Solar Decathlon Smart Home Control
Final Project Proposal

Development Team

Jackson Dawkins
jacksod@g.clemson.edu
Joey Costa
jacosta@g.clemson.edu
Alex Ross
alross@clemson.edu

Target Devices

We aim to support all devices that can run iOS 8, which consists of: the iPhone 4s, 5, 5c, 5s, 6, 6 Plus, and iPod Touch 5th generation. If time allows, we will extend support to the iPad Air and iPad Mini sizes.

Target Audience

The target audience for our app is comprised of three parties: students working on the home, contest attendees, and contest panel judges. Students working on the home (including those not on our software team) will benefit from being able to monitor and adjust conditions within the home for testing, comfort, and convenience. We suspect that the contest attendees that visit our home will largely want to test out the capabilities of our system, so preparing for extreme use cases will be of utmost importance in order to appeal to these attendees. Lastly, we intend to develop and debug our project in such a way that would appeal to the contest judges, who are likely going to attempt various usage patterns and inspect our project for standout innovation and creativity.

Schedule of Implementation

Date	Jackson	Joey	Alex
11/11/14	Dashboard prototype completed	External database created and dummy values inserted	Basic graph functionality implemented
11/18/14	All UI prototypes completed; implements charting	Implemented temperature & humidity View Controllers by retrieving and modifying values in the external database	Graph functionality completed; some user system completed; some user manual completed
11/25/14	Constraints added for different screen sizes/devices	Admin functionality; help with the User & Technical Reference Manual	User system completed; User Manual completed; some Technical Reference completed
12/02/14	Added sound effects to	Added gesture functionality	Technical Reference

	interactions		completed
12/05/14	First draft of Software and Documentation completed		
12/09/14	Final draft of Software and Documentation		
12/11/14	Final presentation (8:00am - 12 noon)		

Summary

The Solar Decathlon Smart Home Control app provides a monitoring and control system for Clemson's entry into the U.S. Department of Energy's Solar Decathlon for 2015. The home is equipped with multiple sensors that have access to a server within the house. This server has Internet access and periodically uploads home statistics to an offsite database. The values stored in this database can be accessible by users of the home control app.

This app is able to access all of the data for analytics purposes. Data on the server is a long-term record of all sensors. Data on the device consists of the most recent of the server's information, to allow the user to view real-time information about the home. When the user selects a specific kind of statistic, the device will retrieve longer-term data for the purpose of charting changes over time.

The app allows for two classifications of users: standard viewers and administrators. The user experience of these two kinds of users is largely similar with a couple differences. When either user opens the app, he is displayed a dashboard-style overview of the current status of the house. This view is able to show statistics at glance as well as their relationship to the preferred value for that data type. For example, if the preferred temperature in the house is 70, and the current temperature is 75, the dashboard shows the user that the temperature is in excess of the preferred.

The dashboard is structured in a tiled format. When any kind of user taps on a tile for a specific kind of statistic, the app displays a detailed view of that statistic. This view includes a larger display of current status alongside the preferred status of the specific sensor. Also, on this screen will be the chart view of the statistic. This is where the summarized statistics will become useful. The user will be able to chart trends over the past few hours, days, weeks, or months.

The usefulness of these kinds of trends is endless. In a world where energy consumption has been becoming a larger concern by the year, it is important for people to know where they are over consuming, and where they can cut back. If the owner of a home has noticed energy consumption has increased over the past couple months, he can check to see why that is. He could display the temperature trends to see if he has been keeping the home cooler, and thus causing the home to do more work to keep cool. These kinds of statistics also apply to water and CO2 levels. Anything that the system collects statistics about can be viewed over time, to allow for greater knowledge and control of the house.

Not just any user could modify the house, though. This would be a detrimental thing in the case of a temporary home for a contest or a permanent home. In a contest home, if any user touring the house could modify the optimal temperature, lighting controls, or other critical

systems in the house, it could become complicated with multiple users. In a permanent home setting, this would be more of a security concern than anything. If someone from outside the home could turn off lights, motion sensors, or other security devices, it would open the home up for crime.

