Technical Report - Project specifications

Beeg Brain

Course: Projeto em Informática

Date: Aveiro, 4 April 2022

Students: 98474: João Reis
98679: João Farias
98388: Ricardo Rodriguez
98390: Mariana Rosa

Project description: This project will be a multi-tenant information system capable of receiving and visualizing EEG files from distinct proveniences, independently of the manufactures. It will also provide tools to manage, visualize and create reports in a web environment.

Index

1. Introduction

Our main goal for the *Projeto em Informática* course is to develop a web platform where it is possible to visualize and manage EEG (electroencephalogram) exams, reaching for the standardization of the EDF (European Data Format) service; this will allow professionals from all over the world to use this platform with EEG exams in the same data format without interoperability problems.

2. Product concept

a. Vision statement

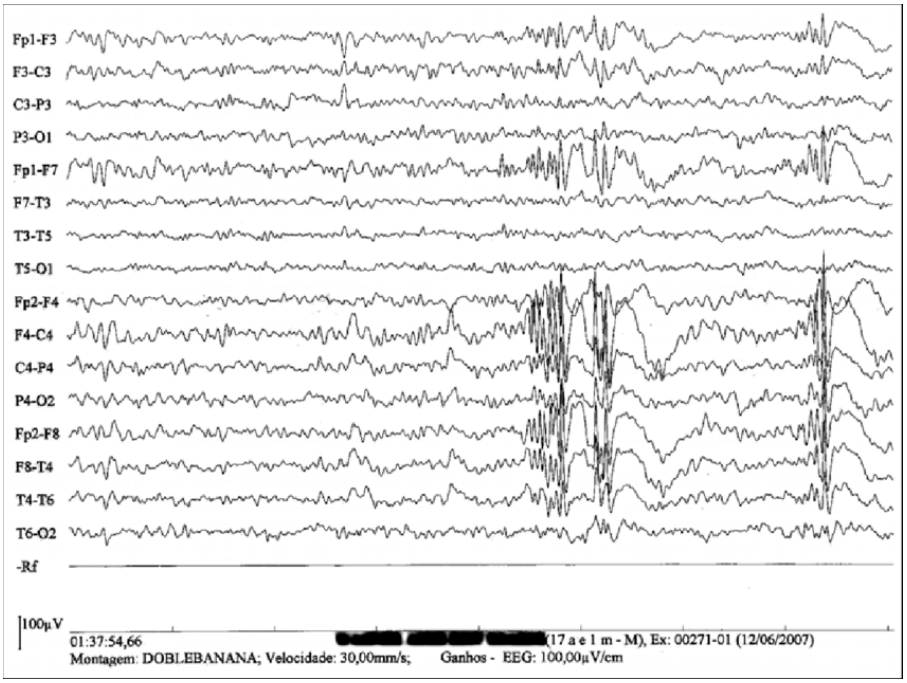


Fig. 1 - EEG example

An EEG is an exam that detects electrical activity in your brain using electrodes (small metal discs) with the main purpose of analyzing brain activity. This activity shows up as wavy lines on an EEG recording (Fig. 1).

However, these exams could have different formats, the main ones are: **EDF**, **BDF** and **GDF**. For this project, we will focus on the EDF (European Data Format).

In every job, collaboration is key. Medicine is not an exception, but sometimes some exams have different data formats from hospital to hospital. This becomes a problem when people from different hospitals want to manage, save or

view the same exam. *BeegBrain* is a platform where all the data is read and written in the same format, so there is not any discrepancy between them.

Therefore, we will develop a multi-tenant system to manage EEG and it will support multiple institutions and multi-users.

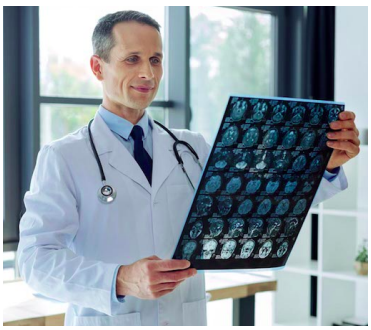
b. Main personas

Alice Torres is a 33 year-old Portuguese neurologist working at the Infante D. Pedro hospital, in Aveiro. She's been married for five years with Vasco Regal and has a 2 year-old son. She's been fascinated with the study of the brain since she was a little child, interested in the diagnostication of brain diseases like multiple sclerosis, epilepsy and Alzheimer's, which affects her grandmother who doesn't recognize her.

