1. Introduction

The target of the project is to develop a web application that remotely manages gas meters installed in the field. Specifically it will:
• Perform the acquisition of the data (e.g. gas consumption), events and alarms elaborated by the gas meter;
• Edit the configuration of the gas meter and send commands (e.g. valve closing).
The web application will be developed, tested and installed in a local server and will be accessed through a web browser.
The piece of software to develop correspond to the SAC (Central Acquisition System) in the Italian legislation.

The data exchange communication protocols and the interface model are described in the DLMS/COSEM specification (IEC 62056, https://www.dlms.com/dlms-cosem/overview). COSEM (Companion Specification for Energy Metering) models the gas meter as a server application used by client applications that retrieve data from, provide control information to, and instigate known actions within the meter via controlled access to the COSEM objects. This objects are uniquely identified through a set of OBIS (Object Identification System) codes.

COSEM and OBIS are specified in the Blue Book (DLMS User Association 1000-1).
DLMS (Device Language Message Specification) is the application layer protocol that turns the information held by the objects into message.

It provides services to connect the clients and the servers and to access the data held by the COSEM object. It also builds the messages (APDUs, Application Protocol Data Units), applies, check and removes cryptographic protection as needed and manages transferring long messages in blocks.

Communication profiles specify how the DLMS/COSEM application layer is supported by the lower protocol layers. The SAC uses a TCP-UDP/IP based communication profile suitable for remote data exchange with gas meter via IP enabled networks such as GPRS.

The application layer, the lower layers and the communication profiles are specified in the Green Book (DLMS User Association 1000-2).

2. Specifications

To reach the target the web application (SAC) will have the following features:

1. A page that lists the gas meters installed in the field. It will provide in a table these information:
? LDN (logical device name) which is mandatory to identify the gas meter during the data exchange;
? PDR (Metering point identifier) which identifies the position where the meter is installed
? Current Index of Converted Volume;
? Current diagnostic;
? Batteries days of autonomy;
? Date and time of the last connection;

2. A page that allows to add a new gas meter and the information related to it (LDN, encryption key) to the page mentioned in point 1. This page should also permit to import several gas meter encryption key using a text file formatted as a csv;

3. A database where to store the encryption key associated with the gas meter. Without the encryption key the DLMS/COSEM client won’t have access to most of the gas meter data;
4. A database where to store the data received from each gas meter. There should also be the possibility to import these data trough SOAP API to an ERP software for billing purpose and organize the maintenance activity or customers support;

5. A detail page of each gas meter that provides the last value and the historical values of the following COSEM object:

- LDN;
- PDR;
- Current Index of Converted Volume;
- Current Index of Converted Volume Under Alarm
- Current diagnostic;
- Batteries days of autonomy;
- Date and time of the last connection;
- Valve status;

6. The page described above will also have a section that allows to change the valve status (enable for opening or closing) by sending the appropriate command.

The object mentioned in point 5 can be retrieved by the SAC by invoking the appropriate DLMS Application layer service or by interpreting a COSEM object called CF (compact frame) sent spontaneously by the gas meter to the SAC (as required by the Italian legislation.

Workflow

The student will take advantage of the open source DLMS library provided by Gurux Ltd (http://www.gurux.fi) which implements the lower layers of the protocol stack.

The student will also have as a starting point a software developed by company that handles the communication between a gas meter and a PC through an optical probe. This software allows to configure and read data from a gas meter locally using an HDLC based communication profile.

In the first phase the student will familiarize with the development environment (Visual Studio), the programming language (C#) and the Gurux DLMS library. Then the student will write the application code on his PC and test it with the equipment made available by company.

Specifically: a local server, where to deploy the application, and a gas meter, which has the ability to show the APDUs exchanged with the SAC as debug strings on a PC.

The Italian legislation defines a list of clients the SAC can authenticate as when it communicates with a gas meter. Each client has different read/write/action privileges to the COSEM object attributes and methods. The two mandatory client the SAC must use are Public (Id 16) and Management (Id 1).

