



**Università
degli Studi
di Palermo**

CdL: Ingegneria dell'innovazione per le imprese digitali

Corso: PROGETTAZIONE DEL SOFTWARE

Docente: Valeria Seidita

A.A. 2020/21

SDD: System Design Document

Progetto:



Tesina a cura di:

Sciacchitano Antonio Pio - 0684157

Scroppo Federico - 0667708

Seminara Giuseppe - 0667494

Tortomasi Gianvito – 0679722

System Design Document

Indice

1. [Obiettivi Generali](#)
 - 1.1. Acronimi
 - 1.2. Definizioni
2. [Architettura Software Corrente](#)
3. [Architetture Software Proposta](#)
 - 3.1. [Scomposizione Del Sistema In Sottosistemi](#)
 - 3.2. [Hardware/Software Mapping](#)
 - 3.3. [Sicurezza E Controllo Degli Accessi](#)
 - 3.4. [Controllo Software Globale](#)
 - 3.5. [Accesso Al Sistema](#)
 - 3.6. [Boundary Conditions](#)
4. [Gestione Dati Persistente](#)
 - 4.1. [Diagramma Entità Relazione](#)
 - 4.2. [Descrizione Delle Tabelle](#)
 - 4.3. [Diagramma Entità Relazione con MySQL Workbench](#)
5. [Messaggi D'errore](#)

1. Obiettivi Generali

L'obiettivo generale del sistema è quello di supportare gli utilizzatori, durante il processo di noleggio dei veicoli, cercando di rendere efficienti le operazioni e lo scambio di informazioni tra i vari utenti che interagiscono all'interno del sistema. Tra i principali obiettivi di progettazione:

- **Facilità di utilizzo:** l'Utente potrà sfruttare tutte le funzionalità del sistema nel minor numero possibile di passaggi;
- **Affidabilità:** i risultati prodotti dalle pagine saranno costantemente aggiornati relativamente alle disponibilità dei veicoli, e delle tempistiche di noleggio in modo da permettere ai vari utenti di non incorrere in errori di sistema ed organizzazione;
- **Sicurezza:** i dati personali non verranno diffusi al di fuori del sistema o utilizzati per altri scopi, così da garantire agli utenti la riservatezza dei dati, secondo le attuali normative sulla privacy;
- **Robustezza:** il sistema, tramite opportuni controlli, impedirà l'inserimento da parte degli utenti di input non validi;
- **Portabilità:** il sistema è portatile in diverse piattaforme, in quanto realizzato in linguaggio Java;
- **Estendibilità:** data la sua versatilità, il sistema potrà essere esteso e modificato in futuro, secondo le esigenze richieste.

1.1. Acronimi

- **DBMS:** Database Management System;
- **UML:** Unified Modelling Language;
- **SDD:** System Design Document;
- **TCP:** Transmission Control Protocol;
- **IP:** Internet Protocol;
- **SMTP:** Simple Mail Transfer Protocol;
- **JVM:** Java Virtual Machine;
- **JDBC:** Java Database Connectivity;

1.2. Definizioni

- **Java:** è un linguaggio di programmazione ad alto livello orientato agli oggetti;
- **Java Virtual Machine:** è una componente della piattaforma Java che esegue i programmi tradotti in bytecode;
- **Java Database Connectivity:** è un connettore per database che consente l'accesso e la gestione della persistenza dei dati sulle basi di dati da parte di un qualsiasi programma scritto in Java;
- **MySQL:** è un relational database management system composto da un client a riga di comando e un server;
- **SHA256:** algoritmo utilizzato per la generazione di un hash relativo alle password;

2. Architettura Software Corrente

Come già esposto nella fase di analisi dei requisiti, si suppone che nel campo della mobilità non venga utilizzato alcun software in grado di compiere le complesse operazioni automatiche proposte dal nostro sistema, le quali sono al momento svolte manualmente degli impiegati

