Bug Report Team U

1. Method Used in Testing

We ran our test suite on the code, and analysed each failing test case, trying to determine if it was failing due to implementation differences or a bug. We decided to use our tests because we had a good suite of functional tests which covered all scenarios and it would be an ideal way to find functional bugs. To avoid finding the same bugs in different scenarios without recognizing them. We started by applying all the unit tests first and only then progressed to integration.

Where it was hard to determine what was going on, or when it was not clear what would be seen by the user, this was complemented by manual testing: we typed commands directly into the application, and observed the external behavior. Throughout most of the process we used a black-box approach, and only looked into and debugged the code in specific issues where it was unclear whether something was a bug.

In certain cases where our test suite made us aware of the problem but did not include a test case designed specifically for it we adapted test cases to make it clearer to diagnose.

We also explored by then looking at the code and understanding if there are any evident bugs (like null checks, planted bugs that weren't fixed) for any structural bugs and any false developer assumptions made for methods, objects etc. One great advantage we got from AutomatedTestingSuite was that running our automated generated tests helped us find 2-3 bugs. But we had to make sure they weren't implementation specific and failing hence it had to be properly examined. Using IntelliJ interpreter, FindBugs tools helped in finding structural possible bugs and allowed us to explore if they can be triggered through user commands. We also ensured that if multiple tests failed due to a single fault, we only reported them once such that resolving each bug resolves the main fault. We provided multiple tests to ensure that other teams can test if all tests pass once the main fault is fixed.

Finally, we put some effort into System Testing by running the Shell and trying different commands in different formats with different functionalities (output, no output, error etc.). This allowed us for testing many real time applications of the System and ensuring they work fine and do manual testing.

To run the failing tests:

Place the `dummyTestFolder` in the same directory level as `src.` The file `TeamUBugs.java` contains all test cases which reveal the bugs. The tests can be run forked by method for best understanding of actual bugs. The JavaDocs for each method state the description of bug and the number in `@DisplayName` can be used to point it to more details in the BugReports.pdf file.

2. Bug Report

#	Description	TestCase	Comments
1	rm with -r flag will not remove empty / non-empty directories. Prints “This is a directory”.	testRmNonEmptyFolderRecursive() / testRmEmptyFolderRecursive()	
2	‘rm -rd .’ removes all files in the current directory, which should not be possible.	testRmFailsDirEndInDot()	
3	ls with Globbing	ls *.txt should output all txt extension files, however it shows: ls: cannot access 'a.txt': No such file or directory testLsWithFileSpecified()	
4	CommandSubs within Double quotes adds extra spaces	echo "This is random:`echo "random"`." => This is random:random. Should not print an extra space before the period. testDoubleQuotesWithBackQuote() testInterpretationOfSpace()	
5	cp copies a file to the same directory it is in, overwriting itself unnecessarily / Also copies file	testCpFailsSingleFileToSameDir() / testCpFailsSingleFileToItself()	

	into itself with similar behavior		

6	Mv: Throws error message if not able to write to output stream when error occurs in not being able to replace a NonEmptyDirectory	testCantMoveToNonEmptyDir()
		
		abc/
		    inside/
		        a.txt
		inside/
		    b.txt
		
		mv inside abc => Exception wrongly caught and different error message given	

7	cp with null output stream displays Null Pointer Exception without specifying the cause	testCpFailsNullOutputStream()	
8	cp directory into itself creates infinite recursive nesting	testCpFolderToItselfWithAnotherValidFile()	
9	paste with null input/output stream displays Null Pointer Exception without specifying the cause	testPasteNullInputStream() / testPasteNullOutputStream()	
10	Incorrect error message when calling paste with a directory as argument. “No such file or directory” implies the user provided a nonexistent file.	testPasteDirOnlyFiles() throws IOException {	
11	‘paste - -’ causes input to stdin to be printed twice such that if the user inputs ‘a’ on a line and ‘b’ on the following line then terminates the command, the output generated is: a\ta b\tb Instead of a\tb	testPasteTwoStdinArgs()	
12	File not found when using ‘paste - file’ (pasting files with stdin) because filenames are not resolved against current directory	testPasteOneFileOneStdinArgs()	
13	NoSuchElementException	testOutputRedirFailsNoDest()	

	exposed to user when the user attempts to redirect output without a destination file (‘cmd args >’, e.g. ‘echo hello >’). Correct behavior would be to report a syntax error		
14	Cannot finish executing paste command with stdin input without terminating the shell	Manual testing: ‘paste -’	
15	Find with globs causes an unexpected error: ‘Paths must precede -name’. The problem should be that you can only search for a single name	testFindGlobInvalid()	
16	Find with quoted glob causes exception ‘Dangling meta character '*' near index 0’	testFindGlob()	
17	Find with multiple flags work, as long as the last flag is -name	testFindWithMultipleFlags	
18	Find throws wrong error when multiple files are provided and everything else is correct	testFindWithMultipleFileNames	
19	Diff command didn’t check for invalid flag	DiffApplicationTest.testFailsInvalidFlag	
20	Diff command didn’t follow output format specified by pdf	DiffApplicationTest.testDiffFilesWithDifferentContentUsingFlagQ	
21	Diff command doesn’t output anything for different directories with files of same content	DiffApplicationTest.testDiffDirContainFilesWithSameContent	
22	Diff command doesn’t read stdin as ‘-’ when displaying in the output	DiffApplicationTest.testDiffFileAndStdinWithSameContentUsingFlagS	
23	Cut command doesn’t detect out of range	CutApplicationTest.testOutOfRange	
24	Cut command doesn’t detect invalid range	CutApplicationTest.testInvalidRange	
25	Cut command doesn’t detect	CutApplicationTest.testIllegalFla	

	illegal flag	g	
26	Extra / when running wc command then find command	WcIntegrationCommandTest.testWcThenFind	
27	Extra unexpected output when running wc command then cut command	WcIntegrationCommandTest.testWcThenCut	
28	Unexpected output when running sort command and then piping output to cut command	SortIntegrationCommandTest.testSortThenCut	
29	Unexpected flags printed in the error message	SortIntegrationCommandTest.testSortThenEchoNegative	
30	Throws error that file doesn't exist when it should be creating the directory needed for files position	echo yo > dir1/dir2/out.txt Create a file in non existent directory 2 level down testCreateDirectoryThroughOutputstreamForFile()	
31	Ls With -R in multiple nested directories gives results in the wrong format. Missing line breaks and empty lines	Given dir structure: dir/ abc/ zz/ b.txt a.txt def/ f.txt new.txt When printing result no space between: dire/abc/zz: b.txt AND dire/def: f.txt Command: ls -R dire/ testLsWithRecursiveDirectory()	
32	Find command with wrong flag generates wrong command	Command: find A -i test	

		FindApplicationTest.testFindWithWrongFlag()	
33	Sed with no file provided and no stdin provided	After some period of time it should timeout and give an error testSedWhenNoFileIsProvided()	
34	Grep with no file provided and no stdin provided	After some period of time it should timeout and give an error testGrepWhenNoFileIsProvided()	
35	Cd to a directory with no read permission	It should throw an exception that directory is not readable testChangeToDirectoryWithNoReadPermission()	
36	Find command with no specified folder	It should throw an exception that the folder is not specified testFindWithNoFolderSpecified()	
37	Find command with a relative folder path specified	The message displayed is wrong. testFindWithRelativePathSpecified()	