



# ACME SYSTEM

**Experiencing the Joy of Kotlin  
with this App: Sip, Save, and  
Savor with Every Order!**

Faculdade de  
Engenharia da  
Universidade do Porto

Mobile Computing – MESW

## **Practical Assignment 1**

Garcia Manuel – up202204409

Adalberto Filipe – up202308707

Fabio Huang– up201806829

Bruno Sena– up202300395

Gabriel Gomes- up202202456

Porto, April 2024

# Contents

Introduction.....	4
Architecture .....	5
Domain Model.....	6
User.....	6
Order.....	6
Product.....	6
Order Products .....	5
Receipt.....	5
Voucher.....	7
Discount Voucher .....	7
Coffee Voucher.....	7
Applied Voucher.....	5
Activity Diagram .....	9
System Features .....	10
Acme Café Server Application (Node.js Server).....	10
Acme Café Mobile Application (Customer App).....	11
Login and Registration .....	11
Acme Café Terminal Application (Terminal App).....	16
Order's Management .....	17
Navigation Map .....	20
Testing .....	20
Test Summary .....	20
Testing Plan .....	20
Tests.....	20
How to run .....	21
Conclusion .....	23
Appendix.....	24



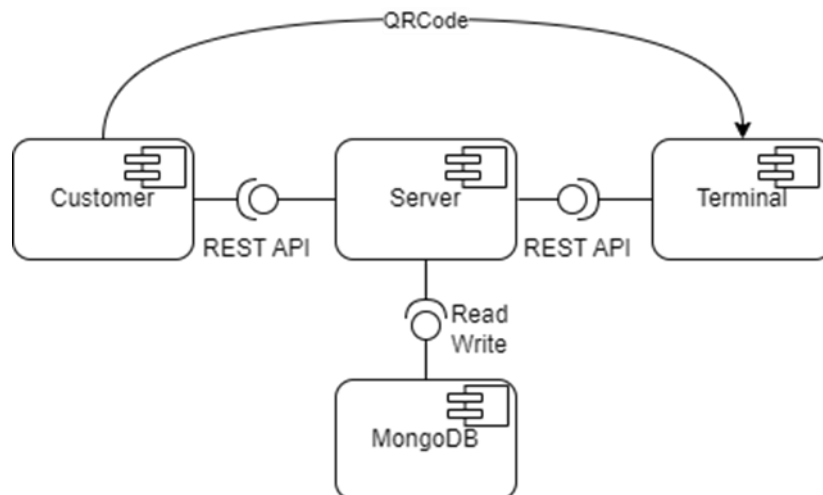
## **Introduction**

Mobile applications have revolutionized the way businesses interact with their customers, particularly in the hospitality and food service industries. For a business-like Acme Café, the implementation of a mobile application can have a profound impact on operations, customer experience, and ultimately, on the business's success.

Acme Café aims to enhance the customer experience through the introduction of an efficient ordering and delivery system, employing a mobile application designed for Android users. This system leverages a loyalty campaign to incentivize regular use by patrons. Customers can compose their orders prior to arrival, select from a comprehensive menu, and transmit their order details, including any vouchers and personal identification data, directly to an in-house terminal. The streamlined process requires customers to simply collect their items at a designated counter upon readiness, thus significantly reducing wait times and improving service delivery.

## Architecture

The system's backbone is the Node.js server application, developed as a REST API with a MongoDB database, which acts as the central processing unit for managing user data, transaction records, and voucher validations. This server interacts with two Kotlin applications: the customer-facing mobile app and the in-house terminal application. The mobile app, equipped with a user-friendly interface, allows customers to interact with the menu, manage their basket, and handle transactions. The terminal application serves as the endpoint for order processing, communicating between the mobile app and the server to finalize transactions and facilitate the pickup process.



*Figure 1 - Components diagram of the ACME System*

## **Domain Model**

The domain model for the Acme Café application defines the structure and relationships of the database entities integral to the system. These entities include Users, Orders, Products, Categories, and Vouchers, each playing a crucial role in the functionality of the application.

### **User**

This represents the customers of Acme Café. Each user has a unique identifier (UUID), a name, a unique email, a password, a unique tax number (NIF), a public key for secure transactions, a counter for the total accumulated payment, and a counter for the accumulated number of coffees purchased (free coffees do not contribute to this accumulation).

### **Order**

An order is linked to a user and contains information such as a unique order identifier, the date of the order, the state of the order (whether it is in “pending” or “verified” state), the original (subtotal), discounted amount (promotionDiscount) and the final prices (total) after any discounts or vouchers are applied.

### **Product**

This represents items available for purchase at Acme Café. Each product has a unique product identifier, a name, a price, and an image.

### **Category**

A category, which is associated with a product, it contains a name, the size this category (how many products are inside this category), and an image.

## **Voucher**

The system includes an inheritance hierarchy for vouchers, with a generic Voucher interface or abstract class that is extended by specific types of vouchers such as Discount Vouchers and Coffee Vouchers. These vouchers are linked to users and can be applied to orders.

### **Discount Voucher**

A specific type of voucher that offers a 5% discount on an order. This voucher is automatically given to a user once a new order is purchased, and the accumulated payment of that user reaches 100 Eur.

### **Coffee Voucher**

A specific type of voucher that offers a free coffee on an order. This voucher is automatically given to a user once a new order is purchased, and the accumulated number of coffees purchased by that user reaches 3.

Additionally, as shown in Figure 2, the model has provisions for scalability in terms of voucher types, the system is designed to easily accommodate new types of vouchers as they are introduced. The use of an interface or abstract class for vouchers allows for polymorphism, where the system can treat all vouchers in a uniform manner despite their specific types or classes.

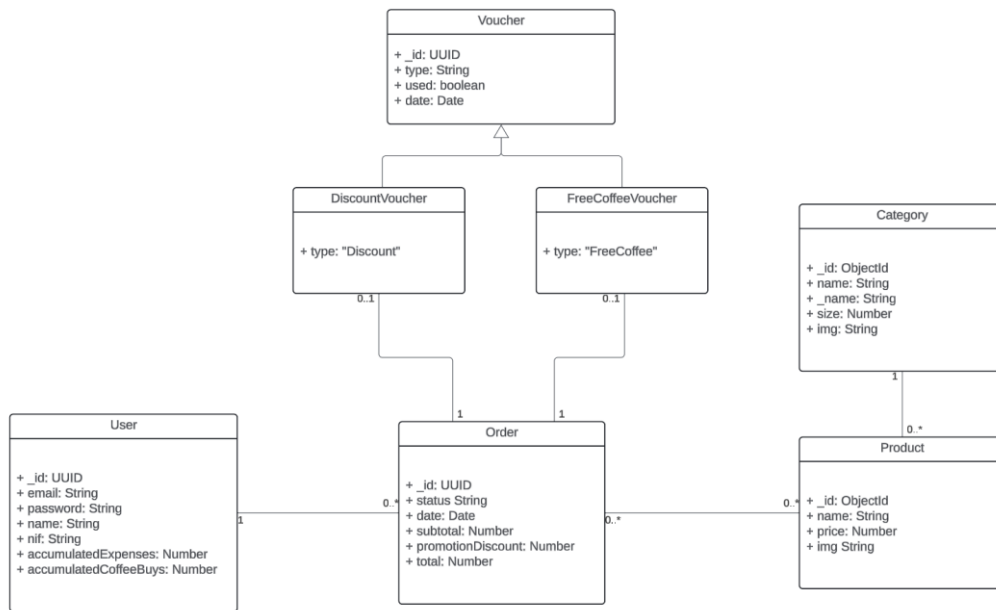


Figure 2 – ACME System's Domain Model



## Activity Diagram

The activity diagram for the Acme System outlines the user flow within the ordering system. Upon interaction with the app, the end user is prompted to either log in if they have an account or to register for a new account. Once logged in, the user can access their profile, orders and manage their vouchers, distinguishing between used and unused vouchers. The main user journey for placing an order begins with viewing the menu, followed by adding selected items to the cart. The user is then asked if they wish to continue shopping. If not, they can proceed to apply any available vouchers to their order. After voucher application, the order is validated. Subsequently, the user can make a payment, after which a receipt is generated. Simultaneously, the order details can be listed at the terminal, likely for the café staff to process the order. This flow ensures a seamless process from the user's perspective, from order creation to payment and receipt generation, emphasizing the ease of use and efficiency of the Acme System.

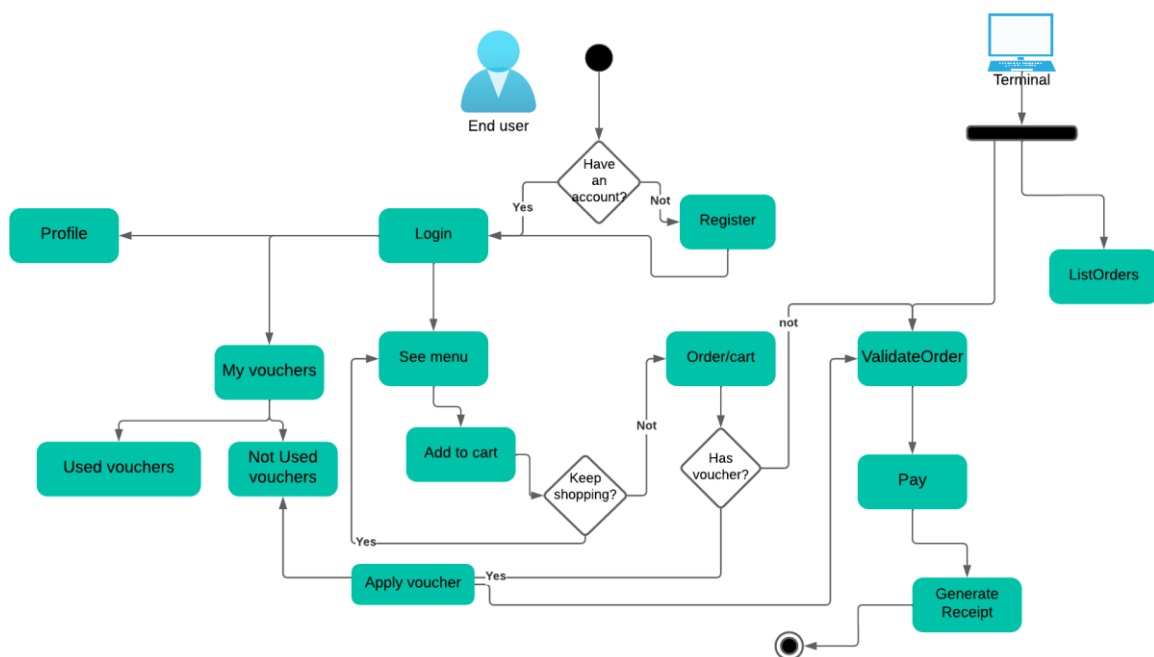


Figure 3 - ACS Activity Diagram

## System Features

The Acme Café system is an integration of three main applications, each serving distinct functions to provide a cohesive user experience and streamline café operations.

### Acme Café Server Application (Node.js Server)

The Node.js server functions as the backend support that keeps the entire system integrated and functioning efficiently.

### **Central Database Management**

Maintains and manages user accounts, order details, product information, and voucher data.

### **API Endpoints**

Offers RESTful services for mobile and terminal apps to access and update server-side information.

### **Security Protocols**

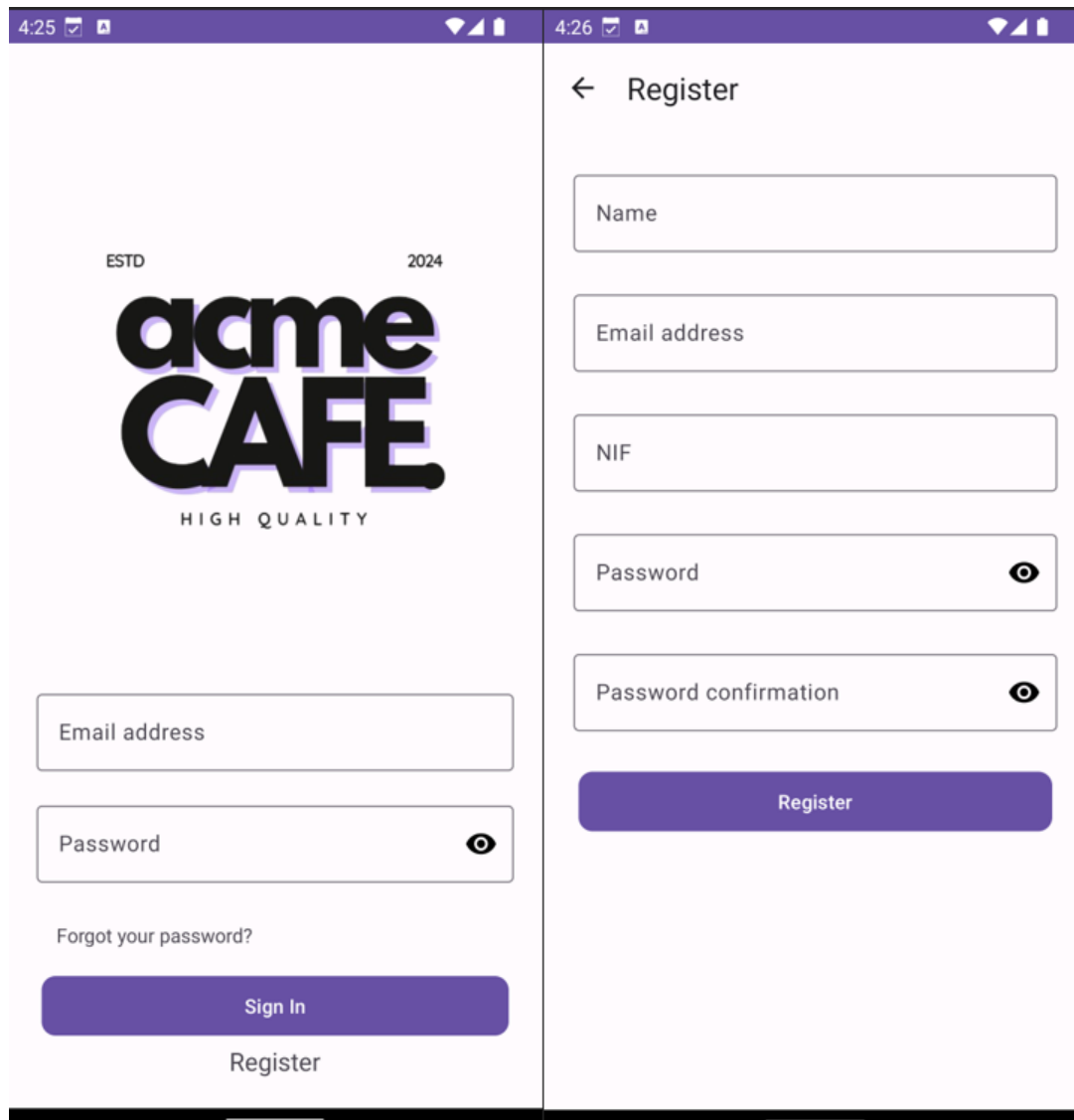
Ensures all data communication between the client apps and the server is encrypted and secure.

## **Acme Café Mobile Application (Customer App)**

The Acme Café mobile app streamlines the customer experience with pre-ordering from the menu, secure user account management for registration and personal data, and basket functionality to manage selections pre-checkout. It accommodates the application of various vouchers for discounts, ensures secure payment processing, offers access to order history for easy reorders, provides real-time updates on order status, and delivers push notifications for order confirmations, pickup readiness, and promotions.

### **Login and Registration**

Secure customer registration and login process, storing critical information such as cryptographic keys, user details, and payment information. In the Figure 4 are depicts login and register activities from the Acme Café Android app. The left screen is the login interface, featuring the café's logo and slogan, fields for email and password entry, a 'Forgot your password?' link, and a 'Sign in' button. The right screen is for new user registration, with fields for name, email, NIF (tax identification number), password, password confirmation, and a 'Register' button. Both screens have a consistent color scheme and design, indicating a user-friendly interface for both existing and new customers of Acme Café.



*Figure 4 – ACME App Login and Registration Activities*

Below, Figure 5 displays the home menu, which features a selection of beverage categories. Each category is represented by a visually appealing image of a coffee drink and a label indicating the type of coffee, such as hot or cold, and specific styles like cappuccino and frappuccino. The number of options available within each category is also shown. The app's user interface is highlighted by a clean and colorful design, with navigation icons at the bottom.

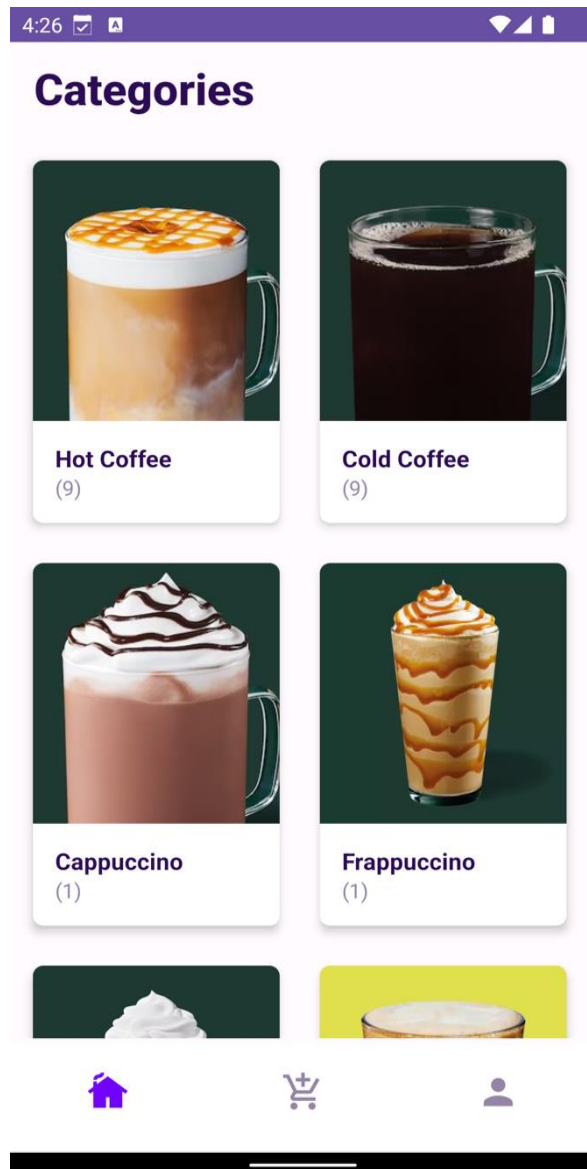


Figure 5 - Customer Application Home Menu

## Vouchers

The voucher management as shown in Figure 6, two screens from the customer app voucher section. Each screen displays different types of vouchers: one offers a 5% discount, and the other two vouchers offer a free coffee each. Each voucher is accompanied by a unique code. There are buttons for applying the vouchers, and the design maintains a consistent color theme and layout, including navigation icons for further user interaction.

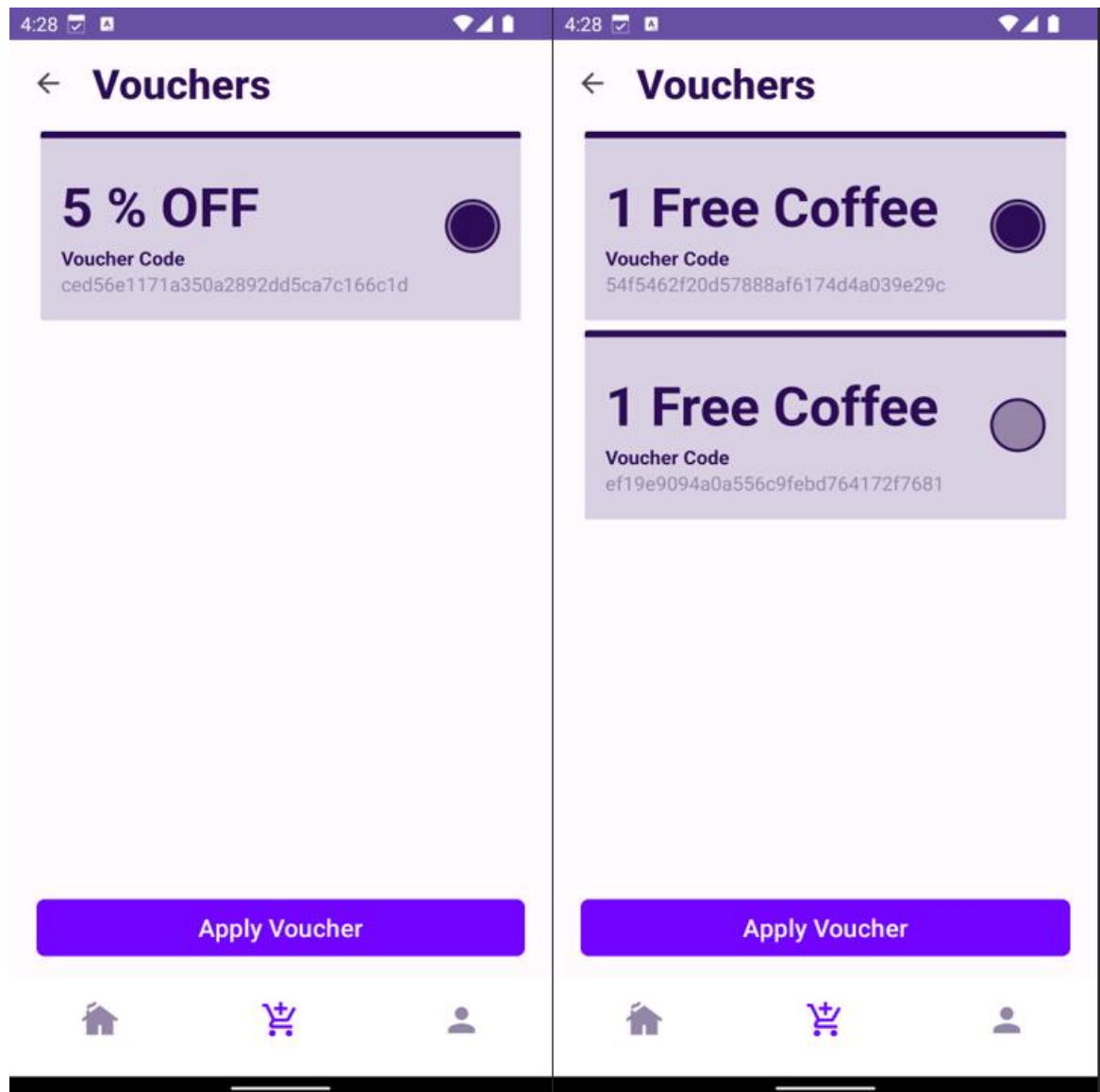


Figure 6 - Customer Application Voucher Management

## Checkout

Figure 7 shows a checkout interface from the Acme Café mobile app. Dominating the upper half of the screen is a large QR code, encoding the details of the customer's order for scanning at the terminal app. Below the QR code, the 'Order Details' section lists order's details such as price, quantity and total. This screen is typically presented by customers during the final stage of placing their order, where the QR code would be scanned by terminal app to finalize the transaction.

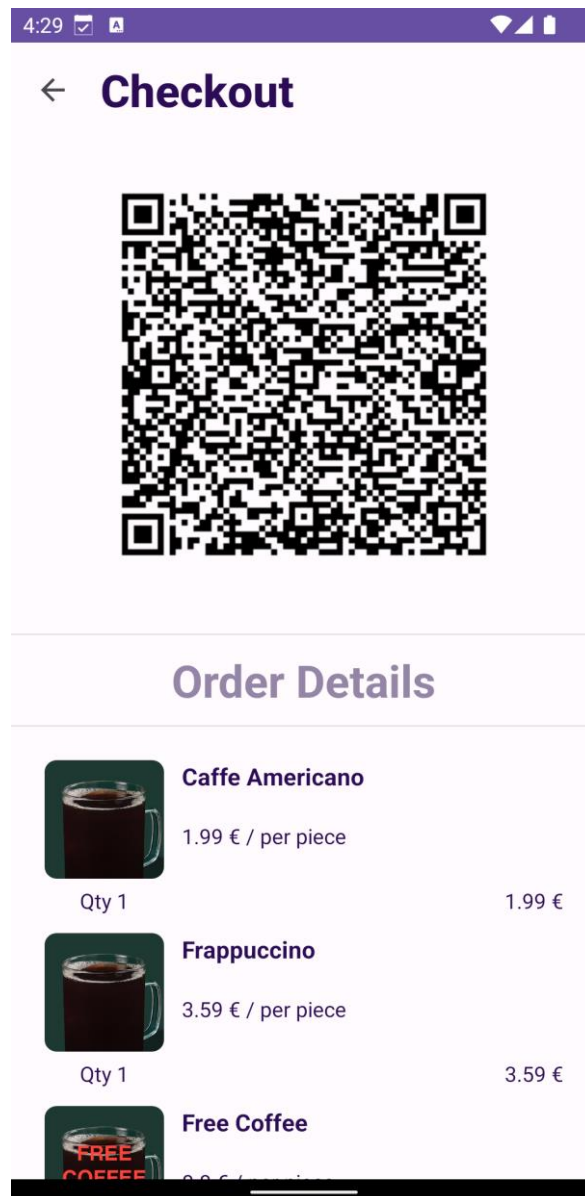


Figure 7 - Customer Application Checkout Page

## Receipts

The image presents two screens from the customer app concerning receipts. On the left screen, there is a detailed individual receipt that includes client details, a summary of the items purchased, such as Caffè Americano and a complimentary coffee, along with vouchers applied to the order, and a financial summary highlighting the subtotal. The right screen lists multiple receipts, each with a brief overview including the date and the number of products purchased. Each entry on the list is associated with a different total amount. The interface utilizes icons and a clean layout for easy navigation and readability.

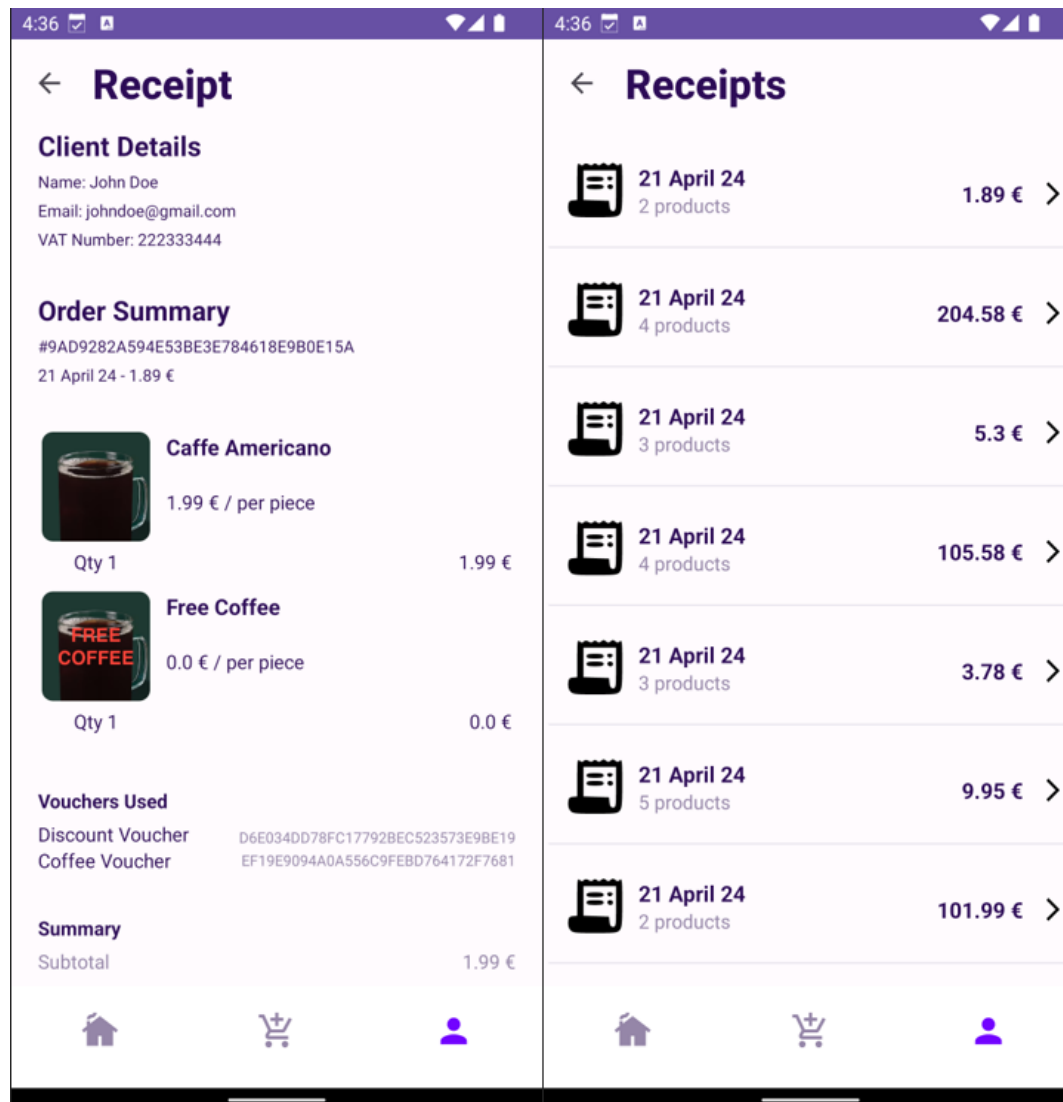


Figure 8 – ACME App Receipts management

## Acme Café Terminal Application (Terminal App)

The terminal app interfaces with the Node.js server to confirm payments, generate digital receipts post-payment, and provides a management interface for café staff to oversee orders. It



also verifies and redeems any vouchers used during checkout, ensuring a streamlined process from order placement to pick up.

## Order's Management

Figure 9 shows a list of orders from the terminal app. Each entry features a date, the number of products ordered, and the total price. The list is uniform, with each order entry depicted by an icon, representing an order summary. A navigation home button is at the bottom of the screen, indicating the ability to return to the main menu of the app. The layout is straightforward, making it easy for users to view their past orders and the associated costs.

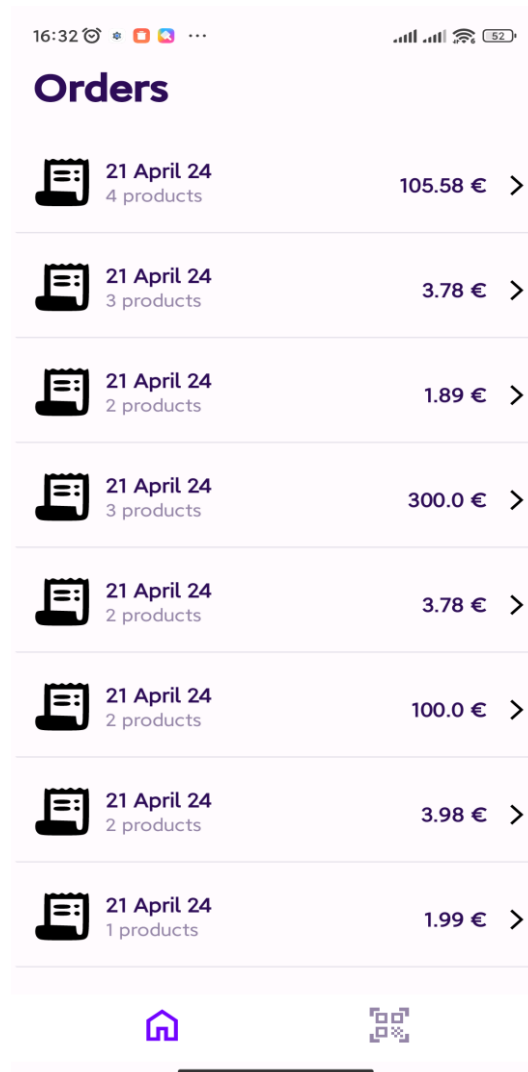


Figure 9 - Terminal Application Order Management



## QRCode Reader

The image shows a feature within the terminal app that serves as a QR code reader. The screen is primarily dedicated to a central graphic indicating that users can tap to scan a QR code, symbolized by a stylized QR code and a magnifying glass icon. This function is likely part of a broader system within the app, related to payments, vouchers, or information retrieval. The app maintains a consistent visual theme with purple tones and navigation icons.



*Figure 10 - Terminal Application Barcode Scanner*

## Navigation Map

## Testing

### Test Summary

The test suite for the ACME Coffee application demonstrates excellent results with 9 tests executed, no failures or ignored tests, and a 100% success rate. This indicates that the tested components of the application are functioning as intended under the current testing conditions.

### Testing Plan

The testing strategy adopted a focused, unit-testing approach with the following key characteristics:

- Isolation:** Tests were designed to isolate specific units of code, minimizing dependencies and enabling precise targeting of functionality.
- Mocking:** Mock objects were used extensively to simulate the behavior of external dependencies. This allowed for controlled environments and a concentrated focus on the code under test.
- Assertions:** Assertions were used to verify expected behaviors and outcomes, ensuring that the code produced correct results.
- Test Methods:** Tests were organized into well-defined test methods, each addressing a distinct aspect of the codebase.
- Test Naming:** Clear and descriptive test method names aided in understanding the intention of each test.
- Test Execution:** The test suite was designed for automated execution, enabling frequent testing and rapid identification of regressions.

### Tests

One JUnit test was created for each method within the following classes, each test class has the same name of the target class, but with “*Test*” at the end of the name:

### Test Summary

9	0	0	1.328s	100%
tests	failures	ignored	duration	successful

Packages

Classes

Class	Tests	Failures	Ignored	Duration	Success rate
<a href="#">com.feup.coffee_order.application.core.crypto.CryptoKeysTest</a>	3	0	0	0.935s	100%
<a href="#">com.feup.coffee_order.application.domain.repository.OrderRepositoryTest</a>	2	0	0	0.326s	100%
<a href="#">com.feup.coffee_order.application.domain.repository.ProductRepositoryTest</a>	2	0	0	0.062s	100%
<a href="#">com.feup.coffee_order.application.domain.repository.UserRepositoryTest</a>	2	0	0	0.005s	100%

## How to run

To prepare a machine for development there are few steps to done:

### Deploy local server

1. Install Docker Desktop
2. Clone the repository on: git clone <https://github.com/FabioMiguel2000> Acme-Caf-Order-System
3. Head inside the server's project directory `/coffee-order-api`, and change the file `local.env` to `.env` with the appropriate environment variables:  
*MONGODB\_URI=mongodb://mongo:27017/acme-coffee*
4. There is a docker-compose file that builds and runs 3 containers, the node server, the MongoDB server and the Mongo-Express (admin interface):  
*docker-compose up -d;*
5. Install NodeJS dependencies:  
*docker exec -it api npm install;*
6. Populate the database with seed, run:  
*docker exec -it api npm run data:import;*
7. If performed the populated the database in step four, to destroy old structure and get new one, run the comand bellow and run again step four:  
*docker exec -it api npm run data:destroy;*

### Run application

Head inside the app's project directory `/coffee-order-application`, and add the server's base URL to the `local.properties`, accordingly:

Using a physical device:

- `apiBaseUrl=http://local_ip_address/api/`

Example: `apiBaseUrl=http://192.168.1.97:3000/api/`

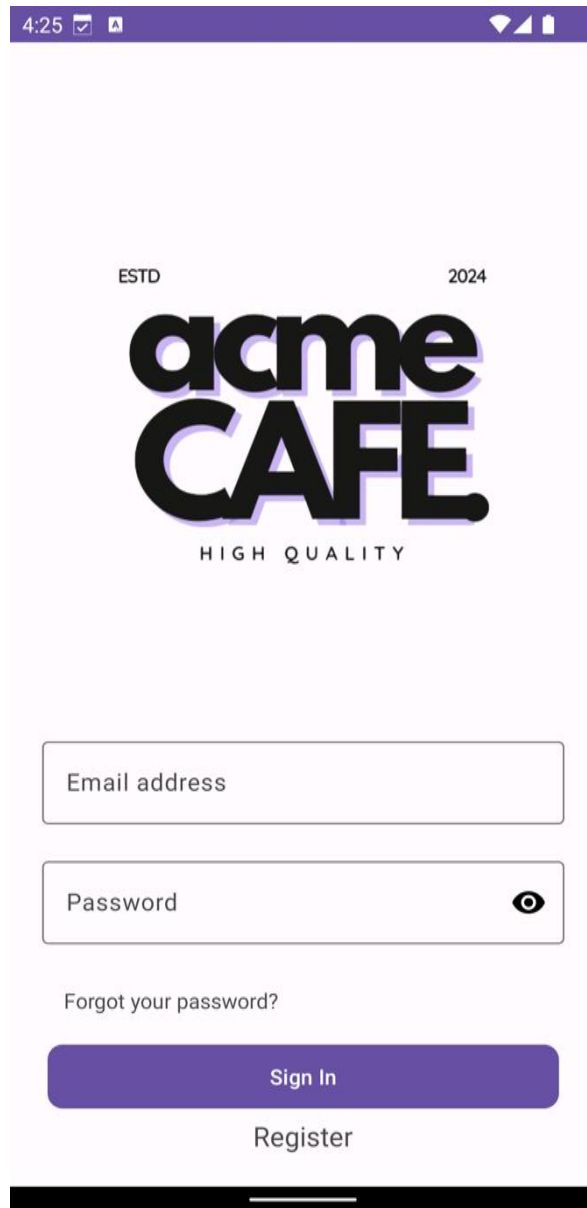
Running on an emulator, just keep with the default base URL:

- `apiBaseUrl=http://10.0.2.2:3000/api/`

## **Conclusion**

Acme Café's implementation of mobile and terminal applications signifies a commitment to enhancing customer service through technology. By streamlining the ordering and pickup process, providing a personal touch with user-specific features, and leveraging the power of a Node.js server for robust data management, the café is poised to elevate its customer experience, improve operational efficiency, and drive business growth.

## Appendix



The image shows a mobile app login screen for 'acme CAFE'. At the top, a purple status bar displays the time '4:25' and various icons. The app's logo, 'acme CAFE', is centered, with 'acme' in a lowercase sans-serif font and 'CAFE' in a larger, bold, uppercase sans-serif font. Above 'acme' is the text 'ESTD' and above 'CAFE' is '2024'. Below the logo, the words 'HIGH QUALITY' are written in a small, uppercase sans-serif font. The login form consists of two white rectangular input fields with thin black borders. The first field is labeled 'Email address' and the second is labeled 'Password'. To the right of the password field is a black eye icon for toggling visibility. Below the password field is a link that says 'Forgot your password?'. At the bottom of the form is a large, rounded purple button with the text 'Sign In' in white. Below this button is a link that says 'Register'. The entire screen has a light pink background. At the very bottom, there is a black horizontal bar with a white line in the center, representing the mobile home indicator.

4:25

ESTD 2024

**acme**  
**CAFE**

HIGH QUALITY

Email address

Password






Forgot your password?

Sign In

Register

*Figure 11 – ACME CAFE App login activity*




4:26     


← Register

Name

Email address

NIF

Password 

Password confirmation 

Register

*Figure 12 – ACME CAFE App register activity*

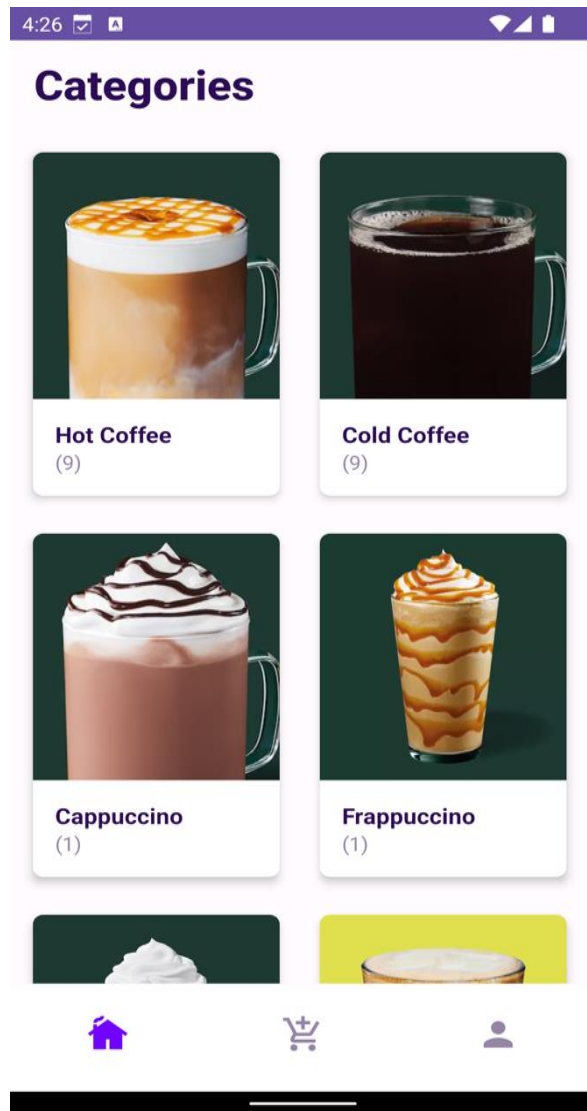


Figure 13 – ACME CAFE Customer App Home Menu Screen

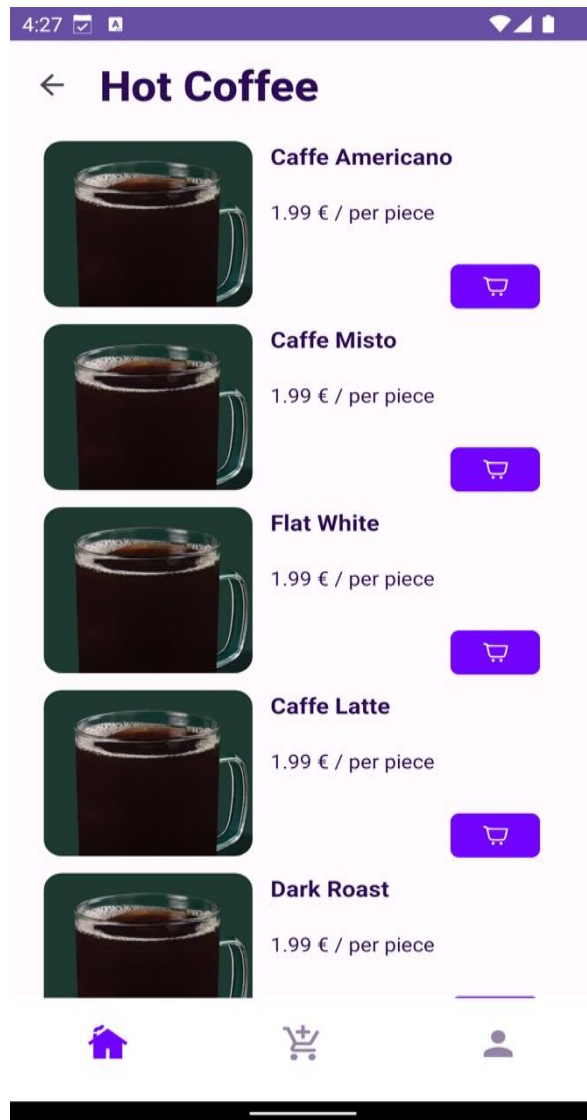
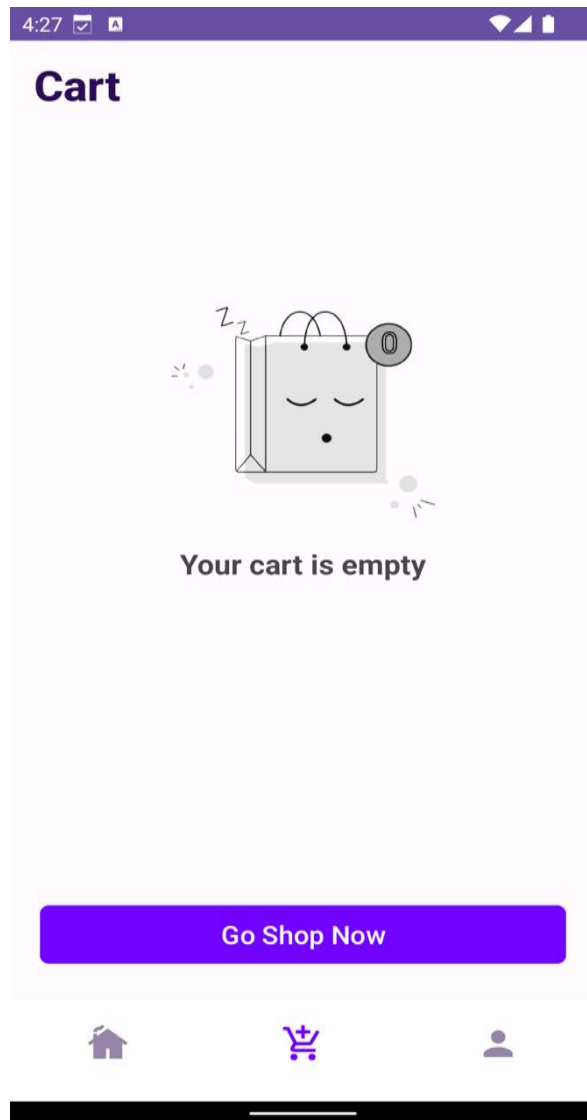


Figure 14 – ACME CAFE Customer App Listing Product in a Category



*Figure 15 – ACME CAFE Customer App Empty Cart activity*

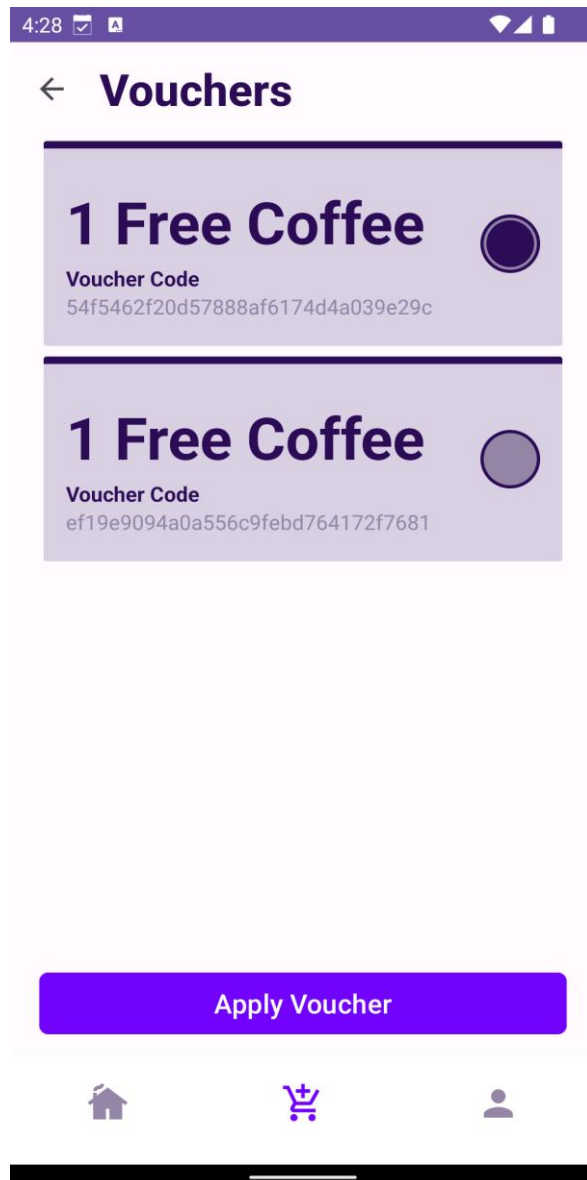


Figure 16 – ACME CAFE Customer App Voucher Screen

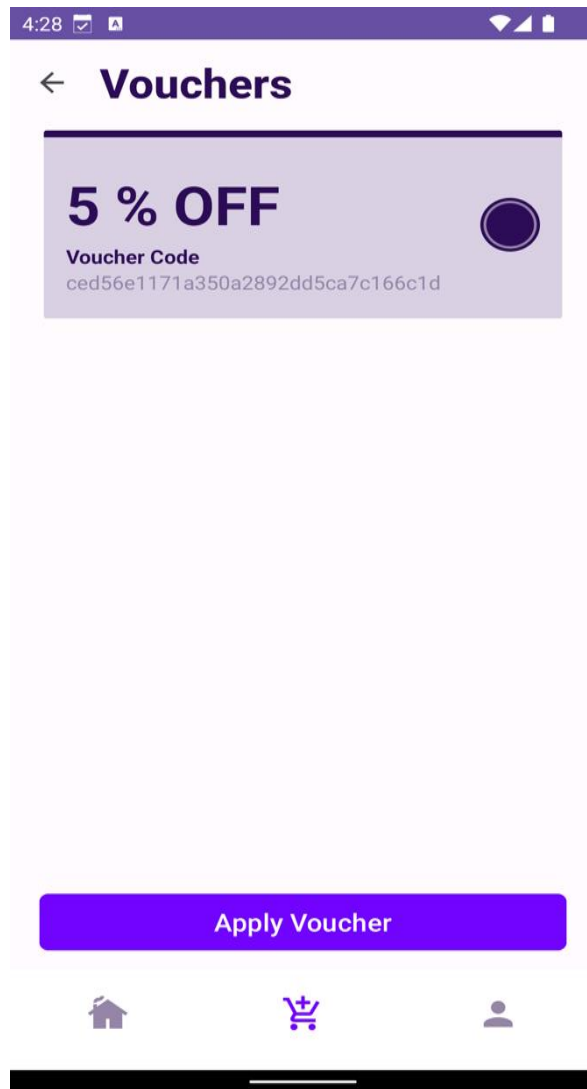


Figure 17 – ACME CAFE Customer App Listing Product in a Category

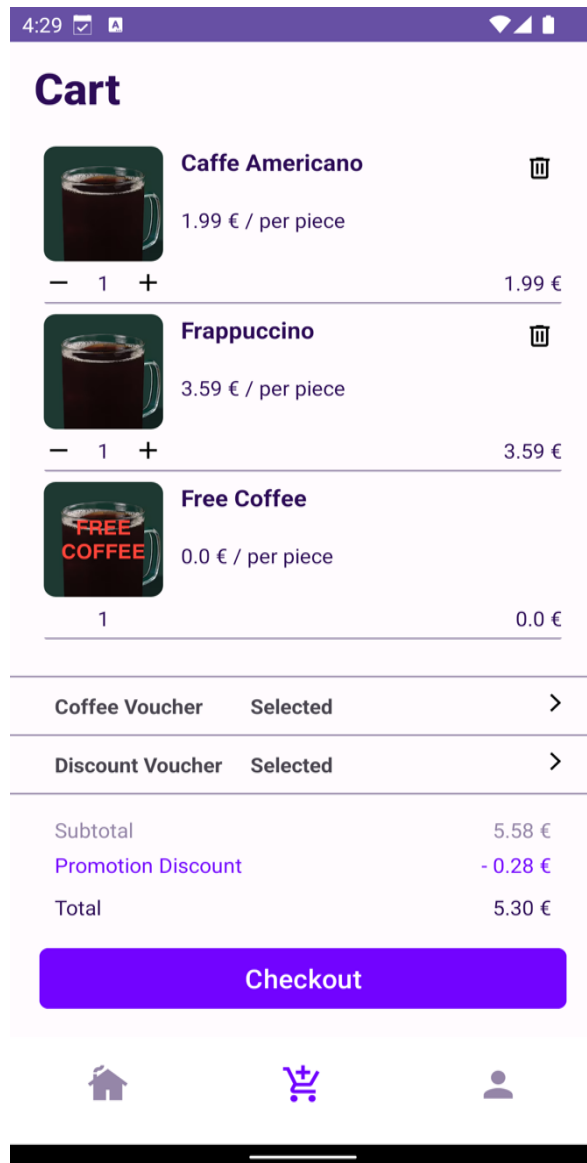


Figure 18 – ACME CAFE Customer App Cart Screen

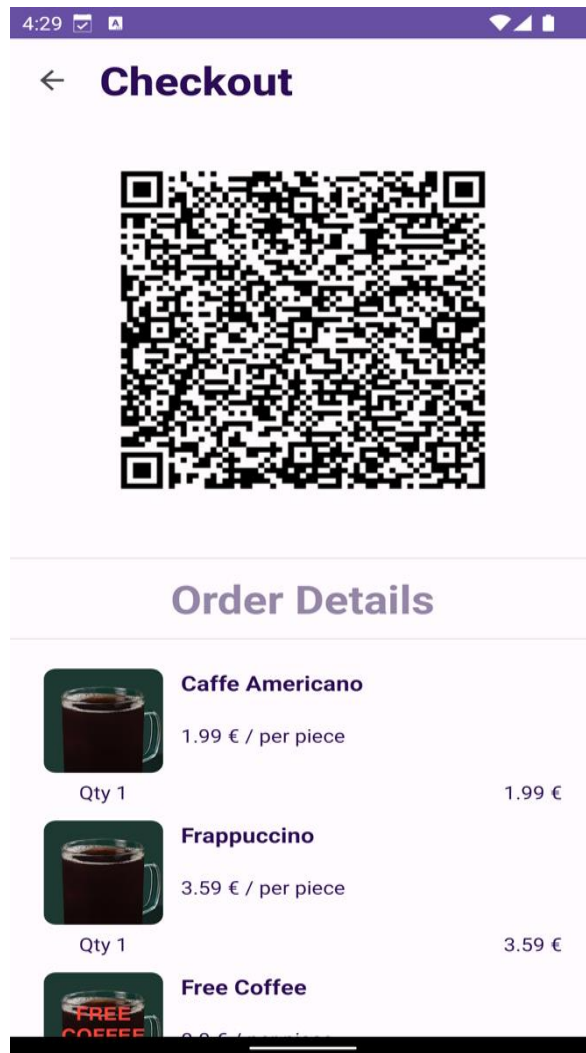


Figure 19 – ACME CAFE Customer App Checkout Activity



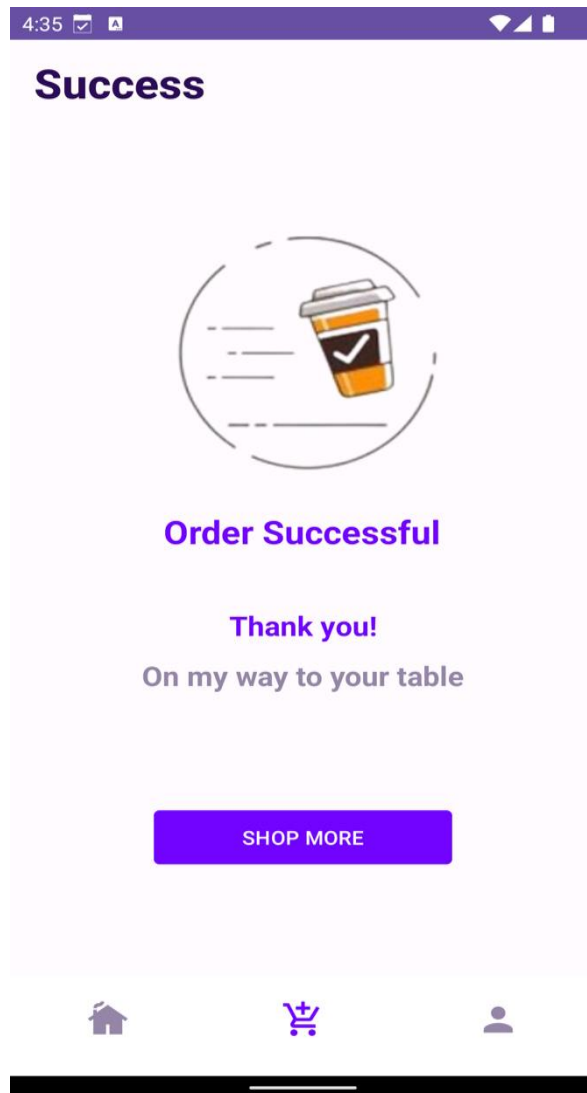


Figure 20 – ACME CAFE Customer App Order Success Screen

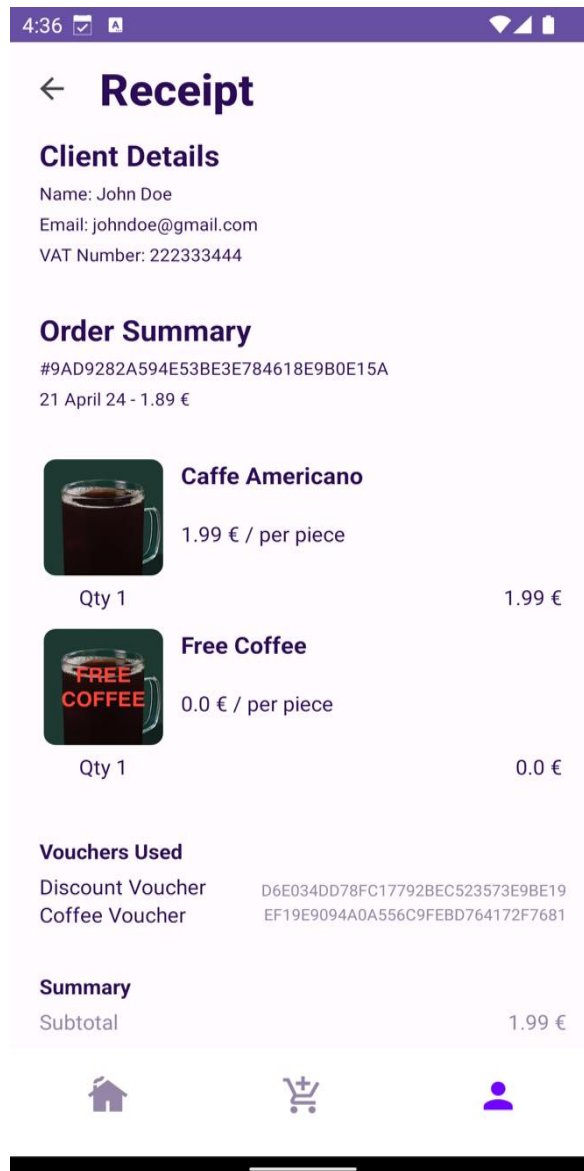


Figure 21 – ACME CAFE Customer App Single Receipt

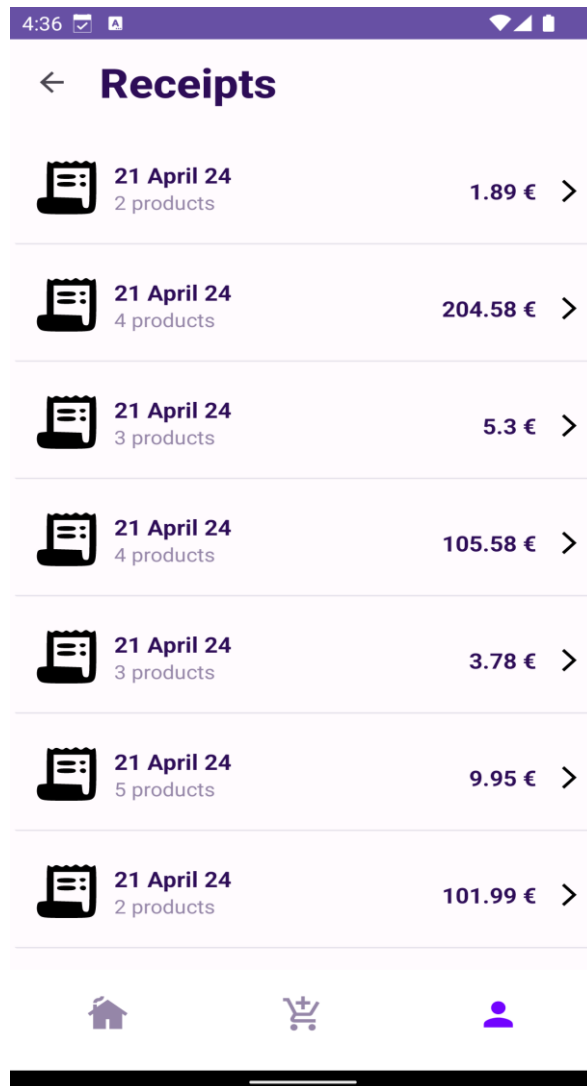


Figure 22 – ACME CAFE Customer App Receipts View

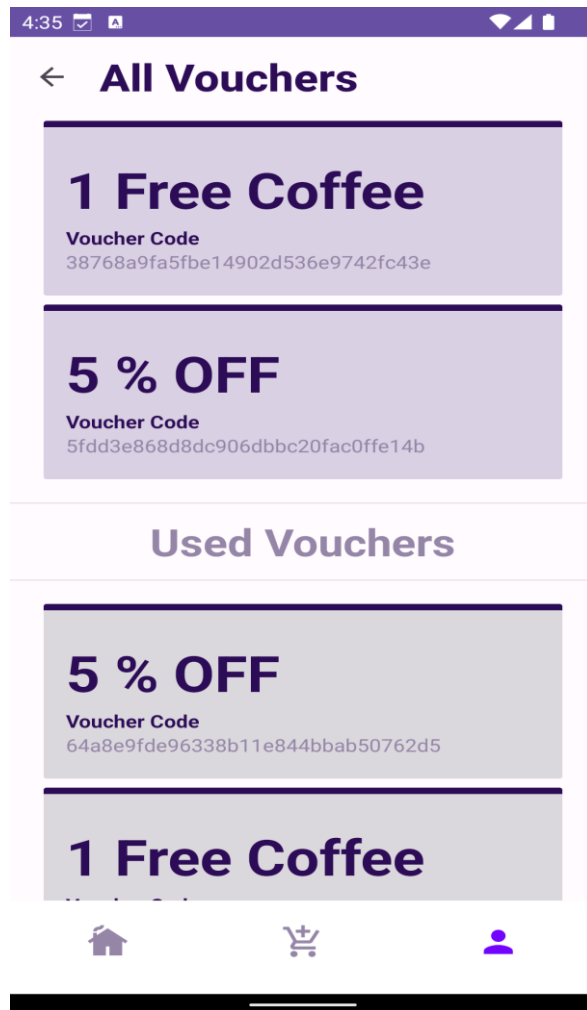


Figure 23 – ACME CAFE Customer App All Available Vouchers

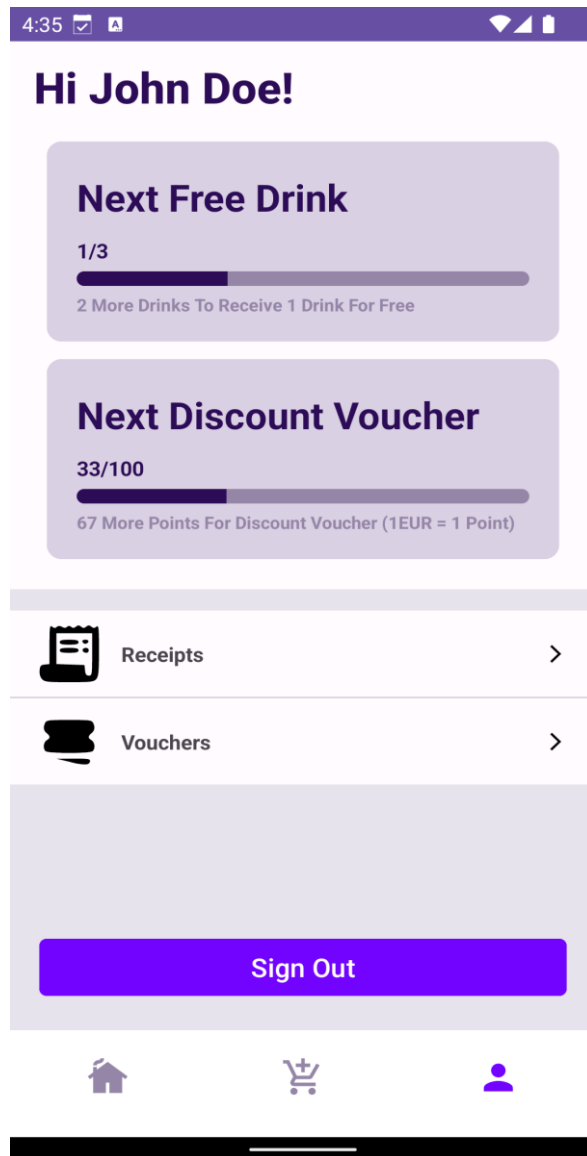
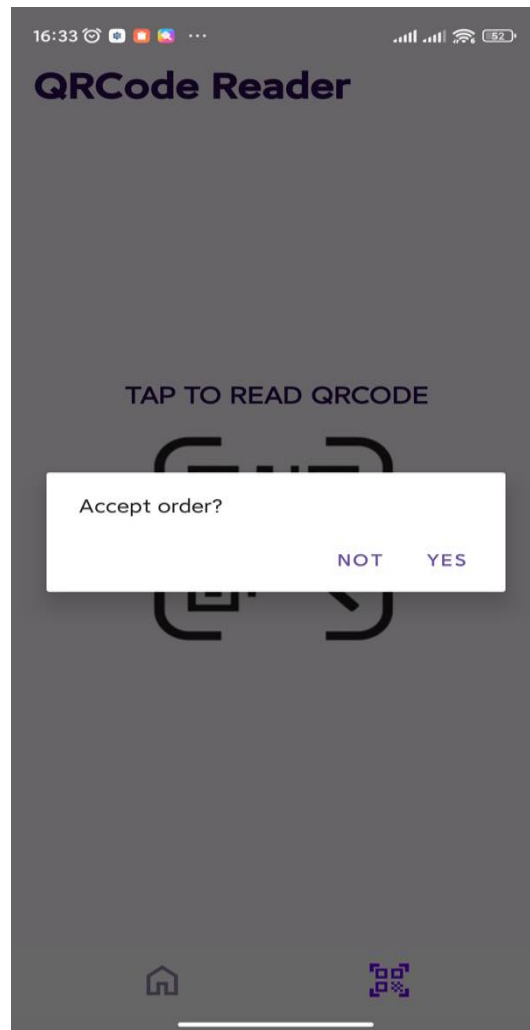


Figure 24 – ACME CAFE Customer App Profile Screen



*Figure 25 – ACME CAFE Terminal App Barcode Reader*



*Figure 26 – ACME CAFE Terminal App Accept Order*

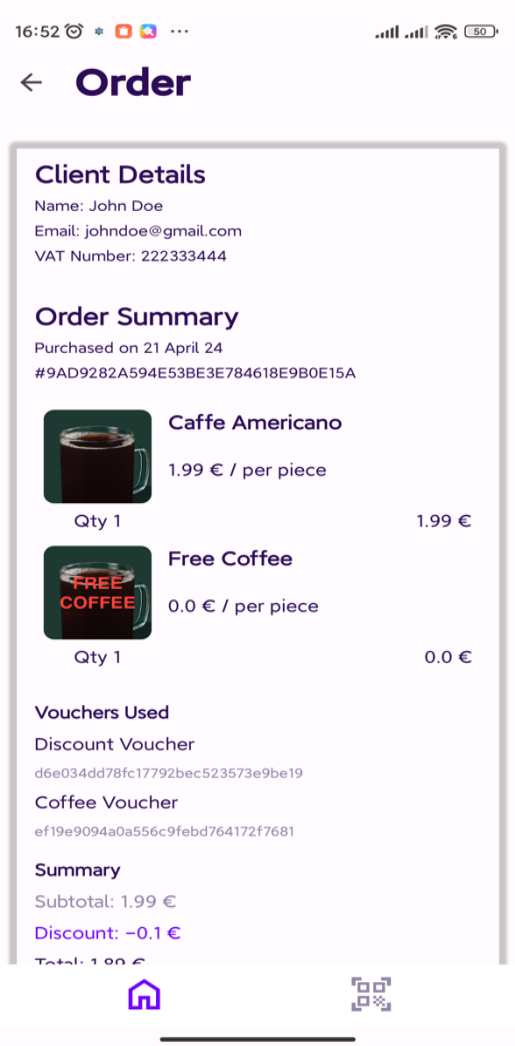
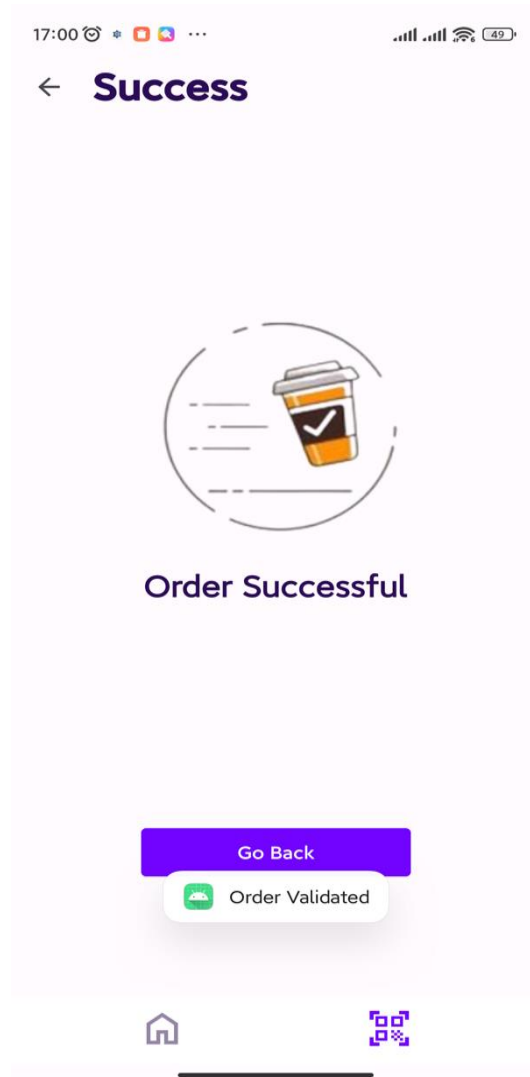


Figure 27 – ACME CAFE Terminal App Order Details







*Figure 29 – ACME CAFE Terminal App Validated Order*

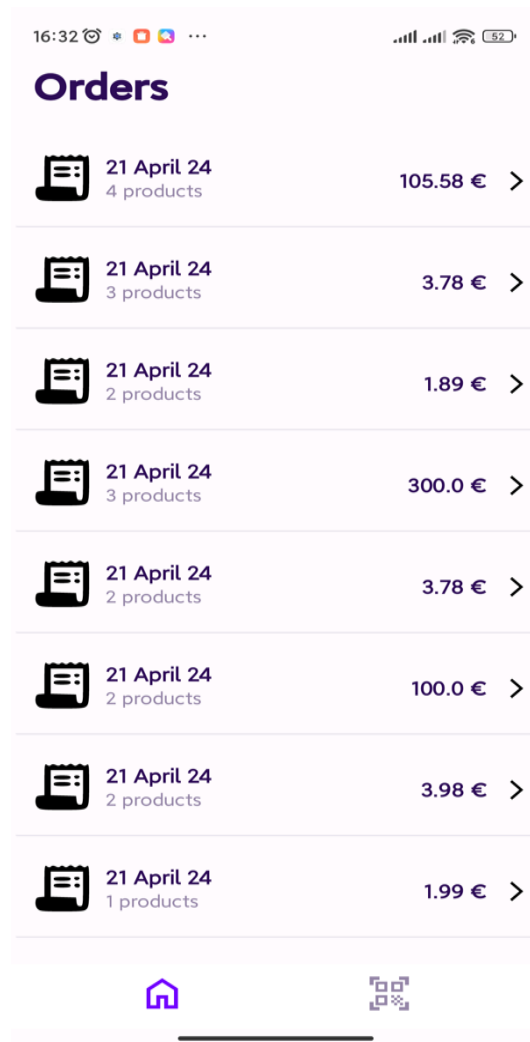


Figure 30 – ACME CAFE Terminal App All Orders