Sprint 1 - Endurance Design Document

October XX, 2019

Use this Requirements Specification template to document the requirements for your product or service, including priority and approval (Must do).

This document will also serve as a System Design Document (How to) and will include sections detailing system flow, algorithms, staffing plan, software/hardware, and Test Plan

This document contains instructions and examples which are for the benefit of the person writing the document and should be removed before the document is finalized.

To regenerate the TOC, select all (CTL-A) and press F9.

Table of Contents

1. EXECUTIVE SUMMARY 3

1.1 Project Overview 3

1.2 Purpose and Scope of this Specification 3

2. PRODUCT/SERVICE DESCRIPTION 3

2.1 Product Context 3

2.2 User Characteristics 3

2.3 Assumptions 3

2.4 Constraints 3

2.5 Dependencies 4

3. REQUIREMENTS 4

3.1 Functional Requirements 5

3.2 Security 5

3.2.1 Protection 5

3.2.2 Authorization and Authentication 6

3.3 Portability 6

4. REQUIREMENTS CONFIRMATION/STAKEHOLDER SIGN-OFF 6

5. SYSTEM DESIGN 6

5.1 Algorithm 6

5.2 System Flow 6

5.3 Software 6

5.4 Hardware 6

5.5 Test Plan 7

5.6 Task List/Gantt Chart 7

5.7 Staffing Plan 7

1. Executive Summary

1.1 Project Overview

Describe this project or product and its intended audience, or provide a link or reference to the project charter.

The purpose of this project is for the robot to be able to go through an obstacle course by following the algorithm and the audience would be the class.

1.2 Purpose and Scope of this Specification

Describe the purpose of this specification and its intended audience. Include a description of what is within the scope and what is outside of the scope of these specifications.

In scope

This document addresses requirements related to phase 2 of Project A:

- modification of Classification Processing to meet legislative mandate ABC.
- modification of Labor Relations Processing to meet legislative mandate ABC.

Out of Scope

The following items in phase 3 of Project A are out of scope:

- modification of Classification Processing to meet legislative mandate XYZ.
- modification of Labor Relations Processing to meet legislative mandate XYZ.

(Phase 3 will be considered in the development of the requirements for Phase 2, but the Phase 3 requirements will be documented separately.)

2. Product/Service Description

In this section, describe the general factors that affect the product and its requirements. This section should contain background information, not state specific requirements (provide the reasons why certain specific requirements are later specified).

The robot is a sphere and us being used to get around an obstacle course without knocking anything over, light up, and speak. In the first sprint the robot has to be able to go in the shape of a square and also light up and speak.

2.1 Product Context

How does this product relate to other products? Is it independent and self-contained? Does it interface with a variety of related systems? Describe these relationships or use a diagram to show the major components of the larger system, interconnections, and external interfaces.

This product is related to other products because it is a robot and can be managed by different types of coding. For this specific project we are using block code. This robot would be self-contained because everything needed to control it is in the app. The app is what connects it to be able to control it through bluetooth.

2.2 User Characteristics

Create general customer profiles for each type of user who will be using the product. Profiles should include:

- Student/faculty/staff/other: students can learn how to use block code, teachers can gain experience by using it to teach.
- experience: gives one experience on computing, especially block code.
- technical expertise
- other general characteristics that may influence the product

2.3 Assumptions

List any assumptions that affect the requirements, for example, equipment availability, user expertise, etc. For example, a specific operating system is assumed to be available; if the operating system is not available, the Requirements Specification would then have to change accordingly.

2.4 Constraints

Describe any items that will constrain the design options, including

- parallel operation with an old system
- audit functions (audit trail, log files, etc.)
- access, management and security
- criticality of the application
- system resource constraints (e.g., limits on disk space or other hardware limitations)
- other design constraints (e.g., design or other standards, such as programming language or framework)

2.5 Dependencies

List dependencies that affect the requirements. Examples:

- This new product will require a daily download of data from X,
- Module X needs to be completed before this module can be built.

3. Requirements

- Describe all system requirements in enough detail for designers to design a system satisfying the requirements and testers to verify that the system satisfies requirements.
- Organize these requirements in a way that works best for your project. See **Error! Reference source not found.Error! Reference source not found.**, **Error! Reference source not found.** for different ways to organize these requirements.
- Describe every input into the system, every output from the system, and every function performed by the system in response to an input or in support of an output. (Specify what functions are to be performed on what data to produce what results at what location for whom.)
- Each requirement should be numbered (or uniquely identifiable) and prioritized.
  See the sample requirements in Functional Requirements, and **Error! Reference source not found.**, as well as these example priority definitions:

  Priority Definitions

  The following definitions are intended as a guideline to prioritize requirements.

