Solar Decathlon Smart Home Control

Final Project Proposal

by The Thunder Ducklings: Jackson Dawkins, Joey Costa, and Alex Ross

Summary

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

This app will be able to access all of the data for analytics purposes. Data on the server will be a long-term record of all sensors. Data on the device will consist of summarized versions of the external database, depending on what the user wants to see.

The app will allow for two classifications of users: standard viewers and administrators. The user experience of these two kinds of users will largely be similar with a couple differences. When either user opens the app, he will be displayed a dashboard-style overview of the current status of the house. This view will be 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 will show 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 will display a detailed view of that statistic. This view will include 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 stored on the internal database 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 extremely 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 would probably be a request system. For the purposes of the competition and this specific app's implementation, there would be some preset admin accounts for judging and testing purposes. New users would be able to register, but would have to be confirmed by an admin to be granted admin privileges.

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.

Features

- Dashboard with real-time metrics for temperature, humidity, power consumption, water consumption, CO_2 ppm, and light status
    - The dashboard will be updated every minute (to save battery) or every time any view controller is loaded (viewDidAppear method).
- Detail view for each of 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 you initially load a detail view for a sensor you will be able to see 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).
    - The view for admin users will only be slightly different. Next to the sensors current status a small (+) and (-) will be shown so the admin would be able to increase/decrease the temperature, humidity etc.
- Detail view for the lights within the house
    - 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
- User system with login, register, and admin functionality
    - If you login as admin you will be able to control the houses sensors, without logging in you will only be able to view the sensors statuses.
- A preference panel
    - Used to save admins login (if the user does not want to have to login each and every time the app is loaded).

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.

Benefits

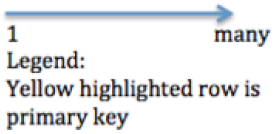
The Thunder Ducklings strongly believe that there are a number of benefits to implementing the Smart Home Control app. These benefits, while broad in scope, can be summarized into three main areas: convenience, assistance, and safety.

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 your home. For example, you could keep tabs on your utilities to make sure money isn't being wasted, and even optimize your usage to save money on your monthly bills. You could also check and adjust these values from within the home, or remotely. If it's too cold in the morning, you could pull up the app to turn the thermostat up to a comfortable level. If you left home for the night and forgot to turn the lights off, you 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. We believe that one of the biggest draws 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 our app only include temperature, humidity, CO_2 ppm, water usage, power usage, light status, and motion detector metrics, we intend to use every actuator available to us to make these daily tasks less tedious and more accessible to all users.

Lastly, we believe that 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 you if you approach an undesirable level and offer tips to air out your home. Additionally, with the array of sensors and actuators at our 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



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 will be the mechanism we use 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. We will have to implement some function to look at the external database (Figure 2) and retrieve the last known values for each sensors before the connection to the external database is loss (for example, if internet access cuts off). We will also store 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 as the actuators adjust the values within the home. When water and power consumption goals are set, their values are also stored within this table, though no physical response occurs. Warning PPM will also be stored this way when set within the app.

