1. System architecture
2. Functional requirement
   1. Appliance Sub Controller Functional Requirement

|  |  |
| --- | --- |
| Arduino Module | |
| ASCFR001 | IR Emitter |
|  | Receive IR pattern from the RF Receiver.  IR patterns are stored in a integer array indicating the on and off time interval of the IR LED.  Decode the IR pattern into Arduino code.  Emit IR signal to turn on or off electronic appliances. |
| ASCFR002 | RF Emitter and Receiver |
|  | Connect Raspberry Pi and Arduino through RF.  After receiving command, delegate to IR Emitter. |
| ASCFR003 | Sensor |
|  | Get sensor data from sensors on Arduino.  Process the data and store it in the memory.  After the preset interval, send the data to main controller for storage.  For temperature sensor and humidity sensor, capture data in every 1 minute, and send back the average in every 5 minutes to the main controller.  For motion sensor, notify the main controller immediately if the state changes. |
|  | |
| Light Arduino Module | |
| ASCFR004 | Relay |
|  | Turn on or off electronic appliances by relay.  Receive command from the RF receiver and control the relay  accordingly. |
| ASCFR005 | RF Emitter and Receiver |
|  | Connect Raspberry pi and Arduino through Wifi. |
| ASCFR006 | Switch State Monitor |
|  | Monitor the light switch state, if the user switches the light via mechanical switch, the state of the switch will change, fire an StateChange event to IR emitter to update the information to main controller.  If the Arduino is rebooted, check and update the switch state. |

* 1. Cellphone Functional Requirement

|  |  |
| --- | --- |
| Display Module | |
| CFR001 | Main Activity |
|  | Display user interface which user is using. |
| CFR002 | Notification |
|  | Notify user when the event is occurred .e.g. the appliance need to be turn on or off |
|  | |
| Authentication Module | |
| CFR003 | User Login |
|  | Check whether the password which users enter correspond to the email address of users. |
| CFR004 | User Registration |
|  | Handle the user list for different email address. When some users start to use the product. Check the correctness of product ID and make sure the user's email address isn’t duplicated. |
|  | |
| Appliance Module | |
| CFR005 | Favorite |
|  | Favorite is a user interface which user can set his favorite appliances to this module  Favorite need to display favorite choice as circular button. |
| CFR006 | Appliance |
|  | Appliance is a user interface which displays all the appliances.  Include special appliance: TV , Light , Air , Conditioner |
| CFR007 | TV |
|  | TV is a user interface which looks like as remote controller  TV must send command to the appliance module when user click button. |
| CFR008 | Light |
|  | Light is a user interface which display in normal. |
| CFR009 | Air Conditioner |
|  | Air Conditioner is a user interface which let user to set the temperature up or down. |
| CFR010 | Others |
|  | Others are also user interface which display in normal view. |
| CFR011 | Add |
|  | Add is a user interface which let user to add appliance |
|  | |
| Chart Module | |
| CFR012 | Chart |
|  | Chart delegate the plot mission to chart TV and chart AV and displayed what they plot. |
| CFR013 | Chart Light |
|  | Chart light plot a chart which shows light used time and price of electricity |
| CFR014 | Chart AC |
|  | Chart AC plot a chart which shows air conditioner used time and price of electricity. |
|  | |
| User Module | |
| CFR015 | Account |
|  | Account is a user interface which lets user to check his account or change password. |
| CFR016 | Profile |
|  | Profile is a user interface which appears when user click a button on Account interface.  Profile shows user’s account and password and let user to change the password. |
| CFR017 | Contact Us |
|  | Contact us provide email address to let user to send email to us. |
|  | |
| Communication Module | |
| CFR018 | Send |
|  | Receive data from other module.  Send data to main controller by cloud server. |
| CFR019 | Receive |
|  | Receive data from main controller by cloud server.  Send data to other module. |
|  | |
| Setting Module | |
| CFR020 | Time Setting |
|  | Time Setting is a ui let user to set when will the appliance to be turned on/off  Time setting must have |
| CFR021 | Proximity Setting |
|  | Proximity Setting is a ui to let user to set how close they are to the home to turn on/off the appliance |
| CFR022 | Energy Saver |
|  | Energy Saver is a user interface which user can set turn on/off appliance when motion sensor don’t detect any person for how many time. |
| CFR023 | Ideal Temperature |
|  | Ideal temperature is a user interface which lets user to set the ideal temperature they want. Briefly the ideal temperature module turn on/off the appliance when time arrived. |

