

# QA Estimations

## Effort Estimation

The QA effort estimation for the CloudNotepad application is based on the complexity and scope of the application, the number of test cases, and the environments to be tested.

1. Requirement Analysis: 5 days
2. Test Planning: 3 days
3. Test Case Development: 10 days
4. Test Environment Setup: 3 days
5. Test Execution: 15 days
6. Defect Reporting and Retesting: 10 days
7. Test Closure: 2 days

## Resource Allocation

- Test Manager: 1 (20% of the time)
- Test Lead: 1 (50% of the time)
- Test Engineers: 3 (full time)

## Timeline

- Total estimated time: 48 days
- Resources: 1 Test Manager, 1 Test Lead, 3 Test Engineers

## Metric Selection

### Test Metrics

1. Test Case Preparation Status
  - Definition: Percentage of test cases prepared versus planned.
  - Formula:  $(\text{Number of test cases prepared} / \text{Total number of test cases planned}) * 100$
2. Test Execution Status
  - Definition: Percentage of test cases executed versus planned.
  - Formula:  $(\text{Number of test cases executed} / \text{Total number of test cases planned}) * 100$
3. Defect Density
  - Definition: Number of defects identified per module.
  - Formula:  $\text{Total number of defects} / \text{Size of the module (LOC or function points)}$
4. Defect Severity and Priority
  - Definition: Classification of defects by severity and priority.
  - Metrics: High, Medium, Low severity and priority levels
5. Test Coverage

- Definition: Percentage of requirements covered by test cases.
  - Formula:  $(\text{Number of requirements covered by test cases} / \text{Total number of requirements}) * 100$
6. Test Pass/Fail Rate
- Definition: Number of test cases passed versus failed.
  - Formula:  $(\text{Number of test cases passed} / \text{Total number of test cases executed}) * 100$
7. Test Effort
- Definition: Time spent on test case preparation and execution.
  - Formula: Total hours spent on testing activities

## Test Plan

### Testing Levels

1. Unit Testing
  - Performed by developers.
  - Ensures individual units work as expected.
2. Integration Testing
  - Verifies interfaces between components.
  - Ensures interactions between different parts of the application.
3. System Testing
  - Conducted on the complete, integrated system.
  - Evaluates compliance with specified requirements.
4. Acceptance Testing
  - Validation by end users.
  - Ensures the application meets user needs and requirements.

### Testing Strategies

1. Manual Testing
  - Exploratory testing.
  - Usability testing.
  - Specific test cases where automation is not feasible.
2. Automated Testing
  - Repetitive test cases.
  - Regression testing.
  - Performance testing.

### Test Items

- Note creation functionality.

- Note editing and deletion functionality.
- Note synchronization across different devices (Mobile, Web, Desktop).
- User authentication and authorization.
- Offline access and data synchronization once online.
- Data encryption and security.

## **Test Types**

1. Functional Testing
  - Verify each function of the application operates according to the requirements.
2. Usability Testing
  - Ensure the application is user-friendly and provides a good user experience.
3. Performance Testing
  - Test the application's responsiveness, stability, scalability, and speed under various conditions.
4. Compatibility Testing
  - Ensure the application works across different devices, operating systems, and browsers.
5. Security Testing
  - Identify vulnerabilities and ensure data protection.
6. Regression Testing
  - Ensure that new changes do not introduce new defects.

## **Test Environment**

1. Hardware Requirements
  - Mobile Devices: Android (various versions), iOS (various versions)
  - Desktops: Windows, macOS, Linux
  - Web Browsers: Chrome, Firefox, Safari, Edge
2. Software Requirements
  - Operating Systems: Latest versions of Windows, macOS, Linux, Android, and iOS
  - Browsers: Latest versions of Chrome, Firefox, Safari, Edge
  - Testing Tools: Selenium, Appium, JUnit, TestNG, JIRA, Jenkins

## **Test Deliverables**

- Test Plan Document
- Test Cases
- Test Scripts
- Test Data
- Test Summary Report
- Defect Report
- Traceability Matrix

