Standby Home

Final Report

Lin Han 997553981
MingFei Wang 997463855
Cole Wheeler 997731450

Word count: 2490

The team requests that code created for the project is not shared.

Introduction

The heating and lighting systems in homes consume large amounts of energy [1] but are not of use for the often-lengthy periods that residents are away from their home [2]. Shutting these systems down manually every day is inconvenient for residence and will likely not be done [3]. The most common solution is programmable thermostats that allow users to manually enter periods when the heating system will turn off. However, only 45% of homes with these thermostats make use of them [4] as:

1. Having to constantly update them is inconvenient [5,6]

2. They result in users returning to a cold house if they arrive earlier than they are supposed to [7]

Newer solutions such as Nest and Honeywell's Wi-Fi Smart Thermostat solve the first problem by automatically learning the users schedule through monitoring their usage patterns but do not address the second [8,9]. Ecobee uses motion sensors to turn on the heat immediately when a user returns unexpectedly but the house will still be cold initially [10]. Honeywell's Lyric thermostat can be set to automatically turn on the heating when you are within either 7miles or 500ft range [11], but while the first will rarely give sufficient time for the heating to return the second will rarely allow the temperature to be turned down for many users.

For this project, the team will design "Standby Home", an android application that will use additional inputs such as GPS to solve this problem. It will both:

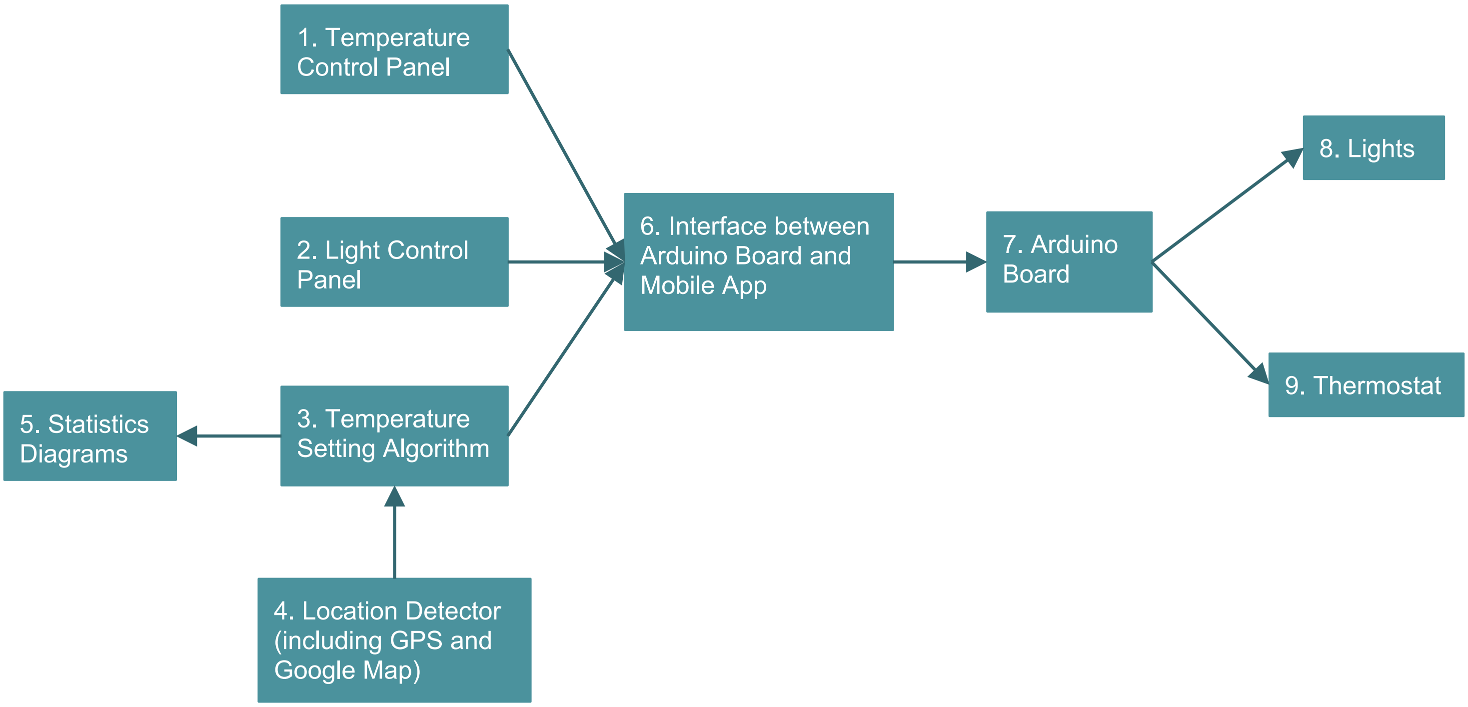
1. Automatically turning down the heating and lighting in the house as far as possible, without the need for any user input.
2. Do so without risking the user being inconvenienced or making them feel as if they have lost control.

These two objectives conflict as setting the temperature low enough to save energy will make it difficult to bring it back to room temperature before the user returns. Standby Home thus uses a Temperature Setting Algorithm to determine when it is safe to turn down the systems and by how much. It uses information such as the users location to estimate variables such as how far away from home they are, or how close to work they are, and in turn uses this information to set an appropriate temperature.

As well as through the algorithm, attempts were made to keep users happy by giving them very easy and accessible control over the systems, and insight into how the app is working. Together this maximizes the apps adoption by ensuring users see value in it and are not confused or frustrated by it.

2. Overall Design

The simplified block diagram below shows the main components of the app and its accompanying hardware. These components are described in more detail below.

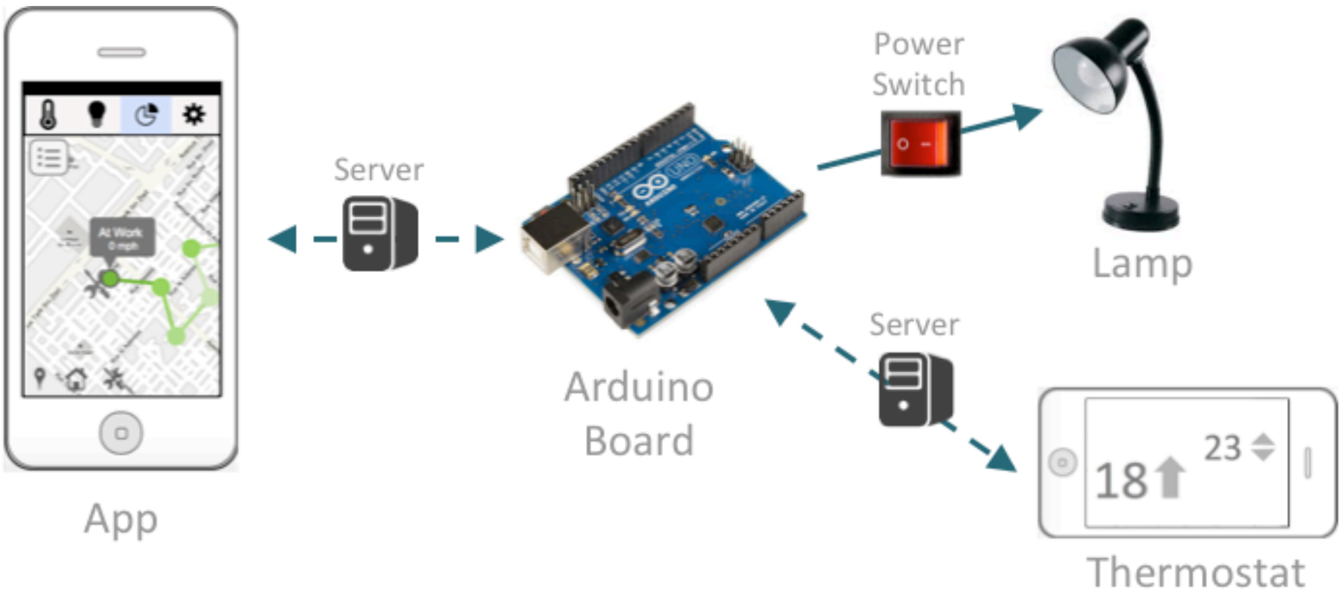


1. Interface used to set the target temperature when the user gets home and current home temperature
2. Interface that allows users can manually turn on or off the lights.
3. Code that runs in the background and calculates the correct temperature and light setting based on inputs such as current location
4. Code used to gather location information. It includes GPS and Google Map.
5. Code that uses past temperatures to display temperature curve based on time
6. This communication protocol connects the mobile phone and arduino board
7. A programmable development board, used to simulate the house control system
8. Two physical lamps used to simulate the kitchen light, and the living room light
9. Separate app running on another phone that displays basic temperature information

3. Functionality Overview

3.1 Hardware

Hardware was used simulate the control system, lights and heating system, that the app would be connected to if implemented in a house. This section describes these components in detail.



Main App Connection

The app interacts with the hardware through an Arduino Yun board. This connection is facilitated by a router that is configured to assign a static IP address to the board's MAC address and forward any requests to the board. Thus the board is globally accessible when connected to any publicly accessible network.

Arduino board

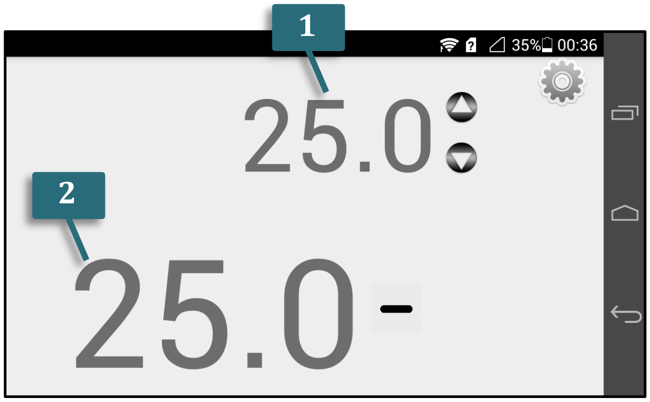
The software run on the Arduino board listens to and executes the http commands sent by the main app. In addition, it simulates the current temperature in the home by increasing or decreasing it at a fixed rate towards the temperature determined by the main app.

Lamps

The two standard lamps are used to simulate the lights in the home and are each connected to the Arduino board through a PowerSwitch, which has a relay circuit built-in. The PowerSwitch is turned on by supplying a constant voltage to its “+” terminal through the output pins on the Arduino board, which in turn switches on the light.

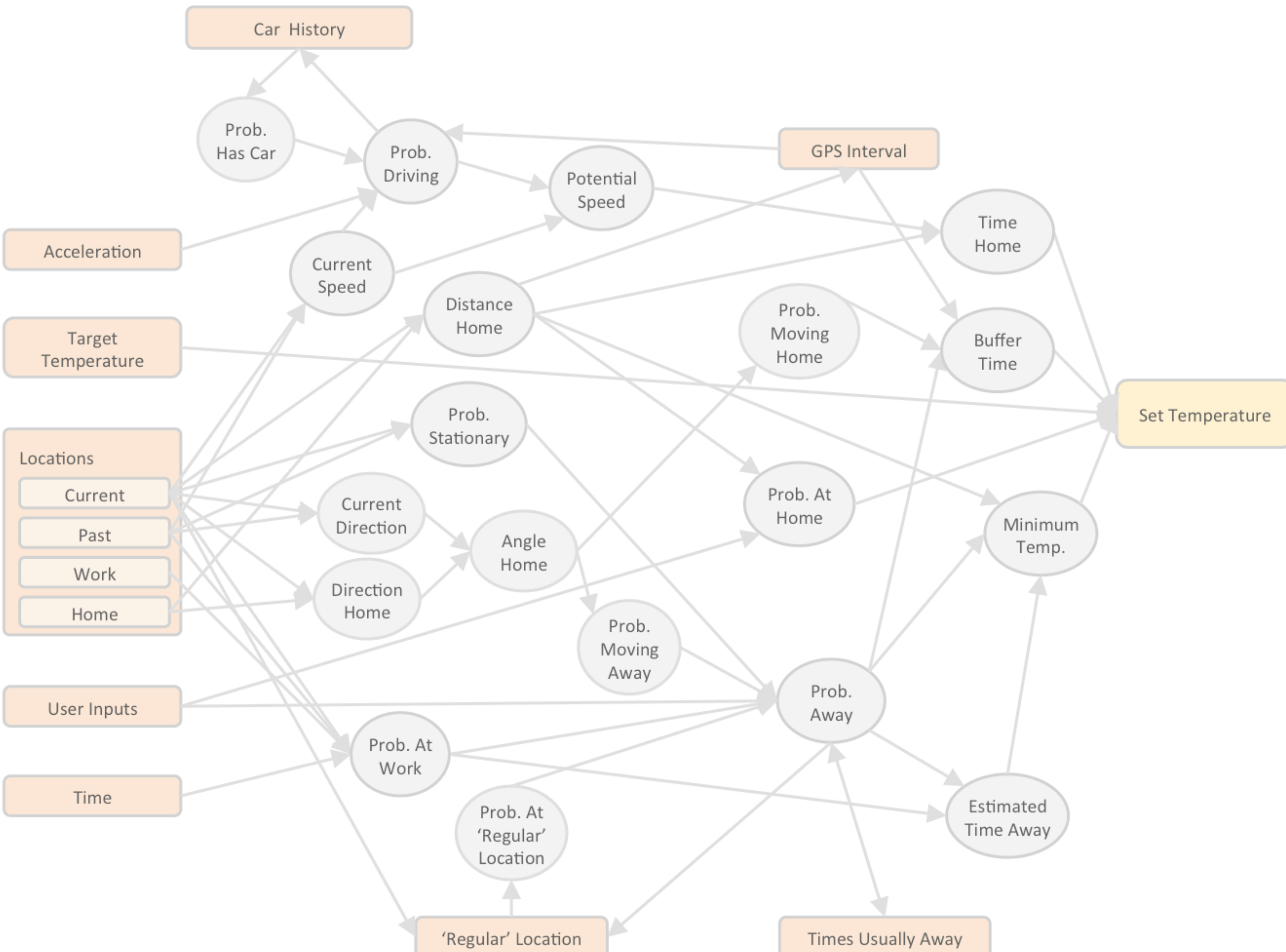
Thermostat App

The thermostat app (see below) simulates a regular thermostat and displays the “thermostat” temperature (**1**) and current temperature (**2**). It is connected to the Arduino board in the same way as the main app and can receive the values of both the current and set temperature, as well as send back any changes made to the set temperature.



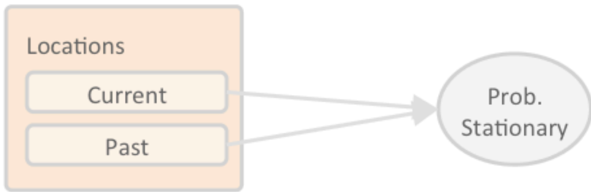
3.2 Temperature Setting Algorithm

The Temperature Setting Algorithm determines how low a temperature it is safe to set and is a key part of the app (see introduction). The simplified overview diagram shown below shows the inputs (shown in orange) it uses to determine a number of parameters (grey), which eventually lead to a certain temperature being set as the output (yellow). To illustrate how this overview translates into the logic in the app, two examples of this are provided below.

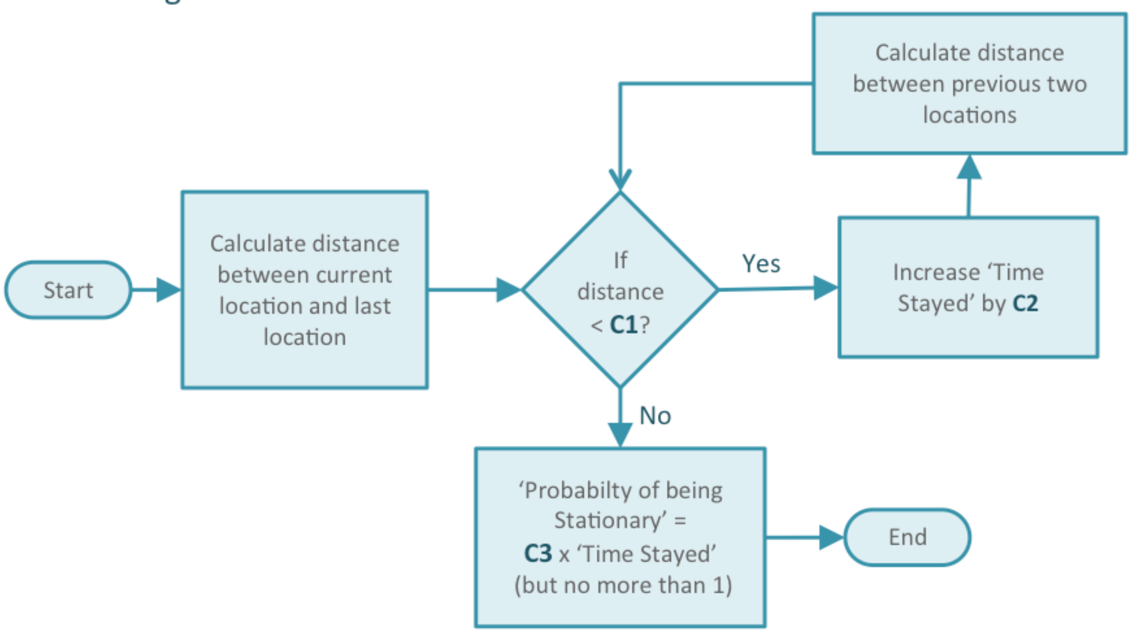

Car History
Prob.
Has Car
Prob.
Driving
GPS Interval
Potential
Speed
Acceleration
Current
Speed
Time
Home
Target
Temperature
Distance
Home
Prob.
Moving
Home
Buffer
Time
Prob.
Stationary
Set Temperature
Locations
Current
Past
Work
Home
Current
Direction
Angle
Home
Direction
Home
Prob. At
Home
Minimum
Temp.
Prob.
Moving
Away
User Inputs
Prob.
Away
Time
Prob. At
Work
Estimated
Time Away
Prob. At
'Regular'
Location
'Regular' Location
Times Usually Away

Probability of being Stationary

Section of Overview Diagram



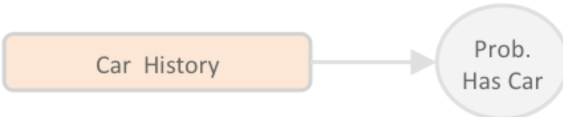
Detailed Diagram



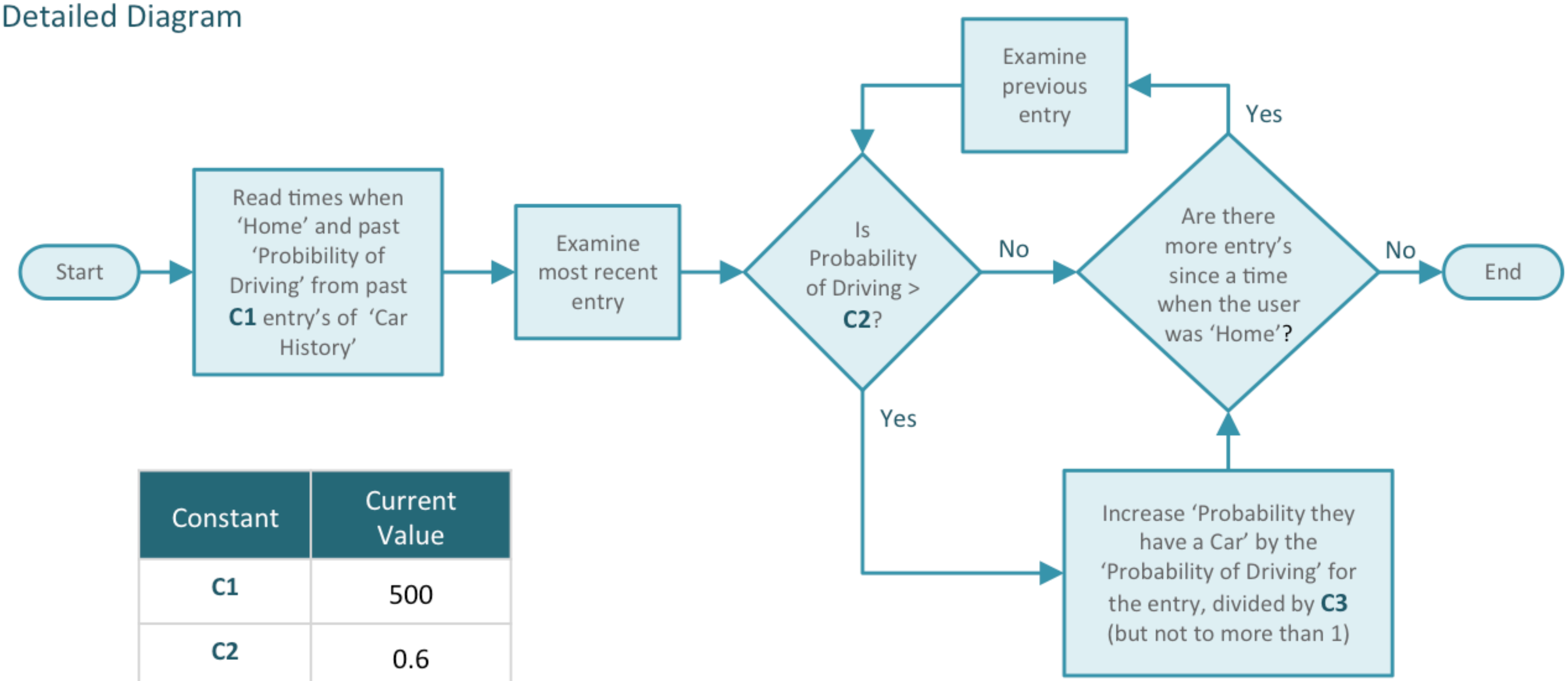
Constant	Current Value
C1	0.2 km
C2	'GPS Interval'
C3	0.01

Probability of being Stationary

Section of Overview Diagram



Detailed Diagram



Constant	Current Value
C1	500
C2	0.6
C3	50

3.3 Interface overview

The following section gives an overview of the interface and describes its functionality.

Temperature Tab

This allows the user to manual control the temperature and get insight into how it is currently behaving, thus minimizing their feeling of lost control. It is the first screen seen when opening the app. displays:

1. The "thermostat" temperature that the house should be at when the residents are at home and allows control over this

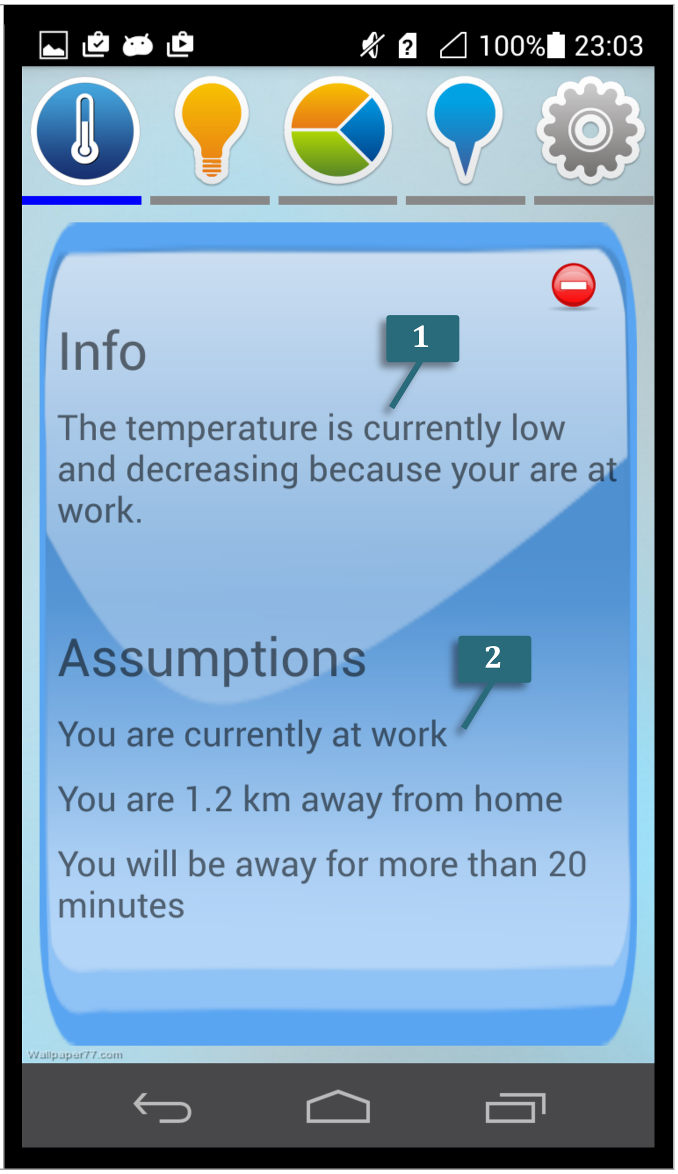
2. The current temperature along with its direction, both read from the Ardiuno board. This value determines the yellow bars height and moves with it.

3. Buttons that the user can select to inform the Temperature Setting Algorithm of there status. While the former will immediately begin increasing the temperature, the latter will usually allow a lower temperature to be set.

Information Panel

This panel gives further insight into how and why the temperature is changing. It is displayed by pressing the “i" icon in the temperature tab and displays the following text determined by the Temperature Setting Algorithm:

1. A description of the current temperature, its direction and the main reason for its direction
2. A list of the assumptions that the app is currently making



Lighting Tab

This tab allows easy control over the lights both to change which are turned on while at home and to override the automation. It displays:

1. The server address of the Arduino board (for debugging and would remove in final app)
2. A switch that will turn on and off all lights connected to the Arduino board
3. A list of lights that each can turn on and off a light connected to the Arduino board

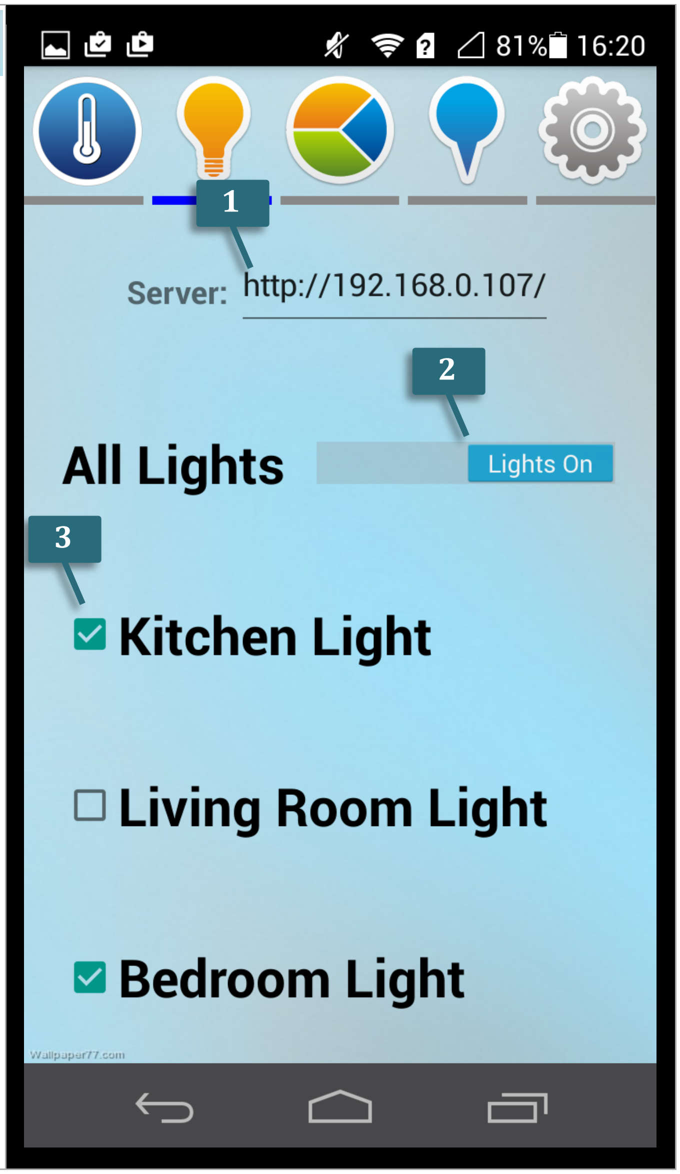
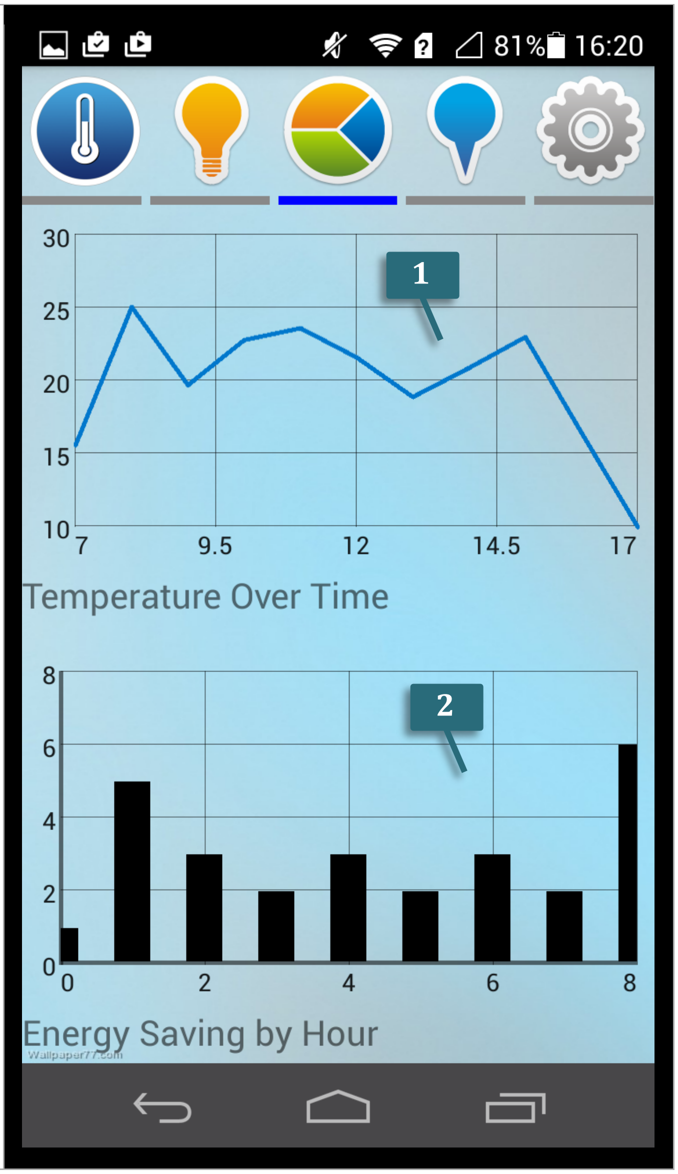


Chart Panel

This tab gives the user insight into how the temperature has been set and encourages them to use less energy by showing savings information. It displays:

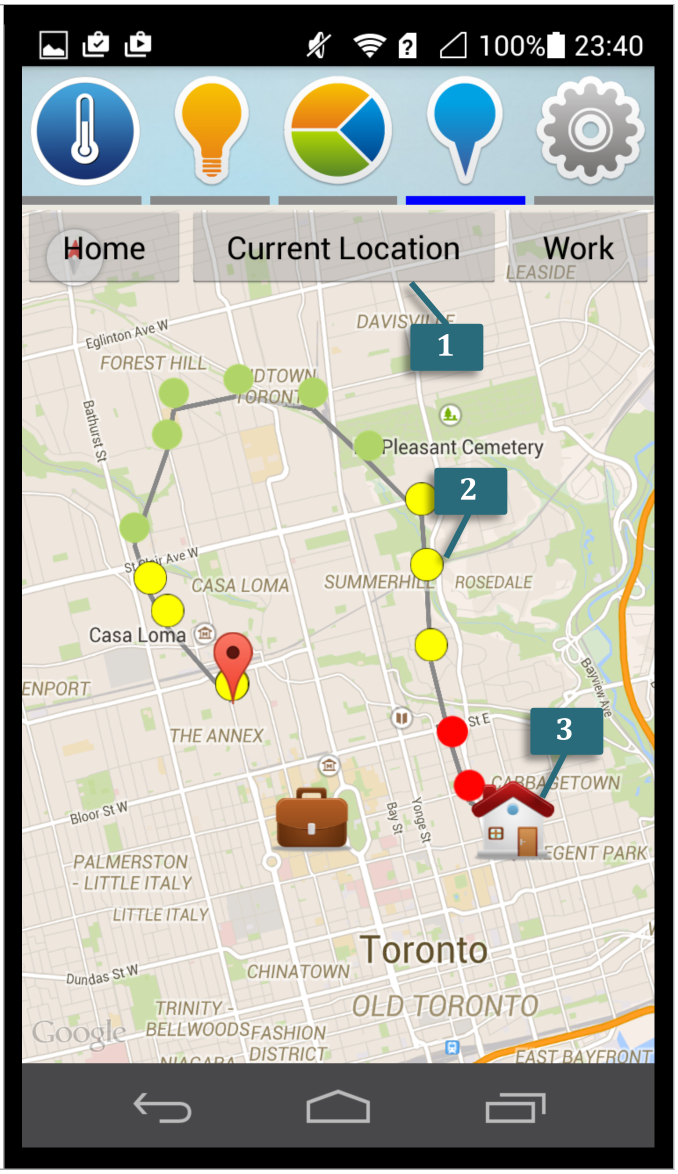
1. A line graph of temperature vs time generated from past data. Scrolling to the left on the graph reveals more data
2. A bar graph of energy savings by hour



Map Tab

This tab gives the user insight into how the temperature is set by showing how it changes based on location. It also allows them to check that their current, home and work locations are correct and adjust the current location if necessary. It displays:

1. Buttons which navigate to their respective locations
2. Points representing the 20 most recent locations the user has been, connected by lines and colour coded according to temperature (red high to green low). Tapping anywhere on the map changes the current location to that location
3. A marker for the current, home and work locations



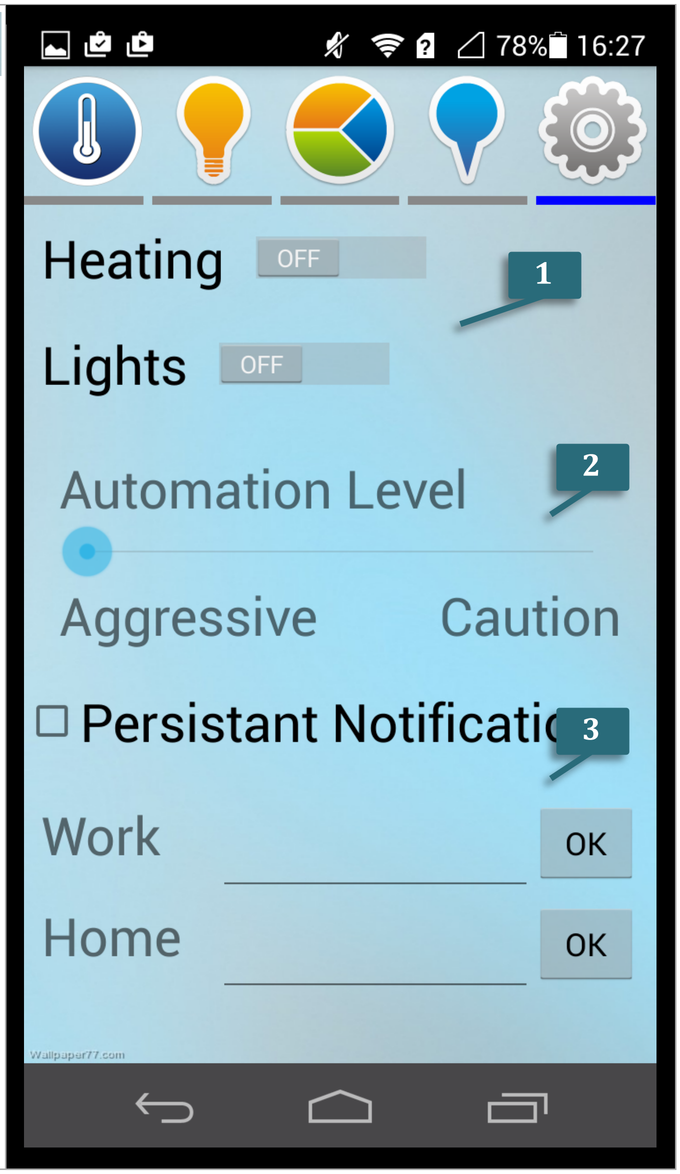
Setting Tab

This tab allows the user to easily control and turn off the automation, ensuring that they retain a feeling of control. It displays:

1. "On/Off" button for heating and lighting that stops them from changing automatically

2. Automation level which controls how aggressively low of a temperature the app will set (i.e. setting to caution would result in generally higher temperatures than aggressive)

3. Settings to turn on and off persistent notification showing temperature and enter the work and home locations. These features are not currently functional.



3.4 Uncompleted Features

A number of proposed features were not completed due to the time constraints of the project and the resulting need to focus on the app's most important features:

- Calendar panel - this would allowed the user to select future dates that they were going to be away, making turning it off while on vacation easier.
- Settings – If fully implemented the last three items in the settings tab would allow the work and home locations to be inputted by the user and allow a persistent notifications showing the current temperature to be turned on, to keep them informed of changes in the app.

Several other parts of the app did not work when implemented:

- Incorrect assumptions – buttons on the information panel would have allowed the user to mark each assumption as false, in order to allow the algorithm to react accordingly. Feeding this information back into the algorithm proved to be too difficult, though this could have been possible had the algorithm code been designed to consider this from the start.
- Algorithm functioning close to home – it was hoped that by monitoring whether a user was staying in one place the algorithm could predict their intent with sufficient precision to allow temperature decreases even when close to home. Unfortunately, through testing it became apparent that it was almost never safe to reduce the temperature within 0.5km of home. This would result in less savings for users who frequently stay at places close the their home.

4. Lessons Learned

The following are the key lessons learned throughout this project:

- Priority should have been placed on completing interfaces such as the map view which could have made earlier testing of the Temperature Setting Algorithm greatly simplified. In future these interdependencies would be more carefully considered.

- The three members coded separately and only tested compiled versions of the app once per week. These long periods between integrated testing resulted in difficulty debugging every week and in future more frequent integration would be done.

