RELAZIONE DEL PROGETTO DI RETI II

RETI II LABORATORIO – UNIVERSITA' DEL PIEMONTE ORIENTALE –

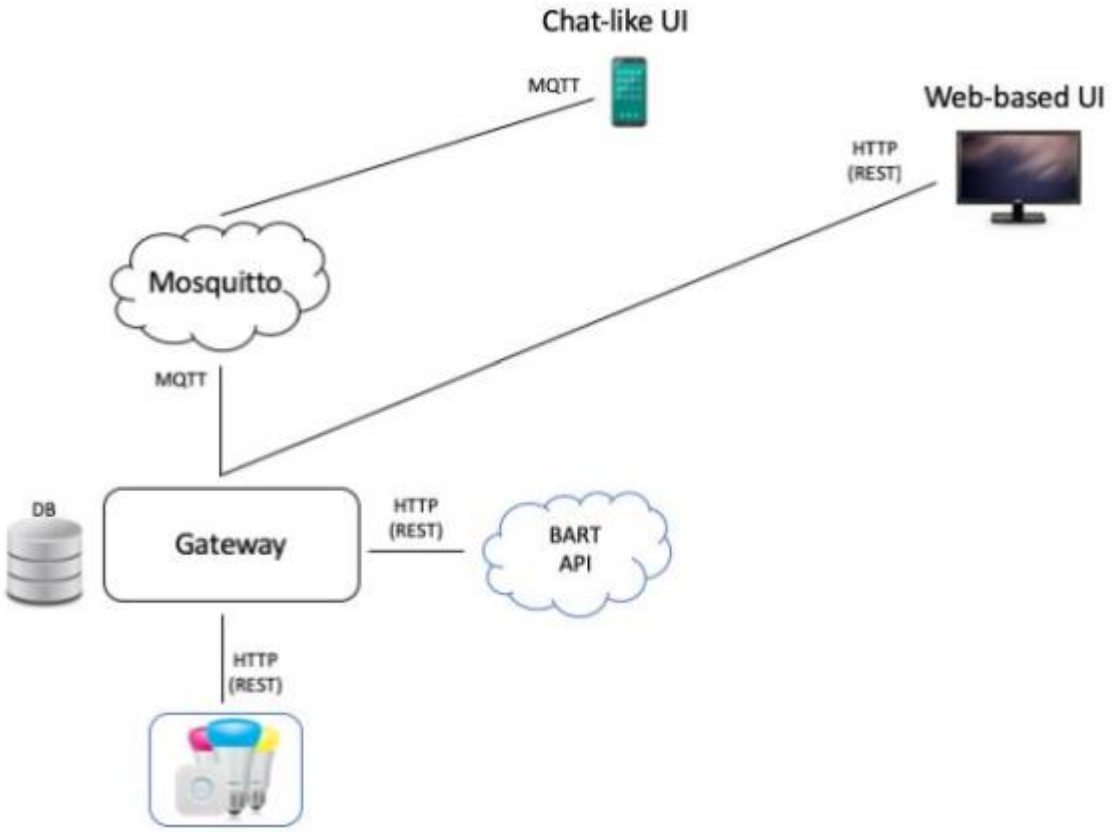
A.A. 2019-2020

Politi Domenico 20012478

SOMMARIO:

1. SPECIFICHE E ARCHITETTURA

L’architettura generale del progetto:



N.B. L’architettura del progetto sviluppato non contiene la parte riguardante Mosquitto.

