1. Customer Statement of Requirements

I would like you to build me a desktop application that I can use to view data from a Bluetooth enabled charge controller, I would like to be able to view the battery voltage, current, temperature and amount of remaining charge on the battery. I would also like to be able to control the charge controller by turning its blue LED on or off. It should be possible to initiate a Bluetooth connection from the user interface

2. Requirements Engineering

2.1. Statement of requirements

Table 2-1 enumerates initial requirements for the Bluetooth charge controller system extracted from the problem description in Section 1.

Table 2-1: Requirements for the Bluetooth charge controller system (see Section 1)

Identifier	Priority	Requirement
REQ1	5	The system shall create a Bluetooth connection with the Bluetooth charge controller when the user clicks the connect button.
REQ2	4	The system shall turn on an LED on the Bluetooth charge controller when the user presses the LED on button.
REQ3	3	The system shall turn off an LED on the Bluetooth charge controller when the user presses the LED off button.
REQ4	2	The system shall receive and display data from the Bluetooth charge controller. The data will consist of Voltage, Current, Remaining charge and Temperature. The data will be received every 30 seconds.

2.2. Use Case Modeling

Table 2-2 summarizes preliminary use cases for our Bluetooth controller system.

Table 2-2: Actors, goals, and the associated use cases for the Bluetooth controller system

Actor	Actor's Goal (what the actor intends to accomplish)	Use Case Name
User	To get the Bluetooth controller application and the Bluetooth charge controller connected.	BluetoothConnection (UC-1)
User	To turn an LED on.	TurnLEDOn(UC-2)
User	To turn an LED off.	TurnLEDOff(UC-3)
User	To view Bluetooth charge controller data.	ViewData(UC-4)
Bluetooth charge controller	To receive and respond to all communication from the Bluetooth application. Initiate sending of data for use in UC-4.	UC-1, UC-2, UC-3, UC-4

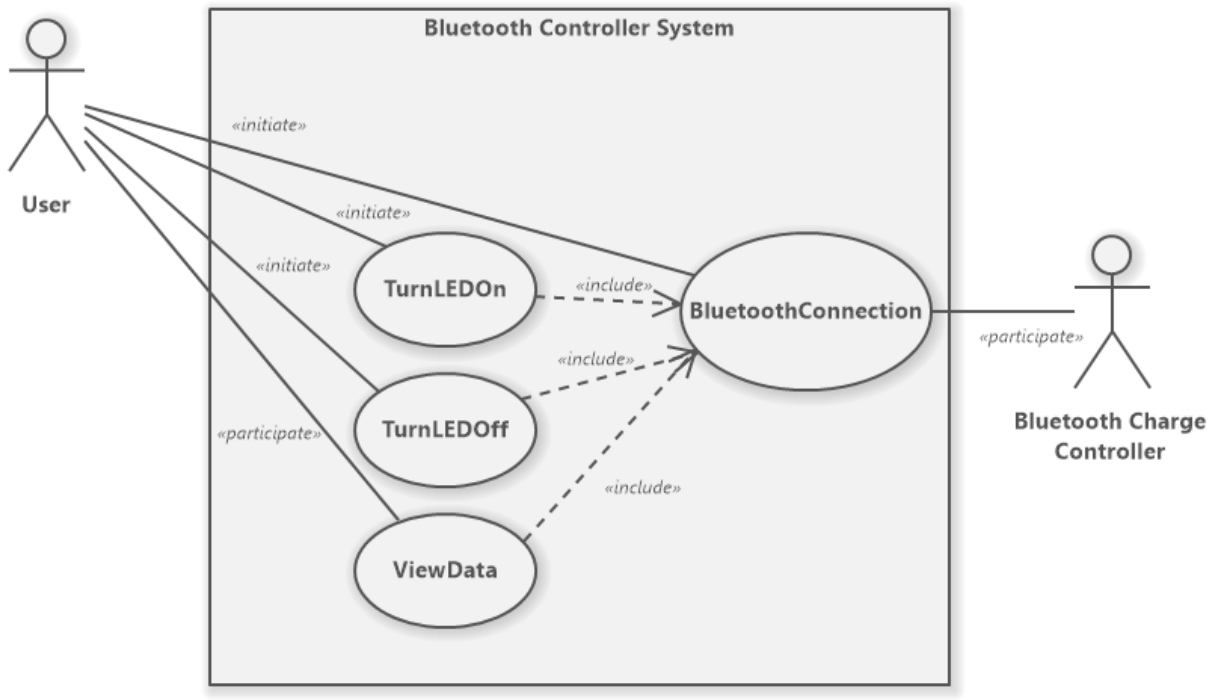

Figure 2-1: Use cases for the Bluetooth controller system.

