Table of Contents

Product Vision 1
User Interfaces 2
Project Architecture
- Introduction 6
- Technological Architecture 7
- Logical Architecture 8
- Physical Architecture 9
- Key design decisions 10

Product Vision

Ao Dispor is a social network that intends to help freelance professionals get more job opportunities and ease people looking for their services to find them without depending on companies/enterprises.

The goal of this project is to create employment opportunity for freelance professionals, and found by people who need their services.This saves both professionals and clients time.Clients can always check professionals information and choose the best.

It's hard to find a professional to solve that problem without having to rely on contacting an enterprise. Meanwhile there are a lot of professionals who could solve our situation for a lower price, the problem is the lack of contact information to reach those professionals.
With this problem in mind, this project will consist on a website and mobile app, where professionals may register and therefore be contacted by someone requiring their services.
This is a project without lucrative purposes, which only intention is to help people get help from those unemployed professionals who work for no company and therefore have no ways of being contacted.

STATEMENT:

The founder of Ao Dispor www.aodispor.pt requires our help to improve the quality of the beforementioned website and the development of a mobile app that serves same purpose as the website. This website offers opportunity for freelance professionals to be easily found and contacted by people who need their services. This website has no lucrative purpose, unlike existing website, which has the same purpose, but demands people to pay in order to use their services.

STAKEHOLDERS AND NEEDS:

Our customer and founder of the website has only the intention of helping those professionals and clients to communicate with each other. He will work alongside the development team to verify if everything goes according to his desires.

The clients will have a much better time when looking for some professional to hire.

The professionals will have a lot of more employment and services due to the visibility the website offers them.

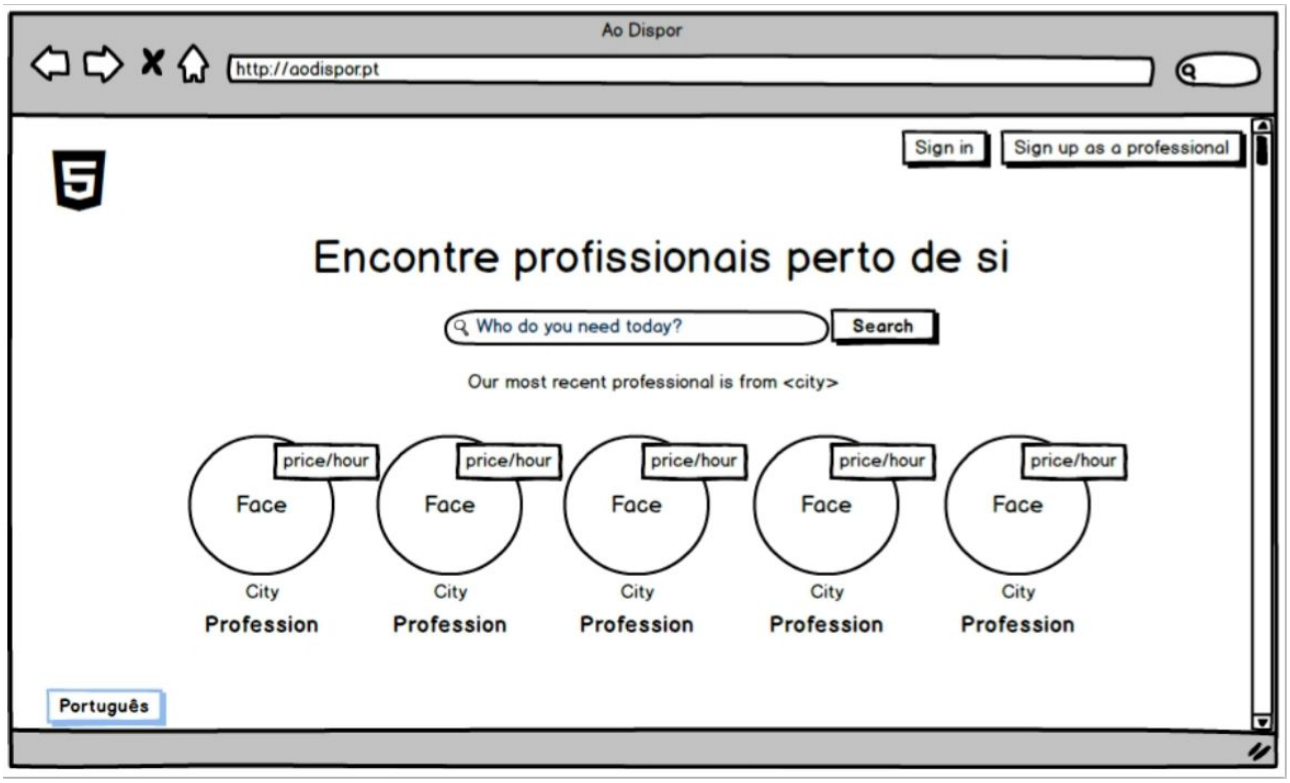
The development team will have to work in the website and develop a mobile app.

