Milestone 2:

Details about TDD Process:

- Testing
  - For all features & applications that were not implemented, we used the test cases given by other teams, and applied TDD to ensure test cases passed.
  - On features that were already implemented and tested by us, we also run the test cases given by other teams to check if we missed out edge cases in our original testing.
  - Before all testing began, we decided to re-test all features that we thought were correct from Milestone1 with the given TDD test cases.
  - Fixing features that were supposed to be “error-free” was our top priority, and we moved on to do TDD on features that were not implemented yet once the given TDD test cases passed.
  - When implementing new features, we focused on clearing test cases given by other teams, and then we re-tested them again based on our original test cases.

- Problems Encountered
  - We did not expect some of our original test cases to fail.
  - There were features that we were unsure on how to do integration test effectively like “pipe operator”, “semicolon”, “io redirection” etc, as there were no new test cases given by other teams for us to do integration testing for these features.
  - Features like “pipe operator” had to make plenty blind integration test cases as all of us had to wait for one another to finish implementing missing features.
  - When integrating new features and updating previous features from Milestone1, some of our original test cases failed, and required modifications.
  - The ‘Exit’ test cases given in TDD could not be executed as it requires the project to be Java <17, but our team is running Java >17.

Planning and Execution of Integration Tests:

- We split our work for milestone2 based on our assigned features in milestone1, where we tested integration test cases given by other teams on our assigned features.
- We brainstormed some ideas on how to do additional integration test cases on features which did not have integration test cases by other teams and by getting inspiration from their test cases.
- For all assigned features, our priority was to effectively test our features with the given integration test cases, and fix all bugs.
- One experimental additional integration test suite that we did was to try pairwise integration testing of all applications with the pipe operator. We reduced test cases from “15^15 = 255” test cases to around 40 test cases using selective pairwise testing as there were some applications that did not use stdin at all, and some applications were not required to test based on careful consideration.

Snippet of Pairwise Integration Testing:
(Subset of the original table shown here based on tests done in PipeOperatorIT in TDD folder)

TC	Applications Under Test	Input
1	echo | wc	echo 'Hello world' | wc
2	echo | sort	echo 'Hello world' | sort
3	echo | cat	echo 'Hello world' | cat
4	echo | ls	echo 'Hello world' | ls
5	echo | paste	echo 'Hello world' | paste > pipeTest.txt
6	echo | uniq	echo 'Hello world' | uniq
7	echo | cut	echo 'baz' | cut -b 3
8	echo | tee	echo 'hehe' | tee teeFileTest.txt
9	echo | grep	echo "java is boring" | grep "java"
10	wc | sort	wc pom.xml | sort
11	wc | cat	wc pom.xml | cat
12	wc | ls	wc pom.xml | ls
13	wc | paste	wc pom.xml | paste > pipeTest.txt
14	wc | uniq	wc pom.xml | uniq
15	wc | cut	wc uniq.txt | cut -c 1-2
16	wc | tee	wc uniq.txt | tee teeFileTest.txt
17	wc | grep	wc uniq.txt | grep 'uniq.txt'

Additional Faults Found:

Feature / Application	Fault Observed
Cut	1. Change runtime exception to cut exceptions
Paste	1. Did not allow empty argument but with input stream 2. Did not consider serial for 1 file 3. Did not handle when file is empty
Cat	1. Directory error should throw “This is a directory” instead of “Is a directory” 2. Line format was incorrect, should be space rather than tab character after the line number
Tee	1. Did not add new line for some test cases in TeeApplicationPublicTest class 2. Did not insert fileA or fileB correctly into the arguments 3. Did not put the correct paths of the files into the arguments
Wc	1. Line format was incorrect, should be tab character in front followed by format specifier for the filename or stdin 2. Did not have a line specifying the total number of lines, words and bytes 3. Did not have checks for null filenames and inputstream
Mkdir	1. Did not have checks for null and empty folderName in createFolder()
Grep	1. Did not have filenames shown by default when multiple files are used
Mv	1. When flag is specified, no exception should be thrown when dest exists
Echo	1. Did not indicate that Echo will terminate with a new line
Exit	1. To run Exit test cases, project structure should be set to java 16 and below

Cd	1. Did not handle null check for stdin and stdout 2. Assumption where the null or blank args will NOT go to root folder, “Cd ..” change the current directory to the parent directory of the current directory
IORedirection	1. Outputstream did not receive anything 2. Output IORedirection did not create a new file if the file did not exists
PipeOperator	1. Pipe Operator did not handle outputstream correctly 2. Output of tee not written to stdout. However, cli and file observed output but stdout is empty
Globbing	1. Did not add directory to files found to pass the regex