Brainsense

Architecture Design

Version 1.1

Reference

Date	Version	Description	Author	Status
08/01/14	1.1	PRD	Roni Eshett	Draft

Revision History

Date	Ver.	Description	Resp.	Approved	Status
09/01/14	1.0	Initial design	Victor	Roni	Draft
25/02/14	1.1	Add DB design, Update system design	Victor	Roni	Final

Table of Contents

Language and Tools

Language

Language: Java
Web service: Restful (Jersey)
Web server: Tomcat
Coding Criteria: Sun Coding Standard
Database: MySql

Development Tools
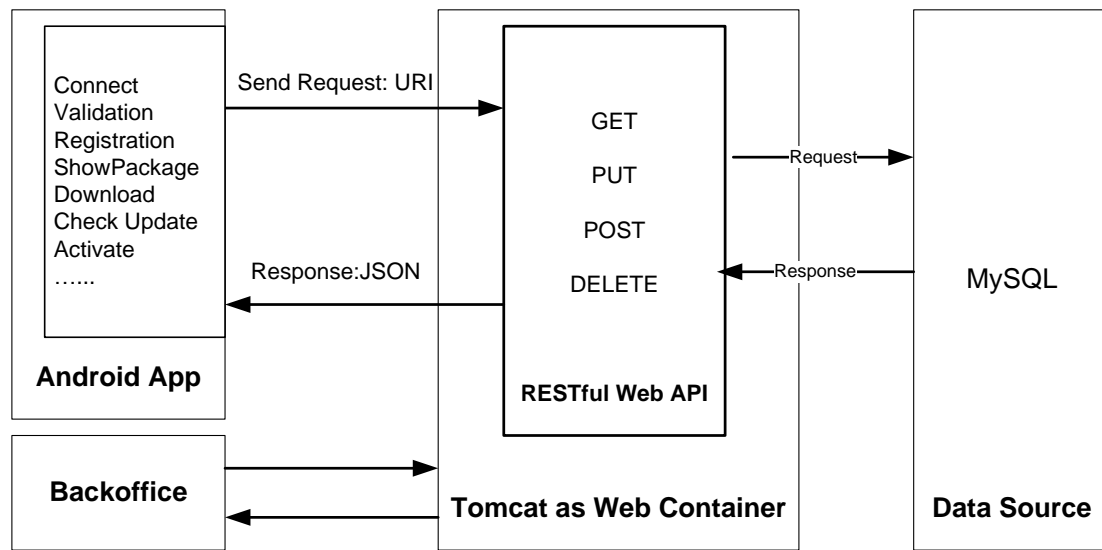
Development Tools: Eclipse and android ADT
Database Tool: Workbench

System Design

The core of system adopted RESTful Web as architecture style, and it is based on HTTP, URI and Json Protocol and standard.

A uniform interface separates clients from servers. This separation of concerns means that, for example, clients are not concerned with data storage, which remains internal to each server, so that the portability of client code is improved. Servers are not concerned with the user interface or user state, so that servers can be simpler and more scalable. Servers and clients may also be replaced and developed independently, as long as the interface between them is not altered.

Below diagram (P-1) describes how to interact between different system (Android App and Web App) and RESTful web.

Connect
Validation
Registration
ShowPackage
Download
Check Update
Activate
…...

Android App

Send Request: URI

Response:JSON

GET

PUT

POST

DELETE

RESTful Web API

Request

Response

MySQL

Data Source

Backoffice

Tomcat as Web Container

P-1

Network Architecture

The whole system includes client side and server side from the functional point of view. As well it includes three layers, application layer, data transport layer and server layer.

Application layer: End user can download and install the app from server.

Transport layer: Data can be transported by internet.

