

CTIS - 477 System Engineering Project

Project Requirements Report



**Ömer Levent Durdalı**

**21702600**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 05/03/2021 | Version 0.1.0 | Levent Durdalı | Initial Document |
|  |  |  |  |

Table of Contents

[1. Identification 5](#_Toc65788874)

[2. Scope 5](#_Toc65788875)

[2.1. System Overview 5](#_Toc65788876)

[2.2. Document Overview 5](#_Toc65788877)

[3. Requirements 5](#_Toc65788878)

[3.1. States and Modes 5](#_Toc65788879)

[3.2. Functional Requirements 5](#_Toc65788880)

[3.3. Non-Functional Requirements 6](#_Toc65788881)

[3.3.1. Performance 6](#_Toc65788882)

[3.3.2. Reliability 6](#_Toc65788883)

[3.3.3. Availability 6](#_Toc65788884)

[3.3.4. Security and Privacy Requirements 6](#_Toc65788885)

[3.4. Other Requirements 6](#_Toc65788886)

[3.4.1. System Environment Requirements. 6](#_Toc65788887)

[3.4.2. System Quality Factors 6](#_Toc65788888)

[3.4.3. Computer Resource and Hardware Requirements 6](#_Toc65788889)

[3.4.4. Interfaces 7](#_Toc65788890)

[3.5. Constraints 7](#_Toc65788891)

[4. Qualification Provision 7](#_Toc65788892)

[4.1. Demonstration 7](#_Toc65788893)

[4.2. Test 7](#_Toc65788894)

[4.3. Inspection **Error! Bookmark not defined.**](#_Toc65788895)

[5. System Models 8](#_Toc65788896)

[5.1. Requirements Diagram 8](#_Toc65788897)

[5.2. Use Case Diagram 9](#_Toc65788898)

[5.3. Activity Diagram 10](#_Toc65788899)

[5.1. Component Diagram 11](#_Toc65788900)

[6. Requirement Traceability 12](#_Toc65788901)

Table of Figures

[Figure 1: Use Case Diagram 9](#_Toc65788870)

[Figure 2: Activity Diagram 10](file:///E:\CTIS\Year_4\Sem_2\CTIS%20-%20477%20-%20Systems%20Engineering\Project\Part-2\LeventDurdalı_Requirements_05032021.docx#_Toc65788871)

[Figure 3: Component Diagram 11](file:///E:\CTIS\Year_4\Sem_2\CTIS%20-%20477%20-%20Systems%20Engineering\Project\Part-2\LeventDurdalı_Requirements_05032021.docx#_Toc65788872)

# Identification

The projects name is, Smart Life and Home Security System, this document will refer to the project with following abbreviation; **SLS**.

SLS is a small Smart Home System that manages and tracks curtains, lights, temperature and

motion in a given area.

# Scope

## System Overview

SLS’s main purpose is to be a Decision support system for a home user, meaning the system can help with managing room temperature, adjust light levels and lastly helps the user to adjust the curtains in the specified room. The system will have 3 different sensors that is connected to a central hub. This hub will communicate with a personalized cloud server using the MQTT communication protocol. The end-user will be able to see the gathered data and control the light level and curtain positions from the local web dashboard panel.

## Document Overview

The rest of this document gives the detailed specifications for SLS. It is organized as follows:

* **Section 3: Requirements**

Each objective gives a desired behaviour for the system, a business justification, and a measure to determine if the final system has successfully met in order for the new system to be considered successful and also includes the constraint or a technical requirement on the overall characteristics of the system.

* **Section 4: Qualification Provision**

This section explains the different levels of testing that should be done.

* **Section 5: System Models**

This section gives a pictorial description of the scope of the system with different diagrams.

* **Section 6: Requirement Traceability**

This section shows the traceability of the requirements.

# Requirements

## States and Modes

SLS will not have states or modes.

