DETI - 2022/2023

MepML

Plataforma de avaliação de modelos ML/AI

Diogo Oliveira Magalhães	102470
Emanuel Gaspar Marques	102565
Leonardo Almeida	102536
Pedro Henrique Figueiredo Rodrigues	102778
Rafael Fernandes Gonçalves	102534

Projeto em Informática

Supervisor: Prof. Mário Antunes

May 30, 2023

Abstract

The need to handle large amounts of data and advances in processing technologies have led to the mass development of intelligent systems. Machine Learning algorithms are often applied to optimise various real-life scenarios, leading to cost savings and increased productivity. Therefore, it is imperative to promote better education in this field of study.

The University of Aveiro has been integrating Artificial Intelligence topics in several areas, such as Informatics, Electronics, Telecommunications, and Mathematics. As part of their coursework, students are faced with exercises that require Machine Learning models to predict a target variable, based on a set of features. Unlike traditional programming, they already have access not only to a large set of input data but also to the corresponding outputs. The assessment of the their work though is still done manually, which is a time-consuming process, especially with a large number of students. In addition, it is also difficult to track the progress of all students.

There are some platforms dedicated to sharing datasets and Machine Learning problems: regressions, classifications and more. However, they have some aspects that limit their applicability in the university environment, namely they lack support for assigning exercises to a restricted group of students, importing classes from a spreadsheet and adding and further reusing of new metrics for model performance evaluation. In addition, there is no clear distinction between the roles of student and teacher as actors in the systems.

We took these shortcomings as an opportunity to create a valuable alternative, targeted at the teachers and students of the University of Aveiro who have to deal with this kind of problems. Hence, we developed a multi-layered web application, where teachers can create and manage exercises, import and manage classes, track the progress of the assignees, and add new evaluation metrics, while students can view and solve exercises, get quick feedback on their models and rank themselves among their peers.

The project demanded much of the knowledge acquired during the bachelor's degree, enforcing an iterative and incremental development approach, and the adoption of good software engineering practices, such as quality assurance testing. So, we believe that the proposed software solution will be a good asset for the university, contributing to the effectiveness of part of the learning process in some courses.

Keywords: Machine Learning, Artificial Intelligence, Data Science, Education, Web Application, RESTful API, Model Evaluation, Performance Metrics

Acknowledgements

We would like to thank:

- Our supervisor, Professor Mário Luís Pinto Antunes, for his guidance and support throughout the project.
- Professor Petia Georgieva, for her availability and invaluable support and collaboration during the usability testing phase.
- "Complements of Machine Learning" students for their willingness to devote their time and energy to the usability testing session that provided us with insights into the user experience and identified aspects where we could make enhancements.
- The conductors of "Projeto em Informática", Professor Osvaldo and Professor Rui Aguiar for their accurate positive criticisms.

On behalf of the entire team

"Nobody does great software alone but a team working together can do extraordinary things."

by OpenUP

Contents

List of Figures

List of Tables

Abbreviations

UA	*Universidade de Aveiro*
ML	*Machine Learning*
AI	*Artificial Intelligence*
IT	*Information Technology*
ICT	*Information and Communications Technology*
IdP	*Identity Provider*
GCP	*Google Cloud Platform*
API	*Application Programming Interface*
GPU	*Graphics Processing Unit*
IDE	*Integrated Development Environment*
PACO	*Portal Académico Online*

Chapter 1

Introduction

The last few decades have been marked by a rapid and relentless technological evolution [1]. Where once access to the Internet and smart devices was a privilege, we now live in an era where everyone and everything is connected. As a result, there is a huge amount of data that needs to be processed in order to optimise processes and operations in a wide range of real-world scenarios. Meanwhile, the increase in computing resources (e.g. cloud computing), has made it easier to collect, store and process information. These technological paradigm shifts have driven areas such as AI, especially ML.

When we had to choose a theme for the Project in Informatics, we took into account the market trends. One of our main sources was Fortune Business Insights, a company that Google, EY, Huawei and many others rely on for market research reports and consulting services. According to their public report FBI102226 [2], the global ML market size was valued at $19.20 billion in 2022 and is projected to reach $225.91 billion by 2030, exhibiting a CAGR (Compound Annual Growth Rate) of 36.2% over the forecast period. To explain this expected growth, the report highlights some recent applications of ML:

- During the pandemic, Massachusetts Institute of Technology (MIT) researchers used ML to predict the spread of COVID-19 in different countries.
- Many e-commerce businesses (e.g. Amazon, Alibaba and eBay) predict customer behaviour to recommend products - this has prooven to be an effective strategy to increase sales.
- IBM Watson for Genomics help oncologists deliver customised cancer treatments to patients, inspecting genomic sequencing data.
- Berg Health, a Boston-based pharmaceutical startup, uses AI to produce drugs targeting cancer, diabetes, and neurological diseases.

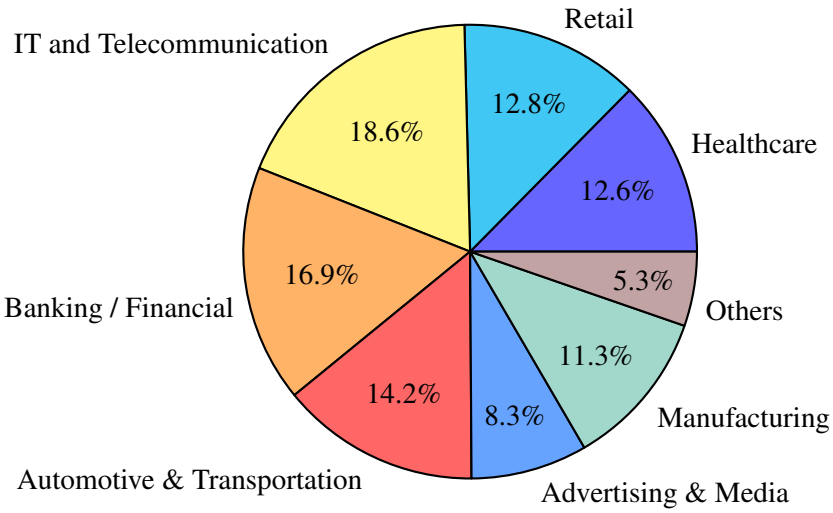

Figure 1.1: Global ML market share per end-use industry (2022)
(Adapted from: *Business Fortune Insights)*

Fig. 1.1 shows the distribution of the ML market by industry. As we can see, very different sectors are investing in this technology in order to extract real-time insights from data.

If we look at the whole AI market, the numbers are even more impressive. According to the FBI100114 report [3], the global AI market size will increase from $428.00 billion in 2022 to $2025.12 billion by 2030. The game changer was indeed OpenAI's ChatGPT (Generative Pre-Trained Transformer) built on top of Transformer, a deep learning model introduced by Google in 2017 that can process entire sequences of data in natural language at once. The reception of this tool has been so enthusiastic that hyperscalers have been responding with many other powerful solutions. Despite this, inaccuracy is still a major issue with ML models, which demonstrates the need for human intervention, for example, with reinforcement learning.

Observing the market segmentation, we also concluded that ML and AI are expected to be the key technologies for small and medium-sized enterprises (SMEs) in the coming years, as they will give them a competitive advantage by automating processes, while investing less in ICT. We can easily confirm this by looking at the job offers in the IT sector. The demand for skilled professionals in these fields is very high. All these factors led us to believe that promoting a better ML/AI education is thinking ahead. Preparing the students - including us - for the upcoming challenges is a must. So, we decided to create an intuitive and collaborative space for students to practice ML with problems proposed by their teachers.

1.1 Context

This project was developed by a team of 5 people, as part of the Project in Informatics course of the Bachelor's Degree in Informatics Engineering at the University of Aveiro. During one semester, we had to develop a software solution to a problem of our choice, strengthening our documentation, communication and application demonstration skills. We accepted the proposal

of Professor Mário Antunes, to develop a ML/AI model evaluation system, based on performance metrics provided by teachers.

Looking at the usual ML workflow (Fig. 1.2), we realise that it differs from Traditional Programming in the sense that we already have access to a lot of input data and the corresponding outputs, and the model found by the engineer encapsulates the relationship/pattern that maps the inputs to the outputs. The purpose of our platform, named MepML, is to evaluate the performance of the models, either in terms of precision, accuracy or any other metric that teachers want to apply. We emphasise that the models submitted by students are trained by them beforehand. Then, they simply upload the resulting prediction dataset to our system, which performs a quick statistical analysis and returns the metrics scores.

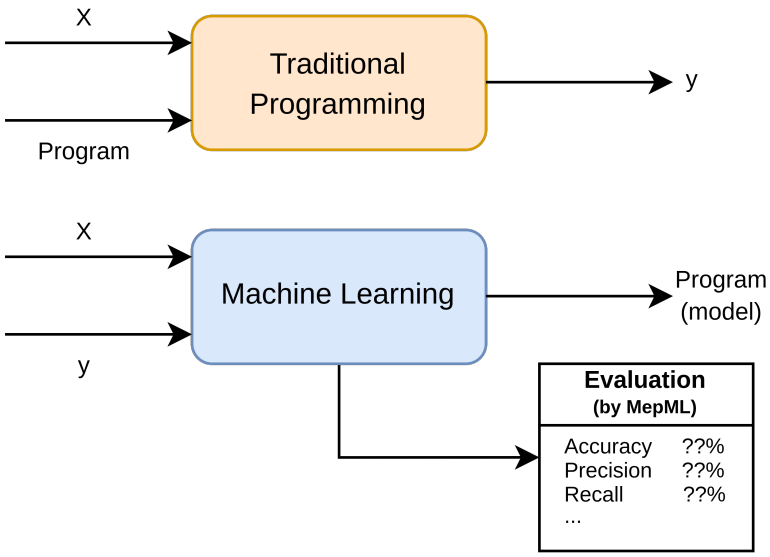

Figure 1.2: Process reengineering with MepML

1.2 Main Motivation

The main problem that motivated the start of this project was the complexity associated with the evaluation of ML models. Nevertheless, other difficulties arise from this complexity:

- It is difficult to keep track of each student's progress.
- Students don't have a clear idea of how good their models are.
- Teachers take a long time grading the students' work.

1.3 Expected Results

Table 1.1: Description of expected results

Integration with UA IdP	Authentication using UA IdP since target users are UA people.
Grouping students	Teachers should be able to create, modify and delete classes.
Exercises creation	Teachers can upload datasets and create exercises with limited attempts, deadlines and specific evaluation methods.
Students engagement	Students may access proposed exercices and submit a solution to them and compare results with other students.

1.4 Document structure

This document has 7 other chapters, totalling 8 with the introduction. In chapter 2, we describe the state of the art, where we present existing solutions and its problems, related papers and used technologies. In chapter 3, we summarise the process to define the product concept, mainly our requirements gathering techniques. Chapter 4 delves into the system architecture, describing the domain, the architecture design and the deployment. We thought it was important to dedicate a chapter to the methodologies adopted by the team, so we detail them in chapter 6. In chapter 5, we describe the implementation of the system, divided into 4 layers: presentation, authentication, business and database. In chapter 7, we present the methods that we used to ensure the quality of the product. Finally, in chapter 8, we reflect on the results obtained, the limitations of the project and the future work.

Chapter 2

State-of-the-Art

2.1 Introduction

State of the Art is the current highest level of development, achievement, or advancement in a particular field. Staying updated with the state of the art of that field is crucial as it drives innovation, improves model performance, and guides research.
Machine Learning (ML) is a rapidly growing field with endless possibilities, and one of the most exciting and needed developments in recent years [2]. The objective of this project is to develop a ML educational platform that helps professors and students communicate and improve the education levels.

2.2 Related Work

In order to have better knowledge of the work that has been done and developed in the field of machine learning education and competitions, we started by searching the most popular platforms in the market that were similar to our objective as a platform. We found some platforms like Kaggle, Weights and Bias, Codalab and MachineHack and proceeded to analyse them in order to find the best features and technologies that we could implement in our platform and where those platforms failed.

2.2.1 Kaggle

Kaggle is a widely recognised and popular platform mostly used to host and participate in ML competitions where participants can compete to build the best predictive models for specific problems, it also offers a rich ecosystem of datasets, collaborative tools, such as a customisable Jupyter Notebooks environment and GPUs access at no cost, and resources that enable individuals and teams to explore, learn, and solve real-world data challenges.
Its limited options and lack of privacy make it a bad option for education purposes.

2.2.2 Weights and Bias

Weights and Biases is a ML platform for developers that helps them to build better models faster. It offers tools and infrastructure to facilitate the development, training, and analysis of ML models to help developers effectively manage and monitor their ML experiments. It allows collaboration between team members, experiment tracking, which allows users to log and record essential information about their experiments and offers a dashboard where users can visualise and compare their experiments and results.

2.2.3 Codalab

CodaLab is an open-source platform that provides an ecosystem for conducting computational research in a more efficient, reproducible, and collaborative manner.
It is an application that surprised us, due to having most of differentiating features that we had thought about, but it was still limited due to its narrow integration with external systems and bad overall appearance and user-experience.

2.2.4 MachineHack

MachineHack is a data science platform with a focus on hosting ML competitions centring their attention on professionals in the area.
MachineHack offers a wide range of data-related challenges and competitions, allowing data professionals to test and highlight their skills and abilities and learn from others in the community.
This platform has a big flaw, regarding the overall goal of our project: it was not purposely developed for students and professors. Furthermore, it does not allow a professor to create a new exercise and assign it to a student, since only registered companies can do it, even though they cannot assign it to a specific user/class.

2.2.5 Conclusions

After the analysis we came to the conclusion that all the platforms analysed failed to provide a good educational experience to the students and that the main focus was on the competitions and not on the education. As we can see on the table 2.1, where each platform functionalities are compared, even though all the platforms failed in our main objective, we found some interesting features and designs that we also took as inspiration for our platform.

Table 2.1: Comparison between platforms

Requirement	Kaggle	Codalab	Weights and Biases	Machine Hack	MepML
Professor can create exercises and students can join them	✓	✓	X	X	✓
Define exercise visibility	✓	✓	X	X	✓
Exercise Leaderboard	✓	✓	X	✓	✓
Define exercise deadline	✓	✓	X	✓	✓
Define exercise maximum number of tries	X	✓	X	X	✓
Add/remove/import students to a restricted group	X	X	X	X	✓
Add new and reuse Metrics	X	X	X	X	✓
Use UA IdP to authenticate students and professors	X	X	X	X	✓
Check assigned exercises	✓	✓	✓	✓	✓
Download exercise related content	✓	✓	✓	✓	✓
View Students Code and results	✓	X	X	X	✓
Students and professor user types	X	X	X	X	✓

2.3 Metrics

To evaluate ML models are used metrics. Since there are a lot of evaluation metrics, we needed to find out which ones were the best and most used, so that those could be quickly included (with maximum priority) in our platform and be chosen by teachers to evaluate students' work.
The authors of the paper "A critical analysis of metrics used for measuring progress in artificial intelligence" [4] presented an analysis of which were the most used metrics in Machine Learning. The study took in consideration 3883 papers, where 2998 distinct datasets were used and where 32209 benchmarks were made. In those benchmarks, 812 distinct metrics were found which becomes a very hard task to manage and implement all of them in our application.

