Smartstat
Detailed Specifications

Milestone 2

CS 170/270 Software Engineering
March 9, 2010

Drew Mason
Joe Cancilla
Simon Fishel
Anna Fornaeus
Helen Tompkins
Kathyrene Villariba

Group Lead:
Drew Mason
amason@mills.edu

Revision History

Date	Version	Description
03/09/10	1	Submission for Milestone 2

1. Product Description

In theory, programmable thermostats are a great idea. The user programs in his or her daily schedule, and the thermostat automatically turns down the heat or air conditioning (i.e., turns the temperature down in the case of heating and up in the case of air conditioning) during periods of sleep or absence. Programmable thermostats are supposed to save money on utility bills and help the environment by making it easy to use heating and air-conditioning systems in a more energy-efficient way.

In practice, however, using a programmable thermostat is anything but easy. Setting up programs requires picking through tedious, confusing manuals and lists of codes—a process so confounding that some users end up having to call in HVAC specialists. The problem is so severe that programmable thermostats were removed from the U.S. Energy Star program as of December 31, 2009, largely due to concerns about the difficulty of using them properly.[1]

The main cause of the usability problem is easy to identify: these thermostats are trying to provide complex programming functionality through a limited on-device interface consisting of nothing more than a few buttons and a small screen. There's nothing inherently difficult about programming a thermostat; all it requires is an improved UI.

That's where the Smartstat comes in. The Smartstat will be web enabled, allowing it to be programmed via browser and smartphone applications that are easy to use and accessible from anywhere. These applications will feature an intuitive, user-friendly interface based on a well-known paradigm: the calendar. Constructing Smartstat programs will be similar to adding repeating appointments in Microsoft Outlook or Google Calendar—a concept with which most people are already familiar.

Homeowners and small businesses will likely find the Smartstat especially attractive. Whereas larger facilities may have complex HVAC systems controlled by facility managers, people who own homes or small businesses just want a single thermostat that they can program quickly and easily without having to consult a specialist. For such people, the Smartstat will provide an ideal solution.

2. Glossary

2.1 **HVAC:** An acronym for “heating, ventilation, and air conditioning.”

2.2 **Programmable Thermostat:** A device that enables the user to set one or more time periods each day when a comfort setpoint temperature is maintained and one or more time periods each day when an energy-saving setpoint temperature is maintained. This device enables the user to save energy because the heating and cooling equipment is not running needlessly at a comfort setpoint temperature 24 hours per day.

2.3 **Setpoint Temperature:** The temperature setting in degrees Fahrenheit or degrees Celsius that is maintained by the thermostat for a given setting period.

2.3.1 **Comfort Setpoint Temperature:** The setpoint temperature maintained when the user is present in the building. Generally further from the outside temperature—cooler in the case of air conditioning, warmer in the case of heat.

---

1 *Consumer Reports* Home & Garden Blog, “Programmable thermostats are no longer part of the federal Energy Star program,” June 2, 2009, http://blogs.consumerreports.org/home/2009/06/programmable-thermostats-energy-star-epa-save-energy-cooling-heating.html.

2.3.2 Energy-Saving Setpoint Temperature: The setpoint temperature maintained when the user is away. Generally closer to the outside temperature—warmer in the case of air conditioning, cooler in the case of heat.

2.4 Setting Period: A scheduled period of time in which the thermostat maintains a predefined setpoint temperature. A setting period can be part of the thermostat's default settings or programmed manually by the user. Common setting periods are *wake, day, evening,* and *sleep*.

2.5 Override: This feature enables the user to temporarily set the thermostat to a different temperature until the next setting period begins.

2.6 Anticipation: A feature in which the thermostat shuts off the furnace before the setpoint temperature is reached in order to keep from overshooting it.

2.7 Hysteresis: The process by which the thermostat maintains the environment temperature near a given setpoint by turning on the heat or air conditioning when the actual temperature moves too far from the setpoint in either direction.

2.8 Recovery: A process in which a thermostat gradually returns from the energy-saving setpoint temperature to the comfort setpoint temperature before the comfort setting period is scheduled to commence, so that the environment has reached has reached the comfort setpoint temperature by the beginning of the comfort setting period.

2.9 Setpoint Schedule: The schedule of all of the thermostat's setting periods and associated setpoint temperatures. The setpoint schedule determines the thermostat's setpoint temperature at any given time. Usually involves either daily or weekly repeating.

2.10 Default Setpoint Schedule: A predefined setpoint schedule that is automatically enabled without any user intervention.

2.11 Energy Star: A government-backed program helping businesses and individuals protect the environment through superior energy efficiency. Programmable thermostats are no longer part of the Energy Star program, but the Energy Star specifications nevertheless provide a useful set of guidelines for an energy-efficient thermostat.

2.12 Alerts: Emails or text messages automatically sent to the user to notify him of unexpected or exceptional behavior (e.g., hardware failures, loss of web connection, password change). See Section 4.3 for more details.

3. High-Level Requirements And Use Cases

The user shall be able to access and modify the Smartstat’s settings through three different interfaces: the device hardware, a web browser application, and an Android smartphone application. The controls on the hardware shall be limited to simple operations, whereas the browser and phone interfaces shall provide more powerful (but still easy to use) programming options. The functionality provided by each of these interfaces is described in more detail in section 4.

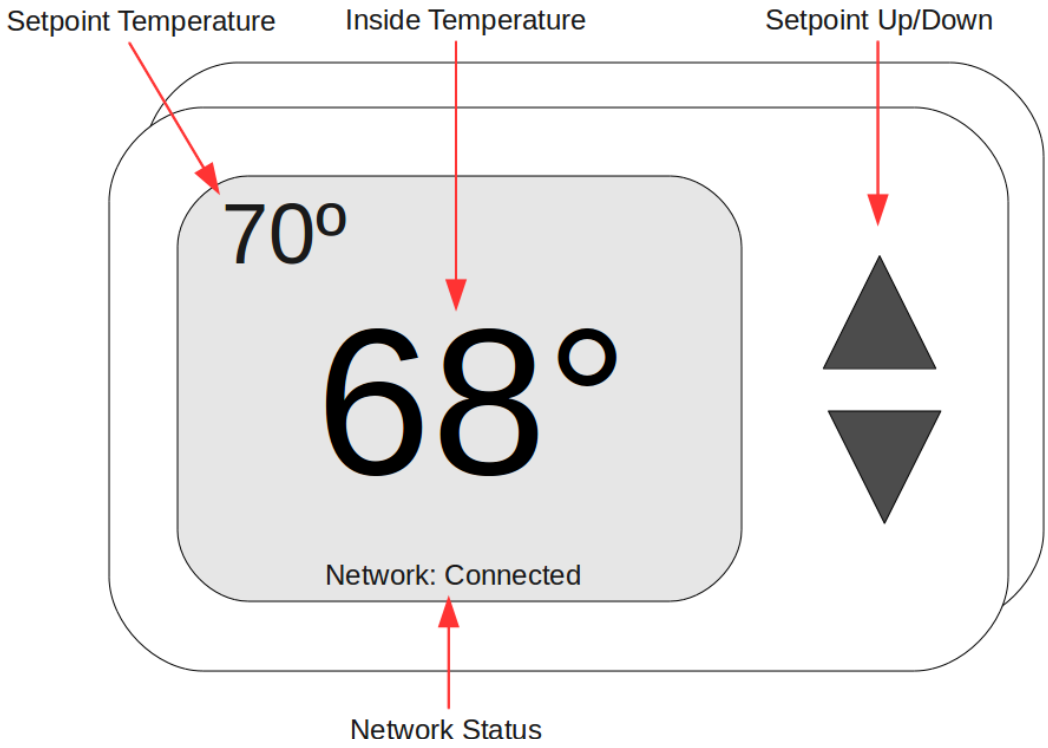
3.1 Device (Hardware) Interface

Use Case: Temporarily Change the Current Setpoint Temperature

- **Scenario:** Linda is feeling cold, so she walks over to the Smartstat and overrides the current setpoint temperature using the buttons. Even if Linda forgets that she made manual changes, no extra energy will be wasted—Linda's override will only be in

effect temporarily. The regular setpoint schedule will automatically take over again when the next setting period begins.

