



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	Obiettivi generali	3
2	Obiettivi di progettazione.....	3
3	Architettura software proposta	3
3.1	Panoramica.....	3
3.2	Decomposizione in sottoinsiemi	3
3.2.1	Mappatura degli oggetti nei sottoinsiemi	5
3.3	Mappatura Hardware/Software.....	10
3.4	Progetto del database	11
3.4.1	Modello ER	11
3.4.2	Descrizione delle tabelle	11
3.5	Accesso al sistema	15

1 Obiettivi generali

Il sistema si pone come obiettivo quello di agevolare i residenti a Palermo negli spostamenti all'interno della città, offrendo un efficiente servizio di noleggio mezzi.

2 Obiettivi di progettazione

- Il sistema deve essere correttamente visualizzato dalle ultime versioni stabili di Mozilla, Chrome, Firefox e Microsoft Edge.
- Il sistema deve impedire l'inserimento di input non corretti da parte degli utenti.
- Il DBMS deve rimanere sempre attivo per consentire agli utenti una corretta fruizione dei servizi offerti dal sistema.

3 Architettura software proposta

3.1 Panoramica

Il sistema è un software costituito da un singolo nodo centrale che memorizza tutti i dati di interesse. Gli utenti interagiscono con il sistema tramite i loro personal device (laptop, smartphone, ecc...). Il sistema non prevede alcun tipo di interazione tra utenti: il tipo di architettura più appropriata è, per tale motivo, quella Client-Server.

I nodi client sono rappresentati da tutti i dispositivi che si connettono al sistema per usufruire dei servizi offerti dallo stesso. quindi?

Il nodo server si occupa di rispondere alle richieste dei nodi client, immagazzinando e gestendo le informazioni inviate da questi ultimi mediante un database relazionale basato sul linguaggio SQL.

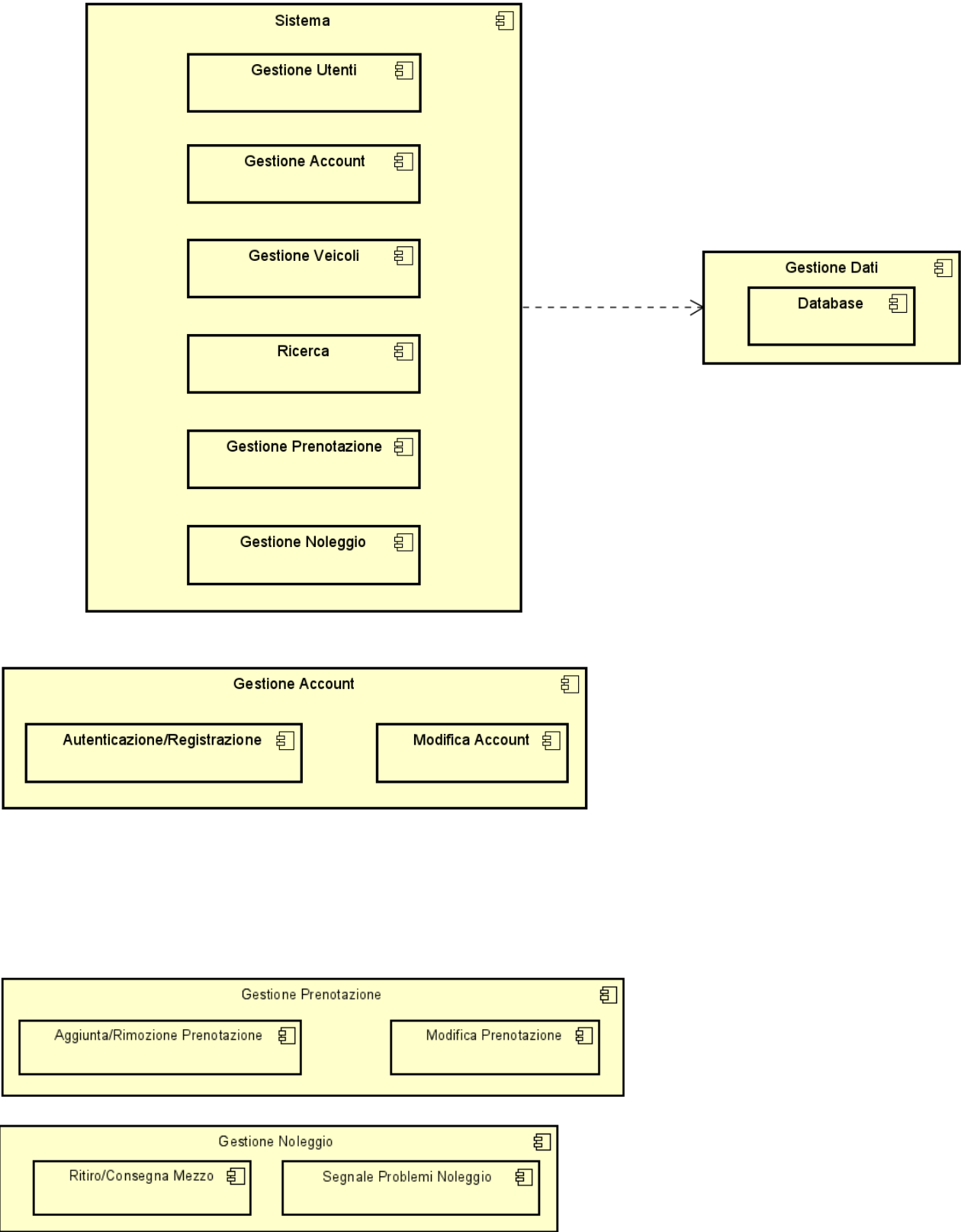
3.2 Decomposizione in sottoinsiemi

Di seguito una panoramica che mostra i sottosistemi che compongono il sistema proposto, realizzata mediante i component diagram di UML.

I sottosistemi che compongono il sistema comprenderanno, ovviamente, una serie di servizi che non sono stati rappresentati nella panoramica sopra.

