

System Design Document

Progetto: PalerMove

Gruppo: Ctrl C Ctrl V

Angelo Ribaudò, Emanuele Muzio, Dario Bologna, Gabriele Fraterrigo.

Sommario

1. Obiettivi Generali
2. Obiettivi di progettazione
3. Architettura Software Proposta
 - 3.1. Panoramica
 - 3.2. Decomposizione in sottosistemi
 - 3.2.1. Mappatura degli oggetti nei sottosistemi
 - 3.3. Mappatura Hardware / Software
 - 3.4. Progetto del DB
 - 3.4.1. Modello EER
 - 3.4.2. Descrizione delle tabelle
 - 3.5. Accesso al sistema

1 Obiettivi Generali

L'obiettivo del sistema è quello di agevolare al massimo la gestione del servizio di car sharing da parte degli amministratori e la prenotazione di un mezzo da parte degli utenti.

Inoltre, è di fondamentale importanza rendere questo servizio accessibile a tutti e sicuro, mettendo a disposizione dei clienti la possibilità di usufruire di un servizio autista e verificando che l'utente abbia una patente in corso di validità per il veicolo che intende prenotare.

2 Obiettivi di progettazione

- Il sistema deve essere compatibile con le ultime versioni dei browser web più utilizzati (Google Chrome, Mozilla Firefox, Opera e Safari).
- Il sistema deve garantire la correttezza dei dati inseriti, nonché la sicurezza nel trattamento di questi, in particolare quelli legati a metodi di pagamento e patente
- Il sistema deve tenere correttamente traccia delle prenotazioni e delle modifiche che gli utenti effettuano su queste

3 Architettura software proposta

3.1 Panoramica

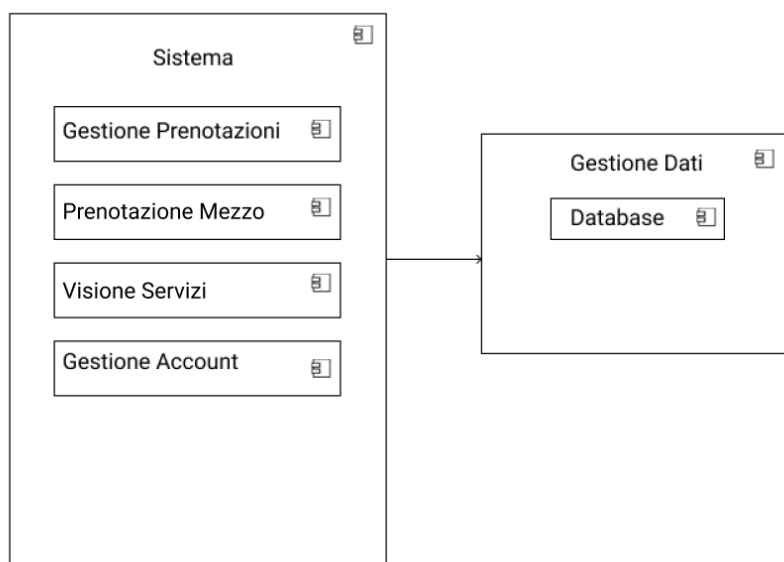
Gli utenti accedono al sistema inviando le proprie credenziali a un nodo centrale, che contiene tutti i dati d'interesse.

Si è, pertanto, deciso di scegliere un'architettura Client-Server.

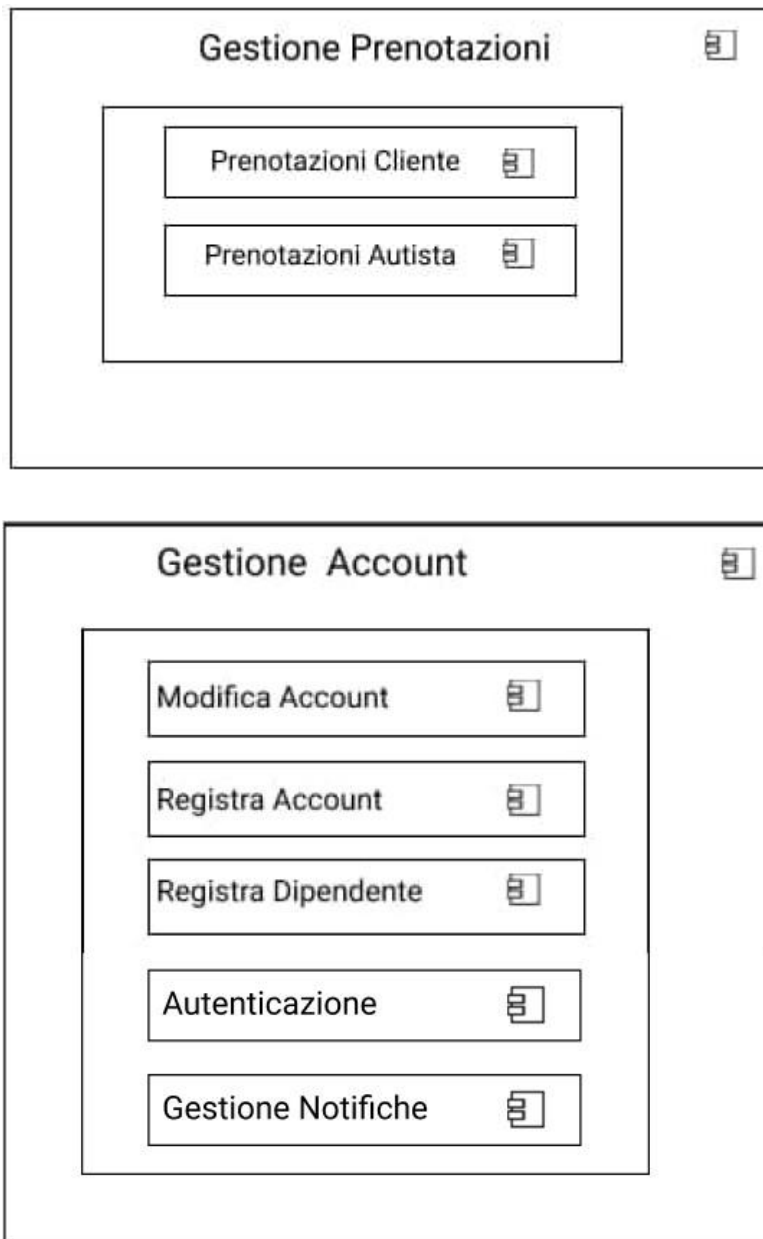
I nodi client rappresentano i device che vengono usati per connettersi e avvalersi del servizio offerto dal sistema.

Per la connessione e la comunicazione tra i nodi viene utilizzato il protocollo HTTP.

3.2 Decomposizione in sottoinsiemi



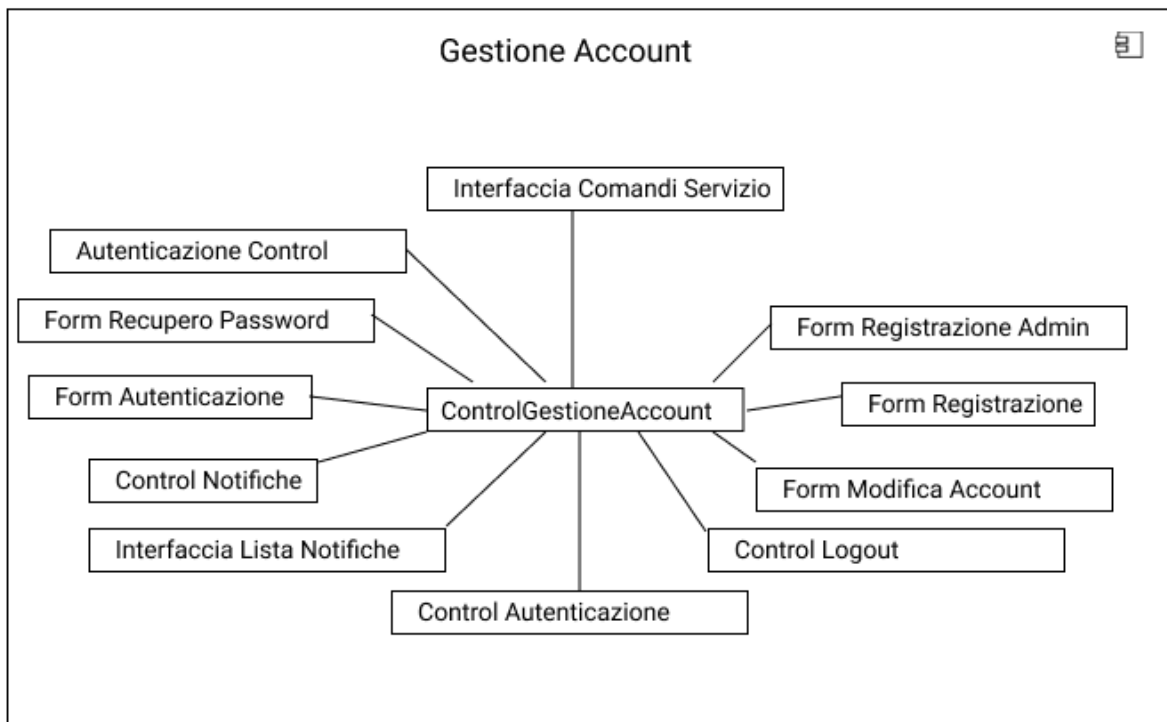
Nel diagramma UML mostrato nella pagina precedente, vengono mostrati i sottosistemi che compongono il sistema proposto. Si è inoltre deciso, al fine di rendere più chiara la comprensione di alcuni sottosistemi, di scomporli in sottosistemi .



Data l'architettura scelta in fase di progettazione, le operazioni di comunicazione con il DBMSBoundary sono state racchiuse nel nodo server.

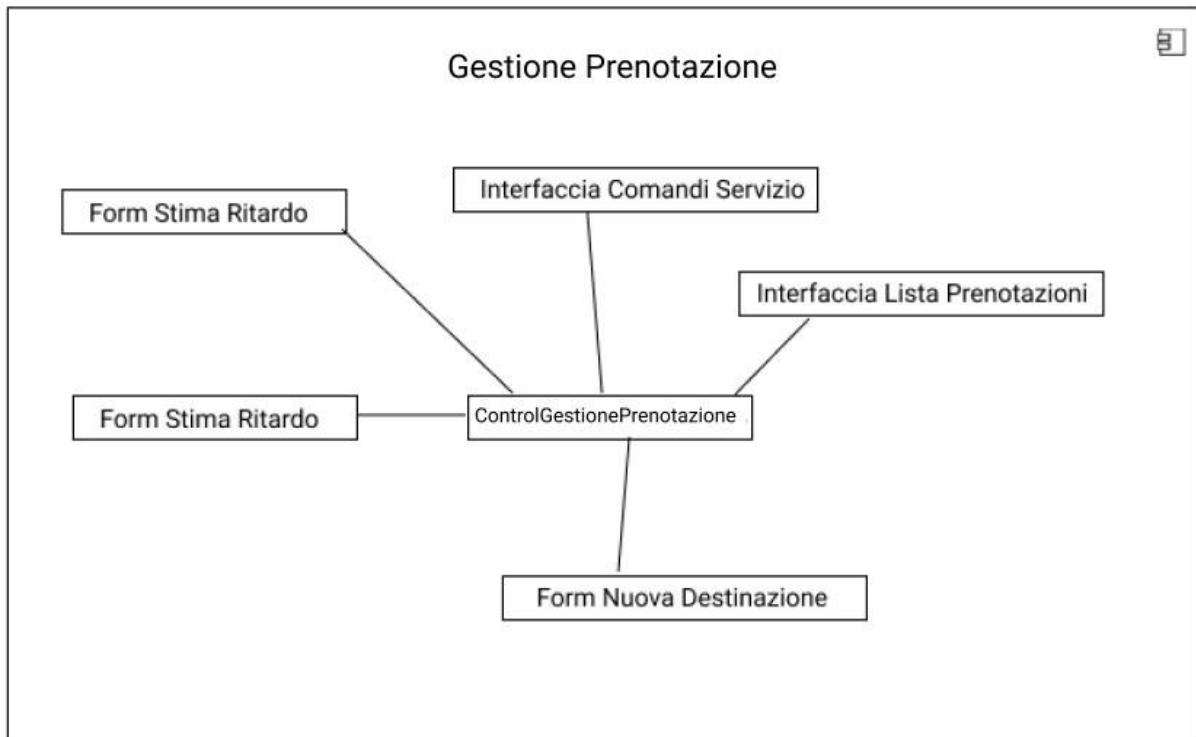
3.2.1 Mappatura degli oggetti nei sottosistemi

3.2.1.1 Gestione Account



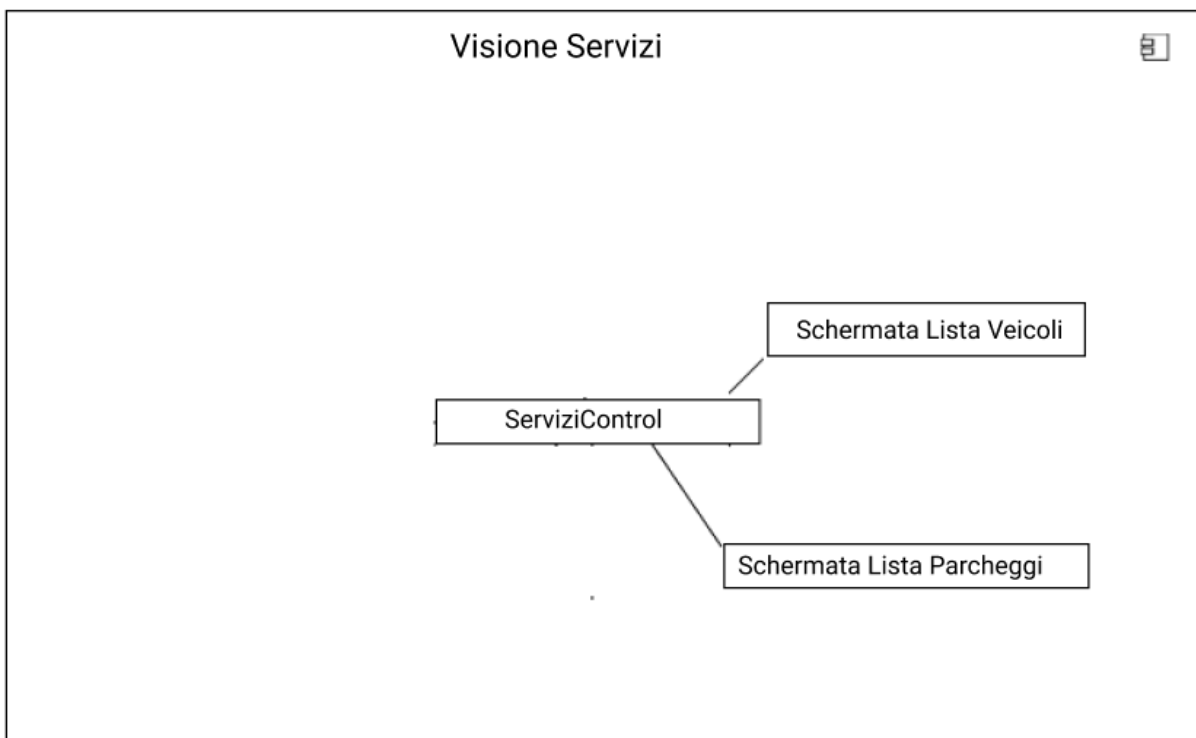
ControlGestioneAccount include ControlModificaAccount, ControlRegistrazione e ControlRegistrazioneAmministratore.

