CS104 - Sprint 3 Agility – System Design Document

Kathryn Schauer, Nicholas Smythe, Joseph Scerbo

Kathryn Schauer

**Algorithm:**

* Start the program
* Input roll at 0 degrees at 50 speed for 2 seconds
* Delay for 2 seconds
* Input roll at 93 degrees at 50 speed for 2.1 seconds
* Delay for 2 seconds
* Input roll at 0 degrees at 50 speed for 2.5 seconds
* Delay for 2 seconds
* Input roll at 90 degrees at 140 speed for 2.5 seconds
* Delay for 2 seconds
* Input 225 degrees at 210 speed for 2.3 seconds
* End the program

**Executive summary:**

The products intended audience is the professor and our classmates and maybe even some outsiders just observing and watching the project. This project is a robot and it can roll around at different speeds, talk, change color for a certain amount of time, and much more.

The purpose of the project in the scope, is to show that the robot can go around an obstacle course. It cannot go off course like straying from the course direction, bump into any objects, and it has to end at the starting point. Outside the scope, the project has to be coded in such a way that no errors happen like bumping or going off course and everything needs to run into order so no malfunctions or mistakes happen while the robot is running.

**Product description:**

This product which is the robot does relate to other products out in the world. One of the products would be the Roomba which a robot or machine that rolls around and sucks up dirt and dust like vacuums. The robot is self-contained and is independent as long someone programs the robot to do certain actions. It has sensors that make the robot change directions whenever it bumps into any object. The robot can interface with any computer systems like laptops, desktops, computers, etc. which are the larger systems. The interconnections would probably be the computer system and coding controlling the robot’s actions. The external interfaces would be the Wi-Fi that is connecting the computer system and the robot.

The customer profile for a student would be that they would have to have some experience and technical expertise in how to write an algorithm, flowchart, syntax language, coding and some knowledge in software engineering to use the robot. They would also have to practice and experiment using the robot a couple of times to know how to use it well.

The customer profile for the faculty/staff would be having complete knowledge and a lot of experience and technical expertise on how to set up this program and how the program will work to use the robot. They will have to know the ins and outs of the robot’s characteristics completely. The customer profile for people that do not know how to use the robot or what is does would probably learning more about computer science, software engineering and observing the students and faculty/staff.

Some assumptions that may affect the requirements would be the equipment availability. If you cannot access the equipment then you will have to change everything around. You will have to contact the person in charge of the robots and see when you can get the robot as soon as possible and then instead of working on the code first, change the Gantt chart and work on the staff plan, executive summary, product description and requirements chart if you have not already done the algorithm and flow chart in the meanwhile. Also, the same would go if we still need to use the robot even after we give it back or need to use Howard Hall room to test robots and it isn’t available.

Some items that will cause constraints would be if the app or even the computer system device itself that is being used to control the robot is malfunctioning. For example, if it is lagging or it is just not working or letting you do certain commands as normal. Perhaps the code is too long and there is an overload on the device so its robot will not be able to perform the activity that it is supposed to do. The people working on the project may also just be having a hard time figuring out the coding or how to work the robot as well.

The dependencies that will affect the requirements are doing the project in the right order. First of all, you will need to sign out the robot to even just experiment and test it. Before even testing or writing the code, you will need to understand what you are trying to do with the program and then write the algorithm and then flowchart. Then you will need to understand the program that you are using to control the robot. After that, you will need to try testing the code that is written to see if it works multiple times.

**Requirements:**

The (#) show the requirement number. The robot is running an obstacle course. The first requirement will be to start on an x and then try to reach the next destination while avoiding three obstacles in the way. While trying to get to the next destination, the robot needs to roll onto and over a ramp. Lastly, the robot should try to knock down all of the pins and then stop at the starting place’s x which ends the obstacle course. The robot needs to do all of this without going off course.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req #** | **Requirement** | **Comments** | **Priority** | **Date Rvwd** | **Reviewed/**  **Approved** |
| 1. Agility | Robot needs to start at the x | Robot can go through course | 1 | 12/1/19 | Nicholas  Kathryn  Joseph |
| 2. Agility | Go through the obstacles within the parameters | Make sure the robot does not stray from the path and knock down any obstacles | 2 | 12/1/19 | Nicholas  Kathryn  Joseph |
| 3. Agility | Roll onto and over a ramp | The robot must worry about the bump so it will not go off course | 3 | 12/1/19 | Nicholas  Kathryn  Joseph |
| 4. Agility | Knock down all the pins | Make sure the speed is alright and robot does not take damage running into pins | 4 | 12/1/19 | Nicholas  Kathryn  Joseph |
| 5. Agility | Stopping at the last x | Robot need to stop and end on the x it started from | 5 | 12/1/19 | Nicholas  Kathryn  Joseph |

