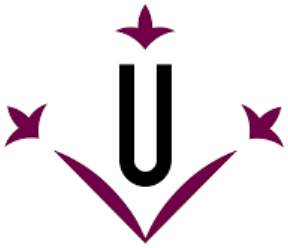

Universitat de Lleida

Sprint 2

Commed

Made by
Oriol Alàs Cercós, Nicolas Arias, Yassine Elkihal, Emina Hanzic
Joaquim Picó Mora, Sergi Simón Balcells

Delivery
17th of November, 2021

Universitat de Lleida
Escola Politècnica Superior
Grau en Enginyeria Informàtica
IT Project Management

Professorate:
Juan Enrique Garrido Navarro
Josep Escribà Garriga

Contents

List of Figures

List of Tables

1 Relevant links

- GitHub backend repository
- GitHub Mobile App repository
- GitHub Web client repository
- GitHub docs
- GitHub project management
- Slides of the presentation
- Spreadsheet documentation

2 Planification

2.1 User Stories

The first thing that was done regarding the planification of the project was to define the behavior of the application in a list of user stories. The next list exposes all of the actions that the user can do with it as well as different ways of interacting with it.

For this sprint, not only it has been changed the name of the user stories to be more precised but also it has been added new ones, related to the formal offer and financial operations. Although the last ones are not implemented, they do exist in set of user stories.

- **AUTH1**: As a guest, I want to register in the application
- **AUTH2**: As a user, I want to log in to the application.
- **AUTH3**: As a registered user, I want to create a profile of my company.
- **PROD1**: As a guest, I want to search for services or products so that I receive a list of services or products.
- **PROD2**: As a guest, I want to have a detailed view of the product/service.
- **PROD3**: As a registered user who has a company profile, I want to create services/products.
- **CHAT1**: As a user, I want to connect to a company because of a publication.
- **CHAT2**: As a user, I need to chat with the company that I connected with.
- **CHAT3**: As a company, I want to chat with the users that have sent a message.

- **FO1**: As a company, I want to send the Formal Offer which contains the contract pdf through the chat.
- **FO2**: As a company, I want to digitally sign contracts.
- **FO3**: As a user, I want to digitally sign contracts.
- **FO4**. As a company, I want to have a list of my sent formal offers.
- **FIN1**: As Commed, I want to get a 5% commission on each contract.
- **FIN2**: As a company, I want to publish the first 3 announcements freely.
- **FIN3**: As a company, I want to have the possibility to pay to promote my announcements.

2.2 Scrum organization and planification

Afterwards, a meeting was held in order to fulfill the product backlog with all the tasks that had to be done. These tasks were related to User Stories, but they were divided so that the product backlog had a small granularity in the given tasks.

According to the kanban, it has been created a milestone named *Sprint 2*, which groups all the issues from this sprint. Also, as in this sprint will be two new main projects added to the former sprint, which will be the web client and the mobile client application, it has been assigned new labels to sort the different types of issues in every repository:

- **wireframe**. The issue is related to the design of the application.
- **web**. The issue is related to the web client.
- **web-dev**. The issue is related to the development of the web.
- **mobile**. The issue is related to the mobile application.
- **mobile-dev**. The issue is related to the development of the mobile application.

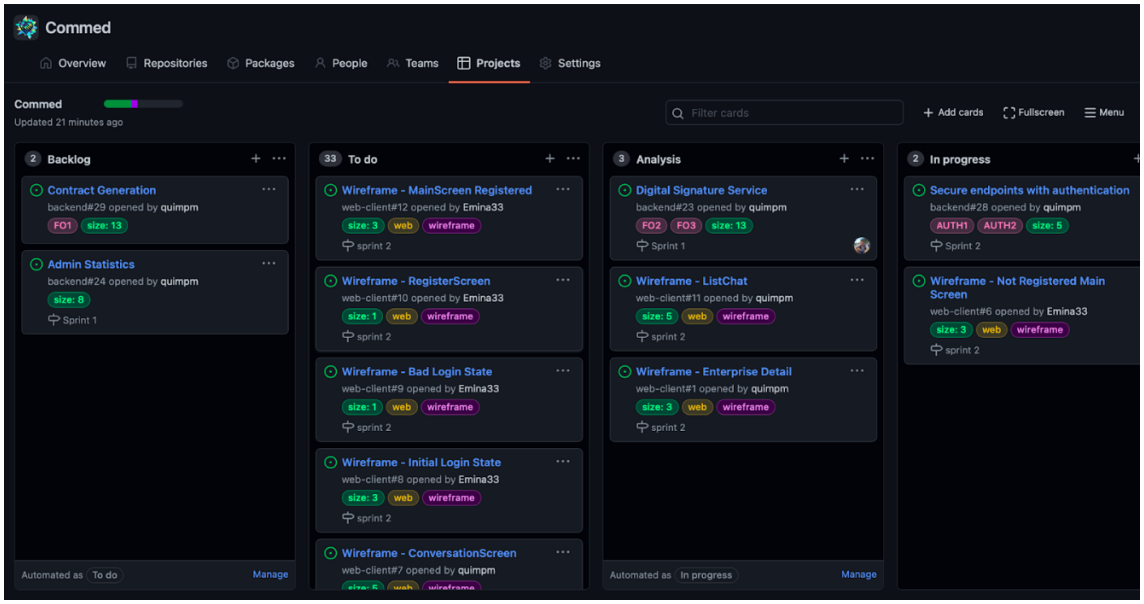

Figure 1: Screenshot of the first days of the project kanban.

2.2.1 Issues

Backend

- List Formal Offers for User
    - size: 3
- List User Products
    - size: 3
- Secure endpoints with authentication
    - size: 5

Web client

- Wireframe - Publication Detail
    - size: 3
- Wireframe - Profile Detail
    - size: 3
- Wireframe - Enterprise Detail
    - size: 3
- Wireframe - ListChat
    - size: 5
- Wireframe - ConversationScreen
    - size: 5
- Wireframe - ListProducts
    - size: 3

- Wireframe - Not Registered Main Screen
    - size: 3
- Wireframe - MainScreen Registered
    - size: 3
- Wireframe - ListFormalOffer
    - size: 3
- Wireframe - Inital Login State
    - size: 3
- Wireframe - Bad Login State
    - size: 3
- Wireframe - RegisterScreen
    - size: 3
- Wireframe - ListFormalOffer
    - size: 3
- Implementation of Register
    - size: 3
- Implementation of Login
    - size: 3
- Implementation of Product Detail
    - size: 5
- Implementation of 404 NotFoundScreen
    - size: 1
- Implementation of the NavBar
    - size: 3
- Implementation of Home
    - size: 5
- Implementation of Enterprise Detail
    - size: 5
- Implementation of Search Products
    - size: 8
- Implementation of Enterprise Edit
    - size: 5
- Implementation of Edit Product
    - size: 8
- Implementation of Create Product
    - size: 3
- Implementation of Chat

    - size: 8
- Implmentation of Formal Offer List
    - size: 3
- Implementation of Formal Offer Form
    - size: 5

Mobile application

- Wireframe Planification - State Diagram
    - size: 5
- Wireframe - RegisterScreen
    - size: 2
- Wireframe - Bad Login State
    - size: 1
- Wireframe - Initial Login State
    - size: 2
- Wireframe - Not Registered Main Screen
    - size: 3
- Wireframe - MainScreen Registered
    - size: 5
- Wireframe - PublicationDetail
    - size: 5
- Implementation of Localization Language
    - size: 2
- Implementation and design of Redux architecture
    - size: 13
- Implementation of Copy static method to create a new State in Redux
    - size: 1
- Implementation of RegisterScreen
    - size: 3
- Implementation of LoginScreen
    - size: 5
- Implementation of GenericSummaryWidget
    - size: 2
- Implementation of CarrousleSliderWidget
    - size: 3
- Implementation of Product/Service Publication Screen
    - size: 5
- Implementation of ListChatScreen

```
    - size: 5
- Implementation of FormalOfferDetailScreen
    - size: 3
- Implementation of ListFormalOfferScreen
    - size: 5
- Implementation of ConversationScreen
    - size: 8
- Implementation of Profile Enterprise Detail Screen
    - size: 3
- Design and Title of the Application - Manifest
    - size: 1
- Implementation of SearcherView
    - size: 5
- Implmentation of Authentication Logic
    - size: 3
- Implementation of MainScreen without registration user
     - size: 5
```

2.3 Scrum analytics

As Github does not support milestones for a group of repositories but for single one, it has been created three different milestones for each project: `backend`, `web-client` and `mobile-app`.

backend

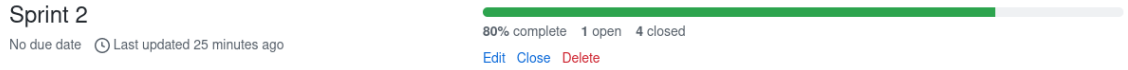

Figure 2: Backend Sprint 2 milestone

As it can be seen, the backend has only little modifications, that is why the percentage can be smaller even though it has been done all the tasks for this sprint. The task that is still missing is the authorization management of the endpoints called, which will be held on the next sprint.

web-client

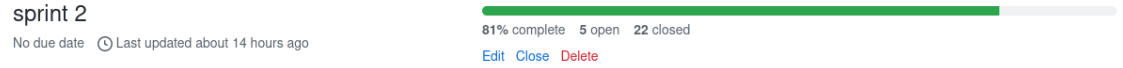

Figure 3: Web client Sprint 2 milestone

In regards of the web client, mostly of the tasks are done. The tasks that are not done are

depending on issues detected in backend, such as image management and chat. This issues will be postponed to the third sprint, where will be created new issues related to the back.

mobile-app

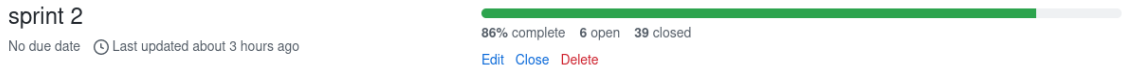

Figure 4: Web client Sprint 2 milestone

According to the mobile application, the things that are still in progress or in to do stage are about the integration of the API backend, which were formerly associated with this sprint. However, while iterating for this sprint, we did realize about that and it was considered that the issues of design and architecture were more important to be finished on this sprint.

3 Requirements

In this section the list of requirements that the application has to offer to the user are on the list below. As matter of fact, the requirements have been more detailed in this sprint, the changes and new requirements are written in *italics*.

Functional Requirements:

- The application has to let all kinds of users search for products or services.
- The application has to let users register into the application *and it will create automatically an enterprise profile.*
- The application has to let users log in to the application if they have an active account on the system.
- The application has to let logged users publish products or services *in the web application.*
- The application has to let logged users interested in either a product or a service to start a chat with the owner of it.
- The application has to let logged users who are owners of a given product to chat with said interested users through a chat.
- The application has to let logged users send a commercial transaction contract when an agreement has been reached .
- *The application has to let logged users download a commercial transaction contract sent by the owner of a product that they are interested.*

- The application has to let logged users sign a commercial transaction contract sent by the owner of a product that they are interested.
- The application has to generate the evidences for both sides of the commercial agreement.
- *The application has to let logged users to view a list of the formal offers they are in.*

Non-Functional Requirements:

- The application has to be the most usable possible.
- The application has to be compliant and respect the laws that run in each country that it's available in.
- The application mustn't have large waiting times for the client.
- The application has to be portable and easy to deploy.
- The application has to be scalable and always leave the code open to the possibility of adding new features in the future.

4 Main use cases

4.1 Sprint 1

- **Register into application**
  - **Actors:** User
  - **Purpose:** Let a user register into the application system
  - **Description:** Provides a screen with a form in which the user is able to fulfill it and send the information to the system in order to be registered.
- **Log In into Application**
  - **Actors:** User
  - **Purpose:** Log in to the application to be able to use some of the application services.
  - **Description:** Provides a screen with a form in which the user will put its email and password. Then, they will log into the application so that they can start using the services that it provides.
- **Search for Products/services**
  - **Actors:** User
  - **Purpose:** Search for any product or service the user is interested in.

