



Progetto P.I.S.S.I.R
Gruppo 06

Galliera - Marino - Ternullo - Vecchio

May 27, 2024

Contents

1	Specifica	1
1.1	Dominio Applicativo	1
1.2	Casi d'uso	3
1.2.1	Login	3
1.2.2	Crea utente	4
1.2.3	Modifica dati	4
1.2.4	Diventa premium	4
1.2.5	Occupa posto	4
1.2.6	Richiede ricarica	5
1.2.7	Richiede estensione ricarica	5
1.2.8	Prenota posto	5
1.2.9	Modifica Prenotazione	6
1.2.10	Cancella prenotazione	6
1.2.11	Monitora parcheggio	6
1.2.12	Aggiorna prezzo	7
1.2.13	Visualizza posti	7
1.2.14	Visualizza ricarica	7
1.2.15	Visualizza prenotazioni	7
1.2.16	Aggiorna stato posto	7
1.2.17	Aggiorna stato MWBot	7
1.2.18	Interrompi ricarica	7
1.3	Requisiti funzionali e Non funzionali	8
2	Progettazione	9
2.1	Architettura del sistema	9
2.2	API REST	10
2.3	MQTT TOPICS	11
2.3.1	ParkCharge/StatoPosti/ID_Posto/data	11
2.3.2	ParkCharge/StatoSensore/ID_Posto/data	11
2.3.3	ParkCharge/StatoMWBot/ID_MWBot/data	11
2.3.4	ParkCharge/StatoRicariche/ID_MWBot/data	11
2.3.5	ParkCharge/RichiediRicarica/ID_MWBot/data	12
2.3.6	ParkCharge/EseguiRicarica/ID_MWBot/data	12
2.3.7	ParkCharge/Notifiche/SostaConclusa/ID_Utente/data	12
2.3.8	ParkCharge/Notifiche/RicaricaConclusa/ID_Utente/data	12

List of Figures

1.1	Diagramma del dominio applicativo	2
1.2	Diagramma dei casi d'uso	3
1.3	RF01 Login	4
1.4	RF06 Occupa posto	5
1.5	RF08 Prenota posto	6
1.6	RF09 Modifica prenotazione	6
1.7	RF11 Monitora Parcheggio	7
1.8	Requisiti Funzionali	8
1.9	Requisiti Non Funzionali	8
2.1	Architettura a microservizi	9
2.2	Diagramma delle classi	10

Chapter 1

Specifica

1.1 Dominio Applicativo

Il dominio applicativo va a fornire una panoramica del sistema di gestione di un parcheggio smart dotato di MWbot. Le classi principali sono:

- Utente: rappresenta una persona, cliente o amministratore, che interagisce col sistema.
- Prenotazione: definisce l'occupazione di un determinato posto da parte di un utente per un determinato lasso di tempo. Sono le prenotazioni non ancora concluse.
- Storico: registro delle prenotazioni passate e concluse, sono già state pagate.
- Ricarica: rappresenta una ricarica effettuata da un MWBot durante una prenotazione. Percentuale richiesta indica la percentuale effettiva di ricarica a cui l'auto deve essere portata.
- Costi: rappresenta i costi di parcheggio, ricarica, costo premium e penale.

utti i clienti possono arrivare al parcheggio e occupare il posto assegnatoli dal sistema. Solo i clienti Premium possono prenotarne uno preventivamente.

Ad ogni prenotazione verrà associato un pagamento che dipenderà dai costi, dalla durata della sosta e dall'eventuale ricarica, la quale può essere richiesta dal cliente al momento del parcheggio.

L'amministratore gestisce le prenotazioni e può, in qualsiasi momento, modificare i costi del parcheggio. Sono state create, inoltre, due nuove classi:

- Carta: oggetto che rappresenta la carta associata al cliente (ad ogni cliente può essere associata una sola carta alla volta).
- TimeStamp: oggetto che rappresenta un istante nel tempo.

