

---

# Software Requirements Specification for <RobotName>

Version 0.5

Prepared by <author(s)>  
<Group X>

<date>

# Table of Contents

[Table of Contents](#)

[Revision History](#)

1. [Introduction](#)
  - 1.1 [Purpose](#)
  - 1.2 [Document Conventions](#)
  - 1.3 [Intended Audience and Reading Suggestions](#)
  - 1.4 [Product Scope](#)
  - 1.5 [References](#)
2. [Overall Description](#)
  - 2.1 [Product Perspective](#)
  - 2.2 [Product Functions](#)
  - 2.3 [User Classes and Characteristics \(Optional\)](#)
  - 2.4 [User Documentation \(Optional\)](#)
  - 2.5 [Assumptions and Dependencies \(Optional\)](#)
3. [External Interface Requirements](#)
  - 3.1 [Robocode Interface](#)
  - 3.2 [Software Interfaces \(Optional\)](#)
  - 3.3 [Communications Interfaces \(Optional\)](#)
4. [System Features](#)
  - 4.1 [System Feature 1](#)
    - 4.1.1 [Description and Priority](#)
    - 4.1.2 [Functional Requirements](#)
  - 4.2 [System Feature 2](#)
    - 4.2.1 [Description and Priority](#)
    - 4.2.2 [Functional Requirements](#)
  - 4.3 [System Feature 3](#)
    - 4.3.1 [Description and Priority](#)
    - 4.3.2 [Functional Requirements](#)
  - 4.4 [System Feature 4 \(and so on\)](#)
5. [Quality Requirements](#)
  - 5.1 [1-vs-1 Battle Performance Requirements](#)
  - 5.2 [Melee Battle Performance Requirements](#)
  - 5.3 [Team Battle Performance Requirements \(Optional\)](#)

#### [5.4 Software Quality Attributes](#)

#### [6. Other Requirements \(Optional\)](#)

#### [Appendix A: Glossary](#)

#### [Appendix B: Analysis Models \(Optional\)](#)

#### [Appendix C: To Be Determined List](#)

# Revision History

<Note that you are might want to create intermediate versions between the releases. This is perfectly fine!>

| Name      | Date | Description   | Version |
|-----------|------|---|---------|
| Author(s) |      | Alpha release: Features described.                                      | 0.5     |
| Author(s) |      | Beta release: Detailed requirements and quality requirements specified. | 0.9     |
| Author(s) |      | Final release: A complete SRS.  | 1.0     |

## 1. Introduction

### 1.1 Purpose

<Briefly describe the robot whose software requirements are specified in this document. Clarify the purpose of the robot, i.e., to be sold on an open market to be used in team rumbles.>

### 1.2 Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance.>

### 1.3 Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, testers, and potential customers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

### 1.4 Product Scope

<Provide a short description of the purpose of the software being specified, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies as specified in your lean canvas.>

## **1.5 References**

<List any other documents or Web addresses to which this SRS refers, e.g., the lean canvas, Java code conventions, and the ETSA02 Robot Communication Protocol. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

# **2. Overall Description**

## **2.1 Product Perspective**

<Describe the context and origin of the robot being specified in this SRS. A simple diagram that shows how the robot is intended to contribute to a robot team can be helpful.>

## **2.2 Product Functions**

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS.>

## **2.3 User Classes and Characteristics (Optional)**

<If your product targets a particular market segment, describe the different types of customers that you anticipate will purchase your robot. Describe the pertinent characteristics of each type of customer, as certain requirements may pertain only to some types of customers. Distinguish the most important types of customer for this robot from those who are less important to satisfy.>

## **2.4 User Documentation (Optional)**

<If you have any, list the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software.>

## **2.5 Assumptions and Dependencies (Optional)**

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use,

issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.>

## **3. External Interface Requirements**

### **3.1 Robocode Interface**

<Describe the characteristics of the interface between the robot and the hardware components of the system. List all events that your Robot intercepts from Robocode, e.g., onScannedRobot() and onRobotDeath(), and describe the purpose of each. Refer to the Robocode application programming interface.>

### **3.2 Software Interfaces (Optional)**

<Describe the connections between the robot and other specific software components (name and version), including databases, tools, and libraries. Identify the data items or messages coming into the system and going out and describe the purpose of each. Refer to documents that describe detailed application programming interface protocols.>

### **3.3 Communications Interfaces (Optional)**

<Describe the requirements associated with any communications functions required by this robot. If you implement the ETSA02 Robot Communication Protocol, either completely or in part, list the implemented commands and describe the purpose of each.>

## **4. System Features**

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. If you have a good reason, you can reorganize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your robot.>

### **4.1 System Feature 1**

<Don't really say "System Feature 1." State the feature name in just a few words. Align this with your lean canvas.>

### 4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority.>

### 4.1.2 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## 4.2 System Feature 2

<Don't really say “System Feature 1.” State the feature name in just a few words. Align this with your lean canvas.>

### 4.2.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority.>

### 4.2.2 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## **4.3 System Feature 3**

<Don't really say "System Feature 1." State the feature name in just a few words. Align this with your lean canvas.>

### **4.3.1 Description and Priority**

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority.>

### **4.3.2 Functional Requirements**

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## **4.4 System Feature 4 (and so on)**

# **5. Quality Requirements**

## **5.1 1-vs-1 Battle Performance Requirements**

<Specify performance requirements for how your robot shall perform in 1-vs-1 battles against standard robots to help the developers understand the intent and make suitable design choices. Make such requirements as specific as possible to allow later verification work, i.e., specify quantitative targets such as "65% win rate against SpinBot.">

## **5.2 Melee Battle Performance Requirements**



<Specify performance requirements for how your robot shall perform in melee battles against standard robots to help the developers understand the intent and make suitable design choices. Make such requirements as specific as possible to allow later verification work, i.e., specify quantitative targets such as “65% win rate in melee battles with one SpinBot, one CrazyBot, and two TargetBots.”>

### **5.3 Team Battle Performance Requirements (Optional)**

<Specify performance requirements for how your robot shall perform in team battles against standard robots to help the developers understand the intent and make suitable design choices. Make such requirements as specific as possible to allow later verification work. This might be useful if you want to state requirements on how multiple copies of your robot work together.”>

### **5.4 Software Quality Attributes**

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: maintainability, testability, source code size, and adherence to Java code conventions.>

## **6. Other Requirements (Optional)**

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

## **Appendix A: Glossary**

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

## **Appendix B: Analysis Models (Optional)**

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

## **Appendix C: To Be Determined List (Optional)**

<In the document, you might postpone some parts until later, i.e., you explicitly state “TBD”. Here you can collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>