Sprint 1 - Endurance Design Document

March 28th, 2024

Sprint 1 - Endurance Design Document

Executive Summary

1.1 Project Overview

This project is to showcase the new and improved functionality of the Sphero robots. This product is intended for anyone who is learning to code either as a profession or a hobby. In the following sections you will find the general makeup of the testing on the Sphero robots that have been conducted onsite, as well as external research into the making of these robots. There will also be a proposal of unique ways to use this software outside of learning moving forward in the future.

1.2 Purpose and Scope of this Specification

The purpose of this specification is to provide clear guidelines and requirements for the design and development of our team's Sphero robot project. It outlines the functionality, constraints, and considerations that need to be addressed during this project's lifecycle.

In scope

Functional Requirements: Ensure the robot completes the obstacle course.

Design Constraints: All programming is done through the Sphero Robot application.

Hardware/Software: This document explains the difference in using different software and the most compatible as well as the required hardware.

Testing: Document provides an in-depth test table with each test our team has run.

Out of Scope

Coding Application: This document does not go into depth with how the Sphero Application coding was developed.

Robot Design: This document does not go into the Sphero robots' design or how they are made.

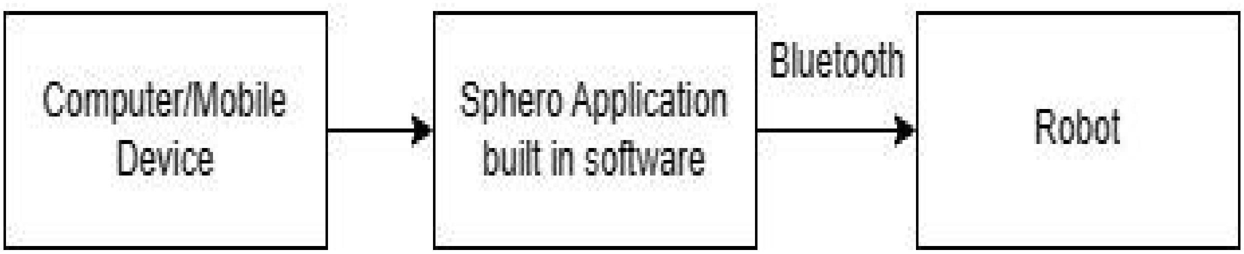
Sales Pitch: It is not our intention to sell the reader on anything, this document is simply research conducted by our team utilizing the Sphero Robots and their software.

2. Product/Service Description

To ensure proper functionality of the robots, the proper operating system is paramount. Due to the developments made by the Sphero team, the Robots work best with Mac OS but do offer operating capabilities with Windows or Linux OS. The Sphero program utilizes block coding for easy-to-use development which reduces development time exponentially.

2.1 Product Context

The Sphero robots utilize state of the art technology for easy-to-use coding with preset "blocks" that can be used to set the robots function as the user sees fit. The robots utilize similar but simplified software that the Roomba Vacuum or RC cars utilize. The Sphero robots are self-contained in that each component is designed to operate that single individual robot. Sphero does work with a variety of related systems including Computer/Mobile Devices via Bluetooth. These computer/mobile devices link directly to the robot and can control the robot as coding is done via the Sphero application and not onto the robot itself.



2.2 User Characteristics

Student:

Role: Student

Experience: Varies based upon users' knowledge.

Technical Expertise: Depends on if the student has experience/is familiar with operating a smartphone or laptop. Typically ranges from low to moderate.

Other Characteristics: Creativity and willingness to learn. Students may use the Sphero Robots to understand creating an algorithm and putting their on-paper tests into physical ones.

Faculty/Staff:

Role: Educator

Experience: Varies, but generally more experienced compared to students. Faculty may have some background in education, technology, or STEM fields.

Technical Expertise: Ranges from moderate to high. Faculty and staff may have experience with educational technology, programming, and curriculum development.

Other Characteristics: Interest in incorporating innovative teaching methods and technologies into their courses. Allows freedom for students to learn proper project management with little interference.

Other:

Role: Individuals either in STEM related fields or enjoy computing as a hobby.

Experience: Varies ranging from complete beginners to experienced individuals with a passion for technology.

Technical Expertise: Varies based upon individuals background, interest, and experience with programming or robotics.

Other Characteristics: Interest in technology and learning. Users may use the Sphero robots for recreational purposes or to learn more about programming and robotics.

