Velomobile Control & Telemetry System

Use Case Specification

Parse Telemetry Stream

Version 1.1

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Ver.** | **Description** | **Author** |
| January 15, 2009 | 1.0 | Initial Composition |  |
| February 16, 2010 | 1.1 | Major revision | Shawn McGinnis |

Table of Contents

[1. Parse Telemetry Stream 4](#_Toc254127386)

[1.1 Brief Description 4](#_Toc254127387)

[1.2 Requirements Trace 4](#_Toc254127388)

[1.3 Involved Actors 4](#_Toc254127389)

[2. Flow of Events 4](#_Toc254127390)

[2.1 Basic Flow 4](#_Toc254127391)

[3. Preconditions 4](#_Toc254127392)

[4. Post Conditions 4](#_Toc254127393)

[5. Scenarios 4](#_Toc254127394)

[5.1 Happy Day 4](#_Toc254127395)

[5.2 Rainy Day 5](#_Toc254127396)

# Parse Telemetry Stream

## Brief Description

Web server receives an XML telemetry stream from a device that must be parsed into the data elements.

## Requirements Trace

2.1.1, 2.1.2, 2.1.4

## Involved Actors

Web System – Web server parsing system.

# Flow of Events

## Basic Flow

This use case begins when the Web System receives a stream of metrics.

1. Web server receives XML stream and stores it.
2. Web server begins parsing stream.
3. Web server verifies that tags are formatted correctly.
4. Web server ignores incorrect tags it cannot identify.
5. Web server breaks tag information into either sensor data or requests.
6. Web server checks sensor data against previous sensor data or boundaries.
7. Web server increments error counter for incorrect/outrageous sensor data.
8. Web server inserts the correct sensor data into database.(Use Case Store Telemetry Data)
9. Web server replies with success on receiving the XML data.
10. Web server replies to requests when it can if any where specified.

# Preconditions

The system has an incoming uncorrupted telemetry stream.

# Post Conditions

The system has stored the data from the XML stream.

# Scenarios

## Happy Day

**Assumptions**: The XML stream is “<velo><heading>120</heading></velo>”.

**Steps:**

1. The web server receives data from its open socket.
2. The web server parses “velo” as the device.
3. The web server begins a response stream “<velo>”.
4. The web server parses “heading” as an element of “velo”.
5. The web server saves “120” as the data of the element “heading” of the device “velo”.
6. The web server finalizes the xml response to “<velo></velo>”.
7. The web server responds to the socket with “<velo></velo>”.

## Rainy Day

**Assumptions:** The XML stream is “<velo></heading>”.

**Steps:**

1. The web server receives data from its open socket.
2. The web server parses “velo” as the device.
3. The web server begins a response stream “<velo>”.
4. The web server parses “heading” as an element of “velo”.
5. The web server identifies that the XML stream is not formatted properly.
6. The web server responds to the socket with “<Exception type="XmlException">The 'velo' start tag on line 1 does not match the end tag of 'heading'. Line 1, position 9.</Exception>”.