Le specifiche del progetto:

- Il gateway deve raccogliere i dati di passaggio dei treni (previsti) utilizzando le API messe a disposizione da https://www.bart.gov/schedules/developers/api
- L’interfaccia grafica, web-based, deve permettere al viaggiatore di visualizzare le informazioni sulla stazione e selezionare uno o più passaggi di treni alla stazione scelta, interrogando il gateway attraverso le sue API REST
- L’interfaccia grafica, sempre utilizzando le API REST del gateway, deve permettere al viaggiatore di prenotare un posto sul treno precedentemente selezionato, facendogli indicare la sua necessità di portare una bici a bordo

- Per ogni prenotazione di posto con bici, il gateway deve accendere una luce (Philips Hue) verde, aumentandone sempre più l'intensità all'aumentare delle prenotazioni con bici
- Nel caso in cui il treno diventi troppo affollato, il gateway deve far diventare la "luce bici" rossa lampeggiante (ci si aspetta che i passeggeri con bici si comportino adeguatamente); se non ci sono bici prenotate, la luce deve rimanere spenta
- Nel caso in cui il treno diventi troppo affollato, l'interfaccia grafica, sempre utilizzando le API REST del gateway, deve visualizzare tale informazione in fase di prenotazione posto (e impedire nuove prenotazioni)
- Il gateway è responsabile di memorizzare le associazioni tra viaggiatore, bici, treno, … in un apposito database

2. OBBIETTIVO

Si vuole realizzare una stazione BART (https://www.bart.gov) intelligente. La stazione conterrà un gateway che dovrà gestire le informazioni sulla stazione stessa (disponibilità di parcheggi, di posti bici, …), quelle sui treni in transito (linee, orari previsti, orari in tempo reale, …) e le informazioni fornite dai passeggeri.

I passeggeri, infatti, utilizzando un apposito sito web, potranno prenotare un posto su uno specifico treno, indicando anche la loro necessità di trasportare una bicicletta.

Il gateway, tenendo conto di tutte queste informazioni, accenderà una luce verde in stazione per indicare la prenotazione di bici. Più prenotazioni con bici vengono effettuate, più la luminosità della luce dovrà essere intensa. Il gateway, inoltre, dovrà stimare quante persone si troveranno su un specifico treno: se il treno diventasse troppo affollato, dovrà notificare i passeggeri con bicicletta del fatto che non potranno salire con la loro bici sul treno; contestualmente, la luce relativa alle

“prenotazioni con bici” dovrà diventare di un rosso intenso e lampeggiante.

3. COMPONENTI HARDWARE E SOFTWARE UTILIZZATI

Componenti hardware utilizzati:

- Portatile Windows
- Raspberry Pi (solo per l’esame)
- Philips Hue Lights (solo per l’esame)

Componenti software utilizzati:

- Eclipse
- IntelliJ IDEA
- Philips Hue Emulator
- Visual Studio Code
- GitHub
- Java
- HTML
- CSS
- Javascript
- JQuery
- Spark
- Handlebars
- SQLite
- Gradle

4. SCELTE IMPLEMENTATIVE

Ho iniziato implementando la classe BARTClient che utilizza le classi JsonOb, Root, Request, Schedule e Trip per simulare la struttura del Json fornito dalle API. BARTClient è implementato come thread.

Ho costruito le classi Passeggero e Treno che contenessero, relativamente, le informazioni dei passeggeri e dei treni.

Ho implementato la classe TrenoService che gestisce tutta la parte relativa al server, quindi interagisce con il DB, esegue come thread HueClient e BARTClient. Il ruolo principale è rendere disponibile le informazioni a TrenoClient. Inoltre consente di aggiungere nuovi passeggeri al DB.

TrenoClient prende le informazioni relative ai treni dall'URL e le pubblica sul sito.

HueClient si occupa di tutta la gestione delle luci Hue.

DBConnect gestisce la connessione con il DB SQLite.

Infine, la classe TrenoDao si occupa di tutta la gestione dei passeggeri e dei treni sul DB, aggiungendoli, cancellandoli, modificandoli e ottenendo informazioni attraverso le query.

I treni gestiti sono quelli che partono dalla stazione di Oakland Airport e arrivano in quella di Coliseum.

6. LIMITAZIONI

Ha poche limitazioni, in quanto può gestire tutti i treni che vengono forniti dalle API, a patto che ci siano lampadine per rappresentare i vari treni. Un'altra limitazione è rappresentata dai posti che sono molto ridotti a fini dimostrativi per mostrare i cambiamenti delle luci e relativi al sito.