Software Architecture Documentation

Urban Environments Monitoring

by:

Excella

Members:

Christopher Moodley - 10457489

Lutfiyya Razak - 10198408

Juan Maree - 28004265

for:

Dr Laurie Butgereit of CSIR

V1.1

23 May 2014

Change Log/History

Date	Version	Description
17/05/2014	V0.1	Created skeleton of requirement specification
19/05/2014	V0.2	Added architectural scope and quality requirements
20/05/2014	V0.3	Added Integration requirements and access channel
21/05/2014	V0.4	Added Architectural patterns, styles and strategies
22/05/2014	V0.5	Added 4, 5, 6 & 7
23/05/2014	V1.0	Final Document Formatting

Contents

1. Architecture requirements

1.1 Architectural scope

Providing an infrastructure for serial communication between a microcontroller and a PC.

Providing a reporting infrastructure for audit log access.

Providing an infrastructure to service client requests.

1.2 Quality requirements

1.2.1 Scalability

Scalability is the highest quality requirement of the system. It should be able to handle parking lots of various sizes and layouts. This would be achieved by adding and removing secondary micro-controllers that interface with the sensors in the parking bays.

The only limit in terms of scalability would be the memory of the main micro-controller, as it determines the number of references to parking bays that may exist.

The deployed system must be able to operate effectively under the load of 5 concurrent users.

1.2.2 Usability

95% of the users should be able to use the system without prior training.

The system functions for Administrators and Security personnel that may need initial training and will also be highlighted in the system's user manual.

1.2.3 Auditability

Audit logs will be created periodically by the system, which will contain statistical information about the usage of the parking lot related to the time of day.

The system will provide functionality to view information from the audit logs and it will not allow the audit log to be modified.

1.2.4 Testability

Software

All services offered by the system must be testable through unit tests which will test

1. That a service is only provided if all pre-conditions are met and that an exception will be raised if one of the pre-conditions is not met.
2. That once a service is provided, post-conditions will hold true.

Hardware

Hardware will use sub-assembly tests to ensure correct functionality of independent units.

It will be possible for sensors to be overridden with manual switches for debugging purposes.

1.2.5 Performance Requirements

Performance is not the highest priority of the system.

- The system should update within 1 second of a parking bay changing state.
- A user should receive the latest version of the parking lot map within 10 seconds.

1.2.6 Security

Administrator functionality is only accessible after authentication of user credentials.

Security Personnel will also need credentials to gain access to their functionality.

Users will gain access to the system when they are within range of the WI-Fi server or extender located within the parking bay.

1.3 Integration and access channel requirements

1.3.1 Access channels

The system will be accessible by human users through the following channels:

1. From a PC through a Java UI for Admin and Security Personnel
2. From a mobile device with an Android operating system for Users

1.3.2 Integration channels

In order to make the system integrable so that the functionality can be accessed by other software systems, layering of the system will be done.

The secondary smaller microcontrollers connected to the sensors will communicate with the main microcontroller through a custom written protocol which will be researched and defined in future.

The system will be able to communicate with the main microcontroller through a serial port (RS232/USB). Through this port, the system will receive a binary stream as to the state of the sensors in the parking bays.

The abstraction between the sensing hardware and the system allows for alternative sensing methods to be implemented later, such as cameras.

The system will integrate with a data service which will pass objects containing map data for the parking lot in a CSV file format. The data service will also receive objects relating to the state of the system to store logging information and output a CSV file.

A separate mobile service will receive processed objects containing information regarding available and taken parking spaces from the system in CSV format. The mobile service will then integrate with mobile devices by passing the CSV data over a TLS protocol as a string.

1.4 Architectural constraints

No architecture constraints have been made by the client, the current personal choices include:

- Micro-controllers will be programmed using C
- The Server, User-Interface and Mobile Application will be programmed in java.
- Data storage will be in CSV file format.

