COEN 6761 Project Deliverable 1

1st Mayank Gupta
Dept. E.C.E.
40198760
Concordia University
mayank.pkgupta@gmail.com

2nd Krishna Bhatt
Dept. E.C.E.
40183716
Concordia University
bhattkrishna605@gmail.com

3rd Uday Putreddy
Dept. E.C.E.
40182878
Concordia University
udayreddy1947@gmail.com

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

As described in the course, humans are prone to making errors. Hence, wherever humans are involved, there are high chances of errors being made. This fact makes testing an essential activity. Software testing is the process of assessing and confirming that a software application or product performs as intended. Testing has advantages such as bug prevention, lower development costs, and better performance. This software testing project involves evaluating and verifying the functionality, performance, and reliability of a software application to ensure that it meets the specified requirements and operates correctly in various environments. The testing process will involve designing and executing test cases, identifying and fixing defects, and documenting results to ensure that the software application is of high quality and meets industry standards.

II. Goal

The goal of this project is to deliver a software application that is user-friendly, efficient, and meets customer expectations. This project is mainly to learn and practice the software testing techniques covered in the course which are used in the real world. The testing process includes execution of created test cases, study the usage of java code coverage tools to verify the functions of the code, White-box, Black-box and regression testing. All these tasks will be performed and delivered in four tasks respectively.

III. Problem Statement

To develop and test a program that simulates the robot capabilities to move on the floor. This project involves creating a robot simulation in Java that moves around a room represented by an N by N array called floor. The robot holds a pen that can be in either the up or down position. When the pen is down, the robot traces shapes on the floor and when it is up, the robot moves freely without writing. The robot starts at position [0, 0] facing north, with its pen in the up position.

IV. Man Hours Distribution

A. Man Hours

By assessing and calculating the need, we determined that team members would work 11 to 13 hours each week in order to finish all of the project's tasks within the allotted time for the tasks. The team is divided up each day of the week as follows:

TABLE I
Man Hours Distribution

Weekday	**Mayank**	**Krishna**	**Uday**
Monday	2 Hours	3 Hours	1 Hours
Tuesday	1 Hours	1 Hour	2 Hours
Wednesday	2 Hours	2 Hours	3 Hours
Thursday	1 Hours	1 Hours	1 Hours
Friday	1 Hours	2 Hour	2 Hours
Saturday	2 Hours	2 Hour	1 Hour
Sunday	3 Hours	2 Hours	2 Hour

To complete the project, we're employing an agile sprint methodology.

B. Sprint Cycle

TABLE II
Sprint Table

Sprint	**Week**	**Start Date**	**Completion Date**
Sprint 1	Week 1	9 Jan	15 Jan
	Week 2	16 Jan	22 Jan
Sprint 2	Week 3	23 Jan	29 Jan
	Week 4	30 Jan	4 Feb
Sprint 3	Week 5	5 Feb	9 Feb

We started analyzing and finding the initial requirements of the project in week-1. By the end of sprint-1, we were able to program the functions satisfying the requirements. In sprint-2, the task of adding requirements and programming them was our major work. By the end of sprint-2, we completed the development of the program for robots and started creating unit test cases. Unit test cases were created by the end of week-5.