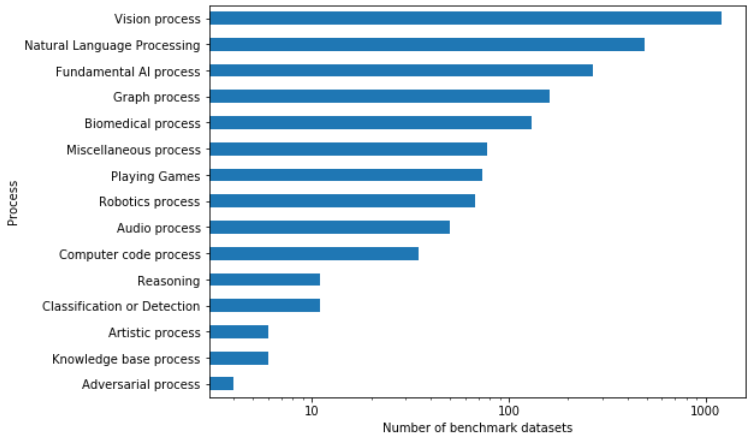

Figure 2.1: Number of benchmark datasets per higher level process

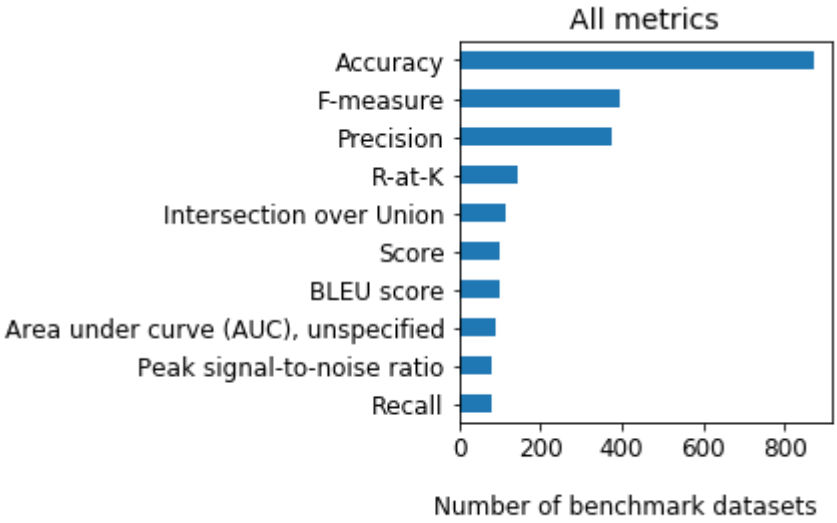

Figure 2.2: Top 10 most frequently reported performance metrics

The most used datasets process types and the most frequently utilised evaluation metrics were comprehended after the analysis of this paper as it is possible to visualise in Figure 2.1 and Figure 2.2, respectively. The significance of this study lies in the fact that it has provided us with the knowledge of where our focus and prioritisation should be when implementing the evaluation metrics in our system.

To gather more information two stakeholders were identified, Professor Pétia Georgieva and Professor Mário Antunes. With their help, we got to know which metrics are the most used in the University of Aveiro and in their ML/AI classes, since students and professors are the main target of our platform.

As an advantage our platform allows teachers to add new evaluation metrics out of python scripts and reuse them later.

2.4 Technologies

To make our platform, many technologies and frameworks needed to be analysed and discussed in order to assure that the end user has the best user-experience and the platform the maximum performance.
We came to the conclusion that the best technologies for our platform would be:

2.4.1 Frontend - React

React is a popular JavaScript library created by Facebook used for developing user interfaces and related components. React has a component-based architecture and its efficient rendering makes it an ideal choice for building dynamic and interactive user interfaces. React allows developers to build large web applications that can alter data without requiring a page reload. The fundamental goal of React is to be quick, scalable, and simple.
React is currently the most popular frontend framework and according to aceinfoway [5] 75.4% of web development companies, and agencies have reported to be specialised in the framework.

2.4.2 Backend - Django

Django REST framework is a powerful and flexible Python web framework used for building Web APIs and it will be used in the backend of our platform. Django encourages fast development and clean, pragmatic design and it's based on the Don't Repeat Yourself (DRY) principle, focusing on code reusability. Django allows an easy way of connecting to many different databases like SQLite, MySQL, Oracle, etc.. Django takes security seriously and helps developers avoid many common security mistakes. It is one of the most popular backend frameworks and according to codingnomads [6], the framework is today the second most popular backend framework.

2.4.3 Database - MySQL

MySQL is a Database Management System (DBMS) that uses SQL Language as its interface to access and manipulate data. MySQL is a free and open-source software and it is ideal for both small and large applications. MySQL is a relational database, that is, a database where data is stored in tables and tables are related to each other by keys. MySQL is one of the most popular databases and according to statisticsanddata [7] it ranks second.

2.4.4 Containerisation - Docker

Docker is a software framework that helps developers build, share, manage and run applications in containers. Docker's containerisation technology provides a lightweight and portable way to package software with all of its dependencies and run it consistently across different environments that makes it easy to deploy and scale applications in different environments.
Compared to virtualisation, containerisation is more lightweight, requires less effort to deploy, run and manage which makes it a great choice for continuous integration and continuous delivery (CI/CD) pipelines and to improve productivity and efficiency.

Chapter 3

Product Concept

3.1 Requirements elicitation

Requirements outline the needs of stakeholders and what the system must include to meet them. In this section, we describe our requirements gathering techniques based on OpenUP model [8].

3.1.1 Talk to domain experts

Our supervisor and stakeholder, Professor Mário Antunes, explained us what he wanted from the project, especially from a teacher's point of view. He also gave us some examples of exercises that he usually proposes to his students. This was very important to understand the domain and the inherent problem. From our very first meetings, we got the following requirements:

R1: integrate UA IdP for authentication
R2: two profiles: student and professor
R3: the students are expected to:
- 3.1: consult the exercises (public and private)
- 3.2: download train and incomplete test dataset (without prediction)
- 3.3: upload results
- 3.4: view the results table (comparison with other students)

R4: the professor is expected to:
- 4.1: set/add metrics
- 4.2: define classes (group of students)
- 4.3: add/remove/import students to the class
- 4.4: define exercises and its visibility
- 4.5: set deadline and/or retry limit
- 4.6: access student results/code

"Tópicos de Aprendizagem Automática" teacher was informally interviewed and has allowed us to use our own classes as a source of continuous feedback on students' needs. As we will show

later in this document (Section 7.3), we were able to carry out usability tests with "Complementos de Aprendizagem Automática" students, who are more experienced in ML.

3.1.2 Mockups

Mockups play a crucial role in the process of requirements elicitation, serving as powerful tools for communication and understanding what the stakeholders expect from the system. They provide visual representations of proposed user interfaces, workflows, and system behavior, allowing stakeholders to visualize and evaluate the proposed solution.

In this project we developed several mockups, not only to help us understand the requirements of the system but also to get early feedback from the stakeholders so that we could minimize the risk of developing a system that does not meet their expectations. The mockups that we developed served as a visual blueprint, providing a clear and concrete reference for implementation.

Even though there are several tools that allow us to create mockups, we decided to use Figma, which is a powerful prototyping tool that allows us to create mockups and prototypes with a high degree of fidelity. The main reason for choosing this tool was the fact that it is free for university students and allows us to work collaboratively, which is extremely important in a project with several team members.

Figures 3.1 and 3.2 show two of the pages that we mocked up, for students and professors, respectively.

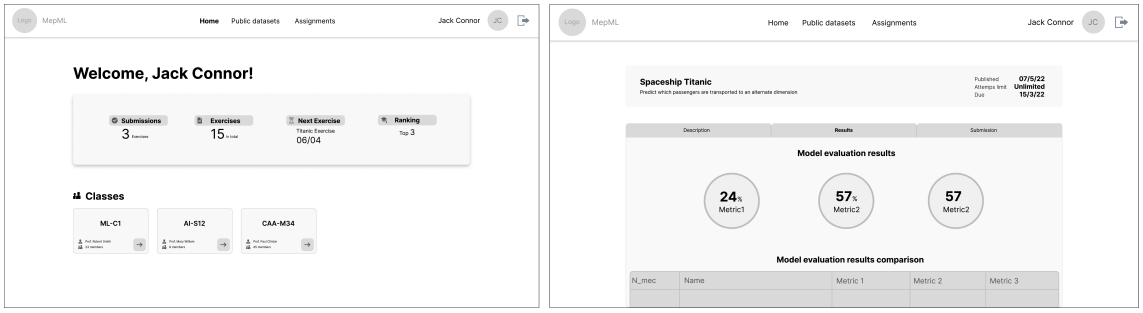

Figure 3.1: Student Homepage and Model Results

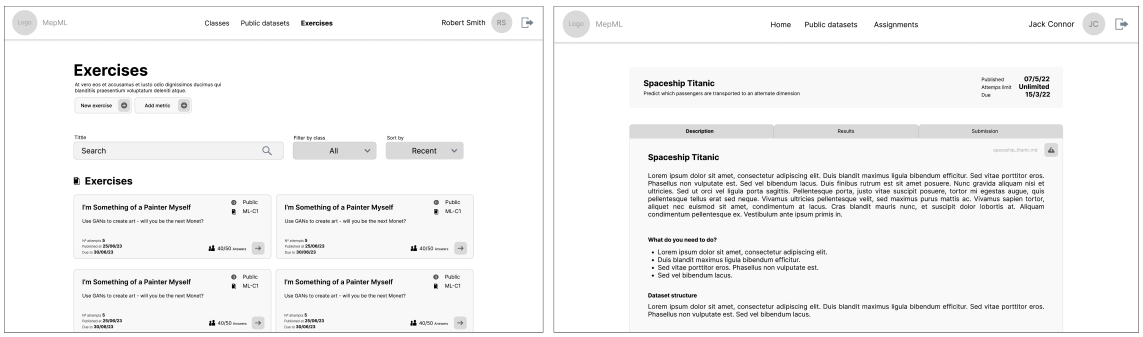

Figure 3.2: Professor's Exercises and Exercise Details

3.1.3 Brainstorming

Since the beginning of the project, the team intended to proof to be proactive and to come up with ideas to improve the product. We did not want to just do what was asked of us, but to go further and try to innovate. We had several brainstorming sessions, some on a private Discord server and others with our supervisor, where we discussed ideas and tried to find solutions to the problems we were facing. A lot of ideas went through our heads, but some of them were not feasible and others would not add value to the product. We ended up with one big idea: a Slack bot that notifies students of new assignments. This **extra feature** was added at the end of the project, after we had met all the other requirements.

Even though it is a separate module and adds more complexity to the system, it also adds additional value to the users. With it, professors don't have to manually notify their students about new assignments. Our platform does that automatically, saving their time and effort. Furthermore, students don't have to check the system manually, which makes it easier for them to stay up-to-date with their assignments.

R5: notify students of new assignments on Slack

3.1.4 Personas and main scenarios

When following an agile software development approach, requirements gathering sometimes entails the creation of personas: fictional entities that represent a certain group of end-users of the system, having realistic characteristics, context, motivations and goals.

Persona 1: Professor

Table 3.1: Daniel Santos' profile

Name	Daniel Santos
Age	35 years
Sex	Male
Job title	Associate Professor
Institution	University of Aveiro
Civil status	Married
Location	Ílhavo, Aveiro, Portugal

Background: Daniel loves programming. He has a PhD in Data Science and his colleagues describe him as a highly methodical and detail-oriented individual. When faced with large data sets, he can easily find the best models to make traditional processes more efficient. In order to assess practical works, Daniel has to run a script with performance metrics for each student, which is time-consuming and error-prone.

Goals: Daniel would like to automate this evaluation procedure, so that he can focus on dealing with the students' doubts and enhancing their analytical and predictive skills. First, he's looking for a tool that allows him to load and modify real classes stored in the university database, upload exercises with datasets and results, assign them to specific students and evaluate their performance. In addition, Daniel wants to track the students' progress and compare results between them.

Persona 2: Student

Table 3.2: Joana Silva's profile

Name	Joana Silva
Age	21 years
Sex	Female
Job title	Master's student in Data Science
Institution	University of Aveiro
Civil status	Single
Location	Mealhada, Aveiro, Portugal

Background: Joana is attending her first year of a Master in Data Science. She already has a Bachelor's degree in Computer Science and she's looking forward to learning more about ML and Data Mining. Joana has a limited knowledge of these fields. She's familiar with some theoretical concepts, but has never applied them in practice.

Goals: As a ML novice, Joana wants to learn more about the subject and prepare herself to tackle real-world challenges. She wants a platform where she can access exercises proposed by her teachers, submit solutions to them and check her position in class ranking. She'd also like to be notified through Slack when a new exercise is available, a deadline has been reached, benchmarks have finished or teachers have reviewed her work.

The context scenarios portray how the proposed system fits seamlessly into the end-users' activities, helping them to achieve their goals more effectively.

> **Daniel uploads a new exercise** - Daniel opens the web application and logs in. On his dashboard, he sees a button to create a new exercise. He clicks on it, triggering a form to appear. First, Daniel enters a title and the guidelines for the exercise. Then, he uploads 2 files: a training dataset and a test dataset, whose last column contains the expected values. He also specifies some rules: deadline, maximum number of submissions (if any), visibility (public or restricted to a group) and, last but not least, the performance metrics to be used in the automatic evaluation of the students' solutions. Finally, he submits the form and the exercise is created.

Joana is one of the students to whom Daniel has assign the task.

> **Joana submits a solution to the exercise** - Joana is notified through Slack that a new exercise was assigned to her class. So, she logs into the web application, goes to

the exercises page and selects an exercise. Joana notes she's able to download 2 files: a training dataset and an incomplete test dataset, i.e., without the expected values. After solving the problem, and training a model, she uploads a prediction file and submits it. The platform evaluates the solution and, a few seconds later, displays the results.

Daniel checks Joana's evaluation report and reviews her solution - Even though the deadline hasn't passed yet, Daniel wants to follow the progress of his students, especially Joana's, since she's the one who's struggling the most in class. So, he logs back into the web application, chooses the exercise and selects her. The page shows the automatic assessment report of Joana's work. Daniel verifies that Joana's model accuracy is 0.6, which is below the minimum expected by the teacher. He then downloads Joana's solution file and reviews it. He notices that Joana's model overfits the training dataset and, therefore, doesn't generalise well to the test dataset.

The underlined excerpts hint at the actions that were relevant for confirming and detailing the requirements we had already gathered.

3.1.5 Risks and dependencies

In every new project, there are inherent risks, and software engineering is no exception. Alongside the common issues that arise, such as internet connectivity failures, poor engagement by the target audience, and potential cyber attacks or sabotage, we also rely on external entities - the ones we identify below as secondary actors (Section 3.3).

Authentication is our biggest strength and weakness. On one hand, an institution-based authentication method offers the ability to establish a subject's identity without uncertainties, ensuring that the system remains accessible only to a specific community. On the other hand, the process of getting the right credentials and authorisation for using an identity provider is slow and throughout the project we had some difficulties on that matter.

Big change in requirement R1: To simulate the use of a university Identity Provider (IdP), we have turned to Firebase as our authentication provider. Additionally, we depend on various platforms provided by Google to support our deployment infrastructure.

R1: integrate Firebase authentication

3.2 Mapping user stories to use cases

As we will see in chapter 6, we started by breaking the work down into user stories to capture the high-level goals and benefits of a feature. We then mapped them to use cases. Each use case describes a sequence of interactions between the system and an actor to achieve a given goal.

For writing user stories, we used the following structure:
As a <type of user>, I want <some goal> so that <some reason>.

Table 3.3: Mapping Teacher user stories to use cases

User Stories	**Use cases**
As a teacher, I want to sign up, so that I can use the system.	Sign up
As a teacher, I want to securely log in to the system, so that I can access my classes and exercises.	Sign in
As a teacher, I want to log out of the system, so that my data can only be accessed and managed by me.	Logout
As a teacher, I want to group students in a new class, so that I can assign them exercises. As a teacher, I want to import a class from a file, so that I can group my students in the same class.	Create class
As a teacher, I want to view a class, so that I can see the students to whom I am assigning exercises. As a teacher, I want to add a student to a class, so that I can assign him/her exercises. As a teacher, I want to remove a student from a class, so that he/she can no longer access the exercises assigned to that class. As a teacher, I want to rename a class, so that I can better identify it. As a teacher, I want to delete a class, so that I can remove it from the system.	Manage class
As a teacher, I want to create a new exercise, so that I can assign it to my students.	Create exercise
As a teacher, I want to view an exercise, so that I can recall its details and rules. As a teacher, I want to edit an exercise, so that I can change its details and rules. As a teacher, I want to delete an exercise, so that I can remove it from the system.	Manage exercise
As a teacher, I want to enter a title and guidelines for an exercise, so that my students know what is expected from them.	Enter title and guidelines
As a teacher, I want to set the visibility of an exercise, so that I can decide whether it is public or only visible to a given class.	Set visibility
As a teacher, I want to choose a performance metrics, so that the system can automatically evaluate my students' solutions.	Choose performance metrics
As a teacher, I want to upload the required datasets for an exercise, so that my students can train and test their models.	Upload required datasets

Table 3.4: Mapping Student user stories to use cases

User Stories	**Use cases**
As a student, I want to sign up, so that I can use the system.	Sign up
As a student, I want to securely log in to the system, so that I can view public or assigned exercises.	Sign in
As a student, I want to log out of the system, so that my data can only be accessed and managed by me.	Logout
As a student, I want to view an assigned exercise, so that I can understand its requirements.	View assigned exercise
As a student, when viewing an assigned exercise, I want to check my ranking among my peers, so that I can see how well I am doing.	Check ranking
As a student, I want to check my model's evaluation, so that I can see how well it performs.	Check model evaluation
As a student, when viewing an assigned exercise, I want to upload my solution, so that I can submit it for evaluation.	Upload a solution
As a student, I want to download the provided datasets, so that I can train and test my model.	Download provided datasets
As a student, I want to view a public exercise, so that I can learn from it and be more prepared for assignments.	View public exercise

3.3 Actors

Primary actors are entities that interact directly with the system to achieve their goals. In this case, we have two primary actors: **Student** and **Teacher**.
Secondary actors are the ones that provide services to the system, enabling primary actors to achieve their goals. Therefore, we identified two secondary actors: **Slack** and **Firebase** systems.

3.4 Use cases

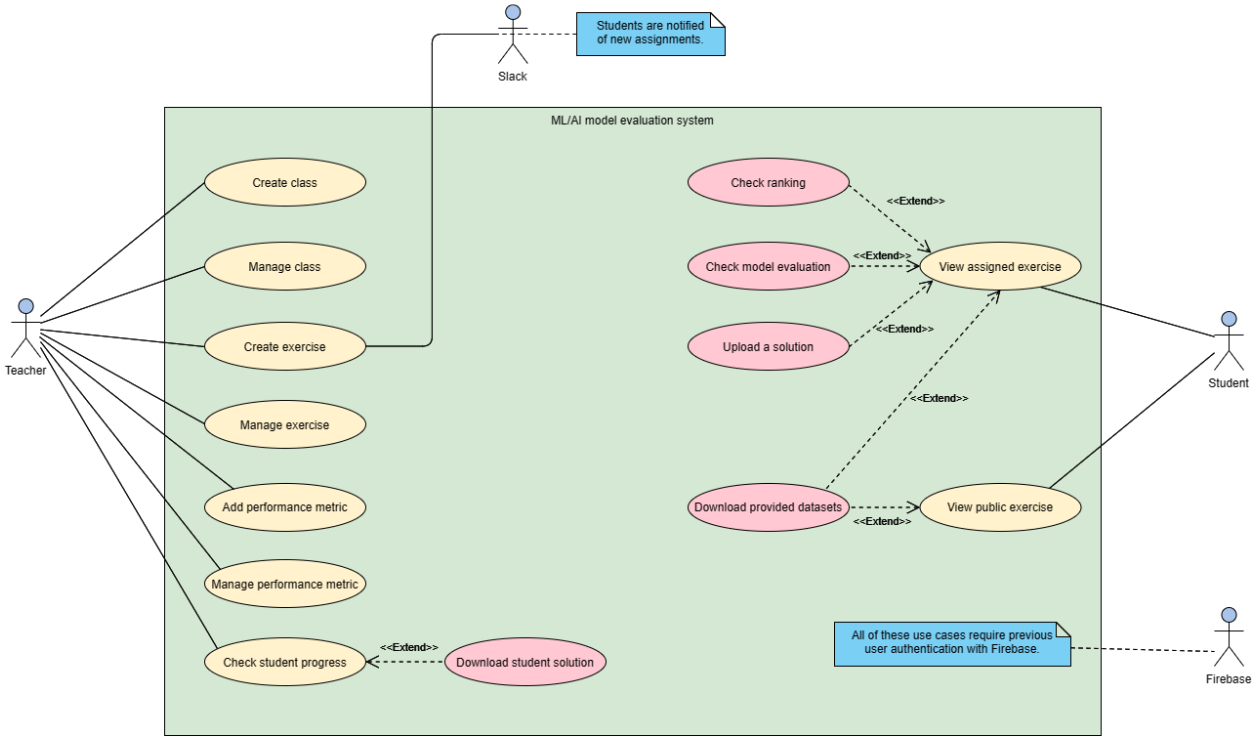

Figure 3.3: Use Case diagram

The authentication use cases - Sign up, Sign in and Logout - were omitted in the diagram (Fig. 3.3), since they are related to both types of users.

We have included an **extended specification** of the use cases in Appendix A. For each one, we document the underlying interaction between the actor and the system, enumerate secondary actors, pre-conditions, post-conditions and exceptions, when applicable, and assign a priority.

3.5 Non-functional requirements

Non-functional requirements are essential for the success of any software project or system implementation. While functional requirements define what the system should do, non-functional requirements specify how the system should behave. They are crucial for building robust, reliable, and user-friendly systems and by addressing them, organisations can deliver high-quality software that meets user needs and aligns with business goals.

For this project, we have identified the following non-functional requirements:

- **Availability**

  The system should be always available and responsive. This means that it should be accessible from any device, at any time, and respond quickly to user requests. This is important because it ensures that users can access the system when they need it and that they don't have to wait for long periods for the system to respond.

- **Security**

  The system should be secure and protect user data. This means that it should use strong encryption to protect sensitive information. It should also use authentication and authorisation to ensure that only authorised users can access the system and perform certain actions.

- **Maintainability**

  The system should be easy to maintain and update. This means that it should be well-documented and have a modular architecture that allows for easy updates and changes. It should also have a clear separation of concerns so that changes in one part of the system don't affect other parts of it, which makes it easier to scale the system as needed.

- **Usability**

  The system should be easy to use and intuitive. This means that it should have a simple user interface that is easy to navigate and understand. It should also have good performance and provide a good user experience, which means that it should respond quickly to user requests and provide feedback when necessary.

Chapter 4

System Architecture

4.1 Domain Model

There are several storage strategisms available, the two most common ones are: relational [9] and No-SQL [10]. For this work we adopted a relational database due to the need for joins and queries involving multiple entities, and because relational databases have a long-standing reputation for their maturity and wide adoption in the industry, which influenced our decision to opt for MySQL as our database management system. We also acquired a MongoDB for our non-relational database to handle unstructured data, as this is a reliable and popular document oriented database.

A good domain model plays a critical role in a database design as it serves as a conceptual representation of the universe we are addressing and it helps better communication between the developing team and stakeholders. It also allows to define a structure that is consistent, scalable and flexible, making queries efficient and it prevents risk and bad decisions in the long run.

Given the requirements from section 3.1.1, we proposed the following domain model, which has the given entities: the teacher, who plays a central role in the platform, can create classes which are an aggregation of students. These students have the responsibility to solve exercises proposed by the teacher which will upload datasets and define metrics to create personalised assignments, with deadlines for submission and may expose it to anyone that uses the platform. The role of the students is to submit their results, try to compete with themselves and colleagues, view the results and make their code available for the teacher to see. Based on these relationships between entities we sketched the database diagram shown in figure 4.1.

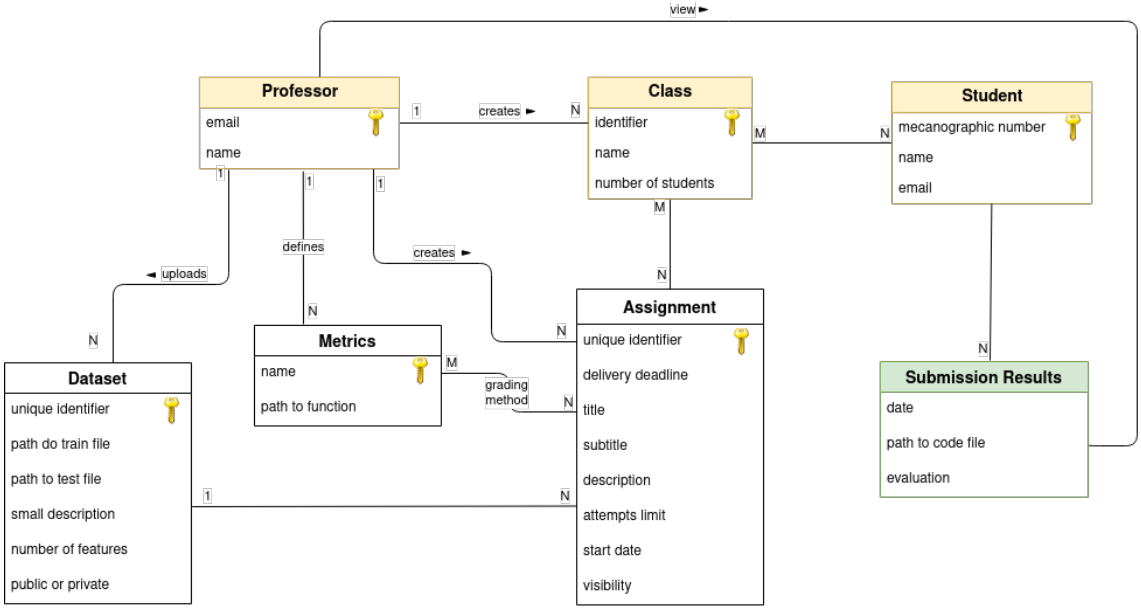

Figure 4.1: Database model - initial version

This was our first domain model with the intuition to store data in the table in green on MongoDB and tables in yellow could change depending on the data we could obtain from PACO.

The final domain model (Fig. 4.2) turned out to have some changes although the base is the same. These changes derive from the need to show more information on the interface, like file size and number of answers given to a certain assignment, even though most of these fields are computed at runtime, so the database model didn't change more than a couple of times. Besides that, we dropped the idea of using MongoDB, so 2 tables needed to appear to replace the unstructured data that would keep record of students' submission. Code submission table now stores information, regarding submitted files and results table stores the computed scores calculated by the metrics in use by the exercise.

Due to the simplicity of our domain and the relatively small data size, our model has proven to be highly efficient and the first draft had almost everything we needed for the final release of the product.

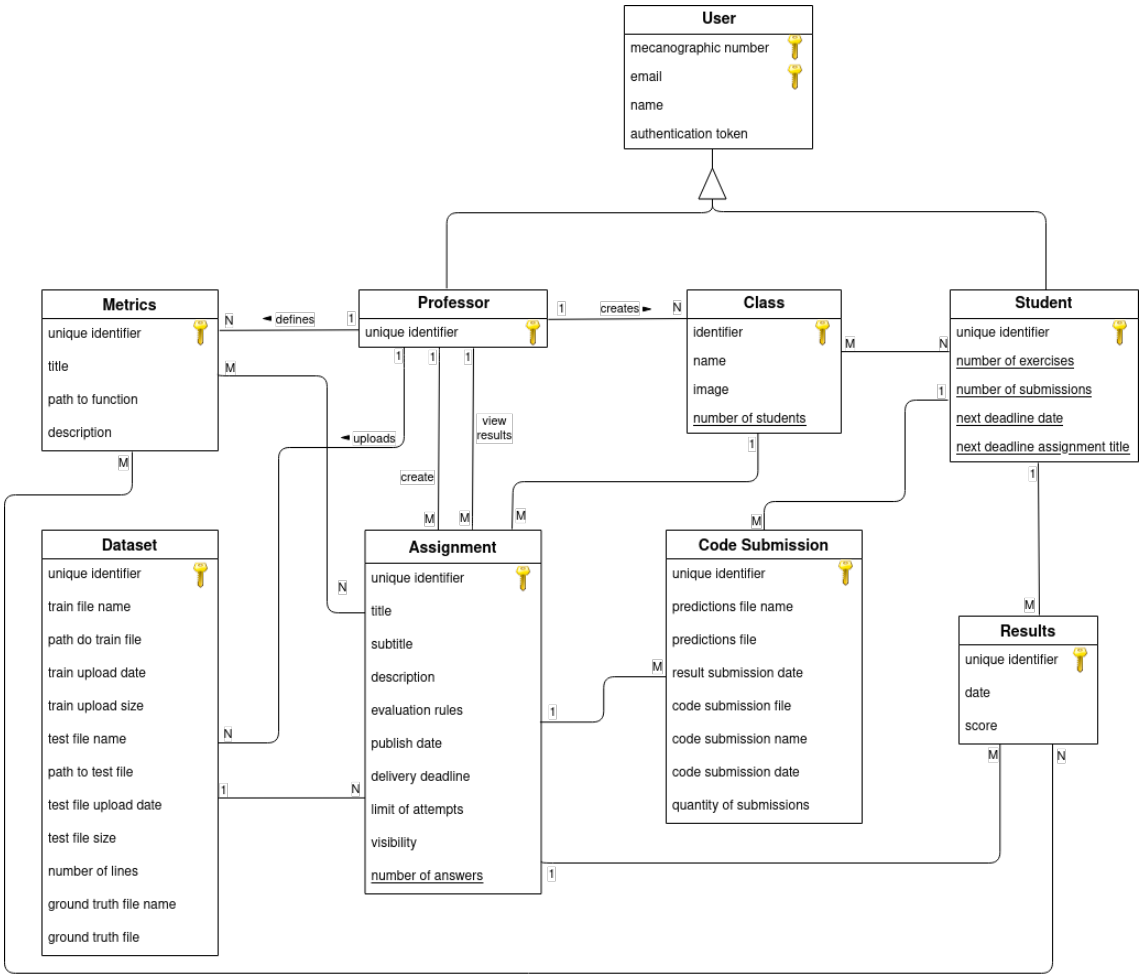

Figure 4.2: Database model - final version

4.2 Architecture Design

In order to fulfil both functional and non-functional requirements specified in the Product Concept section, it is extremely important to design an adequate and realistic system architecture. To begin with, we started analysing different design architectures that are commonly used in the software industry, from the most basic to the most complex ones. For this purpose, we used the *Software Architecture Patterns* [11] book as a reference, written by Mark Richards, which provides a great overview of the most common architectures used nowadays.

After analysing the different architectures, and taking into account the advantages and disadvantages of each one, we concluded that the most suitable architecture for our project is a Layered architecture, also known as N-tier architecture. With this type of architecture, the different components that make up the system are organised into horizontal layers, each one of them with a specific role. The main advantage of this design is that it allows us to separate the different concerns of the system, which makes it easier to maintain and extend.

The first version of the proposed architecture is composed of 4 different layers, namely:

- **Presentation Layer**

  This layer is responsible for handling the user interface. It is composed of a web application that allows the user to interact with the system. The technology used to develop this layer is React, which is a JavaScript library for building user interfaces.

- **Authentication Layer**

  This layer is responsible for handling the authentication and authorisation of the users. It is composed of a web service that provides the necessary endpoints to authenticate and authorise the users. For this purpose, we decided to use the Identity Provider (Idp) provided by the University of Aveiro.

- **Business Layer**

  This layer is responsible for handling the business logic of the system. It is composed of a web service that provides the necessary endpoints to perform the different operations of the system. We decided to use Django Rest Framework to develop this layer, which is a powerful and flexible python framework for building web APIs. This layer is composed of the following components:

  - **URL Dispatcher**: This component is responsible for handling the different requests made by the user. It is composed of a set of endpoints that are mapped to the different operations of the system.
  - **View**: This component is responsible for handling the different requests made by the user. It is composed of a set of functions that are mapped to the different endpoints of the URL Dispatcher.
  - **Cache**: This component is responsible for handling the caching of the datasets and other necessary data.
  - **Serialiser**: This component is responsible for handling the serialisation and deserialisation of the data. It is composed of a set of classes that are mapped to the different models of the system.
  - **Model**: This component is responsible for handling the data persistence of the system. It is composed of a set of classes that are mapped to the different tables of the database.

- **Data Layer**

  The data layer is responsible for handling the data persistence of the system. We decided to use two databases, one for storing structured data and another for storing unstructured data. The first one is a relational database, more specifically, MySQL, and the second one is a non-relational database, more specifically, MongoDB.

