Sprint 3 - Agility Design Document

December 4, 2023

Table of Contents

[1. Executive Summary 3](#_Toc152603049)

[1.1 Project Overview 3](#_Toc152603050)

[1.2 Purpose and Scope of this Specification 3](#_Toc152603051)

[2. Product/Service Description 3](#_Toc152603052)

[2.1 Product Context 3](#_Toc152603053)

[2.2 User Characteristics 3](#_Toc152603054)

[2.3 Assumptions 3](#_Toc152603055)

[2.4 Constraints 3](#_Toc152603056)

[2.5 Dependencies 3](#_Toc152603057)

[3. Requirements 4](#_Toc152603058)

[3.1 AGILITY: 4](#_Toc152603059)

[3.2 Functional Requirements 4](#_Toc152603060)

[3.3 Security 5](#_Toc152603061)

[3.3.1 Protection 5](#_Toc152603062)

[3.3.2 Authorization and Authentication 5](#_Toc152603063)

[3.4 Portability 5](#_Toc152603064)

[4. Requirements Confirmation/Stakeholder sign-off 5](#_Toc152603065)

[5. System Design 6](#_Toc152603066)

[5.1 Algorithm 6](#_Toc152603067)

[5.2 System Flow 6](#_Toc152603068)

[5.3 Software 7](#_Toc152603069)

[5.4 Hardware 8](#_Toc152603070)

[5.5 Test Plan 8](#_Toc152603071)

[5.6 Task List/Gantt Chart 11](#_Toc152603072)

[5.7 Staffing Plan 11](#_Toc152603073)

# Executive Summary

## Project Overview

This project is to test our groups and individual ability to use software engineering skills to design a robot to follow a set of steps for Professor Eckert. All of which display agility for optimal performance in the robotics Triathlon.

<https://monmouth.desire2learn.com/d2l/le/content/316748/viewContent/3838976/View>

## Purpose and Scope of this Specification

This project is going to run a robot around HH208 with certain measurements to successfully display its agility to Professor Eckert. The measurements, speaking a set of words, code program, obstacles, plan, and all requirements will be within scope. Out of scope would be anywhere the robot rolls outside of these measurements or does not speak the correct set of words at the proper location or does not follow the correct code, plan and requirements.

# Product/Service Description

The general factors that affect the product are the code used to run the robot and any outside factors that will disturb its track. Also, the condition of the robot could affect it, all of which give reason to why we need specific requirements, so the robot runs the correct course.

## Product Context

The robot is related to the Sphero coding system program by a Bluetooth connection. The code directs the robot to do certain tasks which are made up by the user (our group).

## User Characteristics

Students - Our group of three will be users of this product. We all have limited experience with it and our technical expertise is not the best, but we all have a general grasp of how to use it. We are in the learning stage and have the basic skills to design this robot and use it.

Professor - Professor Eckert has plenty of experience with this product and has enough technical expertise to teach a class on it.

## Assumptions

Some assumptions that could affect the requirements.

