Supermarket Mobile App

Antonio Pereira - up201307910

Hugo Neves - up201104178

Introduction

We were tasked with creating an android app using android studio for evaluation in the CMOV subject. The project can be described as a full QR scanning based supermarket app, focused in improving and facilitating customer satisfaction.

In this App we have user verification while registering and through most part of the App. When trying to do an act that is not correct or enabled it will display an error given by the Server or the Application.

Design

The Scan'O'Mart mobile app was designed with the basic principle of being simple to use. Corporate policy over at Scan'O'Mart headquarters requires the streamlining of all customer services. With this in mind the app was designed to be used by people of all ages, and follow the color pallete of the company.

Architecture

The whole process from scanning to purchase goes through many systems and modules, which we will expand on in this report.

1) The Server

The Server was done in node js with the help of Express.

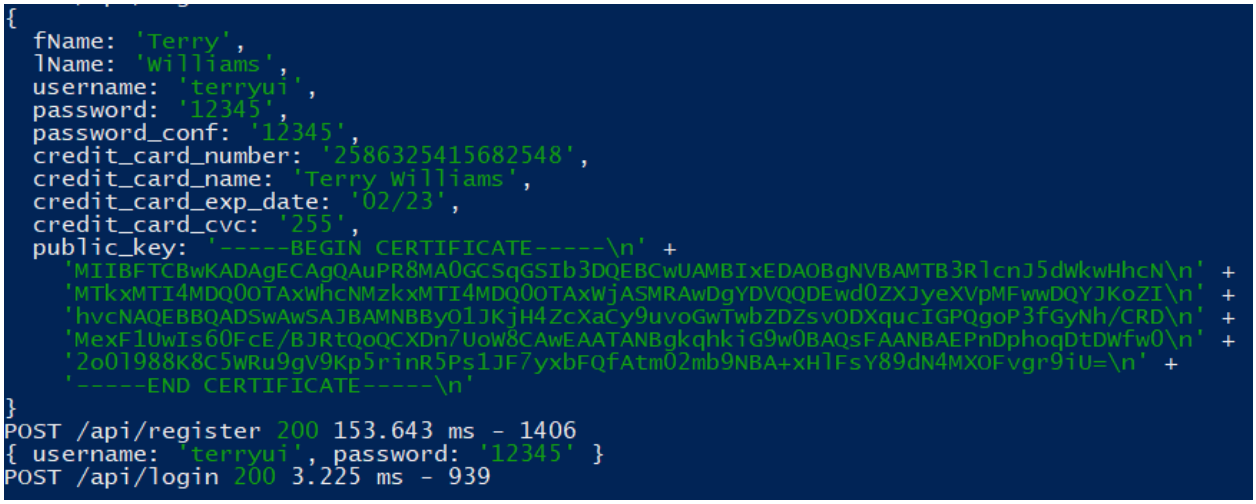
The server establishes all the communication aspects between the app and the terminal, as well as keeping information about the purchasing. The server also serves as a database for the application.

```
PS D:\HARD DRIVE\Documents\FEUP\CMOV\feup-cmov\Server> npm start

> acme@0.0.0 start D:\HARD DRIVE\Documents\FEUP\CMOV\feup-cmov\Server
> set DEBUG=acme & node ./bin/www

  acme Listening on port 3001
Supermarket Server is operational!
```

Example of Server requests being displayed while registering and sign in on the Application.



2) The Terminal App

The terminal app serves as a cashier. The cashier takes place as the middle man in the communication between the Server and the Mobile App while processing a transaction. While making a transaction the App sends the public key of the user through NFC for it to be possible to decode the QR Code after.

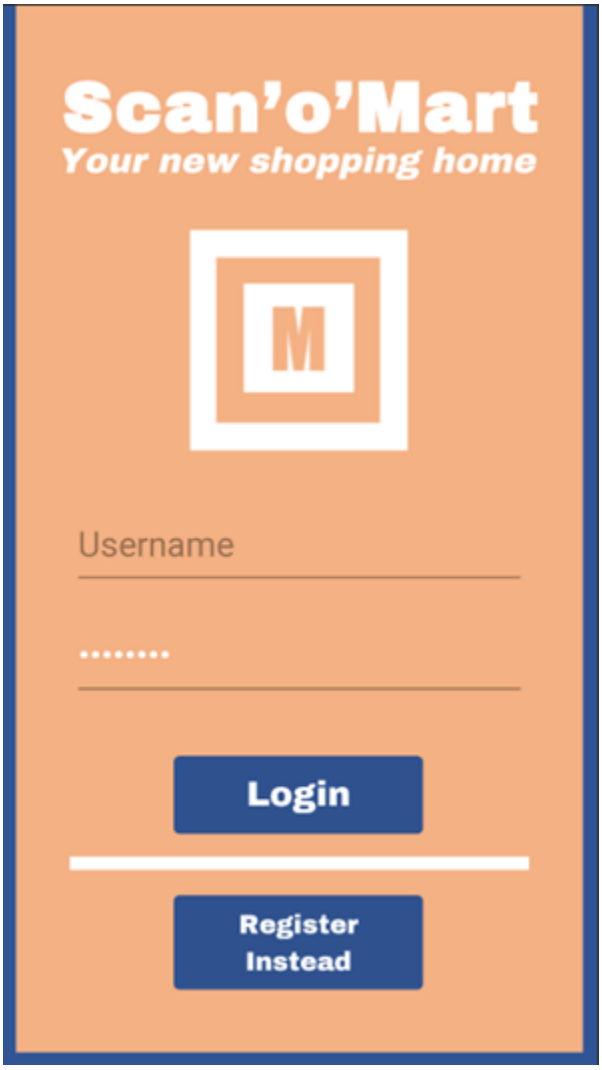
If the Transaction is done correctly it will display a message saying to “Open Terminal Doors”.

3) The Mobile App

The Application can do several actions such as:

- Update Vouchers;
- Create Transaction;

- View Vouchers / Select Vouchers;
- View Profile;
- View Transactions;
- Update Transactions;

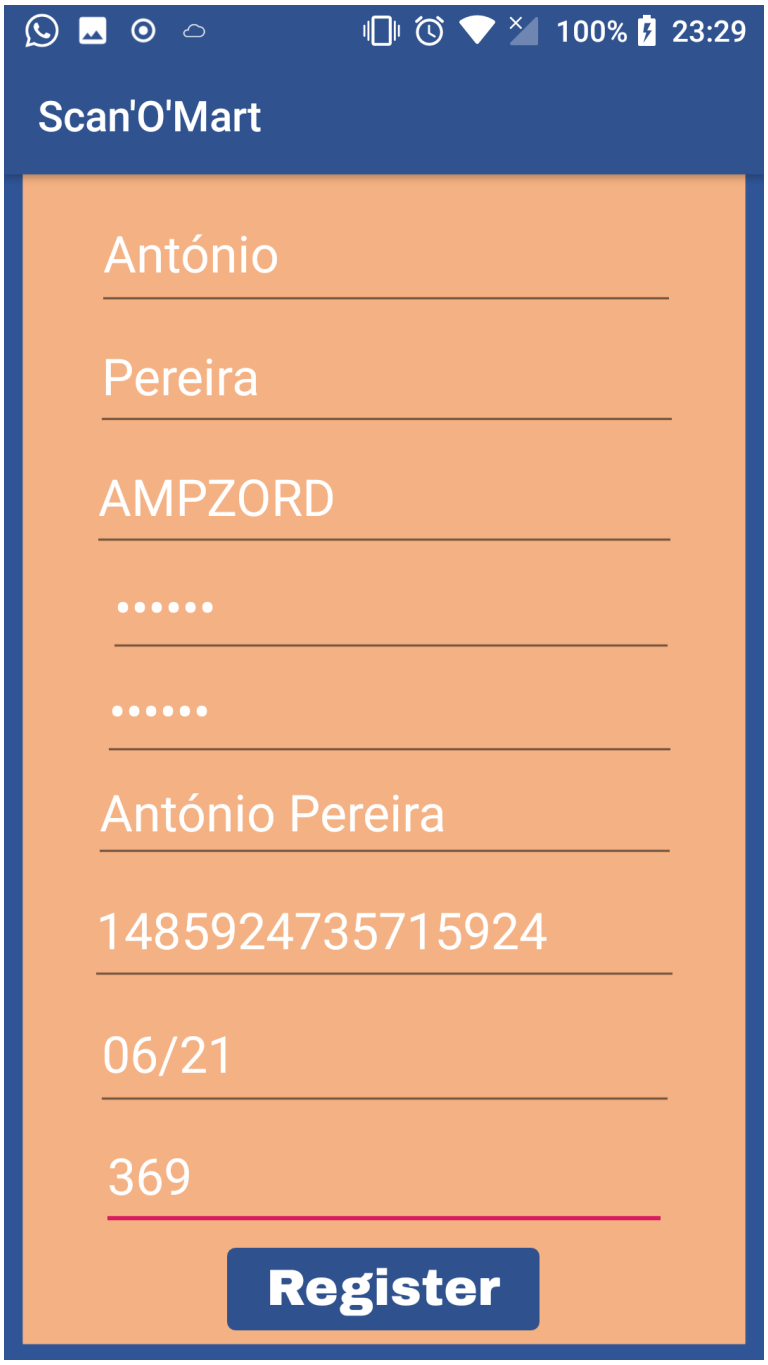


Log in menu

The log in menu serves as a splash page to the app, being the first view the user experiences. In it the user can log in, or instead create a new account

On this menu the user can:

1) Input his username and password and click log in to be logged in the app
2) Register a new account and be logged in the app with that new information



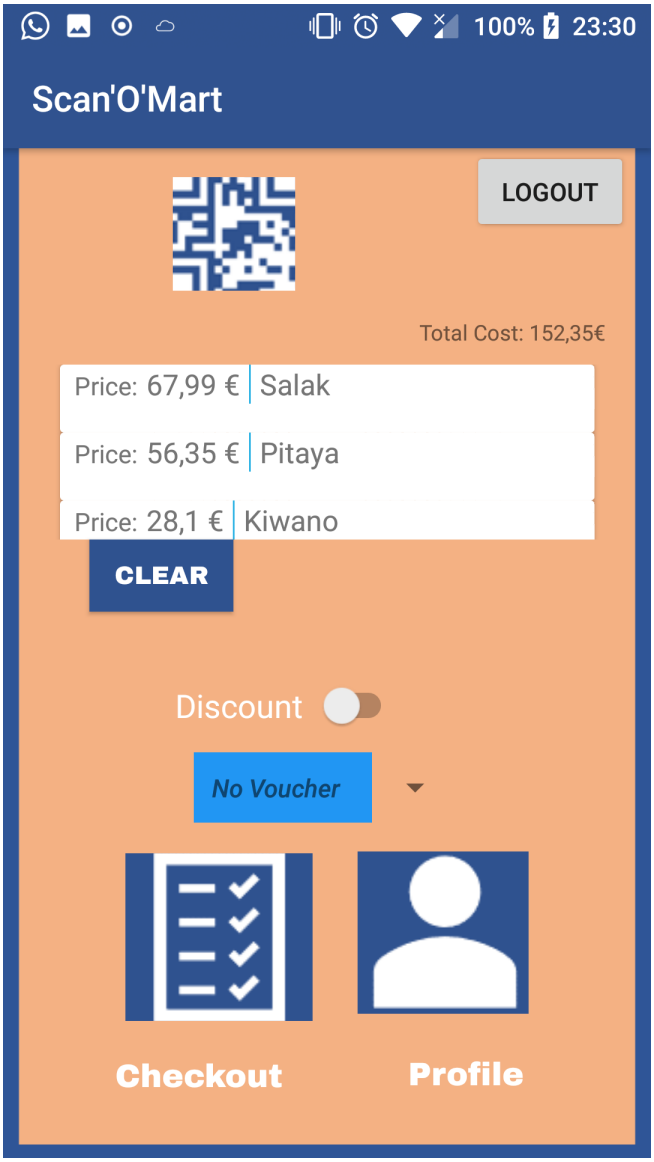
Register menu

Registering is as simple as filling out all the required information on this menu. In addition to regular customer information, we also required a valid credit card template card to be added, seeing we need that data for the core app to function.

On this menu the user can:

1) Input his first name
2) Input his last name
3) Input his username

4) Input his password
5) Input his password again to confirm it
6) Input the name associate with his credit card
7) Input his credit card card number
8) Input his credit card expiration date
9) Input his credit card security code
10) Click to register all this info under a new account



Main Buy Menu

The central menu of the application. This menu allows for complete navigation of the app, and use of all its main features.

Uptop you can see the Scanner button. This initiates the scanner module and allows the app to collect a QR to read an item off of.

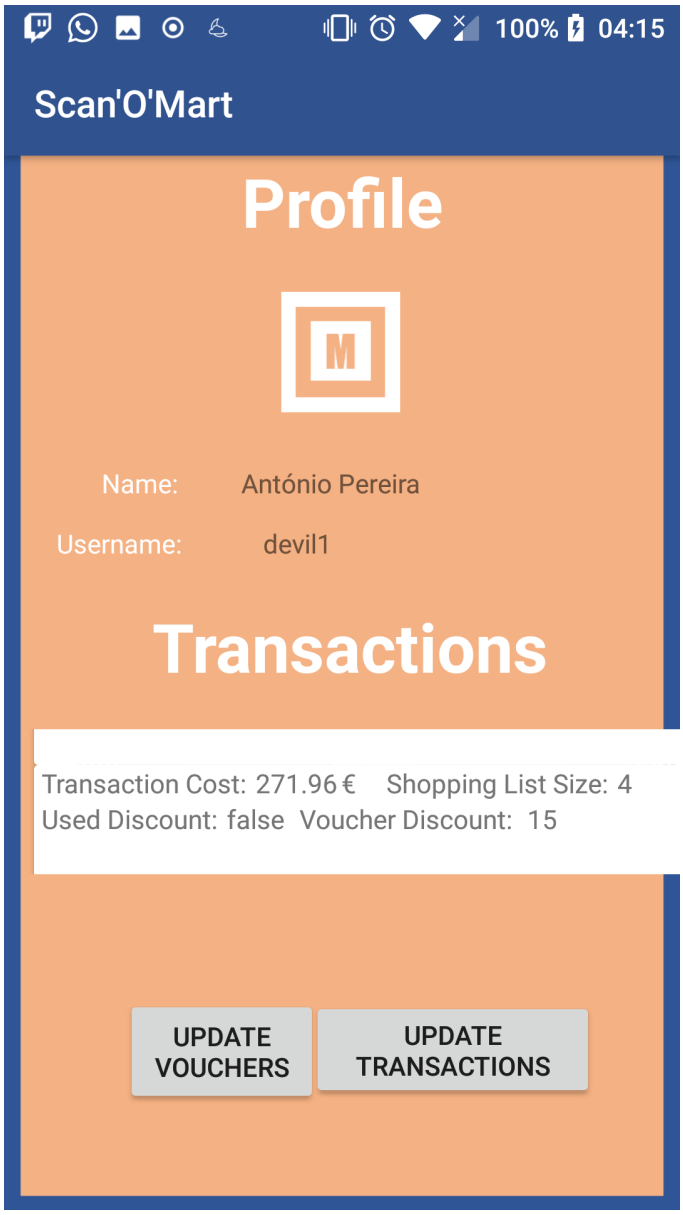
Also on top, but to the sides, we see both the “logout” buttons.

Bellow is the item list, this container holds all the items from the basket.

Next are three buttons, all of them sending us to their respective menus.

On this menu the user can:

1) Click logout to log out and return to the log in menu
2) Click clear to clear all items off of his shopping cart
3) Click checkout to go to the checkout page
4) Click profile to head to his profile page
5) Select if transaction is to discount on the server.
6) Select which voucher to use.



Profile

This menu displays the basic details about the user.

Also displays past transactions of the user.

Here the user can update his voucher list by requesting it to the Server. The same with the Transactions.

Communication Menu

The communication menu is displayed when the user is requested to approach his phone to the NFC terminal in order to complete the purchase.

On this menu the user can:

1) Await for NFC communication to pair and then click the tooltip on screen.

After this will generate a QR Code of the shopping list for the terminal to read.

Checkout Menu

This temporary menu is quickly displayed when generating a QR code to be read by the terminal to establish a secure purchase.

Vouchers

If the user spent less than 100 euros he doesn't receive a voucher. And if he spent between 100-600 euros he will receive a voucher depending on how much he spent. For example

If he spent more than 100 euros and less than 200 euros he will receive a 15% voucher.

200-300 euros he will receive 30% voucher. This will propagate till 60% discount where it will cap.

Security

Security is ensured by the generation of public and private keys. In the registration process both the Application and the Server exchange public keys to be able to decrypt messages from each other.

The QR Code created by the server is after read by the Application with the public key received on the registration process.

When doing a checkout of the shopping list the Application first sends through NFC the user's public key to the terminal.

After the Application will generate an encoded QR Code with the shopping list (voucher, price, etc). The Terminal will read this QR Code and after send the request to the Server.

Conclusions and Final Thoughts

Several of our colleges had already used androids studio before, mostly during the LGP course. However, neither of us two had ever been assigned a project using the software, and we take this experience as a net positive for our understanding of the platform and its capabilities.

Personally, however, we thought android studio to be sluggish and a slow program to work with.

The project itself we feel incopassed a lot of what a “real” app required to be functional. A full server to app interface, complete mobile app experience etc. All these components work together in the ways we expected, and was fun seeing the project slowly be built toward a final working goal.

In the end, we only regret having so many refactoring steps, especially of the main menu, which lead to some pages of the app being more visually fleshed out than others.

All in all a worthwhile experience for both of us.