2.3 Assumptions

Internet Connection- The user must ensure they are connected to the internet via Wi-Fi, hotspot, or cellular data to ensure proper Bluetooth connection to the robot.

Operating System- The Sphero Robots work best with Mac OS but are compatible with Windows, Linux, Android, and Chromebook as well with reduced functions.

Physical Environment- The robot is set to traverse a certain obstacle course and if the course were to change then the code would have to be adjusted to fit.

Age and Experience of User- The simple and easy to use application provides users with an efficient way to design the robots' functions, erasing many abstractions that can be present in coding.

2.4 Constraints

Old System Compatibility: While the Sphero application works best on Mac OS, it can be run on older systems with limited functions. For the best operation you must ensure the most up to date OS and application are downloaded.

Audit Functions: The original code is posted to Github for public viewing, but the SDD is embedded so that every change to the document found can be sourced back to who made the change.

Access and Security: Due to the block code and research being posted to Github, all this information is open source for anyone to use.

Importance of the Project: While not groundbreaking, our research is to show the simplicity of the Sphero application. As our team is new to the world of coding, this project shows how easy the Sphero robots can be to grasp the concept of coding.

Resource Limits: The Sphero Robots utilize a basic software that can be run on a variety of devices if it has the ability to download external applications and connect to wife/Bluetooth.

Design Standards: The Sphero Robots must be programmed within the Sphero application. The Sphero application utilizes block coding and presets that apply directly to Sphero robots. The built program can be used on any Sphero Robot assuming proper Wi-Fi connection to connect the robot via Bluetooth.

2.5 Dependencies

The Sphero Robots act based on the code provided to it via the Sphero application. To ensure proper functionality, the user must ensure the Sphero application is up to date. The proper program can either be downloaded via Github or the web or created from scratch based on the user’s preference and tasking.

3. Requirements

To begin this project, all requirements must be described in adequate detail to satisfy further steps of the project. The following are the requirements in detail which satisfy the project guidelines:

Robot must begin on the yellow square
Robot then changes color to green and states “Ready, set, go!”
Robot then proceeds moving in direction 0 degrees at x speed for approximately 11.8 seconds
Robot then comes to a complete stop at turn one and makes a 90 degree turn
Robot proceeds onward in direction 90 degrees at x speed for 6.4 seconds
Robot then comes to a stop at turn two and completes another 90 degree turn
Robot proceeds forward in direction 180 degrees at x speed for 11.9 seconds
Robot stops at turn three and completes another 90 degree turn
Robot proceeds forward on line at 270 degrees at x speed for 6.3 seconds
Robot makes its final stop on the yellow square in which it started at, turns red and says “I'm tired and I need water”

Sprint 1 - Endurance Design Document

Following these requirements, a designer may design a system to test and verify that they meet the project guidelines, as per what our group completed.

Requirement #	**Priority**	**Input**	**Function**	**Output**
1	**1**	Physically placing the robot on the yellow square	Physically place robot on yellow square	Robot begins on yellow square
2	**1**	Code robot to change color and speak	Robot changes to green and says “Ready, set, go!”	Robot changes color and speaks
3	**1**	Code robot to move in one direction at an approximate speed for a certain amount of time	Robot then proceeds moving in direction 0 degrees at x speed for approximately 11.8 seconds	Robot proceeds in assigned direction
4	**1**	Code robot to stop and make a turn	Robot then comes to a complete stop at turn one and makes a 90 degree turn	Robot completes spin task
5	**1**	Code robot to move in one direction at an approximate speed for a certain amount of time	Robot proceeds onward in direction 90 degrees at x speed for 6.4 seconds	Robot proceeds in assigned direction
6	**1**	Code robot to stop and make a turn	Robot then comes to a stop at turn two and completes another 90 degree turn	Robot completes spin task
7	**1**	Code robot to move in one direction at an approximate speed for a certain amount of time	Robot proceeds forward in direction 180 degrees at x speed for 11.9 seconds	Robot proceeds in assigned direction
8	**1**	Code robot to stop and make a turn	Robot stops at turn three and completes another 90 degree turn	Robot completed spin task
9	**1**	Code robot to move in one direction at an approximate speed for a certain amount of time	Robot proceeds forward on line at 270 degrees at x speed for 6.3 seconds	Robot proceeds in assigned direction
10	**1**	Code robot to make final stop and speak	Robot makes its final stop on the yellow square in which it started at, turns red and says “I'm tired and I need water”	Robot completes final stop and speaks