3.2.1.2 Gestione Prenotazioni

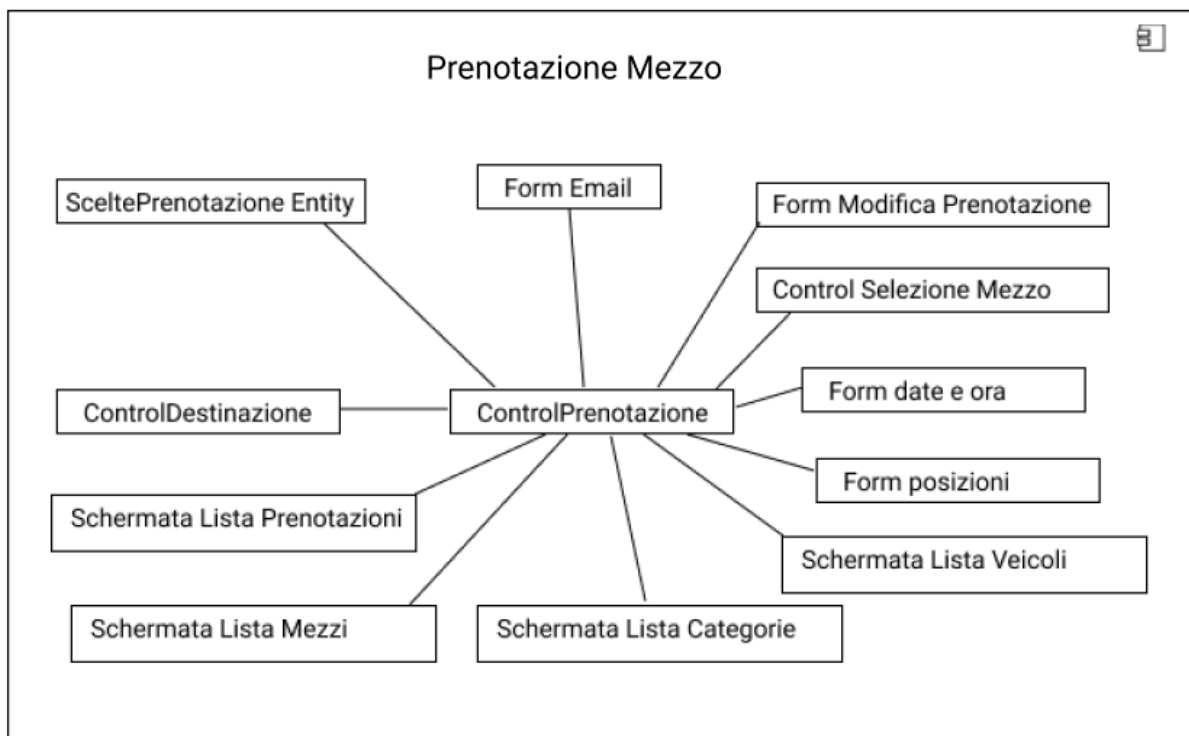


ControlGestioneServizio include ControlTerminaServizioCliente, ControlRitiro e ControlConsegnaMezzo

3.2.1.3 Visione Servizi



3.2.1.4 Gestione Struttura



ControlPrenotazione comprende ControlEliminaPrenotazione, ControlModificaPrenotazione, ControlModificaPrenotazione2, ControlCreaPrenotazione.

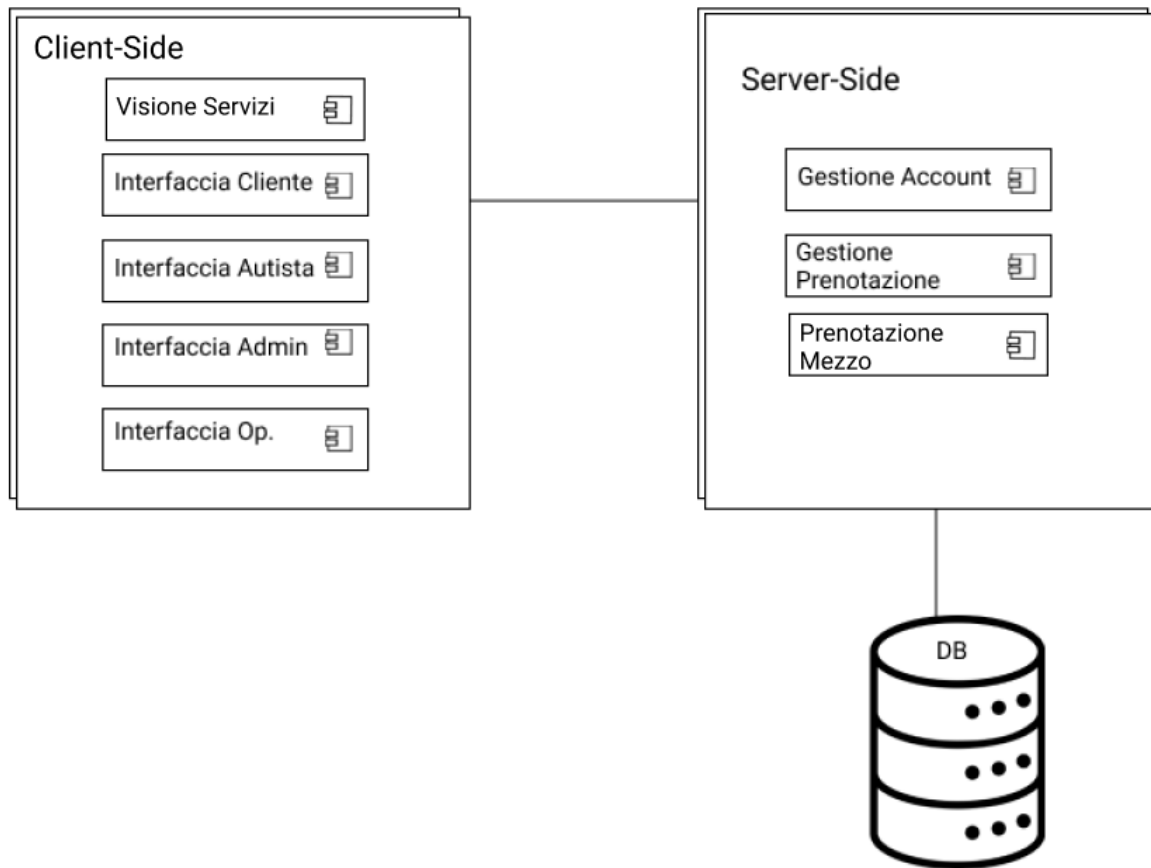
3.3 Mappatura Hardware / Software

La struttura hardware proposta segue il modello client-server. E' costituita da un server centrale e dai dispositivi degli utenti.

La scelta di un'architettura client-server è dettata dalla necessità di condividere una serie di servizi da un server centrale verso i dispositivi degli utenti.

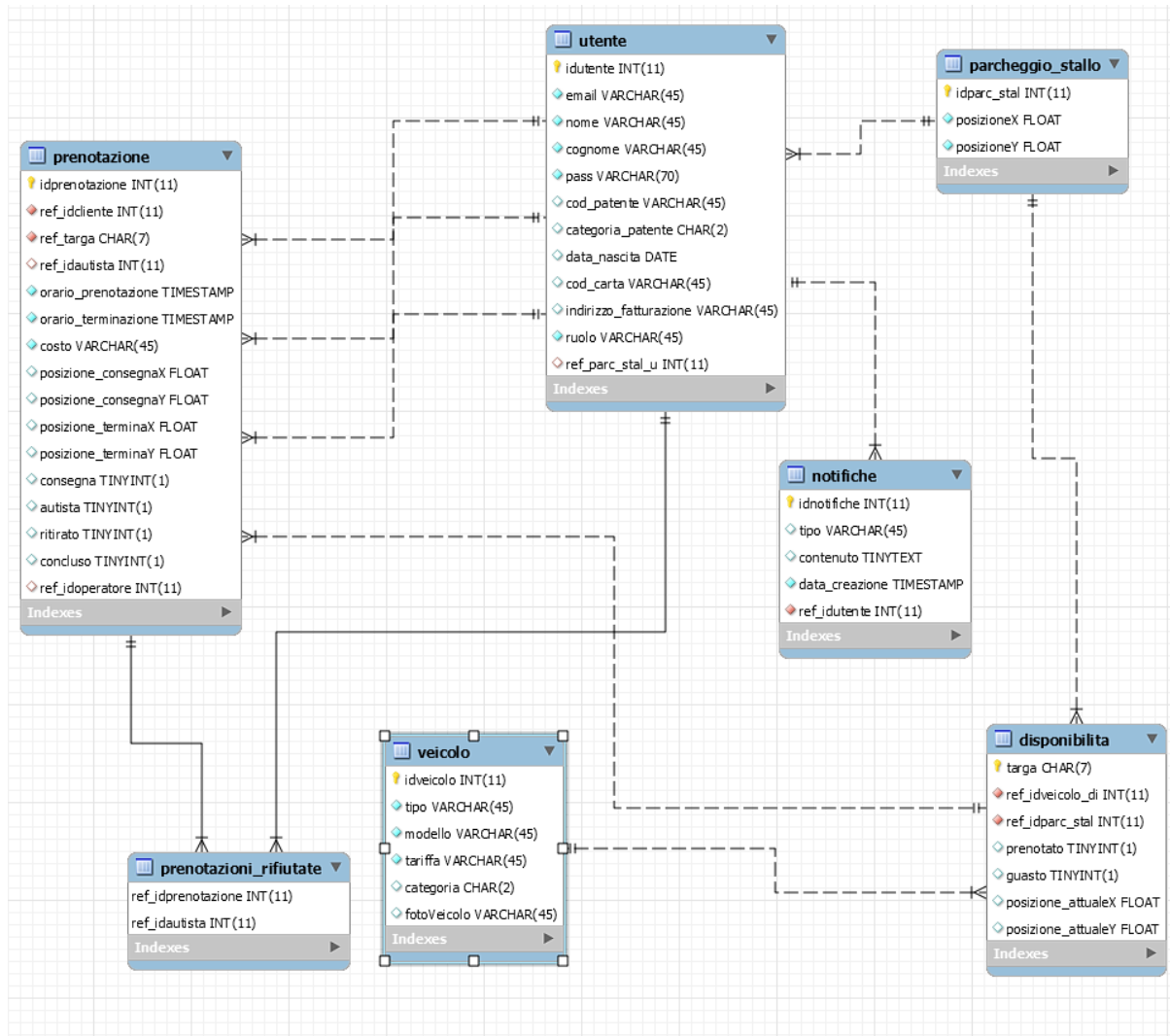
L'utilizzo di questa architettura permette una risposta immediata alle richieste dell'utente, che da parte sua deve soltanto interagire con l'interfaccia client proposta. Sui dispositivi client ogni utente avrà accesso ai sottosistemi relativi al proprio ruolo(cliente, amministratore, autista, operatore).

A seguire viene raffigurato il diagramma di deployment con i componenti suddivisi per ogni nodo.



3.4 Progetto del DB

3.4.1 Modello EER



3.4.2 Descrizione delle tabelle

Disponibilità

Attributo	Datatype	Foreign Key	Default/Expression
<u>targa</u>	CHAR(7)	--	NOT NULL
ref_idveicolo_di	INT(11)	Veicolo (idveicolo)	NOT NULL
ref_idparc_stal	INT(11)	Parcheggio (idparch_stall)	NOT NULL
prenotato	TINYINT(1)	--	'0'
guasto	TINYINT(1)	--	'0'
posizione_attualeX	FLOAT	--	NULL
posizione_attualeY	FLOAT	--	NULL

Notifiche

Attributo	Datatype	Foreign Key	Default/Expression
<u>idnotifiche</u>	INT(11)	--	NOT NULL
tipo	VARCHAR(45)	--	NULL
contenuto	TINYTEXT	--	CURRENT_TIMESTAMP
data_creazione	TIMESTAMP	--	NOT NULL
ref_idutente	INT(11)	Utente (idutente)	NOT NULL

Parcheggio_Stallo

Attributo	Datatype	Foreign Key	Default/Expression
<u>idparc_stall</u>	INT(11)	--	NOT NULL
posizioneX	FLOAT	--	NOT NULL
posizioneY	FLOAT	--	NOT NULL

Prenotazioni_Rifiutate

Attributo	Datatype	Foreign Key	Default/Expression
<u>ref_idprenotazione</u>	INT(11)	Prenotazione (idprenotazione)	NOT NULL
<u>ref_idautista</u>	INT(11)	Utente (idutente)	NOT NULL

Prenotazione

Attributo	Datatype	Foreign Key	Default/Expression
<u>idprenotazione</u>	INT(11)	--	NOT NULL
ref_idcliente	INT(11)	Utente (idutente)	NOT NULL
ref_targa	CHAR(7)	Disponibilita (targa)	NOT NULL
ref_idautista	INT(11)	Utente (idutente)	NULL
orario_prenotazione	TIMESTAMP	--	NOT NULL
orario_terminazione	TIMESTAMP	--	NOT NULL
costo	VARCHAR(45)	--	NOT NULL

posizione_consegnaX	FLOAT	--	NULL
posizione_consegnaY	FLOAT	--	NULL
posizione_terminaX	FLOAT	--	NULL
posizione_terminaY	FLOAT	--	NULL
consegna	TINYINT(1)	--	NULL
autista	TINYINT(1)	--	NULL
ritirato	TINYINT(1)	--	NULL
concluso	TINYINT(1)	--	NULL
ref_idoperatore	INT(11)	Utente (idutente)	NULL

Utente

Attributo	Datatype	Foreign Key	Default/Expression
<u>idutente</u>	INT(11)	--	NOT NULL
email	VARCHAR(45)	--	NOT NULL
nome	VARCHAR(45)	--	NOT NULL
cognome	VARCHAR(45)	--	NOT NULL
pass	VARCHAR(70)	--	NOT NULL
cod_patente	CHAR(2)	--	NULL
data_nascita	DATE	--	NULL
cod_carta	VARCHAR(45)	--	NULL
indirizzo_fatturazione	VARCHAR(45)	--	NULL

ruolo	VARCHAR(45)	--	
ref_parc_stal_u	INT(11)	Parcheggio (idparc_stal)	NOT NULL

3.5 Accesso al sistema

Tutti gli utenti possono accedere alla visione dei servizi, che siano autenticati o meno. Gli utenti possono autenticarsi inserendo le proprie credenziali (e-mail e password) dopo essersi registrati (i clienti) o essere stati registrati da un amministratore (operatori e autisti). Una volta autenticati, vengono riconosciuti secondo 4 tipologie di utente (amministratore, operatore, autista, cliente). Ogni tipologia di utente potrà accedere alle interfacce grafiche, che gli permettono di interagire con le funzioni che gli competono.