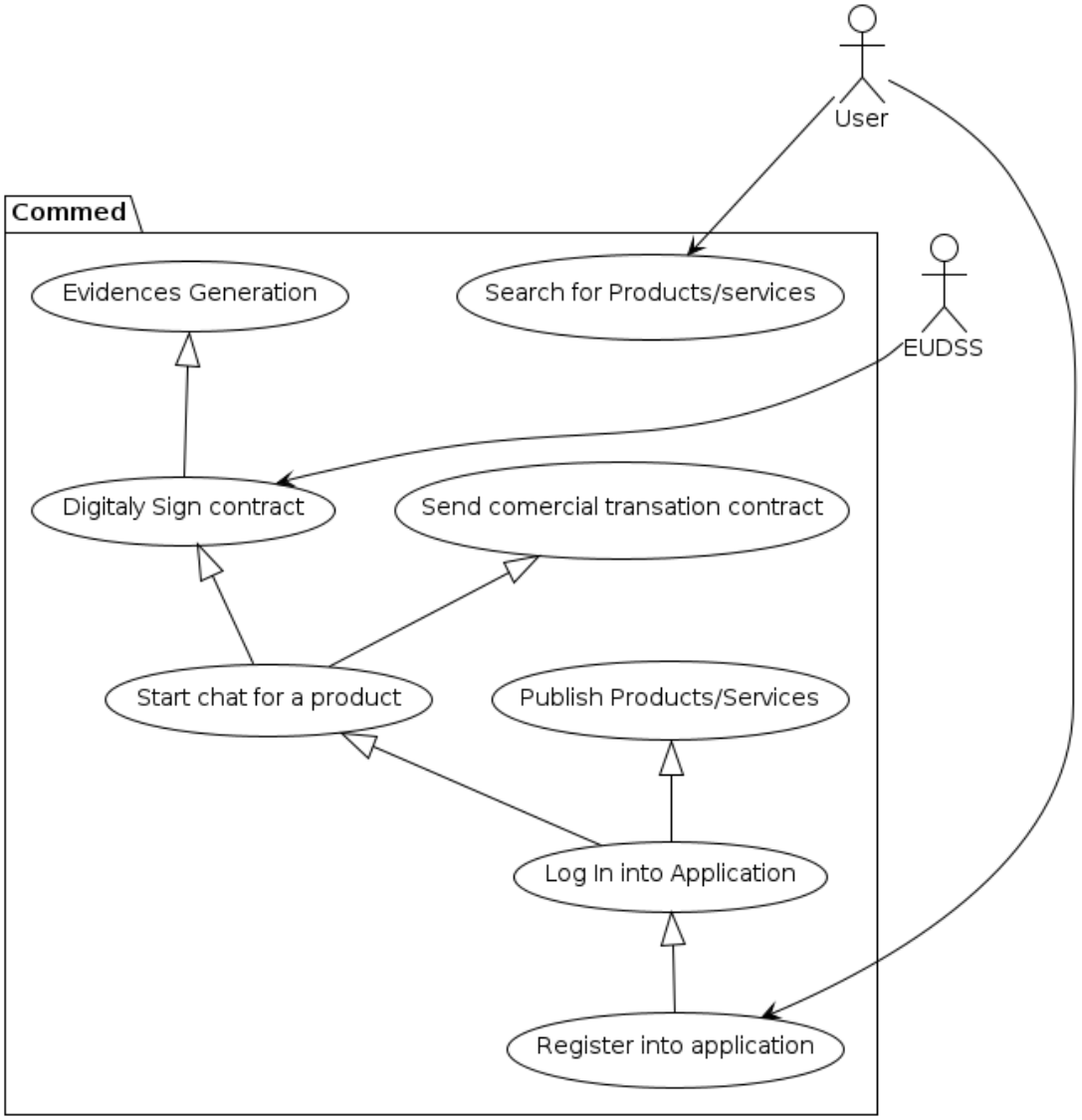

Figure 5: Use case diagram of the application.

  - **Description:** Provides a searcher for every user so that they can look up the products or services that they are interested in.

- **Publish Products/Services**

  - **Actors:** User
  - **Purpose:** Publish services or products in order to be sold to other users.
  - **Description:** Lets a logged user publish the products and services that they offer in order for them to be sold to other interested users.

- **Start chat for a product**

  - **Actors:** User

  - **Purpose:** Users can start a chat when they are interested in a product
  - **Description:** Lets a logged user start a chat with the owners of either a product or a service that they are interested in, so that they can start a negotiation.

- **Send commercial transaction contract**
  - **Actors:** User
  - **Purpose:** Send a formal offer with a commercial transaction contract.
  - **Description:** Lets the owners of given products or services send a formal offer containing a compliant commercial transaction contract within the chat in which the negotiations are taking place.

- **Digitally Sign contract**
  - **Actors:** User EUSSD
  - **Purpose:** Sign a commercial transaction contract sent within a Formal Offer.
  - **Description:** Lets the users of the application sign digitally the contract that was sent as a Formal Offer in the chat in which the negotiations took place.

- **Evidences Generation**
  - **Actors:** User
  - **Purpose:** Provide users with evidences and the billing of a business transaction
  - **Description:** The system will generate for both parts the commercial transaction with all the evidences and the billing of the contract.

4.2 Sprint 2

5 General architecture

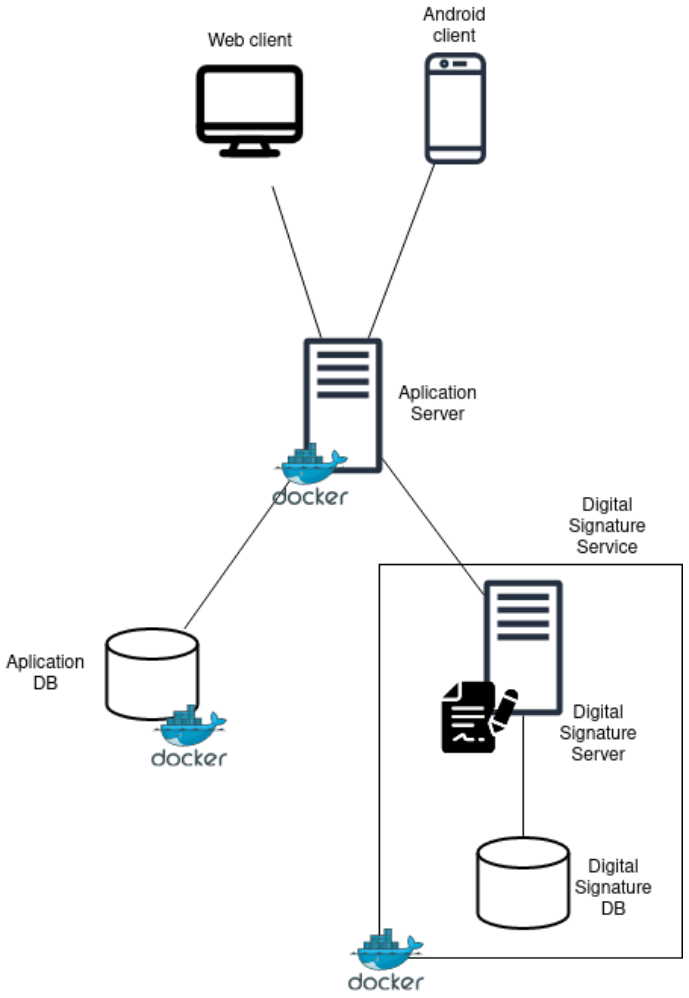

Figure 6: General architecture of the application.

5.1 Mobile app architecture

In this subsection, the decisions regarding the architecture of the mobile app will be explained, but not those that regard the user experience of the mobile. This will be explained in more detail at 7.

First, why we chose flutter and not other frameworks, as Android or React Native, will be explained. Then, it will be introduced the Redux for the state management of the app.

5.1.1 Flutter

Flutter is a multiplatform framework aiming to become a standard for building apps that have to work on either Android or iOS, as well as the web, desktop, and embedded operative systems.

As a tool, it has been proven that it can build more resilient and optimized apps than its counterparts, like React Native, while still being multiplatform. Even though it was taught how to work with native Android Applications, it was decided to choose this tool against it as the team had already some experience with it.

Even if the framework is young, the community behind flutter is active and has several repositories that have some application examples, as well as a libraries showcase that has and will help the development of the app.

5.1.2 Redux

State management in applications is a hot topic, and although both frameworks used by the clients in this project have some way to implement it, it was preferred to use a third-party option that is well consolidated, easy to use, and easy to manage.

Redux is an abstraction that provides a way to implement easier a reactive pattern, that is, instead of having the typical Model-View-Controller, it has a way of drawing a State, the buttons can launch actions, that is taken by reducers along the actual state, and return the next state, as shown at 7.

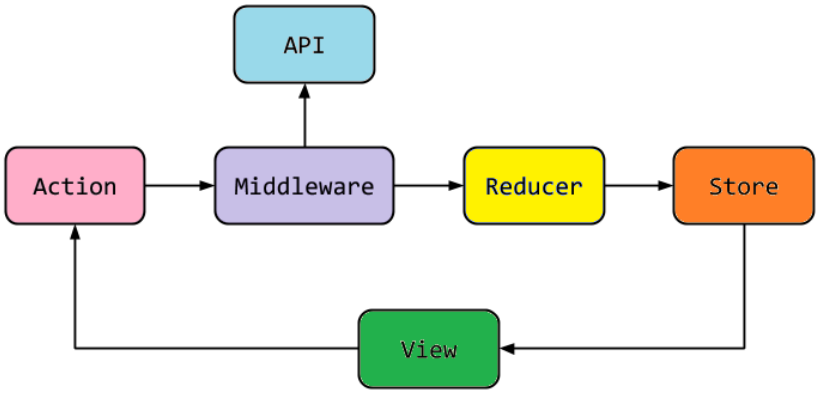

Figure 7: Redux Diagram.

This makes the components stateless, which makes it easier to work with them and provides a

way of mesmerizing them so they aren't computed that many times, making them faster than their counterparts.

This also helps to have a way of monitoring the changes that the state has suffered, as you can keep the states and the actions in a list for debugging, so it can be traced easily the changes. This adds when you have to have a shared object that won't be modified easily, but has to be read by most of the components, as an API token or a theme object. In order to do that, the different functionalities of the mobile application have been splitted and sorted. Therefore, it has been created in every package a `store` and `action` dart files which contain all the logic of the management of the store globally.

Finally, as it was also used in the web applications, it makes the team be able to change quickly between the two frameworks, as the most difficult part is done the same way.

5.2 Reason on implementing a Web Client

As the application has its target user in the administrative world, it was decided that a web will be created alongside the mobile application that will make the usage of our services easier to our customers in administrative environments. It has been decided that the mobile app can be really useful for doing little tramits, for chatting with some customers, for quick searches of products/services and enterprises...
But it has been felt that in order to make it more usable and accessible for administrative porposes a web client should also be created.

Furthermore, the implementation of the web client shined some light on how to communicate with the backend and has lead us to iterate a little bit on the API endpoints that were implemented on the last sprint. This is nice because now on the third sprint the integration of the mobile app with the backend will be a lot more straightforward.

5.3 React

For implementing this Web Client React has been used, which is a library built in javascript for creating User Interfaces. To be more precise React was used because it gives the opportunity of working with Hooks. Hooks are functions that comprehend inside them inmutable components with independent states, wich makes the code a lot more clean and structured and simplifies the amount of chaos that has to be dealt with usually when it's programmed with vanilla javascript.

Also React has a huge community, and it has a lot of libraries that have been key for making a lot

more easier the programming task. Some examples can be:

- react-router-doom
- react-modal
- redux-react-session
- react-bootstrap
- ...

