Financit

By Victor Fresco, Filippo Pagliani, Sharifa Vos and Iñigo Sanz

Phase 1: Research

Competitive Analysis

Identify the competition

Bank applications

Santander Bank is a very popular choice to save your money in. The app holds many useful functionalities that we are interested in. For example, the app itself has a section where it informs the user in what specific aspects they've been spending their money on: basic necessities found in the supermarket, taxes, sport related expenses… We think this kind of functionality can be very enlightening for the user to know not only how much they spend but also on what. This feature is used by all kinds of people, especially young people.
It can also perform transactions to any other account given the IBAN (identification number of a bank account), but it charges some fees that tend to be a bit high. Santander bank, as any other bank, allows users to pay electronic invoices from the app.

The Santander mobile app, as well as most of the other bank apps in Spain, have a functionality called "**Bizum**". Bizum is extremely useful and popular due to how easy it is to send and receive money. It's faster than a classical bank transaction and two users can exchange money just by knowing each other's phone numbers: no need for large bank identification codes. Another improvement regarding transactions is that it does not charge any fee. It's just like giving or receiving cash money by hand, but between bank accounts. It is also very secure.

Another bank application we have analyzed is **Rabo Online Bankieren** (Rabo Online Banking). It is a free finance application for members of Rabobank, one of the most popular Dutch banks. The main goal of the application is to manage your bank transactions and check your balance (mostly like Santander Bank). However, this application has a lot of other functionalities that regular bank apps (at least in Spain) don't have. We want to implement a few of them.

First of all we want to create "vaults" in order to organize the money better. Rabo Online Bankieren has this functionality where you can make a piggy bank with a name where you can transfer money to. It is also possible to make a piggy bank and give it a certain amount that you want to save in a certain amount of time, and it gives you advice based on that (including warnings). You can also save a certain amount of money automatically if you want. This could be part of the 'spending planning' we want to implement in our application. The application also makes a prediction of expenses in the future, which could also be part of spending planning.

PayPal

PayPal was one of the first online payment methods to be available for everyone. Nowadays it can count more than 300 Million of users and billions of transactions every year. At first, PayPal was born just as a safe and fast way to manage online payment. You could link PayPal to a bank account and pay online with that, but not managing or accessing the account data. Now, it offers a much larger and complete environment, also allowing the user to store money in PayPal itself or paying and managing electronic invoices, without needing a bank account. It can even store different currencies in the same PayPal account and check and categorize past transactions (although categories are not very correct).

In addition, it allows the user to send and receive money to any other PayPal user very fast and easily. With only the email linked to the PayPal account of a user, another user can send the amount it wants. This has become the main feature of PayPal. A strong point of this functionality is that you can send or receive money from trusted people without paying any fees, or with just a small fee if you want an “extra protection” on that specific transfer. It lets the user even trace every expense it has done, but just in a chronological order without any actual further information.

Another important functionality we are interested in is PayPal’s Purchase Protection. This feature protects the user when purchasing any item online by making sure that everything is done correctly. If a purchased item arrives broken, it doesn’t arrive or similar faulty circumstances, PayPal makes sure that the money is returned to the user. It also covers irregularities in the account, such as transactions that the user didn’t make and so on.

Apple Wallet

One of the functionalities we want in our application is to store virtual credit cards whenever an implemented account allows it, so they can be used to pay in real world shops. The first application that implemented this functionality and came to our mind was **Apple Wallet**. Apple Wallet comes pre-installed in all iPhones, and it just stores virtual credit cards and all kinds of tickets. Although it’s great to also store tickets for the cinema, theater, festivals… we will only cover the credit cards part, since it’s the functionality we are interested in.

Tricount

Another functionality we want to implement is to be able to organize expenses inside a group of people. For instance, if a group of friends goes on vacation for a week, our application must be capable of managing all the expenses from the trip. It needs to account who paid/participated in what and then compute the amounts that each person must pay in order to pay off all debts. An application used for this purpose is **Tricount**.

Establish the dimensions of the product

Number of users: defined by the app store..
Score: defined in the app store.
Response time: How users perceive changes in the application. Can be either fast or slow.
How easy it is to use: Can be either easy, medium or hard.
Access speed: Can be either fast or slow.
Visual appearance: Can be either clear or confusing.
Functionality: Can be either good, medium or bad.

Analyze the competition

	Number of users	Score in app store	Response time	How easy to use	Access speed	Visual appearance	Functionality
Santander	5M+	4.5/5 (Good)	Fast	Medium	Medium	Good	Medium
Bizum	-	-	Fast	Easy	Medium	Good	Good
RaboBank	4.5M+	4.7/5	Fast	Easy	Fast	Good	Good
PayPal	300M	4.8/5(Very Good)	Fast	Easy	Fast	Good	Good
Apple Wallet	-	-	Fast	Easy	High	Good	Good
Tricount	1M+ (Google Play)	4.8/5 (Very Good)	Fast	Medium	Medium	Medium	Good

Santander

- It's interface and general feeling is not very good
- Sometimes the application freezes and does not work well.
- It can categorize past expenses but it's not very precise and usually categories are incorrect.
- Integration with Bizum
- It can manage electronic invoices

Bizum

- User data is not available since it's not an application itself, just a common service inside most of spanish banks
- Very simple: send or receive money to any phone number
- Works really well

RaboBank

- You can integrate bank accounts from other banks in the application, so you don't need multiple apps
- Users can create vaults to better manage money
- Users can install warnings/alerts for incoming transactions, messages, overspending etc.
- It can manage electronic invoices

PayPal

- It can store money independently of a bank account, with different currencies.
- It can categorize past expenses but it's not very precise and usually categories are incorrect.
- Purchase Protection works very well and protects the user from different inconveniences
- It can send/receive money to any account by its email address
- It can manage electronic invoices

Apple Wallet

- Since it comes pre-installed in all iPhones (aprox. 1 billion owners), there is no usage data in the app store such as user ratings or number of downloads. However, as individuals, we have seen a lot of people using it around us, especially young people.
- Very easy to use: In order to add a new credit card, the user just has to photograph the card or input the values if it cannot do that. To use any card, just select it and authenticate.
- Works really well: Once the user has authenticated with its face, fingerprint or passcode, it's recognised by the card readers very fast.
- It's visual appearance is very simple and minimalistic

Tricount

- The concept can be a bit confusing for new users, and the application itself does not provide any means to better understand it.
- Although it serves its purpose pretty well, it's full of very intrusive advertising which pops up when opening the application and when performing some random actions inside it, degrading the user experience.

Convert results into recommendations

When developing the **integration of different accounts,** we should be able to integrate all bank accounts but also integrate PayPal and similar services.
We need our application to correctly categorize each transaction and be able to modify or add categories in case the user wants to correct assigned ones or create new ones, as most of the bank's apps and PayPal don't do that correctly.
We should focus on the response time, as it's crucial that users are able to perceive as quickly as possible any changes in their accounts. Whether it's a transaction, an expense or any other event, it should be reflected instantly in our application. This is something that Rabobank and PayPal do pretty well.
It's also important to achieve a high access speed, but also offering a good security method to access the application. PayPal and RaboBank also do this correctly, however, Santander lacks a bit in this.

Regarding the feature to **create vaults and warnings/alerts**, RaboBank does a pretty good job. We should follow their model in which vaults can be created and warnings can be placed to alert the user of overspending. It would also be great to be able to place the same warnings without the need of a vault (i.e in any regular account). It's also a good idea to be able to create vaults with assets from different accounts. For instance, if any user has two accounts with $1000 each, a vault of $1500 could be created with $500 from one account and $1000 from the other.

About the **send and receive money** functionality, Bizum and PayPal provide a good experience, each one with their pros and cons. We should let the user perform these transactions either via phone number, email address or by identification number of the bank account. We should also avoid charging fees whenever possible, at least for amounts lower than a threshold (fees may be required by law above a certain quantity depending on the country). Most importantly, it should be fast at least for low quantities, as high quantities may need some legal bureaucracy to take place.

All bank applications we have analyzed and PayPal allow users to **manage electronic invoices** from the app. However, PayPal allows you to pay either with the PayPal balance or with any of the linked bank accounts. We could implement that in our app, so the user can manage e-invoices and pay them with any of its accounts.

Another feature we are interested in is the **warranty management** of purchased products. PayPal is a leading company implementing this, since its Purchase Protection has been a great aliade for customers for several years now. We would like to develop something similar, where not only users can return the defective products they have bought appealing to the warranty, but also request a refund when there is a problem with the shipment and other inconveniences. All these processes will be handled by us, so the user will just have to press a button or fill a form inside the app.

Regarding the **virtual wallet** functionality, it would be very difficult to create a better version of Apple Wallet, if we want it as clean and simple. It is easy to use, minimalistic and it works very well. The best approach would be to try to match the capabilities of it in our application, without trying to add any more features that would just dirty the user experience.

With respect to the functionality for **managing group expenses**, there are some aspects in which we can learn from Tricount. First of all, we need to present it as a user-friendly feature, as it can be confusing for newcomers. We can prepare a short tutorial for new users and, mostly, produce an overall clear layout. If we use advertising, it should not be used randomly along the application, and mostly, not used when performing critical actions such as consulting or paying off a debt. To finish, it would also be nice to integrate the payoffs with the functionality to send and receive money, so it can be all done without leaving the application.

It's also important to maintain a good looking visual appearance on the app, since we want users to feel good using it. PayPal, RaboBank or AppleWallet are all good examples of applications that feel very light and easy to navigate. They present the information in an affordable way so the users can focus on what they want to see instead of displaying large amounts of data at a time.

Since our app will have so many functionalities, it's crucial that they are all well organized, with an user friendly interface. It must be intuitive to access all required information and navigate through the different features. RaboBank has so many functionalities and even so they have accomplished a decent easy to use interface, so we will take them as reference.

User Interviews

Interview Structure

Presentation of the user

1. What is your name?
2. How old are you?
3. Where are you from?
4. Are you studying or working at the moment? And what do you do?

General questions related to the overall experience

1. Which finance applications do you use?
2. What do you use them for?
3. Do they help you in your daily life?

Banking Applications & Manage Expenses (Santander, Rabobank…)

1. What kind of bank application do you use?
2. Why did you download the application?
3. What do you like and dislike about the application?
4. What are the features that you use most frequently? What are the features that you get most frustrated with?

5. What do you think of a functionality that shows you how much money you spend in different sectors, for example, when you go to the supermarket, when you pay your taxes, when you go partying…
6. How often do you backtrack to past transactions? When checking them, do you always remember what you bought?
7. Would knowing what you bought each day help you somehow?

Wallet (Apple Wallet, Google Pay, Paypal)

1. How frequently do you pay virtually (with phone or card)? Is it always with the same card?
2. Do you think it's important to have more than one card? (i.e there are people who like to divide their money between different accounts what do you think)
3. If you have more than one, how important is it to keep every card and payment system united in a single application?
4. If you have more than one bank account, how do you experience taking track of your expenses?

Organize group expenses (Tricount)

1. What do you think of an app that allows you to organize your expenses between a group of people?
2. *If they don't have Tricount:* How often do you think you would need it? Would it be more useful in your country or abroad?
   If they do have Tricount: How often do you use it? Does it help?
   What do you like and dislike about the application?

Make & receive transactions (Bizum..)

1. How often have you had to send or receive money in general?
2. Did you do it virtually or with cash? How long did it take?
3. Do you prefer this way of changing money than using cash? How so?

Warranty

1. How often have you had to return a product that was sent to you in a bad shape or didn't arrive at all?
2. How difficult was it to return the product?
3. When you buy something, how important is it to you that a warranty covers it?

Electronic invoices

1. Do you think it's important to manage your invoices in advance?
2. How do you manage all those future invoices? Do you do it manually?
3. Have you ever created an invoice?
4. If yes, how have you done it?

Interview Narration

Today we're going to be talking about this new application we're designing. This application aims to organize the user's financial information in the most simple and straightforward possible way. With it, the user will be able to easily manage their multiple bank accounts and make transactions in a secure manner.
The scope of this application is for people over the age of 18 that have full control over their bank account and are interested in knowing more about their expenses and how to monitor them.

In order to finish our competitive analysis and find what's most appealing about financial applications that are already in the market, we decided to interview some guests who volunteered to give us more insight about what the average user thinks.

First of all, thank you for taking part in this. We would like to know you better:

Since we're aiming to design an application that rivals the best ones in the market, we want to know your opinion on
our average experience with the most popular financial apps.

Most people nowadays use banking applications on their mobile devices to manage their money with a couple of taps. They allow the user to make transactions, receive money and get a closer look on what and how they spend their money. We want to know your opinion on such apps.

Before, one bank account was always linked to an individual, it was strange to have more than one account. Nowadays, people like to separate their cash and the types of transactions between multiple accounts.// We want to know what you think about this:

However, our financial application shouldn't be limited to managing one's own expenses, but also take into account collective expenses with a group of people and compute from there how much the user should pay.

Many social interactions involve an exchange of money, let it be buying a beer, or giving a friend some money. We would like to know what the user thinks about these repetitive exchanges and if they should be easier to make.

When buying a certain product, customers usually want to have the guarantee that what they're getting is in its optimal condition and that it arrives on time. Which is why we want to add a feature to our application that manages product warranties.

In our society we have to pay many monthly costs such as taxes, licenses, bills… We want to make sure our users can manage such expenses in an organized way, without being surprised by an expense they didn't take into consideration.

We're grateful that you wanted to do this interview with us in your free time. For us it's really helpful to hear about your own experiences with the multiple bank applications that you are using, applications for organizing group expenses, about your paying habits and the other things we talked about. Knowing more about your preferences and opinions about the finance applications that you are familiar with along with the preferences and opinions of some other interviewed guests, it will allow us to focus our application on certain types of people, also known as Personas. Besides that, since we're aiming to design a finance app that integrates all the needs from the future users and eliminates or improves the functionalities from existing applications that can be better, we'll be using your answers to these questions to make the design process as efficient as possible.

Factoids List

GENERAL:

- User Paula is a young student, currently doing a master's degree.
- User Paula is comfortable with technologies, using some financial applications.
- User Paula only has one bank account
- User Paula only has one card
- User Paula usually doesn't carry cash
- User Monica is working, she's an english assistant at a high school.
- User Monica has multiple bank accounts from three different banks
- User Yasmin is a young student, doing an exchange program at Comillas.
- User Yasmin uses multiple financial applications.
- User Yasmin only has one bank account.
- User Yasmin has only one card.

BANKING APP:

- User Paula checks her account movements every month
- User Paula finds difficult to check her account movements in Santander App
- User Paula usually experiences internet connection problems with Santander App.
- User Monica really likes how secure her bank applications are.
- User Monica complains about having to remember three different usernames and passwords for her bank apps.
- User Monica finds it very handy that her bank sends her a message telling her how much she's spent during the month. This message also tells her the specific things she spent money on.
- User Monica doesn't like that her bank logs her out when she goes a long time without using the app.
- User Monica doesn't like that her app asks her security questions and calls whenever she logs in.
- User Yasmin monitors her purchases everyday, to know what she spends the most on. She thinks this feature could be upgraded on her app, for example make it more personalized allowing excepcional, and thus not considered, expenses.
- User Yasmin likes that the application shows her information in multiple ways, for example using charts.

- User Yasmin usually reads budgeting options as well as saving options her bank offers to her.
- User Yasmin dislikes that her expenses aren't always categorized correctly. She'd like to be able to edit it.

SEND/RECEIVE MONEY:

- User Paula sends/receive money very often via Bizum
- User Paula really considers to send/receive money virtually better than interchanging by cash
- User Monica sends and receives money very frequently.
- User Monica is worried that the application she uses in America to send/receive money will tax people when they send over 600$ in one year.
- User Monica specifies how useful this feature is, even for big businesses and other circumstances that aren't just going out.
- User Yasmin finds it very handy that she can pay her debts off via her phone, but she does it virtually only if it's not a small amount.

WALLET:

- User Paula pays very often with her bank card
- User Paula finds cash useful but she really likes paying virtually
- User Paula likes the idea of unifying all payments/cards in a single application
- User Monica has three different cards for different necessities.
- User Monica barely pays with cash.
- User Monica usually pays with the same card.
- User Monica finds it very important to have more than one card, as losing it would keep you out of using your money.
- User Monica would like to have all her accounts united in one single application.
- User Yasmin pays almost always with her card or with her phone.
- User Yasmin says that if she had more than one card it would be very useful to have them unite in one single application.
- User Yasmin always pays with the same card

TRICOUNT:

- User Paula doesn't use Tricount for single expenses within a group
- User Paula really likes to use Tricount when she is abroad with a group of people to manage some expenses through the trip
- User Monica likes that she can give detail to what she bought, so other people don't get confused over that expense.
- User Monica would like to have an easier way to get rid of these debts.
- User Yasmin never used an application like Tricount.
- User Yasmin said that she's interested in starting to use it especially if it's integrated in the bank application.

WARRANTY:

- User Paula takes a lot of time to return products online
- User Paula wants an app that integrates the feature to take care of the warranty of online purchased products

- User Monica has many times experienced how she ordered something online and it never got sent to her.
- User Monica doesn't like how she has to handle something that's the business' fault.
- User Yasmin usually doesn't do online shopping so she never had to return something.
- User Yasmin says that if she needs to buy something online in the future, the warranty is essential.

INVOICES:

- User Paula manages her invoices manually (excel, email)
- User Paula finds interesting to be able to manage invoices via an application
- User Monica thinks it's very important to see all her future invoices in advance.
- User Monica manages her invoices manually as they are always the same.
- User Yasmin isn't so interested in the possibility of an application that reminds her of the incoming invoices

Conclusion Milestone 1:

In conclusion, for this milestone we've had to meet some people and find their likes and dislikes about certain financial applications that are going to influence our product. We've done an extensive interview to get as much information as possible and to get a series of facts called "factoids" that describe in bullet points what each user thinks about each part of our future product.

Phase 2: Modeling and Requirements

Modeling

Brief explanation of the chosen process

We chose the bottom-up approach because we think that, in our case, it's easier to extract the necessary personas from our research data than guessing our target category of users.

Next, we documented each part of the process:

1) Behavioral variables

AXIS:

Activities (What the user does, frequency and volume)
Attitudes (How the user thinks about the product domain and technology)
Aptitudes (What education and training the user has; capability to learn)
Motivations (Why the user is engaged in the product domain)
Skills (User capabilities related to the product domain and technology)

VARIABLES:

- How often they check last transactions (Activity)
- How often they send/receive money virtually (Activity)
- How often they pay with their phone (Activity)
- How often they check electronic invoices (Activity)
- How often they buy online (Activity)
- How interested the user is in managing their expenses (Attitude)
- How much they worry about the application being secure (Attitude)
- How comfortable is the user with technology (Aptitude)
- How much financial education the user has (Aptitude)
- How often they use their bank account (Motivation)
- How often does the user get engaged with money (Motivation)
- How difficult is it to manage their bank online (Skill)
- How difficult is it to return a product (Skill)

2) Map interview subjects to behavioral variables

Frequency / Activity	**Paula**	**Yasmin**	**Monica**
Checking transactions	*Med*	*High*	*Med*
Send/Receive money virtually	*High*	*Med*	*High*
Pay by phone	*Low*	*High*	*High*
Check electronic invoices	*Low*	*Low*	*Med*
Buy online	*Low*	*Low*	*High*
Interest in managing expenses	*High*	*High*	*Med*
Worry about the app's security	*Med*	*High*	*High*
Comfortable using technology	*High*	*High*	*Med*
Financial education	*Low*	*Low*	*Med*
Use bank accounts	*Low*	*Low*	*High*
Money engagement	*High*	*High*	*High*
Difficulty to manage bank online	*High*	*Low*	*Low*
Difficulty returning products	*High*	*Low*	*Med*

3) Identify behavior patterns

We see two types of users: the casual and the more technical/engaged one.

Casual → Paula
Engaged → Yasmin & Monica

The casual behaviour describes someone who doesn't use the full potential of the financial application: they only care about the money they have and what they spend and some other basic operations. They usually don't want to use multiple bank accounts because they don't find any use for it. They simply see the application as a tool to manage their money in a simple and easy way.

Meanwhile, the engaged behaviour describes those users who are more interested in how to organize their money: they use multiple bank accounts where they frequently send money from one account to the other, they have a good overall financial knowledge, they also usually pay by phone using one of their multiple bank accounts to separate types of transactions, they are interested in security and look for a good experience buying online.

4) Synthesize characteristics and relevant goals

CASUAL PERSONA:

BASIC INFO:

- NAME: Maria García
- SEX: Female
- AGE: 45
- LOC: Barcelona
- JOB: Works at a tea shop
- SALARY: 1.400€/month

BEHAVIOUR:

- POTENTIAL USE CONTEXT: María García wants to use this application to get an overall look of her bank status every so often, without going too much into detail, as long as she doesn't see something weird.
- CURRENT SOLUTIONS: She wants an application on her phone that she can always have on her to check her bank balance and her latest transactions.
- CURRENT FRUSTRATIONS: She always uses the internet browser to check in her bank account and transactions, and since it's a long process, she barely does it.

GOALS:

- END GOALS (3-5)
    - Have enough money to gift things to family and friends
    - Anticipate big expenses and save enough for her family.
    - Get a general view of her financial situation
    - Anticipate problems in order to avoid stress.
- LIFE GOAL (1):
    - Live a peaceful life
- EXPERIENCE GOALS (2)
    - Feel relaxed with her money
    - Feel like her money is safe

ENGAGED PERSONA:

BASIC INFO:

- NAME: James Smith
- SEX: Male
- AGE: 29
- LOC: Madrid
- JOB: CTO of IT company (his own boss)
- SALARY: 2.100€/month

BEHAVIOUR:

- POTENTIAL USE CONTEXT: Daily use of the application for every aspect of financial activities.
- CURRENT SOLUTIONS: A lot of different applications for each of the tasks he wants to do
- CURRENT FRUSTRATIONS: Some applications don't work as expected or are not as efficient as he would like

GOALS:

- END GOALS (3-5):
    - Integrate all financial activities into one single place
    - Manage his expenses very efficiently
    - Send/receive money fast and without any taxes
    - Doesn't want to deal with problematic situations involving online purchases
- LIFE GOAL (1):
    - Be successful by having an important job and earning a lot of money
- EXPERIENCE GOALS (2):
    - Feel smart / in control
    - Always feel efficient

5) Check for completeness and redundancy.

Most information present in the data set is present in the personas: from our interviews we defined two very separate behaviours. One of the behaviours is characterized by Paula's interview and the other behaviour is characterized by Yasmin's and Monica's interview.

6) Expand description of attributes and behaviors

These are the personas we have created (one per page), with an overall description as well as how they would be involved in the application:

MARÍA GARCÍA

The casual user

Age: 45
Sex: Female
Nationality: Spanish
Location: Barcelona
Occupation: Owner of tea shop *Tea Time*
Contact: maria.garcia76@gmail.com

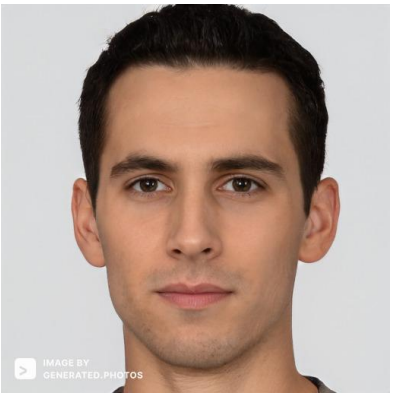
"Nothing beats a good cup of tea in relaxation"

Description:
María is a loving mother of three children of the age 10, 14 and 18 and she is happily married. As an addition to the family, they have a dog. She opened a tea shop with her sister in the center of Barcelona called *Tea Time*. She divides her time by working in Tea Time, taking care of the family and the house and practising yoga. Taking into account all the expenses she pays for the shop, she earns around 1.400€ per month. María wants a peaceful future for her and her children, without ever having to worry about money.

How is María going to use the application?
María is comfortable with technology but is not very into it at the same time. She does not want to spend too much time on managing her money, so she will use the application from time to time only to use the most basic functionalities, for example accessing her bank accounts and having a quick overview on the transactions. It is helpful for her to have multiple accounts together in the application, since she has her own accounts and also one for the tea shop, so she can oversee them easily. She will use the invoice management system which helps with the planning of the expenses of her business and makes transactions easier.

What is important to María when using the application?
María says: *"I'd rather spend time on things I enjoy than spend time focusing on the money that I have."*

- It is important to her that she can see the account balance immediately after verifying she is the user.
- It is important to her to have a quick and clear overview of expenses and transactions.
- It is important to her that the information is very minimal, but if there are some confusions she wants to be able to get more detailed information.



JAMES SMITH

The engaged user

Age: 29
Sex: Male
Nationality: English
Location: Madrid
Occupation: CTO in tech start-up bIT
Contact: james.smith@protonmail.com

"When something is important enough,
you do it even if the odds are not in your favor"

Description:
James is the son of a Spanish woman and an English man and lived in England until he was 17 years of age. He moved to Madrid to start with his study in Computer Science and began working after he graduated. He left the company he was working for one year ago (2020) to start up his own tech company called bIT where he is the CTO. James is in his own words very focused on his work and career and wants to put most of his time, energy and money into the company. He does manage to obtain a relationship with his partner Alex who he lives with in Madrid.

How is James going to use the application?
James is interested in using the application on a daily basis for every aspect of financial activities. He likes being on top of his money and on what it is being spent by managing all of his expenses and transactions in detail. He wants to do this in a very efficient way. Besides that, he will be using the application for sending and receiving money fast and without having to pay any sort of taxes when he does that. He also wants to pay with the application to make online purchases and have an overview of the warranties.

What is important to James when using the application?
For James the most important thing is that the things he does are efficient. He says the following about this: "Time equals money and I don't want to waste either of them."

- On top of the list is efficiency of the different functionalities, but especially when he needs to backtrack the transactions. This is reflected in the importance for him to be able to integrate all his financial activities into one single place.
- James doesn't want to deal with problematic situations involving online purchases.
- He wants to feel in control.
- He wants the application to be totally secure.

7) Designate persona types

The casual and engaged users are both primary personas, named respectively María Garcia and James, since they both represent the main target users.

Requirements

1) Problems and visions

Problem:

- There exist a lot of different applications for all sorts of financial activities, including different banks and financial accounts. This is very inconvenient for the users, since they have to switch between applications to perform all these different things, remember different passwords, etc.

Vision:

- Our application will integrate the majority of these financial activities and all their bank and online wallets accounts, so users can just log in in our application with one password/code or biometrics and be able to manage all their expenses, send and receive money, etc.

Problem:

- Most financial applications don't categorize the expenses properly, giving the users a very confusing view of their financial state.

Vision:

- Our application will let the users easily categorize their expenses and will suggest categories based on other user's choices.

Problem:

- Applications to send/receive money sometimes charge taxes for large amounts of money.

Vision:

- Our application will handle these taxes so the user pays only the minimum amount possible

Problem:

- It could be difficult to manage all the expenses inside a group of people in a situation like a trip or between roommates.

Vision:

- Our application will manage it automatically and in the smartest way possible trying to simplify all the debits and make it easier.

Problem:

- When purchasing products online, if users have any problem along the way they may have to deal with a lot of inconveniences such as reaching out to the company behind the product, refunds, long waits, etc.

Vision:

- Our application will handle this process so if the user has any problem with an online purchase, it will reach out to the company and solve the problem for the user.

Problem:

- When processing electronic invoices, users can only pay them with the application that has processed it. If they want to pay them with a different account, they must import the invoices to the corresponding application.

Vision:

- Our application will let the user import an electronic invoice and then pay with any of the available accounts, making it easier to manage them.

Problem:

- When buying in local stores like a supermarket or a bakery, there is no way to get the information about the bought items digitally

Vision:

- Our application will let the user scan tickets for each transaction so they get digitized and seen in the transaction.

2) Summary of ideas taken from brainstorming

- Our application will allow multiple account integration in order to summarize the transactions. These accounts can be regular bank accounts, online wallets, etc
- Our application will allow categorizing each transaction with user defined categories, default categories automatically set by the application or even categories other users have defined.
- Our application will let the user define expected transactions for the future, so if one of these does not occur, they get notified.
- Our application will support alerts and notifications for the users to know when some important events happen, such as previously explained expected transactions, when they spent too much money, etc.
- Our application will support scanning tickets to digitize and attach them to a transaction as well as attaching other documents such as files or pdfs.
- Our application will have a wallet with virtual credit cards for each account
- Our application will allow to send & receive money from each account
- Our application will support managing expenses inside a group
- Our application will allow integration of electronic invoices whenever a transaction supports it
- Our application will support warranty management of purchased items as long as the transaction allows it
- Our application won't have a trading functionality
- Our application won't support cryptocurrencies

3) Personas expectations

Maria:

Maria is a working mother and wife and likes to spend time with her family instead of having to worry about finances.

Maria would like to see her expenses clearly, so she can easily know if she has spent too much or if she has enough money for a treat, but she's not going to use all the functionality at their maximum potential since she doesn't really like technology.

For these reasons Maria would like the application to be as easy as possible at least for the basic functionalities.

Since Maria has her own activities she likes the opportunity to manage more accounts inside the same application in order to have an easy way to access both her personal account and the one from her shop.
Another very important functionality for Maria is the possibility to manage all the invoices of her store in an easier way and in advance.

James:

James is the CTO of a very important business and he believes that money rules the world. This is why James wants to save his money in the most efficient possible manner.

James likes to divide his personal money and business income between multiple bank accounts so he can do certain transactions with one or the other.. Since James needs to keep track of the business' expenses, he needs to see every transaction made from that account in a very detailed and clear way.

To James, his money is his most important asset which is why he wants it to be in a completely secure environment. He's periodically monitoring his money because he's anxious that something might happen to it. He wants to feel in complete control of his account.

James is too busy in his day to day life so he wants a type of service that doesn't give him any trouble. If he feels like the application isn't working well, he will without a doubt switch to a different application.

4) Context scenarios

(Maria - expense categorization)
It's almost the end of the month and Maria has to control the flow of money of the shops to see how the things are going, so she opens the application and thanks to the automatic categorization of the expenses provided by the application, she can see a detailed chart that shows how much money she has spent and on what.

(Maria - invoices)
Maria wants to take her family to dine at a fancy restaurant but she doesn't know whether she'll have enough money to look after her shop after the end-of-the-month invoices. To make sure she has enough, she looks at the 'Invoices' section in the application, where it clearly indicates how much she will have to pay and why. This way, it won't surprise her once the money is subtracted from her bank account.

(Maria - send/receive money)
Maria is in the tea shop, as every morning, when she receives a message from her oldest child, telling her that he has forgotten the lunch, and as he doesn't have any cash, he cannot buy anything for lunch. Luckily, Maria created a bank account for her oldest child so he could use it for emergencies. Maria sends money to her child and now he can go to the bank near his high school to get some cash to buy lunch.

(Maria - attach tickets and other documents)
Maria is not a very meticulous person, so when she is reviewing past transactions she sometimes forgets about the context in which that transaction took place. Luckily for her, with our application, she can attach whatever file she wants to any transaction, including the ticket, so she can remember what it was. This way, if she goes to a family dinner in which she spends a lot of money, she can scan the ticket and attach it to the transaction and even attach a photo of the family in the restaurant so she remembers clearly everything about the dinner.

(James - custom expenses for each bank account)
Every morning, James sits in his office to check the news and the stats of the global market. After he has done this, he likes to take a look at all his personal accounts. He goes one by one checking all the expenses of the past day, categorizing each one with custom categories he has defined. He also reviews all expected movements, to see if they are correct and to take a look at what's expected to be next if everything goes fine. After he has finished checking each individual account, he likes to see his global status, to get a more general view of all his transactions.

(James - alerts)
James is taking a break at a cafeteria near his company's office, checking Twitter to know what the important CEOs of other companies think about the current value of Bitcoin, when he receives a notification from our application. It says that an important periodic transaction was expected for today, but it did not occur. James goes quickly to his office to check what's going on and he finds out that one of his clients hasn't paid him, so he quickly calls the client and solves the problem.

(James - managing group expenses)
Every last Friday of the month James goes with his friends to their favorite rooftop bar in the center of Madrid to have dinner and catch up. After dinner the guys drink some beers, but since James has to work a bit the following day, he'll be leaving earlier than his friends. One of the guys tells James he'll pay everything and will put the amount everyone has to pay in the application. The next day James checks the application, goes to the managing group expenses section and sees directly how much money he owes his friend.

(James - wallet)
James wants to buy a new computer since his old one is getting slow, but since he uses the PC mainly for his work he doesn't want to pay with the card that he uses for his normal daily purchase, but with the one that he uses for work. So, as always, he's going to pay with the card through the phone, but since his daily life card is selected as the default one, he has to open the application and change the card to the one he wants to use, it being the work card.

(James - warranty)
James had just bought the new Iphone but when it arrived he noticed that the screen was broken. Luckily he used our application to buy it and so now he just has to open the application and require the restitution and we will contact the vendor for him and manage everything.

5) List of requirements

- Allow users to link different financial accounts (regular bank accounts, online wallets, etc.)
- Categorize correctly the expenses of the month in a cleanly manner and without errors
- Allow the user to define expected periodic or promptly transactions
- Alert the user when some events happen, like an expected transaction not taking place or spending too much money.
- Allow custom expense categories so the user doesn't leave out any kind of personal transaction
- Send, or receive, money to other people needing just their email address, phone number or bank account identification number
- Scan and digitize tickets and attach them and other files to a transaction
- Use virtual credit cards from any of the linked accounts, securely and with no delays
- Manage in a smart way the expenses inside a group of people simplifying the exchange of money
- Manage electronic invoices so the user can pay them with any of the linked accounts and consult them later.
- Manage warranties of purchased products so the user doesn't have to get too involved in the refund/return process

Conclusion Milestone 2:

In conclusion, this milestone allowed us to find and describe very specific personas using the bottom-up approach. Our main goal is to satisfy these personas since they make up the most part of the community that will use our application. Since these personas have certain frustrations and opinions about other financial apps, we want to make a list of requirements that shows us which functionalities we should and should not implement.

Phase 3: Design framework and interactive final prototype

Define the form factor, posture and input methods

Form Factor: Our application will be only available for mobile phones. It may be used by users everywhere, since it's an application to manage financial aspects of users lifes.

Posture: Our application will use a "Transient posture". It is specifically designed to manage the user's financial life, so users will mostly use our application during short periods of time, to perform specific tasks such as checking recent transactions or specific ones, sending/receiving money, checking the status of a refund, etc. The most important aspect of this is to create clean and easy to use interfaces, so users can rapidly do what they want without any distraction.

Input methods: The primary input method will be the touch screen of the mobile phone.

Define data and functional elements

Data elements:

Element: Wallet
Attributes: Holds every account linked to the application and displays the total amount of money it contains.
Relationship to other elements: Accounts saved inside

Element: Account (bank, PayPal...)
Attributes: Money saved in the account, transactions done from said account, electronic invoices and warranties related to the account, alerts
Relationship to other elements: Wallet

Element: Contact
Attributes: Money sent and received from each contact, their information (phone number, e-mail address…), how frequently they get money sent to
Relationship to other elements: Bank account, Transaction (in send/receive money, they'll appear as frequented contacts)

Element: Transaction
Attributes: Category of the transaction, money spent/received, date, person or company at the other end of the transaction
Relationship to other elements: Bank account and contacts

Element: Invoice
Attributes: Money related to the invoice and the information linked to the invoice, finalized status
Relationship to other elements: Bank account and Wallet

Element: Warranty
Attributes: The expense it's linked to, money of the order and expire date, status
Relationship to other elements: Transaction, bank account

Element: Group (for group expenses)
Attributes: Account linked, transactions by its members, each member's balances (what they owe to each person)
Relationship to other elements: Contacts

Element: Group Transaction (for group expenses)
Attributes: Account linked, member who made the transaction, money and date
Relationship to other elements: Group

Element: Transaction Alerts
Attributes: Date of the alert when the transaction is expected, amount of money to be received/sent, account from which transaction is expected.
Relationship to other elements: Transaction, Account, Wallet

Element: Account Alerts
Attributes: Money threshold, account(s) linked
Relationship to other elements: Account, Wallet

<u>Functional elements:</u>

ACCOUNTS:

- Users can link any kind of financial account to the Wallet with the press of a button, where they'll have to enter their account details to be able to manage it
- Users can use any kind of account identifier: e-mail (paypal), phone number, IBAN (bank account)... to insert their new account
- Users can delete a certain bank account going into their settings. Since our app only manages the bank account, the account's money as well as all its associated information like transactions etc. will be safe in their bank account.
- Users can pay for something by clicking on an account in the Wallet, this will prompt the NFC sensor for the specific card.

TRANSACTIONS:

- When paying for something, the type of the expense will be inferred based on where you bought it and past similar transactions.
- By tapping in an expense, the user will be able to gather more information about it as well as change its expense type.
- All transactions will be shown in-order of date and account.
- The user can specify new expense types.
- Users can send and receive money from another user by having one of their linked credentials, this may be email, phone number, IBAN..
- When clicking on a transaction, the user can add multiple files (PDF, image…) to add more information of what it was about.
- The user can delete files linked to a transaction if they feel they don't fit anymore.

ALERTS:

- The user can define an alert of type "expected" to inform the app that they expect to receive a transaction from a certain contact with a certain quantity due by a certain date. If the transaction is not received by the specified date, the app will warn the user.
- The user can define an alert of type "threshold, general" where every expense done during the month will be summed up and compared against a threshold. If it goes over it, the app will inform the user.
- The user can define an alert of type "threshold, specific" where every expense of a certain list of categories during the month will be summed up and compared against a threshold. If it goes over it, the app will inform the user.

GROUP EXPENSES:

- Users can be added into an Expense Group with other users to share certain expenses between them all
- When adding an expense in an Expense Group, the user can specify which members have to pay for it.
- Users can see how much each person owes to you and the others.
- Users can see the total balance of the group (how much each person owes in general)
- Users only need to click "Pay" once, and the app will take care of everything.

INVOICES:

- Users can pay in advance a certain invoice before its due date
- Users can filter by "completed" and "not yet completed" invoices.

WARRANTIES:

- From the "Expense" menu, users can decide which expenses to get a warrant of.
- When a user creates a warranty, its process is managed automatically.
- Users can check the current status of a warranty.
- Users can filter by "completed" and "not yet completed" warranties.

Explanation of iterations

We have made five iterations in total. Four of them have been done in a competitive way, where each of us created one sketch without seeing the others. The fifth iteration was made in collaboration, so we took the best aspects of each of the competitive ones and improved some of them.

Victor

The iteration made by Victor was focused on providing the user with the functionalities we offer in the cleanest and easiest way possible. It proposes a modern and attractive design so users like to navigate the app. It includes a menu to go to each individual functionality.

In order to check transactions and accounts data, the user can go to the *Wallet* tab and check any transaction, or go into a specific account and check only its transactions. They can be filtered and searched. To send or receive money, there is the *Exchange* tab, which provides an easy to use interface to send and receive money. *Virtual Cards* tab shows the users all their virtual cards linked to existing accounts and last transactions made with the selected card. *Alerts* tab lets the user configure two types of alarms: threshold and expected. They respectively correspond to alarms notifying the users that their money has gone down a specified threshold and to alarms alerting that an expected transaction has taken place or not. *Warranties* tab provides the user with the possibility to manage their refunds and *Invoices* tab to manage their invoices, both showing in a clear way all the needed information. Lastly, *Group Expenses* tab is for managing expenses inside a group and paying them easily with one of the linked accounts.

Iñigo

Iñigo's iteration was more involved in displaying and explaining as much as possible the information shown in the app, instead of giving the user a simplified look. Right after logging in, the home screen displays general information related to the user's bank accounts: their total money, the accumulated expenses for the last months, as well as a rundown on how much money each account has and their balances. Among the multiple functionalities, the user has the capability to bind certain contacts to an account so that any money they receive from them is directly linked to said bank account. For unbound contacts, the money received will be sent to the user's default account. The user can also set up alerts for specific categories if they want to keep a certain type of expense in check, or general alerts to keep track of all expenses in general. Some other types of alerts, called "expected alerts", can be set up if the user expects a certain amount of money from one of their contacts before a due date. The user will also be able to filter their expenses as needed, in case they want to view certain types of expenses coming from certain accounts.

The user is able to see a lot of information related to their transactions, as well as interacting with the app's AI, which predicts how much money the user will spend by the end of the month, based on the actual and past transactions and buying habits (i.e: drinks every friday). The user can also take a look at what their most usual type of expenses are.

The user can send and receive money easily through our app's menu. They must select the specific account they want to make the transaction with as well as the contact, the amount

and finally input their password. When selecting a contact, the most frequent ones will be displayed under the “frequent contacts” tab so the user gets faster access to them.

With group expenses, the user is able to manage multiple groups of people and their expenses at once. The user can only view how much they owe or must be paid by their friends, as well as the group’s last transactions. Each transaction is specific and can be owed by certain people.

The invoices section allows the user to view all their expected end-of-the-month payments, the date their due and their status. They can be paid by the user before they’re due if they wish to do so.

In the expense section, the user can select a specific transaction and, if they are refundable, they can request a warranty. The warranty will display its status (pending, finished, cancelled…) as well as the price of the item bought. Each time the warranty changes status, a notification will be sent to the user.

In conclusion, this iteration was mainly focused on the advanced user, the one that wants as much information as possible.

Filippo

The iteration made by Filippo was focused more on what was the most frequent functionalities and the one that has to be reached fast and easily. The first one is the possibility to change the card that you want to use to pay if you are paying with your phone with the NFC and that’s why the first thing you see after being logged in is the HomePage with on the top all the cards that you have and the possibility to swap between them just swiping. Other important information like the total balance available and the last transactions are still in the home page, and speaking of this one you can access the more detailed screen about the last transactions just by clicking on it from the homepage. All the other functionalities are reachable from the drop menu that you can open from every point of the application clicking on the top right corner button since they don't usually need to be used in context that require to be fast.

Sharifa

Prior to the sketch a list of all the functionalities was made including the different screens that belong to the functionalities. The first screen is a screen with the name and logo of the app and it automatically goes to the login screen. After a successful login the user will see the menu with all the functionalities. By clicking on a functionality the user goes to that certain screen. I tried to put all the important information on the screens, without it being too much or too little, but with clarity and it being easy to use for the user. During the process of designing the screens, multiple changes were made such as placement of things, forgotten aspects etc.

Collaborative

As we said in the beginning, for the collaborative iteration we took the best ideas from all previous ones, and improved others that didn't quite make the fit. We added security to the app, so the user needs to login with biometrics or a pin. We also used a very clear interface with a menu where the most important functionalities are located.

Our main goal in this collaborative iteration was to merge the simple, aesthetic and modern looks of Victor's and Sharifa's iterations with the more complex and detailed views of Filippo and Iñigo's iterations. To do that, we reviewed and redefined some functional elements to make sure all aspects were included in the collaborative iteration.

This iteration starts off very user-friendly: welcome screen, pin/biometrics request and the possibility of asking for a new password in case the user forgot their previous one.
Then the user is taken to the home screen where they can see all the general information of their accounts as well as the total money.

By tapping on one of their accounts, the user will be able to see in greater detail the money influx as well as a prediction made by an AI on how the account's expenses will grow. They'll also be able to see their current expenses both categorized and separated, with the option of filtering them. At any time, a user can create a new account and either link it to a bank account or simply have it as a new storage account.

Our application is supposed to manage the user's expenses in an efficient and clean manner, which is why they are categorized between multiple categories so the user can look through them easily. The expense categories themselves also provide very useful information, such as the average money spent per month in a certain category, or how often they buy a product of said class. The user will be able to make sure these expenses don't go out of hand by placing an alert (either specific to certain expense types or general) to notify when it's going through a certain threshold. We expect the user to use these tools to manage their expenses responsibly.

The user can easily send and receive money. They only have to input an amount, select a contact, the account they want their money to be sent from/with and a concept of their choosing. For easier and faster access, the user's most frequent contacts will be displayed in case the user needs them. If a user is expecting one of their contacts to send them money, they can place an alert of type "Expected" that will notify the user if said contact doesn't make the transaction before a due date.

At any time, the user can use one of their virtual cards to pay for a transaction. They only have to go to the "Virtual cards" section and choose one of their different cards. Each card, when selected, will display the last transactions made by said card.

Transactions are easily managed with our application, since they're all categorized automatically. They can be filtered to select certain transaction types. If the user selects a transaction, they'll be able to change their category type and if they wish to, add a new

custom category type. The application will end up which kind of transactions are related to this custom category and will automatically categorize those types of expenses too.
A transaction can be linked to a certain document or picture in case the user wants to remember why it was made. They can digitalize the receipt and, if it was digitized incorrectly, edit its contents.

The user can manage group expenses by letting our application properly share the expenses between its members. For each transaction related to the group, the user that uploaded it must specify which members must also pay for it. Users will be able to see their balance and pay their debt at any time.

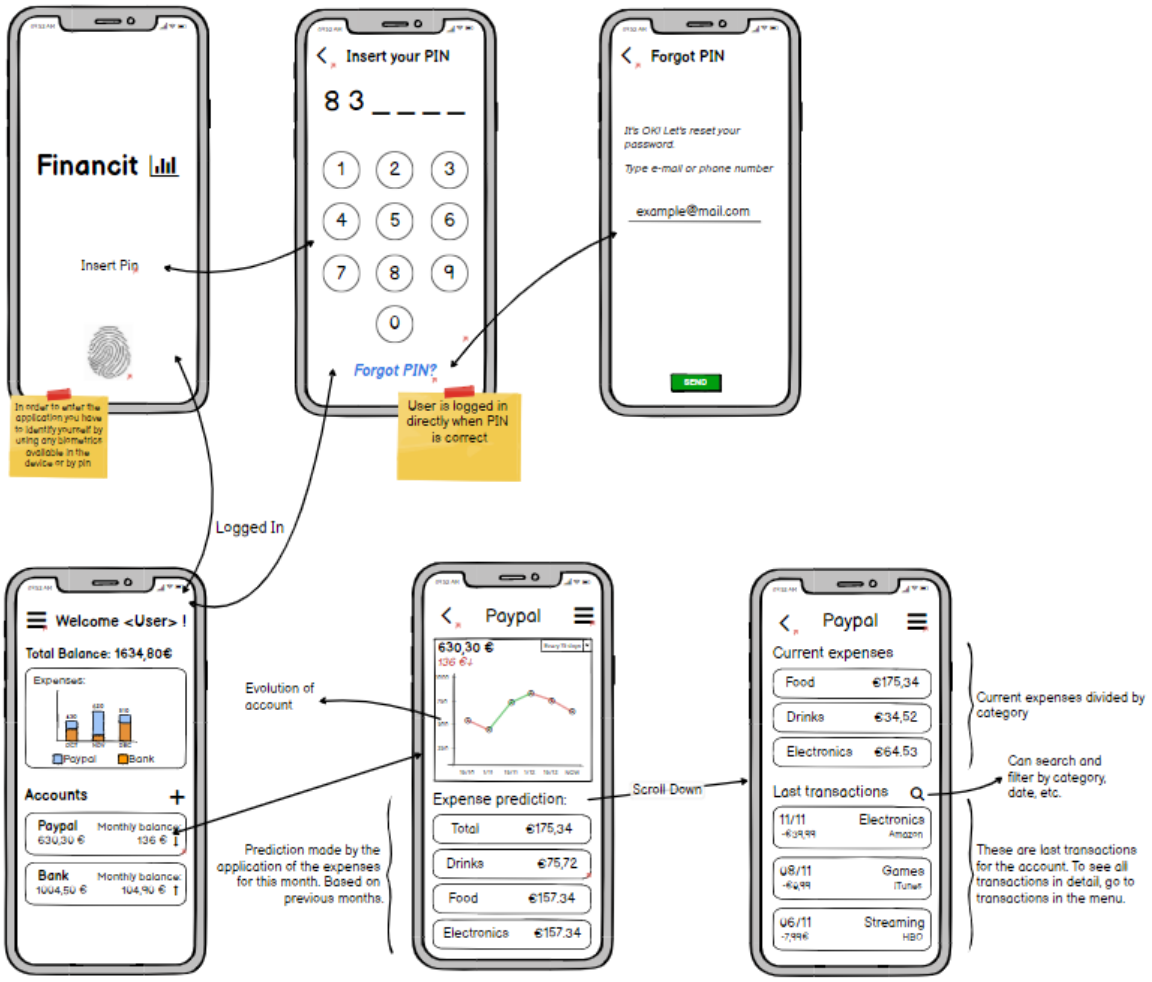
The invoices section allows the user to view all their expected end-of-the-month payments, the date their due and their status. They can be paid by the user before they're due if they wish to do so.

Warranties allow the user to easily request for a refund if the transaction is applicable. Once done, the user will be able to see in the "history section" the status of the request, whether it's been accepted, cancelled, if it's still pending or whether the store wants to get in contact with the user.

In conclusion, this collaborative iteration involves all our previous iterations, where we displayed the best features of each one of them as well as a rework of the ones that didn't make it.

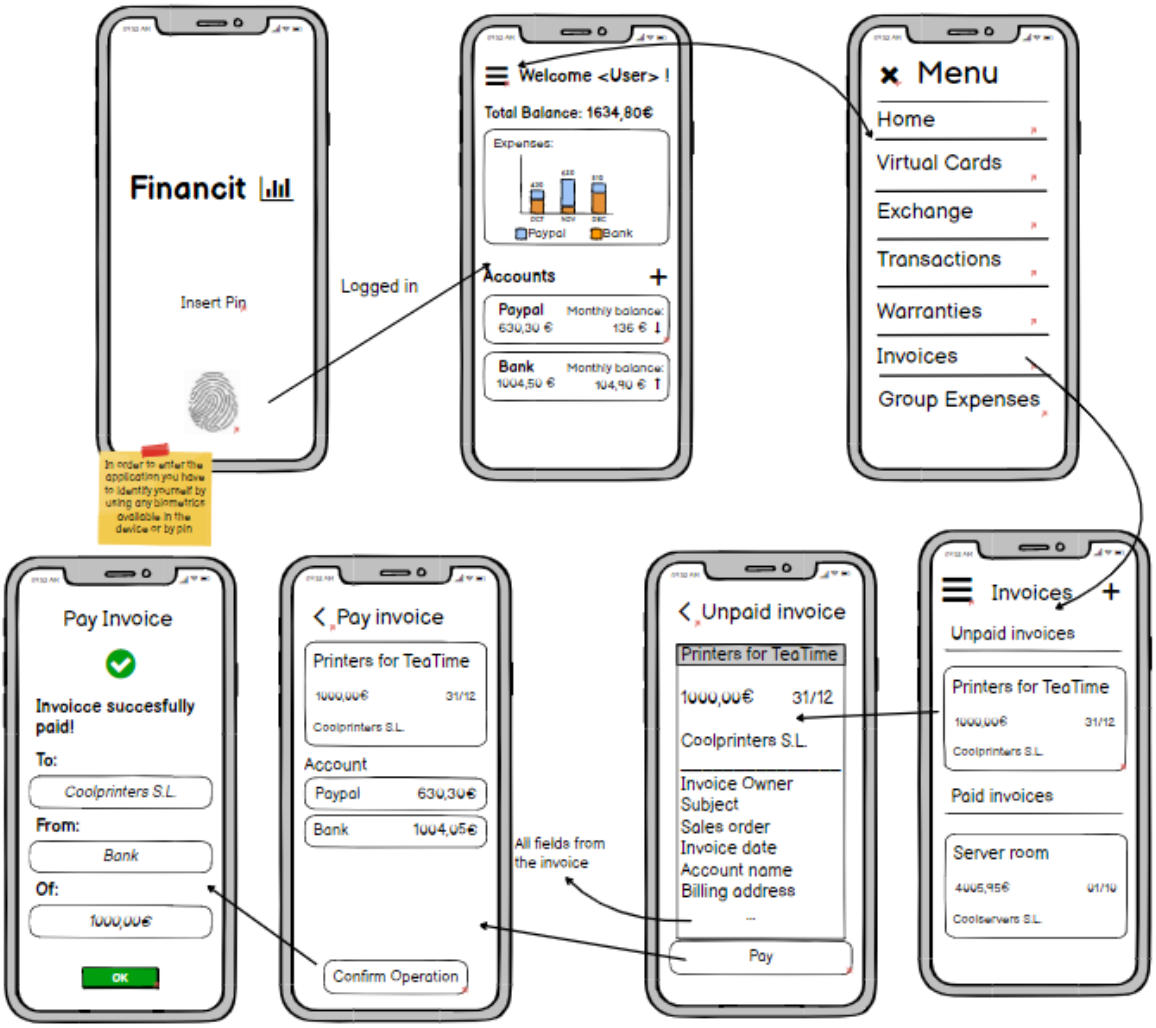
Key path scenarios

(Maria - expense categorization)



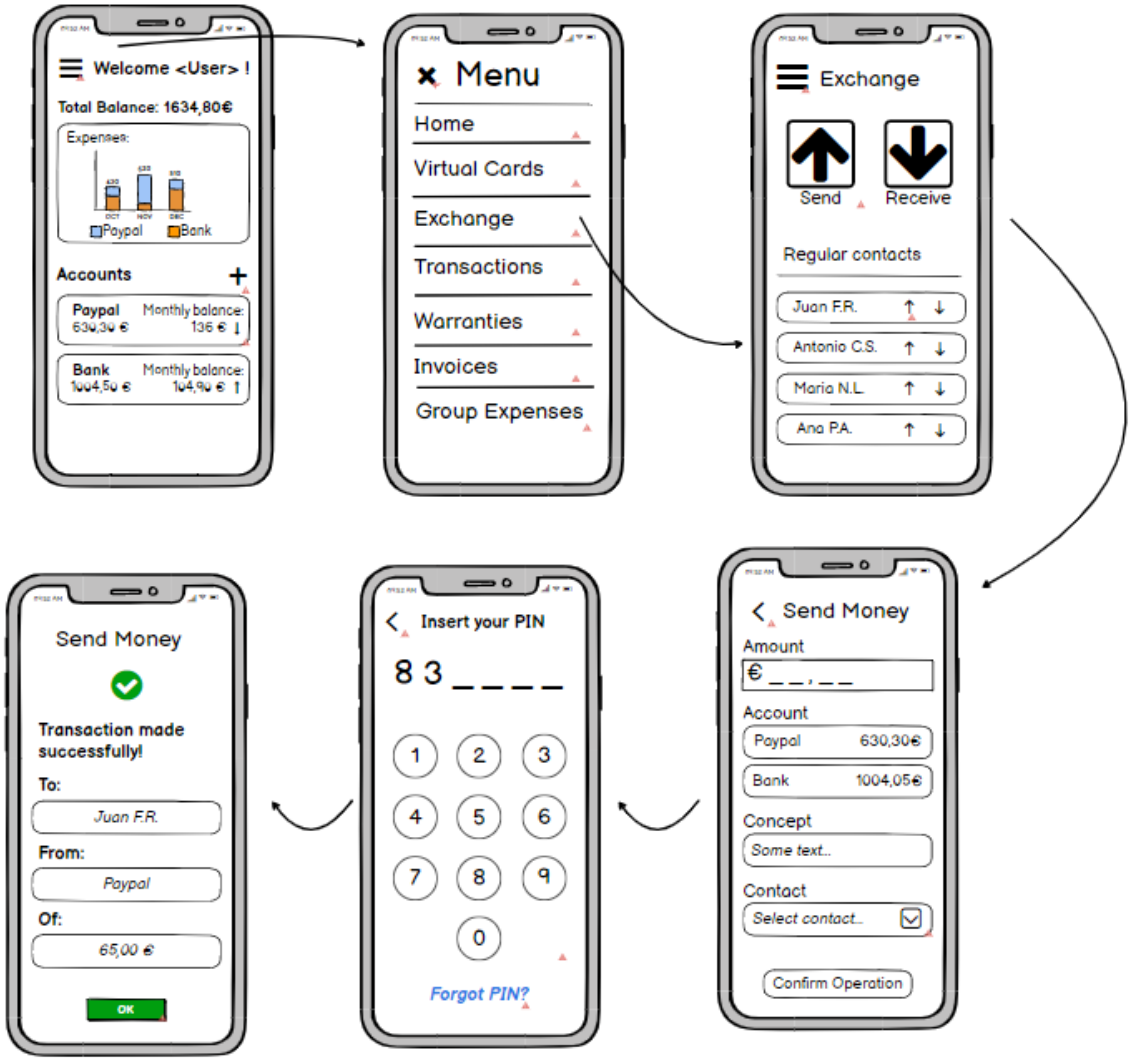
It’s almost the end of the month and Maria has to control the flow of money of the shops to see how things are going, so she takes her phone and opens Financit where she first has to authenticate herself using her fingerprint or her pin. Once she’s logged in she can see in the homepage a first view of the status of each account: how much money they have, how much each account spent during the last months… To gain more insight on how her work account is going, she can click on it to see more information about it: a graph displaying the influx of money from that account, how much money she spent on which particular expenses, as well as the last transactions related to those categories. Finally, the app shows her a prediction of how much money she will spend by the end of the month based on her previous and current purchases, also based on the expense categories.

(Maria - invoices)



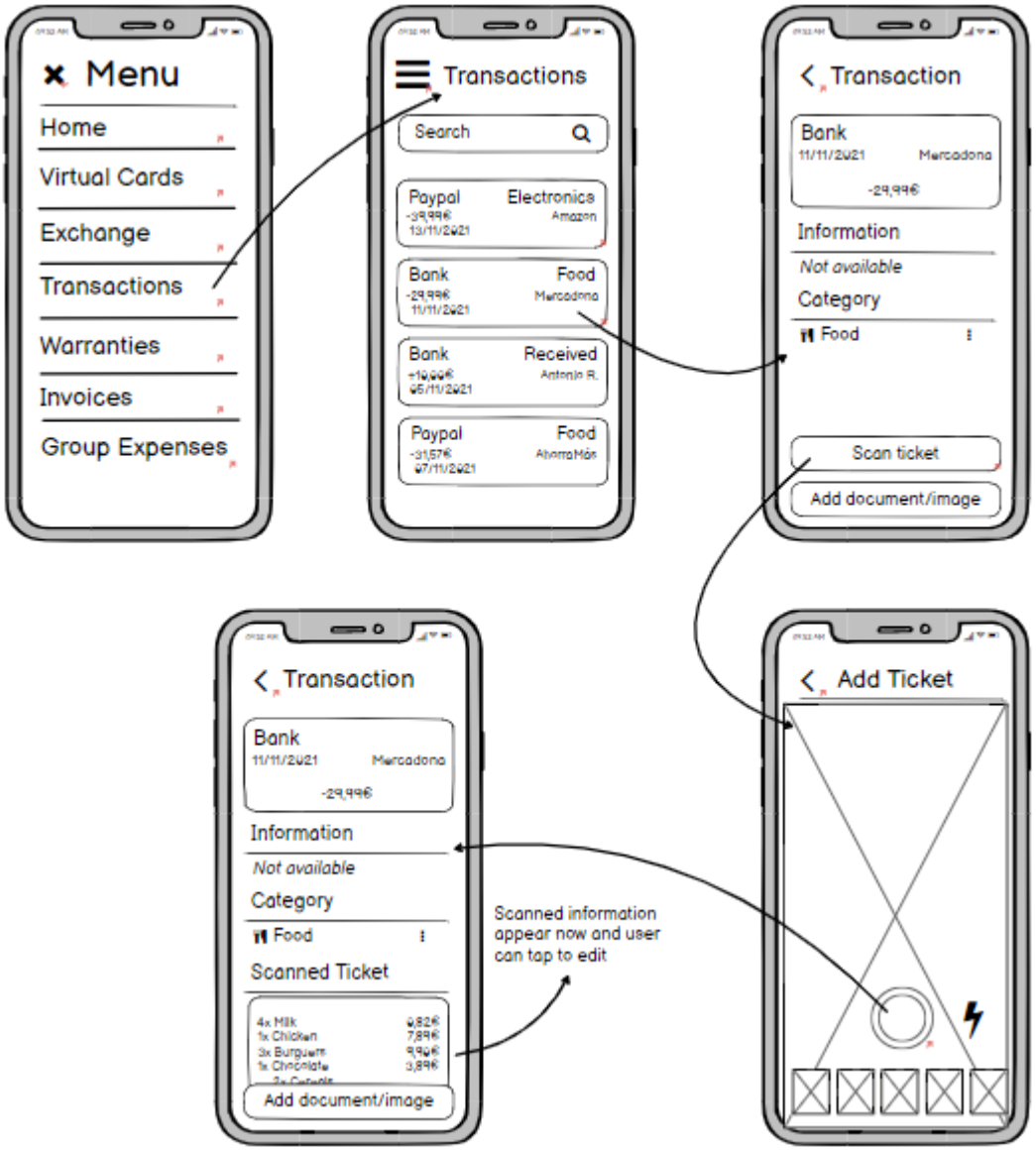
Maria wants to take her family to dine at a fancy restaurant but she doesn't know whether she'll have enough money to look after her shop after the end-of-the-month invoices. To make sure she has enough, she looks at the 'Invoices' section in Financit, where it clearly indicates how much she will have to pay and why. To do that she just has to open the application and, once she's logged in, navigate to the menu on the top left and select "Invoices". There, she can see all the invoices that she has registered and must pay with the amount and deadline. If she wants to pay it before it's due, she can click on the "pay" button, where she'll have to choose an account to pay it with.

(Maria - send/receive money)



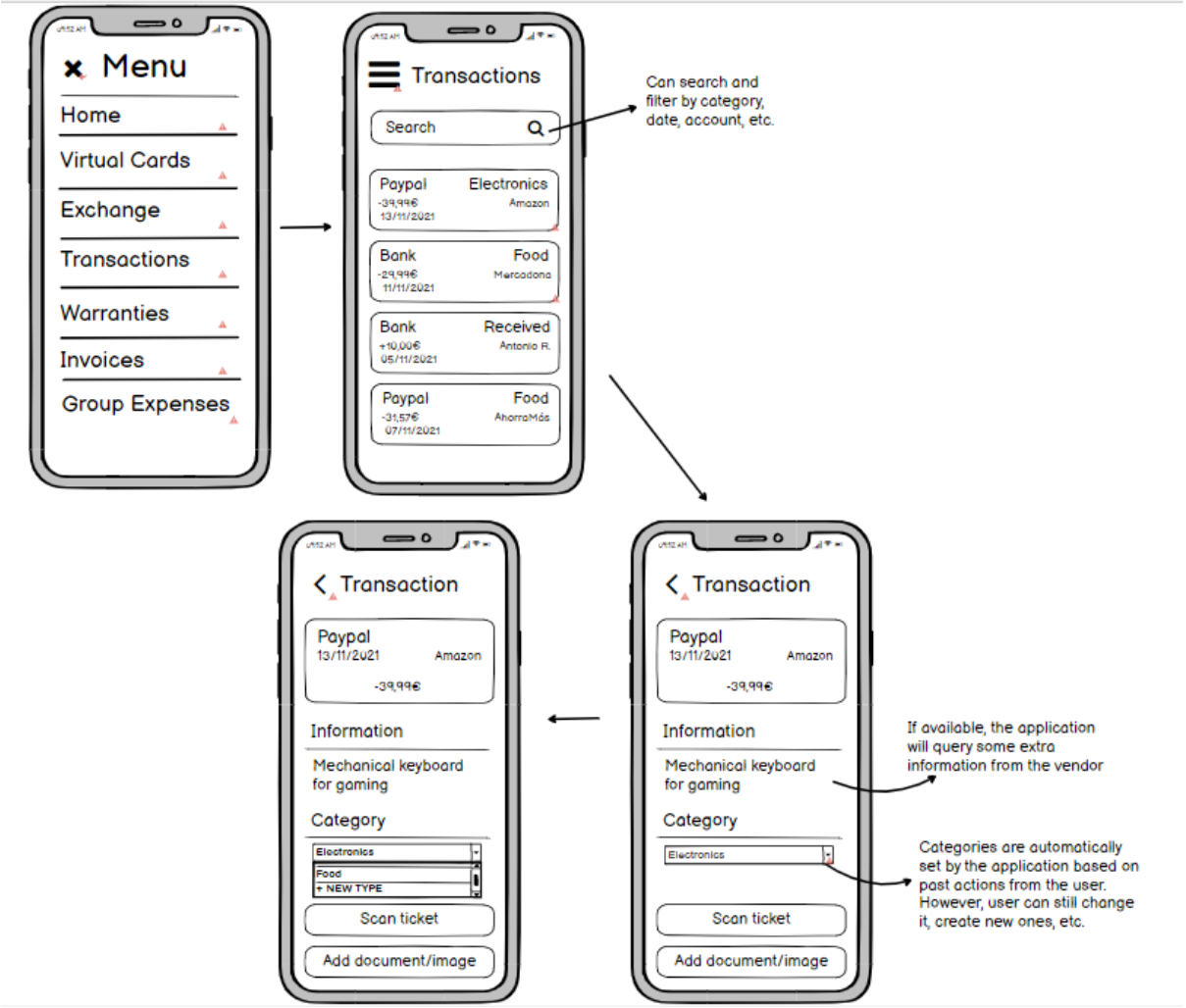
María needs to give her youngest son some money so he can buy himself lunch. Instead of doing it physically, she chooses to send it using Financit. Once logged in, she uses the menu on the top left to go to the ‘Exchange’ tab, where she can either send or receive money. This time, she wants to send her son Juan some money, so, she clicks on “Send” and fills all the needed information such as selecting his contact information, the amount she wants to pay him, with which account as well as a concept. Once everything’s filled out, she can click “Send”, input her PIN and see the check.

(Maria - attach file to transaction)



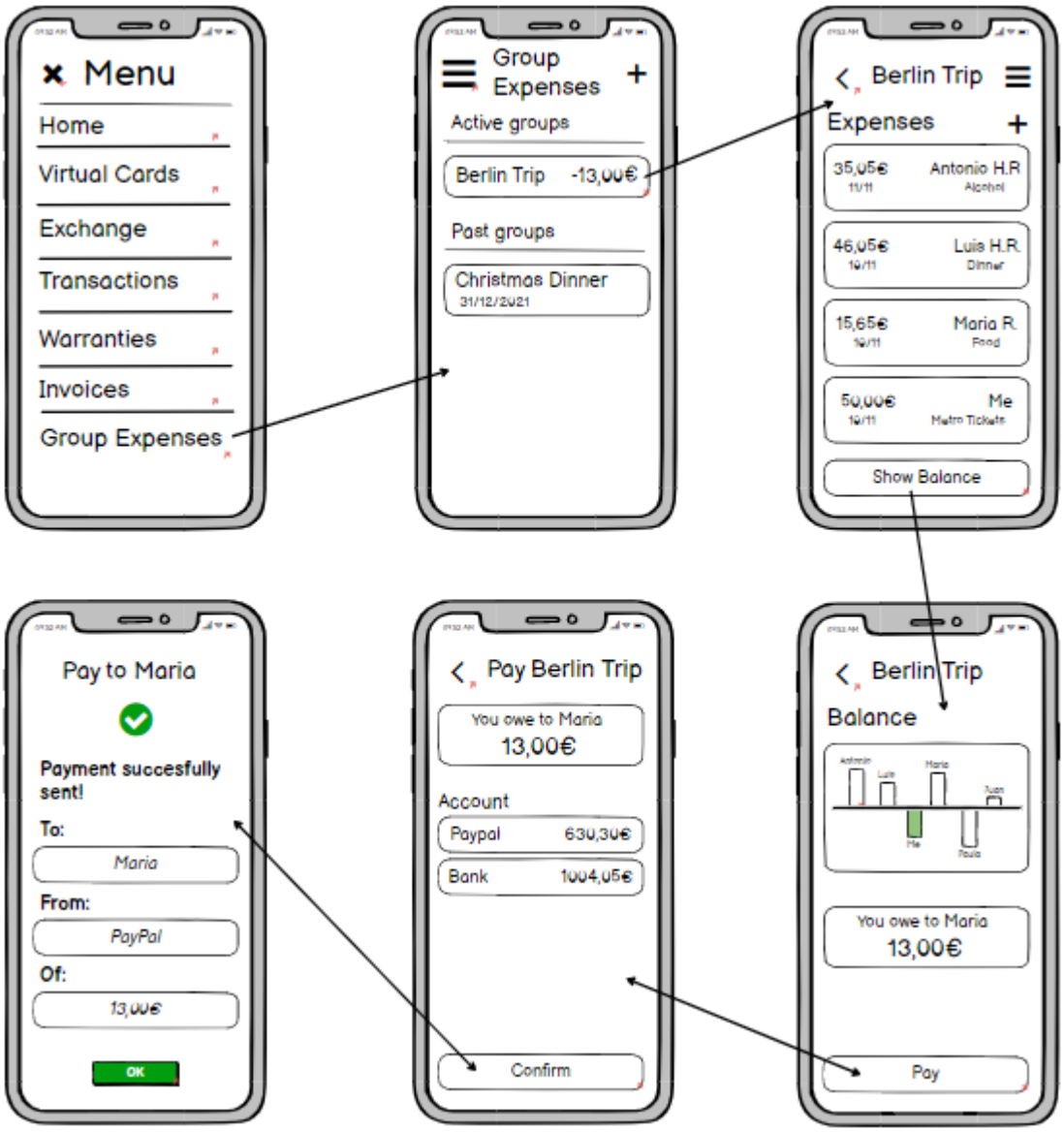
Maria is not a very meticulous person, so when she is reviewing past transactions she sometimes forgets about the context in which that transaction took place. Luckily for her, with our application, she can attach whatever file she wants to any transaction so she can remember what it was. This way, if she goes to a family dinner in which she spends a lot of money, she can take a picture of the ticket and attach it to the transaction, so she can later remember what that transaction exactly was. In order to do it, once she's logged in, she has to open the menu and access the "Transactions" section where all her past transactions are displayed. Here she can look for the one transaction she's searching for. By tapping on it, she can access all the information about the transaction. Below are two options: "Add document/image" and "Scan ticket". The first one is used to attach any type of image that you want to add, the other one is used to scan a recipe that will be processed and digitized by the application before being added to the transaction.

(James - custom expenses for each bank account)



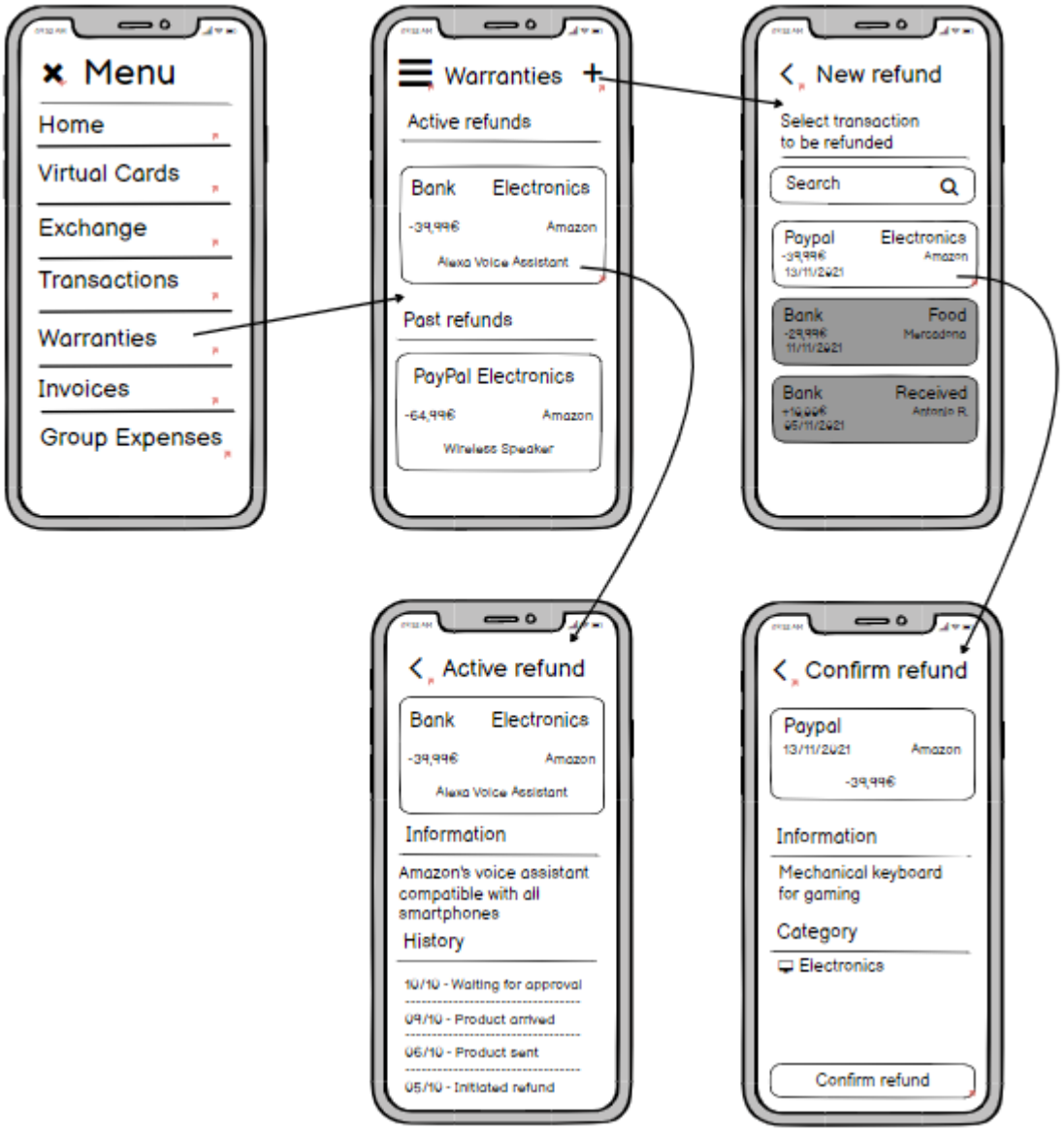
James loves to have everything planned out. During his breaks, he enjoys making sure all his expenses are properly categorized and accounted for. To do that, he goes through the menu to the option “Transactions”, where he clicks on the one he wishes more information about. There, he can see the transaction’s information but more importantly, he can change its expense category by selecting the choice box and either selecting an already existing category or creating a new one.

(James - managing group expenses)



Every last Friday of the month James goes with his friends to their favorite rooftop bar in the center of Madrid to have dinner and catch up. After dinner they drink some beers, but since James has to work the following day, he'll be leaving earlier than his friends. One of the guys tells James he'll pay everything and will put the amount everyone has to pay in the application. The next day James checks the application, through the menu he selects the "Group expenses" where he can see all the groups he's inside and select his friends' one. There, he can see all the recent expenses of the group, as well as how much money he owes his friends, precalculated by the app. To pay off his debt, he goes to "Show balance" where he can see what each person owes, and click on "Pay", where the application prepares the transaction he must make to each of his friends with the selected bank account.

(James - warranty)



James had just bought a new iPhone but when it arrived he noticed that the screen was broken. Luckily he used Financit to buy it and now he just has to open the application, navigate through the menu to the “warranties” section and create a new refund. In the new window he can see all his past transactions and the ones that are non refundable are shown darker. Luckily the iPhone transaction was refundable since he bought it through amazon so he just has to select it and click “confirm refund” and everything else will be managed by the application. Furthermore, once the warranty is requested James will be able to see its status.

Validation scenarios

These validation scenarios have made us define more accurately our application. With each question from below, we were able to find a loophole and resolve it with the scenario that follows.

These are some validation scenarios that we used to improve the design:

What if while a user is paying their group expense debt, someone else adds a new transaction involving the user? Then the debt that the user first agreed to pay would be paid, not involving the new transaction.

What if a user is trying to add a ticket to an expense but for some reason the scan doesn’t recognize the recipe and can’t digitize it from the photo? then the system would ask the user if he wants to add it like a simple photo or if he would try again.

What if a user that isn't bound to any account sends us money? Then the money would be sent to a default account selected by the user.

What if the user is trying to pay an invoice but for any type of error the payment doesn’t arrive at his destination? If at any moment an error happens, the money will not be sent and the invoice will remain pending. A notification will be sent to the user so that they're informed about the invoice's status.

What if one of the user's card expires? The card will be marked as invalid and the user will be notified that from that moment on he won't be able to use the card again. Through the tab “Virtual cards”, the user will have to update the card information.

These scenarios have been looked through and fixed in the final view of the project, in Balsamiq.

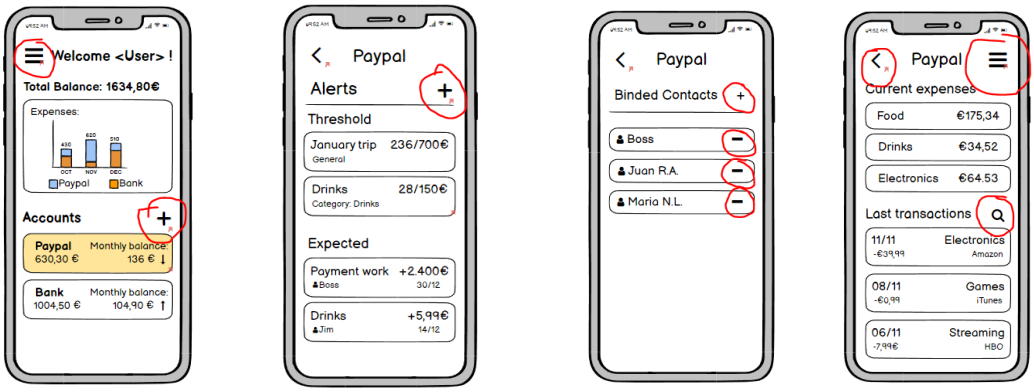
Design principles

And now, we will explain how the design principles seen in class have helped us develop a clear interface.

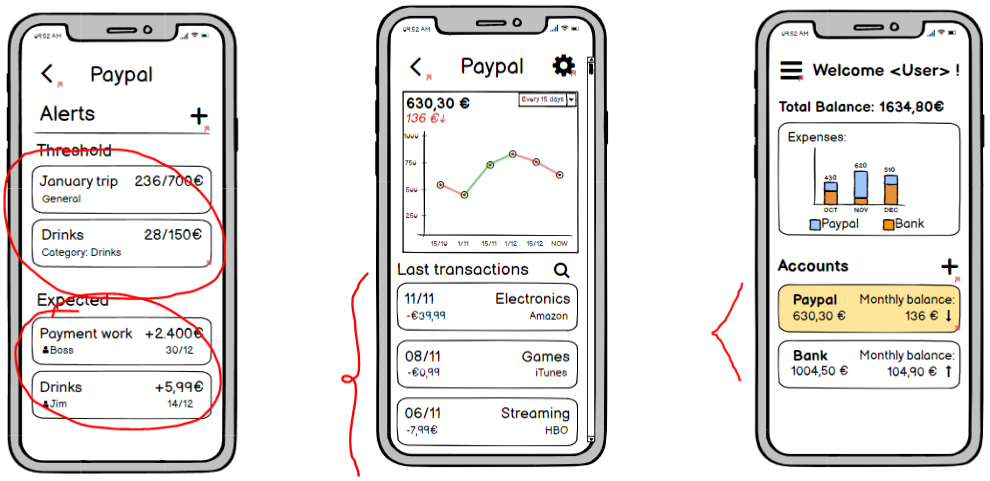
Home Screen

We will start by talking about the first screen seen by users once they login our application: the home screen. This view lets the users take a look at the most recent activity of their accounts. They can also access each individual account and perform actions on them.

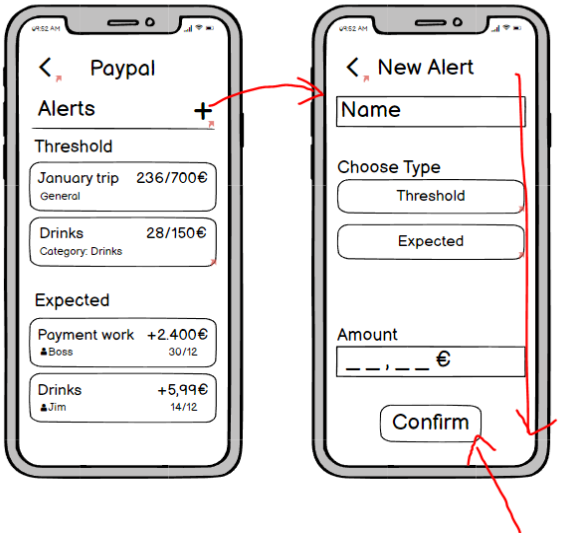
We can appreciate here one of the main principles applied throughout our whole application, the **consistency** principle. We take advantage of the fact that the majority of our users will have already used a lot of mobile applications, so they will already know common ways to navigate them. In order to add new accounts, to add new alerts inside an account or to add new bined contacts, a button with a + symbol is required. This is very common in almost every application nowadays, so users will be familiar with it. Another example of this principle is the usage of an *x* button to close, a - to remove, a magnifying glass to search and filter, three stripes to open the menu, an arrow to the left to go back...Another example of this is the use of a magnifying glass symbol to search for and filter elements in a set.



Another design principle widely used in our interface is the **proximity** principle. We group together elements of the same type, such as accounts, categories, transactions, contacts, alerts, etc. so they are easily identified as the same kind.

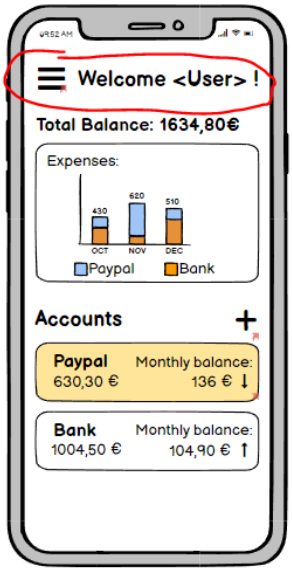


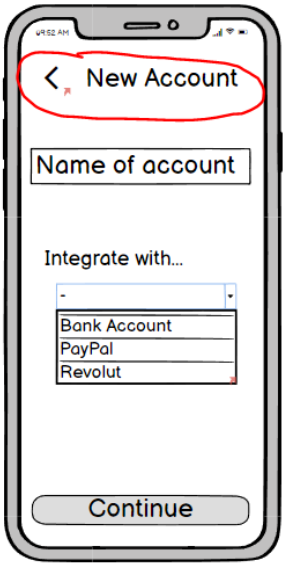
We have also used the **closure** principle in all our screens, since most tasks are done from top to bottom and with very clear actions such as tapping on a button and filling a form, to make them as intuitive as possible. For example, in order to set a new alert, user just have to follow an intuitive and guided process:



Regarding **user control and freedom**, users can add, edit and remove all important elements regarding this screen such as accounts, alerts, binded contacts, etc.

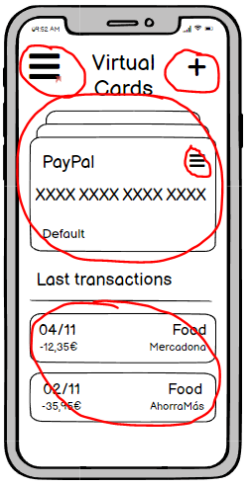
We have also thought about how the users will feel navigating through our application, and we want them to be as comfortable as possible, always knowing where they are. Talking about ***visible state management***, we have implemented an interface in which there is always a title and the main menu button or a go back button in the top left corner. The main menu will only appear for main screens, while the go back menu is for elements inside these main screens. All of them will always be announced in the title. This way, users always know when they are in the main screen of a functionality or taking a look at some element inside it. This is repeated in all the screens of the interface.





Virtual Cards

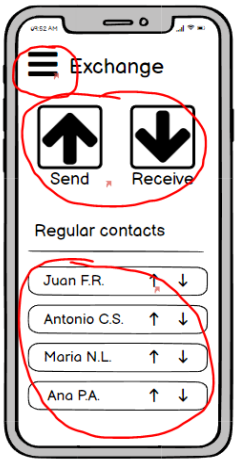
In this screen, we have again applied the **consistency** principle in regular symbols, but also in the way the virtual cards are presented, since they are similar to other virtual card managers such as Google's or Apple's. Again, we grouped elements belonging to a set together, such as the virtual cards and the last transactions made, taking into account the **proximity** principle.

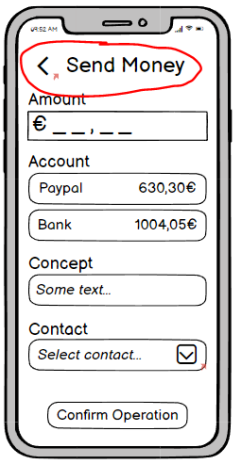


Regarding the **visibility and feedback**, the users can clearly see which card they are using thanks to the selected layout. This was an important aspect since transactions shown here belong to this selected card. Also, users can add, modify and delete each card however they want, providing them **freedom** about their virtual cards.

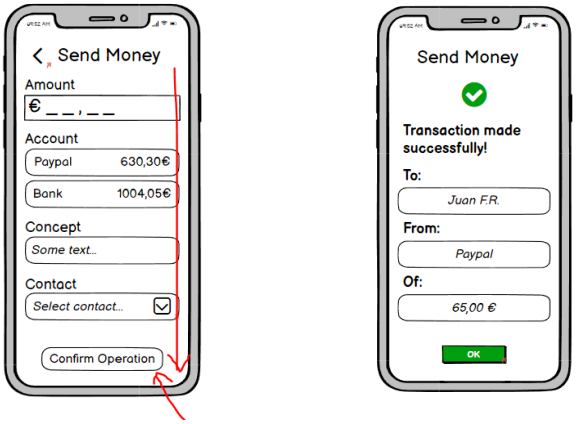
Exchange

Again, we can clearly find the **consistency** and **proximity** principles in the way the elements are presented to the users, so they can understand quickly what is there. As in every other screen, the title and menu button indicate that this is the main screen, whereas in the screens for sending money, there is only a go back button.



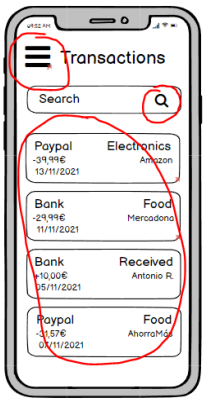


This task is also made so it's easy for the user to follow, since it's done from top to bottom in a very easy and well explained manner. There is also **visibility and feedback** when sending and receiving money, as in any other important action in the application, since there is a screen to let the user know that the process has succeeded.

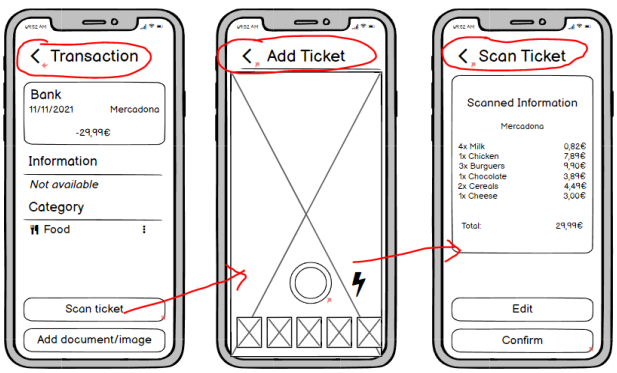


Transactions

Transactions are all grouped together following the **proximity** principle. Again, we also find the **consistency** principle here.
Users can see details of every transaction and once there, scan a ticket or add any other document they want. **Visibility and feedback** was very important here since to scan tickets,



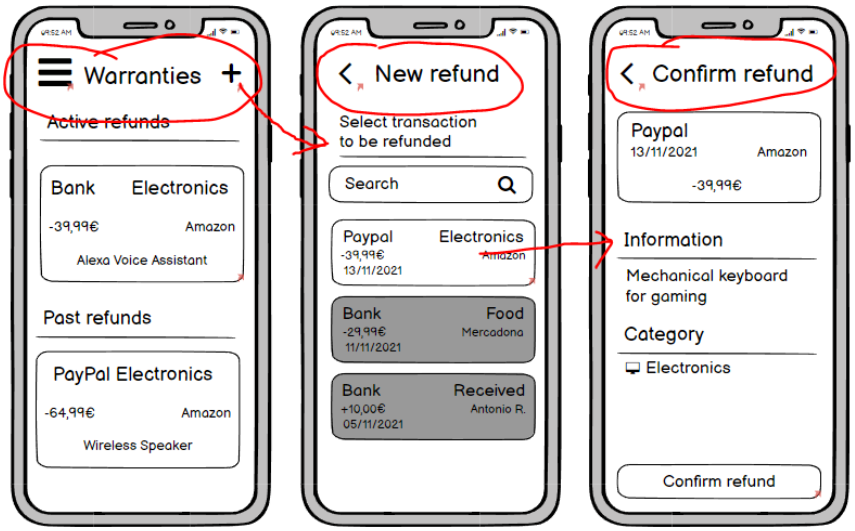
the camera must be used, so the whole interface must change without disrupting the user experience. This was done by showing a camera interface similar to that found in other applications such as WhatsApp or Instagram and maintaining the title and go back button in the top. This way, users recognize that they are still inside the application while feeling familiarized with the camera interface, applying again the **consistency** principle.



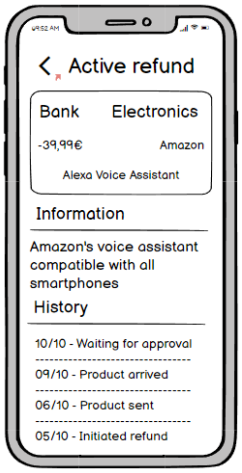
Warranties

Warranties screen is pretty simple. Users can see the state of current and completed refunds and also create new ones. Again, there are still very present both the **consistency** and the **proximity** principles in the whole set of screens providing this functionality.

We also remembered here the **Visibility and feedback** principle when displaying all transactions in order to select the one to refund. Unavailable transactions appear darker and cannot be expanded to see details, this way, users know that they cannot be refunded.

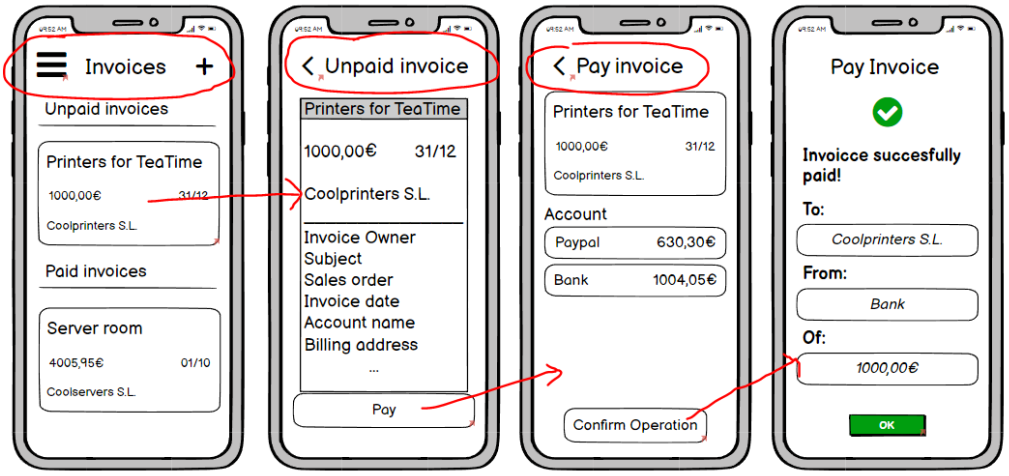


About **visible state management**, users can see at any moment the tracking history as well as other information about their refunds.



Invoices

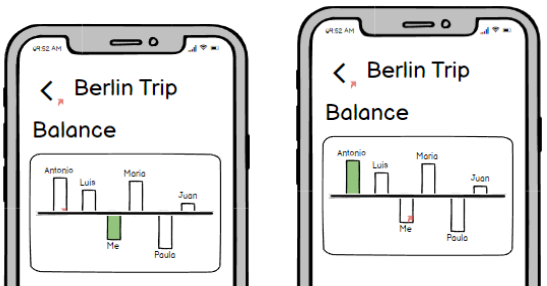
Invoices tab is very similar to the warranties tab: they can see the state of already paid and unpaid invoices and import new ones. Thus, the same **consistency**, **proximity** and **freedom** principles appear here. And also **visibility and feedback** and **visible state management**, since users are prompted with success screens when they successfully pay and they can clearly see all information about their invoices.



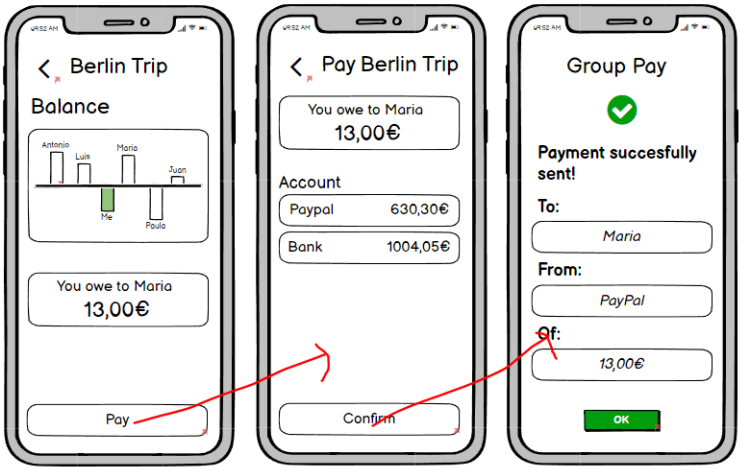
Group Expenses

This screen contains the same principles as the other ones, since the interface is again very similar. However, it includes a functionality to invite other users to a group via a QR code. Most users of our application will be familiarized with QR codes, so it can be easily integrated without any difficulty.

We can note however a good similarity in the balance screen with the virtual cards screen, since depending on the selected user, the information of one or another shown, applying here the **visibility and feedback** principle so it's easier for the users to note whose information is.



The process to pay the debts is very simple. Users just have to select their information and tap on pay, select the account and they will be prompted with a succeed screen if everything has gone well, providing again a very



Conclusion Milestone 3:

This milestone is vital for our application since it not only allowed us to properly design it in Balsamiq, but also get a very specific view of what the application is going to look like thanks to the functional elements and the validation scenarios, which let us find loopholes and fix them up.
We explained our design choices for each functionality in the Design Principles section and showed the most important features our personas would use with the key-path scenarios.

Phase 4: Usability testing

Heuristic evaluation plan

We handed our project along with an evaluation plan to two different groups. The evaluation plan had a description of our application, a script of tasks to be performed by experts on the prototype, a list of heuristics to be used by experts, a severity scale as well as an individual template report for the evaluators.

Brief description of the application to evaluate

Our application is a financial manager for bank accounts and it is aimed to be used from the smartphone.
It includes:

- Multiple account integration in order to summarize the transactions
- Wallet with virtual (credit) cards for each account
- Send and receive money from each account
- Manage expenses inside a group, the expenses inside will be shared accordingly between its members
- Integration of electronic invoices whenever a transaction supports it
- Warranty management of purchased items as long as the transaction allows it

Script of tasks to be performed by experts on the prototype

Log in/making your account/homepage/menu

1. Log into the application by inserting the pin or using the fingerprint
2. Go from the homepage to the menu

Accounts

3. Add new account
4. Check account information (current balance, expense prediction, current expenses, etc.)
5. Remove account

Alerts

6. Add new expected transaction alert
7. Add new threshold alert
8. Edit existing alert
9. Remove alert

Bound contacts

10. Add contacts
11. Remove contacts

Transactions

12. Check transactions
13. Attach a ticket to a transaction
14. Attach another document/picture

Virtual cards

15. Add new card
16. See other cards
17. Change default card
18. Remove card

Send/Receive money

19. Send money to Juan F.R.
20. Request to receive money

Invoices

21. Import new invoice
22. Pay invoice
23. Check invoice information

Warranties

24. Create new refund process (warranties)
25. Check refund status (warranties)
26. Check group expenses (i.e: make new group, check balance)

List of heuristics to be used by experts

N01. Visibility of system status
N02. Match between system and the real world (or use familiar metaphors and language)
N03. User control and freedom
N04. Consistency and standards
N05. Error prevention
N06. Recognition rather than recall
N07. Flexibility and efficiency of use
N08. Aesthetic and minimalist design
N09. Help users recognize, diagnose and recover from errors
N10. Help and documentation

S05. Offer informative feedback
S06. Permit easy reversal of actions
S08. Reduce short-term memory load

Severity Scale

Low
Medium
High
Critical

Individual report template for evaluators

Your name & Group number:		
Comments during analyzing Financit:		
Issues found and Problem:	**Heuristics violated:**	**Severity:**
Problem 1:	Heuristic:	
Problem 2:	Heuristic:	
Problem 3:	Heuristic:	
Problem 4:	Heuristic:	
Problem 5:	Heuristic:	
……	……..	

Expert reviews

After carrying out the Heuristic Evaluation Plan and sending it to our experts, their evaluation of our application gave us insight on how to optimize it.

Group1

We decided to consider some of Group 1's considerations in order to further enhance our prototype.
For example, we made sure to give the user more information on certain screens like what would happen if they sent a PIN recovery email to themselves. We also took care of enhancing the design, changing how some buttons look in order to give the user a more consistent and aesthetically pleasing experience. Moreover, we added more screens, such as how adding a new account to the app would look. Some functionalities seemed to be missing, such as managing your user account and looking over your personal information.

Finally, after fixing a couple of bugs described by the evaluators, we moved on to Group 2's evaluation report.

Group2

The advice and corrections made by Group 2 also helped bring Financit to its finished state. Some of these corrections overlapped with Group 1's, however others were really helpful such as allowing the user to log out, or even delete their user account. We made fixes in order to increase error prevention and user control and freedom, such as adding extra confirmation screens when doing important tasks like managing your account information.

Conclusion of the project:

This last milestone allowed us to clean and fix our prototype from an outside perspective, and bring Financit to its finished state.
By working on this application, we've learned the importance of the process and the different steps that must be carefully carried out in order to plan and design a complex application. Beginning with a competitive analysis and interviewing people about their views of the different applications on the market, then describing the different personas that will represent most of the community that will use the application as well as their requirements, and finally, making the design of the application and performing usability testing in order to optimize and finish it.
This makes us realize the procedure of designing an application is a complex but rewarding one, since the application comes to a closing state, agreed by all members of the team.