**Software Design Specification (SDS)**

Revision History: (Double-Bloom for unifications)

|  |  |  |
| --- | --- | --- |
| Date | Author | Description |
| 2019.4.8 | Renxiang Zhu | The sds document |
| 2019.4.8 | Zhi Zhou | The Detailed design |
| 2019.6.4 | Zhi Zhou | Add ER, Activity Diagram |
| 2019.6.4 | Zhi Zhou | Modify the detailed design |
|  |  |  |
|  |  |  |

Contents

[1. Introduction 1](#_Toc5602370)

[1.1. Intended Audience and Purpose 1](#_Toc5602371)

[1.2. How to use the document 1](#_Toc5602372)

[2. System Design 1](#_Toc5602373)

[2.1. Context 1](#_Toc5602374)

[2.2. Design Pattern 1](#_Toc5602375)

[2.3. Architecture 1](#_Toc5602376)

[2.3.1. <Component Diagram> 2](#_Toc5602377)

[2.3.2. <Deploy Diagram> 2](#_Toc5602378)

[3. Module Interface Design 2](#_Toc5602379)

[4. Detailed Design 2](#_Toc5602380)

[4.1. Server Detailed Design 2](#_Toc5602381)

[4.2. Client (Android) Detailed Design 2](#_Toc5602382)

[4.3. Client (Web) Detailed Design 2](#_Toc5602383)

[4.4. Intelligent Control Detailed Design 2](#_Toc5602384)

[4.5. Database Detailed Design 2](#_Toc5602385)

[A.    Appendices 3](#_Toc5602386)

[A.1    Definitions and acronyms 3](#_Toc5602387)

[A.1.1    Definitions 3](#_Toc5602388)

[A.1.2    Acronyms and abbreviations 3](#_Toc5602389)

[A.2    References 3](#_Toc5602390)

## Introduction

## Intended Audience and Purpose

<Every technical document should clearly specify who the document is written for and what purpose the document should serve for each intended audience. This section describes the purpose and audience for the Concept of Operations and the Software Requirements.>

## How to use the document

<Describes the document organization. This section should answer for the reader: “Where do I find particular information about X?”>

## System Design

<Use this section to give a detailed description of the system contexts from an architect's point of view. It should make clear the expected context of the software, such as the platform, design pattern, etc.>

## Context

<Specifies the system's operational context: i.e., the programming languages to develop the software with, the operating system your software runs on, the database management system your data will be stored, the internet protocol for the component communication, etc.>

Programming languages:python

Operating system:

Linux:

* Red Hat Enterprise Linux 5.5+1, 6.x (32-bit), 6.x (64-bit)2
* Red Hat Enterprise Linux 7.x (64-bit)2 (8u20 and above)
* Ubuntu Linux 12.04 LTS, 13.x
* Ubuntu Linux 14.x (8u25 and above)
* Ubuntu Linux 15.04 (8u45 and above)
* Ubuntu Linux 15.10 (8u65 and above)

Windows:

* Windows 10 (8u51 and above)
* Windows 8.x (Desktop)
* Windows 7 SP1
* Windows Vista SP2
* Windows Server 2008 R2 SP1 (64-bit)
* Windows Server 2012 and 2012 R2 (64-bit)

Mac OS X:

* Intel-based Mac running Mac OS X 10.8.3+, 10.9+

Internet protocol:IPV4

## Design Pattern

<Specifies the technical details of the software system: i.e., model-view-control division, restful service pattern, etc.>

## Architecture

## <Component Diagram>

<Component Diagram (CD) specifies how the system is parted according to the use cases analyzed from RS. >

The whole system is divided into five parts according to the SRS document, known as the Intelligent Control(IC), the Server, the Database(DB), the Web Client and the App Client. The server part mainly deals with the interaction functions with hardware, intelligent control modules, applications and databases. Its role can be considered as a transit station, as a hub for various information processing and interaction. According to the SRS file, the server mainly provides six services: (1) stable link between hardware (2) receiving hardware response (3) accepting hardware status data (4) accepting end user requirements (5) responding to end user query (6) End user access to the database to provide access.

## <Deploy Diagram>

<这里应该是个总图>

## Module Interface Design

< It specifies the contracts with which the modules communicate.>

  (Mogic for System Interface Specifications, extra template available; all groups should contribute via interface design of her own module.)