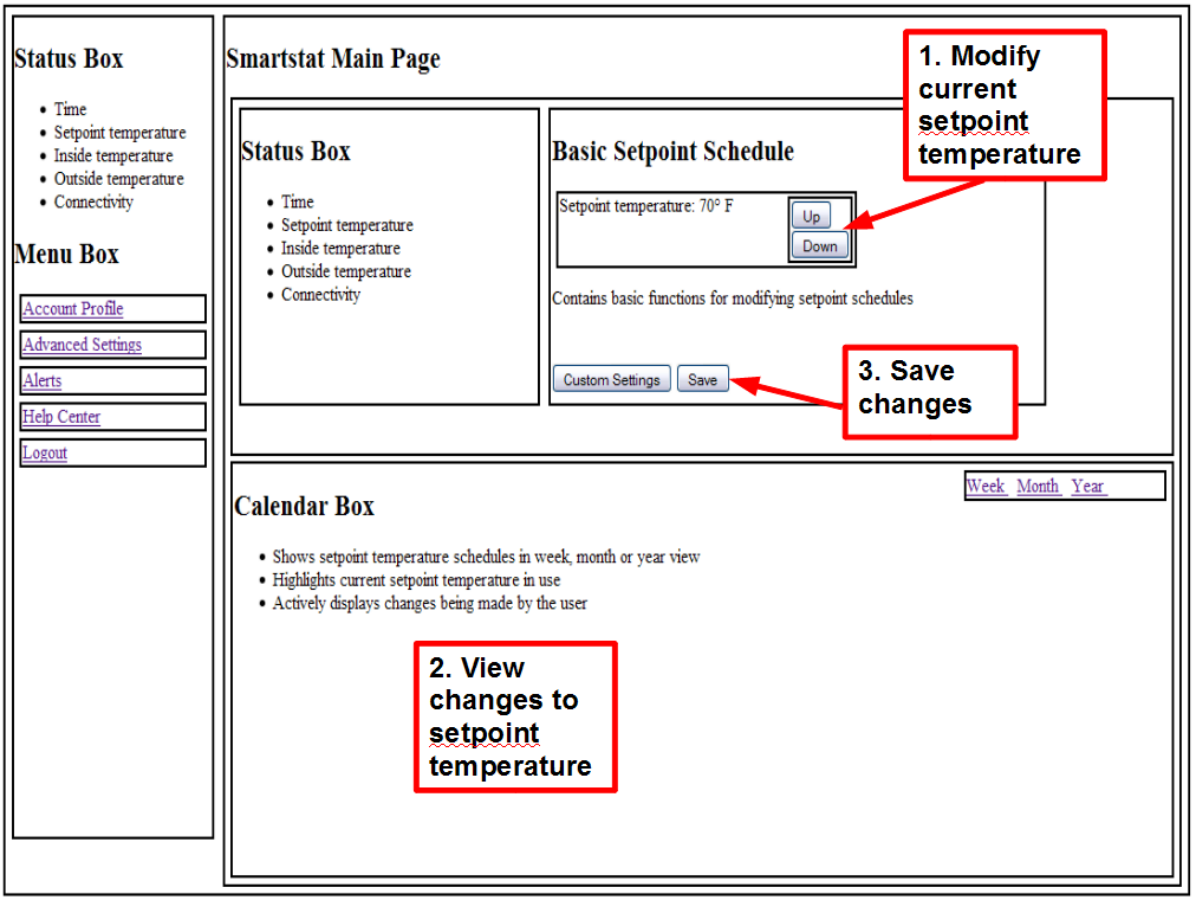
- **Procedure:**
  Linda uses the up and down arrows to adjust the setpoint temperature.



3.2 Browser Application

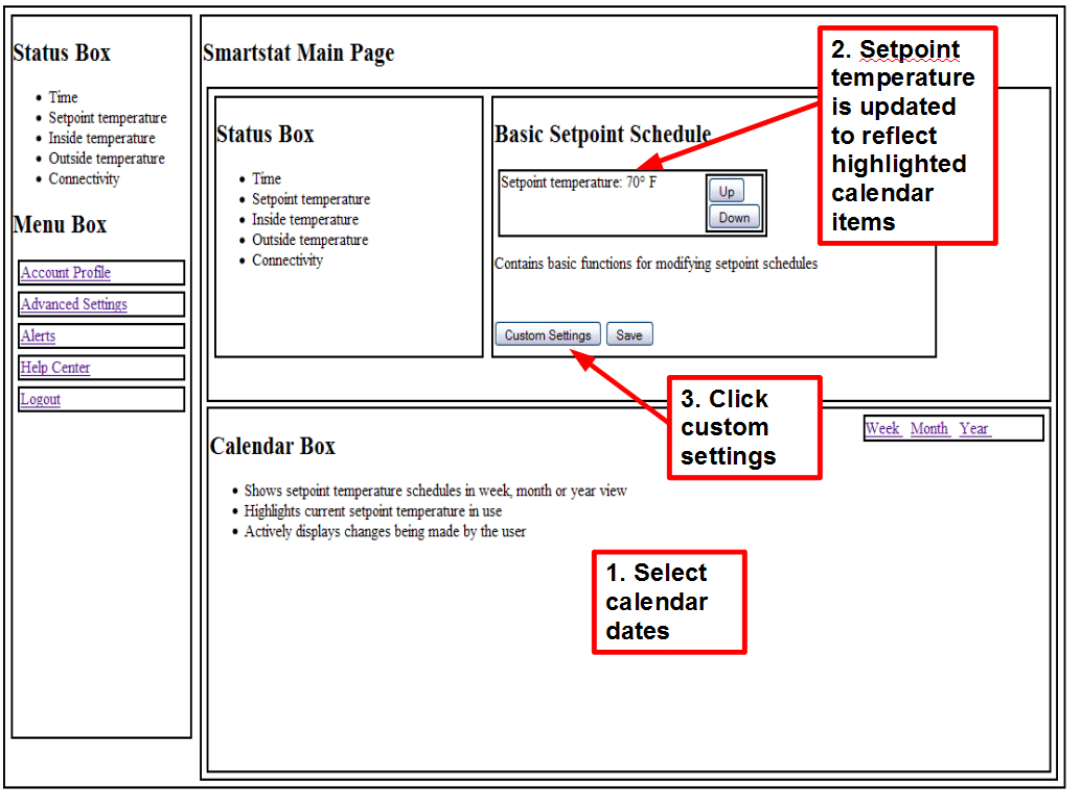
Use Case: Temporarily Change the Current Setpoint Temperature

- **Scenario:** Joe's teenaged children are on winter break and are hanging out at home unsupervised while he's at work. Joe wants to make sure they aren't taking advantage of his absence to crank the heat up, so he uses his computer at work to log in to his Smartstat account and check the house temperature and setpoint. If it's too hot, he'll turn it down.
- **Procedure:**
  - Start from the account home page. The current setpoint temperature shall be displayed in the two status boxes and basic setpoint schedule box. The current setpoint schedule in use shall be highlighted in the calendar box.
  - The user shall modify the current setpoint temperature by clicking on the Up or Down buttons in the basic settings box.
  - The calendar box shall actively display the changes the user has made to the current setpoint temperature.
  - The user shall click Save so that his changes will be reflected on the thermostat.