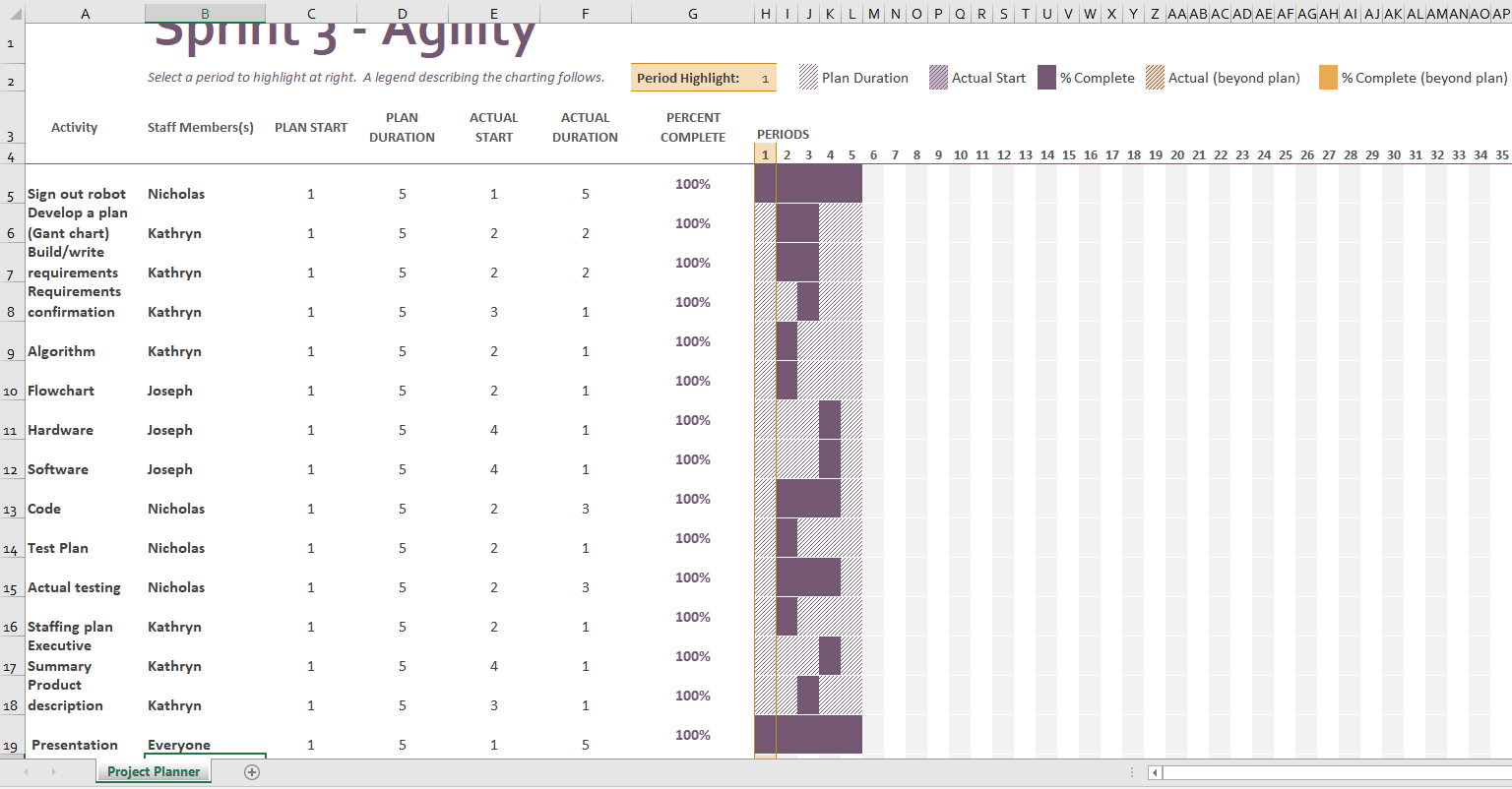
In order to protect the code, one could set up some type of program or defense system to pop up whenever an unknown user is trying to access the code. Also, a password could be set to access the code. Make sure to save the code in multiple locations as well in case the code gets altered or deleted by accident.

In order to be authorized to use and or see the code, the app Sphero Edu requires people to create an account to access their own code. Like protecting the code, one could also download or create multiple programs for someone to be able to access their own code.

