Wimoto iOS App

Software Requirements Specification

Version 0.1 - 2nd December 2013

Marc Nicholas

Revision History

Date	Description	Author	Comments
2-Dec-2013	Version 0.1	Marc Nicholas	First Revision

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	Marc Nicholas	CTO	

Table of Contents

1. Introduction

This document sets out to describe the requirements for Wimoto's iOS application. In the future, this document can also be applied to the requirements for the Android version of the application.

1.1 Purpose

The purpose of this document is to articulate to Wimoto's development partner, Atmosphere Studios, the intent and functionality of the Wimoto iOS app.

1.2 Scope

Create an iPhone app for iOS 6 and 7 that will interface with Wimoto sensors.

1.3 Definitions, Acronyms, and Abbreviations

BLE — Bluetooth Low Energy

1.4 References

TBD

2. General Description

The Wimoto iOS application is responsible for interfacing with Wimoto sensors and provides user interface and a framework for setting sensor characteristics such as alarms.

Because the Wimoto sensors are Bluetooth Low Energy compliant devices, the communication method between the application and the devices will be CoreBluetooth.

Wimoto has a family of five (5) different sensors all built upon the same embedded platform and all with Bluetooth Low Energy communications:

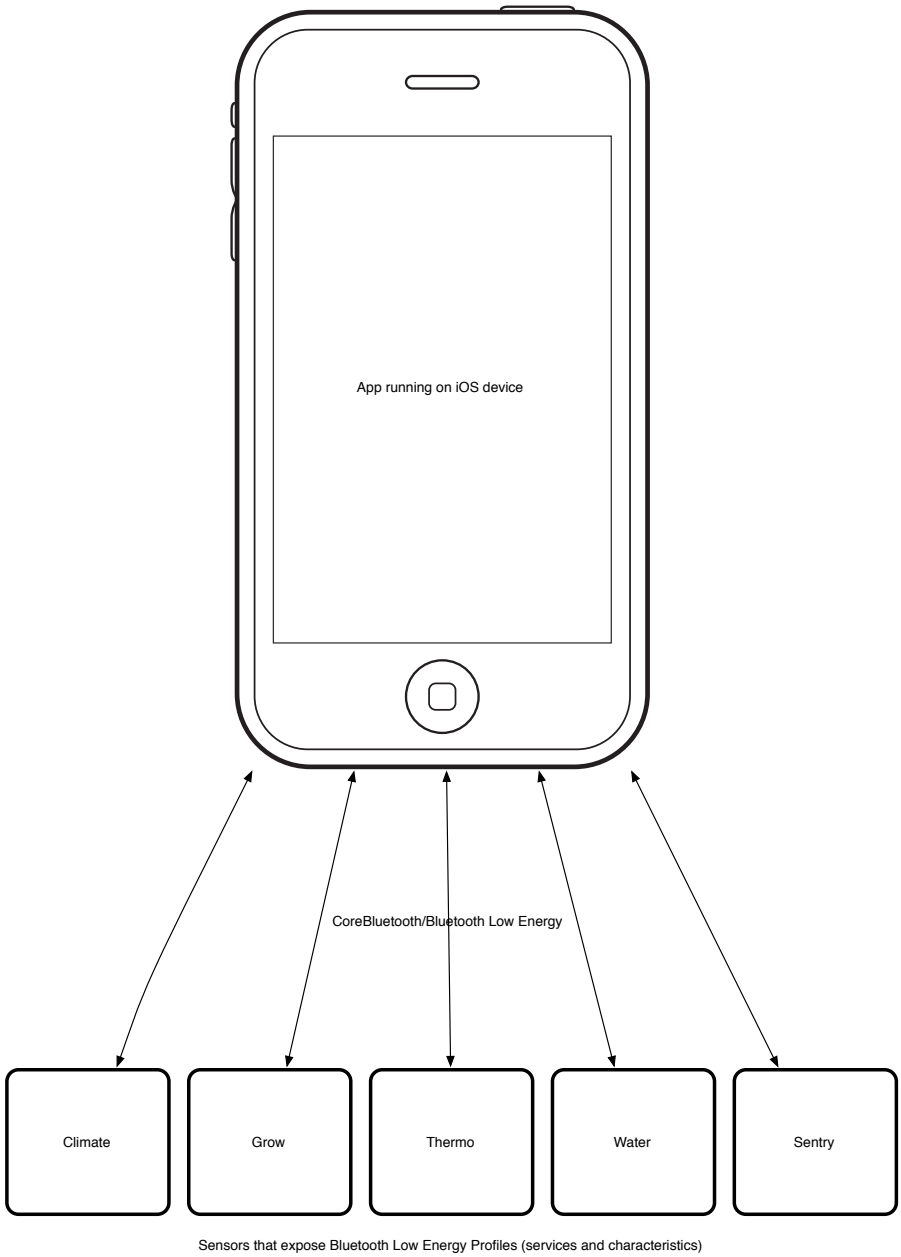
Climate
A climate sensor that provides ambient temperature, humidity and light level information.

Grow
A plant growing condition sensor that measures sunlight, soil moisture and soil temperature.

Thermo
A smart thermometer that measures object temperature with both a non-contact infrared sensor and a probe sensor.

Water
A sensor that detects the presence of water.

Sentry
A sensor that detects human presence (passive infrared), or if it's being moved (via accelerometer).



2.1 Product Perspective

This subsection of the SRS puts the product into perspective with other related products or projects. (See the IEEE Guide to SRS for more details).

2.2 Product Functions

Version 1.0

- Provide a mechanism to store details of a discovered sensor (CoreData?)
- Provide two mechanisms to add new sensors:
  - Scan for new sensors via Bluetooth Low Energy
  - Provide a mechanism to add a new sensor via QR bar code
- Connect to (in Peripheral mode) and display characteristics of the sensor (i..e temperature, humidity, presence of water, etc.
- Enable and disable the alarm characteristic
- Create a notification event so that when the alarm triggered characteristic is set, the app provides an alert
- Set the high and low thresholds for the alarm characteristic
- Set the data logging characteristic on or off
- Provide a mechanism to set whether or not the data-logger is switched on via a characteristic
- Provide a mechanism to download the data-logger data over Bluetooth Low Energy and either save it to a file (Dropbox?) or email it to a user specified address as a CSV
- Implement standard Bluetooth Low Energy Time profile so that the realtime clock of the sensor can be set
- Implement standard Bluetooth Low Energy Device Information profile to read device information from the sensor

Version 1.1

- Provide a mechanism to set the “Device Firmware Update” characteristic
- Provide a mechanism to upload new device firmware to the sensor
- Implement Couchbase Lite as the datastore for sensor data
- Provide feedback to user when Couchbase Lite last performed a sync
- Allow user to manually trigger a sync

2.3 User Characteristics

The application is intended to be used by the general population and should be as straightforward to use as possible.

The use of technical terms should be avoided in the GUI or error alerts.

2.4 General Constraints

- CoreBluetooth is a general constraint as it is the low level framework for communicating with Bluetooth Low Energy devices. Wimoto has no intention to use another communication protocol at this time.
- For v1.1, we require the use of Couchbase Lite as a data store as it has built in synchronization features with our central datastore.

2.5 Assumptions and Dependencies

TBD

3. Specific Requirements

TBD

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.4 Communications Interfaces

3.2 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

3.2.1 <Functional Requirement or Feature #1>

3.2.1.1 Introduction
3.2.1.2 Inputs
3.2.1.3 Processing
3.2.1.4 Outputs
3.2.1.5 Error Handling

3.2.2 <Functional Requirement or Feature #2>

…

3.3 Use Cases

3.3.1 Use Case #1

3.3.2 Use Case #2

…

3.4 Classes / Objects

3.4.1 <Class / Object #1>

3.4.1.1 Attributes
3.4.1.2 Functions
<Reference to functional requirements and/or use cases>

3.4.2 <Class / Object #2>

…

3.5 Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).

3.5.1 Performance

3.5.2 Reliability

3.5.3 Availability

3.5.4 Security

3.5.5 Maintainability

3.5.6 Portability

3.6 Inverse Requirements

*State any *useful* inverse requirements.*

3.7 Design Constraints

Specify design constrains imposed by other standards, company policies, hardware limitation, etc. that will impact this software project.

3.8 Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.

3.9 Other Requirements

Catchall section for any additional requirements.

4. Analysis Models

List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS's requirements.

4.1 Sequence Diagrams

4.3 Data Flow Diagrams (DFD)

4.2 State-Transition Diagrams (STD)

5. Change Management Process

Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.