Integration test

# Version history

|  |  |  |  |
| --- | --- | --- | --- |
| Editor | Date | Version | Description |
| N/A | 20 Feb 2021 | 1.0.0 | Initial release |
| N/A | 8 Mar 2021 | 2.0.0 | Add test cases for file input |

Table of Contents

[Version history 2](#_Toc66128792)

[Welcome page 6](#_Toc66128793)

[Case: Should be able to show welcome page when the application is launched 6](#_Toc66128794)

[Grid size 7](#_Toc66128795)

[Case: Should be able to allow user to input grid size, before handling file input 7](#_Toc66128796)

[Case: Should be able to set default grid size, if grid size input is empty 8](#_Toc66128797)

[Case: Should be able to throw error if width input is greater than the upper limit 9](#_Toc66128798)

[Case: Should be able to throw error if width input is less than the lower limit 10](#_Toc66128799)

[Case: Should be able to throw error if height input is greater than the upper limit 11](#_Toc66128800)

[Case: Should be able to throw error if height input is less than the lower limit 12](#_Toc66128801)

[Case: Should be able to throw error if the input is invalid 13](#_Toc66128802)

[File input 14](#_Toc66128803)

[Case: Should be able to process commands from file first, before waiting for user manual input on console, if file exists 14](#_Toc66128804)

[Case: Should be able to wait for user manual input on console directly, if file is not found in the designated location 16](#_Toc66128805)

[Case: Should be able to process the commands in file input correctly 17](#_Toc66128806)

[Case: Should be able to process the file even if there is empty line at file end 18](#_Toc66128807)

[Case: Should be able to throw error if any of the commands is invalid 19](#_Toc66128808)

[Case: Should be able to rename file after processing 20](#_Toc66128809)

[Manual console input 21](#_Toc66128810)

[Case: Should be able to throw error if user input is empty 21](#_Toc66128811)

[Case: Should be able to throw error if command is invalid 22](#_Toc66128812)

[Case: Should be able to throw error if the PLACE command has invalid format (More than 2 parts including the command) 23](#_Toc66128813)

[Case: Should be able to throw error if the PLACE command has invalid format (More than 3 elements in the part of the params) 24](#_Toc66128814)

[Case: Should be able to throw error if the MOVE command has more than 1 part (it should contain only the command) 25](#_Toc66128815)

[Case: Should be able to throw error if the LEFT command has more than 1 part (it should contain only the command) 26](#_Toc66128816)

[Case: Should be able to throw error if the RIGHT command has more than 1 part (it should contain only the command) 27](#_Toc66128817)

[Case: Should be able to throw error if the REPORT command has more than 1 part (it should contain only the command) 28](#_Toc66128818)

[PLACE command 29](#_Toc66128819)

[Case: Should be able to place the robot on the table 29](#_Toc66128820)

[Case: Should be able to throw error if the x-coordinate param is not an integer 30](#_Toc66128821)

[Case: Should be able to throw error if attempt to place the robot on the table with the x-coordinate < 0 31](#_Toc66128822)

[Case: Should be able to throw error if attempt to place the robot on the table with the x-coordinate > 5 32](#_Toc66128823)

[Case: Should be able to throw error if the y-coordinate param is not an integer 33](#_Toc66128824)

[Case: Should be able to throw error if attempt to place the robot on the table with the y-coordinate < 0 34](#_Toc66128825)

[Case: Should be able to throw error if attempt to place the robot on the table with the y-coordinate > 5 35](#_Toc66128826)

[Case: Should be able to throw error if the facing is invalid 36](#_Toc66128827)

[MOVE command 37](#_Toc66128828)

[Case: Should be able to move robot east 37](#_Toc66128829)

[Case: Should be able to move robot south 38](#_Toc66128830)

[Case: Should be able to move robot west 39](#_Toc66128831)

[Case: Should be able to move robot north 40](#_Toc66128832)

[Case: Should be able to throw error on MOVE command if robot does not have location 41](#_Toc66128833)

[Case: Should be able to throw error if robot will fall off table after moving east 42](#_Toc66128834)

[Case: Should be able to throw error if robot will fall off table after moving south 43](#_Toc66128835)

[Case: Should be able to throw error if robot will fall of table after moving west 44](#_Toc66128836)