2.3. Detailed Use Case Specification

This section elaborates on the use cases provided in section 2.2 by providing more details about them.

Use Case UC-1: BluetoothConnection		
Related Requirements:	REQ1	
Initiating Actor:	User	
Actor's Goal:	To get the Bluetooth controller application and the Bluetooth charge controller connected.	
Participating Actors:	Bluetooth charge controller device.	
Preconditions:	• Bluetooth should be turned on in the computer that the Bluetooth controller application will run from. • The Bluetooth charge controller should be on.	
Postconditions:	There should be a Bluetooth connection between the Bluetooth controller application and the Bluetooth charge controller device.	
Flow of Events for Main Success Scenario:	→	1. User presses connect button in the Bluetooth controller application (System).
	←	2. System sends a Bluetooth connection request to the Bluetooth charge controller device.
	←	3. System lets user know that the Bluetooth controller application and Bluetooth charge controller share a Bluetooth connection.
Flow of Events for Extensions (Alternate Scenarios)	→	1. User presses connect button in the Bluetooth controller application (System).
	←	2. System sends a Bluetooth connection request to the Bluetooth charge controller device.
	←	3. System lets user know that the Bluetooth connection request failed.

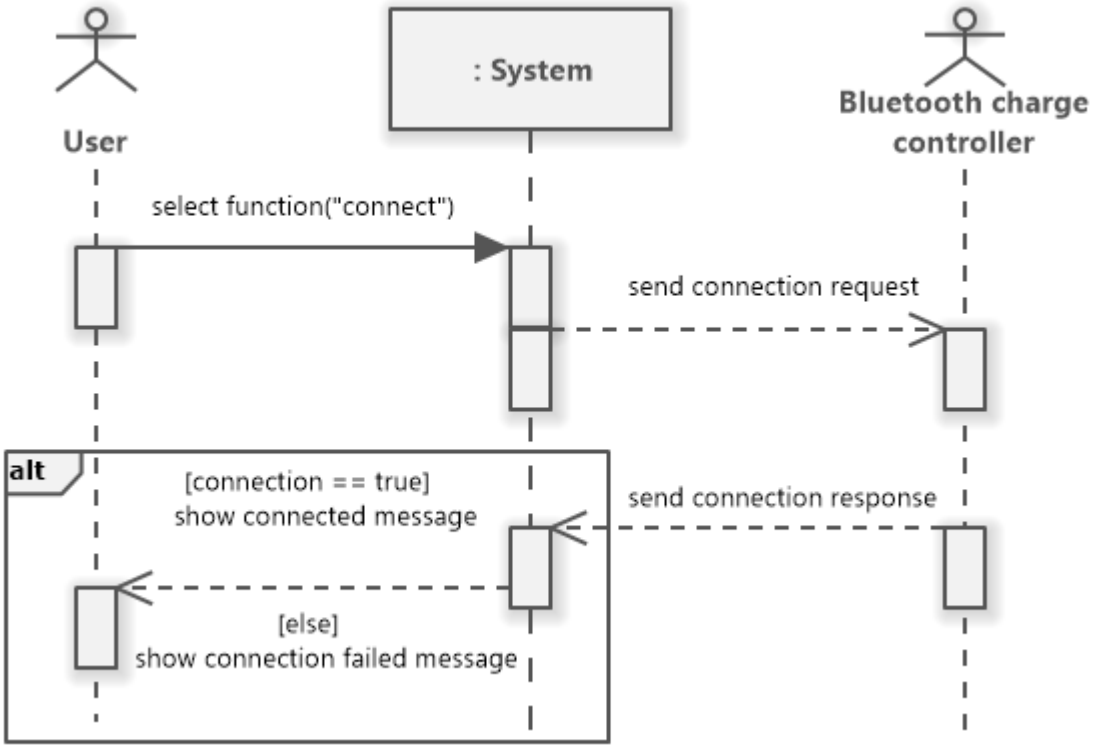

Figure 2-2: Sequence diagram for UC-1: BluetoothConnection.