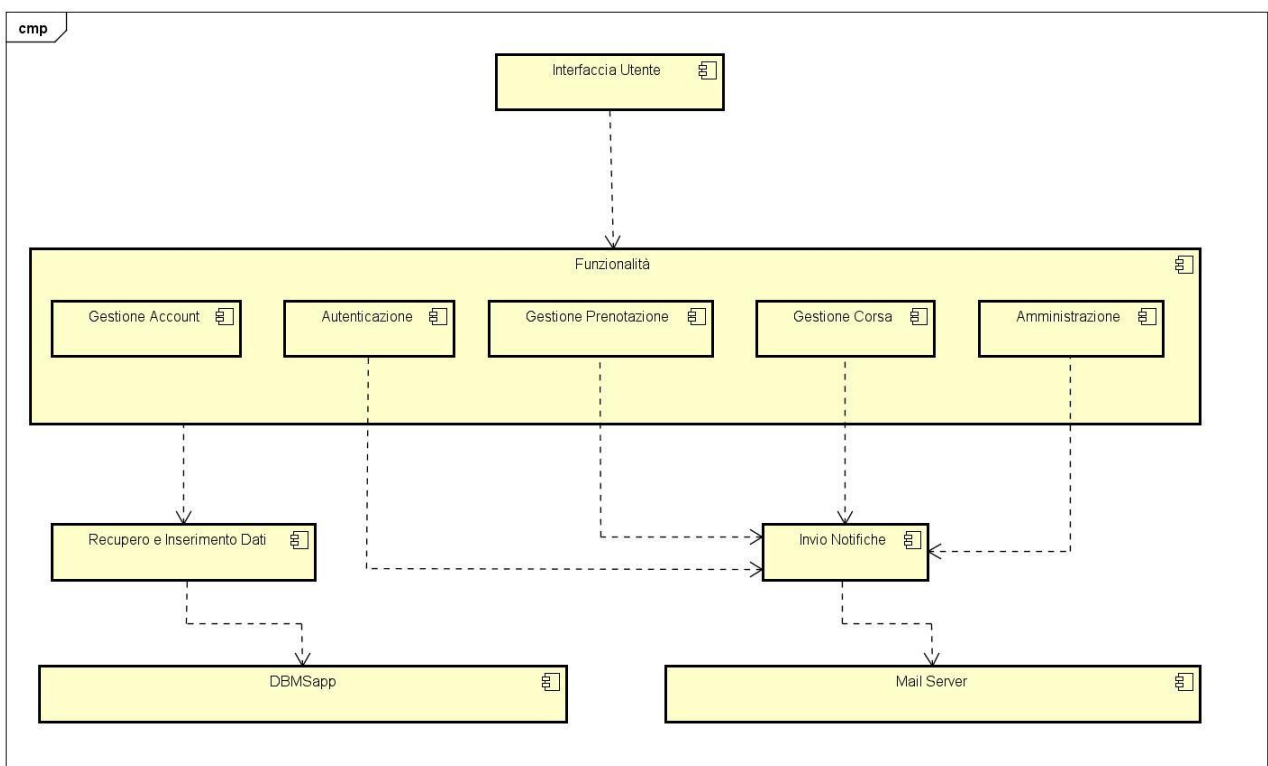
3. Architetture Software Proposta

Per il sistema proposto, il tipo di architettura software che si è scelta è legata alla fase di analisi dei requisiti, pertanto è stata presa in considerazione un'architettura a layer. Introducendo una suddivisione in più livelli organizzati in modo tale da fornire specifici servizi in base al loro posizionamento. La gerarchia dei sottosistemi, inoltre, diminuisce la complessità del sistema.

Il linguaggio di programmazione proposto è Java, mentre il database è realizzato tramite MySQL. La comunicazione tra il codice ed il database è ottenuta col supporto della classe Java JDBC (Java Database Connectivity).

3.1. Scomposizione Del Sistema In Sottosistemi

Al fine di mostrare in modo più semplice la suddivisione del sistema, il seguente diagramma delle componenti è stato generato come segue:

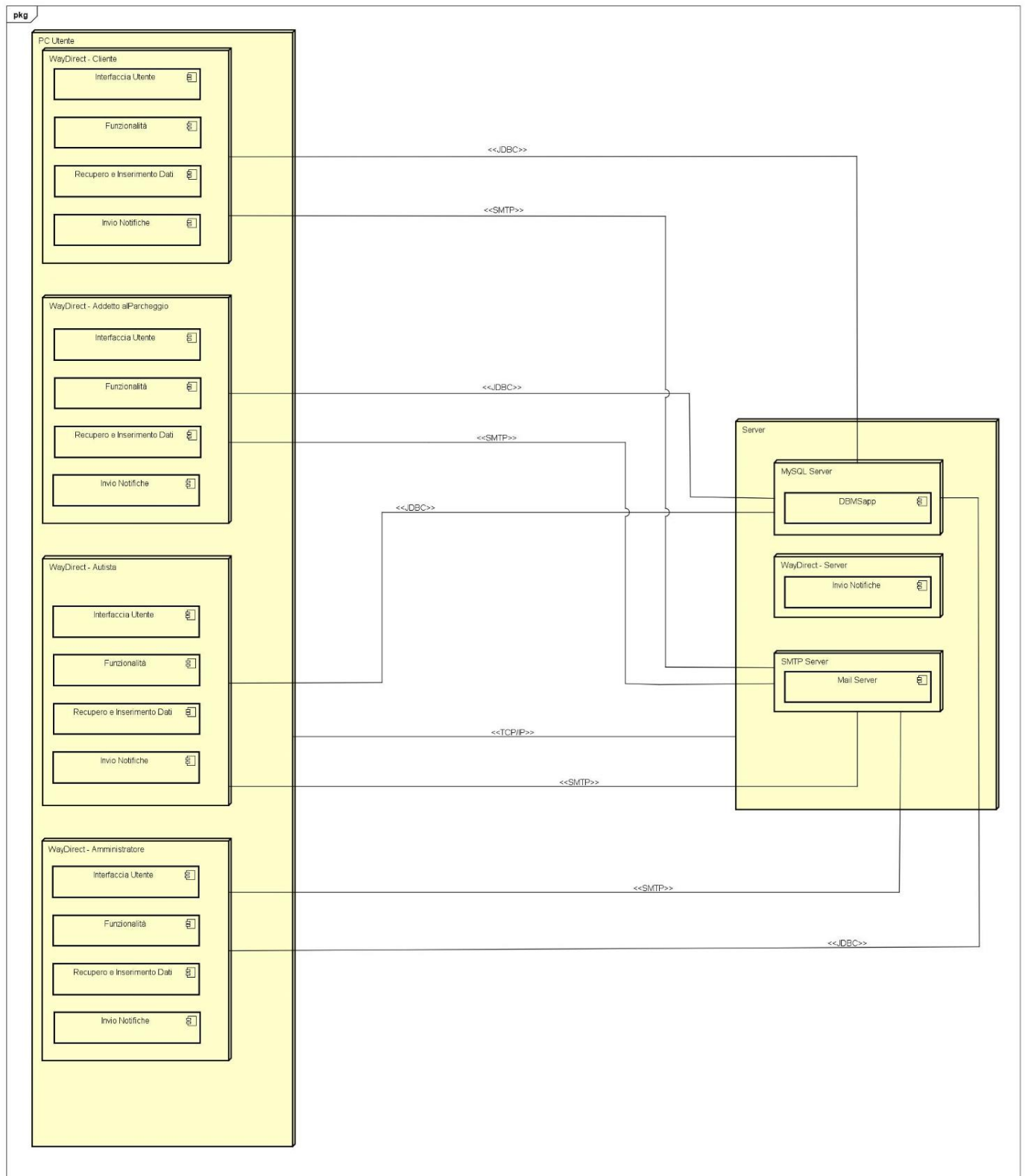


- **Interfaccia Utente:** insieme delle interfacce grafiche tramite le quali, l'Utente interagisce con il sistema;
- **Funzionalità:** il sistema è stato suddiviso in sottosistemi più semplici, ognuno dei quali gestisce la logica di funzionamento di uno dei casi d'uso della vista d'insieme individuati in fase di analisi dei requisiti;
- **Invio Notifiche:** il layer che si occupa di gestire la connessione con il mail server;
- **Recupero e inserimento dati:** gestisce le connessioni col database e si occupa anche della conversione tra il formato in cui i dati sono rappresentati nel database e le entity del sistema.

3.2. Hardware/Software Mapping

Nella mappatura hardware/software, il team ha deciso di concentrare le complessità del sistema sui calcolatori degli utenti al fine di permettere migliore scalabilità. Come si nota dal diagramma riportato in seguito, sono presenti due nodi fondamentali: Pc Utente e Server.

Entrambi i nodi rappresentano dei dispositivi fisici: il primo è un qualsiasi computer personale adibito all'installazione del sistema proposto, mentre il secondo è il Server.



- **Nodo PC Utente:**

All'interno di questo nodo, i componenti "WayDirect – Cliente", "WayDirect – Addetto al Parcheggio", "WayDirect – Autista" e "WayDirect – Amministratore", rappresentano i nodi software contenenti:

- Layer di interfaccia utente;
- Funzionalità e connessioni (database e mail server).

Inoltre, questi nodi sono utilizzabili ed installabili in qualsiasi calcolatore in cui sia presente una Java Virtual Machine.

- **Nodo Server:**

All'interno di questo nodo, oltre al componente software adibito all'invio di notifiche (WayDirect – Server) sono presenti anche un'istanza del "MySQL Server", che gestirà i contenuti del database. Ed un "SMTP Server", che gestirà l'invio dei messaggi di posta elettronica in base alle richieste dei nodi software.

Queste richieste saranno gestite mediante i rispettivi protocolli: TCP/IP (livello trasporto e rete) ed SMTP (livello applicativo).

3.3. Sicurezza E Controllo Degli Accessi

Vista la natura sensibile dei dati gestiti dal sistema, per poter usufruire dei servizi proposti è necessario eseguire una procedura di autenticazione. L'accesso al sistema è consentito tramite l'inserimento di una password e di vari identificativi che daranno la possibilità all'utente di accedere alle diverse aree del sistema. Le password in fase di autenticazione e registrazione vengono crittografate tramite l'algoritmo SHA256 (password hash) per evitare di salvare le stesse in chiaro all'interno del database. Inoltre, alcune funzionalità del sistema saranno accessibili solo ed esclusivamente da determinate tipologie di utente.

3.4. Controllo Software Globale

Quando un utente si autentica, vi è un accesso al database, tramite una query, che permette di controllare l'esistenza dello stesso. Dopo la conferma, l'utente potrà accedere a diverse operazioni messe a disposizione dal sistema, accessibili tramite un'interfaccia grafica. Il controllo del flusso principale, viene effettuato dal Server MySQL, che si occupa di gestire gli accessi concorrenti da parte di più utenti.

Ad esempio, nella fase di creazione di una nuova prenotazione da parte di un cliente, MySQL entra in una sezione critica gestendo la concorrenza ed evitando che più clienti scelgano le stesse caratteristiche di prenotazione (veicolo, orario etc...).

3.5. Accesso Al Sistema

L'accesso al sistema è permesso mediante l'inserimento, nell'apposita Schermata di Accesso, di un indirizzo email ed una password registrata nel database, nella colonna "Utente". In questo modo, l'accesso è permesso solo ai membri precedentemente registrati e riconoscibili.

3.6. Boundary Conditions

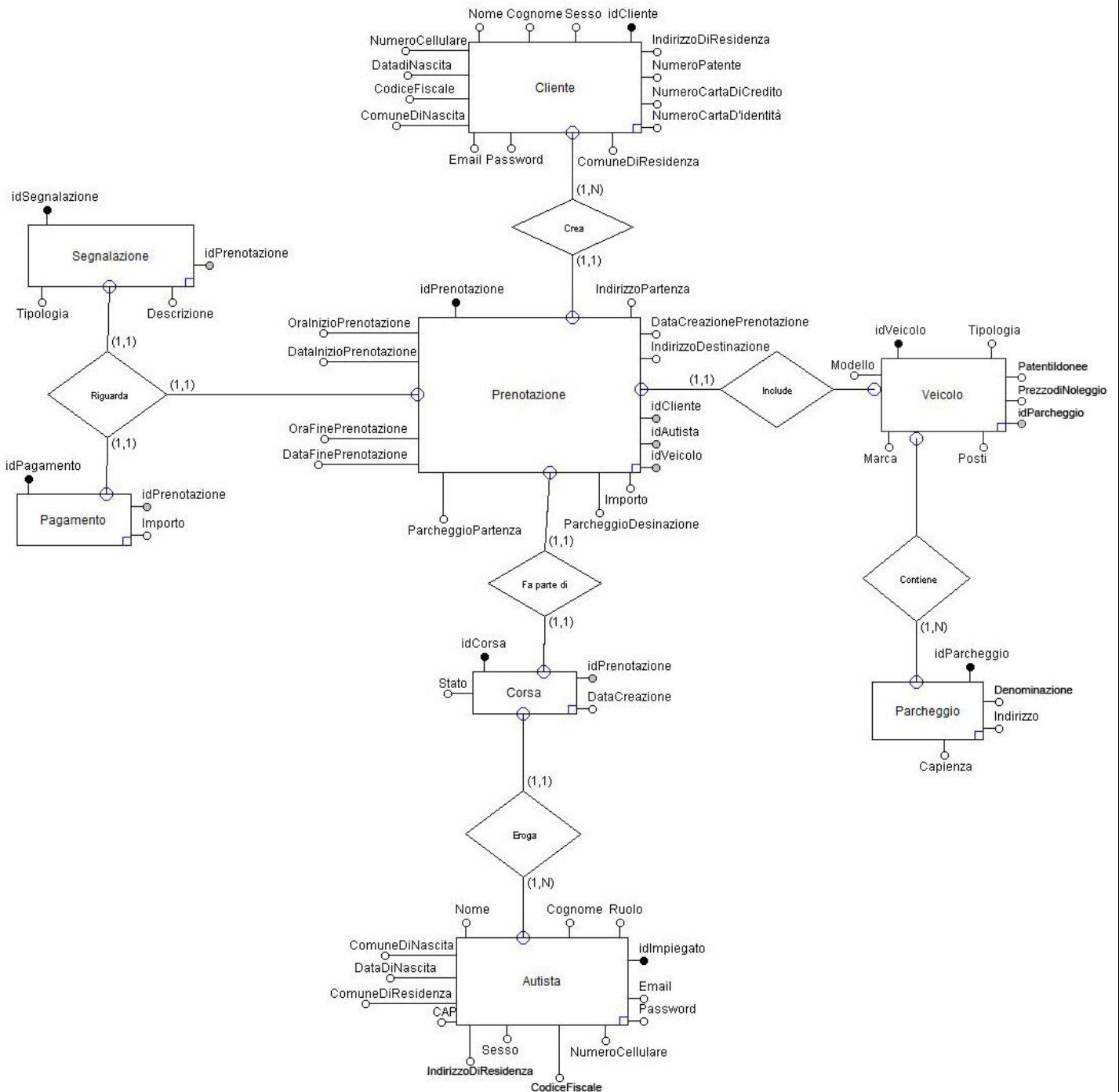
Per permettere all'utente di usufruire delle funzionalità del sistema, deve esserci una connessione fisica, come ad esempio una LAN wireless, e data la facilità di utilizzo, non sono necessari ulteriori passaggi di configurazione.

- **Accensione:** dal momento in cui il sistema verrà avviato per la prima volta, rimarrà attivo per un valore di default di ore, senza mai terminare;
- **Spegnimento:**
 - Il sistema lato server, non terminerà la sua esecuzione a meno che non vi siano dei guasti;
 - Dal punto di vista dell'utilizzatore, invece, terminerà la propria esecuzione solo quando verrà chiuso il sistema dal proprio dispositivo;
- **Eccezioni:**
 - Il sistema lato server potrà non essere funzionante, in caso di problemi hardware, attacchi esterni o mancanza di elettricità;
 - Dal punto di vista dell'utilizzatore, il sistema potrà non essere funzionante, nel caso in cui si verifichi una caduta della connessione di rete.

4. Gestione Dati Persistente

4.1. Diagramma Entità Relazione

In seguito alla fase di raccolta e di analisi dei requisiti, è stata adottata una strategia mista, utilizzando le tecniche di bottom up e top down, per la costruzione del diagramma ER (entità-relazione), eliminando eventuali ridondanze ed introducendo gli identificatori primati, e che risulta così composto:



4.2. Descrizione Delle Tabelle

4.2.1. Tabella Cliente

Field	Type	Null	Key	Default	Extra
idUtente	int	NO	PRI	NULL	auto_increment
Email	varchar(45)	NO		NULL	
Password	char(64)	NO		NULL	
Nome	varchar(45)	NO		NULL	
Cognome	varchar(45)	NO		NULL	
ComuneDiNascita	varchar(45)	NO		NULL	
CodiceFiscale	char(16)	NO		NULL	
Sesso	varchar(45)	NO		NULL	
DataDiNascita	date	NO		NULL	
IndirizzoDiResidenza	varchar(45)	NO		NULL	
CAP	char(5)	NO		NULL	
TipologiaPatente	varchar(3)	NO		NULL	
NumeroPatente	char(10)	NO		NULL	
DataRilascioPatente	date	NO		NULL	
DataScadenzaPatente	date	NO		NULL	
NumeroCartaIdentita	char(9)	NO		NULL	
DataScadenzaCartaIdentita	date	NO		NULL	
ComuneRilascioCartaIdentita	varchar(45)	NO		NULL	
NumeroCartaCredito	char(16)	NO		NULL	
DataScadenzaCartaCredito	date	NO		NULL	
CVV	char(3)	NO		NULL	
NumeroCellulare	varchar(10)	NO		NULL	

4.2.2. Tabella Corsa

Field	Type	Null	Key	Default	Extra
idCorsa	int	NO	PRI	NULL	auto_increment
DataCreazione	date	NO		NULL	
Stato	tinyint	NO		NULL	
idPrenotazione	int	NO	MUL	NULL	

4.2.3. Tabella Impiegato

Field	Type	Null	Key	Default	Extra
idImpiegato	int	NO	PRI	NULL	auto_increment
Nome	varchar(45)	NO		NULL	
Cognome	varchar(45)	NO		NULL	
ComuneDiNascita	varchar(45)	NO		NULL	
CodiceFiscale	char(16)	NO		NULL	
Sesso	varchar(45)	NO		NULL	
DataDiNascita	date	NO		NULL	
ComuneDiResidenza	varchar(45)	NO		NULL	
IndirizzoDiResidenza	varchar(45)	NO		NULL	
CAP	char(5)	NO		NULL	
NumeroCellulare	varchar(10)	NO		NULL	
Email	varchar(45)	NO		NULL	
Password	char(64)	NO		NULL	
Ruolo	varchar(45)	NO		NULL	

4.2.4. Tabella Pagamento

Field	Type	Null	Key	Default	Extra
idPagamento	int	NO	PRI	NULL	auto_increment
Importo	float	NO		NULL	
idPrenotazione	int	NO	MUL	NULL	

4.2.5. Tabella Parcheggio

Field	Type	Null	Key	Default	Extra
idParcheggio	int	NO	PRI	NULL	auto_increment
Denominazione	varchar(45)	NO		NULL	
Indirizzo	varchar(45)	NO		NULL	
Capienza	int	NO		NULL	

4.2.6. Tabella Prenotazione

Field	Type	Null	Key	Default	Extra
idPrenotazione	int	NO	PRI	NULL	auto_increment
DataCreazionePrenotazione	date	NO		NULL	
IndirizzoPartenza	varchar(45)	NO		NULL	
IndirizzoDestinazione	varchar(45)	NO		NULL	
DataInizioPrenotazione	varchar(45)	NO		NULL	
DataFinePrenotazione	varchar(45)	NO		NULL	
ParcheggioPartenza	int	NO	MUL	NULL	
ParcheggioDestinazione	int	NO	MUL	NULL	
Importo	float	NO		NULL	
idVeicolo	int	NO	MUL	NULL	
idAutista	int	NO	MUL	NULL	
idCliente	int	NO	MUL	NULL	

4.2.7. Tabella Segnalazione

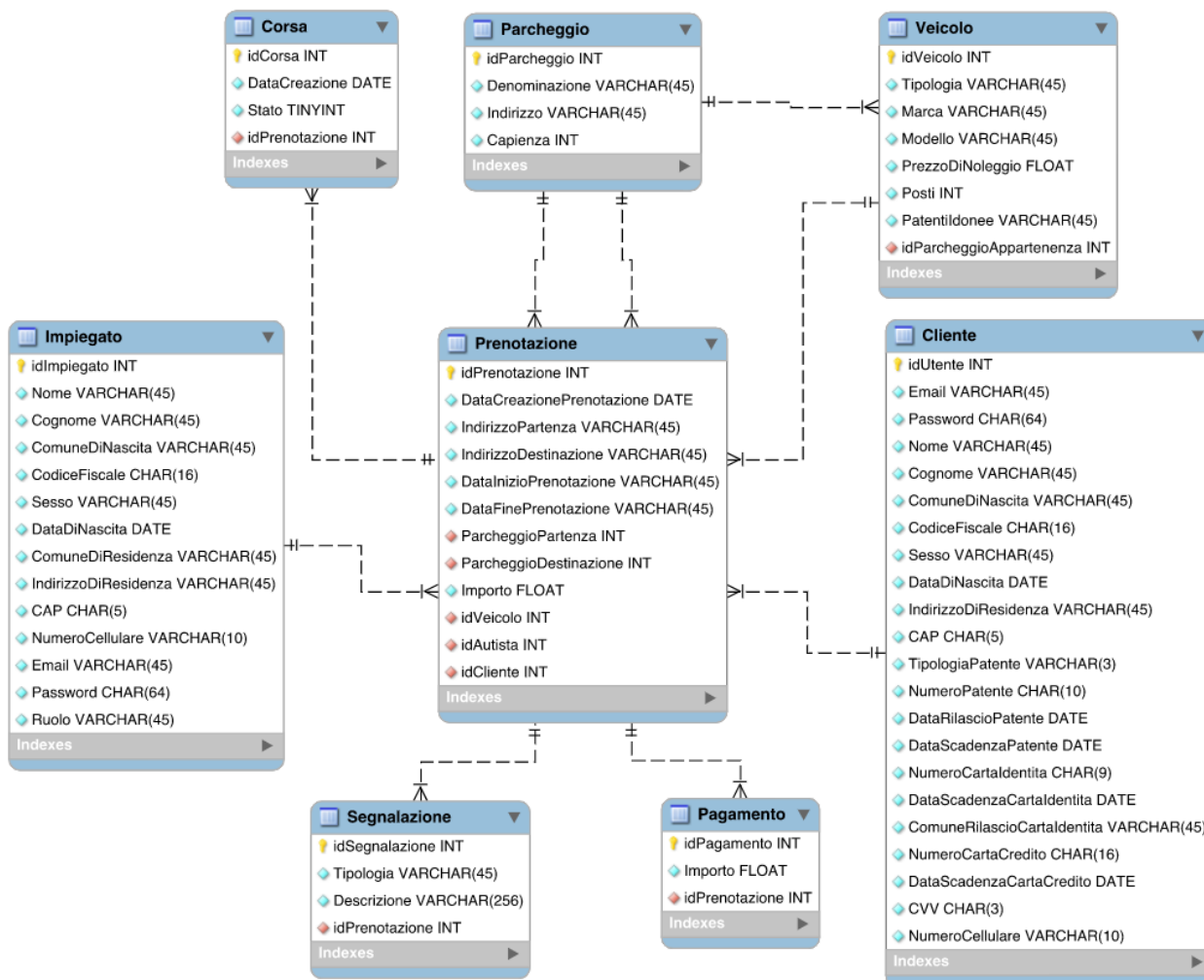
Field	Type	Null	Key	Default	Extra
idSegnalazione	int	NO	PRI	NULL	auto_increment
Tipologia	varchar(45)	NO		NULL	
Descrizione	varchar(256)	NO		NULL	
idPrenotazione	int	NO	MUL	NULL	

4.2.8. Tabella Veicolo

Field	Type	Null	Key	Default	Extra
idVeicolo	int	NO	PRI	NULL	auto_increment
Tipologia	varchar(45)	NO		NULL	
Marca	varchar(45)	NO		NULL	
Modello	varchar(45)	NO		NULL	
PrezzoDiNoleggio	float	NO		NULL	
Posti	int	NO		NULL	
PatentiIdonee	varchar(45)	NO		NULL	
idParcheggioAppartenenza	int	NO	MUL	NULL	

4.3. Diagramma Entità Relazione Con MySQL Workbench

Il precedente diagramma ER è stato realizzato in forma di modello logico, tramite MySQL Workbench.



5. Messaggi D'errore

Nel caso di inserimento di dati o di richieste errati, il sistema, mediante un'opportuna interrogazione del database è in grado di notificare l'errore all'utente ed indirizzarlo opportunamente alla risoluzione degli imprevisti.