It is possible to port the robot’s builder code into JavaScript language on the Sphero Edu app. It is also possible to convert the code created from the robot to a phone from a computer or vice versa as long as you have a Sphero Edu account. It is easy to move the robot because it is small and portable and it is not hard to charge it either.

**Requirements confirmation:**

|  |  |  |
| --- | --- | --- |
| **Meeting Date** | **Attendees** | **Comments** |
| 11/25/19 | Nicholas - Code writer  Joseph – Planner | Confirmed  Started working on project and code |
| 12/1/19 | Kathryn - Information gatherer  Nicholas - Code writer | Confirmed  Testing robot and finalizing code. |
| 12/2/19 | Kathryn - Information gatherer  Nicholas - Code writer  Joseph - Planner | Confirmed  Finalizing all documents and reviewing before submitting |

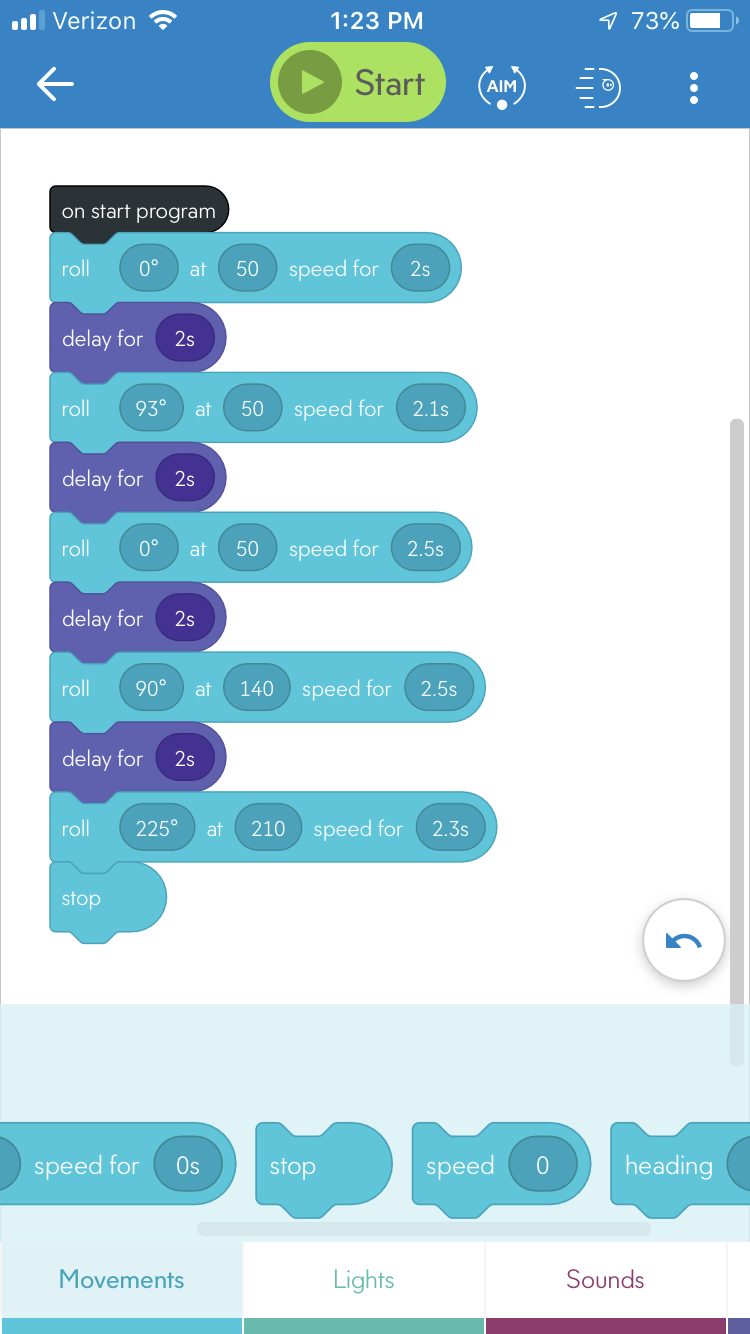
**Gantt chart: **

**Staffing plan:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Responsibility** | **Reports To** |
| Kathryn | Information gatherer | Submitting all documents together on GitHub, reviewing documents, & making team collaborate | 12/1/19 submitting all of the documents |
| Nicholas | Code writer | Creating the code, testing the code out, making adjustments to the code | 12/1/19 finalizing code |
| Joseph | Planner | Planning the flowchart and software and hardware information. | 12/1/19 making sure everything is ok |