Table 2-2: Test case for testing the use case UC-1

Test-case Identifier:	TC-1.01
Use Case Tested:	UC-1, main success scenario
Pass/fail Criteria:	The test passes if a Bluetooth connection is created between the system and the Bluetooth charge controller
Input Data:	None
Test Procedure:	**Expected Result:**
Step 1. Press the connect button on the system user interface	System shows a message box with a message that the Bluetooth connection was created

Use Case UC-2: TurnLEDOn		
Related Requirements:	REQ2	
Initiating Actor:	User	
Actor's Goal:	To turn an LED on the Bluetooth charge controller on.	
Participating Actors:	Bluetooth charge controller device.	
Preconditions:	The Bluetooth controller application and Bluetooth charge controller should have a Bluetooth connection.	
Postconditions:	The blue LED on the Bluetooth charge controller is turned on.	
Flow of Events for Main Success Scenario:	→	1. User presses "LED ON" button in the Bluetooth controller application (System).
	←	2. System sends a request to the Bluetooth charge controller device to turn on the LED.
Flow of Events for Extensions (Alternate Scenarios)	→	1. User presses "LED ON" button in the Bluetooth controller application (System).
	←	2. System sends a request to the Bluetooth charge controller device to turn on the LED.
	←	3. System lets user know that the request to turn on the LED failed.

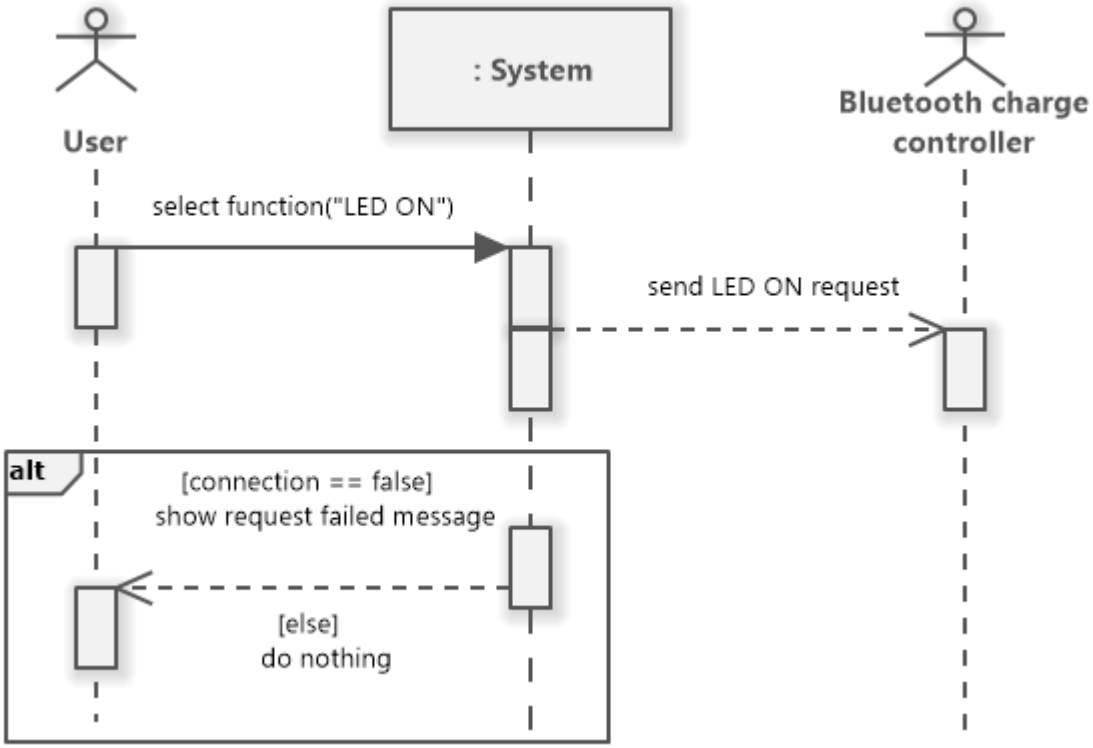

Figure 2-3: Sequence diagram for UC-2: TurnLEDOn.

