Personal Finance

Lorenzo De Luca

lorenzo.deluca4@stud.unifi.it

Università degli Studi di Firenze

February 19, 2021

Outline

Introduction

Today we can use many **payment methods**: cash, credit cards, smartphones and watches

- ✓ More convenient to pay
- ✗ Difficult personal finance management

Personal Finance: planning processes for your financial activities such as generating income, saving or budget monitoring.

Motivation

We need a way to create a **financial plan**

- to have a **clear** overview of monetary inflows and outflows
- to **understand** if we manage money effectively or if improvement is needed

Idea

Create a **mobile app** with the following features:

- Manage multiple checking accounts separately
- Add and view incoming or outgoing transactions.
- Categorize income/expenses
- Define a monthly budget to regulate the flow of expenses.
- Display transaction summary charts

Specifically, build an app that remembers **native iOS applications** to implement clarity and simplicity

Project steps

The creation of the mobile app was based on three steps:

1. **Needfinding**: to identify the needs of users regarding the topic
2. **Code Development**: to implement the idea and characteristics found in the needfinding
3. **Usability Test**: to check if the requirements are met or if there are some problems

Needfinding

The *needfinding* process was carried out with a **survey** on Google Forms to various users, of different age groups and social roles.

The survey included **9 questions** to understand the use and habits of finance management, and thanks to the answers obtained it was possible to develop **Personas**

Personas

- People between 20 and 25 years old, they study and do not receive a salary. Their money comes from personal savings. Their expenses are mainly directed towards food, technology and hobbies. They limit spending to save for future purchases.

- People over 25, they work and have fixed income every month. They have various expenses: from health to home, but few finances are devoted to hobbies. Their interest is based on a generic control of expenses.

Use cases

From the *personas* obtained, it was possible to understand the needs of users and the functions that an app should have.
The functionalities obtained and introduced in the app must solve **hypothetical scenarios**. Here is one of the many examples:

- Marco, a university student, received his pocket money from his grandmother, which he immediately puts in the piggy bank. It is difficult for him to accumulate money quickly to go on a trip with his friends. He has to limit his expenses and to break the piggy bank just before leaving.

Code Development

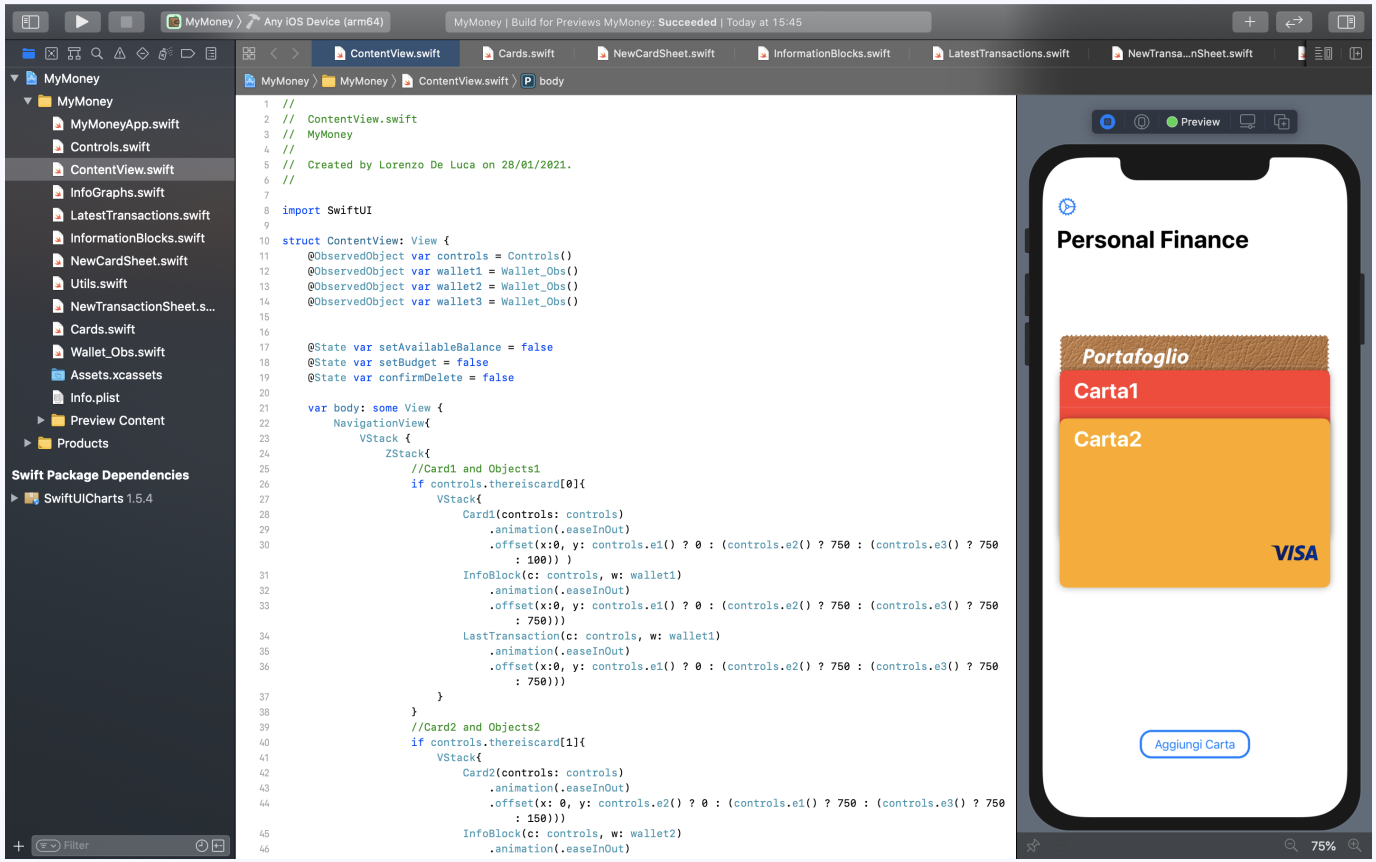
The app was developed **for Apple devices**, respecting Apple's Material Design rules; using the following software:

Xcode is Apple's IDE for macOS, used to develop software for macOS, iOS, iPadOS, watchOS.

SwiftUI a framework to build user interfaces across all Apple platforms with *swift*. It uses a declarative *swift* syntax that's easy to read and natural to write.

about *SwiftUI*

SwiftUI allows to keep the code and design perfectly in sync.



Other useful elements...

Simboli SF was designed by apple to integrate seamlessly with the San Francisco system font; you can use the symbols in a variety of UI elements (such as navigation bars, toolbars, tab bars, ...)

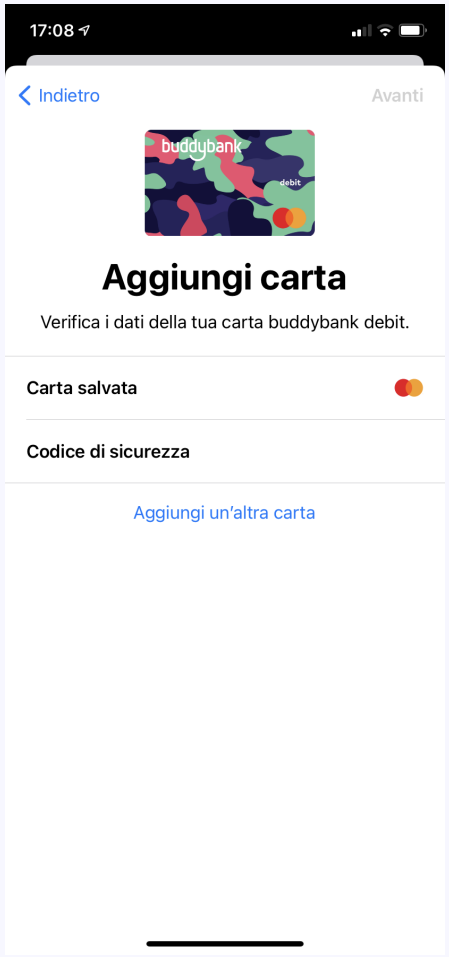
Image2Icon allows you to customize icons. Generate the app icon for the different views (application image, system notifications, ...)

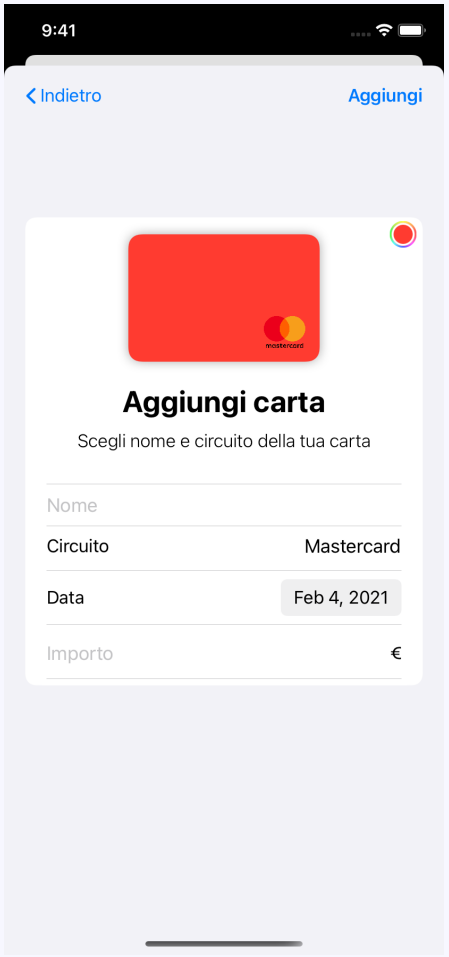
App interface: *main page*

The interface for **managing the bank accounts** was designed taking inspiration from native iOS apps, such as "Wallet"

App interface: *add new bank account*

The interface for **add new bank account** was designed taking inspiration from native iOS apps, such as "Wallet"



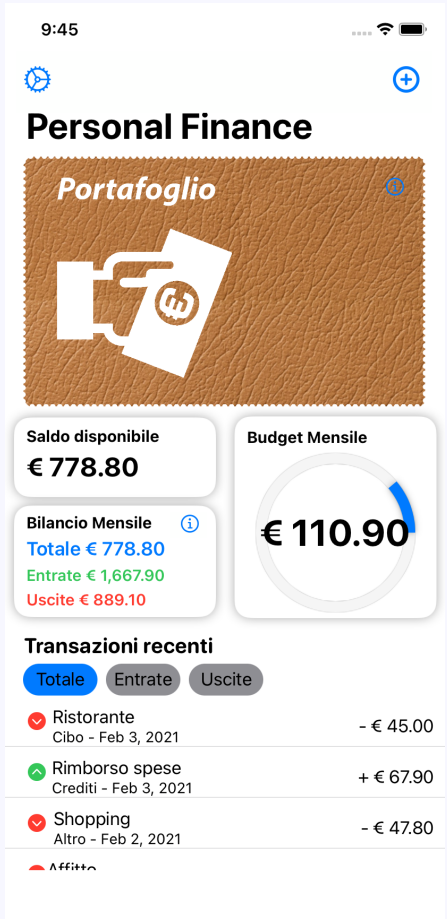


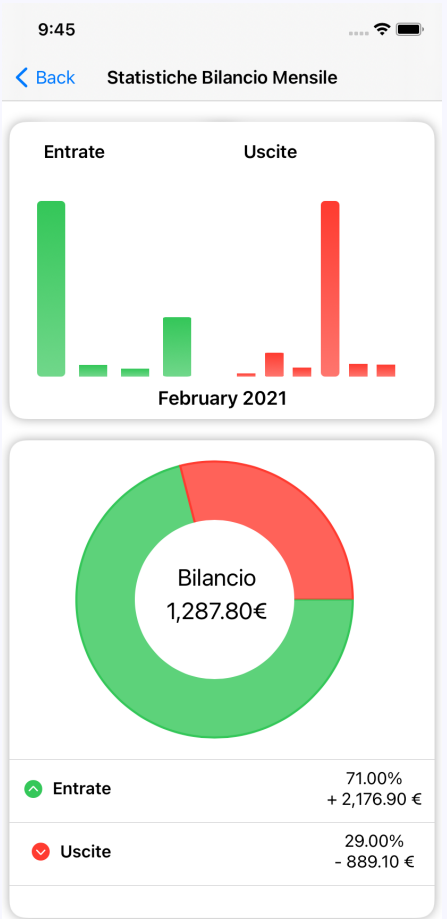
App interface: *view bank account*

The interface for **viewing the details** is reached with a tap on the card, and you can see:

- the available balance
- the income
- the expenses
- the list of transactions
- the budget

Also you can see the **summary graphs**





Usability Test

The usability test was divided into two steps:

1. Test to find and fix any **bugs** in the app, before going to step 2.
2. **Large-scale Test** to understand if the design phase has met the pre-established requirements and if it covers the use cases.

Next, the testers have been asked to answer a short **survey** about the app

Usability Test

After providing the testers with a **guide** that explains the basic functionality of the app, each user:

1. install the application on your device, when possible
2. use the app **freely** for a few days
3. answers the 13-question survey by **evaluating** the app's attractiveness and functionality from 1 to 7 (using Single Ease Question) and answers 2 open questions.

Usability test Results

The results show that:

- The **app** was appreciated and was intuitive,
- The **views** are compatible with Apple's design canons, but have shortcomings, such as haptic feedback.
- The **fluidity of the animations** and the **positioning of the icons**, received substantially positive comments.
- For the **application features**:
  - very positive evaluation for the method and style of **management** of several **bank accounts**.
  - clear display of **statistics** and **graphs**,
  - *'normal'* evaluation for the **add of new transactions**, with the impossibility of entering recurring transactions by default.

Future Works

We can analyze the future works of the app in 3 categories:

1. **Interface improvements**: to make the app similar to a native iOS app, we need introduce some *additional gestures*, *force touch* and *haptic feedback*.
2. **Functionality improvements**: managing the income/expense categories was a key element in the evaluation of expenses, so it is useful *insert new categories* or introduce them manually. Allow *more flexibility*, by modifying transactions once added.
3. **Feature integration**: Adding transactions manually is critical, but *adding transactions automatically* involves major issues from a security and permissions point of view.

Conclusions

Thanks for the attention!