Regression Test Report

After the bugfix, the QA team conducted regression testing. It used test cases that were used during beta testing. Also, this list was expanded with checks of those functionalities that were affected as a result of the bugfix.

While using the old test cases, it was detected experimentally that the changes made in the program code during the bugfix had a big impact on the previously correctly working functions. In this regard, the team of testers paid extra attention to the previously correctly working functions.

Also during regression testing, new bugs (which had not been previously detected by the testing team and the focus group) were identified. Nevertheless, they have been fixed. The checks used are introduced in the table below:

id	Summary	Priority	Module	Steps	Expeted result	Test result
1	Installing of app	High	Installing	1. Run the file "Calaculator.apk" 2. Confirm the intention of installation 3. Wait until the installation is completed	1. The application installer has started 2. The installation process has started 3. Installation completed successfully	Passed
2	Running the application	High	Launching	1. Start the application 2. Wait for the loading screen to appear	1. Application has started 2. Loading screen has appeared	Passed
3	Using the number buttons	High	Calculating	1. Start the application 2. Click button with number	1. Application has started 2. Right number appeared in working field	Passed
4	Using the "addition" function	High	Calculating	1. Start the application 2. Type expression "242+20" 3. Click "equal"	1. Application has started 2. Expression appeared at working field 3. In result field appeared "262.0"	Passed
5	Using the "subtract" function	High	Calculating	1. Start the application 2. Type expression "2-37" 3. Click "equal"	1. Application has started 2. Expression appeared in working field 3. In result field appeared "-35.0"	Passed
6	Using the "multiply" function	High	Calculating	1. Start the application 2. Type expression "54*3" 3. Click "equal"	1. Application has started 2. Expression appeared in working field 3. In result field appeared "162.0"	Passed

Table 1 – Test cases for regression testing

Continued Table 1– Test cases for regression testing

id	Summary	Priority	Module	Steps	Expeted result	Test result
7	Using the "divide" function	High	Calculating	1. Start the application 2. Type expression "980/25" 3. Click "equal"	1. Application has started 2. Expression appeared in working field 3. In result field appeared "39.2"	Passed
8	Using brakets function	High	Calculating	1. Start the application 2. Type expression "34*(98-32)" 3. Click "equal"	1. Application has started 2. Expression appeared in working field 3. In result field appeared "2244.0"	Passed
9	Using dot	High	Calculating	1. Start the application 2. Type expression "12.5/5" 3. Click "equal"	1. Application has started 2. Expression appeared in working field 3. In result field appeared "2.5"	Passed
10	Using "percent" function	High	Calculating	1. Start the application 2. Type expression "123" 3. Click "%"	1. Application has started 2. Expression appeared in working field 3. In result field appeared "1.23"	Passed
11	Using "All clear" function	High	Calculating	1. Start the application 2. Type expression "482+3468" 3. Click "AC"	1. Application has started 2. Expression appeared in working field 3. Working field became empty	Passed
12	Using "Backspace" function	High	Calculating	1. Start the application 2. Type any number buttons 3. Cick "backspace" button	1. Application has started 2. Numbers appeared in working field 3. Last number deleted in working field	Passed
13	Calculating a complex expression	High	Calculating	1. Start the application 2. Type expression "4.5*6.3+(54-17.4)*0.35" 3. Click "equal"	1. Application has started 2. Expression appeared in working field 3. In working field appeared "41.16"	Passed
14	Calculating a expression with mistake	High	Error processing	1. Start the application 2. Type expression "7/(8-1-7)" 3. Click "equal"	1. Application has started 2. Expression appeared in working field 3. In working field appeared "Error in expression"	Passed
15	Using result of previous calculation	High	Calculating	1. Start the application 2. Type expression "66+32" 3. Click "equal" 4. Type "/12" 5. Click "equal"	1. Application has started 2. Expression appeared in working field 3. In working field appeared "98.0" 4. In working field appeared "98.0*12" 5. In workin field appeared "1176.0"	Passed

Continued Table 1– Test cases for regression testing

id	Summary	Priority	Module	Steps	Expeted result	Test result
16	The inability to use two characters in a row	High	Program logic	1. Start the application 2. Type "911" 3. Type "-" twice 4. Click "AC" button 5. Repet points 2-4 with "*", "/", "+", "."	1. Application has started 2. Expression appeared in working field 3. In working field added only one "-" 4. Working field became empty 5. Only one relevant symbol appears **each time**	Passed
17	Character interchangeability	High	Program logic	1. Start the application 2. Type "333" 3. Type "-" 4. Type "+" 5. Click backspace 6. Type "+" 7. Type "/" 8. Click backspace 9. Type "*" 10. Type "/" 11. Click backspace 12. Type "/" 13. Type "*" 14. Click backspace 15. Type "/" 16. Type "-" 17. Click backspace twice 18. Click "*" 19. Click "-" 20. Click "+"	1. Application has started 2. Expression appeared in working field 3. To expression added "-" 4. "-" exchanged to "+" 5. Last symbol deleted from expression 6. To expression added "+" 7. "+" exchanged to "/" 8. Last symbol deleted from expression 9. To expression added "*" 10. "*" exchanged to "/" 11. Last symbol deleted from expression 12. To expression added "/" 13. Last symbol exchanged to "*" 14. Last symbol deleted from expression 15. To expression added "/" 16. To expression added "/" 17. Expression equal "333" 18. To expression added "*" 19. Expression added "-" 20. Two last symbols exchanged to "+" (Expression became "333+")	Passed
18	Automatic closing of brackets	High	Program logic	1. Start the application 2. Type "(9-5" 3. Click "equal" 4. Click "AC" 5. Type "(9-5" 6. Click "percent"	1. Application has started 2. Expression appeared in working field 3. In working field appeared "4" 4. Working field became empty 5. Expression appeared in working field 6. In working appeared "0.04"	Passed

Continued Table 1– Test cases for regression testing

id	Summary	Priority	Module	Steps	Expeted result	Test result
19	The inability to use function characters and dot in start of expression (without minus)	High	Program logic	1. Start the application 2. Type "*" 3. Type "." 4. Type "+" 5. Type "/"	1. Application has started 2. Working field is empty 3. Working field is empty 4. Working field is empty 5. Working field is empty	Passed
20	The inability to use 2 dots in one number	High	Program logic	1. Start the application 2. Type "123" 3. Type "dot" 4. Type "65453" 5. Type "dot"	1. Application has started 2. Working field is "123" 3. Dot added to expression 4. Numbers added to expression 5. Dot didn't add	Passed
21	Correct working with empty working field	High	Program logic	1. Start the application 2. Type "=" more than 5 time 3. Type "%" more than 5 time	1. Application has started 2. Nothing happened 3. "Nothing appeared" after first time	Passed
22	UI check	High	UI	Start the application on different devices: 1. Google pixel 2 2. Google pixel 4 3. Google pixel 5	1. UI works correctly 2. UI works correctly 3. UI works correctly	Passed

The results show that the application is working correctly. This means that the development team can start creating the next version of the program.

QA Engineer: Anatoli Zabauski