



# autoHome Report #1

Group #1

Paul Kania

Elie Rosen

Calvin Chiu

Rohith Dronadula

Elvison Dominguez

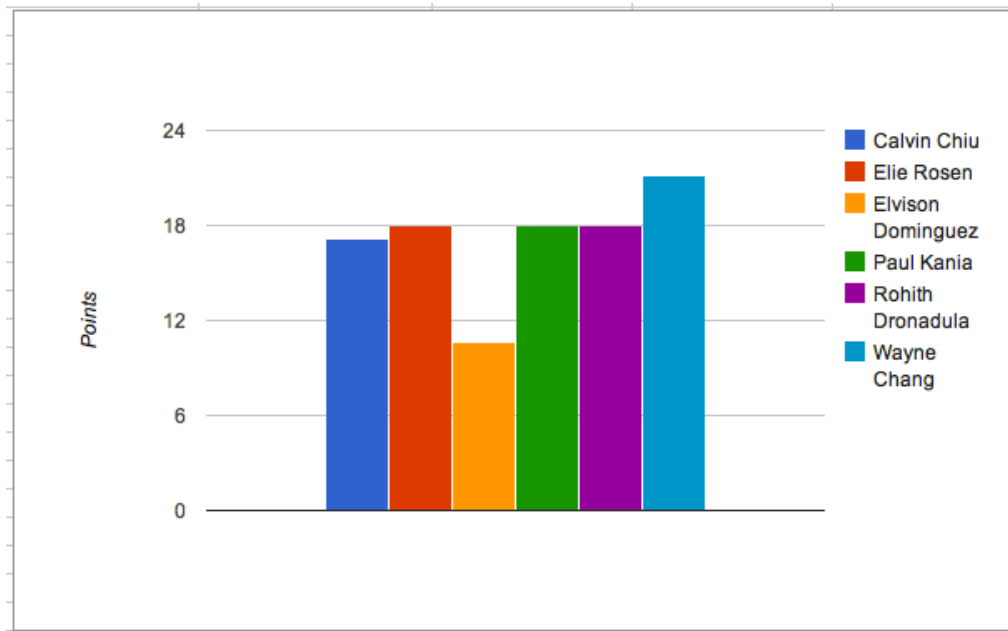
Wayne Chang

**<http://autohome.mylifeiscomputers.net>**

**<https://github.com/autohomeproject>**

# Contribution Breakdown

Responsibilities	Calvin Chiu	Elie Rosen	Elvison Dominguez	Paul Kania	Rohith Dronadula	Wayne Chang	Totals	Possible Points
Project Management		50%		50%			100%	10
Sec. 1 CSR		50%		50%			100%	6
Sec. 1 Term Glossary		20%	20%	20%		40%	100%	3
Sec. 2 SR A		75%				25%	100%	2
Sec. 2 SR B			60%	40%			100%	4
Sec. 2 SR C	N/A	N/A	N/A	N/A	N/A	N/A		0
Sec. 3 FRS A		25%		75%			100%	4
Sec. 3 FRS B	70%	10%		20%			100%	4
Sec. 3 FRS C	90%		10%				100%	16
Sec. 3 FRS D			100%				100%	6
Sec. 4 UIS A					100%		100%	7.5
Sec. 4 UIS B					100%		100%	7.5
Sec. 5 DA A						100%	100%	15
Sec. 5 DA B					50%	50%	100%	6
Sec. 5 DA C		100%					100%	4
Sec. 6 Plan of Work		50%		20%		30%	100%	5
Totals:	17.2	18	10.6	18	18	21.2	100%	100



# Table of Contents

1.	Cover Page.....	0
2.	Contributions Breakdown.....	X
3.	Glossary of Terms.....	4
4.	System Requirements.....	8
	1. Enumerated Functional Requirements.....	8
	2. Enumerated Nonfunctional Requirements.....	9
5.	Functional Requirements Specification	
	1. Stakeholders.....	12
	2. Actors and Goals.....	12
	3. Use Cases.....	13
	4. System Sequence Diagrams.....	37
6.	User Interface Specification	
	1. Preliminary Design.....	38
	2. User Effort Estimation.....	38
7.	Domain Analysis	
	1. Domain Model.....	44
	2. System Operation Contracts.....	48
	3. Mathematical Model.....	49
8.	Plan of Work.....	50
9.	References.....	52

# Customer Statement of Requirements

To our loyal followers:

The day has finally come, it is time to prepare yourselves for a home automation revolution.

After hearing about all of the pit falls that come with various home automation solutions available on the market, we knew it was time for us to step in.

Welcome to Advanced Utility to Operate Home, also known as autoHome. autoHome plans to tackle common everyday household problems by implementing a unique set of sensors and controls to fully maintain and secure your home along with adding more modern conveniences that would further enable you to become more relaxed with your ever-growing stressful lifestyle.



Our product aims to provide a more simplified lifestyle for the average homeowner in the confines of their home. We understand the frustration users have when attempting to use our competitors products and we are here to revolutionize and simplify the realm of Home Automation.

After extensive testing of our competitors products we have come to find that the basis of most users troubles come from the very complicated user interfaces that make it extremely hard to achieve completing tasks as simple as changing the temperature in your home. Instead, autoHome corrects this problem by implementing a centralized online dashboard, the user will be able to control their entire home from their computer or smart phone or even thousands of miles away. Each home that uses our product will be configured to communicate with our dashboard by the use of sensors and automated appliances. This would be useful for people who want to have the most control possible over their home as possible and certainly helpful for people with young children, or simply anyone who wants to have an easier, more enjoyable, living environment.

autoHome all starts with the dashboard. autoHome's dashboard is where you will find a panel of all rooms in the home as well as all devices in their respective rooms. From the dashboard, you will also see water usage and energy usage on a number of different scales (hourly, daily, weekly, and monthly). You will also see the current temperature in the home, with the option to change it to your desired temperature. There will also be

a display showing all appliances, devices, and lights in the home and their current status (on/off), also with the ability to turn them either on or off. All alarms and sensors will also be shown with their current status. There will also be a calendar module that will show scheduled events for the day and will also have your personal calendar shown with any obligations you have for the day. Overall, the dashboard will have an aesthetically pleasing look to it, while remaining simplified and also having a number of key features on it which are mentioned above. Each of these features is explained in detail in the following paragraphs.

Different sensors will be installed throughout the home to collect various data on the status of different elements in your home. The data is then sent to the central server and is analyzed to determine actions and triggers. These triggers are extremely customizable by the user, but autoHome also comes with a default set of rules easy to implement by the user. All events are saved to a log file, (which can be used for homeowners insurance or local law enforcement for their respective uses), which will be available to the user so that they may audit their home and make adjustments to further enhance their comfort and luxury. autoHome will also maintain a default profile that is not publicly available on the site, but will be used to recommend the user the most comfort available, while at the same time, saving the customer money by using energy wisely and effectively. Not only will you be saving money on your electricity bill, but you can take that money and spend it on a new appliance or gadget for your home! This is yet another feature not found in any of our competitors' home automation solutions.

In order to help you save energy, autoHome will be able to control the window blinds in your home. By having control of the blinds, autoHome will be able to save energy by allowing sunlight to come through the windows to serve as natural lighting. autoHome will keep track of sunrise and sunset data and adjust the blinds accordingly to provide optimal lighting in the home throughout the day. Of course, this can all be modified to your liking, such as when to shut the blinds if you decide you want them to shut at a specified time, or if you do not have work on a certain day and do not want to be potentially woken up by sunlight at sunrise that morning. Overall, this is another one of many features that will help you save money on your electricity bill each month using autoHome.

Climate control is a feature that is found in nearly every competitors' home automation solution. Obviously, we've included the feature in our autoHome system, but we've also implemented a clever feature to help control the temperature that the user sets in the home. We plan to implement sliding doors wherever possible in the home to make sure that the door is immediately closed right after a person enters or leaves through it. In ordinary homes, people turn on their air conditioning to cool the home, but then leave the doors open accidentally and make the home warmer. autoHome aims to eliminate this problem with our sliding doors.

One of the biggest features that we saw demand for were the use of motion sensors. Installing motion sensors in the home provides for a number of different features to function properly. First is the home security portion of functionality. autoHome can be armed and disarmed according to if a person is home or not. If our sensors detect motion when nobody is home, you will be notified by means of text messaging, email, or even a phone call. This, alone, makes our system more accessible than anything on the current market. In conjunction with motion, sensors will be wired to monitor the status of fire detectors, carbon monoxide detectors and any other detectors you may use in your home. For instance, if autoHome detects that any of

these alarms are going off while you are not home, namely the fire alarm, not only will you be notified by text message, email, or phone call, but local authorities will be notified at the same time to prevent possible excessive damage to your home. Similarly, if the motion sensors detect motion in your home when nobody is home, you will be notified as well as local authorities to thwart any criminal activity in your home.

In addition to monitoring electricity and our intelligent use of motion sensors, we will be doing some very interesting things with regards to water in your home. First off, autoHome will also monitor the water consumption in the home. Water usage will be an estimation, but will give the user a very good idea of how much water is used, what uses the most water, and what may even be wasting water in your home. Ultimately the user will have an overview of all the resource consumption in their home displayed conveniently on the web panel. Just like electricity monitoring, autoHome can give advice on how to adjust your water consumption to save money on your water and sewer bills. More importantly with regards to water is how autoHome reacts to potential flooding in your home. autoHome's sensors will also be able to determine if water is coming into your home. When it detects water coming into your home, autoHome will intelligently shut off all electrical devices that can be damaged by the incoming water.

Another interesting feature that we're implementing is the option to initialize devices. By this we mean that say you're in a situation where: you're at work until 5 PM, your wife calls and the two of you decide to watch a movie (streaming from Netflix) at 7 PM after dinner. You can simply log on to the autoHome dashboard from work, and set the TV, as well as your Xbox 360, to turn on at 6:55 PM, so that when you and your wife go to watch the movie, everything is set up, ready to go. Instead of waiting for devices to turn on and boot up, devices will now be waiting for YOU! Additionally, all devices and appliances in the home can be scheduled to turn on or off at certain times, as you adjust them to. This proves to be especially useful when you need to turn an appliance on or off at a certain time when you are not home. On top of the scheduled controls feature, we plan to integrate inductive charging "spaces" around the home in order to easily charge your mobile devices anywhere around the home. Adapters will be made for a wide range of devices so that you can easily make sure that your device is compatible with the inductive charging spaces found around the home.

If it is not obvious enough yet, our system is unlike anything available on the market. We looked at what is available on the market, as well as your feedback that you made available online, and we designed autoHome. autoHome morphs a home automation system with a home security system as well as an energy-saving monitor. This is a very unique product entering an already flooded market and we hope you are a part of it!

We truly hope that you are as excited for autoHome as we are. We enjoyed developing it just as much as you'll enjoy using it on a daily basis. On behalf of autoHome, we would like to welcome you to the home automation revolution!

Thanks for sticking with us,

Your friends at autoHome

# Glossary of Terms

Action	An arbitrary setting of a single sensor value.
Action Set	A group of actions to be executed at the same time.
Analog Controller	A controller setting fluctuating data values. (Examined in the Domain Model)
Analog Sensor	A sensor providing fluctuating data values. (Examined in the Domain Model)
autoHome	Needs to be filled in
autoHome Device	Any component that can be interfaced with the autoHome system through a Device Interface Array and scaled with a Device Interface Module.
Condition	A sensor value comparison that can be held true or false.
Condition Set	A group of conditions that must all be met to evaluate as true.
Controller Conductor	A conductor (telephony-grade wire) with voltages controlled on the system side.
Dashboard	A graphical web interface that allows users to view device information and set up rule sets to control devices.
Default Device Value	A value provided by the Device Interface Module to set Controlling Conductors to in the event of no active Action on an autoHome Device.

## Device Interface Array

A physical device that provides many Controlling, Reporting, and Ground Conductors to link with autoHome devices. It aggregates the voltage values of the controlling and reporting conductors into a single connector to the autoHome System.

The Device Interface Array can aggregate data streams into a physical or virtual communications port. At the time of this writing, only USB ports are supported. There are prototypes to allow virtual ports over Ethernet, and these will be implemented in the future.

All Device Interface Arrays use a hardware driver specific to the type of connection used to interface with the system.

## Device Interface Module

A driver-like piece of software providing reasonable scaling and unit conversions from voltages. A Device Interface Module might convert the voltage from a Reporting Conductor for a refrigerator from voltages of 2.0V to 5.0V to temperatures of 0C to 100C.

It might also convert a logical off or on for a lightbulb to a functional 0.0V or 3.0V actually sent through the Controlling Conductor.

The Device Interface Module requires close interaction with a Device Interface Array through a hardware driver.

