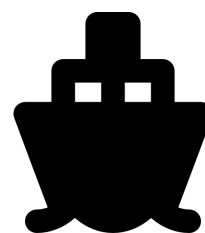
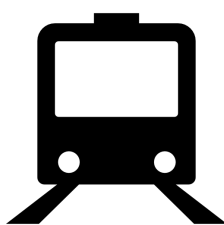


Tesina maturità 2017



Sistema informatico per la gestione dei sistemi di trasporto pubblico



Introduzione (Versione italiana)

Con l'evoluzione delle tecnologie informatiche abbiamo assistito a numerosi cambiamenti, in particolare nei sistemi di amministrazione, dove queste tecnologie hanno rimpiazzato i vecchi sistemi cartacei, rendendo il sistema più efficiente, efficace e veloce. E-transport mira a replicare questo cambiamento nei sistemi di trasporti pubblici, ispirandosi a modelli preesistenti come il sistema di trasporti londinese. Il sistema E-transport prevede la sostituzione dei biglietti cartacei con delle schede elettroniche NFC e delle obliterate con dei lettori NFC a bordo; in questo modo si risparmiano notevoli quantità di carta, si velocizza l'ingresso a bordo dei passeggeri e si riduce notevolmente il numero di passeggeri clandestini.

Il sistema è stato progettato per gestire i seguenti mezzi di trasporto:

- autobus urbani;
- autobus extraurbani;
- metro;
- treni;
- tram;
- traghetti.

La carta di trasporto è unica, un passeggero può quindi usare la sua carta su qualsiasi mezzo.

Funzionamento

Per poter utilizzare qualsiasi mezzo, un cliente ha bisogno di una carta di trasporto. Questa può essere acquistata da un qualsiasi rivenditore autorizzato (edicole, tabacchini, etc...). Una volta acquistata, prima di utilizzarla, il cliente dovrà caricarla con dei soldi, azione che è possibile effettuare dai rivenditori autorizzati, negli sportelli automatici alle stazioni o online. Per poterla ricaricare online è però necessario un account. Il cliente può registrare il suo account sul sito e registrare la sua carta all'account (è possibile registrare più di una carta ad un account ma una carta può essere associata ad un solo account). Una volta registrata vengono messe a disposizione le operazioni di gestione: visione del saldo della carta, ricarica della carta e trasferimento di soldi da una carta ad un'altra (utile anche in caso di smarrimento, entrambe le carte devono essere registrate allo stesso account). Il modo in cui viene usata la carta e con cui vengono calcolati i costi cambia da mezzo a mezzo:

- negli autobus urbani il costo viene calcolato per ora, il passeggero, nel momento in cui sale sul mezzo, poggia la carta sul lettore NFC e gli viene scalato il costo dell'ora dal saldo. Allo scadere dell'ora il passeggero dovrà appoggiare nuovamente la carta sul lettore NFC che provvederà a scalare nuovamente il costo dell'ora dal saldo (se l'ora non è scaduta non viene scalato nulla). Il costo può anche variare da città a città;
- negli autobus extraurbani il costo viene calcolato in ai chilometri percorsi, con un sistema di fasce. Il passeggero poggia la sua carta sul lettore NFC all'ingresso sull'autobus e all'uscita, quando verrà calcolato la fascia di appartenenza e scalato il totale dal saldo della carta. Se il passeggero non poggia la carta in uscita, quando l'autista provvederà a spegnere il lettore, gli

verrà scalato il costo della fascia più alta dal saldo della carta;

- nella metro il costo viene calcolato in base alle zone della città attraversate (la città viene divisa in zone numerate a seconda della distanza dal centro, a partire dalla zona 1). Il passeggero poggia la sua carta sul lettore NFC situato nel tornello all'ingresso della stazione quando entra e quando esce. Se il passeggero non poggia la sua carta all'uscita, gli verrà scalato il costo della tratta più costosa dal saldo della carta;
- nei treni il costo viene calcolato in base alla città origine e alla città destinazione. Il passeggero poggia la sua carta sul lettore NFC all'ingresso sul treno e all'uscita dal treno. Anche in questo caso, se il passeggero non poggia la sua carta all'uscita gli verrà scalato il costo più alto tra tutte le tratte;
- il tram funziona nello stesso modo dell'autobus urbano: il prezzo viene calcolato per ora e il passeggero deve poggiare la sua carta solo all'ingresso;
- I traghetti funzionano invece nello stesso modo dei treni: il prezzo viene calcolato in base all'origine e alla destinazione e il passeggero deve poggiare la carta sia all'ingresso che all'uscita (questo permette ai traghetti di fare più fermate e quindi permette i traghetti turistici urbani).

Eventuali infrazioni possono essere individuate dai controllori che, attraverso un dispositivo dotato di lettore NFC, possono controllare le carte dei passeggeri ed applicare sanzioni. Gli autisti all'inizio del loro turno accendono il lettore NFC con le loro credenziali e le informazioni essenziali per il mezzo e la tratta percorsa. Un cliente registrato può inoltre acquistare abbonamenti online. Una volta acquistato l'abbonamento si lega al suo account fino alla scadenza. Il lettore NFC, prima di scalare dal saldo della carta, controlla se un eventuale abbonamento è attivo.

Il database

In questa tesina non verrà discussa alcuna componente fisica del sistema (schede, lettori) ma sarà incentrata sul sistema informatico su cui tutto è basato. Per poter parlare del sistema informatico è necessario iniziare dalla base di dati, il cuore del sistema. La base di dati è progettata per salvare le informazioni di:

- utenti registrati;
- autisti;
- lettori NFC;
- schede elettroniche;
- abbonamenti;
- tratte dei vari mezzi di trasporto;
- ingressi e uscite degli autisti, per contrastare assenteismo e ritardi;
- transazioni delle schede.

Schema relazionale

Associazioni:

- Effettuato da(M:1)
- Effettuato con(M:1)
- Effettuato sulla(M:1)
- Registrata da(M:1)
- Registra(M:1)
- Posseduta da(M:1)
- Riferito a(M:1)
- Possiede(M:1)
- Aggiunge(M:1)
- Associato a(1:1)
- Risiede a(1:1)
- Residente a (1:1)

Schemi di relazione:

Tratta (id: intero, costo: *decimale*);

Tratta_urbano (citta: *stringa(50)*, id_tratta: intero);

Tratta_extraurbano (da_km: *intero*, a_km: *intero*, id_tratta: intero);

Tratta_metro (da_zona: *intero*, a_zona: *intero*, id_tratta: intero);

Tratta_treno (da: *stringa(50)*, a: *stringa(50)*, id_tratta: intero);

Tratta_tram (citta: *stringa(50)*, id_tratta: intero);

Tratta_traghetto (da: *stringa(50)*, a: *stringa(50)*, id_tratta: intero);

Transazione (id: intero, timestamp: *timestamp*, tipo: *enumerativo("in", "out")*, id_tratta: intero, numero_lettore: stringa(25), numero_carta: stringa(16));

LettoreNFC (numero_serie: stringa(25), data_installazione: *data*);

Carta_trasporto (numero_carta: stringa(16), saldo: *decimale*, data_rilascio: *data*, id_proprietario: intero);

Presenza (id: intero, timestamp: *timestamp*, tipo: *enumerativo("in", "out")*, id_autista: intero, numero_lettore: stringa(25));

Autista (id: intero, nome: *stringa(25)*, cognome: *stringa(25)*, data_nascita: *data*, iban: *stringa(34)*, email: *stringa(254)*, numero_documento: *stringa(9)*, id_indirizzo: intero);

Cliente (id: intero, nome: *stringa(25)*, cognome: *stringa(25)*, email: *stringa(254)*, password: *stringa(32)*, numero_documento: *stringa(9)*, id_indirizzo: intero);

Metodo_pagamento (id: intero, tipo_carta: *enumerativo("visa", "mastercard", "postepay", "maestro", "american_express")*, numero_carta: *stringa(16)*, nome_titolare: *stringa(25)*, data_scadenza: *data*, id_indirizzo: intero, id_cliente: intero);

Indirizzo (id: intero, via: *stringa(25)*, provincia: *stringa(25)*, citta: *stringa(25)*, cap: *stringa(5)*);

Abbonamento (id: intero, data_scadenza: *data*, id_tratta: intero, id_proprietario: intero);

Implementazione SQL

```
CREATE TABLE Tratta (  
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    costo DECIMAL UNSIGNED NOT NULL,  
    PRIMARY KEY (id)  
) ENGINE = INNODB;
```

```
CREATE TABLE Tratta_urbano (  
    citta VARCHAR(50) NOT NULL UNIQUE,  
    id_tratta INT UNSIGNED NOT NULL,  
    PRIMARY KEY (id_tratta),  
    FOREIGN KEY (id_tratta)  
        REFERENCES Tratta(id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
) ENGINE = INNODB;
```

```
CREATE TABLE Tratta_extraurbano (  
    da_km SMALLINT UNSIGNED NOT NULL,  
    a_km SMALLINT UNSIGNED NOT NULL,  
    id_tratta INT UNSIGNED NOT NULL,  
    PRIMARY KEY (id_tratta),  
    UNIQUE (da_km, a_km),  
    FOREIGN KEY (id_tratta)  
        REFERENCES Tratta(id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
) ENGINE = INNODB;
```

```
CREATE TABLE Tratta_metro (  
    da_zona TINYINT UNSIGNED NOT NULL,  
    a_zona TINYINT UNSIGNED NOT NULL,  
    id_tratta INT UNSIGNED NOT NULL,  
    PRIMARY KEY (id_tratta),  
    UNIQUE (da_zona, a_zona),  
    FOREIGN KEY (id_tratta)  
        REFERENCES Tratta(id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
) ENGINE = INNODB;
```

```

CREATE TABLE Tratta_treno (
    da VARCHAR(50) NOT NULL,
    a VARCHAR(50) NOT NULL,
    id_tratta INT UNSIGNED NOT NULL,
    PRIMARY KEY (id_tratta),
    UNIQUE (da, a),
    FOREIGN KEY (id_tratta)
        REFERENCES Tratta(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE = INNODB;

```

```

CREATE TABLE Tratta_tram (
    citta VARCHAR(50) NOT NULL UNIQUE,
    id_tratta INT UNSIGNED NOT NULL,
    PRIMARY KEY (id_tratta),
    FOREIGN KEY (id_tratta)
        REFERENCES Tratta(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE = INNODB;

```

```

CREATE TABLE Tratta_traghetto (
    da VARCHAR(50) NOT NULL,
    a VARCHAR(50) NOT NULL,
    id_tratta INT UNSIGNED NOT NULL,
    PRIMARY KEY (id_tratta),
    UNIQUE (da, a),
    FOREIGN KEY (id_tratta)
        REFERENCES Tratta(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE = INNODB;

```

```

CREATE TABLE Indirizzo(
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    via VARCHAR(25) NOT NULL,
    provincia VARCHAR(25) NOT NULL,
    citta VARCHAR(25) NOT NULL,
    cap VARCHAR(5) NOT NULL,
    PRIMARY KEY(id)
) ENGINE = INNODB;

```

```

CREATE TABLE Cliente (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    nome VARCHAR(25) NOT NULL,
    cognome VARCHAR(25) NOT NULL,
    email VARCHAR(254) NOT NULL UNIQUE,
    password VARCHAR(32) NOT NULL,
    numero_documento VARCHAR(9) NOT NULL UNIQUE,
    id_indirizzo INT UNSIGNED NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id_indirizzo)
        REFERENCES Indirizzo(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE = INNODB;

```

```

CREATE TABLE Transazione (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    timestamp TIMESTAMP NOT NULL,
    tipo ENUM("in", "out") NOT NULL,
    id_tratta INT UNSIGNED NOT NULL,
    numero_lettore VARCHAR(25) NOT NULL,
    numero_carta VARCHAR(16) NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id_tratta)
        REFERENCES Tratta(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (numero_lettore)
        REFERENCES LettoreNFC(numero_serie)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (numero_carta)
        REFERENCES Carta_trasporto(numero_carta)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE = INNODB;

```

```

CREATE TABLE LettoreNFC (
    numero_serie VARCHAR(25) NOT NULL,
    data_installazione DATE NOT NULL,
    PRIMARY KEY (numero_serie)
) ENGINE = INNODB;

```



```

CREATE TABLE Presenza (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    timestamp TIMESTAMP NOT NULL,
    tipo ENUM("in", "out") NOT NULL,
    id_autista INT UNSIGNED NOT NULL,
    numero_letture VARCHAR(25) NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id_autista)
        REFERENCES Autista(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (numero_letture)
        REFERENCES LettureNFC(numero_serie)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE = INNODB;

CREATE TABLE Carta_trasporto (
    numero_carta VARCHAR(16) NOT NULL,
    saldo DECIMAL NOT NULL,
    data_rilascio DATE NOT NULL,
    id_proprietario INT UNSIGNED,
    PRIMARY KEY (numero_carta),
    FOREIGN KEY (id_proprietario)
        REFERENCES Cliente(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE = INNODB;

CREATE TABLE Autista (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    nome VARCHAR(25) NOT NULL,
    cognome VARCHAR(25) NOT NULL,
    data_nascita DATE NOT NULL,
    iban VARCHAR(34) NOT NULL,
    email VARCHAR(254) NOT NULL,
    numero_documento VARCHAR(9) NOT NULL UNIQUE,
    id_indirizzo INT UNSIGNED NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id_indirizzo)
        REFERENCES Indirizzo(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE = INNODB;

```

```

CREATE TABLE Metodo_pagamento (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    tipo_carta ENUM("visa", "mastercard", "postepay", "maestro", "american_express") NOT NULL,
    numero_carta VARCHAR(16) NOT NULL UNIQUE,
    nome_titolare VARCHAR(25) NOT NULL,
    data_scadenza DATE NOT NULL,
    id_indirizzo INT UNSIGNED NOT NULL,
    id_cliente INT UNSIGNED NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id_indirizzo)
        REFERENCES Indirizzo(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (id_cliente)
        REFERENCES Cliente(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE = INNODB;

```

```

CREATE TABLE Abbonamento (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    data_scadenza DATE NOT NULL,
    id_tratta INT UNSIGNED NOT NULL,
    id_proprietario INT UNSIGNED NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id_tratta)
        REFERENCES Tratta(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (id_proprietario)
        REFERENCES Cliente(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE = INNODB;

```

Alcune query

Saldo totale per cliente:

```
SELECT SUM(saldo) AS Saldo, id_proprietario AS "Id proprietario"
FROM Carta_trasporto
GROUP BY id_proprietario;
```

Totale speso per carta in un determinato periodo:

```
SELECT SUM(Tratta.costo) AS "Spesa totale", Transazione.numero_carta AS "Numero carta"
FROM Transazione INNER JOIN Tratta
  ON Transazione.id_tratta = Tratta.id
WHERE DATE(Transazione.timestamp) BETWEEN @data1 AND @data2
GROUP BY Transazione.numero_carta;
```

Abbonamenti non scaduti di un cliente di cui conosciamo l'email:

```
SELECT * AS "Abbonamenti non scaduti"
FROM abbonamento INNER JOIN cliente
  ON abbonamento.id_proprietario = cliente.id
WHERE cliente.email = @email AND
  abbonamento.data_scadenza <= CURRDATE();
```

Il sito web

Altra parte importante del sistema è il sito web. Nonostante l'utente può tranquillamente usufruire dei servizi senza di esso, il sito è una risorsa essenziale attraverso la quale è possibile informarsi su eventuali cambiamenti ed imprevisti, costi delle tratte, modalità di funzionamento del servizio e gestione di account e carta trasporto.

Eventuali espansioni

Il sistema può essere espanso in diversi modi. Per esempio, un applicazione per dispositivi mobili permetterebbe l'utente di controllare facilmente tariffe e il saldo corrente direttamente dal telefono e fuori casa e di ricaricare la carta al volo. Un'altra espansione potrebbe essere la gestione degli orari direttamente nel sistema; in questo modo l'utente può controllare gli orari dei mezzi direttamente dal sito senza doversi rivolgere a servizi esterni, come google maps o volantini affissi alle fermate. Questo permetterebbe inoltre un'espansione dell'applicazione per dispositivi mobili che potrebbe funzionare anche da allarme; l'utente segna la destinazione e l'ora a cui deve essere lì e l'applicazione calcola le tratte migliori e avvisa l'utente quando è il momento di uscire di casa.

Introduction (English version)

With the evolution of the IT sector we've seen different changes, in particular concerning management systems, where new technologies replaced the old paper bases system with a more fast, efficient and effective one. E-transport aims to bring those new technologies to the public transport systems, using a model similar to the one used in London, replacing paper tickets with electronic travel cards, powered by NFC technology, and replacing punching machines with NFC readers; this way we can achieve a reduced waste of paper, faster boarding of the passengers and reduce the frequency of clandestine passengers.

The system is designed to handle the following means of transport:

- City busses;
- Intercity busses;
- Metro;
- Trains;
- Trams;
- Ferries.

All of those share the same travel card, so the passengers can use their own card with every mean of transport.

How it works

In order to use the transport system, you need a travel card. The travel card can be bought from a authorized retail outlet. Once bought, you also need to top up the card with money, action that you can perform directly in the retail outlet where you bought the card, with a terminal at the station or online. You can create an account on the E-transport website and bind your card to your account. This way you get access to all of the online operations like: viewing the travel card credit, topping up the card and transferring the creding between two cards (they both need to be bound to the same account). The fares and the way you're supposed to use the card change based of the mean of transport:

- city busses' fares are defined per hour and per city. When you get on the bus you have to touch the card on the reader, which will provide to deduct one hour worth from the card's total credit. At the end of that hour, if you want to continue to use the bus, you'll have to touch on again (the reader won't deduct anything if the hour isn't over) ;
- intercity busses' fares are calculeted with the travelled distance, with a system bases on intervals. You have to touch on when you board on the bus and touch off when you leave. When you touch off, the reader will calculate the cost of the journey and will provide to deduct it from the card's credit. If you don't touch of, the reader will provide to deduct the most expensive fare from the card's credit at the end of a driver's shift;
- metro's fares are calculated with the city zone where you touched on and the city zone where

you touched off (in the city, different zones or areas are defined bases of their distance from the city center, defined as zone 1). You have to touch on when you enter the station and touch off when you leave it, if you don't touch off, you'll get deducted the most expensive fee from the card's credit;

- trains' fares are calculated with the city where you touched on and the city where you touched off. You have to touch on when you enter the train and touch off when you leave it. If you don't touch off, you'll get deducted the most expensive fee from the card's credit;
- trams work the same way as city busses do: you have to touch on when you enter and the fares are calculated per hour;
- ferries work the same way as trains do: you have to touch on when you board and touch off when you leave and the fares are calculated with your origin and destination.

Infractions are punished with sanctions and can be spotted by the inspectors, who are equipped with an handheld NFC reader, used to check the passengers' cards. At the start of his/her shift, the driver sets up the onboard NFC reader with his/her own credentials and the route's information. You can also buy passes for all the different means of transport available. Every pass that you buy will be bound to your account and the reader will check for an active pass before charging the card.

The database

This thesis will not discuss about the fisical component of the system (travel cards and NFC readers) but will discuss about the informatic system that is behind everything. The most important part of the informatic system is the database, the heart of everything. The database was designed to handle the information of:

- registered users;
- drivers;
- NFC readers;
- travel cards;
- passes;
- routes;
- start and end of drivers' shifts;
- cards' transactions.

Relazional schema

Relationships:

- Made from(M:1)
- Made with(M:1)
- Made on(M:1)
- Registrata da(M:1)
- Registra(M:1)
- Owned by(M:1)
- Refers to(M:1)
- Owns(M:1)
- Adds(M:1)
- Associated with(1:1)
- Resides at(1:1)
- Is residing at(1:1)

Relational scheme:

Route (id: integer, cost: decimal);

Citybus_route (city: string(50), route_id: integer);

Intercity_bus_route (from_km: integer, to_km: integer, route_id: integer);

Metro_route (from_zone: integer, to_zone: integer, route_id: integer);

Train_route (from: string(50), to: string(50), route_id: integer);

Tram_route (city: string(50), route_id: integer);

Ferry_route (from: string(50), to: string(50), route_id: integer);

Transaction (id: integer, timestamp: timestamp, type: enumerative("on", "off"), route_id: integer, reader_number: string(25), card_number: string(16));

NFCReader (serial_number: string(25), installation_date: date);

Travel_card (card_number: string(16), credit: decimal, release_date: date, owner_id: integer);

Shift (id: integer, timestamp: timestamp, type: enumerative("start", "end"), driver_id: integer, reader_number: string(25));

Driver (id: integer, name: string(25), surname: string(25), birthdate: date, iban: string(34), email: string(254), document_number: string(9), address_id: integer);

User (id: integer, name: *string(25)*, surname: *string(25)*, email: *string(254)*, password: *string(32)*, document_number: *string(9)*, address_id: *integer*);

Payment_method (id: integer, card_type: *enumerative("visa", "mastercard", "postepay", "maestro", "american_express")*, card_number: *string(16)*, owner_name: *string(25)*, expiration_date: *date*, address_id: *integer*, user_id: *integer*);

Address (id: integer, street: *string(25)*, province: *string(25)*, city: *string(25)*, postal_code: *string(5)*);

Pass (id: integer, : expiration_date: *date*, route_id: *integer*, owner_id: *integer*);

SQL Implementation

```
CREATE TABLE Route (  
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    cost DECIMAL UNSIGNED NOT NULL,  
    PRIMARY KEY (id)  
) ENGINE = INNODB;
```

```
CREATE TABLE Citybus_route (  
    city VARCHAR(50) NOT NULL UNIQUE,  
    route_id INT UNSIGNED NOT NULL,  
    PRIMARY KEY (route_id),  
    FOREIGN KEY (route_id)  
        REFERENCES Route(id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
) ENGINE = INNODB;
```

```
CREATE TABLE Intercity_bus_route(  
    from_km SMALLINT UNSIGNED NOT NULL,  
    to_km SMALLINT UNSIGNED NOT NULL,  
    route_id INT UNSIGNED NOT NULL,  
    PRIMARY KEY (route_id),  
    UNIQUE (from_km, to_km),  
    FOREIGN KEY (route_id)  
        REFERENCES Route(id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
) ENGINE = INNODB;
```

```
CREATE TABLE Metro_route (  
    from_zone TINYINT UNSIGNED NOT NULL,  
    to_zone TINYINT UNSIGNED NOT NULL,  
    route_id INT UNSIGNED NOT NULL,  
    PRIMARY KEY (route_id),  
    UNIQUE (from_zone, to_zone),  
    FOREIGN KEY (route_id)  
        REFERENCES Route(id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
) ENGINE = INNODB;
```

```

CREATE TABLE Train_route (
    from VARCHAR(50) NOT NULL,
    to VARCHAR(50) NOT NULL,
    route_id INT UNSIGNED NOT NULL,
    PRIMARY KEY (route_id),
    UNIQUE (from, to),
    FOREIGN KEY (route_id)
        REFERENCES Route(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE = INNODB;

CREATE TABLE Tram_route (
    city VARCHAR(50) NOT NULL UNIQUE,
    route_id INT UNSIGNED NOT NULL,
    PRIMARY KEY (route_id),
    FOREIGN KEY (route_id)
        REFERENCES Route(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE = INNODB;

CREATE TABLE Ferry_route (
    from VARCHAR(50) NOT NULL,
    to VARCHAR(50) NOT NULL,
    route_id INT UNSIGNED NOT NULL,
    PRIMARY KEY (route_id),
    UNIQUE (from, to),
    FOREIGN KEY (route_id)
        REFERENCES Route(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE = INNODB;

CREATE TABLE Travel_card (
    card_number VARCHAR(16) NOT NULL,
    credit DECIMAL NOT NULL,
    release_date DATE NOT NULL,
    owner_id INT UNSIGNED,
    PRIMARY KEY (card_number),
    FOREIGN KEY (owner_id)
        REFERENCES User(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE = INNODB;

```

```

CREATE TABLE Transaction (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    timestamp TIMESTAMP NOT NULL,
    type ENUM("on", "off") NOT NULL,
    route_id INT UNSIGNED NOT NULL,
    reader_number VARCHAR(25) NOT NULL,
    card_number VARCHAR(16) NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (route_id)
        REFERENCES Route(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (reader_number)
        REFERENCES NFCReader(serial_number)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (card_number)
        REFERENCES Travel_card(card_number)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE = INNODB;

```

```

CREATE TABLE Shift (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    timestamp TIMESTAMP NOT NULL,
    type ENUM("start", "end") NOT NULL,
    driver_id INT UNSIGNED NOT NULL,
    reader_number VARCHAR(25) NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (driver_id)
        REFERENCES Driver(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (reader_number)
        REFERENCES NFCReader(numero_serie)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE = INNODB;

```

```

CREATE TABLE NFCReader (
    serial_number VARCHAR(25) NOT NULL,
    installation_date DATE NOT NULL,
    PRIMARY KEY (serial_number)
) ENGINE = INNODB;

CREATE TABLE Driver (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    name VARCHAR(25) NOT NULL,
    surname VARCHAR(25) NOT NULL,
    birthdate DATE NOT NULL,
    iban VARCHAR(34) NOT NULL,
    email VARCHAR(254) NOT NULL,
    document_number VARCHAR(9) NOT NULL UNIQUE,
    address_id INT UNSIGNED NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (address_id)
        REFERENCES Address(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE = INNODB;

CREATE TABLE Usser (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    name VARCHAR(25) NOT NULL,
    surname VARCHAR(25) NOT NULL,
    email VARCHAR(254) NOT NULL UNIQUE,
    password VARCHAR(32) NOT NULL,
    document_number VARCHAR(9) NOT NULL UNIQUE,
    address_id INT UNSIGNED NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (address_id)
        REFERENCES Address(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE = INNODB;

CREATE TABLE Address(
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    street VARCHAR(25) NOT NULL,
    province VARCHAR(25) NOT NULL,
    city VARCHAR(25) NOT NULL,
    postal_code VARCHAR(5) NOT NULL,
    PRIMARY KEY(id)
) ENGINE = INNODB;

```

```

CREATE TABLE Payment_method (
  id INT UNSIGNED NOT NULL AUTO_INCREMENT,
  card_type ENUM("visa", "mastercard", "postepay", "maestro", "american_express") NOT NULL,
  card_number VARCHAR(16) NOT NULL UNIQUE,
  owner_name VARCHAR(25) NOT NULL,
  expiration_date DATE NOT NULL,
  address_id INT UNSIGNED NOT NULL,
  user_id INT UNSIGNED NOT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY (address_id)
    REFERENCES Address(id)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY (user_id)
    REFERENCES User(id)
    ON DELETE CASCADE
    ON UPDATE CASCADE
) ENGINE = INNODB;

```

```

CREATE TABLE Pass (
  id INT UNSIGNED NOT NULL AUTO_INCREMENT,
  expiration_date DATE NOT NULL,
  route_id INT UNSIGNED NOT NULL,
  owner_id INT UNSIGNED NOT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY (route_id)
    REFERENCES Route(id)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY (owner_id)
    REFERENCES User(id)
    ON DELETE CASCADE
    ON UPDATE CASCADE
) ENGINE = INNODB;

```

Some queries

Total credit of a user:

```
SELECT SUM(credit) AS Credit, owner_id AS "Owner's id"
FROM Travel_card
GROUP BY owner_id;
```

Total amount spent in a predetermined time period:

```
SELECT SUM(Route.cost) AS "Total spent", Transaction.card_number AS "Card number"
FROM Transaction INNER JOIN Route
    ON Transaction.route_id = Route.id
WHERE DATE(Transaction.timestamp) BETWEEN @date1 AND @date2
GROUP BY Transaction.card_number;
```

Active passes of a user whose email is known:

```
SELECT * AS "Active passes"
FROM pass INNER JOIN user
    ON pass.owner_id = user.id
WHERE user.email = @email AND
    pass.expiration_date <= CURRDATE();
```

The website

Another essential component of the system is the website. Even if the user can use the system and his/her card freely without it, the website is essential for getting to know the changes of the system, routes' fares, the way the system works and to manage his/her account and travel cards.

Future additions

In the future, new components can be added to the system. For exemple a mobile app that lets the user enjoy a way to check fares and current credit of his/her card on the go. Another component might be the manament of the time schedules for the different means of transport directly on the site, without relaying on external services.