The way this system would implement these users in a real life deployment would probably be a request system. For the purposes of the competition and this specific app's implementation, there are some preset admin accounts for judging and testing purposes. New users will be able to register, but will have to be confirmed by an admin to be granted admin privileges.

Features

Dashboard with real-time metrics

This feature includes views for temperature, humidity, power consumption, water consumption, CO_2 ppm, lighting status, and motion detection. The dashboard will be updated every minute (to save battery) or every time any view controller is loaded (viewDidAppear method).

Detail view for each statistic

This includes separate views for the six monitored conditions (except for the lights and motion data), which shows the historical time-series data for the chosen monitor in graph form. When the view is initially loaded, the user sees the current status of the sensor, along with an hourly breakdown (in a chart) of the sensor. Swiping right on the chart will allow the user to expand the data to show, daily, weekly and even monthly breakdown of the sensors data (obviously not for light and motion sensor data).

Administrative Abilities

When a user is logged in as an administrator, the detail view is slightly different. Next to the sensor's current status is a small (+) and (-) so the admin can increase/decrease the preferred temperature, humidity etc. Users without administrative permissions (guests) will only be allowed to view the statistics about the home.

Customizable Lighting Controls

This includes being able to control individual lights with a simple tap on the light or having the ability to create custom scenes where one can control a group of lights.

Settings Bundle and Preferences Panel

This is used to save admin logins (if the user does not want to have to login each and every time the app is loaded), as well as set an acceptable tolerance for each statistic. For example, if the preferred temperature in the home is set to 70 with a 3-degree tolerance, the user is not notified that the home is at an unacceptable temperature until it drops below 67 or rises above 73.

Benefits

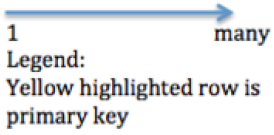
There are a number of benefits to implementing the Smart Home Control app. These benefits, while broad in scope, can be summarized into three main areas: convenience, assistance, and safety.

The convenience of the Smart Home Control app is one of the significant benefits to building the app. There are a number of scenarios with which it would be beneficial to have access to real-time metrics of a home. For example, the owner could keep tabs on utilities to make sure money isn't being wasted, and even optimize usage to save money on monthly bills. The homeowner could also check and adjust these values from within the home, or remotely. If it is too cold in the morning, he or she could open the app to turn the thermostat up to a comfortable level. If he or she left home for the night and forgot to turn the lights off, they could easily pull up the app and turn them off with just a couple taps.

Usage of the Smart Home Control app goes beyond simply adjusting the thermostat, however. One of the biggest potential uses of the app will be for those who are disabled and unable to easily complete minor household tasks, such as turning off the lights or shutting the blinds. While the initial plans of the app only include temperature, humidity, CO_2 ppm, water usage, power usage, light status, and motion detector metrics, it would be simple to use every actuator available to make more of these daily tasks less tedious and more accessible to all users.

Lastly, the safety benefits are another large part of what will make the Smart Home Control app great. Not only will the app actively keep track of the air density of CO_2 molecules, but it will warn the user if the home approaches an undesirable level and offer tips to air out your home. Additionally, with the array of sensors and actuators at the app's disposal, any user can remotely monitor their home—quickly seeing if an intruder has opened a door or turned on a light, for example. Motion sensors will also play a key role in indicating if there is movement in the house when there shouldn't be.

Internal Database



Legend:
Yellow highlighted row is primary key

Last Saved Values Table	
saved_id	Int
savedTemp	Decimal(4,1)
savedHumid	Int
savedWater	Int
savedPower	Int
savedCO2PPM	Int
savedTime	DATETIME

User Sessions Table	
session_id	Int
currentUser	TEXT
adminLevel	BOOL
loginTime	DATETIME

Key Values Table	
key_id	Int
setTemp	Decimal(4,1)
setHumid	Int
goalWaterUsage	Int
goalPowerUsage	Int
warningPPM	Int
updateTime	DATETIME

Figure 1: Internal database schema

Figure 1 depicts our planned internal database schema for the app. There are currently three tables planned: a User Sessions Table, a Last Saved Values Table, and a Key Values Table.