2. Architectural patterns and styles

2.1 Hardware

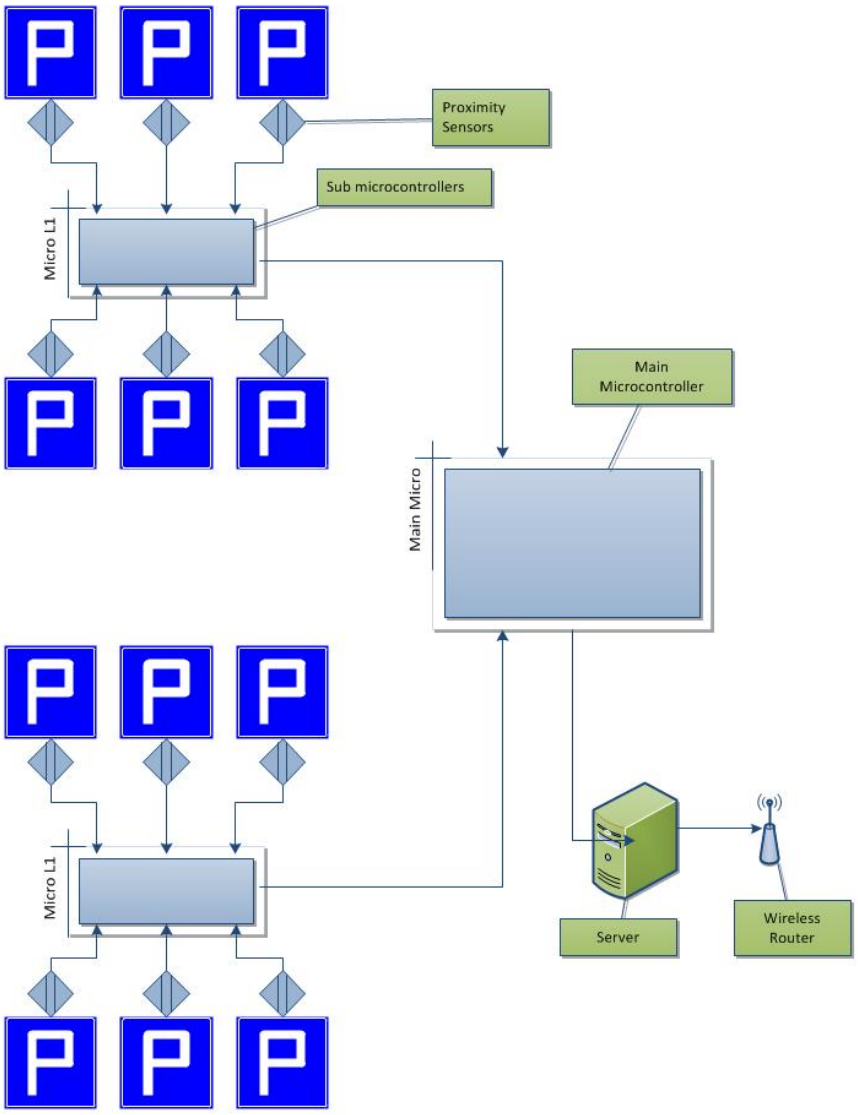

Figure 1.1: Hardware Layers

The main hardware components are shown in Figure 1.1 .The responsibility allocation across the layers is as follows:

1. Providing a wireless data connection to both mobile and java clients - Wireless router
2. Hosting the core application - Server
3. Interfacing the sensor data with the server running the main application - Main controller
4. Managing the input from sensors into a usable format - Secondary controllers
5. Detecting whether a vehicle is occupying a parking bay - Sensors

2.2 Software

The layers and the main components they include are shown in figure 1.2. The responsibility allocation across the layers is as follows:

1. Provide access to humans - Client Layer
2. Provide access to system functionality to human access layer and other systems (including decoding of client messages) - Access Layer
3. Provide domain objects - Domain Objects Layer
4. Interpret hardware serial communication to allow access to backend hardware layer - Hardware Infrastructure Layer
5. Provide a management and communication channel to sub-hardware components - Main Controller Layer
6. Provides structured access to lowest level of hardware sensors - Secondary Controller Layer