## Digital Controller

A controller setting binary data values. (Examined in the Domain Model)

## Digital Sensor

A sensor providing binary data values. (Examined in the Domain Model)

## FURPS+

Acronym that stands for Functionality, Usability, Reliability, and Performance. "+" stands for various requirements such as design, implementation, interface and physical requirements. This is a standard method that is used to describe and classify the non-functional requirements.

## Ground Conductor

A conductor (telephony-grade wire) to ensure a common ground.



Home Automation	Automation of the home, housework or household activity. Home automation may include centralized control of lighting, HVAC (heating, ventilation and air conditioning), appliances, and other systems, to provide improved convenience, comfort, energy efficiency and security.
MySQL	World's most used relational database management system that runs as a server providing multi-user access to a number of databases.
PHP	A hypertext pre-processor. It's an open source, server-side, cross-platform, scripting language used to create dynamic web pages.
Plug & Play	Refers to the ability of a computer system to automatically configure expansion boards and other devices. You should be able to plug in a device and play with it, without worrying about setting DIP switches, jumpers, and other configuration elements.
Reporting Conductor	A conductor (telephony-grade wire) with voltages controlled on the device side.
Rule Set	A relationship between a Condition Set and an Action Set that may be enabled or disabled.

## System Interface Module

A driver-like piece of software acting almost identically to a Device Interface Module, with a readings and controls linked to the computer system and Internet rather than physical devices.

A System Interface Module might have sensor report the value of the weather in a certain zip code.

A different System Interface Module might be controlled to send a text message selecting a text messaging profile (phone number and message) with the controller value.

System Interface Modules will be configured from within the web interface. Each System Interface Module will provide its own configuration web page. An example weather-reporting System Interface Module might have a web configuration page containing a zip code field and a units drop-down box containing Fahrenheit and Celsius. The autoHome System can then call the sensor to receive an analog value of the temperature in the specified units from the Internet.

## Use Case

A usage scenario for our system which shows how a user communicates with the system to get their desired response.

## XSS

A group of interrelated web development techniques used on the client-side to create asynchronous web applications.

# System Requirements

## Enumerated Functional Requirements

Req-x	PW	Description
REQ-1	10	The system shall have a robust authentication and permissions system as to prevent attackers from entering the system, and to prevent normal users from accessing unauthorized devices.
REQ-2	10	The system shall record values from its sensors in a database at specifiable rates, and have the ability to clearly display these values to users.
REQ-3	10	The system shall control devices according a set of user-specified rules.
REQ-4	10	The system shall be redundant and have many precautions in place, allowing swift backups and system recoveries in times of crises.
REQ-5	7	The system shall have a very accessible Graphical User Interface used to allow flexible control of the entire system.
REQ-6	6	The system shall include sufficient System Modules to be able to effectively communicate via text messaging, phone calls, or emails.
REQ-7	5	The system shall allow administrators to add new devices and remove existing devices with ease.
REQ-8	3	The system shall have sufficient System Modules as to be able to query external data from the Internet to use in Rule Sets.

# Enumerated Nonfunctional Requirements

With autoHome, we use the FURPS+ model [1] to describe the nonfunctional requirements as follows:

## **Functionality**

See the Functional Requirements section for details on the functionality of autoHome.

## **Usability**

The autoHome dashboard should be simplified and easy to maneuver, while also being easy on the eyes. This means that the dashboard should have an aesthetically pleasing look to it. This would allow it to be accessible to a wide range of consumers. The dashboard would have a panel with a list of all the rooms in the main menu, with a sub-menu of all the appliances and functions to each specific room. autoHome will support a simple “plug and play” system with all appliances in the home so that they will be easily added to the dashboard when they are installed in the home. The rest of the system should also be designed as intelligently by implementing other features that make the experience the easiest possible for the user.

## **Reliability**

### Data Safety

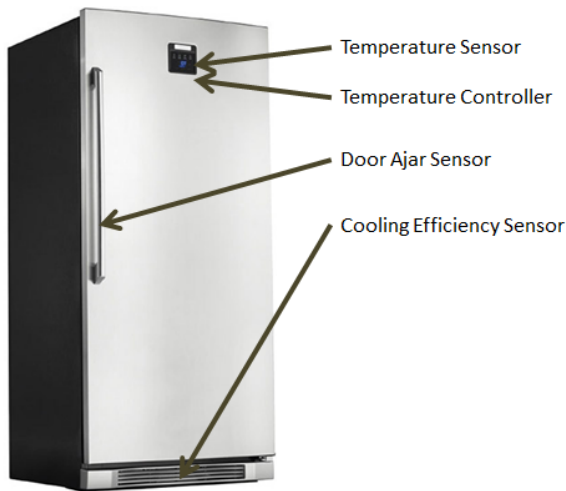
autoHome will create a routine backup in the case of system failure. The backups will be database dumps compatible with many other home automation systems after data manipulation. Backups will be made on a user-specified basis. All user settings, recorded data, and stored rulesets will be backed up. autoHome data and configuration only lives in the database, so a database dump is all that is required. An identical autoHome central server may be installed in the case of another autoHome central server's demise. The reconfiguration would be as simple as loading a database file. The autoHome central server will be installed on an enterprise-grade Uninterrupted Power Supply. This will provide the system long enough time to realize the power loss and back up all data before a potential power failure. The backup system will also be stored to a cloud based storage system. [3]



### Device Resilience

Complicated devices are considered as a bundle of lesser devices to the central system. An automated refrigerator may have multiple sensors such as refrigerator temperature, freezer temperature, refrigerator door-open, and freezer door-open sensors. These are all considered different devices by the system as to provide

a redundant set of controls such that if one sensor malfunctions, the other sensors may behave normally as they are treated as entirely separate devices.



It was considered to add individual devices (with multiple sensor values) to the system as wholes, but then decided against with the realization that it would greatly decrease the modularity and reliability of the system. Instead, individual sensors or controllers may be grouped logically to form the logical association of a whole complex device.

### **Performance**

autoHome should be a very streamlined experience. autoHome system hardware will be tested to work proficiently with its software in order to achieve an optimal experience. Thus, autoHome should perform smoothly and on time with the consumers' inputs. autoHome should be able to recognize all of the appliances in the home by means of the dashboard.

### **Supportability**

Technical support should be available on our website via live chat, email, or phone call. autoHome should have commands that can be processed by either a smartphone, the autoHome website, or the system itself at home. autoHome should have a reboot system with a security code that can be accessed through technical support, in case the system malfunctions. autoHome should receive updates that should be "pushed" to the system through a wireless network to ensure that potential bugs are fixed.

### **± (Other criteria)**

#### **Security**

autoHome should have passwords for each individual that uses the system. After keying in multiple invalid password entries, autoHome should lock up until the system generates an "override password" that will be sent to the user by an email or text notification.

### **Implementation**

autoHome's Dashboard interface will be coded in PHP, MySQL, AJAX and jQuery [4]. autoHome Embedded Devices will use an Arduino [5] coded in C.

### **Operation**

autoHome will be able to detect which appliance is attached to the system with the "plug and play" system. autoHome will have Climate control, inductive charging, ambient lighting and window blind control, bathroom automation, automatic sliding doors, scheduled controls, energy and water usage monitor, fire/CO alarms, and appliance sensors such as an oven/stove. autoHome will also be able to detect home intrusion and alert the police with an alarm system, and it will alert the family members with an alarm and via text or email. For a detailed outline of operations, please refer to the "Statement of Requirements" section.

### **Packaging**

Ideally, autoHome should be installed in new homes. However, autoHome can be installed into previously existing homes, but the same functionality is not guaranteed. In previously existing homes, autoHome will require many holes in the walls for the wiring of the system. Since some previously existing homes may be older than others, the overall conditions may not be optimal for the installation of autoHome. Connections and wiring would be handled by the professional installer of autoHome. The consumer would simply deal with the autoHome dashboard, which is detailed in the "Usability" section above.

**Legal**

The programming, design and ideas of this product will be provided by our own team. A license of this product will be obtained when autoHome is completed and ready to sell as a market product.

# Functional Requirements Specification

## Stakeholders [2]

10. Internal Stakeholders
  1. Developers (Internal): A person or group of people who codes and programs for autoHome and has rights to access all parts of it. Internal developers' interests are advanced functionality, ease of use, and affordability.
11. External Stakeholders
  1. Developers (External): A person or group of people who want to utilize modules of autoHome for their personal use and/or business. External developers' interests are increasing profitability of consumer products and better relations with consumers.
  2. Homeowners (Users): People who have our system installed in their home. Homeowners' interests are quality, value, service and reliability.
  3. Competitors: Companies that provide very similar products to sell on the market to the general public. Competitors' interests are demand, profit, and customers.

## Actors and Goals

### User

- Initiating type
- Goals: To control their home based on their specifications, allowing for a safer and more user-controlled home environment.

### User Web Interface

- Participating Type
- Goals: To accept user configurations to the autoHome system, and display graphs and data values according to logged information.

### autoHome System

- Participating/Initiating Type
- Goals: To collect various data from sensors, analyze data and issue instructions to devices.

### autoHome Database

- Participating Type
- Goals: Store device data logs, user information, and device configurations.

### Device Sensors

- Initiating type
- Goals: To constantly report meaningful binary or fluctuating values to the central server.

### Device Controllers

- Participating type
- Goals: To process binary or fluctuating data values from the central server and control devices.

### System Modules

- Participating type
- Goals: To interface the autoHome system to communications and systems based technologies such as text-messaging, phone calls, system time, system status checks.

# Use Cases

## I. Casual Descriptions

### **UC#1 Authenticate User**

A user wants to log into the system to modify system preferences, add or remove devices, configure schedules or request information. The user must provide the credentials of a valid user account in the autoHome system. The user is prompted to create an administrator-level account upon first boot of the system. An invalid entry shall result in an error notification and logging by the system, and user shall be prompted again for password.

### **UC#2 Create User Account**

An authenticated administrator or initiating user wishes to create additional user accounts. User accounts may only be created without authentication on the first-time system start-up (hence initiating user), in which the first user account may be created with administrator-level privileges. The account creation process will prompt for valid first and last names, a username, a password, and privilege level (user or administrator).

### **UC#3 Change User Account Settings**

An authenticated administrator wishes to change user account settings. The ability to change user account settings is only granted to administrators as to prevent abuse (falsification of names or data). The administrator may change any fields populated at account creation, or delete the user account.

### **UC#4 Configure User Device Permissions**

An authenticated administrator may change the device permissions of a user. Device permissions give or restrict access to individual sensors and controllers to devices. Permission to a device allows a user to use that device in rule sets. Administrators have access to all devices.

### **UC#5 Configure Device Rule Sets**

An authenticated user wants to configure condition sets and related action sets. Users can schedule certain times to enable devices, set up sensor value-dependent actions, or utilize the abilities of systems modules. Examples include locking doors during the night, turning on lights as light sensors detect low luminance, or sending a text message to a phone number if a motion sensor is activated on a certain date and time.

### **UC#6 Display Device Information**

An authenticated user wants to geographically view authorized devices on a floor map and their data values as either text or graphs. The system will poll device map databases for locations, and logs for data values. The values are then pretty-printed or used to generate graphs. Additional calculations will be optionally displayed as well to show total usages, relationships, and trends.

### **UC#7 Export Device Log Data**

An authenticated user wishes to export data values for authorized devices from the system logs. The data is outputted to the user's browser in a MIME-typed common database format.

### **UC#8 Install Device Interface Modules**

An authenticated administrator wishes to update the system to support new devices. The administrator can upload a vendor-provided module to normalize raw sensor and controller data into sensible values and units.



### **UC#9 Uninstall Device Interface Modules**

An authenticated administrator wishes to remove an interface module for a device. The module will be removed and the system will no longer be able to interface with devices using the module and will automatically remove all such devices.

### **UC#10 Add Interface Array**

An authenticated administrator wishes to add an additional physical or virtual set of device slots for controlling and reading device values. The system will attempt to auto-detect the newly inserted physical or virtual interface arrays and request more information upon failure.

### **UC#11 Remove Interface Array**

An authenticated administrator may remove a device interface array in case of malfunction or disuse of the array. All devices on the array are automatically removed during this use case.

### **UC#12 Add Device**

An authenticated administrator may add a device to the system. The administrator is responsible for physical installation of the device and a proper connection to a valid interface array. The administrator is also responsible for installing the correct vendor-provided interfacing modules. The administrator shall specify the correct port of the device interface array and device type from drop-down style lists. The correct database entries to utilize the device will be made.

### **UC#13 Remove Device**

An authenticated administrator may remove a device from the system. All rule sets, conditions, actions, condition sets, and action sets directly or indirectly referencing the device will be removed. This is to prevent security issues with rule sets not meeting full conditions before executing actions. Administrators are advised to refactor all condition and action sets prior to device removal. The device database entries are removed and the device can no longer be interfaced with.