Server Layer: receive request and deal with the data and then response data to end user.   The relevant relationship is the network diagram below (P-2).
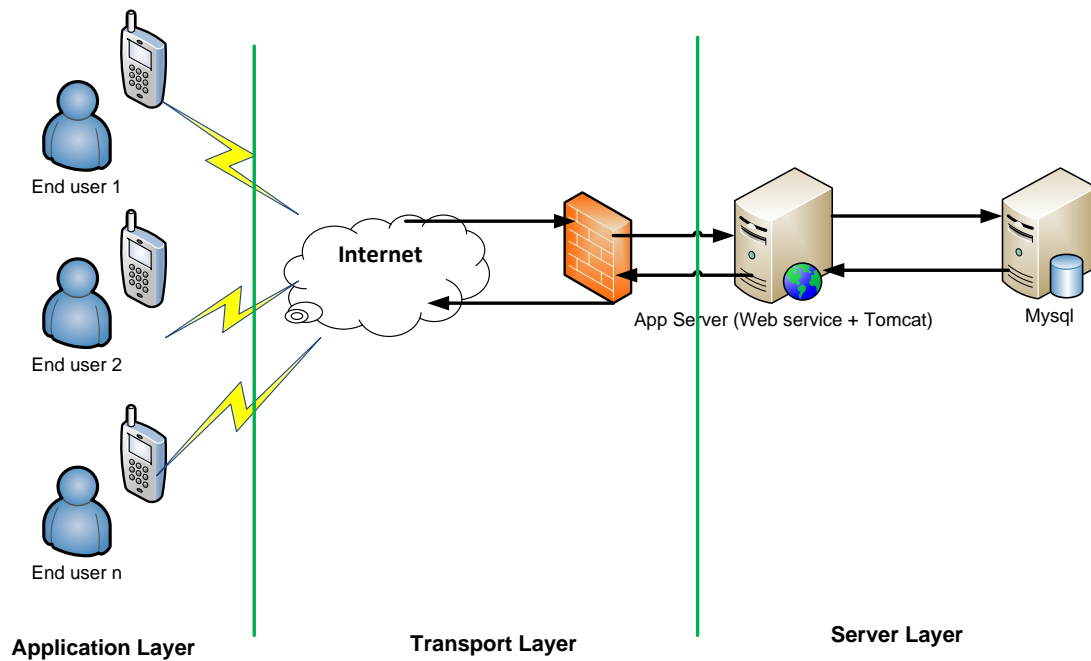
P-2

Modules Design

Use Case Diagram

Based on the PRD, the system includes client side and server side. As well as it should include different role, user includes end user and server admin.

Client side

Client side should include main features below:

