



Laboratorio di Sviluppo di Applicazioni per IoT
a.a. 2022-2023

SMART QUEUE

Studenti:

Turco Francesco

Schettini Francesco

Docente:

Antonio Guerriero

Idea di progetto (Specifica dei requisiti)

Si vuole implementare un'applicazione IoT per la gestione della fila di accesso ad una mostra d'arte.

L'amministratore può accedere al sito tramite login.

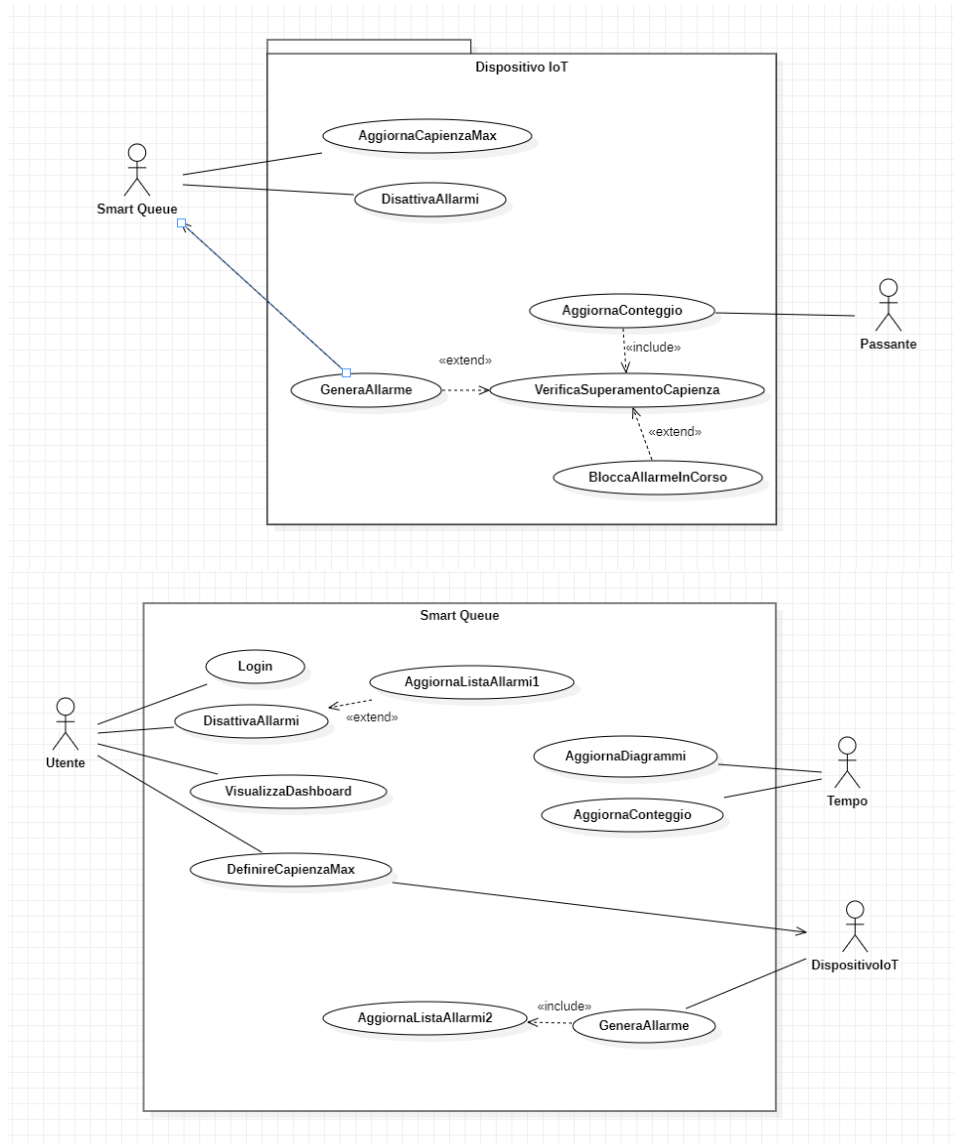
Il dispositivo serve a rilevare il passaggio di una persona, così da tenere traccia del numero totale di persone all'interno di un edificio.

Il dispositivo vieta l'accesso se si è superato il numero massimo di capienza, in particolare segnala l'allarme finché non viene rispettata la capienza massima oppure interrotto dall'amministratore.

L'amministratore può settare tramite dashboard la capienza massima e disattivare tutti gli allarmi.

Inoltre, l'amministratore, attraverso la dashboard, potrà visualizzare l'occupazione attuale (aggiornata ogni dieci secondi), la lista degli allarmi e infine il numero di persone che hanno visitato il museo nell'arco di una giornata mediante opportuni diagrammi che tracciano l'andamento orario delle visite, aggiornati ogni inizio ora.

Diagrammi dei casi d'uso



Scenari dei casi d'uso

Smart Queue Dispositivo IoT

CASO D'USO	Login
Attore primario	Utente
Descrizione	Accesso al sito web tramite autenticazione
Pre-Condizioni	-L'utente dev'essere connesso alla rete locale
Sequenza degli eventi	1. l'utente inserisce l'e-mail 2. l'utente inserisce la password 3. l'utente clicca su accedi
Post-Condizioni per il successo	-L'utente ha inserito delle credenziali valide
Post-Condizioni per il fallimento	-L'utente ha inserito delle credenziali non valide -Problema di connessione
Evento innescante	Apertura del sito dall'utente
Include	
Estende	

CASO D'USO	DisattivaAllarmi
Attore primario	Utente
Descrizione	Disattivazione di tutti gli allarmi. Se in corso, lo fa smettere di suonare.
Pre-Condizioni	-Login effettuato con successo -Accesso alla rete (utente)
Sequenza degli eventi	1. l'utente clicca sul pulsante di disattivazione
Post-Condizioni per il successo	-L'utente ha cliccato sul pulsante
Post-Condizioni per il fallimento	-Problema di connessione
Evento innescante	
Include	
Estende	