External Database

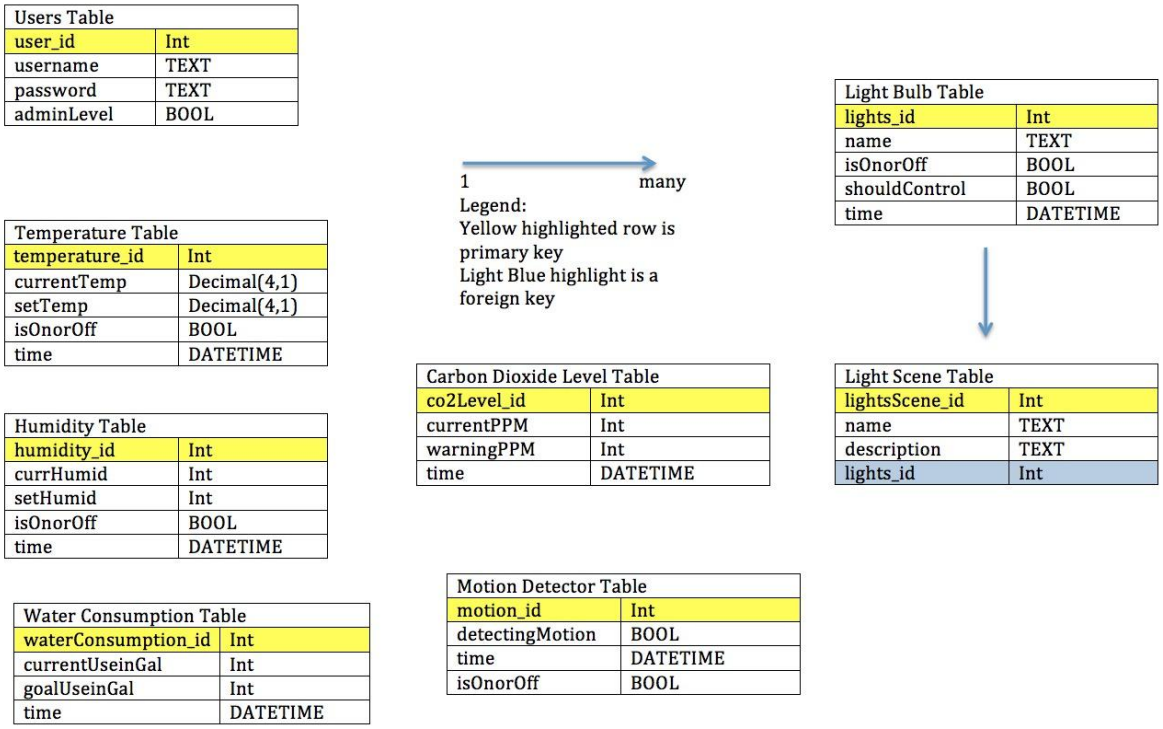


Figure 2: External database

For the lights_id the shouldControl is used only for the lightsScene table (the default value is ON). The idea behind shouldControl is if you have 4 lights kitchen, front door, bedroom, common room and you want to have a scene turn on the kitchen light and common room light then you would set the kitchen light and common room light isOnorOff to ON and then set shouldControl kitchen and common room light to ON. That way when you send the command to turn just those two lights on, because bedroom and common room lights "isOnorOff" would be OFF, those lights wont be affected by this command. Basically don't turn the other lights off because their isOnorOff will be set to OFF.

Storyboard and User Interface Concept

See following pages. For a higher quality view of the storyboards, see the included 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

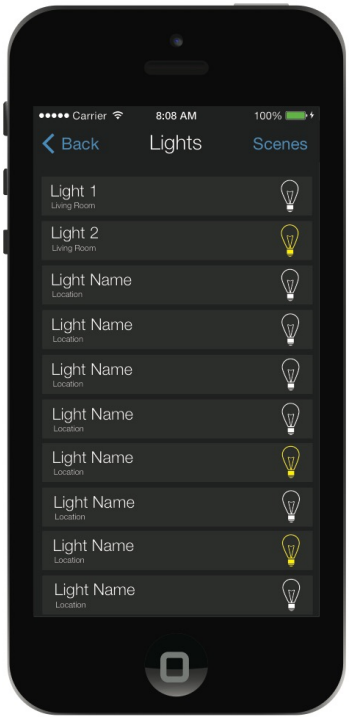
When the user clicks the 'Login' button, they are taken to the Login View

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>

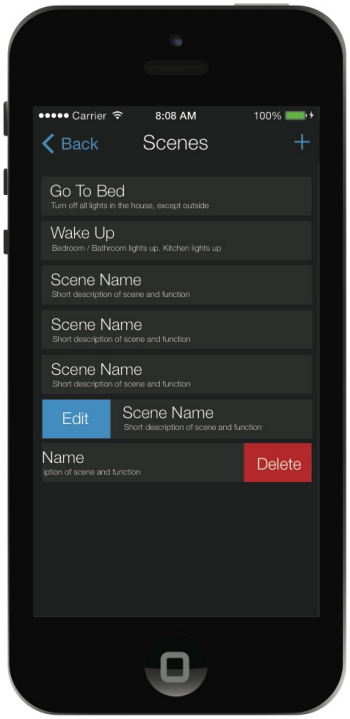
Lighting Status View
<LightingViewController>



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

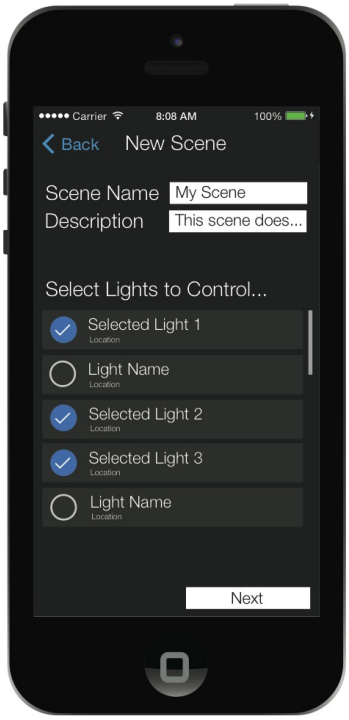
Lighting Scene View
<LightingSceneController>



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.

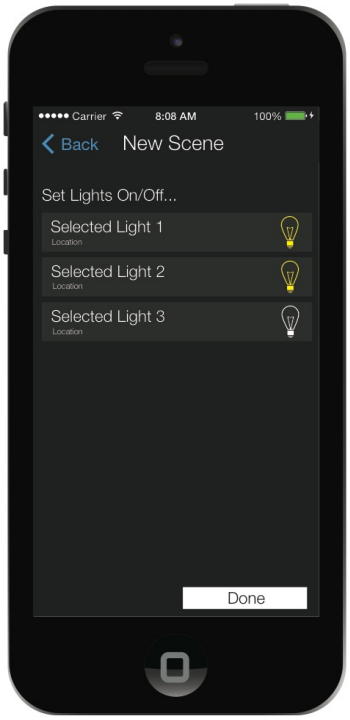
Swiping to edit brings the user to a similar view, but with 'Edit Scene' as the title and vaules already filled.

New Scene View
<NewSceneViewController>

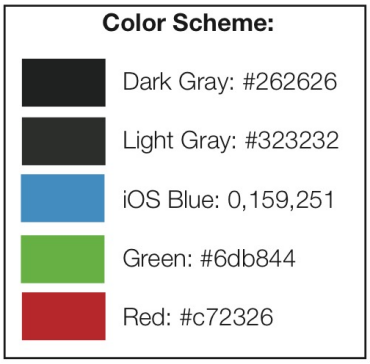


After hitting next, the user picks the state that the scene should cause the selected lights to go into. Once done, the user clicks "Done" and the scene is saved.

New Scene View 2
<NewScene2ViewController>



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