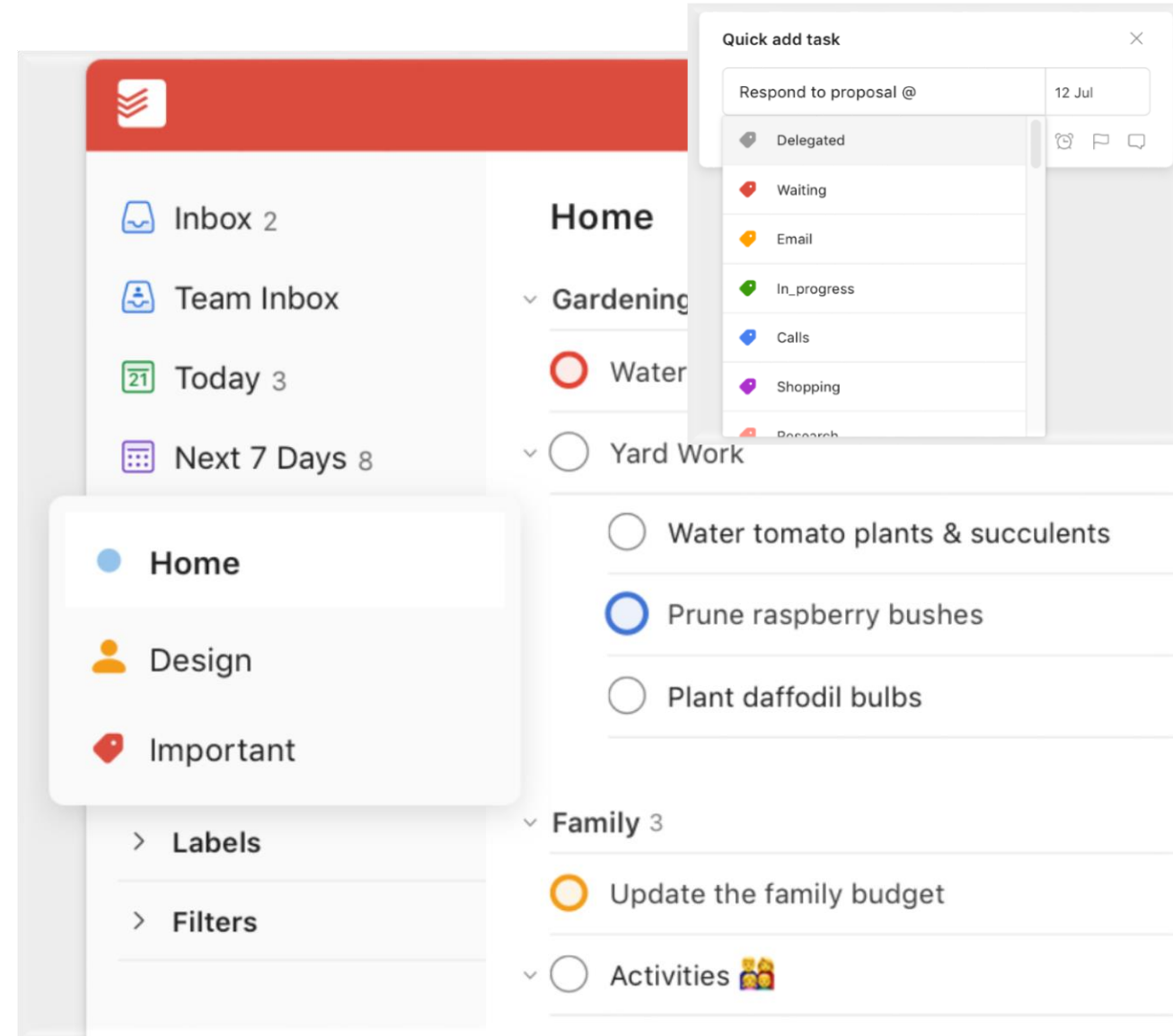


The application overview

Todoist is the task manager, web service, and software suite for project management and task management

The main features:

1. Create tasks and sub-tasks
2. Set reminders
3. Organize tasks into projects
4. Add labels and priorities
5. Collaborate with others
6. Track progress



Smoke testing

Task: Test new version of the Todoist application (Desktop v. 9879 Windows)

Smoke testing procedure: Verify that the most critical functionalities (running the application, creating a task, editing a task, marking task as complete) are working

SCENERIO	STATUS
The app can be launched without any errors	✓
The user can create a new task and add it to the task list	✓
User can edit a task (change the due date or priority level)	✓
User can mark a task as complete	✓

PASS

softserve

Functional testing #1

Requirement: users should be able to create new projects to organize their tasks.