### **UC#14 Send Device Signal**

The system specifies values to set the connected devices to. The device interfacing modules convert these values to meaningful voltages to set the controlling conductors to. When there are no device settings, the interfacing modules provide default values to hold the controllers at.

### **UC#15 Receive Device Signal**

The system receives values from the connected devices. The device interfacing modules convert the voltages from the devices into meaningful values to the system. Values are polled at rates specified by the device interfacing modules. All values are evaluated against relevant rule sets upon change.

### **UC#16 Add System Interface Module**

An authorized administrator may add a system interface module to the autoHome system. These modules provide ways to interact with the computer system and the Internet as though as they were devices.

### **UC#17 Remove System Interface Module**

An authorized administrator may remove a system interface module. All devices using the Interface Module will be removed.

### **UC#18 Send System Interface Module Signal**

Set an active value to the system to use as a controller value in rule sets. Such values may include text message profiles IDs to text messaging modules, email profile IDs to emailing modules, and shell-script profile IDs to computer-control modules. Values and interfacing are handled by the device interfacing modules.

**UC#19 Receive System Interface Module Signal**

Retrieve a value from the system to use as a sensor value in rule sets. Such values may include system time, system date, and information (such as weather) from over the Internet. Polling rates and interfacing are handled by the device interfacing modules. All values are evaluated against relevant rule sets upon change.

**UC#20 Evaluate Data Against Rule Sets**

When new data is received, it is checked against the relevant active rule sets that contain the data value as a condition in its condition set. If the condition set is met (all conditions inside the set are fulfilled), then the associated action set will be executed.

**UC#21 Execute Action Set**

Executing an action set will actively set all devices to specified values until the associated condition set is no longer met (one or more conditions are unfulfilled). The device controllers are then set at their default values.

**UC#22 Backup Database and Device Interface Modules**

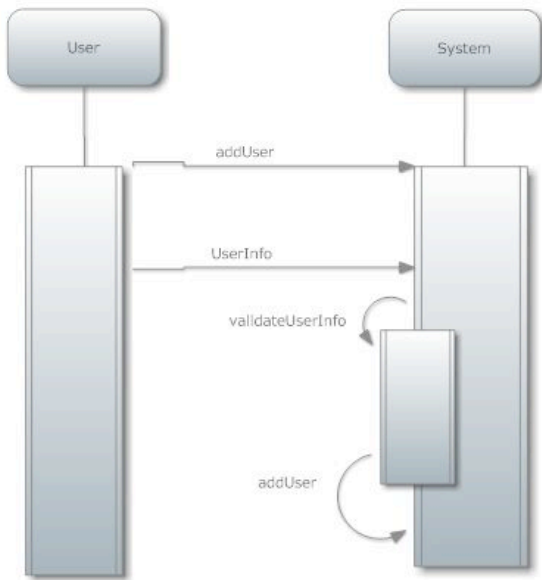
The autoHome Database and Device Interface Modules will be periodically dumped and stored either on server, on removable storage, or through the network at an off-site location. The database and Device Interface Modules store the entirety of the configuration data for autoHome, and are all that are required for complete system recovery. An administrator may also manually start a backup to a specified location.

**UC#23 Restore Database and Device Interface Modules**

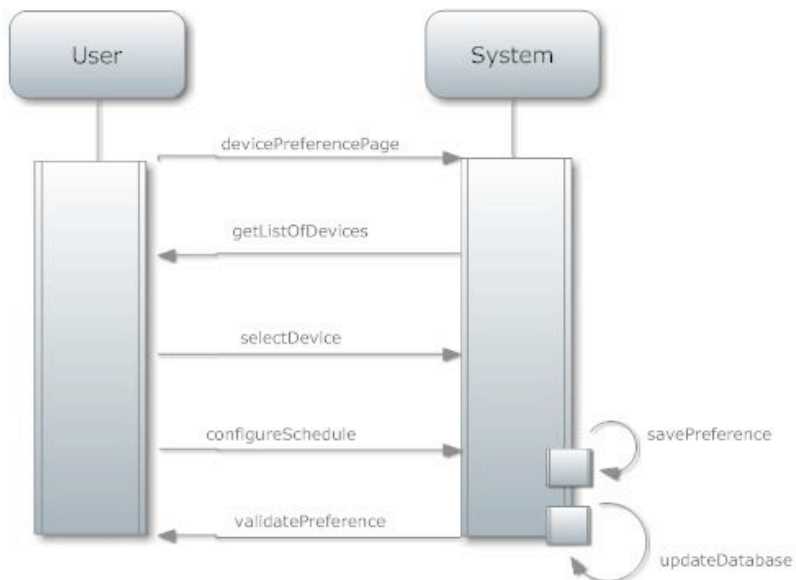
An authorized administrator may restore the autoHome system to a previous state. The autoHome Database contents will be dropped and loaded with a verified backup. The Device Interface Modules will be erased and replaced with the ones in the backup.

## II. Use Case Diagrams

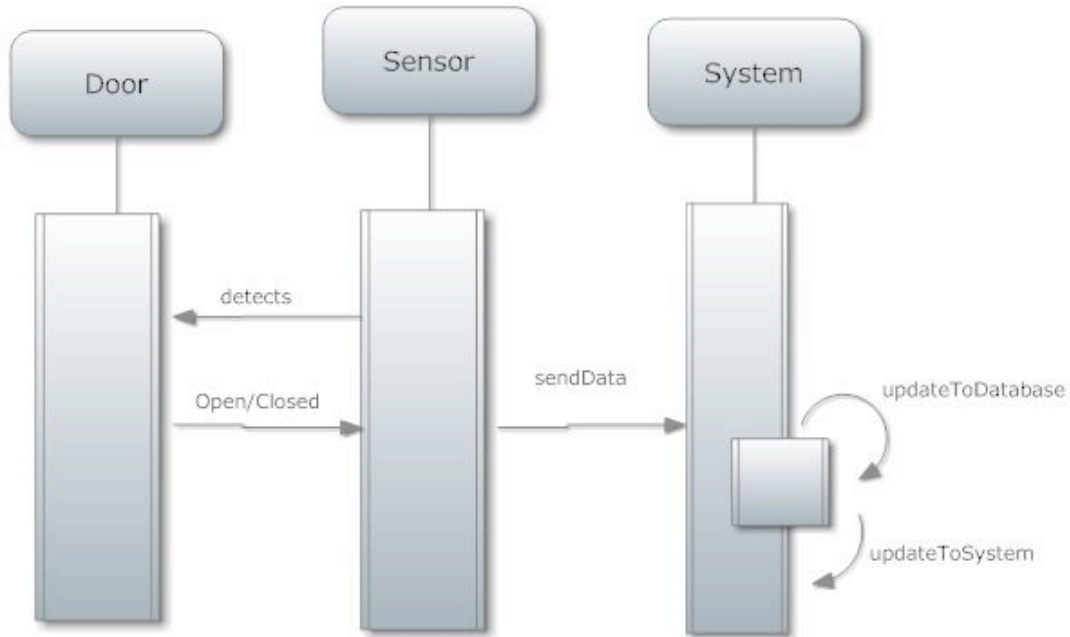
AddUser Sequence Diagram



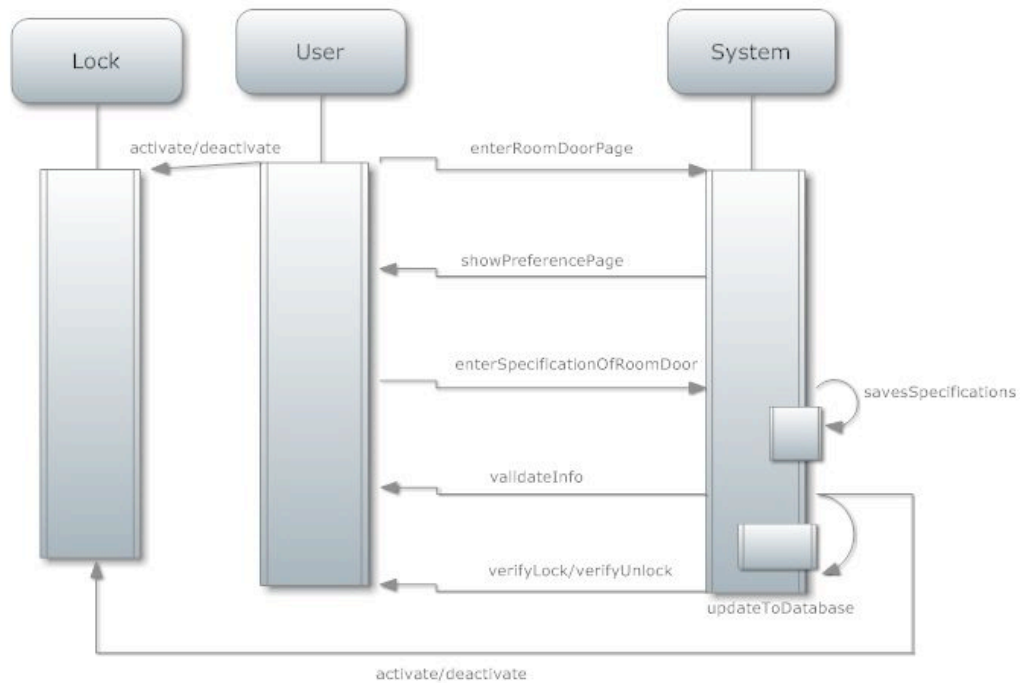
Configure Device Schedule Sequence Diagram



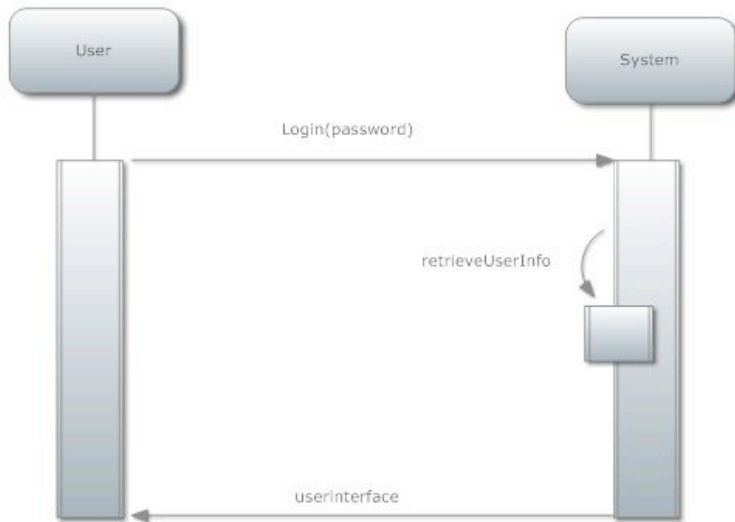
Detect Door Status Sequence Diagram



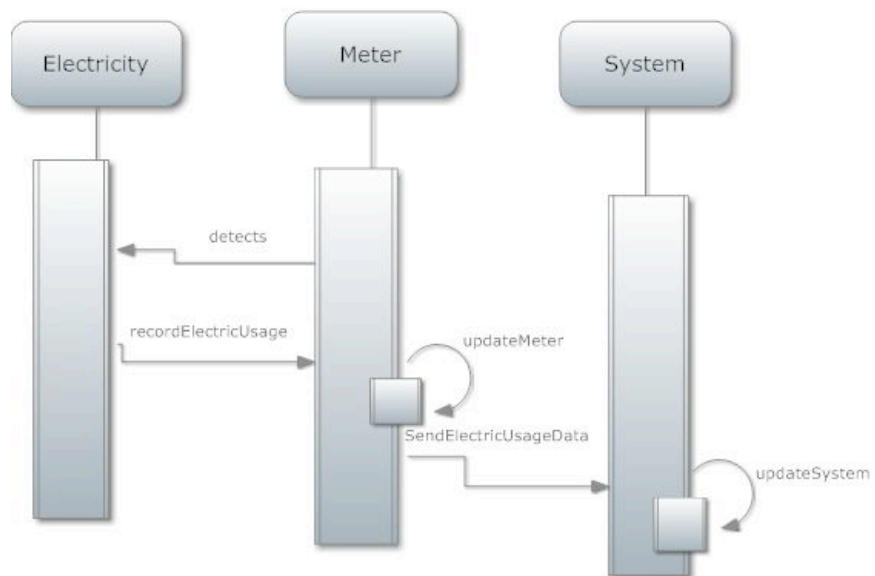
Lock Sequence Diagram



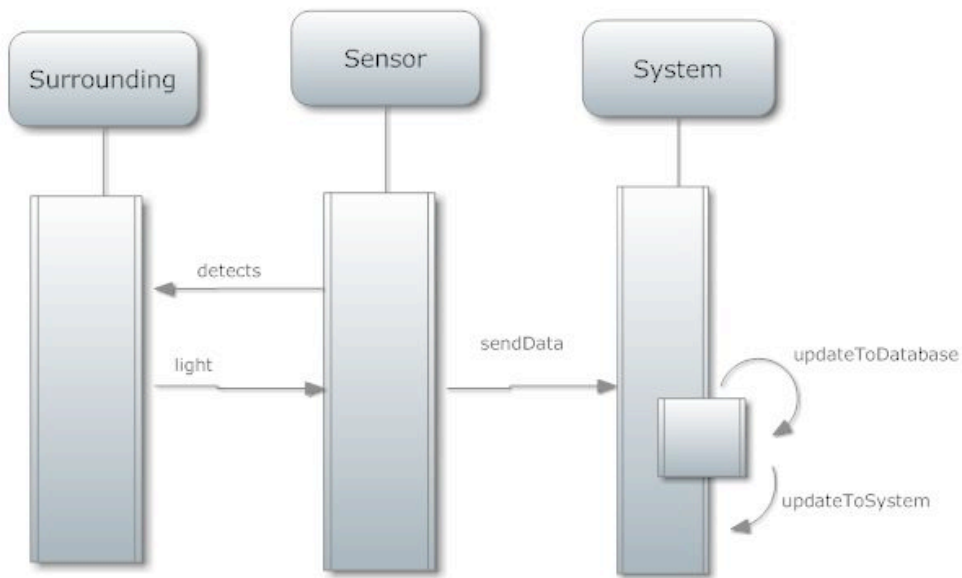
Login Sequence Diagram



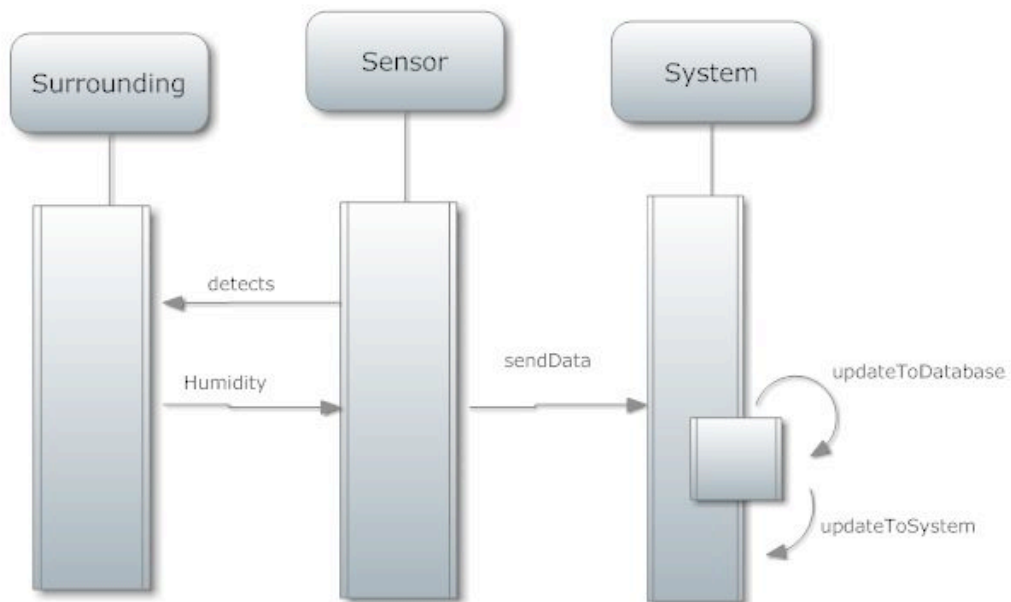
Measure Electricity Usage Sequence Diagram



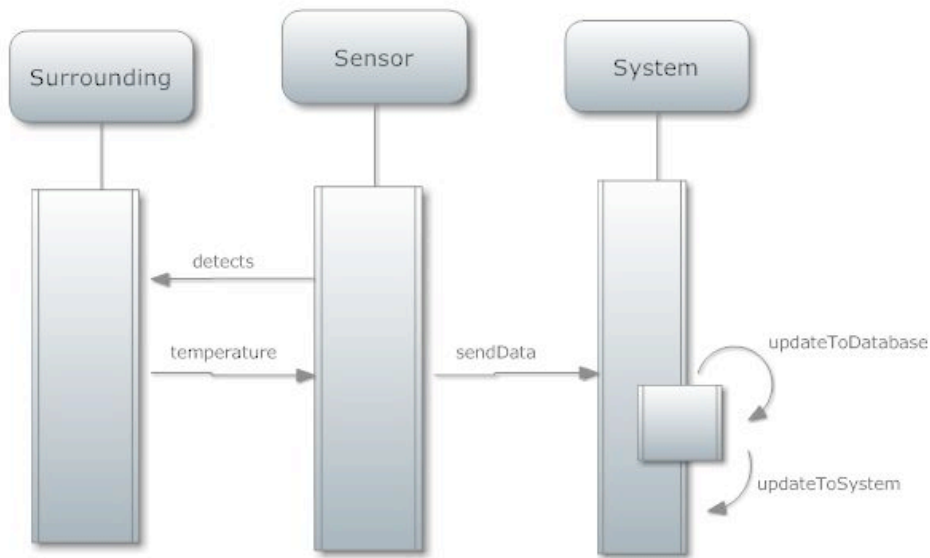
Measure Light Intensity Sequence Diagram