The Figure 4.3 shows the proposed architecture:

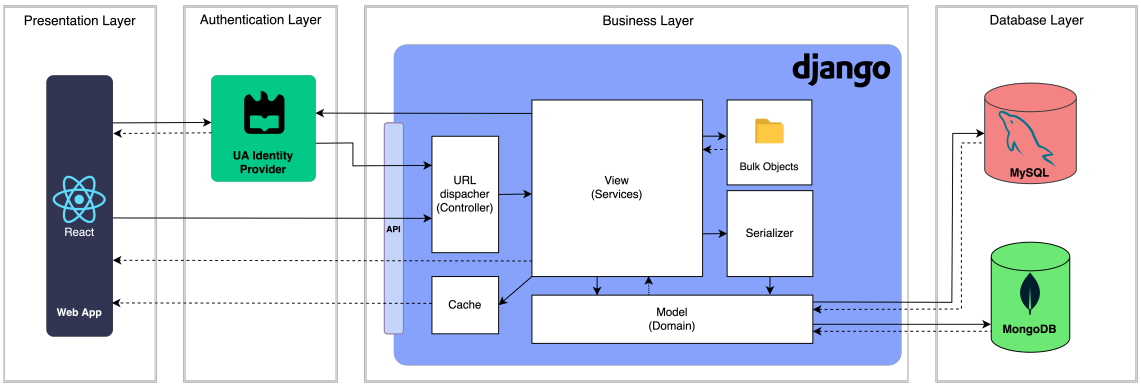

Figure 4.3: System Architecture - initial version

As aforementioned in chapter 3, one of the functional requirements of the system is to allow professors to create their own metrics to evaluate the students' models. In order to guarantee that the execution of the metrics is done in an isolated way and does not affect the system's security and integrity, we decided to add an isolated execution environment component to the business layer.

Due to external constraints related to the University's Identity Provider, we were forced to change the authentication layer. Instead of using the Idp provided by the university, we decided to use a third-party Identity Provider, more specifically, Google's Firebase Authentication Service.

In the first version of the architecture we also decided to use a cache component responsible to store the recently requested datasets. Nevertheless, in machine learning the size of the datasets is usually very large, which makes it difficult to store them in a cache due to memory limitations. For this reason, we decided to remove the cache component from the architecture, so that the system performance is not affected.

Regarding the data layer, we decided to use a single database, more specifically, MySQL, to store all the data. This decision was made to simplify the architecture and reduce the complexity of the system, because using MongoDB was one more point of failure.

As mentioned in the section 3.1.3 section, we considered it important to notify the users about the publication of new assignments via Slack. For this purpose, it was necessary to add a new individual component to the business layer, responsible for handling the communication between the system and Slack. This component was developed using FastApi and Slack's API.

In the first architecture proposal, we decided to store the files submitted by the users (datasets, images, models and results), in the local file system. However, this approach was not the most suitable, because it did not allow us to scale the system horizontally. For this reason, we decided to use a cloud storage service, more specifically, Google Cloud Storage, allowing us to store bulk data in a scalable way.

The Figure 4.4 shows the final version of the proposed system architecture.

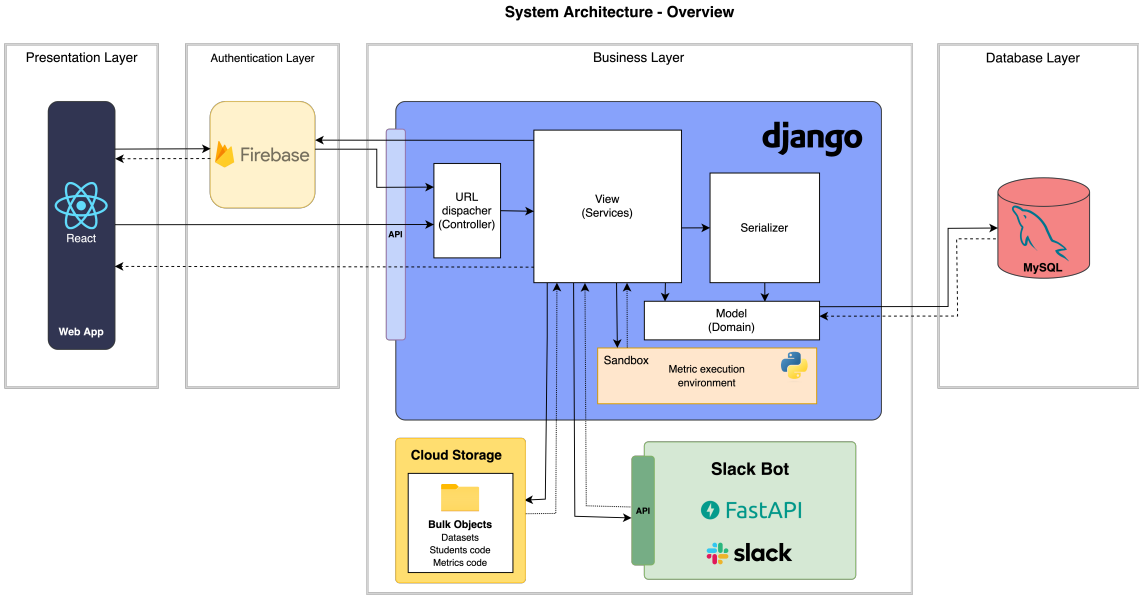

Figure 4.4: System Architecture - final version

Table 4.1: Architecture components that meet the requirements

Requirement	**System component(s)**
R1: integrate Firebase for authentication	Authentication service
R2: two profiles: student and professor	Authentication service, REST API, Database
R3.1: [student] consult the exercises (public and private)	Presentation, REST API, Database
R3.2: [student] download train and incomplete test dataset (without prediction)	Presentation, REST API, Database, Cloud Storage
R3.3: [student] upload results	Presentation, REST API, Sandbox, Database, Cloud Storage
R3.4: [student] view the results table (comparison with other students)	Presentation, REST API, Database
R4.1: [professor] set/add metrics	Presentation, REST API, Sandbox, Database
R4.2: [professor] define classes (group of students)	Presentation, REST API, Database
R4.3: [professor] add/remove/import students to the class	Presentation, REST API, Database
R4.4: [professor] define exercises and its visibility	Presentation, REST API, Database
R4.5: [professor] set deadline and/or retry limit	Presentation, REST API, Database
R4.6: [professor] access student results/code	Presentation, REST API, Database, Cloud Storage
R5: notify students of new assignments on Slack	Slack bot

4.3 Deployment

Deployment is a critical aspect of software engineering, it is the boundary between development code that is in a stable phase and a release product that is ready to show to the client. If this fails, client may lose trust on us and what could be a very good product turns out to be nothing is it only works on developers local environment. The software can be deployed in a variety of ways, from a virtual machine but, since no virtual machine was made available by the teacher and requesting one from the department, we had to turn our eyes to the domain of cloud computing and find a free alternative service to our problem there. Due to some previous exploration of one of us in this software engineering field, and because of our own personal preferences, we opted by using Google Cloud Platform Services as we think what they offer is reliable,

To deploy our Django app, we are using Google Cloud Run, which is a new GCP service (launched in 2019) that brings the best of both worlds — serverless and containers. It allows to run

stateless containers in a fully managed environment, abstracting away the underlying infrastructure and enabling to focus on building and deploying applications.

It has a pay-as-you-go pricing model but up to a limit is free. "In Cloud Run, each revision is automatically scaled to the number of instances needed to handle all incoming requests or events." [12] This guarantees almost infinite scalability as new instances of the base container are created and destroyed as their are needed.

We have established a continuous deployment workflow on the main branch of our GitHub repository, utilising Google Cloud Build. This integration involves a trigger mechanism that automatically activates upon detecting a push event, initiating the deployment process to Google Cloud Run. With this setup, any changes pushed to the main branch are swiftly built and deployed to our Cloud Run environment, ensuring that our application is always up-to-date with the latest code changes.

As it is a stateless environment, any data that requires storage must be processed by an external service, and as we have media to save, a new service was needed. Taking the momentum of using Google Cloud platform services, and after comparing different services as explained in detail by Google itself [13], we opted by using cloud storage. It allows storing any type of media in a structure called a bucket.

Our production database (MySQL) is running on a docker container which in turn is allocated inside a Google compute engine instance. This database could be on cloud SQL but that was discarded due to our lack of budget.

The slack bot is deployed on another Google's compute engine.

Finally, our front-end application is hosted on Cloudflare.

The authentication is external to us, so no deployment was needed for this layer.

For a more robust deployment infrastructure, Kubernetes engine could be in use.

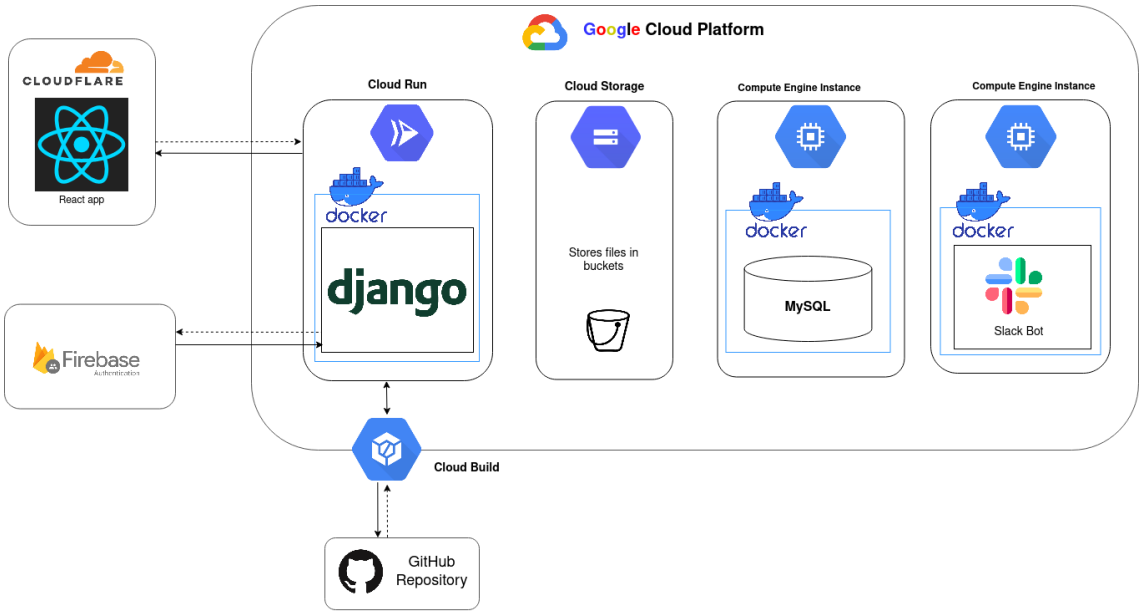

Figure 4.5: Deployment Architecture

Chapter 5

Implementation

5.1 Presentation Layer

This section describes the implementation of the main features of the web application, as well as the decisions made during the development process and its justification.

5.1.1 User interface design

Even though the presentation layer is just a single layer of the system, it is the means through which the user interacts with the system as a whole. For this reason, it is extremely important to design an intuitive and easy-to-use user interface. To guarantee a good user experience, we decided to follow the 10 usability heuristics proposed by Jakob Nielsen, presented and described in the *Usability Inspection Methods* [14] book and discussed in the Human-Computer Interaction course. Of the 10 heuristics, we decided to focus mainly on the following:

- **Visibility of system status**: The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
- **User control and freedom**: Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted action without having to go through an extended process.
- **Consistency and standards**: Users should not have to wonder whether different words, situations, or actions mean the same thing. It is essential to follow platform and industry conventions.
- **Error prevention**: Good error messages are important, but the best designs carefully prevent problems from occurring in the first place. It is crucial either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

- **Aesthetic and minimalist design**: Interfaces should not contain information that is irrelevant or rarely needed. Every extra unit of information in an interface competes with the relevant units of information and diminishes their relative visibility.
- **Help users recognise, diagnose, and recover from errors**: Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution.

5.1.2 UI components and layout

As we used React to develop the web application, we followed a component-based approach to design the user interface. This approach consists of dividing the user interface into a set of components, each one of them with a specific role. This allowed us to reuse the components in different parts of the application, which makes it easier to maintain and extend.

To facilitate and speed up the development of the interface, we decided to use CSS frameworks and component libraries, more specifically, Tailwind CSS and Flowbite React. To give flexibility to the user, and consequently improve the user experience, we made use of several types of components, including forms, tables, buttons, modals, popovers, tooltips, charts, etc.

5.1.3 Data presentation and visualisation

As previously stated, one of our main goals was to allow students to submit their ML models to measure their performance according to a set of metrics. For this reason, it is extremely important to present the results clearly and intuitively, so that the student and professors can easily understand them and make the necessary conclusions. For this purpose, we decided to use tables and charts, as they are the most common and intuitive way to present data, preventing results misinterpretation.

5.1.4 Error reporting and feedback

As previously mentioned, one of the usability heuristics we decided to focus on was the help users recognise, diagnose, and recover from errors. For this reason, we decided to provide the user with feedback on the different actions performed, as well as to report any errors that may occur. To do this, we display messages, informing the result of the action performed, as well as any errors that may have occurred and how to solve them. To achieve this, we made use of alert components, modals, tooltips, icons and warning colours.

5.1.5 Integration with the business layer

As said before, the presentation layer is responsible for handling the user interface. However, it is also responsible for handling the communication between the user and the business layer. For this purpose, we decided to use the Axios library, which is a promise-based HTTP client, allowing us to make asynchronous HTTP requests to the business layer, as well as to handle the responses.

5.1.6 Performance and Optimisation

To increase the performance of the web application, and its responsiveness, we tried to reduce the number of HTTP requests made to the business layer, passing information between components, whenever possible.

5.2 Authentication Layer

Authentication is a fundamental aspect in our system, as it ensures the correct identity of the subject attempting to access the platform and provides a secure way of verifying their credentials.

Initially our plan was to use the university IdP. However, due to some delays, this goal was not possible to achieve. In this final version, we have opted for an alternative IdP that closely mimics the desired functionality. We have turned our focus towards Firebase Authentication, which facilitates a straightforward authentication mechanism using email and password, similar to UA system. It is also very simple to implement, either server side or client side.

When a user enters the platform for the first time, he/she required to create a new account. For that, a sign up page is shown. After all fields are filled, it sends the data to the server, which replies with an authentication token. This token is obtained by making a call to the Firebase Auth API, where the user's email and password are sent and in return, the API provides various details about the user, including a session token.

If the subject has already registered on the platform, they only need to provide the email and password and our system takes charge of forwarding the data to request the token to Firebase Auth.

We adopted this authentication mechanism to minimise the number of client-side calls and prioritise server-side handling. By delegating the riskier aspects to the server, we enhance security and mitigate potential attacks on the user's end.

We also implemented an endpoint which receives the token stored in user's cookies and returns an object of a user, which contains information about user type, teacher or student, email, name and nmec. The relevance of this is to establish a session after a reboot on the client's page.

5.3 Business Layer

5.3.1 REStful API

As we have the frontend and backend modules separated on distinct remote servers, a method of carrying information from one side to another needed to be implemented.

We have implemented an API (Application Programming Interface) that defines a set of rules and protocols, that allows different software applications to communicate and interact with each other, without developers needing to understand the underlying implementation details. Among the various types of APIs available, we have chosen to adopt a RESTful style because of its modularity, scalability, interoperability and flexibility, but also because it is a standard practice in the industry. This approach suits us because the system requires CRUD operations on data and we

can create endpoints to meet our needs, ensuring that each page receives exactly what it is supposed to. It is very adaptable and also very secure, due to the ability of using authentication and authorisation mechanisms, such as API keys, tokens, or OAuth.

CRUD operations, which are supported by a RESTful API, involving creation of data, reads on that data, updates and deletes, all of which combine very well with the system we have at hand.

- Read operations are supported by http GET method and are very useful to inquire our backend server for data, as an example, we need this to obtain all available exercises.

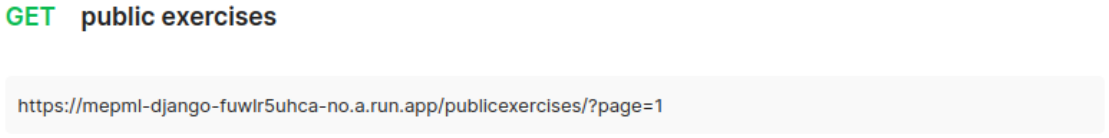

Figure 5.1: GET endpoint for public exercises

- Create operations are handled by POST method. Our main actions, which are creating an assignment and submitting a ML model and predictions, could not be possible without this type of endpoint.

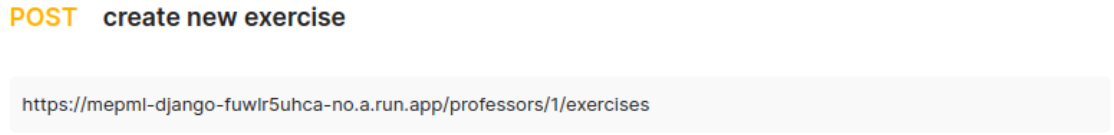

Figure 5.2: POST endpoint to upload a new exercise

- For update operations, the PUT method is required, and we are taking advantage of this, when an assignment or class needs to be changed.

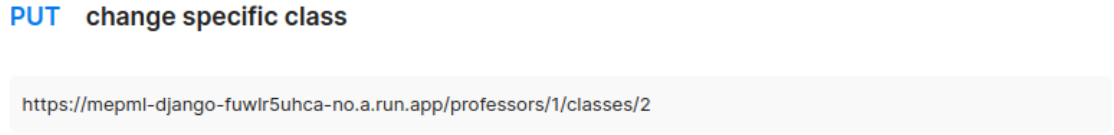

Figure 5.3: PUT endpoint to update a class

- Delete operations are handled by DETELE method and with use it for example, to delete a class, an exercise or a metric created by the teacher himself.

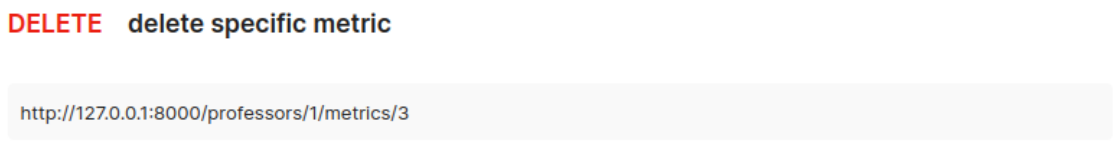

Figure 5.4: DELETE endpoint to delete a metric

To test the APIs endpoints, during the development process, we used Postman.

For communication within the team as well with the professors, we published the API for everyone to see, but only with this purpose, because our API is not meant to be public, is only for the sake of documentation.

The examples provided above are merely an illustration. We made the full API documentation public [15].

5.3.2 Sandbox environment

We should not assume that all users are trustworthy and allow them to run arbitrary code on our server. So, to run the performance metrics, we relied on a sandbox environment, where Python methods are restricted to those that are strictly necessary.

As a first approach, we tried ‘RestrictedPython‘, a library that allows the definition of a whitelist of methods and variables that can be used in the code, but we got several errors, namely not recognising ‘len‘, ‘range‘, even after injecting the correct built-in methods.

To keep up with the project schedule, we decided to implement our own sandbox using the ‘ast‘ module (Fig. 5.5). First, we parse the metric source code into an abstract syntax tree. Then we remove all import statements, since we do not want to allow the user to import external libraries. The sandbox already imports the necessary ones. Finally, we remove all calls to the ‘open‘ method, as we do not want to allow the user to read or write files on our server. We then unparse the tree into a code object and execute it, passing the data to the ‘score‘ function defined within the code. The ‘score‘ function is always called, so other functions are ignored.

5.3.3 SlackBot

As mentioned before the SlackBot is a separate module in the business layer. It has an API made with FastApi, which allows the SlackBot to receive requests from the Django API when a new assignment is created. It is also connected to the Slack API, which allows it to send messages to the Slack channel. It is very flexible and can be easily extended to support other types of notifications, such as notifications about new grades or announcements, which makes it a useful addition to the system, because it can send any message to any channel in the Slack workspace.

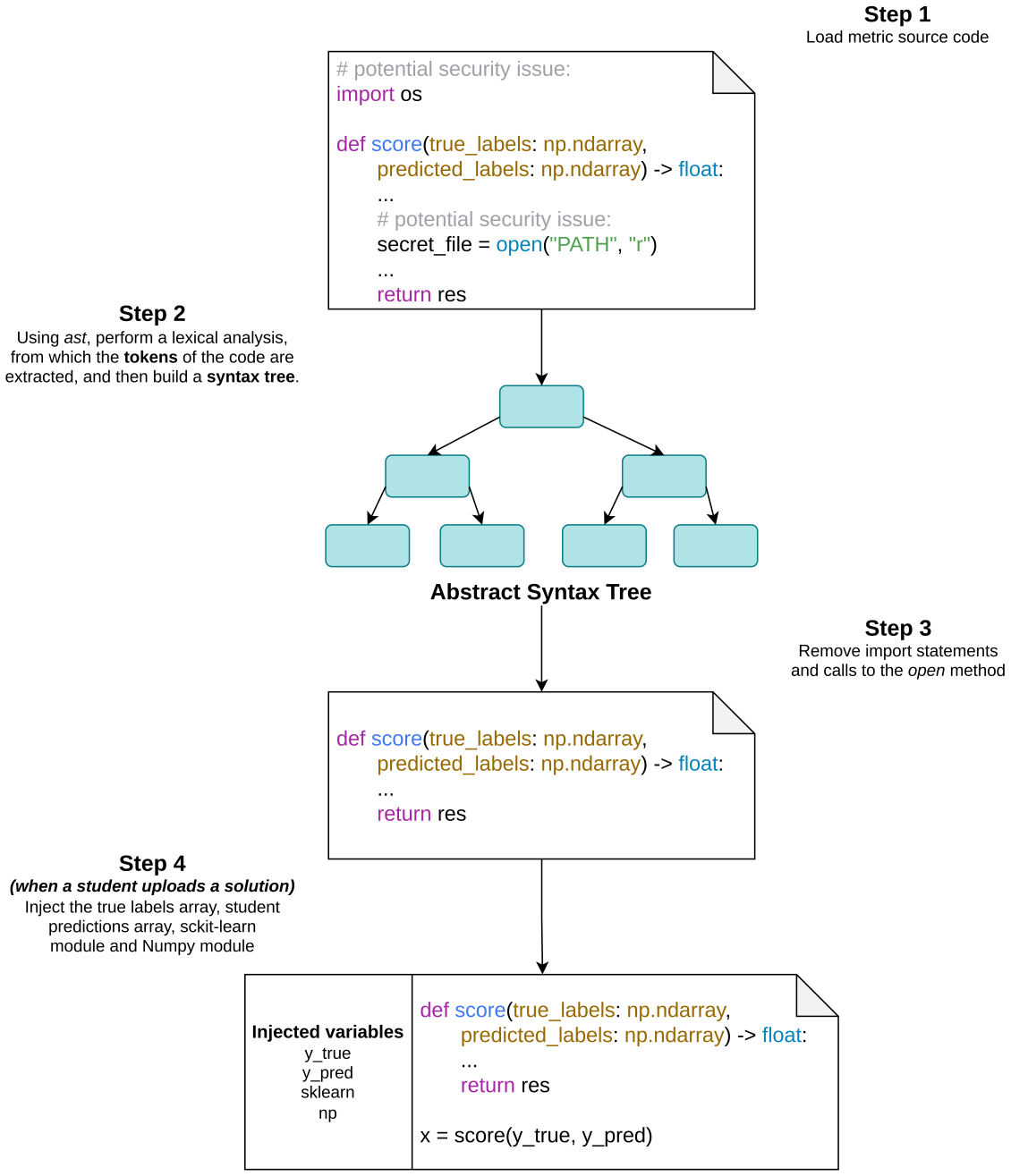

Figure 5.5: Metric execution pipeline

5.4 Database Layer

This project implied persistent data storage, and we opted for MySQL as our preferred database management system. By using MySQL, we can effectively create, manage, and maintain data consistency while preventing concurrent operations.

Since our back-end framework is Django, we are taking advantage of Object Relational Mapper, which is a technique or technology that facilitates the interaction between a relational database and object-oriented programming paradigm. With it, we can write classes and methods to represent data and it abstract us from raw sql commands, mapping python code into tables, columns and rows as well as relationships between tables.

At first, we incorporated MongoDB into our project, assuming its capability to store data with an unstructured schema would be beneficial. However, as the project progressed, we realised that it wasn't essential, and after careful consideration, we decided to remove it as we were adding unnecessary complexity.

The removal of MongoDB resulted in significant improvements to the overall system. It streamlined our dependencies and eliminated a potential point of failure, enhancing the system's reliability and maintainability.

In an environment with more resources we could have a backup database, however, this is overkill for the scope of this project.

Chapter 6

Project Management

Project management serves as a critical foundation for successful software development. It provides a structured approach to planning, organising, and managing resources to achieve specific goals and objectives. Without a project management plan, software development projects can face a variety of challenges and setbacks, including missing deadlines and poor quality outcomes. A good plan helps mitigate risks and ensure that the project is completed on schedule and within the requirements defined.

Given the importance of project management, in this chapter, we will discuss the plan for our software development project. We will start by discussing the team roles and responsibilities, followed by the project schedule and communication plan. Finally, we will discuss the methodology used for this project.

6.1 Team Roles

For this project, we have a team of five members, each with a specific role and responsibilities. The team roles were assigned based on the skills and experience of each member and the needs of the project. These roles follow the OpenUP methodology which allows a software development process focused on people interaction and collaboration since "Nobody does great software alone but a team working together can do extraordinary things" [16].

The team roles are presented in Table 6.1.

Table 6.1: Team Roles

Role	Member
Team Manager	Leonardo Almeida
Product Owner	Rafael Gonçalves
Architect	Pedro Rodrigues
DevOps Master	Emanuel Marques
Quality Assurance Tester	Diogo Magalhães
Developer	All the above

- **Team Manager**

  The Team Manager is responsible for overseeing the entire software development project. He provides leadership, guidance, and direction to the team. He is also responsible for coordinating the team's activities, making strategic decisions, managing risks and resources and ensuring the project's successful execution.

- **Product Owner**

  The Product Owner is responsible for defining the features and functionality of the software product being developed and prioritising the product backlog with the Team Manager. He represents the Stakeholders and is responsible for ensuring that the team delivers value to the business.

- **Architect**

  The Architect is responsible for designing the overall structure and system architecture of the software. He analyses requirements, creates technical specifications, and develops an architectural blueprint that guides the development team. He also ensures that the software is scalable, maintainable, and aligns with industry best practices.

- **DevOps Master**

  The DevOps Master focuses on the integration of development and operations to streamline software development and deployment processes. He is responsible for automating the software development lifecycle, including building, testing, and deploying software, ensuring smooth collaboration between developers.

- **Quality Assurance Tester**

  The Quality Assurance Tester is responsible for ensuring the quality, functionality, and performance of the software. He develops test plans, executes test cases, and reports bugs and issues to the development team. He also works with the Product Owner, defining acceptance criteria and ensuring that the software meets the requirements.

- **Developer**

  In addition to the specific roles mentioned above, all team members are also developers. We collaborated to develop software, write code, and implement features and functionality. We also work together to ensure that the software meets the requirements and is delivered on time.

6.2 Project Schedule

The project schedule is a critical component of project management. It defines the tasks, activities, and milestones required to complete the project. It also provides a timeline for each task and activity, allowing the team to track progress and ensure that the project is completed on time. There are 4 milestones: Inception, Elaboration, Construction, and Transition. Each milestone represents a phase of the project and has a set of deliverables that must be completed before moving on to the next milestone.

- **Inception**

  The Inception phase is the first phase of the project. It is focused on understanding the project's scope and objectives, defining the requirements, creating a project plan and schedule, and identifying potential risks and issues.

- **Elaboration**

  The Elaboration is the next phase, where we do a requirements enhancement, define a solid architecture to support the project and with this mitigate the risks and issues identified in the Inception phase.

- **Construction**

  The Construction phase is where we start developing the product. We start by creating a prototype to validate the requirements and ensure that the product meets the Stakeholder's needs. then we conduct usability testing to ensure that the product is easy to use and meets the Stakeholder's expectations. Finally, we analyse the feedback and make improvements to the product.

- **Transition**

  The Transition phase is the final phase of the project. It is focused on deploying the product to the production environment and ensuring that it is ready for use and do minor adjustments to the product. It is also focused on finalising the project documentation.

For this project, we used a Gantt chart [17] to visualise the project schedule, as shown in Figure 6.1. It provides a visual representation of the project timeline, including the start and end dates for each task and activity. It also shows the dependencies between tasks, allowing the team to identify potential bottlenecks and delays. This schedule was made at the beginning of the project and updated as the project progressed, to reflect the actual progress of the project, as well as changes in priorities and project plan.

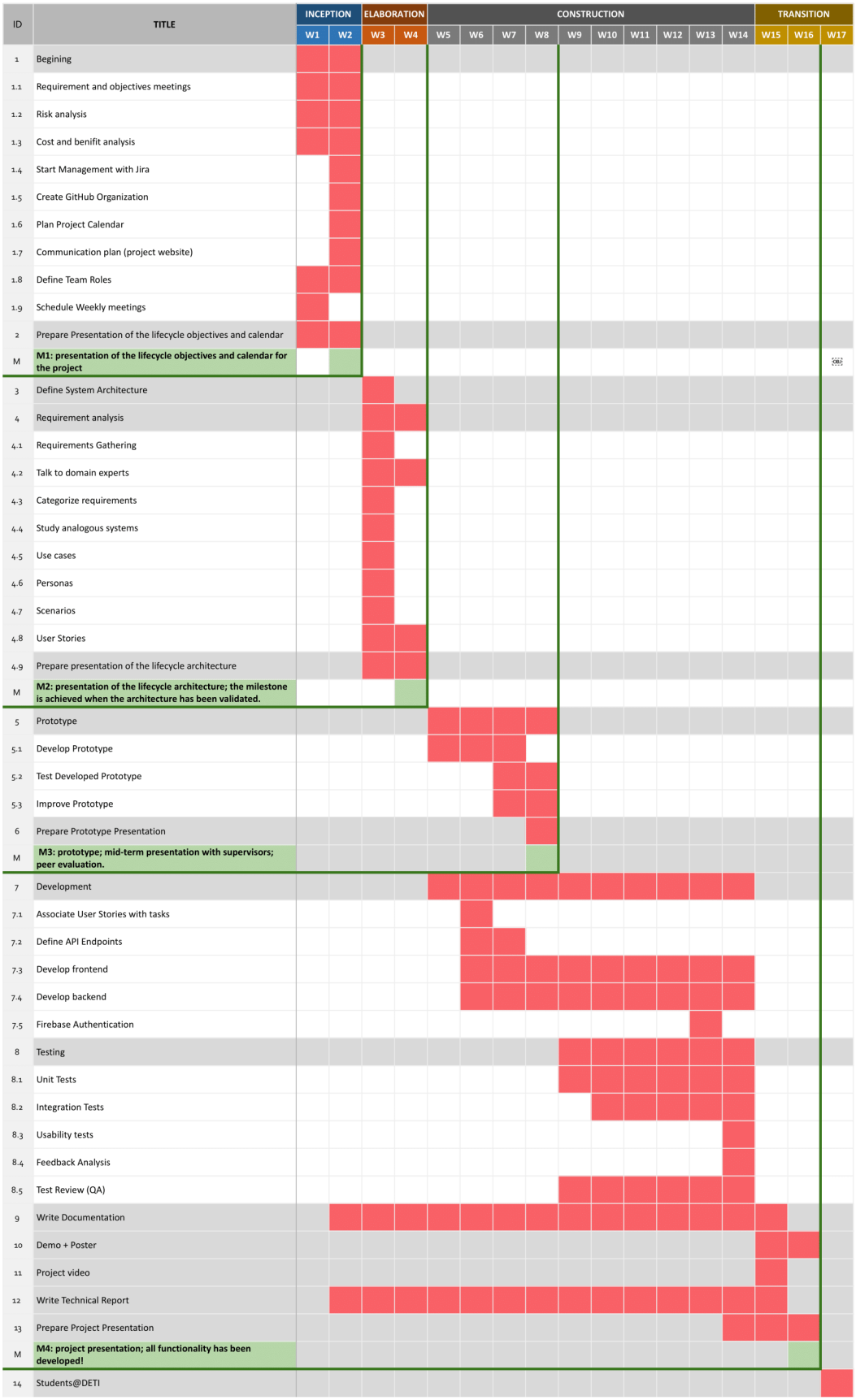

Figure 6.1: Project Schedule

The main changes to the project schedule were related to usability testing. We initially planned to conduct usability testing as soon as the authentication and authorisation features were implemented. However, due to some delays with UA IdP integration, we had to postpone usability testing until the end of the project. Other changes were the fast completion of some tasks, such as the implementation of the user interface, which allowed us to start working on other tasks ahead of schedule and add a new extra feature to the project.