[Case: Should be able to throw error if robot will fall of table after moving north 45](#_Toc66128837)

[LEFT command 46](#_Toc66128838)

[Case: Should be able to turn the robot to face north if it is facing east 46](#_Toc66128839)

[Case: Should be able to turn the robot to face west if it is facing north 47](#_Toc66128840)

[Case: Should be able to turn the robot to face south if it is facing west 48](#_Toc66128841)

[Case: Should be able to turn the robot to face east if it is facing south 49](#_Toc66128842)

[Case: Should be able to throw error on LEFT command if robot does not have location 50](#_Toc66128843)

[RIGHT command 51](#_Toc66128844)

[Case: Should be able to turn the robot to face south if it is facing east 51](#_Toc66128845)

[Case: Should be able to turn the robot to face west if it is facing south 52](#_Toc66128846)

[Case: Should be able to turn the robot to face north if it is facing west 53](#_Toc66128847)

[Case: Should be able to turn the robot to face east if it is facing north 54](#_Toc66128848)

[Case: Should be able to throw error on RIGHT command if robot does not have location 55](#_Toc66128849)

[REPORT command 56](#_Toc66128850)

[Case: Should be able to report the robot's location 56](#_Toc66128851)

[Case: Should be able to throw error on REPORT command if robot does not have location 57](#_Toc66128852)

[Logging 58](#_Toc66128853)

[Case: Should be able to log system logs 58](#_Toc66128854)

[Case: Should be able to log error logs related to program run 59](#_Toc66128855)

[Case: Should not show the logs in the console 60](#_Toc66128856)

# Welcome page

## Case: Should be able to show welcome page when the application is launched

Graphical user interface, text

Description automatically generated

# Grid size

## Case: Should be able to allow user to input grid size, before handling file input

Text

Description automatically generated

Text

Description automatically generated

## Case: Should be able to set default grid size, if grid size input is empty

Expected grid size:

5x5

Result:

Text

Description automatically generated

## Case: Should be able to throw error if width input is greater than the upper limit

Allowed grid size limit:

100x100

Result:

Text

Description automatically generated

## Case: Should be able to throw error if width input is less than the lower limit

Text

Description automatically generated

## Case: Should be able to throw error if height input is greater than the upper limit

Text

Description automatically generated

## Case: Should be able to throw error if height input is less than the lower limit

Text

Description automatically generated

## Case: Should be able to throw error if the input is invalid

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

# File input

## Case: Should be able to process commands from file first, before waiting for user manual input on console, if file exists

File location

Graphical user interface, text, application

Description automatically generated

File content

Graphical user interface, application

Description automatically generated

Result

Text

Description automatically generated

## Case: Should be able to wait for user manual input on console directly, if file is not found in the designated location

File location

Graphical user interface, text, application

Description automatically generated

Result

Text

Description automatically generated

## Case: Should be able to process the commands in file input correctly

Text

Description automatically generated

## Case: Should be able to process the file even if there is empty line at file end

File content

Graphical user interface, text, application

Description automatically generated

Result

Text

Description automatically generated

## Case: Should be able to throw error if any of the commands is invalid

File content

Graphical user interface, text, application

Description automatically generated

Result

Text

Description automatically generated

## Case: Should be able to rename file after processing

Graphical user interface, text, application

Description automatically generated

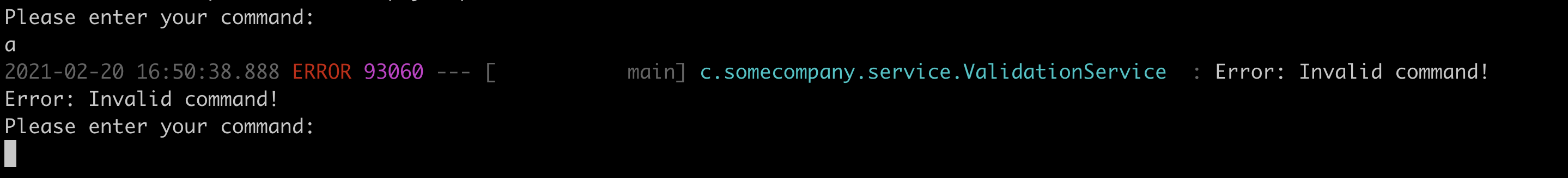
# Manual console input

## Case: Should be able to throw error if user input is empty

Text

Description automatically generated

## Case: Should be able to throw error if command is invalid



## Case: Should be able to throw error if the PLACE command has invalid format (More than 2 parts including the command)

Text

Description automatically generated

## Case: Should be able to throw error if the PLACE command has invalid format (More than 3 elements in the part of the params)

