

Testing report for the "Shopping-List" mobile application

Report prepared by:

Uliana Briakina

Application Overview	3
Testing Scope	3
In Scope	3
Out of scope	4
Items not tested	4
Metrics	5
The ratio of executed test cases to unexecuted ones	5
The ratio of passed test cases to failed ones	5
Number of defects identified and their priority	6
Defects distribution – module wise	6
Types of testing performed	7
Test Environment and Tools	7
Lessons Learnt	8
Recommendations	8
Best Practices	8
Exit Criteria	8
Conclusion/Sign Off	8

Application Overview

The "shopping-list" mobile application is a simple task management app. Its functionality includes the ability to create and store up to 10 short tasks simultaneously. The user can edit tasks, mark them as completed, and delete tasks from the list.

The application is localized only in Russian, and the validation parameters also include the requirement to save tasks using only Cyrillic characters.

Tasks are saved in the device's local storage. Additional permissions and network access are not required.

Testing Scope

In Scope

- Functional testing of the Task management module, including submodules:
 - Adding Tasks
 - Task Validation
 - Editing Tasks
 - Saving Tasks
 - Marking tasks as completed
 - Deleting Tasks
- User Interface testing, including:
 - Usability
 - Localization
- Compatibility with the device hardware and software, including:
 - Installability
 - Interruption
 - Changes in system state
 - Interaction with device buttons
- Performance testing for the main functionality of the application

Out of scope

- Testing for access permissions to device features.
- Testing for quick-access Links
- Testing for interoperability and co-existence with other apps on the device.
- Testing for various connectivity methods.

Items not tested

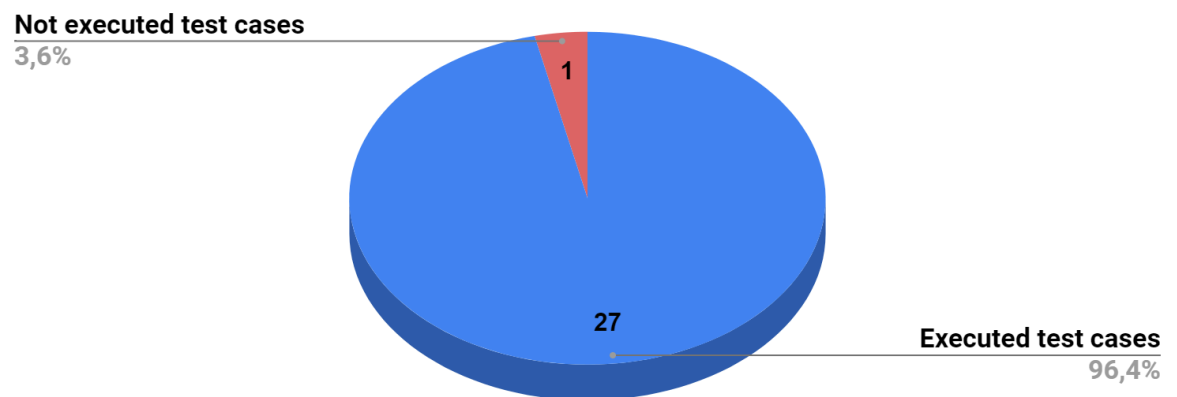
- Testing for power consumption and state.
- Stress Testing.
- Testing for interoperability with multiple platforms and operating system versions - the application has been tested on only one platform at this stage.
- Security Testing.
- Accessibility Testing.