* 1. Cloud server Functional Requirement

|  |  |
| --- | --- |
| Authentication Module | |
| CSFR001 | User login |
|  | Parse the string into email, password, and check the password.  Send the result back to the cell phone. |
| CSFR002 | New Account |
|  | Parse the string into email, password and product id.  Check if the email is duplicated, send false message to cell phone if duplicated.  Check if the product id is Valid, send the result back to the cell phone.  If the product id is valid, save the mapping between the product id and the email in the Router Module. |
| Router Module | |
| CSFR003 | User-controller table |
|  | Store the mapping between the product id and the email |
| CSFR004 | fd-controller table |
|  | Store the mapping between the product id and the Socket.  Store the mapping between the cell phone email and the Socket”s”. |
| Server Service | |
| CSFR005 | Receive POST requests from companies to update the IR code, and store them into Database. |
| Database | |

* 1. Main Controller Functional Requirement

|  |  |
| --- | --- |
| Socket Module | |
| MCFR001 | Send Message |
|  | Connect to the cloud server via TCP connection.  Receive message from other modules, and send them to the cloud server. |
| MCFR002 | Receive Message |
|  | Receive message from the cloud server, and delegate to Parse Message. |
| MCFR003 | Parse Message |
|  | Parse the message and transmit to specific module. |
|  | |
| Management Module | |
| MCFR004 | Status |
|  | Keep track on every appliance’s state.  Receive message if the state of any appliance changes.  Send message to History to save the state change. |
| MCFR005 | History |
|  | Save all the appliance state change and sensor data to the database.  If the user requests to plot the history chart, query the required data and send to Send. |
| MCFR006 | Device Controller |
|  | Check if the arduino id is a valid and existing arduino.  Receive message from the Socket module, and either send message to RF module to perform command immediately or schedule the command to the Scheduler module. |
| MCFR007 | Decision maker |
|  | Receive parameters such as ideal temperature from Parse, and save it in the database.  Keep track of Status to see if the current status is in the range, send notification to user and ask if they want to turn on appliances.  Decisions include:  **ideal temperature environment**, automatically maintain the ideal temperature and ideal humidity;  **motion sensing light control**, if the motion sensor detects motion, then turn on the light, and after the person leaves the detectable range, turn off the light.  **save energy**, when the motion sensor detects nothing in a predefined duration (default 1 hours), turn off the light;  All of the above functions can change to send notification to user to decide whether to turn on or not.  Send on/off command to commander. |
| MCFR008 | Device Info |
|  | Store the Arduino device type, arduino id, arduino nickname into database. |
| Scheduler Module | |
| MCFR009 | Scheduler |
|  | Receive appliance id, start time, and duration command from management module.  Keep track of the time, data, and the up coming event.  Send command to RF module when it’s time.  Save the schedule in database in case if the main controller reboots. |
|  | |
| Internet Module | |
| MCFR010 | Internet connection |
|  | Receive wifi username and password from cellphone module via bluetooth.  Use username and password to connect to wifi.  Save wifi username and password to database.  Notify the Socket Module to connect to Cloud Server.  If the internet connection is lost, turn on the bluetooth and see if the user will connect and update Wifi connection data. |
|  | |
| RF Module | |
| MCFR011 | RF Transmitter |
|  | Receive commands from other module, and transmit according RF signal out to Arduino module.  When the user is adding a new Arduino, broadcast the add-new-arduino signal. |
| MCFR012 | RF Receiver |
|  | Receive RF signals from Arduino, and send them to other modules accordingly.  If receive an acknowledgement signal indicating successful switching appliance or the state of switch has changed, write the update message to Status.  If receive the add-new-arduino signal’s response, add the arduino id to device info. |
|  | |
| Appliance Module | |
| MCFR013 | Add Appliance |
|  | Receive the new Arduino id from Parse, send to RFTX to broadcast the add-new-arduino signal.  If receive the acknowledgement signal from RFRX, save it in device info and database. |
| MCFR014 | Add IR Pattern |
|  | Receive the new IR Pattern from parse including the arduino id to be added in, and the IR source.   * If the IR source is from the cloud server:   + Download it from the cloud server and send the testing IR pattern to RFTX, and wait for the user to confirm.   + Testing IR pattern is noted specially so the result won’t be recorded to History.   + If the user confirmed to add this IR pattern to the specific arduino, save it to the device info and database. * If the IR source is a custom IR pattern:   + Turn on the IR receiver and start listening to incoming IR signal.   + decode the IR pattern and send a message to the Send module to tell the user that the recording process is done, and wait for the user to test it.   + If the user confirms, save the IR pattern to the database with a new ID and nickname. |
|  | |
| Database | |
| MCFR015 | Database |
|  | Store IR code, a table of user account, appliances and id numbers, chart information (temperature, humidity), wifi username and password. User habits (how many times and how long does the appliance turn on per month) |

