# 1 - INTRODUCTION

This document constitutes the Design Document (DD). Its purpose is to analyze the design of the software.

It is divided in 5 section:

* 'Introduction' to give a generic document overview
* 'Components Breakdown' where each single component of the whole system is described
* 'Components Tasks' in which for every component are discussed their own operations.
* 'Components sequence diagram', in this chapter are present the sequence diagrams of the components interactions
* 'Simplification for code' to list all the simplifications adopted to have easy code to execute.

1.1 Design overview

The system adopted is a distributed one, the choice is due to the interest in creating a project that can be as similar as possible to a real project.

The components of the entire system are:

* user interface (front end)
* back end
* database
* optimization algorithm

# 2 - COMPONENTS BREAKDOWN

Here, for every component is described the motivation of the choice to have it, the programming language (if needed)

## 2.1 - Front end

The front end (user interface) is a web app developed with the reactJS framework, the choice is done because it is a very popular framework and in addition web apps are extremely diffuse nowadays, so these choices are done in order to use tools useful in the world of work.

## 2.2 - Back end

The back end is written with the programming language python, since the optimizer algorithm uses libraries that are very popular in optimization.

## 2.3 - Database

The database is a relational one, in particular the adopted one is MySQL, the choice is due to the interest in learning how to use a relational database and being MySQL one of the most used it could be a good tool to learn.

## 2.4 - Optimization Algorithm

The optimization algorithm is written with the coding language Python, the choice is done because Python offers a large number of functions to solve optimization problems.

The following libraries are used to solve the problem

# 3 - COMPONENTS TASKS

Each component has a specific task indeed it is developed in a custom way. In this chapter for each component are listed the operations that it has to do and in the next chapter they are analyzed in detail with a sequence diagram.

## 3.1 - Front end

The front end (user interface) lets the registrar's office personnel insert input data in the database and, once the settings are set, call the optimization algorithm with custom data.

This section has to:

* Allow user to insert starting and ending date for the exam session.
* Allow user to insert unavailability for a specific professor, and his/her relative courses
* Allows user to define other specification for the exam session such as:
  + Minimum distance between exams
  + Minimum distance between calls of exam
  + Number of calls for each session

All this inputs will be part of the custom data settings and sends to the main component in the backend.

At the end the output will be visible on the main page.

## 3.2 - Back end

The back end takes the data entered in the web app interface and inserts them in the database, it also establishes the connection with the database and calls the optimization algorithm when a specific button of the user interface is clicked.

The backend will firstly retrieve the static data about exams from the database (course, associated professors, sections, CFU, average number of students, etc…) and prepare them for the optimization part, building the weight parameters for each exam.

Then it calls the optimization algorithm iteratively for each programme (corso di studi) and can updates the capacity of each day. After each iteration of the algorithm it stores the result data. If a course is common in more than a programme, and it’s been assigned in the previous iteration, the associated date variable will be set.

At the end the output is produce as an excel file, divided by the programme and easy to read.

## 3.3 - Database

The MySQL database is used as data storage. For each PoliMi course the following data are stored: course name, professor name……

## 3.4 - Optimization Algorithm

The optimization algorithm solves the main problem that is the exam scheduling of the session FOR EACH PROGRAMME,

# 4 - COMPONENTS SEQUENCE DIAGRAM

In this section are present the sequence diagram for interactions of each component

## 4.1 - Front end

Is divided in two main pages: *Home* and *AddUnavailabilty*

* *Home* features and functions
  + Insert start-end date
  + Insert new unavailability
  + Modify Session Settings
  + Start

## 4.2 - Back end

## 4.3 - Database

## 4.4 - Optimization Algorithm

# 5 - SIMPLIFICATION FOR CODE

In order to develop a software that can work in a small amount of time and can find a feasible solution some simplifications are done.

In particular, it has been considered only *“x number”* of days available to schedule exams, *“y number”* of exams to schedule and an infinite classroom capacity. In this way the it is possible to find a feasible solution and it takes *“z”* minutes

The algorithm can be implemented with

# 