Metrics

The ratio of executed test cases to unexecuted ones

Planned test cases	Executed test cases	Not executed test cases
28	27	1

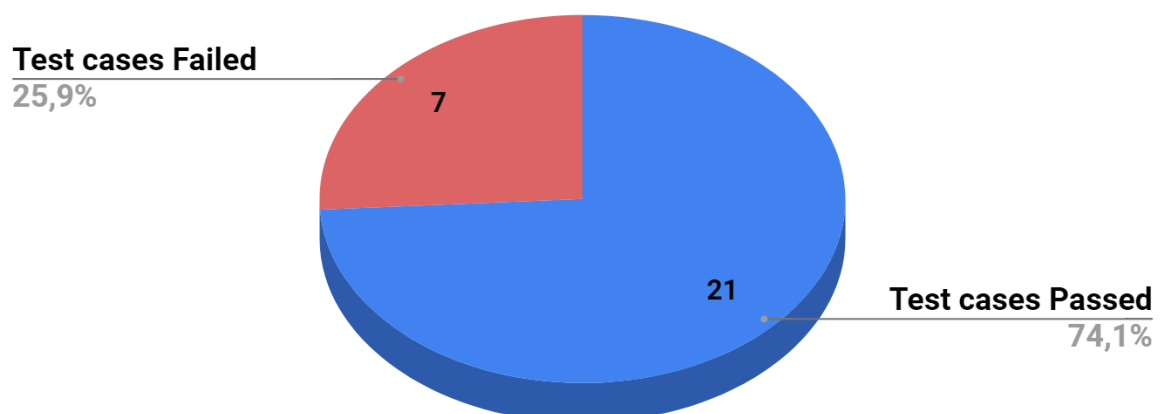
Test Cases Execution



The ratio of passed test cases to failed ones

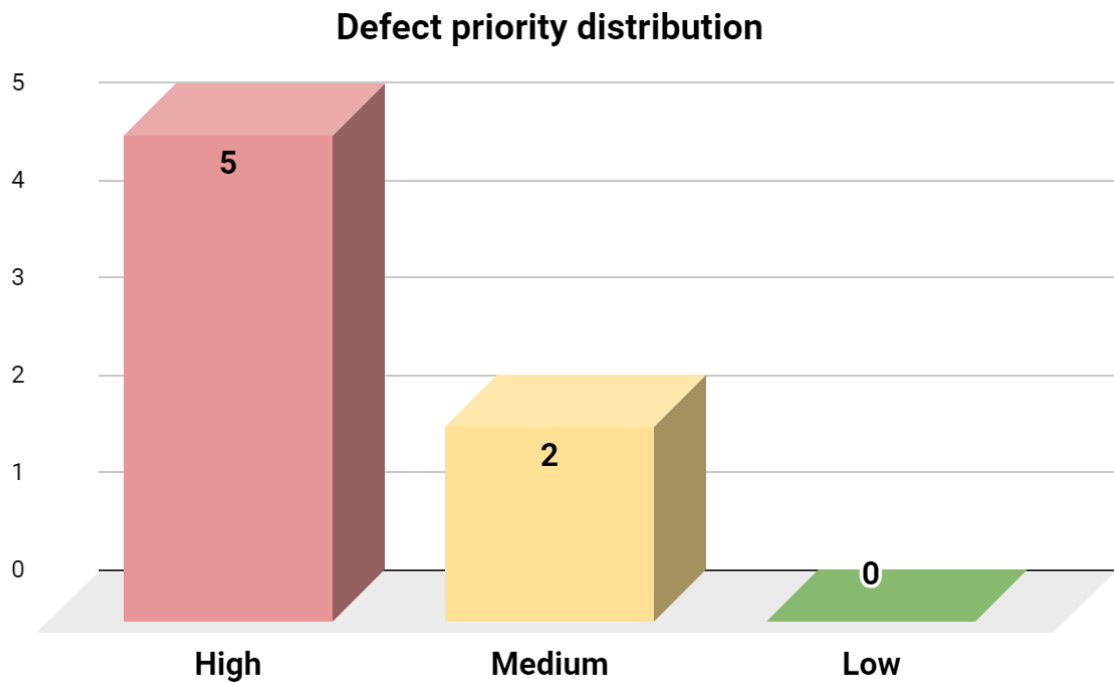
Test cases Passed	Test cases Failed
20	7

Test Cases Failed VS Passed



Number of defects identified and their priority

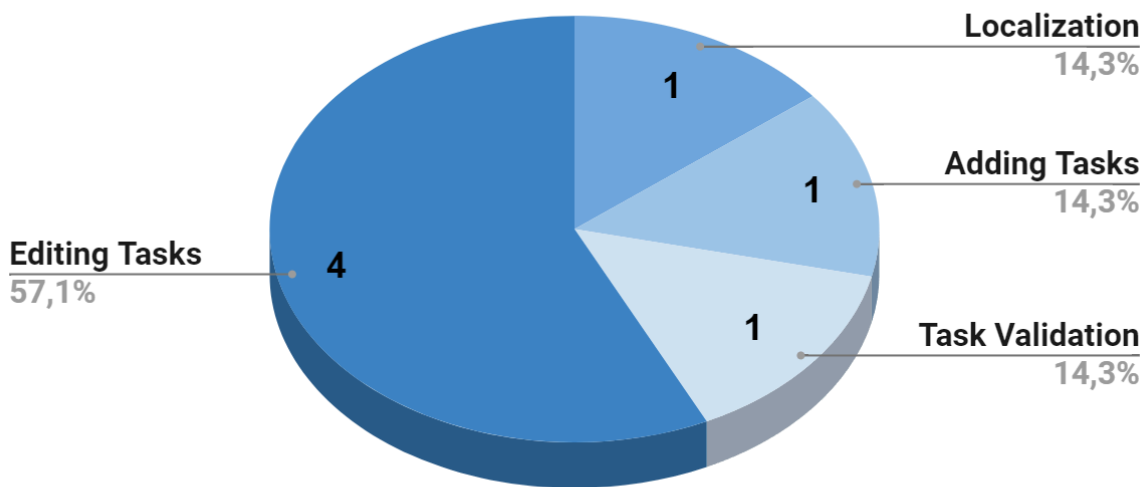
Defect priority distribution			
High	Medium	Low	Total
5	2	0	7



Defects distribution - module wise

	High	Medium	Low	Total
Localization		1		1
Adding Tasks		1		1
Task Validation	1			1
Editing Tasks	4			4
Total	5	2	0	7

Defects distribution – module wise



Types of testing performed

It is important to note that all the types of testing listed below were conducted as part of **manual testing, exploratory** by nature of its formalization. This means that the previously prepared testing scenario was refined during the testing process to reflect the specificities of the interface, as they were not agreed upon in the form of a mock-up during the requirements review.

Dynamic testing

Dynamic testing was used to check the actual behavior of the application based on code execution.

Smoke Testing

Smoke testing was conducted to verify the core functionality of the application was working properly, before the build could be accepted and testing could start. It was necessary to ensure that the user can create a task or a list of tasks, edit tasks, delete them, and mark them as completed.

Installation testing

The installation test was conducted to verify the smooth installation of the application, ensuring no crashes or data loss during the installation process. The following scenarios were tested:

- Installation using the downloaded APK file;
- Appearance of the application icon in the device menu for quick access;
- Uninstallation of the application;
- Reinstallation of the application;

GUI testing

GUI testing was conducted to validate the proper functioning of the application's interface. It was verified that all buttons and fields required for the basic functions of the application are displayed correctly. All functional elements work without errors and behave as expected when interacted with.

Usability testing

Usability testing in this check only included verifying the visibility and accessibility of buttons and input fields.

Localization testing

Localization testing was conducted to verify the correct localization of the application into Russian (including all graphical user interface elements).

Compatibility testing

The application's compatibility with a specific environment was verified. The following aspects were checked:

- Application behavior when interrupted by system notifications, pop-up alerts, quick access panel activation, and a charging cable connection.
- Application behavior when interacting with the device's physical and touch-sensitive buttons.
- Application response to signals from the device's sensors.
- Application response to changes in global settings, such as sound and network access.

Test Environment and Tools

Device: Samsung Galaxy A52s 5G

OS: Android 13, One UI Version: 5.1

Android Studio version: Koala Feature Drop I 2024.1.2 Patch 1

Lessons Learnt

Issue faced: The requirements analysis stage was significantly shortened, and the requirements didn't include an application mockup.

Solution: The graphical interface of the provided build, along with common sense, was used to fill in the missing requirements.

Recommendations

A mockup of the application, provided during the requirements analysis and checklist development, would have significantly reduced testing time, as the requirements for the application would have been clearer.

Exit Criteria

- All test cases should be executed - Yes
- All defects with high and medium priority should be verified and closed – Yes

Conclusion

Even after meeting the Exit Criteria from the previous section, the application **is not ready for release**. More comprehensive testing is required on other devices and operating systems. The testing scenario should also be supplemented with items from the "[Items not tested](#)" section.