The purpose of the Public client is mainly to retrieve the LDN of the gas meter. This resource can be accessed by the Public client within an explicit Application association without encryption.

It means the SAC sends an authentication request (AARQ) and waits for the response (AARE) then it sends a COSEM get request and waits for the answer and lastly it sends a release request (RLRQ) and waits for the response (RLRE).
Once the SAC has identified the gas meter through its LDN it can authenticate using the Management Client. Then the COSEM objects can be accessed within a pre-established association (without sending AARQ and RLRQ APDUs) and the messages are protected with an encryption key.

The target of the first stage of development will be to successfully read the logical device name of one gas meter using the Public Client. Then the student will work on the part of the data exchange that involves the Management Client with the goal of retrieving all the COSEM object listed in the specifications. Lastly the web application will be able to work with multiple gas meters.

The Italian legislation introduced the possibility for the gas meter to send an unsolicited message called push. This message groups several COSEM object in a compact form and is sent to the SAC using the DATA-NOTIFICATION service (refer to the Green Book chapter 10.5.7.4).

The web application should also be able to receive push messages.

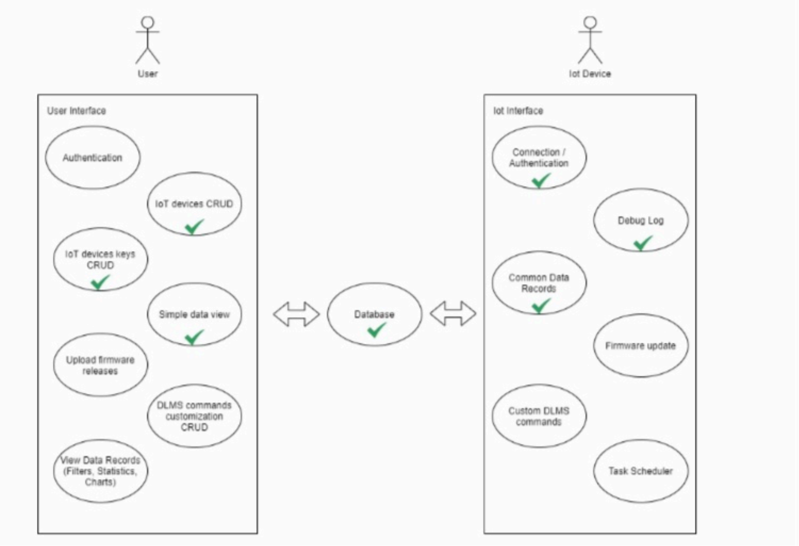
4. Use case diagram

The following simple use case diagram shows the complete functionalities of a web application that manages IoT devices communications.
The web application will serve two kind of agents:

? human user: uses CRUD (create, read, update, delete) pages to manage IoT devices, reads data, statistics, charts;
? Gas meter (IoT device): connects to the web application, receives DLMS commands and transmits DLMS objects.

The minimum set of features we suggest developing in this project are those with the check marks.