The User Sessions Table is the mechanism used to keep track of the logged in user, as app functionality depends largely on permission level. Not being logged in, or being logged in as a standard user, will allow you to view trends and current metrics for the various sensors. Being logged in as an admin, on the other hand, allows the user access to manipulating desired values—e.g., adjusting the temperature.

The Last Saved Values Table contains the last known values for each sensor. The app includes function to look at the external database (Figure 2) and retrieve the last known values for each sensor before the connection to the external database is lost (for example, if internet access cuts off). The app also stores the last time these values were captured within the table.

Similar to the Last Saved Values Table, the Key Values Table is comprised of all of the desired values for each sensor. When temperature and humidity are set, their values are updated within the table and the actuators within the home attempt to bring the values into an appropriate range, similar to how a thermostat works in a typical home with air conditioning and heat. When water and power consumption goals are set, their values are also stored within this table, though no physical response occurs. The level at which CO2 warnings will be given will also be stored this way when set within the app.

External Database

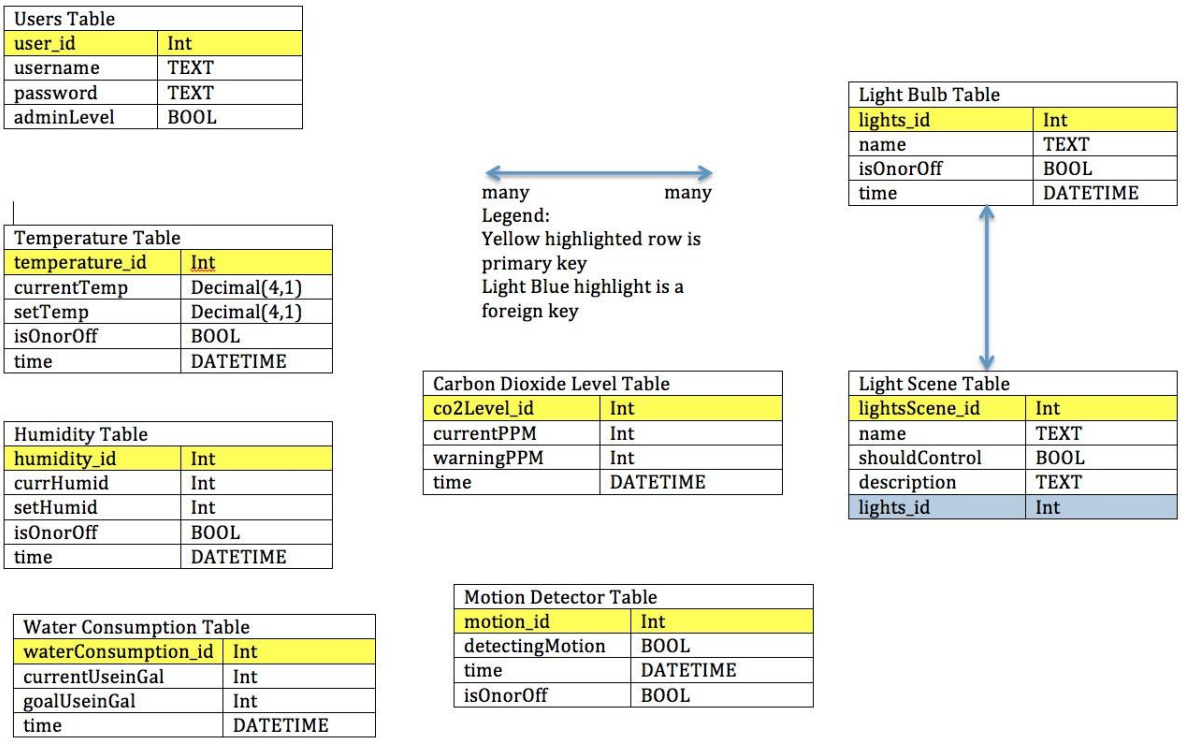


Figure 2: External database

This database contains separate tables for each type of sensor included in the house. This makes it simple to obtain more detailed information about a specific sensor without having to sort through large amounts of data. Each of these sensor tables includes a current and optimal field for the respective sensor.

