

Software Test Plan - STP

"Gmail"

Avihai, Gitit, Ariel Dotan and Yiftach
Production environment
9/11/2023

Version Control

CURRENT VERSION

Title	Software Test Plan - STP
Author	Avihai, Gitit, Ariel Dotan and Yiftach

Table of Contents

1	3
1.1	4
1.2	4
1.3	5
1.4	5
2	5
2.1	6
2.2	Error! Bookmark not defined.
2.3	7
2.4	9
3	Error! Bookmark not defined.
3.1	Error! Bookmark not defined.
3.2	10
3.2.1	Error! Bookmark not defined.
3.3	Error! Bookmark not defined.
3.3.1	Error! Bookmark not defined.

1 Document Overview

1.1 Introduction

This document serves as the Software Test Plan for <"Gmail" website> <production environment>.

The purpose of this STP is to define the framework and Strategy for the testing of "Gmail" web site.

The plan is tailored to support the Agile Scrum methodology, emphasizing on flexibility, and iterative development.

Our objective is to validate the High Quality of "Gmail" web site.

We will verify "Gmail" web site behaves as expected by testing its features and functionality.

In alignment with Scrum principles, this document will try to stay as short and focused on Testing needs so it could be easily updated and evolve throughout project iterations.

1.2 Objectives

At a high level The primary objectives of this Software Test Plan for gmail are as follows:

✓ **Ensure Product Quality:**

To uphold the high standards of quality for which Gmail is known, verifying that all features work as intended and meet user and business requirements.

✓ **Enable Efficient Development Cycles:**

To align testing activities with Scrum sprints, facilitating swift identification and resolution of defects, and supporting the development team in quick iterations.

✓ **Support Business Goals:**

To ensure that the testing process aligns with the overarching business objectives, contributing to the sustained success and growth of Gmail.

1.3 Scope

- The scope of this document is only for version production environment of "Gmail" product.
- This STP won't include the Test Planning and Test Execution of "Gmail" on the following OS: Linux, MacOS

1.4 References

<If applicable you can list here any reference you have about the specification of the product like tutorials / User Manuals / SRS etc'.

In case there's none you can state that No references were available e.g. "N/A">

No	Document Title	File Name (Path) / HyperLink
1	N/A	
2		
3		

2 Scope of testing

2.1 *Features to be tested*

Here you'll state all the Modules Features you plan to test.

<Note that because it's an evolving document some features/Modules could be added/deleted while the project is in process depends on timetables and complexity>

- Sending Emails- Gitit
- Receiving emails - Gitit
- Inbox settings - Gitit
- Connection Status - Avihai
- Last activity Status - Avihai
- Add contacts- Avihai
- Offline settings - Avihai
- synchronization of Emails with the Gmail app - Ariel
- theme in settings - Ariel
- labels in settings- Ariel
- Display settings - Ariel
- Chat- Yiftach
- Spaces - Yiftach
- general settings - Yiftach

2.2 *Features not to be tested*

- Help and guidance
- Garbage folder / permanent delete
- add-ons
- pop/Imap
- accounts and import
- filters and blocked addresses
- Chat settings
- Integration with other websites (connect with google)
- Grabage & permanent delete
- Search and filters

2.3 Testing Types

Outlined below are the test types that will be planned and performed during this project:

- **Functionality Verification:**

To ensure all features of Gmail, such as query input, search execution, Filters, and tabs, send and receive Emails as intended, All settings work correctly, Chat, Spaces and Meet work as the SRS explains.

- **Usability Assessment:**

To evaluate the user interface for intuitiveness, ease of use, and accessibility. This includes ensuring the home page is easily navigable and that the interface elements are responsive to user interactions.

- **Compatibility Testing:**

To confirm that Gmail works seamlessly across different browsers (e.g., Chrome, Firefox, Edge), operating systems (Windows), and devices (desktops, tablets, smartphones).

- **Localization and Internationalization Verification:**

To ensure that Gmail provides accurate results and a user-friendly experience in different languages and regions.

- **Search Result Accuracy:**

To validate the relevancy and accuracy of search results provided by the search algorithms.

This includes testing the effectiveness of filters and the ranking of search results.

-
- **Smoke Testing**
 - **load Functionality:**
 - **Check if the login page is accessible**
 - **Sign Up Functionality**
 - **verify the user can Sign up to Gmail**
 - **Login Functionality:**
 - Verify that users can log in with valid credentials.
 - **Inbox Access:**
 - Check if the inbox loads properly after login.
 - Ensure that new emails appear without delay.
 - Verify that the email list displays correctly.
 - **Compose Email:**
 - Confirm that users can open the compose email window.
 - Check if sending an email is successful.
 - Verify the email appears in the sent folder.
 - **Receive Email:**
 - Send a test email from another account and verify its receipt.
 - Ensure that received emails are displayed correctly in the inbox.
 - **Basic Navigation:**
 - Verify that various sections (Inbox, Sent, Drafts, Spam, Trash) are accessible and functional.
 - Ensure that the "Search" bar works to find emails.
 - **Log Out:**
 - Confirm users can log out successfully.

- **User Interface Testing**
- 1. Maximum Page Size:
 - Ensure the email client can display a specified maximum number of items per page.
 - Verify smooth scrolling and responsive layout.
 - Confirm clear indicators for reaching the maximum page size.
- 2. Page Navigation:
 - Test "Next Page" and "Back" functionality.
 - Verify correct button states and smooth transitions.
 - Ensure accurate email display during navigation.
- 3. Email Search:
 - Test keyword, sender, and subject search.
 - Check for accurate and responsive search results.
 - Assess performance with a large mailbox.
- 4. Word Combinations and Filtering:
 - Test filtering by criteria and complex word combinations.
 - Confirm easy configuration and performance with a high email volume.
- 5. Message Marking / Email Tagging:
 - Test marking/tagging for the organization.
 - Verify marking options and visual distinctions.
 - Assess management efficiency.

- **Error handling**

Check how the Website responds when making expected / unexpected errors.

<List here all the testing types that you plan to use in this Project>

Smoke

sanity

GUI

Boundary

Compatibilly

2.4 Test Strategy and Approach

Our test approach is systematic and structured to ensure thorough and efficient validation of each build received from the Development team.

The following outlines our planned testing progression for each release cycle:

Initial Build Assessment with Smoke Testing:

Upon receipt of a new build, the Quality Assurance (QA) team will execute a Smoke Testing Suite.

This suite is designed to quickly check the stability of the build and ensure that the core functionalities of Gmail are operating as expected.

Only after a build passes the smoke test will it move forward in the testing process.

Focused Testing on New Features and Bug Fixes with Sanity Testing:

After the build has passed the Smoke Testing phase, the QA team will proceed to Sanity Testing.

This phase is targeted at the new features and bug fixes included in the release.

The objective is to ensure that specific updates are functioning correctly in the application without any immediate issues.

Comprehensive Regression Testing:

Following the Sanity Testing phase, comprehensive Regression Testing will be conducted.

This is critical to ensure that new code changes have not adversely affected existing functionalities of Gmail.

The Regression Testing will be extensive and is designed to cover all areas of the application that could potentially be impacted by the changes.

Incorporation of Exploratory Testing:

Parallel to the structured testing phases, we allocate approximately 20% of the total testing effort during the execution phase for Exploratory Testing.

This approach allows testers to go beyond predefined test cases and scenarios, using their insights and experience to uncover issues that may not have been anticipated in the test planning stages.

Iterative Feedback and Continuous Integration:

The testing strategy is aligned with the Agile Scrum framework, which advocates for continuous integration and iterative feedback.

Testing phases will be tightly integrated with the sprint cycles, ensuring prompt feedback to the Development team and allowing for quick iteration and refinement of the application.

The proposed testing approach ensures a balance between structured testing and the flexibility to discover unforeseen issues, making it highly effective in an Agile development environment.

By following this approach, the QA team contributes to the delivery of a stable, high-quality product that meets the rigorous standards expected of Gmail.