Test Schedule

Milestones

- Test Planning: 2024-08-01 to 2024-08-05
- Test Case Development: 2024-08-06 to 2024-08-15
- Test Environment Setup: 2024-08-16 to 2024-08-18
- Test Execution: 2024-08-19 to 2024-08-30
- Test Closure: 2024-08-31 to 2024-09-02

Detailed Schedule

Activity	Start Date	End Date	Responsible
Test Planning	2024-08-01	2024-08-05	John Doe
Test Case Development	2024-08-06	2024-08-15	Jane Smith
Test Environment Setup	2024-08-16	2024-08-18	Alice Johnson
Test Execution	2024-08-19	2024-08-30	Michael Brown
Defect Reporting and Retesting	2024-08-21	2024-08-29	Emily Davis
Test Closure	2024-08-31	2024-09-02	Robert Wilson

Risk and Contingencies

Risks

- Unavailability of test environment.
- Delays in test case development.
- Incomplete requirements.
- Insufficient resources.

Mitigation Plans

- Maintain backup environments.
- Prioritize critical test cases.
- Regular communication with stakeholders.
- Allocate additional resources if needed.

Roles and Responsibilities

Role	Responsibilities
Test Manager	Planning, coordination, and management of test activities.
Test Lead	Overseeing test execution, reporting, and resource allocation.
Test Engineers	Designing, executing test cases, and reporting defects.
Developers	Fixing defects and assisting in test environment setup.
Business Analysts	Ensuring requirements are clear and testable.

# Test Strategy

## 1. Introduction

The purpose of this document is to outline the test strategy for the CloudNotepad application. CloudNotepad is a simple application designed for notes creation and working with non-formatted text, supporting environments such as Mobile, Web, and Desktop.

## 2. Objectives

The objective of this test strategy is to define the approach, resources, and schedule for the testing activities. It ensures that the application is thoroughly tested to meet the quality standards and functional requirements.

## 3. Scope

The scope of this test strategy includes:

- Functional and non-functional testing of the CloudNotepad application.
- Compatibility testing across supported environments (Mobile, Web, Desktop).
- Regression testing to ensure new changes do not adversely affect existing functionalities.

## 4. Testing Approach

### 4.1 Manual Testing

- Used for exploratory testing, usability testing, and specific cases where automation is not feasible.

### 4.2 Automated Testing

- Employed for repetitive test cases, regression testing, and performance testing.

## 5. Test Levels

### 1. Unit Testing

- Conducted by developers to test individual units of code.
- Tools: JUnit, TestNG.

### 2. Integration Testing

- Ensures that different modules or services interact correctly.
- Tools: JUnit, TestNG, Postman.

### 3. System Testing

- Comprehensive testing of the complete system against requirements.
- Tools: Selenium, Appium.

### 4. Acceptance Testing

- Validation by end-users to confirm the application meets their needs.

- Tools: Manual testing, UAT sessions.

## **6. Test Types**

1. **Functional Testing**
  - Verify each function of the application operates as expected.
2. **Usability Testing**
  - Ensure the application is user-friendly and provides a good user experience.
3. **Performance Testing**
  - Assess the responsiveness, stability, scalability, and speed under various conditions.
4. **Compatibility Testing**
  - Verify the application works across different devices, operating systems, and browsers.
5. **Security Testing**
  - Identify vulnerabilities and ensure data protection.
6. **Regression Testing**
  - Ensure new changes do not introduce new defects.

## **7. Test Environment**

### **7.1 Hardware Requirements**

- Mobile Devices: Various versions of Android and iOS.
- Desktops: Windows, macOS, Linux.
- Web Browsers: Chrome, Firefox, Safari, Edge.

### **7.2 Software Requirements**

- Operating Systems: Latest versions of Windows, macOS, Linux, Android, and iOS.
- Browsers: Latest versions of Chrome, Firefox, Safari, Edge.
- Testing Tools: Selenium, Appium, JUnit, TestNG, JIRA, Jenkins.

