CS104 - Sprint 1 Endurance – System Design Document

Kathryn Schauer, Nicholas Smythe, Joseph Scerbo

Kathryn Schauer

**Algorithm:**

* Start the program
* Input the color green
* Make the robot say “Ready, set, go” and then continue the program.
* Heading at 0 degrees
* Roll at 0 degrees at 207 speed for 5 seconds - this makes the robot roll straight until the end of the path
* Delay for 3.5 seconds in order for the robot to have some time to stop
* Roll at 90 degrees at 160 speed for 3.8 seconds to turn right and roll until the end of the path
* Delay for 3.5 seconds in order for the robot to have some time to stop
* Roll at 180 degrees at 211 speed for 5 seconds to turn the robot right again and roll until the end of the path
* Delay for 3 seconds in order for the robot to have some time to stop
* Roll at 270 degrees at 175 speed for 3.3 seconds to turn the robot right for the last time until the end of path which will be the starting point of the robot
* Stop the program so the robot stays in place and will stop moving completely
* Input change the color to red
* Make the robot say “I’m done and I need water” and then make the robot wait
* 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, and much more.

The purpose of the project in the scope, is to show that the robot can go around the classroom four feet from the outside wall. It cannot go off course, bump into any objects, and it has to end at the starting point. It also has to glow green at first and say “ready set go” and then at the end it has to glow red and say “I’m done and I need water.” 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 first requirement (1) will be to contact the person that is in charge of the robots to set a date to sign out a robot in Howard Hall in order to use that robot later to create and test the codes. Next, we should assign the staffing plan (1) to make sure everyone has their own roles and responsibilities so that they will know that they will have to work on for the project. The Gantt chart (1) will be created to know exactly when everyone should start working on their part, and what parts they are working on in the project. It shows when people have finished their parts and how many tasks there has to be completed in the project. The requirements (1) describe the system requirements in much detail to satisfy and verify the system requirements. The algorithm (1) are the extremely important instructions of the program that show what exactly what should be in the code and how the code should look like the exact units/numbers, equations and what it is that is being written. The flow chart (1) is the algorithm converted into a chart like picture that is easier to understand since it visually shows what the code of the project should look like in order. The coding (1) affects how the robot will move around and what actions the robot is doing as long as everything is programmed correctly. The test plan (1) shows what mistakes have occurred and the corrections to that mistake. The actual testing (1) is the trial and error that made sure the code was working and running smoothly. The requirements confirmation (2) shows the group confirming what has been done on the days that we were working together on the project. The hardware (3) describes the hardware like the computer systems that created the code and application. The software (3) describes the software like the apps and syntax language used that created the code and application. The executive summary (3) described the objective and purpose of the program. The product description (3) compares the product to other products out there, the customer profiles and the constraints affecting the requirements. The presentation (2) at this point was just created and shared but eventually, it will be about the entire process of creating the project.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req #** | **Requirement** | **Comments** | **Priority** | **Date Rvwd** | **Reviewed/**  **Approved** |
| 1. | Sign out robot | Properly signed out robot | 1 | 10/28/19 | Nicholas |
| 2. | Staffing plan | Assigned people tasks | 1 | 10/28/19 | Kathryn |
| 3. | Gantt chart | Saw if everything was completed | 1 | 10/28/19 | Kathryn |
| 4. | Requirements | Verify system requirements | 1 | 10/28/19 | Joseph |
| 5. | Algorithm | The instructions of how exactly the code and project should run | 1 | 10/28/19 | Kathryn |
| 6. | Flow Chart | Made algorithm into a chart | 1 | 10/28/19 | Joseph |
| 7. | Coding | Created final code | 1 | 10/28/19 | Nicholas |
| 8. | Test Plan | Tested code to see errors | 1 | 10/28/19 | Nicholas |
| 9. | Actual testing | Made sure it ran smoothly | 1 | 10/28/19 | Nicholas |
| 10. | Requirements confirmation | We made sure we did all the documents correctly | 2 | 10/28/19 | Kathryn |
| 11. | Hardware | Described hardware used to demonstrate application | 3 | 10/28/19 | Joseph |
| 12. | Software | Described software used to demonstrate application | 3 | 10/28/19 | Joseph |
| 13. | Executive summary | Summarized objective of robot and project | 3 | 10/28/19 | Kathryn |
| 14. | Product description | Described the robot itself and its requirements | 3 | 10/28/19 | Kathryn |
| 15. | Make presentation | Shared presentation with groupmates | 2 | 10/28/19 | Kathryn |