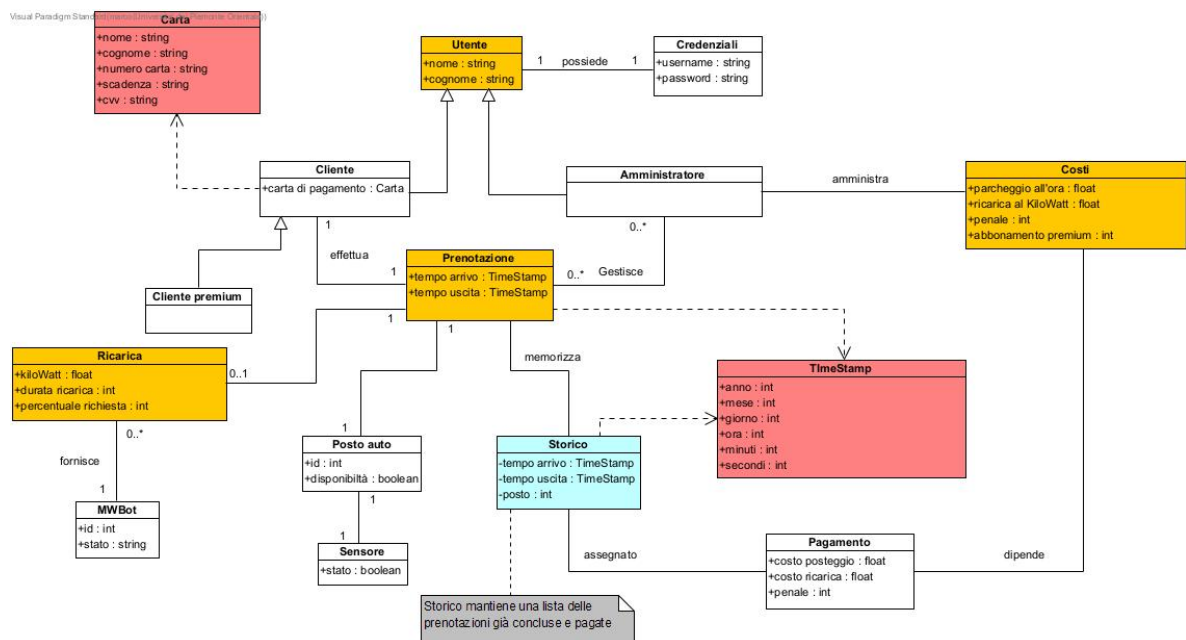


Figure 1.1: Diagramma del dominio applicativo

1.2 Casi d'uso

Diagramma dei casi d'uso

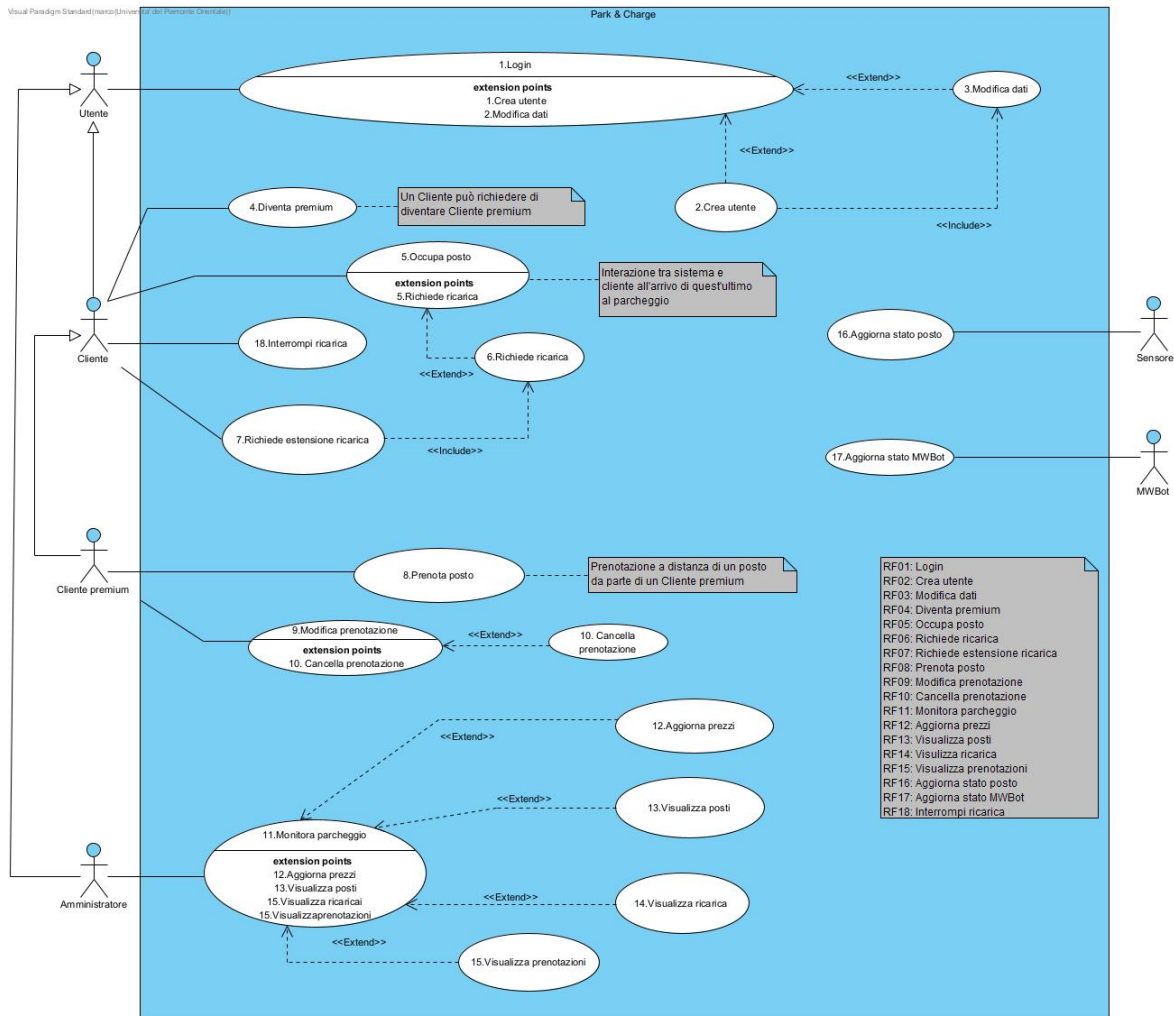


Figure 1.2: Diagramma dei casi d'uso

1.2.1 Login

Un utente inserisce nel form le proprie credenziali (username e password). Se le credenziali sono corrette l'utente accede al sistema nel menù a lui dedicato (menù cliente, menù cliente premium, menù amministratore). Se le credenziali sono errate viene mostrato un messaggio di errore e gli viene ripresentato il form di login.

Dal form di login sarà anche possibile accedere al form di registrazione.

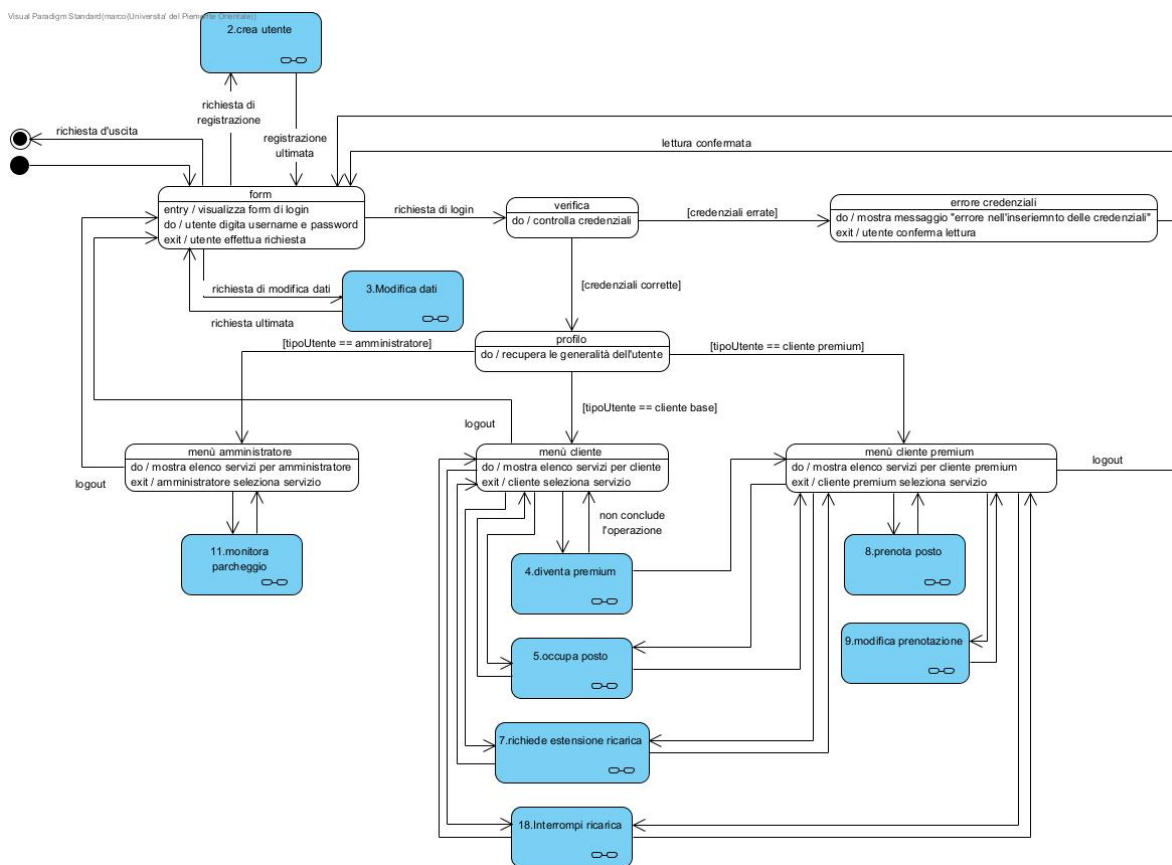


Figure 1.3: RF01 Login

1.2.2 Crea utente

Nel form di registrazione l'utente potrà andare ad inserire i propri dati, quali: Nome, Cognome, Username e Password; successivamente verrà richiesto di inserire i dati di una carta di credito da associare all'account appena creato. Dopo la registrazione l'utente viene riportato al form di login.

1.2.3 Modifica dati

Nel form l'utente potrà andare a modificare i propri dati, quali: Nome, Cognome, Username, Password e dati della carta.

1.2.4 Diventa premium

Un utente base può scegliere di diventare un utente premium con un pagamento una tantum. Se un utente premium prova a fare l'upgrade, il sistema non glielo permette.

1.2.5 Occupa posto

Un utente, base o premium, che arriva al parcheggio richiede che gli venga assegnato un posto. Nel caso in cui l'utente, ovviamente premium, abbia prenotato un posto gli verrà sicuramente assegnato, altrimenti solamente se ci sono posti non prenotati nel lasso di tempo scelto.

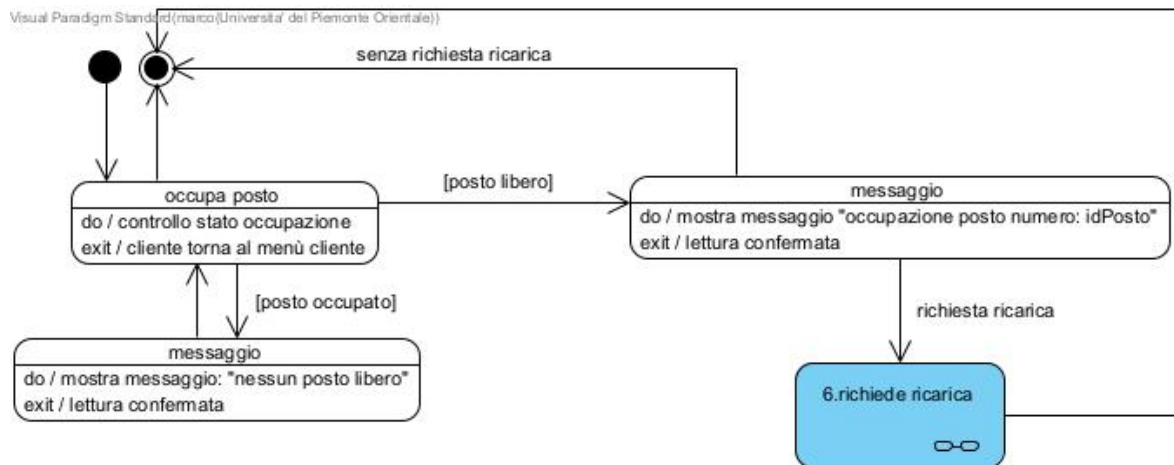


Figure 1.4: RF06 Occupa posto

1.2.6 Richiede ricarica

Un utente, base o premium, che arriva al parcheggio può richiedere che gli venga fornita una ricarica, richiedendo che essa venga fornita fino al raggiungimento di una determinata percentuale di carica della vettura. A quel punto il sistema darà all'utente una previsione del tempo richiesto dal MWBot, in base alla disponibilità, per ricaricare la vettura alla percentuale di carica desiderata.

Nel caso in cui la ricarica non dovesse essere soddisfacibile entro il tempo di sosta il sistema mostrerà un messaggio di avviso. Al termine della ricarica l'utente riceve una notifica.

La policy di scheduling delle ricariche è **EDF**, *Earliest Deadline First*. Questa politica garantisce il rispetto della deadline prefissata (il timestamp di fine posteggio) e massimizza il numero di job (ricariche) soddisfabili. L'algoritmo è preemptive (una ricarica iniziata non deve essere per forza conclusa direttamente) e dinamico (le ricariche vengono prese in carico in modo dinamico se soddisfacibili).

1.2.7 Richiede estensione ricarica

Un utente che ha richiesto la carica del proprio veicolo può, al termine di questa, può richiederne un'estensione.

1.2.8 Prenota posto

Un utente premium può prenotare un posto per un determinato lasso di tempo tramite un form: il sistema accetterà la richiesta solo se ci sono posti liberi in quel lasso di tempo.

Nel caso di successo il sistema mostrerà un messaggio di successo senza specificare il posto assegnatogli, altrimenti mostra un messaggio di errore e riporta l'utente nel form iniziale.