1. External Interface Requirement

3.1 Cellphone External Interface Requirement

|  |  |
| --- | --- |
| CEIR001 | User to Main Activity |
|  | User do all activities through main activity page which will be continuing changed when user click a button or some. |
| CEIR002 | User to Notification |
|  | Notification do all notification things which user can know what happen in his home. |

3.2 Cloud Server External Interface Requirement

|  |  |
| --- | --- |
| CSEIR001 | Update IR code |
|  | Company can upload the IR code of their new product for users.  The updated data have to be in IR code format. |
| CSEIR002 | Administrator Login |
|  | Check the correctness of the administrator’s password. |

1. Internal Interface Requirements

4.1 Appliance Sub Controller Internal Interface Requirements

|  |  |
| --- | --- |
| Internal Interface Requirements | |
| ASCIIR001 | Turn on/off command by IR signal |
|  | Wifi Emitter and Receiver send command from main controller to IR Emitter. |
| ASCIIR002 | Turn on/off command by relay |
|  | Wifi Emitter and Receiver send command from main controller to Relay. |
| ASCIIR003 | Sensory data transmit |
|  | Sensor transmit sensory data to Wifi Emitter and Receiver . |
| ASCIIR004 | Main Controller command to Arduino |
|  | Main Controller send turn on/off command to Arduino by Wifi.  Main Controller send appliance ID and IR pattern to tell the subcontroller which appliances to command.   * Appliance ID is type of int which save defined appliances information. * IR pattern may have five formats follows with hex number (e.g. SONY 68B92) , or Raw format follows with raw code   + NEC   + Sony   + RC5   + RC6   + Raw   Main Controller and Sub Controller connect each other by TCP/IP protocol. |
| ASCIIR005 | Main Controller command to Light Arduino |
|  | Main Controller send turn on/off command to Light Arduino by Wifi. |
| ASCIIR006 | Transmit sensory data to Main Controller |
|  | Arduino transmit sensory data to Main Controller by Wifi connection. |
| ASCIIR007 | Send an update if the switch state changes |
|  | Tell the RF transmitter to send a message to the main controller to update the switch state. |