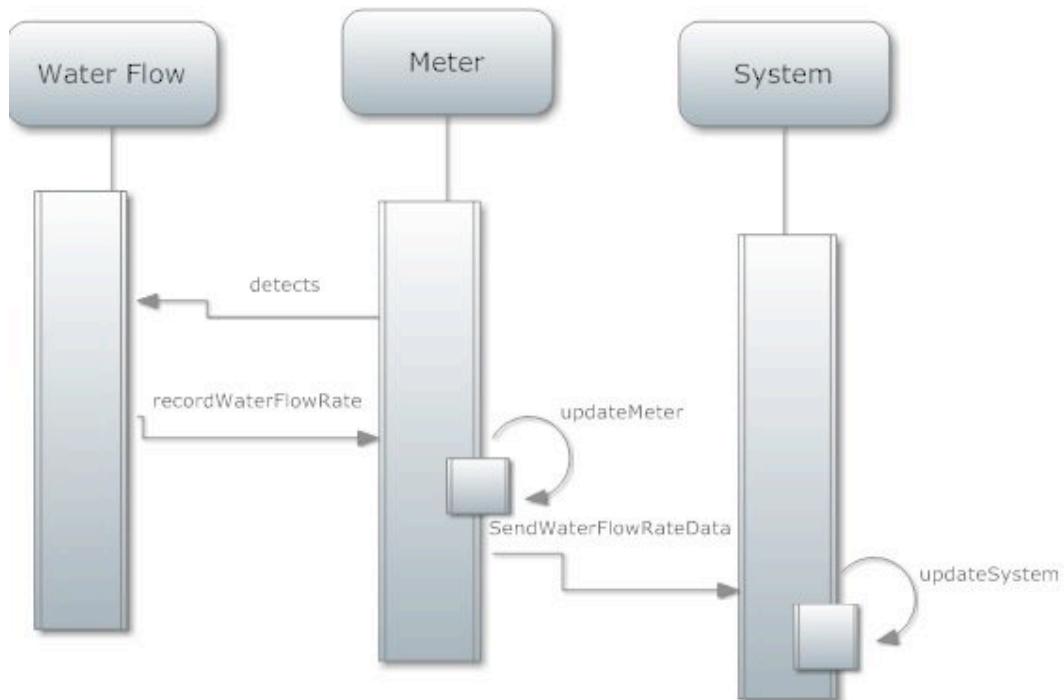
Measure Humidity Sequence Diagram



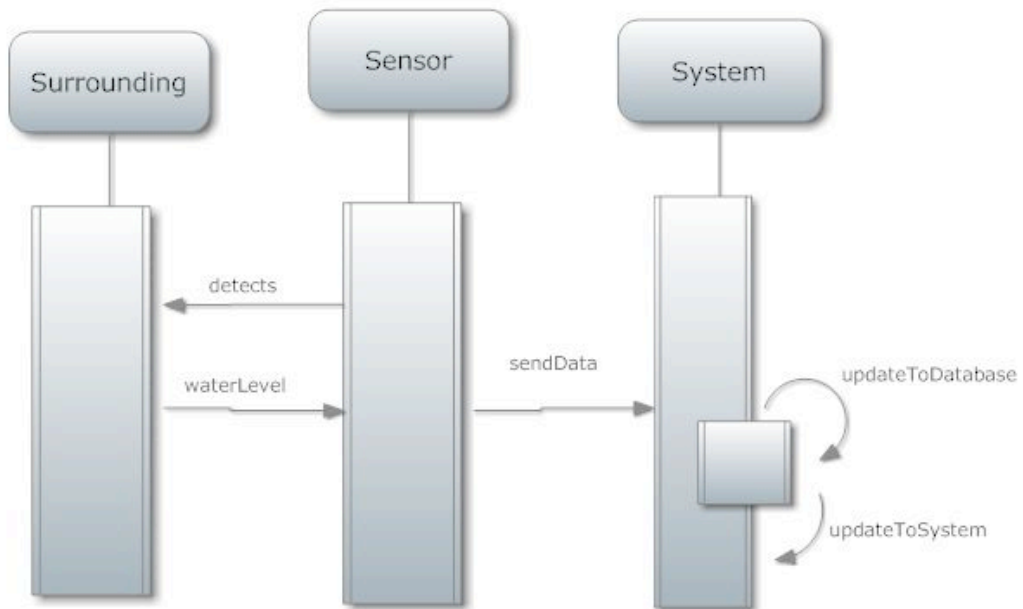
Measure Temperature Sequence Diagram



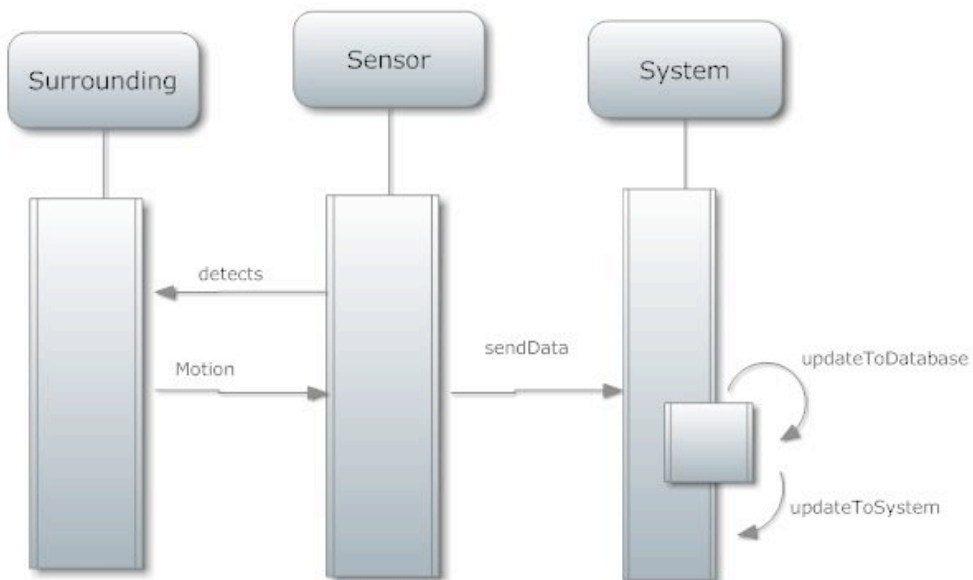
Measure Water Flow Sequence Diagram



Measure Water Level Sequence Diagram

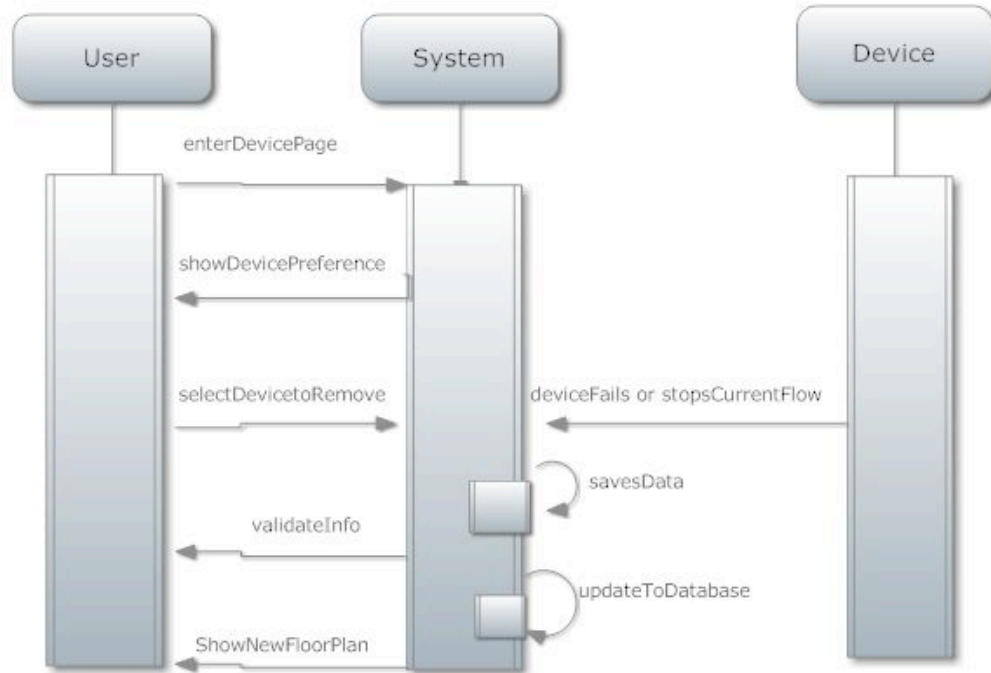


Motion Sensor Sequence Diagram





Remove Device Sequence Diagram



### III. Fully-Dressed Descriptions

UC-#1: Authenticate User		
Related Requirements	REQ-1, REQ5	
Goal In Context	Administrator attempts to log into the central control system and configure the preference or requests information	
Preconditions	System is activated and working	
Successful End Condition	Administrator inputs correct password using the factory default password and the interface is provided to the user	
Failed End Condition	Administrator inputs incorrect password and the login interface is provided to the user	
Actors	Administrator, Server	
Trigger	Administrator enters password number on the login screen	
Main Flow	Step	Action
	1	Administrator enters password and attempts to login
	2	System reads the password entered by the administrator authenticates with the password. Then the system provides the administrator with an interface that allows administrator to perform system configuration or request information.
Extensions	Step	Action
	2.1	Administrator enters invalid password. System notifies the administrator and returns to log-in screen
	2.2	Administrator verifies his password and enters the password again.

UC-#2: Create User Account		
Related Requirements	REQ-1, REQ-5	
Goal In Context	Administrator creates user account for the system after first time initiation	
Preconditions	First-time system initiation has not been done and administrator has logged into the system using factory default password	
Successful End Condition	Administrator creates a new account and a new password is set up	
Failed End Condition	Administrator does not create a new account nor a new password	
Actors	Administrator, System	
Trigger	Administrator starts up the system on the first time and logs in using factory default password	
Main Flow	Step	Action
	1	Administrator starts up the system on first time and logs in using factory default password
	2	System provides the administrator an account set up interface
	3	Administrator enters preferences and password and saves the configuration
	4	System validates the information and saves the new account to the database

	5	System provides the administrator an interface that allows him to perform system configuration or request information
Extensions	Step	Action
	4.1	System detects incorrect information entered by the administrator.
	4.2	System prompts the administrator about the incorrect information and stays on the new user account configuration screen with errors highlighted.
	4.3	Administrator corrects the error and saves it.

UC-#3 : Change User Account Settings		
Related Requirements	REQ-1, REQ-5	
Goal In Context	Administrator updates the account preferences	
Preconditions	A user account has been set up, the system is logged in	
Successful End Condition	Administrator successfully updates the account preferences	
Failed End Condition	Administrator does not update the account preferences	
Actors	Administrator, System	
Trigger	Administrator enters the account settings page	
Main Flow	Step	Action
	1	Administrator enters the account settings page
	2	Administrator enters new information and saves
	3	System validates the information and saves the new information to the database
	4	System returns the administrator to home screen
Extensions	Step	Action
	3.1	System detects incorrect information entered by the administrator.
	3.2	System prompts the administrator about the incorrect information and stays on the new user account configuration screen with errors highlighted.
	3.3	Administrator corrects the error and saves it.

UC-#4 : Configure User Device Permissions		
Related Requirements	REQ-3, REQ-5	
Goal In Context	Administrator changes device permission for a user	
Preconditions	System is logged in and devices are connected to the system	
Successful End Condition	Administrator changes device permission for a user	
Failed End Condition	Administrator does not change device permission for a user	
Actors	Administrator, System, Device	
Trigger	Administrator enters Device Preference page	

Main Flow	Step	Action
	1	Administrator enters Device Preference page and select a specific device
	2	System provides an interface that allows administrator to edit the device preference
	3	Administrator changes device permission and saves
	4	System validates the integrity of the configuration and updates the database.
	5	System returns the administrator the Device Preference page.
Extensions	Step	Action
	1.1	Administrator cannot find a specific device and proceed to UC#
	3.1	System detects incorrectness in the configuration
	3.2	System prompts the administrator about the incorrectness and stays on the same screen with errors highlighted
	3.3	Administrator corrects the errors and saves it.

UC-#5 : Configure Device Rule Sets		
Related Requirements	REQ-3, REQ-5	
Goal In Context	Administrator configures the device condition sets and related action sets	
Preconditions	System is logged in and devices are connected to the system	
Successful End Condition	Administrator configures the device rule sets	
Failed End Condition	Administrator does not configure the device rule sets	
Actors	Administrator, System, Device	
Trigger	Administrator enters Device Preference page	
Main Flow	Step	Action
	1	Administrator enters Device Preference page
	2	System provides an interface that allows administrator to edit the device preference
	3	Administrator configures the device condition sets and related action sets
	4	System verifies the integrity of the configuration and updates the database
Extensions	Step	Action
	4.1	System detects incorrect configuration entered by administrator
	4.2	System prompts the administrator and stays on the same screen with errors highlighted
	4.3	Administrator corrects the configuration and saves it

UC-#6 : Display Device Information		
Related Requirements	REQ-5	
Goal In Context	System displays various device information to the administrator	
Preconditions	System is logged in and devices are connected to the system	
Successful End Condition	System displays device information with respect to administrator's request. Information including device map, logs, usages, trends, relationships, in text form and graphs.	
Failed End Condition	System does not display device information to administrator nor the information is correct	
Actors	Administrator, System, Device	
Trigger	Administrator enters Device Information screen and request information	
Main Flow	Step	Action
	1	Administrator enters Device Information screen and request information
	2	System polls device map from database for locations and logs, as well as analyzed data from UC#18 and displays information to user in text form and graphs
Extensions	Step	Action
	2.1	Some device information is not shown
	2.2a	Administrator manually force-refreshes the system to detect newly connected device.
	2.2b	Administrator manually adds device to the system database
	2.3	System re-polls device map from updated database and display information

UC-#7 : Export Device Log Data		
Related Requirements	REQ-5, REQ-6	
Goal In Context	Administrator exports data values for specific device from the system log	
Preconditions	System is logged in and devices are connected	
Successful End Condition	Data for specific device is exported in MIME-type format	
Failed End Condition	Data is not exported	
Actors	Administrator, System, Device	
Trigger	Administrator requests system to export device data	
Main Flow	Step	Action
	1	Administrator requests system to export device data
	2	System polls data of the specific device and saves it in a MIME-typed common database format. Then the data is accessible on administrator's mobile device or PC's browser
Extensions	Step	
	2.1	System cannot find any data for the specific device
	2.2	System acknowledges the administrator that no data for that device is found.

UC-#8: Install Device Interface Modules		
Related Requirements	REQ-5, REQ-7	
Goal In Context	Administrator updates the system to support new devices	
Preconditions	System is logged in and some new devices are not recognized by the system	
Successful End Condition	Administrator updates the system to support new devices and system detects them	
Failed End Condition	Administrator does not update the system to support new device and thus the system cannot recognize the new devices	
Actors	Administrator, System, Device	
Trigger	Administrator enters Add/Remove Device Interface Modules page	
Main Flow	Step	Action
	1	Administrator enters Add/Remove Device Interface Modules page
	2	System provides an interface that allows administrator to choose to update the modules from internet, driver disc or USB
	3	Administrator chooses the media and updates the system database
	4	System displays a list of newly updated devices to the administrator
Extensions	Step	Action
	4.1	Some devices are still not missing after the update is completed
	4.2	Administrator requests system to reinstall the package
	4.3	System removes the package in UC#9 and reinstalls it back. Then a list of newly updated devices is displayed to the administrator

UC-#9: Uninstall Device Interface Modules		
Related Requirements	REQ-5, REQ-7	
Goal In Context	Administrator removes a device interface module	
Preconditions	System is logged in and devices are connected	
Successful End Condition	Administrator removes a device interface module from the database	
Failed End Condition	Administrator does not remove a device interface module from the database	
Actors	Administrator, System, Device	
Trigger	Administrator enters Add/Remove Device Interface Modules page	
Main Flow	Step	Action
	1	Administrator enters Add/Remove Device Interface Modules page
	2	System provides a list of devices for administrator to choose
	3	Administrator chooses a device and removes the interface module
	4	System refreshes the database and display the updated list
Extensions	Step	Action

	4.1	After the system refreshes the database the device is still shown on the list. System prompts administrator about unsuccessful removal
	4.2	Administrator checks if the corresponding device is still in use or not and stops it. Then he uninstalls the module again.

UC-#10 : Add Interface Array		
Related Requirements	REQ-5, REQ-7	
Goal In Context	Administrator adds a physical or virtual set of device slots for controlling and reading device values	
Preconditions	System is logged in	
Successful End Condition	Administrator adds a physical or virtual set of device slots	
Failed End Condition	Administrator does not add a physical or virtual set of device slots	
Actors	Administrator, System, Device	
Trigger	Administrator plugs in the physical device to the system or installs a virtual device like a software probe	
Main Flow	Step	Action
	1	Administrator plugs in the physical device to the system or installs a virtual device like a software probe
	2	System automatically detects the device and updates the database
	3	Administrator is notified upon successful installation
Extensions	Step	Action
	3.1	Upon failure, system requests the administrator to provide additional information for the interface array. System provides a interface configuration page for the administrator to edit the details
	3.2	Administrator enters the configuration and saves it
	3.3	System refreshes the database and re-detects the interface array. Then it updates the database.

UC-#11 : Remove Interface Array		
Related Requirements	REQ-5, REQ-7	
Goal In Context	Administrator removes a physical or virtual set of device slots	
Preconditions	Interface array is installed	
Successful End Condition	Administrator removes a physical or virtual set of device slots	
Failed End Condition	Administrator does not remove a physical or virtual set of device slots	
Actors	Administrator, System, Device	
Trigger	Array malfunction or misuse of the array is detected by the system	
Main Flow	Step	Action
	1	Array malfunction or misuse of the array is detected by the system
	2	System automatically stops the service and removes the interface array. Then it notifies the administrator about the removal

Extensions	Step	Action
	2.1	System does not stop the service nor remove the interface array. Then it notifies the administrator about this error and requests manual removal. It provides a interface for administrator to remove the array manually (software) or requests the administrator to physically remove the device.

UC-#12 : Add Device		
Related Requirements	REQ-5, REQ-7	
Goal In Context	Administrator adds a new device to the system and the system updates the device database	
Preconditions	A new device has just connected to the system and configured	
Successful End Condition	Administrator adds a new device to the system and the system updates the device database	
Failed End Condition	Administrator does not add a new device and the system does not update the device database	
Actors	User, Server, Device	
Trigger	Administrator connects a new device to a valid interface array and enters Add/Remove Device page	
Main Flow	Step	Action
	1	Administrator connects a new device to a valid interface array and enters Add/Remove Device page
	2	System checks the database for a corresponding device interface module
	3	System provides an interface for administrator to specify the correct port of the device interface array and device type
	4	Administrator enters the required information and saves it
	5	System updates the database and notifies administrator upon success
Extensions	Step	
	2.1	System cannot find the corresponding device interface module. Then it prompts the administrator to install the interface module in UC#8
	5.1	System cannot find the specified array port. Then it prompts the administrator to install a new array in UC#10

UC-#13 : Remove Device		
Related Requirements	REQ-5, REQ-7	
Goal In Context	Administrator removes a device from the system and its corresponding configurations	
Preconditions	The device is connected and configured	
Successful End Condition	Administrator removes a device from the system and its corresponding configurations such as rule sets, conditions, actions, condition sets, and action sets directly or indirectly referencing the device.	