## Functional Requirements

1. All sensors shall be able to read its intended values.
2. The system shall communicate with NodeRED server over MQTT protocol.
3. NodeRED server shall be able to send instructions the systems hardware.
4. The system shall provide the user with a visual output.
5. The system shall be able to open and close the light automatically if the light intensity is over/below the determined threshold.
6. The system shall be able to start and stop the motor automatically if the light intensity is over/below the determined threshold.
7. The system shall be able to turn on the lights if the motion sensor detects motion.
8. The user shall be able to open and close the connected light.
9. The user shall be able to start and stop the connected motor.
10. The user shall be able to see the temperate values.
11. The user shall be able to see the humidity level.
12. The user shall be able to log in to the system with valid credentials.

## Non-Functional Requirements

### Performance

* The system shall display the visualized output to the user in less than 3 seconds.
* 90% of the operations shall be processed in less than 15s

### Reliability

* The devices within the MQTT network shall have the fault tolerance mechanisms to restart itself.

### Availability

* The system shall have an availability rate of 99.99%.

### Security and Privacy Requirements

* The user passwords shall be hashed in rest.
* The data in transit shall be encrypted with the use of HTTPS.

## Other Requirements

### System Environment Requirements.

* The system shall be able to work in an average room environment
* The systems sensors shall be able to work between 0⁰C - 50⁰C
* The system shall be able to operate in side of a LAN
* The System shall be able to communicate over a local network.

### System Quality Factors

* The sensors shall be swappable for to achieve easy maintainability.
* The sensors shall be easily moved to a new environment.

### Computer Resource and Hardware Requirements

* SLS shall run on a single ESP8266 board with the following hardware resources:
  + Motion Sensor
  + DHT22 Temperature and Humidity Sensor
  + 5mm LDR Sensor
  + 5mm RGB LED Module
  + Servo Motor
* The SLS board shall be powered with a 5V USB connection.

### Interfaces

* + All external and internal interfaces are to be left to the design.

## Constraints

* All updates to the codebase will be done using Git in order not to lose data and manage versions more efficiently.
* In order to maintain a persisting code style and standards will be followed.
* Only analog signal sensors and card drivers can be used.
* Hardware specified in the Proposal Plan must be used.

# Qualification Provision

## Demonstration

* SRQ-SLS-01. All sensors shall be able to read its intended values.
* SRQ-SLS-03. NodeRED server shall be able to send instructions the systems hardware.
* SRQ-SLS-04. The system shall provide the user with a visual output.
* SRQ-SLS-05. The system shall be able to open and close the light automatically if the light intensity is over/below the determined threshold.
* SRQ-SLS-06. The system shall be able to start and stop the motor automatically if the light intensity is over/below the determined threshold.
* SRQ-SLS-07. The system shall be able to turn on the lights if the motion sensor detects motion.
* SRQ-SLS-08. The user shall be able to open and close the connected light.
* SRQ-SLS-09. The user shall be able to start and stop the connected motor.
* SRQ-SLS-010. The user shall be able to see the temperate values.
* SRQ-SLS-011. The user shall be able to see the humidity level.
* SRQ-SLS-012. The user shall be able to log in to the system with valid credentials.

## Test

* SRQ-SLS-02. The system shall communicate with NodeRED server over MQTT protocol.

