“Anotepad.com” Functionality Test Plan

General Information

This document describes methodologies and approaches that will be applied to the testing of the functionality of http://aNotepad.com. It contains objectives, test responsibilities, scope, entry and exit criteria and major milestones. Test Plan helps to determine the time and effort required to test the product. It also specifies the resource requirements and environmental needs to manage the testing process successfully.

Scope

The following table defines the scope of the http://aNotepad.com functional test effort:

Functions to be tested	Functions not to be tested
Note operations: Create Update, Delete. Move	Upgrading to Premium Account
Saving notes without creating an account	Send Note by Email (Available in Premium Plan only)
Signing up with a free account	
Sharing notes via Twitter, Facebook etc.	
Downloading notes in PDF, MS Word, ODT, and Text format	
Browsing through notes with the note preview button	
Making the notes protected with the password	
Creating multiple folders and sort notes by date or title	
Autosave feature for the notes	
“Note Title” field containing the symbols quantity in range between 3 and 100	
Folder name field containing the symbols quantity in range between 3 and 100	
Logging with correct Email and Password and password remembering feature	
Sending email for a newly registered user	
Allowing changing password (at least six characters)	
Changing time zone	
Changing color theme	
Downloading all notes (for registered users)	

Glossary

The table below clarifies the test terms and other terms used in this document:

Term	Definition
Entry Criteria	The conditions that must be met before the test starts.
Exit Criteria	The conditions that must be completed before testing is concluded.
Scope	The areas of a product that are supposed to get tested, what

	functionalities to focus on and what areas or features should not be covered by a test process.
Test environment	A setup of software and hardware to execute the test. It may contain operating system, applications, database server, browser, network configuration.
Test case	A set of actions executed to verify a specific functionality or feature of the software application.
Milestones	Small project targets that allow tracking the progress and timeline.
Bug	A defect, which means that software or application is not working as per the requirement

Schedule of Milestones

The following table shows the scheduled events that affect this test effort:

Milestone/Effort	Start	End
Planning	08.12.2021	10.12.2021
Plan agreement by project management	10.12.2021	11.12.2021
Areas to be tested are identified	11.12.2021	12.12.2021
Required staff assignment	12.12.2021	13.12.2021
Entry criteria are met	13.12.2021	13.12.2021
Tests are identified	13.12.2021	13.12.2021
High priority tests details documented	14.12.2021	15.12.2021
Test environment ready	15.12.2021	16.12.2021
Functional tests run	17.12.2021	25.12.2021
Exit criteria are met	26.12.2021	26.12.2021
Test results and errors summary review	27.12.2021	28.12.2021

Entry criteria

The entry criteria are the conditions that must be met before test starts:

- Required test engineers are defined.
- The budget has been agreed and allocated.
- All hardware test platforms are successfully installed, configured and functioning properly.
- All necessary documentation, requirements, design and information is available.
- All standard software tools including the testing tools are successfully installed and working properly.
- Proper test data is available.
- Test environment is prepared and the http://aNotepad.com is loading properly in the test environment.
- Test engineers have completely understood the requirements.

Exit criteria

These are the conditions that must be met before the test is completed:

- All the previously planned tests are performed.
- A certain level of requirements coverage has been achieved.
- All the errors of high priority and severity are fixed.
- The budget has been spent within the planned limit.
- The test results are evaluated, discussed and approved.

Test environment and test configurations

The following environment will be used on server and client workstations:

Term	Definition
Hardware	PC and laptop minimum requirements: • CPU: Intel Core 3 (sixth generation or newer) or equivalent • HDD: 500 GB • RAM: 8 GB • 15" LCD monitor
Software	Operation System: • Windows 7 SP1 • Windows 10 • Windows NT • Mac OS X Software tools: • IE 11+, Chrome (latest), Firefox (latest), Safari (latest) • Microsoft Office 2010 or higher • TestLink 1.9.20
Network	100 Mbps or above

There are three servers running:

System	Name	IP Address	OS	Status
Srv1	Kyiv	192.168.2.8	WinNT	Available
Srv2	Lviv	192.168.2.9	WinNT	Available
Srv3	Dnipro	192.168.2.7	WinNT	Available

Responsibilities and Roles

The following table describes the human resources required to execute this plan:

Role	Responsibilities	Member
Test Manager	Planning, control and coordination of test activities. Defining the scope of testing concerning the context. Building up and leading a test team to the project	Bill Sharpe

	success.	
Project Manager	Maintaining the balance between workflow productivity, reliability of the product, cost of the work and development time.	Jill Saward
Manual Test Engineer	Analyzing the product requirements, writing and executing the test cases in order to detect, report and prevent product defects.	Roger Odell

Test development

The test team develops test cases according to a test development/execution schedule. This schedule is needed to allocate human resources and reflect development due dates. The test team also monitors development progress and produces progress status reports.