Per offrire al lettore una maggiore comprensione dei sottosistemi rappresentati sopra, si è deciso di

suddividerli ulteriormente in sottosistemi più piccoli.

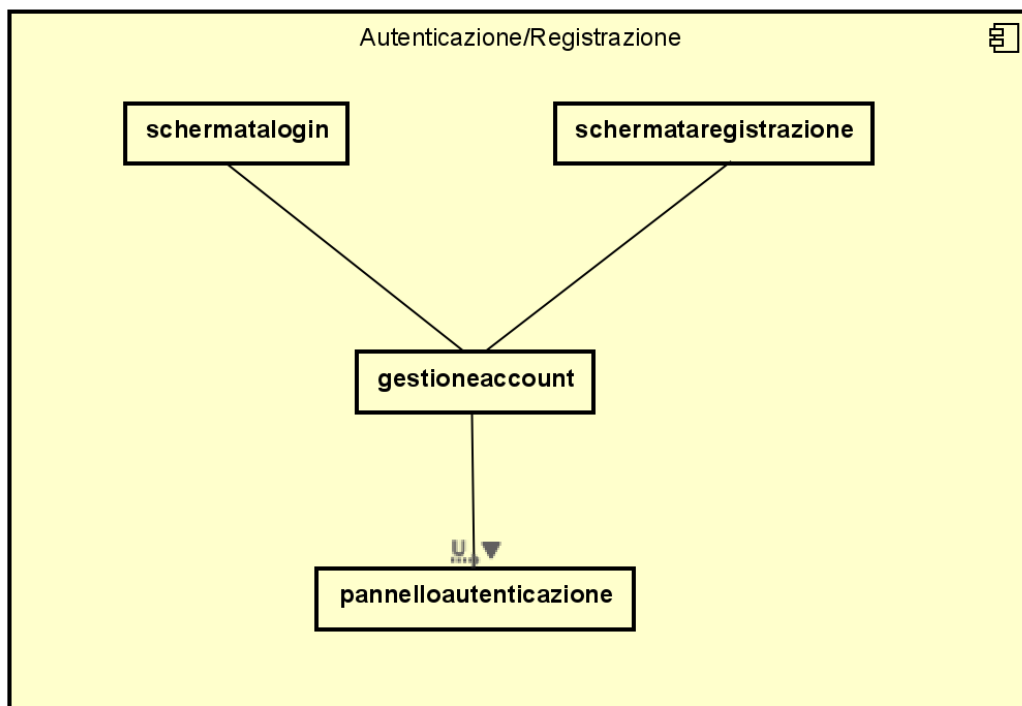


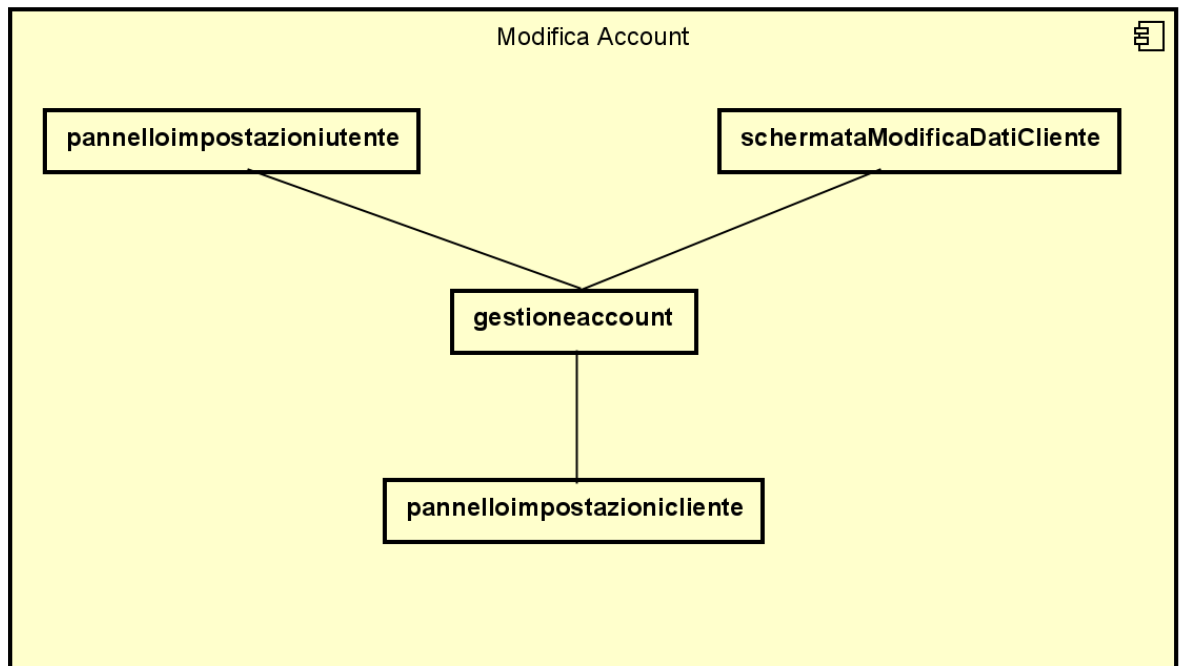
Alla luce dell'architettura scelta, si è deciso di inglobare tutte le operazioni che richiedono uno scambio di messaggi con il DBMS -e quindi con la BoundaryDBMS- all'interno del nodo server.