Table 2-3: Test case for testing the use case UC-2

Test-case Identifier:	TC-2.01
Use Case Tested:	UC-2, main success scenario
Pass/fail Criteria:	The test passes if the blue LED on the Bluetooth charge controller is turned on
Input Data:	None
Test Procedure:	**Expected Result:**
Step 1. Press the "LED ON" button on the system user interface	The blue LED on the Bluetooth charge controller is turned on

Use Case UC-3: TurnLEDOff		
Related Requirements:	REQ3	
Initiating Actor:	User	
Actor's Goal:	To turn an LED on the Bluetooth charge controller off.	
Participating Actors:	Bluetooth charge controller device.	
Preconditions:	The Bluetooth controller application and Bluetooth charge controller should have a Bluetooth connection.	
Postconditions:	The blue LED on the Bluetooth charge controller is turned off.	
Flow of Events for Main Success Scenario:	→	1. User presses "LED OFF" button in the Bluetooth controller application (System).
	←	2. System sends a request to the Bluetooth charge controller device to turn off the LED.
Flow of Events for Extensions (Alternate Scenarios)	→	1. User presses "LED OFF" button in the Bluetooth controller application (System).
	←	2. System sends a request to the Bluetooth charge controller device to turn off the LED.
	←	3. System lets user know that the request to turn off the LED failed.

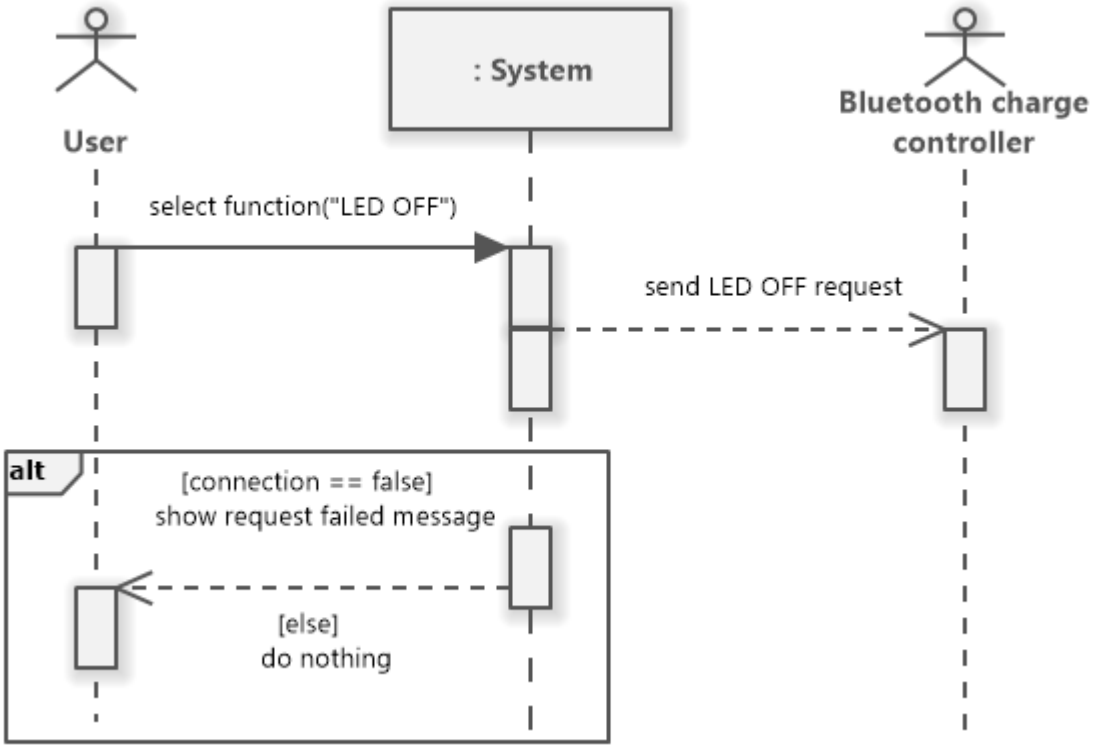

Figure 2-4: Sequence diagram for UC-3: TurnLEDOff.

Table 2-4: Test case for testing the use case UC-3

Test-case Identifier:	TC-3.01
Use Case Tested:	UC-3, main success scenario
Pass/fail Criteria:	The test passes if the blue LED on the Bluetooth charge controller is turned off
Input Data:	None
Test Procedure:	**Expected Result:**
Step 1. Press the "LED OFF" button on the system user interface	The blue LED on the Bluetooth charge controller is turned off

Use Case UC-4: ViewData		
Related Requirements:	REQ4	
Initiating Actor:	Bluetooth charge controller	
Actor's Goal:	To send Voltage, Current, Remaining charge and Temperature data to the Bluetooth controller application every 30 seconds.	
Participating Actors:	User	
Preconditions:	The Bluetooth controller application and Bluetooth charge controller should have a Bluetooth connection.	
Postconditions:	The Bluetooth controller application displays the received data.	
Flow of Events for Main Success Scenario:	→	1. Bluetooth charge controller sends data to the Bluetooth controller application (System).
	←	2. System receives data and displays it.
Flow of Events for Extensions (Alternate Scenarios)	→	1. Bluetooth charge controller sends data to the Bluetooth controller application (System).
	←	2. Two outcomes are possible if System fails to receive data. a. System continues to display previous data. b. System displays blanks on the data fields in case it has not received any data during the new session.