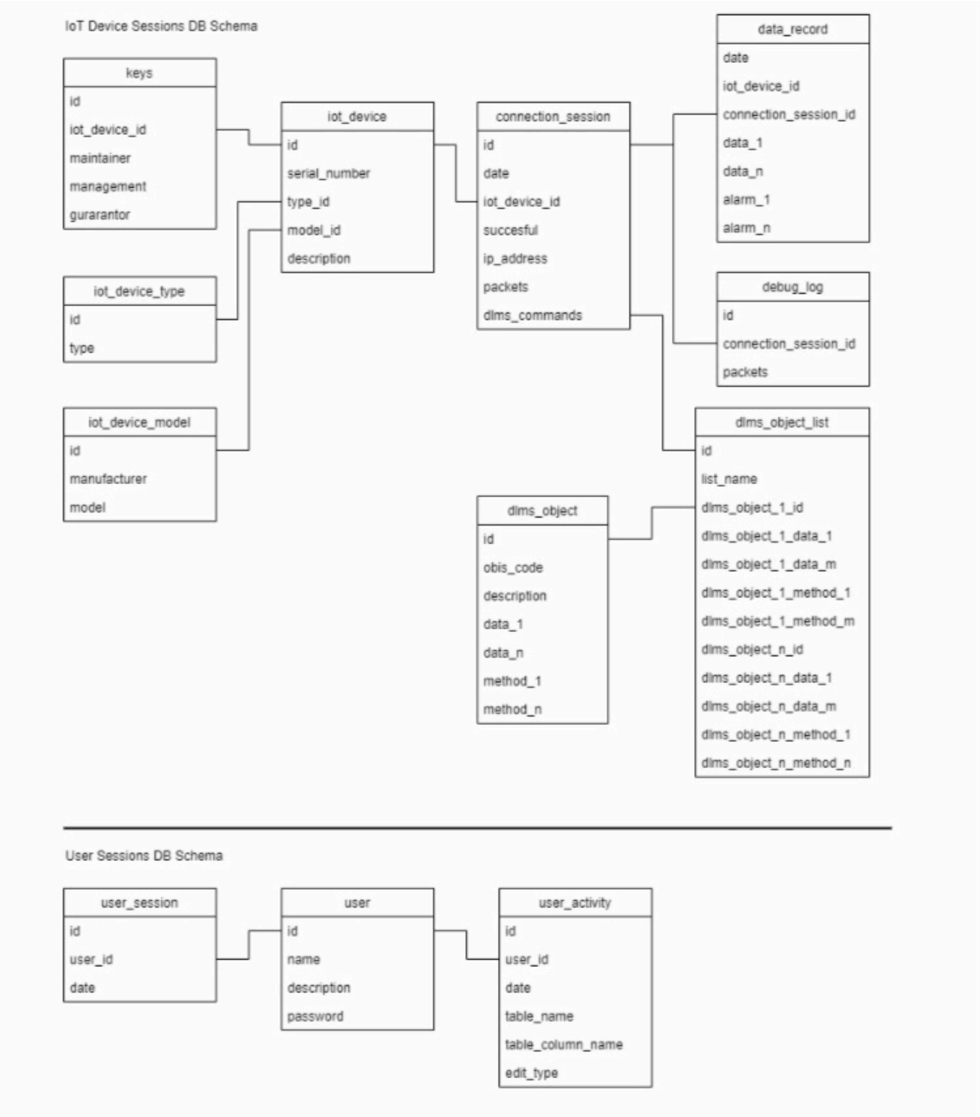
Even if the required services are only a subset of a complete web application, the software should be developed according to object oriented programming, to achieve a high level of abstraction, reusability and scalability.



5. A potential class diagram scheme

Below we represent a possible implementation of a class diagram for the underlying database and corresponding object model.

This diagram should not intended as a full specification of the web application architecture, but as a high level overview of the object model.
During the development of the project, the student will add more level of details, and will focus only on the subset of the model needed to achieve the minimum requirements for this project.


IoT Device Sessions DB Schema
keys
id
iot_device_id
maintainer
management
gurarantor
iot_device_type
id
type
iot_device_model
id
manufacturer
model
iot_device
id
serial_number
type_id
model_id
description
connection_session
id
date
iot_device_id
succesful
ip_address
packets
dlms_commands
data_record
date
iot_device_id
connection_session_id
data_1
data_n
alarm_1
alarm_n
debug_log
id
connection_session_id
packets
dlms_object_list
id
list_name
dlms_object_1_id
dlms_object_1_data_1
dlms_object_1_data_m
dlms_object_1_method_1
dlms_object_1_method_m
dlms_object_n_id
dlms_object_n_data_1
dlms_object_n_data_m
dlms_object_n_method_1
dlms_object_n_method_n
dlms_object
id
obis_code
description
data_1
data_n
method_1
method_n
User Sessions DB Schema
user_session
id
user_id
date
user
id
name
description
password
user_activity
id
user_id
date
table_name
table_column_name
edit_type