Taking into account the priority domains for the end-users as well as the domains with possible crucial technical issues, the areas of testing are identified and prioritized. Some of these areas will need complementary test cases created.

Test execution

The main purpose of the Functional Test phase is to make sure that the whole product works according to the requirements and no significant errors appear. This is implemented by means of running a set of manual test cases against the web site. It involves checking of different aspects of the system, such as executing each use case or function using valid and invalid data to verify that the expected results occur when valid data is used. On the contrary, the appropriate error or warning messages are displayed when invalid data is used.

Based on designed scenarios and test data, test cases will be executed by respective test engineers. The actual result will be updated in Defect Logging and Reporting document.

Bug Reports

Bug reports are created in order to provide the development team and the Project Manager with complete information about the discovered defects. This must be helpful in determining root causes of the errors and correcting them.

The severity of the defect can be classified into the following categories:

- **Critical (blocker) defects** are the issues that require immediate removal, affect core functionality of the product and can cause system crashes or block the operability. It also applies to data loss failures and to the processes that leave inconsistent data stored.
- **Major defects** are those which affect main system functions but do not cause crash-down of the whole system or when a crucial module in the web site is malfunctioning, but the rest of the system works fine.
- **Minor defects** do not result in system failure but causes the system to show incorrect, incomplete or inconsistent results. These defects do not have much impact on system

functionality and do not affect severely on the workflow but can cause a negative impact on user experience.
- **Trivial defects** are not related to system functionality and by that can be skipped: grammar mistakes, typos, incorrect terminology etc.

Bug report provides the following information concerning the defect:

- Short description of the problem location of the defect in the product.
- Frequency of the defect occurrence.
- Severity of the defect.
- Steps to reproduce the error.
- Additional information about the defect (attached screenshots).

Test case tracking

Test cases are to be tracked in order to provide both detail level and summary level information on test cases for each test pass. The test case may have the following states:

- The test case is completed successfully (Passed)
- The test case completed with an error or being deferred by the project team (Warn)
- A defect occurred as a result of running a test case (Fail)
- The test case with a failure that was resolved later (Closed)
- The test case is under execution (In Queue)
- The test is skipped. A mandatory explanation in “Comments” section must be provided (Skipped)
- The test cannot be executed. A mandatory explanation in “Comments” section must be provided (Blocked)

The test case also contains the information about the name of a tester, who runs a test; a Bug ID that is assigned to the bug found in case the test failed; time needed to run the test; the planned and actual effort (in person-hours), the date when the test was run for the first time.

Bug tracking

Bug tracking is the process of logging and monitoring bugs or defects. Each defect needs to be evaluated, monitored and prioritized for debugging. Defect tracking helps to ensure that bugs found in the system actually get fixed. The severity of the effects of each potential failure classifies from the most damaging to the least damaging:

- Loss of data;
- Loss of functionality;
- Loss of functionality with a workaround;
- Partial loss of functionality;
- Cosmetic error.

The bugs are prioritized upon the importance of fixing a problem (from the most important to fix until the least important):

- Urgent
- Essential
- Valuable
- Desirable
- Discretionary

The bug report may have the following states:

- **Review** (another tester reviews current problem)
- **Rejected** (the problem is not confirmed and is not recognized as a bug)
- **Reported** (test engineer considers the problem to be explicit)
- **Assigned** (development accepts the problem as being explicit and clearly described. The owner responsible for the problem fix is assigned)
- **Build** (the problem is regarding to be fixed and the corresponding changes were made into the code in order to be tested in the future test release)
- **Test** (the fix is under the evaluation)
- **Reopened** (the fix has failed the retest)
- **Defer** (the fix is deferred for future release)
- **Closed** (the retest confirmed the problem to be fixed)

Deliverables

Test deliverables are the artifacts that are produced by staff involved in the test process. They are:

- Test Plan
- Test cases and test data
- Bug reports
- Test summary reports
- Test status reports
- Test closure report

Revision history

The following table depicts the revision history for the test plan.

Version	Date of Release	Description	Author
0.1	01.12.2021	First draft presented to the team	Bill Sharpe
1.0	04.12.2021	Reviewed after minor changes	Bill Sharpe
1.2	06.12.2021	Final review	Bill Sharpe
1.3	08.12.2021	Agreed by the stakeholders	Bill Sharpe

Approvals

No	Name	Signature	Date
1	Brian Culbertson		
2	Marcus Miller		
3	Keiko Matsui		