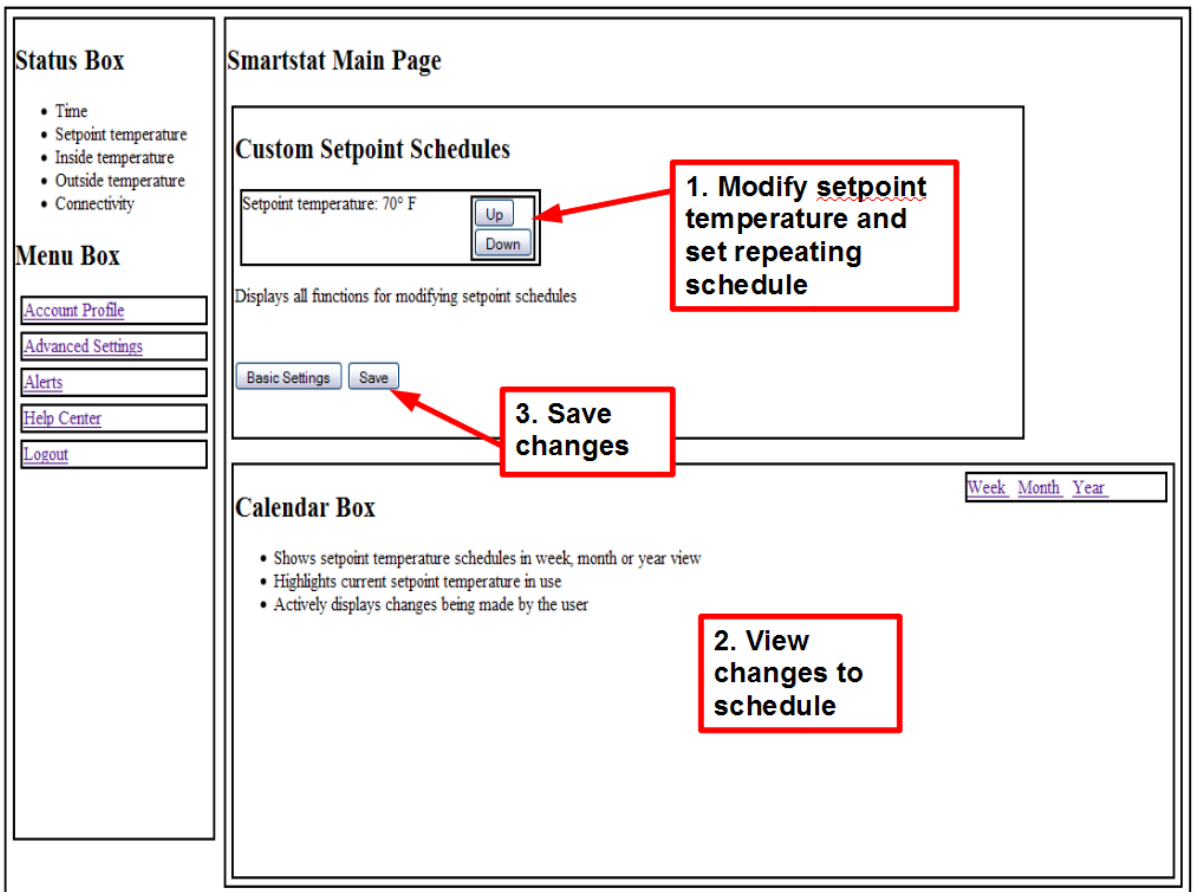


Use Case: Create/Modify a Setpoint Schedule

- **Scenario:** Kat's company is sending her to Hong Kong for six months, and she's going to let her brother Drew use her house while she's away. Drew, a bartender, works a completely different schedule from Kat. Kat and Drew go online and use the Smartstat's browser application to create a new setpoint schedule for Drew. When Drew moves in, he goes online and activates his setpoint schedule.
- **Procedure:**
  - Page 1: Start from the account home page.
  - Select a date or a range of dates in the calendar box.
  - The setpoint temperature in the basic settings box is updated to reflect the setpoint temperature scheduled for the date or range of dates that the user highlighted.
  - To set custom or repeating schedules, click the Custom Settings button.



- Page 2: Clicking the Custom Settings button will open the custom setpoint schedules box. The date or range of dates that the user selected will be highlighted in the calendar box.
- To set a new setpoint temperature, the user shall click the up or down buttons in the custom setpoint schedules box. The custom setpoint schedules box will also include additional functionality that will allow the user to set repeating schedules.
- The calendar box shall actively display the changes the user has made to the setpoint schedule.
- The user shall click Save so that his changes will be reflected on the thermostat.

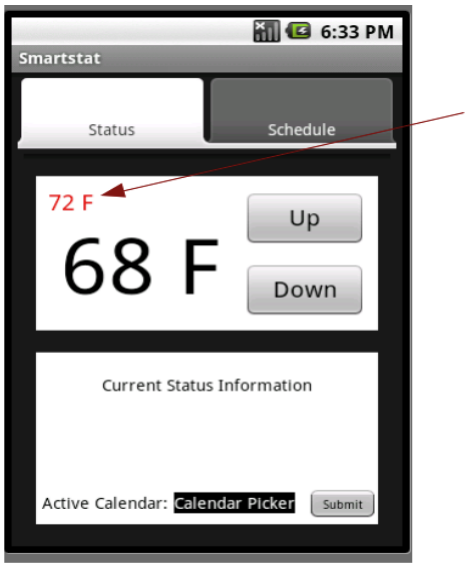


3.3 Android Application

Use Case: Temporarily Change the Current Setpoint Temperature

- **Scenario:** George is at work one winter afternoon when he gets a call from his daughter Ada's elementary school. She's come down with a severe flu and needs to be taken home immediately. As he walks to his car, he uses his phone to check the temperature in his house and turn it up so that Ada will be warm and toasty the moment she gets home.
- **Procedure:**
  - George loads the Smartstat application on his phone and is immediately brought to the status screen.
  - He can see the current temperature in his house, as well as the setpoint temperature.
  - He taps the Up button (right) to override the setpoint temperature to a higher value.

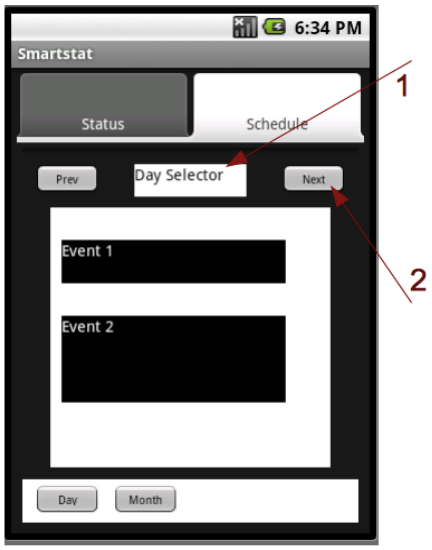
- As he changes the setpoint temperature, its color changes to red (right). Once it has not been changed for 2 seconds, the color will return to black, indicating that the override instruction has been sent to the Smartstat device.



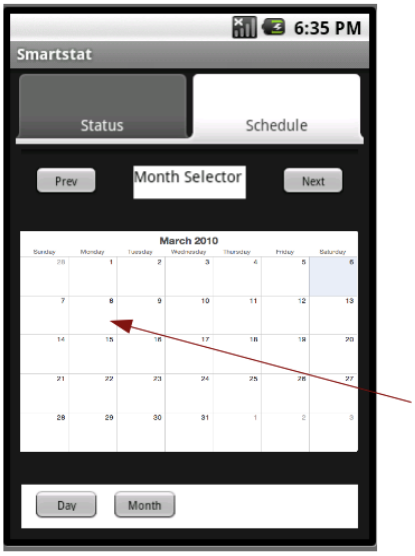
Use Case: View/Change the Setpoint Schedule

- **Scenario:** Helen is at the airport waiting for her flight to Hawaii when she realizes that in the hustle and bustle of getting ready to leave, she has forgotten to change the setpoint schedule on her Smartstat. She uses her phone to set it to the vacation schedule for the time she'll be away.
- **Procedure:**
  - Helen loads the Smartstat application on her phone and is immediately brought to the status screen.
  - She can see the setpoint schedule that is currently active.
  - She uses the calendar picker (arrow 1 right) to choose her vacation schedule.
  - She taps the Submit button (arrow 2 right) to send the update to the Smartstat device.