## **8. Test Tools**

1. **Selenium**
  - Automated web testing.
2. **Appium**
  - Automated mobile testing.
3. **JUnit/TestNG**
  - Unit and integration testing.
4. **JIRA**
  - Issue and project tracking.
5. **Jenkins**
  - Continuous integration and delivery.

## **9. Test Deliverables**

- Test Strategy Document
- Test Plan Document
- Test Cases
- Test Scripts
- Test Data
- Test Summary Report
- Defect Report
- Traceability Matrix

## 10. Risk and Mitigation

### 10.1 Risks

- Unavailability of test environment.
- Delays in test case development.
- Incomplete requirements.
- Insufficient resources.

### 10.2 Mitigation Plans

- Maintain backup environments.
- Prioritize critical test cases.
- Regular communication with stakeholders.
- Allocate additional resources if needed.

## 11. Roles and Responsibilities

Role	Responsibilities
Test Manager	Planning, coordination, and management of test activities.
Test Lead	Overseeing test execution, reporting, and resource allocation.
Test Engineers	Designing, executing test cases, and reporting defects.
Developers	Fixing defects and assisting in test environment setup.
Business Analysts	Ensuring requirements are clear and testable.

## Difference Between Test Plan and Test Strategy

### Test Plan

- **Purpose:** A document outlining the detailed approach for testing a specific product or release.
- **Content:** Includes test objectives, scope, resources, schedule, test environment, test cases, and test deliverables.
- **Level of Detail:** High level of detail specific to the project or release.
- **Audience:** Test team, project managers, developers, and stakeholders directly involved in the project.

Test Strategy

- **Purpose:** A high-level document describing the general approach and principles for testing across multiple projects.
- **Content:** Includes testing objectives, approach, testing levels, testing types, tools, environment, and risk management.
- **Level of Detail:** High-level overview, applicable across projects or releases.
- **Audience:** Senior management, QA managers, and stakeholders interested in the overall testing approach and standards.

Test Cases

Test Case ID	Scenario	Description	Test Steps	Prerequisite	Test Data	Expected Result	Status	Environment	Comments
TC001	User Registration	Verify that a new user can register successfully with a valid phone number.	1. Open the Telegram app. 2. Tap on "Start Messaging" 3. Enter a valid phone number and tap "Next." 4. Enter the OTP received via SMS. 5. Fill in the required personal details and tap "Next"	Telegram app is installed User not already registered	1. Phone number: +1234567890 2. OTP: 123456 3. Personal Details: a. Name - John Doe b. Username - JDoe	The user is successfully registered and navigated to the home screen	Pass	iOS 15 Telegram v10.14.2	Ensure network connectivity for OTP reception
TC002	User Registration	Verify that a new user cannot register with an invalid phone number.	1. Open the Telegram app. 2. Tap on "Start Messaging" 3. Enter and invalid phone number and tap "Next"	Telegram app is installed	1. Phone Number: 123	An error message is displayed indicating an invalid phone number	Pass	iOS 15 Telegram v10.14.2	Verify error message and content behaviour
TC003	User Login	Verify that a user can log in with the correct password	1. Open the Telegram app 2. Tap on "Start Messaging" 3. Enter the registered phone number. 4. Enter the correctly associated password	User Registered	1. Phone number: +1234567890 2. Password: password	The user is successfully logged in and navigated to the home screen	Pass	iOS 15 Telegram v10.14.2	Ensure network connectivity
TC004	User Login	Verify that the user cannot login with an incorrect password	5. Open the Telegram app 6. Tap on "Start Messaging" 7. Enter the registered phone number.	User Registered	1. Phone Number: +1234567890 2. Password: passw0rd	An error message is displayed indicating an incorrect password	Pass	iOS 15 Telegram v10.14.2	Verify error message and content behaviour