Task: verify that the Project creation feature works as expected.

Preconditions:

1. The Todoist app is installed
2. The user is logged in with a valid account

Steps:

1. Hover the cursor over "Projects" tab in the navigation menu
2. Click on the "+" icon
3. Enter a name for the new project in the "Name" field
4. Click on the "Add" button

Expected result: The new project is created and displayed in the list of projects and in the list of "Favorites".

PASS

Add project

Name

New project

Favorites

• New project

Projects

• New project

☒ List ☐ Board

View is synced between teammates in shared projects. [Learn more.](#)

Cancel Add

Functional testing #2

Requirement: users should be able to use the search function to locate specific tasks.

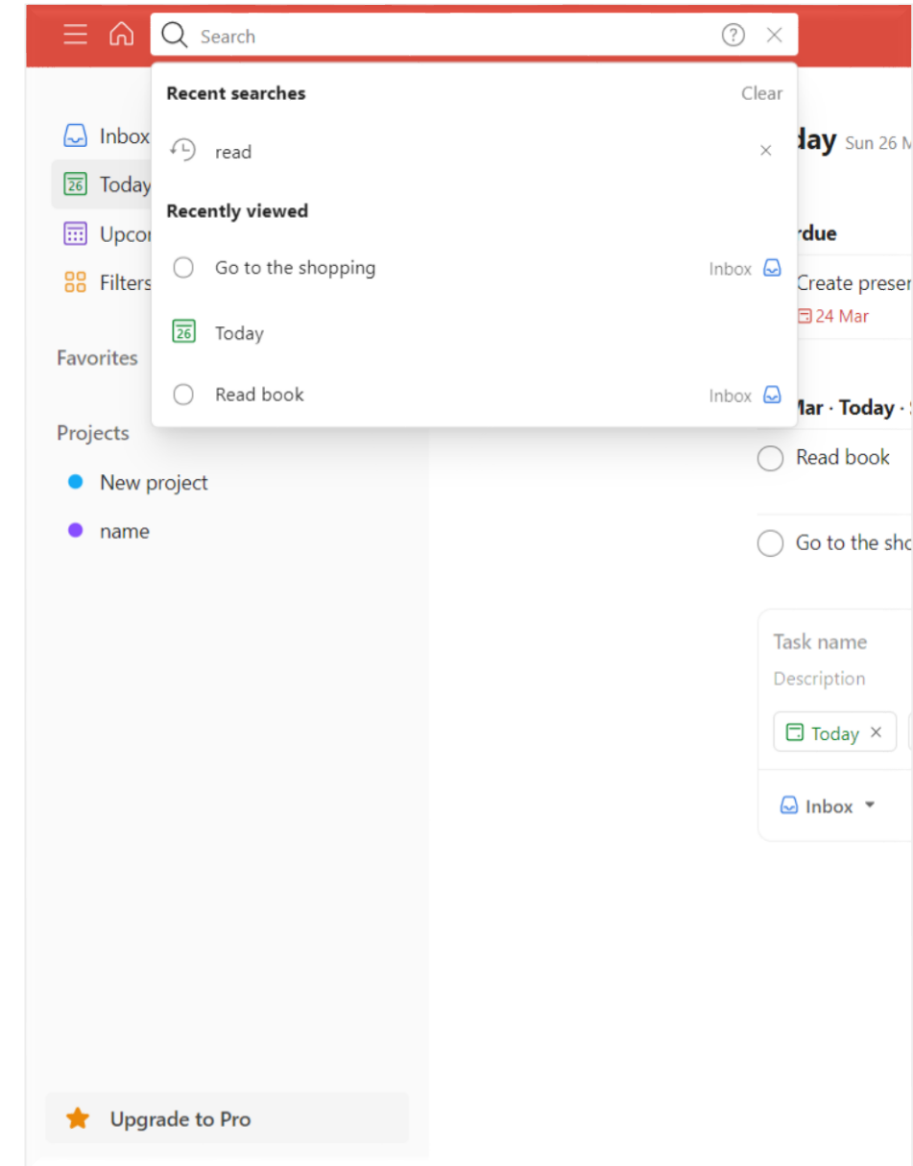
Task: verify the functionality of the search bar in Todoist app

Preconditions:

1. The Todoist app is installed and logged in with a valid account
2. Several tasks are added to the account

Steps:

1. Click on the search bar to activate it.
2. Enter a search query for a specific task or keyword related to a task.
3. Click on the "X" button next to the search bar to close the search window.
4. Click on the search bar once again
5. Click the button "clear " in the "recent searches" drop-down menu.



Functional testing #2 (Bug report)

Summary: The "Recent" searches drop-down menu results are not cleared after clicking on the "Clear" button

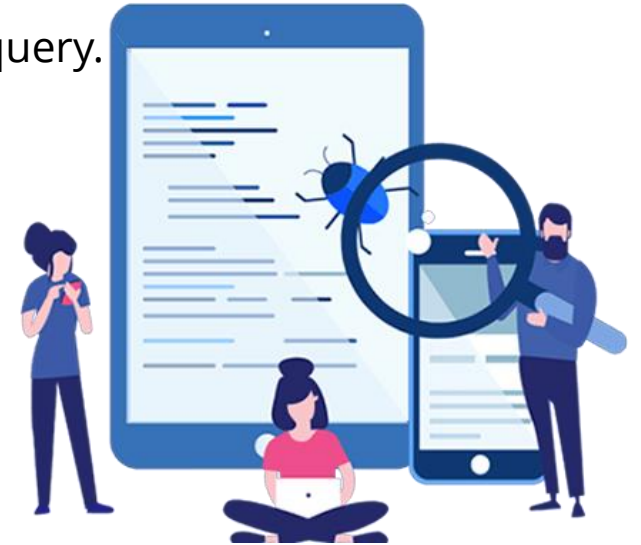
Actual Result: system does not clear the "Recent searches" dropdown menu by clicking on the "Clear" button

Expected Results: the app should display relevant results for the entered search query. Search window could be closed by clicking on the "X" button. The recent searches drop-down menu could be cleared by clicking on the "Clear" button

Priority: Medium

Severity: Low

FAIL



softserve

Nonfunctional testing



softserve

Compatibility testing

Requirement: the users should be able to create and edit tasks in the Todoist from Google Calendar and vice versa, and see the results of their actions on both platforms

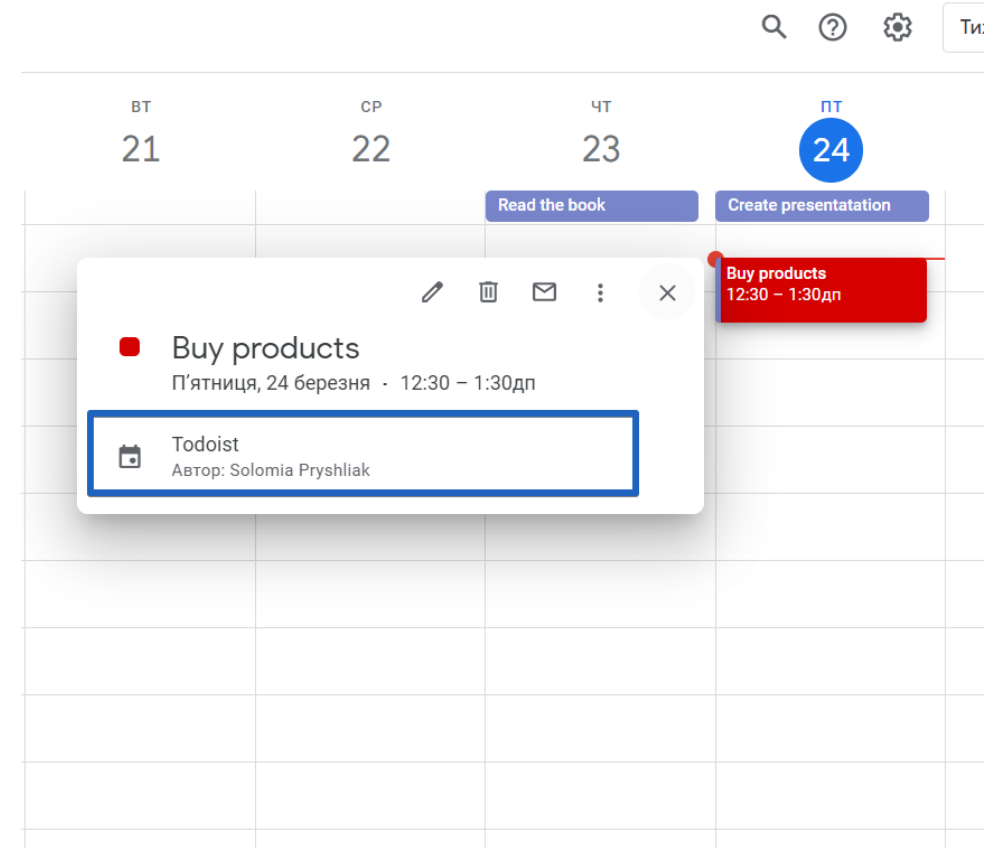
Task: verify that the Todoist desktop app is compatible with Google Calendar

Preconditions:

1. The Todoist desktop app is installed.
2. A Google Calendar account is set up.

Steps:

1. Open the Todoist app and navigate to the "Settings" menu.
2. Click on "Integrations" and select "Google Calendar."
3. Log in to the Google Calendar account.
4. Test the app's ability to create new tasks in Google Calendar from the Todoist app.
5. Test the app's ability to update tasks in Google Calendar from the Todoist app.



Compatibility testing (Bug report)

Summary: changes made with tasks from Todoist are not reflected in the calendar

Preconditions:

1. The Todoist desktop app is installed.
2. A Google Calendar account is set up.

Steps to reproduce:

1. Open the Todoist app and navigate to the "Settings" menu.
2. Click on "Integrations" and select "Google Calendar."
3. Test the app's ability to create new tasks in Google Calendar from the Todoist app.
4. Test the app's ability to update tasks in Google Calendar from the Todoist app.

Actual result: user is unable to edit tasks in Google Calendar from Todoist app.

Expected result: the app is able to create, update, edit and retrieve tasks in the Todoist app from Google Calendar and vice versa without any errors or delays. The synchronization of data between Google Calendar and the Todoist app is consistent and up to date.

Severity: medium

Priority: low

Environment: Windows 11



softserve

Portability testing

Requirement: the app should be portable across different versions of Windows operating systems

Task: test the compatibility of the Todoist desktop app in various versions of Windows operating systems

PASS

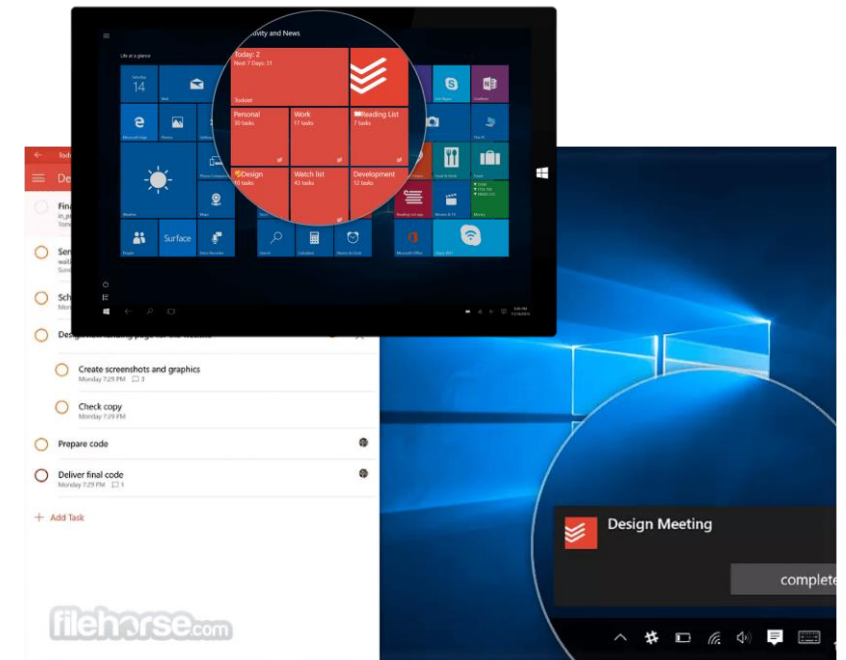
Preconditions:

1. The Todoist desktop program is downloaded and installed on a Windows machine
2. Access to different versions of Windows operating systems (Windows 7, Windows 8, Windows 10, Windows 11)

Steps:

1. Install and open Todoist on a machine running Windows 7
2. Conduct a smoke testing
3. Repeat Step 1 and 3 with Windows 8, Windows 10 and Windows 11

Expected result: the app works seamlessly and smoothly across different versions of Windows operating systems.