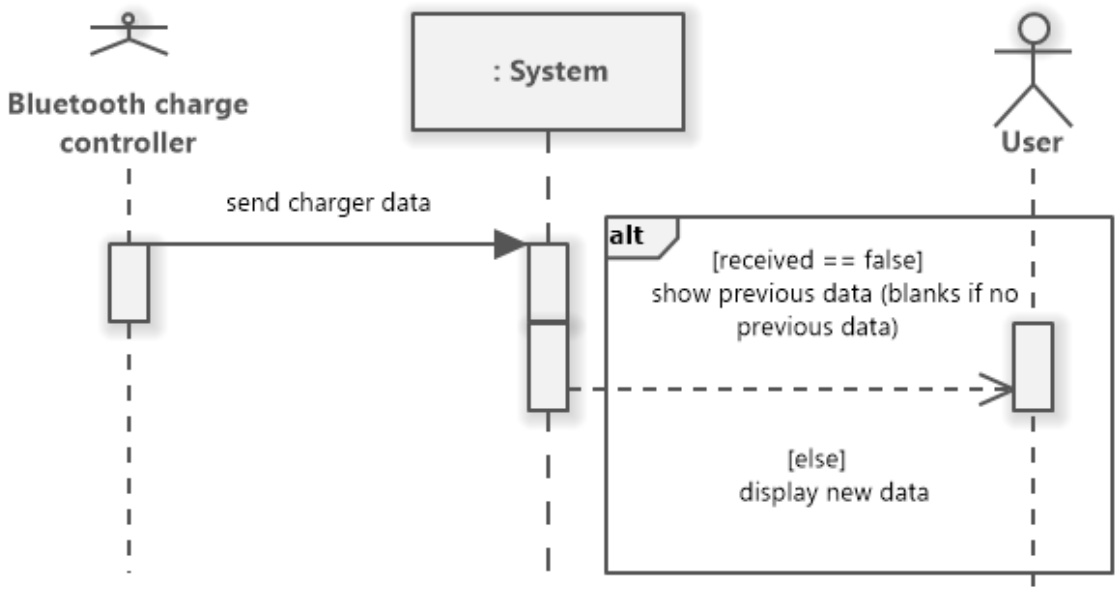

Figure 2-5: Sequence diagram for UC-4: ViewData.

Table 2-5: Test case for testing the use case UC-4

Test-case Identifier:	TC-4.01
Use Case Tested:	UC-4, main success scenario
Pass/fail Criteria:	The test passes if the system receives and displays received data.
Input Data:	None
Test Procedure:	**Expected Result:**
Step 1.Start the system software Step 2. Observe if the data section of the display	The data section of the display shows new data every 30 seconds

3. Analysis

Figure 3-1 shows the domain model of the system.