3.1 Functional Requirements

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
ENDUR_ 01	Robot must start in yellow square	Start	1	18MAR 24	Chris K
ENDUR_ 02	Robot must turn green and say “ready set go!”	Speak and change color	1	18MAR 24	Chris K
ENDUR_ 03	Robot must proceed on blue line and stop at first corner	Proper movement	1	18MAR 24	Chris K
ENDUR_ 04	Robot must make a right at the first corner and proceed to the second corner on the blue line	proper stop and turn	1	18MAR 24	Chris K
ENDUR_ 05	Robot must stop and execute another proper right turn at the second corner.	proper stop and turn	1	18MAR 24	Chris K
ENDUR_ 06	Robot must proceed on the blue line and stop at the third corner.	proper movement	1	18MAR 24	Chris K
ENDUR_ 07	Robot must stop and execute a right turn at the third corner	proper stop and turn	1	18MAR 24	Chris K
ENDUR_ 08	Robot must proceed on the blue line and stop back at the starting point.	proper movement	1	18MAR 24	Chris K
ENDUR_ 09	Robot must stop turn red and say “I’m done and I need water”.	stop at yellow square and speak	1	18MAR 24	Chris K

3.2 Security

3.2.1 Protection

The use of GitHub, Google Docs, and Sphero.edu incorporates protective measures like encryption, activity logging, and data integrity checks. GitHub encrypts code repositories to prevent unauthorized access, while activity logging in both GitHub and Google Docs tracks changes for detection of unauthorized modifications. These measures collectively defend against malicious or accidental threats, bolstering system security

3.2.2 Authorization and Authentication

To fortify security measures for our project, we utilized GitHub and Google Docs to enforce strict authorization and authentication protocols. GitHub served as our primary platform for version control and collaboration, allowing only authorized team members to modify our code repositories, ensuring maximum safety. Meanwhile, our System Design Document on Google Docs featured authentication features such as timestamps and user verification, providing clear accountability for any changes made. By integrating these measures, we bolstered security, safeguarding our data and ensuring the integrity of our project.

3.3 Portability

GitHub, Google Docs, and Sphero.edu are built with portability in mind. They use code that can easily work on different machines and operating systems. These platforms prioritize simple languages that are known to work well everywhere. Sphero.edu is especially good at working the same way no matter where it's used. The robots are made to be portable as they come with a carrying case, allowing for easy access to bring anywhere. This also means they can be used on different computers without any problems, making them flexible and easy to use across various setups.

4. Requirements Confirmation/Stakeholder sign-off

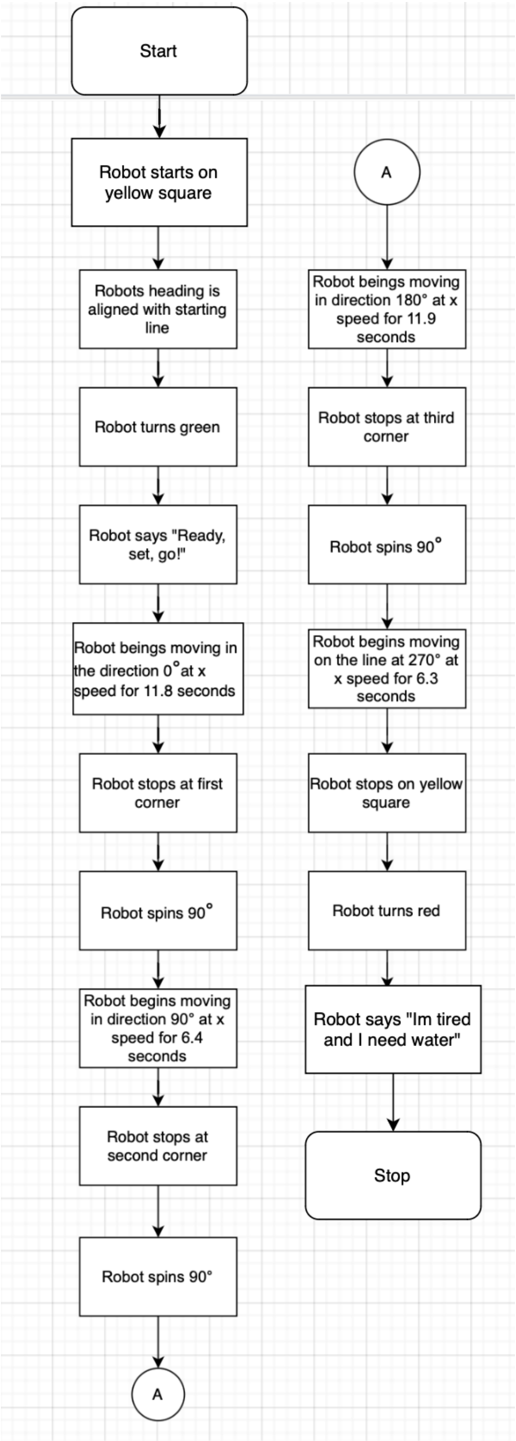
Meeting Date	Attendees (name and role)	Comments
03/18/24	Chris Kenny (Leader) and Aaron Guensch (Developer)	Confirmed all requirements
03/19/24	Chris Kenny (Leader) and Aaron Guensch (Developer)	Confirmed all tests passed
03/26/24	Chris Kenny (Leader) and Aaron Guensch (Developer)	Completed SDD

