

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.

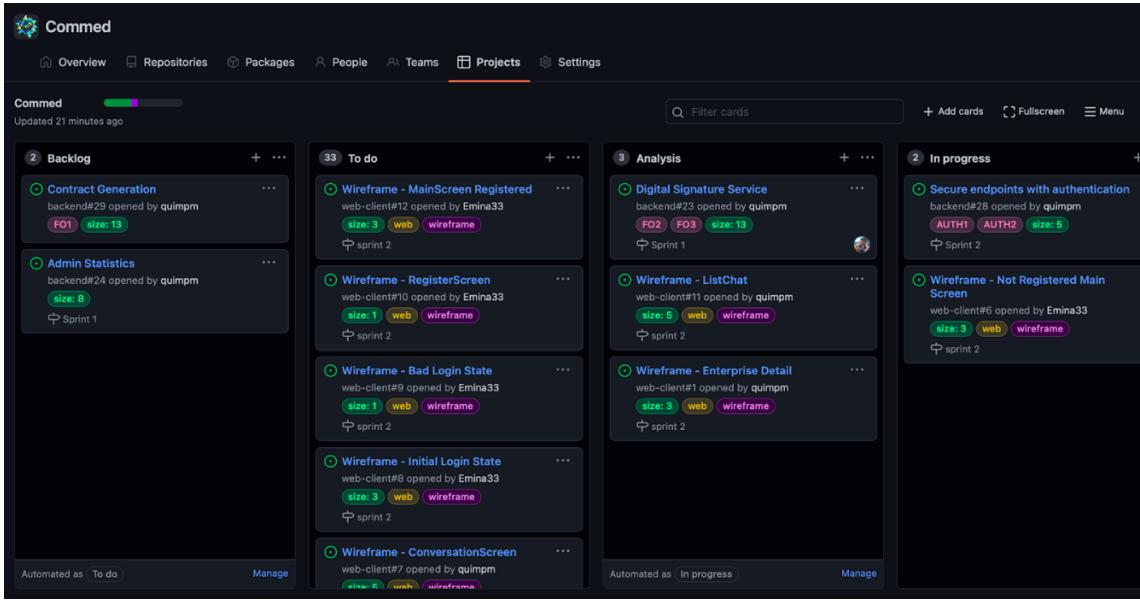


Figure 1: Screenshot of the first days of the project kanban.

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
- Implementation 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
- Implementation 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

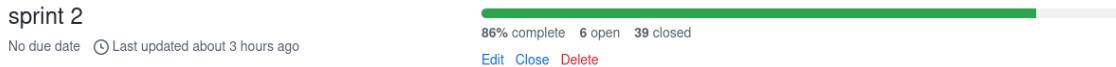


Figure 4: Web client Sprint 2 milestone

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.

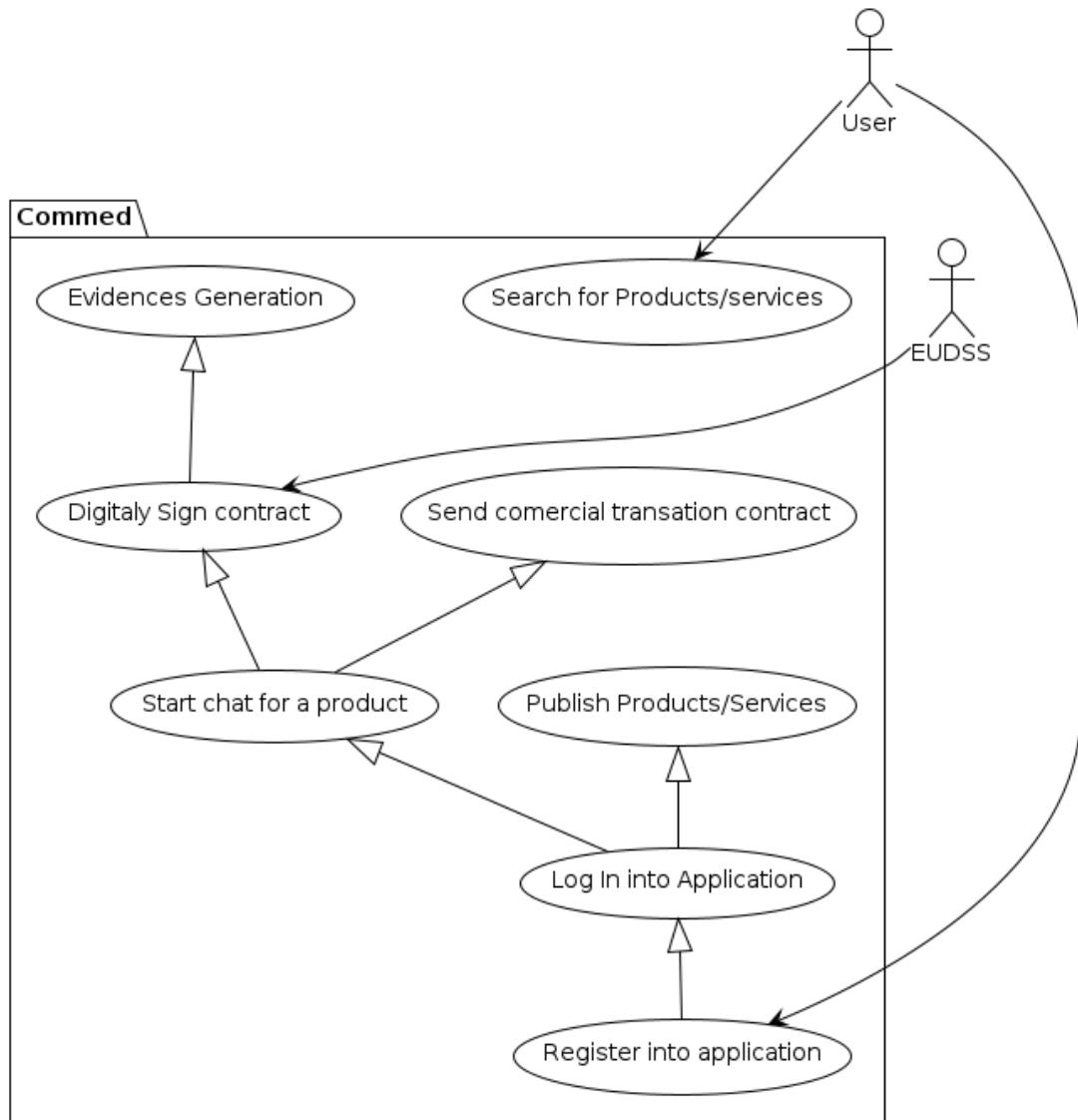


Figure 5: Use case diagram of the application.

