

# **Relazione progetto Smart Room - IoT**

*Marco Antolini, Luca Pasini, Lorenzo Tosi, Andrea Zavatta*

*marco.antolini6@studio.unibo.it, luca.pasini9@studio.unibo.it,  
lorenzo.tosi10@studio.unibo.it, andrea.zavatta3@studio.unibo.it*

# Indice

<b>1</b>	<b>Requisiti</b>	<b>3</b>
<b>2</b>	<b>Room Controller</b>	<b>4</b>
<b>3</b>	<b>Room Sensor Board</b>	<b>5</b>
<b>4</b>	<b>Room Service</b>	<b>6</b>
<b>5</b>	<b>Room Dashboard</b>	<b>7</b>
5.1	Interfaccia . . . . .	7
5.1.1	Window . . . . .	7
5.1.2	History . . . . .	7
5.2	Software . . . . .	7
<b>6</b>	<b>Room App</b>	<b>9</b>

# 1 Requisiti

Desideriamo realizzare un sistema IoT che implementi una versione semplificata di una stanza intelligente, finalizzata a monitorare e controllare lo stato di una stanza in un campus.

1. Room Sensor Board (esp): Sistema incorporato per monitorare lo stato della stanza tramite sensori. Interagisce con il Room Service tramite MQTT o HTTP.
2. Room Service (backend - pc): Funziona come unità principale per la gestione della stanza. Comunica tramite seriale con il Controller, tramite MQTT con la Room Sensor Board e tramite HTTP con la Dashboard.
3. Room Controller (Arduino): Sistema incorporato che controlla la luce e le tapparelle. Comunica tramite seriale con il Room Service e tramite Bluetooth con la Room App. Deve implementare la logica di controllo tramite macchine a stati finiti (sincrone o asincrone).
4. Room App (Android - smartphone): App mobile per il controllo manuale delle luci e delle tapparelle. Comunica con il Controller tramite Bluetooth (o tramite seriale in caso di emulatore Android).
5. Room Dashboard (Frontend/web app su PC o tramite sockets): Interfaccia per visualizzare e tracciare lo stato della stanza. Interagisce con il Room Service.

Il sistema Smart Room controlla l'illuminazione e le tapparelle secondo la seguente politica:

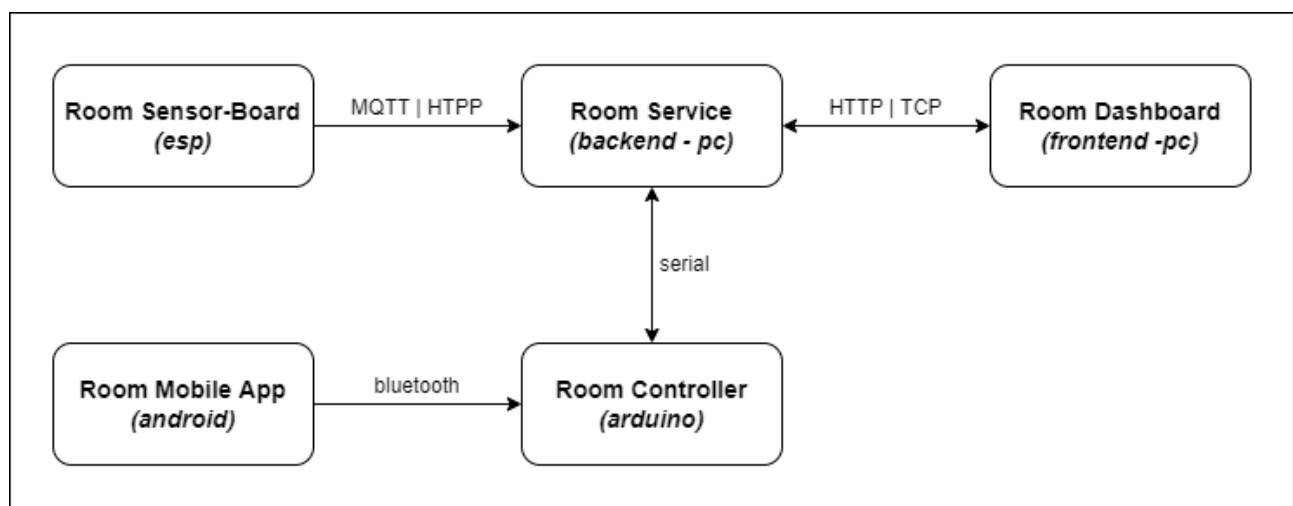
- Se nessuno è nella stanza, la luce è spenta.
- Se qualcuno entra in una stanza buia, la luce si accende.
- Le tapparelle si sollevano completamente automaticamente alla prima entrata a partire dalle 8:00.
- Le tapparelle si abbassano completamente alle 19:00 se sono sollevate e nessuno è in stanza, o non appena chi è ancora in stanza alle 19:00 lascia.

Attraverso l'app mobile, l'utente può:

- Accendere/spegnere la luce.
- Sollevare/abbassare le tapparelle in modo parziale (0-100- Attraverso la dashboard, il responsabile della stanza può:
- Monitorare lo stato della stanza.
- Controllare completamente la luce e le tapparelle.
- Si presume che la stanza sia accessibile dalle 8:00 alle 19:00.

Ulteriori dettagli:

- La Room Sensor Board deve accendere il LED quando qualcuno è in stanza e spegnerlo quando nessuno è presente.
- Il Controller controlla/simula le tapparelle con un servomotore. 0° rappresenta le tapparelle completamente sollevate, 180° completamente abbassate.