			8. Enter an incorrect password						
TC005	Messaging	Verify that the user can send a text message to a contact	1. Open a chat with a contact 2. Type a message in the text box 3. Tap the send button	User logged in, at least one contact available	1. Message Content: "Test"	The message is successfully send and appears in the chat	Pass	iOS 15 Telegram v10.14.2	Ensure contact is available for testing
TC006	Messaging	Verify that the user cannot send a text message without a network connection	1. Disable network connection 2. Open a chat with a contact 3. Type a message in the text box 4. Tap the send button	User logged in, at least one contact available	1. Message Content: "Test"	An error message is displayed indicating the absence of a network	Pass	iOS 15 Telegram v10.14.2	Ensure network is disabled
TC007	Media Upload	Verify that the user can upload and send an image.	1. Open a chat with a contact. 2. Tap the attachment icon. 3. Select an image from the gallery. 4. Tap the send button.	User logged in, at least one contact available	1. Image File: "sample_image.jpg"	The image is successfully uploaded and sent	Pass	iOS 15 Telegram v10.14.2	Ensure image file is accessible
TC008	Media Upload	Verify that the user cannot upload and send an image without a network connection.	1. Disable network connection. 2. Open a chat with a contact. 3. Tap the attachment icon. 4. Select an image from the gallery. 5. Tap the send button.	User logged in, at least one contact available	1. Image File: "sample_image.jpg"	An error message is displayed indicating the absence of a network connection, and the image is not uploaded or sent.	Pass	iOS 15 Telegram v10.14.2	Ensure network is disabled
TC009	Notification Settings	Verify that the user can change notification settings	1. Tap on the menu icon. Select "Settings." 2. Go to "Notifications and Sounds." 3. Change notification settings as desired.	User logged in	1. Notification Preferences: Enable or disable sound, vibration, and popup notifications.	The notification settings are successfully updated	Pass	iOS 15 Telegram v10.14.2	Check all notification options
TC010	Logout	Verify that user can logout successfully	1. Tap on the menu icon. Select "Settings." 3. Tap "Log Out."	User logged in	N/A	The user is logged out and redirected to the login screen	Pass	iOS 15 Telegram v10.14.2	Ensure proper logout behavior

Checklist

Functional Areas	Priority	Status	Tester	Jira #	Notes
User Registration	P1	OK	Matth er		
- Register with valid phone number	P1				
- Register with invalid phone number	P1				
- Register with already registered phone number	P1				
User Login	P1	OK			
- Login with correct password	P1				
- Login with incorrect password	P1	In Progress			
Messaging	P2				
- Send a text message	P2	OK			
- Send a text message without network	P2	OK	Mark		
- Send an image	P2				
- Send a video	P2				
- Send a document	P2				
- Send a voice message	P2				
Notifications	P3				
- Receive push notifications	P3				
- Disable notifications	P3				
- Customize notification sound	P3				
Contacts	P3				
- Sync contacts	P3	Problem		SW-1000	Syncing contacts that are no longer saved
- Add a new contact	P3				
- Block a contact	P3		Luke		
Groups	P2				

- Create a new group	P2				
- Add members to a group	P2				
- Remove members from a group	P2				
- Send a group message	P2				
Settings	P2				
- Change profile picture	P2				
- Change username	P2				
- Change password	P2				
- Enable two-step verification	P2				
Media Sharing	P2		John		
- Upload and send images	P2				
- Upload and send videos	P2				
- Upload and send documents	P2	OK			
- Upload and send voice messages	P2				
Privacy and Security	P1				
- Set privacy settings for last seen	P1				
- Set privacy settings for profile photo	P1				
- Enable two-factor authentication	P1				
- Set up a local passcode	P1				
Miscellaneous	P3				
- Logout from the app	P3				
- Delete account	P3				
- Clear chat history	P3				

A test case is a detailed document that provides specific steps, expected results, and test data for testing a particular scenario, making it suitable for complex and critical testing. In contrast, a checklist is a high-level list of items or tasks to verify, offering simplicity and speed but relying on the tester's knowledge. While test cases ensure thorough documentation and consistency, they require more effort to create and maintain. Checklists, however, are easier and quicker to update, making them ideal for repetitive or straightforward tasks.

