Kyle Cole, Nea Culbreath, Alvaro de Landaluce,
Logan Ford, Sawyer Jones, & Amshah Mushtaq
COMP 330/474 001
12/11/2021

T5 - Complete Project Requirements

Purpose

A complete set of Atomic Requirements including a suitable glossary for the Electronic Thermometer.

System

The Electronic Thermometer will be a contactless method to determine if a person has a temperature over the fever limit, which is a variable set point.

Overview of Requirements

Table I. Set of Requirements

Identifier	***Title***	***Requirement Statement***
Req 1	Power On-Off	Users may turn the thermometer on and off.
Req 2	Self-Test	**ThermometerSystem** performs a self-test when it is first turned on to check for battery power and sensor responding.
Req 3	Measure Temperature	**ThermometerSystem** displays measured temperature recording.
Req 4	Fever Calculation	**ThermometerSystem** calculates if the user has a fever.
Req 5	Temperature Log	Users may view a log of the temperatures.
Req 6	Inactivity Timer	**ThermometerSystem** shall turn off after thirty seconds of inactivity.
Glossary		Terms that appear in **PascalCase** are defined in the glossary.

Complete Set of Atomic Requirements

Requirement 1: Power On-Off

Importance: Essential

The **ThermometerSystem** shall allow users to turn the thermometer on and off using a **PowerButton**.

1.1 If the user presses the **PowerButton** on the thermometer while the thermometer is turned off, the thermometer shall turn on.

1.2 The **MainScreen** of the thermometer shall display a welcome message for five seconds when it is successfully turned on. The welcome message shall be "Welcome to the Electronic Thermometer."

1.3 If the user presses the **PowerButton** while the thermometer is currently turned on, the **ThermometerSystem** shall turn off.

1.4 The **AverageTemperatureValue** shall be deleted automatically from the **MainScreen** of the **ThermometerSystem** when the **ThermometerSystem** is turned off.

1.5 Only the **PresetValueTemperature** and the temperature readings in the log shall be stored when the device is turned off.

Requirement 2: Self-Test

Importance: Essential

The **ThermometerSystem** shall perform a self-test when it is first powered on to check for battery power and sensory responding.

2.1 The **ThermometerSystem** shall display a self-test message on the **MainScreen** when the thermometer is first turned on. The self-test message shall explicitly state "Self-Test Beginning Now" to the user on the **MainScreen** for five seconds. The **ThermometerSystem** shall automatically start the self-test upon turning on, and the user cannot request a self-test on their own from the **MainMenu.**

2.2 The **ThermometerSystem** shall check the **BatteryLife** to determine if the battery has at least a 20% charge remaining, in which case the **MainScreen** shall be displayed.

2.3 If the thermometer **BatteryLife** has less than a 20% charge remaining, a warning message shall be displayed for five seconds, then the **MainScreen** will be displayed. The battery alert message shall explicitly state "Low Battery" to the user.

2.4 The **ThermometerSystem** shall check if the sensor is able to detect the temperature, in which case the user is allowed access to the **MainScreen**; otherwise, an error message is displayed for five seconds, and the user is not allowed access to the **MainScreen** until a sensor response is verified. The error message shall explicitly state “Sensor Error.”

2.5 If the battery power and the sensor response tests pass, the **ThermometerSystem** shall produce two **BeepSounds**.

2.6 The **ThermometerSystem** shall display **BatteryLife** as a graphical battery icon on the **MainScreen** to update the user on the **ThermometerSystem’s BatteryLife.**

Requirement 3: Measure Temperature

Importance: Essential

The **ThermometerSystem** shall be able to display measured temperature readings from the sensor on the **MainScreen** of the **ThermometerSystem**.

3.1 During the **ReadPeriod**, the thermometer shall repeatedly read the temperature from the sensor until ten successive readings fall within a range of 0.1 degrees Celsius.

3.2 When ten successive readings fall within a range of 0.1 degrees Celsius, the **ThermometerSystem** shall take a snapshot of those ten values, add them up, and divide them by ten, resulting in an **AverageTemperatureValue.** This shall be the temperature value the user is presented.

3.3 Once an **AverageTemperatureValue** has been determined, the **ThermometerSystem** shall produce two **BeepSounds**, as an indication of the end of the successful **ReadPeriod**.

3.4 Following the two **BeepSounds**, the **ThermometerSystem** shall display a single value to the user: the **AverageTemperatureValue** to the nearest tenth degree on the **MainScreen** of the **ThermometerSystem**. The ten values in the snapshot will not be made available for the user and are solely for calculation purposes.

3.5 If ten successive readings within 0.1 degrees Celsius are not found within 10 seconds, the **ThermometerSystem** shall produce one **BeepSound**, indicating a failed reading.

3.6 Once the **ReadPeriod** has concluded, the corresponding message shall be displayed for five seconds if the reading was successful: “Reading Successful.”