- If Helen would like to verify the updated schedule, she taps the Schedule tab to load the schedule screen.
- The default view will show her the setpoint event schedule for the current day.
- She can use either the day selector (arrow 1 right) or the Next button (arrow 2 right) to navigate to a later date.



- Alternatively, she can tap the Month button to change to a monthly calendar.
- She can then tap any day (right) to zoom to the event schedule screen for that day.



4. Functional Specifications

4.1 Hardware/Device Interface Functionality

4.1.1 The user shall be able to override the current setpoint temperature. P1

4.1.2 Overrides shall remain in effect until the start of the next setting period. P1

4.1.3 The user shall be able to reset the Smartstat to the default settings. P2

4.1.4 The user shall be able to deactivate or reactivate the Smartstat. P2

4.1.5 The user shall be able to view the current temperature (i.e., the actual temperature). P1

4.1.6 The user shall be able to view the current setpoint temperature. P1

4.1.7 The user shall be able to view the Smartstat's network status. P2

4.1.8 The hardware shall ship with a default setpoint schedule based on Energy Star guidelines. This setpoint schedule shall remain in effect until and unless modified by the user. P1

4.1.9 The hardware shall perform self-calibrations to optimize anticipation for energy efficiency. P3

4.1.10 The hardware shall perform self-calibrations to optimize hysteresis for energy efficiency. P3The hardware shall perform self-calibrations to optimize recovery for energy efficiency. P3

4.1.11 The Smartstat shall perform recovery without any need for intervention by the user. P1

4.1.12 The Smartstat shall be capable of keeping the temperature to within ±2ºF of the setpoint temperature. P1

4.1.13 The Smartstat hardware should generate alerts and send them to the web service as needed. P3

4.1.14 The Smartstat hardware should report usage statistics to the web service. P3

4.1.15 The hardware shall be expandable to incorporate advanced HVAC system components such as multi-stage furnaces, air conditioners and fans, damper systems, and multiple temperature sensors. P3

4.2 Browser Application Functionality

4.2.1 The user shall be able to register and set up an online account in order to use the browser application. P1

4.2.2 The user shall be able to log in to her account with a username and password. P1

4.2.3 The user shall be able to change his password. P2

4.2.4 The user shall be able to change her username. P2

4.2.5 The user shall be able to view the current temperature. P1

4.2.6 The user shall be able to view the current setpoint temperature. P1

4.2.7 The user shall be able to view the current outside temperature. P2

4.2.8 The user shall be able to toggle the Smartstat's temperature format between Fahrenheit and Celsius. P3

4.2.9 The user shall be able to override the current setpoint temperature. P1

4.2.10 The user shall be able to view his existing setpoint schedules. P1

4.2.11 The user shall be able to view which setpoint schedule is currently in effect. P1

4.2.12 The user shall be able to switch the Smartstat to a different setpoint schedule. P1

4.2.13 The user shall be able to set the setpoint schedule to be used on specific dates in the future. P1

4.2.14 The user shall be able to modify her existing setpoint schedules. P1

4.2.15 The user shall be able to specify whether modifications to setpoint schedules are to take effect on a recurring/permanent basis or as a one-time exception. P1

4.2.16 The user shall be able to create new setpoint schedules. P1

4.2.17 The user shall be able to reset the Smartstat to its default setpoint schedule. P2

4.2.18 The user shall be able to download the Android application from the browser application. P1

4.2.19 The user should be able to set his alert preferences (see Section 4.3 below). P3

4.2.20 The user shall be able to view performance statistics, such as how closely the thermostat program matches the actual temperature of space.

4.3 Alert Functionality

4.3.1 The user should be able to receive alerts by email, text message, or both. P3

4.3.2 The following alert types should be defined: P3

4.3.2.1 HVAC system stops responding to Smartstat (unable to achieve setpoint temperature)

4.3.2.2 Smartstat loses internet connectivity

4.3.2.3 Password on user's account is changed

4.3.3 The user should be able to opt in or out and choose method of receiving alert for each of these alert types individually. P3

4.4 Android Application Functionality

4.4.1 The user shall be able to log in to her account with the same username and password as she uses for the browser application. P1

4.4.2 The user shall be able to view the current temperature. P1

4.4.3 The user shall be able to view the current setpoint temperature. P1

4.4.4 The user shall be able to override the current setpoint temperature. P1

4.4.5 The user shall be able to view his existing setpoint schedules. P1

4.4.6 The shall be able to view which setpoint schedule is currently in effect. P1

4.4.7 The user shall be able to switch the Smartstat to a different setpoint schedule. P1

4.4.8 The user should be able to modify her existing setpoint schedules. P2

4.4.9 The user should be able to specify whether modifications to setpoint schedules are to take effect on a recurring/permanent basis or as a one-time exception. P2

4.4.10 The user should be able to create new setpoint schedules. P2

4.4.11 The user should be able to reset the Smartstat to its default setpoint schedule. P2

5. Non-Functional Specifications

5.1 Hardware

5.1.1 The Smartstat hardware shall be an embedded system. P1

5.1.2 The Smartstat shall require a persistent Internet connection. P1

5.1.3 The Smartstat shall connect to a home network using an Ethernet cable. P1

5.1.4 The Smartstat shall connect to a home network using WiFi. P2

5.1.5 The Smartstat shall have networking hardware capable of HTTP client requests. P1

5.1.6 The Smartstat hardware shall contain a temperature sensor accurate to within a tenth of a degree Fahrenheit. P1

5.1.7 The hardware shall be a drop-in replacement for any standard thermostat. P2

5.1.8 The hardware shall work with all single-stage and dual-stage furnaces and air conditioners. P1

5.1.9 The hardware shall include an on-device LCD. P1

5.1.10 Installation of the Smartstat hardware shall take no more than 30 minutes when replacing an existing home thermostat. P2

5.1.11 Users who have experience with basic home maintenance and repair shall be able to install the Smartstat without the help of a professional. P2

5.2 Web Service

5.2.1 The web service shall provide an API for communication between the hardware and the client apps. P1

5.2.2 The web service should receive alerts from the hardware and browser application. P3

5.2.3 The web service should receive usage statistics from the hardware. P3

5.2.4 The web service shall receive state information from the hardware. P1

5.2.5 The web service should send alerts to the user as needed. P3

5.3 Browser Application

5.3.1 The browser application shall not require the download of any special software. P1

5.3.2 The browser application shall be compatible with Firefox 3.0 and higher (P1), Internet Explorer 6 and higher (P2), Apple Safari 4.x, and Google Chrome 3.0 and higher (P3).

5.3.3 The browser application shall require at least a 56Kbps dial-up internet connection. P1

5.3.4 The browser interface shall have a response time of no more than 5 seconds. P2

5.3.5 The browser application shall adhere to standard principles of good usability and implement modern UI design technologies such as JavaScript and CSS. P1

5.3.6 The interface for creating and setting setpoint schedules shall be similar to that of commonly used calendar programs such as Google Calendar and Microsoft Outlook. Any user who has experience with such programs shall be able to understand the interface at a glance. P1

5.3.7 Any internet-literate user with a high-school education shall be able to master the browser application and all of its functions within 15 minutes, without any formal training or use of a manual. P2

5.3.8 The browser application should generate security alerts and send them to the web service as needed. P3

5.4 Android Application

5.4.1 The Android application shall work with any smartphone running the Android operating system (version 1.6). P1

5.4.2 The Android application shall be available for free in the Marketplace. P2

5.4.3 The phone application shall adhere to standard principles of good usability and implement modern UI design technologies. P1

5.4.4 The interface for setting setpoint schedules shall use the same calendar paradigm as the browser application. P1

5.4.5 The Android interface shall have a response time of no more than 5 seconds. P2

5.4.6 Any smartphone-literate user with a high-school education shall be able to master the smartphone application and all of its functions within 15 minutes, without any formal training or use of a manual. P2

6. Competitive Analysis

Further research into competing systems has revealed that there are quite a few web-programmable thermostats already on the market that claim ease of use. It seems that our key innovations will be in ease of installation and the creation of a mobile application for interfacing with the device.