CASO D'USO	AggiornaListaAllarmi1
Attore primario	Utente

Descrizione	Aggiornamento della lista degli allarmi
Pre-Condizioni	-Allarme appena disattivato dall'utente (DisattivaAllarmi) -Allarme stato in corso -Accesso alla rete (utente)
Sequenza degli eventi	0. Cessa l'allarme 1. La lista viene aggiornata
Post-Condizioni per il successo	
Post-Condizioni per il fallimento	-Problema di connessione
Evento innescante	Disattivazione dell'allarme da parte dell'utente
Include	
Estende	DisattivaAllarmi

CASO D'USO	VisualizzaDashboard
Attore primario	Utente
Descrizione	Visualizzazione interfaccia web con relative informazioni
Pre-Condizioni	-Login effettuato con successo -Accesso alla rete (utente)
Sequenza degli eventi	
Post-Condizioni per il successo	
Post-Condizioni per il fallimento	-Problema di connessione
Evento innescante	-Login effettuato con successo
Include	
Estende	

CASO D'USO	DefinireCapienzaMax
Attore primario	Utente
Attore secondario	Dispositivo IoT
Descrizione	Definire la capienza massima dell'edificio
Pre-Condizioni	-Login effettuato con successo -Accesso alla rete (utente) -Accesso alla rete (dispositivo IoT)
Sequenza degli eventi	1. L'utente scrive la capienza massima 2.L'utente clicca sull'icona
Post-Condizioni per il successo	-L'utente ha inserito un numero valido
Post-Condizioni per il fallimento	-Problema di connessione -L'utente ha inserito un numero minore uguale di 0
Evento innescante	
Include	
Estende	

CASO D'USO	AggiornaDiagrammi
Attore primario	Tempo
Descrizione	Aggiornamento dei diagrammi visualizzabili dalla dashboard con il nuovo report dati
Pre-Condizioni	-Accesso alla rete (dispositivo IoT)
Sequenza degli eventi	0. Il dispositivo manda il report (ogni ora) 1. I diagrammi vengono aggiornati
Post-Condizioni per il successo	
Post-Condizioni per il fallimento	-Problema di connessione
Evento innescante	Raggiungimento orario prestabilito: inizio dell'ora
Include	
Estende	

CASO D'USO	AggiornaConteggio
Attore primario	Tempo

Descrizione	Aggiornamento del valore numerico conteggio occupazione attuale, visualizzabile dalla dashboard
Pre-Condizioni	-Accesso alla rete (dispositivo IoT)
Sequenza degli eventi	il valore viene aggiornato (ogni 10 secondi)
Post-Condizioni per il successo	Il dispositivo manda il report (ogni variazione)
Post-Condizioni per il fallimento	-Problema di connessione
Evento innescante	Periodico 10 secondi
Include	
Estende	

CASO D'USO	GeneraAllarme
Attore primario	DispositivoIoT
Descrizione	Generazione dell'allarme
Pre-Condizioni	-Accesso alla rete (dispositivo IoT) -Non disattivato dall'utente (DisattivaAllarmi)
Sequenza degli eventi	Il dispositivo segnala l'allarme
Post-Condizioni per il successo	
Post-Condizioni per il fallimento	-Problema di connessione
Evento innescante	Superamento della soglia
Include	AggiornaListaAllarmi2
Estende	

CASO D'USO	AggiornaListaAllarmi2
Attore primario	DispositivoIoT
Descrizione	Aggiornamento della lista degli allarmi
Pre-Condizioni	-Allarme appena terminato in quanto si è raggiunto un numero adeguato di occupazione -Accesso alla rete (dispositivo IoT)
Sequenza degli eventi	0. Cessa l'allarme 1. La lista viene aggiornata
Post-Condizioni per il successo	
Post-Condizioni per il fallimento	-Problema di connessione
Evento innescante	Raggiungimento della soglia (inferiore)
Include	
Estende	

CASO D'USO	AggiornaCapienzaMax
Attore primario	Smart Queue
Descrizione	Aggiornamento della capienza massima da remoto sulla dashboard
Pre-Condizioni	-Accesso alla rete (dispositivo IoT)
Sequenza degli eventi	0. Il sistema ha comunicato il valore 1. Aggiornamento (caso successo)
Post-Condizioni per il successo	- Accesso alla rete - Il sistema ha comunicato un numero ammissibile
Post-Condizioni per il fallimento	- Problema di connessione - Il sistema ha comunicato un numero minore uguale di 0
Evento innescante	-Premere bottone sulla dashboard
Include	
Estende	

CASO D'USO	DisattivaAllarmi
Attore primario	Smart Queue
Descrizione	Disattivazione degli allarmi
Pre-Condizioni	-Accesso alla rete (dispositivo IoT)
Sequenza degli eventi	Il sistema ha comunicato di disattivare gli allarmi Se in corso, lo blocca
Post-Condizioni per il successo	- Accesso alla rete
Post-Condizioni per il fallimento	-Problema di connessione
Evento innescante	Richiesta da remoto (dashboard, pulsante)
Include	
Estende	

CASO D'USO	AggiornaConteggio
Attore primario	Passante

Attore secondario	Smart Queue
Descrizione	Aggiornamento valore occupazione al passaggio del passante
Pre-Condizioni	-Accesso alla rete (dispositivo IoT)
Sequenza degli eventi	0. passa il turista 1. sensori rilevano passaggio 2. aggiornamento dell'occupazione attuale (entrata -> ++, uscita -> --)
Post-Condizioni per il successo	-Sensori funzionanti
Post-Condizioni per il fallimento	-Problema ai sensori
Evento innescante	Passaggio turista (entrata o uscita)
Include	VerificaSuperamentoCapienza
Estende	

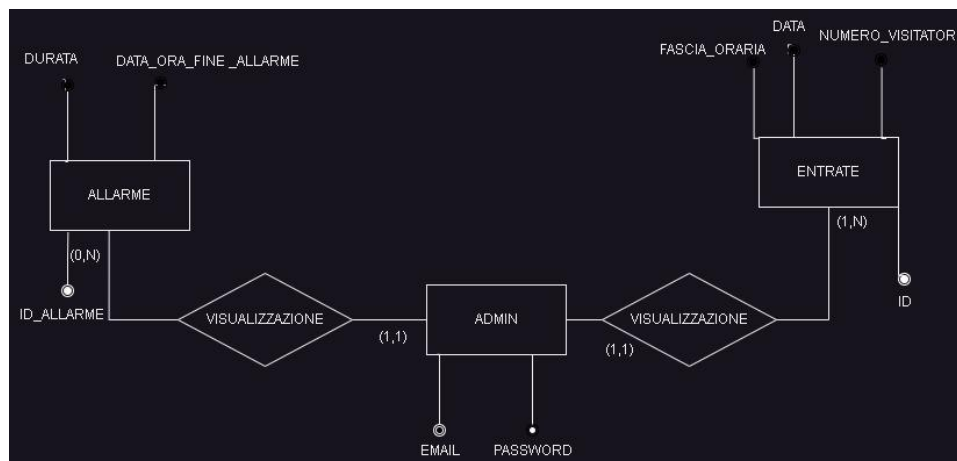
CASO D'USO	VerificaSuperamentoCapienza
Attore primario	Passante
Attore secondario	SmartQueue
Descrizione	Verifica del superamento della soglia prevista

CASO D'USO	BloccaAllarmeInCorso
Attore primario	Passante
Attore secondario	Smart Queue
Descrizione	Disattivazione dell'allarme
Pre-Condizioni	-Accesso alla rete (dispositivo IoT) -La soglia minima è stata raggiunta
Sequenza degli eventi	
Post-Condizioni per il successo	- Accesso alla rete
Post-Condizioni per il fallimento	-Problema di connessione
Evento innescante	Raggiungimento soglia (inferiore)
Include	
Estende	VericaSuperamentoCapienza

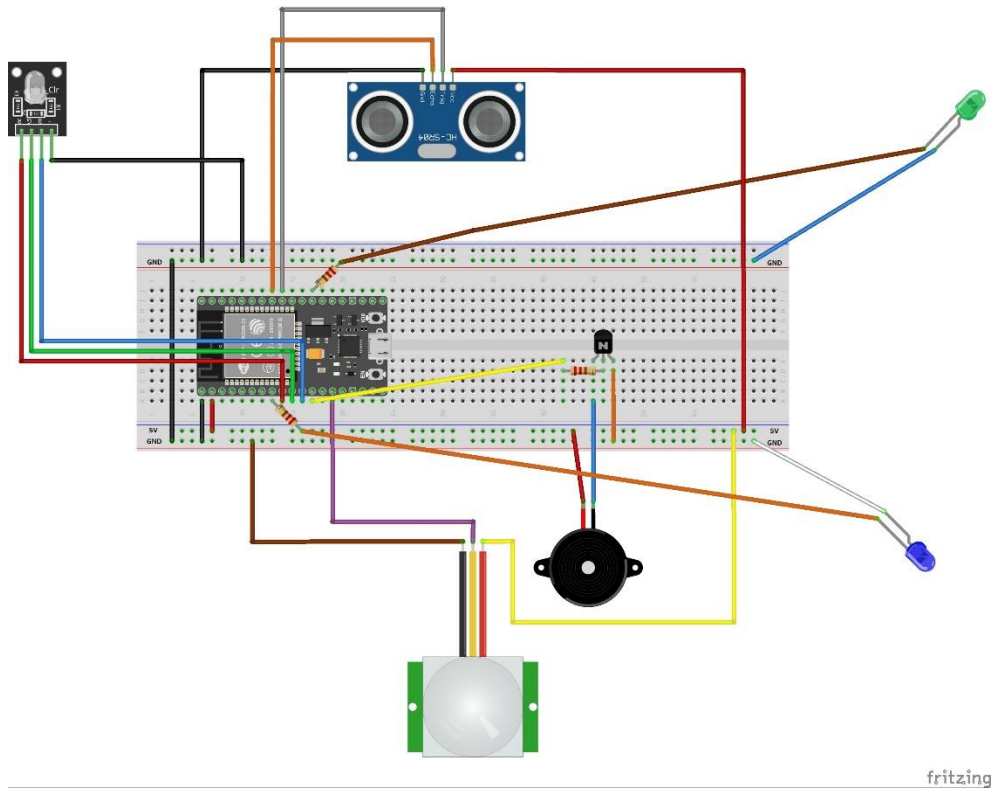
CASO D'USO	GeneraAllarme
Attore primario	Passante
Attore secondario	Smart Queue
Descrizione	Attivazione dell'allarme

Pre-Condizioni	-Accesso alla rete (dispositivo IoT) -La soglia massima è stata raggiunta -Allarmi non disattivati (DisattivaAllarmi)
Sequenza degli eventi	
Post-Condizioni per il successo	- Accesso alla rete
Post-Condizioni per il fallimento	-Problema di connessione
Evento innescante	Raggiungimento soglia (superiore)
Include	
Estende	VericaSuperamentoCapienza

Diagramma E-R



Schema elettronico del progetto



COMPONENTI UTILIZZATI:

- ESP32 + adapter (non riportato nello schema) + breadboard
- Sensore Pir (x1)
- Distanziometro (x1)
- Buzzer + BJT type n + resistenza 1k Ω (x1)
- Led + resistenza 220 Ω (x2)
- Led RGB (x1)

Parti di codice più importanti

1) Codice Arduino

-1.1) main

(main.ino)

```
main.ino
1
2 /*Classi e Librerie*/
3 /*mie classi*/
4 #include <DistanceMeter.h>
5 #include <Led.h>
6 #include <SensorPIR.h>
7 #include <LedRGB.h>
8 #include <Buzzer.h>
9
10 /*mie librerie*/
11 #include "Arduino.h"
12 #include <WiFi.h>
13 #include <WiFiClient.h>
14 #include <HTTPClient.h>
15 #include <String.h>
16
17
18
19 /*etichette*/
20 #define TIME_LAMP 500 //tempo di lampeggio dei led
21
22 const char *ssid = "realme"; //Enter the router name
23 const char *password = "napolii926"; //Enter the router password
24
25 const char* scriptGetThreshold = "http://192.168.56.98/smart_queue/getCardinalityThreshold.php";
26 const char* scriptSetOccupation = "http://192.168.56.98/smart_queue/setOccupation.php";
27 const char* scriptSetAlarmDB = "http://192.168.56.98/smart_queue/setAlarmDB.php";
28 const char* scriptGetAlarmActivation = "http://192.168.56.98/smart_queue/getAlarmActivation.php";
29 const char* scriptGetActualMinutes = "http://192.168.56.98/smart_queue/getActualMinutes.php";
30 const char* scriptSetHourVisitorsDB = "http://192.168.56.98/smart_queue/setHourVisitorsDB.php";
31
32
33 /*objects*/
34 /*components*/
35 Led Lout(25);
36 Led Lin(4);
37 LedRGB RGB(26, 27, 14);
38 DistanceMeter DM(5, 18);
39 SensorPIR PIR(13);
40 Buzzer B(12);
41
42
43
44 /*variabili*/
45 int peoplePresence = 0; //presenza attuale
46 int cardinalityThreshold; //occupazione massima da non superare
47 int peopleThisHour = 0; //totale visitatori quest'ora
48
49 int startTime, endTime; //tempi di inizio e fine del suono dell'allarme (per calcolare durata allarme)
50
51 bool alarmActivationOn; //accende o spegne definitivamente l'allarme (può entrare chiunque)
52 bool alarmStateOn = false; //stato attuale dell'allarme (relativo al se sta suonando o no)
53
54 int actual_min;
55 bool FLAG_ALREADYREAD = false; //è alto se min = 0 ma ha già inviato la richiesta di aggiornare i DB
56 //serve a mandare la richiesta una volta sola sul minuto 0*/
57
58
59 /*Funzioni*/
60 void gestioneLedRGB();
61 void alarmGeneration();
62 void alarmStop();
63 String timeFormat(); //formatta da secondi a stringa di time
64
65 String get_from_server(const char* serverName);
66 void send_to_server(String getData, const char* serverName);
67
```

```
main.ino
67
68
69 void setup() {
70     Serial.begin(9600);
71
72     //connessione wi-fi
73     WiFi.begin(ssid, password);
74     while (WiFi.status() != WL_CONNECTED)
75     {
76         delay(1000);
77         Serial.println("Connecting to WiFi..");
78     }
79
80     /*inizializzazione delle variabili dai salvataggi*/
81     cardinalityThreshold = get_from_server(scriptGetThreshold).toInt();
82     alarmActivationOn = get_from_server(scriptGetAlarmActivation).toInt();
83     actual_min = get_from_server(scriptGetActualMinutes).toInt();
84 }
85
86
87 void loop() {
88
89     /*rilevazione ingresso*/
90     if(DM.detectedMovement())
91     {
92         //Serial.print("Trovato che entra!!");
93         Lin.lampeggia(TIME_LAMP);
94         peoplePresence++;
95         peopleThisHour++;
96
97         //riporta al server l'occupazione attuale
98         send_to_server(String(peoplePresence), scriptSetOccupation);
99     }
100
101     /*rilevazione uscita*/
102     if(PIR.detectedMovement() && peoplePresence > 0) //movimento in uscita
103     {
104         //Serial.print("Trovato che esce!!");
105         peoplePresence--;
106         Lout.lampeggia(TIME_LAMP);
107
108         //riporta al server l'occupazione attuale
109         send_to_server(String(peoplePresence), scriptSetOccupation);
110     }
111
112     /*gestione degli allarmi*/
113     if(alarmActivationOn) //gestisco gli allarmi se non sono totalmente disattivati dalla dashboard
114     {
115         /*gestione generazione allarme*/
116         if(!alarmStateOn && peoplePresence > cardinalityThreshold)
117             alarmGeneration();
118
119         if(alarmStateOn && peoplePresence <= cardinalityThreshold)
120             alarmStop();
121
122         /*gestisce il semaforo*/
123         gestioneLedRGB();
124     }
125     else
126     {
127         RGB.turnOff(); //spengo il semaforo
128     }
129
130     if(alarmStateOn) //caso limite: utente disattiva allarme durante allarme in corso.
131         alarmStop();
132 }
133
134
```

```

135
136 /*aggiornamento grafici*/
137 if(actual_min == 0 && !FLAG_ALREADYREAD)
138 {
139     send_to_server(String(peopleThisHour), scriptSetHourVisitorsDB);
140     peopleThisHour = 0;
141
142     FLAG_ALREADYREAD = true; //flag di avvenuta lettura della presenza oraria, lo metto a true
143 }
144 else if(FLAG_ALREADYREAD && actual_min != 0){
145     FLAG_ALREADYREAD = false; //lo rimetto a false per il prossimo controllo
146 }
147
148
149
150
151
152 /*aggiornamento variabili dalla dashboard*/
153 cardinalityThreshold = get_from_server(scriptGetThreshold).toInt();
154 alarmActivationOn = get_from_server(scriptGetAlarmActivation).toInt();
155 actual_min = get_from_server(scriptGetActualMinutes).toInt();
156
157
158
159 delay(1000);
160 }
161
162
163 /*metodi*/
164 void gestioneLedRGB(){ //gestisce semaforo quando allarme è acceso
165
166     if (alarmStateOn) //superato capienza, allarme in corso
167         RGB.lampeggia('r', TIME_LAMP);
168     else if(peoplePresence == cardinalityThreshold) //equagliato capienza
169         RGB.setActualColor('r');
170     else if((float)peoplePresence >= (0.8*cardinalityThreshold)) //sto sopra 80% capienza
171         RGB.setActualColor('y');
172     else //sto tranquillo
173         RGB.setActualColor('g');
174 }
175
176
177 void alarmGeneration(){
178     B.turnOn(); //buzzer comincia a suonare
179     startTime = round(millis() / 1000);
180
181     alarmStateOn = true;
182 }
183
184
185 void alarmStop(){
186     B.turnOff();
187     endTime = round(millis() / 1000);
188
189     send_to_server(timeFormat(), scriptSetAlarmDB); //aggiorna il DB degli allarmi e la pagina con la tabella
190
191     alarmStateOn = false;
192 }
193
194
195 String timeFormat(){
196     int last = endTime - startTime; //numero di secondi di durata dell'allarme
197
198     int h = floor ((float) last / 3600),
199         m = floor ((float)(last - h*3600) /60),
200         s = floor ((float)(last - h*3600 - m*60));
201
202
203     String orario = ""; //format hh:mm:ss
204
205     /*nel caso h, m o s <10 per rispettare il formato aggiungo uno 0*/
206
207     if(h<10) orario += "0";
208     orario = orario + h + ":";
209
210     if(m<10) orario += "0";
211     orario = orario + m + ":";
212
213     if(s<10) orario += "0";
214     orario += s;
215
216     return orario;
217 }
218