  - Priority 1 – The requirement is a “must have” as outlined by policy/law
  - Priority 2 – The requirement is needed for improved processing, and the fulfillment of the requirement will create immediate benefits
  - Priority 3 – The requirement is a “nice to have” which may include new functionality

  It may be helpful to phrase the requirement in terms of its priority, e.g., "The value of the employee status sent to DIS **must be** either A or I" or "It **would be nice** if the application warned the user that the expiration date was 3 business days away". Another approach would be to group requirements by priority category.

- A good requirement is:
  - Correct
  - Unambiguous (all statements have exactly one interpretation)
  - Complete (where TBDs are absolutely necessary, document why the information is unknown, who is responsible for resolution, and the deadline)
  - Consistent

- Ranked for importance and/or stability
- Verifiable (avoid soft descriptions like “works well”, “is user friendly”; use concrete terms and specify measurable quantities)
- Modifiable (evolve the Requirements Specification only via a formal change process, preserving a complete audit trail of changes)
- Does not specify any particular design
- Traceable (cross-reference with source documents and spawned documents).

3.1 Functional Requirements

In the example below, the requirement numbering has a scheme - BR_LR_0## (BR for Business Requirement, LR for Labor Relations). For small projects simply BR-## would suffice. Keep in mind that if no prefix is used, the traceability matrix may be difficult to create (e.g., no differentiation between '02' as a business requirement vs. a test case)

The following table is an example format for requirements. Choose whatever format works best for your project.

For Example:

Req#	**Requirement**	**Comments**	**Priority**	**Date Rvwd**	**SME Reviewed / Approved**
ENDUR_01	robot will start with green light and say ‘ready, set, go’ then move forward to first corner and stop	starts on yellow square with blue tape			
ENDUR_02	after stopping robot turns red and says ‘I’m done and I need water’ then turn right at corner	moves forward then stops			
ENDUR_03	robot will move forward to next corner, stop, and turn right	third corner			
ENDUR_04	robot will stop at the third corner turn 90 degrees and then move forward	robot should end at beginning corner			
ENDUR_05	robot stops at beginning corner	robot ends			
ENDUR_XX					

3.2 Security

3.2.1 Protection

Specify the factors that will protect the system from malicious or accidental access, modification, disclosure, destruction, or misuse. For example:

- encryption
- activity logging, historical data sets

- restrictions on intermodule communications
- data integrity checks

3.2.2 Authorization and Authentication

Specify the Authorization and Authentication factors. Consider using standard tools such as PubCookie.

3.3 Portability

If portability is a requirement, specify attributes of the system that relate to the ease of porting the system to other host machines and/or operating systems. For example,

- Percentage of components with host-dependent code;
- Percentage of code that is host dependent;
- Use of a proven portable language;
- Use of a particular compiler or language subset;
- Use of a particular operating system;
- The need for environment-independence - the product must operate the same regardless of operating systems, networks, development or production environments.

4. Requirements Confirmation/Stakeholder sign-off

Include documentation of the approval or confirmation of the requirements here. For example:

Meeting Date	**Attendees (name and role)**	**Comments**
03/22/2021	Adrianna Lanfranco	confirmed
03/22/2021	Dylan Leray	confirmed…………

5. System Design

This section will provide all details concerning the technical design, staffing, coding, and testing the system

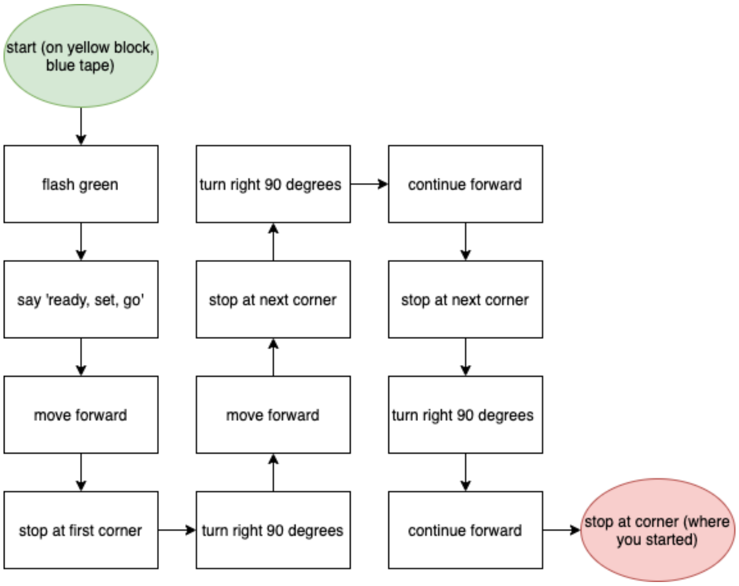
5.1 Algorithm

Develop and describe here the algorithm that will be used to provide the required performance of your software

