Wrocław University
of Science and Technology

Faculty of Computer Science and Management

Report

Work time registration system
using the Internet of Things technology

Course:	Introduction to IoT Laboratory
Author's first and last name:	Vladyslav Gavryliuk
Index No:	245603
Semester:	4
Date:	May 2020 r.
Laboratory Teacher:	mgr inż. Dariusz Gall

HR EXCELLENCE IN RESEARCH
Wrocław University
of Science and Technology
Wybrzeże Wyspiańskiego 27
50-370 Wrocław
wiz.pwr.edu.pl

Table of contents: **N. of page**

Design requirements:

Functional requirements:

-Client:

- User has to log in by pressing the button.
- System asks him login and password.
- If data is correct, then connection is accepted, otherwise no.
- User is able to send Card id by client.
- If press the button “Check card” system sends message with data of given ID.

-Server:

- User can create new employee or delete an existing employee.
- User can give or take out card of employee.
- User can create in the end the report with all enters and exits.
- Press “Create report” to generate output.csv.

Non-functional requirements:

-Openssl:

- Use 2048-Bit Private Keys.
- Protect Private Keys.
- Ensure Sufficient Hostname Coverage
- Obtain Certificates from a Reliable CA
- Use Strong Certificate Signature Algorithms.

-Mosquitto:

- Never use a leading forward slash.
- Never use spaces in a topic.
- Keep the topic short and concise.
- Use only ASCII characters, avoid non printable characters.
- Embed a unique identifier or the Client Id into the topic.
- Don't subscribe to #
- Don't forget extensibility.
- Use specific topics, not general ones.

Description of system architecture:

First of all, we have to write login and password. (auth/login)
After it was done by client(user), data sends to server.
Server reads it and if data is correct sends to topic. (auth/res)
After that, client must receive message (Success or not).
Next step is sending id card by client to topic(card/id).
It has to be the number of card ID which exists in the system, otherwise it won't work.
If there exists given ID, then we can manage/see data of employee.

Description of implementation and solutions used:

***Code fragments of the most important application functions with a description**:*

- seedr.py – use library (from faker Import Faker), fake = Faker() this is using for auto-generating full names for employees.

```
- DB.py - CREATE TABLE records (
    id INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE,
    card_id INT NOT NULL,
    worker_id INT,
    timestamp DATETIME DEFAULT CURRENT_TIMESTAMP
```

The same 2 tables creates for cards and workers.

Code fragments of the MQTT implementation with a description:

emplr.*py* – main method is receiving message from broker is on_message(). It depends on the topic on which the message was sent, we compare the correctness of the input data.
issuer.py – sends number of card id to the topic card/id.

Code fragments of the implementation of encryption and authentication with a description:

Information goes from client to broker , then server reads data from broker and reply to broker, finally client sees did he write correct login and password or not , if yes he is able to do some work with given opportunities.

```
    from dotenv import load_dotenv - responds for process
    client.tls_set(os.getenv('CERTIFICATE'))
```

Other information :

In this project the client independent of the server , so that's why the script that can be used on any device that has no project configuration.

Description of operations and presentation of the interface:

Description of how to install and run the application:

1 - Install Python 3 and newer versions.

2 - Install Mosquitto(to use broker)

3 - Install Openssl

4 - Create folder in which will be saved ca.crt, server.crt, server.key.

5 - Generate certificates or copy sample files

6 - Change data in mosquitto.conf.

7 - Create file.env.

8 - To the folder copypaste file ca.crt

9 - Install needed packets for properly working : pip install -r requirements.txt

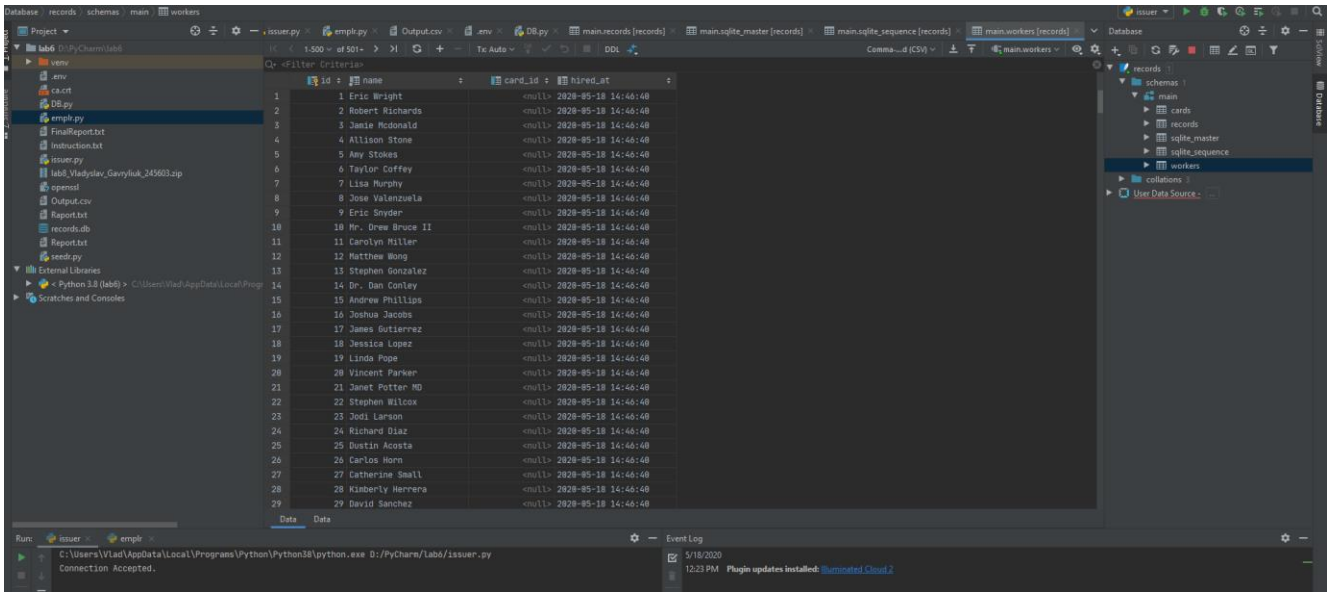
10 - Run db.py to create database with employees.