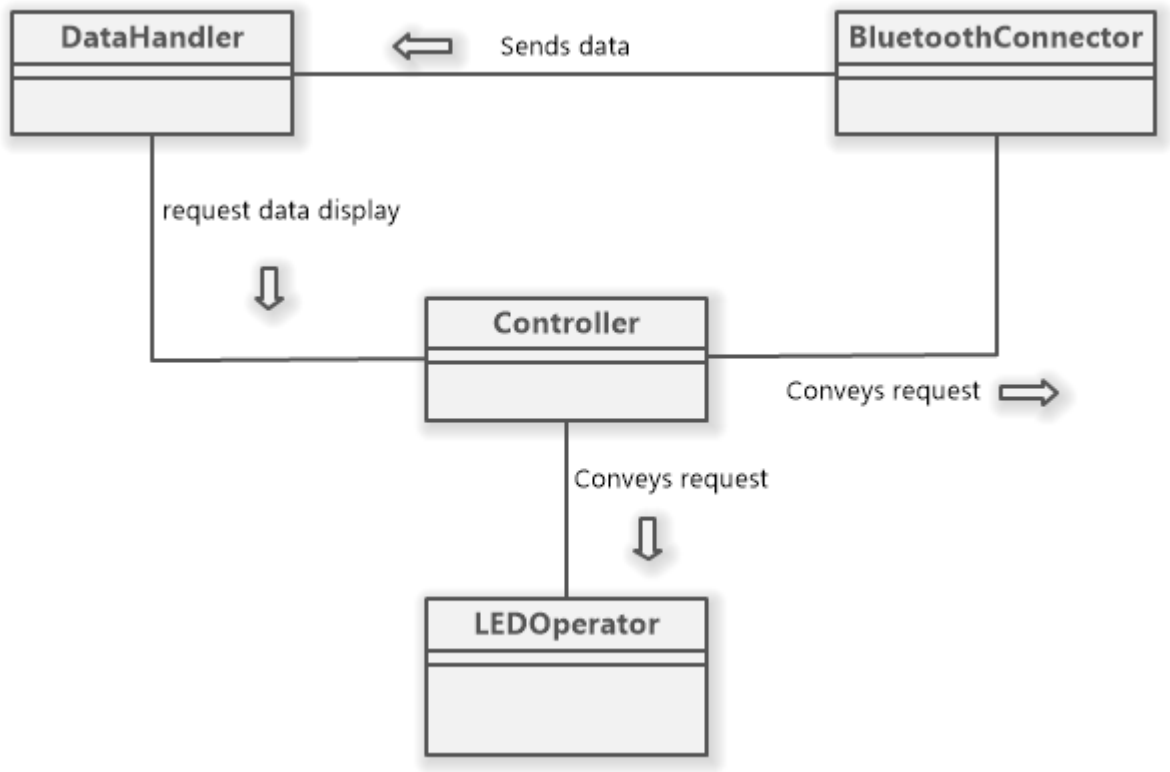

Figure 3-1 Domain model of the system.

4. Design

Figure 4-1 shows the class diagram of the system.

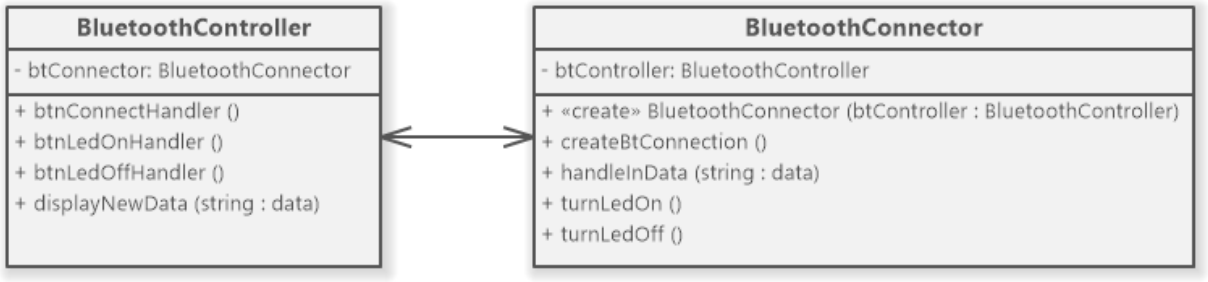

Figure 4-1 System Class Diagram.

Figure 4-2 shows the user interface diagram of the system.

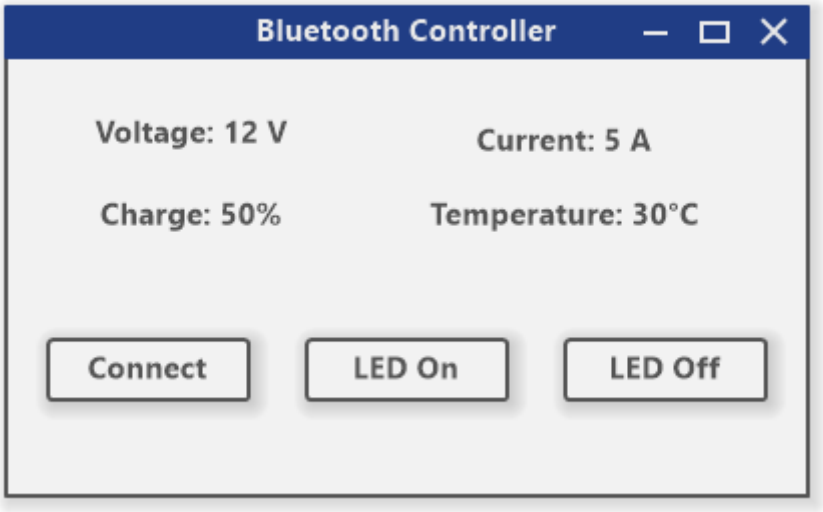

Figure 4-2 System User Interface.