6.3 Communication Plan

To promote communication and collaboration within the team, we used Slack [18] to communicate and discuss project-related issues between team members and the Stakeholder, Zoom [19] for weekly meetings and Jira [20] to manage the project backlog, track progress, and assign tasks to team members. For the development process, we used GitHub [21] to manage the source code and track changes made to the project. Finally to show the project's progress we made a website [22] with all the information about the project, including the documentation, presentations, source code, and other relevant information.

The GitHub repositories we committed at are all within a the same GitHub organisation created by us for this purpose [23]. We created the repositories according to logic components that met the requirements of PI curricular unity as well as our interests for the platform and to have a clear separation between work across different layers. With this in mind, a repository for the website for communication [24] was created in the beginning, another for the functional prototype [25] that was later migrated to the deployed version on "MepML-frontend" repository [26], another repository for the entire backend server [27] and finally a small repository to store the code the allows us to send slack notifications [28].

6.4 Methodology

For this project, we used an agile methodology [29], specifically Scrum [30] , to manage the software development process. Scrum is an iterative and incremental framework for software development that focuses on delivering value to the business. It is based on the principles of transparency, inspection, and adaptation, allowing the team to adapt to changing requirements and deliver high-quality software. This methodology enhances flexibility, collaboration, and adaptability which reduces the risks and facilitates the development process.

To develop the product we divided the project into several sprints. At the beginning of each sprint, we planned the tasks and activities to be completed during that sprint. We also defined the sprint goals and the acceptance criteria for each task. At the end of each sprint, we reviewed the work done and presented the results to the Stakeholder. We also discussed the lessons learned and identified areas for improvement. Finally, we planned the next sprint, based on the feedback received from the Stakeholder and the lessons learned from the previous sprint.

We also did Feature-driven development (FDD) [31] to develop the product, using user stories as the unit of development. A user story is a short description of a feature, written from the perspective of the user. It describes the feature's functionality and the value it provides to the user. It also includes acceptance criteria that define the conditions that must be met for the feature to be considered complete. User stories allow us to prioritise features, and ensure that the software meets the user's needs.

Chapter 7

Quality Assurance

Quality assurance is a critical component in modern software development practices that ensures that the delivered software is reliable, meets its requirements, and results in a high-quality product. This type of process usually involves a set of systematic approaches that are used to identify, assess, and mitigate risks and issues that may compromise the final quality of the software during its development time.
This was a hard task to accomplish since the team was composed of only 5 members and the time to develop the application was very limited, and since all elements of the team were still starting to learn what quality assurance was and which were the best practices and methods used in the industry for this purpose in another course unit. This section describes the quality assurance processes, tools, and protocols that were used in the development of the application and showcase how they helped to improve the quality of the final product.

7.1 Static Code Analysis

Static code analysis plays a very important role in the regard of quality assurance best practices. This type of analysis can be performed during the development and writing of the code and its main goal is to identify programming errors, coding standard violations, security vulnerabilities, and potential bugs and errors in the code before running and executing the program. This is usually done by analysing the code against a set of coding rules and standards.
After some discussion on what would be the best tool to perform this type of analysis, we decided that we would be using SonarLint, and everyone on the team has committed to using this tool when developing the application and its features.
SonarLint is a tool that can be integrated and used in the most popular IDEs and editors and is capable of detecting code smells, which are any characteristics in the source code of a program that possibly indicates a deeper problem, detect bugs, that are errors that lead to unexpected or incorrect results, detect security vulnerabilities, that are weaknesses in the code that can be exploited by a malicious actor to perform unauthorised actions, and enforce coding standards, that are a set of guidelines, best practices, and conventions that developers should follow when writing code.

SonarLint supports several programming languages, such as Python and JavaScript, which were the languages chosen for the development of the application.

7.2 Unit Testing

Unit Testing is another essential aspect of quality assurance in software development. The purpose of this type of testing is to ensure that each individual unit of the software work as intended and meets its requirements. This process is possible by isolating each unit and testing discrete sections of code, which helps to identify defects, bugs, and unexpected behaviour early in the development process.
Unit tests, properly written and executed, were only developed for the backend of the application since the frontend is mostly used for showing data coming from the backend and it is easy to check manually if the data is being correctly displayed. To develop those tests, we used Python unittests, which is a unit-testing framework that is part of the Python standard library.

The first step in the development of the Unit tests was creating models with the expected data for each page of the web application. With the expected results defined, we were able to use a Test-Driven Development approach, which is a software development process that emphasises writing automated tests before writing the actual code, to develop tests for each serialiser of the Django rest framework application. This approach helped us to ensure that the code of each serialiser was working as intended since some of the serialisers had some logic that needed to be tested. In addition to testing the serialiser by itself, those tests were also used as guidelines to develop the views of the application, since most of the GET requests views were just returning the data serialised, which was already tested.

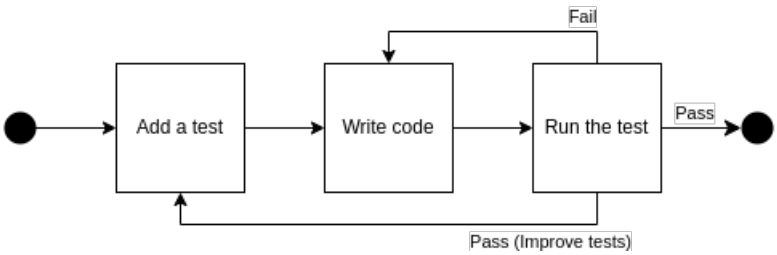

Figure 7.1: Ideal test workflows.

The ideal workflow for the development of the tests and features of the application would be the one shown in Figure 7.1. The problem is that, since we were using the Agile methodology, sometimes we had to change some of the models and serialisers, which would break the tests. This was a problem since we lost some time fixing the tests that should already be working and finished.

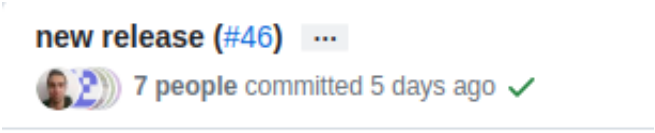

Figure 7.2: Github Actions Continuous Integration Pipeline.

To prevent that from happening and to prevent untested code to be deployed, we used Continuous Integration. Using GitHub actions all the tests were run automatically after each push to the repository, notifying the team if any of the tests were not passing. As we can see in Figure 7.2, we had a checkmark or a cross next to the commit message, which means that all the tests were passing or not.

7.3 Usability Testing

Usability tests play a crucial role in software development by assessing the effectiveness, efficiency, and overall user experience of a software application. The primary purpose of usability testing is to evaluate how easily and intuitively users can interact with the software, identify any usability issues or bottlenecks, and gather valuable feedback to enhance its design.

The following sections describe the usability tests performed, as well as the results obtained.

7.3.1 Sample

With the collaboration of Professor Petia Georgieva, we had the opportunity to perform usability tests in one of her "Complementos de Aprendizagem Automática" (CAA) classes. This course is part of the Master's degree in Informatics Engineering and Data Science.

The usability tests were performed by 9 users, where 8 of them were students and 1 was the professor herself. As our system has two different types of users - students and professors - we decided to divide the users into two groups as follows:

- **Group 1**: 4 elements that will test the system with the role of students.
- **Group 2**: 5 elements that will test the system with the role of professors.

Obviously, the professor who helped us with the usability tests was part of the second group. We also considered that the second group should be larger than the first one, as the system has more features for professors than for students and, therefore, it is more likely that more usability issues would be found.

Even though the number of users was slightly lower than the recommended for usability tests, the obtained results allowed us to identify the main usability issues and bottlenecks of the system, as well as to gather valuable feedback to enhance its design.

7.3.2 Method

In order to not take too much time from the class, and taking into account that the creation of user accounts is a trivial task, we decided to create the accounts for the users in advance, more specifically 5 student accounts and 5 professor accounts. Before we gave the users access to the system, we briefly described the theme of the project and the main features of the system, as well as the purpose of the usability tests.

After this brief introduction, we gave the users access to a link where they could find the following information:

- Link to the platform
- Microsoft Forms questionnaire for users with student role
- Microsoft Forms questionnaire for users with professor role
- Zip file with the necessary files to perform the tests, including a training dataset, a test dataset, a simple metrics file, and a model and its results.

Before users started performing the tasks, we asked them to fill out the first part of the questionnaire, where they had to answer two questions about their experience with machine-learning topics and machine-learning online platforms. The purpose of these questions was to understand the users' background and experience, as well as to identify possible biases that may influence the results of the usability tests. The asked questions were the following:

- How would you rate your level of knowledge on Machine Learning topics?
- How would you rate your experience with Machine Learning digital platforms (e.g. Kaggle, CodaLab, Weights and Biases, Machine Hack, etc.)?

After the users answered these questions, we asked them to perform the tasks described in the second part of the questionnaire, where they had to classify the difficulty of performing each task from 1 to 5, where 1 is very easy and 5 is very difficult. The tasks were the following:

- **Students' Tasks**

  1. Log into the platform with the provided account.
  2. Select one of the classes to which the student belongs. Verify how many members the class has and the Professor's email address.
  3. Go to the list of "Assignments" assigned by Professors and select the most recent one.
  4. Download the training and test dataset of the selected assignment.
  5. Upload and submit the provided model and its results, in order to measure its performance.

- **Professors' Tasks**

    1. Log into the platform with the provided account.
    2. Create a new class and add a new student.
    3. Access the metrics that already exist in the platform.
    4. Create a new metric.
    5. Create a new exercise.
    6. Check students' results.

Regarding the observation technique, as the total of elements that were testing the system was greater than our team, it was not possible to apply the think-aloud technique, where the user is asked to verbalise his thoughts while performing the tasks. For this reason, we decided to use a less intrusive technique, where we observed the users while they were performing the tasks, taking notes of the difficulties they encountered and the feedback they gave us. After the users finished performing the tasks, we asked them to fill out the third part of the questionnaire, where they had to answer 13 multiple-choice questions about the usability of the system and user experience with a 5-point Likert scale, where 1 is totally disagree and 5 is totally agree. The asked questions were the following:

1. The system is easy to use
2. I can easily find what I am looking for in the system
3. The system is slow
4. The system is pleasant to use
5. I feel the need for help with some features
6. The usage of the website requires more in-depth knowledge or previous experience
7. The size of the characters on the screen makes them easy to read
8. The most important information has a good highlight
9. The amount of information displayed per screen is adequate
10. The layout of the information displayed per screen is adequate
11. The icons are intuitive
12. The visual aspect is attractive
13. It is easy to navigate the system

7.3.3 Results

After the usability tests were completed, we analysed the results obtained and identified the main usability issues and bottlenecks of the system.

In terms of time spent performing the tests, the average time spent by users with the role of students was 24 minutes and 23 seconds, while the average time spent by users with the role of professors was 18 minutes and 9 seconds.

Regarding the background and experience of the users, the average knowledge of machine-learning topics on a scale of 1 to 5 was 3.88, while the average experience with machine-learning online platforms was 3.44. This means that, in general, the users had a good knowledge of machine-learning topics and a good experience with machine-learning online platforms.

In terms of the difficulty of performing the tasks from a student perspective, all the users classified the tasks as 1, which means that they considered the tasks very easy to perform. On the other hand, in terms of the difficulty of performing the tasks from a professor's perspective, the users did not classify all the tasks as very easy to perform, probably due to the fact that the tasks were more complex and required more steps to be performed. The different difficulty classifications of the professor's tasks are shown in the Figure 7.3. The scale corresponds to the level of difficulty in performing the tasks, where 0 represents a very low level of difficulty and 5 a very high level of difficulty.

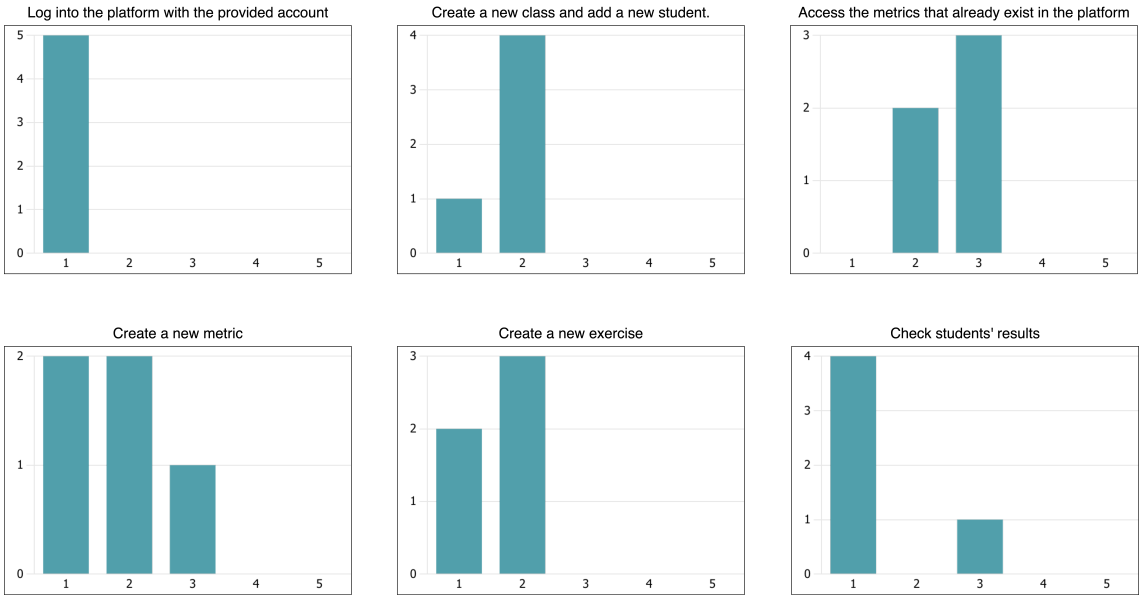

Figure 7.3: Difficulty on performing professor tasks

Regarding the usability of the system, the users gave us a positive feedback, as they agreed with the majority of the statements.

Based on the feedback received from the users, also presented in the appendix section, we identified several usability issues with different levels of severity. The following list shows the main usability issues identified:

- Difficulty in finding the "Metrics" page where the professor can see all available metrics and create a new one. There is only one button that allows professors to access that page, which is not intuitive.
- The dropzone component used to upload files does not respond to the user's actions.
- There is no warning message when professors leave the page when are creating classes, metrics, or a new exercise, which may lead to the loss of the information entered.
- Some UI components in the classes page do not adapt to the screen size when the user resizes the window.
- Classes with the same name are visually indistinguishable on the classes page.
- The "Create Class" button is not visually distinguishable from the other buttons when creating a new class.

In addition to the usability issues identified and aforementioned, we also received valuable suggestions that could be implemented in future versions of the system, such as:

- Allow professors to edit code when creating a new metric, similar to what happens when creating a new exercise with the markdown editor.
- Add an algorithm to calculate the highest grade that a student got in an exercise, based on the grades of the metrics that are part of the exercise.
- Add more filters in the assignments and public exercises pages, so that the search returns more specific results. Those filters could be the type of problem (classification or regression), the type of algorithm (supervised or unsupervised), the metrics used to evaluate the performance (accuracy, precision, recall, etc.), and even the department name of the class associated with the exercise.
- Allow students to see the highest scores of their classmates so that they can compare their performance.
- Allow submissions to public exercises even though those results are not going to be evaluated by the professor.

In general, the feedback received from the users was very positive, as they considered the system easy to use and intuitive. Nevertheless, the users also gave us valuable suggestions that could be implemented in future versions of the system, which would improve the user experience and the usability of the system. All the content used in the usability tests, including the questionnaires and the results obtained, can be found in the appendix section.

Chapter 8

Conclusions

8.1 Main Results

In this project, we have developed a web app that allows professors to make machine learning exercises for their students, where the students can solve the exercises and get immediate feedback on their solutions. The web app is designed to be intuitive and easy to use. We achieved all the objectives defined, including an extra feature.

The app has two main workflows, one for each type of user: professors, shown in Figure 8.1 and students, shown in Figure 8.2. The professor can create and manage classes, exercises and metrics and analyse the results of the students. The student can solve exercises and see their results, in comparison with the rest of the class.

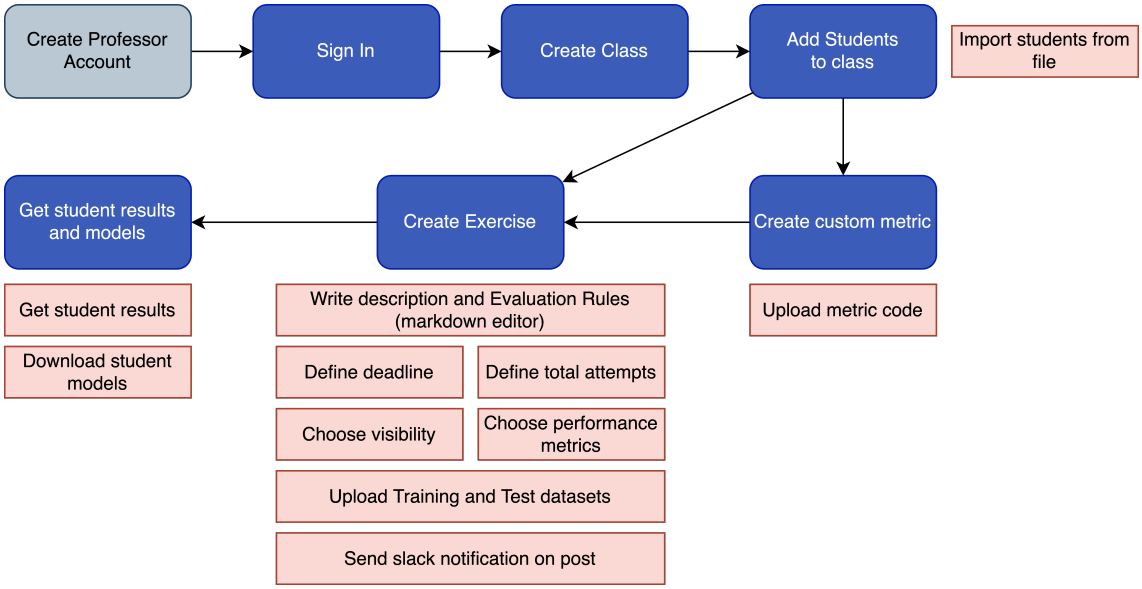

Figure 8.1: Professor workflow

While developing this project we have learned a lot about ML, the technologies used and how to work as a team to solve problems and unforeseen situations. We have also learned the

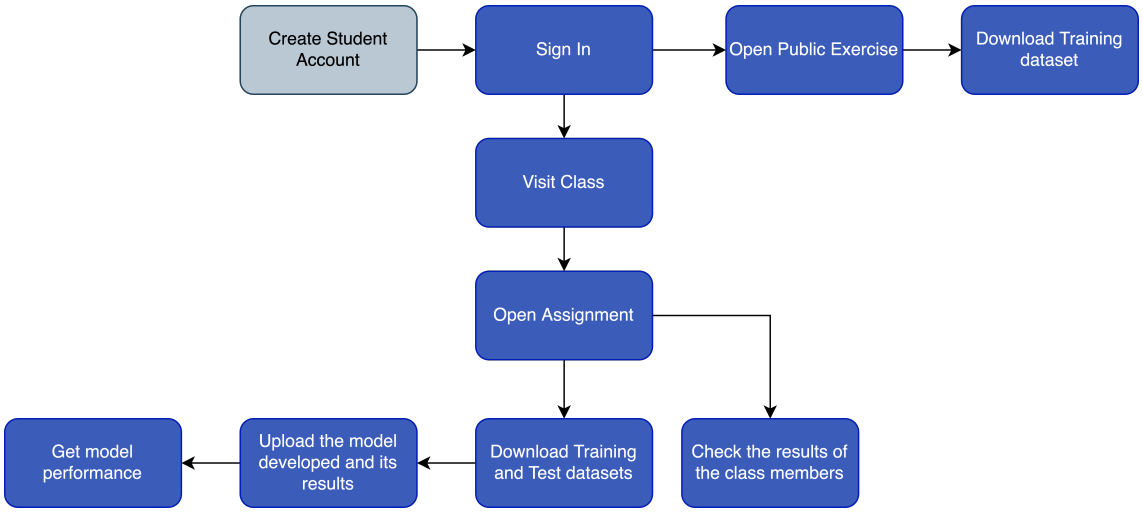

Figure 8.2: Student workflow

importance of planning and the importance of user feedback to improve the product.

8.2 Limitations

Throughout the development of the web app, we have encountered several limitations and challenges that have impacted the final product, being the most noticeable the Identity Provider of the University of Aveiro. This limitation had an impact on the overall implementation and we had to rethink some of the features that we wanted to implement.
To solve that problem we had to choose another authentication service and decided to use Google Firebase as that service. This decision had an influence on the final product, as we had to add a way for the user to register in the web app, which was not in the initial plan. Another problem with using this service is that it was only able to store the user credentials and not the type of user, which was a problem because we needed to know if the user was a student or a professor.
To solve this problem we had to create a way of storing the user type and the user id in the backend, which was not in the initial plan.
Even though this is a limitation, all the product functionalities were implemented and would be working the same way even if we had used the UA IdP, except for the user registration, which would not be needed and people can choose between being a professor and a student.

8.3 Future Work

Throughout the development of the web app, significant progress has been made in achieving the project's objectives. However, there are numerous opportunities for future work to further enhance and expand upon the current implementation through iterative testing and continuous incorporation of user feedback, such as the feedback gathered in the usability test session.

Furthermore, as the user base expands and the web app gains more traction, continuous monitoring of user analytics and metrics will provide valuable insights into user behaviour patterns, usage trends, and potential areas for optimisation. Analysing data such as click-through rates, user engagement, and conversion rates will enable data-driven decision-making and guide future iterations to align the web app more closely with user expectations.

By emphasising a user-centric approach and actively involving users in the development process, the web app can be continuously improved to better meet the needs and preferences of its target audience. Through a combination of usability testing, gathering user feedback, and leveraging data analytics, we can ensure that the web app remains responsive, intuitive, and aligned with user expectations.

References

[1] World Economic Forum. Waves of change: Understanding the driving force of innovation cycles. URL: https://www.weforum.org/agenda/2021/07/this-is-a-visualization-of-the-history-of-innovation-cycles. Accessed on March 18, 2023.

[2] Fortune Business - Machine Learning (ML) Market Size. URL: https://www.fortunebusinessinsights.com/machine-learning-market-102226. Accessed on March 18, 2023.

[3] Fortune Business - Artificial Intelligence (AI) Market Size. URL: https://www.fortunebusinessinsights.com/industry-reports/artificial-intelligence-market-100114. Accessed on March 18, 2023.

[4] Kathrin Blagec, Georg Dorffner, Milad Moradi, and Matthias Samwalda. A critical analysis of metrics used for measuring progress in artificial intelligence. pages 4–12, August 2020.

[5] Aceinfoway. URL: https://www.aceinfoway.com/blog/best-frontend-frameworks. Accessed on March 18, 2023.

[6] Coding Nomads. URL: https://codingnomads.co/blog/best-backend-frameworks/. Accessed on March 18, 2023.

[7] Statistics & Data. URL: https://statisticsanddata.org/data/the-most-popular-databases-2006-2023/. Accessed on March 18, 2023.

[8] OpenUP. OpenUP Requirement Gathering Techniques. URL: https://www.utm.mx/~caff/doc/OpenUPWeb/openup/guidances/guidelines/req_gathering_techniques_8CB8E44C.html. Accessed on May 20, 2023.

[9] Oracle. What is a Realational Database. URL: https://www.oracle.com/pt/database/what-is-a-relational-database/. Accessed on March 8, 2023.

[10] IBM. What are NoSQL databases. URL: https://www.ibm.com/topics/nosql-databases. Accessed on March 8, 2023.

[11] Mark Richards. *Software Architecture Patterns*. O'Reilly Media, Inc., Second edition, 2015.

[12] Google Cloud. Cloud Run Auto scalling. URL: https://cloud.google.com/run/docs/about-instance-autoscaling. Accessed on May 10, 2023.

[13] Google Cloud. Cloud StoragesServices Comapration. URL: https://cloud.google.com/blog/topics/developers-practitioners/map-storage-options-google-cloud. Accessed on April 14, 2023.

[14] Robert L. Mack Jakob Nielsen. *Usability Inspection Methods*. Wiley, 1994.

[15] MepML. API documentation. URL: https://documenter.getpostman.com/view/13973483/2s93Y5PKTx. Created on April 23, 2023.

[16] OpenUP. OpenUP Roles. URL: https://www.utm.mx/~caff/doc/OpenUPWeb/openup/rolesets/openup_roles_5CDDEEDA.html. Accessed on May 20, 2023.

[17] Gantt chart. URL: https://www.gantt.com/. Accessed on May 20, 2023.

[18] Slack. URL: https://slack.com/. Accessed on May 26, 2023.

[19] Zoom. URL: https://zoom.us/. Accessed on May 26, 2023.

[20] Jira Software. URL: https://www.atlassian.com/software/jira. Accessed on May 26, 2023.

[21] GitHub. URL: https://github.com/. Accessed on May 26, 2023.

[22] MEP-Org. Website for Communication. URL: https://mep-org.github.io/website-for-communication/. Accessed on May 26, 2023.

[23] MEP organisation. URL: https://github.com/MEP-org. Created on February 25, 2023.

[24] GitHub Repo for website for communication. URL: https://github.com/MEP-org/website-for-communication. Created on February 25, 2023.

[25] GitHub Repo for Prototype. URL: https://github.com/MEP-org/Prototype. Created on March 14, 2023.

[26] GitHub repo for Frontend. URL: https://github.com/MEP-org/MepML_Frontend. Created on April 27, 2023.

[27] GitHub Repo for Backend. URL: https://github.com/MEP-org/MepML_Backend. Created on February 26, 2023.

[28] GitHub Repo for slack bot. URL: https://github.com/MEP-org/SlackBot. Created on May 7, 2023.

[29] Wrike. What Is Agile Methodology in Project Management? URL: https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/. Accessed on May 26, 2023.

[30] Scrum. URL: https://www.scrum.org/. Accessed on May 26, 2023.

[31] Feature Driven Development. URL: https://www.productplan.com/glossary/feature-driven-development/. Accessed on May 26, 2023.

Appendix A

Use Cases Full Specification

Table A.1: UC-0 Sign up specification

ID	UC-0 Sign up
Creation date	2023-03-06
Use case	Sign up
Primary actor	Teacher / Student
Secondary actor	Firebase
Description	User creates an account in the system.
Trigger	User wants to access the system.
Preconditions	PRE-1: There is no account associated with User's email address.
Postconditions	POST-1: User is signed up to the system.
Normal flow	0.0 Sign up 1. User clicks on the "Sign in" button in the home page. 2. System redirects User to sign in page. 3. User clicks on the "Sign up" link, since he/she does not have an account yet. 4. User submits the form. 5. System redirects User to sign in page again.
Exceptions	0.0.E1 Sign up with invalid credentials 1. System highlights the invalid fields. 2. User corrects the invalid fields and submits the form again. 3. System redirects User to sign in page again.
Priority	Low

Table A.2: UC-1 Sign in specification

ID	UC-1 Sign in
Creation date	2023-03-06
Use case	Sign in
Primary actor	Teacher / Student
Secondary actor	Firebase
Description	User logs in to the system.
Trigger	User wants to access the system.
Preconditions	PRE-1: User has an account in the system.
Postconditions	POST-1: User is logged in to the system.
Normal flow	CaU 1.0 Sign in 1. User clicks on the "Sign in" button in the home page. 2. System redirects User to sign in page. 3. User submits the form with valid data. 4. The system redirects User to the dashboard page.
Exceptions	1.0.E1 Sign in with invalid credentials 1. System displays error ‘Invalid credentials’. 2. User corrects the invalid fields and submits the form again. 3. The system redirects User to the dashboard page.
Priority	Low

Table A.3: UC-2 Logout specification

ID	UC-2 Logout
Creation date	2023-03-06
Use case	Logout
Primary actor	Teacher / Student
Description	User logs out of the system.
Trigger	User wants to end the session.
Preconditions	PRE-1: User is logged in to the system.
Postconditions	POST-1: User is logged out of the system.
Normal flow	CaU 2.0 Logout 1. User clicks on his/her profile picture. 2. User clicks on the "Logout" button. 3. System logs User out. 4. System redirects User to the login page.
Priority	Low

Table A.4: UC-3 Create class specification

ID	UC-3 Create class
Creation date	2023-03-06
Use case	Create class
Primary actor	Teacher
Description	Teacher creates a class.
Trigger	Teacher wants to create a class.
Preconditions	PRE-1: Teacher is logged in to the system.
Postconditions	POST-1: The class is created.
Normal flow	CaU 3.0 Create class 1. Teacher navigates to the "Classes" page. 2. Teacher clicks on the "Create a new class" button or "Create one" link (if visible). 3. Teacher enters the class name and optionally uploads a class picture. 4. Teacher adds students. 5. Teacher clicks on the "Create class" button. 6. System redirects Teacher to the "Classes" page.
Priority	Medium

Table A.5: UC-4 Manage class specification

ID	UC-4 Manage class
Creation date	2023-03-06
Use case	Manage class
Primary actor	Teacher
Description	Teacher views, edits or deletes a class.
Trigger	Teacher wants to view, import/remove students, edit class details or delete it.
Preconditions	PRE-1: Teacher is logged in to the system. PRE-2: Teacher has created at least one class.
Postconditions	POST-1: The class could be edited or deleted.
Normal flow	CaU 4.0 View class 1. Teacher navigates to the "Classes" page. 2. Teacher clicks on the class he/she wants to view. 3. System displays the class details.
Alternative flows	CaU 4.1 Edit class 3. Teacher edits the class name and uploads a new class picture. 4. Teacher adds or removes students. 5. Teacher clicks on the "Save" button. 6. System redirects Teacher to the "Classes" page. CaU 4.2 Delete class 3. Teacher clicks on the bin button. 4. System redirects Teacher to the "Classes" page.
Priority	Medium

Table A.6: UC-5 Create exercise specification

ID	UC-5 Create exercise
Creation date	2023-03-06
Use case	Create exercise
Primary actor	Teacher
Description	Teacher creates a public exercise or an assignment (private exercise).
Trigger	Teacher wants to create an exercise.
Preconditions	PRE-1: Teacher is logged in to the system. PRE-2: Teacher has created a class.
Postconditions	POST-1: The exercise is created.
Normal flow	CaU 5.0 Create exercise 1. Teacher navigates to the "Exercises" page. 2. Teacher clicks on the "Create exercise" button or "Create one" link (if visible). 3. Teacher enters title, class and subtitle. 4. Teacher writes a long description or uploads the corresponding markdown file. 5. Teacher defines visibility, deadline date, attempts limit and performance metrics. 6. Teacher writes the evaluation rules or uploads the corresponding markdown file. 7. Teacher uploads the training and test datasets. 8. Teacher clicks on the "Create" button. 9. System notifies students on a Slack's custom workspace. 10. System redirects Teacher to the "Exercises" page.
Exceptions	5.0.E1 Teacher enters invalid exercise details 1. System detects that the exercise details are invalid. 2. System displays a modal with the missing/invalid fields. 3. Teacher corrects the fields and submits the creation form again.
Priority	Medium

Table A.7: UC-6 Manage exercise specification

ID	UC-6 Manage exercise
Creation date	2023-03-06
Use case	Manage exercise
Primary actor	Teacher
Description	Teacher manages a public exercise or an assignment (private exercise).
Trigger	Teacher wants to view, edit or delete an exercise.
Preconditions	PRE-1: Teacher is logged in to the system. PRE-2: Teacher has created at least one exercise.
Postconditions	POST-1: The exercise could be edited or deleted.
Normal flow	CaU 6.0 View exercise 1. Teacher navigates to the "Exercises" page. 2. Teacher clicks on the exercise he/she wants to view. 3. System displays the exercise details.
Alternative flow	CaU 6.1 Edit exercise 3. Teacher edits the exercise details. 4. Teacher clicks on the "Save" button. 5. System redirects Teacher to the "Exercises" page. CaU 6.2 Delete exercise 3. Teacher clicks on the bin button. 4. System redirects Teacher to the "Exercises" page.
Priority	Medium

Table A.8: UC-7 View assigned exercise specification

ID	UC-7 View assigned exercise
Creation date	2023-03-06
Use case	View assigned exercise
Primary actor	Student
Description	Student views the assigned exercise.
Trigger	Student has been notified of a new assignment.
Preconditions	PRE-1: Student is logged in. PRE-2: An exercise was assigned to one the student's classes.
Normal flow	CaU 7.1 View assigned exercise 1. Student navigates to the "Exercises" page. 2. Student clicks on the exercise he/she wants to view.
Priority	Medium

Table A.9: UC-8 Download provided datasets specification

ID	UC-8 Download provided datasets
Creation date	2023-03-06
Use case	Download provided datasets
Primary actor	Student
Description	If the exercise is public, student can download the train dataset. If the exercise is assigned to him/her, student can download the train and the incomplete test datasets, i.e., without the real labels that he/she is trying to predict.
Trigger	Student wants to solve an exercise.
Preconditions	PRE-1: Student is logged in. PRE-2: An exercise was assigned to one the student's classes.
Normal flow	CaU 8.0 Download provided datasets 1. Student navigates to the "Exercises" page. 2. Student goes to "Datasets" tab. 3. Student clicks on the download button(s).
Priority	Medium

Table A.10: UC-9 Upload a solution specification

ID	UC-9 Upload a solution
Creation date	2023-03-06
Use case	Upload a solution
Primary actor	Student
Description	Student uploads a solution to the assigned exercise.
Trigger	Student wants to upload a solution to the assigned exercise.
Preconditions	PRE-1: Student is logged in.
Postconditions	POST-1: A solution is uploaded, so metrics scores are made available.
Normal flow	CaU 9.0 Upload a solution 1. Student navigates to the "Exercises" page. 2. Student clicks on the exercise he/she wants to upload a solution to. 3. Student goes to the "Submission" tab. 4. Student uploads his/her results (predicted labels) and the code (Python / Jupyter Notebook). 5. Student clicks on the "Submit my answer" button. 6. System redirects Student to the "Results" tab.
Exceptions	CaU 9.0.E1 Results file is invalid 1. System displays an error message. 2. Student uploads a valid results file. 3. Student clicks on the "Submit my "answer" button. 4. System redirects Student to the "Results" tab.
Priority	Medium

Table A.11: UC-10 Check ranking specification

ID	UC-10 Check ranking
Creation date	2023-03-06
Use case	Check ranking
Primary actor	Student
Description	Student checks the ranking of the assigned exercise.
Trigger	Student wants to check how he/she is performing compared to other students.
Preconditions	PRE-1: Student is logged in. PRE-2: An exercise was assigned to one the student's classes. PRE-3: Student has submitted a solution.
Normal flow	CaU 10.1 Check ranking 1. Student navigates to the "Exercises" page. 2. Student clicks on the exercise he/she has solved. 3. Student goes to the "Results" tab and checks the bottom table.
Priority	Medium

Table A.12: UC-11 Check model evaluation specification

ID	UC-11 Check model evaluation
Creation date	2023-03-06
Use case	Check model evaluation
Primary actor	Student
Description	Student checks the evaluation of the model he/she has submitted.
Trigger	Student wants to check how his/her model is performing, before or after the deadline.
Preconditions	PRE-1: Student is logged in. PRE-2: An exercise was assigned to one the student's classes. PRE-3: Student has submitted a solution.
Normal flow	CaU 11.1 Check model evaluation 1. Student navigates to the "Exercises" page. 2. Student clicks on the exercise he/she has solved. 3. Student goes to the "Results" tab and see the metrics scores.
Priority	Medium

Table A.13: UC-12 View public exercise specification

ID	UC-12 View public exercise
Creation date	2023-03-06
Use case	View public exercise
Primary actor	Student
Description	Student views a public exercise.
Trigger	Student wants to learn ML by solving an exercise.
Preconditions	PRE-1: Student is logged in. PRE-2 There is at least one public exercise.
Normal flow	CaU 12.1 View public exercise 1. Student navigates to the "Exercises" page. 2. Student clicks on the exercise he/she wants to view.
Priority	Medium

Appendix B

Usability Tests Questionnaires

Agradecemos a sua colaboração neste estudo, que tem como intuito avaliar e melhorar a interface de utilizador do website MepML, no âmbito da unidade curricular "Projeto em Informática" da Universidade de Aveiro.

Ao responder ao presente formulário, **está a consentir** o tratamento dos dados recolhidos durante a experiência.

Consentimento informado

Procedimento
Os participantes realizarão um conjunto de tarefas, utilizando uma aplicação Web para interagir com o sistema. Durante a experiência, serão recolhidos dados sobre os seus comentários e dificuldades na execução das tarefas.

Duração
A experiência terá uma duração máxima de 15 minutos.

Riscos para o participante
Não existem riscos para o participante.

Benefícios para o participante
Enquanto alunos e professores que lidam com problemas de Machine Learning, os participantes são potenciais utilizadores da plataforma em estudo.

Confidencialidade
Todos os dados recolhidos durante a experiência serão anónimos e confidenciais e só serão utilizados sob esta forma para análise e discussão do sistema baseado na Web previsto.

Participação voluntária
A sua participação é totalmente voluntária. Mesmo que concorde em participar, pode interromper a sua participação em qualquer altura, bastando para tal comunicar a sua vontade aos observadores. Nesse caso, todos os dados recolhidos até essa altura serão eliminados.

Contacto
Para qualquer questão relativa a esta experiência, contactar um dos seguintes elementos:

- Diogo Magalhães - d.magalhaes@ua.pt
- Emanuel Marques - emanuel.gmarques@ua.pt
- Leonardo Almeida - leonardoalmeida7@ua.pt
- Pedro Rodrigues - pedrofrodrigues4@ua.pt
- Rafael Gonçalves - rfg@ua.pt

Figure B.1: Usability Test Consent

Dados do Utilizador

1. Como classifica o seu nível de conhecimento em tópicos de Machine Learning?
2. Como classifica a sua experiência em plataformas digitais de Machine Learning (e.g. Kaggle, CodaLab, Weights and Biases, Machine Hack, etc.)?

Tarefas a Realizar - Aluno

Esta secção apresenta um conjunto de tarefas que deverá completar. Após concluir cada tarefa, deve classificar o nível de dificuldade que sentiu.

1. Inicie sessão na plataforma com as credenciais fornecidas.
2. Selecione uma das turmas a que pertence. Verifique quantos membros a turma possui e o email do docente responsável pela turma.
3. Aceda à lista de "Assignments" atribuídas pelos seus professores e selecione a mais recente.
4. Descarregue o dataset de treino e de teste associados ao exercício.
5. Submeta os resultados obtidos pelo seu modelo (predictions.csv), bem como o próprio modelo (drugsML.ipynb), de modo a medir a sua performance.

Tarefas a Realizar - Professor

Esta secção apresenta um conjunto de tarefas que deverá completar. Após concluir cada tarefa, deve classificar o nível de dificuldade que sentiu.

1. Inicie sessão na plataforma com as credenciais fornecidas.
2. Crie uma turma com o nome "T1" unicamente com um aluno, cujo número mecanográfico é 123456.
3. Consulte as métricas já existentes na plataforma.
4. De modo a medir a performance dos modelos dos seus alunos, crie uma métrica personalizada com o nome "All ones", com a descrição "This is a dummy metric", cujo seu comportamento é definido pela função presente no ficheiro "dummy metric.py".
5. Crie um exercício com as seguintes informações:
   (a) Title: "Drug Classification"
   (b) Subtitle: "The goal is to predict the drug type that should be prescribed to a patient based on features"

(c) Description: Upload do ficheiro de descrição "description.md"

(d) Visibility: Private

(e) Evaluation Rules: —

(f) Deadline: 20/05/2023

(g) Attempts: Unlimited

(h) Metrics: 'Precision', 'Accuracy'

(i) Upload training set: Upload do ficheiro "train.csv"

(j) Upload test set: Upload do ficheiro "test.csv"

6. Futuramente, quando os seus alunos submeterem os seus modelos poderá analisar a sua performance e efetuar o download dos mesmos. Para o exercício criado anteriormente, dirija-se até à zona dos resultados (neste momento não existem resultados disponíveis.).

Sugestões após utilização do sistema

1. Considera que existem certos apetos relativos à usabilidade do sistema que devem ser melhorados? Se sim por favor indique-os em baixo.

Opinião geral sobre o sistema

Após a utilização do sistema e tendo em conta a sua avaliação final, selecione a opção que melhor reflecte a sua opinião em relação à utilização do sistema. Caso considere que estas quantificações não são aplicáveis, não responda.

1. É fácil orientar-me no sistema.
2. Encontro facilmente o que procuro no sistema.
3. O sistema é lento.
4. O sistema é agradável de utilizar.
5. Sinto necessidade de ajuda em algumas funcionalidades.
6. A utilização do site requer conhecimentos mais aprofundados ou experiência anterior.

Opinião sobre aspectos específicos do sistema

1. O tamanho dos caracteres no ecrã torna-os fáceis de ler.
2. A informação mais importante possui um bom destaque.

3. A quantidade de informação que pode ser apresentada por ecrã é adequada.
4. A disposição da informação que pode ser apresentada por ecrã é adequada.
5. Os ícones apresentados são intuitivos.
6. O aspecto gráfico é atrativo.
7. É fácil navegar no sistema.

Aspetos positivos do sistema

1. Indique os principais pontos que considera positivos da plataforma.

Appendix C

Usability Tests Student Results

Table C.1: Average classification for the first part questions

Question	Average Classification (1-5)
Como classifica o seu nível de conhecimento em tópicos de Machine Learning?	3.50
Como classifica a sua experiência em plataformas digitais de Machine Learning (e.g. Kaggle, CodaLab, Weights and Biases, Machine Hack, etc.)?	2.75

Table C.2: Average difficulty to perform each student task

Task	Average Difficulty (1-5)
Inicie sessão na plataforma com as credenciais fornecidas.	1.00
Selecione uma das turmas a que pertence. Verifique quantos membros a turma possui e o email do docente responsável pela turma.	1.00
Aceda à lista de "Assignments" atribuídas pelos seus professores e selecione a mais recente.	1.00
Descarregue o dataset de treino e de teste associados ao exercício.	1.00
Submeta os resultados obtidos pelo seu modelo (predictions.csv), bem como o próprio modelo (drugsML.ipynb), de modo a medir a sua performance.	1.00

ID ↑	Nome	Respostas
1	anonymous	Mais métricas de pesquisa (principalmente tipo de problema: regressão, classificação, etc.). Possibilidade de um estudante obter avaliação dos trabalhos públicos, para avaliação pessoal (mesmo que não haja interesse de submeter ao professor).
2	anonymous	No que diz respeito aos exercícios, seria interessante existir mais secções de modo a diminuir as opções, como por exemplo área de estudo, linguagem, departamento, entre outras. No global considero que a plataforma é intuitiva e bastante útil.

Figure C.1: Suggestions

Table C.3: Average classification for each usability question

Usability Question	Average Classification (1-5)
É fácil orientar-me no sistema.	4.75
Encontro facilmente o que procuro no sistema.	4.75
O sistema é lento.	2.00
O sistema é agradável de utilizar.	5.00
Sinto necessidade de ajuda em algumas funcionalidades.	1.25
A utilização do site requer conhecimentos mais aprofundados ou experiência anterior.	1.75
O tamanho dos caracteres no ecrã torna-os fáceis de ler.	5.00
A informação mais importante possui um bom destaque.	4.25
A quantidade de informação que pode ser apresentada por ecrã é adequada.	4.50
A disposição da informação que pode ser apresentada por ecrã é adequada.	4.50
Os ícones apresentados são intuitivos.	4.25
O aspecto gráfico é atrativo.	4.25
É fácil navegar no sistema.	5.00

ID ↑	Nome	Respostas
1	anonymous	Fácil usabilidade e boa funcionalidade.
2	anonymous	Acho que a plataforma é bastante intuitiva, e graficamente atrativa. Na parte dos assignments talvez dar para ver a classificação do aluno no exercicio submetido seria interessante.
3	anonymous	Fácil e intuitivo.

Figure C.2: Positive aspects

Appendix D

Usability Tests Professor Results

Table D.1: Average classification for the first part questions

Question	Average Classification (1-5)
Como classifica o seu nível de conhecimento em tópicos de Machine Learning?	4.20
Como classifica a sua experiência em plataformas digitais de Machine Learning (e.g. Kaggle, CodaLab, Weights and Biases, Machine Hack, etc.)?	4.00

Table D.2: Average difficulty to perform each student task

Task	Average Difficulty (1-5)
Inicie sessão na plataforma com as credenciais fornecidas.	1.00
Crie uma turma com o nome "T1" unicamente com um aluno, cujo número mecanográfico é 123456.	2.60
Consulte as métricas já existentes na plataforma.	1.0
Crie uma métrica personalizada com o nome "All ones", com a descrição "This is a dummy metric".	1.80
Crie um exercício.	1.60
Dirija-se até à zona dos resultados.	1.40

ID ↑	Nome	Respostas
1	anonymous	Facilitar o separador para as metricas, tentar apresentar de outra forma.
2	anonymous	O sistema esta bem pensado. Poucos aspectos de detalhas podem ser melhorados.
3	anonymous	Metrics deviam estar destacada num tab separado, não no meio dos exercícios. Criar o link para inscrição em turma/exercício. Tenho 2 turmas com o mesmo nome de forma não destinguiveis, deviam mostrar sempre o ano. Seria interessante também ter algum género de Highscore entre alunos. Gostava de ter este Highscore com ou sem mostrar o código, por exemplo como professor, posso gostar que a turma veja os highscore entre eles, mas sem que possam copiar código entre eles (tal como acontece no kaggle).
4	anonymous	Introduzir mensagem de aviso para sair da criação de turma (sair sem guardar)
5	anonymous	Para criar a turma e criar o exercício, por exemplo, dar mais ênfase ao botão "criar" no fim. Pode ser facilmente esquecido. Para as métricas, também não era óbvio que estariam na secção "Exercise". Se calhar adicionar um filtro de pesquisa para isso.

Figure D.1: Suggestions

Table D.3: Average classification for each usability question

Usability Question	Average Classification (1-5)
É fácil orientar-me no sistema.	4.75
Encontro facilmente o que procuro no sistema.	4.75
O sistema é lento.	2.00
O sistema é agradável de utilizar.	5.00
Sinto necessidade de ajuda em algumas funcionalidades.	1.25
A utilização do site requer conhecimentos mais aprofundados ou experiência anterior.	1.75
O tamanho dos caracteres no ecrã torna-os fáceis de ler.	5.00
A informação mais importante possui um bom destaque.	4.25
A quantidade de informação que pode ser apresentada por ecrã é adequada.	4.50
A disposição da informação que pode ser apresentada por ecrã é adequada.	4.50
Os ícones apresentados são intuitivos.	4.25
O aspecto gráfico é atrativo.	4.25
É fácil navegar no sistema.	5.00

ID ↑	Nome	Respostas
1	anonymous	Apresentação, Simplicidade
2	anonymous	Boa e intuitiva apresentação. Fácil de entender.
3	anonymous	Gosto é simples e intuitivo. Gosto especialmente de poder colocar em dark mode.
4	anonymous	Acessibilidade da informação, feedback rápido de professores para alunos
5	anonymous	Rápido, prático, e razoavelmente intuitivo.

Figure D.2: Positive aspects

Appendix E

Contributions

Table E.1: Workloads by student

	Diogo Oliveira Magalhães 102470 (QA Tester)	**Rafael Fernandes Gonçalves 102534 (Product Owner)**	**Leonardo Almeida 102536 (Team Manager)**	**Emanuel Gaspar Marques 102565 (DevOps Master)**	**Pedro Henrique Figueiredo Rodrigues 102778 (Architect)**
Frontend Aluno	-	-	20%	-	80%
Frontend Professor	-	-	80%	-	20%
Backend Tests	100%	-	-	-	-
Backend Endpoints	35%	30%	-	35%	-
Backend Sandbox	-	100%	-	-	-
Backend SlackBot	-	-	100%	-	-
Backend Documentation	-	-	-	100%	-
Usability tests Forms	-	20%	-	-	80%
Deployment	-	-	10%	90%	-
Video	5%	75%	5%	10%	5%
Poster	5%	80%	5%	5%	5%
Presentations	20'%	20%	20%	20%	20%
Report Chapters	2, 7.1, 7.2, 8.2	abstract, 1, 3, 5.3.2	3.5, 5.3.3, 6, 8.1, 8.3	3.1.5, 4.1, 4.3, 5.2, 5.3.1, 5.4	3.1.2, 4.2, 5.1, 7.3
Overall	**20%**	**20%**	**20%**	**20%**	**20%**