Table of Contents

[Purpose 2](#_Toc170065232)

[Project Overview 2](#_Toc170065233)

[2 Scope 2](#_Toc170065234)

[In-Scope 2](#_Toc170065235)

[Out-of-Scope 2](#_Toc170065236)

[3 Testing Strategy 3](#_Toc170065237)

[3.1 Test Objectives 3](#_Toc170065238)

[3.2 Test Assumptions 3](#_Toc170065239)

[3.3 Data Approach 3](#_Toc170065240)

[3.4 Testing types 3](#_Toc170065241)

[3.4.1 Functional Testing 4](#_Toc170065242)

[3.4.2 non-Functional Testing 4](#_Toc170065243)

[Performance Testing: 4](#_Toc170065244)

[Security Testing: 4](#_Toc170065245)

[Usability Testing: 5](#_Toc170065246)

[3.4.3 White-box Testing 5](#_Toc170065247)

[3.5 testing levels 6](#_Toc170065248)

[3.5.1 Unit Testing 6](#_Toc170065249)

[3.5.2 User Acceptance Testing 6](#_Toc170065250)

[Overall User Experience: 6](#_Toc170065251)

[Security and Privacy: 7](#_Toc170065252)

[3.5.3 Regression Testing 7](#_Toc170065253)

[4 Execution Strategy 8](#_Toc170065254)

[Entry Criteria 8](#_Toc170065255)

[Exit criteria 9](#_Toc170065256)

[5 Validation and Defect Management 10](#_Toc170065257)

[6 Environment Requirements 11](#_Toc170065258)

[Test Environments 11](#_Toc170065259)

[7 Dependencies 11](#_Toc170065260)

1. **Introduction**

## Purpose

To ensure the core functionalities, user interface, security, and performance of Gmail are thoroughly tested and meet the required standards.

## Project Overview

This project provides users with the ability to send, receive, organize, and search emails. This project involves testing Gmail’s functionalities, UI consistency, security features, and performance to maintain its reliability and user satisfaction.

# 2 Scope

## In-Scope

**Functional Testing**: Email operations (send, receive, forward, delete), search functionality, labels, filters, attachments.

**UI Testing**: Layout consistency, accessibility, responsiveness.

**Security Testing**: Authentication, encryption, data protection.

**Performance Testing**: Load time, response time, scalability, volume.

## Out-of-Scope

Non-email related features like Google search, Google maps, YouTube, Google drive, Google translate.

Non included categories: Starred, Snoozed, Important, Scheduled, Drafts, Spam, Trash, Categories, Manage labels, Create new label.

# Testing Strategy

## 3.1 Test Objectives

* Verify all email operations work correctly and efficiently.
* Ensure the user interface is intuitive, consistent, and accessible across different devices and browsers.
* Confirm robust security measures are in place to protect user data.
* Assess performance to handle various loads and maintain fast response times.

## 3.2 Test Assumptions

* Gmail servers and infrastructure are operational and accessible during testing.
* Test environments accurately represent production environments.
* Users have stable internet connections for testing.

## 3.3 Data Approach

* Utilize a mix of synthetic and anonymized real data to simulate various email scenarios.
* Test with different attachment types, email formats, and user configurations to cover diverse use cases.

## 3.4 Testing types

List the types of testing to be performed.

|  |  |  |
| --- | --- | --- |
| **Test Type** | **Description** | **Responsible Parties** |
| Functional Testing | Validate core email operations and functionalities | QA Team |
| Non-Functional Testing | Assess performance, security, and usability aspects. ￼ | QA Team, Security Team, Performance Team |
| White-box Testing | Evaluate internal structures, algorithms, and code paths. | Development Team |
| Change-related Testing | Verify the impact of changes on existing functionalities. | QA Team, Development Team |

## 3.4.1 Functional Testing

**Email Operations:**

Sending emails, receiving emails, forwarding emails, Deleting emails.

**Search Functionality:**

Accuracy of search results, previous searches.

