

Object Design Document

Progetto Moovy

A chi è rivolto:

Analisti e sviluppatori.

Membri del gruppo:

Alessia Bevilacqua, Marena Jestin, Irene Pellicane, Laura Guaggenti

Sommario

1. Introduzione

- 1.1. Compromessi e scelte nella progettazione degli oggetti

2. Package

- 2.1. Package Models
- 2.2. Package Controllers
 - 2.2.1. Business Controllers
 - 2.2.2. Routers
- 2.3. Package Middleware
- 2.4. Package Utilities
- 2.5. Package Views

3. Interfacce delle classi

1. Introduzione

1.1. Compromessi e scelte nella progettazione degli oggetti

La realizzazione del sistema si è basata sull'utilizzo di diversi framework.

Per quanto riguarda la realizzazione delle interfacce grafiche responsive, si è scelto di utilizzare il framework Bootstrap. Abbiamo inoltre utilizzato EJS, un template engine per la creazione di pagine web interattive.

Invece per la gestione lato server è stato utilizzato NodeJS, che permette la gestione degli eventi asincroni e dei metodi mediante il linguaggio javascript.

In particolare la gestione del back-end è stata affidata al framework ExpressJS, che permette di gestire il routing, rispondere alle richieste HTTP effettuate dal front-end ed interfacciarsi al database.

Infine, per la realizzazione e manipolazione del database è stato utilizzato MySQL.

In particolare sono stati utilizzati i seguenti moduli:

- ❖ mysql, per la gestione del database;
- ❖ md5, che permette di eseguire l'hashing delle password;
- ❖ express-session, per la gestione delle sessioni;
- ❖ express-mysql-session, per memorizzare i dati delle sessioni su un database mysql;
- ❖ nodemailer, per l'invio delle mail;
- ❖ @turf/distance, per il calcolo della distanza tra due punti su una mappa
- ❖ Mapbox GL, per la creazione di mappe vettoriali interattive e personalizzabili per il web.

2. Package

Dipendenze:

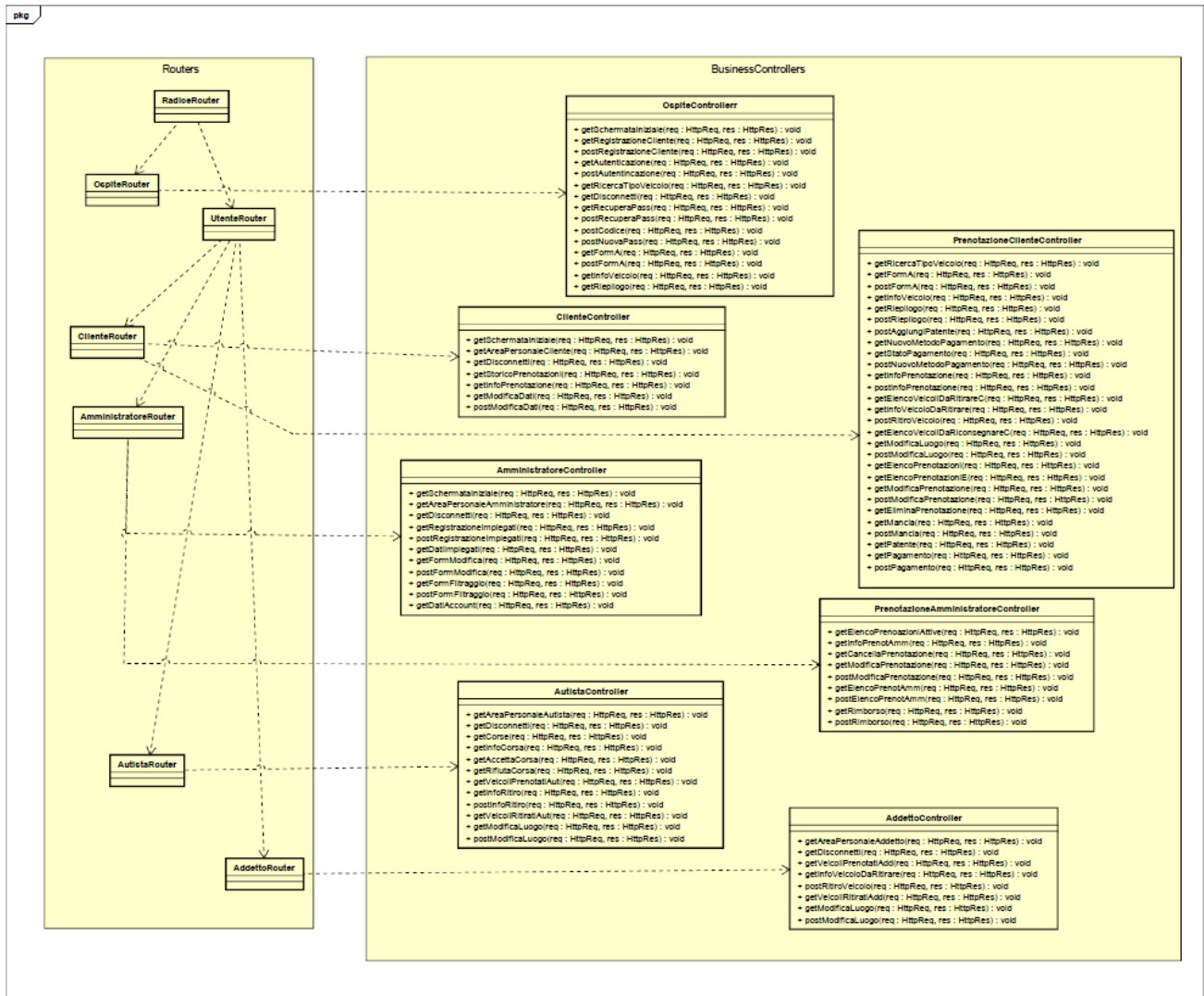
- I Controller dipendono da Model e dalle Utilities
- I Model dipendono dalle Utilities
- I Router dipendono dai Controller e dalle View

2.1. Package Model

pkg	
	<pre> accountModel + getAccount(dbPool :dbPool, utentid :int) :Object + getImpiegato(dbPool :dbPool, utentid :int) :Object + getAccountImpiegato(dbPool :dbPool) :Object + getAccountFinanziario(dbPool :dbPool, ruolo :varchar) :Object + login(dbPool :dbPool, email :varchar, password :varchar) :void + recuperoPassword(dbPool :dbPool, utentid :int, nuovaPassword :varchar) :void + registrazioneCliente(dbPool :dbPool, nome :varchar, cognome :varchar, email :varchar, data_d_nascita :Date, num_telefono :varchar, cnaPassword :varchar, codice_patente :varchar, scadenza_patente :Date, tipo_a :int, tipo_b :int, tipo_am :int, tipo_a1 :int, tipo_a2 :int, numero_carta :varchar, nome_intestatario :varchar, cognome_intestatario :varchar, scadenza_carta :varchar, cvv :int) :void + registrazioneImpiegato(dbPool :dbPool, ruolo :varchar, nome :varchar, cognome :varchar, email :varchar, data_d_nascita :Date, num_telefono :varchar, cnaPassword :varchar, codice_patente :varchar, scadenza_patente :Date, tipo_a :int, tipo_b :int, tipo_am :int, tipo_a1 :int, tipo_a2 :int) :void + aggngiPatenteCidbPool :dbPool, ref_account :int, codice_patente :int, scadenza_patente :Date, tipo_a :boolean, tipo_b :boolean, tipo_am :boolean, tipo_a1 :boolean, tipo_a2 :boolean) :void + modificaCidCarnet(dbPool :dbPool, utentid :int, num_telefono :varchar, ruolo :varchar, codice_patente :varchar, scadenza_patente :Date, tipo_a :int, tipo_b :int, tipo_am :int, tipo_a1 :int, tipo_a2 :int, numero_carta :varchar, scadenza_carta :varchar, nome_intestatario :varchar, cognome_intestatario :varchar, cvv :int) :void + modificaCidImpiegato(dbPool :dbPool, utentid :int, nome :varchar, cognome :varchar, email :varchar, data_d_nascita :Date, num_telefono :varchar, nuovaCnaPassword :varchar, codice_patente :varchar, scadenza_patente :Date) :void + eliminaAccount(dbPool :dbPool, utentid :int) :void + getParcheggioAutto(dbPool :dbPool, id_Auto :int) :Object + aggiornaMontoDiPagamento(dbPool :dbPool, numero_carta :int, ref_account :int, nome_intestatario :varchar, cognome_intestatario :varchar, scadenza_carta :varchar, cvv :int) :void </pre>
	<pre> prenotazioneModel + getPrenotazione(dbPool :dbPool, prenotazioneId :int) :Object + aggiungiprenotazione(dbPool :dbPool, ref_cliente :int, ref_autista :int, tipo_veicolo :varchar, ref_veicolo :int, marcia :float, data_ritiro :Date, data_riconsegna :Date, luogo_ritiro :varchar, luogo_riconsegna :varchar, prezzo_stimato :float) :void + confermaPrenotazione(dbPool :dbPool, id_prenotazione :int, ref_carta :int) :void + getConsegna(dbPool :dbPool, ref_autista :int) :Object + accettaConsegna(dbPool :dbPool, id_prenotazione :int, ref_autista :int) :void + verificaPrenotazione(dbPool :dbPool, id_prenotazione :int, stato_prenotazione :varchar) :void + annullaPrenotazione(dbPool :dbPool, id_prenotazione :int) :void + getConsegnePrenotazione(dbPool :dbPool, ref_cliente :int) :Object + modificaPrenotazione(dbPool :dbPool, id_prenotazione :int, data_riconsegna :Date, luogo_riconsegna :varchar) :void + setPrezzoFinali(dbPool :dbPool, id_prenotazione :int, prezzo_finale :float) :void + getPrenotazioneDelVeicolo(dbPool :dbPool, ref_veicolo :int) :Object + getPrenotazioneAttivaCidbPool :dbPool, ref_cliente :int) :Object + getPrenotazioneAttivaAutto(dbPool :dbPool) :Object + cercaVeicolo(dbPool :dbPool, ve :int) :Object + getVeicolo(dbPool :dbPool, id_veicolo :int) :Object + getRitiraprenotazione(dbPool :dbPool) :Object + getVeicoloDaRitirareCidbPool :dbPool, ref_cliente :int) :Object + getVeicoloDaRitirareAutto(dbPool :dbPool, ref_autista :int) :Object + getVeicoloDaRiconsegnareCidbPool :dbPool, ref_cliente :int) :Object + getVeicoloDaRiconsegnareAutto(dbPool :dbPool, ref_autista :int) :Object + getVeicoloDaRiconsegnareAutto(dbPool :dbPool) :Object + modificaLuogoRiconsegna(dbPool :dbPool, id_prenotazione :int, luogo_riconsegna :int) :void + verificaRitiro(dbPool :dbPool, id_prenotazione :int, luogo_riconsegna :varchar, motivo_ritiro :varchar) :void + riconsegnaVeicolo(dbPool :dbPool, id_prenotazione :int, stato_prenotazione :varchar, id_veicolo :int, luogo_riconsegna :varchar, prezzo_finale :float) :void + getPrezzoVeicolo(dbPool :int, id_veicolo :int) :Object + getAutistaLibero(dbPool :dbPool, data_ritiro :Date, data_riconsegna :Date) :Object </pre>

Nome	Descrizione
accountModel.js	Oggetto che permette di gestire i dati delle tabelle account, carte_di_credito e patenti presenti nel Database
prenotazioneModel.js	Oggetto che permette di gestire i dati delle tabelle prenotazioni, veicoli e parcheggi presenti nel Database

2.2. Package Controller



2.2.1. Business Controller

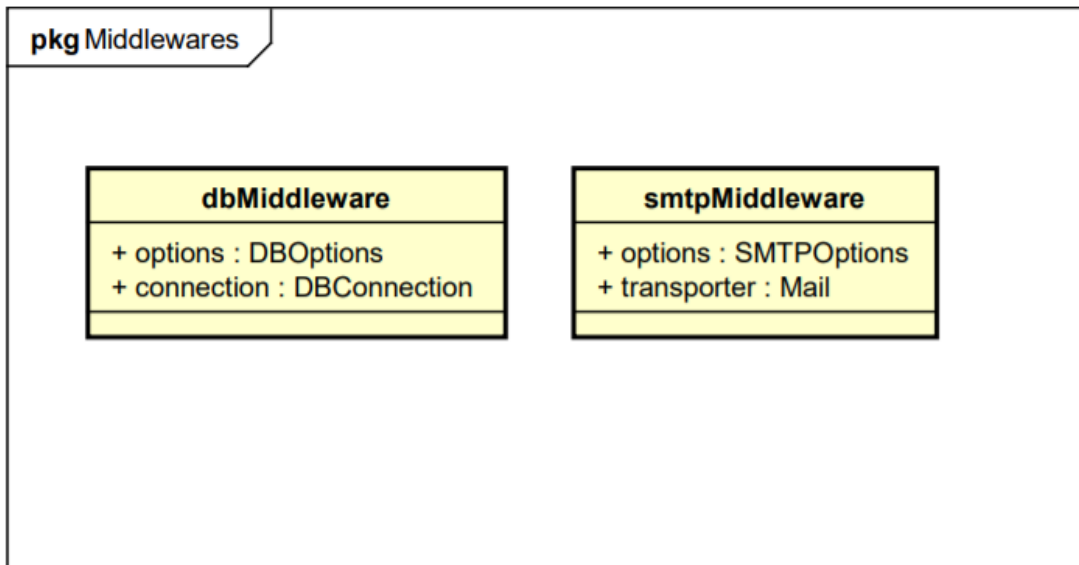
Nome	Descrizione
OspiteController.js	Controller che si occupa di gestire le operazioni di mostra schermata iniziale, registrazione cliente, autenticazione e recupero password.
cliente/ClienteController.js	Controller che si occupa di gestire le operazioni di mostra schermata iniziale, mostra Area personale, modifica dati personali e mostra storico prenotazioni.
cliente/PrenotazioneClienteController.js	Controller che si occupa di gestire le operazioni di ricerca e prenota veicolo, inserimento di nuova carta di credito e di patente, aggiunge mancia per l'autista (se è richiesto), pagamento, mostra veicoli da ritirare e da riconsegnare, modifica luogo di riconsegna, modifica prenotazione e

	elimina prenotazione.
amministratore/AmministratoreController.js	Controller che si occupa di gestire le operazioni di mostra Area Personale, registrazione impiegati, modifica dati impiegati e elimina Account degli utenti .
amministratore/PrenotazioneAmministratoreController.js	Controller che si occupa di gestire le operazioni di mostra elenco prenotazioni attive, modifica prenotazione , elimina prenotazione e rimborso.
autista/AutistaController.js	Controller che si occupa di gestire le operazioni di mostra Area Personale, mostra corse da accettare e accettate, mostra veicoli da ritirare e riconsegnare, modifica luogo di riconsegna.
autista/AddettoController.js	Controller che si occupa di gestire le operazioni di mostra Area Personale, mostra veicoli da ritirare e riconsegnare, modifica luogo di riconsegna.

2.2.2. Routers

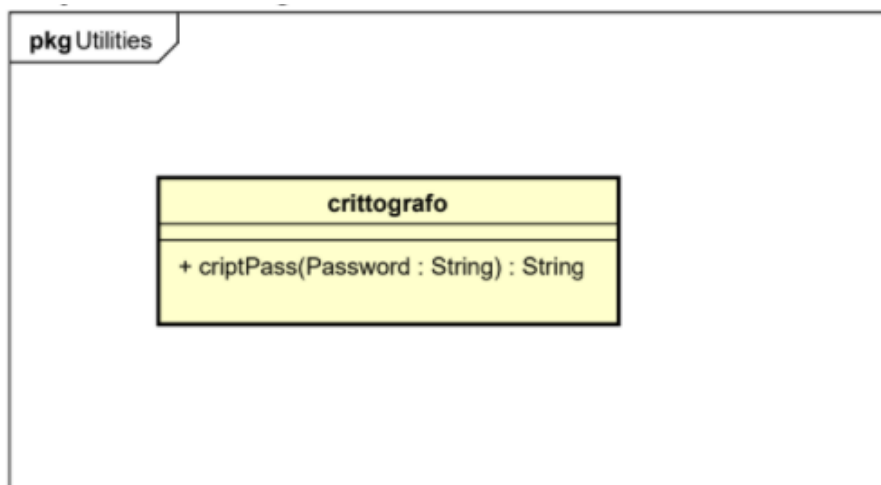
Nome	Descrizione
RadiceRouter.js	Router che si occupa di reindirizzare verso i router OspiteController e UtenteController, in base al fatto che l'utente sia autenticato o no.
OspiteRouter.js	Router che si occupa di gestire le routes per l'autenticazione e registrazione del cliente e di reindirizzare la logica di business all'OspiteController.
UtenteRouter.js	Router che si occupa di reindirizzare verso i router ClienteRouter, AmministratoreRouter, AutistaRouter e AddettoRouter, in base al ruolo dell'utente autenticato.
ClienteRouter.js	Router che si occupa di reindirizzare la logica business ai controller: ClienteController e PrenotazioneClienteController.
AmministratoreRouter.js	Router che si occupa di reindirizzare la logica business ai controller: AmministratoreController e PrenotazioneAmministratoreController.
AutistaRouter.js	Router che si occupa di reindirizzare la logica business al controller AutistaController.
AddettoRouter.js	Router che si occupa di reindirizzare la logica business al controller AddettoController.

2.3. Package Middleware



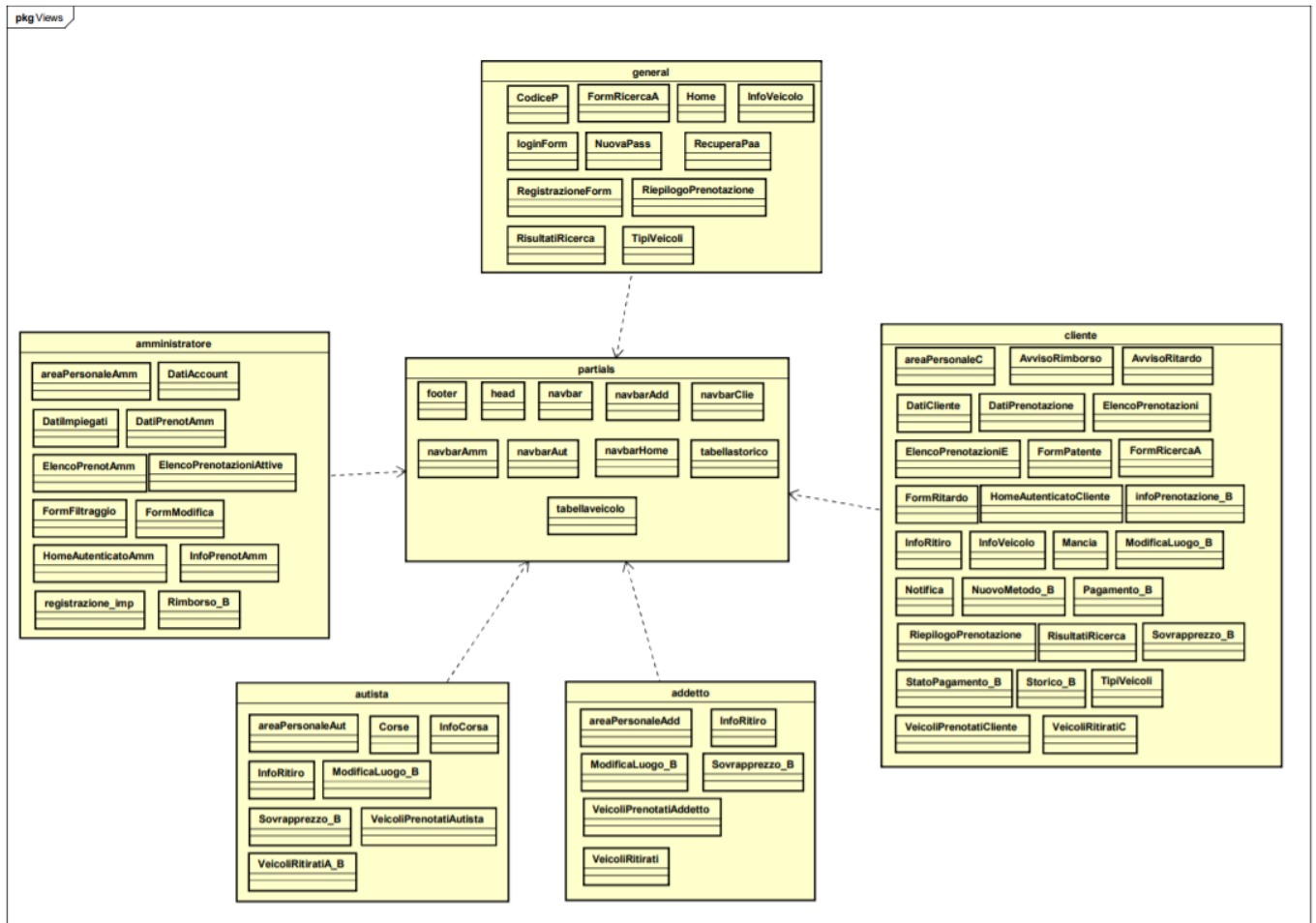
Nome	Descrizione
db.js	Middleware per la connessione al database.
smtp.js	Middleware per il servizio mail.

2.4. Package Utilities



Nome	Descrizione
crittografo.js	Oggetto che si occupa di crittografare la password.

2.5. Package Views



Nome	Descrizione
general/Home.ejs	Oggetto che descrive la vista della pagina da renderizzare "Schermata Iniziale", prima dell'autenticazione.
general/login.ejs	Oggetto che descrive la vista della pagina da renderizzare "login".
general/RecuperaPass.ejs	Oggetto che descrive la vista della pagina da renderizzare "Recupera Password".
general/CodiceP.ejs	Oggetto che descrive la vista della pagina da renderizzare "Codice".
general/NuovaPass.ejs	Oggetto che descrive la vista della pagina da renderizzare "aggiornare la password".

general/RegistrazioneForm.ejs	Oggetto che descrive la vista della pagina da renderizzare "Registrazione Cliente".
cliente/HomeAutenticatoCliente.ejs	Oggetto che descrive la vista della pagina da renderizzare "Schermata Principale dopo l'autenticazione".
cliente/areaPersonaleC.ejs	Oggetto che descrive la vista della pagina da renderizzare "Area Personale Cliente".
cliente/TipiVeicoli.ejs	Oggetto che descrive la vista della pagina da renderizzare "Tipi Veicoli".
cliente/FormRicercaA.ejs	Oggetto che descrive la vista della pagina da renderizzare "Form Ricerca veicoli".
cliente/RisultatiRicerca.ejs	Oggetto che descrive la vista della pagina da renderizzare "Risultati Ricerca".
cliente/InfoVeicolo.ejs	Oggetto che descrive la vista della pagina da renderizzare "Descrizione veicolo".
cliente/RiepilogoPrenotazione.ejs	Oggetto che descrive la vista della pagina da renderizzare "Riepilogo".
cliente/Mancia.ejs	Oggetto che descrive la vista della pagina da renderizzare "Aggiungere Mancia per l'autista".
cliente/FormPatente.ejs	Oggetto che descrive la vista della pagina da renderizzare "Aggiungere patente".
cliente/Pagamento_B.ejs	Oggetto che descrive la vista della pagina da renderizzare "Pagamento".
cliente/NuovoMetodo_B.ejs	Oggetto che descrive la vista della pagina da renderizzare "Aggiungere Nuova carta".
cliente/StatoPagamento_B.ejs	Oggetto che descrive la vista della pagina da renderizzare "Stato Pagamento".
cliente/DatiCliente.ejs	Oggetto che descrive la vista della pagina da renderizzare "Modifica Dati Cliente".
cliente/ElencoPrenotazioniE.ejs	Oggetto che descrive la vista della pagina da renderizzare "Elenco prenotazioni da Eliminare".
cliente/AvvisoRimborso.ejs	Oggetto che descrive la vista della pagina da renderizzare "Elimina Prenotazione cliente".
cliente/ElencoPrenotazioni.ejs	Oggetto che descrive la vista della pagina da renderizzare "Elenco prenotazioni cliente".
cliente/DatiPrenotazione.ejs	Oggetto che descrive la vista della pagina

	da renderizzare "Dati Prenotazione".
cliente/Storico_B.ejs	Oggetto che descrive la vista della pagina da renderizzare "Elenco prenotazioni".
cliente/InfoPrenotazioe_B.ejs	Oggetto che descrive la vista della pagina da renderizzare "Info Prenotazionei".
cliente/VeicoliPrenotatiCliente.ejs	Oggetto che descrive la vista della pagina da renderizzare "Elenco veicoli prenotati".
cliente/InfoRitiro.ejs	Oggetto che descrive la vista della pagina da renderizzare "Ritiro veicolo".
cliente/VeicoliRitiratiC.ejs	Oggetto che descrive la vista della pagina da renderizzare "Elenco veicoli ritirati".
cliente/ModificaLuogo_B.ejs	Oggetto che descrive la vista della pagina da renderizzare "Modifica Luogo Riconsegna".
cliente/Sovrapprezzo_B.ejs	Oggetto che descrive la vista della pagina da renderizzare "Sovrapprezzo".
amministratore/HomeAutenticatoAmm.ejs	Oggetto che descrive la vista della pagina da renderizzare "Schermata Principale dopo l'autenticazione".
amministratore/areaPersonaleAmm.ejs	Oggetto che descrive la vista della pagina da renderizzare "Area Personale Amministratore".
amministratore/RegistrazioneImp.ejs	Oggetto che descrive la vista della pagina da renderizzare "Registrazione Impiegati".
amministratore/DatiImpiegati.ejs	Oggetto che descrive la vista della pagina da renderizzare "Elenco degli impiegati".
amministratore/FormModifica.ejs	Oggetto che descrive la vista della pagina da renderizzare "Modifica Dati Impiegato".
amministratore/FormFiltraggio.ejs	Oggetto che descrive la vista della pagina da renderizzare "Form Filtraggio", per eliminazione dell'account.
amministratore/DatiAccount.ejs	Oggetto che descrive la vista della pagina da renderizzare "Elimina Account".
amministratore/ElencoPrenotazioniAttive.ejs	Oggetto che descrive la vista della pagina da renderizzare "Elenco Prenotazioni Attive".
amministratore/InfoPrenotAmm.ejs	Oggetto che descrive la vista della pagina da renderizzare "Dati Prenotazione".
amministratore/DatiPrenotAmm.ejs	Oggetto che descrive la vista della pagina da renderizzare "Modifica Prenotazione".

amministratore/ElencoPrenotAmm_B.ejs	Oggetto che descrive la vista della pagina da renderizzare "Elenco Prenotazioni Attive".
amministratore/Rimborso_B.ejs	Oggetto che descrive la vista della pagina da renderizzare "Rimborso".
autista/areaPersonaleAut.ejs	Oggetto che descrive la vista della pagina da renderizzare "Area Personale Autista".
autista/Corse.ejs	Oggetto che descrive la vista della pagina da renderizzare "gestione corse".
autista/InfoCorsa.ejs	Oggetto che descrive la vista della pagina da renderizzare "Corsa da accettare / rifiutare".
autista/VeicoliPrenotatiAutista.ejs	Oggetto che descrive la vista della pagina da renderizzare "Elenco veicoli prenotati".
autista/InfoRitiro.ejs	Oggetto che descrive la vista della pagina da renderizzare "Ritiro veicolo".
autista/VeicoliRitiratiAut.ejs	Oggetto che descrive la vista della pagina da renderizzare "Elenco veicoli ritirati".
autista/ModificaLuogo_B.ejs	Oggetto che descrive la vista della pagina da renderizzare "Modifica Luogo Riconsegna".
addetto/areaPersonaleAdd.ejs	Oggetto che descrive la vista della pagina da renderizzare "Area Personale Addetto".
addetto/VeicoliPrenotatiAddetto.ejs	Oggetto che descrive la vista della pagina da renderizzare "Elenco veicoli prenotati".
addetto/InfoRitiro.ejs	Oggetto che descrive la vista della pagina da renderizzare "Ritiro veicolo".
addetto/VeicoliRitiratiAdd.ejs	Oggetto che descrive la vista della pagina da renderizzare "Elenco veicoli ritirati".
addetto/ModificaLuogo_B.ejs	Oggetto che descrive la vista della pagina da renderizzare "Modifica Luogo Riconsegna".
partials/head.ejs	Microcomponente da importare in altre view contenente i metadati del tag head.
partials/navbar.ejs	Microcomponente da importare in altre view contenente navbar(home, veicoli,login, recupera password e registrati).
partials/navbarClie.ejs	Microcomponente da importare in altre view contenente navbar(HomeAutenticatoCliente, veicoli,Area personale e logout).

partials/navbarAmm.ejs	Microcomponente da importare in altre view contenente navbar(HomeAutenticatoAmm, Area Personale e logout).
partials/navbarAdd.ejs	Microcomponente da importare in altre view contenente navbar(Area Personale e logout).
partials/navbarAut.ejs	Microcomponente da importare in altre view contenente navbar(Area Personale e logout).
partials/navbarHome.ejs	Microcomponente da importare in altre view contenente navbar(Home).
partials/footer.ejs	Microcomponente da importare in altre view contenente il footer.
partials/tabellastorico.ejs	Microcomponente da importare in altre view contenente il template della scheda " Mostra Storico Prenotazioni "
partials/tabellaveicolo.ejs	Microcomponente da importare in altre view contenente il template della scheda "Riconsegna Veicolo"

3. Interfacce delle classi