```

-1.2) classi

(Led.h)

```
1  #ifndef Led_H
2  #define Led_H
3
4  #include "Arduino.h"
5
6  /*convenzione: se pin = -1, il componente non è stato integrato bene
7   | | | | (ad esempio, se attaccato ad un pin non valido)*/
8
9
10 class Led{
11 public:
12     /*COSTRUTTORI*/
13     Led(int pin);
14
15     /*METODI PUBBLICI*/
16     /*metodi d'accesso*/
17     int getAttachedPin();
18     bool getState();
19
20     /*metodi d'aggiornamento, funzionamento*/
21     void turnOn();
22     void turnOff();
23     void lampeggia(int millTime);
24
25 private:
26     /*ATTRIBUTI*/
27     int attachedPin=-1;
28     bool state=LOW;
29
30 };
31
```

(Buzzer.h)

```
9
10 class Buzzer{
11 public:
12     /*COSTRUTTORI*/
13     Buzzer(int pin);
14
15     /*METODI PUBBLICI*/
16     /*metodi d'accesso*/
17     int getAttachedPin();
18
19     /*metodi d'aggiornamento, funzionamento*/
20     void turnOn();
21     void turnOff();
22
23 private:
24     /*ATTRIBUTI*/
25     int attachedPin=-1;
26
27 };
28
```

(LedRGB.h)


```

20
21 class LedRGB{
22     public:
23         /*COSTRUTTORI*/
24         LedRGB(int redPin, int greenPin, int bluePin);
25
26         /*METODI PUBBLICI*/
27         /*metodi d'accesso*/
28         int getAttachedRedPin();
29         int getAttachedGreenPin();
30         int getAttachedBluePin();
31         char getActualColor();
32
33         /*metodi d'aggiornamento, funzionamento*/
34         void setActualColor(char color);
35         void lampeggia(char color, int millTime);
36         void turnOff();
37
38     private:
39         /*ATTRIBUTI*/
40         int attachedRedPin=-1;
41         int attachedGreenPin=-1;
42         int attachedBluePin=-1;
43         char actualColor='/';
44
45         /*METODI PRIVATI*/
46         void setRed(int lowOrHigh);
47         void setGreen(int lowOrHigh);
48         void setBlue(int lowOrHigh);
49 };
50
51

```

(LedRGB.cpp)

```

31
32 void LedRGB::setActualColor(char color){
33
34     switch(color){
35
36         case 'r':
37             setRed(HIGH);
38             setGreen(LOW);
39             setBlue(LOW);
40             break;
41
42         case 'g':
43             setRed(LOW);
44             setGreen(HIGH);
45             setBlue(LOW);
46             break;
47
48         case 'b':
49             setRed(LOW);
50             setGreen(LOW);
51             setBlue(HIGH);
52             break;
53
54         case 'y':
55             setRed(HIGH);
56             setGreen(HIGH);
57             setBlue(LOW);
58             break;
59
60         default: //colore non valido -> spegne
61             setRed(LOW);
62             setGreen(LOW);
63             setBlue(LOW);
64     }
65
66     actualColor = color;
67 }
68

```

```

69 void LedRGB::turnOff(){setActualColor('/');}
70
71 void LedRGB::lampeggia(char color, int millTime){
72     switch(color){
73
74         case 'r':
75             setGreen(LOW);
76             setBlue(LOW);
77             setRed(HIGH);
78             delay(millTime);
79             setRed(LOW);
80             break;
81
82         case 'g':
83             setRed(LOW);
84             setBlue(LOW);
85             setGreen(HIGH);
86             delay(millTime);
87             setGreen(LOW);
88             break;
89
90         case 'b':
91             setRed(LOW);
92             setGreen(LOW);
93             setBlue(HIGH);
94             delay(millTime);
95             setBlue(LOW);
96             break;
97
98         default: //colore non valido
99             setRed(LOW);
100             setGreen(LOW);
101             setBlue(LOW);
102     }
103
104     actualColor = 'L';
105 }

```

(SensorPIR.h)

```

9
10 class SensorPIR{
11     public:
12         /*COSTRUTTORI*/
13         SensorPIR(int signalPin);
14
15         /*METODI PUBBLICI*/
16         /*metodi d'accesso*/
17         int getAttachedSignalPin();
18         bool detectedMovement(); //rilevato movimenti
19
20         /*metodi d'aggiornamento*/
21
22     private:
23         /*ATTRIBUTI*/
24         int attachedSignalPin=-1;
25
26 };
27

```

(SensorPIR.cpp)

```
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

C: > Users > USER > Documents > Arduino > libraries > SensorPIR > SensorPIR.cpp
1 #include "Arduino.h"
2 #include "SensorPIR.h" //header che stiamo implementando
3
4
5 /*COSTRUTTORI*/
6 SensorPIR::SensorPIR(int signalPin){
7     if(signalPin >= 0 && signalPin <= 35){ //pin valido
8         attachedSignalPin = signalPin;
9         pinMode(signalPin, INPUT);
10    }
11 }
12
13 /*METODI PUBBLICI*/
14 /*metodi d'accesso*/
15 int SensorPIR::getAttachedSignalPin(){return attachedSignalPin;}
16
17 bool SensorPIR::detectedMovement()
18 {
19     return digitalRead(attachedSignalPin) == HIGH;
20     //restituisce 1 solo se signal pin alto, cioè ha rilevato movimento
21 }
22
23 /*metodi d'aggiornamento*/
24
25
```