## Detailed Design

< It specifies the design information inside the modules.>

(Each group should contribute, via diagrams either for the whole system or for her module. Optional diagrams are ER diagram, Sequence diagram, Class diagram)

## Server Detailed Design

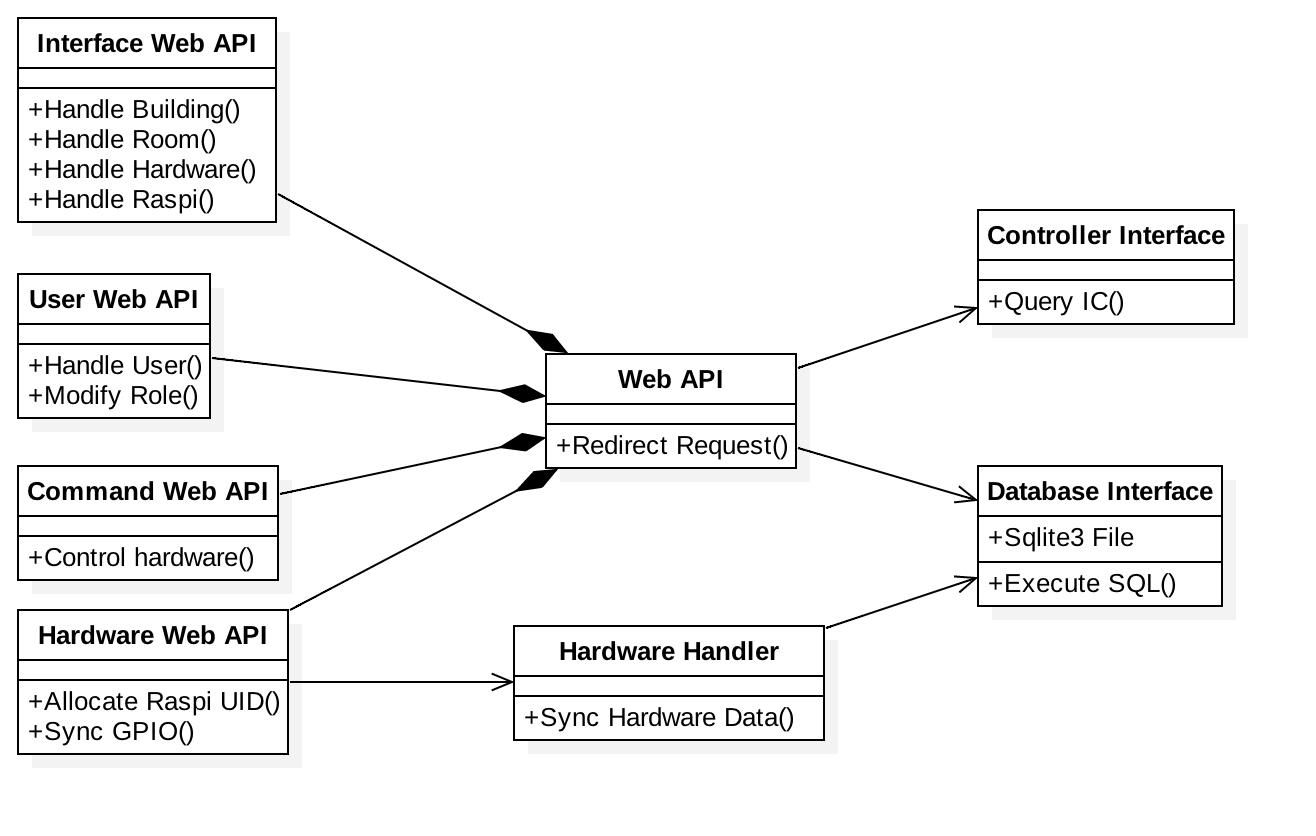


图 1 ER Diagram of Server

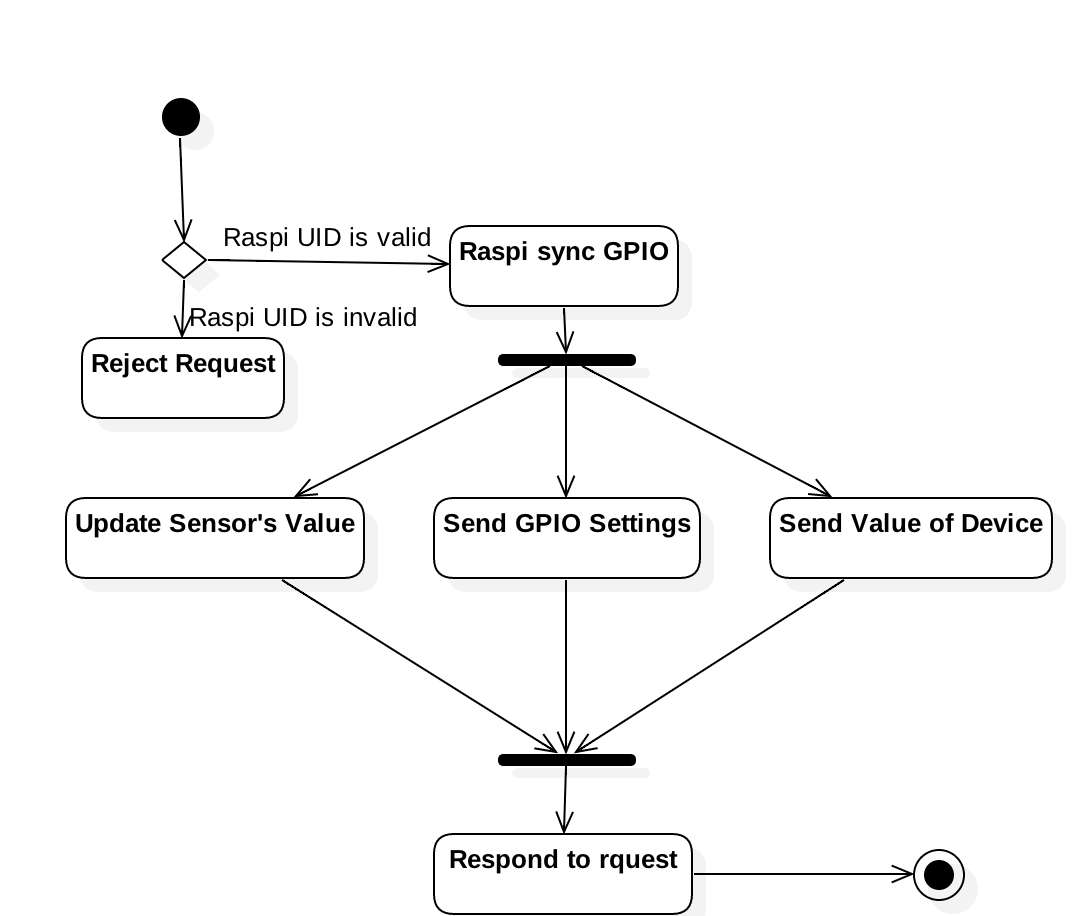


图 2 Sync GPIO Activity Diagram

## Services Provided

|  |  |  |  |
| --- | --- | --- | --- |
| # | Service | Provided By | Tested By |
| 1 | Hardware requests UID from Server. | server\_allocate | Server - T1 -5.1 |
| 2 | Hardware syncs GPIO with Server. | server\_report | Server - T3 -TC 5.3 |
| 3 | End User send command to server. | server\_command |  |
| 4 | End User updates Server’s data. | server\_update | Server - T2 -TC 5.2 |
| 5 | End user queries Server’s data. | server\_query | Server - T2 -TC 5.2 |
| # | Service | Provided By | Tested By |

## Access Method

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Access Method** | **Parameter name** | **Parameter type** | **Description** | **Exceptions** | **Map to services** |
| server\_allocate | http://SERVER\_ADDR/hardware/allocate | Nothing | Nothing | The Raspberry requests Server to allocate it a unique ID. |  | 2, 3 |
| server\_report | http://SERVER\_ADDR/command/report | data | Text in JSON format | The JSON format data should contains the following fields:   1. uid-String-Raspi Unique ID   content-String-Sensor Data | Wrong hardware UID. | 1 |
| server\_command | http://SERVER\_ADDR/command/command | data | HTTP request parameters package | The JSON format data should contains the following fields:   1. uid-Integer-User ID 2. token-String-User Token   command-String-Json String | The user don’t have permission to do that. | 4 |
| server\_update | http://SERVER\_ADDR/interface/<task> | data | HTTP request parameters package | The JSON format data should contains the following fields:   1. user-String-User Account (Authentication) 2. password-String-User Password (Authentication) 3. Other text filed according to task | The user don’t have permission to do that. | 5 |