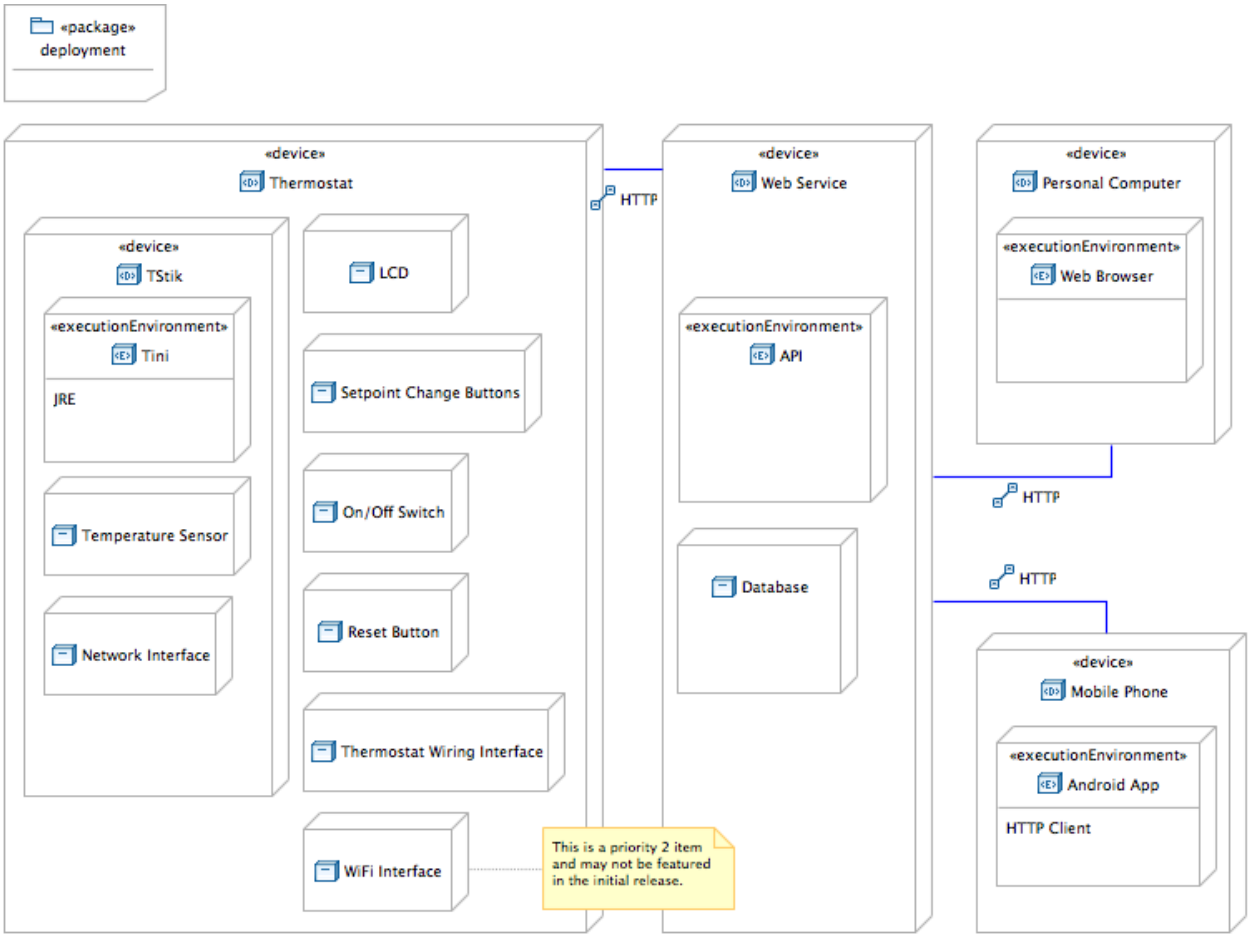
Product	Works with Existing HVAC Systems	Web App	Mobile App	Drop-in Replacement*	DIY Install	Tracks & Reports Usage Statistics	Alerts
Smartstat	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TXU Energy iThermostat	Yes	Yes	No	No	No	Yes	Yes
Proliphix Basic Series Thermostat	Yes	Yes	No	No	Yes	No	No
Ecobee Smart Thermostat	Yes	Yes	No	No	No	Yes	Yes
Aprilaire Model 8870 Thermostat	Yes	Yes	No	Unknown	Unknown	No	No
Comverge SuperStat	Yes	Yes	No	No	Unknown	Yes	Yes

* Requires no additional hardware

7. High-Level System Architecture

The system shall consist of three tiers: embedded system, web service, and client applications.

The following deployment diagram illustrates the three tiers of the Smartstat architecture: embedded system, web service, and client applications.



7.1 Embedded System

The embedded system shall be based on the Tstik platform, which uses the TINI microcontroller. The embedded system shall communicate with the web service using HTTP.

7.2 Web Service/Database

The web service shall serve as the interface between the client applications and the Smartstat hardware. It shall provide a REST- or SOAP-based API to be used by both the embedded system and client applications to send and receive state change information.

7.3 Client Applications

The client applications shall be the means by which the Smartstat can be programmed. They shall communicate with the web service using HTTP.

