

System 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. Obiettivo generale
 - 1.2. Obiettivi di progettazione
 - 1.3. Acronimi
 - 1.4. Riferimenti
- 2. Architettura software attuale**
- 3. Architettura software proposta**
 - 3.1. Overview
 - 3.2. Decomposizione in sottosistemi
 - 3.2.1. Models
 - 3.2.2. Controllers
 - 3.2.3. Views
 - 3.3. Mappatura hardware e software
 - 3.4. Gestione dati persistenti
 - 3.4.1. Descrizione delle tabelle
 - 3.4.1.1. Account
 - 3.4.1.2. Prenotazioni
 - 3.4.1.3. Patenti
 - 3.4.1.4. Carte di credito
 - 3.4.1.5. Veicoli
 - 3.4.1.6. Parcheggio
 - 3.5. Controllo degli accessi e sicurezza
 - 3.6. Boundary conditions

1. Introduzione

1.1. Obiettivo Generale

L'obiettivo del sistema è quello di offrire un servizio di noleggio veicoli e, in particolare, la prenotazione di un mezzo da parte dei clienti e la relativa gestione anche da parte degli impiegati.

1.2. Obiettivi di progettazione

- ❖ Il Sistema dovrà consentire all'Utente di tornare alla schermata precedente o alla schermata principale in qualsiasi momento.
- ❖ Il Sistema dovrà impedire l'inserimento di dati non validi.
- ❖ Il DBMS dovrà essere sempre attivo per fare in modo che le informazioni siano accessibili dall'utente in qualsiasi momento.
- ❖

1.3. Acronimi

DBMS: Database Management System

UML: Unified Modeling Language

SQL: Structured Query Language

MVC: Model View Controller

1.4. Riferimenti

- ❖ Libro di testo: "Object-Oriented Software Engineering: Using UML, Patterns and Java. Third edition." - B. Bruegge, A. H. Dutoit.
- ❖ Libro di testo: "Ingegneria del software" 8/Ed. 2007 - Ian Sommerville
- ❖ Libro di testo "Basi di Dati" 4th edizione Paolo Atzeni, Stefano Ceri, Piero Fraternali, Stefano Paraboschi, Riccardo Torlone
- ❖ Materiale didattico fornito dalla professoressa Seidita Valeria

2. Architettura software attuale

Si suppone che non esista alcun sistema informatico precedente.

3. Architettura software proposta

3.1. Overview

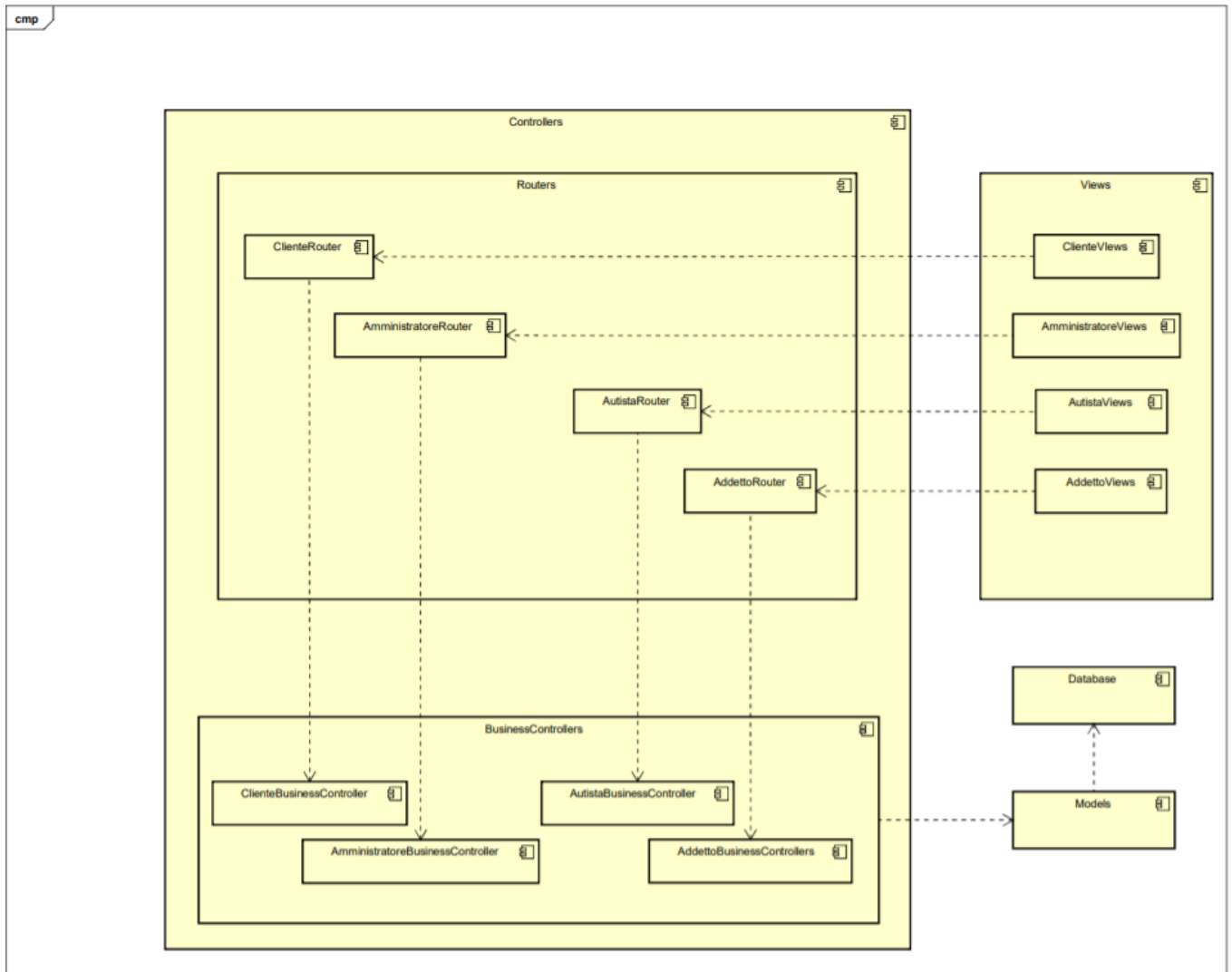
L'architettura di questo sistema si basa sul pattern Model-View-Controller (MVC).

Il modello MVC è strutturato in tre tipi di componenti:

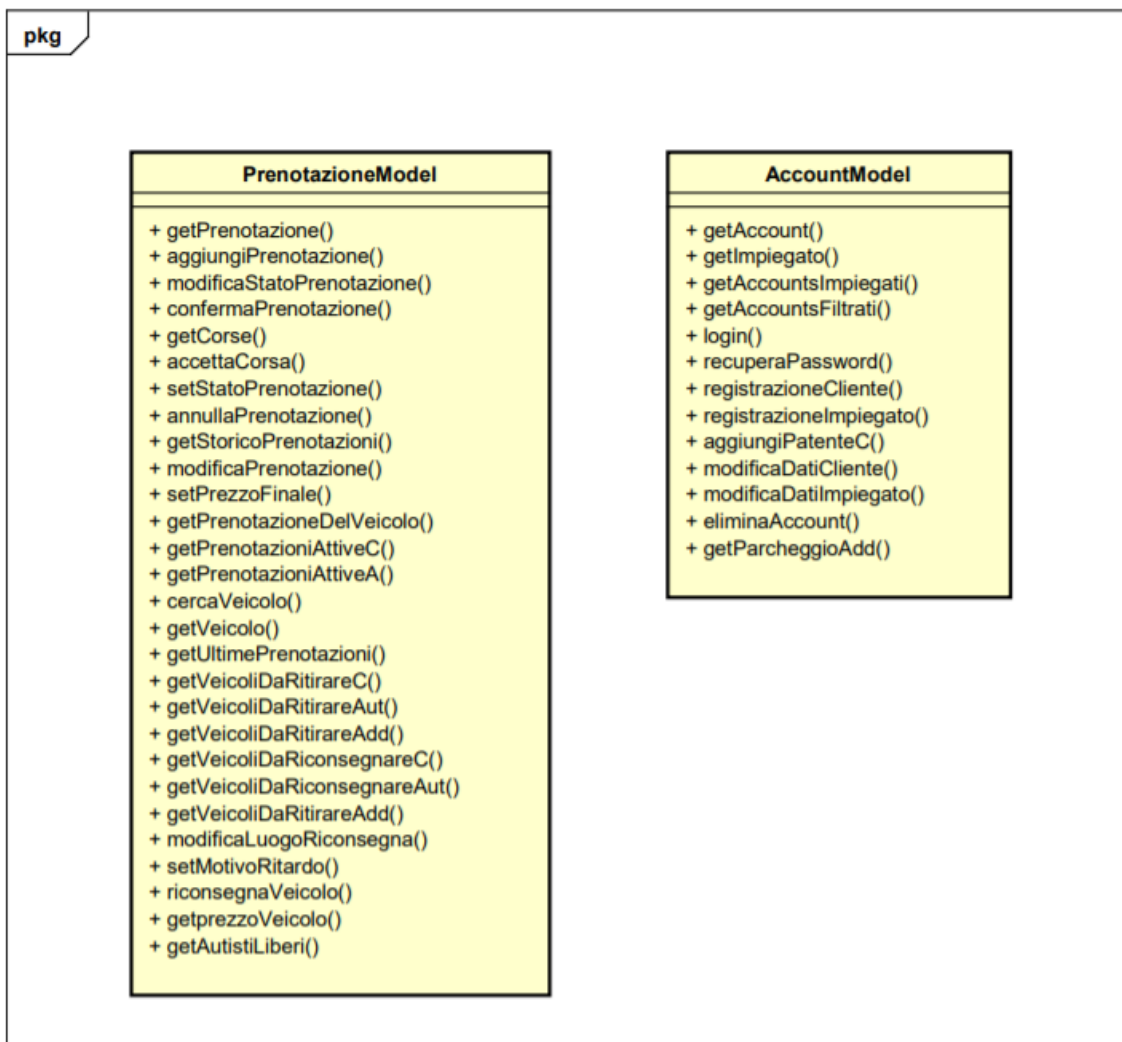
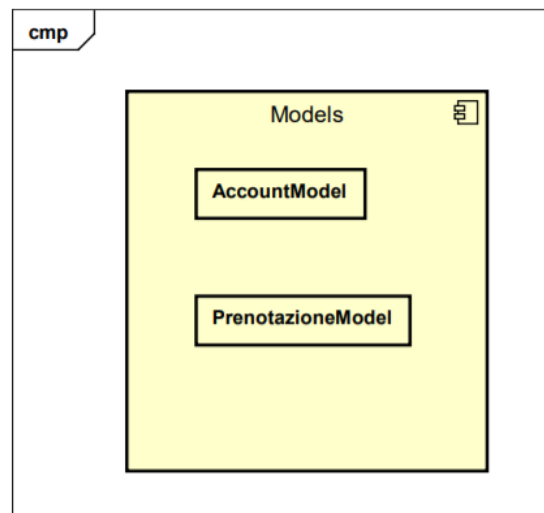
- ❖ Model, che contiene i metodi di accesso ai dati
- ❖ View, che si occupa di mostrare i dati all'utente
- ❖ Controller, che si occupa di gestire la sequenza di interazioni con l'utente

3.2. Decomposizione in sottosistemi

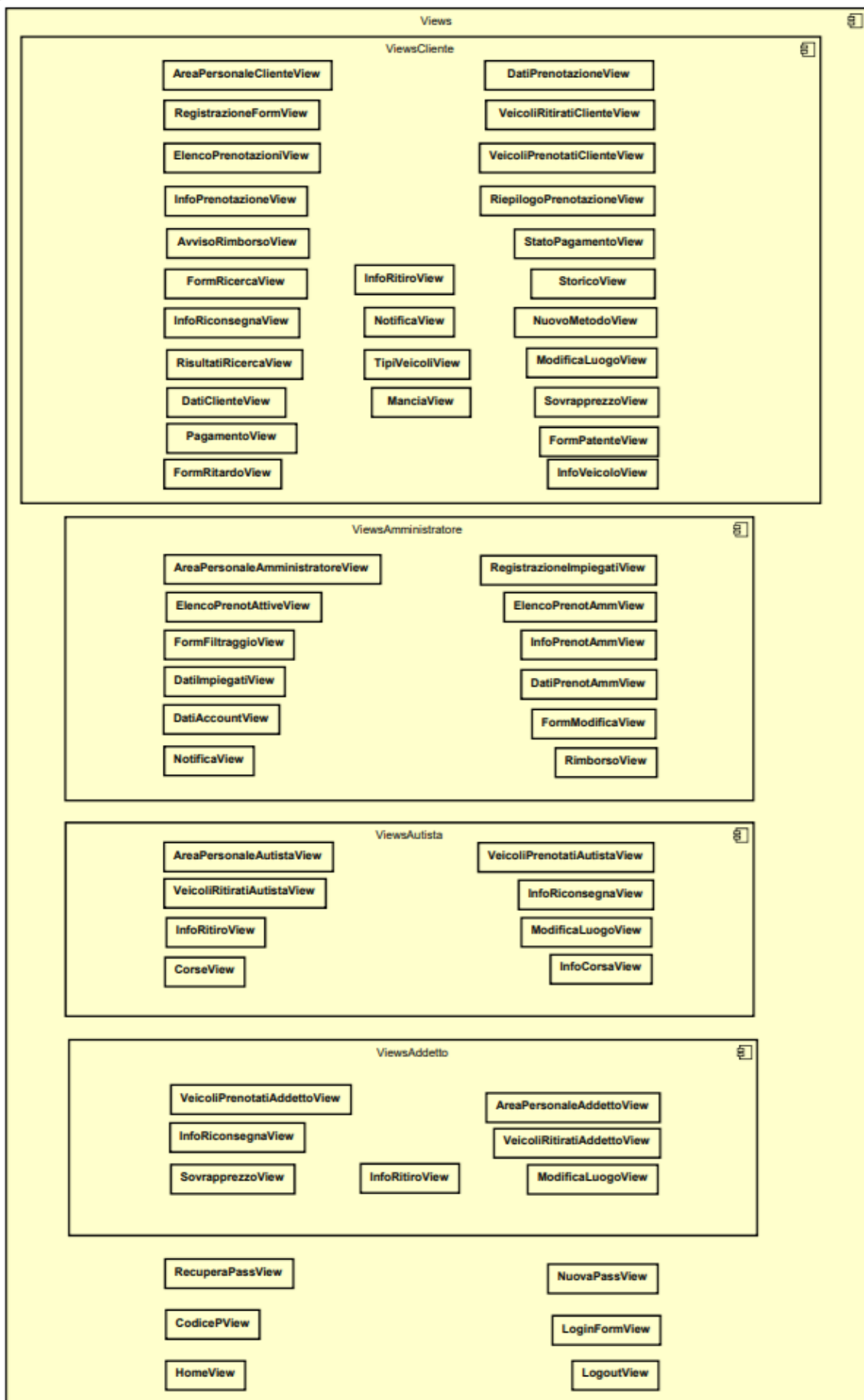
Il sistema è stato diviso in macroaree che rappresentano le funzionalità principali ed evidenziano le dipendenze tra i vari sottosistemi.



3.2.1. Models



3.2.2. Views



3.2.3. Controllers

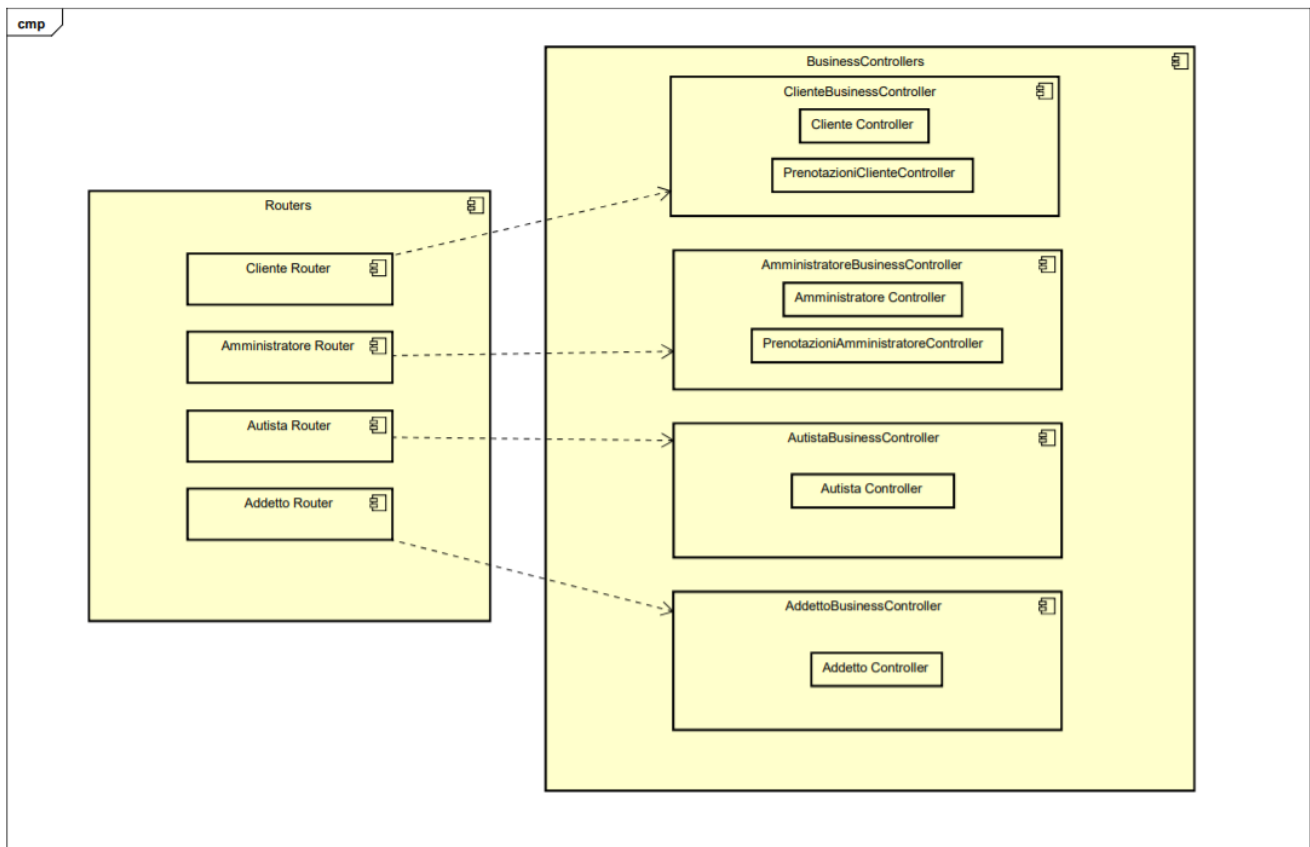
Per garantire efficienza e scalabilità i controllers sono stati divisi nei seguenti sottoinsiemi:

Router

I Router gestiscono le rotte delle richieste http. In seguito queste richieste saranno elaborate attraverso i BusinessControllers.

BusinessControllers

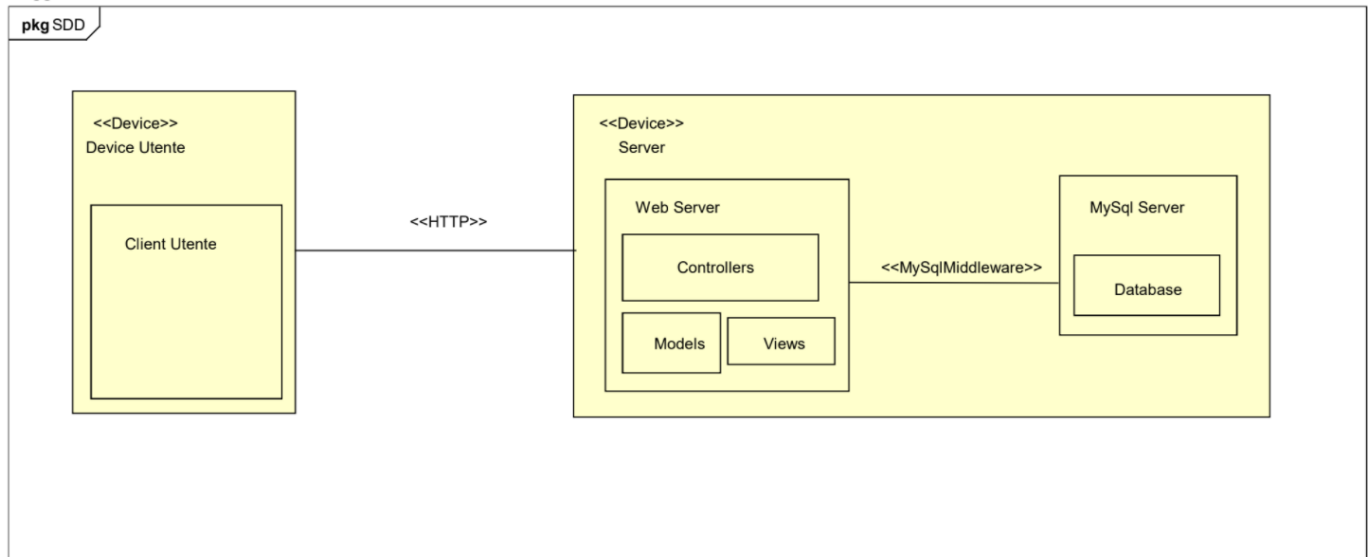
Il compito dei BusinessControllers è quello di gestire la logica di elaborazione del sistema.