*Schema funzionamento applicazione*

## 2 Room Controller

### 3 Room Sensor Board

# 4 Room Service

## 5 Room Dashboard

La Room Dashboard è un'applicazione web hostata su localhost grazie al servizio di web server Xampp che comunica tramite richieste HTTP con il Room Service per interagire con i componenti hardware.

### 5.1 Interfaccia

La dashboard comprende due pagine:

- Window: mostra lo stato attuale delle tapparelle e della lampadina all'interno della stanza e permette anche di modificarli;
- History: mostra dati relativi all'attività delle tapparelle e della lampadina.

#### 5.1.1 Window

Nella Window è visibile un orologio che mostra l'orario attuale e lo sfondo del sito cambia dinamicamente in base al fatto che sia giorno o notte. È presente l'immagine di una lampadina che rispecchia lo stato della lampadina nella stanza e che è possibile cliccare per accendere e spegnere la luce. Accade lo stesso per le tapparelle, per cui è presente uno slider tramite il quale è possibile controllare l'apertura totale o parziale delle tapparelle della stanza.

#### 5.1.2 History

Nella History sono presenti due logs che mostrano i momenti (con gli orari riportati in formato HH:mm:ss) in cui la lampadina è stata spenta o accesa (quindi rispettivamente con le diciture On o Off) e in cui la tapparella è stata aperta o chiusa (riportando la percentuale di apertura rilevata alla fine del movimento).

Inoltre è presente un grafico a torta che mostra l'utilizzo della lampadina con due fette che raffigurano il tempo in cui è stata accesa e il tempo in cui è stata spenta e le relative percentuali.

Tutti i dati riportati in questa pagina sono relativi alle ultime 24 ore.

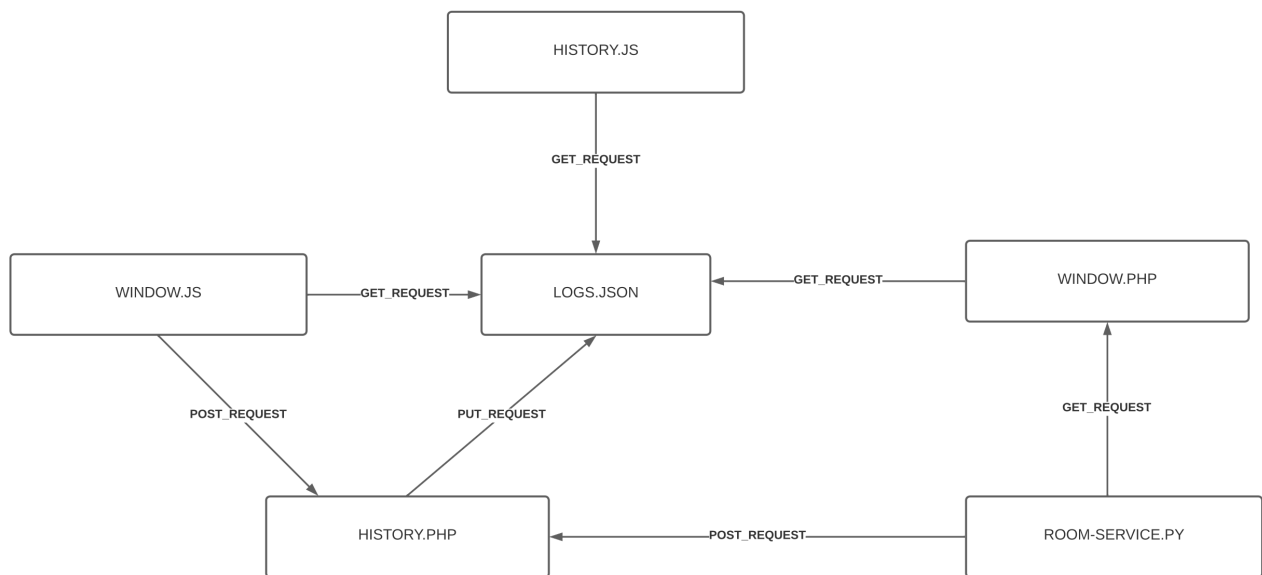
### 5.2 Software

Come introdotto in precedenza la dashboard utilizza le richieste HTTP per comunicare con il Room Service in python che permette di scambiare informazioni e comandi in tempo reale con l'hardware.

Tutti i logs delle ultime 24 ore vengono salvati (e aggiornati dinamicamente) sul file `logs.json`. Il file `room-dashboard-history.js` deve semplicemente accedere ai logs quindi fa una richiesta di tipo GET al file json ogni volta che la pagina viene aggiornata.

Il file `room-dashboard-window.js` deve invece sia fare una richiesta di tipo GET al file json per ottenere i dati aggiornati, sia gestire i comandi provenienti dalla dashboard e farli arrivare all'hardware. Ciò avviene tramite una richiesta POST al file `room-dashboard-history.php` (che tra le altre cose controlla i dati e li filtra per fare in modo che vengano salvati solo quelli delle ultime 24 ore) il quale scrive i dati sul file dei logs.

Dall'altro lato dell'applicazione c'è il Room Service sviluppato in Python che legge i dati dal file dei logs per passarli all'hardware e scrive i cambiamenti hardware sempre sul file json. La lettura avviene tramite una richiesta GET al file `room-dashboard-window.php` che a sua volta fa una richiesta di tipo GET al file json. La scrittura invece avviene ugualmente alla controparte della dashboard quindi tramite una richiesta POST al file `room-dashboard-history.php` che successivamente scrive i dati sul file dei logs.



*Schema funzionamento richieste*



## 6 Room App