3.2.1 Mappatura degli oggetti nei sottoinsiemi

3.2.1.1 *Gestione Account*

3.2.1.1.1 Autenticazione/Registrazione

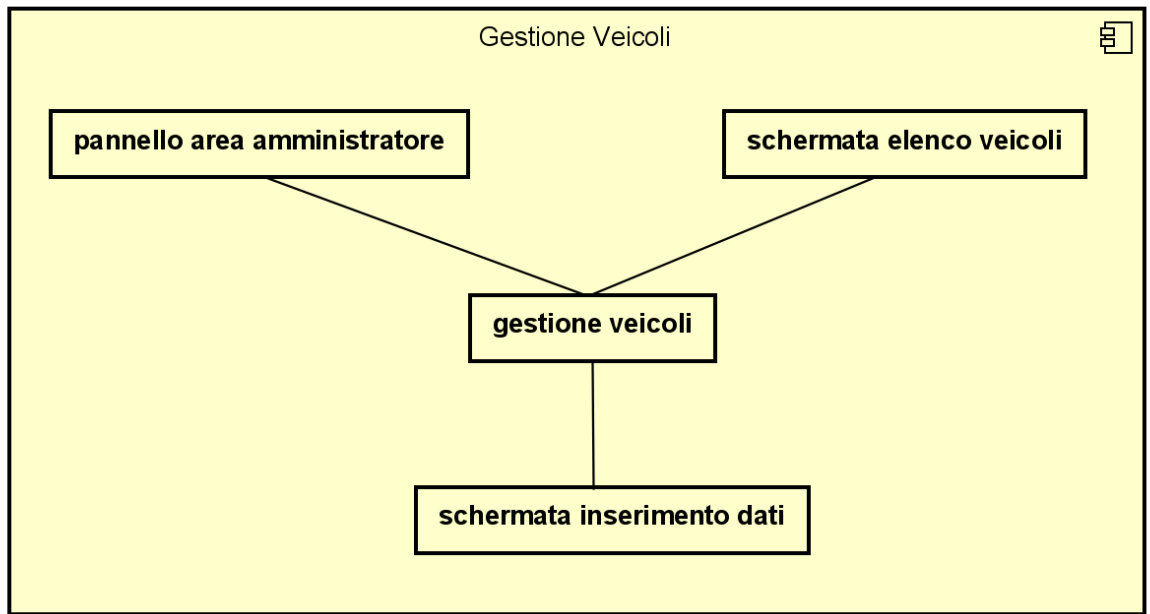




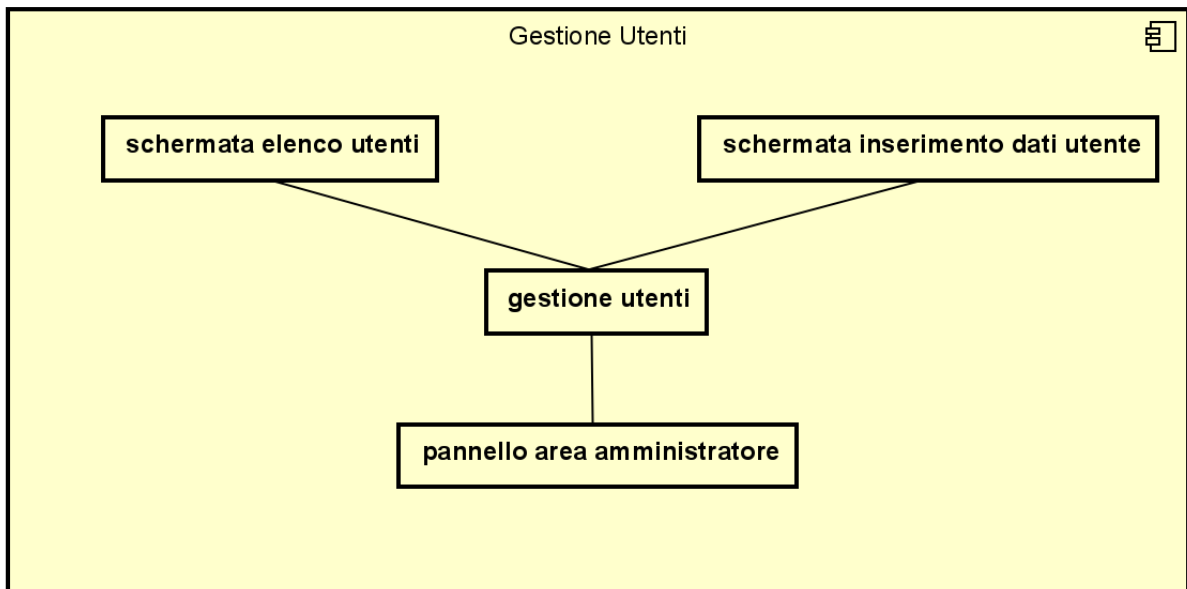
La schermataModificaDatiCliente comprende:

- schermataInserimentoMetodoPagamento;
- schermataInserimentoPatente;
- schermataAggiornaPatente;
- schermataElencoMetodiPagamento;
- schermataModificaPassword;
- schermataModificaDati.

