Corso di Tecnologie e linguaggi per il web
A.A. 2019/2020

Valerio Cislaghi

10 giugno 2020

Indice

Capitolo 1

Introduzione

Il progetto realizzato consiste in una web live chat: Un utente, dopo essersi registrato, può iniziare a chattare con i propri amici/conoscenti solamente conoscendo il loro username (scelto appositamente nella fase di registrazione).
Per ottimizzare la velocità di esecuzione ho deciso di non affidarmi a framework lato client (es: jquery, angular ecc..), ma ho realizzato tutto il codice con javascript nativo, attraverso il pattern Model View Controller con la specifica standard ECMAScript 6.
Per il server ho utlizzato Node.js con il framework Express.
come DBMS ho utilizzato Postgresql, con la libreria 'pg-promise' per interfacciarmi ad esso.

Figura 1.1: Logo dell'applicazione

1.1 Destinatari

L'applicazione è rivolta al 'grande pubblico', essendo una chat molto semplice da usare; infatti basta registrarsi ed iniziare a chattare. Per chattare, essendo una web application, si potrà usare qualsiasi dispositivo che si possa connettere ad internet (è preribile utilizzare una versione desktop dell'applicazione)

1.2 Flusso dei dati

I dati richiesti per la registrazione sono:

- Username
- Password
- URI di un immagine per il profilo

1.3 Aspetti tecnologici

1.3.1 Struttura del codice

Model View Controller

Tutto il progetto si basa sulla realizzazione del pattern di sviluppo: Model View Controller:

- View = componenti che si occupano di gestire una parte ben specifica dell'interfaccia grafica, come per esempio le anteprime delle chat (side-panel.js) e la chat vera e propria (chat-frame.js).
- Controller = componenti che reagiscono ad un evento e passano il controllo al model adatto a gestire una specifica situazione. Un

evento può essere generato dalla view (es: click sul bottone di invio del messaggio), ma anche un evento esterno dalla GUI (es: sono arrivati dei nuovi messaggi dal server).

- Model = gestisce le iterazioni con il REQUSTER (unico oggetto che comunica con il server) e aggiorna di conseguenza la view associata a lui.

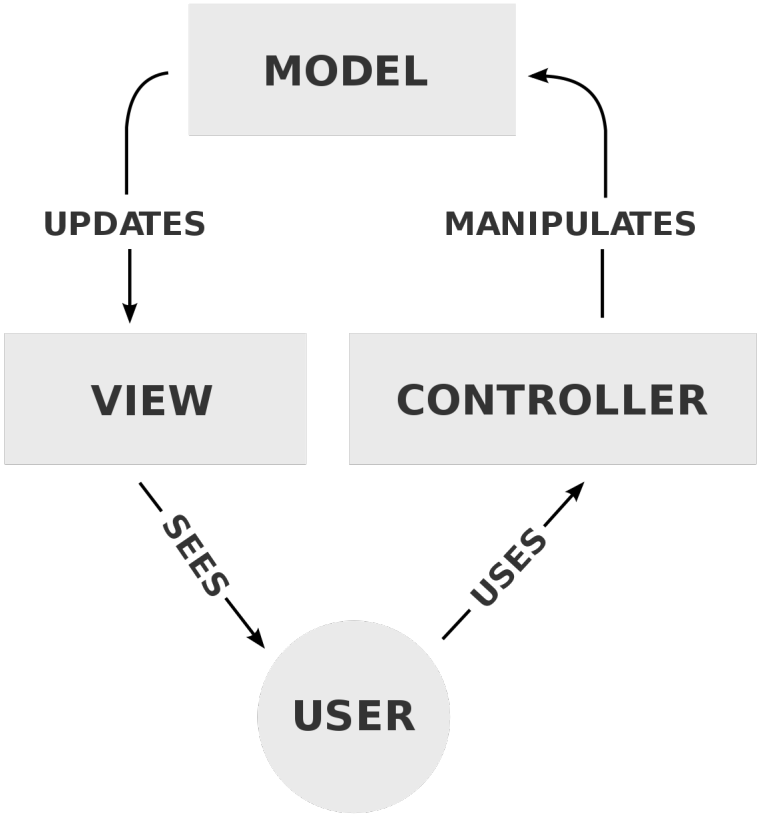

Figura 1.2: Model View Controller

Prototype Chain

Ho deciso di utilizzare le potenzialità dei prototipi con cui creare classi e oggetti per modellare tutto il progetto. Infatti, Per la gestione di Chat, Utenti, Messaggi ecc... **javascript client e Node.js utilizzano gli stessi oggetti, per una maggiore compatibilità**.

1.3.2 Tecnologie utilizzate

Inizialmente ho strutturato il progetto con l'utilizzo di jQuery, ma dopo aver fatto qualche analisi sulle performance (con gli strumentri di chrome), mi sono convinto che utilizzare le funzionalità native del linguaggio Javascript, come i prototipi, lo scope, il binding delle funzioni ecc... avrebbe migliorato le performance e la stabilità del progetto.
per il server, mi sono invece affidato a Node.js e Express con il quale ho realizzato le routes opportune per le viste e per le API. Come moduli ho utiilzzato

- ejs, per il rendering dell'html.
- pg-promise, per l'interrogazione del DataBase
- nodemon, per il refresh automatico del server dopo qualche cambiamento del codice

Capitolo 2

Interfacce

2.1 Login

L'utente, se già possessore di un account, potrà accedere all'applicazione tramite username e password inseriti durante l'iscrizione, altrimenti dovrà iscriversi.

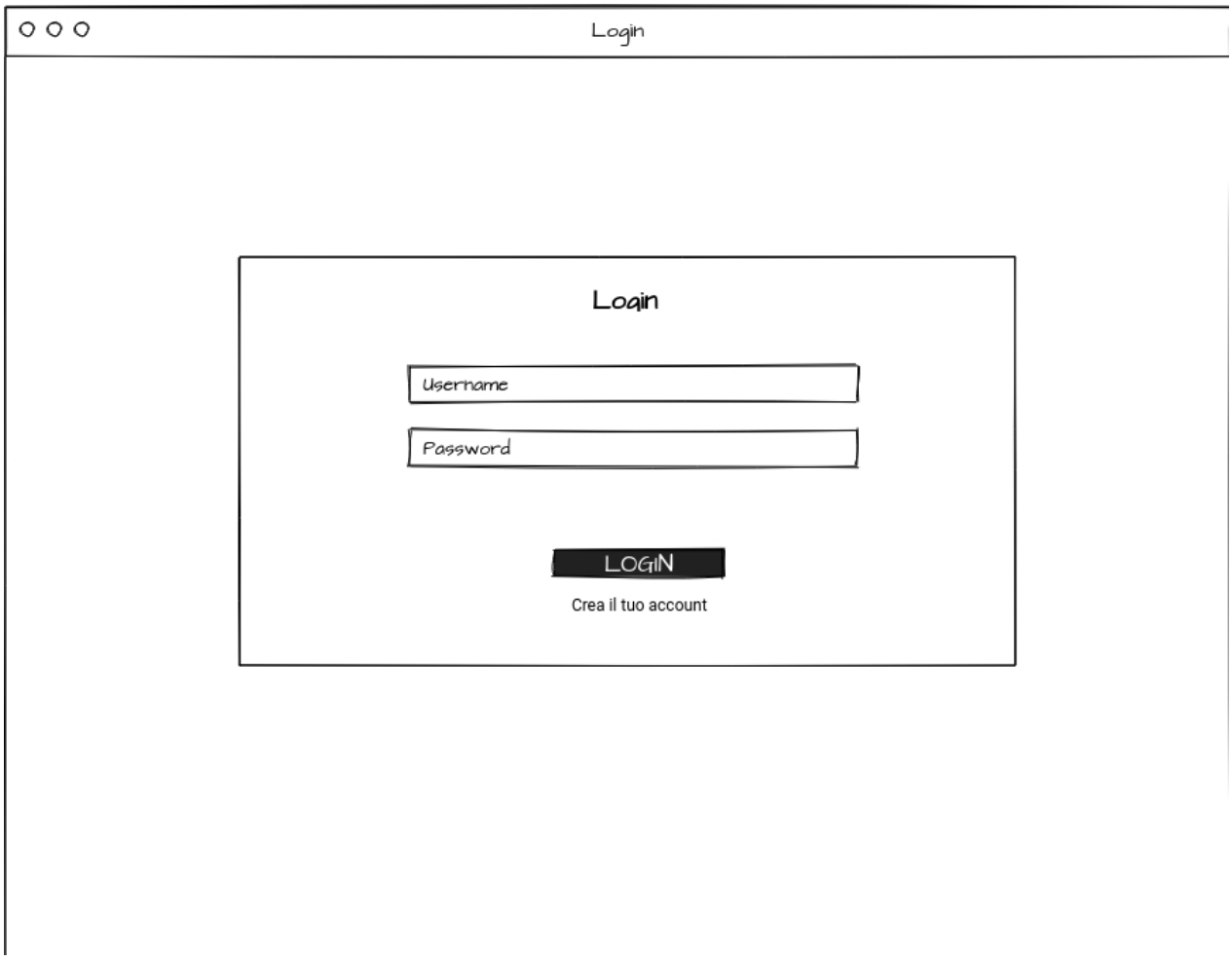

Figura 2.1: Mock Login

Durante la fase di registrazione il client, tramite chiamate REST, controllerà se lo username iserito è già presente nel database, in caso di successo mostrerà un messaggio di errore all'utente chiedendogli di inserire un altro nome utente.
Quindi prima di inviare la richiesta di inserimento dell'utente il client controllerà se lo username non è occupato e se le password inserite

corrispondono.

2.2 Chat

La pagina principale della web application, a sinistra troviamo le nostre chat già attive, mentre attraverso la barra di ricerca possiamo trovare gli utenti, con cui non abbiamo chat attive, per poter iniziare a messaggiare.

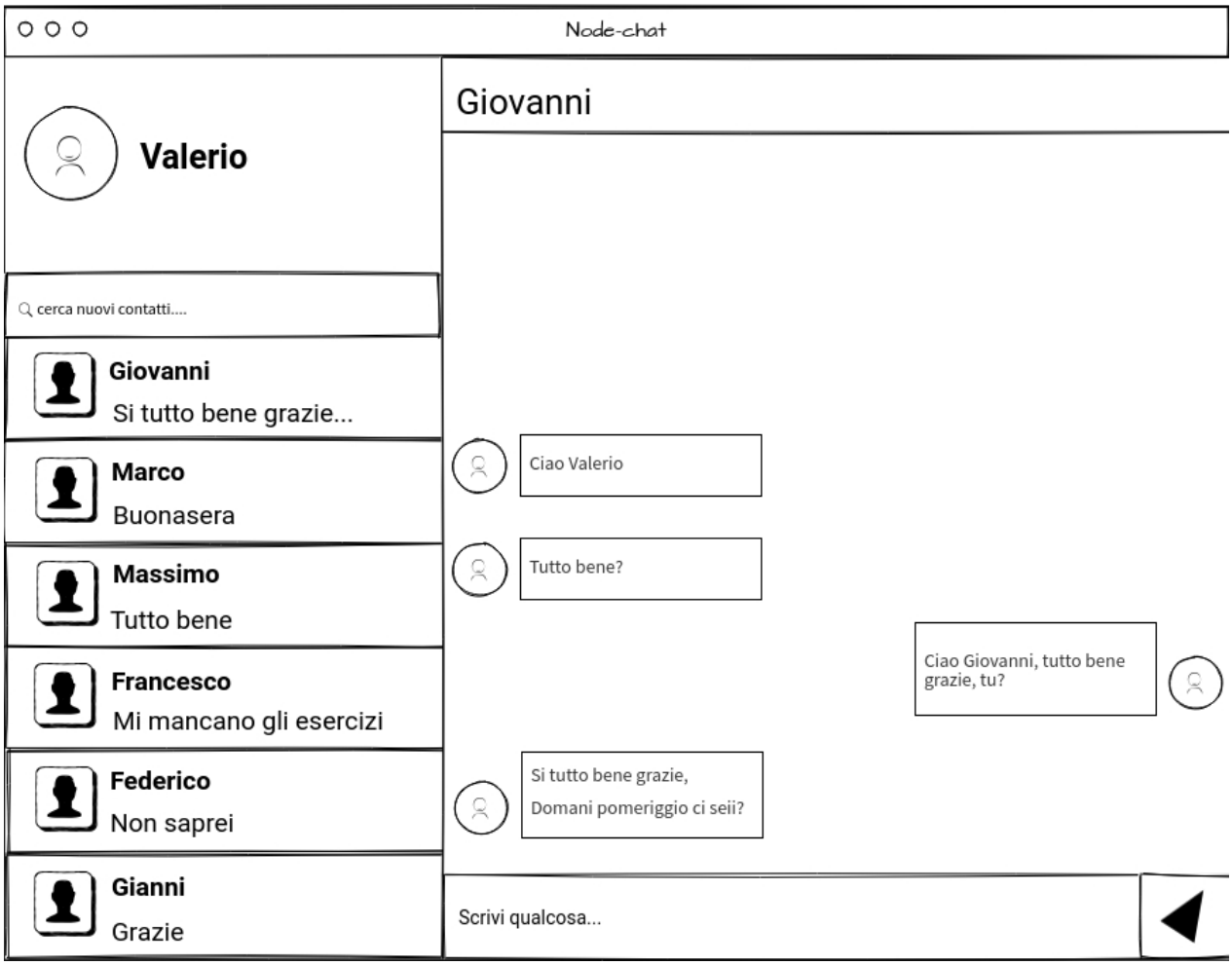

Figura 2.2: Mock Chat

Le chat verranno aggiornate automaticamente, dunque non c'è bisogno di aggiornarle manualmente. Quando inviamo un messaggio e lo vediamo apparire sulla chat, vuol dire che il messaggio è stato inserito correttamente nel DataBase, e di conseguenza ricuvuto dal destinatario.

Capitolo 3

Conclusione

"I always thought something was fundamentally wrong with the universe" [1]

Bibliografia

[1] D. Adams. *The Hitchhiker's Guide to the Galaxy*. San Val, 1995.