11 - Run issuer.py

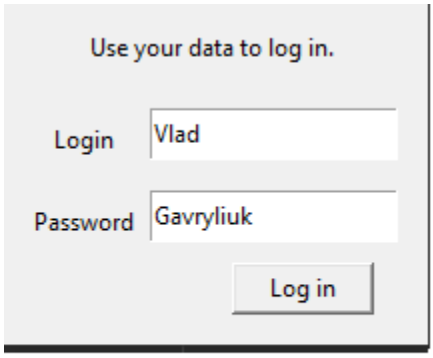
12 - Run emplr.py

Screenshots showing the operations of the application with a description:

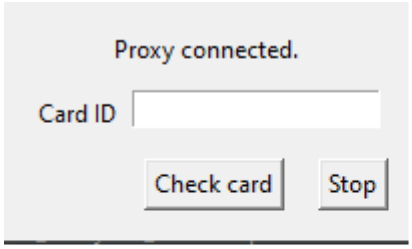
Database :



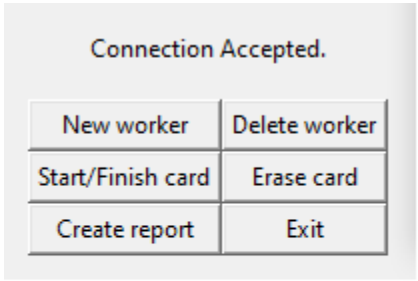
Process of log in:



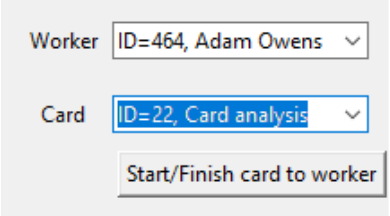
If data is correct:



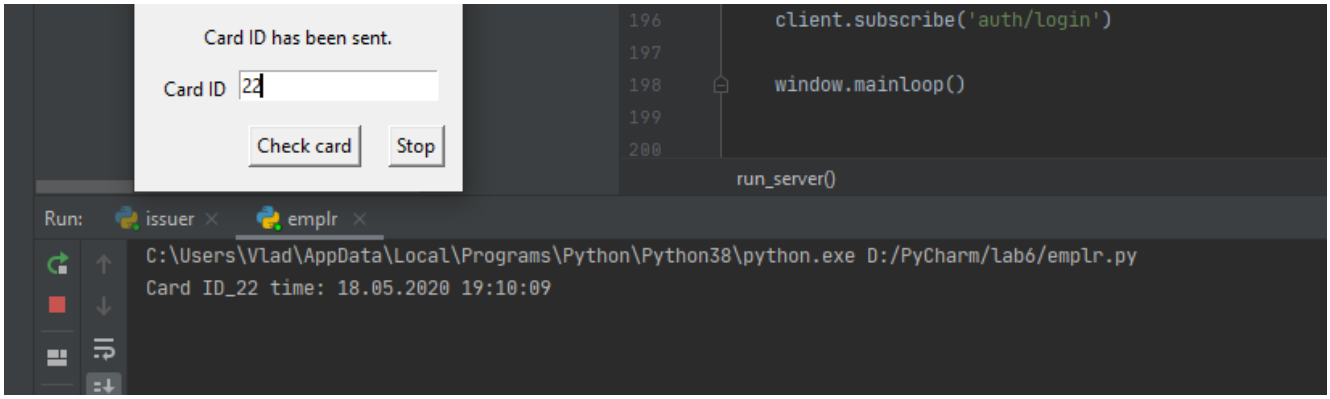
If connection is accepted :