- Lost equipment

- Not knowing the correct code to go with correct measurements

- The Hh208 room not being available

- Broken equipment

- If the Sphero site is not working

-Uneven floor

-The tape affecting the robot rolling

-Not having the obstacles

## Constraints

- Due dates

- Resources

- HH208 room availability

- Types of code blocks available

## Dependencies

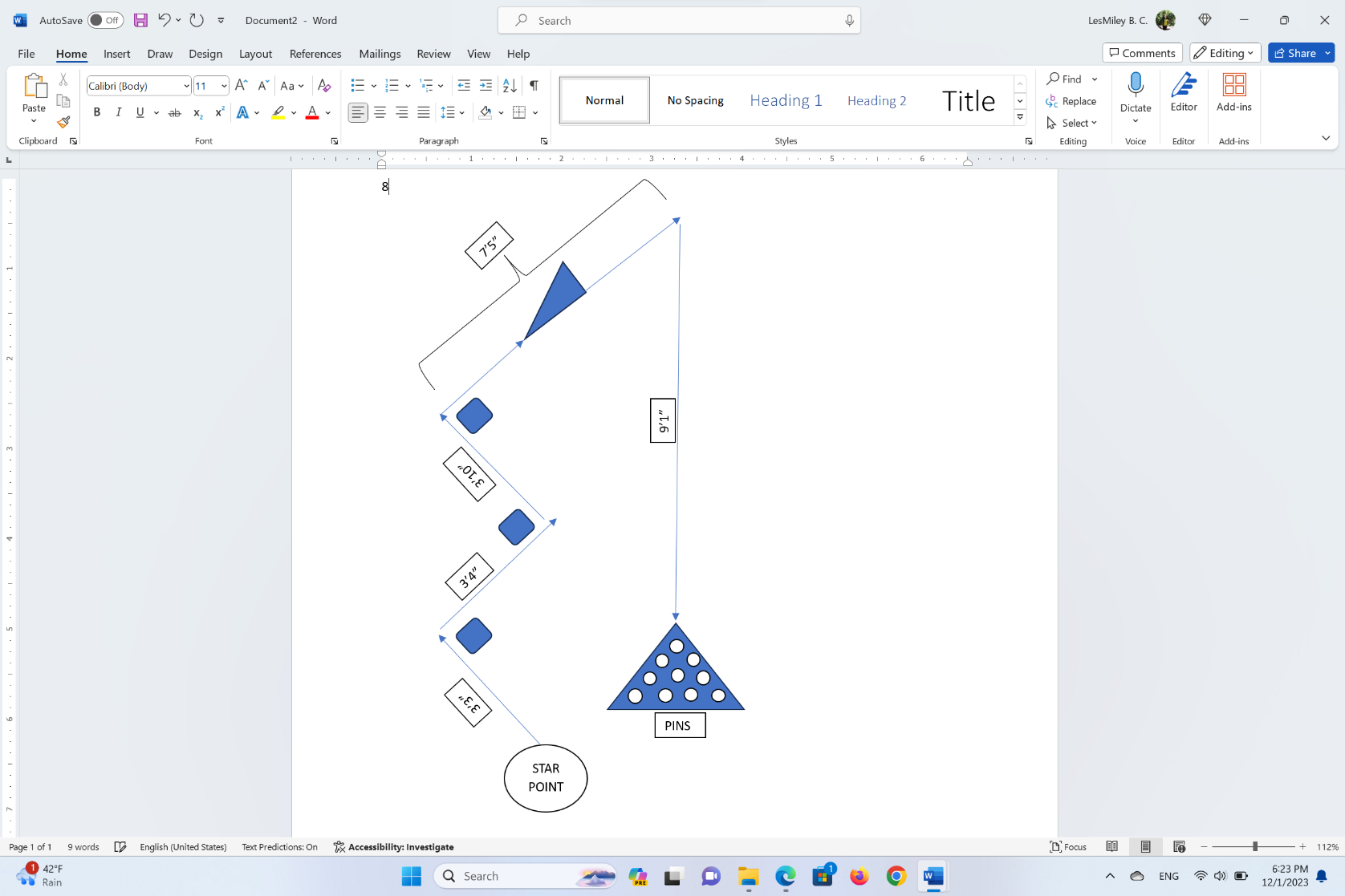
- Coding must be completed for the robot to be tested

- The room must be available to test the robot

# Requirements

## AGILITY:

1. On the floor of room HH208, there will be a obstacle course marked, which will be the route that the robot will run. (See figure below)
2. The robot will be placed at the starting point.
3. At the starting point the robot will start running a distance of 3'3".
4. It will avoid an obstacle, making a right of 90ᵒ and advancing for 3'4" distance.
5. It will then avoid the second obstacle by making a left with a 90ᵒ and continue for 3'10".
6. The robot will avoid a third obstacle, turning to the right with an angle of 90ᵒ and for a distance of 7'5".
7. The robot will go over an inclined ramp.
8. The robot will make a right turn at an angle of approximately 45ᵒ and will move directly for a distance of 9'1".
9. The robot will impact with a group of pins and knock them down.