The shouldControl field in the Lights Scene table has to do with creating a scene. The user needs the ability to turn on or off certain lights and not affect other ones. For example, if the user wants to turn on the kitchen light and bedroom light but not change the status of the front door light, then the shouldControl field in this scene would be set false for the front door light, but true for the kitchen and bedroom lights.

The time field in each table will record the date and time of each sensor. When a specific sensor adds a new entry into the database the exact date and time is recorded. This is useful for example, when setTemp is changed by 2 users the most recent setTemp is the one the sensors read. The boolean value, isOnorOff is for each sensor's actual on or off status. When setTemp is different than the currentTemp then isOnorOff should be set to 'on'. When a light bulb is physically on, then the isOnorOff status will be set accordingly.

The name value in the Light Bulb Table is used to signify the naming of a specific bulb (instead of the bulb being named Light bulb 35, for example, the user can name each of the individual bulbs they have placed in their house).

Storyboard
See attached PDF.

Initial View: Dashboard
<DashboardViewController>

Tapping a dashboard tile takes the **Non-Admin** user to a detailed view, with a chart over time. Swiping on the chart chooses different trends.

Trends include: Hourly, Daily, Weekly, Monthly, and Yearly.

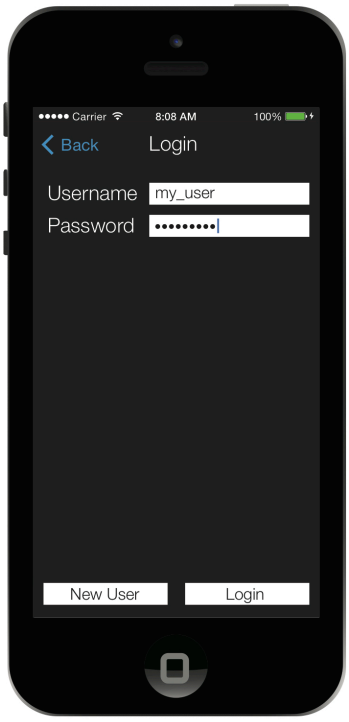
Tapping different tiles takes the user to different, but similarly laid out views.

Detailed Temperature View
<DetailedStatsViewController>

Lighting is shown in a different view on the following page

When the user clicks the 'Login' button, they are taken to the Login View

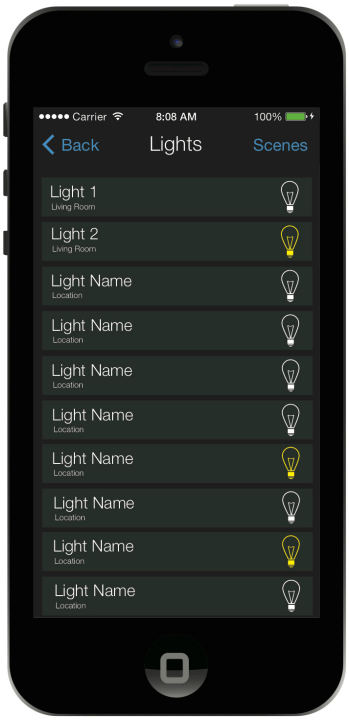
Login View
<LoginViewController>



Logged in users are taken to a similar detailed view, with step control buttons to increment or decrement the default (ideal) value.