**Filters:**

Creation of filters. Application of filters.

**Attachments:**

Uploading attachments. Downloading attachments. Virus scanning for attachments.

**Participants:**

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| Bahaa | QA | QA Tester |
| Shibel | QA | QA Tester |

## 3.4.2 non-Functional Testing

# Performance Testing:

* **Load Time**: Measure the time taken for the inbox and individual emails to load under various conditions.
* **Response Time**: Assess the responsiveness of the interface, especially during peak usage times.
* **Scalability**: Evaluate how well the Gmail web app handles increasing numbers of users and larger volumes of emails.

# Security Testing:

* **Authentication**: Verify the security of login and logout processes, ensuring strong password policies and multi-factor authentication.
* **Encryption**: Ensure that emails and attachments are encrypted both in transit and at rest.
* **Data Protection**: Confirm that user data is protected against unauthorized access and breaches.
* **Vulnerability Scanning**: Conduct regular scans to identify and address potential security vulnerabilities.

# Usability Testing:

* **User Interface**: Evaluate the intuitiveness, accessibility, and overall user experience of the Gmail web app.
* **Accessibility**: Ensure compliance with accessibility standards (e.g., WCAG) to make the web app usable for individuals with disabilities.
* **Consistency**: Verify that the layout and design are consistent across different browsers and devices.
* **User Feedback**: Collect and analyze feedback from users to identify areas for improvement.

## 3.4.3 White-box Testing

**Code Review:**

* **Code Analysis**: Perform a thorough review of the code to ensure it adheres to best practices and coding standards.
* **Algorithm Efficiency**: Evaluate the efficiency of algorithms used in email operations, search functionality, and data handling.
* **Error Handling**: Verify that the code includes proper error handling mechanisms to manage exceptions and unexpected inputs.

**Code Coverage:**

* **Test Coverage**: Measure the extent to which the code is exercised by the tests, aiming for high coverage to minimize untested paths.
* **Path Coverage**: Ensure that all possible paths through the code are tested, including edge cases and error conditions.

**Database-Related Tests:**

* **Schema Validation**: Ensure that the database schema is correctly defined and adheres to the required design standards.
* **Data Integrity**: Verify that data is correctly stored, retrieved, and maintained without corruption or loss.
* **Performance Testing**: Evaluate the performance of database queries, ensuring they execute efficiently under various load conditions.
* **Backup and Recovery**: Test the backup and recovery procedures to ensure data can be restored accurately and promptly in case of failure.

## 3.5 testing levels

## 

## 3.5.1 Unit Testing

* Email sending functionality.
* Email receiving functionality.
* Email forwarding functionality.
* Email deletion functionality.
* Attachment handling.
* Advanced search filters creation and application.

**Participants:**

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| Bahaa | Development | Software Engineer |
| Shibel | Development | Software Engineer |

## 3.5.2 User Acceptance Testing

# Overall User Experience:

Ease of composing and sending emails. Clarity of inbox organization (filters). Effectiveness of search functionality. User interaction with attachments.

# Security and Privacy:

Account login/logout. Handling of sensitive information. Data protection measures.

**Performance:** Load time for inbox and individual emails. Responsiveness of interface during heavy usage. Stability under varying network conditions.

Participants:

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| Bahaa | QA | QA Tester |
| Shibel | QA | QA Tester |

## 3.5.3 Regression Testing

## 

**Features to be Tested:**

1. **Core Email Operations:** Sending, receiving, forwarding, and deleting emails.
2. **Search Functionality:** Accuracy and speed of search results.
3. **Filters:** Creation and application of filters.
4. **Attachments:** Uploading and downloading attachments. Virus scanning for attachments. **Interface Consistency:** Layout and design elements across different devices and browsers.