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** |
| 10/23/19 | Kathryn - Information gatherer  Nicholas - Code writer | Confirmed  Started working on project |
| 10/24/19 | Kathryn - Information gatherer  Nicholas - Code writer | Confirmed  Testing robot and finalizing code |
| 10/28/19 | Kathryn - Information gatherer  Nicholas - Code writer  Joseph - Planner | Confirmed  Finalizing all documents and reviewing before submitting |

**Gantt chart:**

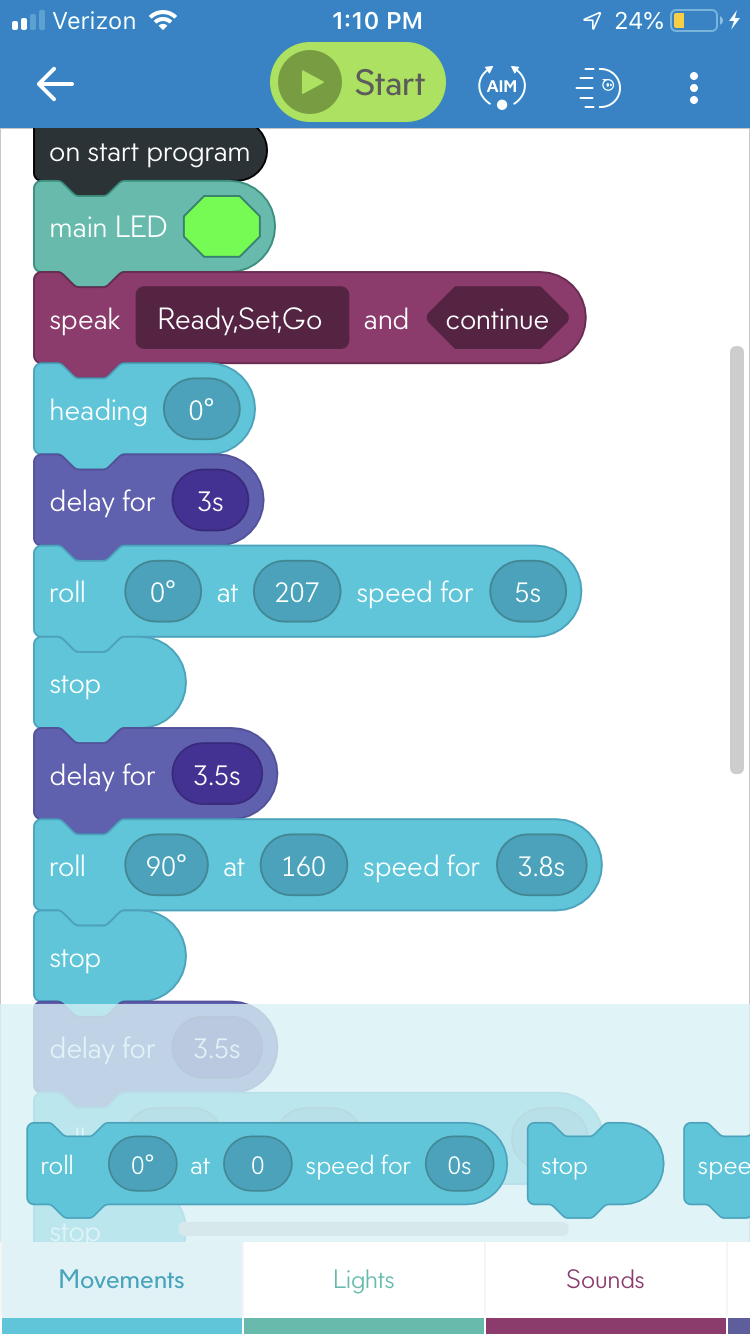
In separate excel sheet document

