



HomeSense

Cheng Hao Chuang - Brandon Johnson - Jared Jones - Michael Austin Kidd

Paper Prototype, Testing, and Refinement Report

CSE 440

Spring 2011

Roles

Group Manager: Jared Jones

Design: Cheng Hao Chuang

Documentation: Brandon Johnson

Testing: Michael Austin Kidd

Problem and Solution Overview

When you're in a rush to get out the door, it can be inconvenient to have to run around the house checking that everything is off and locked. Sometimes you might not even have the time to check your entire house. With HomeSense, the user can quickly determine the state of many critical areas around their home – including windows, doors, and appliances – with the press of a button and access that information later at their leisure.

Tasks – 3 representative tasks of the design

Easy Task: Verifying that the house is ready for you to leave

You're in a hurry to get to work, but as you prepare to leave you realize that you don't remember if you closed the window in your bedroom. Pressing "Ready->To Leave" on the HomeSense console quickly tells you that not only is the window still open, but you forgot to turn the oven off after making breakfast! As you correct these things, they disappear from the list, and you can leave the house secure in the knowledge that you didn't forget anything else.

Medium Task: Checking the state of a particular item in the house

You just got home from the store, but as you set the groceries down inside you wonder whether you remembered to close the door of your detached garage after parking. It's raining, and you don't want to go all the way down the dark, wet drive to check manually. Using the HomeSense console, you go to the "Garage" floorplan then press the "Door" label, which brings up a display telling you that the garage door is in fact closed.

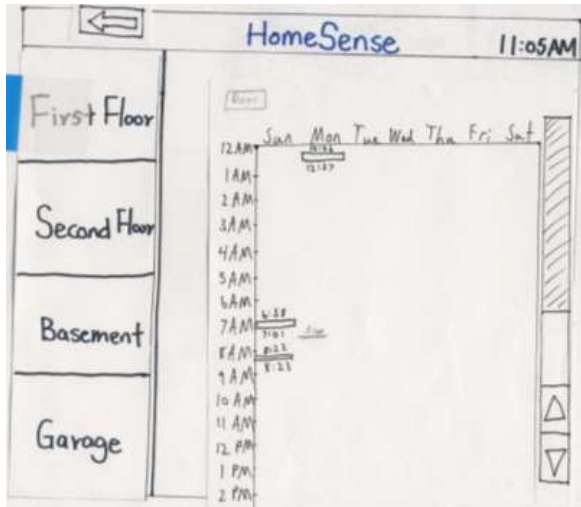
Hard Task: Using the log of previous states of an item in the house

Your son wants to stay out late Sunday night, but you tell him he has to be home by midnight because of school the next day. You have to wake up especially early Monday morning, so you'd rather not stay up until midnight to confirm his arrival. Instead, when you wake up, you go to the HomeSense console and bring up the status screen for the front door. Pressing the "History" button shows you a textual log of the previous openings and closings of the door, showing you that your child got home at around 12:15AM the night before; he was slightly late for his curfew.

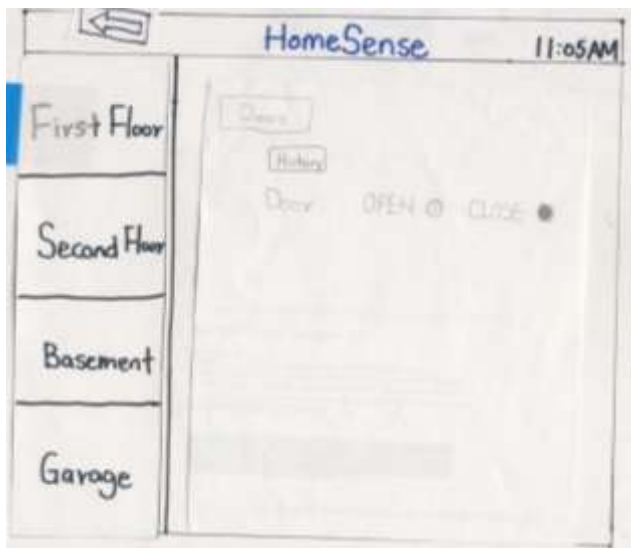
When we first determined these tasks we decided to assign them difficulty based not on how hard they were to complete without our device, but rather how hard, with relation to each other, we wanted them to be to accomplish through our application. We saw the first task as the most common and the one most likely to be performed under rushed circumstances, so we made accomplishing it as transparent and fast as possible with the giant ready button in the bottom right corner of the screen. Checking the current status of a single item is correspondingly one button press shallower to access than the history of that item.