**Participants:**

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| Bahaa | QA | QA Tester |
| Shibel | QA | QA Tester |

# 4 Execution Strategy

## Entry Criteria

* The entry criteria for initiating test execution for Gmail are as follows:

|  |  |
| --- | --- |
| **Entry Criteria** | **Notes** |
| Test environment(s) is available | Ensure that the test environment, including Gmail's staging environment, is set up and accessible for testing. |
| Test data is available | Verify that the necessary test data, including sample emails, attachments, and user configurations, is prepared and accessible for testing scenarios. |
| Code has been merged successfully | Confirm that all code changes related to the Gmail application have been successfully merged into the testing environment. |
| Development has completed unit testing | Ensure that all individual components of the Gmail application have undergone unit testing and are ready for integration testing. |
| Test scripts are completed, reviewed and approved by the Project Team | Validate that all test cases and scripts related to Gmail testing have been developed, reviewed, and approved by the project team. |

## Exit criteria

* *The exit criteria are the desirable conditions that need to be met to proceed with the implementation.*
* *Exit criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions and provide a recommendation.*

|  |  |
| --- | --- |
| **Exit Criteria** | **Notes** |
| 100% Test Scripts executed | All test scripts have been executed without any skipped or unresolved scripts. |
| 90% pass rate of Test Scripts | Ensure that at least 90% of the executed test scripts have passed successfully, allowing for minor exceptions. |
| No open Critical and High severity defects | Verify that all critical and high severity defects have been addressed and closed. |
| All remaining defects are either cancelled or documented as Change Requests for a future release | Ensure that any remaining defects are either resolved or documented for future consideration. |
| All expected and actual results are captured and documented with the test script | Confirm that all test results, including expected and actual outcomes, are documented alongside the test scripts. |
| All test metrics collected based on reports from daily and Weekly Status reports | Gather all relevant test metrics from daily and weekly status reports for analysis and reporting. |
| All defects logged in Defect Tracker/Spreadsheet | Ensure that all identified defects are logged and tracked in the designated defect tracking system or spreadsheet. |
| Test environment cleanup completed and a new back up of the environment | Perform a thorough cleanup of the test environment and create a new backup to ensure readiness for future testing phases. |

## 5 Validation and Defect Management

**Validation of Test Cases / Test Scenarios:**

* Test cases/test scenarios should be validated by executing them according to the defined test procedures.
* Test results should be compared against expected outcomes to ensure that the Gmail application behaves as expected.

**Defect Management:**

* Defects found during testing will be tracked using a Defect Tracker or Spreadsheet.
* Testers are responsible for opening defects, retesting to verify fixes, and closing defects once they are resolved.
* Defects will be categorized based on severity and impact as follows:

|  |  |
| --- | --- |
| **Severity** | **Impact** |
| 1 (Critical) | * Functionality is blocked and no testing can proceed * Application/program/feature is unusable in the current state |
| 2 (High) | * Functionality is not usable and there is no workaround, but testing can proceed |
| 3 (Medium) | * Functionality issues but there is workaround for achieving the desired functionality |
| 4 (Low) | * Unclear error message or cosmetic error which has minimum impact on product use. |

# Environment Requirements

## Test Environments

* **Staging Environment:** A replica of the production environment where testing will be conducted.
* **Access to Gmail Test Accounts:** Testers should have access to test accounts with various configurations to simulate different user scenarios.
* **Attachment Testing Environment:** A dedicated environment to test the handling of attachments in Gmail, including uploading, downloading, and virus scanning.
* **Compatible Browsers and Devices:** Ensure compatibility with major web browsers (Chrome, Firefox, Safari, Edge) and devices (desktop, tablet, mobile).
* **Internet Connectivity:** Stable internet connection to access Gmail services and perform testing activities.

# Dependencies

* **Test-Item Availability:** Availability of test items such as test environments, test data, and test scripts is crucial for testing to proceed smoothly.
* **Testing-Resource Availability:** Availability of testing resources including testers, tools, and equipment is necessary to execute tests effectively.
* **Deadlines:** Adherence to project deadlines and milestones is essential for coordinating testing activities within the project timeline.