- **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

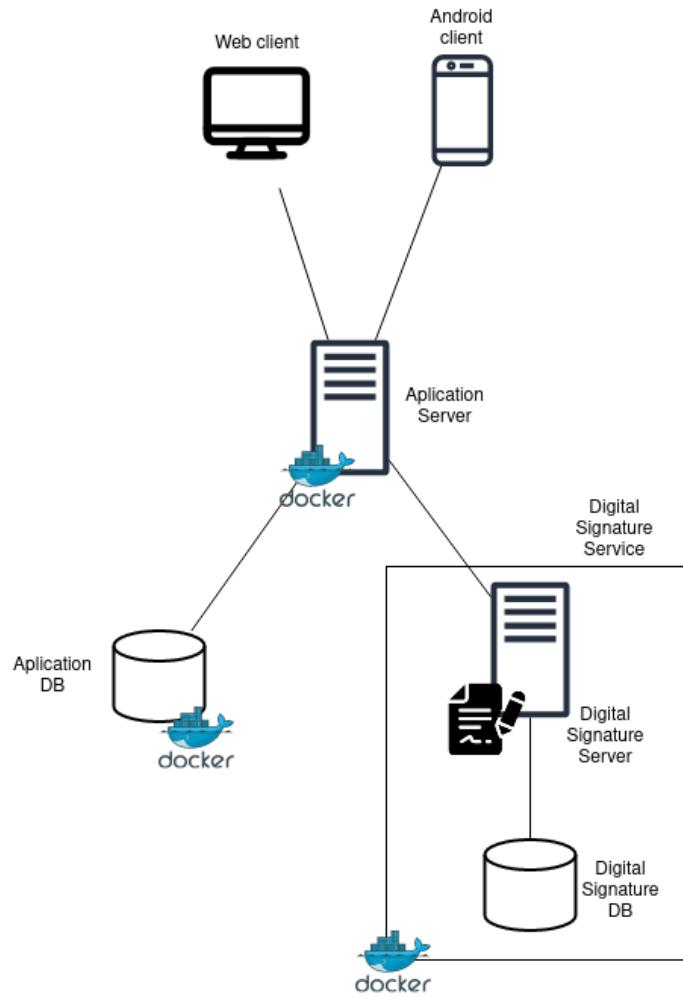


Figure 6: General architecture of the application.

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.

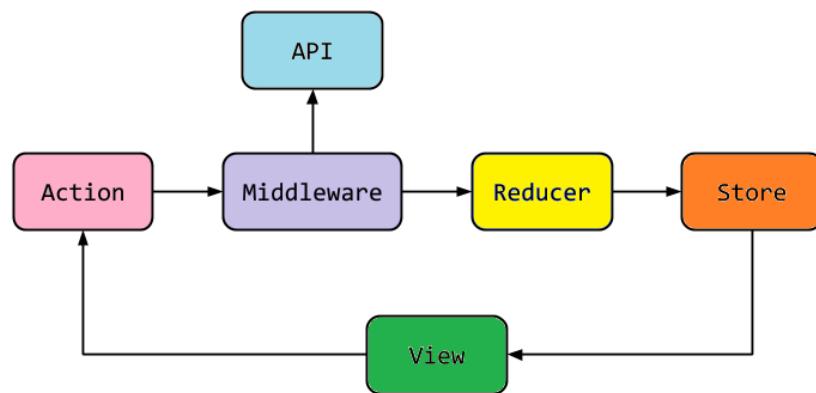


Figure 7: Redux Diagram.

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 his target user in the administrative world, we decided to implement a web alongside the mobile application that will make easier the usage of our servies to our customers in administrative enviroments. We though that the mobile app can be a really usefull for doing little tramsits, for chatting with some customers, for quick searches of products/services and enterprises... But we felt that in order to make it more usable and accessible for administrative porpouses we should create a web client also.

Furthermore, the implementation of the web client showed 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 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 we used React, which is a library built in javascript for creating User Interfaces. To be more precise we used React because it gives the oportunity of working with Hooks. Hooks are functions that comprehend inside him immutable components with independent states, which makes the code a lot more clean and structured and simplifies the amount of chaos that we have to deal usually when we program 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 we are using the default testing server that React provides to us. Although, we expect to use a nginx+docker 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, we also used redux. In fact we use a react library called redux-react-session, that builds a store to mantain the state of our session. Also it creates the correspondig Session Cookie to mantain the value of the authorization token provided by our backend. This way, every time we have to make a request that needs authoritzation, we will be able to pass within the request the value of the token stored in the browser cookie.

6 Database model

The database model can be seen at figure 8.

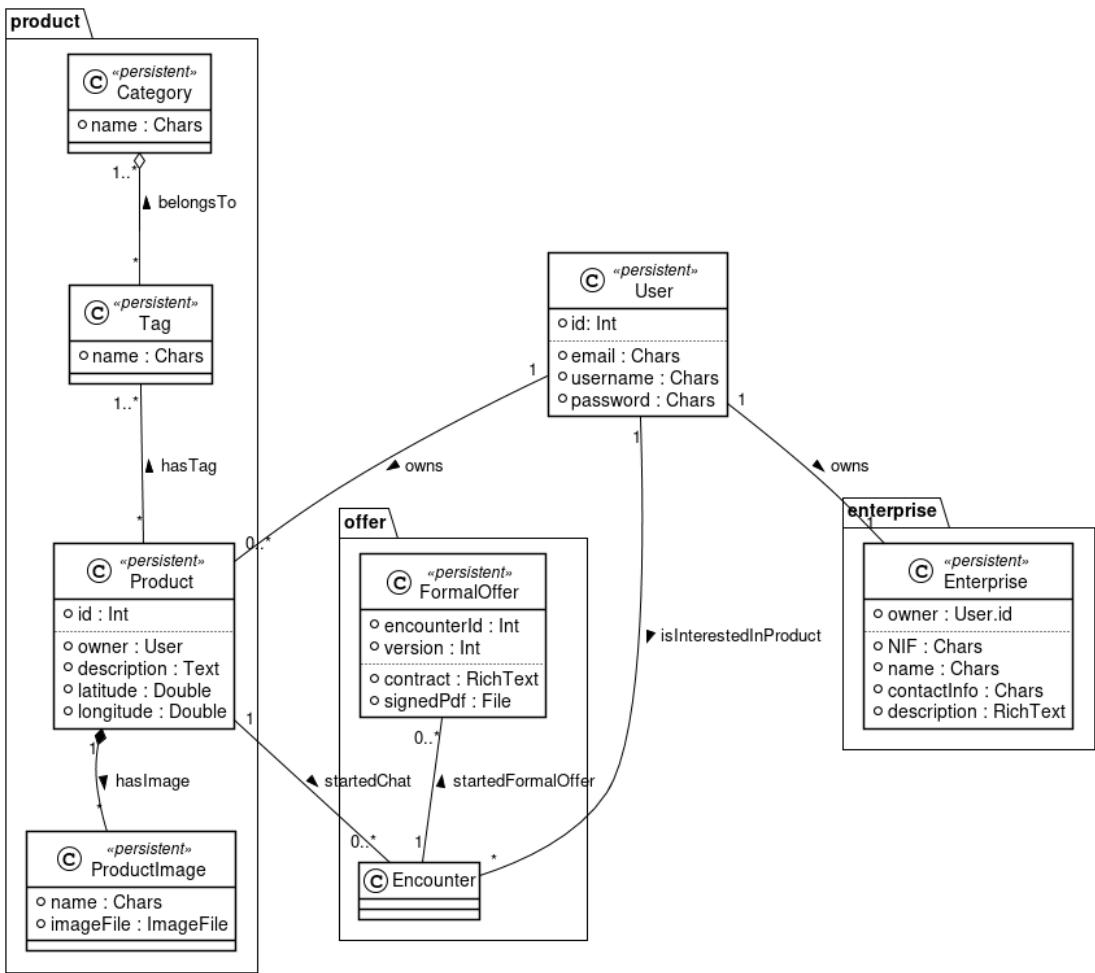


Figure 8: UML diagram of the models.

6.1 Modifications on the models

7 Main Screens

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 9. The next image 10 contains the login screen, which can be accessed by clicking on the user icon the image before. The register 11 is accessed by clicking at the sign-up option.

The image 12 contains the list of chats that you currently have for every product or service, and the 13 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 14. 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 15 and a submenu for things related to the user 16. 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.

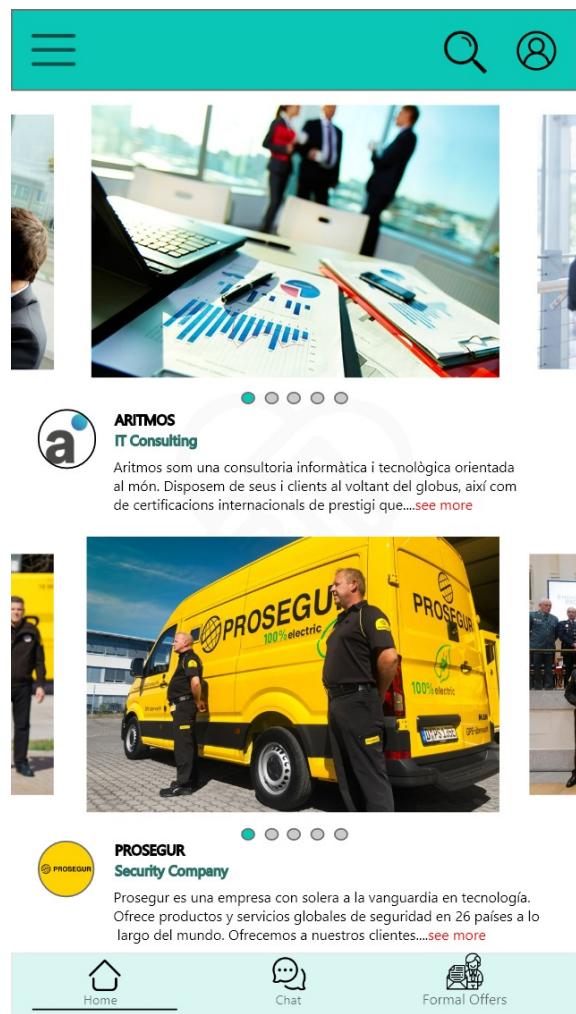


Figure 9: Home mobile screen, it lists the products

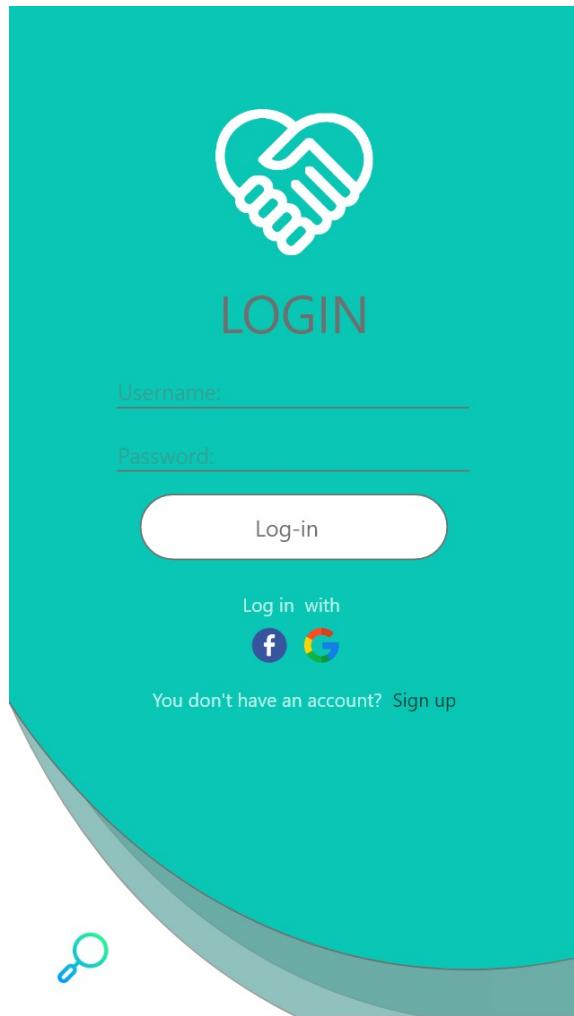


Figure 10: Login view

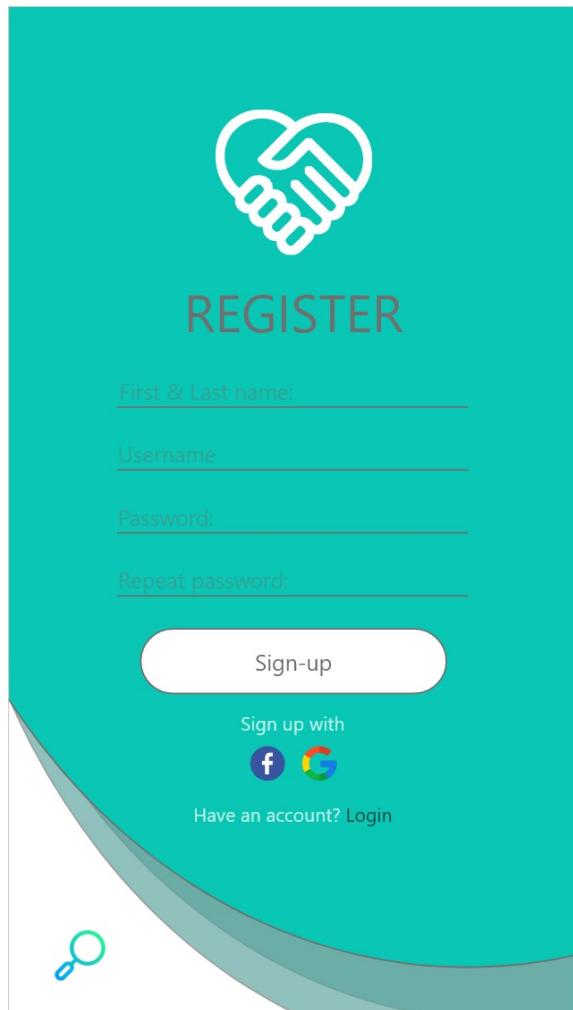


Figure 11: Register view

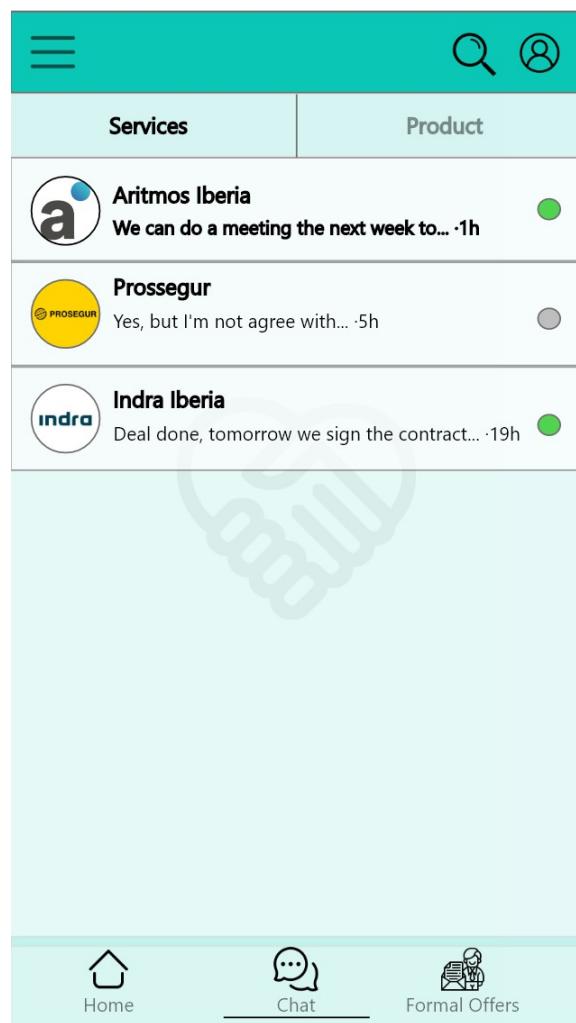


Figure 12: List of the current chats.

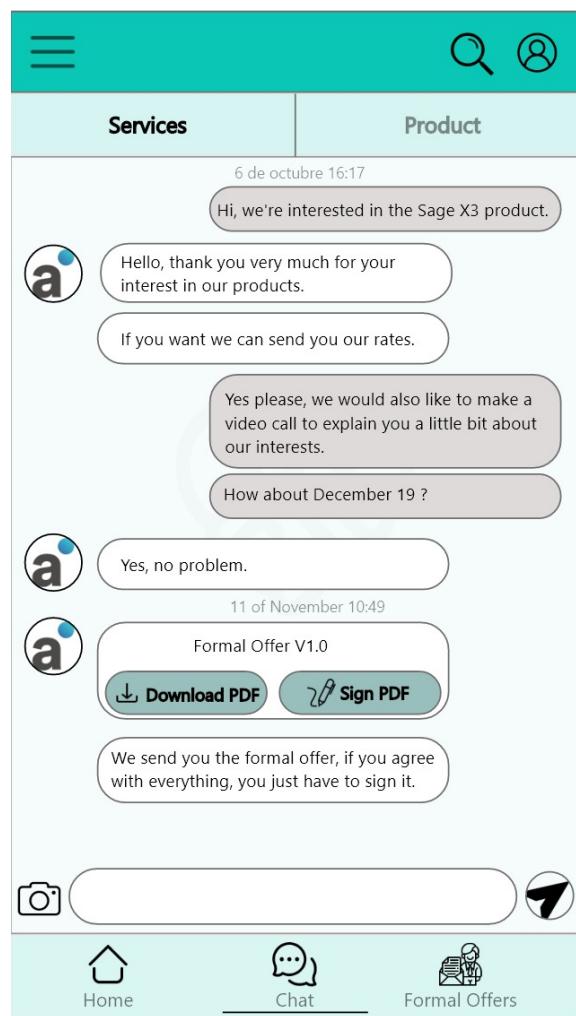


Figure 13: A chat conversation.

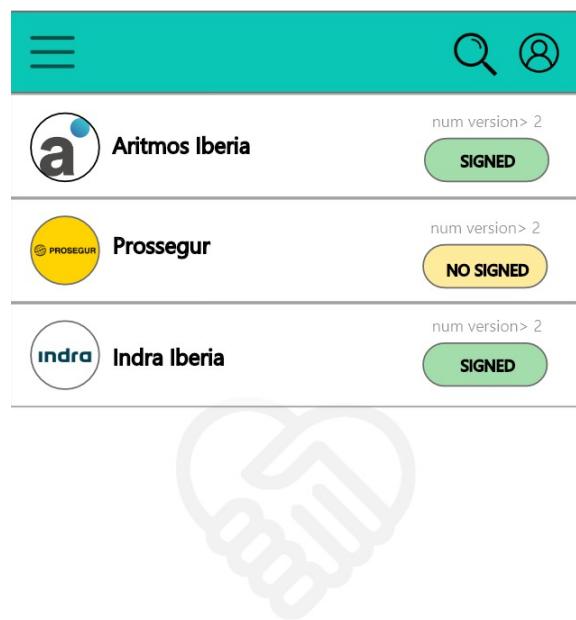


Figure 14: List of formal offers.



Description

Prosegur és una empresa de serveis globals de seguretat fundada el 1976, i que el 1987 es converteix en la primera empresa espanyola de seguretat que cotitza a la Borsa de Madrid.[1] Prosegur és, a més, la tercera companyia mundial en el sector de la seguretat privada.

Prosegur neix el 1976 fundada per Heriberto Gut, apostant per un sector incipient a Espanya: la seguretat privada.



Home



Chat



Formal Offers

Figure 15: An enterprise profile.

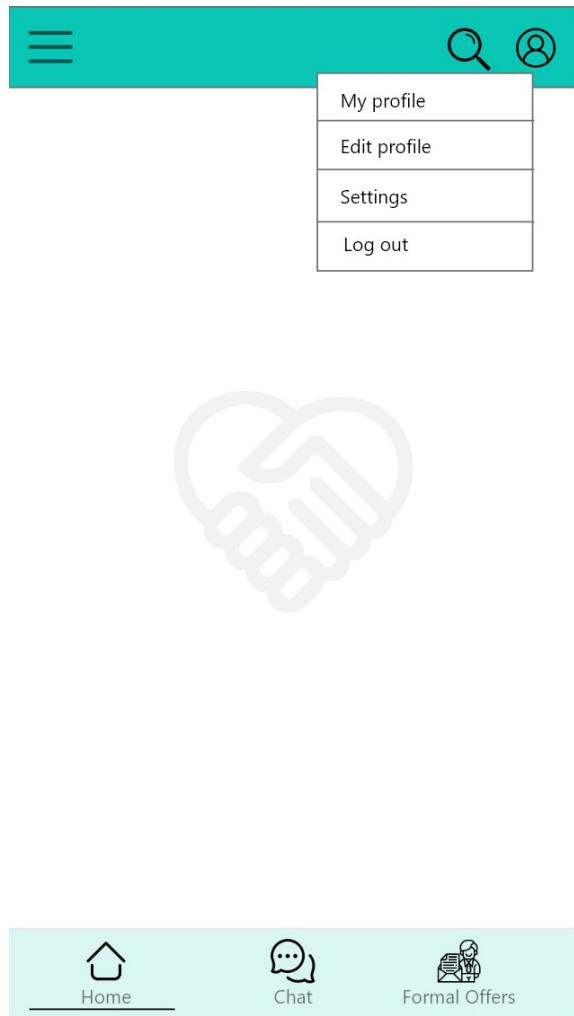


Figure 16: Menu Profile.

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 20 is the profile of an enterprise, which is triggered when clicking a logo of the enterprise, in the pages before. In the figure 21, 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 thought that showcasing both items in the mock would be better. The next image 22 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 23, 24, it is shown the registration and the login screens.

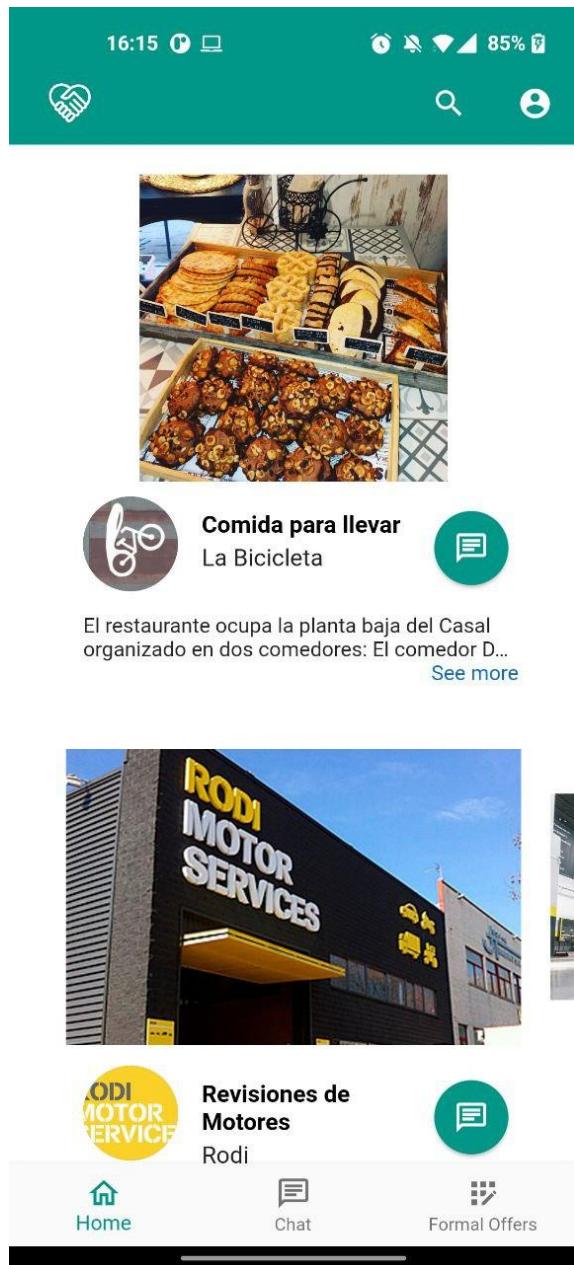


Figure 17: Home mobile screen, it lists the products

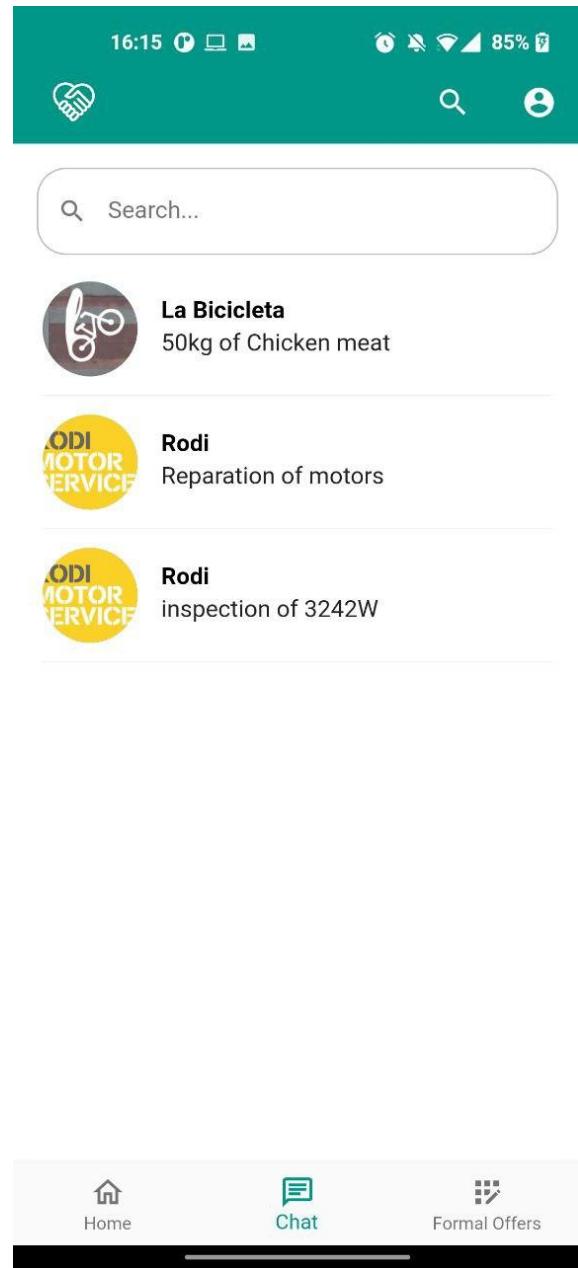


Figure 18: List of the current chats.

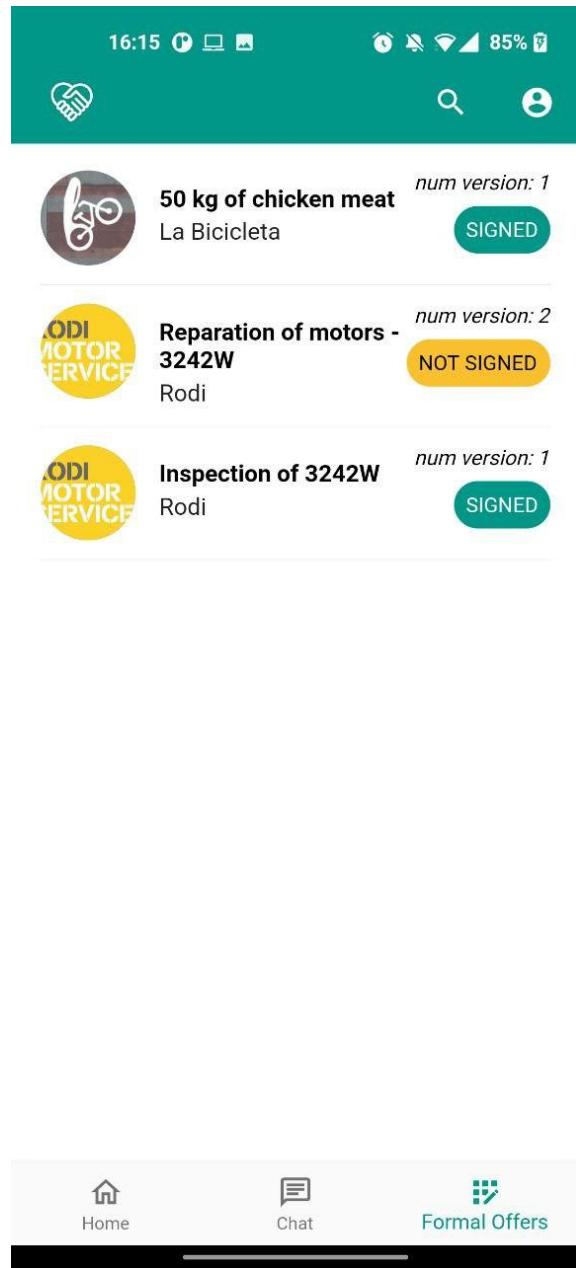


Figure 19: List of the current formal offers.

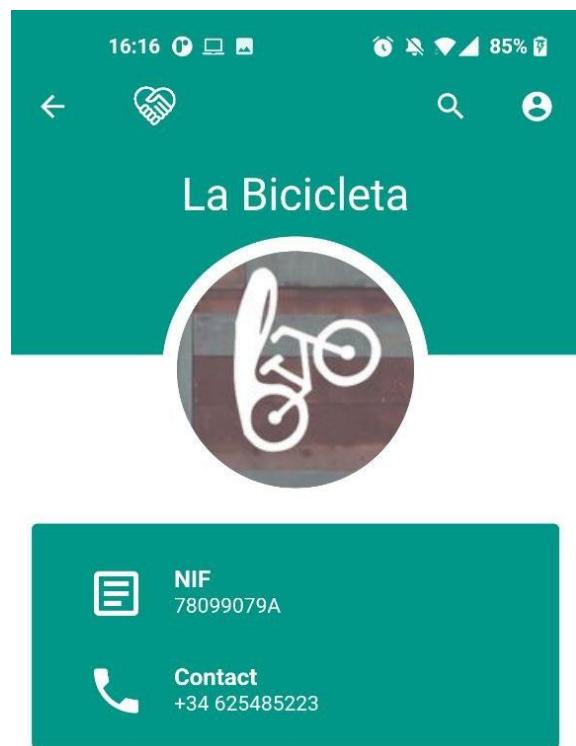


Figure 20: Detail of an enterprise, it can be found when clicking the logo of it.



Figure 21: Chat screen.

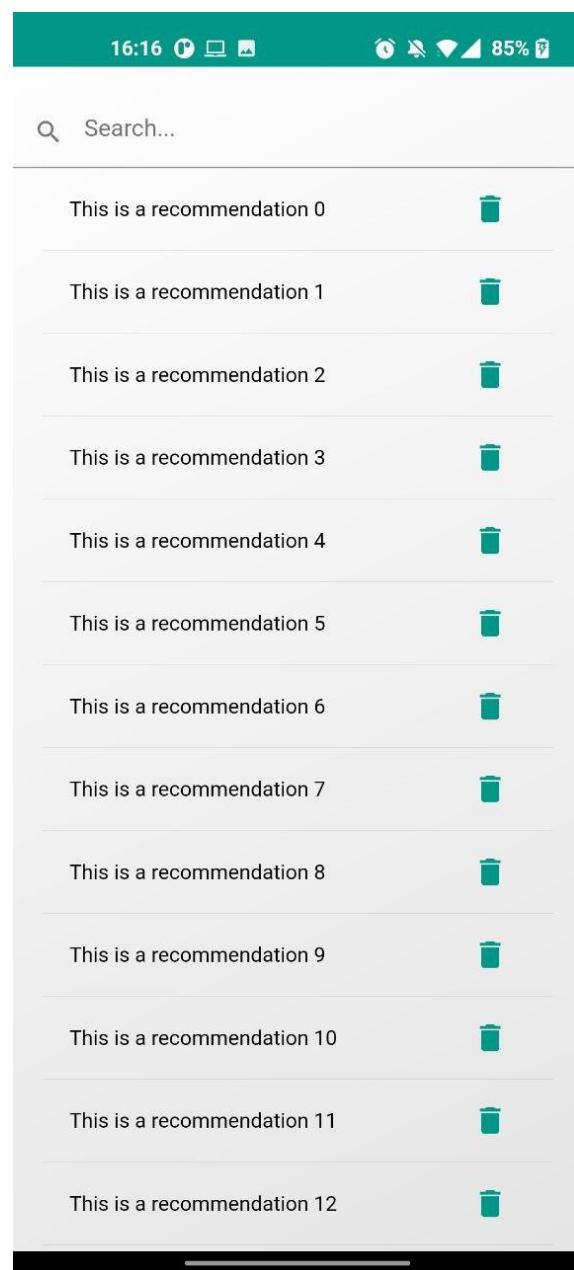


Figure 22: Recommendations screen.

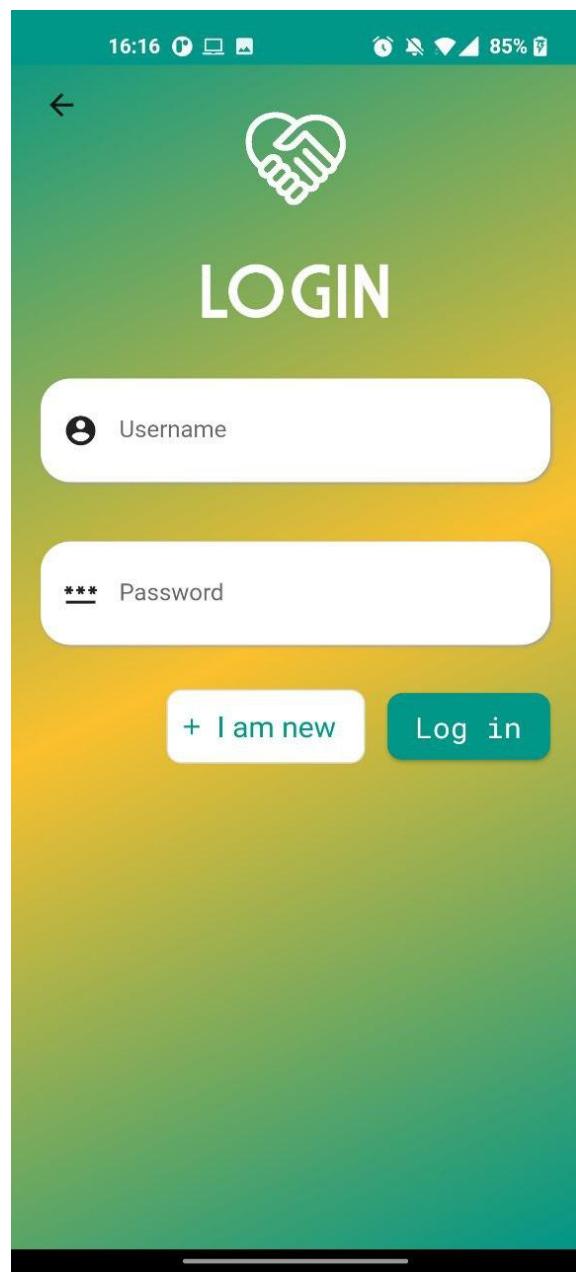


Figure 23: Login view.

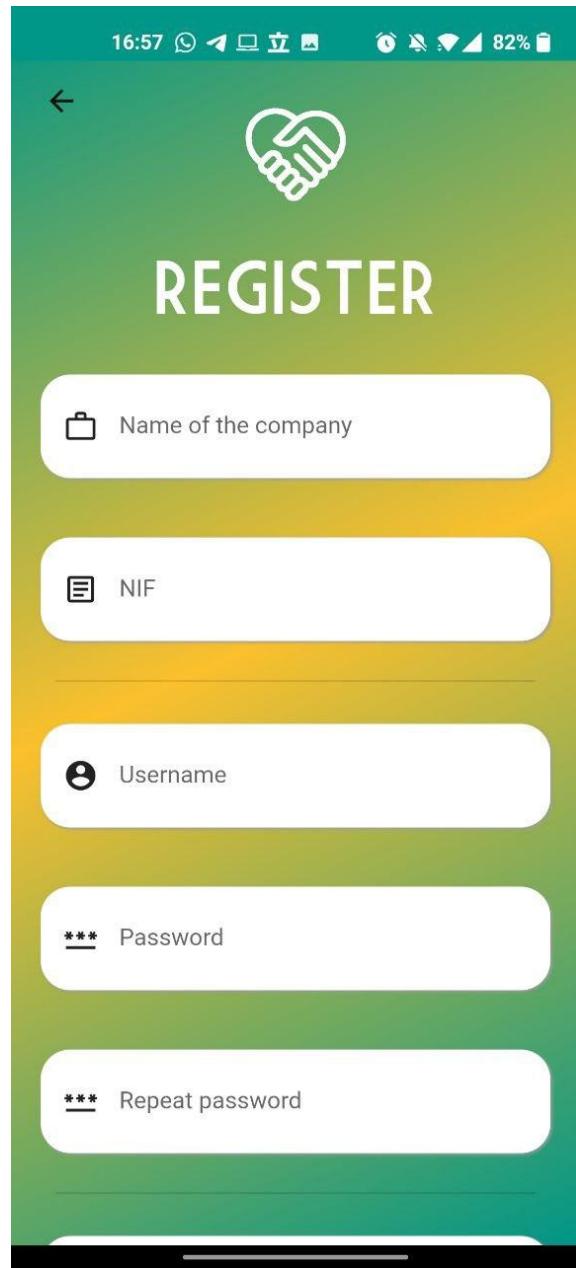


Figure 24: Registration screen.

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 design of the web application.

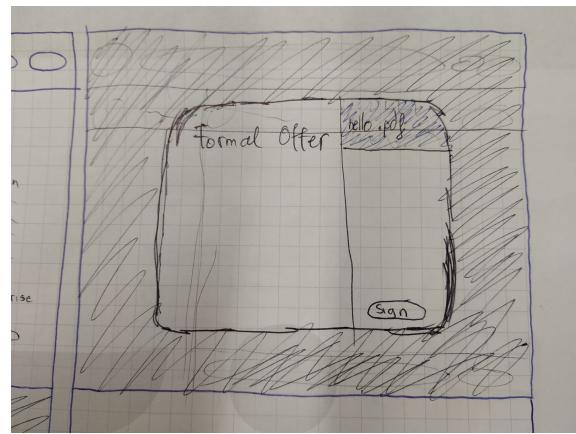


Figure 25: FormalOffer Detail Sign screen.

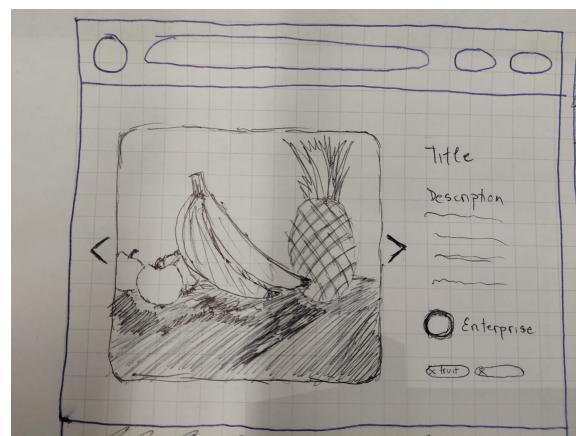


Figure 26: Product Detail screen.

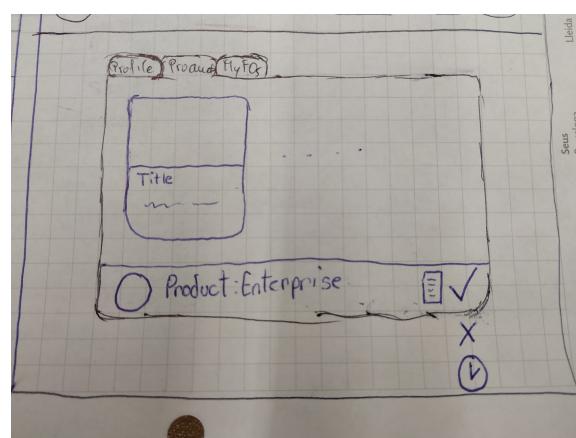


Figure 27: Products/Formal Offers Profile screen.

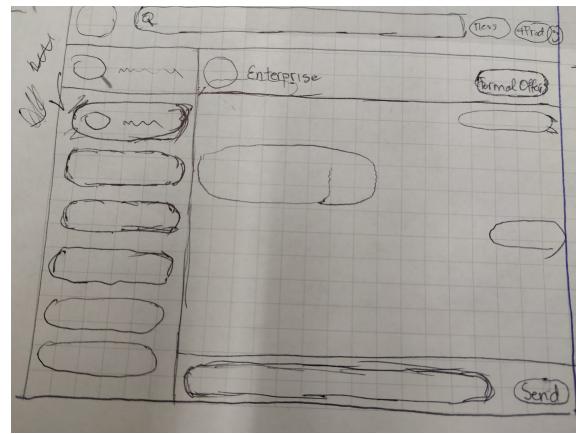


Figure 28: Chat screen.

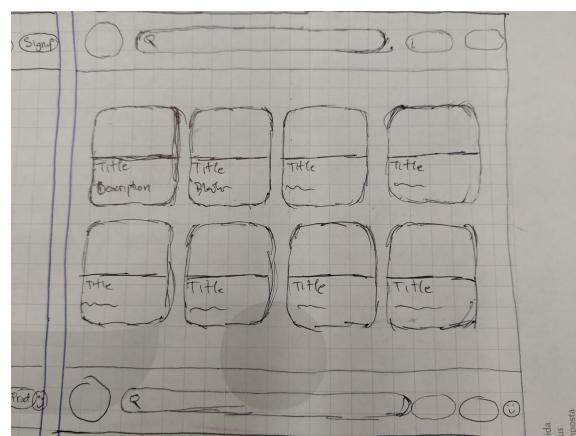


Figure 29: Search screen.

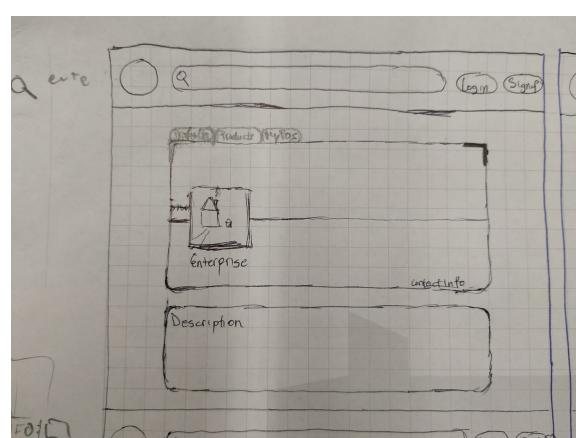


Figure 30: Profile screen.

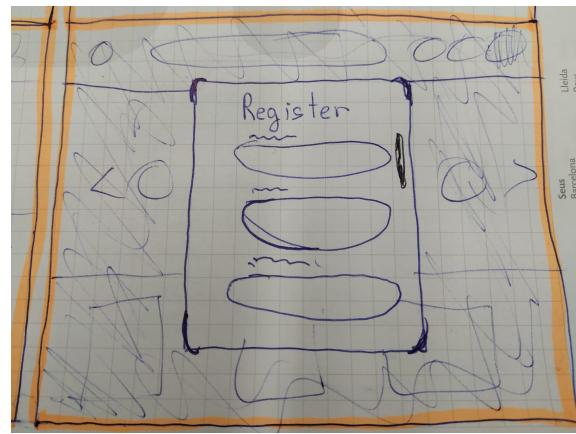


Figure 31: Register screen.

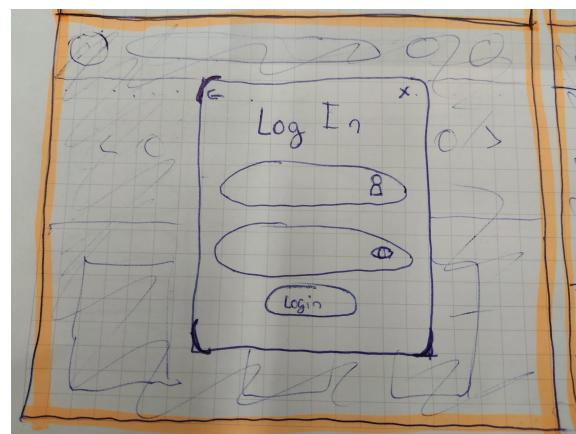


Figure 32: Log In screen.

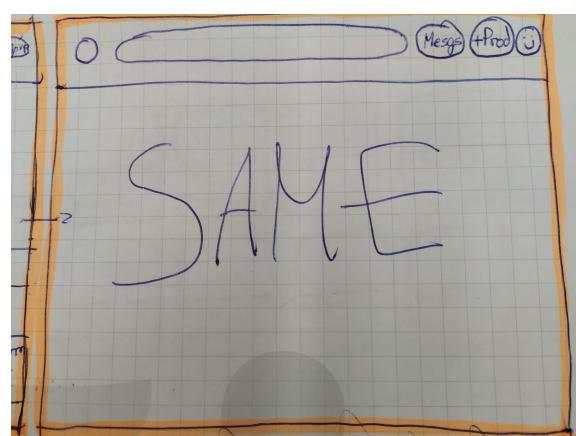


Figure 33: Home Logged In screen.



Figure 34: Formal Offer Send screen.



Figure 35: Home screen.

7.2.2 Application

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

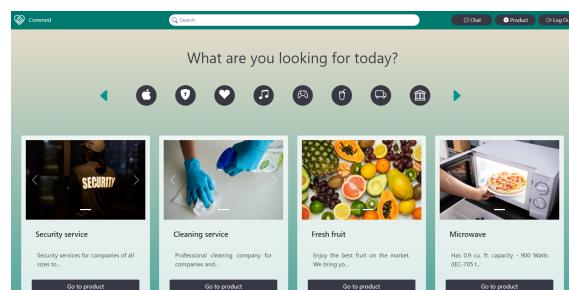


Figure 36: Home screen.

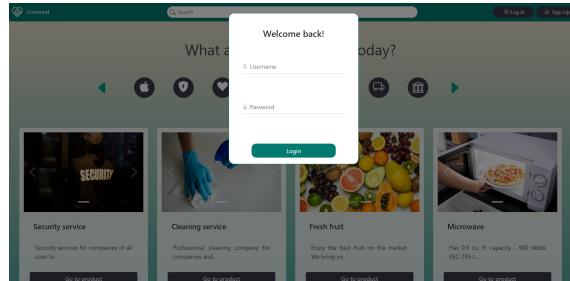


Figure 37: Log In screen.

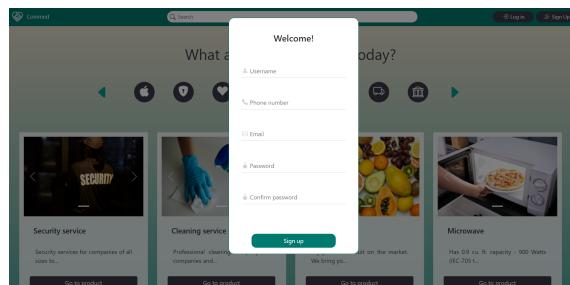


Figure 38: Registration screen.

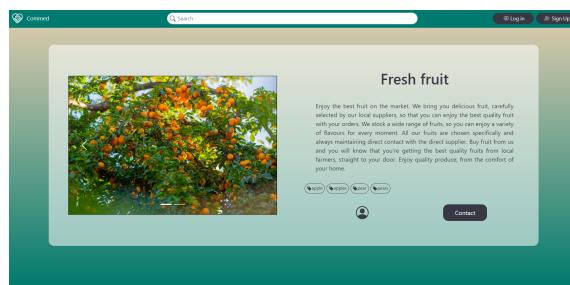


Figure 39: Product Details screen.

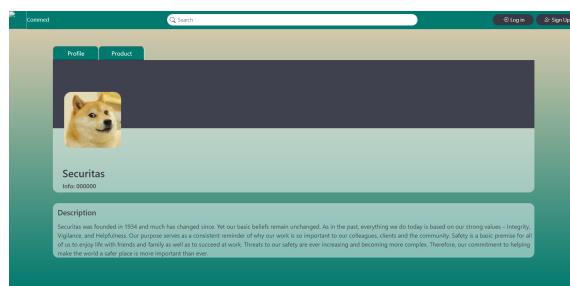


Figure 40: Profile screen.

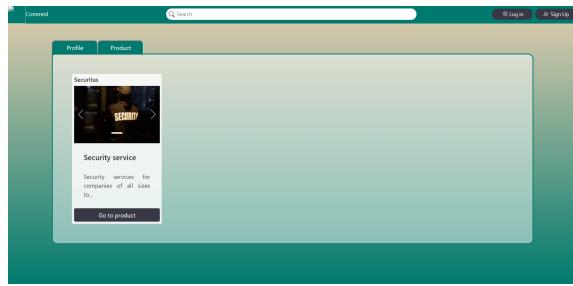


Figure 41: Profile Products screen.

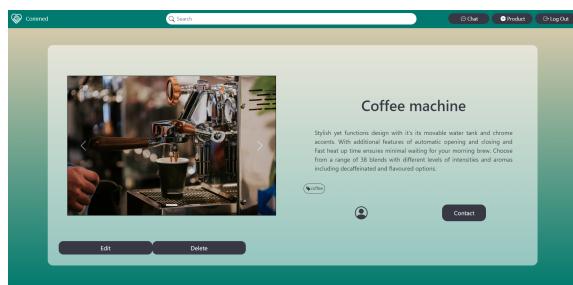


Figure 42: Profile Products Loged In screen.

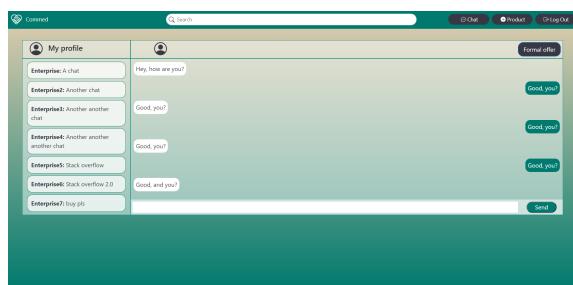


Figure 43: Chat screen.

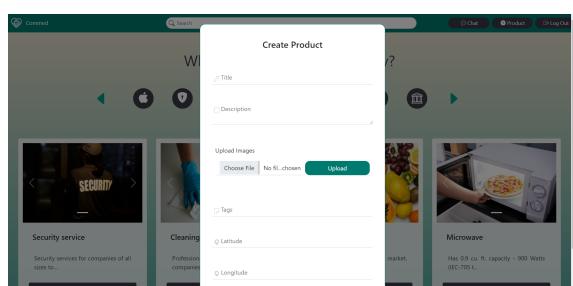


Figure 44: Create Product screen.

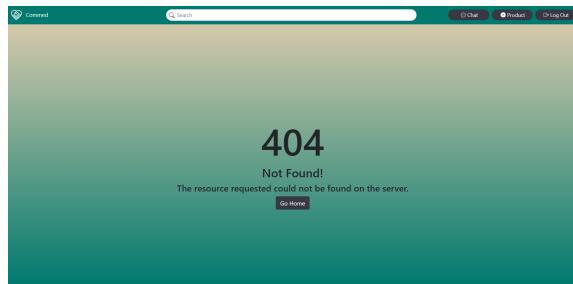


Figure 45: 404 screen.

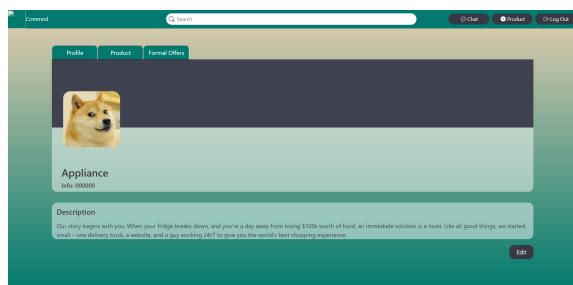


Figure 46: Edit screen.

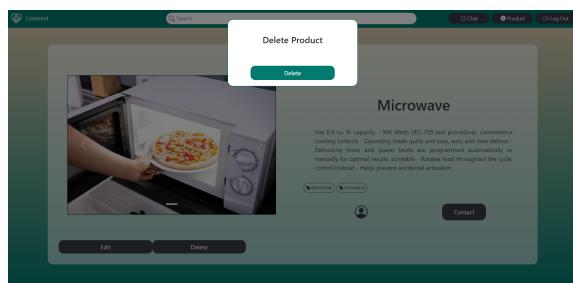


Figure 47: Profile Loged In screen.

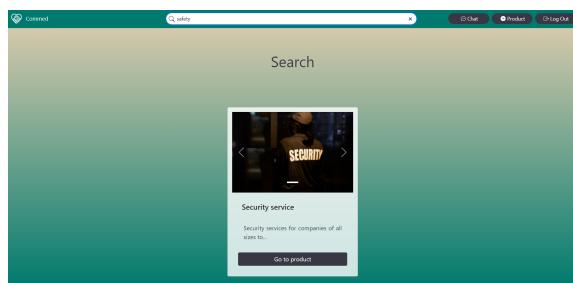


Figure 48: Delete Product screen.

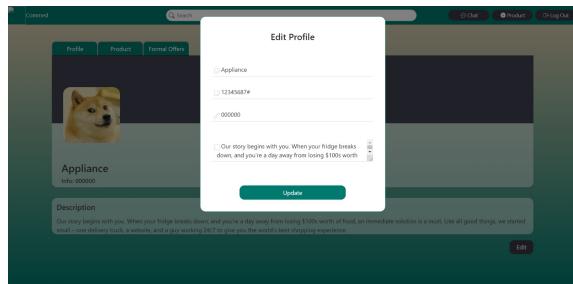


Figure 49: Search screen.

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 Application 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 in our app and mobile app
- Decided the logos that we are using in our web client and App
- Put In common design decisions 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 first steps of the Web app.
- Jass showed to us some screens of the wireframe, and opened some design discussion.
- Oriol and sergi showed bottom 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.