Failed End Condition	Administrator does not remove a device nor the corresponding configurations are removed	
Actors	Administrator, System, Device	
Trigger	Administrator enters Add/Remove Device page	
Main Flow	Step	Action
	1	Administrator enters Add/Remove Device page
	2	System display a list of configured and connected device to administrator
	3	Administrator remove a device from the system
	4	System prompts the user to edit the configurations in UC#4 and UC#5 to prevent security issues with the residue rule sets and conditions
	5	User shall remove the device configuration in UC#4 and UC#5
	6	System updates the database after all the configurations are made
Extensions	Step	
	6.1	System prompts the administrator upon unsuccessful removal in UC#18 and requests the administrator to check if the device is still running
	6.2	Administrator checks the device and redo the process again

UC-#14 : Send Device Signal		
Related Requirements	REQ-2	
Goal In Context	System sends instruction via specific voltage to control devices	
Preconditions	Device is connected to the system	
Successful End Condition	System sends instruction to the device and the device responds accordingly	
Failed End Condition	System does not send instruction to the device nor the device responds accordingly	
Actors	System, Device, Administrator	
Trigger	UC#18 is successful	
Main Flow	Step	Action
	1	UC#18 is successful and system detects a need to alter the device status
	2	System sends instruction via a specific value and a device interfacing module convert the value into meaningful voltages to set the controlling conductors.
	3	The corresponding device is thus initiated with respect to the instruction and responds to the system in UC#15
Extensions	Step	
	3.1	The device is not initiated by the instruction and abnormality in UC#15 is detected by the system
	3.2a	The system attempts to send instruction again

	3.2b	Upon failure, administrator is notified in UC#18 and is requested to check the connection of the device
	3.3b	Administrator checks the device connection and acknowledges the system

UC-#15 : Receive Device Signal		
Related Requirements	REQ-2	
Goal In Context	System receives values from the connected devices and the values are converted into meaningful values by device interfacing modules	
Preconditions	Sensors are connected to the system	
Successful End Condition	System receives meaningful values and can be used in UC#20	
Failed End Condition	System does not receive values nor the device interfacing gives meaningful values to the system	
Actors	System, Device, Administrator	
Trigger	System receives signals from device	
Main Flow	Step	Action
	1	System receives signals from device
	2	Device interfacing module converts the value into system readable value
	3	System saves the value into a log file and prepares the value to be used in UC#14
Extensions	Step	
	2.1	System cannot read the value from device interfacing module
	2.2a	System attempts to send an instruction to request device status via UC#14
	2.2b	Upon failure, administrator is notified in UC#18 and requested to check the status of the device interfacing module configuration and the device connection
	2.3b	Administrator corrects the problem and system attempts 2.2a

UC-#16 : Add System Interface Module		
Related Requirements	REQ-5, REQ-7	
Goal In Context	Administrator adds a system interface module to the system to fetch Internet data	
Preconditions	System is logged on	
Successful End Condition	Administrator adds a system interface module to the system	
Failed End Condition	Administrator does not add a system interface module	
Actors	Administrator, System	
Trigger	Administrator enters Add/Remove External Data Source page	
Main Flow	Step	Action

	1	Administrator enters Add/Remove External Data Source page
	2	System provides a page for administrator to configure the source of the data
	3	Administrator choose to fetch data from internet or from a software that is installed on the system
	4	System updates the database and starts fetching data from the new source in UC#19
Extensions	Step	
	4.1	System cannot fetch any data from the specified source and prompts the administrator about the error
	4.2	Administrator corrects the error and system updates the database and starts fetching data from the source in UC#19

UC-#17 : Remove System Interface Module		
Related Requirements	REQ-5, REQ-7	
Goal In Context	Administrator removes a system interface module	
Preconditions	A system interface module is installed	
Successful End Condition	Administrator removes a system interface module and system no longer receives data from UC#19	
Failed End Condition	Administrator does not remove a system interface module	
Actors	Administrator, System	
Trigger	Administrator enters Add/Remove External Data Source page	
Main Flow	Step	Action
	1	Administrator enters Add/Remove External Data Source page
	2	System provides a page for administrator to configure the source of the data
	3	Administrator removes a source from the preference
	4	System updates the database and the source is no longer involved in UC#19
Extensions	Step	
	N/A	N/A

UC-#18 : Send System Interface Module Signal		
Related Requirements	REQ-6, REQ-7	
Goal In Context	System sets an active value to control modules such as text message, email and computer-control modules to initiate them.	
Preconditions	System is connected to the internet and devices are connected to system	
Successful End Condition	System sets an active value to control specific modules	
Failed End Condition	System does not set an active value to control specific modules	
Actors	System, Device	

Trigger	UC#20 is successful and system detects the need of UC#18	
Main Flow	Step	Action
	1	UC#20 is successful and system detects the need of UC#18
	2	System sets an active value with respect to the module
	3	A device interface module converts the value into meaningful commands and initiates the corresponding modules
Extensions	Step	
	3.1	The corresponding module does not respond to the commands
	3.2a	System attempts to resend the command to the module
	3.2b	Upon failure, system checks if the internet connection is established. An error message on screen is generated.

UC-#19 : Receive System Interface Module Signal		
Related Requirements	REQ-6, REQ-7	
Goal In Context	System receives data from internet	
Preconditions	Internet connection is established	
Successful End Condition	System receives data from internet	
Failed End Condition	System does not receive any data or incorrect data from internet	
Actors	System, Internet	
Trigger	UC#20 is successful and system detects the need to fetch data from internet for analysis. System requests data from internet in UC#18.	
Main Flow	Step	Action
	1	UC#20 is successful and system detects the need to fetch data from internet for analysis. System requests data from internet in UC#18.
	2	System receives data from internet and the data are saved to the database. Device interface module controls the poll rate and interfaces
Extensions	Step	
	2.1	System does not receive any data from a specific source. Administrator is prompted about this error and requested to check the data source in UC#16

UC-#20 : Evaluate Data Against Rule Sets	
Related Requirements	REQ-2, REQ-8
Goal In Context	System analyzes data collected and data stored in condition set and rule sets
Preconditions	Data is collected in UC#19 and UC#5 are successful
Successful End Condition	System analyzes the data and stores it in the database
Failed End Condition	System does not analyze the data nor the analyzed data is correct

Actors	System	
Trigger	System received data from UC#15 and followed by UC#19	
Main Flow	Step	Action
	1	System received data from UC#15 and followed by UC#19
	2	System analyzes data base on the mathematical models and saves it into the database
Extensions	Step	
	N/A	N/A

UC-#21 : Execute Action Set		
Related Requirements	REQ-3, REQ-6	
Goal In Context	System activates a set of devices at the same time	
Preconditions	UC#20 is successful and the need of UC#21 is detected	
Successful End Condition	System activates a set of devices at the same time	
Failed End Condition	System does not activate a set of device at the same time	
Actors	System, Device	
Trigger	UC#20 is successful and the need of UC#21 is detected	
Main Flow	Step	Action
	1	UC#20 is successful and the need of UC#21 is detected
	2	System sends instructions to the set of corresponding devices in UC#14
Extensions	Step	
	2.1	One or more devices are not initiated at the same time
	2.2	System prompts the administrator to check the condition sets and rule sets in UC#5 and the connections of the devices
	2.3	Administrator corrects the problem in UC#5 and the connection of the devices

UC-#22 : Backup Database and Device Interface Modules		
Related Requirements	REQ-4, REQ-5	
Goal In Context	System performs backup and saves it to a specified media	
Preconditions	N/A	
Successful End Condition	System backup is saved to a specified media	
Failed End Condition	System backup fails	
Actors	Administrator, System	
Trigger	Administrator manually initiates backup or initiated by schedule	
Main Flow	Step	Action

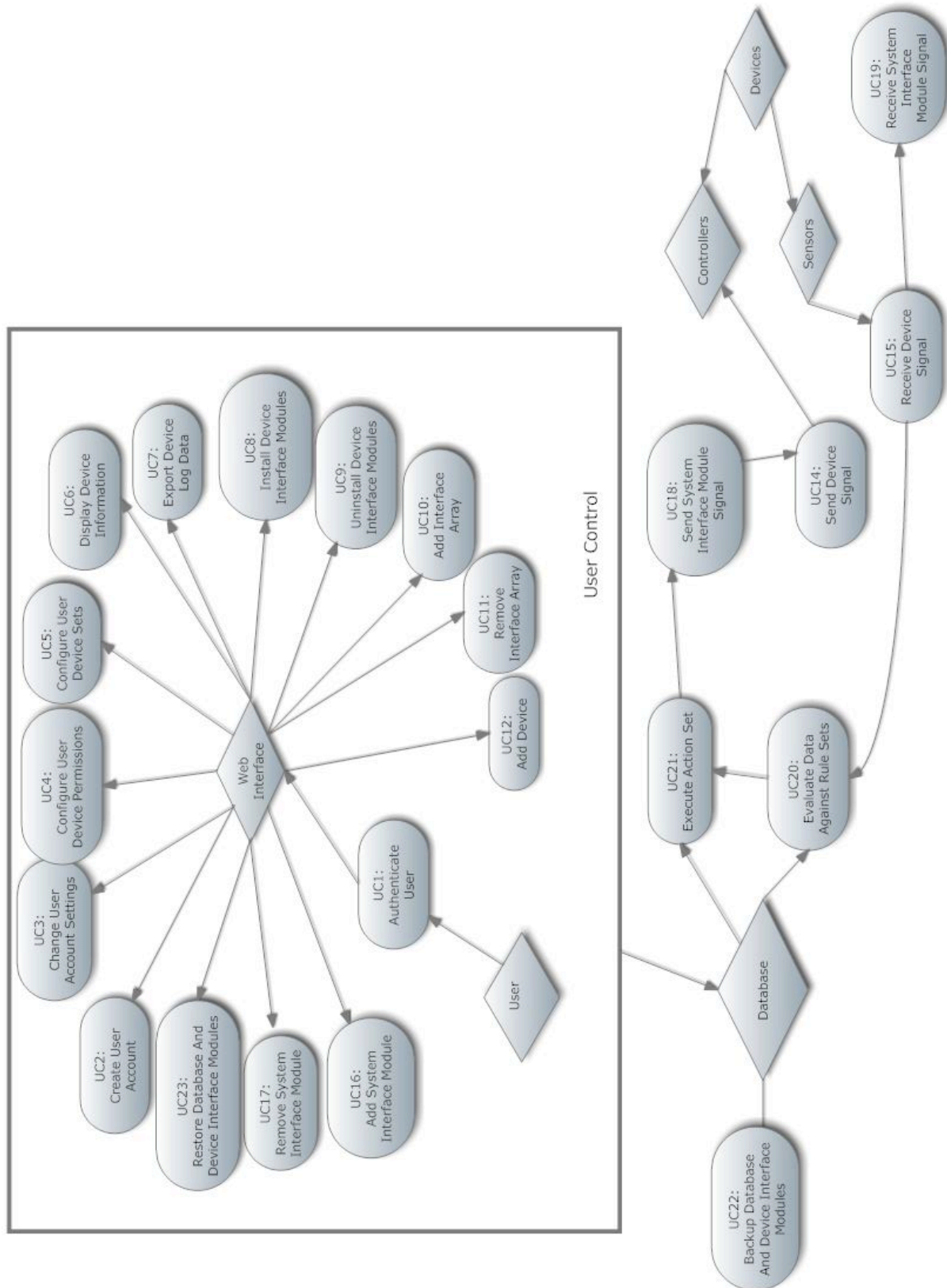
	1	Administrator manually initiates backup or initiated by schedule
	2	System performs a full backup and saves the data to a specified media
Extensions	Step	
	2.1	System fails to perform backup due to either the media is full or no media is connected
	2.2	Administrator is notified and requested to check the media
	2.3	Administrator corrects the error and proceed with the backup procedure

UC-#23 : Restore Database and Device Interface Modules		
Related Requirements	REQ-4, REQ-5	
Goal In Context	Administrator performs a full restore on the system	
Preconditions	System is malfunctioning	
Successful End Condition	System is restored	
Failed End Condition	System is not restored	
Actors	Administrator, System	
Trigger	Administrator enters Backup/Restore page and initiates restore procedure	
Main Flow	Step	Action
	1	Administrator enters Backup/Restore page and initiates restore procedure
	2	System verifies the backup image and performs restore
Extensions	Step	
	2.1a	The backup image is non-functioning
	2.2a	System prompts the administrator about the error and requests for another backup image
	2.1b	Restore process fails
	2.2b	System performs roll-back to the previous state and prompts administrator about the error

#### IV. Traceability Matrix

Use Cases	REQ 1	REQ 2	REQ 3	REQ 4	REQ 5	REQ 6	REQ 7	REQ 8	MAX PW	Total PW
PW	10	10	10	10	7	6	5	3		
UC#1 AuthenticateUser	x				x				10	17
UC#2 CreateUser	x				x				10	17
UC#3 ChangeUserSetting	x				x				10	17
UC#4 ConfigPermissions			x		x				10	17
UC#5 ConfigRuleSets			x		x				10	17
UC#6 DisplayDevInfo					x				7	7
UC#7 ExportLog					x	x			7	13
UC#8 InstallDevMod					x		x		7	12
UC#9 UninstallDevMod					x		x		7	12
UC#10 AddIntArray					x		x		7	12
UC#11 RemoveIntArray					x		x		7	12
UC#12 AddDevice					x		x		7	12
UC#13 RemoveDevice					x		x		7	12
UC#14 SendDevSignal		x							10	10
UC#15 ReceiveDevSignal		x							10	10
UC#16 AddSysMod					x		x		7	12
UC#17 RemoveSysMod					x		x		7	12
UC#18 SendSysSig						x	x		6	11
UC#19 ReceiveSysSig						x	x		6	11
UC#20 EvalData		x						x	10	13
UC#21 ExecuteAction			x			x			10	16
UC#22 BackupDBDev				x	x				10	17
UC#23 RestoreDBDev				x	x				10	17

## V. System Sequence Diagram





# User Interface Design

## Preliminary Design

For the design of the web interface it was essential that the user use the least amount of effort to navigate the interface. To make things very clean and simple to work with a minimalist approach was taken to the design of the interface.