# System Models

## Requirements Diagram



## Use Case Diagram



Figure 1: Use Case Diagram

## Activity Diagram

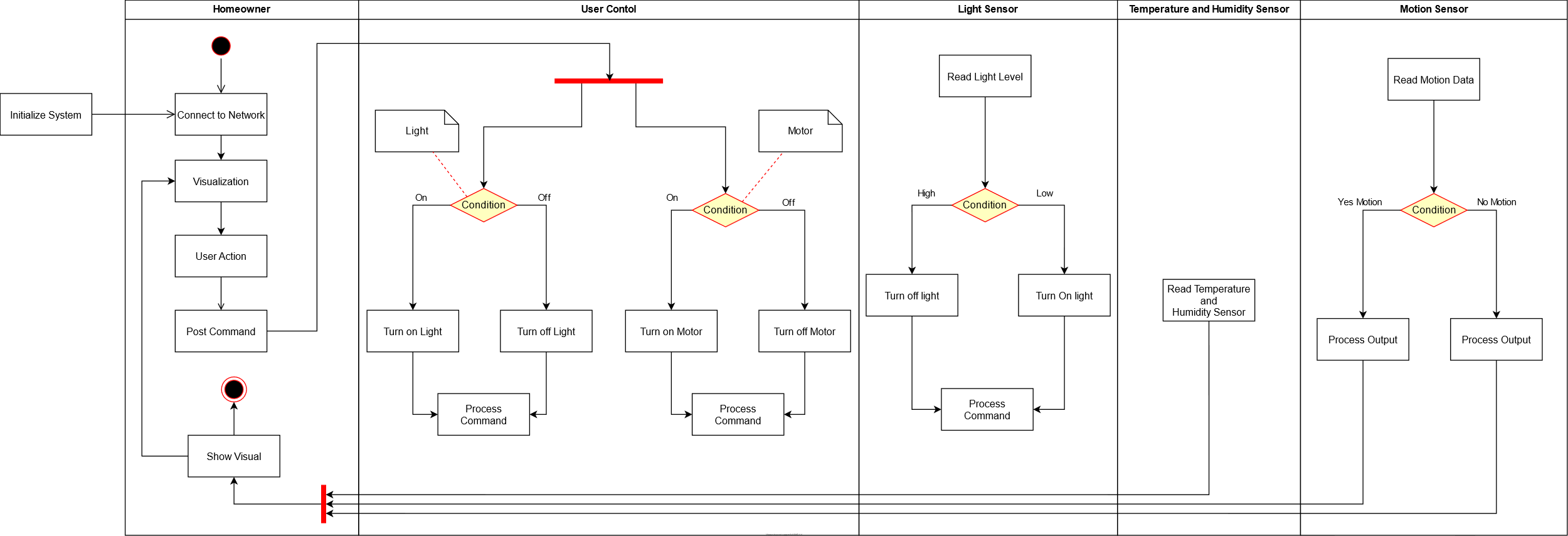


Figure 2: Activity Diagram

## Component Diagram

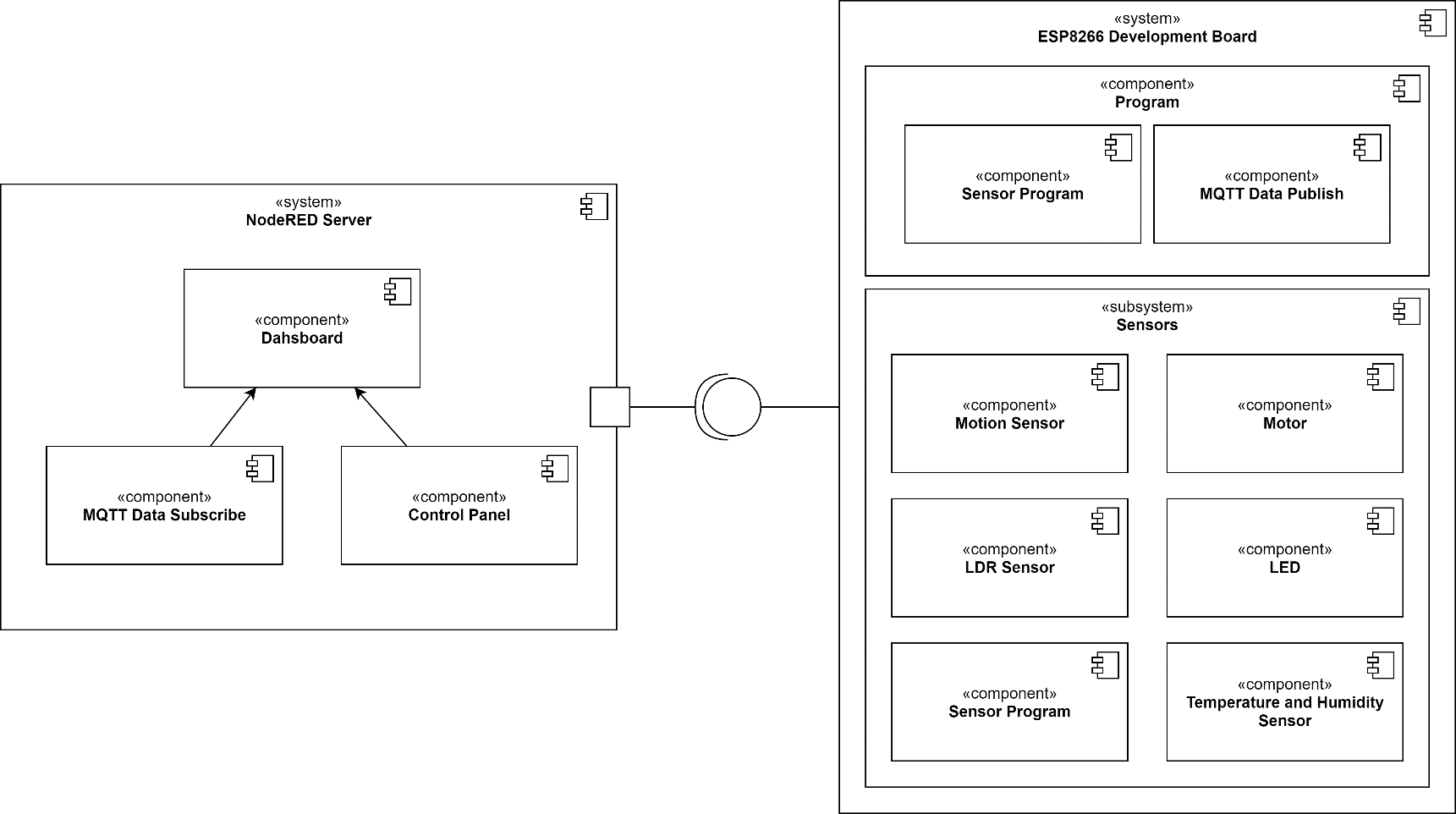


Figure 3: Component Diagram

# Requirement Traceability

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement ID** | **Requirement Name** | **Objective** | **Verification** | **Validation** | **Revision in Document** |
| SRQ-SLS-01 | Get Sensor Info | The NodeRed server will be able to gather sensor data | Having readable values | Demonstration | V0.1.0 |
| SRQ-SLS-02 | Communication | The device will be able to connect to the server with MQTT protocol. | - | Test | V0.1.0 |
| SRQ-SLS-03 | Send Command | The buttons will control the connected devices. | Change output of devices | Demonstration | V0.1.0 |
| SRQ-SLS-04 | Show Visual | The data and buttons will be displayed on  dashboard. | Having working dashboard | Demonstration | V0.1.0 |
| SRQ-SLS-05 | Auto-Toggle Light | System will be able to automatically open/close the light according to the LDR sensor. | Change sent output | Demonstration | V0.1.0 |
| SRQ-SLS-06 | Auto-Toggle Motor | System will be able to automatically open/close the light according to the LDR sensor. | Change sent output | Demonstration | V0.1.0 |
| SRQ-SLS-07 | Motion Detection | The sensor will send alerts to the dashboard and activate the LED. | Change sent output | Demonstration | V0.1.0 |
| SRQ-SLS-08 | Toggle Light | Ability to open and close the light from the dashboard. | Send command | Demonstration | V0.1.0 |
| SRQ-SLS-09 | Toggle Motor | Ability to start and stop the motor from the dashboard. | Send command | Demonstration | V0.1.0 |
| SRQ-SLS-010 | Read Temperatures | The data will be displayed in the  dashboard. | Validating device output | Demonstration | V0.1.0 |
| SRQ-SLS-011 | Read Humidity | The data will be displayed in the.  dashboard. | Validating device output | Demonstration | V0.1.0 |
| SRQ-SLS-012 | Login | Log in to the system with valid credentials. | Login to the system with valid credentials | Demonstration | V0.1.0 |