## Errors/Defects/Failures

### Defect 1: Login Button Not Responding

- ID: D001
- Title: Login button does not respond when clicked

- Description: The login button on the login screen is unresponsive when clicked, preventing users from accessing their accounts.
- Severity: Critical
- Priority: High
- Steps to Reproduce:
  1. Open the Telegram app.
  2. Enter valid credentials.
  3. Click on the Login button.
- Expected Result: The user should be logged in and redirected to the home screen.
- Actual Result: The button does not respond, and the user remains on the login screen.

#### Defect 2: App Crashes on Sending Media

- ID: D002
- Title: Application crashes when sending a media file
- Description: The Telegram app crashes when the user tries to send a media file in a chat.
- Severity: Major
- Priority: High
- Steps to Reproduce:
  1. Open a chat.
  2. Tap the attachment icon.
  3. Select a media file.
  4. Click the Send button.
- Expected Result: The media file should be sent successfully.
- Actual Result: The app crashes.

#### Defect 3: Incorrect Error Message for Invalid Phone Number

- ID: D003
- Title: Incorrect error message displayed for invalid phone number
- Description: When an invalid phone number is entered during registration, the error message displayed is not user-friendly.
- Severity: Minor
- Priority: Low
- Steps to Reproduce:
  1. Open the Telegram app.
  2. Tap on "Start Messaging."
  3. Enter an invalid phone number.
- Expected Result: Display a user-friendly error message.
- Actual Result: Error message "Invalid input" is displayed.

#### Defect 4: Missing "Forgot Password" Option

- ID: D004

- Title: Missing "Forgot Password" link on login screen
- Description: The login screen does not have a "Forgot Password" link, preventing users from recovering their passwords.
- Severity: Major
- Priority: Medium
- Steps to Reproduce:
  1. Open the Telegram app.
  2. Navigate to the login screen.
- Expected Result: A "Forgot Password" link should be present.
- Actual Result: No such link is available.

#### Defect 5: Slow Load Time for Chat History

- ID: D005
- Title: Slow load time for chat history
- Description: Chat history takes an excessively long time to load, especially for large conversations.
- Severity: Moderate
- Priority: Medium
- Steps to Reproduce:
  1. Open a chat with extensive history.
- Expected Result: Chat history should load within a few seconds.
- Actual Result: Chat history takes over 10 seconds to load.

#### Defect 6: Notification Settings Not Saving

- ID: D006
- Title: Notification settings do not save
- Description: Changes made to notification settings are not saved and revert to default settings.
- Severity: Major
- Priority: High
- Steps to Reproduce:
  1. Go to "Settings."
  2. Modify any notification setting.
  3. Save changes.
  4. Reopen notification settings.
- Expected Result: Changes should be saved.
- Actual Result: Settings revert to default.

#### Defect 7: Inconsistent Emoji Display

- ID: D007
- Title: Emojis display inconsistently across devices
- Description: Emojis appear differently on different devices, leading to miscommunication.

- Severity: Minor
- Priority: low
- Steps to Reproduce:
  1. Send an emoji from Device A.
  2. View the emoji on Device B.
- Expected Result: Emojis should appear consistently.
- Actual Result: Emojis display differently.

#### Defect 8: Profile Picture Upload Fails

- ID: D008
- Title: Profile picture upload fails
- Description: Users are unable to upload profile pictures; the upload process fails with an error.
- Severity: Major
- Priority: High
- Steps to Reproduce:
  1. Go to "Settings."
  2. Tap on "Profile Picture."
  3. Select an image to upload.
- Expected Result: Profile picture should be uploaded successfully.
- Actual Result: Upload fails with an error.

#### Bug Life Cycle

The Bug Life Cycle is the process that a software defect undergoes from its identification to its resolution and closure. It provides a structured approach to tracking and managing bugs to ensure they are properly addressed.