4.2 Cellphone Internal Interface Requirement

|  |  |
| --- | --- |
| CIIR001 | Main Activity to User Login |
|  | User Login display itself when user click login in button on main activity |
| CIIR002 | Main Activity User Registration |
|  | User Registration display itself when user click registration button. |
| CIIR003 | Main Activity to Favorite |
|  | Favorite display itself when user click favorite button. |
| CIIR004 | Main Activity to Appliance |
|  | Appliance display itself when use click appliance button. |
| CIIR005 | Main Activity to Chart |
|  | Chart display itself when user click chart button |
| CIIR006 | Main Activity to Account |
|  | Account display itself when user click account button |
| CIIR007 | Appliance Module to Main Activity |
|  | Appliance Module will display child interface when user click a button such as TV , Light … |
| CIIR008 | Chart Module to Main Activity |
|  | Chart module display its child when user want to see his electricity usage |
| CIIR009 | Appliance to Notification |
|  | Appliance will send message to Notification to notify user which event occur. |
| CIIR010 | User Module to Main Activity |
|  | User Module will display itself when user click the to see his profile |
| CIIR011 | Appliance to TV |
|  | TV will show itself to appliance interface when user click TV button |
| CIIR012 | Appliance to Light |
|  | Light will show itself to appliance interface when user click light button |
| CIIR013 | Appliance to Air conditioner |
|  | Air conditioner will show itself to appliance interface when user click Air conditioner button |
| CIIR014 | Appliance to Others |
|  | Other will show itself to appliance interface when user click Others button |
| CIIR015 | Appliance to Add |
|  | Add interface will show itself to appliance interface when user click Add button |
| CIIR016 | Chart to Chart Light |
|  | Chart will get Chart Light’s plot when user wants to see Light’s usage |
| CIIR017 | Chart to Chart AC |
|  | Chart will get Chart AC’s plot when user wants to see AC’s usage |
| CIIR018 | User Login to Send |
|  | User Login will make a user login command which get info from user interface.  Command looks like : /UserLogin [username] [password] |
| CIIR019 | User Registration to Send |
|  | User Login will make a user registration command which get info from user interface.  Command looks like : /UserRegistration [username] [password] |
| CIIR020 | Favorite to Send |
|  | Favorite will make a appliance on/off command to Send  Command looks like : /switch [username] [applianceID] [on/off] |
| CIIR021 | Appliance to Send |
|  | Appliance will make a appliance on/off command to Send  Command looks like : /switch [username] [applianceID] [on/off] |
| CIIR022 | Air Conditioner to ideal temperature |
|  | Ideal temperature display itself when user wants to set air conditioner ideal temperature. |
| CIIR023 | L.A.O to T.P.E |
|  | Light, Air conditioner, Others shows Time setting, Proximity setting, Energy Saver interface when user click button. |
| CIIR024 | Add to Send |
|  | Add will make a appliance added command to Send  Command looks like : /addAppliance [username] [applianceID] |
| CIIR025 | Chart Light to Send |
|  | Chart Light send a message to Send to tell he : I want plot data!!! |
| CIIR026 | Chart AC to Send |
|  | Chart AC send a message to Send to tell he : I want plot data!!! |
| CIIR027 | Receive to Chart Light |
|  | Receive send Light usage data to Chart Light to plot.  Data include time usage , price of electricity |
| CIIR028 | Receive to Chart AC |
|  | Receive send AC usage data to Chart AC to plot.  Data include time usage , price of electricity |
| CIIR029 | T.P.E.I to Send |
|  | Time, Proximity, Energy Saver, Ideal Temperature make command to Send to tell user’s setting.  Time command looks like: /timeSet [username] [applianceID] [startTime] [duration]  Proximity command looks like: /proximitySet [username] [applianceID] [distance]  Energy Sacer command looks like : /EnergySaver [useranem] [appliance] [times]  Ideal Temperature commands looks like: /ideaaTemp [username] [appliacneID] [upTmep] [lowTemp] |
| CIIR030 | Receive to T.P.E.I |
|  | Receive notify the Time, Proximity, Energy Saver, ideal Temperature when main controller have done the mission.  Notification message looks like: /mission [username] [missionType] [succecc] |