Nicholas Smythe

**Code:**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reason for Test Case | Test Date | Expected Output | Observed Output | Pass/Fail |
| Go from 70 to 60 | 12/2 | Go within the line for the turn to complete the sprint | Overshot a little Just need to decrease the speed by a tad bit and should be fine | Fail |
| Go from 60 to 50 on first speed | 12/2 | Pass from conclucion of test 1 | It worked for the test conducted | Pass |
| Go from 70 to 60 | 12/2 | Decrease the speed to not overshoot | Overshot the marker again | Fail |
| Go from the turn of 90 degrees to 93 for more maneuverability | 12/2 | To follow in the line perfectly and complete it | Completed just a little off at the end will adjust speed and time | Pass |
| 40 to 45 for the 3rd turn | 12/2 | Too completely clear the 3rd turn and complete the entire sprint | A little short since the turn was adjusted | Fail |
| 45 to 50 for the 3rd turn | 12/2 | To completely pass the 3rd turn | Perfectly adjusted the speed for the test now needs to fix the last area to decrease the speed and add a stop | Pass |
| Decreased speed from 225 to 210 | 12/2 | To hit all the markers in the triangle and stop | Hits the triangle and all the markers | Pass |

**Tested robot**

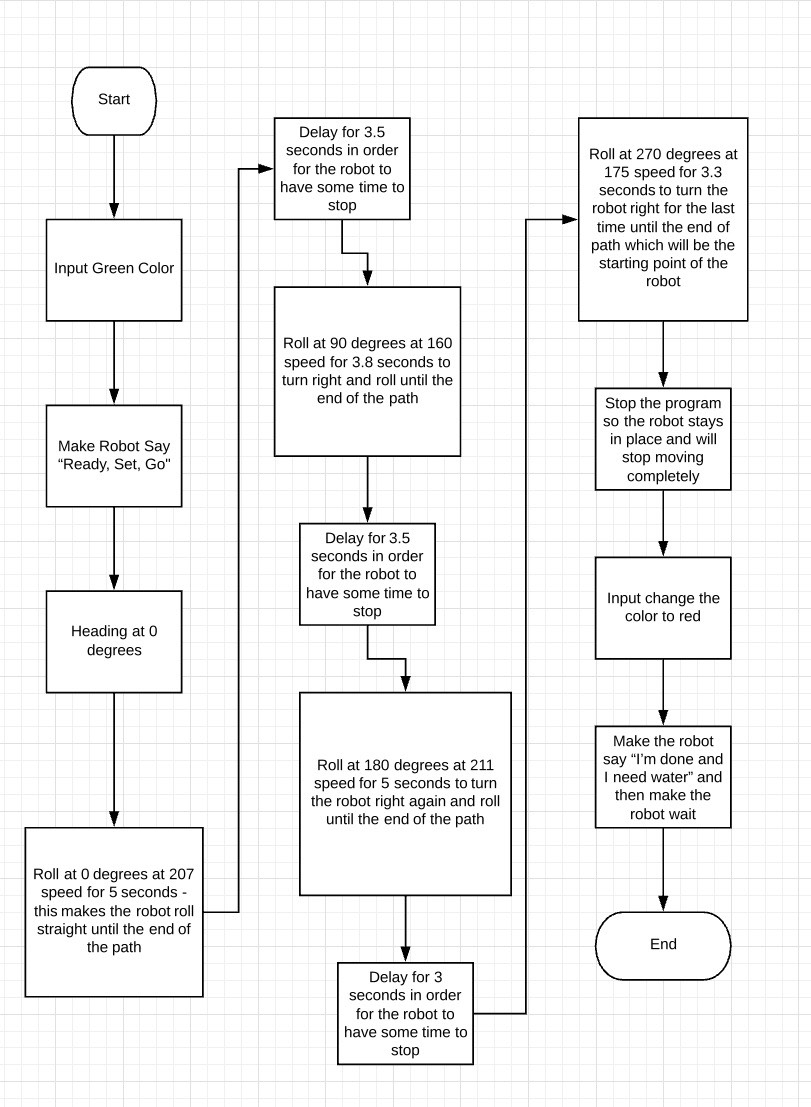
**Signed out robot**

Joseph Scerbo

**Hardware:** The hardware that the robot uses is the computer and the phone or any computer system that can use the website/app Sphero Edu that is used to be able to work on the program. The robot itself is also part of the hardware that is used in the testing and development part of the project/product.

**Software:** The software that the robot uses is the Sphero Edu app which you can manually program the actions and functions or you can write JavaScript code since it is also part of the app. It is also possible to convert the programs functions and actions into JavaScript. As long as there is software, the program will be able to run its course.

**Flowchart:**

****