Alice has a PhD in neuroscience at the *Faculdade de Medicina de Lisboa*.

Osmond Billie Burns is a 52 years-old British man, living in Berkshire, United Kingdom, a historic county near London. He studied neurology in Oxford for 8 years and right now he is working in London. Since he graduated a lot of things have changed in the technology domain. He's been married since 2000 with Rebecca Jimenez, a very famous cardiologist, and together they had 3 children: Otis, Yasmin and Sienna.

The brain always was a thing that fascinated him, so he decided to follow this career which completely fulfills his life.



Duarte Pereira is a 61 year-old man, born in Coimbra. He's been married to Helena Ramos for 36 years and both of them have 2 kids: Hugo and Jéssica. He studied in *Faculdade de Medicina da Universidade de Coimbra*, where he graduated with a Doctorate Degree in Medicine.

Since he started working at *CHUC*, Coimbra's main hospital, he's been observing the evolution of neuroscience and started diverging to that area, where he currently works as a Lead Neuroscientist. He and his team are studying how the brain constructs dreams.

c. Main Scenarios

- Alice Torres has a patient who is sent from Infante D. Pedro hospital to Coimbra University hospital, who uses the **european data format (EDF)** format to represent the EEG **exam**. In order to solve this problem, when Alice arrives in Coimbra, the neurologist doctor uses this system to submit the EEG file done in Infante D. Pedro hospital and the file will be converted in **EDF** format;

- Osmond wants to check an **EEG** exam while he is writing a report. So, he uploads the exam in the application, sees the exam and can write important reports about what he saw. He has intentions of generating a PDF directly on the platform with all the important information gathered in that exam.

- Duarte is a very experienced doctor and he has done thousands of exams in his whole life. Imagining that he has access to all of the exams of his institution, Duarte wants to search for a certain patient to study his case and see how he's going.

3. Architecture notebook

a. Key requirements and constraints

The application should be able to support distributed service requests, in a multi-user and multi institutional situation.

Functional Requirements

- The system should be a centralized cloud-based platform;
- The system must have a RESTful API, in order to upload files and do other important tasks;
- A user should be able to monitor a folder or a set of folders;
- There will be a shared folder, associated with an institution, where the user can upload/drag the files into it and an automatic process will load the new files to the RESTful API;
- The user should also be able to load files inside the application, with the help of an upload button or a drag and drop area.
- The file processing area will be divided in two workspaces: Transferred area (with the files that have been successfully transfered) and the Error area (with the ones that weren't able to parse or sent);
- In the Error area, it will be displayed not only the exams with parsing errors but also with other error types. The user should be notified of these errors to solve these conflicts manually.
- The files in the shared folder will be kept in a temporary cache for 7 days;
- Standardization of EDF Service: a service will be able to receive a file (EEG) in a proprietary format and convert it to a standard *de facto* EDF;
- The user should be able to visualize, monitor and track EEG;
- The system should be able to deal with signal display in an easy and intuitive manner for exam review;
- The platform should be able to generate a PDF document with the content written by the doctor;
- The application should have a monitoring area, where the metrics about patients, EEGs and other different entities are displayed for statistical purposes.

Non-Functional Requirements

- The application EEG viewer should have high performance;
- The system must be scalable, allowing the use of all application functionalities for various users at the same time, without sacrificing performance;
- The EDF files are usually files that vary between 50 to 100Mb;
- EEG visualizer tools must be easy for the user to use and understand its functionalities;