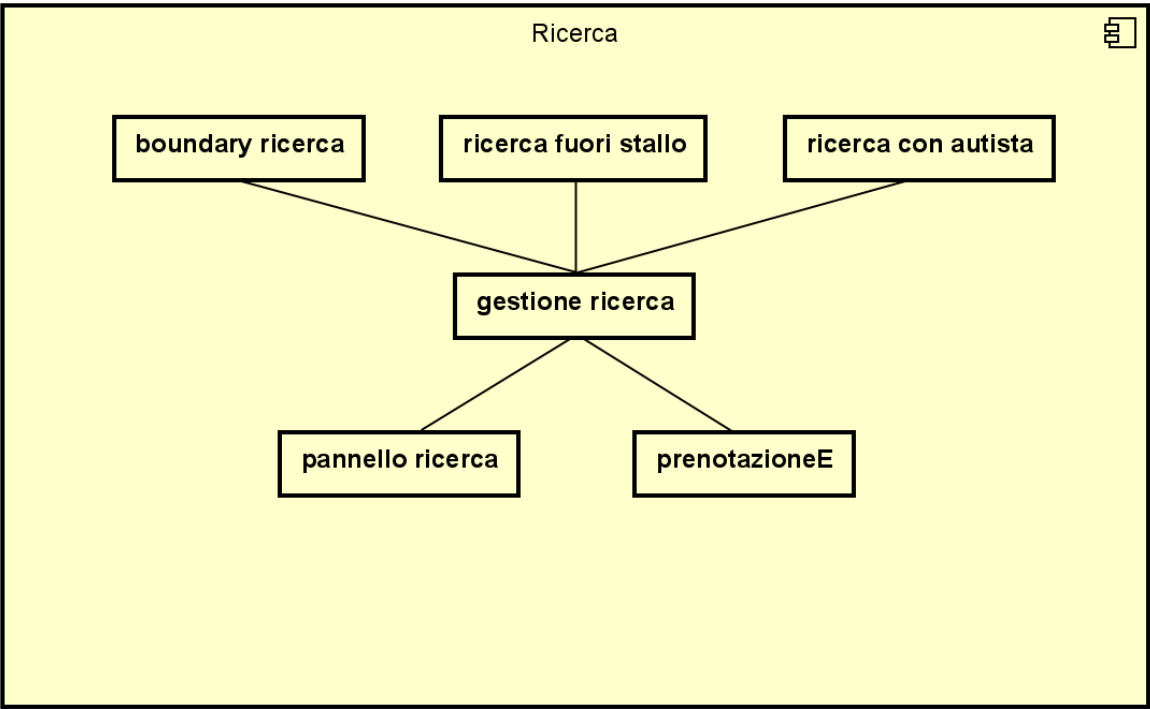
3.2.1.1.3 Gestione Veicoli



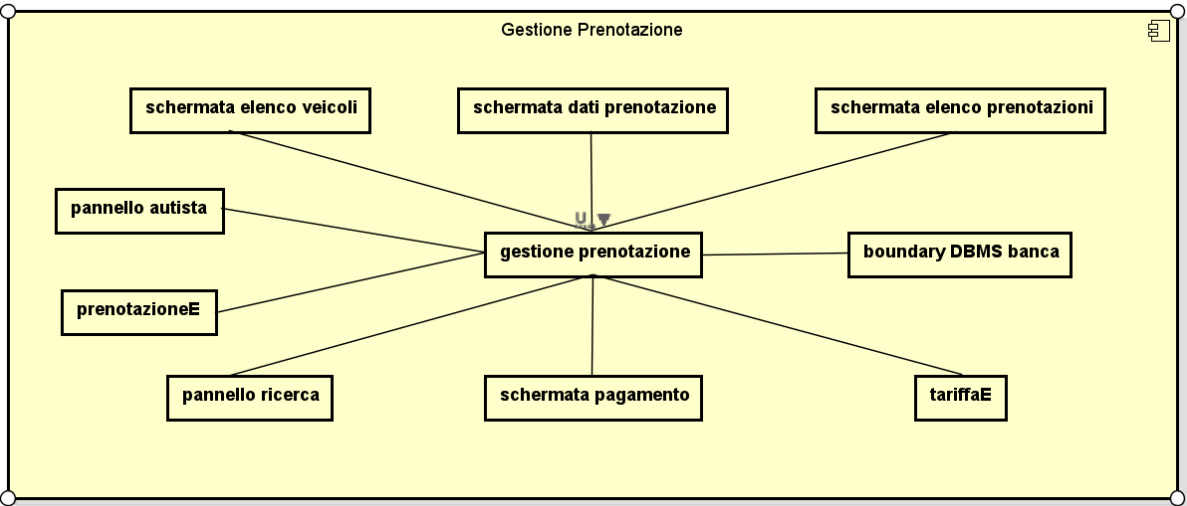
3.2.1.2 Gestione Utenti



3.2.1.3 Ricerca

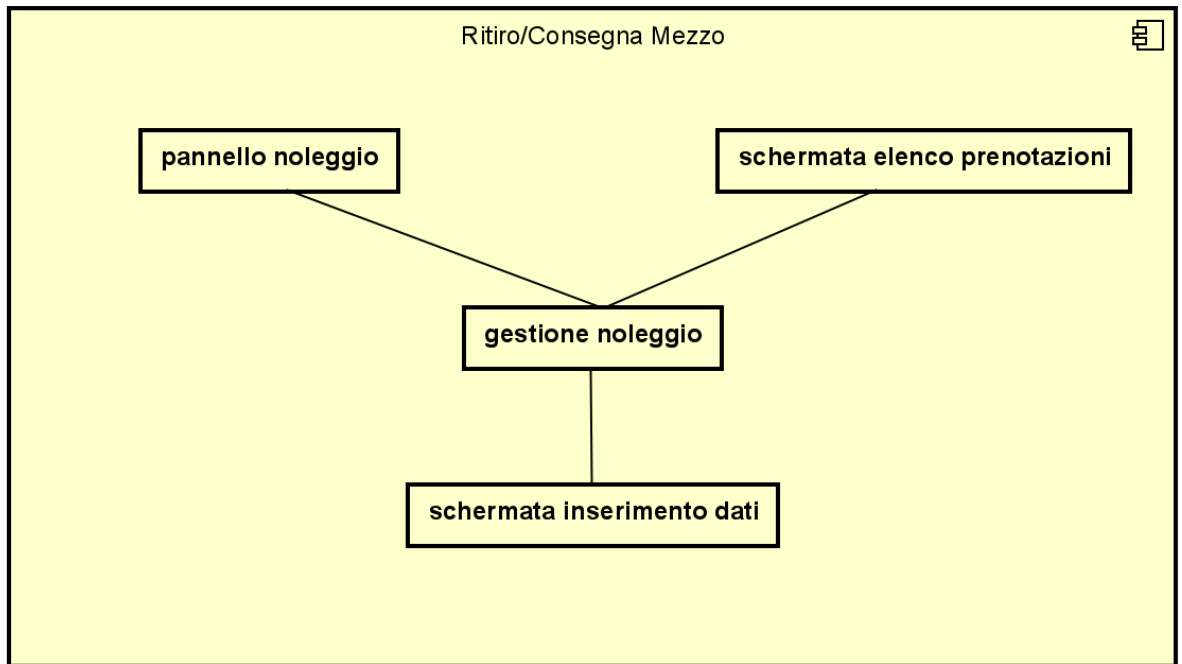


3.2.1.3.1 Prenotazione

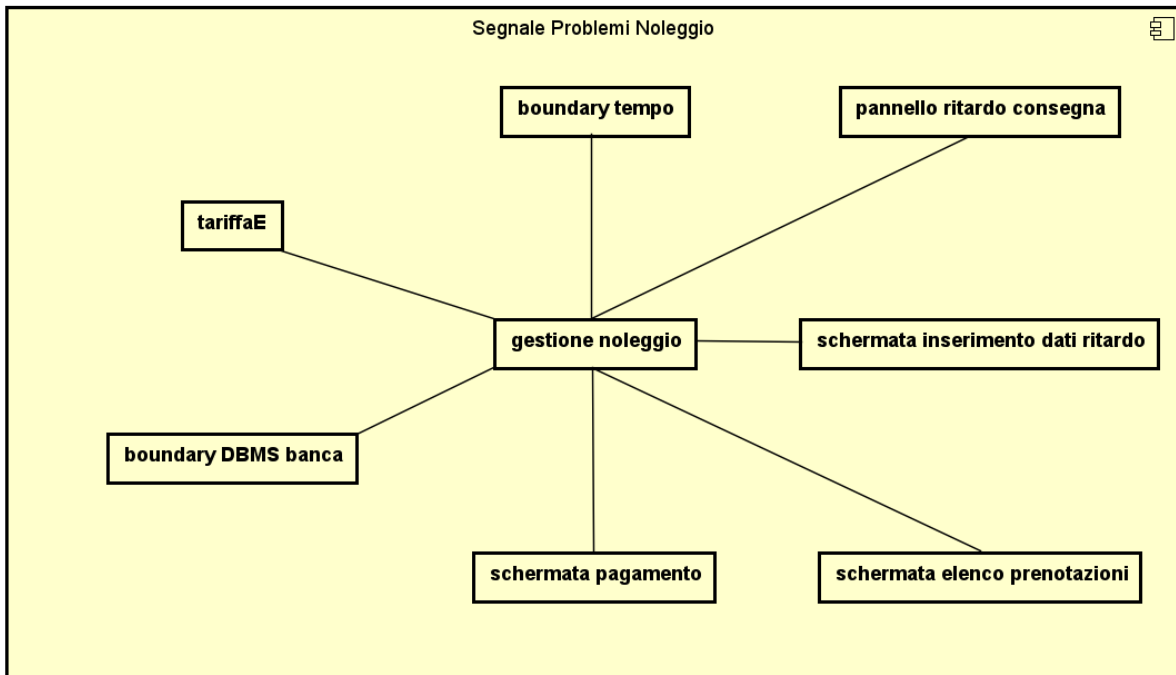


3.2.1.4 Gestione Noleggio

3.2.1.4.1 Ritiro/Consegna Mezzo



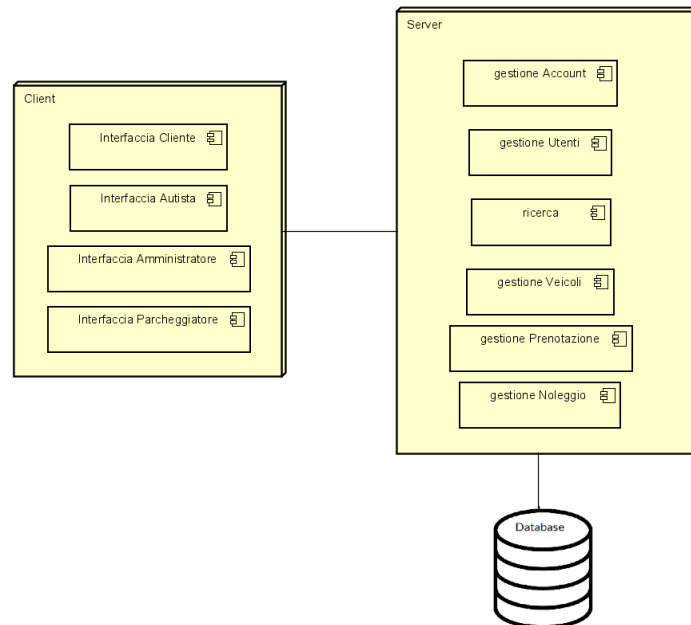
3.2.1.4.2 Segnala Problemi Noleggio



3.3 Mappatura Hardware/Software

La struttura hardware proposta segue, anch'essa, il modello client-server: è costituita da un server centrale e da tanti client che si identificano coi device degli utenti.

Al server centrale si collegano il DBMS e tutti i client, i quali potranno richiedere di usufruire dei servizi che



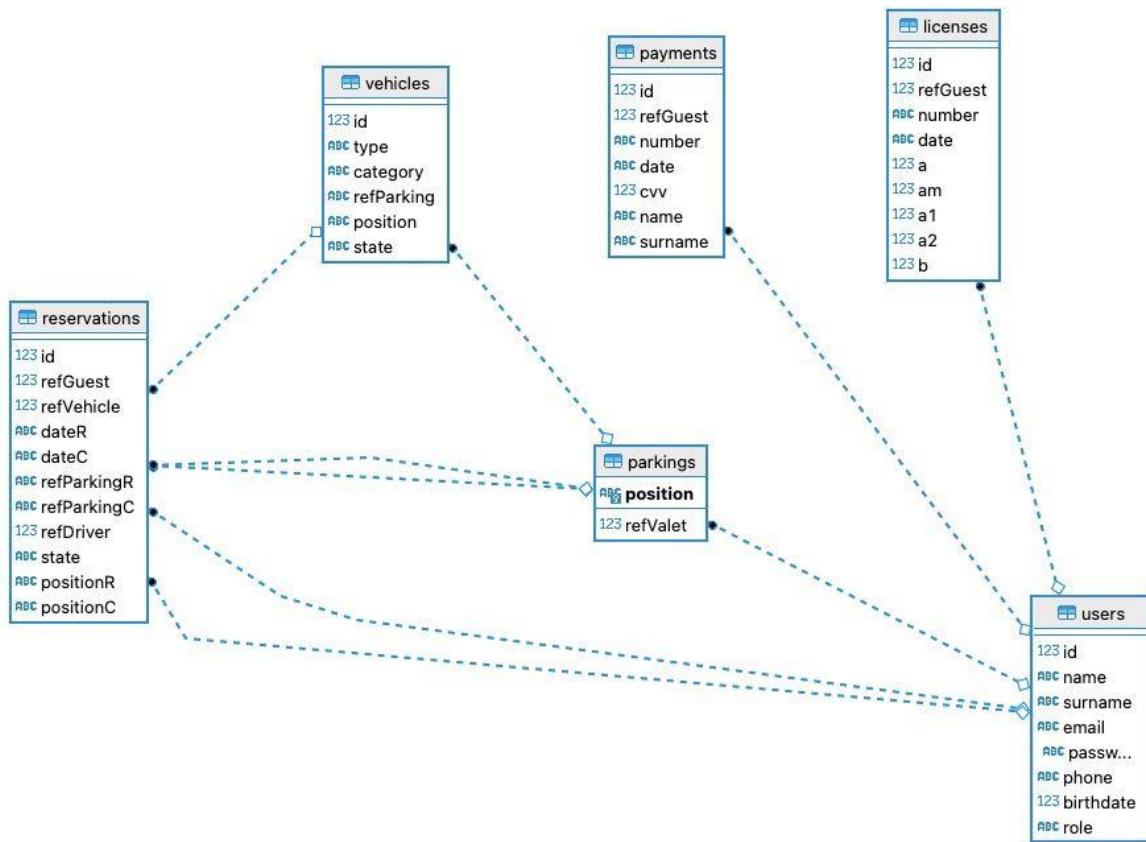
sono stati loro designati.

In particolare, sul client, risiederanno quattro diverse tipologie di sottosistemi, differenziate dal ruolo che l'utente riveste nel sistema.

A sinistra è rappresentato il deployment diagram con i componenti suddivisi per ogni nodo.

3.4 Progetto del database

3.4.1 Modello ER



3.4.2 Descrizione delle tabelle

Users

Nome attributo	Tipo	Vincoli	Descrizione
"id"	INTEGER	NOT NULL, UNIQUE	codice identificativo univoco dell'utente
"name"	TEXT	NOT NULL	nome dell'utente
"surname"	TEXT	NOT NULL	cognome dell'utente
"email"	TEXT	NOT NULL, UNIQUE	email dell'utente
"password"	TEXT	NOT NULL	password dell'utente
"phone"	TEXT	NOT NULL	numero di telefono dell'utente
"birthdate"	TEXT	NOT NULL	data di nascita dell'utente

"role"	TEXT	NOT NULL, DEFAULT 'guest', CHECK("role" IN ('guest', 'admin', 'valet', 'driver'))	ruolo dell'utente all'interno del sistema
--------	------	---	---

Altri vincoli:

- PRIMARY KEY("id" AUTOINCREMENT).

Parkings

Nome attributo	Tipo	Vincoli	Descrizione
"position"	TEXT	NOT NULL, UNIQUE	via in cui si trova il parcheggio
"refValet"	INTEGER	NOT NULL	id del parcheggiatore assegnato allo specifico parcheggio

Altri vincoli:

- PRIMARY KEY("position").
- FOREIGN KEY ("refValet") REFERENCES "users" ("id").

Payments

Nome attributo	Tipo	Vincoli	Descrizione
"id"	INTEGER	NOT NULL, UNIQUE	codice identificativo univoco del metodo di pagamento
"refGuest"	INTEGER	NOT NULL	riferimento all'id dello user a cui è associato il metodo di pagamento
"number"	TEXT	NOT NULL, UNIQUE	numero della carta
"date"	TEXT	NOT NULL	data di scadenza della carta
"cvv"	INTEGER	NOT NULL	cvv della carta
"name"	TEXT	NOT NULL	nome intestatario della carta (può non coincidere con il nome dell'utente avente come id il refGuest)
"surname"	TEXT	NOT NULL	cognome intestatario della carta (può non coincidere con il cognome dell'utente avente come id il refGuest)

Altri vincoli:

- FOREIGN KEY("refGuest") REFERENCES "users"("id") ON DELETE CASCADE ON UPDATE CASCADE.
- PRIMARY KEY("id" AUTOINCREMENT).

Vehicles

Nome attributo	Tipo	Vincoli	Descrizione
"id"	INTEGER	NOT NULL, UNIQUE	codice identificativo univoco del veicolo
"type"	TEXT	CHECK("type" IN ("electric scooter", "scooter", "car", "bicycle"))	tipologia del veicolo
"category"	TEXT	CHECK("category" IN ("sedan", "utilitaire", "suv"))	categoria dell'auto, null se il type del veicolo non è 'car'
"refParking"	TEXT		riferimento al parcheggio in cui si trova il veicolo, null se il veicolo è fuori stallo
"position"	TEXT		via in cui si trova il veicolo, null se il veicolo si trova in un parcheggio
"state"	TEXT	NOT NULL, DEFAULT 'available' CHECK("state" IN ("in use", "damage", "available"))	stato del veicolo

Altri vincoli:

- PRIMARY KEY("id" AUTOINCREMENT).
- FOREIGN KEY("refParking") REFERENCES "parkings"("position").