(DistanceMeter.h)

```
9
10 class DistanceMeter{
11     public:
12
13     /*COSTRUTTORI*/
14     DistanceMeter(int trigPin, int echoPin);
15
16     /*METODI PUBBLICI*/
17     /*metodi d'accesso*/
18     int getAttachedTrigPin();
19     int getAttachedEchoPin();
20     float measureDistance();
21     bool detectedMovement(); //rilevato movimento
22     bool detectedPresence(); //rilevata presenza
23
24     /*metodi d'aggiornamento*/
25
26
27     /*VARIABILI STATIC*/
28     static float thresholdSensitivityDistance; //metri
29     static float thresholdSensitivityTime; //secondi
30
31     /*METODI STATIC*/
32     static void setThresholdSensitivityDistance(float newTreshold);
33     static void setThresholdSensitivityTime(float newTreshold);
34
35
36     private:
37     /*ATTRIBUTI*/
38     int attachedTrigPin=-1;
39     int attachedEchoPin=-1;
40
41     bool presenceFlag = false;
42
43 };
44
```

(DistanceMeter.cpp)

```
C:\Users\USER\Documents\Arduino\libraries\DistanceMeter> DistanceMeter.cpp
1  #include "Arduino.h"
2  #include "DistanceMeter.h"          //header che stiamo implementando
3
4
5  /*etichette-costanti di classe*/
6  #define MAX_DISTANCE 5.0           //5 m massima distanza misurabile
7  #define SOUND_SPEED 340            //340 m/s velocit  del suono
8
9  float const TIME_OUT = 2* MAX_DISTANCE / SOUND_SPEED * 1000000;
10 //tempo massimo di attesa in microsecondi per il ritorno dell'echo
11
12 float thresholdSensitivityDistance = 1.0;
13
14
15
16 /*COSTRUTTORI*/
17 DistanceMeter::DistanceMeter(int trigPin, int echoPin){
18     if(trigPin >= 1 && trigPin <= 35 &&
19         echoPin >= 1 && echoPin <= 35 &&
20         trigPin != echoPin)          //pin validi
21     {
22         attachedTrigPin = trigPin;
23         pinMode(trigPin, OUTPUT);
24         digitalWrite(trigPin, LOW);
25
26         attachedEchoPin = echoPin;
27         pinMode(echoPin, INPUT);
28     }
29 }
30
31 float DistanceMeter::measureDistance()
32 {
33     digitalWrite(attachedTrigPin, HIGH);
34     delayMicroseconds(10);
35     digitalWrite(attachedTrigPin, LOW);
36
37     float pingTime = (float) pulseIn(attachedEchoPin, HIGH, TIME_OUT) / 1000000; //tempo di rimbalzo in sec
38     return pingTime * SOUND_SPEED / 2;
39 }
40
41 bool DistanceMeter::detectedPresence()
42 {
43     float m = measureDistance();
44
45     //debug
46     //Serial.print("Misura: ");
47     //Serial.println(m);
48
49     return (m <= thresholdSensitivityDistance);
50 }
51
52 bool DistanceMeter::detectedMovement()
53 {
54     bool risultato = false;
55
56     if(!presenceFlag && detectedPresence())
57         presenceFlag=true;
58
59     if(presenceFlag && !detectedPresence()){
60         presenceFlag = false;
61         risultato = true;
62     }
63
64     return risultato;
65 }
66
67 /*metodi d'aggiornamento*/
68
69 /*VARIABILI STATIC*/
70 float DistanceMeter:: thresholdSensitivityDistance = 1.0; //metri
71 float DistanceMeter:: thresholdSensitivityTime = 0.5;      //secondi
72
73
74 /*METODI STATIC*/
75 void DistanceMeter::setThresholdSensitivityDistance(float newTreshold){thresholdSensitivityDistance=newTreshold;}
76 void DistanceMeter::setThresholdSensitivityTime(float newTreshold){thresholdSensitivityTime=newTreshold;}
77
78
79
80
81
82
83
84
85
```

2) Parte web

(config.php) //file che serve a tutte le sottocategorie

```
C: > xampp > htdocs > smart_queue > config.php
1
2  <?php
3
4  /*realizza la connessione al database */
5
6  $host= "127.0.0.1";
7  $user= "root";
8  $password ="";
9  $db= "smart_queue";
10
11  $connessione = new mysqli($host,$user,$password,$db);
12
13  if($connessione == false){
14      die("Errore durante la connessione" . $connessione->connect_error);
15  }
16
17  ?>
```

-2.1) Login

(login.html)

```
1  <html lang="en">
2  <head>
3      <meta charset="UTF-8">
4      <meta http-equiv="X-UA-Compatible" content="IE=edge">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Accedi</title>
7      <link rel="stylesheet" href="style_login2.css">
8  </head>
9
10
11  <body>
12      <div class="container">
13          <h1>Login Admin</h1>
14          <form name="modulo" action="accesso_admin.php" method="POST">
15
16              <div class="form-control">
17                  <label for="email">Email</label>
18                  <input type="email" id="email" name="email" placeholder="Inserisci Email" required>
19              </div>
20
21              <div class="form-control">
22                  <label for="password">Password</label>
23                  <input type="password" id="password" name="password" placeholder="inserisci Password" required>
24              </div>
25
26              <button class="btn">ACCEDI</button>
27          </div>
28      </form>
29  </div>
30
31  </body>
32  </html>
```

(accesso_admin.php)

```
dashboard.php  accesso_admin.php X
C: > xampp > htdocs > smart_queue > accesso_admin.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Errore!</title>
8  </head>
9  <body>
10
11
12
13  <?php
14
15
16      require("config.php");
17
18      /*prendo email e password inseriti dall'utente*/
19      $email = $_POST['email'];
20      $password = $_POST['password'];
21
22      /*confronto con dati del database*/
23      if($_SERVER["REQUEST_METHOD"] == "POST")
24      {
25          $sql = "SELECT * FROM tab_admin WHERE email = '$email'";
26
27          if($result = $connessione ->query($sql)){
28              if($result->num_rows == 1)
29
30              if($result = $connessione ->query($sql){
31                  if($result->num_rows == 1)
32                  {
33                      $row = $result->fetch_array(MYSQLI_ASSOC); //creo un array associativo prelevando dal database
34
35                      if($password == $row['password']) //password corretta
36                      {
37                          //login a buon fine
38                          session_start();
39                          $_SESSION['loggato'] = true;
40                          $_SESSION['email'] = $row['email'];
41
42                          header("location: dashboard.php"); //mi reindirizza alla pagina dashboard.php
43                      }
44                      else
45                      {
46                          echo "La password non e' corretta";
47                      }
48                  }
49                  else
50                  {
51                      //non ha trovato quell'email (result == null)
52                      echo "Non ci sono account con quella email";
53                  }
54              }
55              else
56              {
57                  echo "Errore in fase di login!";
58              }
59          }
60          $connessione ->close();
61      }
62
63  >>
64
65      <h1 style="color:white; margin-top:40px; font-size:40px;">Username o Password errata!</h1>
66      <br><br>
67      <a href="../../login_admin.html">Ritorna al Login</a>
68
69  </div>
70
71 </body>
72 </html>
```

