Software Requirements Specification

For

Chariot

Submitted by

Chariot Dev

|  |  |
| --- | --- |
| **Instructor:** | Dr. Gregory Hislop |
| **Team Members:** | Medard Azandegbe, Kamalludin Colaire, Ryan Hassing, John Ho, Juan Garcia Lopez, Christopher Mak, Enioluwa Segun |
| **Cycle:** | 2 |
| **Date Submitted:** | 1/26/2020 |

Grading Rubric - Requirements Specification

This rubric outlines the grading criteria for this document. Note that the criteria represent a plan for grading. Change is possible, especially given the dynamic nature of this course. Any change will be applied consistently for the entire class.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Achievement** | **Minimal** | **Exemplary** | **Pts** | **Score** |
| **Content** | Section(s) missing, not useful, inconsistent, or wrong. | Provides all relevant information correctly and with appropriate detail |  |  |
| Introduction |  |  | 10 |  |
| Overall Description |  |  | 10 |  |
| Functions and Constraints |  |  | 40 |  |
| Logical Database |  |  | 20 |  |
| **Grammar and Spelling** | Many serious mistakes in grammar or spelling | Grammar, punctuation, and spelling all correct | 10 |  |
| **Expression** | Hard to follow or poor word choices | Clear and concise. A pleasure to read | 5 |  |
| **Tone** | Tone not appropriate for technical writing | Tone is consistently professional |  |  |
| **Organization** | Information difficult to locate | All information is easy to find, and important points stand out | 5 |  |
| **Layout** | Layout is inconsistent, visually distracting, or hinders use | Layout is attractive, consistent, and helps guide the reader |  |  |
| **Late Submission** |  |  |  |  |
| **Total** |  |  | 100 |  |

# Contents

[Contents 2](#_Toc30628057)

[1 Introduction 4](#_Toc30628058)

[1.1 Purpose 4](#_Toc30628059)

[1.2 Definitions 4](#_Toc30628060)

[2 Overall Description 4](#_Toc30628061)

[2.1 Product Functions 5](#_Toc30628062)

[2.2 User Characteristics 5](#_Toc30628063)

[3 Specific Requirements 6](#_Toc30628066)

[3.1 Functional Requirements 6](#_Toc30628067)

[FR1 – Installation 6](#_Toc30628068)

[FR2 – Running Chariot 6](#_Toc30628069)

[FR3 – Authentication 7](#_Toc30628070)

[Network Management 7](#_Toc30628071)

[FR4 – Networks 7](#_Toc30628072)

[FR5 – Manage Networks 7](#_Toc30628073)

[FR6 – Network Initialization 7](#_Toc30628074)

[FR7 – Network Deletion 8](#_Toc30628075)

[FR8 – Manage Network Permissions 8](#_Toc30628076)

[IoT Device Management 8](#_Toc30628077)

[FR9 – IoT Device Profile 8](#_Toc30628078)

[FR10 – IoT Device Configuration 8](#_Toc30628079)

[FR11 – Adding IoT Devices 9](#_Toc30628080)

[FR12 – IoT Device Addition During a Data Collection Episode 9](#_Toc30628081)

[FR13 – IoT Device Removal 10](#_Toc30628082)

[FR14 –Unintentional IoT Device Removal During a Data Collection Episode 10](#_Toc30628083)

[FR15 – Intentional IoT Device Removal During a Data Collection Episode 10](#_Toc30628084)

[FR16 – IoT Device Status 10](#_Toc30628085)

[Data Collection 11](#_Toc30628086)

[FR17 – Data Collection Episode (DCE) 11](#_Toc30628087)

[FR18 – IoT Device Data Collection 12](#_Toc30628088)

[FR19 – Data Collection Episode Configuration 13](#_Toc30628089)

[FR20 – Concurrent Network Data Collection 13](#_Toc30628090)

[Data Access 14](#_Toc30628091)

[FR21 – Data Storage 14](#_Toc30628092)

[FR23 – Timestamped Received Data 14](#_Toc30628094)

[FR24 – Timestamped Collected Data 15](#_Toc30628095)

[FR25 – Load Collected Data from a File 15](#_Toc30628096)

[Data Output 16](#_Toc30628097)

[FR26 – Data Output 16](#_Toc30628098)

[FR27 – Data Output during a Data Collection Episode 16](#_Toc30628099)

[3.2 Nonfunctional Requirements 17](#_Toc30628100)

