

NOME e COGNOME: EMANUELE MATIOLI

EMAIL UNIMORE: 186979@studenti.unimore.it

## Prova di Laboratorio - Sistemi e Applicazioni di Rete

*Appello 11 settembre 2018 - Modalità di esame 2017/2018*

Si realizzi un'applicazione web su Google App Engine per la gestione delle prenotazioni in un cinema multisala. L'applicazione permette a clienti di prenotare un posto per la visione di uno spettacolo ad un determinato orario. L'applicazione deve mantenere lo stato delle prenotazioni e implementare le interfacce necessarie per l'accesso alle funzionalità da parte di utenti e applicazioni esterne.

Si assuma che il cinema abbia varie sale identificate da un nome univoco del tipo *salaN*, dove *N* identifica il numero della sala (ad esempio, *sala2*). I posti sono identificati univocamente all'interno di ogni sala tramite una lettera (A, B, ...) e un numero (01, 02, ...), che identificano rispettivamente la fila e la colonna. Ad esempio, un posto in seconda fila e sesta colonna è identificato dalla sigla B06.

Si assuma che il cinema abbia 6 sale delle seguenti dimensioni:

- sala1, sala2, sala3: 15 file, 10 colonne
- sala4, sala5: 8 file, 10 colonne
- sala6: 22 file, 14 colonne

Per semplicità, si assuma che gli orari di proiezione dei film nelle sale siano uguali e fissati ai seguenti valori:

- 17:30
- 20:00
- 22:30

Si implementino le seguenti funzionalità:

1. interfaccia di visualizzazione e manipolazione delle prenotazioni tramite API REST;
2. interfaccia di gestione dei film in proiezione e della visualizzazione dello stato di occupazione dei posti tramite pagine HTML;
3. client per la richiesta del servizio bar al posto realizzato tramite interfaccia MQTT.

### API REST

L'applicazione deve esporre le seguenti funzionalità tramite opportune Web API RESTful:

- allo URI `/api/v0.1/reservations/<room>/<time>` è possibile visualizzare e manipolare lo stato delle prenotazioni di una sala;
- allo URI `/api/v0.1/users/<email>` è possibile ottenere la lista delle prenotazioni effettuate dall'utente specificato;

L'interfaccia di utilizzo delle API deve soddisfare **rigorosamente** il file di specifica *OpenAPI* disponibile presso il sito del corso<sup>1</sup>.

### Interfaccia HTML

L'interfaccia HTML deve prevedere due pagine:

1. programmazione dei film in cartellone e relativa sala di proiezione;
2. visualizzazione dello stato di occupazione dei posti in sala.

La prima pagina deve permettere di definire i film per ogni proiezione utilizzando una form Web che include i seguenti campi di input:

<sup>1</sup>[https://weblab.ing.unimore.it/sar/1718/lab/openapi\\_multisala.yaml](https://weblab.ing.unimore.it/sar/1718/lab/openapi_multisala.yaml)



- il nome univoco della sala (e.g., *sala1*);
- l'orario di proiezione (e.g., *17:30*);
- l'identificativo IMDb del film (*vedere note sotto per dettagli*).

La funzionalità deve verificare la disponibilità della sala all'orario e la validità del codice IMDb indicato prima di accettare la programmazione.

La seconda pagina deve mostrare la lista dei film disponibili, mostrando per ognuno le seguenti informazioni:

- sala e orario di proiezione del film (e.g., *sala1, 17:30*);
- titolo del film;
- anno del film;
- locandina del film.

Le informazioni riguardanti il film devono essere ottenute tramite il servizio di terze parti OMDb Api<sup>2</sup> utilizzando l'identificativo ottenuto dagli utenti nella prima pagina. Per semplicità, è possibile progettare il servizio richiedendo le informazioni del film ad ogni richiesta di informazioni (non si richiede quindi di memorizzare le informazioni dei film in alcun database del servizio).

## Interfaccia MQTT

Si richiede di implementare un sistema Pub/Sub basato su MQTT per la realizzazione di un sistema automatizzato di *bar-al-posto*, in cui gli spettatori possono effettuare degli *ordini* a dei *robot* automatizzati presenti in sala.

Si assuma la presenza di **5** robot per ogni sala, ognuno in ascolto su un particolare topic relativo alla sala di servizio, e di tre tipi di ordini: bibita, popcorn, menu.

Nell'ambito dell'esame il sistema deve essere emulato localmente. Si richiede di modellare degli opportuni topic per rappresentare al meglio la realtà in esame.

Il sistema implementato è composto da due moduli python:

- uno script eseguibile `ordinazione.py` con interfaccia da linea di comando che permette effettuare un ordine presso un posto in una sala. In dettaglio:

– l'interfaccia dello script deve essere la seguente:

`ordinazione.py <item={BIBITA|POPCORN|MENU}>`

Ad esempio: *Sala Posto Ordine*

`ordinazione.py sala1 A05 POPCORN ordinazione.py sala6 C08 BIBITA ordinazione.py sala4 B06 MENU`

- uno script eseguibile `robot.py` che monitora lo stato dei robot di una sala. Questo servizio è implementato tramite un subscriber MQTT che si occupa di mostrare a video le richieste ricevute. Si consideri che un robot viene considerato occupato per 30 secondi dopo il ricevimento di un ordine.

– l'interfaccia dello script deve essere la seguente:

`robot.py`

*Note aggiuntive:*

- L'IMDb ID è un particolare codice identificativo utilizzato dal popolare sito IMDb l'identificazione univoca dei film. Per ottenere un codice IMDb, si cerchi all'indirizzo <https://www.imdb.com> un film in particolare e lo si copi dall'URL della scheda del film. L'IMDb ID è un codice alfanumerico generalmente preceduto dalla coppia di lettere **tt**. Ad esempio, l'IMDb ID per il film *Interstellar* (<https://www.imdb.com/title/tt0816692/>) è **tt0816692**.
- è necessario richiedere una API key all'indirizzo <http://www.omdbapi.com/apikey.aspx> per utilizzare il servizio OMDbAPI. Assicurarsi di inserire un indirizzo email corretto in fase di registrazione e di selezionare un tipo di account "FREE".

<sup>2</sup><http://www.omdbapi.com/>