Localization testing

Requirement: users should be able to change localization of the application. All UI elements have to be translated.

Task: verify that all UI controls, content, and functionality of the app are translated when the user changes the language from English to German.

Preconditions:

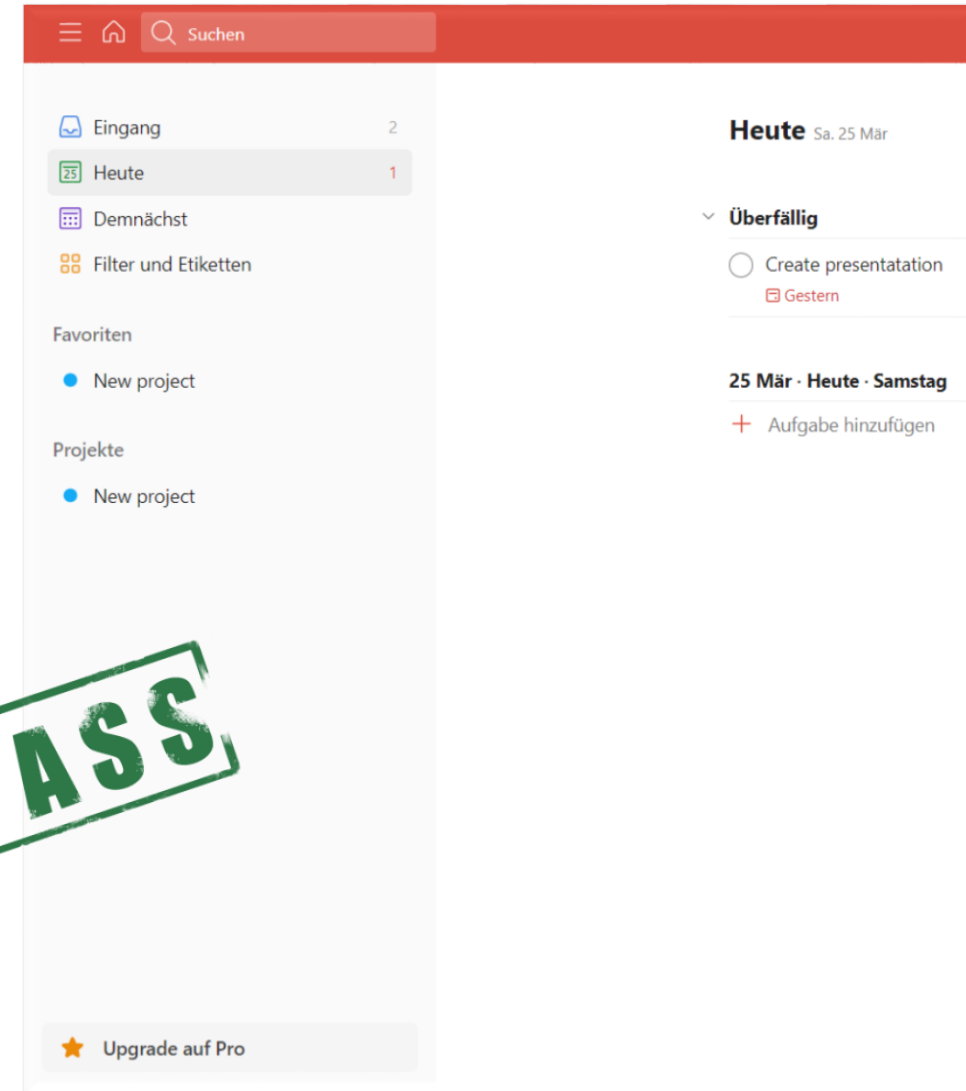
1. The Todoist app is installed and set to the English language.
2. The user is logged in and has a few tasks added to the account.

Steps:

1. Navigate to the General settings page.
2. Change the language setting from English to German.
3. Check all labels and captions whether they are translated into German, localized and not truncated

Expected result: the app's content, including task descriptions, help pages, error messages, pop-ups and settings, are translated into German

PASS



softserve

Accessibility testing

Requirement: Todoist desktop app should be available for the users using screen readers.

Task: verify that basic functions can be performed by users who rely on screen readers.

Preconditions:

1. The Todoist app is installed and launched
2. A screen reader is installed and enabled on the computer (e.g. Narrator, JAWS, NVDA).