Hardware registration, user registration, key code validation, choose and download package, check and update software version and choose different language, etc.
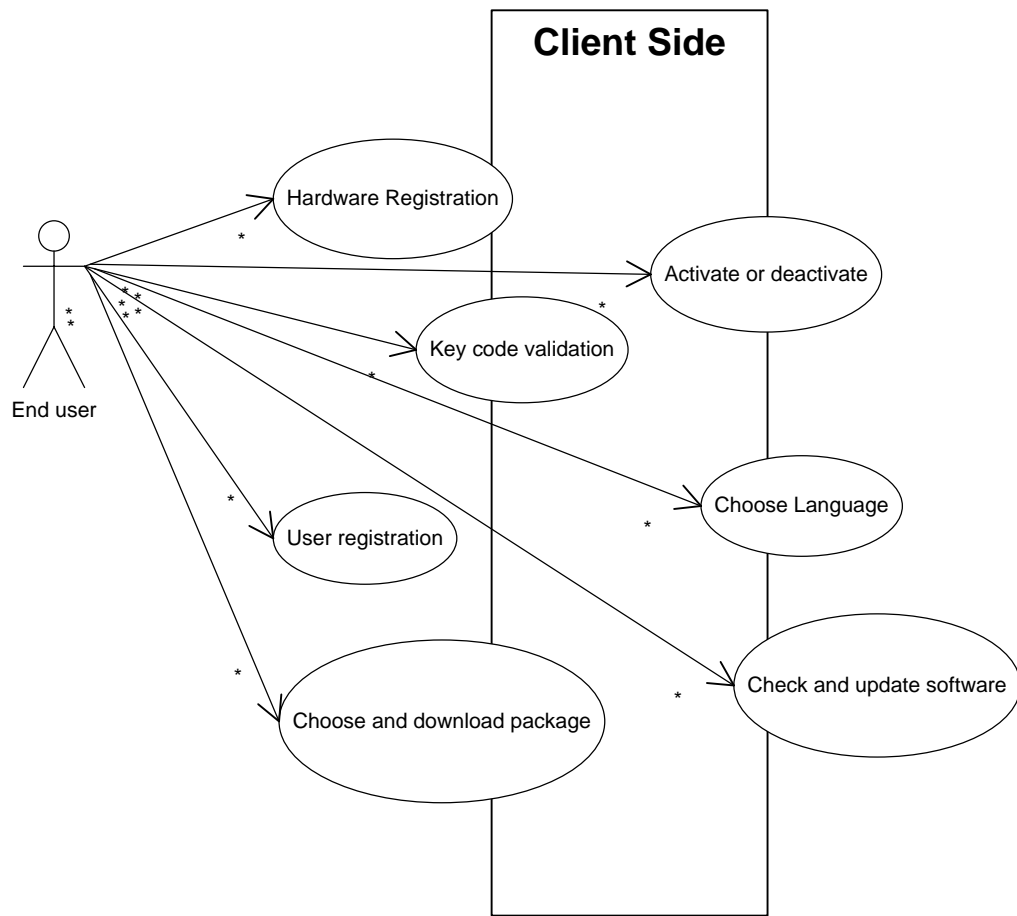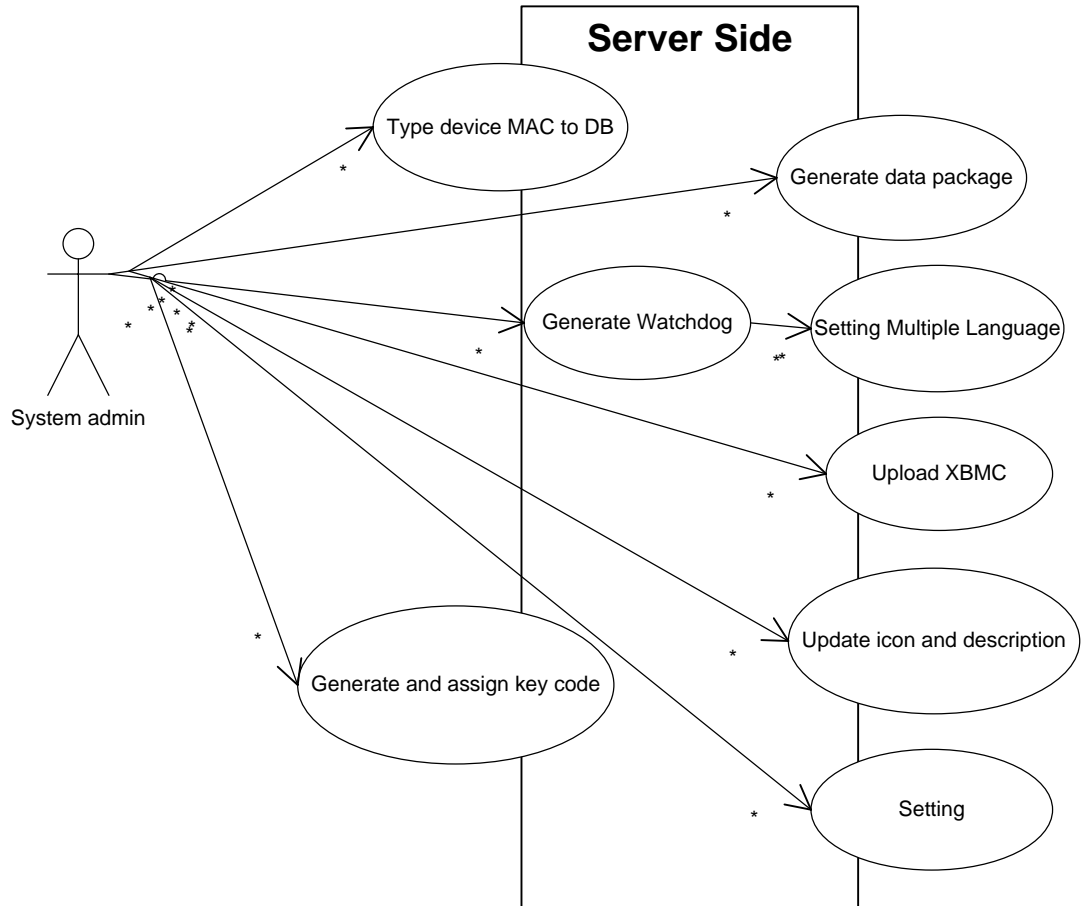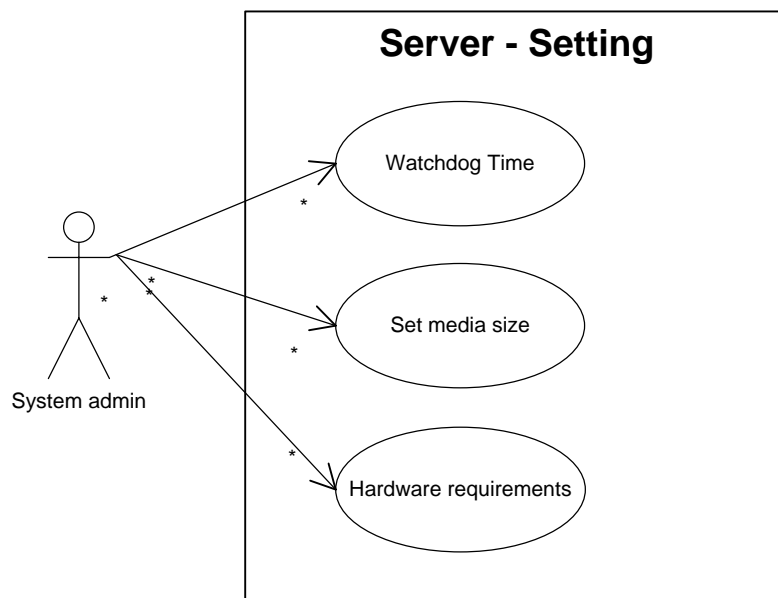
P-3

