1 Requirements

The requirements are split into functional and non-functional requirements, where the former are definitions of what a system is supposed to do and the latter are requirments describe how the system is supposed to be. The functional requirements The functional requirements are straight forward and listed in section 1.

The criteria for the non-functional requirements are shown in section 1. In contrast to many consumer facing technologies usability is not part of the requirements, as the enduser should have as little interaction with the system as possible and physical maintenance is only carried out by professionals[1].

[1] They would still be nice, but due to time limitations we excluded them.

Functional Requirements

ID	Name	Class	Description
100	Register goals	Must	The system must be able to automatically register goals
101	Transmit goals	Must	The system must be able to automatically transmit goal information to the edge device.
102	OTA updates	Should	The microcontroller should support over the air (OTA) updates
103	Low energy transmission	Must	The microcontroller and its transmission partner must be able to communicate via a low energy transmission protocol
104	Power modes	Must	The microcontroller must be able to support different power mode in which it can operate to save power during unused periods.
105	Wake up from sensor	Must	The microcontroller must be able to wake up from an external sensor input.
106	Wake up from timeout	Must	The microcontroller must be able to wake up after a predefined timeout.
107	Transmission range	Must	The transmission range must be larger than 5 meters without blockage.
108	Pin reads	Must	The microcontroller must support reading from input pins.
109	Centralized configuration	Should	All configurations should be synchronized across the nodes in the system.
110	Node fault tolerance	Should	The state of each node should be synchronized in the system, so that in case of an error, the last operable state can be recovered
111	Adjustable Goals setup	Should	The goals setup need to able to adjust in their width to cover a variety of different goal sizes

Functional Requirements

ID	Name	Class	Description
112	Containers (excl microc.)	Should	All applications, excluding the microcontroller, should run inside containers.
113	Containers	Would	The microcontroller code runs inside containers as well.
114	Single binaries	Could	All applications should be packaged in one binary.
115	Edge storage	Would	The edge part should be able to count goals and syncronize with user devices without the Internet.

Non-Functional Requirements

Area	ID	Name	Description
Reliability	100	Downtime	The downtime due needs to be less once per month.
	101	Recording failures	Less than 5 percent incorrect readings.
Robustness	200	Interference	The system should be able to deal with at least 10 bluetooth enabled devices nearby.
	201	Transmission failures	There should not be more 0.01 percent of transmission failures, including all reasons.
	203	Incorrect data	There should not be more 0.0001 percent of wrong data transmission.
	202	Crashes	In case of a crash, the software must be able to recover automatically.
Portability	300	Supported platforms	The application needs to supported on a wide variety of IoT devices for future changes.
Maintainability	400	Updates	The software needs to be updatable over the air (OTA).
	401	Centralization	As long as a table is connected to the Internet (also indirectly through the raspberry pi), all updates must be available through a central point.
	402	Bug fixes	All bugs need to be addressed latest 6 months after discovery
Efficiency	500	Battery life	The system needs to able to life on a single battery charge for at least two weeks with no more than 20 games per day.
	501	Battery replacement	The battery needs to be able to charge to above 80 percent of its original value after 2 years, with 25 recharges a year.