Steps:

1. Using the screen reader, navigate to the "Add Task" button
2. Create a new task by entering a task name and pressing the enter key
3. Using the screen reader, edit the task by changing the task name
4. Using the screen reader, navigate to the checkbox next to the task you just edited
5. Mark the task as completed by checking the checkbox.

Expected result: the screen reader reads out all the relevant information. The user is able to perform all the basic functions (creating a new task, editing a task, and marking it as completed) using the screen reader. The program responds appropriately to user inputs



UI testing

1.	Alignment	The alignment of all UI controls matches the mockup in terms of the position and size.	✓
2.	Typography	The font family, font size, and font weight are consistent with the mockup	✓
6.	Colors	The color scheme used in the page matches the mockup	✓
3.	Spelling	Spelling of all displayed elements does not contain errors	✓
4.	Resizing	All elements adjust to changes in window sizes	✓
5.	Controls	All required controls are present	✓
6.	Tabulation	Tabulation is consistent and properly aligned	✓

PASS

softserve

Performance testing

Requirement: the app should handle a load of 3000 concurrent users with an average response time of 3 seconds per request.

Preconditions:

1. The Todoist app is installed and running.
2. JMeter is installed and configured

Steps:

1. Define the user actions that will be used (creating, completing and editing a task)
2. Set the load level of 3000 concurrent users.
3. Start the performance test and monitor the response time and resource utilization of the app.

Expected results: the Todoist app can handle a load of 3000 concurrent users with an average response time of no more than 3 seconds per request.



PASS

Load testing

Requirement: the app should handle a load of 3000 concurrent users with an average response time of 3 seconds per request.

Task: To measure the response time of the app while increasing the user load

Preconditions:

1. The Todoist app is installed and running.
2. JMeter is installed and configured

Steps:

1. Define the user actions that will be used (creating, completing and editing a task)
2. Set the load level of 3000 concurrent users and gradually increase it to (4000, 5000, 6000).
3. Start the load test and monitor the response time and resource utilization of the app.

Expected results: the Todoist app can handle a load of 300-6000 concurrent users with an average response time of no more than 3 seconds per request.



softserve

Stress testing

Requirement: the app should handle a load of 3000 concurrent users with an average response time of 3 seconds per request.

Task: Verify the load-handling capacity of the Todoist app by simulating a high volume of concurrent users and tasks.

Preconditions:

1. The Todoist app is installed
2. Stress testing software is installed available

Steps:

1. Define the user actions that will be used (creating, completing and editing a task)
2. Emulate amount of request more than the allowed value: 3001, 3010, 3500 and 4000 concurrent users.
3. Start the load test and monitor the response time and resource utilization of the app.

Expected results: the server isn't crashed due to overload and any data isn't lost.

PASS



Installation/uninstallation testing

Requirement: user should be able to install and uninstall the app properly on the desktop computer

Task: ensure that the app installation/ installation process can be complete without any errors

Preconditions:

1. Available Todoist Desktop app package file.
2. Available desktop computer with supported operating system.
3. Sufficient storage space available on the computer.



Steps:

1. Once the installation process is completed, launch the app and check if it opens successfully without any errors. Make a smoke testing.
2. Verify that the Todoist app is located on the computer and displayed in the list of application
3. Complete the uninstallation process and check if the app is removed from the desktop completely (with all the files written on the disk)

Confirmation testing



softserve

Confirmation testing

Task: re-running tests for founded bugs in 1.3.1. build to ensure that these bugs were fixed and the 1.3.2. build of the application works according to the requirements.

Defect ID	Summary	Status
D0001	The "Recent" searches drop-down menu results are not cleared after clicking on the "Clear" button	Fixed
D0002	Changes made with tasks from Todoist are not reflected in the calendar	Reopened

Regression testing



softserve

Regression testing

Requirement: verify that the application still meets requirements after fixing bug D0001 and these changes don't impact on other functionality such way, that new bugs could appears in the 1.3.2. build .

Tasks:

- Verify the login functionality
- Verify that the user can create, edit and mark a new task as done
- Test the search functionality by entering a keyword in the search bar and verifying that relevant tasks and projects are displayed.
- Test the search functionality with labels by applying different labels to tasks and then searching for tasks with specific labels.
- Test the search history functionality by entering multiple search queries and then using the "Recent" searches drop-down menu to quickly access previous search queries.
- Test the «X» button in the search field to verify that the search window can be closed.

Result: regression testing have done successful, bugs in the any other feature or of any type of testing weren't found.



softserve