Interface Revision Sketches

Before user testing, the interface for viewing sensor logs was through a calendar display. We liked this interface because it seemed like it would be more aesthetically pleasing and quicker to use than a text-based record of events. However, user testing showed that our graphical, calendar view was confusing and difficult to read for the sort of information the user would want to find. Our task required the user to see when the door was opened overnight, which proved to be difficult to find on a calendar. The text log displays the changes in state (door opening or closing, oven being turned on or off) in a clear and sequential manner, as below.



Our next change was simply to make better use of the space we had available to us in order to provide more descriptive labels. In particular, we changed the radio boxes showing the state of a sensor into unambiguous full-sentence descriptions of the state of the item. We originally chose radio boxes because only one radio box can be selected, which makes sense with the binary states of the items in the house. Unfortunately, radio boxes afford a feature that is not encompassed by our system: the ability to change the state. Users thought that they could select the unchecked box to change the state, which is something we did not anticipate, and is not supported by our system. Using complete sentences to describe the state of the system avoids the confusion that the “On” and “Off” labels gave.



Prototype Overview

Overview of implementation

A cursory glance of our interactive prototype will reveal that we decided to use an iPad blind to border our interface. This might be surprising given that previously, we referred to the device as being wall-mounted in the manner of a security system, but we wished to emphasize that the information could be available in a more portable manner if desired, so that the user could access it while working at their computer or lying in bed.

We implemented a basic fixed-path prototype to allow users to step through each of our three tasks, exploring representative routes through the interface at the prompting of the script to the right of the device. A back button allows the user to reset the current task if they get lost or confused, but the script should make it clear how to proceed.

In this iteration of our design, by contrast with our paper prototype, we were able to experiment with the aesthetic specifics. For instance, we kept to a simple sans-serif font throughout the interface, and used our standard blue in most cases to contrast with the bright red of the problem item list that appears when you use the ready button feature and you've left a window or door open. Additionally, dark blue highlighting indicates which item on the column on the left is currently selected.

As in our paper prototype, this area on the left serves two purposes. In the normal view, it gives a list of the floors, while after using the ready button, it lists the problem items around the house along with their current state (as in "Oven: on"). The area on the right displays the information available at the current level, allowing the user to click deeper into the house's hierarchy to view specific rooms, the status of the sensors in a given room, and the log history of those sensors.

Storyboards for 3 task scenarios

First Task



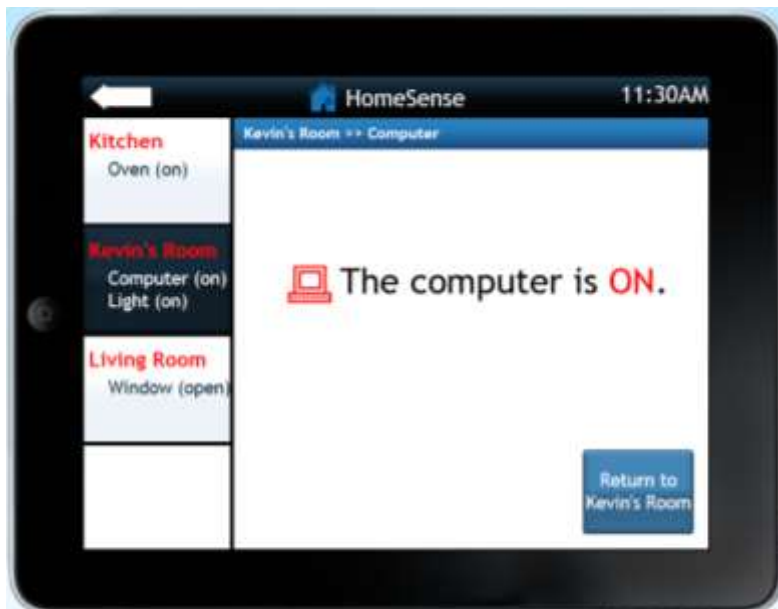
This is the first screen users see when using HomeSense. On this and every screen, the Ready button is accessible in the bottom right corner.



After the user touches the Ready button, the button is replaced with the available Ready States. For this task, the user selected "To Leave" because they're checking if the house is ready for them to leave.



After selected “To Leave” from the previous screen, the user interface changes slightly. Left pane now shows a list of all not-ready items and the rooms that contain them. Selecting a room from the pane jumps the view to floorplan of that room.