5. System Design

5.1 Algorithm

1. Robot must start on yellow square
2. Set robots heading in line with first line
3. Robot turns green
4. Robot says “Ready, set, go!”
5. Robot moves in direction 0° at x speed for 11.8 seconds
6. Robot must stop at first corner
7. Robot spins 90°
8. Robot continues in direction 90° at x speed for 6.4 seconds
9. Robot must stop at second corner
10. Robot spins another 90°
11. Robot continues in direction 180° at x speed for 11.9 seconds
12. Robot must stop at third corner
13. Robot spins another 90°
14. Robot proceeds on line at 270° at x speed for 6.3 seconds
15. Robot must stop back on yellow square
16. Robot turns red
17. Robot says “I’m tired and I need water”

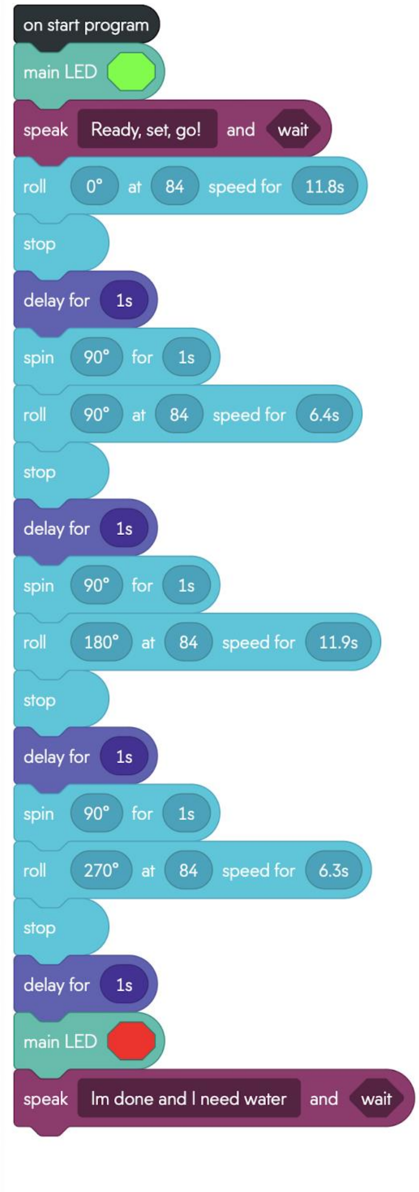
5.2 System Flow


Start
Robot starts on yellow square
Robots heading is aligned with starting line
Robot turns green
Robot says "Ready, set, go!"
Robot beings moving in the direction 0°at x speed for 11.8 seconds
Robot stops at first corner
Robot spins 90°
Robot begins moving in direction 90° at x speed for 6.4 seconds
Robot stops at second corner
Robot spins 90°
A
A
Robot beings moving in direction 180° at x speed for 11.9 seconds
Robot stops at third corner
Robot spins 90°
Robot begins moving on the line at 270° at x speed for 6.3 seconds
Robot stops on yellow square
Robot turns red
Robot says "Im tired and I need water"
Stop