QUALITY REQUIREMENTS:

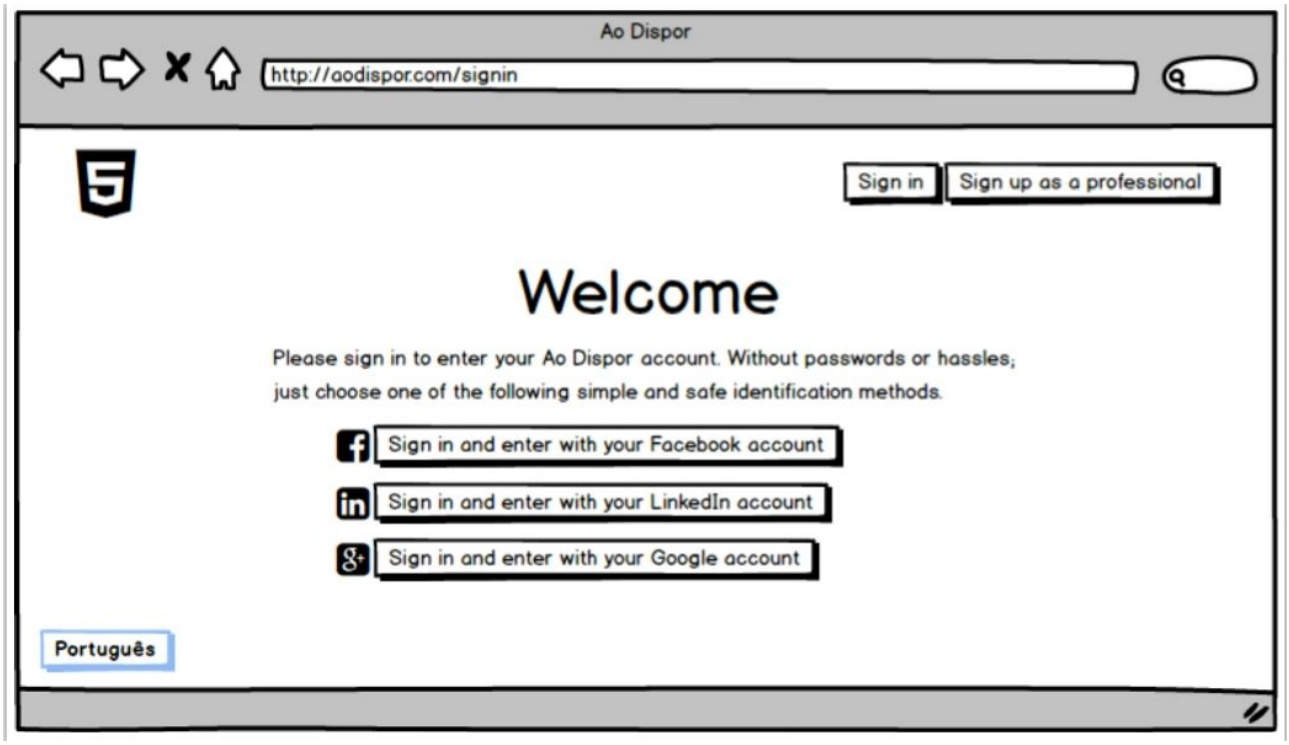
The site and mobile app must be really easy to use and understand, so that it may be usable by people with very low informatic associated knowledge.

User Interfaces

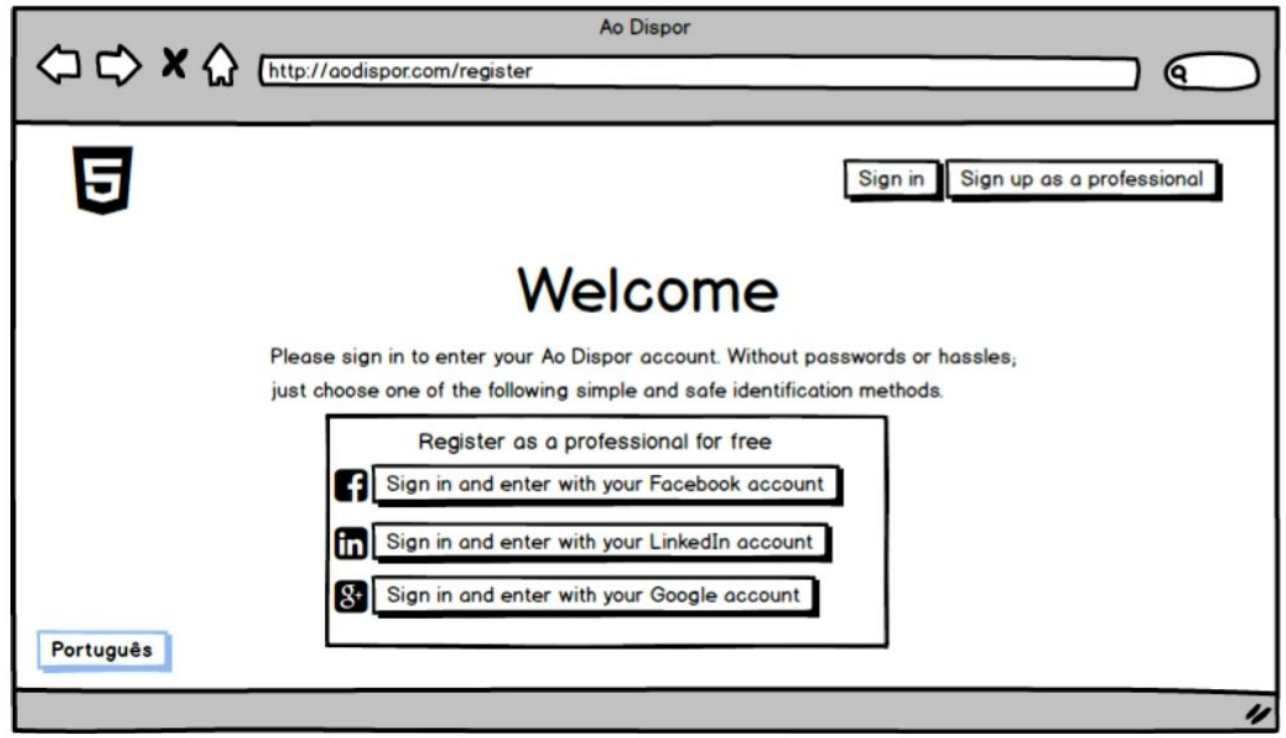
Homepage



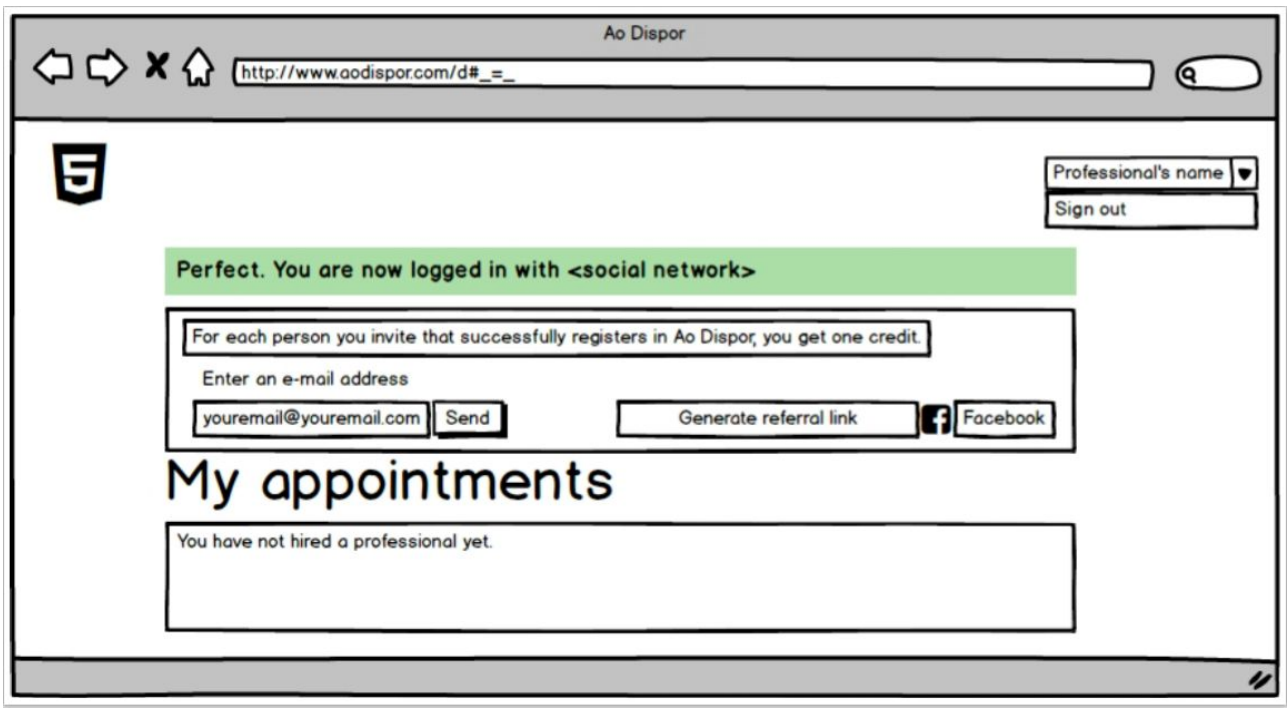
Sign in



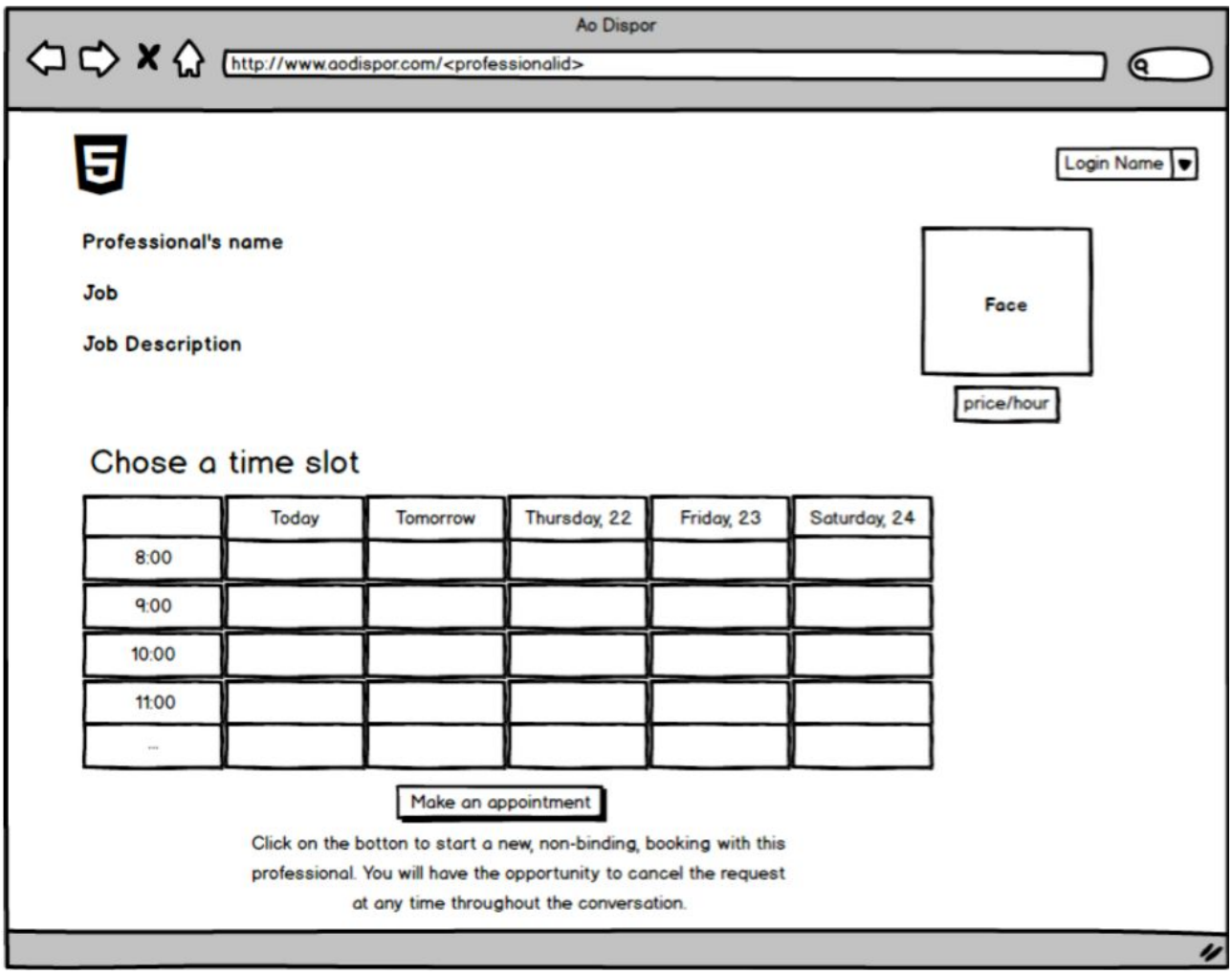
Sign up



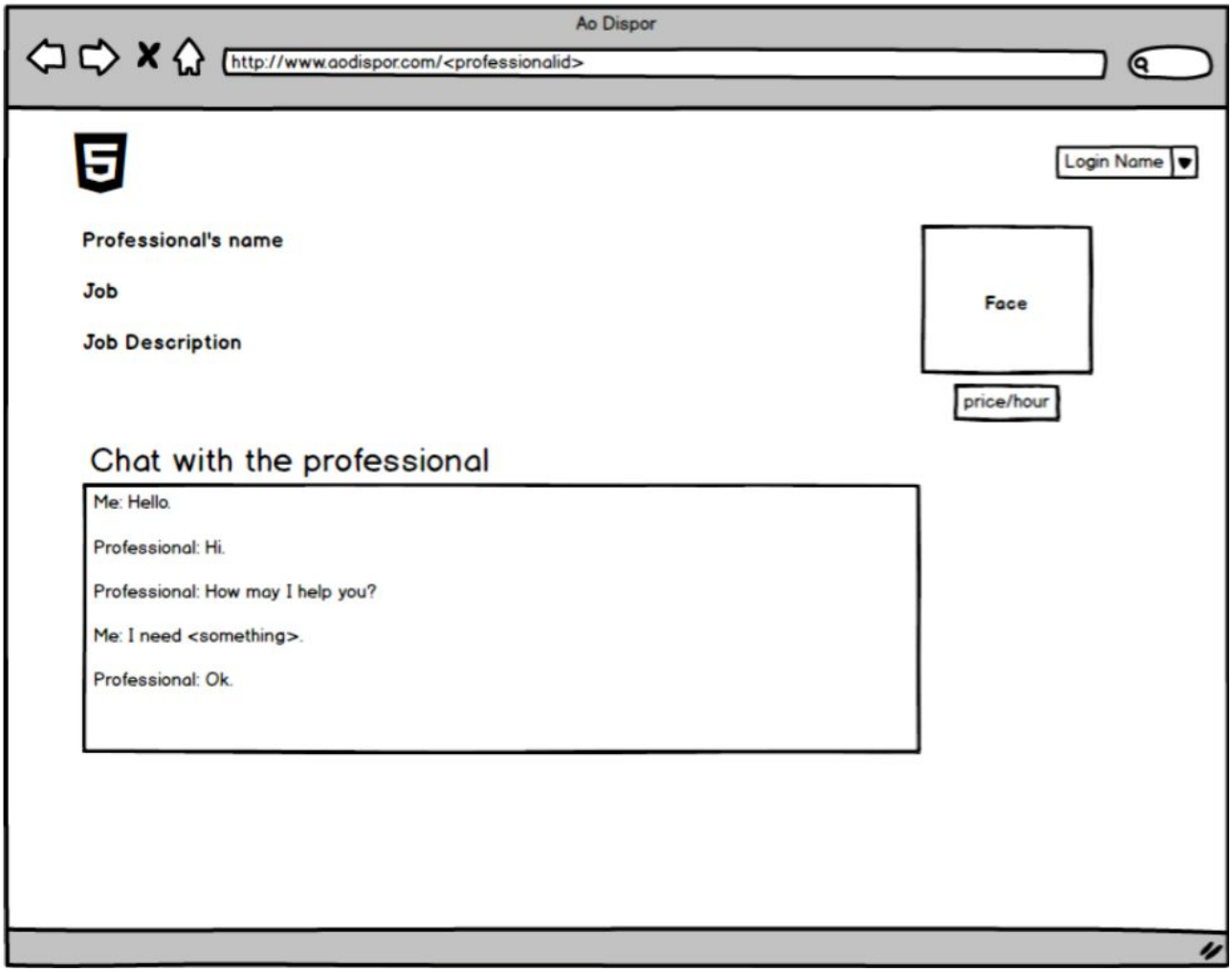
Dashboard



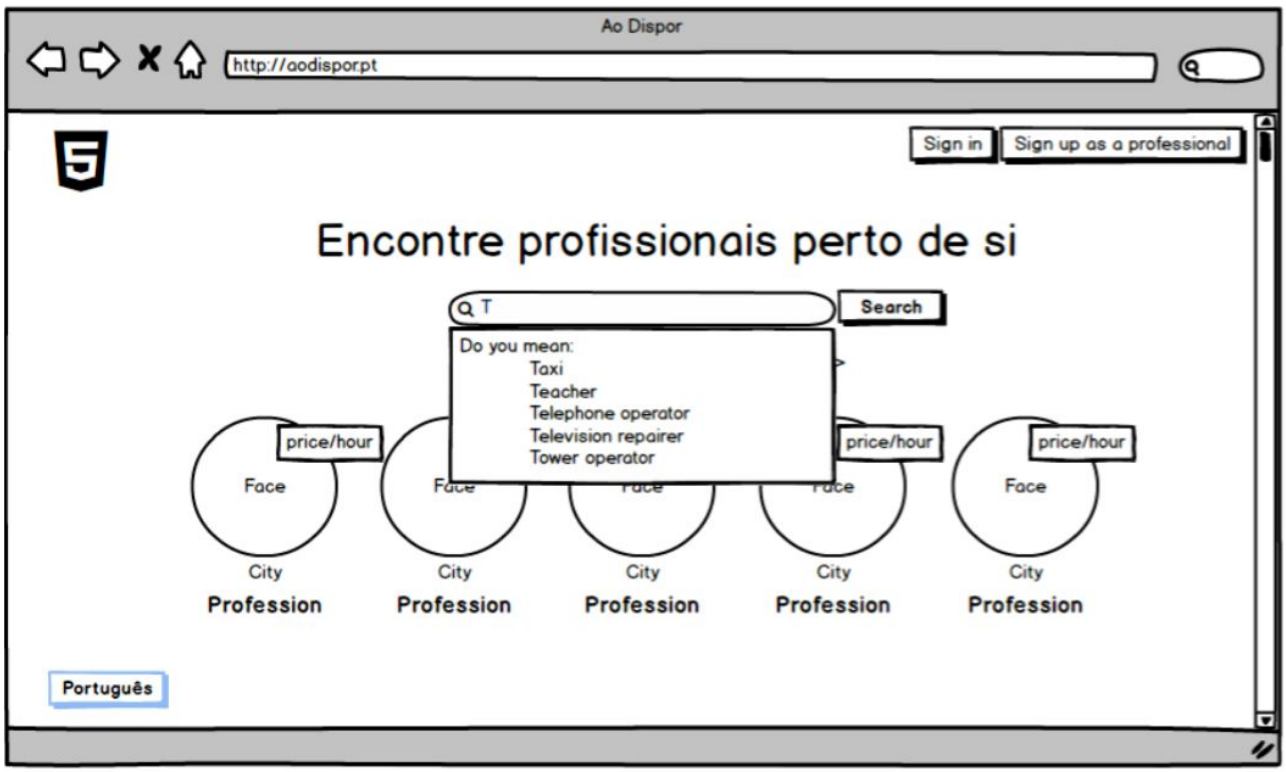
Reservation



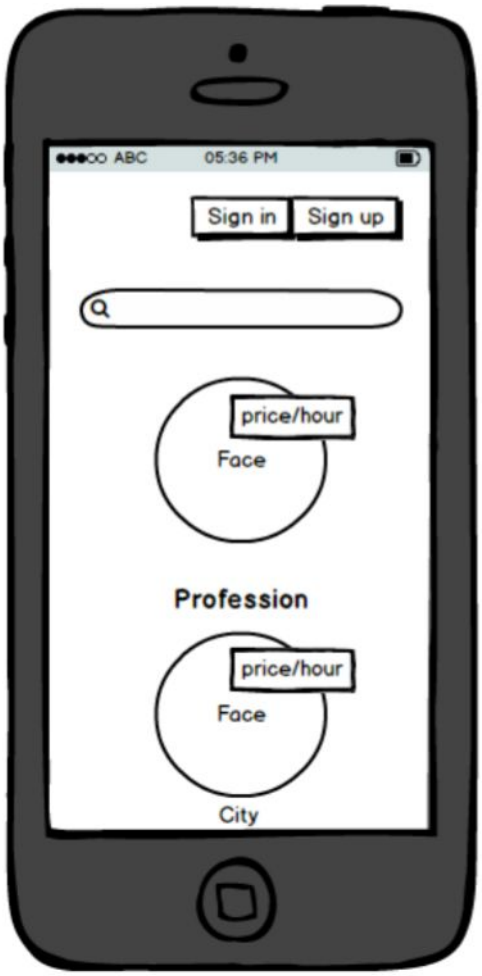
Chat


Ao Dispor
http://www.aodispor.com/<professionalid>
Login Name
Professional's name
Job
Job Description
Face
price/hour
Chat with the professional
Me: Hello.
Professional: Hi.
Professional: How may I help you?
Me: I need <something>.
Professional: Ok.

Search


Ao Dispor
http://aodispor.pt
Sign in
Sign up as a professional
Encontre profissionais perto de si
T
Search
Do you mean:
Taxi
Teacher
Telephone operator
Television repairer
Tower operator
price/hour
Face
City
Profession
Português

Mobile homepage



Project Architecture

Introduction

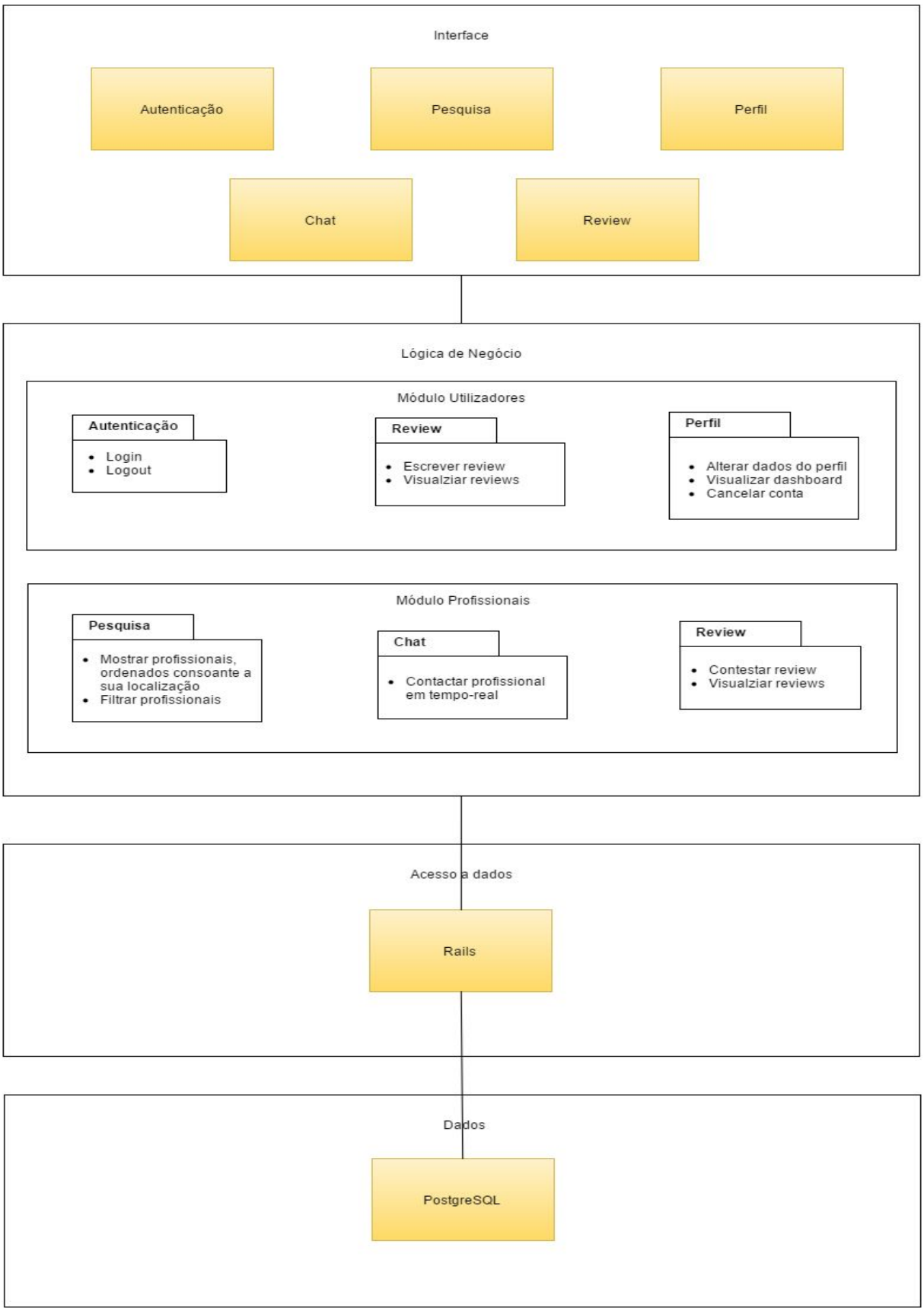
For this project we plan on creating a web platform and the related mobile client. This platform´s aim is to facilitate the communication between those who need a task done and those who are willing to do it. Another main task is to keep the interaction on the platform as simple and intuitive as possible.

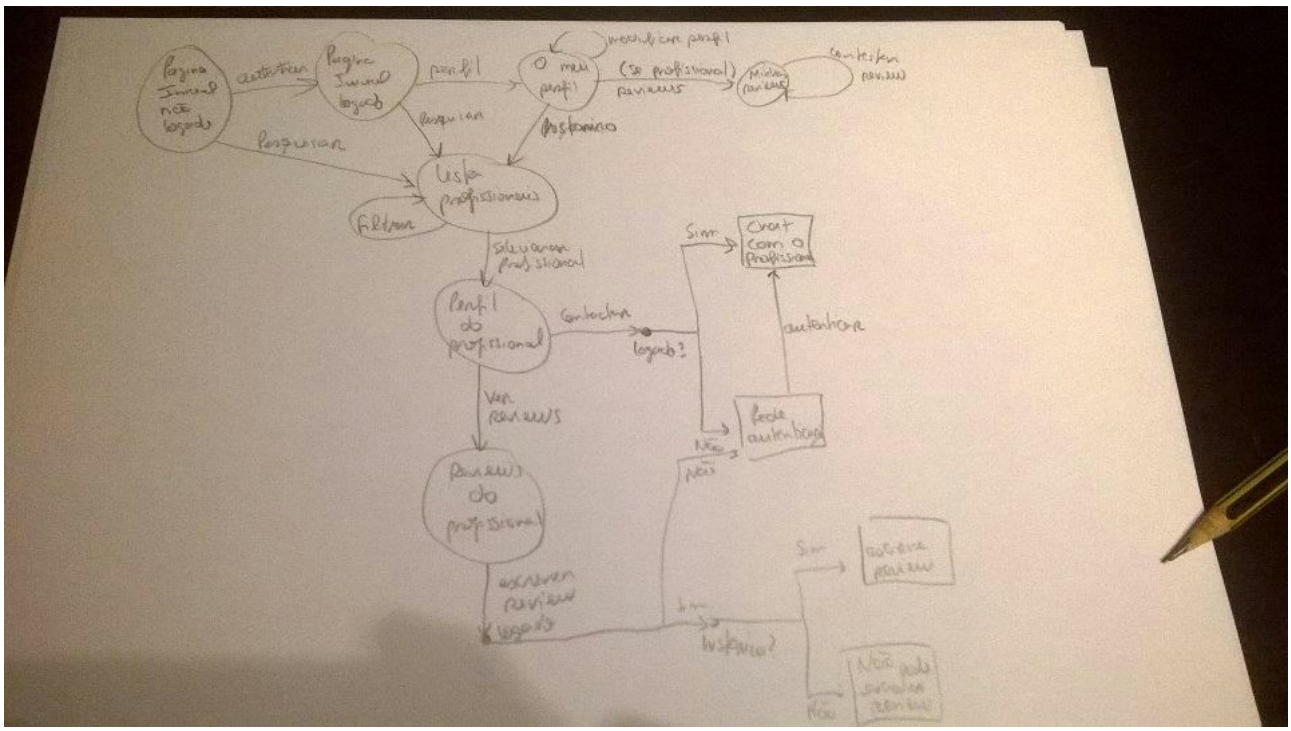
We have structured the report on the following way: Introduction, Technological Architecture, Logical Architecture, Physical Architecture and key design decisions. In the Introduction we describe the aplication to be developed. In the Technological Architecture we describe the technologies selected for this project. Next in the Logical Architecture we focus on the high-level locial structure of the software system. In the Physical Architecture we focus on the high-level physical structure of the software system. And last point is the key design decisions that contains the Software patterns (architectural patterns and design patterns) document proven solutions to recurring problems

Technological Architecture

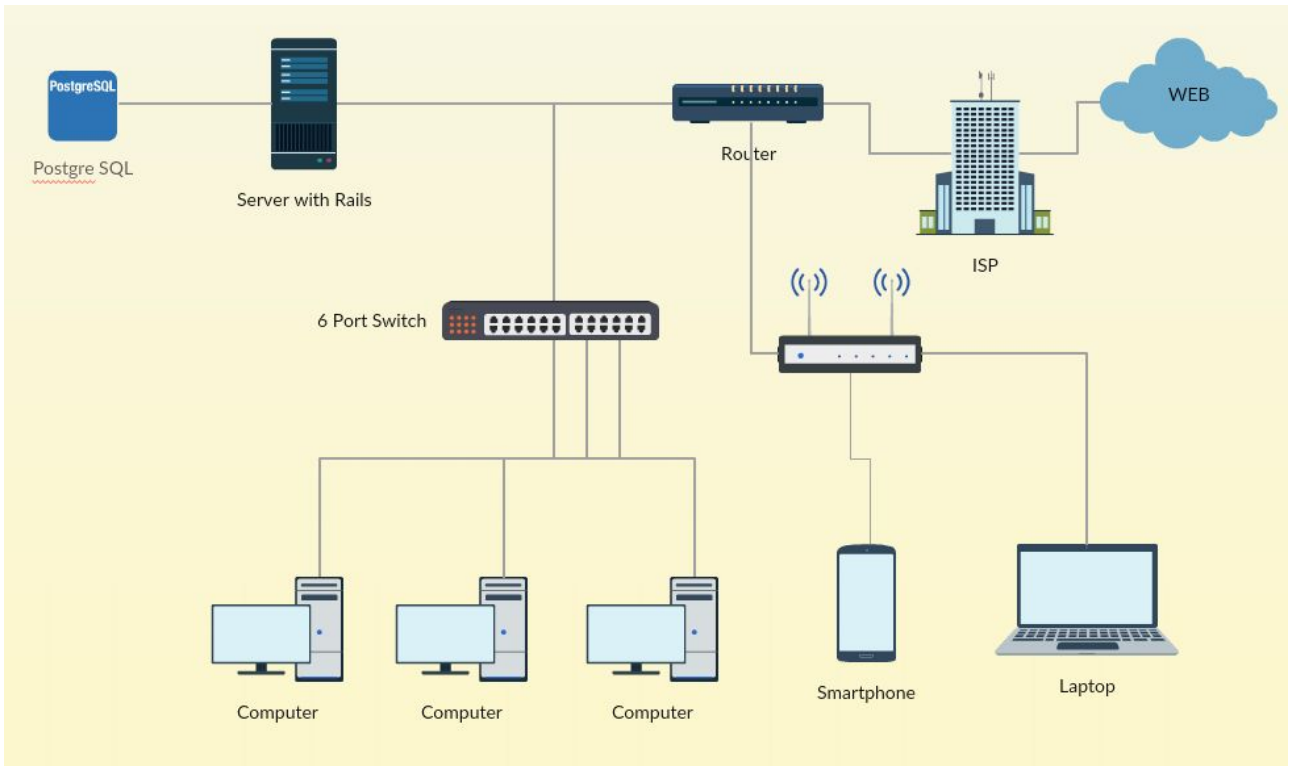
Language	Ruby on Rails 4.2.4
Database	PostgreSQL
Chat Implementation	XMPP
Mobile application	Android
Javascript Framework	AngularJS

Logical Architecture





Physical Architecture



Key design decisions

Regarding authentication and authorization, Rails has a gem called 'devise' that does all the hard work for us. Rails uses a token parameter called authenticity_token that protects against CSRF attacks. For authorization use will use the 'gem' cancancan and rolify. Rolify allow us to add roles to users(admin, regularUsers), while cancancan allow us to define which users can access certain pages. Rails is a MVC framework so we'll opt to use Model-View-Controllers patterns.

In order to store objects persistently in the database we'll use PostgreSQL. The 'pg gem' is an interface to the PostgreSQL RDBMS. We'll take advantage of the Active Record to manage and represent the business data and logic. Active Record will be used as an ORM(Object Relational Mapping) Framework. ORM connects objects of a Rails Application to tables in a relational database system. The advantage is that we avoid to write long SQL queries, altough it may suffer when it comes to performance.

In the frontend we'll use AngularJS. This technology provides us a way to accomplish really powerful things easily. For example, Angular supports two way data-binding that updates the DOM elements instantly. Normally, we would need to update the DOM elements and interpret the user intentions with events, which is difficult when the application grows. Angular also supports 'directives' that can be used to create custom HTML tags, which can be programmed to do actions the programmer wants.