**Google - GMAIL| TEST STRATEGY**

**Table of contents**

[**1 Context of the testing**](#_mj84l6vzpzo) **2**

[1.1 Test objectives](#_t7pte9f9zqyc) 2

[1.3 Test scope](#_hdqhbprlvk4n) 3

[1.3.1 In scope](#_cr4o077d2zhw) 3

[1.3.2 Out of scope](#_ait5xe6mc99u) 4

[**2 Test approach**](#_orvjm87z8r0p) **5**

[2.1 Test automation](#_10nra6yhvt64) 5

[2.2 Test tools](#_fsiwa1luoeka) 6

2.3 Test Levels

[2.4 Test environment requirements](#_jtc2pwty5oaw) 8

[2.5 Test data requirements](#_l3wdyejw8pru) 8

[**3 Test management**](#_evf935qcnjrm) **8**

[3.1 Stakeholders](#_rv5qpldicamp) 9

[3.2 Test artefacts](#_kr9x7fash4s0) 9

[3.4 Defect Management Process](#_4o01cxwf63je) 10

[3.6 Test completion criteria](#_x6xhh7wv8csp) 10

[3.8 Reporting](#_13wsh62eeohv) 11

[3.9 Assumptions, constraints and risks](#_8ah68l5g2mv) 12

# **1 Context of the testing**

**The test strategy document involves a systematic approach to ensure the reliability, security, and optimal user experience of this critical communication service for the Google users.**

**This section highlights the objectives and dimensions for testing to be carried on the GMAIL platform to ensure the coverage and quality of the product.**

## **1.1 Test objectives**

## **Functional Testing:**

* + Conduct thorough functional testing covering core email functionalities:
  + Include testing for special features such as priority inbox, smart replies, and conversation threading.

## **Non-functional Testing:**

* + performance testing to assess the responsiveness and scalability of GMAIL, especially during peak usage periods.
  + Security testing with a focus on email privacy, encryption, and protection against phishing attacks.
  + Usability testing to ensure an intuitive and user-friendly interface.
  + Accessibility testing to ensure that GMAIL is accessible to users with disabilities.

## **Compatibility Testing:**

* + Validate compatibility across various browsers (Chrome, Firefox, Safari, Edge) and operating systems (Windows, macOS, Linux).
  + Test GMAIL on different devices, including desktops, laptops, tablets, and smartphones.

## **Integration Testing:**

* + Verify the integration of GMAIL with other Google services (Google Drive, Google Calendar, Google Meet) and third-party applications.

## **User Experience Testing:**

* + Assess the overall user experience, including the responsiveness of the user interface, navigation, and the consistency of features across platforms.
  + Test the efficiency of search functionalities and filters to help users manage their emails effectively.
  + Hold User research groups for testing the user experience on the app using eye tracking technology on real people recruited according to the target audience specified by the product owner to get intuitive and precise results.

## **1.3 Test scope**

**This section identifies features that are in scope (covered by testing) and features that are out of scope (not covered by testing). The scope determines the boundaries of testing efforts and ensures that resources are allocated effectively.**

### **1.3.1 In scope**

* - **Email Composition and Sending:**
  + Testing the functionality of composing emails
  + Adding attachments
  + Formatting options
  + Sending emails.
* **- Email Receiving and Reading:**
  + Verifying the correct receipt
  + Display of emails
  + Email rendering
  + Email threading and grouping.
* - **Attachments and File Handling:**
  + Testing the attachment functionality, ensuring that users can attach files, view attachments, and download them.
* - **Email Filters and Labels:**
  + Testing the creation, application, and functionality of filters
  + Labels for organising and categorising emails.
* **- Search Functionality:**
  + Verifying the effectiveness and accuracy of the search functionality.
* **- Integration with Other Google Services:**
  + Testing the seamless integration with other Google services, such as Google Calendar, Google Drive, and Google Meet.
* - **Security Features:**
  + Conducting security testing for email transmission, encryption, and protection against phishing attacks.
* - **User Authentication:**
  + Verifying the user authentication process
  + Login
  + Password recovery
  + Multi-factor authentication.
* **- Performance Testing:**
  + Assessing the responsiveness and performance of GMAIL, especially under various load conditions.
* **- Usability:**
  + Testing the user interface for consistency, clarity, and accessibility for users with disabilities.
* **- Accessibility:** 
  + Validating the accessibility for users with disabilities against the WCAG 2.2 guidelines.

### **1.3.2 Out of scope**

* **- Third-Party Integrations:**
  + Testing integrations with third-party applications or services that are not officially supported by GMAIL.
* - **Non-GMAIL Email Services:**
  + Testing the functionality of GMAIL with email services from providers other than Google.
* **- Hardware-Specific Testing:**
  + Testing GMAIL on specific hardware devices that are not part of the supported list.
* **- Browser-Specific Features:**
  + Testing features that are specific to a particular browser if it is not listed as a supported browser.
* **- Offline Mode:**
  + Testing features related to offline mode if it is not a supported or advertised feature.
* **- Legal and Compliance Testing:**
  + Testing for legal and compliance requirements that fall outside the standard functional and security testing scope.
* **- Advanced Settings and Customizations:**
  + Testing advanced settings or customizations that are not part of the standard GMAIL user experience.
* **- Beta Features:**
  + Testing features labelled as experimental or beta unless explicitly included in the testing scope.

### **- Localization and Globalization Testing:**

* + To assess the software's adaptability to different languages, regions, and cultural settings. (To be carried by the designated localisation team)

### 

# **2 Test approach**

The test approach defines, at a high level, the approach for testing based on information available during the project definition phase.

## **2.1 Test automation**

This section defines the level of test automation to be undertaken by the test effort and how any automation will be approached.

* **Regression Testing:**
  + Regression test suites for critical functionalities, ensuring that existing features remain intact with each new release.
* **Data-Driven Testing:**
  + Cover the variety of scenarios, such as testing with different types of attachments, email content, and user profiles.
* **Continuous Integration:**
  + Automate tests of the continuous integration process to provide quick feedback on code changes and detect issues early in the development cycle.

## **2.2 Test tools**

This section states the combination of tools to address various testing aspects, including functional testing, performance testing, security testing, and automation testing.

* **Test Automation:**
  + Selenium WebDriver:
  + Use Selenium WebDriver for browser automation to perform end-to-end testing of Gmail's web interface.
  + Implement test scripts in Java language.
  + Appium:
  + For testing Gmail mobile applications across Android and iOS platforms.
  + TestNG:
  + For structuring and organising automated test scripts.
* **Performance Testing:**
  + JMeter:
  + To conduct performance testing on Gmail, simulating various user scenarios and load conditions.
  + Analyze response times, throughput, and server performance under different levels of concurrent users.
* **Security Testing:**
  + Burp Suite:
  + Use for manual and automated security testing, especially during the development and testing of new features.
  + Perform thorough security assessments and penetration testing.
* **Test Management and Collaboration:**
  + Jira and Zephyr:
  + Utilize Jira for test case management, tracking, and collaboration.
  + Integrate Jira with Zephyr to create a seamless workflow for test execution, issue tracking, and reporting.
  + Confluence:
  + Use Confluence for creating and maintaining test documentation, including test plans, test cases, and test strategy documents.
* **Continuous Integration/Continuous Delivery (CI/CD):**
  + GitLab:
  + Automate build and test processes for efficient development workflows.
* **Accessibility Testing Tools:**
  + Axe Accessibility Checker:
  + Integrate axe Accessibility Checker to automate accessibility testing.
* **Cross-Browser Testing Tools:**
  + BrowserStack:
  + Use BrowserStack to perform cross-browser testing of Gmail.
* **Collaboration Tools:**
  + Slack:
  + Use Slack for real-time communication within the testing team.

**2.3 Test levels**

### **Unit Testing:** To verify that individual units or components of the software perform as designed.

* + To be conducted by developers.
  + Focuses on isolated modules or functions.
  + Validates correctness at a code level.
  + Automated testing tools to be used.

### **Integration Testing:** To test the interactions between integrated components and ensure that they work together seamlessly.

* + To be conducted after unit testing.
  + To be conducted with both manual and automated testing.

### **System Testing:** To evaluate the entire system as a whole, ensuring it meets specified requirements and functions correctly.

* + Follows integration testing.
  + Tests functional and non-functional requirements.
  + Verifies system behavior in different environments.
  + To be conducted with manual and automated testing.

### **User Acceptance Testing:** To assess whether the system meets business requirements and is ready for deployment.

* + Conducted by end-users or stakeholders. (Involves the user research sessions)
  + Validates if the system satisfies acceptance criteria.

### **Regression Testing:** To ensure that new changes or enhancements do not negatively impact existing functionality.

* + To be conducted with every new added feature or bug fixes release.
  + To be automated to save time and effort.

## **2.4 Test environment requirements**

This section defines the test environment needs of the test effort whether existing test environments will be sufficient and suitable. New or additional test environments will need to be included in the project budget.

* **Supported Browsers:**
* Google Chrome: The primary browser for Gmail. Test across different versions.
* Mozilla Firefox, Safari, Microsoft Edge
* **Supported Operating Systems:**
* Windows, macOS
* **Test Environment Configurations:**
* Multiple Test Environments: Set up multiple test environments in isolation from the development environments to facilitate parallel testing and avoid conflicts.

## **2.5 Test data requirements**

* **Realistic User Profiles:** Create test accounts with different user profiles to cover a variety of usage scenarios.
* **Backup Email Accounts:** Create backup email accounts for testing recovery and continuity scenarios.
* **Different User Roles:** Simulate different user roles (e.g., administrators, standard users) to test role-based access control.
* **Permission Settings:** Verify that user permissions are configured correctly.

# **3 Test management**

Test management describes how progress of the test effort will be monitored, controlled, measured and communicated.