5.4 Web Client Architecture

By now This Web Client is still in a development phase, so for now the default testing server that React provides to us is being used. Although, a nginx+docker is expected to be used when deploying our application to a production enviroment.

Right now the application is able to fully connect with the backend, beeing able to Register, SignUp and interacting with all the different features that our application provides.

5.5 Redux

In this part of the project, redux has also been used. In fact a react library called redux-react-session that builds a store to maintain the state of our session is being used. Also it creates the correspondig Session Cookie to maintain the value of the authorization token provided by our backend. This way, every time a request that needs authorization has to be made, it will be possible to pass the value of the token stored in the browser cookie within the request .

6 Database model

The database model can be seen at figure 8.

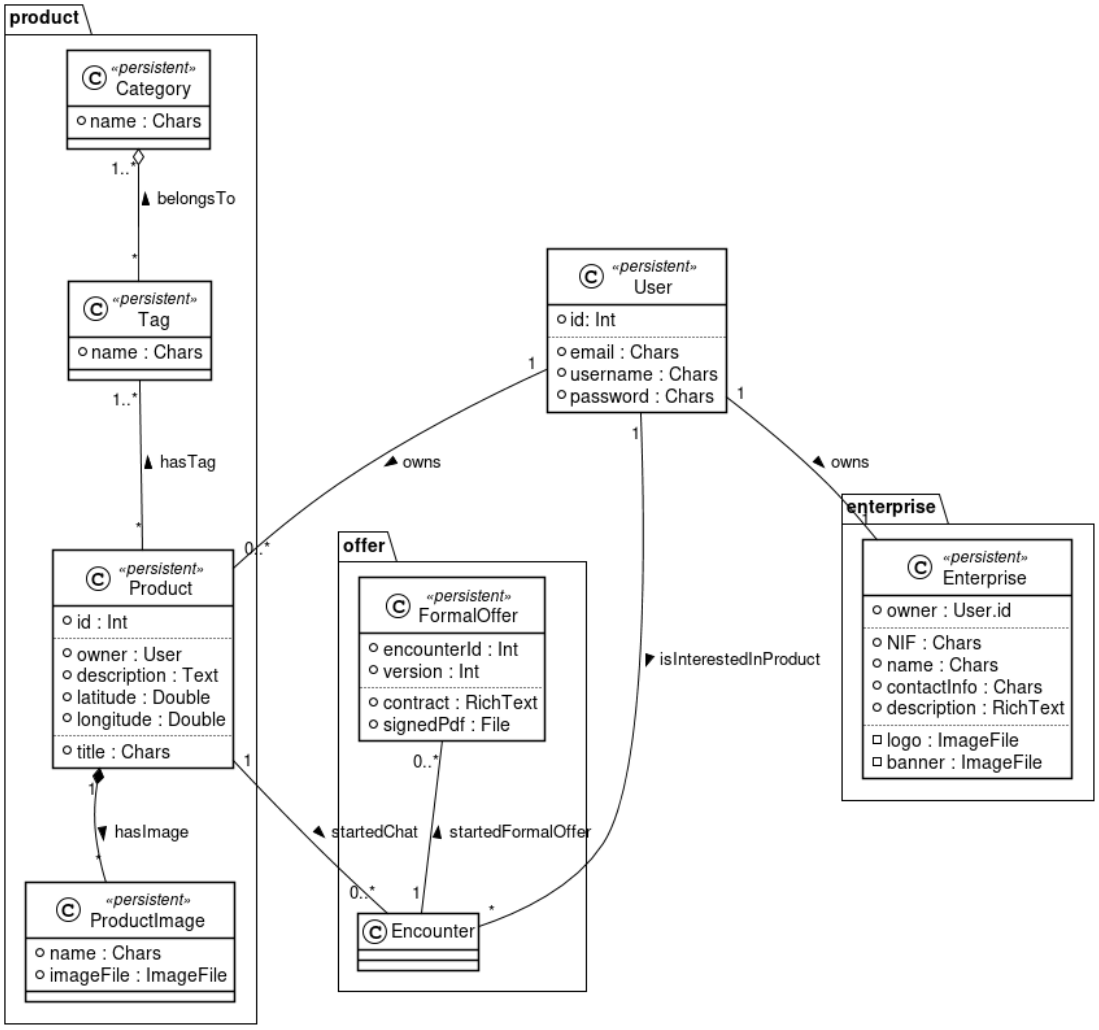

Figure 8: UML diagram of the models.

6.1 Modifications on the models

During the implementation of the clients, it was found the lack of the title of the product. As it was discovered in early ages of the sprint, it has been created an issue for it. Also, it was found the lack of an image that would represent the logo of the enterprise and an another one which would correspond to the banner.

7 Main Screens

In the first stage of the designing of the mobile application, it was held a state diagram to start representing the management and the navigation of the user. This diagram can be shown in 7.

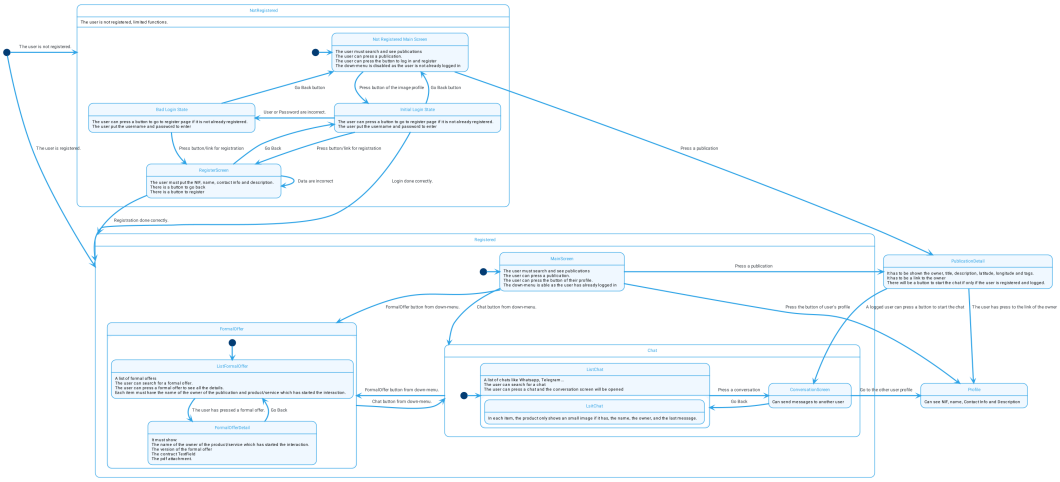

Figure 9: State diagram of the application.

7.1 Mobile Application

7.1.1 Wireframe

It can be seen in the next images the main screens of the wireframe. At first, it can be seen a list of product as the main page of the application, in the image 10. The next image 11 contains the login screen, which can be accessed by clicking on the user icon the image before. The register 12 is accessed by clicking at the sign-up option.

The image 13 contains the list of chats that you currently have for every product or service, and the 14 has the detail of a particular chat, that can be seen when clicking in a item from the list of chats. The next image contains the list of formal offers 15. Both lists can be accessed when clicking on the icon in the bottom menu if the application.

The last two images contains the detail of an enterprise 16 and a submenu for things related to the user 17. The first image can be accessed when clicking the logo of an enterprise, and the last one can be seen when clicking on the user logo.

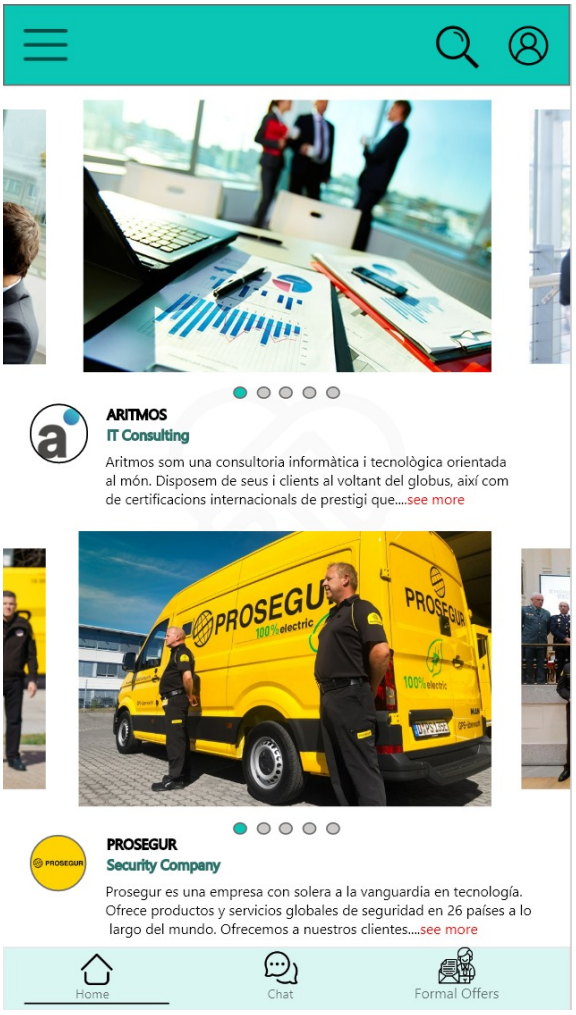

Figure 10: Home mobile screen, it lists the products

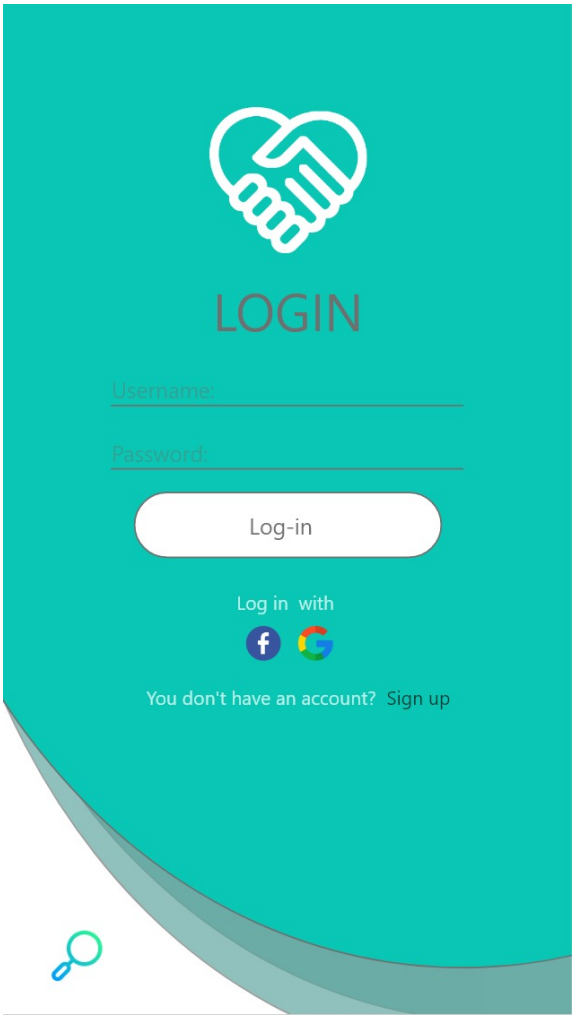

Figure 11: Login view

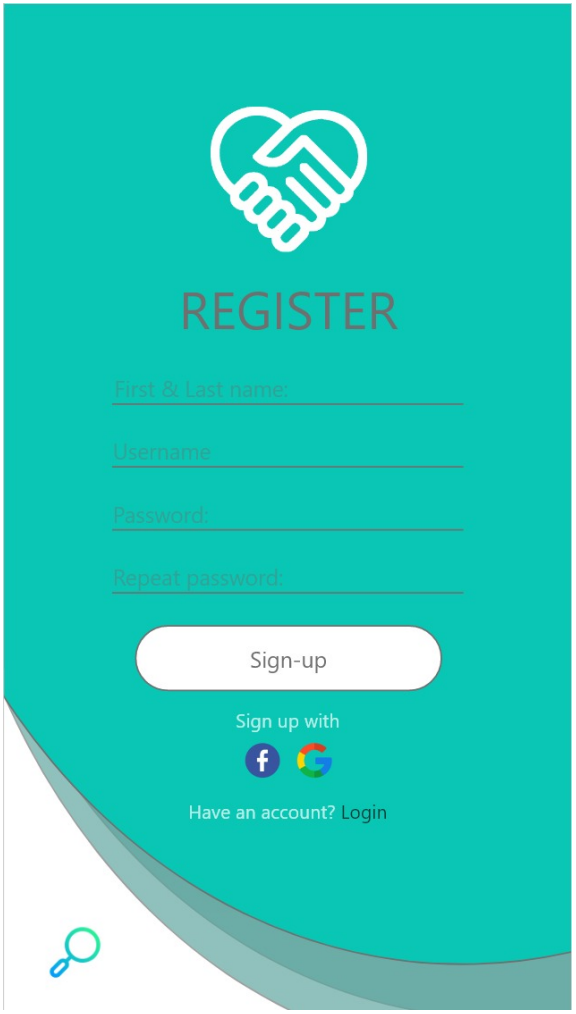

Figure 12: Register view

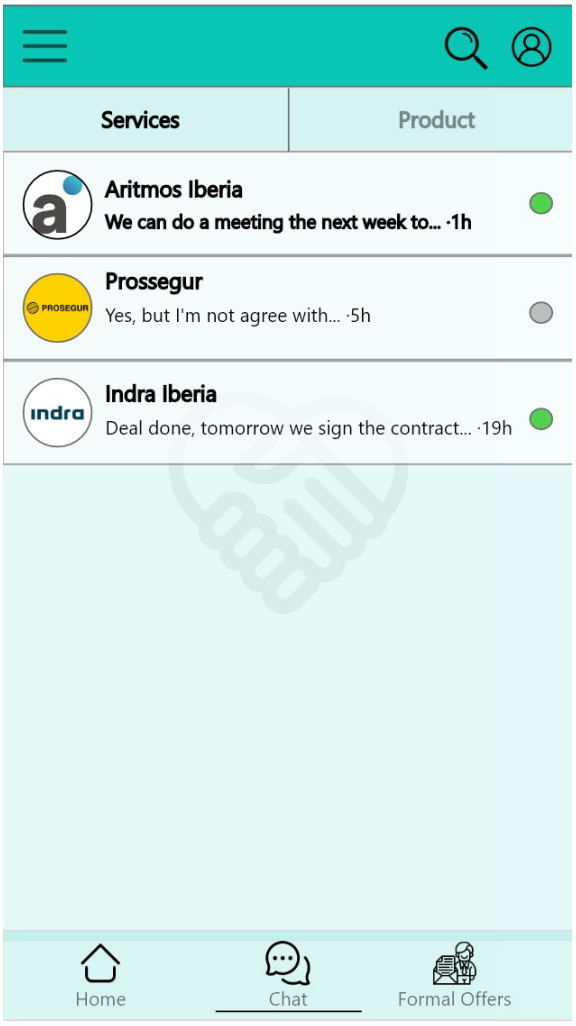

Figure 13: List of the current chats.

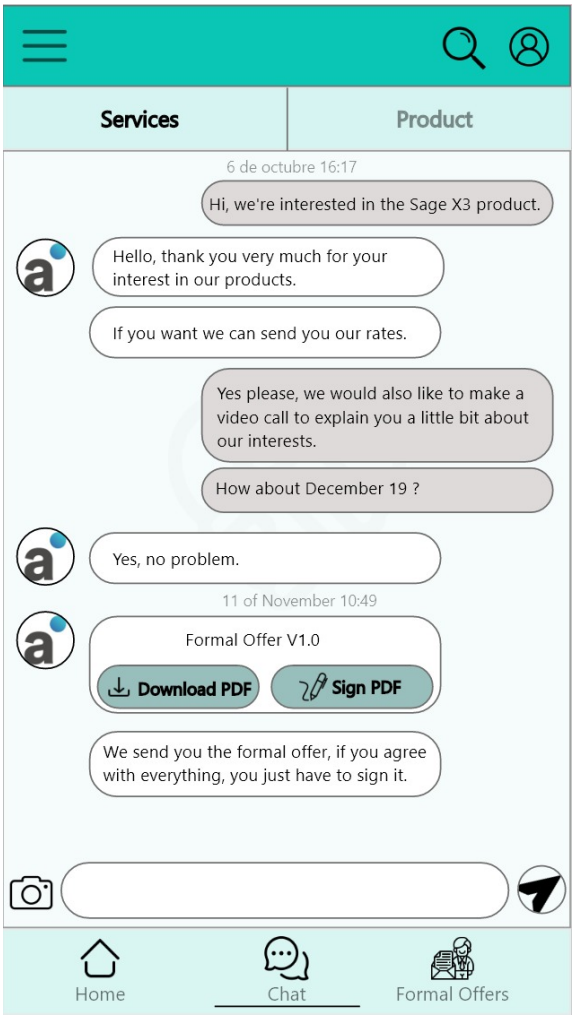

Figure 14: A chat conversation.

Figure 15: List of formal offers.

Figure 16: An enterprise profile.

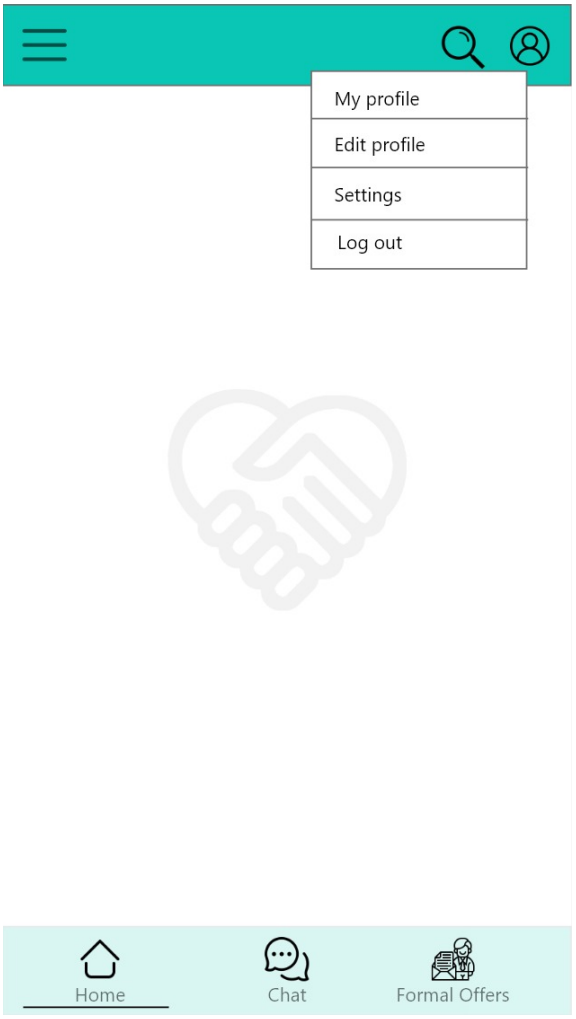

Figure 17: Menu Profile.

7.1.2 Application

In this section, you will encounter all the screens that the application has currently. The first three images correspond to the home, which lists some products, the list of chats, and the list of formal offers. This can be accessed when clicking at the bottom view. It can not be slides, as it was found confusing when using the images' carousel.

The next page 21 is the profile of an enterprise, which is triggered when clicking a logo of the enterprise, in the pages before. In the figure 22, the chat between two users can be seen. It also can be seen a formal offer version, and although a formal offer version can only be send by the owner of the product, it was though that showcasing both items in the mock would be better. The next image 23 is the searcher of the app, that can be triggered when clicking the magnifying glass icon. It has the history of the items searched, which can be deleted when clicking the trash button, or replicated when clicking the text.

Finally, in the images 24, 25, it is shown the registration and the login screens.

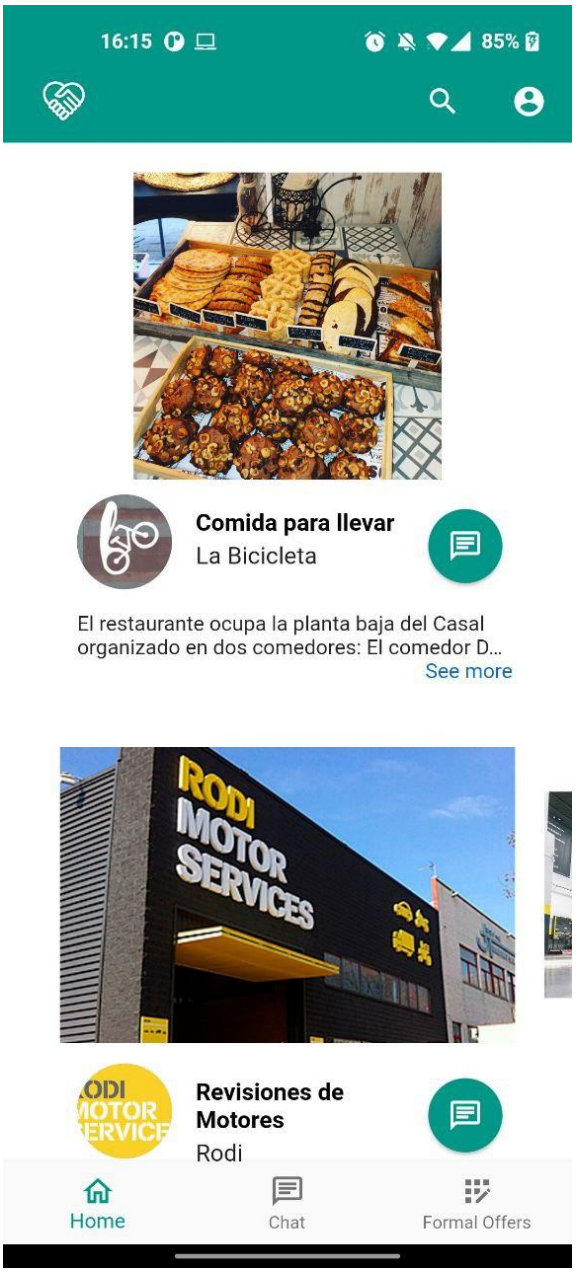

Figure 18: Home mobile screen, it lists the products

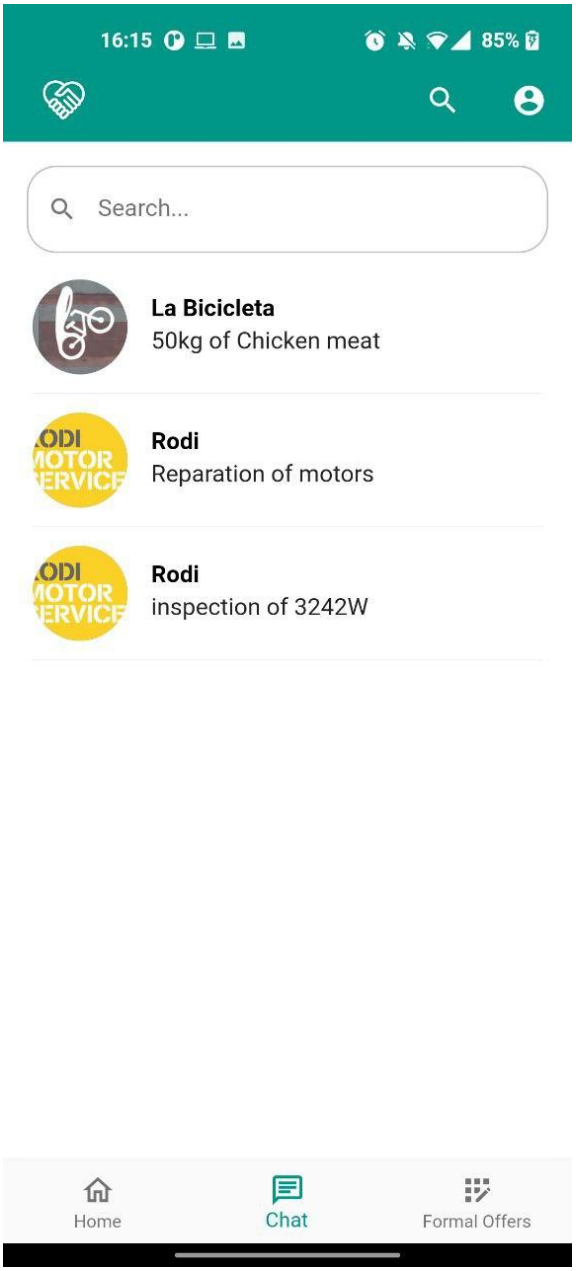

Figure 19: List of the current chats.

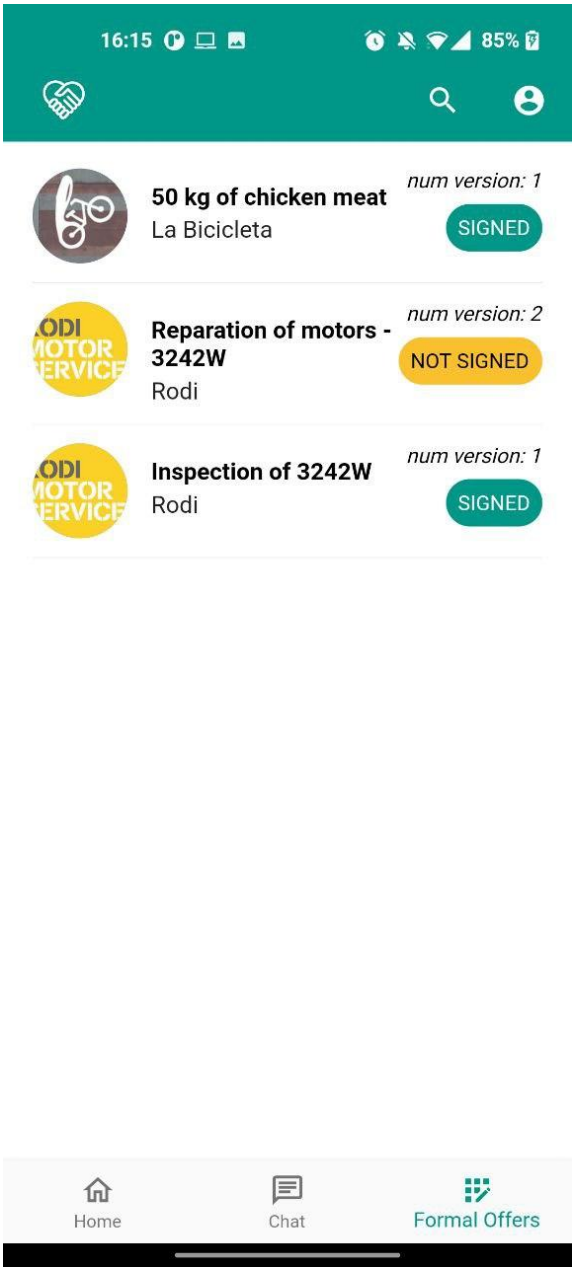

Figure 20: List of the current formal offers.

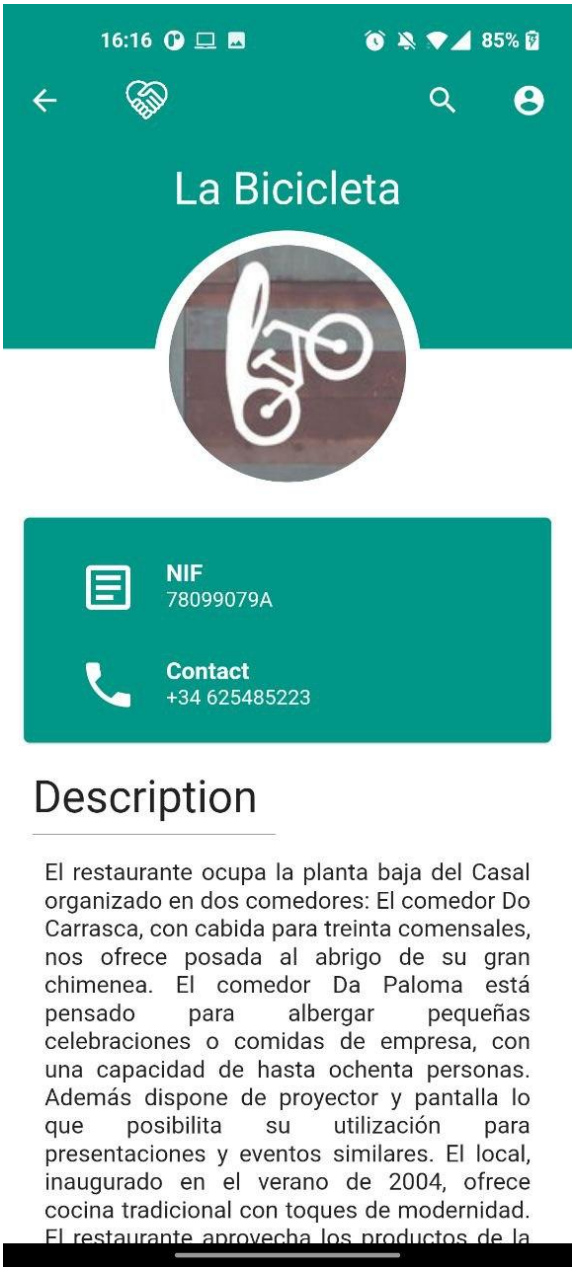

Figure 21: Detail of an enterprise, it can be found when clicking the logo of it.

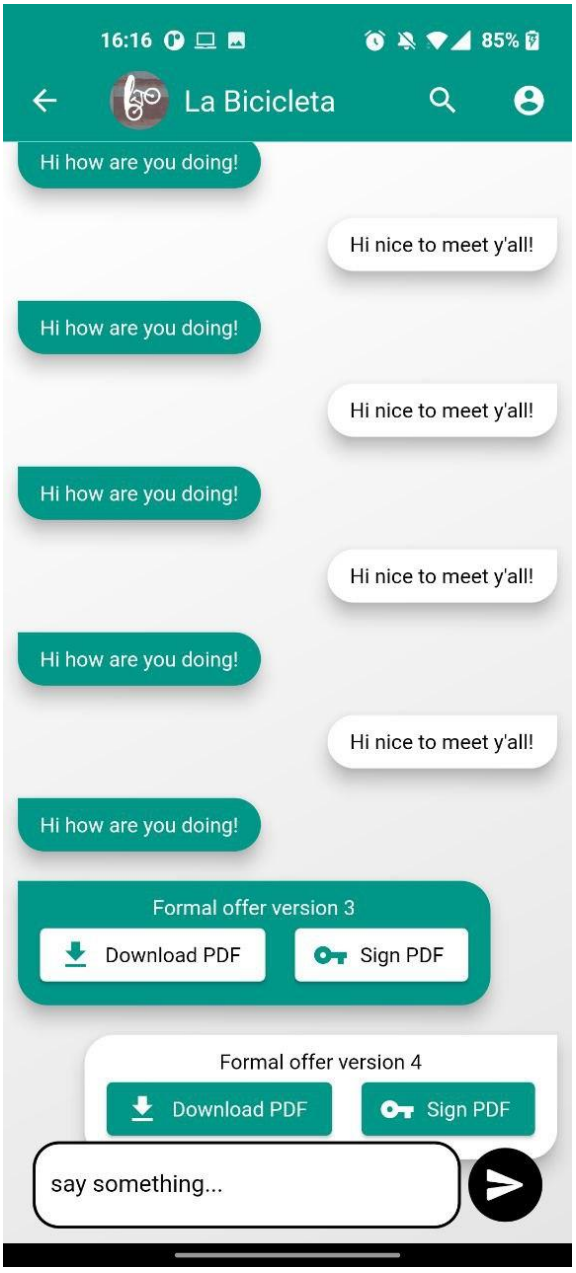

Figure 22: Chat screen.

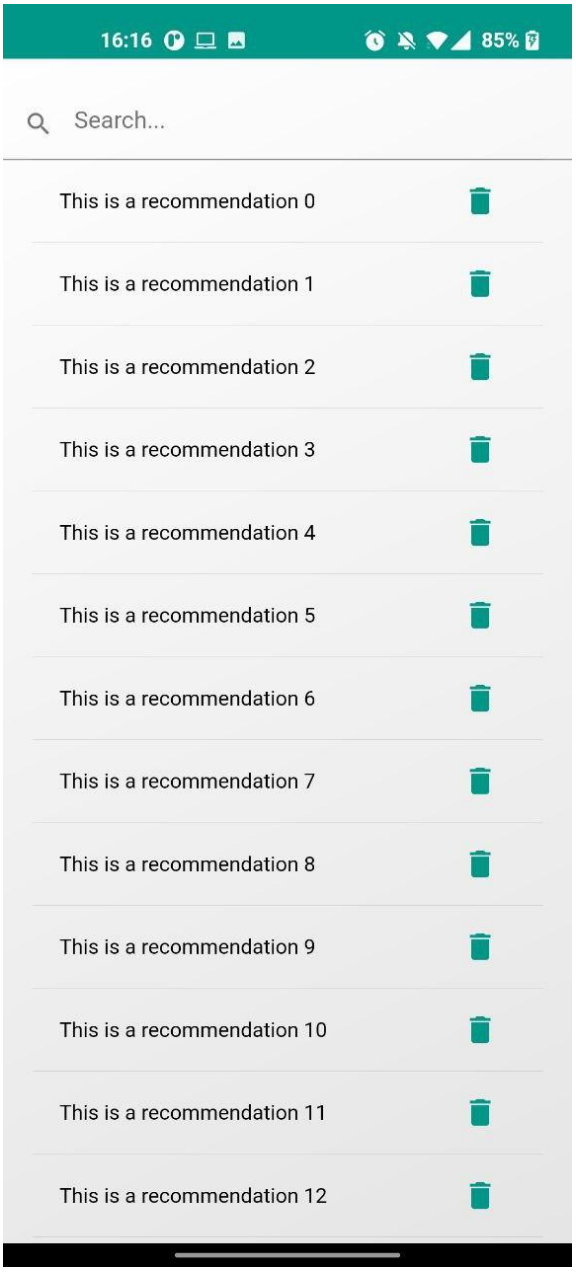

Figure 23: Recommendations screen.

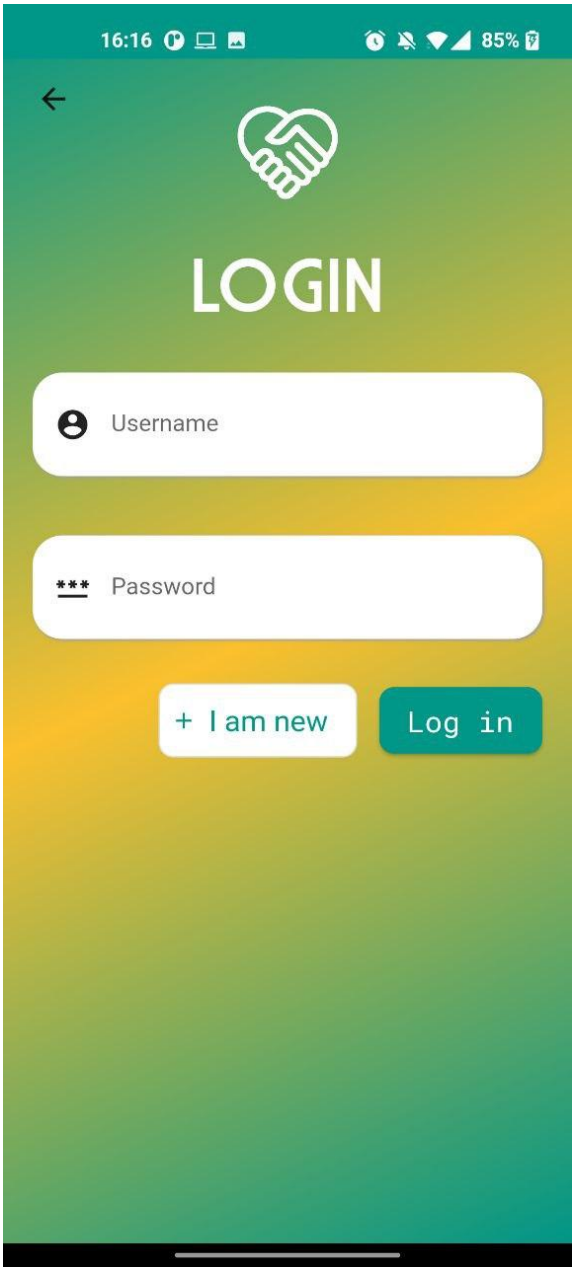

Figure 24: Login view.

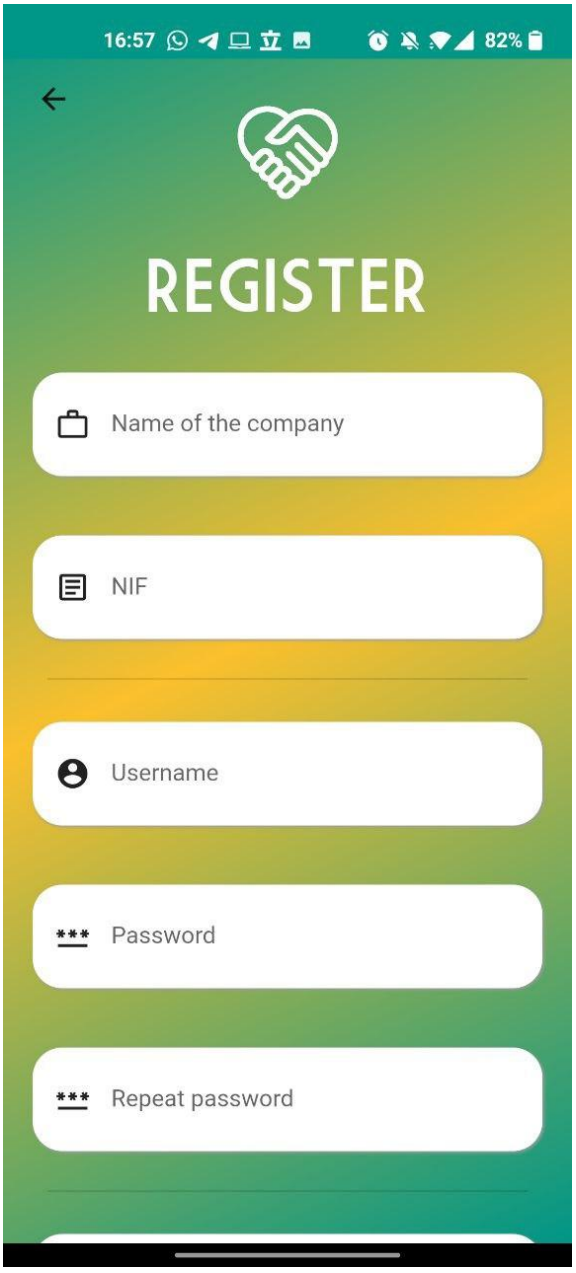

Figure 25: Registration screen.

7.2 Web Client

7.2.1 Wireframe

In this section, it is going to be shown the paper wireframe that was drawn. This contains the first ideas that had been thought for the dessign of the web application.

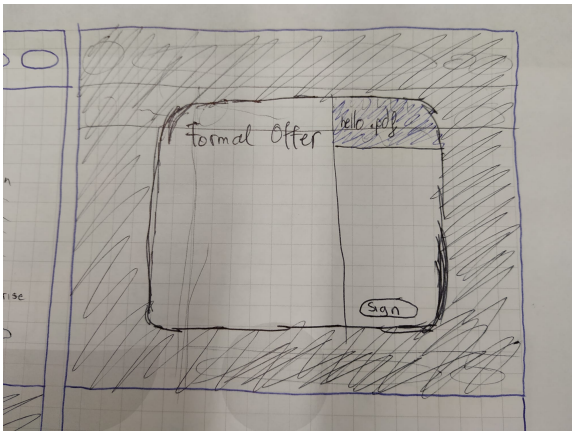

Figure 26: FormalOffer Detail Sign screen.

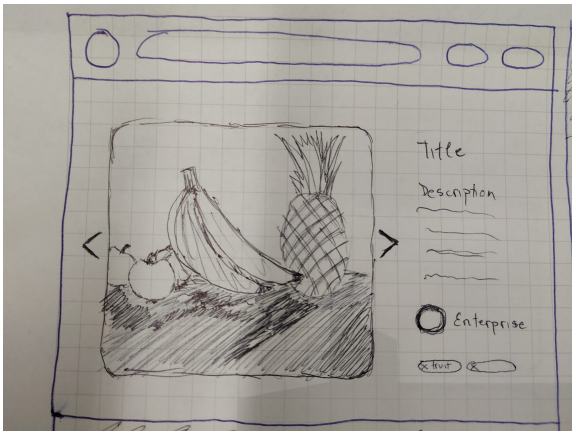

Figure 27: Product Detail screen.

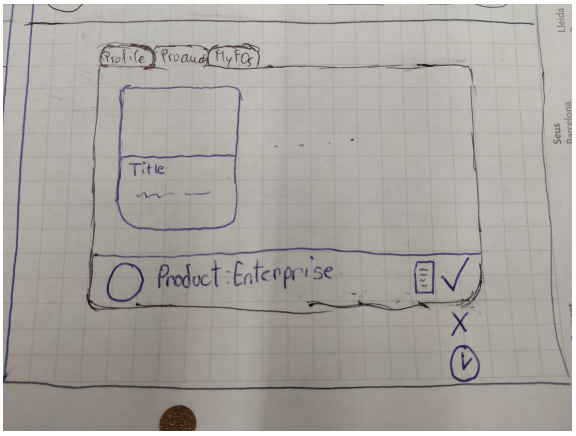

Figure 28: Products/Formal Offers Profile screen.

Figure 29: Chat screen.

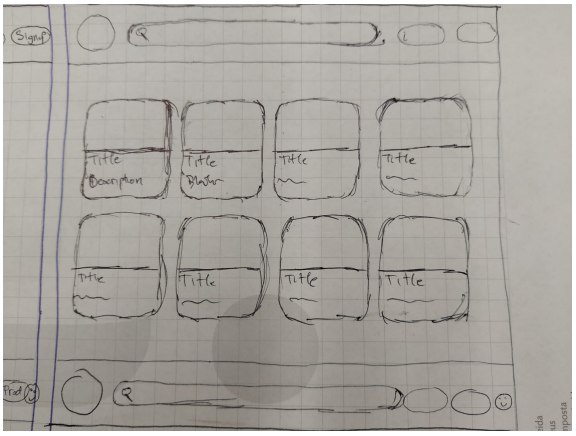

Figure 30: Search screen.

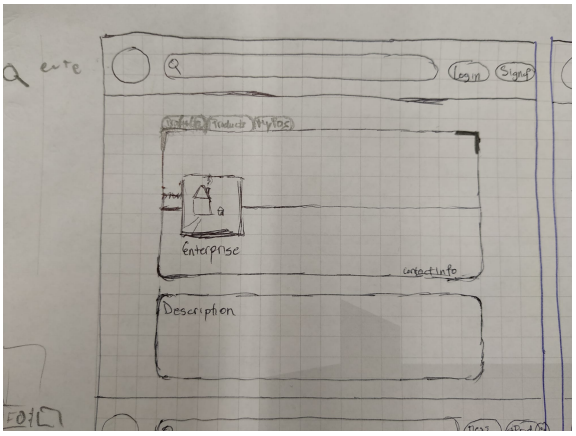

Figure 31: Profile screen.

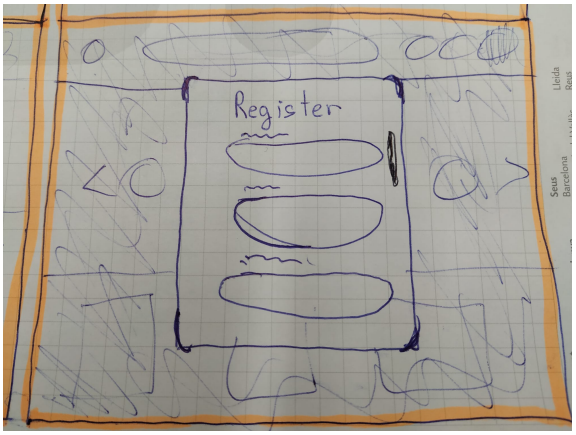

Figure 32: Register screen.

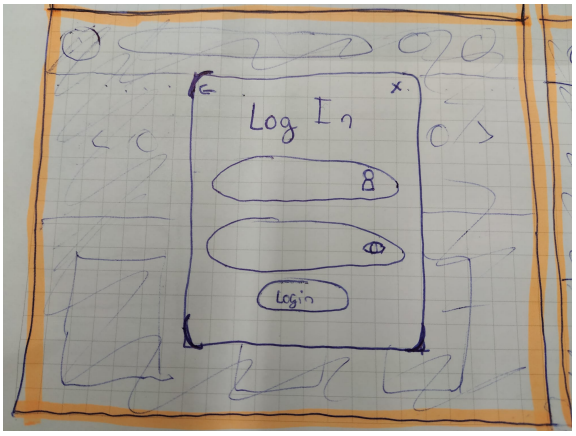

Figure 33: Log In screen.

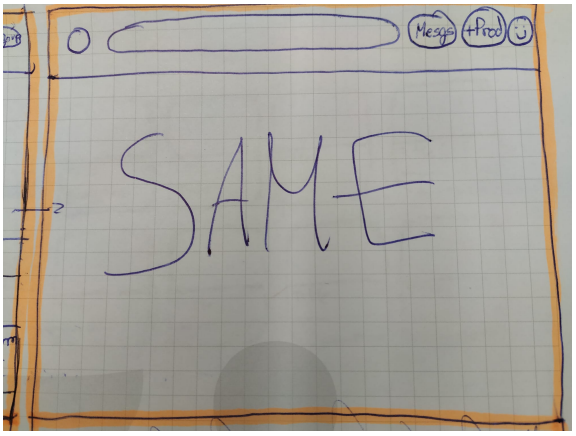

Figure 34: Home Loged In screen.

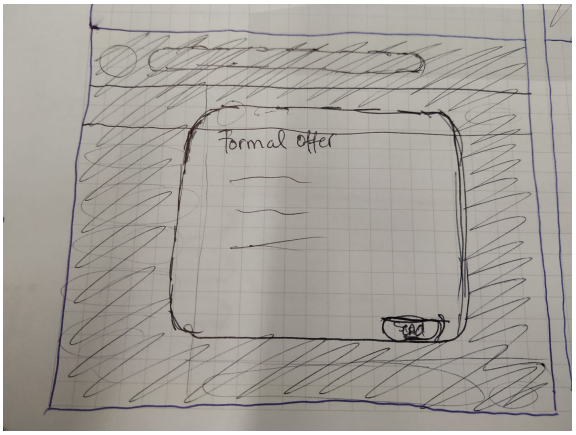

Figure 35: Formal Offer Send screen.

Figure 36: Home screen.

7.2.2 Application

In this section we are going to show all the Screens that are actually in our application.

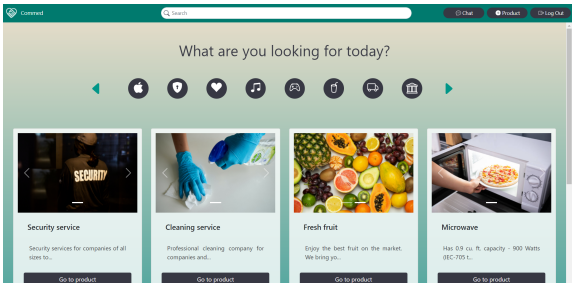

Figure 37: Home screen.

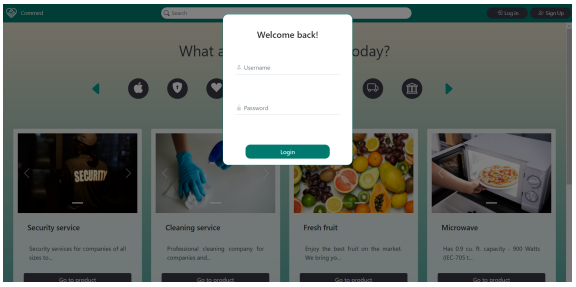

Figure 38: Log In screen.

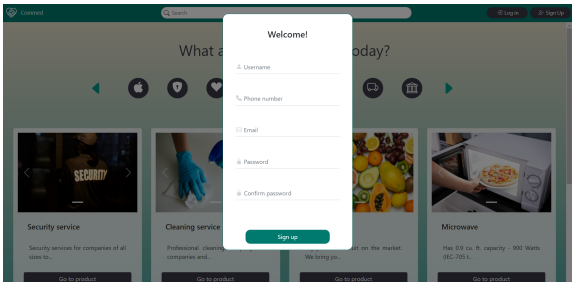

Figure 39: Registration screen.

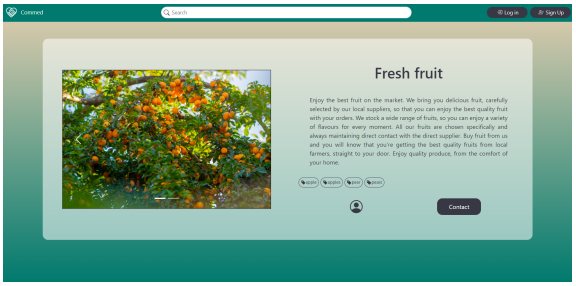

Figure 40: Product Details screen.

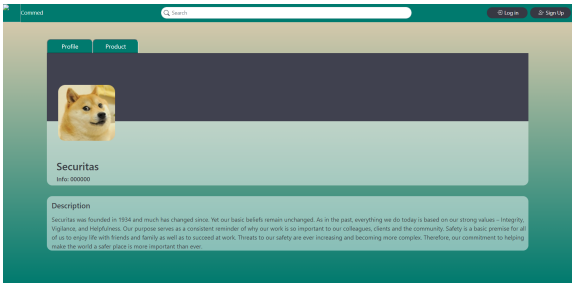

Figure 41: Profile screen.

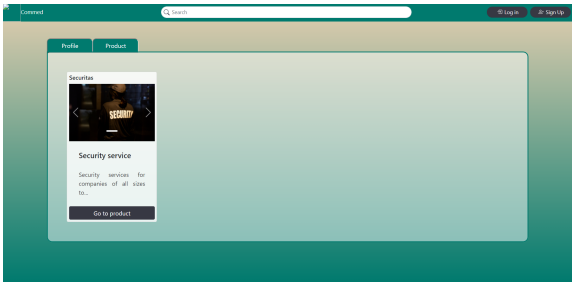

Figure 42: Profile Products screen.

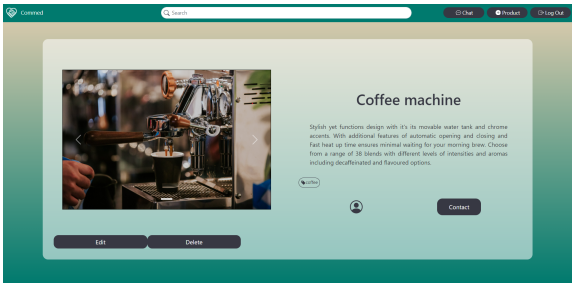

Figure 43: Profile Products Loged In screen.

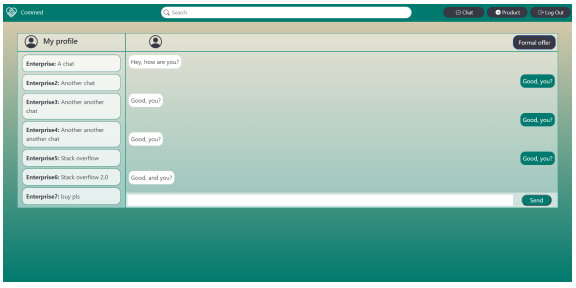

Figure 44: Chat screen.

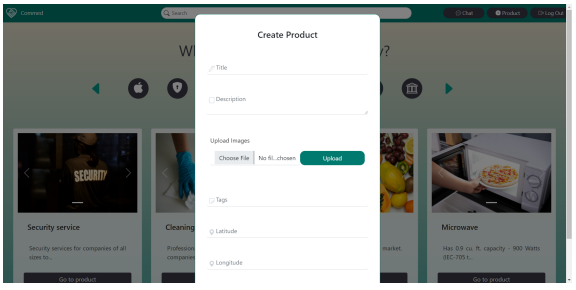

Figure 45: Create Product screen.

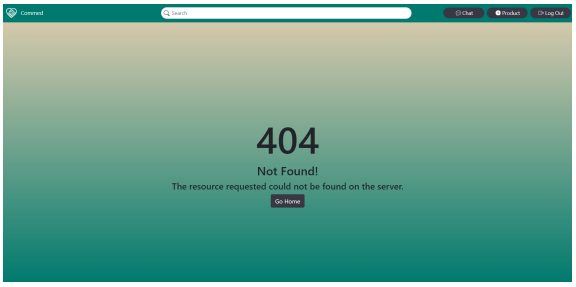

Figure 46: 404 screen.

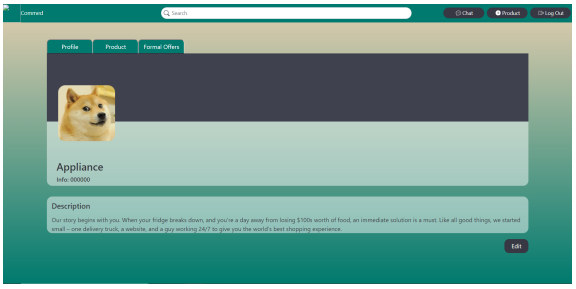

Figure 47: Edit screen.

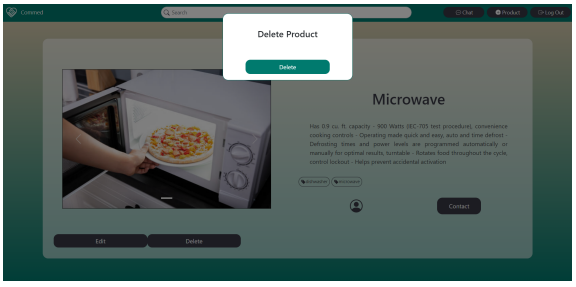

Figure 48: Profile Loged In screen.

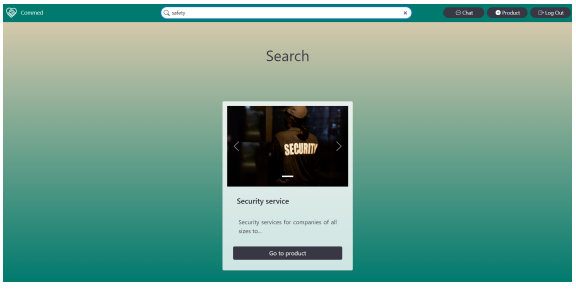

Figure 49: Delete Product screen.

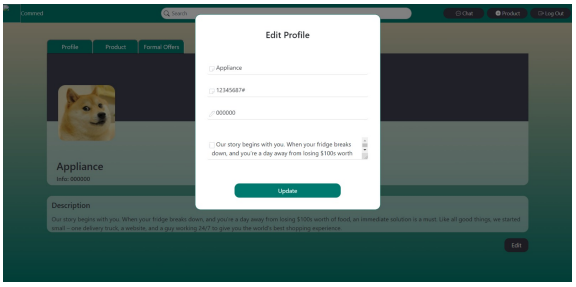

Figure 50: Search screen.

8 Dailies

8.1 01-11-2021

- Creation of the state diagram of the Mobile Application.
- Accorded that Jass was going to be the responsible of doing the Wireframe of the application.
- Accorded that Quim and Emina are working on the React Web Client
- Accorded that Sergi and Oriol are working with the Flutter Mobile App

8.2 02-11-2021

- Talked a little bit about the style of the application
- Emina and Quim started With Wireframe
- Mobile Aplication Team started also with the design and the Wireframe of the application.

8.3 03-11-2021

- Decided the palette of colors we are using oin our app and mobile app
- Decided the logos that we are using in our web client and App
- Put In commont design decissions from both subteams.
- Talked about the work to be done in the rest of the week

8.4 08-11-2021

- Update of the status of the project after the week of work
- Quim and emina showed wireframe of the app and irst steps of the Web app.
- Jass showed to us some screens of the wireframe, and opened some design discussion.
- Oriol and sergi showed botom navigation menu, list of products and started.

8.5 09-11-2021

- Jass keep the work with wireframe and started to learn some flutter.
- Oriol and keep with the work and start login and register.
- Quim and emina start with the arquitecture of the web.

8.6 10-11-2021

- Oriol and sergi keep working with register and login
- Jass working with wireframe and flutter
- Quim and Emina srat doing Home screen and comunicate with backend
- Set the amout of work to do for nexct week

8.7 22-11-2021

- Update of the current state od the project
- Oriol and Sergi showed some Mobile App Screens
- Quim and emina Showed some screens of the client + Session managment and login/register communicating with with backend

8.8 23-11-2021

- Sergi and Oriol Working with More mocked screens of the mobile App
- Quim and Emina also working in some more Screens of the web application

8.9 24-11-2021

- Quim and Emina put on the table dessign questions that had been debated by all
- Sergi and Ori Showed to all the group the progression of the app screens and also leaded on some design discussion.

9 Financial Case

9.1 Monetization Strategy

The monetization strategy of Commed is mainly based on a percentage of the value of the commercial agreement contracts. When a contract has been finally set, an invoice will be generated. The value of the invoices depends on the value of the generated contract in order to make the invoices

affordable for all types and sizes of companies. Therefore, in the first four years, the search of the companies and the other features will be free. After these beginning years, it is something to study and plan if some other features can become freemium.

In order to get more publications in the first few years and popularize our service quickly, the first three publications will be free for any company. The next ones will cost 250 € per year, which will not depend on the size of the company publishers. In addition, any kind of publication could get a better position in the search by paying monthly. The fee of this kind of advertising will be up to 450 € per month and the position will depend on how much the company pays for that advertising.

Therefore, there are three possible incomes:

- 5% commission of each contract generated.
- Publish a 4^{th}, 5^{th},..., n^{th} offer cost 250 € per new offer each year.
  Publishers have 3 free publications.
- Payments for advertising by month, the payment will be chosen by the customer in the same way it is done on instagram and facebook. The more the company pays for the advertising, the more the offer will appear. The maximum fee should be 450 € per month.

As the search will be free, the companies could be able to negotiate and set the contract out of our reach. In order to sort this inconvenience out, Commed will finally generate the contract making this process so easy and the invoice very affordable as the payment is only the 5% of the contract.

9.2 Marketing Strategy

According to the marketing strategy, it has been set that the main focus in the first years will be getting the small and medium sized companies, in order to quickly popularize the platform and create small business ecosystems. For example, the main ecosystems to focus on in the first year will be in order of importance:

- Restoration, butchers and other food providers.
- Food industry and supermarkets.
- Organizations for seasonal work to get temporary contracts.
- Cleaning services.
- Security services.
- Logistic services.

After creating the small ecosystem, the focus will be on strengthening the ecosystem and widening the smaller ones in order to join with the others and make them become a greater one.

This Sprint, we decide to be more specific about our Marketing Strategy. We choose to operate a door-to-door strategy to convince our first clients. Our sales mans will convince them by proposing the pros of using our service, and state our welcome offer. As for the advertising channels, we have chosen Linkedin, because we are in a B2B market, so our target are companies. As things progress, we rely on word of mouth to acquire more customers, both publishers and service seekers.

9.3 Speculation and flow chart

In order to create the simulation of the revenue, a speculation has been done on how many contracts will be held by our application in a month. This conjecture has been simulated over the first four years of the company. The revenue functions consists on these three variables:

- The income from the percentage of the contract between the entities. This variable has the main weight of the function as it is the main income generator. Mean value for small business contracts : 1000€. Mean value for big business contracts : 4000€.
- The revenue from the paid publications. In order to ease the simulation and know how many publications will be paid, a relaxation of the problem has been made. The paid publications ratio has been calculated with an approximation, which also is directly correlated with the amount of contracts held.
- The income from the payments for advertising a publication. This variable has also been approximated making a correlation from the number of contracts. We assumed that 5% of the publications are ads, for a cost of 30€ each.

According to the costs, its function depends on these variables:

- The cost of paying developers, which can be splitted into junior or senior developers. Compared to Sprint 1, we assume that the developers are not payed the first 6 months.
- Marketing cost.
- As the platform will be growing, the costs of the server will be higher. Despite that, it is thought about creating a serverless backend using AWS Lambda to minimize this cost and only pay for the requests.

Three scenarios have been made in order to show a possible but different projection of the incomes generated by the platform:

- A pessimistic scenario, where the number of contracts increases in a linear way amongst the four years.

- An optimistic scenario, in which the contracts' function has an exponential form.
- A more realistic scenario, which also has linear function in the first years but in the lasts, it gets more the way of an exponential function.

9.4 Pessimistic Scenario

Number of contracts per year :

Year	1	2	3	4
Small Business Contracts	518	1448	2262	3713
Big Business Contracts	0	0	381	1079

In this scenario, the number of contracts increase linearly. In comparison to the Sprint 1, the marketing cost are lower on the fourth year. This change is explained by the necessity to drop the costs, due to a pessimistic scenario.

Pessimistic	**Years**					
Project	**1**	**2**	**3**	**4**		
Income of contracts	19 250€	46 000€	129 350€	261 200€		
Income of publications	5 688€	8 750€	14 875€	24 500€		
Total Income	**24 938€**	**54 750€**	**144 225€**	**285 700€**		
Senior Employee	12 000€	24 000€	24 000€	24 000€		
Junior Employees	8 500€	17 000€	17 000€	17 000€		
Marketing	36 000,00 €	36 000,00 €	25 500,00 €	18 000,00 €		
Variables cost	288,75 €	690,00 €	1 371,00 €	2 475,75 €		
Total Cost	**56 789€**	**77 690€**	**67 871€**	**61 476€**		
Cashflow	**-31 851€**	**-22 940€**	**76 354€**	**224 224€**		
ROI %	**-156,09%**	**-129,53%**	**12,50%**	**264,74%**	**-2,10%**	
Net Income				**245 787€**		
PBP Years				**1,07339**		
PBP Month				**13**		
NVP				**222 936€**	**Interest NVP**	1,05
IRR				**107%**		
					Sales unit price	90,00 €
BEP	**53**	**72**	**63**	**56**	**Cost of unit**	0,75 €
Number of contracts	**385**	**920**	**1828**	**3301**		

Table 1: Pessimistic Cash Flow

$$\text{ROI} = \frac{\text{Total Income} - \text{Total Cost}}{\text{Total Cost}}$$

9.5 Realistic Scenario

Number of contracts per year :

Year	1	2	3	4
Small Business Contracts	518	1448	2262	3713
Big Business Contracts	0	0	381	1079

In this scenario, the number of contracts is the mean between the pessimistic and the optimistic scenarios.

Realistic	Years					
Project	**1**	**2**	**3**	**4**		
Income of contracts	25 900€	72 400€	189 300€	401 450€		
Income of publications	6 738€	14 000€	20 125€	36 050€		
Total Income	**32 638€**	**86 400€**	**209 425€**	**437 500€**		
Senior Employee	12 000€	24 000€	24 000€	24 000€		
Junior Employees	8 500€	17 000€	17 000€	17 000€		
Marketing	36 000,00 €	36 000,00 €	36 000,00 €	36 000,00 €		
Variables cost	362,60 €	1 013,60 €	1 850,10 €	3 354,40 €		
Total Cost	**56 863€**	**78 014€**	**78 850€**	**80 354€**		
Cashflow	**-24 225€**	**8 386€**	**130 575€**	**357 146€**		
ROI %	**-142,60%**	**-89,25%**	**65,60%**	**344,46%**	**44,55%**	
Net Income				**471 882€**		
PBP Years				**0,62321**		
PBP Month				**8**		
NVP				**428 011€**	**Interest NVP**	1,05
IRR				**231%**		
					Mean revenue per u	90,00 €
BEP	**53**	**72**	**72**	**72**	**Cost of unit**	0,75 €
Number of contracts	**518**	**1448**	**2643**	**4792**		

Table 2: Realistic Cash Flow

9.6 Optimistic Scenario

Number of contracts per year :

Year	1	2	3	4
Small Business Contracts	610	1801	2720	4415
Big Business Contracts	0	0	472	1372

In this scenario, the number of contracts increase exponentially. In comparison to the Sprint 1, the marketing cost are higher from the third year. This change is explained by the possibility to raise the costs, thanks to an optimistic scenario.

Optimistic	Years					
Project	1	2	3	4		
Income of contracts	30 500€	90 050€	230 400€	495 150€		
Income of publications	7 438€	17 500€	23 625€	43 750€		
Total Income	**37 938€**	**107 550€**	**254 025€**	**538 900€**		
Senior Employee	12 000€	24 000€	24 000€	24 000€		
Junior Employees	8 500€	17 000€	17 000€	17 000€		
Marketing	36 000,00 €	36 000,00 €	41 250,00 €	45 000,00 €		
Variables cost	396,50 €	1 170,65 €	2 074,80 €	3 761,55 €		
Total Cost	**56 897€**	**78 171€**	**84 325€**	**89 762€**		
Cashflow	**-18 959€**	**29 379€**	**169 700€**	**449 138€**		
ROI %	**-133,32%**	**-62,42%**	**101,25%**	**400,37%**	**76,47%**	
Net Income				**629 259€**		
PBP Years				**0,49130**		
PBP Month				**6**		
NVP				**570 756€**	**Interest NVP**	1,05
IRR				**361%**		
					Sales unit price	90,00 €
BEP	**53**	**72**	**77**	**81**	**Cost of unit**	0,75 €
Number of contracts	**610**	**1801**	**3192**	**5787**		

Table 3: Optimistic Cash Flow

9.7 Economic indices comparison between scenarios

All information related to the simulation and prediction about the first four years of the platform can be found in the spreadsheet attached to this document. Even though, the cash flows of each scenario can be found in *Tables 1, 2 and 3.* In this section, we will begin by comparing the three scenarios using the different indexes calculated above.

First, we want to show you the `ROI` index between the different scenarios, which can be found in Table 4. The `ROI` has been calculated every year, as it is more significant to us to analyze this information. Although it can be found that the `ROI` index in the first year is similary between the scenarios, the pessimistic scenario has some challenges to increase the ROI so as to be positive while optimistic and, actually, the realistic appear to have a better return on investment.

	ROI				
Project / Year	**1**	**2**	**3**	**4**	**Mean**
Pessimistic	-168,94%	-132,15%	-68,73%	41,15%	-82,17%
Realistic	-159,39%	-93,35%	-10,22%	166,56%	-24,10%
Optimistic	-152,81%	-67,50%	29,73%	249,98%	14,85%
Mean	-160,38%	-97,67%	-16,41%	152,56%	-30,47%

Table 4: ROI Index comparison between the three scenarios

According to the other indices, mostly all of them have better values in the optimistic scenario. One of the most valuable indices is the `PBP` in terms of months. Here, we can see that the optimistic has only 9 months of `PayBack Period`. On the other hand, realistic scenario is a promising case,

as in only one year and a month we would be able to recover the cost of the investment.

Although `ROI` gave us bad values, specially in the pessimistic scenario, we can see that in all the scenarios the `IRR` is positive. For the interest rate, we chose 1,05, that is quite similar to the inflation rate. That gave us good information to know if someone is going to invest on the company. The only one index that has the same behaviour in all the cases it is the `BEP`, which is calculated monthly. That is because the costs are fixed, as the lifecycle of the software would be large.

Indices	PBP (year)	PBP (month)	NVP	IRR
Pessimistic	1,07	13	222 936€	107%
Realistic	0,62	8	428 011€	231%
Optimistic	0,49	6	570 756€	361%
Mean	0,73	9	407 234 €	233%

Table 5: Indices comparison between the three scenarios

	BEP (monthly)				
Project / Year	1	2	3	4	Avg
Pessimistic	53	72	63	56	61
Realistic	53	72	72	72	67,25
Optimistic	53	72	72	72	67,25
Mean	53	72	69	66,66666667	

Table 6: BEP comparison between the three scenarios and years

In Figure 51, we can see the contracts' function of all the scenarios and the total cost function, which is the same in all scenarios. In the optimistic scenario, we will start to recover the initial investment at the beginning of the first year whereas in the worst case, we will start generating benefits at the beginning of the third year.

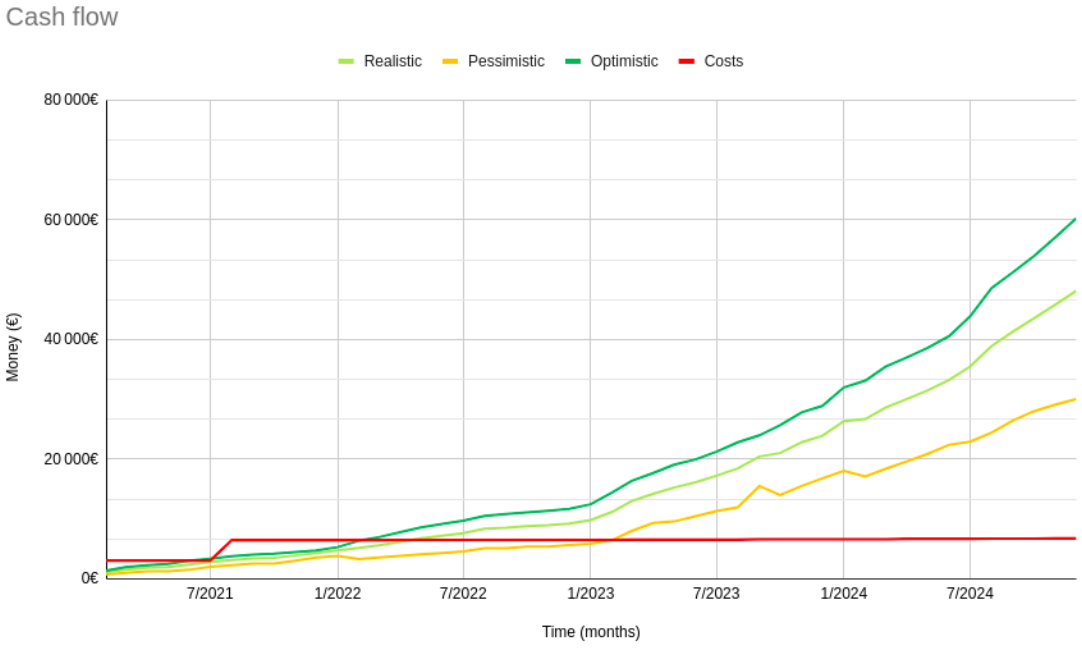

Figure 51: Cash flow of the different revenue function from scenarios and cost function