4.3 Cell phone Cloud Server Internal Interface Requirement

|  |  |
| --- | --- |
| CSIIR001 | Send to User Login |
|  | Send will send command to Server User login to tell server I want to Login |
| CSIIR002 | Send to new account |
|  | Send will send command to Server new account to tell server I want to create a new account. |
| CSIIR003 | Send to Server Router |
|  | Send will send commands to Server Router to tell server who am I and what I want to do. |
| CSIIR004 | Send to fd-controller table |

4.4 Cloud Server Internal Interface Requirement

|  |  |
| --- | --- |
| CSIIR001 | User and controller authentication |
|  | Database hold a table which maps user and controller.  Database save administrator and companies’ login information. |
| CSIIR002 | Data forward |
|  | Cloud server will forward data to user cellphone when chart need to be displayed. |

4.5 Main Controller Internal Interface Requirement

|  |  |
| --- | --- |
| MCIIR001 | Commands transform |
|  | Assignment module’s Receive will deliver on/off command, functionalities settings to cellphone module’s Command. |
| MCIIR002 | Draw chart command transform |
|  | Assignment module’s Receive will deliver draw chart command to cellphone module’s Chart. |
| MCIIR003 | Scan IR command transform |
|  | Assignment module’s Receive will deliver scan IR code command to cellphone module’s IR scan. |
| MCIIR004 | User location |
|  | Assignment module’s Receive will deliver cellphone GPS location to cellphone module’s GPS. |
| MCIIR005 | Appliance on/off message |
|  | Command will send appliance on/off message to assignment module’s Send. |
| MCIIR006 | Chart information |
|  | Chart will send chart information to assignment module’s Send. |
| MCIIR007 | IR scan message |
|  | IR scan will send “scan finished” message to assignment module’s Send. |
| MCIIR008 | Notification transform |
|  | Notification will send notifications to assignment module’s Send. |
| MCIIR009 | Command to Scheduler |
|  | Command send user id, appliance id, start time, duration commands to set Scheduler. |
| MCIIR010 | Command to decision maker |
|  | Command send temperature, humidity, settings to set decision maker. |
| MCIIR011 | Command to commander |
|  | Command send on/off command to commander. |
| MCIIR012 | Scheduler notification |
|  | Scheduler will send message to notification when needed.  Notify when Schedule mission executed. |
| MCIIR013 | Decision maker notification |
|  | Decision maker will send message to notification when needed.  Notify when   1. Temperature out of range 2. Humidity out of range 3. User nearby home |
| MCIIR014 | GPS to decision maker |
|  | GPS send user location to decision maker. |
| MCIIR015 | Database to Chart |
|  | When needed, Chart get chart information from database. |
| MCIIR016 | IR code to database saving |
|  | IR scan send IR code to database. |
| MCIIR017 | Scheduler to commander |
|  | Scheduler send on/off command to commander. |
| MCIIR018 | Decision maker to commander |
|  | Decision maker send on/off command to commander. |
| MCIIR019 | Commander to Arduino communicator |
|  | Commander send on/off command, IR code, and appliance id to Arduino communicator. |
| MCIIR020 | Commander and database |
|  | Commander get IR code and from database.  Commander send user habits to database saving. |
| MCIIR021 | Setting to database |
|  | Save wifi settings from parameter setting. |
| MCIIR022 | Update IR code |
|  | Receive IR code from IR receiver. |
| MCIIR023 | Wifi setting to parameter setting |
|  | When first connected to cellphone through blue tooth, Wifi setting sets wifi information includes username and password to Parameter Setting. |
| MCIIR024 | Appliance Sub Controller data received |
|  | Decision maker will receive sensor data from Arduino communicator.  Sensor data includes temperature, humidity and motion sensor. |
| MCIIR025 | Save sensor data |
|  | Sensor Data will be saved in txt file format.  Save a year data in txt file which file size may not bigger than 52KB. |