- A number of times errors arose from different members having developed different understandings of how parts of the app should function. Clear documentation of each function would help ensure the teams understanding was consistent.

- With the limited timeline and resources available on the project completing all proposed functions proved to be challenging. In future the team would focus on the main aspect of the app, the temperature control, instead of also working on the lighting control.

6. Contribution by Group Members

This section outlines the contributions of each group member to the project.

L – Led work: took a leading role for this task and completed some if not all of the work

A – Assisted with work: contributed to the task and assisted person leading it

Task	Ming Fei	Cole	Lin
Design Work			
Designing app and defining scope of functions	A	L	A
Creating mockups of app		L	
Coordinating team and defining deliverables	A	L	A
Hardware			
Arduino board setup and configuration	L		
Router configuration and http protocol definition	L		
Arduino board programming	L		
Arduino board connection to other hardware	L		
Hardware to software integration	L		A
Coding thermostat application and linking to hardware	L		
Temperature Setting Algorithm			
Algorithm design		L	
Algorithm coding		L Text	
Testing and improvement		L	
General Coding			
Initial coding of navigation			L

Integration with algorithm	A	A	L
Design and coding of basic page layouts			L
Structure and navigation refinement	L		A
App code integration	A		L
Temperature Tab			
Layout design and initial coding		A	L
Temperature simulation	A		L
User controls	A		L
Information Panel		A	L
Temperature graphic implementation	L		A
Lighting Tab			
Layout design and coding			L
Charts Tab			
Chart creation	L		A
Integrating charts and reading data	A		L
Map Tab			
Google maps setup and integration			L
Location markers and buttons	A		L
Current location detection	A		L
Point and line plotting			L
Setting Tab			
Coding and integration	L		A
Project Deliverables			
Project proposal	A	L	A
Presentation slides	A	L	A
Final Report	A	L	A

7. Apper Context

While the Apper was an industrial Engineering by training, their role for the past three years as a leading member of the Toronto Solar Decathlon Team was used as the original impetuous for this project. The Decathlon team consists of students and faculty from four universities and colleges. Their goal is to design and build an innovative net-zero energy house to compete in the Solar Decathlon competition and they are currently in the process of developing the detailed design. The house will be a semi-automated smart home and as part of this the user will control the home via a mobile application. The app created in this course essentially acts as a prototype for this final app and has allowed some of its key features to be tested.

One of the most important contributions of this project is in providing a proof of concept for the automated home shutdown function and will likely result in the Decathlon Team choosing to include this as part of the final app they create. The current app will then be used as the start of future iterations, which will focus on other features such as control and informing users of their energy use in the home. The app can be used by the team, along with other aspects of the homes design, to market the house as a truly smart home that carefully balances automation and user experience.

Outside of the Decathlon project the success of this app also have repercussions for the smart-home industry at large. As the industry continues to grow in market, the inclusion of simple and effective ways to save energy will likely be increasingly important. If it functions correctly, as this project has shown that it can, the automated home shutdown feature has the potential to save a substantial amount of energy and is thus a prime addition to future smart-home systems.

8. Future Work

If this project is continued the functions described in the Uncompleted Functions section would first be added. Following this, the additional functions below would be added:

- Support for multiple users. This would allow all residents to control the systems through the app. While most of the work to enable this function is complete, minor adjustments and testing still need to be done.
- Work to connect the app to an actual control system in a house in order to conduct more thorough testing and potential pilot studies.
- Work to further improve the user interface with a particular focus on improving the readability of the charts, in order to ensure a positive user experience.
- Improve security of data transferred from the phone to the control system to ensure information such as the set temperature cannot be used to determine when the resident is away.

References

[1] Statistics Canada (2012, December 19). *Households and the Environment: Energy Use: Analysis*. Available from: http://www.statcan.gc.ca/pub/11-526-s/2010001/part-partie1-eng.htm

[2] Bureau of Labor Statistics (2014, June 18). *American Time Use Survey Summary.* Available from: http://www.bls.gov/news.release/atus.nr0.htm

[3] Therese Peffer, Marco Pritoni, Alan Meier, Cecilia Aragon, Daniel Perry (2011). *How people use thermostats in homes: A review.* Available from: http://eec.ucdavis.edu/files/How_people_use_thermostats_in_homes.pdf

[4] Energy Information Administration (EIA). *Residential energy consumption survey, housing characteristics tables, table HC14.5: space heating usage; Feb. 16, 2011.* Available from: http://www.eia.gov/emeu/recs/recs2005/hc2005_tables/detailed_tables2005.html

[5] Kempton W, Feuermann D, McGarity AE. *"I always turn it on super": user decisions about when and how to operate room air conditioners*. Energy and Buildings 1992;18:177-91.

[6] Diamond RC, Remus J, Vincent B (1996). *User satisfaction with innovative cooling retrofits in sacramento public housing.* Proceedings of the 1996 ACEEE summer study on energy efficiency in buildings.

[7] Linden A-L, Carlsson-Kanyama A, Eriksson B. *Efficient and inefficient aspects of residential energy behaviour: what are the policy instruments for change?* Energy Policy 2006;34:1918-27.

[8] Nest. *Nest: Thermostat.* Available from: https://nest.com/ca/thermostat/life-with-nest-thermostat/

[9] Honeywell. *Wi-Fi Smart Thermostat.* Available from: http://yourhome.honeywell.com/home/Products/Thermostats/7-Day-Programmable/Wi-Fi+Smart+Thermostat.htm

[10] Ecobee. *ecobee3: The smarter wi-fi thermostat with remote sensors.* Available from: https://www.ecobee.com/

[11] Honeywell. *Lyric.* Available from: http://lyric.honeywell.com/