Faculdade de Engenharia da Universidade do Porto

Mestrado Integrado em Engenharia Informática e Computação

Mobile Computing

Mobile Computing Project Report

Autores:
Diogo Reis - up2015——
Tiago Magalhães - up201607931

November 2019

Contents

1 Architecture

1.1 Server

1.1.1 REST WebAPI

The server uses a REST WebAPI provided by the ASP.NET framework to handle all the communication with it. All routes are present in a single controller under the router of server.

1.1.2 Database

Communication with the PostgreSQL database is done using the Npgsql package, it is then abstracted into a singleton class that provides the common insert and select operation. Prepared statements are done via an entry system where each entry has a name, a value and a boolean indicating if the entry is a UUID or not, the entry names are then matched to the query markers for parameters.

All responses to select queries are done via list of dictionaries that map strings to object, each element in the list is a row in the database, and the elements of the dictionaries contain the values of the various columns in the database, the entries have the same name as they do in the database schema.

1.1.3 RSA Encryption & Signing

1.2 Store And Client Applications

1.2.1 HTTP Requests

1.2.2 QR Codes

1.2.3 RSA Encryption & Signing

1.2.4 Data Caching

2 Data Scheme

2.1 Database Data Schema

```
CREATE EXTENSION IF NOT EXISTS "uuid-ossp";

DROP TABLE IF EXISTS Client CASCADE;
CREATE TABLE Client(
        id uuid primary key default uuid_generate_v4(),
        name text not null,
        username text not null,
        password text not null,
        credit_card int not null,
        public_key text not null,
        current_total_spent_euro INTEGER not null default 0,
        current_total_spent_cent INTEGER not null default 0,
        current_accumulated_euro INTEGER not null default 0,
        current_accumulated_cent INTEGER not null default 0
);

DROP TABLE IF EXISTS Voucher CASCADE;
CREATE TABLE Voucher(
        id uuid primary key default uuid_generate_v4(),
        client uuid not null REFERENCES Client(id),
        was_used BOOLEAN not null DEFAULT FALSE
);

DROP TABLE IF EXISTS Purchase CASCADE;
CREATE TABLE Purchase(
        id uuid primary key default uuid_generate_v4(),
        client uuid not null REFERENCES Client(id),
        voucher uuid REFERENCES Voucher(id) DEFAULT NULL,
        should_discount BOOLEAN not null DEFAULT false
);

DROP TABLE IF EXISTS Product CASCADE;
CREATE TABLE Product(
        id uuid primary key default uuid_generate_v4(),
        price_euro INTEGER not null,
        price_cent INTEGER not null,
        name text not null,
        image_url text DEFAULT NULL,
        purchase uuid REFERENCES Purchase(id) DEFAULT null
);
```

2.1.1 Client

This table contains information about users of the platform and has the following fields:

- id - UUID representing the identification of the user.
- name - String representing the name of the user.
- username - String representing the username or nickname of the user.
- password - String representing the password of the user.
- credit_card - Integer representing the credit card number of the user.
- public_key - String containing the users RSA public key.
- current_total_spent_euro - The amount of money spent by the user, this is the euro component.
- current_total_spent_cent - The amount of money spent by the user, this is the cent component.
- current_accumulated_euro - The amount of money the user has accumulated from voucher, this is the euro component.
- current_accumulated_cent - The amount of money the user has accumulated from voucher, this is the cent component.

2.1.2 Voucher

This table contains the voucher registered in the system and has the following details:

- id - UUID representing the identification of the voucher.
- client - UUID representing the identification of the user the voucher belongs to.
- was_used - Boolean representing wether or not the voucher has been used, by default this value is false.

2.1.3 Purchase

This table contains information about purchases and has the following details:

- id - UUID representing the identification of the purchase.
- client - UUID representing the identification of the user the purchase is associated with.
- voucher - UUID representing the identification of the voucher used in this purchase, this value is optional and is null by default.
- should_discount - Boolean representing weather or not the cost of the purchase was amortized with money the user had accumulated via vouchers.

2.1.4 Product

This table contains the products registered in the system and has the following information:

- id - UUID representing the identification of the product.
- price_euro - This is the price of the product, this is the euros component.
- price_cent - This is the price of the product, this is the cents component.
- name - String representing the name of the product.
- image_url - String representing the link to an image of the product.
- purchase - UUID representing the purchase this product is in, this is optional and is null by default.

2.2 Checkout Information

2.3 Data Verification Signatures

3 Features & Tests

3.1 Features

3.1.1 Available Product List In Store App

3.1.2 User Registering

3.1.3 User Login

3.1.4 User Login Caching

3.1.5 Adding Products To Cart Through QR Codes

3.1.6 Viewing and Editing Cart

3.1.7 Viewing Past Transaction

3.1.8 Viewing Products Bough In Past Transaction

3.1.9 Viewing Vouchers Available To User And Amount Of Money Usable For Discounting

3.1.10 Checkout

3.2 Testing

4 Usage Manual

4.1 Server

4.2 Store Application

4.2.1 Viewing Available Product List

4.2.2 Checkout

4.3 Client Application

4.3.1 User Registering

4.3.2 User Login

4.3.3 User Login Caching

4.3.4 Adding Products To Cart Through QR Codes

4.3.5 Viewing and Editing Cart

4.3.6 Viewing Past Transaction

4.3.7 Viewing Products Bough In Past Transaction

4.3.8 Viewing Vouchers Available To User And Amount Of Money Usable For Discounting

4.3.9 Checkout