

# Invoice App

Project developed by  
Emanuele Deanesi 185457  
Claudio Bettin 186690

# Invoice App

- ▶ This simple application was implemented using the web-application framework Ruby on Rails. This app was designed for companies which want to keep track of their customers and the related activities and invoices to pay.

# Welcome to the Bill app

This is the home page for the bill application.

[Customers](#)[Activities](#)[Bills](#)

# Description of the system

- ▶ Each customer entity must be filled in with name, surname, company, address, phone number, VAT number and email in order to have a complete representation of the client.
- ▶ When you create an activity you have to choose from a select box, which contains all the fullname of the customers created, the customer you want to relate at the activity. Each activity has a short description, a start date, an end date and the total hours of work.
- ▶ When you want to know the total hours charged for a customer you have to create the bill. As for the activities you have to choose from a select box the customer. Each bill is characterized from the date of issue, the terms of payment, the hourly rate customizable for each customer, the tax rate, the withholding and the discount. Thanks to the implementation of specific methods the app will sum all the hours of works related with the customer and it will provide you the taxable and the total cost to pay. When you change the total hours of works, you add an activity or you delete ones, the app will provide you automatically the new total hours, taxable and total cost. Once the customer paid you for that activities you can label it as paid.

# Technical requirement

In order to implement this app we used:

- ▶ ruby 2.4.2
- ▶ rails 5.1.4
- ▶ Atom

Attached to the app there are two file of user stories. The first was created before starting the project and the second after the creation of the Rails Skeleton App. We chose to write the second file, because we were new using rails and after we started using the framework we realized that the activities, the effort and the priority changed. You can find this file in the doc directory inside the project.

In order to test the app according to the Behaviour Driven Development paradigm, we add the gems:

- ▶ gem 'factory\_bot\_rails'
- ▶ gem 'ffaker'
- ▶ gem 'cucumber-rails'
- ▶ gem 'selenium-webdriver'

We implement the test on the main functions and methods as edit, view, delete entities and the result of the methods.