Lamp

System Design Description

Abstract
This document provides the system design description for the **Lamp System**.

Contents

1 Overview

This document describes the Lamp system, which provides automated room lighting.

In Section 2, we describe implementation details of the system. In Section 3, we summarize the services produced and consumed by the system

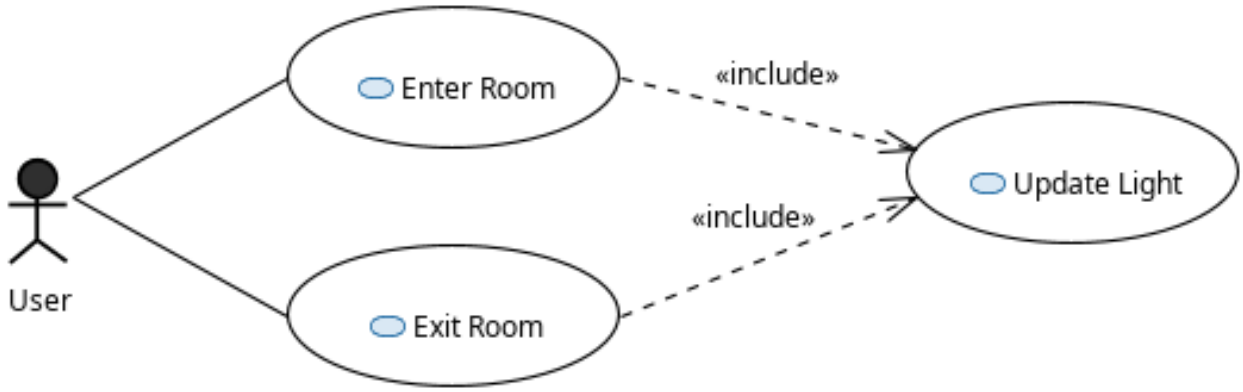

Figure 1: SysML use case diagram. A user can enter a room and the light will be turned on, depending on the movement of the person. A user can also leave a room and if no other person is in the room, the light will be turned off.

2 Implementation

This implementation is based on the System Design Document (SysD) **Lamp**.

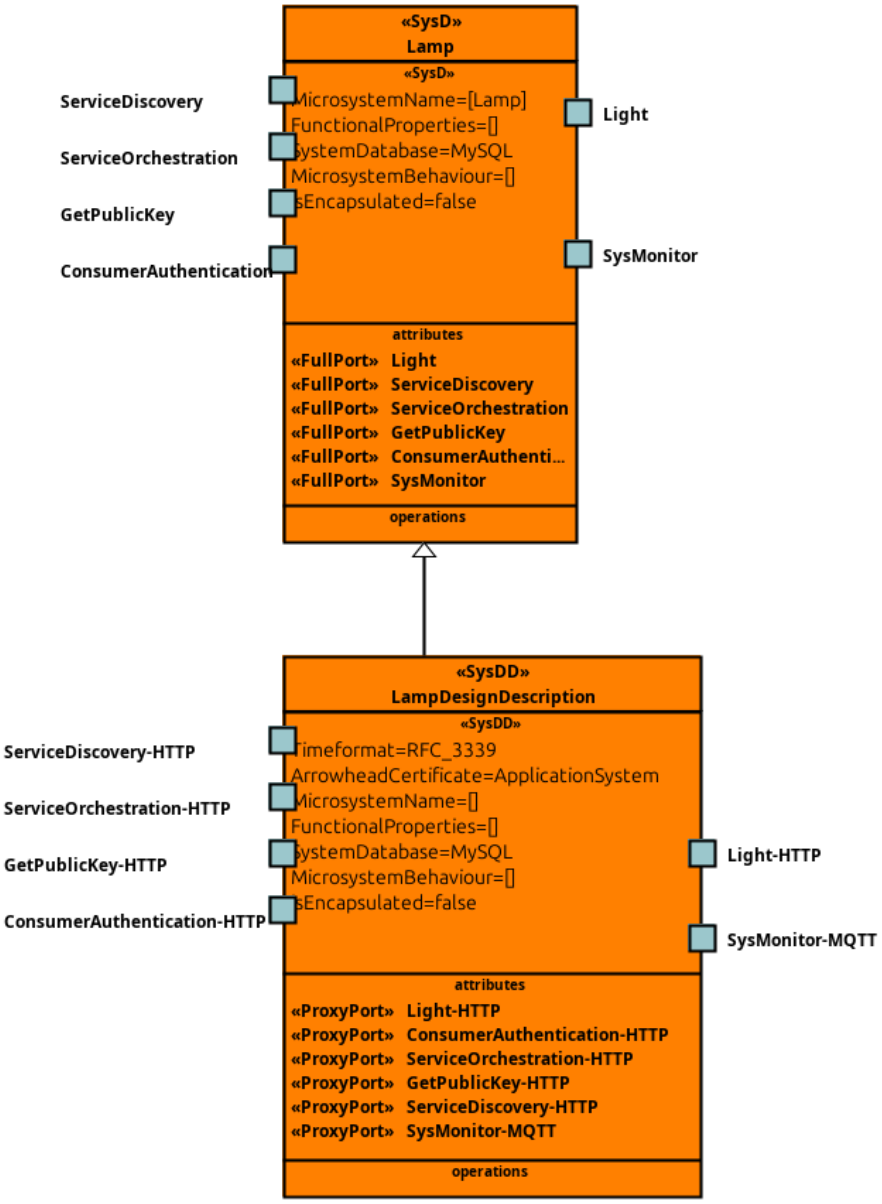

Figure 2: System Definition Block Diagram of the Lamp

2.1 Implementation language and tools

- *Programming Language:* **Python 3.12.3**
- *Programming Framework:* **Flask 3.0.3**
- *IDE:* **Visual Studio Code + Black Formatter**
- *Building Tool:* **Pipenv 2024.1.0**
- *Database Management System:* **MySQL 8.4**

- *State:* **Stateful**

2.2 Functional properties implementation

2.2.1 Database structure

Implementation of data storage functionality was done as describe by Figure 3.