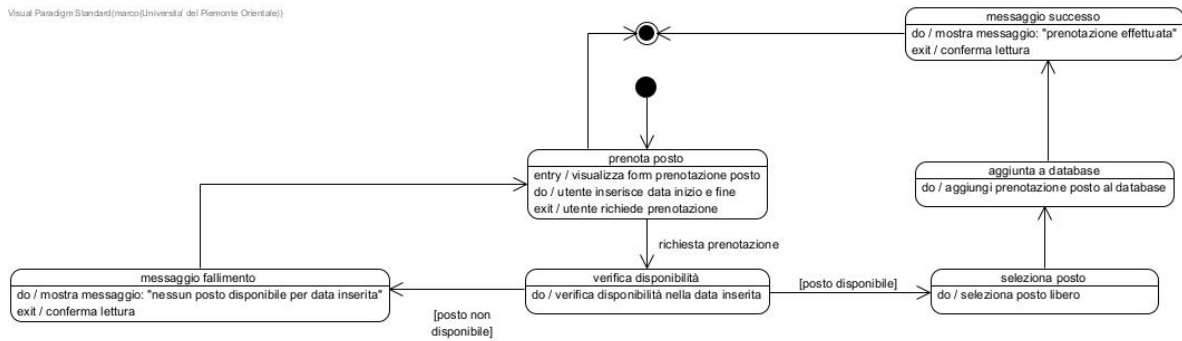


Figure 1.5: RF08 Prenota posto

1.2.9 Modifica Prenotazione

L'utente premium pu  modificare la prenotazione effettuata prima dell'ora di inizio.

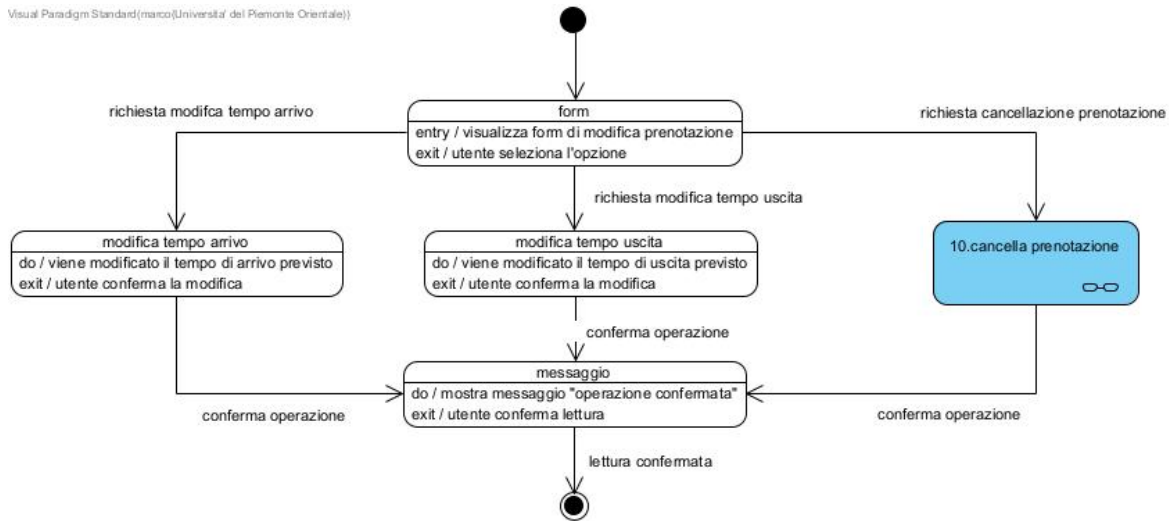


Figure 1.6: RF09 Modifica prenotazione

1.2.10 Cancella prenotazione

Un utente premium, prima dell'ora di inizio della prenotazione effettuata, pu  cancellare tale prenotazione senza penale. Il posto che gli era stato assegnato ritorna libero nel lasso di tempo della prenotazione.

1.2.11 Monitora parcheggio

L'Amministratore pu , attraverso un apposito form, accedere agli altri form di aggiornamento prezzi, visualizzazione posti, visualizzazione ricariche e le prenotazioni effettuate da utenti premium.