8. UI Design Principles

8.1 Hardware

8.1.1 Colors shall be used sparingly.

8.1.2 The function of each component shall be intuitive to the user.

8.1.3 The number of components shall be kept to a minimum.

8.2 Browser Application

8.2.1 The browser application shall use a consistent color scheme.

8.2.2 Certain colors shall have symbolic connotations (e.g., red = hot, blue = cold).

8.2.3 UI elements shall be organized in a neat and consistent manner.

8.2.4 No page shall be more than 2 clicks deep.

8.2.5 The position of standard elements (e.g., the navigation bar) shall not change.

8.2.6 Element groups shall be aligned along common axes.

8.2.7 The calendar visual metaphor shall be used for the schedule-management interface.

8.2.8 When the user makes changes, a visualization of those changes shall be provided.

8.3 Phone Application

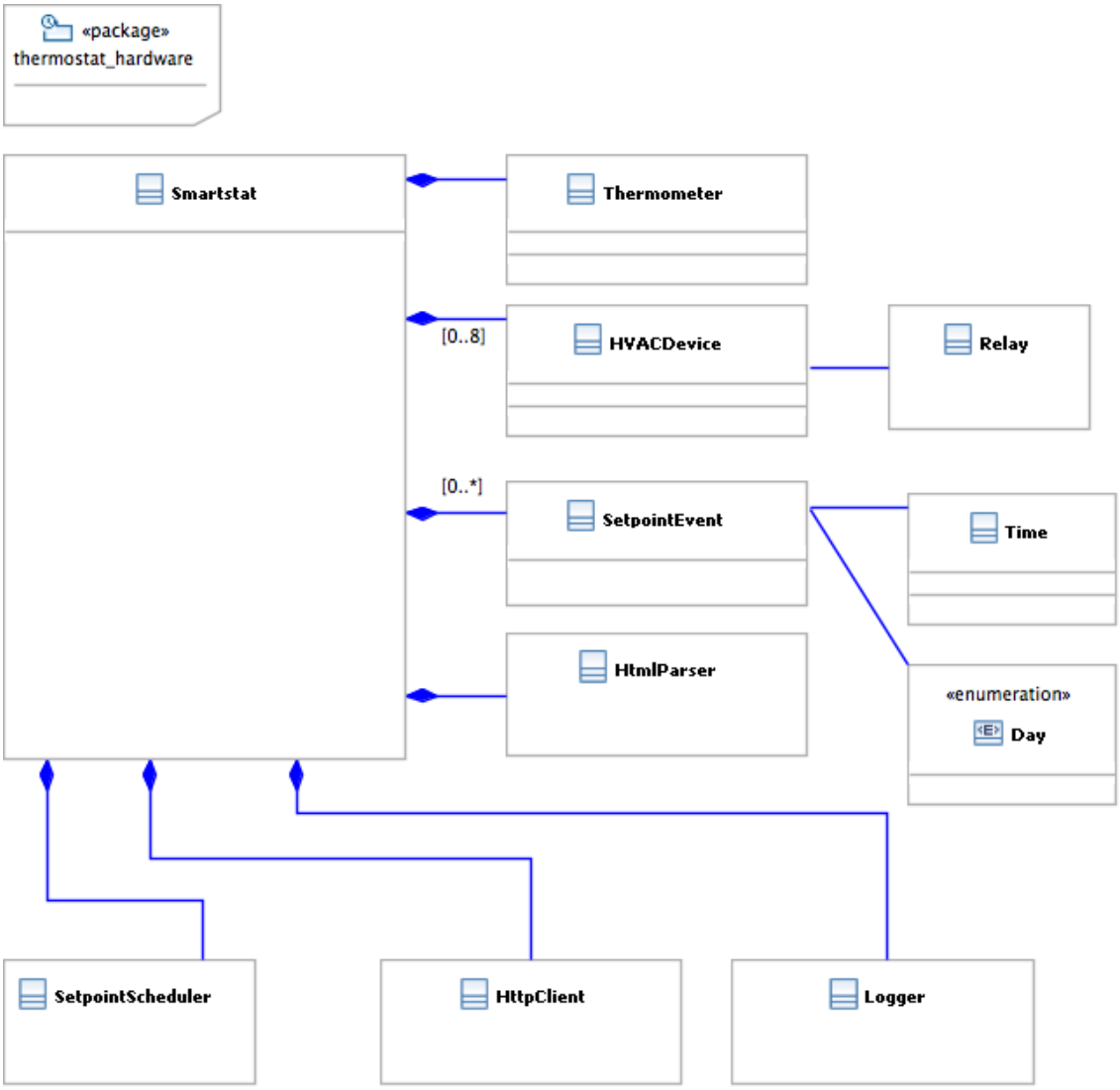
8.3.1 The use of the keyboard shall not be necessary (all tasks can be done using the touch screen and buttons).

8.3.2 The phone application shall use a consistent color scheme. Its palette shall echo that of the browser application.

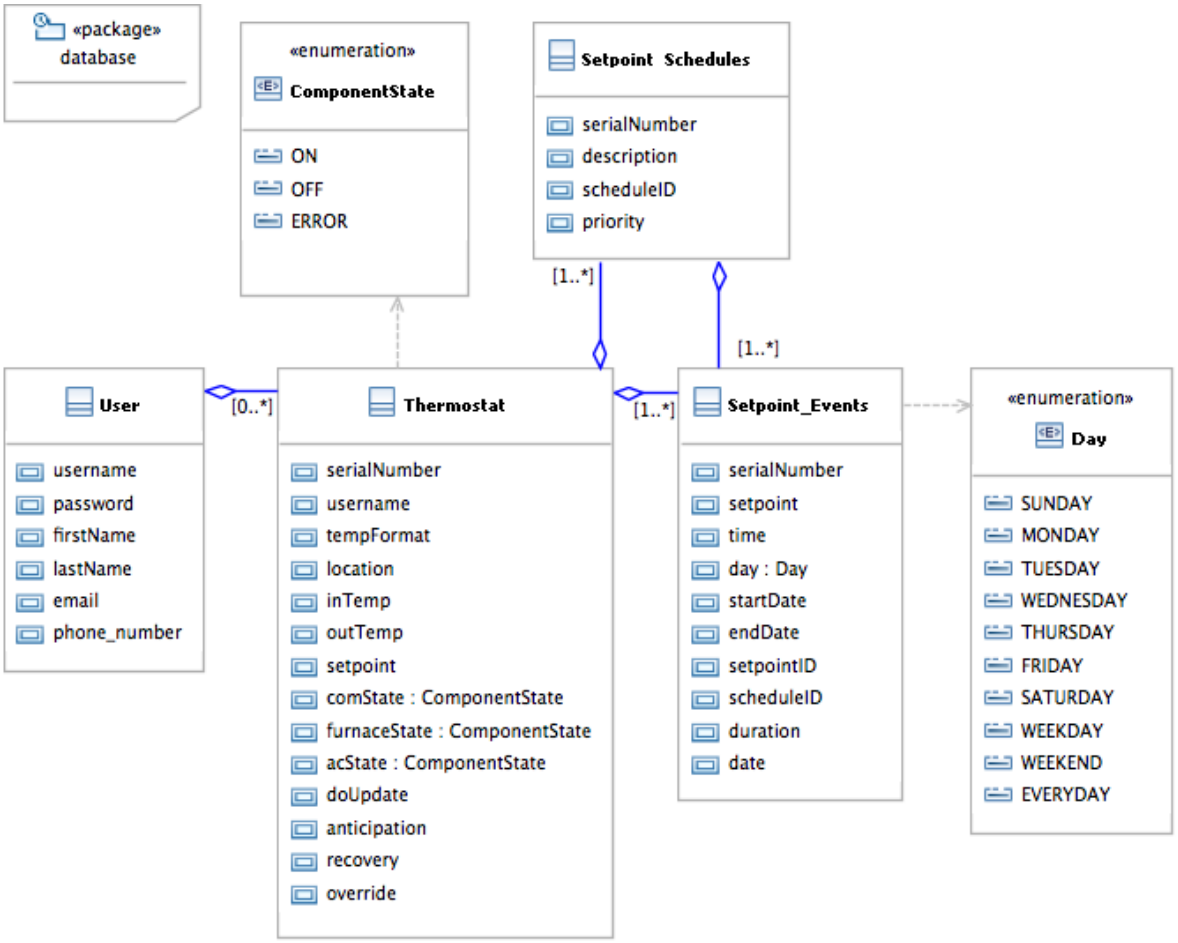
8.3.3 The calendar visual metaphor shall be used for the schedule-management interface.