- EEG files must be converted to the standard EDF (European Data Format) type;
- EEG files should be stored in subparts in the database when uploaded and these parts should be assembled into one single EEG unit when requested for visualization;
- EEG files should be compressed when uploaded to the database, in order to be space efficient;
- When the system fails during a EEG upload, the costumer should be notified of that event;
- The system should have a 2 factor authentication or a Single Sign-On, allowing the user, associated with a medical institution, to be authenticated;
- The system must encrypt sensitive data about doctors and patients for security prevention;
- The system must track events related to the life cycle of an EEG (e.g. upload, delete, PDF generation…) and associated it with an owner (doctor or an automatic service) and a timestamp;
- EEG deletion must be double confirmed by the user, in order to avoid;
- Deleted EEGs must reside 1 week in the database for possible restoration of the file, being permanently deleted afterwards;
- Users can have two different type of access to a file: read or read+write;
- EEG access invitation should be confirmed by the invited doctor before giving him the access to the file;
- When generating a PDF, the user should be able to add/remove sections in the text editor and format them the way he likes;

b. Architectural view

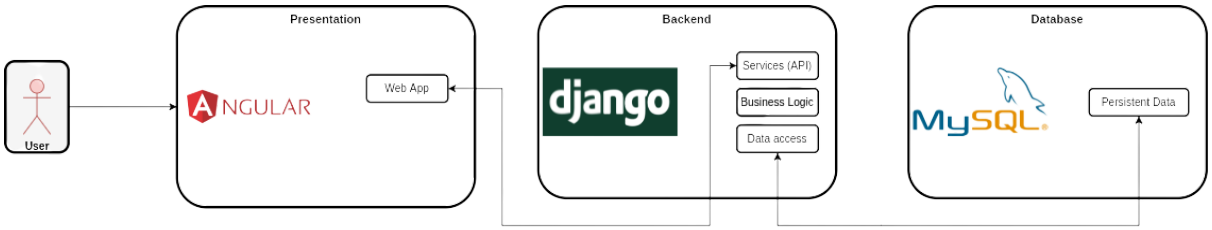


Fig. 2 - Project's architecture

Our project will be developed with Angular Framework for the presentation, Django for the backend (including the construction of the API) and our database will

be created with MySQL, which is supported by **django**.

The frontend module should be able to interact with the users by displaying the EEG data in the browser, allowing users to upload files in several ways and many other functionalities that require user input. These events should be registered in a logging system and sent to the backend module.

The backend module should receive API requests triggered by user events, process them and store/retrieve information from the database.

The database should persist all structured information critical for the system to work (e.g. entities and relations between them). This module is explained in detail in chapter 4.

c. Deployment view

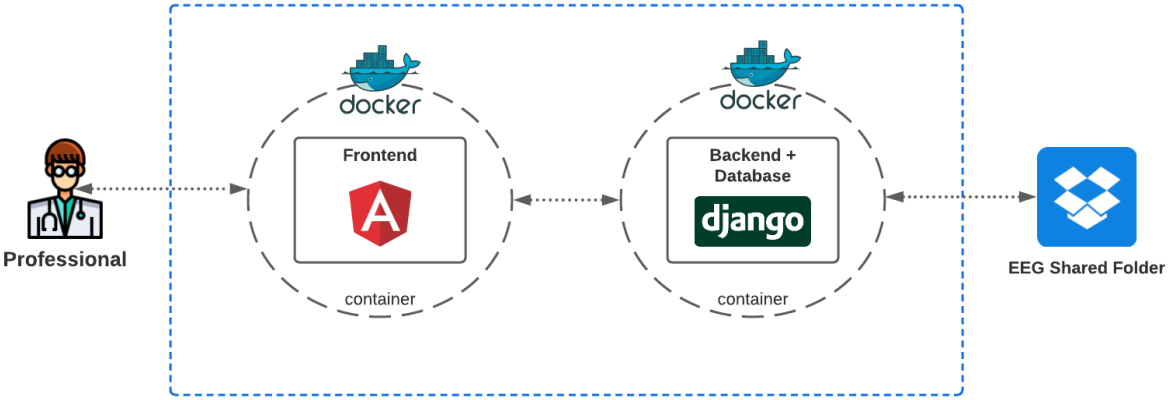


Fig. 3 - Deployment diagram

As we explained in the previous chapter, the doctor is going to access our application through a web page, developed in **Angular**, which is going to communicate with the backend, developed in **django**. These two components will be deployed separately using **docker containers**. In the backend module is where the system will communicate with the EEG files shared folder.

4. Information model

This diagram represents the database of our application. In it we show our entities, the relationships between them and their fields.

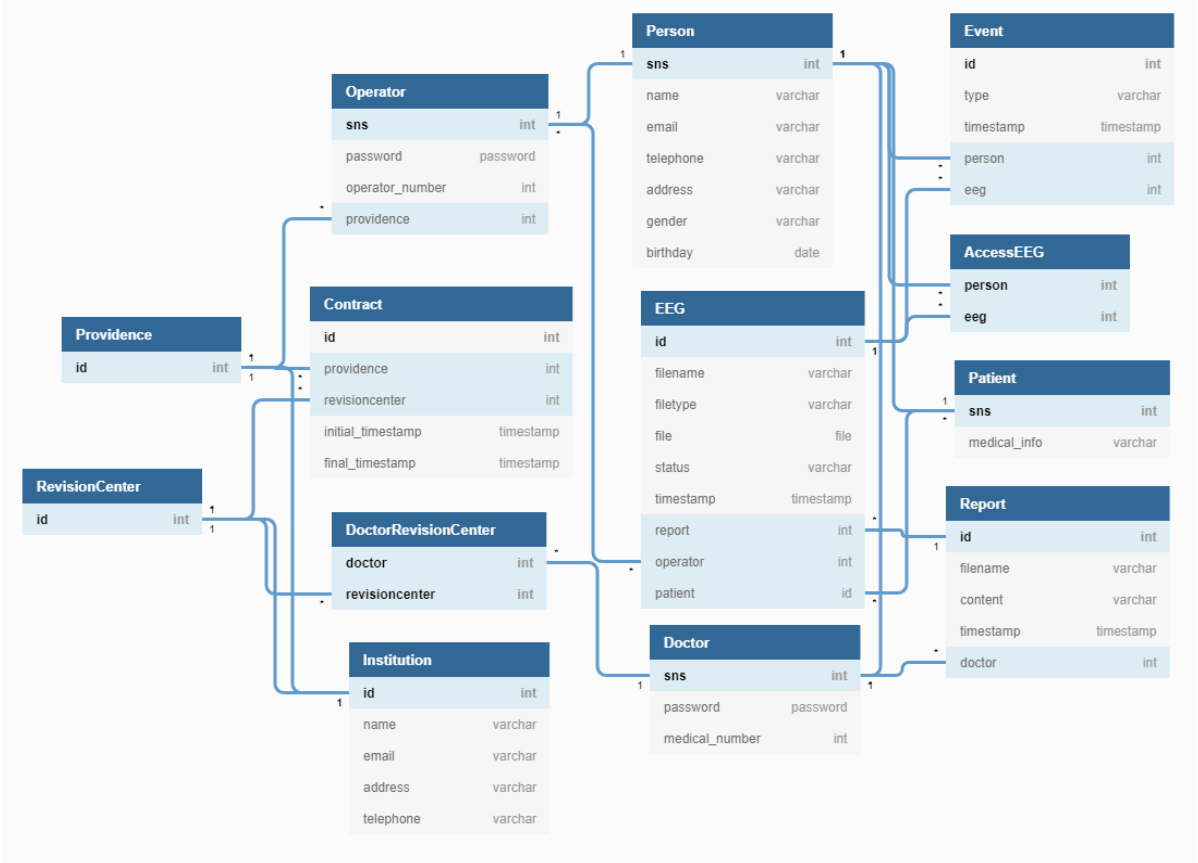


Fig. 4 - EER diagram

Important notes:

- EEG's access attribute has a list of the doctors who have access to that specific exam, no one else have the right to read/edit them;
- The owners of the file can be a doctor or an automatic procedure;
- Events is like a log file that has regist of events. For example, a doctor that sees an EEG and generates a PDF, it's an **Event**.
- A **Contract** is shared between a providence (institutions where EEGs are produced) and a revision center (institutions where EEGs are reviewed) and they are mutually exclusive, meaning that a providence can only work with a revision center and vice versa.

5. Conclusion

To conclude this report, we think that we explained our system, how it's going to work and all goals we want to reach. The calendar and more detailed information is in the links below. With this work we hope to help patients, doctors and institutions' lives towards contributing to the world of medicine.

6. Useful links

Our GitHub Repository: Github project

Our Website: Project Website

Our Calendar: Project Calendar
Tasks for each team member: Each member tasks