Reti: Modulo Laboratorio III
HOTELIER: an HOTEL advIsor sERvice
Progetto di Fine Corso A.A. 2023/24

1. Descrizione del servizio

Il progetto consiste nell'implementazione di un servizio che si ispira a *TripAdvisor*. Viene richiesto di implementare un sottoinsieme di funzionalità molto semplificate del popolare sito di recensioni di alberghi, ristoranti ed altri contenuti relativi ai viaggi. In questo paragrafo ricordiamo brevemente le principali funzionalità di *TripAdvisor*, mentre il paragrafo successivo descriverà le funzionalità richieste per *HOTELIER*.

Tripadvisor richiede la registrazione di un account ed il login nel caso si desideri rilasciare recensioni per i contenuti messi a disposizione dal servizio, mentre non è necessario un account per la consultazione dei contenuti. La piattaforma mette a disposizione diverse funzioni di ricerca, l'utente può cercare una specifica struttura (hotel, ristorante, casa vacanze,...) di cui vuole conoscere le caratteristiche, oppure trovare le informazioni relative a una specifica classe di strutture di una località. I risultati della ricerca mostrano diverse informazioni sulla/su ognuna delle strutture trovate.

Gli utenti possono pubblicare delle recensioni sulle strutture (ristoranti, hotel, ...) che hanno visitato, assegnando un punteggio da 0 a 5 a *specifiche categorie* (che dipendono dal tipo della struttura) e un *punteggio sintetico* sulla struttura, sempre da 0 a 5. E' anche possibile pubblicare una recensione a testo libero, in cui, in base alla tipologia di struttura, l'utente descrive cosa ha gradito e cosa no.

Per ogni struttura viene calcolato un *rank locale* che la posiziona rispetto ad altre strutture della stessa zona, il rank è calcolato in base ai punteggi sintetici attribuiti dagli utenti a quella struttura nelle loro recensioni. La classifica viene ricalcolata *quotidianamente*, in base alle nuove recensioni ricevute nella giornata. TripAdvisor non ha pubblicato l'algoritmo che utilizza, ma ha reso noto solo che si basa sulla *qualità* delle recensioni, sulla loro *attualità*, e sulla loro *quantità*. L'attualità delle recensioni riguarda il fatto che le recensioni più recenti hanno maggior valore rispetto a quelle pubblicate da molto tempo, e questo vale sia per le recensioni positive che per quelle negative.

La Fig.1 esemplifica la ricerca di uno specifico hotel. Come mostrato nella figura, sono presenti diverse informazioni riguardanti la struttura, tra cui di particolare rilievo è il punteggio sintetico calcolato sulla base dei punteggi sintetici dei singoli utenti, e i punteggi specifici relativi a posizione, pulizia, servizi e rapporto qualità/prezzo assegnati come media delle recensioni collezionate dagli utenti. E' presente inoltre il numero totale di recensioni e il rank della struttura rispetto alle altre collocate nella stessa zona geografica.

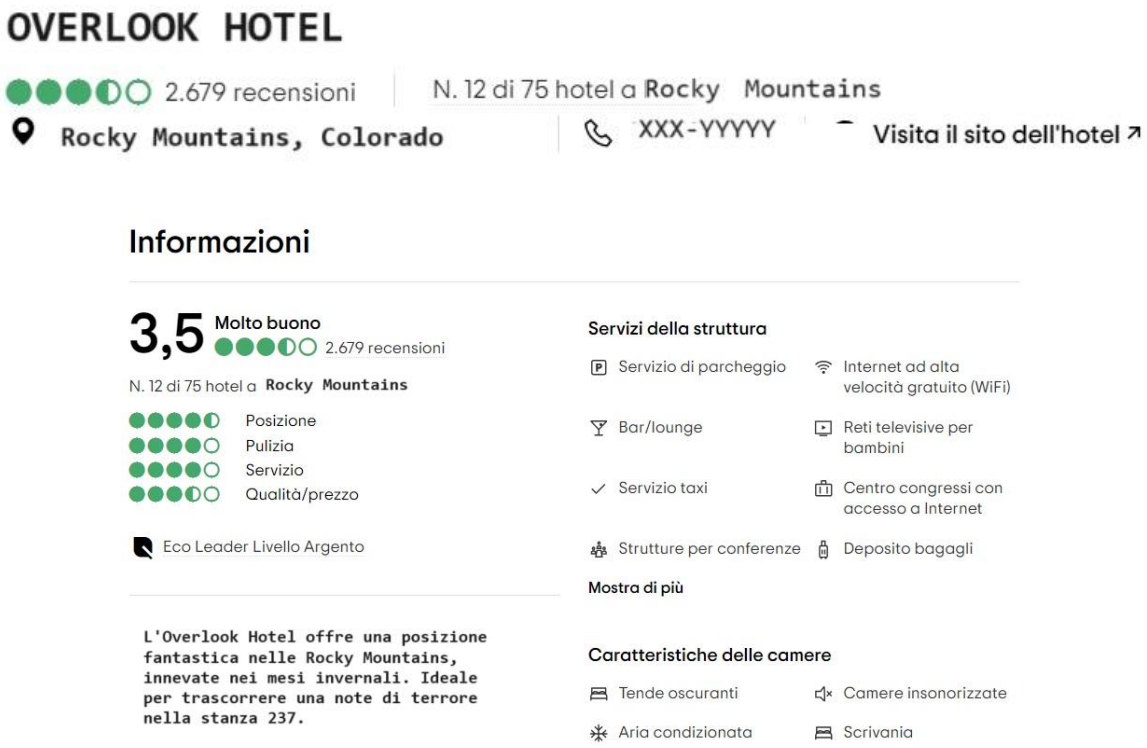

Fig. 1

Tripadvisor offre ha anche elaborato un sistema di "distintivi" che vengono attribuiti ad un utente ogni volta che supera una certa quota di recensioni. Ci sono diversi livelli di “esperienza ”, ciascuno associato a un diverso distintivo. I distintivi non danno origine a ricompense, tuttavia, il “distintivo” di un utente viene sempre indicato vicino alle recensioni e serve solo ad indicare quanto il recensore sia "esperto".

TripAdvisor offre molte altre funzionalità, che tuttavia non riguardano questo progetto. Ad esempio, i gestori di una struttura (hotel, ristorante) devono registrarsi, creando un account, per poter pubblicare i dati della propria struttura sulla piattaforma. Ogni gestore può iscriversi ad un servizio di notifica che lo avvisa tutte le volte in cui è stata sottomessa una nuova recensione per la sua struttura.

2. HOTELIER: Funzionalità

Si richiede di implementare una versione semplificata di un sottoinsieme della funzionalità di *TripAdvisor*, che conservi però la logica di base delle funzionalità mostrate nel paragrafo precedente. In particolare si tengano in considerazione le seguenti specifiche:

- le uniche strutture gestite da HOTELIER sono gli hotel
- gli hotel recensiti sono collocati solo in un insieme predefinito di città, ad esempio si possono considerare le città capoluogo delle 20 regioni italiane. HOTELIER non permette la registrazione di hotel posizionati fuori da questo insieme di città.

- per ogni hotel vengono registrate solo informazioni di tipo testuale (non si prevede il caricamento di foto)
- le recensioni effettuate dagli utenti consistono solo nel punteggio (da 0 a 5) sintetico e nei punteggi da 0 a 5 dati alle singole categorie, queste ultime coincidono con quelle che TripAdvisor mostra per gli hotel, ovvero quelle mostrate in Fig.1 (Posizione, Pulizia, Servizio, Prezzo). Non sono supportate recensioni testuali.
- La valutazione del ranking locale si basa sulla attribuzione di un punteggio ad ogni hotel. L'algoritmo per l'attribuzione di tale punteggio deve essere specificato dallo studente e deve tenere in considerazione i tre parametri, qualità, quantità e attualità delle recensioni. La qualità delle recensioni riguarda i punteggi sintetici attribuiti da ogni utente a quella struttura. La quantità è il numero totale di recensioni ricevute dall'hotel. Il rank locale di un hotel viene calcolato in base alla posizione di quell'hotel rispetto a tutti gli altri hotel della stessa città.
- ad ogni utente vengono attribuiti livelli diversi di esperienza in base al numero di recensioni effettuate. Si prevedono 5 livelli: Recensore, Recensore esperto, Contributore, Contributore esperto, Contributore Super. Ogni livello corrisponde ad un diverso "distintivo".

Distingueremo in seguito l'insieme di funzionalità di base di HOTELIER da quelle aggiuntive. L'insieme di funzionalità di base dovrà essere implementate sia dagli studenti del vecchio ordinamento che da quelli del nuovo ordinamento, quelle aggiuntive solo dagli studenti del vecchio ordinamento. Inoltre l'implementazione di alcune funzionalità di base sarà diversa per gli studenti del nuovo rispetto a quelli del vecchio ordinamento, come specificato nella sezione 2.1 e 2.2.

