



UNIVERSITÀ
DEGLI STUDI
DI PALERMO

Requirements Analysis Document
Ingegneria Informatica – AA 2020/2021

A cura di:
Domenico Mereu
Emanuele Seminara
Flavio Giuseppe Amato
Samuele Marino

Indice

1	Introduzione.....	3
1.1	Compromessi e scelte nella progettazione	3
2	Packages.....	5
2.1	Amministratore.....	5
2.2	Cliente.....	6
2.3	Autista	7
2.4	Parcheggiatore	7
2.5	Ricerca	8

1 Introduzione

1.1 Compromessi e scelte nella progettazione

Per la realizzazione del sistema si è scelto un approccio modulare diviso su più livelli, in maniera tale da rendere il sistema scalabile e semplice da implementare. Il sistema segue il pattern architetturale Three-Tier che permette di implementare un'architettura multi-livello.

I livelli presenti nel sito si possono riassumere in:

- React, per quanto riguarda la creazione delle interfacce utente.
- AXIOS, per permettere al front-end di dialogare con il back-end, e viceversa.
- Express, per implementare le funzionalità del sistema tramite controller.
- SQLite, per la creazione e manipolazione dei dati nel database

Per quanto concerne l'interfaccia grafica si è scelto di utilizzare il framework React.

L'utilizzo di componenti rappresenta un notevole vantaggio in quanto permette estrema riusabilità del codice ed in generale una maggiore leggibilità in fase di scrittura e debugging del codice.

Per la realizzazione del front-end sono stati utilizzati i seguenti moduli:

- @date-io/date-fns: 1.3.13, Questo progetto fa parte di date-io monorepo e contiene la unified interface di date-fns.
- @material-ui/core: 4.11.4, Componente React per uno sviluppo web più rapido e semplice.
- @material-ui/lab: 4.0.0-alpha.58, Questo pacchetto ospita i componenti dell'incubatrice che non sono ancora pronti per essere trasferiti al core.
- @material-ui/pickers: 3.3.10, selettori di data e ora accessibili e personalizzabili.
- availity-reactstrap-validation: 2.7.1, componenti di validazione React facili da usare compatibili per reactstrap.
- axios: 0.21.1, client HTTP promise-based che permette di effettuare richieste HTTP come GET, POST PUT e DELETE.
- bootstrap: 4.6.0, framework di front-end elegante, intuitivo e potente per uno sviluppo web più rapido e semplice.
- date-fns: 2.22.1, fornisce il set di strumenti più completo, ma semplice e coerente per manipolare le date JavaScript in un browser.
- jquery: 3.6.0, rende cose come l'attraversamento e la manipolazione di documenti HTML, la gestione degli eventi, l'animazione e Ajax molto più semplici con un'API facile da usare che funziona su una moltitudine di browser.
- jspdf: 2.3.1, serve a creare PDF tramite tecnologia Web Client-Side.
- moment: 2.29.1, modulo utilizzato per la gestione delle date.
- react: 17.0.2, è una libreria JavaScript per la creazione di interfacce utente.
- react-dom: 17.0.2, questo pacchetto funge da punto di ingresso al DOM e ai renderer server per React. È pensato per essere abbinato al pacchetto React generico, che viene spedito come react to npm.
- react-router-dom: 5.2.0, modulo che permette di effettuare le operazioni di routing così da implementare il multipage.
- react-scripts: 4.0.3, questo pacchetto include script e configurazione utilizzati dall'app Create React.

- reactstrap: 8.9.0, modulo che permette la creazione di componenti esteticamente curati, in maniera semplice.
- web-vitals: 1.1.2, è una libreria per misurare tutte le metriche Web Vitals sugli utenti reali, in un modo che corrisponda accuratamente a come vengono misurati da Chrome e segnalati ad altri strumenti di Google.

Per quanto riguarda il lato server, si è scelto di usare NodeJS, che permette la gestione degli eventi (asincroni) e dei metodi mediante la scrittura di codice JavaScript.

Il motivo principale di questa scelta è legato alla capacità di NodeJS di gestire e risolvere eventi asincroni. Logica conseguenza di ciò è la massimizzazione del throughput e la minimizzazione della latenza.

Nello specifico, è stato utilizzato Express.js come framework principale per la gestione del backend, che rappresenta anche il middleware del sistema.

Express.js permette di gestire il routing lato back-end, rispondere alle richieste HTTP effettuate dal front-end, ed interfacciarsi al database SQLite.

Per la realizzazione del back-end sono stati utilizzati i seguenti moduli:

- bcrypt: 5.0.1, una libreria che permette di eseguire l'hashing delle password.
- dotenv: 10.0.0, è un modulo che carica le variabili di ambiente da un file .env in process.env.
- express: 4.17.1, framework web veloce, disinvolto e minimalista per node.
- express-session: 1.17.2, libreria per la gestione delle sessioni.
- express-validator: 6.12.0, libreria per la validazione dei dati.
- generate-password: 1.6.0, libreria per la generazione automatica delle password.
- google-distance: 1.0.1, API google per il calcolo della distanza tra due vie.
- morgan: 1.10.0, Middleware del registratore di richieste HTTP per node.js
- nodemailer: 6.6.2, libreria per l'invio delle email
- passport: 0.4.1, è un middleware di autenticazione compatibile con Express per Node.js.
- passport-local: 1.0.0, strategia per passport per l'autenticazione con nome utente e password.
- sqlite3: 5.0.2, gestore di database SQLite con chiamate asincrone e non bloccanti per Node.js.

Per quanto riguarda l'implementazione dell'App Mobile si è scelto di utilizzare Android Studio, un ambiente di sviluppo per la piattaforma Android.

Nello specifico è stata implementata come WebView: un componente di sistema Android che permette alle app Android di mostrare contenuti web.

Il motivo di questa scelta è dovuto, sicuramente, ad una facile e veloce implementazione, ma soprattutto al fatto che questo componente è preinstallato su qualsiasi dispositivo Android e quindi farà sempre visualizzare l'App, senza alcun problema di visualizzazione né compatibilità

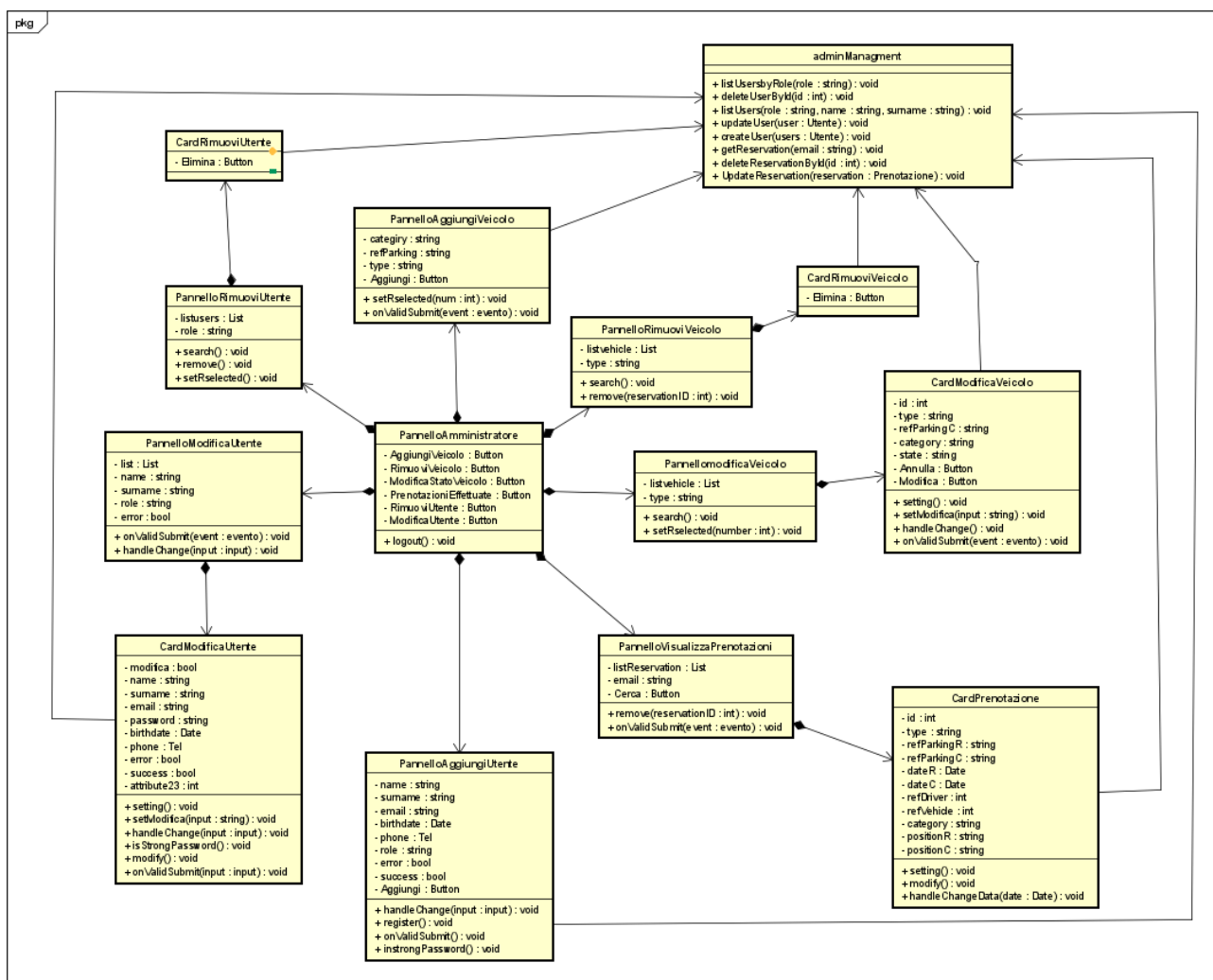
2 Packages

Per un'organizzazione ordinata e semplice durante lo sviluppo del software, si è scelto di dividere le componenti in diversi packages.

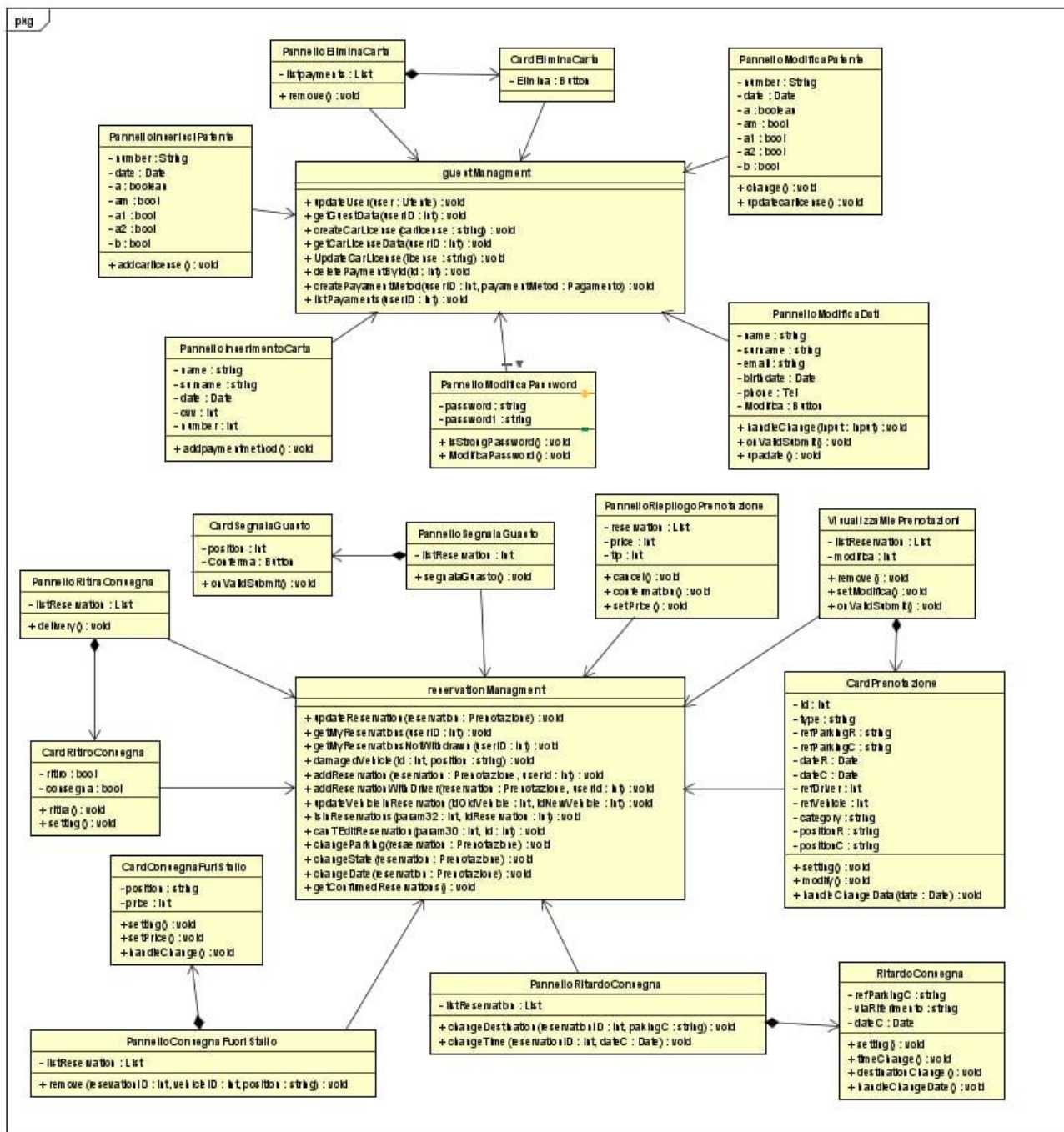
Il software è stato suddiviso in 2 parti: ciò che riguarda l'interfaccia con il client (front-end) e ciò che riguarda i meccanismi di esecuzione del sistema (back-end).

A seguire, l'elenco dei package:

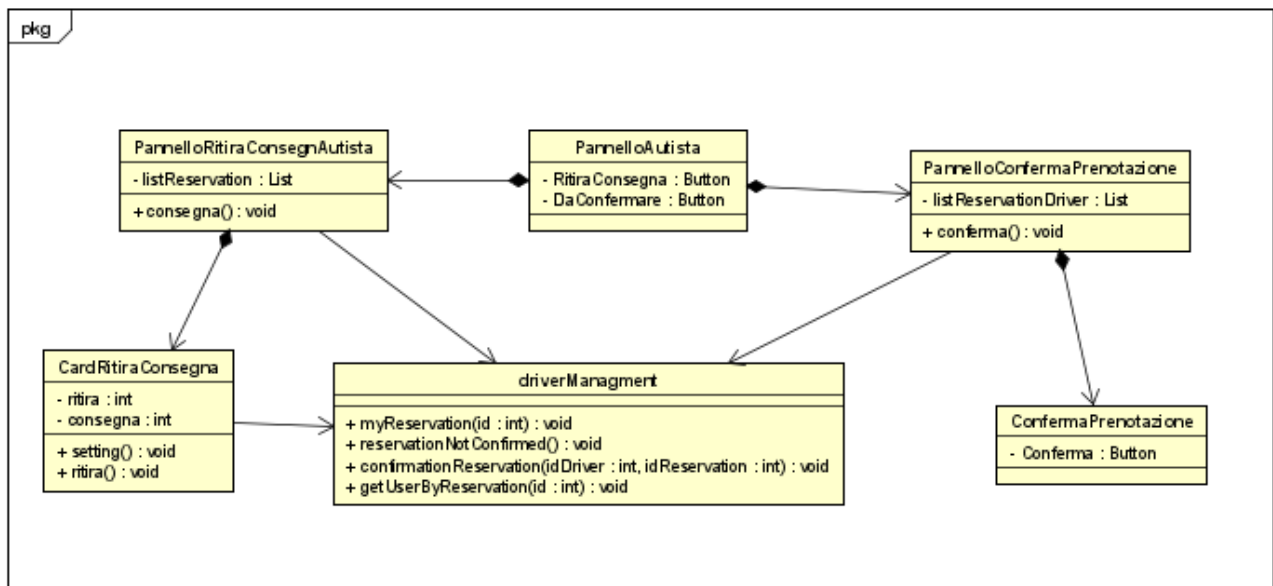
2.1 Amministratore



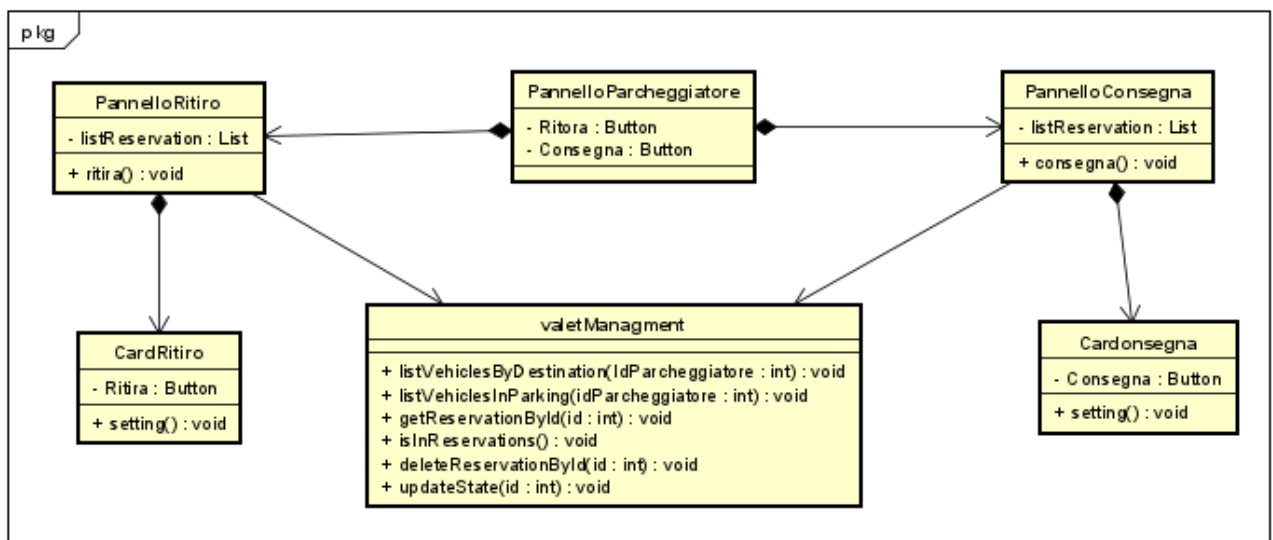
2.2 Cliente



2.3 Autista



2.4 Parcheggiatore



2.5 Ricerca