Selecting the icon for the specific room that is not ready brings the user to a detailed page for the item.

Second Task



From the start screen, the user can select Garage from the left pane to jump to the garage floorplan. Because the garage is both a floor and a room, sensors are immediately visible instead of rooms.

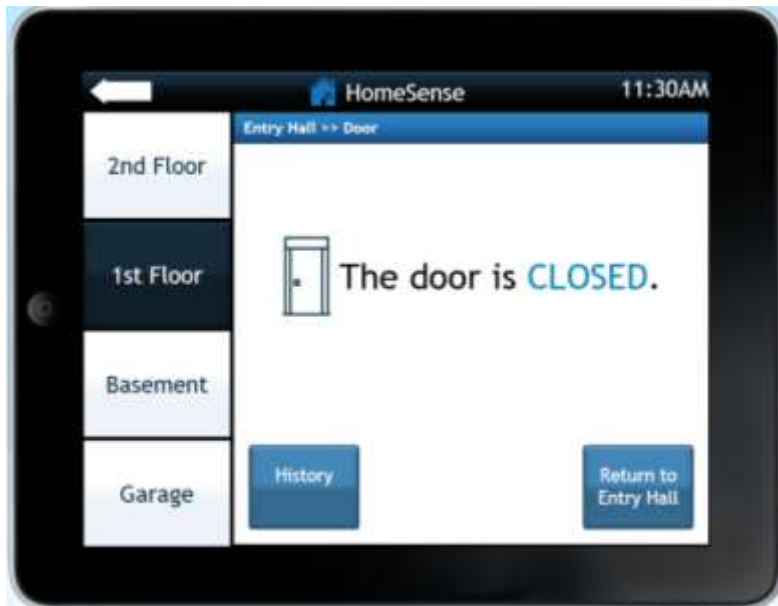


After jumping to the garage level, touching the door icon brings up the current state of the door.

Third Task



From the start screen, selecting the Entry Hall brings up the detailed view of that room.



Touching the door icon brings up the door's detailed information.



From the detailed view of the front door, the user can touch the History button, which brings the view to a detailed log of the recent front door activity.

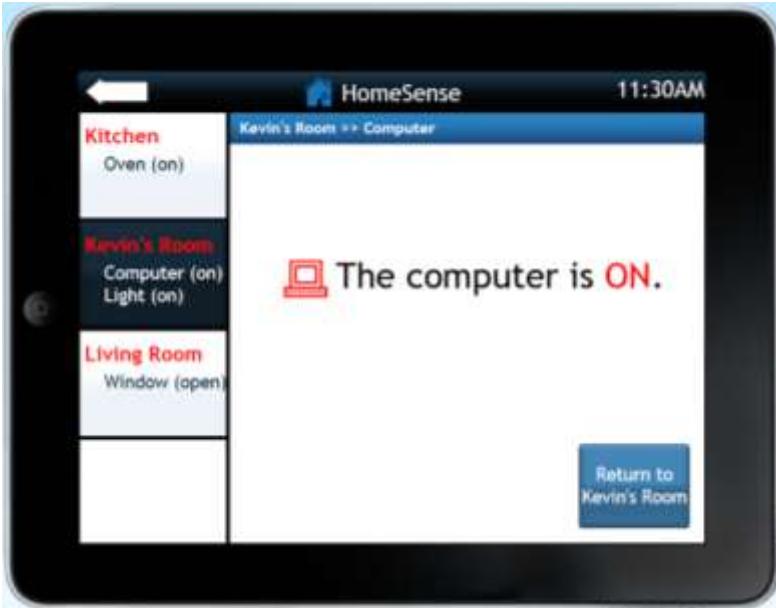
Description of tools used

Because of our familiarity with coding directly in HTML/CSS and JavaScript, it was more time effective to approach the development of our prototype without the use of any particular interface tools. We obtained permission from the professor before doing so. That said, we used a simple text editor for the code and various image software (i.e. Photoshop, Adobe Illustrator, etc.) to create our interactive prototype. All figures of the prototype in this report are screenshots of the web interface pasted into the document.