## **3.1 Stakeholders**

|  |  |
| --- | --- |
| **Role** | **Team member Name** |
| Project Manager |  |
| Product Owner |  |
| Test lead/Senior |  |
| Test Engineer |  |
| Software Lead |  |
| Software Engineer |  |
| UX Designer |  |

## **3.2 Test artefacts**

Testing artefacts are documents or deliverables that the testing team is responsible to create and add to the project document repository.

* **Test Plan:** Outlines the overall testing strategy, objectives, scope, schedule, resources, and deliverables.

### **Test Case Specification:** Provides detailed information on individual test cases, including inputs, expected outcomes, and execution steps.

### **Test Scripts:** A set of instructions written in a scripting language for automated testing.

### **Defect Report:** Documents information about defects or issues identified during testing.

### **Test Summary Report:** Summarises the testing activities, including test results, statistics, and overall assessment.

### **Traceability Matrix:** Maps requirements to test cases to ensure that all requirements are covered by test cases.

## **3.4 Defect Management Process**

This section defines the defect management processes to be followed during the test effort

* **Defect Identification:** Testers identify defects during different testing phases, covering functional, usability, performance, and security issues.
* **Defect Logging:** Testers document detailed information about each defect in a defect tracking system, including ID, description, steps to reproduce, severity, and priority.
* **Defect Classification and Prioritization:** Defects are classified based on their nature and assigned severity levels. Prioritization is done based on impact.
* **Defect Assignment:** The responsible team member, often a developer, is assigned to review and resolve the defect.
* **Defect Resolution:** Developers work on fixing defects by making necessary code changes, with potential collaboration between testers and developers.
* **Defect Verification:** Testers verify fixed defects to ensure successful resolution by following documented steps for reproduction.
* **Defect Reopening (If Necessary):** Defects may be reopened if issues persist or new problems arise, restarting the defect lifecycle.
* **Defect Closure:** Successfully verified and resolved defects are marked as closed in the tracking system, forming part of the final documentation.
* **Defect Metrics and Reporting:** Metrics, including defect numbers and trends, are collected and reported to assess the system's health.

## **3.6 Test completion criteria**

This section specifies specific conditions or requirements that must be met to consider a testing phase or the entire testing process as complete.

* **Test Execution Coverage:** A minimum of 90% of high priority test cases must be executed, covering both functional and non-functional aspects of the system.
* **Defect Closure:** All critical and high-severity defects must be addressed and closed. A maximum of 10% of lower-severity defects may also be acceptable.
* **Test Case Pass Rate:** A minimum of 90% high priority test cases must pass successfully without critical defects.
* **Regression Testing Success:** Testing team must confirm that new changes have not introduced critical defects or negatively impacted existing functionality.
* **Test Logs and Reports:** Test logs and reports, including defect reports, test execution logs, and summary reports, must be reviewed by the Test lead and finalized.
* **Stakeholder Approval:** Test lead and Project manager must review and approve the test results, indicating their satisfaction with the testing outcomes.

## **Reporting**

This section specifies the purpose and frequency of test reporting.

* **Test Execution Status:** Overview of test case execution progress, including the number of test cases executed, passed, failed, and pending.
* **Defect Status:** Summary of open and closed defects, including their severity, priority, and status.
* **Performance Metrics:** Results of performance testing, including response times, throughput, and resource utilization.
* **Security Testing Summary:** Overview of security testing results, highlighting any vulnerabilities identified.
* **Usability Testing Feedback:** Summary of usability testing findings and user experience insights.
* **Regression Testing Results:** Confirmation of whether regression testing was successful and any issues discovered.

**Frequency of Reporting:**

* **Regular Reporting:** Daily updates during active testing phases to provide real-time insights in Standup meetings.
* **Milestone Reporting:** At the completion of each sprint.
* **Release Reporting:** Before the major releases, providing a comprehensive overview of testing outcomes.
* **Ad-Hoc Reporting:** As needed, in response to critical issues, unexpected events, or stakeholder requests.

## **3.9 Assumptions, constraints and risks**

This section to be used to log any assumptions, constraints or risks / issues associated with the test effort, including actions planned to mitigate each risk / issue.

|  |  |
| --- | --- |
| **Ref** | **Assumption / Constraint** |
|  |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref** | **Risk / Issue** | **Likelihood** | **Mitigating Action** |
|  |  |  |  |
|  |  |  |  |

**Appendix A**

**Table 1**

|  |  |  |
| --- | --- | --- |
| **Column 1** | **Column 2** | **Column 3** |
| **Item 1** | Item 2 | Item 3 |
| **Item 1** | Item 2 | Item 3 |
| **Item 1** | Item 2 | Item 3 |

**Appendix B Related documents**

**Further details are available in the following related documents:**

|  |
| --- |
| **DOCUMENT** |
| Test Plan |
|  |

**Appendix R Acronyms, terms and definitions**

**The following jargon is used within this document.**

|  |  |
| --- | --- |
| **TERM** | **DEFINITION** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |