Test Plan for

Project “NotePad”

General information	
Customer	
Created by	Seleznev Aleksandr
Preparation Date	07.03.2023
Version	
Status	

Revision History					
Version	Description	Author	Date	Approved by	
				Author	Date

Summary

1. INTRODUCTION

1.1. GENERAL INFORMATION

This document describes the methods and procedures that will be used by **its partner** team in the functional testing process mobile application.

It is meant to be used as a manual during testing works. It describes the procedure of the testing process. The test plan is intended for project managers, product developers and QA engineers.

The objective of the testing activities is to check functions and features of a software product developed for Android 4+ devices.

1.2. PURPOSE

This Test Plan document for the Notepad project supports the following objectives:

- Identify existing project information and software components to be tested.
- Recommendation and description of the testing strategies to be employed.
- Identify required resources and provide a test effort estimate.
- List the test project deliverable elements.

The results of test execution will be sent to the customer as reports. All found bugs will be tracked using the Jira bug tracker.

2. SCOPE OF PROJECT

Testing of mobile applications is in the scope of this test plan. The following components and functions would be tested:

1. Installation, uninstallation, updating.
2. Launch of application.
3. Creation of a new note.
4. Editing, coping and removal of notes.
5. Editing of note’s titles.
6. Minimize and expand of application.

7. Screen orientation (landscape/portrait).
8. Work off application in the background.

3. WORK PLAN

The parties agreed to follow the next work plan:

1. Test plan preparation.
2. Test plan approval.
3. Functional testing and bugs reporting.
4. Daily reports preparation.
5. Final report preparation.

4. TEST PLAN SRATEGY

4.1. FUNCTIONAL TESTING

The objective of functional testing is to make sure that the whole software product works according to the requirements, and no significant errors appear in the application.

Functional testing is the most substantial part of software testing. It involves checking different aspects of the system. A software product must pass all the planned tests. Only in this case its quality can be assured.

Test Objective:	Ensure proper target-of test functionality
Technique:	Execute each use case, use-case flow, or function, using valid and invalid data, to verify the following: ● The expected results occur when valid data is used. ● The appropriate error or warning messages are displayed when invalid data is used. ● Each rule is properly applied.
Entry Criteria:	● The application construction is completed. ● The test engineers are dedicated. ● Necessary devices, instruments, and other equipment are acquired. ● Test environment is prepared, and the application is released to the test environment.
Completion Criteria:	● All the planned tests are performed. ● There are no show-stopping errors. ● All the errors of high priority and severity are fixed. ● The test results are evaluated, discussed and approved.
Special Considerations:	None.

4.2. TEST PROCEDURE

Test procedure assumes the next points:

- Reporting of found software bugs.

Various aspects of the tested software should be checked; this requires executing of different testing types.

The main testing types that would be executed:

- Functional Testing.
- UI Testing.
- Usability Testing.
- Compatibility Testing (4 old fashion and modern devices).
- Regression testing.
- Retesting (during the second round if needed).

It also will be checked how the software product is run on devices that are supposed to support it, how it starts and stops, and how much time it needs to launch.

During this test round the next types of testing will NOT be applied:

- Security testing.

4.3. BUG REPORTS

Bug reports are created in order to provide the development team and the project managers with exhaustive information about the discovered defects. They must be helpful in determining causes of the errors and correcting them.

Defect Severity can be classified into four categories:

- Critical (blocker) defects are the failure of the complete software system or of a critical subsystem, and no work or testing can be carried out after the occurrence of the defect. It also applies to data loss failures and with processes that leave inconsistent data stored in the database.
- Major defects (and crashes) are those which also cause failure of the entire or part of the system, but there are some processing alternatives which allow further operation of the system. It also applies to the system crashing, or aborting, during normal operation of a non-critical flow.
- Minor defects do not result in failure but cause the system to show incorrect, incomplete, or inconsistent results.
- Trivial defects are small errors that do not affect the functionality: typos, grammar mistakes, wrong terminology, etc.

The information that is indicated in each bug report:

- the software product name;
- version number of the software product (if tested on mobile);
- the browser on which the tests were performed.

Each report provides the next information about the defect:

- summary, which is short description of the problem;
- location of the defect in the software product;
- steps to reproduce the error;
- frequency of the defect occurrence;
- severity of the defect;
- additional information about the defect in the form of attached screenshots or video records.

Third party software will be used for reporting and maintaining discovered errors. The test team will log in that software all the defects that will be found during the testing process.

5. RESOURCES

5.1. TOOLS

The following tools will be used for this project:

Name of process	Tool
Defect Tracking	Jira
Test Cases	Testrail
Screenshots / Video capture	Snagit
Emulation of devices	Android studio

5.2. THE LIST OF DEVICES

Name of devices	OS
Asus_Z00ED	Android 6.0.1
ZTE Blade A3 2019RU	Android 9
Samsung Galaxy A13	Android 13

6. THE CRITERIA OF QUALITY

The product should operate in accordance with the requirements and the functional specification (if present).

The product should not contain critical and blocking defects in the final version of the project.

7. TESTING PROCESS RISKS

The next issues may influence testing works:

- changes and modifications of the software product that were not planned and discussed with the test team beforehand;
- changes in the software requirements that were not discussed with the test team beforehand;
- delays in correcting/fixing errors;
- delays in delivering new builds to the test team.

8. TEXT TEAM EXPEXTATIONS

The test team must be provided with valid, updated documents during the whole testing process.

All the required equipment, instruments, devices and software must be acquired and prepared before the beginning of the testing process.

All show-stopping errors must be corrected as soon as possible.

Release notes should be added to each software release to the test team. The note must explain which elements, functions and features were added to the program and how these additions affect the software.

The developers should correct all the errors in the software modules before releasing a new version.

9. RESPONSIBILITIES OF TEST TEAM MEMBERS

Project Manager

- Managing the whole testing process.
- Providing all the needed resources for the testing activities.

QA Tech Lead

- Managing the QA team from a technical perspective.
- Analyzing the tasks and distributing them between team members.
- Communicating with the client team and discussing all issues, providing recommendations before an update or release.
- Experience in participation of different SDLC models like Agile, Scrum, Kanban, Sequential, Iterative and Incremental.
- Creating test documentation, including test cases, test plans, etc.
- Proposing best practices and tools for a project.

QA Engineer

- QA process / logging found errors into the approved bug tracking system.

10. DELIVERABLES

- Test Plan.
- Bug reports and reports regarding the testing progress.