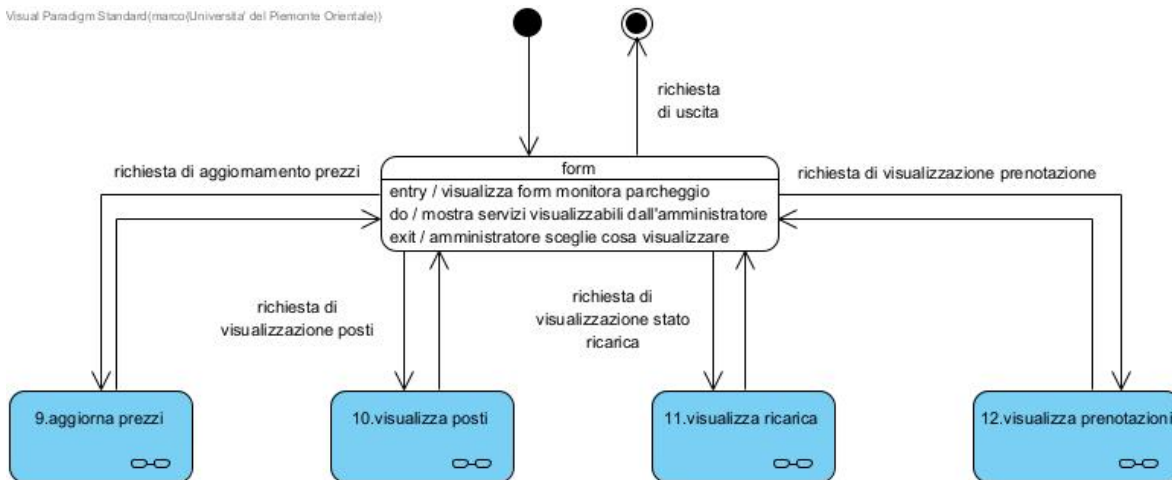


Figure 1.7: RF11 Monitora Parcheggio

1.2.12 Aggiorna prezzo

L'Amministratore pu , attraverso un apposito form, andare a modificare i costi: del parcheggio all'ora, della ricarica al KiloWatt, della penale e il costo per l'abbonamento al servizio Premium.

1.2.13 Visualizza posti

L'Amministratore   in grado di visualizzare tutti i posti del parcheggio e le loro caratteristiche: se sono prenotati, occupati o liberi. Nel caso un posto sia occupato, o prenotato, verr  visualizzato l'utente che lo occupa, o lo ha prenotato, ed i relativi tempi di sosta e ricarica richiesti.

1.2.14 Visualizza ricarica

L'Amministratore   in grado di visualizzare le ricariche in corso e quelle prenotate, pu  anche vedere dove sono gli MWBot e cosa stanno facendo in quel momento.

1.2.15 Visualizza prenotazioni

L'Amministratore   in grado di visualizzare le prenotazioni future degli utenti.

1.2.16 Aggiorna stato posto

Il sensore aggiorna periodicamente lo stato del posto che controlla.

1.2.17 Aggiorna stato MWBot

L'MWBot aggiorna il proprio stato.

1.2.18 Interrompi ricarica

Un utente pu  richiedere che la ricarica in corso venga istantaneamente interrotta.   obbligatorio interrompere la ricarica in corso per spostare il veicolo.

1.3 Requisiti funzionali e Non funzionali

REQUISITI FUNZIONALI	ID	GRUPPO
login	RF01	
crea utente	RF02	
modifica dati	RF03	
diventa premium	RF04	
occupa posto	RF05	
richiede ricarica	RF06	
richiede estensione ricarica	RF07	
prenota posto	RF08	
modifica prenotazione	RF09	
cancella prenotazione	RF10	
monitora parcheggio	RF11	
aggiorna prezzi	RF12	
visualizza posti	RF13	
visualizza ricariche	RF14	
visualizza prenotazioni	RF15	
aggiorna stato posti	RF16	
aggiorna stato MVBot	RF17	
interrompi ricarica	RF18	

Figure 1.8: Requisiti Funzionali

ID	DESCRIZIONE
RNF01	Specifica e progettazione saranno definite tramite diagrammi UML e linguaggio naturale.
RNF02	Il sistema sarà implementato in Java.
RFN03	L'interfaccia utente sarà realizzata tramite Java Swing.
RFN04	I database saranno implementati usando SQLite e Java DataBase Connectivity.
RFN05	Verrà utilizzato SQLiteBrowser per visualizzare e modificare i contenuti dei database.
RFN06	Verrà utilizzato Visual Paradigm per disegnare i diagrammi UML e le bozze dell'interfaccia utente.
RFN07	Verrà utilizzato Gradle per compilare, eseguire e testare il codice.
RFN08	Verrà utilizzato Git per condividere documentazione e codice tra gli sviluppatori.
RFN09	Verrà utilizzato GitLab come repository on-line.
RFN10	Verrà utilizzato Swagger OpenAPI per la documentazione dell'Api REST.
RFN11	Le date saranno memorizzate nel pattern: YYYY-MM-DD HH-MI-SS.
RFN12	L'username deve contenere almeno 4 caratteri.
RFM13	La password deve contenere almeno 6 caratteri di cui almeno una cifra e un carattere speciale.

Figure 1.9: Requisiti Non Funzionali

Chapter 2

Progettazione

2.1 Architettura del sistema

I componenti principali dell'architettura sono:

- Frontend: creazione di una desktop app che opera da punto di accesso al sistema, che contatta il backend tramite una API REST e comunica col sistema esterno Auth2 per l'autenticazione. Permette alle varie categorie di utenti (Clienti, Clienti Premium, Amministratore) di accedere a funzionalità diverse.
- Backend: permette agli utenti di accedere ai servizi del parcheggio e di ricarica. Gestisce lo scheduling delle ricariche con algoritmo EDF. Comunica attraverso MQTT col MQTT Broker per il controllo degli MWBot, l'accesso dati dei sensori e per inviare notifiche a utenti (quando viene conclusa la ricarica e a sosta conclusa). Si interfaccia inoltre agli utenti (Cliente e Amministratore) attraverso una REST API dove espone varie funzionalità tra cui prenotazione del posto, prenotazione ricarica e monitoraggio da parte dell'amministratore.
- Gestore IoT: permette al sistema di comunicare con MWBot e i sensori del parcheggio tramite un message broker (mosquitto).
- Database: permette al sistema di tenere delle prenotazioni, dei posti occupati e dei pagamenti.

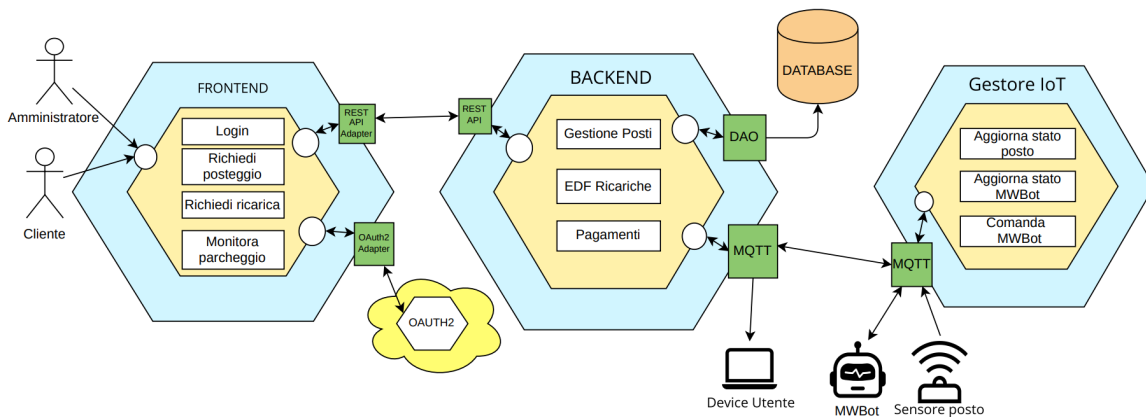


Figure 2.1: Architettura a microservizi

La rappresentazione di tale architettura nel diagramma delle classi è la seguente:

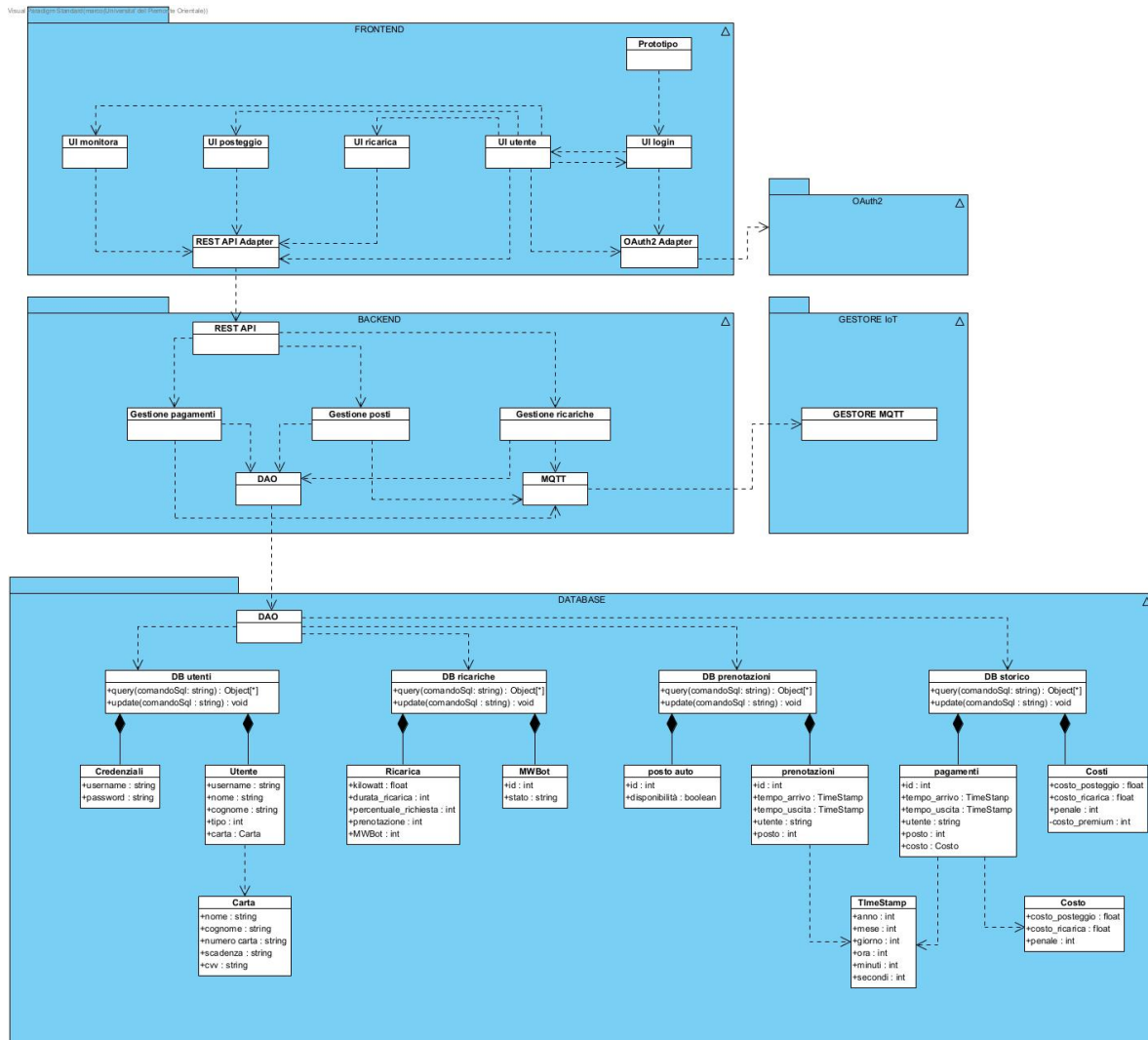


Figure 2.2: Diagramma delle classi

2.2 API REST

Abbiamo sviluppato la documentazione dell'API REST tramite l'editor Swagger ed è la seguente:

- link documentazione API rest¹
- link codice yaml²

¹<https://nicomarinio.altervista.org/RestAPI.html>

²<https://app.swaggerhub.com/apis/MARCOVECCHIO/ParkCharge/1.0.0>

2.3 MQTT TOPICS

2.3.1 ParkCharge/StatoPosti/ID_Posto/data

Il GestoreIoT aggiorna lo stato del singolo posto definito da ID_Posto.
Il formato di data è

```
{
  "stato" : String
}
```

dove il campo stato $\in \{ \text{"libero"}, \text{"occupato"} \}$.
Permessi: GestoreIoT Write, BackEnd Read.

2.3.2 ParkCharge/StatoSensore/ID_Posto/data

Il sensore aggiorna lo stato del posto che monitora definito da ID_Posto.
Il formato di data è:

```
{
  "stato" : String
}
```

dove il campo stato $\in \{ \text{"libero"}, \text{"occupato"} \}$.
Permessi: Sensore Write, GestoreIoT Read.

2.3.3 ParkCharge/StatoMWBot/ID_MWBot/data

Ogni MWBot comunica il proprio stato.

```
{
  "statoCarica" : String,
  "posizione" : Integer,
  "percentualeRicarica" : Integer
  "KW_Emessi" : Float
}
```

dove il campo statoCarica $\in \{ \text{"caricando"}, \text{"finito"} \}$.
Permessi: MWBot Write, GestoreIoT Read.

2.3.4 ParkCharge/StatoRicariche/ID_MWBot/data

Il GestoreIoT comunica lo stato del singolo MWBot.

```
{
  "statoCarica" : String,
  "posizione" : Integer,
  "percentualeRicarica" : Integer
  "KW_Emessi" : Float
}
```

dove il campo statoCarica $\in \{ \text{"caricando"}, \text{"finito"} \}$.
Permessi: GestoreIoT Write, BackEnd Read.

2.3.5 ParkCharge/RichiediRicarica/ID_MWBot/data

Il sistema comunica il prossimo comando per l'MWBot al GestoreIoT.

```
{
  "target" : Integer,
  "percentualeRicarica" : Integer
}
```

Permessi: BackEnd Write, GestoreIoT Read.

2.3.6 ParkCharge/EseguiRicarica/ID_MWBot/data

Il GestoreIoT comunica il prossimo comando all'MWBot.

```
{
  "target" : Integer,
  "percentualeRicarica" : Integer
}
```

Permessi: GestoreIoT Write, MWBot Read.

2.3.7 ParkCharge/Notifiche/SostaConclusa/ID_Utente/data

Il BackEnd comunica al dispositivo dell'utente il periodo finale di sosta, ed eventuale ricarica, con il costo pagato. Un utente può fare subscribe a tutto con ParkCharge/Notifiche/#/ID_Utente/

```
{
  "tempoSosta" : Integer,
  "costoSosta" : Integer,
  "kilowattUsati" : Integer,
  "costoRicarica" : Integer
}
```

Permessi: BackEnd Write, UserDevice Read.

2.3.8 ParkCharge/Notifiche/RicaricaConclusa/ID_Utente/data

Il BackEnd comunica al dispositivo dell'utente che la ricarica è stata conclusa, dando la possibilità di richiedere un'ulteriore ricarica.

```
{
  "kilowattUsati" : Integer,
  "costoRicarica" : Integer
}
```

Permessi: BackEnd Write, UserDevice Read.