Sito internet: IIS "Marzotto-Luzzatti"
E-Mail: viis02004@istruzione.it
E-Mail Certificata: viis02004@pec.istruzione.it

Elaborato Esame di Stato

Informatica e Sistemi - Anno Scolastico 2020/2021

Introduzione

Candidato

Nome: Alex
Cognome: Sandri

Sommario

Traccia

Per promuovere la micro-mobilità sostenibile il Ministero delle infrastrutture e della mobilità sostenibili commissiona lo sviluppo di una nuova piattaforma per gestire i noleggi di monopattini elettrici a livello nazionale.
L'utente accede al servizio tramite il Sistema Pubblico di Identità Digitale (SPID). Il sistema permetterà agli utenti del servizio di trovare dei mezzi nelle loro vicinanze tramite l'utilizzo del sistema GPS. Il noleggio parte dopo la conferma dell'utente successiva all'aver inquadrato il codice QR, posto su ogni mezzo, tramite la fotocamera del dispositivo.
L'utente possiede un portafoglio virtuale nel quale può aggiungere fondi con cui poter pagare i noleggi.
Si descriva l'infrastruttura che offre il servizio web e una ipotetica stazione di noleggio. Si approfondiscano le tecnologie per l'identificazione SPID.

Il candidato, fatte le opportune ipotesi aggiuntive, sviluppi tutti i seguenti punti:

1. *un'analisi della realtà di riferimento ipotizzata descrivendo, per la soluzione proposta:*
    - *architettura di rete e caratteristiche dei sistemi server;*
    - *modalità di comunicazione tra server e dispositivi, protocolli e servizi software per gestire la rete e fornire le pagine;*
    - *gestione della sicurezza dei sistemi informatici realizzati o utilizzati;*
    - *modello concettuale e logico del database;*
    - *implementazione dello schema logico mediante linguaggio SQL.*
2. *Implementi una parte significativa del progetto sia per quanto riguarda l'applicazione web (o App per dispositivi mobili Android o IoS), sia per quanto riguarda la configurazione dei servizi e dispositivi di rete.*

Svolgimento

Rete: infrastruttura e sicurezza

Per l'infrastruttura di rete si è scelto di utilizzare vari servizi in ***cloud***, così da esternalizzare (*outsource*) la gestione e la sicurezza di certi aspetti, quali:

1. *Back-up*
2. Aggiornamenti *software* e *hardware*
3. Ridondanza dei dati

Tutti i servizi comunicheranno tra di loro esclusivamente in modo sicuro attraverso **HTTPS**, per quindi garantire che i dati dell'utente siano mantenuti riservati mentre sono in transito attraverso Internet.

Tutte le rotte dell'API sono protette da un sistema di autenticazione basato su sessioni, fatta eccezione per quella richiamata da Stripe, in questa vengono però verificate l'autenticità e l'integrità della richiesta tramite una firma basata su HMAC inviata con essa.

L'identificativo della sessione viene inviato insieme ad ogni richiesta diretta all'API tramite l'*header* **Authorization**, con questo contenuto: **Bearer *id_sessione***.

Viene ora riportato il codice per la creazione di una nuova sessione, l'aggiunta del *cookie* e il reindirizzamento al sito principale:

```
$session_id = "ses_" . bin2hex(random_bytes(50));

$session_expires_at = new DateTime("now", new DateTimeZone("UTC"));
$session_expires_at->add(new DateInterval("P1M"));

try
{
    pg_query_params(
        $connection,
        '
        insert into "sessions"
            ("id", "user", "expires_at")
        values
            ($1, $2, $3)
        ',
        [
            $session_id,
            $user_id,
            $session_expires_at->format(DateTime::ATOM),
```

```
        ]
    );
}
catch (Exception $e)
{
    exit;
}

setcookie(
    "session_id",
    $session_id,
    [
        "expires" => 0,
        "path" => "/",
        "domain" => Config::is_prod()
            ? parse_url($_ENV["CLIENT_HOST"], PHP_URL_HOST)
            : false,
        "secure" => Config::is_prod(),
        "httponly" => false,
        "samesite" => "Strict",
    ]
);

header("Location: " . $_ENV["CLIENT_HOST"]);
```

Viene ora riportata la porzione di codice significativa per quanto riguarda la validazione del *token* di sessione:

```
const authorization = request.raw.req.headers.authorization;

if (!authorization)
{
    throw Boom.unauthorized();
}

const session = await Session.retrieve(authorization.split(" ")[1]);

if (session.hasExpired())
{
    throw Boom.unauthorized();
}

const { user } = session;
```

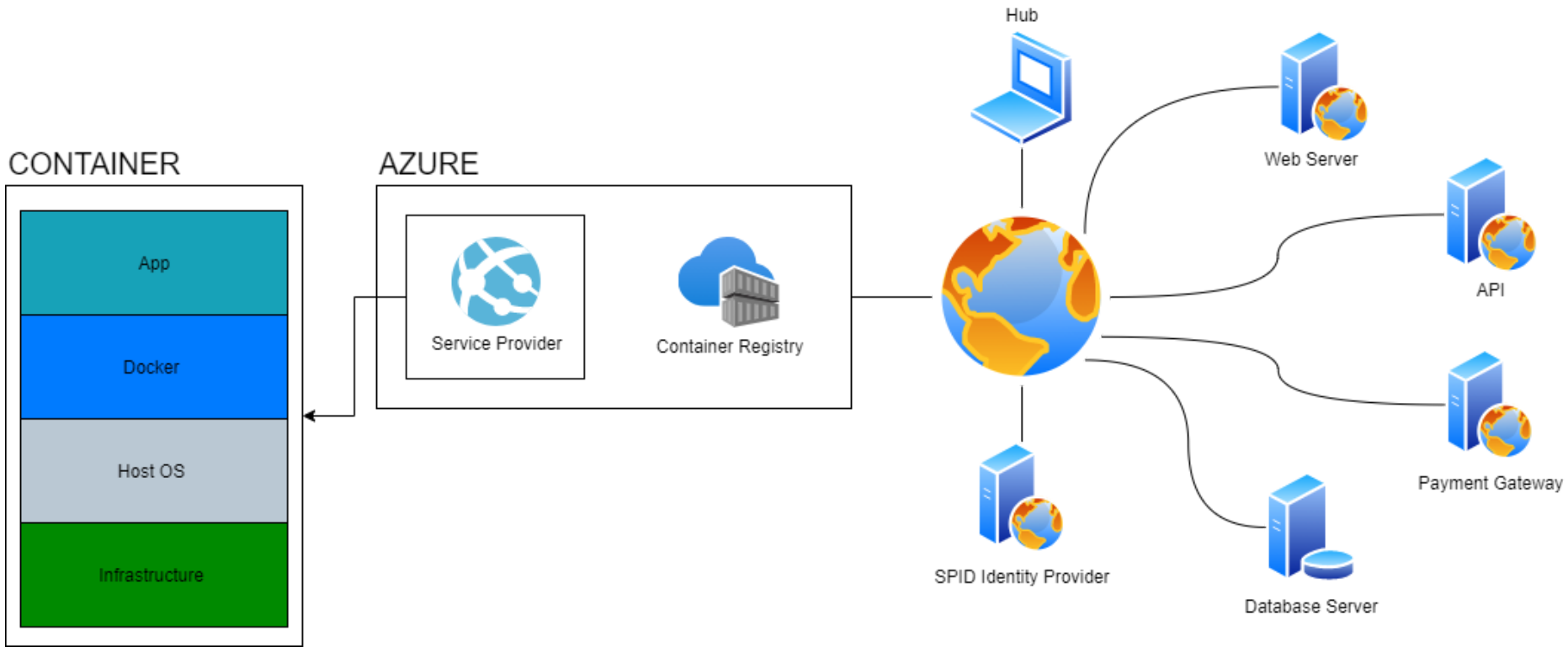
```
const scope = [ user.type ];

if (user.type === "admin")
{
    scope.push("user");
}

return h.authenticated({
    credentials: {
        user,
        scope,
    },
});
```

Infine viene riportato l'oggetto che contiene gli schemi, in questo caso della risorsa ***WALLET***, che vengono utilizzati dall'API per validare richieste e risposte:

```
public static readonly SCHEMA = {
    OBJ: Joi.object({
        id: Schema.ID.WALLET.required(),
        name: Schema.STRING.max(30).required(),
        balance: Schema.MONEY.required(),
        user: User.SCHEMA.OBJ.required(),
        __metadata: Joi.object({
            is_default: Schema.BOOLEAN.required(),
        }).optional(),
    }),
    CREATE: Joi.object({
        name: Schema.STRING.max(30).required(),
    }),
    UPDATE: Joi.object({
        name: Schema.STRING.max(30).optional(),
    }),
} as const;
```

- **OBJ** viene utilizzato per garantire che l'API risponda sempre con la stessa struttura
- **CREATE** viene utilizzato per validare le richieste di creazione
- **UPDATE** viene utilizzato per validare le richieste di aggiornamento

Diagramma

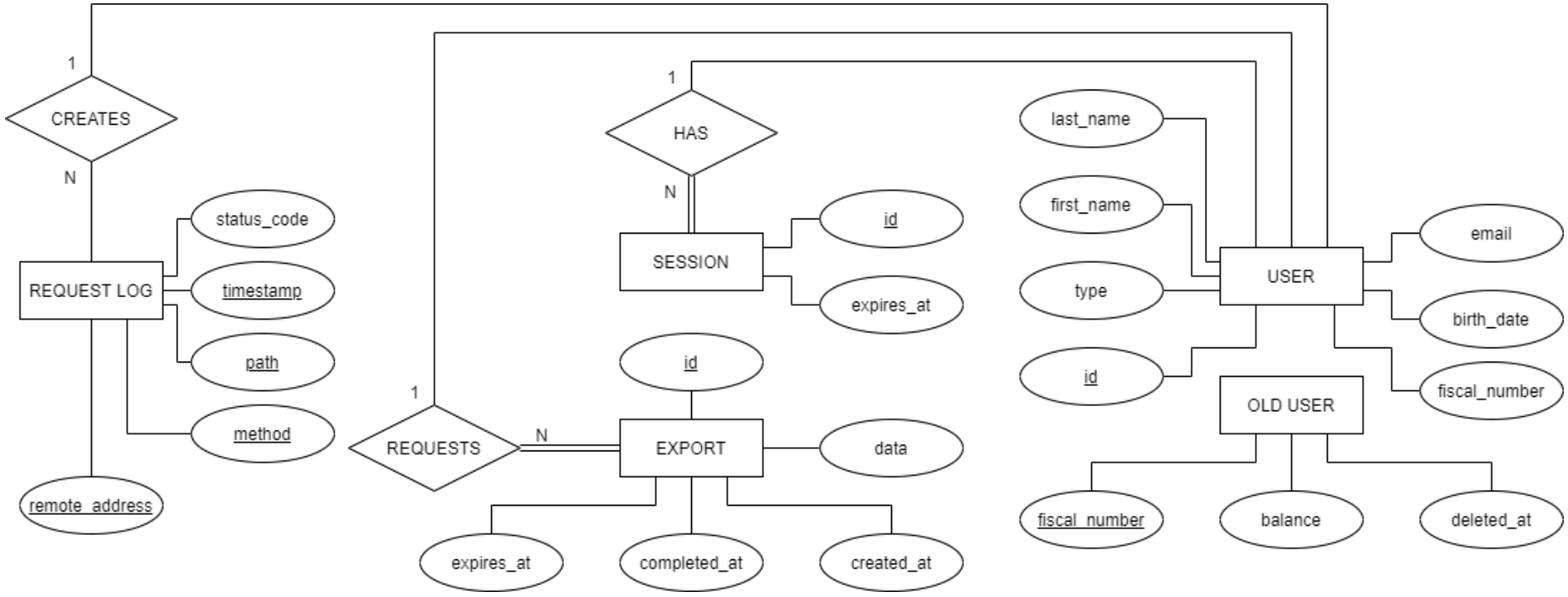

Hub
Web Server
API
Payment Gateway
Database Server
SPID Identity Provider
CONTAINER
App
Docker
Host OS
Infrastructure
AZURE
Service Provider
Container Registry

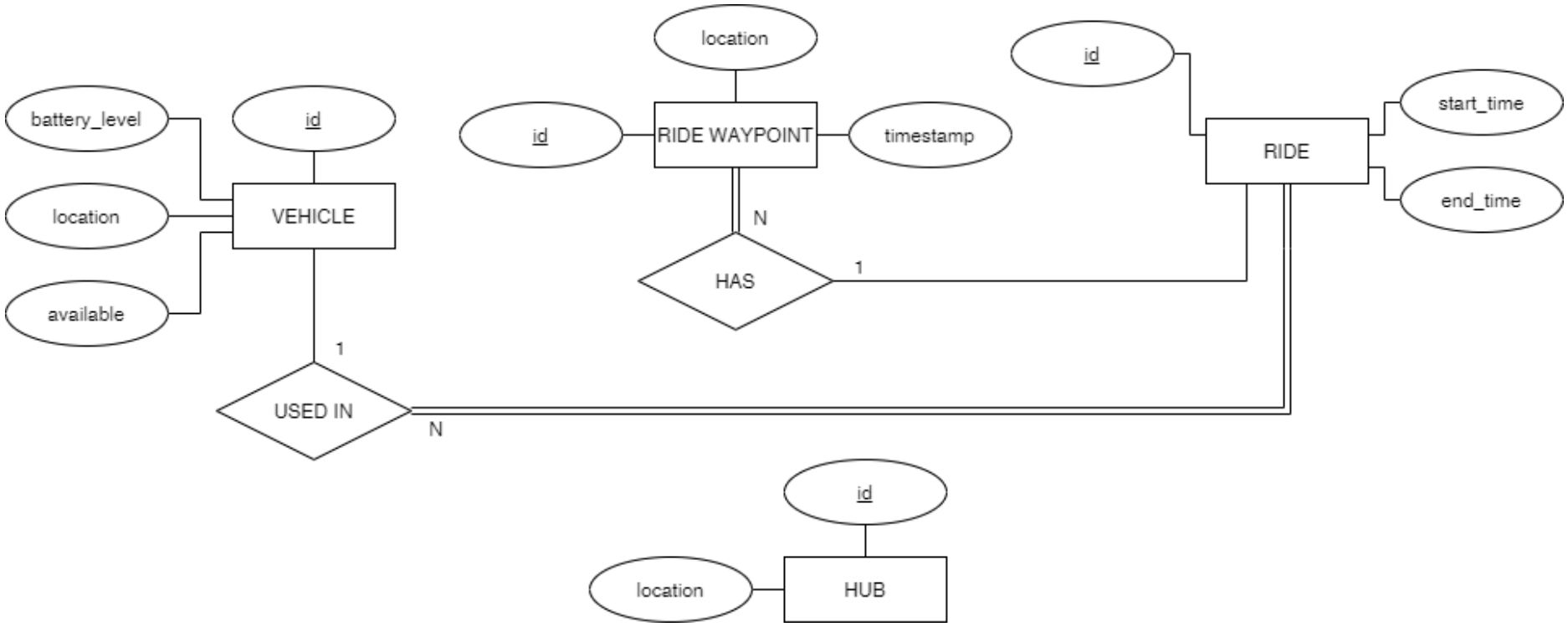
Progettazione del database

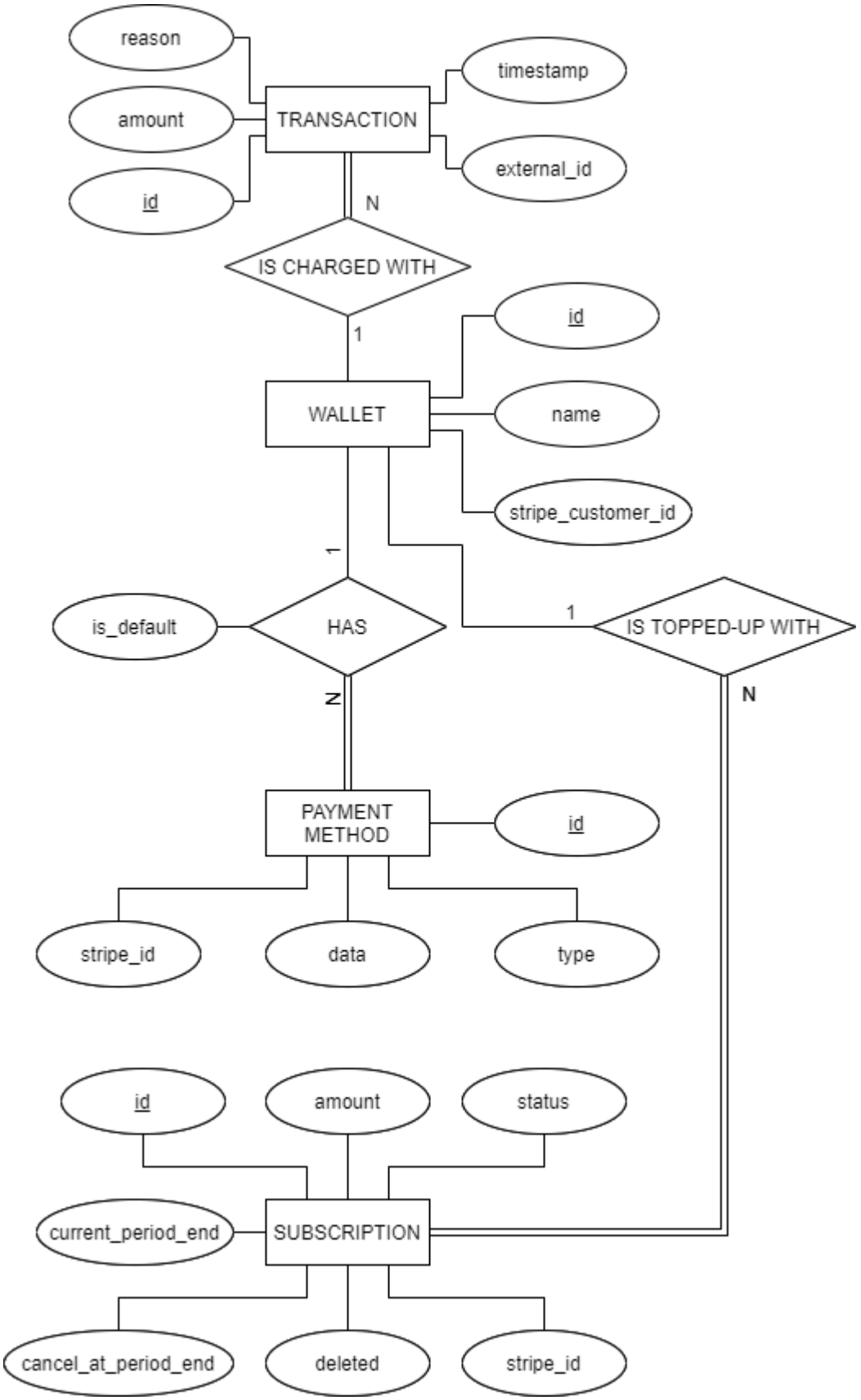
Schema concettuale

Lo schema è stato suddiviso in tre sotto-schemi, rappresentanti le tre entità principali (***USER***, ***WALLET*** e ***RIDE***), per migliorarne la leggibilità. Sono ora quindi riportate le associazioni necessarie a collegare il tutto:

1. **USER - WALLET**: 1 - N (un utente può avere più portafogli, ma un portafoglio può appartenere ad un solo utente)
2. **WALLET - RIDE**: 1 - N (un portafoglio può essere utilizzato per pagare più corse, ma una corsa viene pagata con un solo portafoglio)




location
id
battery_level
id
id
RIDE WAYPOINT
timestamp
RIDE
start_time
location
VEHICLE
N
end_time
HAS
1
available
1
USED IN
N
id
location
HUB


reason
amount
id
TRANSACTION
timestamp
external_id
N
IS CHARGED WITH
1
id
WALLET
name
stripe_customer_id
1
is_default
HAS
1
IS TOPPED-UP WITH
N
N
PAYMENT METHOD
id
stripe_id
data
type
id
amount
status
current_period_end
SUBSCRIPTION
cancel_at_period_end
deleted
stripe_id

Schema logico

Specifica

- Gli **id** sono tutti realizzati in questo modo:
  - un prefisso di esattamente tre caratteri
  - un trattino basso
  - stringa di caratteri casuali di lunghezza non necessariamente fissa
- Gli indirizzi **email** sono definiti come **varchar(320)** in quanto un indirizzo email valido non può superare i 320 caratteri totali (RFC3696)
- Le **coordinate** vengono specificate attraverso il tipo **geography(point, 4326)**, parte dell'estensione per PostgreSQL "PostGIS".
- I campi di testo che non sono per qualche motivo necessariamente ristretti in lunghezza (come email e codice fiscale) sono definiti come **text**
- Salvo diversamente specificato tutte le chiavi esterne vengono aggiornate o eliminate all'aggiornamento (**on update cascade**) o eliminazione (**on delete cascade**) delle righe a cui si riferiscono

Implementazione

È stato scelto di riportare la tabella **TRANSACTIONS** come esempio rappresentativo. Questa è la tabella che viene utilizzata per tenere traccia di tutti i movimenti all'interno di un portafoglio, viene inoltre utilizzata per calcolarne il saldo tramite la query SQL seguente, dove `$1` è il parametro che viene valorizzato con l'id del portafoglio:

```
select coalesce(sum("amount"), 0) as "balance"
from "transactions"
where "wallet" = $1
```

Nel caso non siano ancora presenti transazioni allora `sum("amount")` restituisce **null**, entra quindi in gioco la funzione `coalesce` che, in questo caso, restituisce il valore 0, dato che il saldo deve essere valorizzato.

La ***reason*** specifica il tipo di transazione avvenuta e può essere:

- **restore-balance**: Transazione relativa al ripristino del credito precedente: questo avviene in quanto un utente può eliminare il suo account in qualsiasi momento e viene quindi aggiunto un *record* in **OLD_USERS** con il suo Codice Fiscale, il suo credito (negativo o positivo) e il momento in cui questa operazione è avvenuta. Viene poi ripristinato questo credito al prossimo accesso con lo stesso Codice Fiscale.
- **ride**: Transazione relativa ad una corsa
- **subscription**: Ricarica periodica
- **top-up**: Ricarica singola

TRANSACTIONS

(

id: text,
amount: numeric(10, 2),
timestamp: timestamp,
wallet: text,
reason: text,
external_id: text,

)

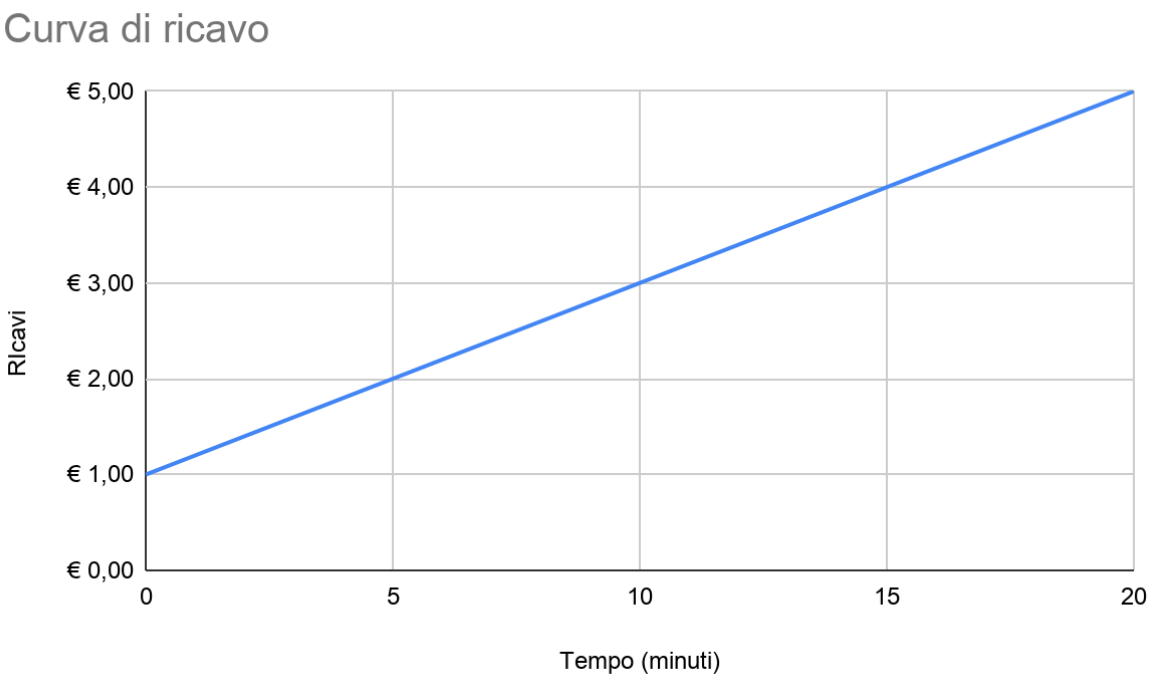
- **id** comincia con '**trx_**'
- **timestamp** ha come valore predefinito **current_timestamp**
- **timestamp** <= *current_timestamp*
- **{ wallet }** chiave esterna riferita a **WALLETS**
- **{ reason }** chiave esterna riferita a **TRANSACTION_REASONS**
- **{ external_id }** chiave candidata relativa a:
    - Una **RIDE** se questo è l'addebito dopo la fine della corsa (*reason* **ride**)
    - Un **Payment Intent** (Stripe) se questo è l'accredito dovuto ad una ricarica, periodica (*reason* **top-up**) o meno (*reason* **subscription**)
    - **null** se la *reason* è **restore-balance**

```
create table "transactions"
(
    "id" id not null,
    "amount" numeric(10, 2) not null,
    "timestamp" timestamp not null default current_timestamp,
    "wallet" id not null,
    "reason" text not null,
    "external_id" text not null,

    primary key ("id"),

    unique ("external_id"),

    foreign key ("wallet") references "wallets" on update cascade on
delete cascade,
    foreign key ("reason") references "transaction_reasons" on update
cascade on delete cascade,

    check ("id" like 'trx_%'),
    check ("timestamp" <= current_timestamp)
);
```

Viene ora riportata la parte significativa del codice che calcola l'ammontare da addebitare all'utente alla fine della corsa:

```
const amount = Config.RIDE_FIXED_COST
    + (differenceInMinutes(endTime, this.start_time) *
Config.RIDE_COST_PER_MINUTE);
```

Segue la curva dei ricavi di una corsa.



Sito web

Il sito web completo è disponibile all'indirizzo scootr.it.
È possibile effettuare l'accesso tramite l'*Identity Provider* di test all'indirizzo spid.scootr.it/signin.php?idp=idp_testenv2_prod, con *username* **test** e *password* **test**.

Requisiti

Come da traccia il sistema verrà realizzato come applicazione web che dovrà consentire all'utente:

1. La creazione di uno o più portafogli virtuali (al primo accesso ne verrà già creato uno predefinito), con la possibilità di scegliere quello predefinito
2. L'aggiunta di uno o più metodi di pagamento ad un portafoglio (non necessariamente solo carte di credito)
3. La ricarica del portafoglio (periodica o meno)
4. La visualizzazione dello storico delle transazioni per ogni portafoglio
5. La visualizzazione di tutte le ricariche periodiche (attive, incomplete o disattivate)
6. La visualizzazione dei dettagli dell'account (nome, cognome, ecc.)
7. La visualizzazione di tutte le corse effettuate e, per ogni corsa, i dettagli e una mappa con il percorso effettuato

8. La possibilità di chiudere l'account, in tal caso viene salvato il saldo complessivo dell'utente (positivo o negativo che sia)
9. La possibilità di esportare tutti i dati relativi all'account
10. La possibilità di cominciare una nuova corsa scansionando il Codice QR posto su ogni veicolo e scegliendo su che portafoglio addebitare il costo (si può ipotizzare che un utente voglia separare le spese sostenute per recarsi al posto di lavoro e quelle personali)
11. La visualizzazione della corsa attiva (anche se la pagina viene ricaricata o se l'utente accede da un altro dispositivo)
12. La visualizzazione di una mappa con tutte le stazioni e i veicoli, nel caso questi non siano presso una stazione, nelle vicinanze
13. La possibilità di cercare un indirizzo e spostare dunque la mappa su un altro punto (potrebbe essere necessario se il dispositivo non può fornire l'accesso al GPS o se l'utente ne ha negato il consenso)
14. L'utilizzo del servizio completo con o senza l'accesso al GPS

Premessa

L'applicazione web verrà realizzata utilizzando questi strumenti:

- Angular per il *front-end*, si tratta quindi di una SPA (*Single Page Application*) e perciò non necessita di un linguaggio lato server in quanto il tutto viene distribuito agli utenti staticamente e l'interfaccia viene di conseguenza personalizzata nel client.
- Node.js per il *back-end* utilizzando il *framework* hapi per realizzare API RESTful che mette già a disposizione diversi strumenti per velocizzare e semplificare lo sviluppo e la messa in sicurezza del servizio.
- PostgreSQL come DBMS.
- Docker per il servizio di autenticazione SPID, questo strumento permette di virtualizzare un ambiente di esecuzione garantendo quindi che sia identico sia in locale per il *testing* sia in un server pubblico rimuovendo così i problemi che potrebbero sorgere da versioni dei vari *software* diverse tra i vari ambienti.

Realizzazione

Come parte significativa viene proposta la realizzazione della *home page*.

La *home page* del sito ha due "stati":

1. **Accesso Eseguito**:
   In questo caso l'utente visualizza una mappa che ricopre l'intera superficie della pagina (la cui visualizzazione è delegata alla libreria JavaScript Leaflet che visualizza i dati provenienti da OpenStreetMap) con dei *marker* che mostrano la posizione dei vari veicoli.
   Altri elementi della pagina sono:
   a. Header con il logo, un *input* per consentire all'utente di cercare un indirizzo sulla mappa e un pulsante che al *click* visualizza un ulteriore menu con pulsanti per andare alla pagina del proprio *account* o terminare la sessione.
   b. Floating Action Buttons che consentano queste azioni:
      - Impostare la visualizzazione della mappa alla posizione corrente

        - Scansionare Codice QR, questo viene visualizzato solo se non c'è una corsa attiva
        - Terminare corsa, questo viene visualizzato solo se c'è una corsa attiva
    c. Sezione con le informazioni (tempo trascorso e costo stimato) della corsa attualmente in corso (se presente)
    d. Messaggio in caso l'utente non abbia concesso l'accesso al GPS o il dispositivo e/o il browser non supportino questa funzionalità
2. **Accesso NON Eseguito**:
   In questo caso l'utente visualizza l'header con il solo logo, il messaggio principale del sito (*hero message*) e due pulsanti:
    1. **Entra con SPID**: Permette all'utente di accedere al servizio autenticandosi con le proprie credenziali SPID, scegliendo prima l'*Identity Provider* tramite un menu che viene visualizzato al *click* sul lato destro della pagina
    2. **Entra con CIE**: Permette all'utente di accedere al servizio autenticandosi con la propria Carta d'Identità Elettronica

Il *rendering* condizionale viene effettuato tramite il seguente codice:

```
<signed-in-home *ngIf="auth.user"></signed-in-home>

<signed-out-home *ngIf="!auth.user"></signed-out-home>
```

App per dispositivi mobili

Per rendere più facile e semplice l'utilizzo del servizio da un dispositivo mobile si è pensato di realizzare una applicazione apposita per le due piattaforme più diffuse: **Android** e **iOS**.

L'app deve poter offrire un numero comparabile di funzionalità al sito web, inoltre deve essere organizzata in modo analogo, per evitare eventuali confusioni dell'utente in caso volesse passare dal sito web all'app o viceversa.

Questa applicazione verrà realizzata utilizzando Flutter, un SDK (*Software Development Kit*) basato sul linguaggio di programmazione Dart per realizzare applicazioni native per varie piattaforme senza la necessità di dover mantenere più versioni della stessa app.

Screenshot disponibili su GitHub.

Codice sorgente

Il codice sorgente di tutto il progetto è disponibile ai seguenti indirizzi:

- **Sito web**
- **API**
- **Servizio di autenticazione SPID**
- **App per dispositivi mobili**

Stazione di noleggio

Il servizio offre la possibilità all'utente di lasciare il veicolo in praticamente tutto il territorio senza particolari restrizioni, tuttavia non si esclude la possibilità che possa essere implementato un divieto di parcheggio in certe aree.
È però consigliato, e si potrebbero anche ipotizzare eventuali incentivi, lasciare il veicolo a fine corsa in una delle stazioni disponibili nelle vicinanze.
Il servizio non obbliga questo comportamento in quanto non si può sapere se dove l'utente è diretto sia disponibile una stazione, viene perciò lasciata a discrezione dell'utente la decisione.

La stazione non è altro che un "parcheggio" per i vari veicoli del servizio, possibilmente coperto, che offre queste possibilità:

1. Prelievo / consegna di un veicolo, questa funzione è comunque delegata all'utente semplicemente questo può essere un punto più comodo dove effettuare queste azioni
2. Ricarica dei veicoli tramite la rete elettrica comunale

Il numero di veicoli di ciascuna stazione che viene mostrato sulla mappa è ottenuto tramite la seguente *query*:

```
select "h"."id" as "hub", count(*) as "count"
from
    "v_vehicles" as "v"
    left outer join
    "hubs" as "h"
    on st_dwithin("v"."postgis_location", "h"."location", 10)
group by "hub";
```

Non viene perciò aggiunto un campo che determina se un veicolo è in una stazione o meno, ma si è optato per ottenere questa informazione calcolando il numero di veicoli che sono nel raggio di **10 metri** da ogni stazione.

Autenticazione SPID (e CIE)

Come metodo di accesso è stato scelto SPID, abbinato a CIE.
Si è deciso di utilizzare questo metodo (e non una classica combinazione *email* e *password*) in quanto si ha così la certezza che l'utente sia una persona reale e che i dati inseriti siano corretti e verificati, evitando così account falsi o con dati errati.

Questo metodo offre molti vantaggi ma richiede anche molto più tempo e risorse per essere integrato correttamente:

1. È necessario per l'implementazione diretta all'utente finale (ambiente di produzione) fare richiesta per diventare un *Service Provider* all'AgID (Agenzia per l'Italia Digitale):
    a. SPID: https://www.spid.gov.it/come-diventare-fornitore-di-servizi-pubblici-e-privati-con-spid
    b. CIE: https://federazione.servizicie.interno.gov.it/
2. È richiesto di rispettare una lunga serie di norme dettagliate nella documentazione tecnica

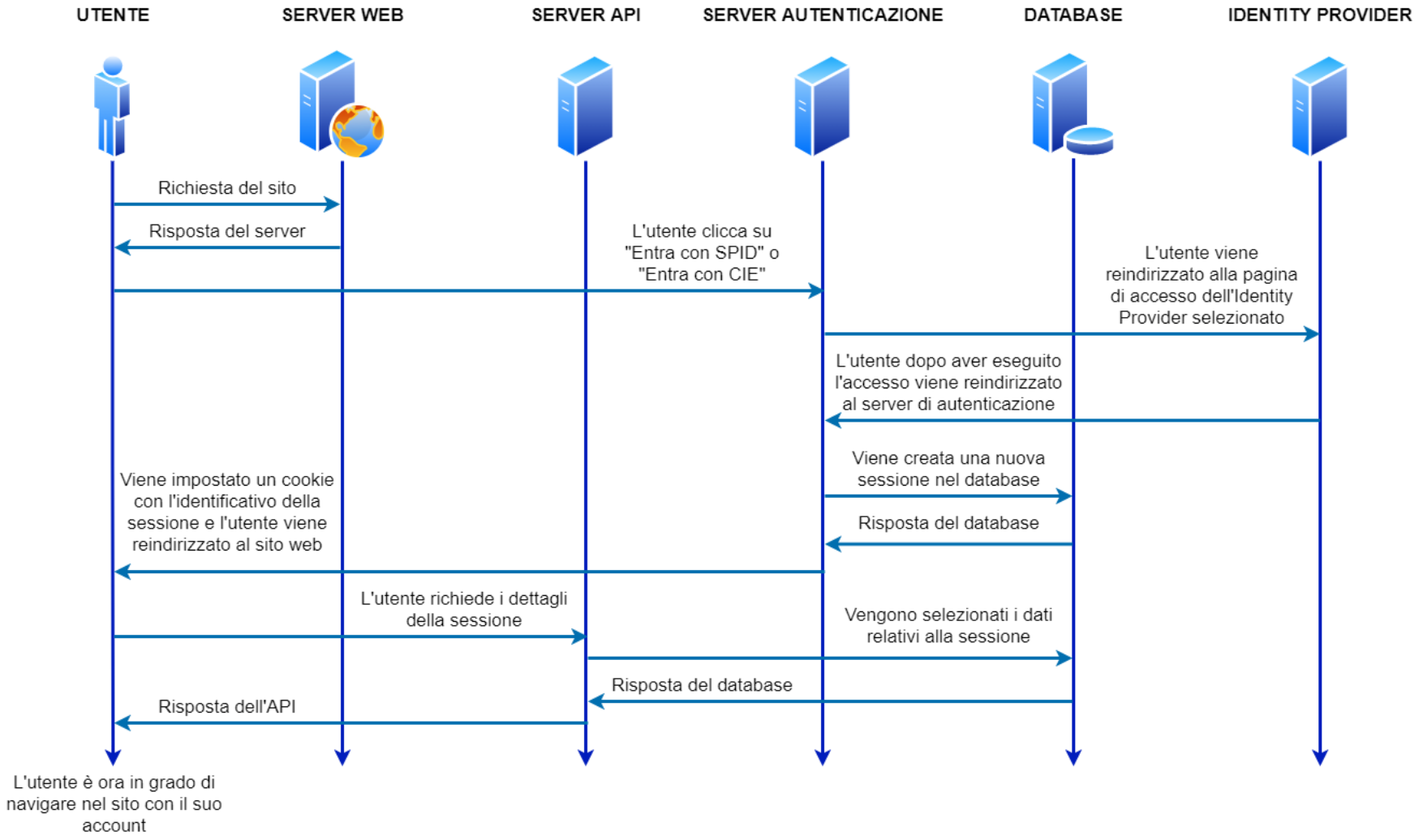
L'autenticazione SPID è suddivisa in **tre livelli**, con un livello di sicurezza crescente:

1. Semplice accesso con nome utente e *password*
2. Accesso con nome utente, *password* e un codice OTP (*one time password*), fornito tramite SMS o app
3. Oltre al nome utente e alla *password* viene richiesto un supporto fisico che gestisca dei certificati digitali, come una *smart card*

L'*Identity Provider* verifica l'autenticità della richiesta proveniente dal *Service Provider* tramite la verifica dei metadati di quest'ultimo, contenenti dei certificati **X.509**.
La stessa cosa vale anche per il *Service Provider*, che verifica le richieste provenienti dall'*Identity Provider*.
È possibile vedere un esempio di metadati all'indirizzo https://testidp.scootr.it/metadata.

Qui sotto viene riportato un diagramma di sequenza che rappresenta il funzionamento dell'autenticazione tipo, ovvero in caso vada tutto correttamente.
Per semplicità non è presente tutto il processo di scambio dei certificati, che si collocherebbe tra la terza e la quarta fase.

Diagramma


UTENTE
SERVER WEB
SERVER API
SERVER AUTENTICAZIONE
DATABASE
IDENTITY PROVIDER
Richiesta del sito
Risposta del server
L'utente clicca su "Entra con SPID" o "Entra con CIE"
L'utente viene reindirizzato alla pagina di accesso dell'Identity Provider selezionato
L'utente dopo aver eseguito l'accesso viene reindirizzato al server di autenticazione
Viene creata una nuova sessione nel database
Risposta del database
Viene impostato un cookie con l'identificativo della sessione e l'utente viene reindirizzato al sito web
L'utente richiede i dettagli della sessione
Vengono selezionati i dati relativi alla sessione
Risposta del database
Risposta dell'API
L'utente è ora in grado di navigare nel sito con il suo account