5.3 Software

Mac Operating System for optimal results and feedback

Sphero Robot application for coding



```
on start program
main LED 🟩
speak Ready, set, go! and wait
roll 0° at 84 speed for 11.8s
stop
delay for 1s
spin 90° for 1s
roll 90° at 84 speed for 6.4s
stop
delay for 1s
spin 90° for 1s
roll 180° at 84 speed for 11.9s
stop
delay for 1s
spin 90° for 1s
roll 270° at 84 speed for 6.3s
stop
delay for 1s
main LED 🟥
speak Im done and I need water and wait
```

5.4 Hardware

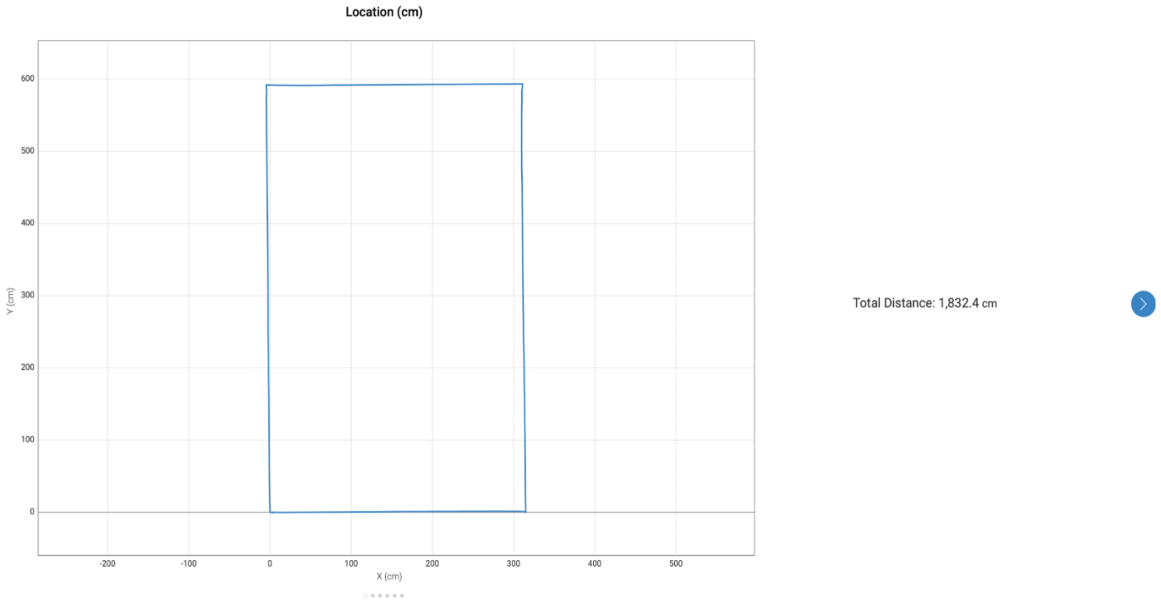
Apple Mac computer for coding and accurate results following tests