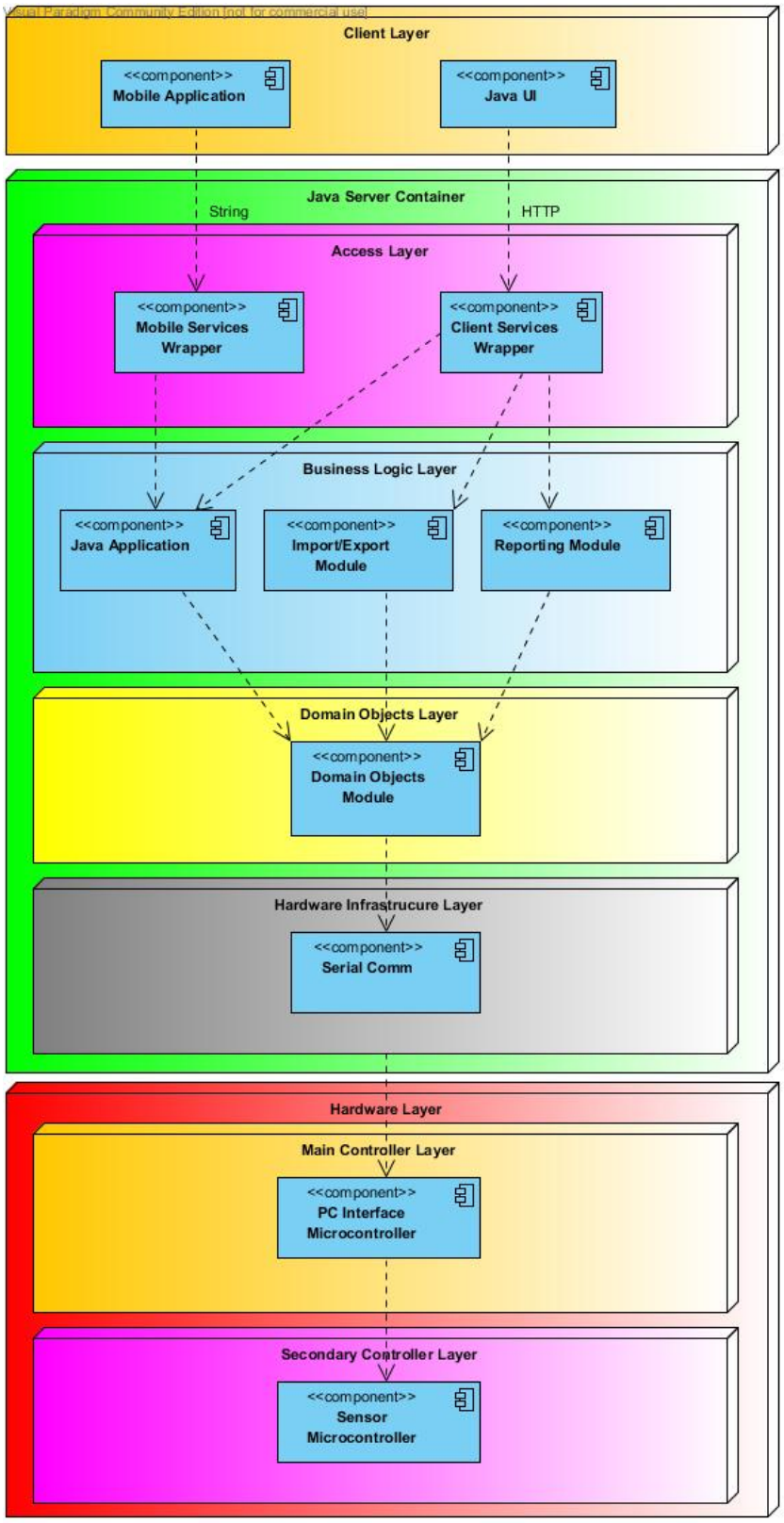
Visual Paradigm Community Edition [not for commercial use]
Client Layer
<<component>>
Mobile Application
<<component>>
Java UI
Java Server Container
String
HTTP
Access Layer
<<component>>
Mobile Services
Wrapper
<<component>>
Client Services
Wrapper
Business Logic Layer
<<component>>
Java Application
<<component>>
Import/Export
Module
<<component>>
Reporting Module
Domain Objects Layer
<<component>>
Domain Objects
Module
Hardware Infrastrucure Layer
<<component>>
Serial Comm
Hardware Layer
Main Controller Layer
<<component>>
PC Interface
Microcontroller
Secondary Controller Layer
<<component>>
Sensor
Microcontroller

Figure 1.2

3. Architectural tactics or strategies

Multi-tier Architecture will be used to aid with interoperability and maintenance by separating functional units, allowing for flexible and reusable applications. For example if the physical state of the parking lot is determined by another functional unit like image processing with cameras, this unit should be interchangeable with the proximity sensor functional unit and the rest of the system will not need to be modified to maintain a full working system. Or if an apple mobile application were to be added, all lower functionality would be retained.

Threading clients, each client will be represented as a thread to allow for concurrent users on the system and ensure that correctness properties hold whilst multiple clients are manipulating a single data structure.

To greatly improve hardware performance, secondary microcontrollers are chained to create independent functional units which will allow for a system using hardware interrupts instead of polling for sensor readings, which will greatly increase efficiency.

4. Reference architectures and frameworks

JavaFX for Java User Interface - This is the latest UI framework with most features and is well documented and supported.

PhoneGap Mobile Development Framework for multi-platform development (Written in HTML, CSS, Javascript).

Service-oriented architecture (SOA) - This architecture provides application functionality as services to other applications. This will be used to provide map functionality to mobile applications.

5. Access and integration channels

The system will be accessible by human users through the following channels:

1. From a PC through a Java UI.
2. From a mobile device with an Android operating system.

6. Technologies

- Micro-controllers will be programmed using C
- The Server, User-Interface and Mobile Application will be programmed in java.
- Data storage will be in CSV file format.

7. Glossary

Android - is an operating system based on the Linux kernel with a user interface based on direct manipulation, designed primarily for touchscreen mobile devices such as smartphones and tablet computers.

CSV - comma-separated values file stores tabular data (numbers and text) in plain-text form.

Proximity Sensor - is a sensor able to detect the presence of nearby objects without any physical contact, in this instance using infrared radiation.

Microcontroller - a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals.