

SDD

(System Design Document)

Indice

1. Architettura Software Proposta.....	1
1.1 Panoramica.....	1
2. Gestione dei dati persistenti – Progetto del Database.....	2
2.1 Modello E-R.....	2
2.2 Descrizione delle tabelle.....	2
3. Decomposizione in sottoinsiemi.....	6
4. Mappatura Hardware/Software.....	10
5. Applicazione Android.....	11

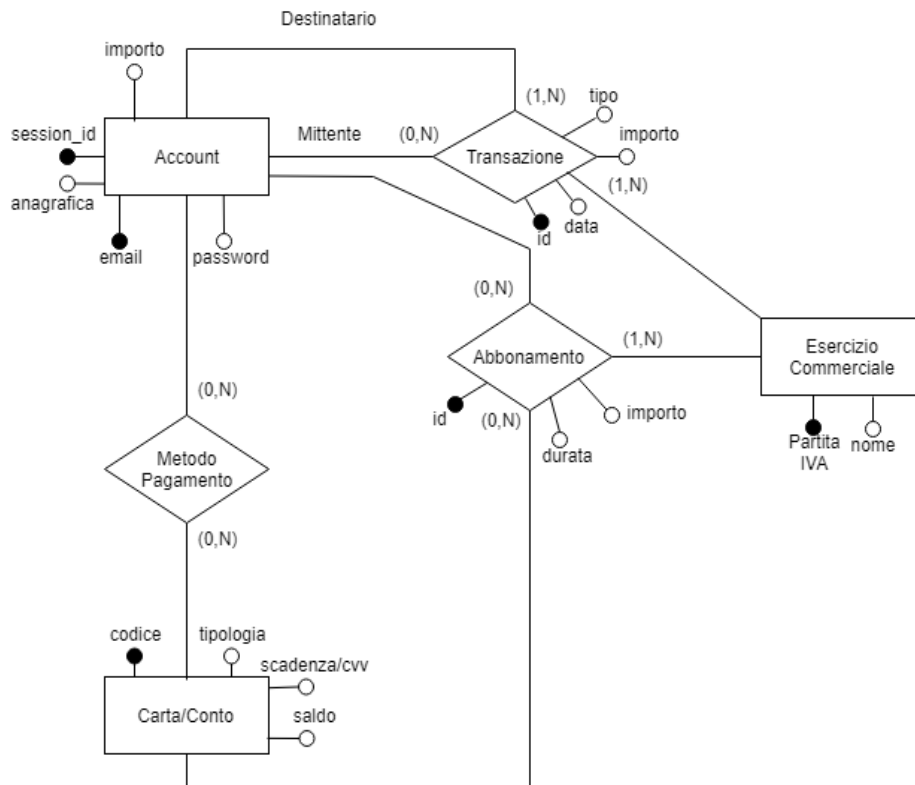
1. Architettura di Software proposta

1.1. Panoramica

Il sistema prevede un'architettura client-server. I nodi client sono I dispositivi degli utenti che accedono al sito web dell'applicazione, o gli smartphone su cui è installata l'app mobile. Il nodo server è rappresentato dal vero e proprio server del committente.

2. Gestione dei dati persistenti– Progetto del database

2.1. Modello E-R



2.2. Descrizione delle tabelle

Qui si presenta una traduzione del modello E-R sopra rappresentato.

ACCOUNT

Nome colonna	Tipo	Vincoli	Descrizione
session_id	varchar(32)	UNIQUE	Codice criptato in md5 che memorizza la sessione corrente.
email	varchar(20)	PRI_KEY	Email univoca dell'utente
password	varchar(32)	NOT_NULL	Password dell' utente. Criptata in md5.
importo	double	POSITIVE NOT_NULL	Saldo virtuale dell'utente
anagrafica	varchar(50) date	NOT_NULL	Anagrafica dell'utente (Codice fiscale etc)

CARTA/CONTO

Nome colonna	Tipo	Vincoli	Descrizione
codice	varchar(27)	PRI_KEY	Codice di una carta o di un conto IBAN
tipologia	varchar(20)	NOT_NULL	Flag che discrimina carta da conto IBAN
Scadenza/cvv	varchar(32)	NOT_NULL	Campi di scadenza e cvv
saldo	double	POSITIVE NOT_NULL	Saldo della carta

TRANSAZIONE

Nome colonna	Tipo	Vincoli	Descrizione
emailmittente	varchar(20)	FOR_KEY Account(email)	Email dell'utente che versa denaro
destinatario	varchar(32)	NOT_NULL	Destinatario del pagamento. Può essere un utente o un esercizio
data	date	NOT_NULL	Data della transazione
importo	double	POSITIVE NOT_NULL	Importo trasferito dal mittente al destinatario
tipo	boolean	NOT_NULL	Flag che discrimina una transazione completata da una in pending (ossia una richiesta di denaro da parte del destinatario al mittente)
id	integer	PRI_KEY, AUTO_INC	Id della transazione

METODO DI PAGAMENTO

Nome colonna	Tipo	Vincoli	Descrizione
email	varchar(20)	PRI,FOR_KEY Account(email)	Email dell'utente che possiede il metodo
codice	varchar(27)	PRI,FOR_KEY Carta/Conto(codice)	Codice della carta o conto associato al metodo

ABBONAMENTO

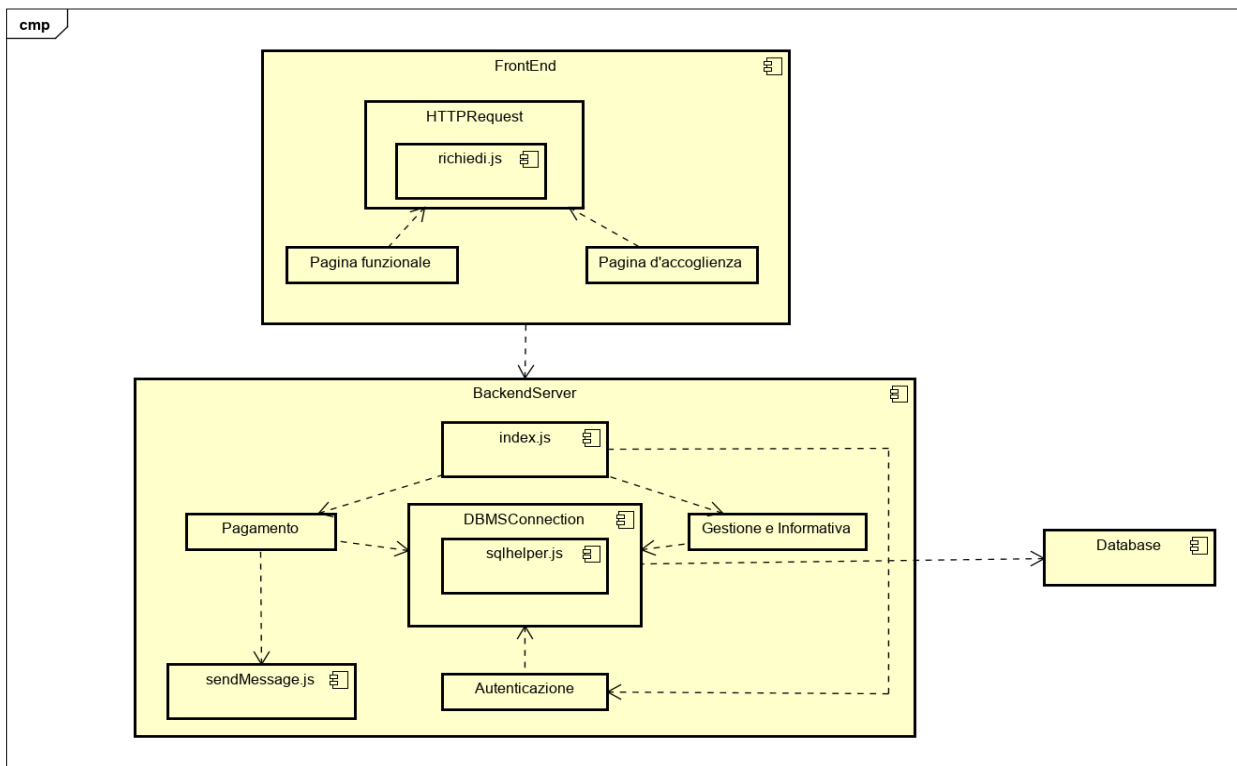
Nome colonna	Tipo	Vincoli	Descrizione
email	varchar(20)	FOR_KEY Account(email)	Email dell'utente che sottoscrive l'abbonamento
Partita IVA	varchar(11)	FOR_KEY Esercizio Commerciale (Partita IVA)	Partita IVA dell'esercizio commerciale a cui l'utente si abbona
durata	date, integer	NOT_NULL POSITIVE	Durata dell'abbonamento. E' espressa in Data di Avvio, numero di pagamenti, periodicità.
importo	double	POSITIVE NOT_NULL	Importo periodicamente versato all'esercizio
id	integer	PRI_KEY, AUTO_INC	Id dell'abbonamento

ESERCIZIO COMMERCIALE

Nome colonna	Tipo	Vincoli	Descrizione
Partita IVA	varchar(11)	PRI_KEY	Partita Iva dell'esercizio
Nome	varchar(20)	NOT_NULL	Nome dell'esercizio

3. Decomposizione in sottoinsiemi

In questa sede, si utilizzano le componenti diagram di UML e si mostrano i sottosistemi come component UML per ragioni di flessibilità del modello. Si è scelto di decomporre il sistema nel seguente modo



Il sottosistema “Frontend” contiene file statici che riceve il client quali html, css e javascript. Tutti I file rappresentati all'interno di questo sottosistema sono presenti all'interno della cartella “Frontend”. L'applicazione mobile contiene al suo interno tali files.

In basso, I dettagli delle componenti di “Frontend”.

cmp

Pagina d'accoglienza

index.html

login.html

home.html

register.html

cmp

Pagina funzionale

Area Saldo

panel.js

panel.html

Area Movimenti

transactions.js

transactions.html

Area Notifiche

panel.js

panel.html

Logout

logout.js

Area Versamenti

sendmoney.js

sendcommonmoney.js

periodicalpayments.js

addperiodicalpayments.js

sendmoney.html

sendcommonmoney.html

periodicalpayments.html

addperiodicalpayments.html

Area Richiesta

requestmoney.js

requestmoney.html

Area Gestione Profilo e Metodi di Pagamento

profile.js

profile.html

changepwd.html

addpayments.js

delet payments.js

linkiban.html

linkcard.html

“Pagina d'accoglienza” contiene le componenti utili all'accesso di un visitatore al sito web o all'apk. In particolare, home.html è la pagina d'accoglienza che incontra il visitatore una volta aperta l'apk, e conterrà un form per inserire l'indirizzo IP del server. Successivamente il visitatore verrà indirizzato a index.html.

I visitatori che accedono da browser entrano direttamente in index.html.

“Pagina funzionale” contiene le componenti effettivamente funzionali per l'utente registrato. E' stata divisa in sette aree, sei aree più una di logout. L'area saldo e l'area notifiche visualizzano il saldo virtuale dell'utente e le notifiche di richieste di pagamento.

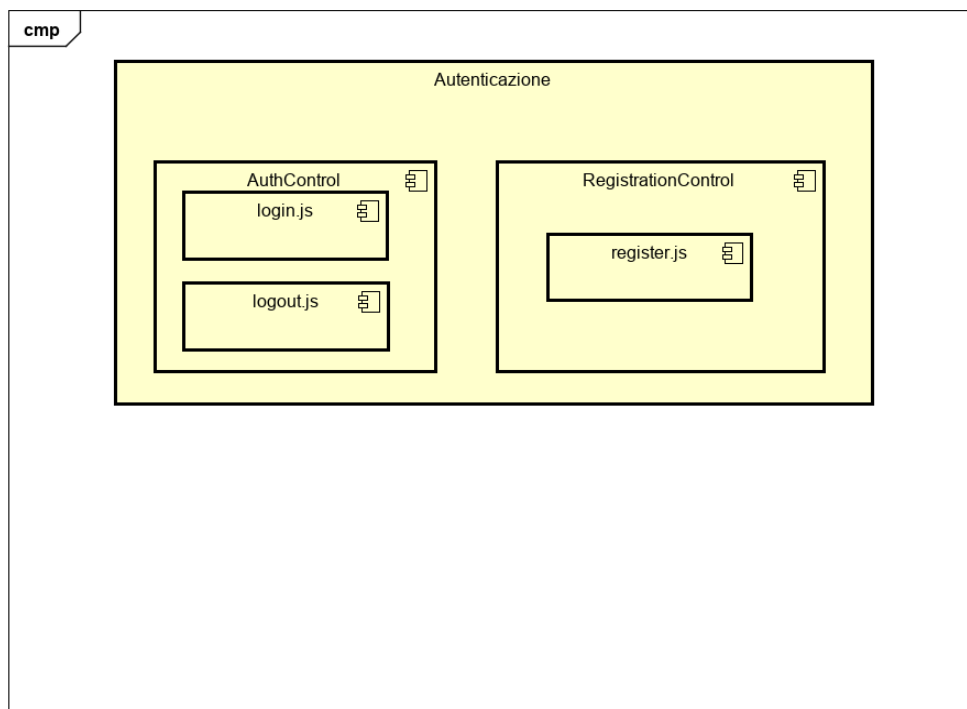
L'area Movimenti visualizza la lista delle transazioni dell'utente.

L'area Versamenti indirizza a pagine web di invii di denaro o di gestione di pagamenti periodici.

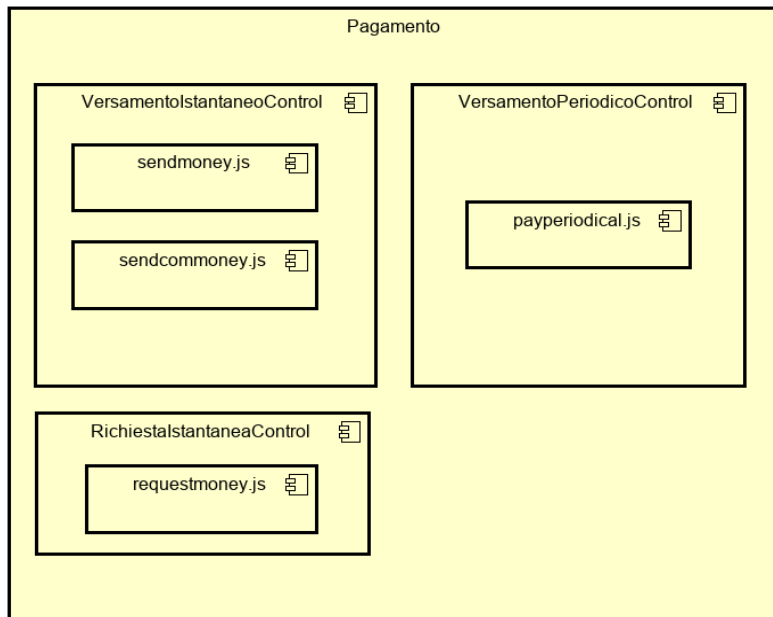
L'area Richiesta indirizza alla pagina web di richiesta di pagamento.

L'area Gestione Profilo e Metodi Di Pagamento indirizza alle pagine web di gestione del profilo e dei metodi di pagamento.

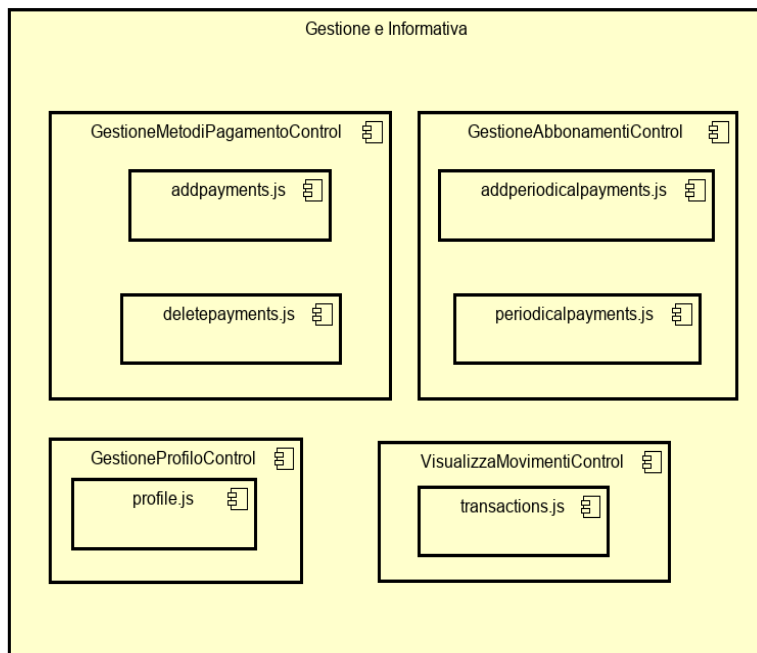
Di seguito si mostra la divisione in sottoinsiemi delle parti di “Backend”. Si noti che sono stati usati gli stessi nomi dei codici javascript di frontend, tuttavia il loro funzionamento è diverso (uno responsabile del frontend, L'altro del backend).



cmp



cmp



“Autenticazione” contiene le control relative all'autenticazione o alla registrazione di un visitatore.

“Pagamento” contiene le control relative a versamenti istantanei o periodici di un visitatore.

“Gestione e Informativa” contiene le control relative alla gestione e all'informativa di dati anagrafici o di pagamento (metodi, transazioni, etc).

In fase di progettazione si è scelto di assegnare tutte le operazioni che nei sequence diagram apparivano come messaggi inviati al DBMS, ad un apposito oggetto nella componente “DBMSConnection” (che non ha un corrispettivo in fase di analisi e quindi nel RAD, in quanto trattasi di dettaglio non avente significato a quel livello di astrazione), che di fatto si occuperà di sostituire qualunque interazione diretta con il DBMS all’interno del sistema.

DBMSHelper
+ richiestaInvioDati() : void + richiestaAggiornamentoDati() : void + richiestaEliminazioneDati() : void + richiestaInserimentoDati() : void

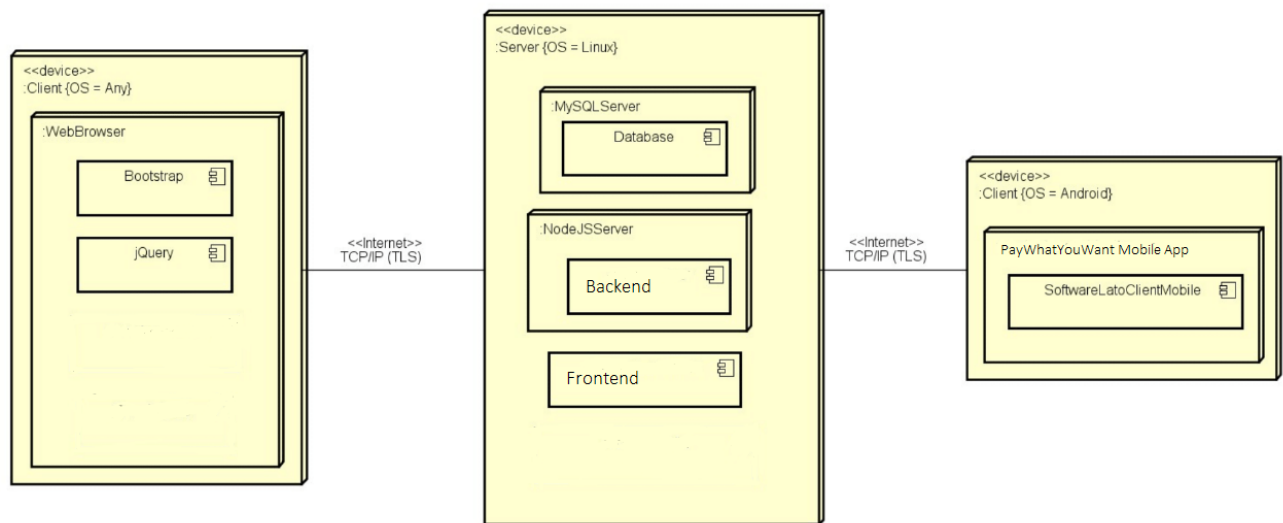
Per completezza, si riporta qui di fianco insieme alle sue operazioni.

4. Mappatura Hardware/Software

La mappatura è stata effettuata sulla base del modello architetturale client-server, come illustrato nella panoramica.

Il nodo server conterrà:

1. Un'istanza di un server NodeJSServer, che si occuperà di servire le richieste dei client e delle operazioni di gestione interna del server. Queste richieste saranno in particolare richieste HTTPS, che Utilizzeranno il protocollo TCP/IP con lo strato di crittografie TLS.
2. Un'istanza di MySQLServer, che si occuperà di gestire i contenuti del Database, di fornirli e di modificarli a seguito delle richieste provenienti dal web server di Node.js. Tali richieste saranno gestite dal protocollo TCP/IP e, lato software, dal modulo mysql di Node.js.
3. Il software lato client (i file html, css, javascript), che verrà servito al client che si collega al sito web dell'applicazione.



5. Applicazione Android

L'applicazione Android è stata creata utilizzando il software Cordova. La struttura dell'applicazione è identica a quella del software lato client, e come nel software del lato client al suo interno sono presenti files di libreria. Come menzionato precedentemente, in fase di avvio il visitatore approderà alla pagina web home.html, dove gli verrà richiesto l'indirizzo IP per collegarsi al server dell'applicazione. Una volta inserito l'IP, il server viene contattato per richiedere la pagina web iniziale (index.html); da lì si svolge il normale corso del funzionamento, comune a quello del software lato client.