## User Effort Estimation

Here described are two scenarios for which user effort estimation is going to be shown for. The first scenario involves adding a device to the autoHome system and the second one is the alert system for which the demo is going to be done on.

### Scenario 1: Add a device to the system, in this case the user will be adding an old fire alarm which is connected to the legacy-connections-hub.

- o Login into System:
  - § Data Entry: 3 clicks OR 1 click and 2 keystrokes
    - Click in “user ID” field and enter user ID
    - Press “Tab” or Click in the “PASSWORD” field and enter user password
    - Press “Enter” or Click the arrow button to commence login
- o Selection of Settings option:
  - § Navigation: 1 click
    - Click the “settings” button to proceed into the settings page
- o Adding device:
  - § Navigation: 4 clicks [Only used if fire alarm is autoHome compatible]
    - Click “Add Device” option to open up dialogue for adding a device
    - Click “Choose from list”, this option has a basic list of previously configured devices that user can select from.
    - Click “Fire Alarm”, from the list user chooses the Fire Alarm they wish to add.
    - Click “Apply” button to finish setup and add device.
  - § Data Entry: 6 Clicks [This is used for the old fire alarm connected to the hub]
    - Click “Add Device” option to open up dialogue for adding a device
    - Click “Manual Entry” to manually setup and add device
    - User types in the name of the device, i.e Name: “Fire Alarm”
    - Click “Port #” option to select the port which the alarm is connected to.
    - Click “Type of Device”, basic selection of Emergency device, Appliance, etc...
    - Click “Emergency Device” since in this scenario a fire alarm is being added.
    - Click “Apply” to finish setup and exit the dialogue.

### Scenario 2: This scenario will show how the effort needed to check any alerts

- o Login into System:
  - § Data Entry: 3 clicks OR 1 click and 2 keystrokes
    - Click in “user id” field and enter user ID
    - Press “Tab” or Click in the “PASSWORD” field and enter user password
    - Press “Enter” or Click the arrow button to commence login
- o Selection of “Alerts” option:
  - § Navigation: 1 click

- Click the “Alerts” button to proceed into the alerts page
- o Viewing and Checking Alerts:
  - § Navigation: 5 clicks
    - Click the alert name to open dialogue box for that alert, i.e click “Fire Alarm is active” to open dialogue box for that alert, in this case it would display the room(s) where the fire alarm has been activated.
    - Click “Details” to open up the details for this alert for further investigation, additionally if the alert is not considered as important the user can also select from “Ignore” and “Cancel” to exit the alert dialogue.
    - Click “Contact Emergency”, generally for this kind of scenario the system will automatically contact emergency services if the <emergency device> has been active for longer than 5-10 minutes, however in this specific case it is assumed that the user is operating from a different location. There are two other options “Shutdown System” (similar to contact emergency, except the system shuts down instead to prevent further damage to it) and “Attempt Fix” (attempts to fix the issue).
    - Click “Accept”, this option is will acknowledge the alert and exit the dialogue.



**Description:** The user is initially greeted with the login page for the web interface of autoHome. Here they will login in with their given ID and PIN to access the interactive settings section of the web interface.



**Description:** After Login the user is presented with a very simple selection of options. This is in effort to make the system as user friendly as possible. The selections range from Status, Settings, Alerts and Exit. Status button redirects the user to the status page which displays the status of the devices connected to the system. Settings redirects the user to the settings page where the user can configure the devices. Alerts will redirect to the alerts page where all the alerts are displayed in a simple format for the user to check. Exit button just exits from the web interface, it's made big to augment with the entire minimalist and user friendly theme.



Instructions: Click on device from the floor map to view its status

**Description:** Selecting the Status option will bring the user to this page where the user can check the status of the devices connected to the system. To eliminate any complexity a graphical interface is used where all the connected devices are shown in their respective location on the floor map, clicking the “next floor” button will change the floor, there is also a “previous floor” button as well but it only appears with the user is on a higher floor. The 3-D house map will show the entire house in a three dimensional view for aesthetic purposes. The “Display All Settings” will show the basic working status of all the devices connected to the system.

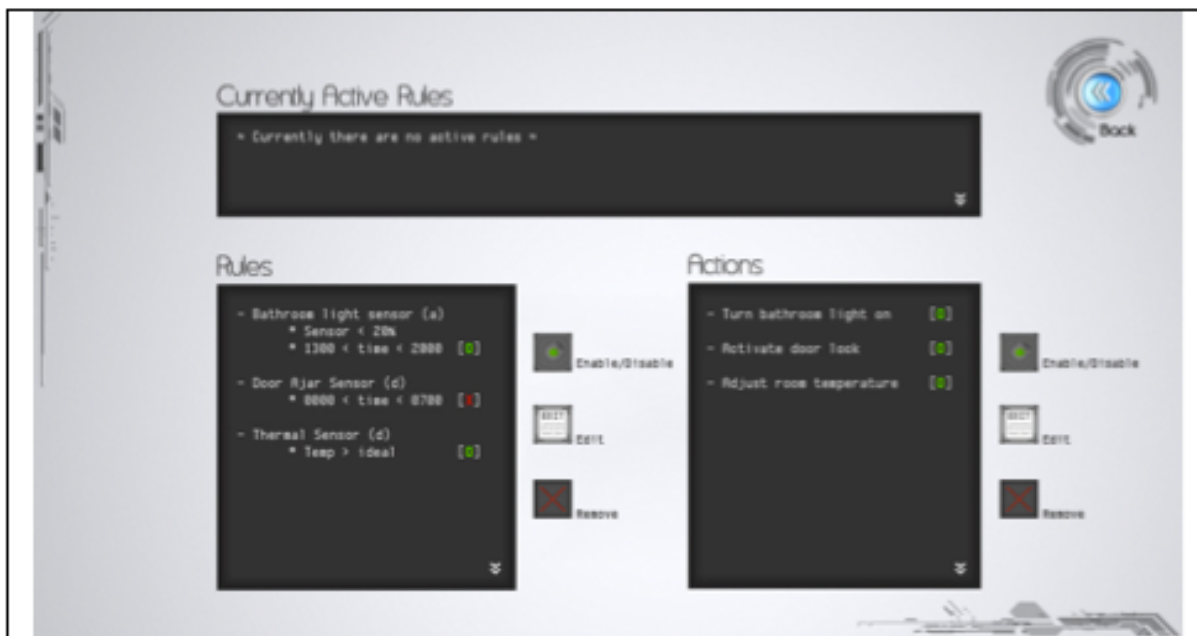


**Description:** The Settings page gives the user the options to add, remove, configure or diagnose a device. The add and remove function open a

dialogue box where the user can simply select from a preconfigured list of common devices or manually setup a device. Remove device is very simple, clicking remove device will prompt the user to select a device from the center panel device list and the selected device disconnect from the system and no longer appear on the list. Diagnose device prompts the user to select a device from the list and automatically begins to check if there are any connection issues with the device, and reports to the user the issue. The device list has an online and offline section, this basically shows which devices are being used and which devices are not being used and the locations of those devices.



**Description:** The alerts page displayed is separated into two columns major alerts and minor alerts, this allows the user quick access to the any messages generated by the system. The minor alerts section displays any small updates that the system receives and user acknowledged alerts. The major alerts section is separated so that the severe alerts are not buried within alerts that are not as important, these alerts include system warnings and general cautions. The user can click the alert and decide to act on it by selecting from the options, “Details”, “Ignore”, and “Cancel”. The detailed option gives a much more elaborate description of the alert, such as where the location of the problem device is and details about the issue itself.



**Description:** The Device rules page allows the user to set rules to certain devices. Such as setting the lighting to turn on at a certain time. The allows flexibility to completely customzie rules to how the user wishes to “run” the house. Rules are set and Actions can be enabled to augment the rules.



# Domain Analysis

## Domain Model

### Standardization

Home automation has many moving parts and one of the main challenges is to standardize all the different equipment being used to monitor and control household appliances. Devices controlled or monitored may include electronic ovens, fire alarms, ajar door sensors, individual power meters, motion sensors, and luminescence detectors. Most, if not all, of the useful sensors that this project plans to utilize will send or process a single data value. This makes it very convenient to lay out a generic way to process the inputs from, or outputs to these devices. Each device can then be reduced to an object bearing a single data value to report to the system, or an object receiving a single data value from the system to govern the device's actions, or both. These standardized objects will be the device nodes as shown in the diagram. The data values obtained from these devices can then be categorized into analog, digital, or none (if the device hasn't a value to report, or cannot be controlled).



## Sensors

### Alarm-type (Digital) Sensors

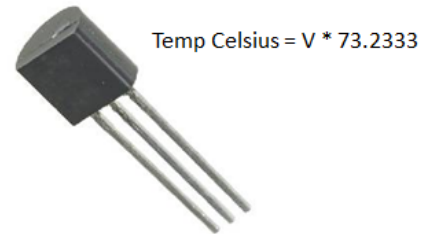
These devices have two states. We will denote them in our application as logical 0 (off) or logical 1 (on). They can be either active or inactive. A fire alarm will be either blaring due to hazardous conditions, or not.

A window-mounted burglar alarm will show if the window is open or not. A simple light detector will be active if it receives light above a certain intensity threshold. A carbon-monoxide alarm will be active only if the carbon-monoxide levels are above a certain threshold. The data collected from these devices can be put into a meaningful graph that will look like a digital signal. Users would be able to determine exactly when a fire alarm went off or what time windows were open at and for how long.

## Reading-type (Analog) Sensors

If required to track values over a period of time, reading-type sensors with analog-reporting capabilities are required. Power sensors would report a varying signal value correlated to the use of electricity. Complex light intensity sensors would give numbers correlating to how intense light is shining on them. Water usage sensors might give values indicating the electromagnetic disturbance varying volumes of flowing water might have on them. Thermometer devices give the current temperature. The values obtained are then logged according to specific device standards to produce meaningful and useful graphs and data sets.

### Temperature Sensor



## Controllers

### On/off-type (Digital) Controllers

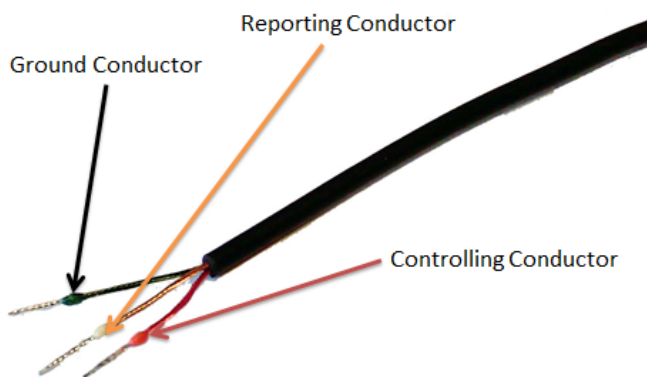
Digital controllers will send commands to devices with only two states. Again, these states will be represented as logical 0 (off) and logical 1 (on). Remotely controlled light-switches would only have an on or off state. Controlling whether a power outlet receives power only involves two states--off or on. Unlocking or locking a door would also deal with only two states.

### Adjustable-type (Analog) Controllers

Analog controllers send commands to devices and govern their behavior by means of a meaningful value. A temperature control system would receive an analog value specifying what temperature to keep a certain room at. A speaker system could receive volume controls to adjust the sound output. Dimmer-light switches might receive values indicating how much to dim the lighting by. With these fine-tuned controls, a user could, for example, set up rules to control the intensity of the built-in interior lights according to the levels of light already in the room from the sun or other sources.