## Functional Requirements

| Req# | Requirement | Comments | Priority | Date Rvwd | SME Reviewed / Approved |
| --- | --- | --- | --- | --- | --- |
| AGILITY\_1 | At the starting point the robot will start runninga distance of 3'3". | Meets the requirements |  | 11/28/23 | Approved |
| AGILITY\_2 | It will avoid an obstacle, making a right of 90ᵒ and advancing for 3'4" distance | Meets the requirements |  | 11/28/23 | Approved |
| AGILITY\_3 | It will avoid the second obstacle by making a left with a 90ᵒ and continue for 3'10" | Meets the requirements |  | 11/28/23 | Approved |
| AGILITY\_4 | The robot will avoid a third obstacle, turning to the right with an angle of 90ᵒ and for a distance of 7'5". | Meets the requirements |  | 11/30/23 | Approved |
| AGILITY\_5 | In the middle of the 7'5" distance, the robot will go over an inclined ramp. | Meets the requirements |  | 11/30/23 | Approved |
| AGILITY\_6 | The robot will make a right turn at an angle of approximately 45ᵒ and will move directly for a distance of 9'1". | Meets the requirements |  | 12/04/23 | Approved |
| AGILITY\_7 | The robot must impact with a group of pins, in order to knock them down. | Meets the requirements |  | 12/04/23 | Approved |

