Software requirement specification (SRS) document template



Review history



Approval history

Project name: Date: Version:

By:



Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Verson description | Date completed |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Approving party | Version approved | Signature | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Reviewer | Version reviewed | Signature | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Table of contents



1. Introduction
   1. Product scope
   2. Product value
   3. Intended audience
   4. Intended use
   5. General description

2 Functional requirements

1. External interface requirements
   1. User interface requirements
   2. Hardware interface requirements
   3. Software interface requirements
   4. Communication interface requirements
2. Non-functional requirements
   1. Security
   2. Capacity
   3. Compatibility
   4. Reliability
   5. Scalability
   6. Maintainability
   7. Usability
   8. Other non-functional requirements

5 Definitions and acronyms



1

Introduction

Describe the purpose of the document.

* 1. Product scope

List the benefits, objectives, and goals of the product.

The objective of BT-896 (Robot) is to have the ability to solve a maze whilst identifying the benefits of BT-896 is the industrial setting. Overall, the goal of Kettle chili chips is to solve a maze with the harsh environment.

* 1. Product value

Describe how the audience will find value in the product.

* 1. Intended audience

Write who the product is intended to serve.

The target market market in which we will value this product is keen on engineering, more specifically robotics engineering

* 1. Intended use

Describe how will the intended audience use this product.

The intended use for clank is to solve and identify obstacles in a maze but can be used for different problems. This may include moving blocks identifying different colors.

* 1. General description

Give a summary of the functions the software would perform and the features to be included.

The product will follow a track in a specific maze whilst clearing obstacles on its way but using the wheels for direction and the other parts which allow the product to maneuver over these obstacles such like different colored titles ect.

# Functional requirements



2

List the design requirements, graphics requirements, operating system requirements, and constraints of the product.

The robot must have all the compiled parts, each of the up to date including the battery

External interface requirements



3

* 1. User interface requirements

Describe the logic behind the interactions between

the users and the software (screen layouts, style guides, etc).

* 1. Hardware interface requirements

List the supported devices the software is intended

to run on, the network requirements, and the communication protocols to be used.

* 1. Software interface requirements

Include the connections between your product and other software components, including frontend/backend framework, libraries, etc.

* 1. Communication interface requirements

List any requirements for the communication programs your product will use, like emails or embedded forms.

# Non-functional requirements



4

* 1. Security

Include any privacy and data protection regulations that should be adhered to.

Basic cybersecurity standards

Implement secure communication

* 1. Capacity

Describe the current and future storage needs of your software.

The robot must have the capable storage of mapping data for the maze

* 1. Compatibility

List the minimum hardware requirements for your software.

Arduino?

* 1. Reliability

Calculate what the critical failure time of your product would be under normal usage.

The robot should continuously operate for at least a specified time

Ability to avoid basic obstacles

* 1. Scalability

Calculate the highest workloads under which your software will still perform as expected.

* 1. Maintainability

Describe how continuous integration should be used to deploy features and bug fixes quickly.

* 1. Usability

Describe how easy it should be for end-users to use your software.

* 1. Other

List any additional non-functional requirements.

Definitions and acronyms



5

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |