

Università degli studi di Palermo

Dipartimento di Ingegneria
2178 - Ingegneria Informatica
Anno accademico 2020/21

PickMeUp!

Object Design Document

Docenti

Prof.ssa Valeria **SEIDITA** (Ingegneria del software)

Prof. Roberto **PIRRONE** (Programmazione web e mobile)

A cura del gruppo **Bytecoders**

Object Design Document

Introduzione	3
Scelte nella progettazione degli oggetti	3
Packages	5
Organizzazione directory (front-end)	7
Organizzazione directory (back-end)	8

ODD

1. Introduzione

1.1. Scelte nella progettazione degli oggetti

Per la realizzazione del software proposto, in forma di applicazione web desktop e mobile, si è scelto di utilizzare come gestore dei pacchetti Javascript NPM (v6.14.12).

Per semplificare lo sviluppo dell'applicazione, si è deciso di utilizzare il framework **ExpressJS (v4.16.1)** lato backend e la libreria Javascript **ReactJS (v17.0.2)** lato frontend.

Per la realizzazione delle interfacce grafiche, si è scelto di utilizzare il framework CSS **Bootstrap (v5.0.1)**, insieme al framework **React-Bootstrap (v1.6.1)**.

I pacchetti necessari per lo sviluppo del software per il lato frontend sono:

- **@fortawesome/fontawesome-svg-core (v1.2.35)**,
@fortawesome/free-solid-svg-icons (v5.15.3),

@fortawesome/react-fontawesome (v0.1.14) sono delle librerie di icone svg open source. Nello specifico si è utilizzata la libreria compatibile con react.

- **@popperjs/core (v2.9.2)** : è un engine di posizionamento, ha lo scopo di calcolare la posizione di un elemento per posizionarlo vicino ad un altro elemento.
- **@react-leaflet/core (v1.0.2)** : è un pacchetto di leaflet compatibile con react.
- **@testing-library/jest-dom(v5.13.0)**
- **@testing-library/react(v11.2.7)**
- **@testing-library/user-event(v12.8.3)**
- **axios (v0.21.1)** : è un client HTTP promise-based per node.js e il browser. Lato server usa il modulo nativo di node.js, mentre lato client (browser) utilizza il modulo basato su XMLHttpRequest.
- **bootstrap (v5.0.1)** : è un framework CSS che definisce delle classi per modellare il CSS dei componenti.
- **codice-fiscale-js (v2.3.9)** : è una libreria di Javascript per calcolare e convalidare il codice fiscale italiano.
- **crypto-js (v4.0.0)** : è una libreria di Javascript per la crittografia standard.
- **.framer-motion (v4.1.17)** : è una libreria opensource per le animazioni e le transizioni.
- **jquery (v3.6.0)** : è una libreria di Javascript e nasce con lo scopo di semplificare la selezione, la manipolazione, la gestione degli eventi e l'animazione di elementi DOM in pagine HTML.
- **leaflet (v1.5.1)** : è una libreria opensource di Javascript per la visualizzazione di mappe interattive.
- **node-sass (v6.0.1)** : è una libreria che consente di associare Node.js a LibSass e permette di compilare nativamente file .scss in cc in maniera automatica tramite un middleware di connessione.
- **react-router-dom (v5.2.0)** : è un API che mi permette di gestire la navigazione e il reindirizzamento tra le varie schermate.
- **react-places-autocomplete(v7.3.0)** : libreria che definisce componenti React per l'utilizzo della API Google Places.

I pacchetti necessari per lo sviluppo del software per il lato backend sono:

- **cookie-parser (v1.4.4)** : è un pacchetto di utilità per la gestione dei cookie ed è usato per ricavare dagli headers della richiesta il cookie associato all'utente.
- **cors (v2.8.5)** : è un pacchetto di utilità per sfruttare un middleware che può essere usato per abilitare CORS(Cross-origin resource sharing) con diverse opzioni.
- **crypto-js (v4.0.0)** : è una libreria di Javascript per la crittografia standard.

- **http-errors (1.6.3)** : utilizzato per creare errori HTTP lato server e reindirizzarli lato client.
- **jsonwebtoken (v8.5.1)** : è una libreria usata per generare dei token definiti in modo compatto e autonomo per la trasmissione sicura di informazioni tra le parti come oggetti JSON.
- **jspdf (v2.3.1)** : è una libreria per generare PDF in Javascript.
- **mongodb (v3.6.9)** : è un driver per node.js per implementare le interrogazioni e la comunicazione con un database MongoDB.
- **morgan (v1.9.1)** : è un middleware per le richieste HTTP per node.js
- **nodemailer (v6.6.2)** : è un modulo utilizzato per l'invio delle e-mail agli utenti per comunicazioni di interesse.
- **secure-random-password (v0.2.3)** : è un generatore casuale di password, usato nel nostro progetto, per reimpostare la password.

Per gestire le richieste **HTTP CRUD** tra client e server, si è preferito utilizzare la libreria **axios (v0.21.1)**.

Infine, a supporto dell'interfaccia grafica, si è scelto di utilizzare la libreria di icone **Font Awesome (v5.15.3)**.

La scelta dell'uso combinato di **ReactJS** ed **ExpressJS** ha fornito potenti strumenti per la mappatura degli oggetti Boundary, Control ed Entity individuati in fase di analisi.

Le Boundary del sistema presenti nel **RAD** sono state tradotte e programmate in componenti **React**, le Control precedentemente individuate sono state mappate nelle rotte di **Express** e per le Entity si sono utilizzati gli Hooks di React e il **localStorage** della **Web Storage API** di Javascript.

Per interfacciarsi al database si è creato un middleware che fornisce metodi per la gestione della connessione al DBMS.

Per effettuare le query, invece, è stato scelto di attenersi al paradigma di progettazione di ExpressJS che prevede l'utilizzo dei cosiddetti **Model**, nei quali sono state divise tutte le operazioni effettuate in fase di analisi dalla DBMS Boundary.

2. Packages

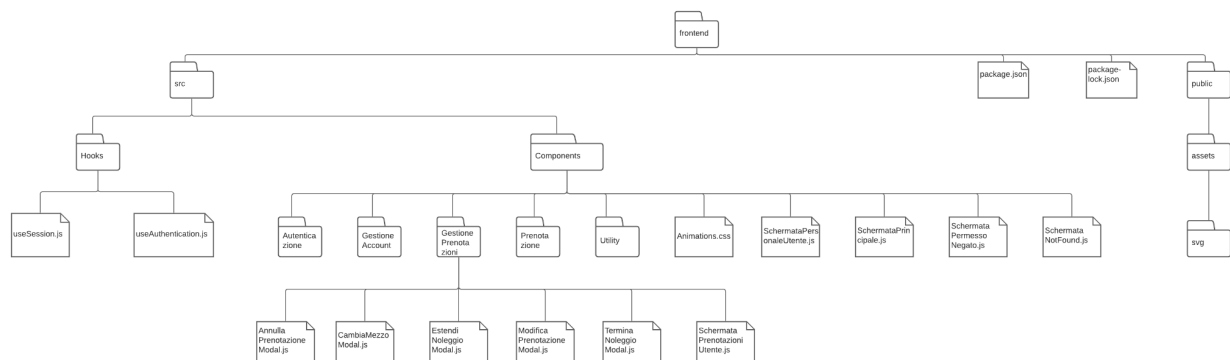
Durante lo sviluppo del software si è scelto di dividere i componenti in packages, per avere un'organizzazione migliore e fedele alla documentazione presentata.

Uniformemente all'architettura scelta, si è deciso di suddividere l'applicativo in due package principale, cioè frontend e backend.

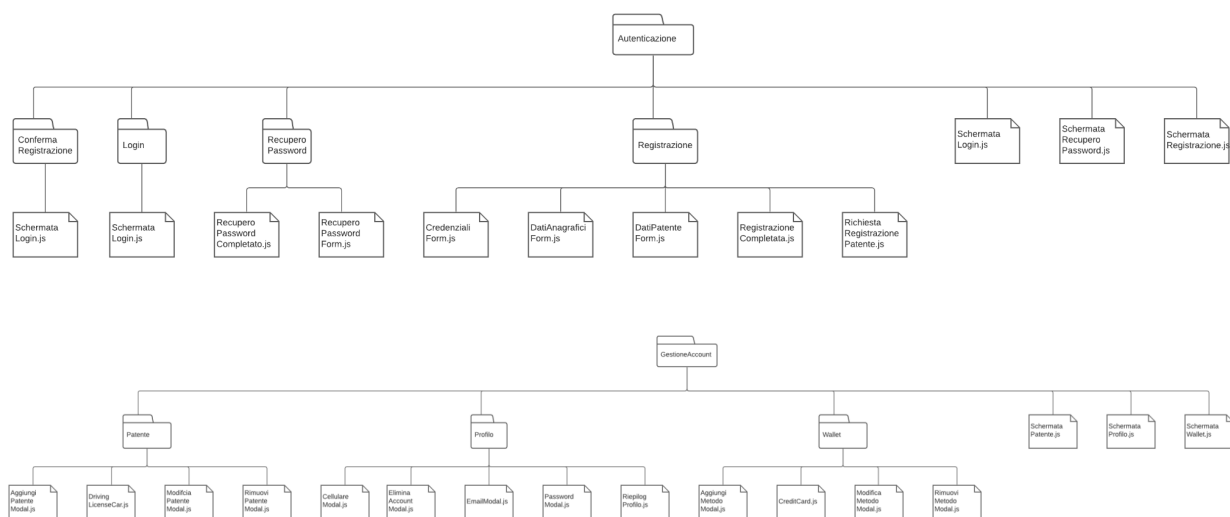
Di seguito è riportato l'elenco dei package.

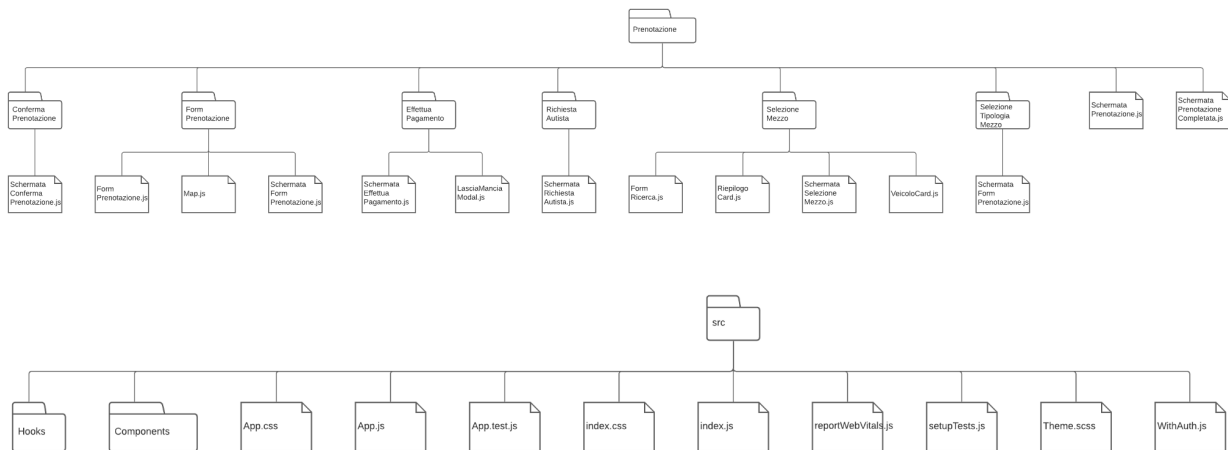
Autenticazione	Contiene tutto ciò che serve per effettuare l'autenticazione da parte degli utenti. Essi possono effettuare la registrazione del proprio account, effettuare il login recuperare la password del suddetto account.
GestioneAccount	Contiene tutto ciò che serve per la gestione dell'account e la modifica dei dati personali.
GestionePrenotazioni	Contiene tutto ciò che serve per la gestione della prenotazione da parte degli utenti Il cliente può richiedere il noleggio di un mezzo, può modificare quest'ultimo, può anche annullarlo, ed inoltre può visualizzare la schermata della prenotazione. L'amministratore può fare tutto ciò che possono fare il cliente e l'autista.
Prenotazione	Contiene tutto ciò che serve per la prenotazione di un mezzo.
Impiegato	Contiene tutto ciò che riguardano le funzionalità degli Impiegati, tra cui la gestione Utenti, gestione Mezzi, Gestione impiegati e registrazione impiegati.

2.1. Organizzazione directory (front-end)



Per quanto riguarda le directory Autenticazione, GestioneAccount, Prenotazione e src, si è scelto di rappresentarli separatamente, per evitare di creare uno schema troppo confusionario.





2.2. Organizzazione directory (back-end)

