Sprint 3 - Speed Design Document

November, 2019

Table of Contents

[1. Executive Summary 3](#_Toc21616852)

[1.1 Project Overview 3](#_Toc21616853)

[1.2 Purpose and Scope of this Specification 3](#_Toc21616854)

[2. Product/Service Description 3](#_Toc21616855)

[2.1 Product Context 3](#_Toc21616856)

[2.2 User Characteristics 3](#_Toc21616857)

[2.3 Assumptions 3](#_Toc21616858)

[2.4 Constraints 3](#_Toc21616859)

[2.5 Dependencies 4](#_Toc21616860)

[3. Requirements 4](#_Toc21616861)

[3.1 Functional Requirements 5](#_Toc21616862)

[3.2 Security 5](#_Toc21616863)

[3.2.1 Protection 5](#_Toc21616864)

[3.2.2 Authorization and Authentication 6](#_Toc21616865)

[3.3 Portability 6](#_Toc21616866)

[4. Requirements Confirmation/Stakeholder sign-off 6](#_Toc21616867)

[5. System Design 6](#_Toc21616868)

[5.1 Algorithm 6](#_Toc21616869)

[5.2 System Flow 6](#_Toc21616870)

[5.3 Software 6](#_Toc21616871)

[5.4 Hardware 6](#_Toc21616872)

[5.5 Test Plan 7](#_Toc21616873)

[5.6 Task List/Gantt Chart 7](#_Toc21616874)

[5.7 Staffing Plan 7](#_Toc21616875)

# Executive Summary

## Project Overview

Creating a program that will allow a spherical robot to do a complete figure 8 in as little time as possible

## 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 what is outside of the scope of these specifications. For example:

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.)

# 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).

## 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.

## User Characteristics

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

* Student/faculty/staff/other
* experience
* technical expertise
* other general characteristics that may influence the product

## 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.

## 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)

## 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.

# 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).

## 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 |
| --- | --- | --- | --- | --- | --- |
| SPEED\_01 | Robot must go in a straight line to the next point+ |  | 1 | 11/02/19 |  |
| SPEED\_02 | Must make a 90 degree turn to go to the next point |  | 2 | 11/02/19 |  |
| SPEED\_03 | Must set the degree back to 0 and accelerate to get to the next point |  | 3 | 11/02/19 |  |
| SPEED\_04 | Must gather enough speed to go over the ramp but at the same time stop at the desired point |  | 4 | 11/02/19 |  |
| SPEED\_05 | Must turn to finish the course in a straight line |  | 5 | 11/02/19 |  |
| SPEED\_06 | Repeat if neccesary |  | 6 | 11/02/19 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| SPEED\_XX |  |  |  |  |  |

## Security

### 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

### Authorization and Authentication

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

## 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.

# 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 |
| 12/03/19 | Hassan, Sterling, Michael | Meet so we can go over sprint 3, collaborate, and make sure we complete the course |
| MM/DD/YY | My group member names |  |

# System Design

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

## Algorithm

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

* Place Robot in starting position
* Accelerate, slightly decelerate upon turn
* Have robot make wide U-turn (x) degrees towards the left, facing center
* Accelerate
* Move straight, past the center, (x) distance
* Slight deceleration
* Begin a wide turn towards the right (x) degrees until facing center once again
* Accelerate
* Move straight, past center, (x) distance
* Slight deceleration
* Make turn towards left until at the starting point.

## System Flow

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

## Software

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

## Hardware

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

## Test Plan

| **Reason for Test Case** | **Test Date** | **Expected Output** | **Observed Output** | **Staff Name** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| Test Ability to go in a straight line | 12/02/19 | The robot will properly go to the intended point | It took a lot of trial and error but the robot made it to the point | Michael, Sterling, Hassan | Pass |
| Test ability to rotate | 12/02/19 | The robot will turn at the correct angle to go to the next point | At 90 degrees the robot successfully turned to face the right point | Michael, Sterling, Hassan | Pass |
| Test the balance between speed and decleration | 12/02/19 | The robot will have enough speed to go over the ramp but at the same time it will have enough speed to decelerate. | Its not perfect, but to some extent it can go over the ramp and make it to the general area where it needs to devclerate. | Michael, Sterling, Hassan | Fail |
| Test the ability to make a full rotation to go in a straight line to finish | 12/02/19 | The robot will make the proper turn and go in a straight line to finish | We couldn’t really get the angle right so, it goes to the correct distance but not exactly the right angle | Michael, Sterling, Hassan | Fail |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Task List/Gantt Chart

Embed your gantt chart here

## 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 |
| --- | --- | --- | --- |
| Sterling | Programmer | Programs robot | The group |
| Hassan | Manager | Manages project | The group |
| Michael | Organizer | Makes sure all items are organized | The group |