Server Side

Server side should include main features below:

Import device mac to DB, create apk file, download apk, generate key code, update hardware requirements, update language (English and Hebrew) and update icon and description, etc.

Server Side

- Type device MAC to DB
- Generate data package
- Generate Watchdog
- Setting Multiple Language
- Upload XBMC
- Update icon and description
- Generate and assign key code
- Setting

System admin

P-4

Server - Setting

- Watchdog Time
- Set media size
- Hardware requirements

System admin

P-4.1

Sequence Diagram

Hardware registration and validation

| View | UserRestful | UserAction | DeviceService | DeviceDAO |

1.0 getDeviceMac()

Top Package::End user

1.1 send the device MAC address

1.2 validateDevice(devicemac)

1.3 validateDevice(devicemac)

1.4 getDeviceByDeviceId(devicemac)

If exists device mac

1.5 return device class

1.6 return true or false

1.7 return true or false

1.8 return true or false

Procedure description:
1.0 get user device mac;
1.1 send the device mac to restful;
1.2 - 1.4 validation if exists device mac in database;
1.5 according to return value(true or false), client side will display reject message or jump step2.
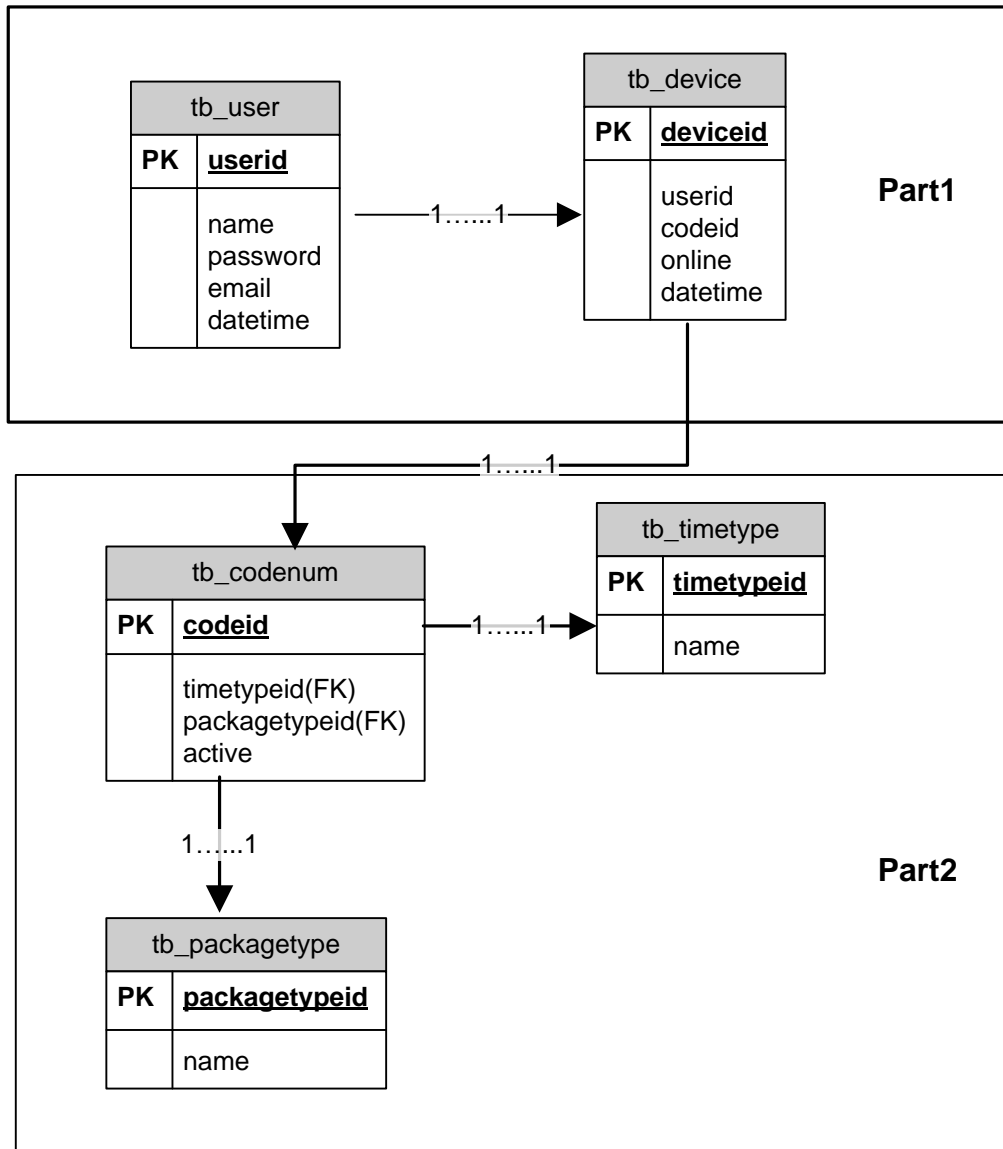
P-5

Database Design

E-R Diagram

Associated between user, device and key code

P-6

Description:

Part 1:

1, one user only can bind with a device, so tb_user and tb_device are one to one relationship.

Part 2:

2, one device can bind with a software key code number, so tb_device and tb_codenum are one to one relationship.

3, a software key code number can relate to a type of package and a type of time, so tb_codenum and tb_timetype are one to one relationship, as well as tb_codenum and tb_packagetype are one to one relationship.

Associated between package, package history, and category of package (country, content)
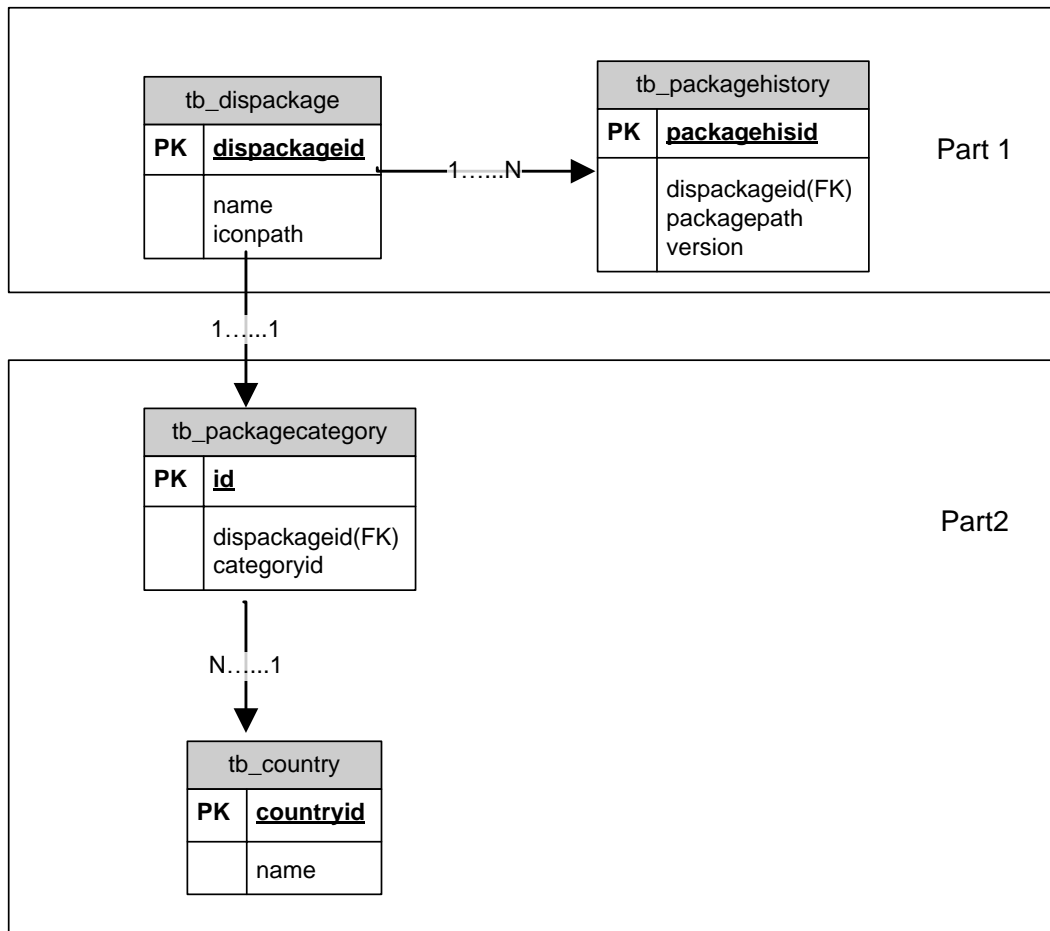
P-7

