CEN4010 Principles of Software Engineering, Summer 2023

IOT Smart Thermostat

Group: 6

Repository: https://github.com/AbirFaisal/CEN4010

Trello: https://trello.com/b/2uNrNM0m/project-board

Firebase: https://console.firebase.google.com/u/0/project/cen4010-group-6/overview

YouTube Page: https://youtu.be/A0Lh9OXyH2w

Team members:

Abir Faisal – Scrum Master	afaisal2018@fau.edu
Isaiah Cruz - Develop Team	icruz2020@fau.edu
Gabriel Giani - Develop Team	ggiani2022@fau.edu
Brandon Hyppolite – Develop Team	bhyppolite2020@fau.edu
Kevin Pham - Develop Team	kpham2019@fau.edu

Milestone 4

7/25/2023

Revision 1

Product Summary

Name of Product:

The name of our product is the “IOT Smart Thermostat”.

Explicit list of all major committed functions:

Change the Temperature

Select operating mode (heating/cooling)

Measure and maintain temperature

Unique features of our product: Smart thermostat that can operate on it’s own without any external services that most smart home devices require.

URL to product:

URL: https://cen4010-group-6.web.app/

*Product is a hardware product; URL is for demo only.

Usability Test Plan

Select one major function to be tested for usability:

Changing the temperature.

Test Objectives:

Changing the temperature on the front end changes the corresponding parameter on the back end.

Test Plan:

1. Check the initial value in the database (if any)
2. Visit web page
3. Change the temperature on the web page using the user interface.
4. Check value in database.
5. If value changes to the same value as the user interface, the test passes, otherwise it fails.

Questionnaire form:

1. Does the UI allow the user to change the temperature?

2. Does the backend reflect those changes?

QA test plan

Create a formal QA test plan from the class material

Test objectives:

Verify functionality of temperature dial.

Hardware and software setup:

As a demo using firebase or prototype hardware or from local filesystem in a browser.

Feature to be tested:

Changing the temperature.

Actual test cases:

Test#	Test Title	Test Description	Test Input	Expected Output	PASS/FAIL
1	Change Temp Browser 1	Change temperature to from Chrome	65	65	PASS
2	Change Temp Browser 2	Change temperature from Firefox	75	75	PASS

Code Review

Coding style: Spaghetti

Code chosen for usability testing:

```
<!--The jquery range slider -->
<div id="slider"></div>
<script type="text/javascript">
$(document).ready(function() {
/*Cool & Heat prompt*/
window.changeTooltip = function(e) {
var val = e.value.toFixed(0), speed;
if (val < 80) speed = "COOLING";
else speed = "HEATING";
return "<div>" + speed + "</div>" + "<b>" + val + "</b>";}
$("#slider").roundSlider({
sliderType: "min-range", handleShape: "round",
width: 22, radius: 100, value: 75, startAngle: 315,
lineCap: "round", circleShape: "pie", min: "60", max: "90",
tooltipFormat: "changeTooltip", editableTooltip: false,
svgMode: true,
/*Button function*/
create: function() {
var that = this;
var btn1 = $("<button id='sub'>&#9660;</button>");
var btn2 = $("<button id='add'>&#9650;</button>");
this.innerContainer.append(btn1);
this.innerContainer.append(btn2);
btn1.click(function() {
that.setValue(that.options.value - 1);
});
btn2.click(function() {
that.setValue(that.options.value + 1);});});}});
</script>
```

List containing peer review and commented code:

1. Commenting: Nice use of comments consider adding more details to explain complex parts of the code.
2. Code Organization: To keep things organized, move the JavaScript to an external file and link it with the HTML using the <script> tag.
3. Function Naming: Great effort! Use descriptive names like "getCoolingHeatingTooltip" to improve clarity.
4. “Magic” Numbers: Avoid using numbers without context; use named constants or variables with comments instead.
5. Event Handlers: Use named functions for event handlers to enhance code maintainability.

```
<!--The jquery range slider -->
<div id="slider"></div>

<script type="text/javascript">
$(document).ready(function() {

  // Feedback 1: Good use of comments maybe consider adding more details to explain complex parts of the code.
  /*Cool & Heat prompt*/
  window.changeTooltip = function(e) {
    var val = e.value.toFixed(0), speed;

    // Feedback 4: Avoid using numbers without context; use named constants or variables with comments instead.
    if (val < 80) speed = "COOLING";
    else speed = "HEATING";

    return "<div>" + speed + "</div>" + "<b>" + val + "</b>";
  }

  // Feedback 2: Consider moving the JavaScript code to an external file and link it with the HTML using the <script> tag.

  $("#slider").roundSlider({
    sliderType: "min-range",
    handleShape: "round",
    width: 22,
    radius: 100,
    value: 75,
    startAngle: 315,
    lineCap: "round",
    circleShape: "pie",
    min: "60",
    max: "90",
    tooltipFormat: changeTooltip, // Feedback 3: Use descriptive names like "getCoolingHeatingTooltip" to improve clarity.
    editableTooltip: false,
    svgMode: true,

    /*Button function*/
    create: function() {
      var that = this;

      // Feedback 6: Use named functions for event handlers to enhance code maintainability.
      var btn1 = $("<button id='sub'>&#9660;</button>");
      var btn2 = $("<button id='add'>&#9650;</button>");
      this.innerContainer.append(btn1);
      this.innerContainer.append(btn2);
      btn1.click(function() {
        that.setValue(that.options.value - 1);
      });
      btn2.click(function() {
        that.setValue(that.options.value + 1);
      });
    }
  });
});
</script>
```

Self-check on best practices for security

List of major assets being projected:

1. Web interface passcode
2. Wifi Key

Confirm that passwords is hashed and not stored in plain text:

Team is aware that passwords should be stored as a hash.

Confirm input validation:

Backend can ensure temperature values are within a safe range.

Self-check on Adherence to original non-functional specifications

Initial list of high-level functional requirements from M1 document:

User Functions:

Allow user to set the temperature. - DONE

Allow the user to set a schedule for the temperature. - ISSUE (Time)

Allow the user to select their preferred operating mode. - ON TRACK

Display current temperature and settings – ON TRACK

Allow the user to set temperature threshold for heating and cooling. - ISSUE (Redundant)

System Functions:

Allow the system to sense the temperature of the room. - ON TRACK

Allow the system to get the current weather from a Weather API. - ISSUE (Removed)

Allow the system to connect to a wireless network. - ON TRACK

Allow the system to learn and adapt to the user's preference over time. - ISSUE (Removed)

Allow the system to optimize energy usage. - ISSUE (Not physically possible.)