Sphero robot

5.5 Test Plan

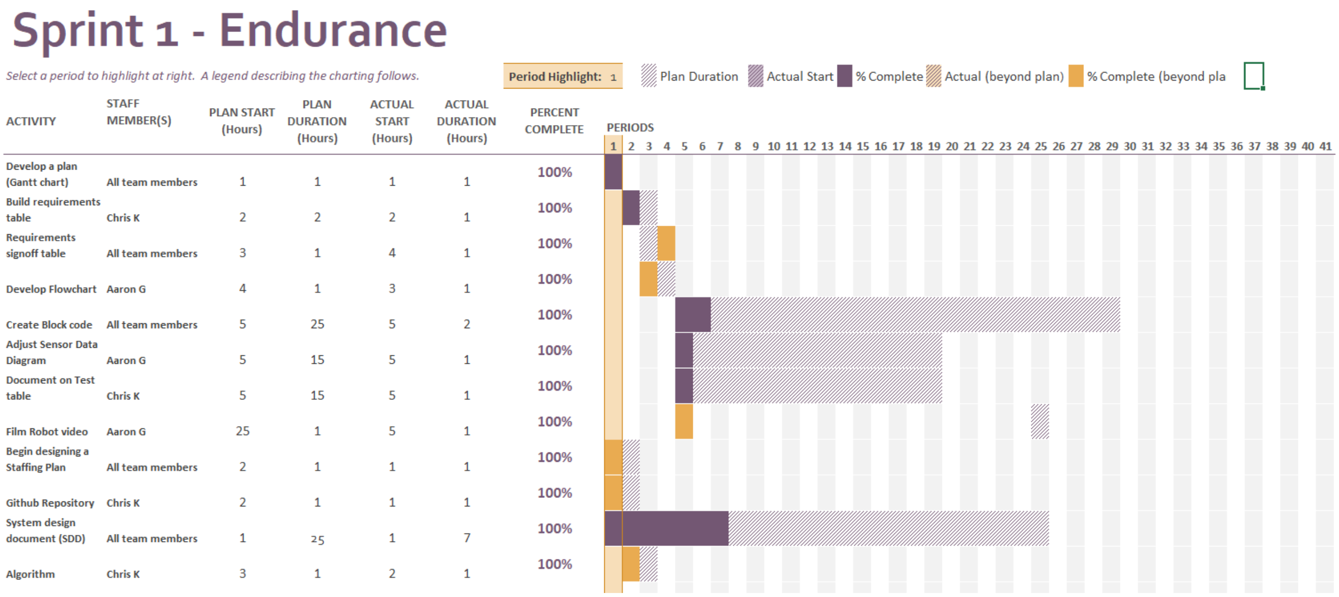
Reason for Test Case	Test Date	Expected Output	Observed Output	Staff Name	Pass/Fail
Ensure turns green	03/19/24	Turns green	Turns green	Chris	Pass
Ensure speaks	03/19/24	Says ‘ready set go”	Says “ready set go”	Aaron	Pass

Reason for Test Case	Test Date	Expected Output	Observed Output	Staff Name	Pass/Fail
Moves along line and stops	03/19/24	Stops at line	Stops short	Chris Aaron	Fail
Moves along line and stops	03/1924	Stops at line	stops at line	Chris Aaron	Pass
makes right turn	03/19/24	Turns right	turns right	Chris Aaron	Pass
Moves along second line and stop	03/19/24	moves along line and stops accurately	moves along line and stops accurately	Chris Aaron	Pass
Turns right at second stop	03/19/24	turns right at second stop	Stops and turns correctly	Chris Aaron	Pass
Moves along third line and stops correctly	03/19/24	Moves along third line and stops	moves and stops accurately	Chris Aaron	Pass
Full test	03/19/24	Robot completes full sprint and record video	first and last legs inaccurate	Chris Aaron	Fail
Full test	03/19/24	Robot completes full sprint	Robot completed full sprint	Chris Aaron	Pass
Turns red and says “Im done and I need water”	03/19/24	Turns red and accurately speaks	Turns red and says “im done and need water”	Chris Aaron	Pass
Full test and video	03/19/24	Robot completes full sprint with video	Robot completed full sprint accurately	Chris Aaron	Pass



Total Distance: 1,832.4 cm

5.6 Task List/Gantt Chart



Sprint 1 - Endurance

Select a period to highlight at right. A legend describing the charting follows.

Period Highlight: 1

ACTIVITY	STAFF MEMBER(S)	PLAN START (Hours)	PLAN DURATION (Hours)	ACTUAL START (Hours)	ACTUAL DURATION (Hours)	PERCENT COMPLETE
Develop a plan (Gantt chart)	All team members	1	1	1	1	100%
Build requirements table	Chris K	2	2	2	1	100%
Requirements signoff table	All team members	3	1	4	1	100%
Develop Flowchart	Aaron G	4	1	3	1	100%
Create Block code	All team members	5	25	5	2	100%
Adjust Sensor Data Diagram	Aaron G	5	15	5	1	100%
Document on Test table	Chris K	5	15	5	1	100%
Film Robot video	Aaron G	25	1	5	1	100%
Begin designing a Staffing Plan	All team members	2	1	1	1	100%
Github Repository	Chris K	2	1	1	1	100%
System design document (SDD)	All team members	1	25	1	7	100%
Algorithm	Chris K	3	1	2	1	100%

5.7 Staffing Plan

Name	Role	Responsibility	Reports To
Chris Kenny	Leader/Developer	Develop, Ensure smooth operations, complete work	Professor Eckert
Aaron Guensch	Co-leader/developer	Develop and complete work	Chris
Skylyn	Quit	Dealing with leaving us behind	herself