Licenses

Nome attributo	Tipo	Vincoli	Descrizione
"id"	INTEGER	NOT NULL, UNIQUE	codice identificativo univoco della patente
"refGuest"	INTEGER	NOT NULL, UNIQUE	riferimento all'id dello user a cui è associata la patente
"number"	TEXT	NOT NULL, UNIQUE	numero della patente
"date"	TEXT	NOT NULL	data di scadenza della patente
"a"	INTEGER	NOT NULL, DEFAULT 0 CHECK("a" IN (0, 1))	categoria della patente: 0 non posseduta, 1 posseduta

"am"	INTEGER	NOT NULL, DEFAULT 0 CHECK("am" IN (0, 1))	categoria della patente: 0 non posseduta, 1 posseduta
"a1"	INTEGER	NOT NULL, DEFAULT 0 CHECK("a1" IN (0, 1))	categoria della patente: 0 non posseduta, 1 posseduta
"a2"	INTEGER	NOT NULL, DEFAULT 0 CHECK("a2" IN (0, 1))	categoria della patente: 0 non posseduta, 1 posseduta
"b"	INTEGER	NOT NULL, DEFAULT 0 CHECK("b" IN (0, 1))	categoria della patente: 0 non posseduta, 1 posseduta

Altri vincoli:

- PRIMARY KEY("id" AUTOINCREMENT).
- FOREIGN KEY("refGuest") REFERENCES "users"("id").

Reservations

Nome attributo	Tipo	Vincoli	Descrizione
"id"	INTEGER	NOT NULL, UNIQUE	codice identificativo univoco della prenotazione
"refGuest"	INTEGER	NOT NULL	riferimento all'id dello user a cui è associata la prenotazione
"refVehicle"	INTEGER	NOT NULL	riferimento all'id del veicolo prenotato
"dateR"	TEXT	NOT NULL	data e ora di ritiro del veicolo
"dateC"	TEXT	NOT NULL	data e ora di consegna del veicolo
"refParkingR"	TEXT		riferimento alla posizione del parcheggio di ritiro, null nel caso di prenotazione di veicoli fuori dallo stallo
"refParkingC"	TEXT	NOT NULL	riferimento alla posizione del parcheggio di consegna
"price"	REAL	NOT NULL	costo della prenotazione in euro
"refDriver"	INTEGER		riferimento all'id dell'autista che ha preso in carico la prenotazione, non è null solo se il cliente ha richiesto una prenotazione con autista
"state"	TEXT	NOT NULL, DEFAULT 'confirmed' CHECK("state" IN ('confirmed', 'not confirmed', 'withdrawn', 'late delivery'))	descrive lo stato della prenotazione: confermata, non confermata, ritirata o consegna in ritardo

"positionR"	TEXT		posizione in cui viene ritirato il mezzo, non è null nel caso di prenotazioni di veicoli fuori dallo stallo
"positionC"	TEXT		posizione in cui viene consegnato il mezzo, non è null nel caso in cui il cliente per non incorrere in sovrapprezzi decida di lasciare il mezzo in un posto diverso da uno dei parcheggi in parkings

Altri vincoli:

- PRIMARY KEY("id" AUTOINCREMENT).
- FOREIGN KEY("refDriver") REFERENCES "users"("id") ON DELETE CASCADE ON UPDATE CASCADE.
- FOREIGN KEY("refGuest") REFERENCES "users"("id") ON DELETE CASCADE ON UPDATE CASCADE.
- FOREIGN KEY("refParkingR") REFERENCES "parkings"("position") ON DELETE CASCADE ON UPDATE CASCADE.
- FOREIGN KEY("refParkingC") REFERENCES "parkings"("position") ON DELETE CASCADE ON UPDATE CASCADE.
- FOREIGN KEY("refVehicle") REFERENCES "vehicles"("id") ON DELETE CASCADE ON UPDATE CASCADE

3.5 Accesso al sistema

Per poter usufruire di qualsiasi tipo di servizio offerto dal sistema è necessario che l'utente sia autenticato, ossia abbia effettuato il login inserendo le credenziali -email e password- relative al proprio account.

Nel caso in cui un utente abbia dimenticato la password per poter accedere al proprio account, il sistema ha previsto la possibilità di recuperarla, infatti verrà inviata una mail all'indirizzo fornito dall'utente stesso nella fase di recupero contenente una nuova password che rispetti i criteri di sicurezza stabiliti dal sistema.

Ogni tipologia di utente -guest, driver, admin e valet- visualizzerà un'interfaccia grafica differente, affinché l'utente possa fruire delle funzionalità che il sistema ha designato alla tipologia a cui appartiene.