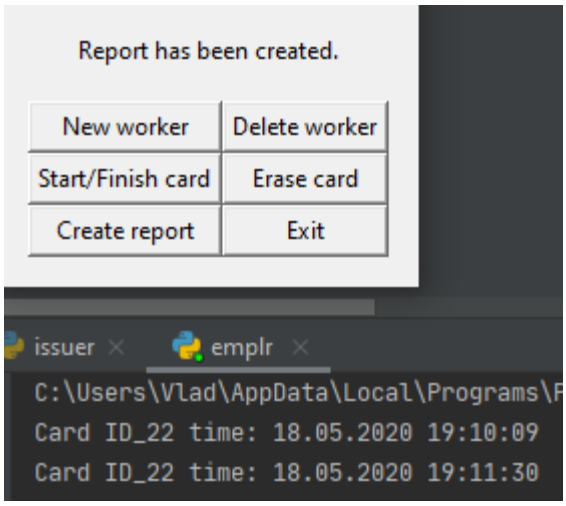
Start of finish work for someone:



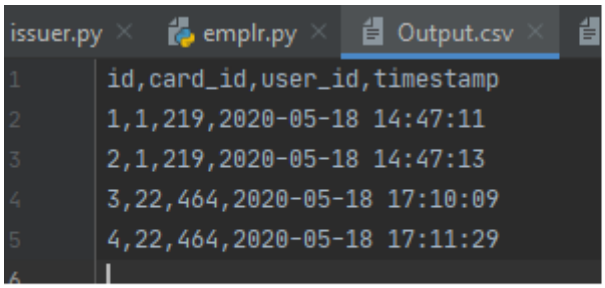
Start work of employee:



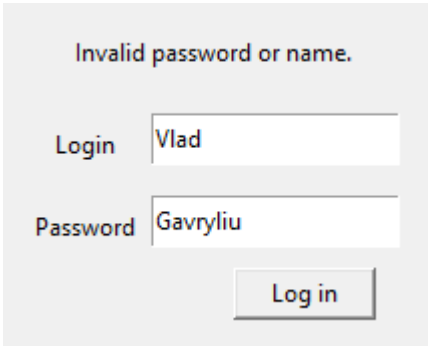
Create Report



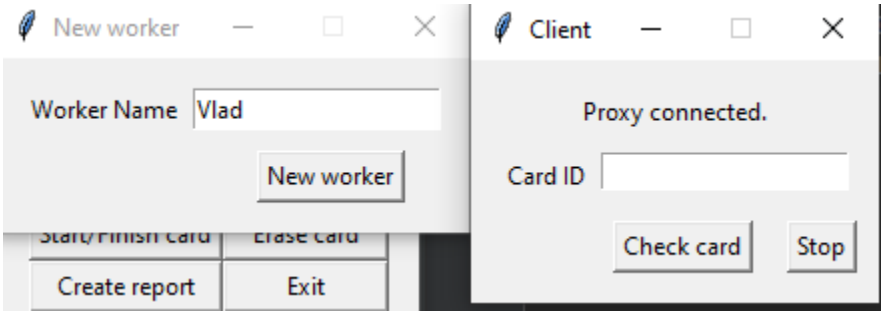
Output.csv



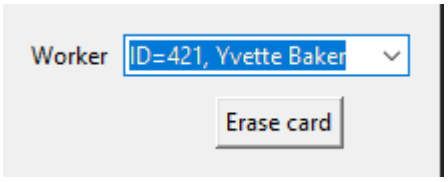
If data is incorrect:



To create new Worker:



If we want to erase card for someone.



Other information:

Settings of broker :

-BROKER=DESKTOP-GI6OBR7
-PORT=8883
-CERTIFICATE=ca.crt

-CLIENT_USER=client
-CLIENT_PASS=password

-SERVER_USER=server
-SERVER_PASS=admin

Settings of authorization :
-USER=Vlad
-PASS=Gavryliuk

Summary:

Summary of project implementation. Compliance of the project with the requirements:

- There were used libraries, such as :
    import sqlite3(DB-API 2.0 interface for SQLite databases)
    import tkinter as tk(Python interface to Tcl/Tk , to create terminals with buttons)
    import paho.mqtt.client as mqtt(To run broker)
    from datetime import datetime(to know current time)
    from dotenv import load_dotenv(Reads the key-value pair from .env file and adds them to environment variable.)
    import csv(Read data from file 'filename.csv').
-The aim of project is :
    to create a client.
    to create a broker.
    to create a server.
    to connect database with all generated employees.
    install and modify mosquitto.
    make implementation of the MQTT protocol for communication between server and client.
    install and modify openssl.
    manage login and password for client.(for security - authorization & authentication)
- The project is done as it was required
- In summary, I can say that after all laboratories which I've done I learned and understood meaning of broker , server , client and how it works.

Notes on implementation difficulties encountered:

- Due to the lack of access to the Raspberry Pi understand the simulation of card downloading using the graphical interface on the client.

Comments on proposals for changes or development of the project:

- In the future it is possible to build a project database.

- When applying the card, for example, open the door to the company office.

Literature:

Information about tasks of laboratories **:** https://eportal.pwr.edu.pl/course/view.php?id=1740
OpenSSL: https://www.openssl.org/source/
MQTT Version 5.0: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html

Annex

The whole code is added to zip file.