

Building Management System

(Sistemul de management integrat al unei cladiri)

Prezentarea proiectului:

Proiectul va consta intr-o aplicatie cu Interfata Grafica, in care userul va avea acces deplin la instalatiile electrice ale cladirii respective. In primul rand sa definim ce inseamna un Building Management System , in ziua de astazi cunoscut ca si BMS.

Detalii Tehnice:

- Mediul de dezvoltare: JetBrains PyCharm Community Edition, Arduino IDE(Optional)
- Limbaj: Python
- Baza de date: SQLite
- Versiunea: Python 3.7

Resurse software necesare:

- Sistem de operare: Windows 10, Ubuntu
- Baza de date: curent_users.sqlite3
- Librarii:
 - ◆ Intefata Grafica:
 - Tkinter (ceva mai complex)
 - PyQt
 - ◆ Programarea Senzorilor :
 - PySerial -> Librarie folosita pentru mediul de dezvoltare Arduino
 - ◆ Convertirea programului intr-un rulabil:
 - PyConvert
- Descararea software-ului necesar placii de dezvoltare

Resurse hardware necesare:

- Procesor: 2 Gh
- Memorie RAM: 4 Gb
- Memorie HDD: 250Gb
- Placa de dezvoltare compatibila cu Arduino

In ce consta un BMS?

BMS (Building Management System) - Acest sistem poate integra semnalizarile de la mai multe tipuri de instalatii cum ar fi:

- Instalatii Electrice
- Supraveghere Video
- Control Accesul in cladire
- Instalatia de detectare si stingere a incendiului

Totodata prin intermediul BMS , in masura in care sistemele care itegreaza permit acest lucru, se pot prelua semnalizari si se pot transmite comenzi de la distanta.

Estimarea performantelor:

Perfomanta depinde de hardware-ul utilizatorului. Desi in ziua de astazi spectul hardware nu mai este o problema pentru marea majoritate a utilizatorilor, deoarece pana si cel mai mediocru calculator are o putere de calcul uriasa.

Aplicatia nu va avea un trafic mare de utilizatori, cum denota si numele aplicatiei doar un numar restrans de persoane va avea acces la acest system.

Estimand circa 2-3 persoane care se vor ocupa cu administrarea cladirii cat si owner-ul cladirii.

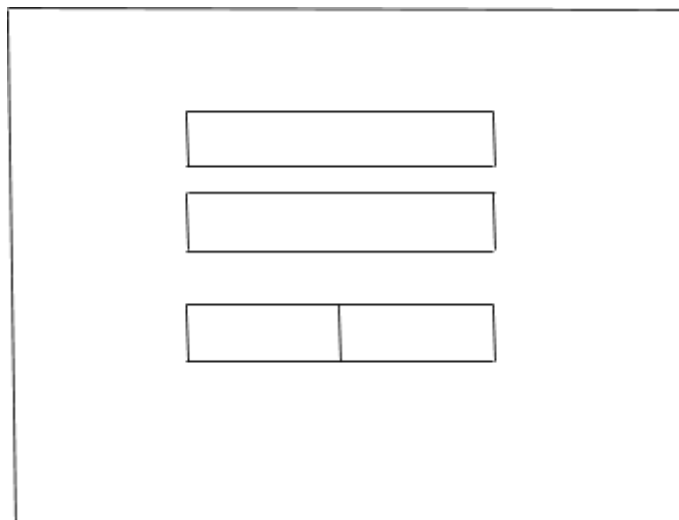
Modul de concepere si construire a aplicatiei

Interfata Grafica cu Utilizatorul

Scheletul:

→ Main Frame:

- ◆ Label User
- ◆ Label Password
- ◆ Button Login as Administrator
- ◆ Button Login as Owner



A diagram of the Main Frame, which is a large rectangle containing three smaller rectangular input fields stacked vertically in the center. The top field is a single-line text box. The middle field is a single-line password box. The bottom field is a two-line text box, represented by two adjacent rectangles of the same width.

→ Second Frame:

- ◆ Label Control Acces
- ◆ Label Supraveghere Video
- ◆ Label Det. Incendiu
- ◆ Label Log Monitorizare User(Owner Only)
- ◆ Button Logout

C.Acc	Frame
Sup.Vid	
Ins.Inc	
Ist.User	

A diagram of the Second Frame, which is a large rectangle divided into two parts. On the left is a vertical stack of four smaller rectangles, each containing text: 'C.Acc', 'Sup.Vid', 'Ins.Inc', and 'Ist.User'. To the right of this stack is a large rectangular area labeled 'Frame' in the center.

→ Control Acces:

- ◆ Frame pentru Log-ul persoanelor care au intrat intr-o anumita zona din cladire.