-2.2) Dashboard

--2.2.1) File principale

(dashboard.php)

```
<div class="card">
  <h2>OCCUPAZIONE ATTUALE</h2>

  <div class="row">
    <div class="leftcolumn2">
      <h5>su un totale di </h5>
    </div>

    <form action="setCardinalityThreshold.php" method="get">
      <div class="centralcolumn2">
        <input type="text" id="userSetThreshold" name="userSetThreshold" >

      </div>

      <div class="rightcolumn2">
        <span>
          <input type="submit" id="thresholdSetButton" value="">
        </span>

        <div class="dropdown-content">
          <p>Cambia la capienza massima</p>
        </div>
      </div>
    </form>
  </div>
</div>
```

risultato:

OCCUPAZIONE ATTUALE

su un totale di 

0 su 40

```

176 <h2>Lista allarmi</h2>
177 <?php
178 $sql="SELECT* FROM tab_allarme";
179 if($result = $connessione->query($sql)){
180     if($result -> num_rows > 0){
181         echo'
182         <table style="font-family: arial, sans-serif;
183         border-collapse: collapse;
184         width: 100%;
185         border: 2px solid #dddddd;
186         text-align: left;
187         padding: 1px;
188         background-color: #5c5858;
189         ">
190         <thead>
191         <tr style = "border: 2px solid #dddddd">
192         <th style = "border: 2px solid #dddddd">Allarme</th>
193         <th style = "border: 2px solid #dddddd">data ora fine allarme</th>
194         <th style = "border: 2px solid #dddddd">durata</th>
195         </tr></thead>';
196
197         while($row = $result->fetch_array()){
198             echo'
199             <tr>
200             <td style = "border: 2px solid #dddddd">'. $row['id_allarme'] . '</td>
201             <td style = "border: 2px solid #dddddd">'. $row['data_ora_fine_allarme'] . '</td>
202             <td style = "border: 2px solid #dddddd">'. $row['durata'] . '</td>
203             </tr>';
204         }
205
206         echo'
207         </table>';
208     }
209     else
210     {
211         echo "Non ci sono allarmi";
212     }
213 }

```

Risultato:

Lista allarmi

Allarme	data ora fine allarme	durata
1	2023-06-25 15:56:56	00:00:15
8	2023-06-27 19:14:02	00:00:10
9	2023-07-01 17:58:10	00:00:01

```

C: > xampp > htdocs > smart_queue > dashboard.php
217 </div>
218 </div>
219
220 <div class="rightcolumn3">
221     <div class="card">
222         <h3 id="headerDisattivaAllarme">Disattiva allarme</h3>
223
224         <div class="row">
225
226             <div class="leftcolumn4">
227
228                 <div class="leftcolumn5">
229                     stato:
230                 </div>
231
232                 <div class="rightcolumn5">
233                     <?php require("getFormatAllarmActivation.php"); ?>
234                 </div>
235
236                 <form action="setAllarmActivation.php" method="get">
237                     <input type="submit" class="disableAllarmButton" name="setActivation" value="disattiva/attiva">
238                 </form>
239             </div>
240
241             <div class="rightcolumn4">
242                 
243             </div>
244
245         </div>
246     </div>

```

Risultato:

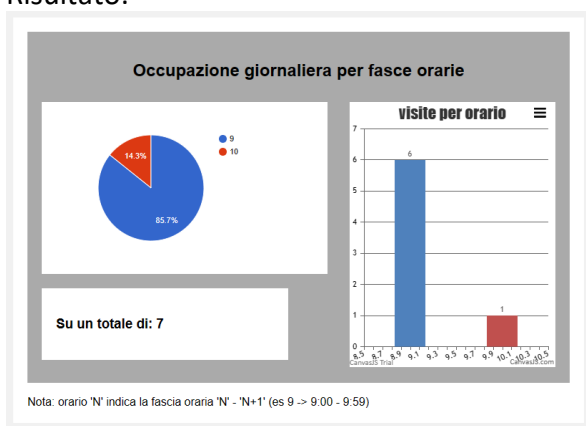


```

C: > xampp > htdocs > smart_queue > dashboard.php
253     </div>
254
255     </div>
256
257     <div class="card">
258         <div class="fakeimg" style="height:490px;">
259             <h2 id="graphicsHeader">Occupazione giornaliera per fasce orarie</h2>
260
261             <div class="leftcolumn6">
262                 <div class="row">
263                     <div id="chart_container"></div>
264                 </div>
265
266                 <div class="card" style="width: 80%;">
267                     <h3>Su un totale di: <?php require("getTotal.php") ?></h3>
268                 </div>
269             </div>
270
271             <div class="rightcolumn6">
272                 <script src="https://cdn.canvasjs.com/canvasjs.min.js"></script>
273                 <div id="chartContainer" style="height: 370px; width: 100%;"></div>
274             </div>
275         </div>
276
277         <br>
278         Nota: orario 'N' indica la fascia oraria 'N' - 'N+1' (es 9 -> 9:00 - 9:59)
279     </div>
280
281 </div>
282 </div>
283

```

Risultato:



--2.2.2) Script eseguiti dalla dashboard non relativi alla comunicazione

(getOccupation.php)

```
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

getOccupation.php X
C: > xampp > htdocs > smart_queue > getOccupation.php
1  <?php
2
3      /*leggo da file*/
4      $f = fopen("occupazione.txt", "r");
5      $content = fread($f, filesize("occupazione.txt"));
6      fclose($f);
7
8
9      echo $content;
10
11  ?>
```