2.1 Funzionalità di base (per gli studenti di entrambi gli ordinamenti)

Il gioco deve essere implementato mediante un programma client e un programma server. Le due componenti che interagiscono usando diversi protocolli e paradigmi di comunicazione di rete, sono le seguenti:

- **HOTELIERCustomerClient**: Gestisce l'interazione con l'utente, tramite una CLI (Command Line Interface), comunica con il Server per eseguire le azioni richieste dall'utente. Le principali operazioni sono le seguenti:
  - *register(username, password)*: registrazione a HOTELIER. L'utente deve fornire username e una password. Il server risponde con un codice che può indicare l'avvenuta registrazione, oppure, se lo username è già presente, o se la password è vuota, restituisce un messaggio d'errore. Lo username dell'utente deve essere univoco. Come specificato in seguito, le registrazioni sono tra le informazioni da persistere lato server.
  - *login(username, password)*: login di un utente già registrato per accedere al servizio. Il server risponde con un codice che può indicare l'avvenuto login, oppure, se l'utente ha già effettuato la login o la password è errata, restituisce un messaggio d'errore.
  - *logout(username)*: effettua il logout dell'utente dal servizio.

- *searchHotel(nomeHotel, città):* ricerca i dati di un particolare hotel appartenente a una città e li invia all'utente. Questa operazione può essere effettuata anche dagli utenti non loggati.
- *serchAllHotels(città):* ricerca i dati di tutti gli hotel di quella città e li invia all'utente. Gli hotel devono essere inviati ordinati secondo il ranking, calcolato in base alle recensioni. Questa operazione può essere effettuata anche dagli utenti non loggati.
- *insertReview(nomeHotel, nomeCittà, GlobalScore, [ ] SingleScores):* inserisce una review per un hotel di una certa città. Viene indicato sia il punteggio complessivo per quell'hotel che i singoli punteggi per le varie categorie. L'utente deve essere registrato ed aver effettuato il login per effettuare questa operazione.
- *showMyBadges()*: l'utente richiede di mostrare il proprio distintivo, corrispondente al maggior livello di expertise raggiunto. L'utente deve essere registrato ed aver effettuato il login per effettuare questa operazione.

- **HOTELIERServer.** Implementa le seguenti funzionalità

  - inizialmente carica da un file JSON la descrizione degli hotel. A questo proposito si suppone che esista un file, già creato, che riporta la descrizione di un insieme significativo di hotel. Per ogni hotel, vengono registrate le caratteristiche più importanti, tra cui il punteggio sintetico attribuito all'albergo e i punteggi specifici relativi alle diverse categorie e la sua posizione nel ranking locale. Questi punteggi sono pari a 0 alla prima attivazione del server.
  - gestisce la fase di registrazione e di login degli utenti
  - memorizza le informazioni relative a tutti gli utenti e a tutti gli alberghi
  - effettua le ricerche richieste dagli utenti
  - riceve e memorizza le recensioni, e aggiorna il calcolo delle recensioni di quell'utente per l'attribuzione dei distintivi
  - aggiorna periodicamente il ranking degli alberghi di ogni città. Il periodo di tempo che intercorre tra un aggiornamento e il successivo è un parametro di input della applicazione
  - invia su richiesta degli utenti l'ultimo distintivo acquisito da quell'utente
  - invia una notifica a tutti gli utenti loggati ogni qual volta che cambia l'identità dell'hotel primo classificato in qualche ranking locale
  - persiste periodicamente le strutture dati utilizzate per la memorizzazione di utenti e di contenuti su una struttura dati permanente.

~~2.2 Funzionalità aggiuntive (per gli studenti del vecchio ordinamento)~~

~~Gli studenti del vecchio ordinamento dovranno implementare anche le seguenti funzionalità~~

- ~~**HOTELIERCustomerClient.**~~
  - ~~il client deve, al momento del login, registrare il suo eventuale interesse a ricevere notifiche sull'aggiornamento di uno o più ranking locali, indicando le città di interesse~~

- ~~**HOTELIERServer.**~~

~~● Il server deve, ogni volta che un ranking locale varia, mandarlo a tutti gli utenti che hanno registrato interesse per quel ranking.~~

3. Specifiche per l'implementazione

Nella realizzazione del progetto devono essere utilizzate molte delle tecnologie illustrate durante il corso. In particolare:

- gli utenti interagiscono con HOTELIER mediante un client che utilizza un'interfaccia a linea di comando. E' facoltativa l'implementazione di un'interfaccia grafica.
- fase di registrazione
    - (studenti nuovo ordinamento): questa fase viene implementata instaurando una connessione TCP con il server
    - ~~(solo studenti vecchio ordinamento) questa fase viene implementata mediante RMI~~
- Fase di login: deve essere effettuata come prima operazione, dopo che è stata effettuata la registrazione, usando una connessione TCP instaurata con il server. Il login è necessario nel caso si voglia publicare una recensione o vedere il proprio distintivo.
    - ~~(solo studenti vecchio ordinamento) In seguito alla login il client si registra a un servizio di notifica del server per ricevere aggiornamenti su alcuni ranking di suo interesse. Il servizio di notifica deve essere implementato con il meccanismo di RMI callback. Il client mantiene una struttura dati per tenere traccia dei ranking modificati, che è aggiornata in seguito alla ricezione della callback.~~
- Dopo la login effettuata con successo, l'utente interagisce, secondo il modello client-server (richieste/risposte), con il server sulla connessione TCP persistente creata, inviando uno dei comandi elencati nella sezione 2.1. Tutte le operazioni sono richieste su questa connessione TCP.
- quando il server ricalcola i ranking locali, verifica se c'è un cambiamento nella prima posizione e, in questo caso, invia il nome del nuovo albergo che occupa la prima posizione e della città in cui si trova, a tutti gli utenti loggati. La trasmissione avviene tramite l'invio di un pacchetto UDP su un gruppo di multicast a cui sono iscritti tutti gli utenti loggati.
- il server può essere realizzato con JAVA I/O e threadpool oppure può effettuare il multiplexing dei canali mediante NIO (eventualmente con threadpool per la gestione delle richieste).
- Il server definisce opportune strutture dati per memorizzare le informazioni relative agli utenti registrati, agli hotel, alle recensioni. Tali informazioni devono essere persistite periodicamente.
- i file per la memorizzazione degli utenti e dei contenuti memorizzano le informazioni in formato JSON.

4. Modalità di svolgimento e di consegna del progetto

Il progetto deve essere eseguito individualmente.

Il materiale da consegnare comprende:

- Il codice dell'applicazione e di eventuali programmi utilizzati per il test delle sue funzionalità. Si tenga presente che il codice
  - deve compilare correttamente da riga di comando (ovvero invocando direttamente il compilatore *javac*). In caso contrario, il progetto non verrà considerato valido.
  - deve essere ben commentato.
  - le classi che contengono un metodo main devono contenere "main" nel nome, es. ServerMain.java; per le altre classi non ci sono vincoli, ma nomi mnemonici sono ovviamente apprezzati.
  - oltre al codice sorgente, è necessario consegnare un file JAR eseguibile per ogni applicazione (es. un file JAR per i client e uno per il server).
  - i parametri di input delle applicazioni (numeri di porta, indirizzi, valori di timeout, intervallo per aggiornamento dei ranking, ecc.) devono essere letti automaticamente da appositi file di configurazione testuali da consegnare assieme al resto del codice (due file separati per client e server). Non è consentito leggere i parametri in modo "interattivo" (ovvero facendo in modo che sia il programma a chiederli dopo essere stato avviato).
  - in caso di progetti realizzati con Eclipse, IntelliJ IDEA o altri IDE, è obbligatorio consegnare solamente il codice sorgente, rimuovendo eventuali altri file (o directory) creati dall'IDE per gestire il progetto.
  - eventuali librerie esterne utilizzate (jar) vanno allegate al progetto.

- la relazione in formato pdf, deve contenere
  - la definizione delle scelte effettuate nei punti del progetto lasciati alla personale interpretazione (es l'algoritmo per il calcolo del punteggio degli hotel)
  - una definizione delle strutture dati utilizzate sia lato server che lato client
  - uno schema generale dei thread attivati sia lato server che lato client
  - una descrizione delle eventuali primitive di sincronizzazione utilizzate dai thread per accedere a strutture dati condivise
  - una sezione di istruzioni su come compilare ed eseguire il progetto (librerie esterne usate, argomenti da passare al codice, sintassi dei comandi per eseguire le varie operazioni...). Questa sezione deve essere un manuale di istruzioni semplice chiaro per gli utilizzatori del sistema.
  - l'organizzazione e la chiarezza della relazione influiranno sul voto finale.

Relazione e codice sorgente devono essere consegnati su Moodle in un unico archivio compresso in formato zip.