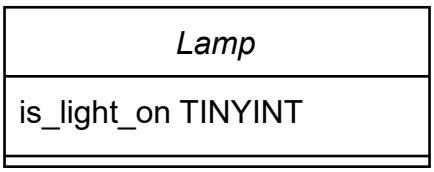

Figure 3: Database model of the Lamp

2.2.2 Configuration

There is no further configuration needed for the system to work, as it only server one purpose, to control the light. Property configuration can be found in section 2.3.4.

2.3 Non-functional properties implementation

2.3.1 Security

The system's security is relying on SSL Certificate Trust Chains. The system does not contain configuration for Arrowhead unsecure/ secure mode. The Arrowhead trust chain consists of three levels:

- Master certificate: `arrowhead.eu`
- Cloud certificate: `my-smarthome.smarthome.arrowhead.eu`
- Client certificate: `my-lamp.my-smarthome.smarthome.arrowhead.eu`

The trust chain is created by issuing the cloud certificate from the master certificate and the client certificate from the cloud certificate. with other words, the cloud certificate is signed by the master certificate's private key and the client certificate is signed by the cloud certificate's private key, which makes the whole chain trustworthy.

For Arrowhead certificate profile, see: `https://github.com/eclipse-arrowhead/documentation`

2.3.2 Power management

The state of the lamp is preserved on hard disk, so the state is saved even when the lamp goes down. That means, the system is very robust against power outages.

2.3.3 Internal monitoring

The system provides monitoring information which can be consumed by a monitoring aggregator. It simply logs when the state of the lamp has changed.

2.3.4 Configuration

The system configuration properties can be found in the `application.properties` file, which is located at `src/main/resources` folder.

- **database.datasource.url**

  URL to the database.

- **database.datasource.username**
  Username to the database.
- **database.datasource.password**
  Password to the database.
- **database.datasource.driver-class-name**
  The driver provides the connection to the database and implements the protocol for transferring the query and result between client and database.
- **database.jpa.show-sql**
  Set to true in order to log out the SQL queries.
- **database.jpa.properties.hibernate.format_sql**
  Set to true to log out SQL queries in pretty format.
- **database.jpa.hibernate.ddl-auto**
  Auto initialization of database tables. Value must be always 'none'.
- **server.address**
  IP address of the server.
- **server.port**
  Port number of the server.
- **domain.name**
  Set this when the system is available via domain name within the network.
- **domain.port**
  Set this when the system is available via domain port within the network.
- **core_system_name**
  Name of the system.
- **log_all_request_and_response**
  Set to 'true' in order to show all request/response in debug log.
- **server.ssl.enabled**
  Set to 'false' in order to disable https mode.
- **server.ssl.key-store-type**
  Type of the key store.
- **server.ssl.key-store**
  Path to the key store.
- **server.ssl.key-store-password**
  Password to the key store.
- **server.ssl.key-alias**
  Alias name of the certificate.
- **server.ssl.key-password**
  Password to the certificate.

- **server.ssl.client-auth**
  Must be always ’need’ which means that SSL client authentication is necessary when SSL is enabled.
- **server.ssl.trust-store-type**
  Type of the trust store.
- **server.ssl.trust-store**
  Path to trust store.
- **server.ssl.trust-store-password**
  Password to trust store.
- **disable.hostname.verifier**
  If true, http client does not check whether the hostname is match one of the server’s SAN in its certificate.

3 Services

The implementation services is based on the follwoing SD and IDD documents:

- SD: Light
- IDD: Light-HTTP

Table 1: References to documentation for services produced and consumed.

Services produced	**SysD ref**	**SD ref**	**IDD ref**
Light	SysD Lamp	SD Light	IDD Light-HTTP
SysMonitor	SysD MicrosystemMonitor	SD SysMonitor	IDD SysMonitor-MQTT
Services consumed	**SysD ref**	**SD ref**	**IDD ref**
ServiceDiscovery	SysD ServiceRegister	SD ServiceDiscovery	IDD ServiceDiscovery-HTTP
ServiceOrchestration	SysD ServiceOrchestration	SD ServiceOrchestration	IDD ServiceOrchestration-HTTP
GetPublicKey	SysD ConsumptionAuthorisation	SD GetPublicKey	IDD GetPublicKey-HTTP
ConsumerAuthentication	SysD Authentication	SD ConsumerAuthentication	IDD ConsumerAuthentication-HTTP

4 References

5 Revision History

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	2024-10-12	4.6.2		Max Lütkemyer

5.2 Quality Assurance

No.	Date	Version	Approved by
1	2024-10-12	4.6.2	