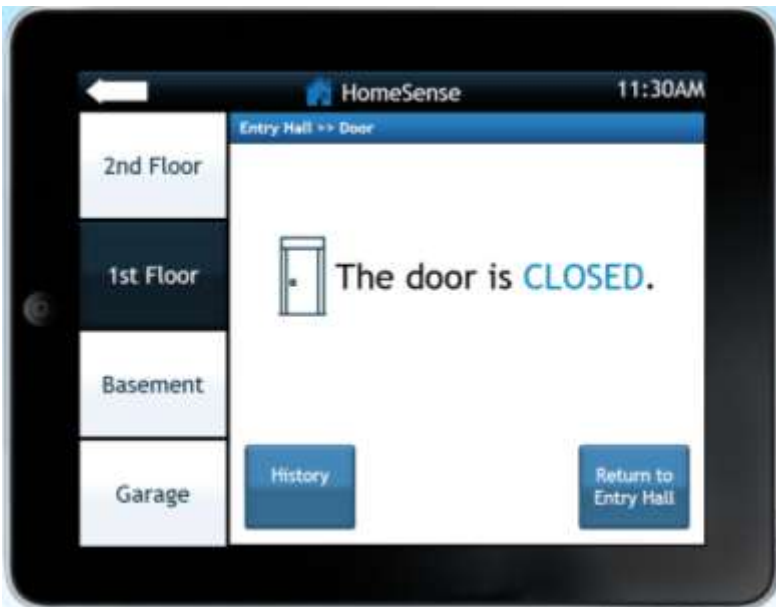
What was left out of the prototype and why

A full implementation of our application would allow the user to not only use the ready button to check the status of the house but also to peruse the current status and history log of each and every sensed item in the house, in the manner of our medium and hard tasks. Due to time and complexity limitations we were not able to complete such a full implementation, but we feel that the fixed-paths storyboarded by the tasks in our interactive prototype adequately portray the functionality available to the user to its deepest level.

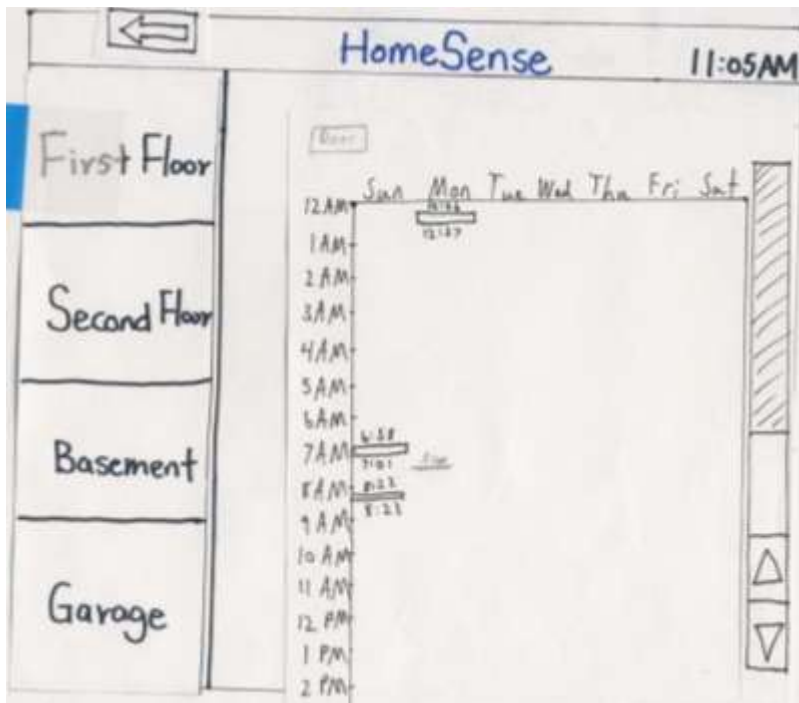
To be clear, the screens and functionality that we decided not to implement are just the same screens and functionality as those presented in the second and third tasks, except for the rest of the floors, rooms, and sensors in the house, and we didn't believe that such redundancy was necessary to show what the application is capable of.



The computer details page (above) from the first task is missing the History button, which is available from the front door details page (below). The implementation would have such functionality on both screens; the functionality was omitted from the demo to focus time and resources.



Further, the calendar history interface was not included in the interactive prototype. This decision was made because it offered no new functionality over the text log. A full implementation of this system would most likely have a calendar interface for the history.



A paper prototype of the calendar view that is not included in the interactive prototype.

Nothing special is going on behind the scenes, except that the purported sensors and the doors and windows they're monitoring don't actually exist, so we've supplied a filler floorplan and data to show what the interface would show for the given house. For the sake of displaying the interface, the users pretend that the given floorplan is of their own house.