3.7 Once the **ReadPeriod** has concluded, the corresponding message shall be displayed for five seconds if the reading failed: “Reading Failed.”

Requirement 4: Fever Calculation

Importance: Essential

The **ThermometerSystem** shall evaluate if the user has a fever using the **AverageTemperatureValue** and the **PresetValueTemperature.**

4.1 The **ThermometerSystem** shall check if the **AverageTemperatureValue** is greater than or equal to the **PresetValueTemperature**, in which case the user has a fever.

4.2 If the user has a fever, the **ThermometerSystem** shall display a message on the **MainScreen** notifying the user that a fever has been detected by explicitly stating “Fever Detected” for five seconds.

4.3 In the case of a detected fever, the thermometer shall produce a **BeepSound**.

4.4 If the **AverageTemperatureValue** is less than the **PresetValueTemperature**, the **ThermometerSystem** shall display a message on the **MainScreen** notifying the user that no fever has been detected by explicitly stating “No Fever Detected” for five seconds.

4.5 The user may change the **PresetValueTemperature** through the **MainMenu** of the **ThermometerSystem** to a value that is considered a fever temperature.

4.6 If the user does not have a detected fever, then the **ThermometerSystem** shall not produce a **BeepSound**.

Requirement 5: Temperature Log

Importance: Moderate

The user shall be able to view a log of the recorded temperatures using the **MainMenu** of the **ThermometerSystem**.

5.1 On the **MainMenu** of the **ThermometerSystem**, the user shall have an option to view a log of the last five **AverageTemperatureValues** the **ThermometerSystem** has recorded.

5.2 The **ThermometerSystem** shall record in memory the last five temperatures the user has taken, the **AverageTemperatureValues**, in order of newest to oldest temperature reading.

5.3 On the **MainMenu** of the **ThermometerSystem**, which is reachable through the **MainScreen,** the user shall be able to select whether they want to view the temperature readings in Celsius or Fahrenheit.

Requirement 6: Inactivity Timer

Importance: Essential

The **ThermometerSystem** shall have a background timer that detects inactivity.

6.1 The **ThermometerSystem** shall start a 30 second long timer when it is turned on.

6.2 The timer shall be reset back to 30 seconds each time the thermometer receives any input or is turned on.

6.3 When the **ThermometerSystem's** timer reaches 0 seconds, the thermometer shall turn off.

Glossary

AverageTemperatureValue: The average temperature of the ten successive temperature readings that fall within a range of 0.1 degrees Celsius. This number may be viewable to the user in Celsius or Fahrenheit to the tenth value and shall be used to detect a fever.

BatteryLife: A graphical battery icon that has 4 shaded cells that are each separated by a blank line. The shading of the cells depends on the battery of the **ThermometerSystem**.

- When 4 of 4 cells are shaded in, that indicates 100% battery (or less) remaining
- When 3 of 4 cells are shaded in, that equates to 80% battery (or less) remaining
- 2 of 4 cells shaded in means that the **ThermometerSystem** has 50% battery (or less) remaining
- 1 of 4 cells shaded in means the **ThermometerSystem** has 20% (or less) battery remaining

A battery percentage is displayed to the right of the graphical battery icon. This display reflects the actual battery percentage remaining before the battery is out of energy.

BeepSound: An audible tone lasting one second.

MainMenu: A menu accessible through the **MainScreen** that houses the temperature log and options to change the **PresetValueTemperature** and degree type (Celsius or Fahrenheit).

MainScreen: The screen the user sees on the Electronic Thermometer, which contains messages and alerts as they appear, temperature readings, and access to the **MainMenu**. The **MainScreen** is the screen that the user views temperature readings and messages on, and it is different from the **MainMenu.**

PowerButton: An accessible button on the Electronic Thermometer that allows users to turn the thermometer on and off. When the user pushes the button once while the **ThermometerSystem** is turned off, the **ThermometerSystem** will be turned on. When the **ThermometerSystem's** power button is pressed a second time, the **ThermometerSystem** will turn off.

PresetValueTemperature: Any value between 95.0 degrees Fahrenheit and 105.0 degrees Fahrenheit that the user may select according to what they consider a fever. The **ThermometerSystem** will then be comparing the detected temperature, the **AverageTemperatureValue**, to the **PresetValueTemperature.** If the **AverageTemperatureValue** is equal to or greater than the **PresetValueTemperature,** then the **ThermometerSystem** will give the user a warning in the form of a **BeepSound**. The **ThermometerSystem** will have a default **PresetValueTemperature** of 100 degrees Fahrenheit, which the user may change from the **MainMenu.**

ReadPeriod: The interval of time where successive readings for the calculation of the **AverageTemperatureValue** are recorded.

ThermometerSystem: Represents the electronic thermometer. The electronic thermometer is a contactless method to determine if a person has a temperature over the fever limit, which is a variable set point.