## Physical (Layer 1) Device Communication

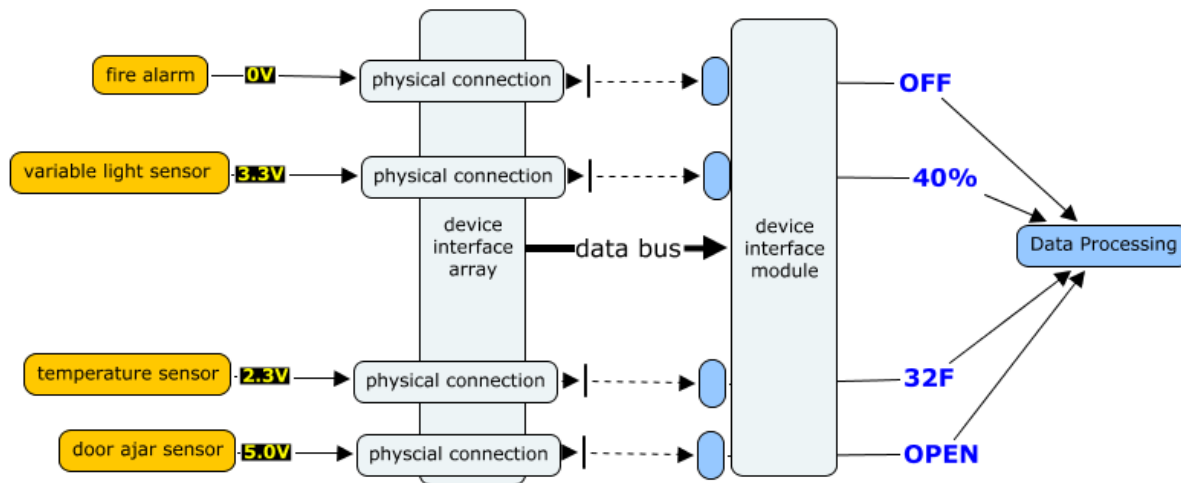
Although the home automation system is geared towards new homes being built (as retrofitting wiring is quite laborious), some methods are introduced to reduce the wiring requirements of new homes, and to ease the transition of older homes into automation.



The physical interaction of these devices will consist of three different types of conductors (wires). The first (required) conductor is the relative ground of the system and the device. The second type of conductor acts as a data channel that the device will use to report either digital or analog values to the central system. The third type of conductor acts as a central system-controlled channel to relay commands (in either analog or digital encoding) to the device. All devices will have a ground. They may have one or many of any of the combinations of the other conductors.



## Device Interface Array and Device Interface Modules



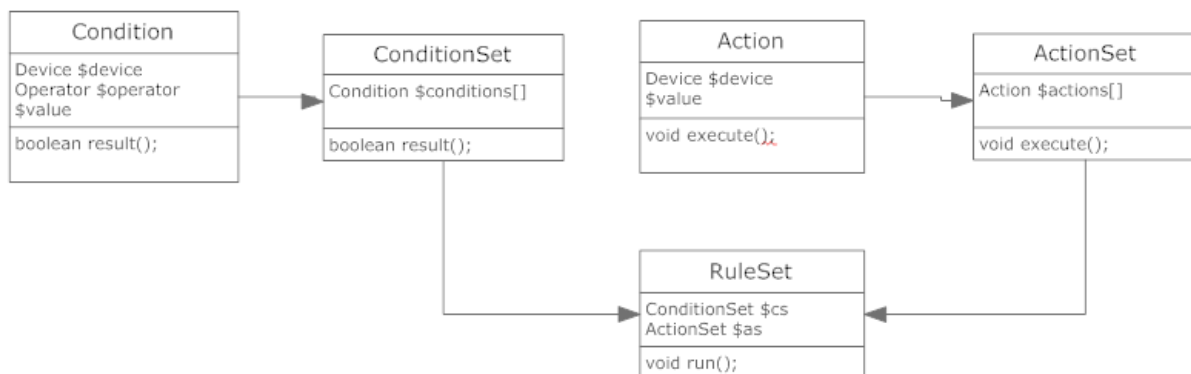
### System Devices

- The autoHome system comes with software-based sensor devices that report useful information such as time and date.
- There are also software-based controller devices that can use the Internet to send text messages, make phone calls, check server status, and many other useful things.

### Ruleset

A home is hardly automated if it does not intelligently follow a set of rules configurable by the home owner. We introduce two rulesets used to govern the collecting of data, and the execution of commands to devices. By having a generic way to add conditions that will result in actions, the user is granted extreme flexibility in deciding exactly what they want their home to do.

Actions are simple commands that relay commands to control individual values. Each action has a respective controller and value defined by the user.



Examples of Actions:

1. Dim "Living Room Light" to 50%
2. Lock "Basement-Outside Door"
3. Set "A/C System" to 50.00 Degrees Fahrenheit
4. Log "Temperature Sensor" Value

Actions are chained together to form action sets--sets of events that execute quickly enough in sequence as to be considered instantaneous..

Example Action-Set:

{Action #1, Action #2} => Dim "Living Room Light" to 50%, Lock "Basement Door"

The execution of action sets are governed by condition sets. Condition sets are made of conditions, which are simple variable comparisons to sensor values.

Examples of Conditions:

1. "Bathroom Light Sensor" > 60%
2. "Front Door Status" == Open
3. "Fire Alarm Status" == Active
4. "Time of Day" == 08:23
5. "Time of Day" > 06:30
6. "Time of Day" < 23:30
7. "Date" == 2012-02-17

Conditions must have all events evaluated as true to execute an associated action set.

For example, in the following condition set and action set relation:

{Condition #5, Condition #6} => {Action #1, Action #2}

As long as conditions 5 and 6 both remain satisfied, actions 1 and 2 will be continuously held.

I.E. as long as it is between 6:30AM and 11:30PM, the living room light will be dimmed to 50% and the basement-outside door will be locked.

Allowing users to define their own rules gives many possibilities for complex and useful rulesets. A user could, for example, keep porch lights on during the evening, only on weekends.

## System Operations Contract

Operation	Poll Sensor Data
Preconditions	<ul style="list-style-type: none"><li>• Device Interface Module thread for reading sensor device running</li><li>• Device-specific wait-time (specified in Device Interface Module) elapsed</li></ul>
Post-conditions	<ul style="list-style-type: none"><li>• Push updated data to device objects</li><li>• Drive all RuleSets</li></ul>

Operation	Set Controller Value
Preconditions	<ul style="list-style-type: none"><li>• Device Interface Module thread for setting controller device running</li><li>• Device object set function called</li></ul>
Post-conditions	<ul style="list-style-type: none"><li>• Lock Device Interface Module thread to value.</li></ul>

Operation	Execute ActionSet
Preconditions	<ul style="list-style-type: none"><li>• New data updated in device</li><li>• Relevant ConditionSet successful</li></ul>
Post-conditions	<ul style="list-style-type: none"><li>• Actions in ActionSet executed.</li></ul>

Operation	Database Backup
Preconditions	<ul style="list-style-type: none"><li>• User-specified backup time has elapsed</li></ul>
Post-conditions	<ul style="list-style-type: none"><li>• System dumps databases and module code into compressed tarball.</li><li>• Tarball is backed up at a different location.</li></ul>

## Mathematical Model

The autoHome software contains a few mathematical models as listed below:

- Electricity Usage [11]
  - A database containing common power usage of most common home appliances will be used for an estimate
    - Power is measured as **Power(Watts) = Volts(V) \* Amperes(A)**
  - The amount of time that a device is turned on will be measured in seconds
    - **Work(Joules) = Power(Watts) \* Time(s)**
  - Cost of energy will be calculated
    - **Total Cost(\$ ) = Work \* (\$ per joules)**
- Water Usage [12]
  - Flow meters will be installed on showers, sinks, outdoor hoses, and toilets
    - **Flow = Gallons(gal) \* Time(min)**
  - Cost of water will be calculated
    - **Total Cost(\$ ) = Flow \* (\$ per gallon)**
- Inductive Charging [13]
  - The voltage transferred to the device we want to charge is

$$V_s = \frac{N_s}{N_p} V_p$$

$V_s$  = voltage to the device being charged

$N_1 = N_2$  = the number of copper coils

$V_p$  = the voltage from the source

- The current is calculated as

$$I_s = \frac{N_p}{N_s} I_p$$

$I_s$  = current to the device being charged

$N_1 = N_2$  = the number of copper coils

$I_p$  = the current from the source

- The device power can then be calculated as described from “Electricity Usage”

## Plan of Work

<b>Milestone</b>	<b>Expected Completion</b>	<b>Tasks/ People Involved</b>
<b>Complete Database</b> <ul style="list-style-type: none"> <li>• Relational DB</li> <li>• Tables for rooms</li> <li>• Tables for devices</li> <li>• Tables for device properties</li> </ul>	2/15/2012	Wayne has created the majority of the database structure, Elie will help to add to the database with minor fixes and changes
<b>Design Website</b> <ul style="list-style-type: none"> <li>• Full GUI</li> <li>• Working Buttons</li> <li>• Working Dials</li> <li>• List of Rooms</li> <li>• List of Devices</li> </ul>	3/01/2012	Rohith will be working on the main design of the web page and Paul will implement the basic design structure of the site.
<b>Connect Website to Database</b> <ul style="list-style-type: none"> <li>• Implement PHP hooks</li> <li>• implement a log of all events in the system</li> </ul>	3/03/2012	Elie will connect the web interfaces with the DB
<b>Second Report</b>	3/09/2012	All of us will collectively complete
<b>Develop API</b> <ul style="list-style-type: none"> <li>• Enable hooks/functions for use throughout the site</li> </ul>	3/15/2012	Calvin and Elvison will develop simple functions to interact with the software
<b>First Demo</b> <ul style="list-style-type: none"> <li>• Completed website</li> <li>• Completed DB</li> <li>• Hooks to outside API <ul style="list-style-type: none"> <li>◦ Weather</li> </ul> </li> <li>• Map of home</li> <li>• Ability to simulate adding devices</li> </ul>	3/27/2012	All of us will collectively complete
<b>Temperature Sensor</b> <ul style="list-style-type: none"> <li>• Link to site</li> <li>• Display data</li> </ul>	4/05/2012	Wayne, Elie, and Paul will create some hardware
<b>Create Light Array</b> <ul style="list-style-type: none"> <li>• Using relay turn on array of lights</li> <li>• controlled through website</li> </ul>	4/10/2012	Rohith, Calvin, and Elvison will design a working light array
<b>Text Messaging API</b> <ul style="list-style-type: none"> <li>• Introduce ability to send text notifications</li> </ul>	4/15/2012	Elie will set up the text messaging service
<b>Third Report</b>	4/27/2012	All of us will collectively complete

Second Demo <ul style="list-style-type: none"> <li>• Ability to receive text notifications and email</li> <li>• Turn on/off devices</li> <li>• Receive analog input</li> <li>• Final working product</li> </ul>	5/01/2012	All of us will collectively complete
Completed e-archive	5/03/2012	All of us will collectively complete

# References

- [1] FURPS model - [http://en.wikipedia.org/wiki/FURPS;  
http://www.ibm.com/developerworks/rational/library/4706.html#N100A7](http://en.wikipedia.org/wiki/FURPS;http://www.ibm.com/developerworks/rational/library/4706.html#N100A7)
- [2] Requirements analysis - [http://en.wikipedia.org/wiki/Requirements\\_analysis#Stakeholder\\_identification](http://en.wikipedia.org/wiki/Requirements_analysis#Stakeholder_identification)
- [3] Cloud based backup solution - <http://www.backblaze.com/>
- [4] jQuery documentation and usage examples - [http://docs.jquery.com/Main\\_Page](http://docs.jquery.com/Main_Page)
- [5] Arduino microcontroller - <http://arduino.cc/>
- [6] Version control system - <http://www.github.com>
- [7] Infrared motion detector - [http://en.wikipedia.org/wiki/Passive\\_infrared\\_sensor](http://en.wikipedia.org/wiki/Passive_infrared_sensor)
- [8] Temperature / humidity monitor POE - <http://www.itwatchdogs.com/product-detail-microgoose-9.html>
- [9] Water level sensor - <http://www.amazon.com/United-Security-Products-Water-Sensor/dp/B001HLAOX4>
- [10] Light intensity sensor, outdoor - <http://www.onsetcomp.com/products/sensors/light-intensity>
- [11] How to measure electricity - [http://en.wikipedia.org/wiki/Electric\\_power](http://en.wikipedia.org/wiki/Electric_power)
- [12] How to measure water - [http://en.wikipedia.org/wiki/Flow\\_measurement#Liquid](http://en.wikipedia.org/wiki/Flow_measurement#Liquid)
- [13] Coupled Induction for charging devices - [http://en.wikipedia.org/wiki/Mutual\\_inductance#Coupled\\_inductors](http://en.wikipedia.org/wiki/Mutual_inductance#Coupled_inductors)
- [14] House blueprint - <http://homeinteriortrend.com/wp-content/uploads/2011/06/sketch-smart-house-design-6.jpg>