(setCardinalityTreshold.php)

```
C: > xampp > htdocs > smart_queue > setCardinalityThreshold.php
1  <?php
2
3
4      if(!empty($_GET)){ //se cliccato sulla matita
5
6          if($_GET["userSetThreshold"] > 0){ //fa un controllo se maggiore di 0 (soglia valida)
7              $f = fopen("soglia.txt", "w");
8              fwrite($f, $_GET["userSetThreshold"]);
9              fclose($f);
10         }
11     }
12
13
14     header("Location: dashboard.php");
15
16
17  ?>
```

(setAlarmActivation.php)

```
getOccupation.php  getTotal.php  setAlarmActivation.php X
C: > xampp > htdocs > smart_queue > setAlarmActivation.php
1  <?php
2
3      header("Location: dashboard.php");
4
5      if(!empty($_GET)){ //get che viene eseguita se ho cliccato sul bottone "attiva/disattiva"
6
7
8          /* leggo lo stato dell'allarme sul file */
9          $f = fopen("alarmActivation.txt", "r");
10         $content = fread($f, filesize("alarmActivation.txt"));
11         fclose($f);
12
13         /* in base allo stato attuale, devo cambiarlo nel suo opposto */
14         $toWrite="";
15
16         if(strcmp($content, "1") == 0) // acceso -> devo spegnere
17             $toWrite = "0";
18         else // spento -> devo accendere
19             $toWrite = "1";
20
21         $f = fopen("alarmActivation.txt", "w");
22         fwrite($f, $toWrite);
23         fclose($f);
24     }
25
26  ?>
```

(getFormatAllarmActivation.php)

```
C: > xampp > htdocs > smart_queue > getFormatAllarmActivation.php
1  <?php
2
3      /* leggo lo stato dell'allarme sul file */
4      $f = fopen("allarmActivation.txt", "r");
5      $content = fread($f, filesize("allarmActivation.txt"));
6      fclose($f);
7
8      /* lo formato per l'output */
9      if($content == "0")
10         echo "Spento";
11     else
12         echo "Acceso";
13
14  ?>
```

(getTotal.php)

```
getOccupation.php  getTotal.php X
C: > xampp > htdocs > smart_queue > getTotal.php
1  <?php
2
3      /*leggo da DB tutte le entrate orarie giornaliere e le sommo tutte*/
4      require("config.php"); //connessione DB
5
6      $sql = "SELECT * FROM tab_entrare WHERE tab_entrare.Data = CURRENT_DATE()";
7      $result = mysqli_query($connessione, $sql);
8
9
10     $total=0;
11
12     while($row = mysqli_fetch_assoc($result))
13     {
14         $total += $row['numero_visitatori'];
15     }
16
17     echo $total;
18
19  ?>
```

--2.2.3) Script per comunicazione con ESP (ESP -> Server)

(setOccupation.php)

```
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage
getActualMinutes.php getAlarmActivation.php dashboard.php
C: > xampp > htdocs > smart_queue > setOccupation.php
1 <?php
2
3 /*scrivo su file dato da ESP*/
4 if(!empty($_GET)){
5     $f = fopen("occupazione.txt", "w");
6     fwrite($f, $_GET["value"]);
7     fclose($f);
8 }
9
10
11
12 ?>
```

(setAlarmDB.php)

```
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
setOccupation.php setHourVisitorsDB.php setAlarmDB.php
C: > xampp > htdocs > smart_queue > setAlarmDB.php
1 <?php
2
3 /*scrivo su DB valore dell'ESP (durata) */
4
5 require("config.php"); //connessione al DB
6
7
8
9
10 if(!empty($_GET)){
11     $x = $_GET["value"];
12
13     $sql = "INSERT INTO tab_allarme (data_ora_fine_allarme, durata) VALUES (current_timestamp(), '$x')";
14     $connessione->query($sql);
15 }
16
17 ?>
```

(setHourVisitorsDB.php)

```
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
setOccupation.php setHourVisitorsDB.php
C: > xampp > htdocs > smart_queue > setHourVisitorsDB.php
1 <?php
2
3 /*scrivo su DB valore dell'ESP (occupazione oraria) */
4
5 require("config.php"); //connessione al DB
6
7
8 if(!empty($_GET)){
9     $hours = date('h', time());
10
11     $specifier = date('a', time()); //am oppure pm
12     if(strpos($specifier, "pm") == 0)
13         $hours+=12;
14
15     echo $hours;
16
17     if($hours >= 10 && $hours <= 18) //se l'ora appena passata quella dell'apertura del museo, cioè da 9:00 a 17:59
18     {
19         $hours = $hours - 1; //trovo l'orario da registrare (fascia oraria = ora - 1)
20         $x = $_GET["value"]; //numero di visite in quest'ora trascorsa
21
22         $sql = "INSERT INTO tab_entrare (fascia_oraria, numero_visitatori, Data) VALUES ('$hours', '$x', CURRENT_DATE())";
23         $connessione->query($sql);
24     }
25
26
27
28
29 ?>
```

--2.2.4) Script per comunicazione con ESP (Server -> ESP)

(getCardinalityThreshold.php)

```
C: > xampp > htdocs > smart_queue > getCardinalityThreshold.php
1  <?php
2
3      /* leggo la soglia da file */
4      $f = fopen("soglia.txt", "r");
5      $content = fread($f, filesize("soglia.txt"));
6      fclose($f);
7
8
9      echo $content;
10
11  ?>
```

(getAllarmActivation.php)

```
getActualMinutes.php  getAllarmActivation.php X
C: > xampp > htdocs > smart_queue > getAllarmActivation.php
1  <?php
2
3      /* leggo lo stato dell'allarme sul file */
4      $f = fopen("allarmActivation.txt", "r");
5      $content = fread($f, filesize("allarmActivation.txt"));
6      fclose($f);
7
8      echo $content;
9
10
11  ?>
```

(getActualMinute.php)

```
getActualMinutes.php X
C: > xampp > htdocs > smart_queue > getActualMinutes.php
1  <?php
2
3      $minutes = date('i', time()); //prende il minuto attuale
4      echo $minutes;
5
6
7  ?>
```

++) File di testo usati per la comunicazione:

- occupazione.txt

- soglia
- alarmActivation.txt

++) File Css usati per lo stile:

- style_login2.css
- style_dashboard.css