## Security

### Protection

There is one chosen person to hold onto the robot, making sure it does not go missing. There is only allowed one Bluetooth connection at a time so there is no accidental access. All activity with the robot is logged as well to make sure everything is in order.

### Authorization and Authentication

You need an account to access the Sphero app and the newly released version. You must also have an authorized computer.

## Portability

The robot's portability is versatile. The robot can be run anywhere and will run the correct route if it does not hit any obstacles. As for writing the code, the Sphero site is not so versatile. You cannot do all it offers on Windows, so we must use a Mac computer for the coding to work and get the graphs. But besides that, the robot is easily portable and can be used anywhere with enough space and internet and Bluetooth connections.

# Requirements Confirmation/Stakeholder sign-off

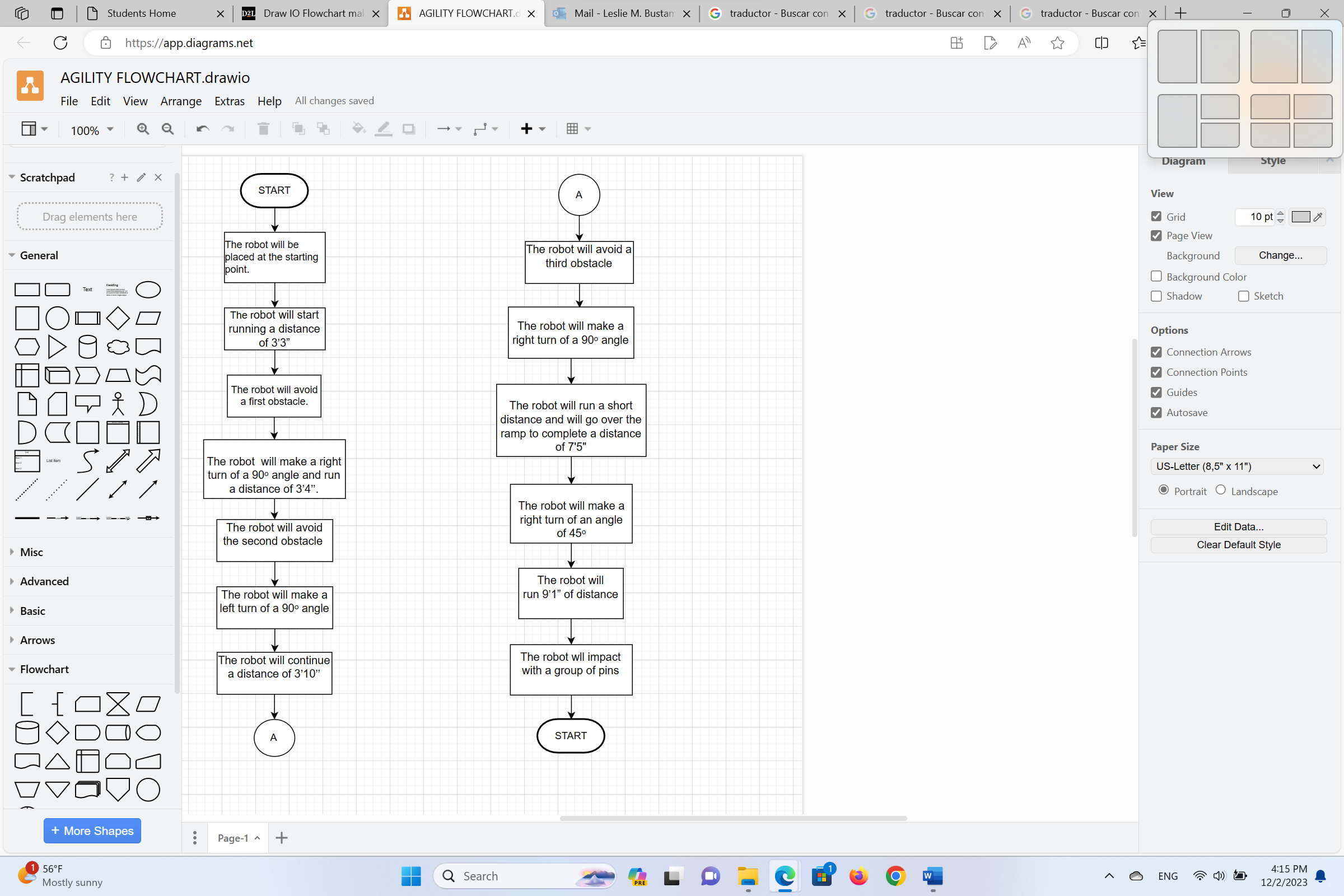
|  |  |  |
| --- | --- | --- |
| Meeting Date | Attendees (name and role) | Comments |
| 11/7/23 | Jon Veltri, Jack McGovern, Leslie B | Confirmed all |