3.3. Mappatura hardware e software

La struttura hardware proposta è costituita da un server centrale connesso con il Database, in modo tale da poterlo interrogare per gestire le richieste client. I client inviano le richieste al server tramite protocollo HTTP.

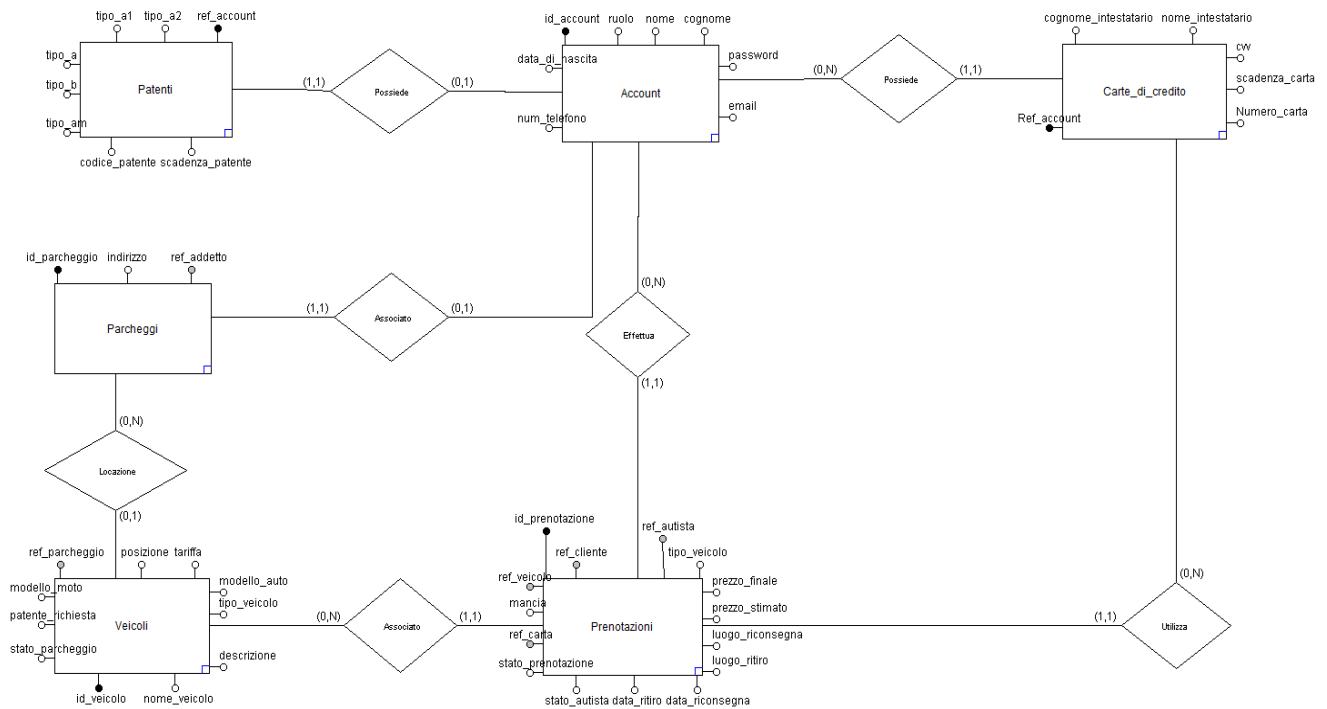
Lo scambio di informazioni tra Server e Database avviene tramite un middleware.



3.4. Gestione dati persistenti

Per la gestione dei dati persistenti viene utilizzato un database relazionale MySQL. Nel database vengono salvati relativi a utenti, veicoli, parcheggi e prenotazioni.

3.4.1. Diagramma E - R



3.4.2. Descrizione delle tabelle

3.4.2.1. Account

Nome	Tipo	Vincoli	Descrizione
id_account	int	PRIMARY KEY AUTO INCREMENT	Codice identificativo univoco dell'utente, generato automaticamente dal DBMS alla creazione di una nuova tupla.
ruolo	ENUM	NOT NULL, CHECK ("ruolo" IN ('Cliente', 'Autista', 'Addetto', 'Amministratore'))	Tipologia di account
nome	varchar(255)	NOT NULL	Nome dell'utente
cognome	varchar(255)	NOT NULL	Cognome dell'utente

data_di_nascita	date	NOT NULL	Data di nascita dell'utente
num_telefono	varchar(255)	NOT NULL	Numero di telefono dell'utente
email	varchar(255)	NOT NULL, UNIQUE	Email dell'utente
password	varchar(255)	NOT NULL	Password dell'utente

3.4.2.2. Prenotazioni

Nome	Tipo	Vincoli	Descrizione
id_prenotazione	int	PRIMARY KEY AUTO INCREMENT	Codice identificativo univoco della prenotazione, generato automaticamente dal DBMS alla creazione di una nuova tupla
ref_cliente	int	FOREIGN KEY REFERENCES "Account" ("id_account")	Codice identificativo del cliente che ha effettuato la prenotazione
ref_autista	int	FOREIGN KEY REFERENCES "Account" ("id_account")	Codice identificativo dell'autista associato alla prenotazione, è null se il cliente non ha richiesto un autista
tipo_veicolo	ENUM	NOT NULL, CHECK ("tipo_veicolo" IN ('Automobile', 'Moto', 'Bicicletta', 'Monopattino'))	Tipologia del veicolo associato alla prenotazione
ref_veicolo	int	FOREIGN KEY REFERENCES "Veicoli" ("id_veicolo")	Codice identificativo del veicolo associato alla prenotazione
mancia	float		Importo della mancia volontaria per l'autista
ref_carta	varchar(255)	FOREIGN KEY	Numero identificativo

		REFERENCES "carta di credito" ("numero_carta")	della carta di credito associata alla prenotazione
stato_prenotazione	ENUM	NOT NULL, CHECK ("stato_prenotazio ne" IN ('Pagato', 'Non pagato','Veicolo ritirato', 'Veicolo riconsegnato', 'Annullato', 'Rimborsato'))	
stato_autista	ENUM	CHECK ("stato_autista" IN ('Da confermare', 'Confermato'))	
data_ritiro	datetime	NOT NULL	Data e ora di inizio prenotazione
data_riconsegna	datetime	NOT NULL	Data e ora di fine prenotazione
luogo_ritiro	varchar(255)	NOT NULL	Luogo di ritiro del veicolo associato alla prenotazione
luogo_riconsegna	varchar(255)	NOT NULL	Luogo di riconsegna del veicolo associato alla prenotazione
prezzo_stimato	float	NOT NULL	Prezzo stimato della prenotazione
prezzo_finale	float	NOT NULL	Prezzo finale della prenotazione, comprende eventuali sovrapprezzi
motivo_ritardo	varchar(255)		Descrizione della motivazione del ritardo nella riconsegna del veicolo

3.4.2.3. Patenti

Nome	Tipo	Vincoli	Descrizione
codice_patente	varchar(255)	NOT NULL	Codice identificativo della patente
scadenza_patente	date	NOT NULL	Data di scadenza della patente
tipo_a	tinyint	NOT NULL, DEFAULT '0'	Tipologia della patente
tipo_b	tinyint	NOT NULL, DEFAULT '0'	Tipologia della patente
tipo_am	tinyint	NOT NULL, DEFAULT '0'	Tipologia della patente
tipo_al	tinyint	NOT NULL, DEFAULT '0'	Tipologia della patente
tipo_a2	tinyint	NOT NULL, DEFAULT '0'	Tipologia della patente
ref_account	int	PRIMARY KEY, FOREIGN KEY REFERENCES Account ("id_account")	

3.4.2.4. Carte_di_credito

Nome	Tipo	Vincoli	Descrizione
numero_carta	varchar(255)	NOT NULL	Numero identificativo univoco della carta di credito
ref_account	int	PRIMARY KEY , FOREIGN KEY REFERENCES "Account" ("id_account")	
nome_intestatario	varchar(255)	NOT NULL	Nome dell'intestatario della carta di credito

cognome_intestatario	varchar(255)	NOT NULL	Cognome dell'intestatario della carta di credito
scadenza_carta	varchar	NOT NULL	Data di scadenza della carta di credito
cvv	int	NOT NULL	Codice CVV della carta di credito

3.4.2.5. Veicoli

Nome	Tipo	Vincoli	Descrizione
id_veicolo	int	PRIMARY KEY	Codice identificativo univoco del veicolo, generato automaticamente dal DBMS alla creazione di una nuova tupla.
nome_veicolo	varchar(255)	NOT NULL	Nome del veicolo
descrizione	varchar(500)	NOT NULL	Descrizione del veicolo
tipo_veicolo	ENUM	NOT NULL, CHECK ("tipo_veicolo" IN ('Automobile', 'Moto', 'Bicicletta', 'Monopattino'))	Tipologia di veicolo
modello_auto	ENUM	CHECK("modello_auto" IN 'Suv', 'Utilitaria', 'Berlina')	Categoria di auto, null se il tipo_veicolo non è auto
modello_moto	ENUM	CHECK("modello_moto" IN 'Ciclomotore - 50cc', 'Scooter - 125cc', 'Turistica - 600cc', 'Adventure - 1200cc')	Categoria di moto, null se il tipo_veicolo non è moto
patente_richiesta	ENUM	CHECK ("patente_richiesta" IN ('tipo_a', 'tipo_b',	Tipologia di patente richiesta per guidare il

		'tipo_am', 'tipo_a1', 'tipo_a2', 'no'))	veicolo
stato_veicolo	ENUM	NOT NULL, CHECK ("stato_veicolo" IN ('Ritirato', 'Riconsegnato'))	Stato attuale del veicolo
ref_parcheggio	int	FOREIGN KEY REFERENCES "Parcheggio" ("id_parcheggio")	Codice identificativo univoco del parcheggio in cui si trova il veicolo
posizione	varchar(255)	NOT NULL	Indica la posizione del veicolo all'interno del parcheggio o, se fuori dallo stallo, la via in cui si trova.
tariffa	float	NOT NULL	Tariffa oraria per il noleggio del veicolo

3.4.2.6. Parcheggi

Nome	Tipo	Vincoli	Descrizione
id_parcheggio	varchar(255)	PRIMARY KEY	Nome identificativo e univoco del parcheggio.
indirizzo	varchar(255)	NOT NULL	Indirizzo del parcheggio
ref_addetto	int	FOREIGN KEY REFERENCES "Account" ("id_account")	Codice identificativo univoco dell'addetto associato al parcheggio

3.5. Accesso e sicurezza

Il sistema consente agli utenti di accedere alle varie funzionalità solo dopo aver effettuato l'autenticazione. Per quel che riguarda la funzionalità di ricerca dei veicoli, invece, essa è disponibile anche agli utenti non autenticati.

Il controllo degli accessi avviene tramite l'immissione di credenziali valide da parte dell'utente. È prevista, inoltre, la cifratura della password immessa dall'utente tramite la funzione hash crittografica MD5.

Il sistema permette all'utente di modificare la propria password, nel caso l'abbia dimenticata, inviando un codice tramite email.

3.6. Boundary conditions

In caso di errore nell'inserimento dei dati il sistema avverte l'utente con un messaggio di errore e ne consente il reinserimento.