

DOCUMENTAZIONE TECNICA

“Mr. Alfred”

Introduzione

Come caso di studio per il corso di Sistemi ad agenti abbiamo deciso di sviluppare un chatbot che gestisca le prenotazioni di un ristorante.

Nello specifico abbiamo implementato, dal lato del cliente che intende effettuare una prenotazione, un chatbot su Telegram utilizzando la piattaforma Dialogflow mentre dal lato del ristoratore abbiamo implementato una pagina web che permetta la gestione delle prenotazioni effettuate mediante il bot.

Per il salvataggio delle prenotazioni invece abbiamo utilizzato Firebase.

Il codice sorgente del fulfillment del bot e il codice sorgente del sito sono disponibili in questa repository GitHub: <https://github.com/nicolapiccolo/bookingBot>



Chatbot

Dialogflow è una piattaforma di comprensione del linguaggio naturale utilizzata per progettare e integrare un'interfaccia utente conversazionale in app mobili, nel nostro caso il chatbot è disponibile su Telegram.

Per la gestione delle prenotazioni abbiamo individuato quelli che secondo noi sono gli intent principali (ovvero ciò che l'utente finale può richiedere all'agente), di conseguenza l'agente cercherà di interpretare la richiesta dell'utente individuando l'Intent che più si avvicina.

Gli intent individuati sono:

- **Prenotazione** permette l'acquisizione dei dati necessari per inserire una nuova prenotazione nel database;
- **Cancellazione** permette di eliminare una prenotazione una volta acquisiti i dati necessari per identificarla;
- **Disponibilità** permette di visualizzare i posti liberi di un determinato giorno a una determinata fascia oraria;
- **Modifica** permette di modificare una prenotazione una volta acquisiti i dati necessari per la modifica;
 - ➔ **ModificaOrario** in particolare permette di modificare la data e l'ora di una prenotazione;
 - ➔ **ModificaPosto** in particolare permette di modificare il numero di posti di una prenotazione;
 - ➔ **ModificaPostoOrario** dà la possibilità di modificare sia orario che numero posti della prenotazione
- **MostraPrenotazioni** permette di visualizzare tutte le prenotazioni effettuate da un determinato utente;
- **Orari Ristorante** permette di visualizzare l'orario di apertura e di chiusura del ristorante.

Dialogflow permette, nella sezione Fulfillment, di inserire codice in linguaggio Node.js.

In particolare per ogni intent è presente una funzione che definisce il comportamento dell'agente.

Gestione funzionalità

La funzione *disponibilita()* richiede come input dell'utente il giorno e l'ora in cui questo vuole verificare la disponibilità del locale. Una volta acquisiti questi dati viene richiamata la funzione *read()*, a cui vengono passati la data e l'ora e come risultato ritorna il numero di posti disponibili in quella fascia oraria. Nel caso in cui la fascia selezionata non sia valida viene informato l'utente che a quell'ora il ristorante è chiuso.

La funzione *prenota()* richiede in input un nominativo, l'indirizzo email a cui verrà inviata la conferma di prenotazione in caso di esito positivo, un recapito telefonico, il numero di posti e la data e l'ora della prenotazione. Acquisiti questi dati viene richiamata la funzione *read()* per verificare la disponibilità nella fascia oraria richiesta, in caso positivo verrà assegnato un id univoco alla prenotazione (comunicato all'utente nel bot e nell'email di conferma) e verrà salvata la prenotazione nel database con lo stato "PENDING".

In caso negativo verrà comunicato all'utente il motivo per cui la prenotazione non è stata accettata.

La funzione *Cancellazione()* richiede in input l'email dell'utente e id della prenotazione per verificare che la prenotazione è presente nel database e che appartenga all'utente che ne fa richiesta. Se il controllo ha esito positivo verrà cambiato lo stato della prenotazione in "CANCELLED" e verrà inviata una mail di avvenuta cancellazione, altrimenti verrà inviato un messaggio indicando che la prenotazione cercata non è presente.

La funzione *Modifica()* come la *Cancellazione()* richiede in input l'email dell'utente e id della prenotazione per verificare che questa sia presente nel database e che appartenga all'utente che ne fa richiesta. Successivamente l'utente può richiedere la modifica della data e dell'ora, del numero di posti o di entrambi. In base al tipo di modifica selezionata, verranno richiamate le funzioni *ModificaOrario()*, *ModificaPosto()* e *ModificaOrarioPosto()*:

- La funzione *ModificaOrario()* richiede in input la nuova data e l'ora della prenotazione. Verifica che il numero di posti precedentemente prenotati siano disponibili per la nuova data e ora, se il controllo ha esito positivo la modifica verrà memorizzata nel database e verrà inviata una mail di conferma, altrimenti verrà inviato un messaggio indicando che il numero di posti richiesto non è disponibile in quella data;
- La funzione *ModificaPosto()* richiede in input il nuovo numero di posti della prenotazione. Verifica che il nuovo numero di posti sia disponibile, in caso positivo verranno memorizzati gli aggiornamenti e verrà inviata un'email di conferma altrimenti verrà inviato un messaggio indicando che il numero di posti richiesto non è disponibile;
- La funzione *ModificaOrarioPosto()* combina le funzionalità di *ModificaPosto()* e *ModificaOrario()*.

La funzione *mostraPrenotazioni()* richiede in input l'email utilizzata in fase di prenotazione e, una volta recuperate le prenotazioni effettuate da quell'indirizzo email ancora nello stato "PENDING", stamperà un riepilogo contenente le informazioni sulle prenotazioni di quell'utente qualora ci fossero.

Firestore

Come database abbiamo utilizzato Firestore nella versione realtime, in modo tale che tutti i client connessi al database ricevano in tempo reale gli aggiornamenti. Il database contiene le informazioni relative alle varie prenotazioni e inoltre contiene anche i parametri:

- **span**, ovvero la durata media di una consumazione che viene utilizzata nel calcolo dei posti a disposizione;
- **time**, che a sua volta contiene l'orario di apertura e di chiusura del ristorante;
- **tot_seat**, ovvero il numero totale di coperti del ristorante;

Tutti questi parametri vengono stabiliti dal gestore del ristorante mediante la pagina web dedicata.

```
booking-1091b
├── pren
│   ├── -MBtc1hXeuGOi5Ke0x_p
│   │   ├── cell: 328965421
│   │   ├── customer: "Oronzo"
│   │   ├── email: "nacci.aronzo@gmail.co
│   │   ├── id: 29
│   │   ├── n_posti: "1"
│   │   ├── orario: "2020-07-11T19:00:00+02:0
│   │   ├── status: "pending"
│   │   └── time: "2020-07-10T16:47:45+02:0
│   ├── -MBtcvvtileXXXdhnKsF
│   ├── -MBxGCN_89phXJanChb5
│   ├── -MC7VLCsv672xBXoNzyh
│   ├── -MCBva4hhcfVYjPiTxXt
│   ├── -MCCf71NI2db9UJxgJMo
│   └── -MCChXfqDH_JVRgneduR
├── span: "1"
├── time
│   ├── closing: "01:00"
│   └── opening: "12:00"
└── tot_seat: "50"
```

Pagina Web

Si è pensato di creare una pagina web back-end per permettere al ristoratore di avere il pieno controllo dei contenuti del database, in modo da fornire uno strumento che faciliti la lettura e la modifica delle prenotazioni e delle informazioni riguardanti il ristorante.

Quindi tramite php viene effettuata la connessione al database Firebase tramite le rispettive credenziali. Vengono effettuate delle interrogazioni alla base di dati restituendoci un file JSON con i risultati. Dunque viene scansionata la struttura del file al fine di ritrovare le informazioni di interesse per costruire dinamicamente le componenti html per visualizzare le prenotazioni. In particolare la home page presenta una lista di prenotazioni sotto forma di tabella schedulata con i filtri today, future e history, in maniera da facilitare il ristoratore nella ricerca delle prenotazioni: del giorno corrente, di quelle nei giorni a venire o dello storico di prenotazioni ricevute.

Per ogni prenotazione nella sezione Today e Future è presente un pulsante "edit" per permettere all'utente di modificare direttamente data, ora, numero di posti e lo stato della prenotazione (Pending->Confermed se la prenotazione è stata consumata o Pending->Cancelled nel caso in cui il cliente non si è presentato o ha disdetto la prenotazione). Quando viene premuto il pulsante "Save changes" viene fatto un controllo per verificare la disponibilità con i nuovi parametri inseriti, se il controllo ha esito positivo viene effettuata una query per aggiornare i valori nel database e contemporaneamente viene mandata una mail di notifica all'utente. In caso di esito negativo viene visualizzato un pop-up con il messaggio dell'errore riscontrato (numero posti non disponibile, ecc...).

Nel menù presente lateralmente abbiamo inserito una pagina Settings per permettere la modifica di parametri sul ristorante quali: orario di apertura e di chiusura, il numero di posti totale e il tempo stimato di consumazione. Quest'ultimo è utilizzato nella fase di controllo di disponibilità della prenotazione, in quanto si verifica che i posti occupati dal range impostato dal ristoratore all'orario indicato nella prenotazione non eccedano i posti totali del ristorante. I parametri, in caso di modifica, vengono salvati nel database e visualizzati in real time.