- robot will begin at starting spot
- robot will flash green
- robot will say ‘ready, set, go’
- robot will move forward
- robot will stop at first corner
- robot will turn red
- robot will speak ‘i’m done and need water’
- robot will turn 90 degrees
- robot will move forward
- robot will stop at next corner
- robot will turn right
- robot will move forward
- robot will turn right at third corner
- robot will move forward
- robot will stop at original corner

5.2 System Flow

Develop a flowchart (and show here) that accurately depicts how your software application will act to fulfill the algorithm



5.3 Software

Describe software languages/platforms/api's used to develop and deploy this application

- Sphero
    - block code

5.4 Hardware

Describe hardware platforms that were used to develop, test and demonstrate this application

- sphero robot
- adrianna: mac

5.5 Test Plan

Include a test plan showing all unit tests performed for this application, Include test rational, test date, staff member, pass/fail status

Reason for Test Case	Test Date	Expected Output	Observed Output	Staff Name	Pass/Fail
create code	3/30/21	figure out code, record video	robot kept going at a angle, fixed it with 'aim'	adrianna lanfranco	P
test code	3/30/21	code to work	took about an hour	adrianna lanfranco	P

5.6 Task List/Gantt Chart

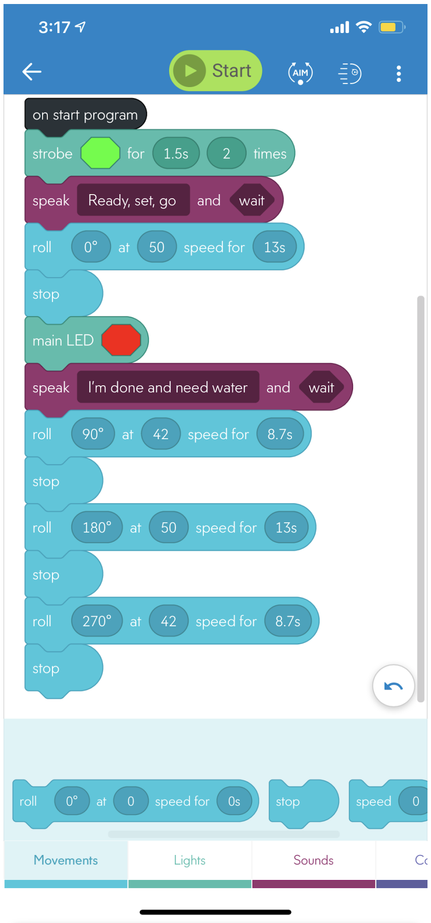
Embed your gantt chart here
https://docs.google.com/spreadsheets/d/1mQ2GAvJUJ4ceKCrHbuEUnojyfPLMvCCb/edit#gid=1479078586

5.7 Staffing Plan

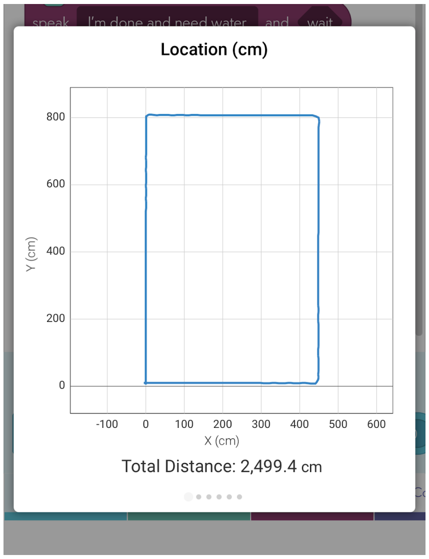
Insert a chart/table that depicts the roles and responsibilities of each team member that worked on this project

Name	Role	Responsibility	Reports To
adrianna lanfranco	member	flow chart, algorithm, code,video,	all members

Block Code:



```
on start program
strobe 🟩 for 1.5s 2 times
speak Ready, set, go and wait
roll 0° at 50 speed for 13s
stop
main LED 🟥
speak I'm done and need water and wait
roll 90° at 42 speed for 8.7s
stop
roll 180° at 50 speed for 13s
stop
roll 270° at 42 speed for 8.7s
stop
```



Total Distance: 2,499.4 cm