Detailed Temperature Admin View
<DetailedStatsAdminViewController>

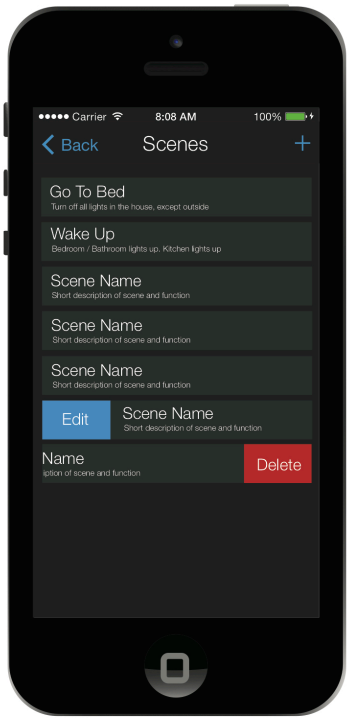
Lighting Status View
<LightingViewController>



Lighting scenes are presets for lights. Pressing "scenes" takes the admin user to the scene view. Displaying user-defined titles and descriptions of scenes. Tapping a scene toggles it, and sets the lighting values accordingly.

Tapping on an individual light cell toggles the state of the light from off to on or vise versa. The yellow light bulb represents a light in the "on" state

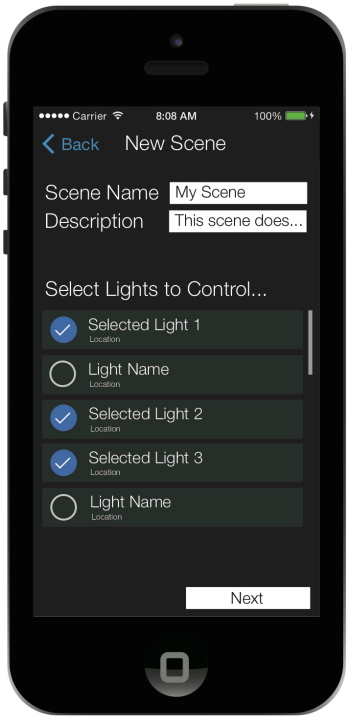
Lighting Scene View
<LightingSceneController>



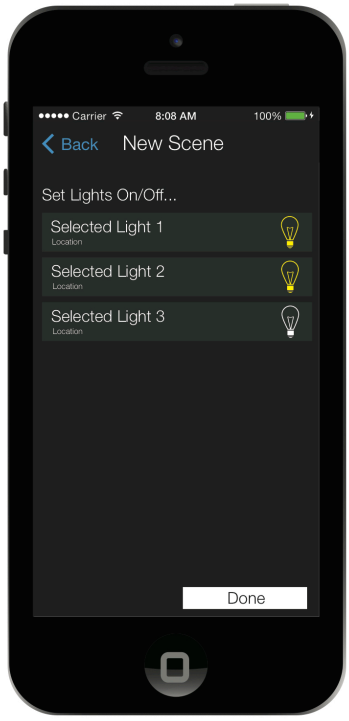
Tapping "+" brings the admin user to the **new scene view**, where the user sets a name and description to be shown. The user then selects which lights the scene should act upon, as scenes don't have to control every light in the house.

Swiping to edit brings the user to a similar view, but with 'Edit Scene' as the title and vaules already filled.

New Scene View
<NewSceneViewController>

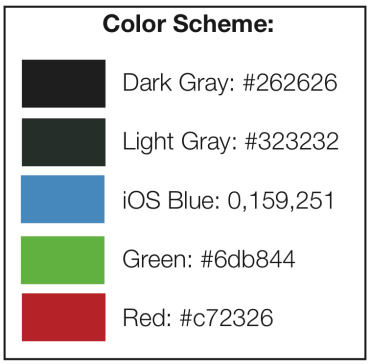


New Scene View 2
<NewScene2ViewController>



After hitting next, the user picks the state that the scene should cause the selected lights to go into. Once done, the user clicks "Done" and the scene is saved.

Solar Decathlon Home Controller Storyboard and UI Prototype



Assets Used

Font: Helvetica Neue (Light/Bold)
iPhone 5 Vector Graphic: *Sean Thomas Sweeney* http://seantsweeney.com/free-vector-iphone-5-image/
Light Bulb Vector Graphic: *Freepik*
http://www.freepik.com/free-vector/utility-identifies-small-icon-vector-material_573351.htm
iOS System Graphics: *Rusty Mitchell* http://mercury.io/blog/ios-8-illustrator-vector-ui-kit-update