[NFR1 – Extensibility 17](#_Toc30628101)

[NFR2 – Deployment 17](#_Toc30628102)

[NFR3 – Data Integrity 17](#_Toc30628103)

[NFR4 – Real-Time Data Collection 17](#_Toc30628104)

[NFR5 – Error Handling 18](#_Toc30628105)

[NFR6 – Security 18](#_Toc30628106)

[NFR7 – Documentation 18](#_Toc30628107)

[3.3 Data Requirements 19](#_Toc30628108)

[3.4 Design Constraints 20](#_Toc30628109)

[DC1 – Graphic User Interface 20](#_Toc30628110)

[DC2 – External Library Usage 20](#_Toc30628111)

[DC3 – LGPL Licensing 20](#_Toc30628112)

[DC4 – Networking Standard Compliance 20](#_Toc30628113)

[DC5 – Adaptation of Drexel Wireless’ IoT Sensor Framework 20](#_Toc30628114)

[4 References 21](#_Toc30628115)

# Introduction

## Purpose

This document shall provide a description and the requirement specifications for Chariot, an IoT sensor framework for the development and operation of heterogeneous sensor networks. Chariot is built upon the foundations provided by Dr. Mongan’s IoT Sensor Framework, but Chariot will expand and adapt from it and other IoT Sensor Frameworks. This document shall outline the requirements that were specified by stakeholders and the product development team for the development and review of the final product.

## Definitions

This section describes technical terms commonly used in the Software Requirements Specification.

1. **Internet of Things (IoT)** - the interconnection via the Internet of computing devices embedded in everyday objects, enabling them to send and receive data.
2. **Core Component** – components of Chariot that provide core functionality for Chariot to perform its tasks.
3. **IoT device** – a hardware device that records data about its environment and has the capability to connect to a network and communicate with other devices.
4. **Data Collection Episode** **(DCE)** – the timeframe during which Chariot is actively collecting and storing data received from IoT devices.
5. **Received Data** – data transmitted, or in transmission from an IoT device that has not yet been stored.
6. **Collected Data** – data received from an IoT device that has been saved to a storage unit
7. **Network** - a collection of configured, connected IoT devices
8. **Data Output Receiver** – an external program or service that connects to Chariot and receives outputted data.

# Overall Description

The Internet of Things (IoT) is a system of connected devices capable of collecting and transferring data. Different users of IoT devices may be interested in different types of data. A gardener might want to track the pH value of their soil and the air humidity. A city planner may look for a way to track pedestrian traffic in common spaces. A factory manager might look for a way to track the status of machinery and production rates.

IoT infrastructure has exploded in the past decade, so there are devices on the market to collect data for a wide variety of applications. But the wide variety of devices requires special considerations. A Wi-Fi 4K camera connects to a network much differently than a home thermometer. A motion detector may collect and transfer data in a different form than a pressure sensor.

So, despite progress in the IoT industry, it is still difficult to integrate different types of devices into a unified system. Chariot solves this problem. Chariot is a platform for IoT device management, data storage, and analysis. Users can add, remove, and configure a variety of devices on a single network. With Chariot, a gardener could purchase a pH and humidity IoT devices off the shelf and gather data from his soil and the air from one convenient interface.

## Product Functions

Chariot will provide the following functions:

* Easy addition and removal of IoT devices on a network
* Configuration of devices
* Simultaneous usage of heterogeneous devices on a single network
* Real-time data collection
* Device status reporting
* Flexible data export, including a web API
* Data analysis tools for data visualization and processing
* Modular expansion

## User Characteristics

This section identifies and describes the user types Chariot is designed for.

### Technical – A user with technical familiarity of the system and its underlying code. They will have the knowledge to add or improve data models and IoT drivers used by Chariot.

### Non-technical – This user does not need to interact with the system on a deep technical level. They use existing modules, rather than creating their own, with a focus on the final collected data.

# Specific Requirements

The below table describes the priorities of each requirement.

|  |  |
| --- | --- |
| **Priority** | **Description** |
| **Essential** | A requirement that is necessary for the final product to meet the stakeholder's expectations. |
| **High** | A requirement that is not required but will significantly improve the quality of the final product. Will be added after Essential requirements, time permitting. |
| **Low** | A requirement that offers minor improvements to the final product. Will be added after High requirements, time permitting. |

## Functional Requirements

### FR1 – Installation

**Priority: Essential**

**Target User: All**

Chariot shall have a single program install all components of the system to a computer.

### FR2 – Running Chariot

**Priority: Essential**

**Target User: All**

Upon system startup, Chariot shall prompt for the user to login, **See FR3**, and then display a welcome screen.

### FR3 – Authentication

**Priority: High**

**Target User: All**

Chariot shall have an authentication system for the management of user access.

### Network Management

The following section describes the core component of Chariot’s IoT Network Management, and its administration of collection of IoT devices.

### FR4 – Networks

**Priority: Essential**

**Target User: All**

Chariot shall group IoT devices into user-defined heterogeneous collections called networks. Each network will have configuration information for its devices and data collection.

### FR5 – Manage Networks

**Priority: Essential**

**Target User: All**

Users shall be able to manage initialized networks, allowing them to view devices in a network and make changes to devices.

### FR6 – Network Initialization

**Priority: Essential**

**Target User: All**

A user shall be able to create a new network of IoT devices, adding it to the list of initialized networks. When adding devices to the network, a user may either manually configure or load a pre-existing configuration file.

### FR7 – Network Deletion

**Priority: Essential**

**Target User: All**

A user shall be able to delete a network from the list of initialized networks, removing its device configurations and data collection parameters from the system.

### FR8 – Manage Network Permissions

**Priority: Low**

**Target User: Technical**

A technical user shall be able to control other users’ access to their initialized networks. They can allow or deny other users from editing or using their networks.

### IoT Device Management

The following section describes the core component of IoT Device Management, and how Chariot will interact with external IoT Devices.

### FR9 – IoT Device Profile

**Priority: Essential**

**Target User: All**

Chariot shall utilize device profiles to discern different IoT device models. Technical users will provide profiles to the system to allow for the interaction with new IoT devices.

### FR10 – IoT Device Configuration

**Priority: Essential**

**Target User: All**

Users shall be able to create configurations for IoT devices from within Chariot. These configurations will determine how the IoT device will perform, and how Chariot will interact with it.

#### FR10.1 – Save IoT Device Configuration

**Priority: Essential**

**Target User: All**

A user can save the configuration of an IoT device to an external file.

#### FR10.2 – Load IoT Device Configuration

**Priority: Essential**

**Target User: All**

Users shall be able to load in previously saved IoT device configurations in place of the manual input of configuration parameters.

### FR11 – Adding IoT Devices

**Priority: Essential**

**Target User: All**

A user shall be able to add IoT devices to new or existing networks. The user will either provide the configuration details for a new IoT device profile or use an existing device profile.

### FR12 – IoT Device Addition During a Data Collection Episode

**Priority: High**

**Target User: All**

A user shall be able to add IoT devices to a network that is currently in a Data Collection Episode (DCE), **See FR17**.

### FR13 – IoT Device Removal

**Priority: Essential**

**Target User: All**

A user shall be able to remove a connected IoT device from an initialized network. Removal will prevent the further reception and collection of data from that IoT device onto the network.

### FR14 –Unintentional IoT Device Removal During a Data Collection Episode

**Priority: High**

**Target User: All**

If an IoT device is unintentionally disconnected from a network during its DCE, Chariot will attempt to reconnect, before setting its status to disconnected, preventing the receiving of data from that device until manually reconnected by a user.

### FR15 – Intentional IoT Device Removal During a Data Collection Episode

**Priority: Low**

**Target User: All**

A user can remove an IoT Device from a network during a DCE. Chariot will continue to store any previously received data and remove the IoT device from the network.

### FR16 – IoT Device Status

**Priority: High**

**Target User: All**

Chariot shall record and make accessible the status of all IoT devices on a network every second. A device status will include the device ID, uptime, resource utilization, and other information pertinent to the model of device.

### Data Collection

The following section describes the core component of Data Collection and the Data Collection Episode.

### FR17 – Data Collection Episode (DCE)

**Priority: Essential**

**Target User: All**

Chariot shall take received data from a network’s IoT devices and save it as collected data into a database during a DCE.

#### FR17.1 – Initialize Data Collection Episode

**Priority: Essential**

**Target User: All**

Users will initialize a DCE to receive data from a network’s connected devices and save it to a data storage unit.

#### FR17.2 – Pause Data Collection Episode

**Priority: High**

**Target User: All**

A user shall be able to pause an ongoing DCE, suspending the receiving of data from a network’s IoT devices.

#### FR17.3 – Resume Data Collection Episode

**Priority: High**

**Target User: All**

A user shall be able to resume a paused DCE. This will allow for the further collection of data the network’s IoT devices after it is resumed.

#### FR17.4 – Terminate Data Collection Episode

**Priority: Essential**

**Target User: All**

A user shall be able to terminate a data collection episode. This will immediately cause Chariot to stop receiving data from IoT devices and save all received data as collected data.

### FR18 – IoT Device Data Collection

**Priority: Essential**

**Target User: All**

Chariot shall receive and collect data from at least one IoT device from a single network during a DCE.

#### FR18.1 – Concurrent IoT Device Data Collection

**Priority: Essential**

**Target User: All**

Chariot shall be able to receive and collect data from at least two IoT devices from a single network simultaneously during a data collection episode.

#### FR18.2 – Concurrent Heterogenous IoT Device Data Collection

**Priority: Essential**

**Target User: All**

Chariot shall be able to receive and collect data from at least two heterogeneous IoT devices from a single network simultaneously during a data collection episode.

### FR19 – Data Collection Episode Configuration

**Priority: Essential**

**Target User: All**

A user must create a DCE configuration to define what data is to be collected from the network’s IoT devices, the storage unit to use, formatting of the collected data, and duration of the data collection episode.

#### FR19.1 – Save Data Collection Episode Configuration

**Priority: High**

**Target User: All**

A user shall be able to save a DCE configuration to a file.

#### FR19.2 – Load Configuration for Data Collection

**Priority: High**

**Target User: All**

A user shall be able to load previously saved data collection episode configurations in place of the manual input of configurations.

### FR20 – Concurrent Network Data Collection

**Priority: Low**

**Target User: All**

Chariot shall be able to receive and collect data from at least two separate networks simultaneously.

#### FR20.1 – Concurrent Network Data Collection of Network Sharing IoT Devices

**Priority: Low**

**Target User: All**

Chariot shall allow for at least two separate networks that share at least one IoT device amongst their collections to perform concurrent data collection episodes if all shared devices have the same IoT device configuration, **See FR12**.

### Data Access

The following requirements indicate how Chariot manages user access to data.

### FR21 – Data Storage

**Priority: Essential**

**Target User: All**

Chariot shall have interfaces to external databases for the storage and accessing of collected data.

### FR23 – Timestamped Received Data

**Priority: Essential**

**Target User: All**

Chariot shall timestamp all received data from an IoT device to the storage unit with the time it was received.

### FR24 – Timestamped Collected Data

**Priority: Essential**

**Target User: All**

Chariot shall timestamp all collected data to the storage unit with the time it was collected.

### FR25 – Load Collected Data from a File

**Priority: High**

**Target User: All**

A user can load in an existing collected data file for use within Chariot.

#### FR25.1 – Append Data

**Priority: High**

**Target User: All**

A user shall be able to append new collected data to an existing storage unit with collected data.

#### FR25.1 – Edit Data

**Priority: Low**

**Target User: All**

A user shall be able to edit collected data from a file, this will include changing the formatting, stored values, or deleting collected data.

### Data Output

The following requirements specify requirements for Chariot’s data output capabilities, and how data will be accessible by other external programs and systems. Examples of a data output tool include visualizers, data processing tools and algorithms.

### FR26 – Data Output

**Priority: Essential**

**Target User: All**

Chariot shall maintain an output source that internal and external Data Output Receivers shall be able to receive data from.

### FR27 – Data Output during a Data Collection Episode

**Priority: Essential**

**Target User: All**

A user can connect and disconnect output receivers to Chariot, and this will not affect the DCE or the connection of other outputs.

**FR28 – Visualizer**

**Priority: Essential**

**Target User: All**

Chariot shall have a visual graphing tool that will display collected data-labelled line graphs during a DCE. There will be a legend for which source device is used for each line.

#### FR28.1 – Visualizer Replay

**Priority: High**

**Target User: All**

A user shall be able to simulate a live visualizer graphing by using existing collected data.

## Nonfunctional Requirements

The statements below define the non-functional requirements for the system.

### NFR1 – Extensibility

**Priority: Essential**

**Target User: Technical**

A technical user shall be able to increase Chariot’s functionality by creating and adding IoT devices, databases, and data output receivers to a Chariot system. There shall be documentation and guidelines for what is required for each type to interact with Chariot.

### NFR2 – Deployment

**Priority: Essential**

**Target User: Non-Technical**

Chariot’s deployment process, and usage documentation, shall be streamlined to the point where a non-technical user described in **Section 2.2.2** can install and begin data collection without external assistance.

### NFR3 – Data Integrity

**Priority: Essential**

**Target User: All**

Chariot shall maintain integrity of collected data in such a way that Chariot does not alter values during the collection process.

### NFR4 – Real-Time Data Collection

**Priority: High**

**Target User: All**

For every second of data sent to Chariot, Chariot shall receive and save the data sent in that one second interval to allow for real time analysis.

### NFR5 – Error Handling

**Priority: High**

**Target User: All**

Chariot must not crash due to the unexpected failure of an IoT Device or network. The system shall have error handling measures to ensure the continued, uninterrupted collection of data.

### NFR6 – Security

**Priority: High**

**Target User: Technician**

A technical user shall be able to encrypt the collected data and data streams used and generated by Chariot too their own encryption standards.

### NFR7 – Documentation

**Priority: High**

**Target User: All**

Chariot shall be packaged with extensive internal and external documentation. Internal documentation will clearly describe how Chariot’s code functions. External documentation will describe how to operate the system.

## Data Requirements

The following table describes the various kinds of data expected to be utilized by Chariot

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Size** | **Comment** |
| User information | Dictionary | Variable | Username/password information for each user of the system, **See FR3** |
| Access list | List | Variable | For each network, there is a list of users with access to network, **See FR8** |
| Networks | List | Variable | List of initialized networks. **See FR4** |
| Devices on network | List | Variable | List of devices connected to a network. **See FR5** |
| Device information | Dictionary | Variable | ID, name, and other information specific to a device. |
| Device Configuration | JSON | Variable | Configurable settings of a device and their values. **See FR9** |
| Device status | JSON | Variable | Uptime, resource utilization, and other status information pertinent to device. Different from Device information, as Device status will frequently change. **See FR14** |
| Device data stream | Stream | Variable | Data received from device sensors. |
| Network Configuration | JSON |  | Configurable settings for a network. |
| Data collection episode status | Integer |  | Indicates if data collection episode is in progress, paused, or terminated. Also indicates errors in data collection. |
| Collected data | Various | Variable | Incoming data from data analysis episode is written to a database. Previously collected data can also be loaded from a file. The type is dependent on the storage unit used by the data collection episode. |
| Output Source | Stream | Variable | An output stream that can be connected to by an external service or program to receive data from Chariot. |

## Design Constraints

The following section specifies the design constraints restricting design of the system.

### DC1 – Graphic User Interface

The user shall be able to use the system via a graphic user interface. Stakeholders have expressed frustration using the command line for tasks this system would accomplish, so a graphic user interface is preferred.

### DC2 – External Library Usage

Chariot will make use of external libraries to interface with IoT devices. The system will use these libraries as specified.

### DC3 – LGPL Licensing

Chariot shall be licensed with the GNU Lesser General Public License (LGPL) (Free Software Foundation, Inc., 2007). Due to the potential use of proprietary IoT Devices, Databases, and Data Output Receivers, we cannot guarantee that all components of Chariot are modifiable. But the core components of the system will be made modifiable by end users.

### DC4 – Networking Standard Compliance

Chariot shall comply will need to be able to communicate with IoT devices using various networking standards such as, IEEE 802.3, 802.11 and continue to develop for newer networking technologies such as LoRa. (Institute of Electrical and Electronics Engineers, 2018).

### DC5 – Adaptation of Drexel Wireless’ IoT Sensor Framework

Chariot will use and adapt related work in the field of IoT data collection. The system will use several key features of Dr. Bill Mongan’s IoT Sensor Framework (Mongan, 2019). Chariot will adhere to the conventions of original work and will document where work is used or adapted.

# References

Free Software Foundation, Inc. (2007, June 29). *GNU Lesser General Public License*. Retrieved from GNU Operating System: https://www.gnu.org/licenses/lgpl-3.0.en.html

Institute of Electrical and Electronics Engineers. (2018, November 14). *IEEE Standards Activities in the Internet of Things.* Retrieved from IEEE Standards Association Web site: https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/other/iot.pdf

Mongan, W. (2019). *IoT Sensor Framework*. Retrieved from William Mongan portfolio: http://www.billmongan.com/portfolio/iotframework/