9. High-Level UML

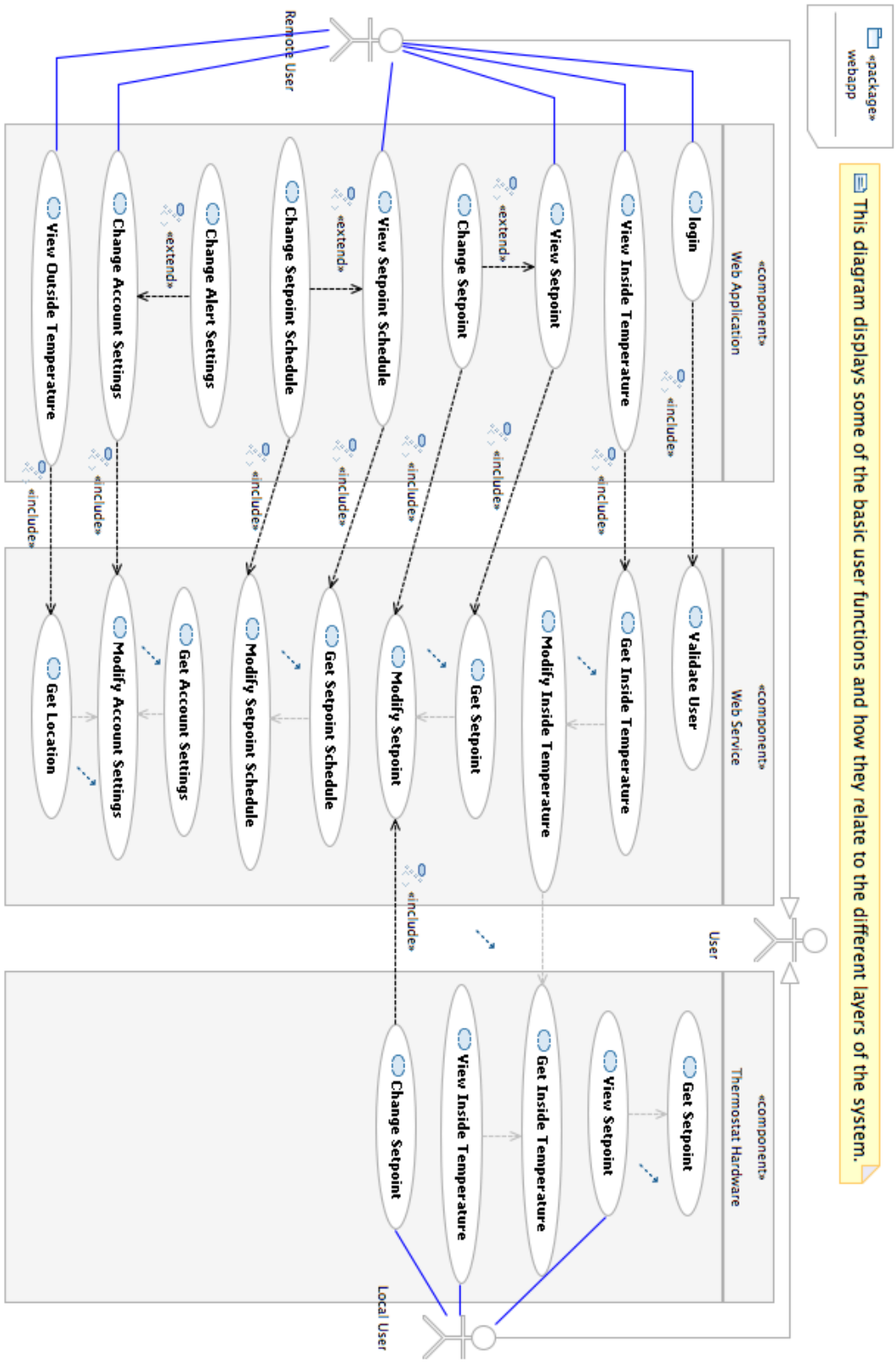
9.1 Conceptual class diagram for the embedded system software.



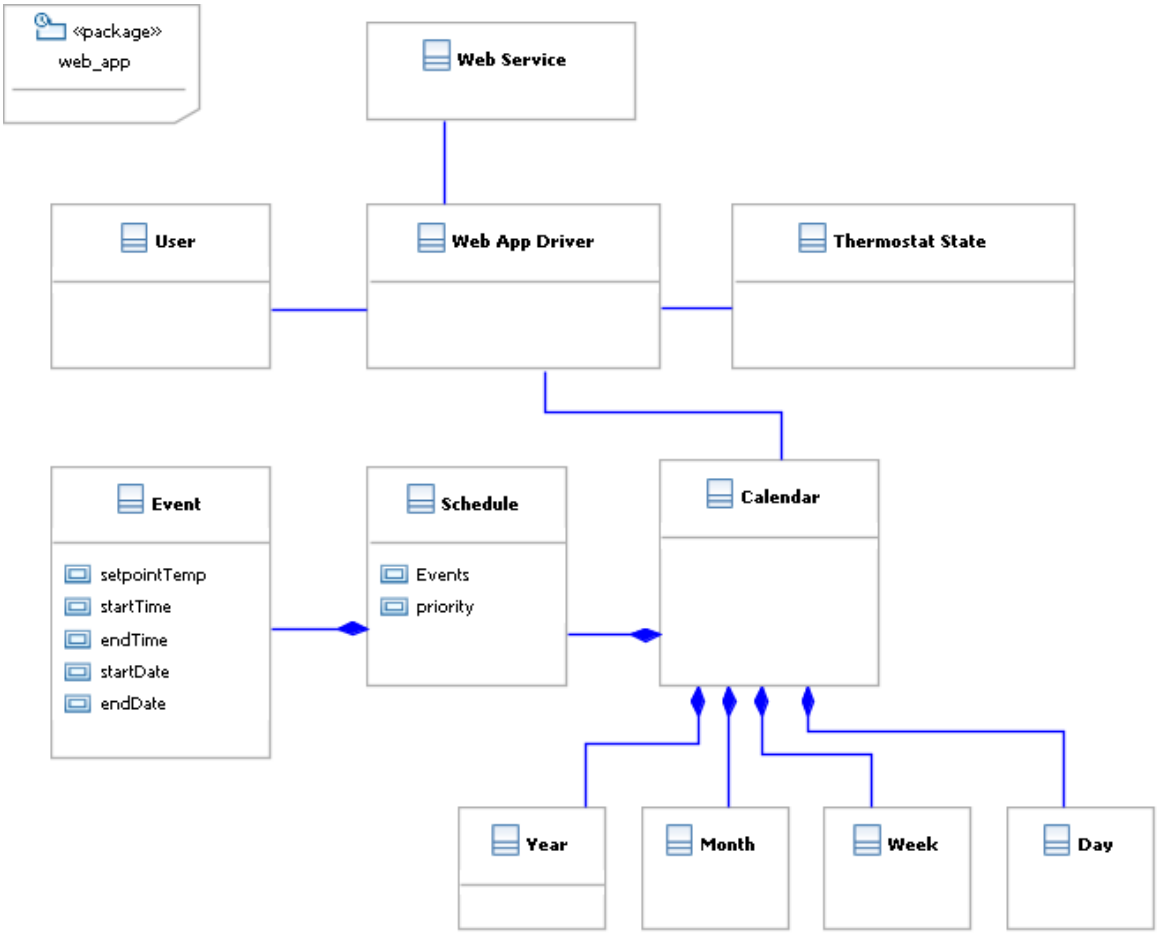
9.2 Conceptual model for the web service database.



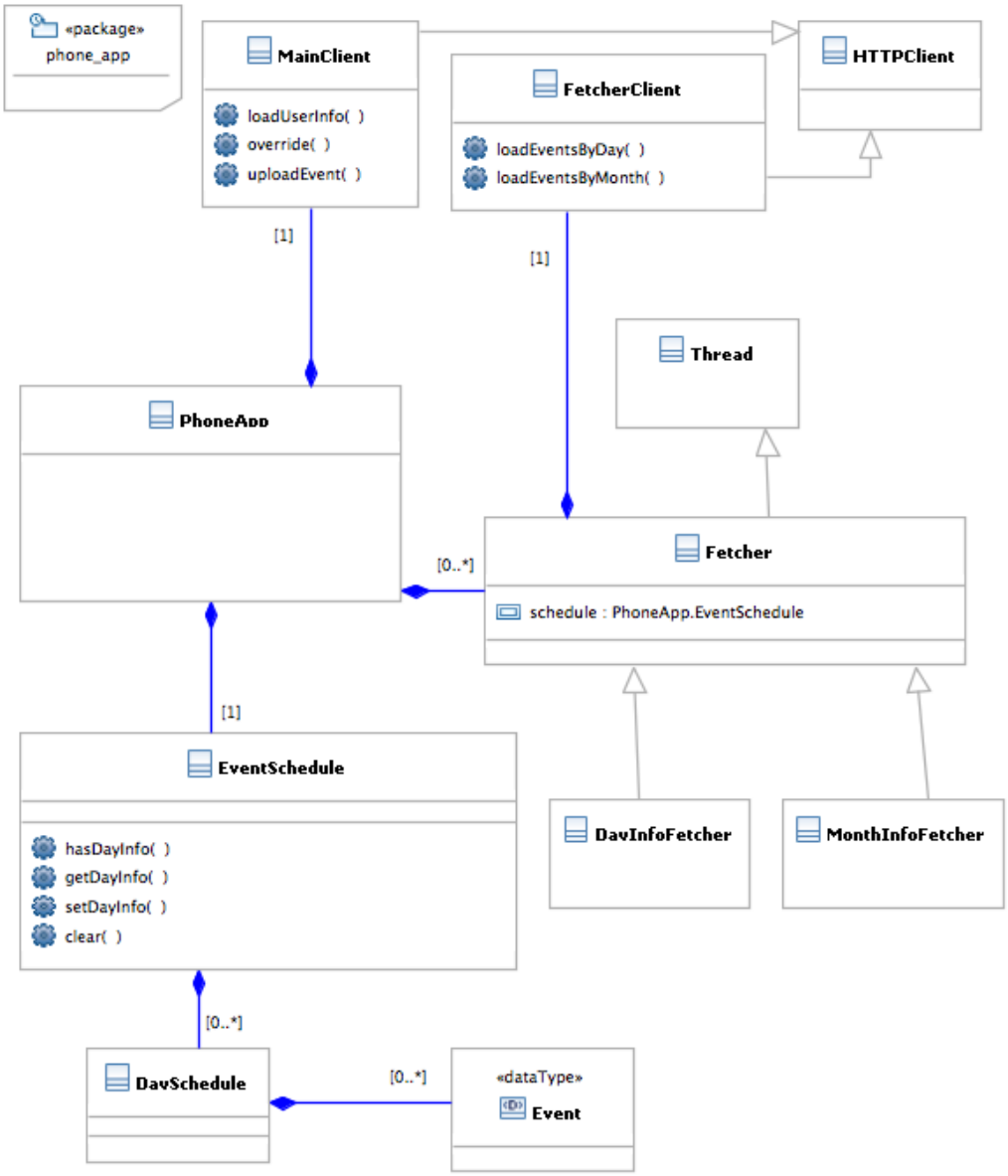
9.3 Use case diagram illustrating some of the basic functions available to remote and local users.



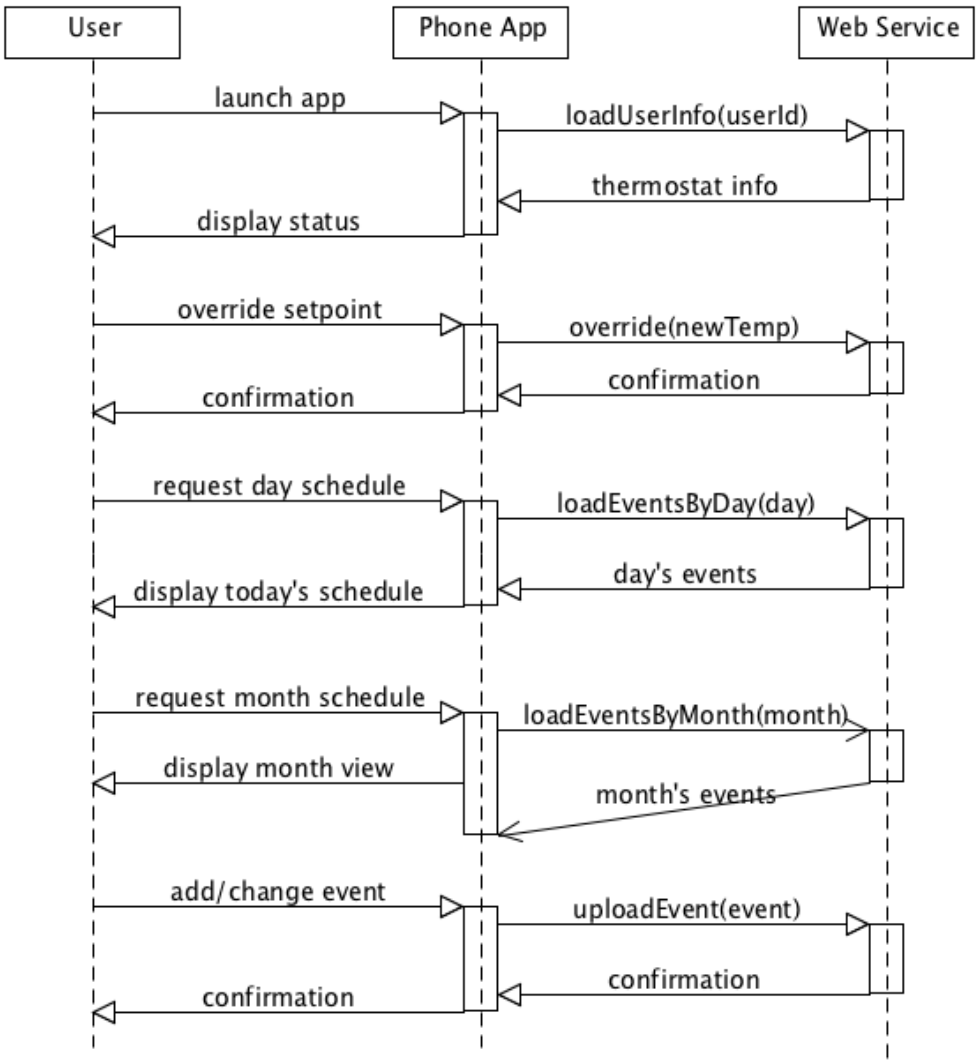
9.4 Web app class diagram illustrating the main classes to be used in the browser application.



9.5 Phone app class diagram. The PhoneApp class serves as a main class that receives prompts and inputs from the user and delegates tasks to other components. The PhoneApp class will use one or more instances of the Fetcher class to query the web service for schedule information, either synchronously or asynchronously. The Fetchers will pass information to the PhoneApp via the shared EventSchedule data structure. Both the PhoneApp and the Fetcher classes contain an instance of a subclass of HTTPClient, which they use to send requests to the web service and process the responses.



9.6 Phone app sequence diagram. For the tasks of loading a user's current status information, overriding a setpoint, viewing the setpoint schedule for a single day, and adding or changing an event, the phone app will send queries to the web service and display the results to the user in a synchronous manner. In the case of loading the setpoint schedule for an entire month, the month calendar view is displayed to the user immediately while the information is being loaded asynchronously in a separate thread.



10. Key Risks

Skills

The development team believes it has adequate knowledge and experience in computer programming to satisfy the software demands of the top-priority items of the project. We do, however, face a lack of expertise in electrical engineering, embedded systems, and HVAC, and we will be relying on research and consulting in these areas in order to complete significant aspects of the project.

Schedule

Due to the small group size, there will be a significant workload for all members of the development team. Additionally, a small amount of our progress hinges on the availability of consultants and hardware components; both of these factors are potential scheduling risks.

Technical

At the time of publication of this document, there are some technical problems that remain unresolved. First, the embedded platform we will be using does not have built-in support for WiFi communication. Second, we do not have a solution for how the Smartstat housing will be fabricated. Third, our deployment plan includes hosting the web service and web application at Sourceforge.net, and we are therefore susceptible to any risks or performance failures experienced by their hardware. Finally, the availability of our phone application in the Android Marketplace is ultimately determined by Google, and is therefore outside of our control.

Legal

We do not perceive any legal risks pertaining to the Smartstat project while it is under the premise of academic research. There would be several legal factors to face in the event of the Smartstat going into the public market, including FCC regulations regarding WiFi features, local and federal electrical/HVAC safety regulations, possible patent infringements, trademarking a product name, unauthorized use of preexisting technologies, and perhaps more.

11. Team and Roles

Team

Drew Mason
Simon Fishel
Helen Tompkins
Kathyrene Villariba

Roles

Hardware: Drew, Helen
Web Interface: Kat, Simon
Phone Interface: Simon, Kat
Native Interface: Simon, Helen

Group Lead: Drew
Documentation Czar: Helen