Technical Report


# Application for expenses and budget management

## ICM – Introduction to Mobile Computing

**Group 02**
91322 – Isadora Loredo
91359 – Juan Lessa
98411 – Ricardo Ferreira

Aveiro, 06 de maio de 2022

## Objective

The purpose of this Project is to implement a mobile app to practice the concepts of Flutter framework by exploring its UI widgets and libraries.

## Application Context

The app's objective is to help users to manage its financial expenses by setting a budget and registering expenses, with the possibility to see reports according to the type of expense or time frame.
The user will be able to establish priorities for how he/she spends money to achieve financial goals, therefore registering how much money is spent and on what.

## Requirements

For the solution this application aimed to achieve, it was proposed the implementation of the following features has:

- Centrally record user's expenses
- Improve financial awareness
- Provide reports on the expenses by using Graphics
- Provide integration with google maps to locate nearby ATMs by using GPS
- Allow user to register their expenses through QR code
- Safety with financial data by using fingerprint reading to access it and shake movement sensor to lock it

## Architecture of implementation

Persistency: sqflite

Services:

- gps
- google_maps
- geolocator
- http

Libraries:

- fl_chart
- qr_code_scanner
- shake
- local_auth

# Milestones

## Milestone 1 – Draft storyboard/UI of the application on the device

### Application Context

This app's objective is to help users to manage its financial expenses by setting a budget and registering expenses. It will be possible to see reports according to the type of expense or time frame. The user will be able to establish priorities for how he/she spends money to achieve financial goals, therefore registering how much money is spent and on what.

### Application Purpose

1. Centrally record user expenses
2. Improve financial awareness in users
3. Provide reports on the expenses
4. Provide integration with maps to locate nearby ATMs.
5. Allow users to register their expenses through QR code reader or NFC sensor

### Sensors & mobile functionalities

Camera (QR code) - Will be used to read QR codes from invoices to create expenses records.
GPS - Will be used to find ATMs nearby.
Map - Will display the user's current location and point ATMs.
Fingerprint - Will enable users to unlock the application using its fingerprint.
Graphics - Will allow the user to analyze the data regarding its expenses.
NFC - Will be used to detect expenses.
Shake Event - Will enable user to lock the app

## Milestone 2 – 2 mandatory elements running on the device

1. Fingerprint sensor implementation to unlock the application
2. Persistency and query implementation with SQLite (sqflite)

## Milestone 3 – Overall solution and suitability to the original problem

This application aimed to tackle the problem of reporting expenses in real time and help users to track its financial planning and management.

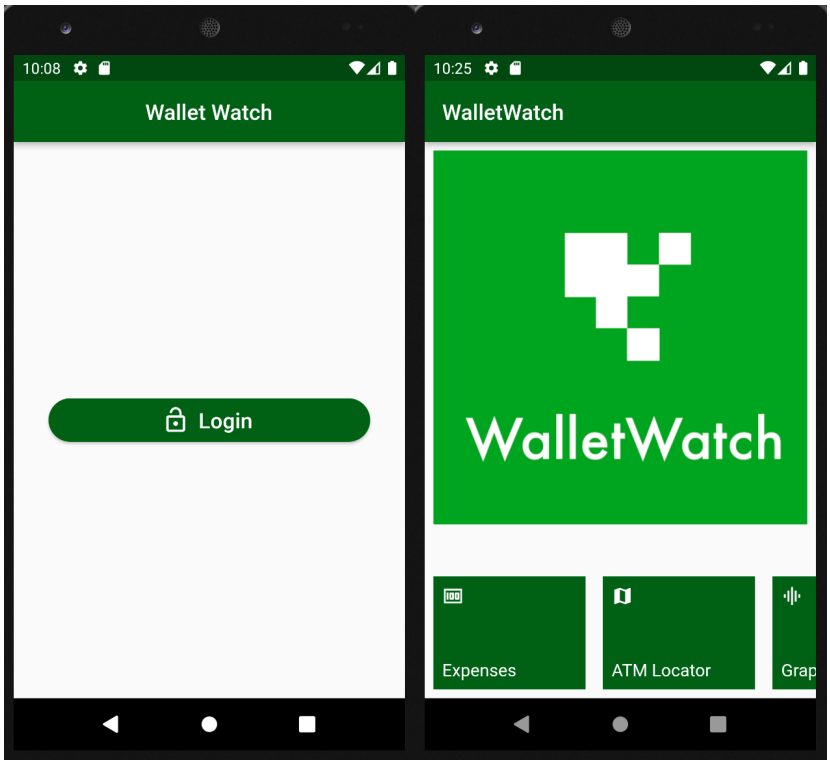
To that end, this solution allows users to record expenses by filling up a form or reading from a QR code scan. There is also the feature of charts drawing from the expenses recorded that provides users awareness on the type of expenses. Finally, users can find nearby ATMs by its current location, and have it on a map and a list with the distance, address and status.

## Demo

Note: is better run using *Adroid Studio Emulator* to test the ATMs Locator feature, as it was noted that on the tested device the call to get the current location of the device eventually took longer than expected.

For the objective of this project, it was mandatory the implementation of 4 elements, the use of services and sensors.
Follow the working implemented elements:

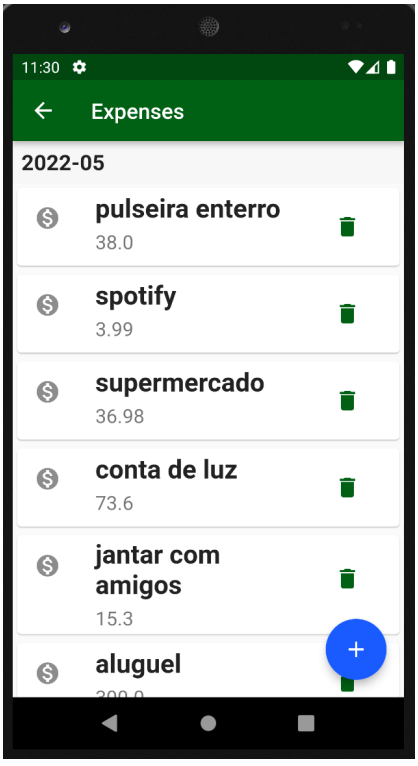
1. The application secures its data using the **fingerprint reading** to unlock it



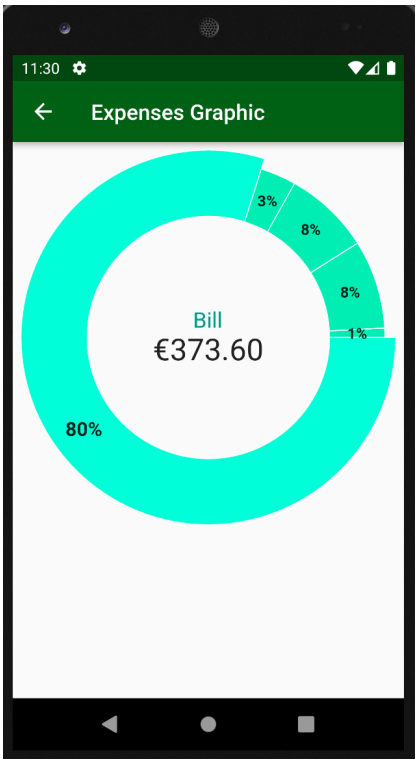

2. The **movement sensor** is used to allow locking the application shaking the mobile

3. For recording expenses easily, it allows reading a **QR code** with the data of an expense

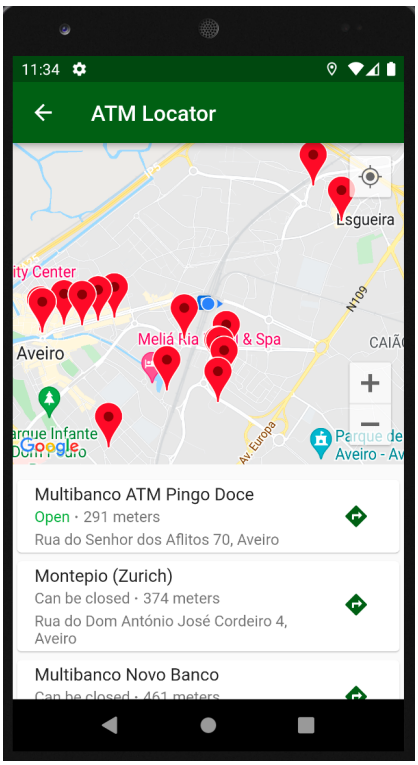

4. The expenses are recorded in a **local database** using SQLite




5. With the expenses recorded in the database the application draws a chart using **graphics**




6. The application gets its current location using the **GPS**, through **http** calls makes a nearby search to find the closest ATMs shown in a **Google Maps** widget, and can open on Google Maps app any of the locations from the list



## Contribution Assessment

Isadora – 50%

Juan – 25%

- Graphics implementation

Ricardo – 25%

- Fingerprint reading sensor