Description:

Part 1

A package with latest version is stored in tb_dispackage; history version is stored in tb_packagehistory. So tb_dispackage and tb_packagehistory are one to multi relationship.

Part 2

A package with latest version is stored in tb_dispackage; the package associated to category of country and content is stored in tb_packagecategory, so tb_dispackage and tb_packagecategory are one to multi relationship.

Data Dictionary

Table Name	Description
tb_user	Storage user's information
tb_device	Storage user's device information
tb_version	Storage the version number of app
tb_codenum	Storage software key code

tb_timetype	Category: Basic table
	Storage the type of time. It is as attribute of package.
tb_packagetype	Storage the type of package. It is as attribute of package.
tb_dispackage	Storage the display information of package for user
tb_packagehistory	Storage the version information of data updated package
tb_hardwarereq	Storage hardware requirements
tb_userhardware	Storage user hardware configuration
tb_country	Category: Basic table
	Storage type of country, It is as attribute of package.
tb_content	Category: Basic table
	Storage type of content, it is associated to tb_country.
tb_packagecategory	Storage category of package, It is as attribute of package.
tb_group	Category: Basic table
	Storage group of package, It is as attribute of package.
tb_public	Category: Basic table
	Storage configuration information in backoffice
tb_operate	Storage delete information when MAC or keycode is deleted
tb_localinfo	Storage local installation information
tb_admin	Storage administrator's info
tb_wdhistory	Storage version history info of watchdog

Tab-1

Project Structure

- Brainsense
  - JAX-WS Web Services
  - Deployment Descriptor: Brainsense
  - Java Resources
    - src
      - com.matrix.brainsense.action
      - com.matrix.brainsense.dao
      - com.matrix.brainsense.dao.impl
      - com.matrix.brainsense.entity
      - com.matrix.brainsense.filter
      - com.matrix.brainsense.restful
      - com.matrix.brainsense.service
      - com.matrix.brainsense.service.impl
      - com.matrix.brainsense.servlet
      - com.matrix.brainsense.test
      - com.matrix.brainsense.thread
      - com.matrix.brainsense.tool
      - jdbc.properties
      - timetype.properties
    - Libraries
  - JavaScript Resources
  - build
  - WebContent
    - Android
    - apkTool
    - Brainsense_Android
    - css
    - images
    - js
    - META-INF
    - uploadTool
    - WEB-INF
    - apkManager.html
    - DeleteRegister.jsp
    - Generate PinNum.html
    - GenerateAPK.html
    - GenerateWatchdog.html
    - HardwareReqManage.html
    - Login.html
    - MatchDeviceAndCodeNum.html
    - NewPackageManager.html
    - PackageTypeManage.html
    - SetDataFiles.html
    - SetMediaFolderSize.html
    - SetWatchDogTime.html
    - updateXMBC.html
    - waiting.html

P-8