| server\_query | http://SERVER\_ADDR/interface/<task> | data | HTTP request parameters package | The JSON format data should contains the following fields:  1. user-String-User Account (Authentication)  2. password-String-User Password (Authentication)  3. Other text filed according to task | The user don’t have permission to do that. | 6 |

## Access Method Effects

|  |  |
| --- | --- |
| **Access Method** | **Description** |
| server\_allocate | The Raspberry PI got an UID from server. And then Raspbery PI could communicate with Server by using this UID. |
| server\_report | Raspberry PI sync the data with server. To be more specifically, Raspberry PI report the sensors’ GPIO value to server and then got the latest GPIO setting and GPIO value of device (light and alarm). |
| server\_command | The server check the permission of user and then apply command to hardware. |
| server\_update | The server check the permission of user and then apply modification. |
| server\_query | The server check the permission of user and then answer the queries. |

## Services Required

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Access Method** | **Parameter name** | **Parameter type** | **Description** | **Exceptions** | **Map to services** |
| DB\_Interface | Sqlite Connection | Nothing | Nothing | Server communicate with DB file by sqlite3 connection. |  | 4, 5 |
| **Name** | **Access Method** | **Parameter name** | **Parameter type** | **Description** | **Exceptions** | **Map to services** |
| DB\_Interface | Sqlite Connection | Nothing | Nothing | Server communicate with DB file by sqlite3 connection. |  | 4, 5 |

## Client (Android) Detailed Design

## Client (Web) Detailed Design

## Intelligent Control Detailed Design

## Database Detailed Design

## A.    Appendices

## A.1    Definitions and acronyms

## 

## A.1.1    Definitions

|  |  |
| --- | --- |
| **Keyword** | **Definitions** |
|  |  |
|  |  |
|  |  |
|  |  |

## A.1.2    Acronyms and abbreviations

|  |  |
| --- | --- |
| **Acronym or**  **Abbreviation** | **Definitions** |
|  |  |
|  |  |
|  |  |

## A.2    References