C.Acc	Log
	Log
	Log
	Log
	Log
	Log

→ Supraveghere Video:

- ◆ Frame pentru supraveghere live
- ◆ Frame pentru supraveghere live in zone restrictionate(Owner Only)
- ◆ Frame pentru inregistrari stocate de camera in momentul cand a detectat miscarea.

Vd.Sup	Live Vid	Stored Rec



→ Det. Incendiu:

- ◆ Frame pentru Log-ul zona in care sa detectat incendiul.

	Log
	Log
	Log
Ins.Inc	Log
	Log
	Log

→ Istoric User:

- ◆ Frame pentru Log-ul user-ilor(nume, data, ora, timp de activitate)

	Log
	Log
	Log
	Log
	Log
Ist.User	Log

Perspectiva de Dezvoltare:

Neavand o cladire la dispozitie pentru a ne luat momentan Input-uri vom folosi place de dezvoltare Arduino , ca si scop principal pentru testarea si demonstrarea aplicatiei.

Control Acces-ul in cladire:

Ce inseamna Control Acces intr-o cladire?

Controlul accesului in ziua de astazi este necesar in cadrul firmelor cu mai multi angajati sau a corporatiilor. Acest lucru se face pentru a pona timpul pe care o persoana il petrece intr-o locatie sau pentru a restrictiona accesul anumitor persoane de a avea acces la informatiile ce nu trebuie dezvaluite unui alt departament si pentru a prevenii furturile de informatii sau bunuri.

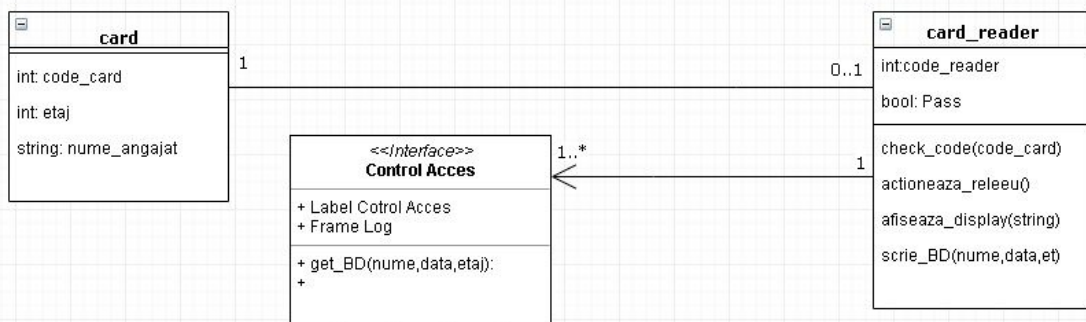
Acesta masura se foloseste prin cartelele de acces ce determina in functie de angajat , zona firmei in care are acces. Dezavantajul acestor cartele magnetice este ca sunt usor de pierdut sau furat, asa ca o alternativa sigura ar fi un cod de acces.

Resurse Hardware Arduino:

- Placa de dezvoltare Arduino
- Real Time Clock (RTC)
- Mini-Display
- FID Reader (Radio-Freq Identificator)
- FID Card(Radio-Freq Identificator Card)



Abstractizare:



Ca si imagine de ansamblu. In ziua de astazi securitatea usilor din cladirile moderne sunt echipate cu sensori magnetici. In momentul in care se programeaza Reader-ul si Card-ul i se atribuie o valoare sau valori(in cazul in care in incaperea respectiva au acces mai multe persoane) pre-stabilita ca si cod de acces. Daca codul card-ului corespunde cu cel al Reader-ului atunci actioneaza un releu care inchide circuitul senzorului magnetic iar persoana respectiva poate intra. Desi pare compilcat defapt se reduce la instructiunea IF eventual switch() spre ex:

```
cod_reader = 165;
cod_card = 195;
daca cod_card == cod_reader atunci
    afisare_Display("Poti Intra") //pass
    actioneaza_releu(); //inchide circuit senzor magnetic
    get_NumePers(); //preia numele persoanei
    get_DateTime(); //data si ora
    scrie_BD(time,NumePers,cod_reader);

altfel
    afisare_Display("Card Invalid") //pass
    get_NumePers(); //preia numele persoanei
    get_DateTime(); //data si ora
    scrie_BD();
```

Supravegherea Video:

Ce presupune Supravegherea Video:

Dupa cum spune si numele - supraveghere video este un sistem de servicii de securitate in cladiri. Practic este un ansamblu de echipamente(camere video) interconectate in scopul preluarii unor imagini din diferite zone. Desi in ziua de astazi echipamentul video a evoluat enorm ca si tehnologie spre ex: exista asa numitele camere Wi-Fi care nu mai necesita un cablu de 100m tras prin cladire, sau asa numitele camere IP care pot fi accesate printr-un REMOTE Server.