# System Design

## Algorithm

* The robot will be placed at the starting point.
* The robot will start running a distance of 3’3”.
* The robot will avoid a first obstacle. .
* The robot will make a right turn of a 90ᵒ angle and run a distance of 3’4’’.
* The robot will avoid the second obstacle
* The robot will make a left turn of a 90ᵒ angle.
* The robot will continue a distance of 3’10’’.
* The robot will avoid a third obstacle.
* The robot will make a right turn of a 90ᵒ angle.
* The robot will run a short distance and will go over the ramp to complete a distance of 7'5".
* The robot will make a right turn of an angle of 45ᵒ.
* The robot will run 9’1” of distance.
* The robot will impact with a group of pins.

## System Flow



## Software

The official and supported Sphero EDU app was used to develop and run the code. The Sphero program utilizes block code (Shown below) to program the robots.

A screenshot of a phone

Description automatically generatedA graph on a screen

Description automatically generated

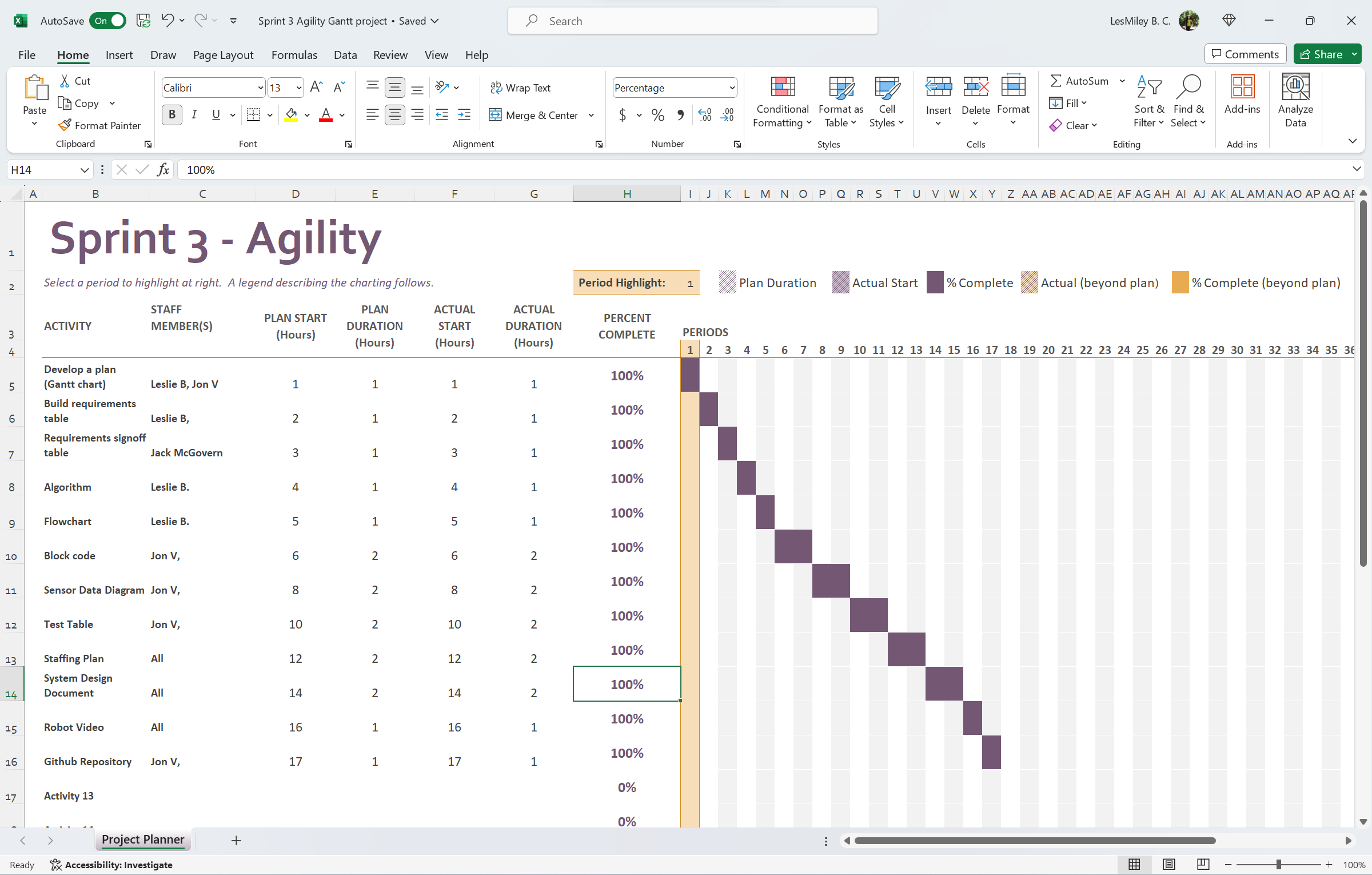
## Hardware

Laptops and desktops were used to develop the code and test the application on the Sphero BOLT robot.