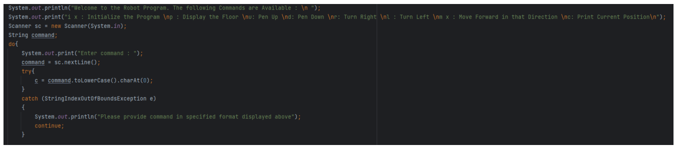
V. Requirements

TABLE III
Requirements Table

R1	The user should be able to choose the following command for the robot. Commands(i,m,c,r,l,p,u,d) irrespective of the case.
R2	Initialization - The floor size, Floor array, robot position, pen direction are to be initialized when ‘i’ command is used.
R3	The user should be able to set the position of the pen to UP when the ‘u’ command is used.
R4	The user should be able to set the position of the pen to DOWN when the ‘d’ command is used.
R5	The user should be able to turn the robot’s direction to Right when the ‘r’ command is used.
R6	The user should be able to turn the robot’s direction to Left when the ‘l’ command is used
R7	The current position of the robot should be displayed when the ‘c’ command is used.
R8	The robot should move forward to the given ‘s’ number of spaces when [M s] command is used.
R9	The NxN matrix should be displayed when the ‘p’ command is entered
R10	The [M s] and [I s] command should have M—I followed by zero or one space followed by an integer s.
R11	The [M s] or [I s] should have s as an Integer greater than 0.
R12	The position of the robot should be updated when [M s] command is used irrespective of the pen being up or down.
R13	The robot should not move if the given space in [M s] is greater than available space.
R14	The other commands should not execute until the floor is not initialized.
R15	The program must end when the ‘q’ command is pressed.
R16	Array floor is not changed to ‘1’ , when pen is UP
R17	Robot should have Position: 0, 0 - Pen: up - Facing: north everytime floor is initialized

VI. Screenshots Of Functions Mapped

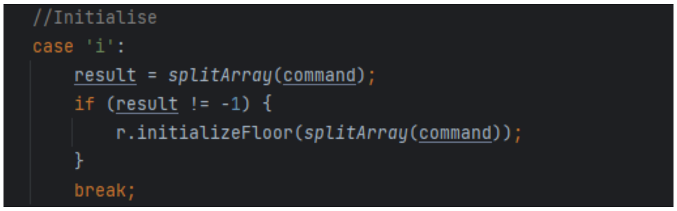
A. Fulfilling Requirement R1



```
System.out.println("Welcome to the Robot Program. The following Commands are Available : \n ");
System.out.print("i x : Initialize the Program \np : Display the Floor \nu: Pen Up \nd: Pen Down \nr: Turn Right \nl : Turn Left \nm x : Move Forward in that Direction \nc: Print Current Position\n");
Scanner sc = new Scanner(System.in);
String command;
do{
    System.out.print("Enter command : ");
    command = sc.nextLine();
    try{
        c = command.toLowerCase().charAt(0);
    }
    catch (StringIndexOutOfBoundsException e)
    {
        System.out.println("Please provide command in specified format displayed above");
        continue;
    }
```

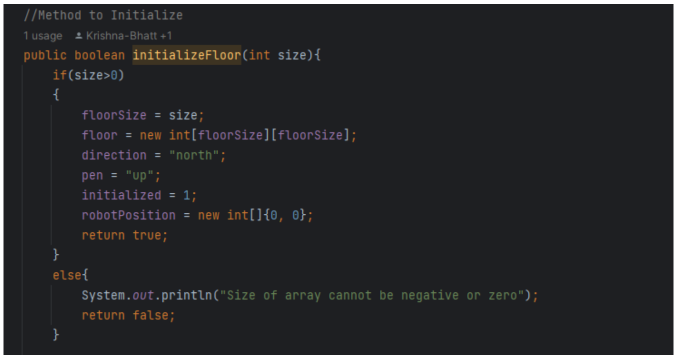
Fig. 1. Screenshot fulfilling R1

B. Fulfilling Requirement R2 & R17



```
//Initialise
case 'i':
    result = splitArray(command);
    if (result != -1) {
        r.initializeFloor(splitArray(command));
    }
    break;
```

Fig. 2. Screenshot fulfilling R2 & R17 Part 1



```
//Method to Initialize
1 usage  Krishna-Bhatt +1
public boolean initializeFloor(int size){
    if(size>0)
    {
        floorSize = size;
        floor = new int[floorSize][floorSize];
        direction = "north";
        pen = "up";
        initialized = 1;
        robotPosition = new int[]{0, 0};
        return true;
    }
    else{
        System.out.println("Size of array cannot be negative or zero");
        return false;
    }
```

Fig. 3. Screenshot fulfilling R2 & R17 Part 2

References

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.