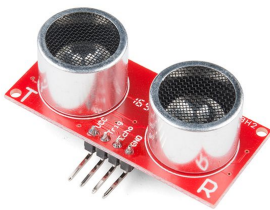
Resurse Hardware Arduino:

→ Placa de dezvoltare Arduino

→ Ultrasonic Sensor



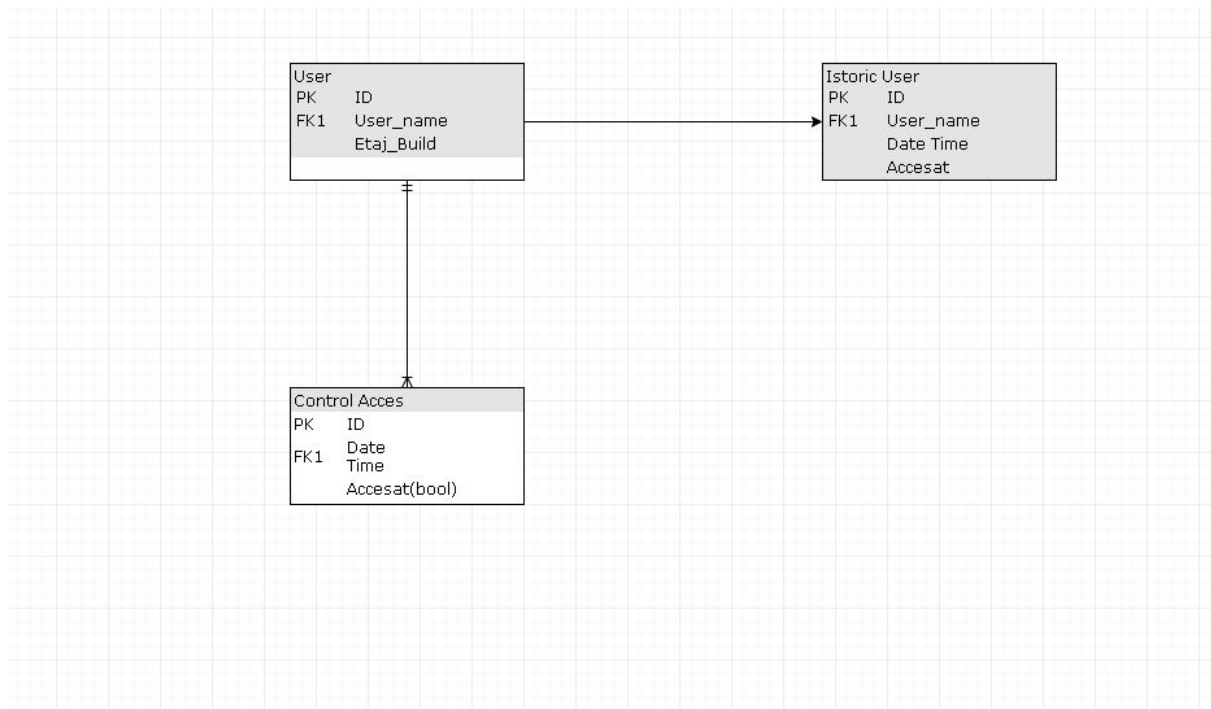
→ Mini-camera Wi-Fi



Abstractizare:

Dupa cum am aratat in scheletul GUI, acesta va fi impartit in 2 Frame-uri, unul pentru LIVE Video , iar celalalt va stoca pe HDD ulterior se poate vedea,strege. Pentru a realiza aceasta functie avem nevoie de o Camera Video, in care doar vom vedea imaginea LIVE(spre ex o camera de telefon), iar pentru cel deal doilea Frame avem nevoie pe langa o camera video si un senzor de miscare. Care va fi logica din spate in momentul in care senzorul detecteaza miscarea un releu va actiona camera video si acesta va incepe sa inregistreze si va stoca pe HDD. De ce facem asta? Este un mijloc mult mai eficient in care nu ocupam degeaba spatiul HDD cu fisiere inutile, doar ganditiva cat de mult spatiu ar costa sa tii un video de 24 ore uitandu-te la o masa spre ex plus timpul operatorului in care trebuie sa stea si sa deruleze in cazul in care este nevoie petru a detecta momentul respectivului incident. Cand acesta opate avea insta cu ajutorul senzorului exact cele 4-5 minute de activitate din cele 24 ore. Ulterior aplicatia va sterge inregistrarile mai vechi de 30 de zile, conform legii inregistrarile mai vechi de 30-60 sunt penalizate.

Istoric User:



DIstoricul user-urilor va fi o functie a aplicatiei accesibila doar de catre Owner, acesta va avea acces la data , ora si timpul in care administratorul(ii) le-am petrecut in aplicatie. Desi pare o functie banala poate fi folosita pentru pontaj-ul administratorilor, iar in cazul unui eveniment neplacut spre ex(spargerea cladirii) poate fi folosita pe post de "martor" in care specifica exact cine era de "garda" in momentul respectiv.