## Test Plan

| **Reason for Test Case** | **Test Date** | **Expected Output** | **Observed Output** | **Staff Name** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| Testing to see if the robot runs the first line, turns 90 degrees and avoids the first obstacle | 11/30/23 | Robot is going to run down the first length, stop and turn 90 degrees | The robot went too far over the line but turned at the right angle | Jon Veltri | Fail |
| Testing to see if the robot runs the first line, turns 90 degrees and avoids the first obstacle | 11/30/23 | Robot is going to run down the first length, stop and turn 90 degrees | The robot was placed slightly off-angle, resulting in it hitting the first obstacle. | Jon Veltri | Fail |
| Testing to see if the robot runs the first line, turns 90 degrees and avoids the first obstacle | 11/30/23 | Robot is going to run down the first length, stop and turn 90 degrees | The robot overran the line once again. | Jon Veltri | Fail |
| Testing to see if the robot runs the first line, turns 90 degrees and avoids the first obstacle | 11/30/23 | Robot is going to run down the first length, stop and turn 90 degrees | The robot ran too short and bumped into the first obstacle. | Jon Veltri | Fail |
| Testing to see if the robot runs the first line, turns 90 degrees and avoids the first obstacle | 11/30/23 | Robot is going to run down the first length, stop and turn 90 degrees | The robot stopped and turned exactly where it needed to | Jon Veltri | Pass |
| Testing to see if the robot drives down the second line and turns 90 degrees at the correct spot. | 11/30/23 | The robot is going to run the length of the second line, stopping at the end and turning 90 degrees in the other direction. | The robot ran too short and turned in the wrong direction. | Jon Veltri | Fail |
| Testing to see if the robot drives down the second line and turns 90 degrees at the correct spot. | 11/30/23 | The robot is going to run the length of the second line, stopping at the end and turning 90 degrees in the other direction. | The robot went too short again and bumped into the second obstacle. | Jon Veltri | Fail |
| Testing to see if the robot drives down the second line and turns 90 degrees at the correct spot. | 11/30/23 | The robot is going to run the length of the second line, stopping at the end and turning 90 degrees in the other direction. | The robot went over the line slightly and turned in almost the right direction. | Jon Veltri | Fail |
| Testing to see if the robot drives down the second line and turns 90 degrees at the correct spot. | 11/30/23 | The robot is going to run the length of the second line, stopping at the end and turning 90 degrees in the other direction. | The robot turned in the right direction but went too far. | Jon Veltri | Fail |
| Testing to see if the robot drives down the second line and turns 90 degrees at the correct spot. | 11/30/23 | The robot is going to run the length of the second line, stopping at the end and turning 90 degrees in the other direction. | The robot turned in the right direction, avoided the obstacle, and stopped at the correct spot. | Jon Veltri | Pass |
| Testing to see if the robot runs the third line, stops, and turn towards the right direction | 12/3/23 | The robot is going to run the distance of the third line, stop for a second, and turn and move towards the right direction. | The robot overran the line and turned in the wrong direction. | Jon Veltri | Fail |
| Testing to see if the robot runs the third line, stops, and turn towards the right direction | 12/3/23 | The robot is going to run the distance of the third line, stop for a second, and turn and move towards the right direction. | The robot was placed slightly off and hit the first and second obstacle. | Jon Veltri | Fail |
| Testing to see if the robot runs the third line, stops, and turn towards the right direction | 12/3/23 | The robot is going to run the distance of the third line, stop for a second, and turn and move towards the right direction. | The robot stopped at the right spot and started moving in the right direction towards the jump. | Jon Veltri | Pass |
| Testing to see if the robot turns in the right direction following the jump and runs down the final line towards the markers. | 12/3/23 | The robot is going to turn towards the markers after the jump and start rolling towards it. | The robot jumped off the binder the wrong way and the direction couldn’t be tested | Jon Veltri | Fail |
| Testing to see if the robot turns in the right direction following the jump and runs down the final line towards the markers. | 12/3/23 | The robot is going to turn towards the markers after the jump and start rolling towards it. | The robot turned in the completely wrong direction following the jump. | Jon Veltri | Fail |
| Testing to see if the robot turns in the right direction following the jump and runs down the final line towards the markers. | 12/3/23 | The robot is going to turn towards the markers after the jump and start rolling towards it. | The robot turned in a more accurate angle towards the markers but still missed. | Jon Veltri | Fail |
| Testing to see if the robot turns in the right direction following the jump and runs down the final line towards the markers. | 12/3/23 | The robot is going to turn towards the markers after the jump and start rolling towards it. | The robot turned in the right angle following the jump and followed the line, colliding with the markers | Jon Veltri | Pass |
| Final Test | 12/3/23 | The robot will complete the entire course and collide with the markers | The robot was placed slightly wrong and collided with an obstacle. | Jon Veltri | Fail |
| Final Test | 12/3/23 | The robot will complete the entire course and collide with the markers | The robot overturned at the end and missed the markers | Jon Veltri | Fail |
| Final Test | 12/3/23 | The robot will complete the entire course and collide with the markers | The robot successfully completed the course and collided with the markers | Jon Veltri | Pass |

## Task List/Gantt Chart



## Staffing Plan

| Name | Role | Responsibility | Reports To |
| --- | --- | --- | --- |
| Jon V. | Coding | Code the robot, setup the GitHub, record the robot video | Everyone |
| Jack M. | SDD | Fill in the missing information in the system design document | Everyone |
| Leslie B. | Gantt Chart / SDD | Fill in the missing information in the system design document; Also fill out the Gantt chart | Everyone |