Text

Description automatically generated

## Case: Should be able to throw error if the MOVE command has more than 1 part (it should contain only the command)

Text

Description automatically generated

## Case: Should be able to throw error if the LEFT command has more than 1 part (it should contain only the command)

Text

Description automatically generated

## Case: Should be able to throw error if the RIGHT command has more than 1 part (it should contain only the command)

Graphical user interface, text

Description automatically generated

## Case: Should be able to throw error if the REPORT command has more than 1 part (it should contain only the command)

Graphical user interface, text

Description automatically generated

# PLACE command

## Case: Should be able to place the robot on the table

Graphical user interface, text, application

Description automatically generated

## Case: Should be able to throw error if the x-coordinate param is not an integer

Text

Description automatically generated

## Case: Should be able to throw error if attempt to place the robot on the table with the x-coordinate < 0

Graphical user interface, text

Description automatically generated

## Case: Should be able to throw error if attempt to place the robot on the table with the x-coordinate > 5

Graphical user interface, text

Description automatically generated

## Case: Should be able to throw error if the y-coordinate param is not an integer

Text

Description automatically generated

## Case: Should be able to throw error if attempt to place the robot on the table with the y-coordinate < 0

Graphical user interface, text

Description automatically generated

## Case: Should be able to throw error if attempt to place the robot on the table with the y-coordinate > 5

Graphical user interface, text

Description automatically generated

## Case: Should be able to throw error if the facing is invalid

Text

Description automatically generated

# MOVE command

## Case: Should be able to move robot east

Shape

Description automatically generated with medium confidence

## Case: Should be able to move robot south

Shape

Description automatically generated with medium confidence

## Case: Should be able to move robot west

Shape

Description automatically generated with medium confidence

## Case: Should be able to move robot north

Shape

Description automatically generated with medium confidence

## Case: Should be able to throw error on MOVE command if robot does not have location

Text

Description automatically generated

## Case: Should be able to throw error if robot will fall off table after moving east

Text

Description automatically generated

## Case: Should be able to throw error if robot will fall off table after moving south

Text

Description automatically generated

## Case: Should be able to throw error if robot will fall of table after moving west

Text

Description automatically generated

## Case: Should be able to throw error if robot will fall of table after moving north

Text

Description automatically generated

# LEFT command

## Case: Should be able to turn the robot to face north if it is facing east

Shape

Description automatically generated with medium confidence

## Case: Should be able to turn the robot to face west if it is facing north

Shape

Description automatically generated with medium confidence

## Case: Should be able to turn the robot to face south if it is facing west

Shape

Description automatically generated with medium confidence

## Case: Should be able to turn the robot to face east if it is facing south

Shape

Description automatically generated with medium confidence

## Case: Should be able to throw error on LEFT command if robot does not have location

Text

Description automatically generated

# RIGHT command

## Case: Should be able to turn the robot to face south if it is facing east

Shape

Description automatically generated with medium confidence

## Case: Should be able to turn the robot to face west if it is facing south

Shape

Description automatically generated with medium confidence

## Case: Should be able to turn the robot to face north if it is facing west

Shape

Description automatically generated with medium confidence

## Case: Should be able to turn the robot to face east if it is facing north

Shape

Description automatically generated with medium confidence

## Case: Should be able to throw error on RIGHT command if robot does not have location

Text

Description automatically generated

# REPORT command

## Case: Should be able to report the robot's location

A black background with white text

Description automatically generated with low confidence

## Case: Should be able to throw error on REPORT command if robot does not have location

Text

Description automatically generated

# Logging

## Case: Should be able to log system logs

Graphical user interface, text

Description automatically generated

## Case: Should be able to log error logs related to program run

Graphical user interface, text

Description automatically generated

## Case: Should not show the logs in the console

A picture containing text

Description automatically generated