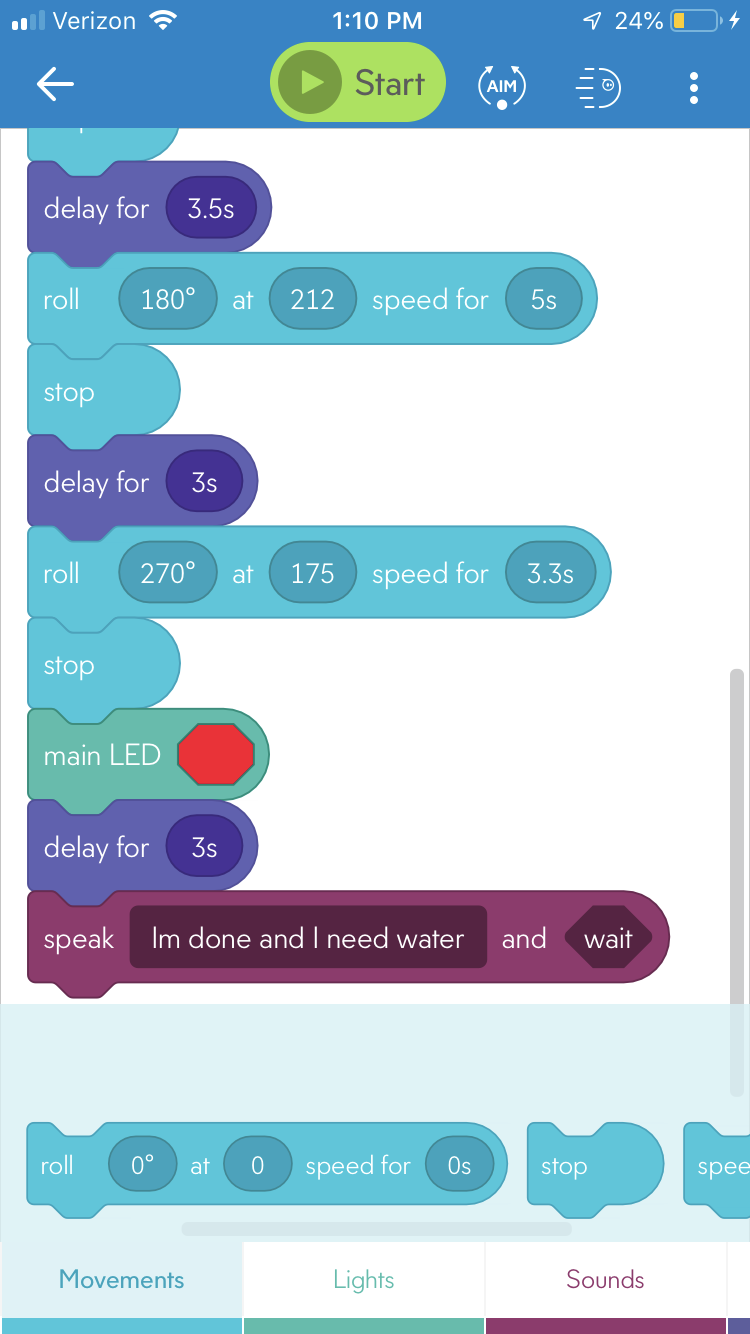
**Staffing plan:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Responsibility** | **Reports To** |
| Kathryn | Information gatherer | Submitting all documents together on GitHub, reviewing documents, & making team collaborate | 10/28/19 submitting all of the documents |
| Nicholas | Code writer | Creating the code, testing the code out, making adjustments to the code | 10/28/19 finalizing code |
| Joseph | Planner | Planning the flowchart and software and hardware information. | 10/28/19 making sure everything |

Nicholas Smythe

**Code:**

****

****

**Test plan:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Reason for Test Case** | **Test Date** | **Expected Output** | **Observed Output** | **Pass/Fail** |
| First Start | 10/24 | To get a gist of the course on the first try. Trying to complete course | Completely off went wide left on first long burst, turned too fast and over shot 2nd burst and under shot 3rd burst and came up short on last burst | Fail for the test conducted |
| Fixing Speed Increased by 20 from 250 to 270 | 10/24 | To fix the issues of the robot going all over the place and making go on a straight path | Went wide right and hit the chair test immediately failed | Fail for the test conducted |
| Decreasing speed from 270 to 240 | 10/24 | Maybe by decreasing the speed the robot will have more of a sense of control to itself and pace itself to move in a straight line | The robot did indeed go straight in the first burst and the 3rd burst but the 2nd and 4th burst were thrown off balance but came up short for the 4th burst | Pass for the test conducted |
| Increasing the delay times by a second for all bursts | 10/24 | By increasing the delay times the robot will be able to have more time to stop and situate itself to move to the next burst | The Bursts were more accurate and allowed for the robot to move in a more controlled manner but were short when it came to speed | Pass for the test conducted |
| Decreased speed by 20 to 220 and increased the time the burst occurred to 4 seconds | 10/24 | By decreasing the speed and increasing the time of the burst it will balance the robot and go more accurately to the target | The target was delivered more accurately but when this happened it came up short | Fail for the test conducted |
| Increasing the time of the burst to 5 seconds for both long bursts and decrease the speed for the first burst to 207 and the 3rd burst to 212 | 10/24 | Increasing the time by a whole second will allow the robot to coast to the destination of the long bursts | This allowed for the robot to hit the target the most accurately the entire test but the 2nd and 4th bursts were thrown off | Pass for the test conducted |
| Increase the 2nd burst to 160 from 140 and decrease the time from 5 to 3.8 | 10/24 | This will allow for the 2nd burst to land on the corner more accurately | It was a success but the last burst was still off completely if copy of the 2nd burst | Pass for the test conducted |
| 4th burst increase from 140 to 175 and decrease the time from 5 to 3.3 | 10/24 | It will land on the X and conduct the test | It was a little off from the X but it is not consistent but gave the results it needed | Pass |
| Overall last test as a whole | 10/24 | It will start and end completely | It hit the X sometimes it did not others based on the rooms floor being indented to make the robots path being altered at the start of the speed test overall if it goes perfectly it will go on the x or very close | Pass |

**Tested robot**

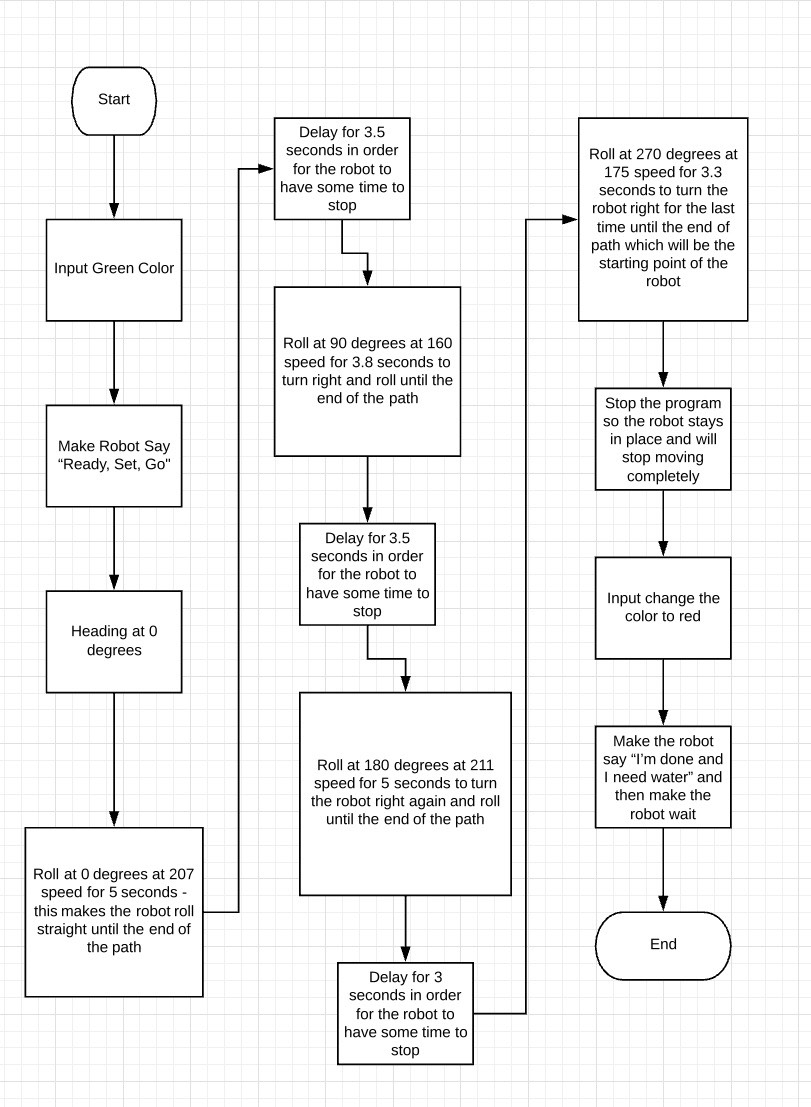
**Signed out robot**

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:**

****