4.5 Main Controller Internal Interface Requirement

|  |  |
| --- | --- |
| MCIIR001 | sending socket to parsing class |
|  | it send socket to the parsing class,and parsing class will parse the socket string |
| MCIIR002 | sending command to management module according to parsing result |
|  | according to the parsing result,it send different command to the specific object which is in management module. |
| MCIIR003 | sending command to appliance module according to parsing result |
|  | according to the parsing result,it send different command to the specific object which is in appliance module. |
| MCIIR004 | return the appliance status and environment status to user |
|  | if the user want the status in his home(on/off ,temperature,humidity),it will send all information to the sending class,so that the user can get the information. |
| MCIIR005 | return the appliance using history to user |
|  | if the user want the using history of each appliance in his home,it will send the using history to the sending class,so that the user can get the information. |
| MCIIR006 | send the notification to user |
|  | if the main controller find the temperature is so high and want to suggest user open the air conditioner, and it will send the notification to the sending class,so that the user can get the notification. |
| MCIIR007 | send all device information to the user |
|  | if the user first register this main controller,and he is not the first user of this main controller, the main controller will send the existing appliance to the user, and he won’t need to add existing appliance by himself. |
| MCIIR008 | tell the user the appliance has turned on or off |
|  | when the scheduler turn on/off the appliance according to the schedule which is sat by the user,it will send the notification to notify the user the appliance has turner on/off. |
| MCIIR009 | Update the history of user |
|  | When the status of appliance changes, it will notify the history module to update the history data. |
| MCIIR010 | Check the ideal environment |
|  | Decision maker would compare the ideal temperature and humidity with the present environment, and make decision to change the appliance setting based on that. |
| MCIIR011 | Store history data into database |
|  | Store history data into the database of main controller, and load the data from database when users request. |
| MCIIR012 | Store the device information into database |
|  | Store device information into the database when new appliance was added to the system. |
| MCIIR013 | Send scheduling command |
|  | Send scheduling command to scheduling module. The scheduling command from users would be executed by scheduling module. |
| MCIIR014 | Send immediately command from user |
|  | Send the command from user to RF module immediately for some immediately command from user. |
| MCIIR015 | Send command from system |
|  | After the decision making, the system would make some system command to appliance, and send these command to RF module to execute. |
| MCIIR016 | Tell the New Appliance module that the new arduino has been successfully detected. |
|  | Tell the New Appliance module that the new arduino has been successfully detected. the New Appliance module can then add it to device info. |
| MCIIR017 | Update the Status after receiving the acknowledgement signal |
|  | After transmitting a command to other arduino modules, to ensure the arduino really gets the message, the arduino will send an acknowledgement signal back.  After receiving this acknowledgement signal, tell Status to update. |
| MCIIR018 | Save Wifi SSID and password in the database |
|  | After receiving the Wifi SSID and password from the user, save it in the database. |
| MCIIR019 | Update the Device info after adding a new appliance |
|  | After the user adds a new appliance, such as a new arduino, save it in the Device info. |
| MCIIR020 | Save new added IR pattern into database |
|  | Save the IR pattern nickname, arduino id and every pattern’s name, content into database. |
| MCIIR021 | Send signal to RF module when the time and date meets the schedule. |
|  | Send the according signal to the RFTX module when the Scheduler found an event that is time to change the state of some appliance. |
| MCIIR022 | Save the Schedule in database |
|  | Save all the Schedules in the Scheduler into database in case the main controller lose power or reboot. |
| MCIIR023 | Query the database for a specific IR pattern. |
|  | when the RFTX module receives a command to transmit a RF signal to the IR arduino, RFTX has to first query the database for the IR pattern, and decode the IR pattern into String object, then send it out to the arduino module. |
|  |  |