Manuale d'uso

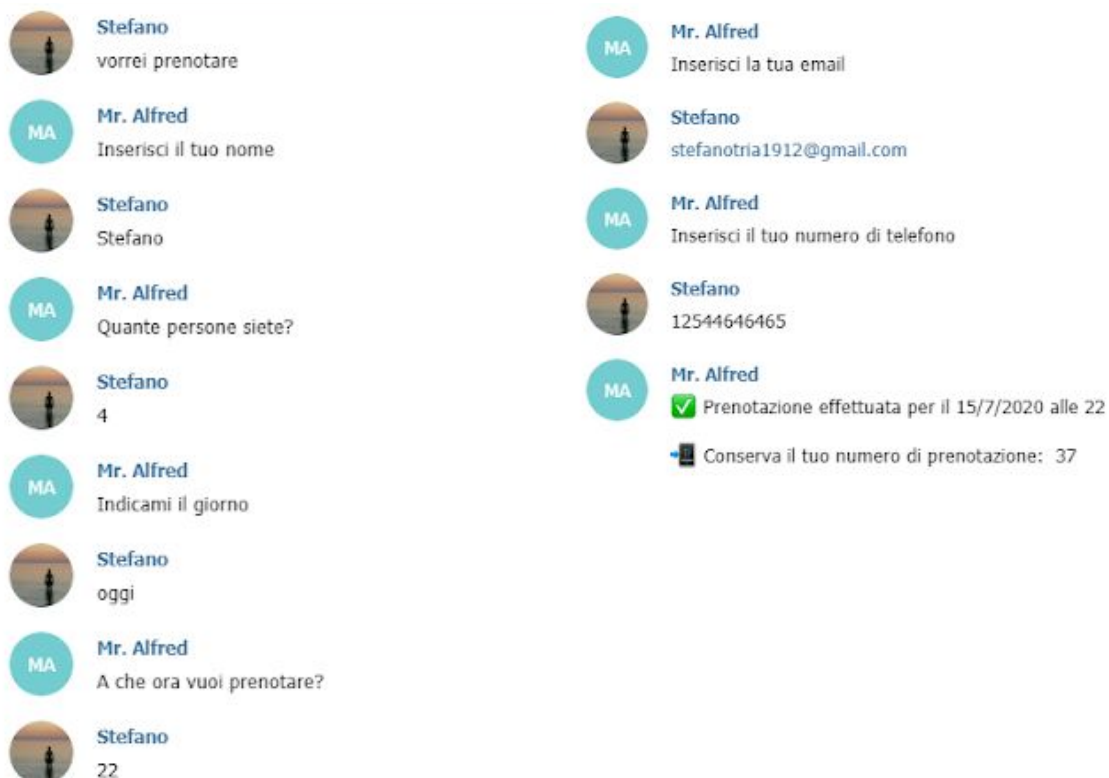
Per testare il chatbot è necessario disporre di un account Telegram.

L'username del chatbot è **gvng_booking_bot**, mentre il nome è **Mr. Alfred**.

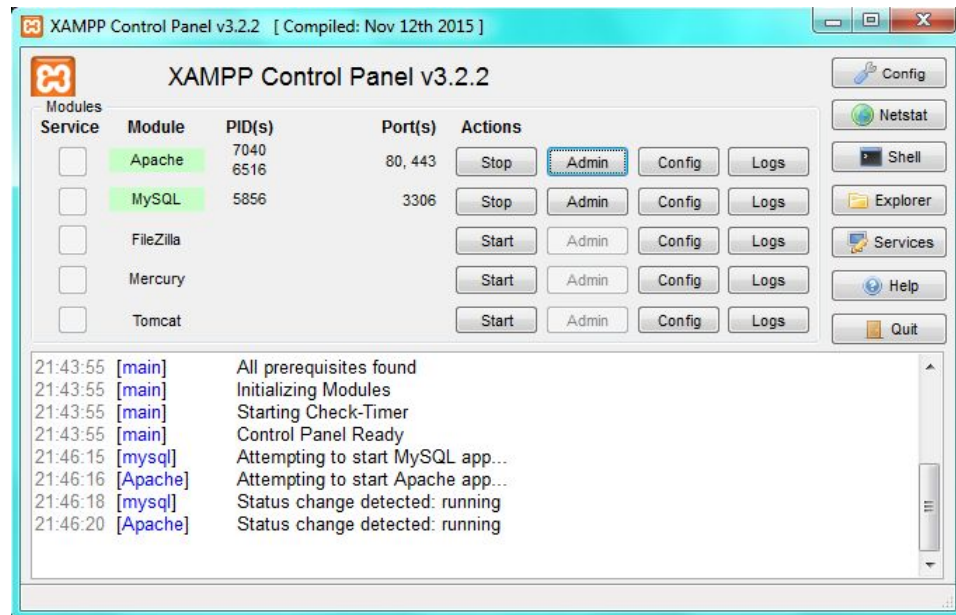
Una volta avviato il bot è possibile interagire sia tramite i pulsanti, selezionando l'operazione desiderata:



Oppure inviando al bot un messaggio testuale:



Per quanto concerne la pagina web di gestione, essendo questa scritta in php, è necessario installare un software come XAMPP per poterla visualizzare in locale. Una volta installato ed avviato XAMPP è necessario avviare il server Apache.



Fatto questo bisogna inserire la cartella di progetto “booking” all’interno del seguente percorso: xampp\htdocs

Per poter poi visualizzare la pagina web basterà aprire il browser all’indirizzo:
`http://localhost/booking/index.php`

In alternativa è anche possibile accedere al sito web collegandosi al seguente link:
<https://mralfredmanager.altervista.org/>

La pagina che verrà visualizzata sarà questa:

