NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

RUI MARQUES FERNANDES

Bachelor in Computer Science and Engineering

AUDIT AI

CO-DESIGNING WEB PLATFORM FOR EXTERNAL ALGORITHM AUDITING

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
November, 2021

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

AUDIT AI

CO-DESIGNING WEB PLATFORM FOR EXTERNAL ALGORITHM AUDITING

RUI MARQUES FERNANDES

Bachelor in Computer Science and Engineering

Adviser: Francisco Nunes
Senior Scientist, Fraunhofer Portugal AICOS

Co-adviser: Sofia Cavaco
Assistant Professor, NOVA School of Science and Technology

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
November, 2021

Audit AI

Acknowledgements

Firstly, I would like to express my gratitude towards Francisco Nunes, who provided me all the means and support I needed in order to develop my dissertation. I would also like to thank the Human-Centred Design group of Fraunhofer Portugal AICOS for all the feedback and support given since the beginning. Additionally, I would like to thank the lecturers from the Department of Computer Science from NOVA SST, especially professor Sofia Cavaco, who gave me some valuable insights.

Invariably, I wouldn't be able to finish my dissertation without the support and motivation from my colleagues, family, and girlfriend, who always helped me achieve my goals.

"Try not to become a man of success, but rather try to become a man of value." (Albert Einstein)

Abstract

Artificial Intelligence (AI) algorithms are becoming more crucial and decisive every day as the promise that these systems are able to mitigate human biases intensifies, thus providing fair opportunities to everyone. When unchecked, algorithms might be subjected to bias and discriminatory outcomes, principally due to the use of biased datasets or even by defective implementations of the algorithms. Notwithstanding, there has not been a follow-up regarding the monitorization of the algorithmic systems, despite the increasing tendency in the use of these systems during decision-making processes. In order to increase the trustworthiness of these algorithms, auditing practices are starting to gain some relevance among companies, researchers, and developers. Nonetheless, the amount of auditing practices performed is still nowhere close the amount needed.

Algorithmic auditing practices can prevent abuses by unethical stakeholders, providing greater scrutiny to the algorithms by searching for discrimination and bias. However, ensuring that AI algorithms reflect societal values and do not present any severe case of bias and discrimination is not easy nor simple, considering that most of these algorithms are confidential and not easily accessible. The owners of the algorithms choose not to share their algorithm's data to protect their intellectual propriety, which often took years to create. This lack of access to the algorithms' source code, datasets, and results does not allow the auditor to inspect the algorithm entirely, thus making the auditing task far more difficult.

This work aims to support the auditing of algorithms without accessing their source code by proposing a solution based on a web platform. The goal is to allow developers to connect their algorithms with the platform, which the auditors can then test using their datasets. So, it is possible to maintain the "invisibility" of the algorithms' source code, but at the same time, allow them to be audited. The platform was built based on interviews with developers, auditors, and researchers working in this space, which allowed us to understand and design an adequate solution that fulfilled the user's requisites.

Keywords: Auditing, Artificial Intelligence, Machine Learning, Algorithms

Resumo

Os algoritmos de Inteligência Artificial desempenham cada vez mais um papel crucial e decisivo na nossa sociedade, existindo a perspetiva de que através da sua utilização seja possível reduzir a influência do preconceito do ser humano durante os processos de decisão, permitindo assim, maior igualdade de oportunidades para todos os envolvidos. No entanto, se auditar algoritmos fosse uma prática comum, verificar-se-ia regularmente que estes algoritmos também apresentam algum grau de discriminação e preconceito, devido à utilização de conjuntos de dados enviesados, ou simplesmente por implementações defeituosas dos próprios algoritmos. Não obstante, na prática, não existe uma verificação adequada dos algoritmos, quer por desconhecimento relativamente à prática de auditoria dos mesmos, quer por incapacidade e dificuldade em realizá-la.

Através da auditoria de algoritmos é possível precaver abusos de poder de certos intervenientes, bem como garantir um maior escrutínio no que diz respeito a situações de discriminação ou desigualdade. Realizar auditorias de algoritmos é uma tarefa extremamente complexa e exigente visto que a maior parte dos algoritmos são confidenciais, ou seja, o seu código fonte e datasets não estão disponíveis ao público. Os responsáveis pelos algoritmos optam por tomar estas medidas com o intuito de proteger a sua propriedade intelectual, no entanto, a inacessibilidade a estas informações complica o trabalho dos auditores, uma vez que estes não conseguem desta forma analisar por completo o sistema.

Seguindo esta ordem de ideias a nossa proposta de solução consiste na criação de um protótipo de uma plataforma Web, onde é possível aos programadores introduzir as informações dos seus algoritmos, incluindo os endpoints para auditá-los, mantendo assim a "invisibilidade" dos mesmos, mas ao mesmo tempo, permitindo que estes sejam auditados. O protótipo foi desenvolvido com base no feedback e ideias dos auditores e programadores, que foram os principais utilizadores do sistema.

Palavras-chave: Auditoria, Inteligência Artificial, Aprendizagem Automática, Algoritmos

Contents

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence vi, xi, 1, 2, 4, 6, 7, 10, 11, 13, 16, 17, 20

AIAs Algorithmic Impact Assessments 7

API Application Programming Interface 2, 5, 34, 35, 37, 39, 40, 41, 42, 44, 46, 47, 50, 55

GDPR General Data Protection Regulation 2, 7, 30

HCI Human-Computer Interaction 18

ICO Information Commissioner's Office xi, 6, 16

IEEE Institute of Electrical and Electronics Engineers 4

JWT JSON Web Token 40, 41

ML Machine Learning 1, 4, 13, 14, 16

UCD User-Centered Design xii, 3, 18, 19

1 Introduction

1.1 Motivation and Context

Governments, companies, and societies are increasingly relying on algorithmic systems to make significant decisions in areas such as employment, financial credit, criminal justice, government resource allocation, among others [KCL20; Kim17]. One of the principal premises for using algorithms in the decision-making processes is the reduction of the influence of human bias in the final decision, thus allowing equal opportunities to all the subjects involved. This premise is adopted and accepted since it exists a concept that algorithms follow pre-determined rules in every situation, and so, they do not discriminate according to the input received [Ske20]. Nevertheless, and considering the increasing tendency in the use of Artificial Intelligence (AI) algorithms during decision-making processes, there has not been a follow-up in terms of verification and monitorization of the respective algorithms, thus leaving them exposed to ethical infractions, such as biased outcomes, disparate impacts on different groups, and even spread of rumors and disinformation. In theory, algorithms are supposed to be fair, regardless of the person affected. Nonetheless, in practice, this attribute is not always verified, thus leading to the creation of mechanisms to ensure it, such as testing the algorithm for specific situations to look for discriminatory outcomes [Gus+18; KCL20]. Amazon recruitment tool, which produced biased outcomes against women during the hiring processes, or the COMPAS justice software, where different ethnic groups had disparate sentences for similar crimes, are famous illustrations of how the use of biased Machine Learning (ML) models and datasets not only do not fulfill the promise of equality but instead, reinforce the inequality, affecting directly the users involved [20b; Das18; Rah20] .

Guaranteeing that societal values (such as accountability, fairness and honesty) are present in the design of the algorithms, and that bias and discrimination are not a factor in the ML models and AI algorithms is not simple. Technical tools alone cannot prevent discriminatory or biased outcomes, since the causes of bias often lie outside the source code [Kim17]. Nonetheless, even within the source code of the algorithm, it is natural to find the reflection of the developers' values and assumptions, thus adding another

possible cause of bias to the algorithmic system [BL94]. Considering all of these factors, it is possible to understand that algorithmic auditing assumes a crucial role in order to achieve the goal of non-discriminatory and free of bias models and algorithms.

Algorithmic auditing can prevent unethical abuses by improper authors, providing greater scrutiny to the algorithm through the detection of unintended bias or harmful discrimination cases. All these arguments make algorithmic auditing a relevant method to promote transparency. Some regulations, like General Data Protection Regulation (GDPR) for example, have started to discuss topics about explainability and algorithm transparency (which highly correlates with the algorithmic auditing process), consequently increasing the attention and focus on these questions. GDPR, for example, requires organizations to be able to explain their algorithmic decisions, when requested to do so [GF17; Gus+18].

Even though awareness for potential discriminations of AI algorithms has increased, there is still little understanding on how to address the challenges of bias and discrimination. To summon this idea, by citing Galdon Clavell, CEO of Eticas Consulting, "the main barrier at the moment is people don't know that algorithmic auditing exists", which highlights the importance of this topic [Ske20].

1.2 Problem Definition

One of the major problems in the algorithmic auditing field is the lack of access to the algorithms. Usually, governments and companies do not disclose the source code of their algorithms (thus the denomination "black-box algorithms") in an attempt to protect their intellectual property, which often takes years to create or even to avoid exposing fundamentals and ideas the algorithm encodes [Gus+18]. Regardless of this fact, the auditing community still recognizes that the ability to access the source code of an algorithm does not make it interpretable. Algorithms affecting a substantial percentage of the population are generally translated into complex code packages, developed by different researchers, and cannot be evaluated through a simple code inspection. Although, it is important to refer that it might exist cases where access to the source code could be useful during the auditing process, like the discovery of badly implemented functions, for instance [GEB14; Kim17].

In addition to algorithms' source code being concealed most of the time, it can also be very tough to communicate with them. Most of the protected algorithms, especially algorithms used in government decisions, do not have public access like an Application Programming Interface (API) for example, thus making algorithmic auditing almost impossible in these situations.

1.3 Goals and proposed solution

Given the context and problems involved in the algorithmic auditing processes, the main goal of this thesis project is to make the auditing of black-box algorithms simple and efficient so that it is possible to fulfill the requisites of both auditors and developers. The proposed solution consists of a web-platform prototype, where researchers can submit the details and endpoint of their algorithms, thus maintaining the intellectual property and specifications untouched. With the algorithm details, including the endpoint, auditors can navigate through the different available algorithms in the platform and audit them through the input of their private datasets and by observing the output created. In this way, it is possible to audit the algorithms while maintaining the source code of the algorithms confidential, which are the principal goals of both auditors and developers, respectively.

Moreover, it is significant to notice that the prototype intended to satisfy the users' needs, which could be accomplished through User-Centered Design (UCD) processes, with special attention to in-depth interviews with the users. Below, it is possible to visualize the summary of the principal contributions of the dissertation:

- This dissertation introduces a prototype of a new external algorithmic auditing tool that supports the auditing of algorithmic systems using auditors' own datasets and without having to access the source code of the algorithms. The prototype was crucial to transmit the ideas of the system's functioning.
- This dissertation offers an analysis of different in-depth interviews performed with scientists and auditors, which were executed to gather more information about the problems of the algorithmic auditing field and receive feedback about the design of the prototype.
- Finally, this dissertation provides a web platform for external algorithmic auditing, where developers can introduce their algorithms' details, including the endpoint used to interact with the algorithm. The algorithms are then available to be audited by auditors, who can select the appropriate algorithms and introduce their own datasets to audit the algorithms. Attached to the illustration of the system is also described the complete interaction between the multiple components.

Together with Fraunhofer Portugal [22] we focused on this subject due to its increasing relevance in society. We wanted to create a tool that could support the auditing of algorithms while having the needs of its users always in consideration.

2

Literature Review

2.1 Algorithm Auditing and Algorithmic Impact Assessment

Algorithm auditing and algorithmic impact assessment are two approaches used in the fairness, accountability, and transparency fields of Artificial Intelligence (AI) algorithms and Machine Learning (ML) models [20d]. Both methodologies are usually connected. Nevertheless, their interpretation may differ according to the context where they are used. It is significant to mention that in the context of this thesis, the usage of the concepts models and algorithms is indistinct.

2.1.1 Algorithm Auditing

The Institute of Electrical and Electronics Engineers (IEEE) standard for software development defines auditing as "an independent evaluation of conformance of software products and processes to applicable regulations, standards, guidelines, plans, specifications, and procedures" [Raj+20]. Following this definition, we can consider algorithmic auditing as a method performed by independent third parties, such as external auditors or auditing companies, used to ascertain if the algorithms comply with all the legal and ethical procedures required.

Algorithmic auditing has become a major weapon to achieve transparency, accountability, and trustworthiness of AI algorithms [Kim17]. There are usually two different versions of algorithmic auditing: the first is done from the point of view of the software engineer, it focuses on testing particular cases and hypothesis about a system, through the observation of inputs and outputs, to ascertain the amount of bias or to check if it exists discrimination in the results, this version of auditing is called **bias audit**; the second version is performed with the perspective of the word audit in its more common sense, it is an approach focused in verifying if the algorithms are behaving according to the rules and norms predefined, it is referred as **regulatory inspection** [20d].

2.1.1.1 Bias Audit

Bias auditing processes are usually conducted by external and independent agents and are performed mostly on algorithms already deployed and in use [20d]. During the auditing process, the auditor focuses mainly on comparing the data introduced in the system, the input, and data returned by the system, the output. As a result of output observation and comparison, it is possible to verify the presence of cases related to discrimination or bias [20d]. It is significant to mention that this version of auditing has great relevance in the auditing of black-box algorithms since it is done without accessing its source code.

Nevertheless, bias auditing also has its downsides. One of them is the need for a concrete hypothesis to test, meaning, a particular use case, or feature. Such hypotheses are frequently very limited characteristics, for example, auditing to test for racial bias, age discrimination, amongst others. This means that bias auditing alone cannot provide a complete picture of the system [20d; San+14].

The majority of bias auditing reports were made by independent researchers or journalists, being one of the most famous examples, the 'Gender Shades' paper by Buolamwini and Gebru (see Figure 2.1) [BG18], who decided to audit three commercial facial recognition Application Programming Interface (API)s. The reason for auditing these APIs was the discovery that the accuracy rates were different when applied to diverse ethnic groups [BG18].

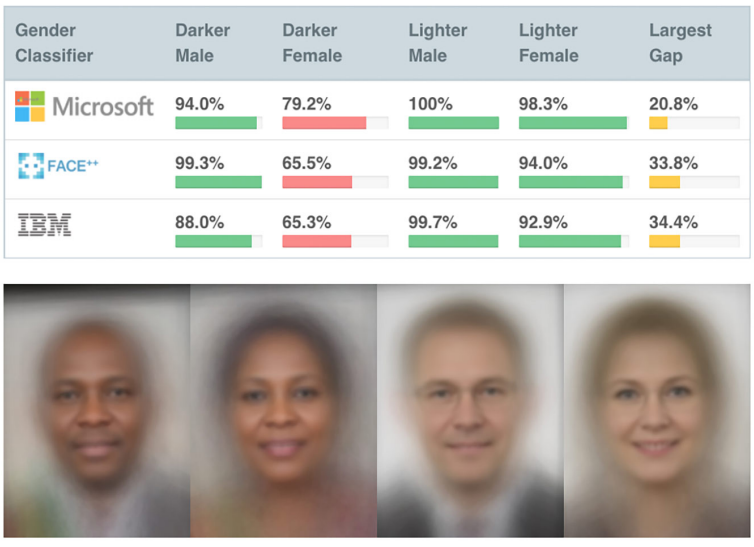

Figure 2.1: Results obtained from the auditing performed on three facial recognition algorithms. The figure illustrates the overall accuracy of the three audited algorithms on all subjects present in the pilot parliaments benchmark (2017). It is possible to notice the gap of accuracy according to the gender and race attributes [BG18] .

2.1.1.2 Regulatory Inspection

A regulatory inspection is an algorithmic auditing approach mainly used to verify if the algorithm complies with data protection laws, legislation, or industry requirements [20d]. This auditing technique[1] requires the collaboration of developers responsible for deploying the algorithm, or if the system is yet in development, the programmers in charge. In order to be robust, a regulatory inspection must not be limited to examining the source code, datasets, inputs, and outputs. It must as well, take into consideration the context where the algorithms operate [20d; Kim17].

Nevertheless, this methodology also presents some disadvantages: firstly, there is no standardized approach to perform regulatory inspections since the context where the algorithms are used differ immensely; secondly, as described above, transparency between entities and auditors is a must, and as already discussed, many companies and governments prefer to maintain the algorithm source code invisible, the denominated black-box algorithms, thus making it extremely hard for auditors to execute a regulatory inspection (see Figure 2.2) [20d].

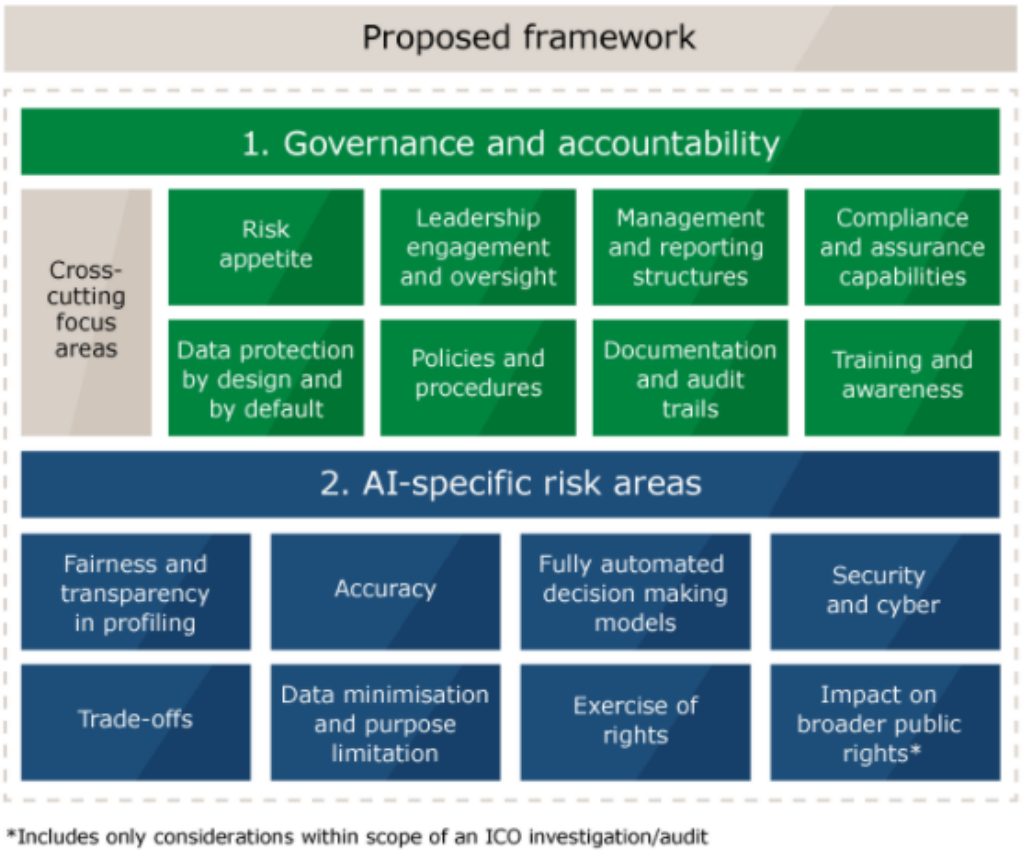

Figure 2.2: UK Information Commissioner's Office (ICO) proposed framework to audit AI algorithms [20a]

[1] This version of auditing is the most common within companies and governments. It is performed during the diverse development stages of the algorithm.

2.1.2 Algorithmic Impact Assessments

Algorithmic Impact Assessments (AIAs) aim to shape how AI algorithms are constructed, monitored, and governed from the beginning, thus encouraging the development of transparent and accountable systems [Koe19; Sec20]. AIAs rely mainly on evaluations elaborated in distinct fields, such as environmental policy, privacy law, and data protection, to assess if the algorithms comply with present regulations, like General Data Protection Regulation (GDPR) for instance. These assessments provide developers and researchers a measure to evaluate the AI solutions from an ethical point of view, to build transparent and responsible algorithmic systems [OS18; Sec20]. Furthermore, it is important to refer that AIAs depend on the context and specifications where the algorithmic systems are applied [Koe19]. Notwithstanding, the meaning of AIAs may differ depending on the algorithm lifecycle stage where they are applied.

There are two versions of AIAs, the first version is the **Algorithmic risk assessments**, which are mainly used before the deployment of an algorithmic system. Algorithm risk assessments determine the main areas where the algorithm will have a greater impact and the risks associated with the system. Algorithmic risk assessments generally try to be holistic by looking not only at the data and model but also through the investigation on how the system will be used, and by whom [20d]. The second approach of AIAs is the **Algorithmic impact evaluations**. This approach is conducted after the algorithmic systems have been deployed. It focuses more on the effects (economic or political, for example) of the system in a particular population. Algorithmic impact evaluations have particular pertinence in the public sector since there is enough evidence within the population to evaluate the impact of the algorithmic systems used. Meanwhile, in the private sector, this evidence is much more challenging to gather [20d; OS18] .

As mentioned before in this chapter, algorithm audits and algorithm impact assessments are applied in different circumstances, and their meaning may differ (see Figure 2.3). However, in the context of this thesis, we will not take into consideration the differences between both versions of algorithm auditing nor both versions of algorithm impact assessments. Hence, leading us to include the four concepts when referring to algorithm auditing.

	Algorithm audits		Algorithmic impact assessments	
	Bias Audit	Regulatory inspection	Algorithmic risk assessment	Algorithmic impact evaluation
What?	A targeted approach focused on assessing algorithmic systems for bias	A broad approach focussed on an algorithmic system's compliance with regulation or norms, and requiring a number of different tools and methods	Assessing possible societal impacts of an algorithmic system before the system is in use (with ongoing monitoring advised)	Assessing possible societal impacts of an algorithmic system on the users or population it affects after it is in use
When?	After deployment	After deployment, potentially ongoing	Before deployment, potentially ongoing	After deployment
Who by?	Researchers, investigative journalists, data scientists	Regulators, auditing and compliance professionals	Creators or commissioners of the algorithmic system	Researchers, policymakers
Origin	Social science audit studies	Regulatory auditing in other fields e.g. financial audits	Environmental impact assessments, data protection impact assessments	Policy impact assessments, which typically are evaluative after the fact
Case study	'Gender shades' study of bias in classification by facial recognition APIs	UK Information Commissioner's Office AI auditing framework draft guidance	Canadian Government's algorithmic impact assessment	Stanford's 'Impact evaluation of a predictive risk modeling tool for Allegheny County's Child Welfare Office'
Status	More established methodology in algorithm context; limited scope	Emerging methodology, skills and capacity requirements for regulators, more established approaches for compliance teams in tech sector	Some established methodologies in other fields, new to algorithm context; requiring evidence as to its applicability and best practice	Established methodology new to algorithm context; requiring evidence as to its applicability and best practice

Figure 2.3: Overview of the different tools used to assess algorithmic systems [20d]

2.2 Internal Auditing *vs* External Auditing

As we have already seen, algorithmic auditing processes can be divided into two different methodologies, bias auditing, an approach more focused on the technical part, and regulatory inspections, related to the legal side. Nevertheless, algorithmic auditing processes can also be separated on whether the processes are conducted by the researchers and developers responsible for the algorithmic system, the denominated internal auditing, or by third parties, external auditing.

Internal audits are often performed by companies and governments before deploying the algorithmic system, hence making it possible to act proactively instead of reactively and even allowing the discovery of hidden bugs or risks [20b]. Considering that most companies, like Google, for example, do not provide information about the algorithms to external auditors, in an attempt to protect intellectual property and maintain business confidentiality, internal audits are becoming a crucial step towards the development of transparent and accountable systems (see Figure 2.4) [20b; Raj+20] .

Conversely, external audits are conducted by independent third parties, with diminutive knowledge of the details and internal processes of the algorithmic system, like the source code or datasets [2] used for training the algorithm, for example. Notwithstanding, despite all of its limitations, external auditing is still an essential strategy to detect unintended bias and to reduce the discriminatory effects of the algorithms already in use [Kim17].

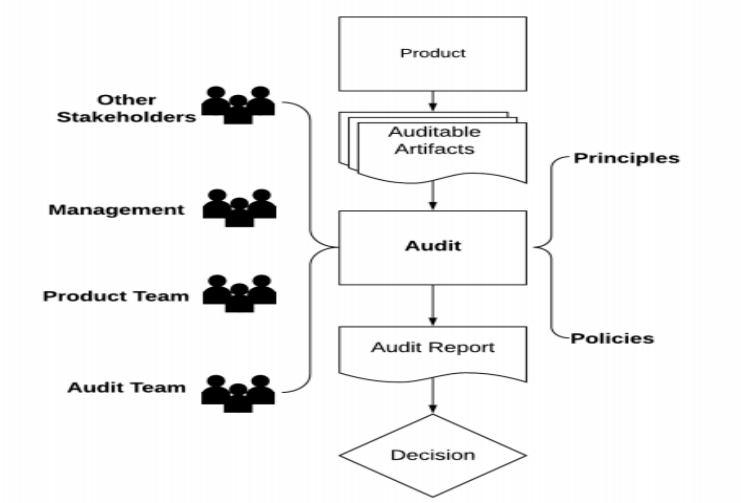

Figure 2.4: High-level overview of the context of an internal algorithmic audit. The audit is conducted during product development and prior to launch. The audit team leads the product team, management and other stakeholders in contributing to the audit. Policies and principles, including internal and external ethical expectations, also feed into the audit to set the standard for performance. [Raj+20]

[2] Datasets are collections of similar data sharing the same structure, which are then used by computers as units for analytic and predictive purposes [Syd21].

2.3 Reasons to Audit Algorithms

As highlighted previously, algorithmic systems are increasingly being used as part of the decision-making processes, both in the public and private sectors [Koe19]. While algorithmic systems are frequently seen as weapons to fight inequality, it was already evidenced that sometimes, these systems can reproduce or amplify bias and discriminations, spread rumors and disinformation, and even take control of people's attention [20b; Gus+18]. These failures must not be handled lightly, considering the impact AI algorithms have on our society. Algorithms that affect a large number of people, mainly those who are used in safety-critical fields, like health or advocacy, when poorly designed, can have a tremendous impact on the life of the users, considering that the failures of the algorithms in question do not result in easy to fix bugs or issues, their decisions can affect immensely our present and future, like deciding whether we are given financial credit or even judicial sentences for instance.

Citing Galdon Clavell, "most engineers have had no training on social issues, so they are being asked to develop algorithms to address social issues that they do not understand. We have created this technological world where engineers are calling all the shots, making all the decisions, without knowing what could go wrong". Both sentences summarize some of the reasons for such cases where bias and discriminatory outcomes were reproduced, thus emphasizing the relevance of auditing algorithms methods to help the detection and mitigation of all these issues, and so, improving the AI algorithms [BL94; Ske20].

2.3.1 How is the bias introduced in the algorithmic systems?

Typically, bias is already present in the model after the choice of datasets, even though developers often do not notice it. Algorithmic systems encode complex societal values through mathematical functions in an attempt to make appropriate predictions based on data from the past [Ske20]. However, this data comes from previous situations, where it could have been subjected to human bias and discrimination. Therefore, using biased datasets will likely turn the model also biased [Agg20; Ske20].

Moreover, bias can also be introduced during the algorithmic decision phase. Developers and researchers occasionally do not understand or are not properly elucidated how the algorithm should decide, thus leaving the system exposed to developers' and researchers' own bias, which is natural considering that all humans have their own prejudices and ideas [BL94; Ske20]. This human factor that is invariably present during the development of the system implies the absence of bias-free systems, thus reinforcing the significance of auditing the algorithmic systems [BL94].

2.3.2 Types of bias

We have already seen how bias is introduced in algorithmic systems and some of the impacts caused by biased models. Although, bias can have different forms and sources, being possible to categorize it in four different ways: **dataset bias**, **association bias**, **automation bias**, and **interaction bias** [CIM].

1. **Dataset bias** — Dataset bias occurs when the data used to train the algorithms does not represent the diversity of subjects expected to be impacted by it. Some generalizations on the datasets used that do not consider the whole variety of users, thus underrepresenting them, may contribute to this type of bias. [Agg20; CIM]. This bias version causes the algorithmic systems to perform well often, but only on a small subset of users [CIM].

2. **Association bias** —Association bias takes place when the data used to train algorithms reinforces and amplifies the existent cultural bias [3]. Language translation tools that create gender assumptions, like for instance, doctors are male and nurses female, are the perfect example of this kind of bias [CIM].

3. **Automation Bias** — Automation bias occurs when the algorithmic system makes decisions based on assumptions that may be against the diversity of users. During the algorithm's design, it is significant to consider the goals of all types of users, not just the average users [CIM].

4. **Interaction Bias** — Interaction bias arises when it exists human manipulation of AI algorithms to create biased results. This type of bias is commonly observed in chatbots, where human users deliberately introduce offensive words into the chatbot, with the sole purpose of training the chatbot to say indecent sentences. One of the most famous examples, was the Tay chatbot developed by Microsoft, which started to post inflammatory content on social media platforms, after just sixteen hours of been released (see Figure 2.5) [Agg20; CIM].

[3] Cultural bias can be defined as "a tendency to interpret a word or action according to culturally derived meaning assigned to it" [HDP19]

Figure 2.5: Inflammatory post by TayChatbot, developed by Microsoft, that after just 16 hours of being released, started to upload controversial posts [Agg20].

2.3.3 Using Algorithmic auditing processes to mitigate unintended Bias and Discrimination

As already pointed out, algorithmic systems are subject to encode unintended bias and discriminations. Technical tools alone cannot prevent these situations since many potential sources of bias lie outside the source code. Therefore, no amount of technical design during the development of the algorithm can ensure that a model will never produce discriminatory outcomes [Kim17]. Removing protected attributes, like gender, race, age, and others, is not a solution since other features highly correlate with those, thus maintaining the model biased [Kim17].

Following this idea, algorithmic auditing surges as a relevant tool to achieve more accountable, transparent, non-discriminatory, and bias-free systems. Auditing processes focus not only on comparing outputs and inputs but also on the foundations of the algorithm, such as goals and datasets used, for instance, fields that technical tools cannot ascertain [Ske20].

2.4 Algorithmic Auditing Barriers

Algorithmic auditing is becoming a fundamental process to guarantee the reliability and fairness of the algorithms. Regardless, many obstacles affect and difficult this process. Concerning the external audit approach, the principal obstruction is the lack of transparency of the algorithms that are being audited. As pointed out, companies and governments prefer to maintain the source code and details of the algorithm confidential, which makes the auditing process more challenging, rather than disclosing this information to be audited [Cla+20; Kim17]. It is relevant to mention that there is a shortage of platforms and guidelines to perform algorithmic auditing feasibly.

Regarding the internal audit approach, many barriers are also often present. Firstly, internal auditors regularly have difficulties remaining independent and objective during the auditing [Raj+20]. Secondly, some problems that the algorithm may present can only be discovered once the system is released [20b; Raj+20].

To conclude, by citing once again Galdon Clavell, "the main barrier at the moment is people don't know that algorithmic auditing exists" , which reflects the final obstacle in the algorithmic auditing field. [Ske20].

2.5 Technologies and Tools available

As addressed in previous sections, one of the main problems of algorithmic auditing is the lack of platforms and frameworks to turn algorithmic auditing into a more manageable and accessible task. By analyzing prior literature, we uncovered different technologies that can support auditing processes, namely: **bias and fairness libraries**, **auditing frameworks**, and **external audit consulting companies**.

2.5.1 Bias and Fairness libraries

Bias and fairness libraries are usually open-source toolkits that allow users to audit datasets used in AI algorithms and ML models for bias and discrimination. These libraries enable the creation of specific reports about the dataset, thus permitting the user to see if there is some degree of prejudice or discrimination in the dataset selected.

These libraries are valuable tools and contribute to the transparency and fairness of the algorithms, given that biased datasets are an influential factor in the existence of biased and discriminative algorithms.

2.5.1.1 Aequitas

Aequitas is an open-source bias audit toolkit that allows developers and researchers to audit the datasets used in AI algorithms and ML models (Figure 2.6). Aequitas allows users to introduce their own datasets and then generate a report with the details about the dataset in question. This report is often called a bias report and specifies if the dataset submitted might be biased or discriminative[Oal+20; ray]. This tool can be used online, through a web application, and also offline, by importing the python libraries or even through the command line [ray].

It is worth noting that the web version of aequitas is the platform that most resembles the system we proposed to create. The main difference between the online version of aequitas and our proposed solution is that the aequitas online tool only allows us to verify the dataset while our web application will focus on auditing the existent algorithms.

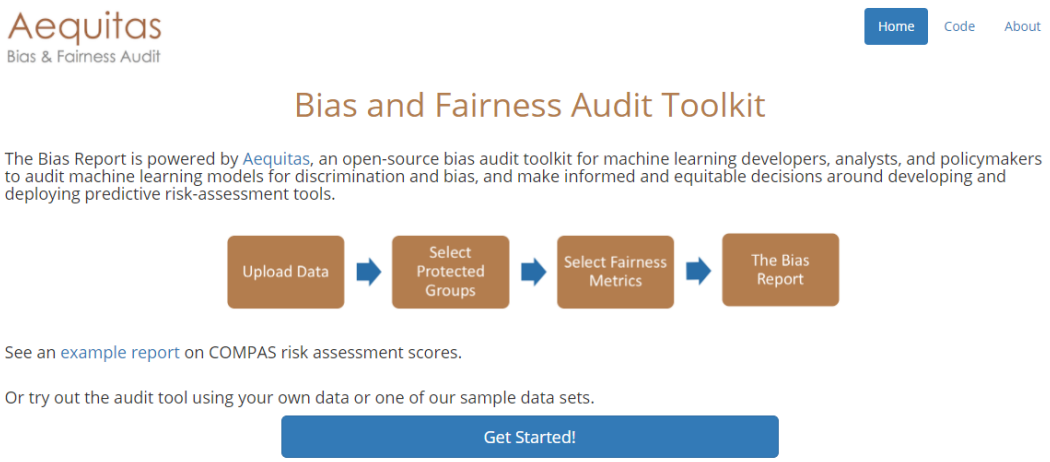

Figure 2.6: Homepage image of the web version of aequitas, here it is possible to observe the different steps required to create a bias report of the users' datasets [ray] .

2.5.1.2 ML-fairness-gym

ML-fairness gym is a group of components that allow users to run simple simulations that explore the potential long-term impacts of ML decision systems in social environments. It implements a framework, to study and search for long-term fairness effects in the simulation scenarios, where a learning agent interacts with the environment over time. Achieving a deeper understanding of such long-term effects is thus a critical step for ML fairness research [21d].

2.5.1.3 AI Fairness 360

The initial release of the AI Fairness 360 Python Package (Figure 2.7), developed by IBM, contained nine different algorithms produced by the algorithmic fairness community, which have the purpose of mitigating unwanted bias [18]. AI360 is different from the other open-source toolkits due to its focus on bias mitigation (not just on the metrics) and industrial usability.

Figure 2.7: Results obtained with the AI Fairness 360 tool using the COMPAS dataset [18].

2.5.2 Auditing frameworks

Auditing frameworks are different kinds of tools. They are more focused on providing a variety of guidelines and practices to auditors. This variety of tools is related to internal auditing processes since these guidelines can be used by internal auditors, researchers, and developers, to audit the algorithms that are being produced or that will be released soon.

2.5.2.1 SMACTR and ICO Internal Framework

The SMACTR framework (Figure 2.8), or more extensively, Scoping, Mapping, Collection, Testing, and Reflection, is a proposed internal audit framework, developed and proposed by Google AI, with the purpose to guide auditors on how they should audit their algorithms. Moreover, the ICO Internal Framework (Figure 2.2) follows the same concepts, having similar purposes, however, it is more focused on government algorithms [20a; 20b; Raj+20].

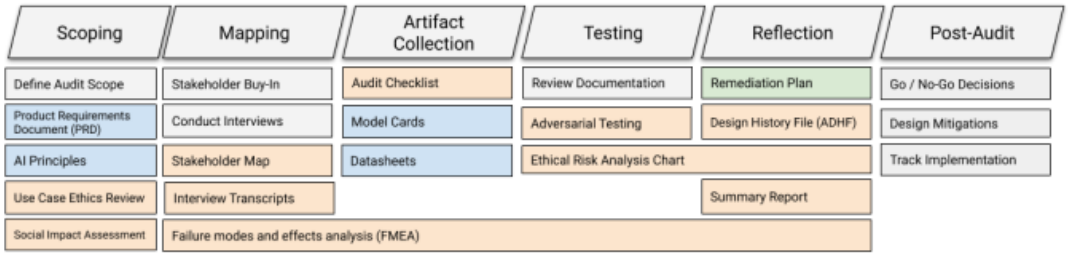

Figure 2.8: Overview of Internal Audit Framework. The sections colored in gray indicates a process, and the colored sections represent documents. Documents in orange are produced by the auditors, blue documents are produced by the engineering and product teams and green outputs are jointly developed [Raj+20].

2.5.2.2 TensorFlow

TensorFlow (Figure 2.9) is one of the most popular ML Frameworks. It is used not only across Google but also by many external developers and researchers around the world, helping them build more inclusive, ethical, and accountable algorithms [20f]. TensorFlow is committed to helping make progress in the responsible development of AI by sharing a collection of resources and tools with the ML community, which can be applied in the different development stages of the algorithm [20e].

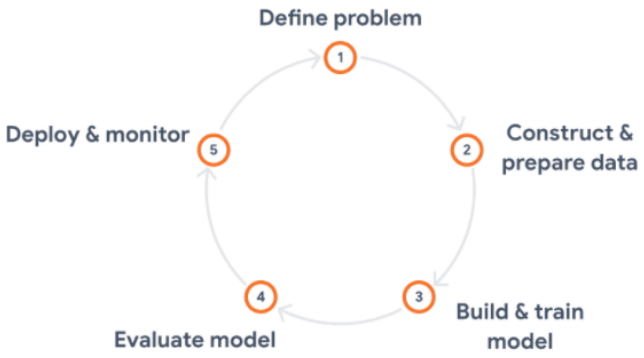

Figure 2.9: The five stages of the TensorFlow framework used to ensure more responsible AI practices [20e] .

2.6 Summary

AI algorithms have already a notable impact in our society, being used in almost every decision-making processes, from government decisions and resource allocation to advertising, media preferences, and even encryption of our data (section 2.1). Nonetheless, as the impact of algorithmic systems arises, we observe that the monitoring of these systems is not adequate, which leads to the existence of AI algorithms that produce discriminatory and biased outcomes (section 2.3).

While governments and corporations understand the relevance of these problems, they frequently do not know how to solve them and prefer to maintain their algorithms specifications (such as datasets, source code, or decision logic, for example) confidential, rather than providing this information to external researchers and auditors for them to search for unintended bias and discriminatory outcomes (section 2.4). Internal auditing tools appear as possible solutions to the public and private sector, as they provide guidelines to audit the algorithms during the development stage (section 2.2). Notwithstanding, external auditing should remain a crucial procedure considering that internal guidelines and technical tools alone cannot mitigate discriminatory outcomes and unintended bias since the source of these problems often lies outside the source code (sections 2.2 - 2.3).

Conclusively, there is a vast set of tools, technologies, and frameworks being developed that aim to help auditors, developers, researchers, politics, and anyone whom these topics might concern (section 2.5). Although, there is still limited knowledge about these tools and the concept of algorithmic auditing in general. Performing an audit on AI algorithms is not simple nor easy, and the tools and technologies available are not sufficient to fulfill the needs of the users, especially when referring to black-box algorithms.

3

Methodology

This chapter exposes the methods followed to approach the problem previously described. It starts with a brief introduction of User-Centered Design (UCD) and Human-Computer Interaction (HCI), and then with a description of the In-depth interviews that took place and respective thematic analysis. Finally, it explains the different stages of web application development.

3.1 Introduction to HCI and UCD

HCI can be defined as "a discipline concerned with the design, evaluation, and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them" [US92]. HCI in a global perspective can be interpreted as an interdisciplinary area, intersecting fields like computer science, sociology, industrial design, or even anthropology. It is crucial to understand how human users interact with technology in general. If the system is hard to interact with then the design must be changed. Otherwise, there is a high probability of the system becoming obsolete since the users will likely stop using the system. Following this idea, HCI has the central purpose of designing and developing systems that improve the interaction between humans and technology [10; US92].

UCD is an approach that aims to build interfaces, products, and services that are appropriate and accessible to the maximum number of users possible, within the constraints of the system itself (see Figure 3.1 [21j]) [Boy17; WD14]. In the context of this thesis, we will use UCD methods to develop a version of the prototype that better fits both auditors' and programmers' needs. For this approach to work, we will need to gather the users' ideas and feedback, which can be accomplished through the execution of **in-depth interviews**.

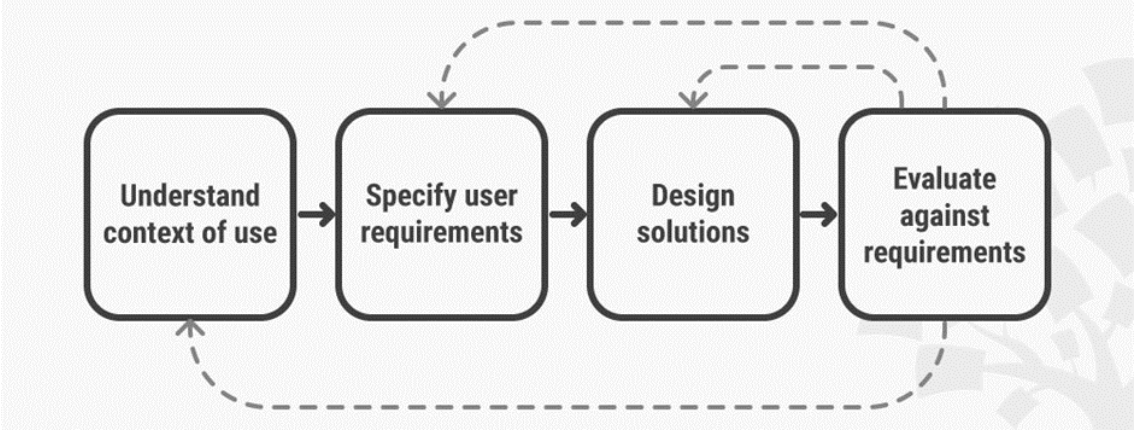

Figure 3.1: UCD overview: First, there is an understanding of the context in which users may use a system. Then, we identify and specify the users' requirements. A design phase follows, in which the design team develops solutions. The team then proceeds to an evaluation phase. In this stage, there is an assessment of the users' evaluation of the system to check how well a design is performing. From here, there are further iterations of these four phases and a continuation until the evaluation results are satisfactory [21j] .

3.2 In-depth Interviews

As pointed above, for the UCD approach to work, it is important to capture user information and feedback since there is a belief that users have something to offer at every stage of the system developing process [WD14]. In-depth interviews appear as one of the principal methods for data collection in qualitative research, allowing interviewers to gather and understand some of the detailed information, experiences, and personal opinions of the interviewees [GDM11; LLL95]. These interviews are used in situations where the interviewer wants to ask open-ended questions, in contrast with surveys, for example, where the answers are more quantitative, thus permitting the interviewer to deeply explore the respondent's feelings, opinions, and perspectives on a particular subject [GDM11].

An in-depth interview is supposed to be structured yet flexible. The interviewer presumably has a script with topics to follow (see appendix B), however, the interviewees should have the freedom to change the order of when these topics are addressed in a way that better suited them. This approach also provides the respondent a more relaxed atmosphere, leaving them more open to discuss some subjects [LLL95].

In-depth interviews, however, also has disadvantages: firstly, these interviews might take several minutes ending up being excessively intensive; second, the interviewer should have experience in questioning the participants to get the most detailed and rich data [GDM11].

In the context of this thesis, during the in-depth interviews with auditors and developers, we aimed to understand all the issues of algorithmic auditing processes, sharing algorithm's executables, amongst others, to perceive the better way to design the system.

We opted to use this method of interviews since we considered that in-depth interviews allowed us to discuss these topics more freely and receiving feedback about the system more interactively and purely. To achieve this goal, we used a qualitative analytic method designated **Thematic Analysis** (section 3.4).

In total, **nine** interviews were conducted: four with developers/data scientists and five with auditors/authors with articles related to the algorithmic auditing field. All the interviews were fully recorded and transcribed, and together, they result in a cumulative amount of **3 hours and 49 minutes** of recordings. More details about the participants can be found below in table 3.1.

Participant	Sex	Country	Occupation	Experience
P1	F	Portugal	Scientist	Algorithm development
P2	F	Spain	IT Analyst and CEO	Internal algorithmic audits
P3	F	USA	Scientist in auditing related fields	Internal algorithmic analysis and testing
P4	M	Portugal	Scientist	Algorithm development
P5	M	Portugal	Scientist	Algorithm development
P6	F	Portugal	Professor and Director	Algorithm development
P7	M	USA	PHD Student	External algorithmic audits
P8	M	Portugal	Director of Research	Algorithm development and AI Ethics studies
P9	M	Austria	PHD Student	Qualitative algorithmic analysis

Table 3.1: Participants details.

3.2.1 In-depth Interviews Protocol

The nine participants were recruited mainly in two different ways: the first was through the analysis of articles related to the algorithmic auditing field, where we sent multiple emails to the various authors of those articles. The second was through existent connections and contacts within Fraunhofer.

All interviews were conducted online via Microsoft Teams and led on my personal computer. During the interviews, we followed the scripts present in appendix B. Nonetheless, the structure of the interviews was flexible, and so, some questions were added or changed according to the responses provided by the interviewees.

Moreover, during the interviews, a prototype of the system was shown to the interviewees. The prototype was exhibited on my computer, and interviewees did not have any of its control.

3.3 Initial Prototype

During the in-depth interviews, it is usual for the interviewees to respond abstractly since they are not presented with an illustrative and concrete example. For that reason, we decided to build the design of the initial prototype using the Balsamiq Wireframes software [21a]. The purpose of this chapter section is to explain more thoroughly what was shown to the interviewees, thus allowing the reader to get a superior understanding of their feedback and opinions. Here, we provide the initial design of the system's main functionalities, including some explanation about the interaction between activities.

Once more, it is significant to notice that we understood that this version of the prototype needed to be modified. However, it was of extreme value to present a version of the prototype during the interviews in order to better channelize the goals and specifications of the system.

Below, we can observe the main screens of the initial prototype used to explain the usage of the system to the interviewees. A full illustration of the prototype can be found in annex I .

3.3.1 Home screen

The home screen is the users' first glance at the system. In this screen, it was provided some context regarding the significance of auditing algorithms, as well as the prominence of uploading the algorithms to be audited. It was also illustrated how the users could access the statistics of the algorithms uploaded previously and the auditing requests created.

3.3.2 Upload the algorithm screen

In this screen, we illustrated to interviewees the requirements needed for the users to upload their algorithms (which can then be audited). We provided interviewees with a brief description of each requirement and its respective importance. We also described how the user could upload their algorithms' executables (by introducing in the platform a .exe file). It was explained how the input and output specifications could be defined, including an image of an input example, highlighting the multiple available formats for both input and output (Figure 3.2).

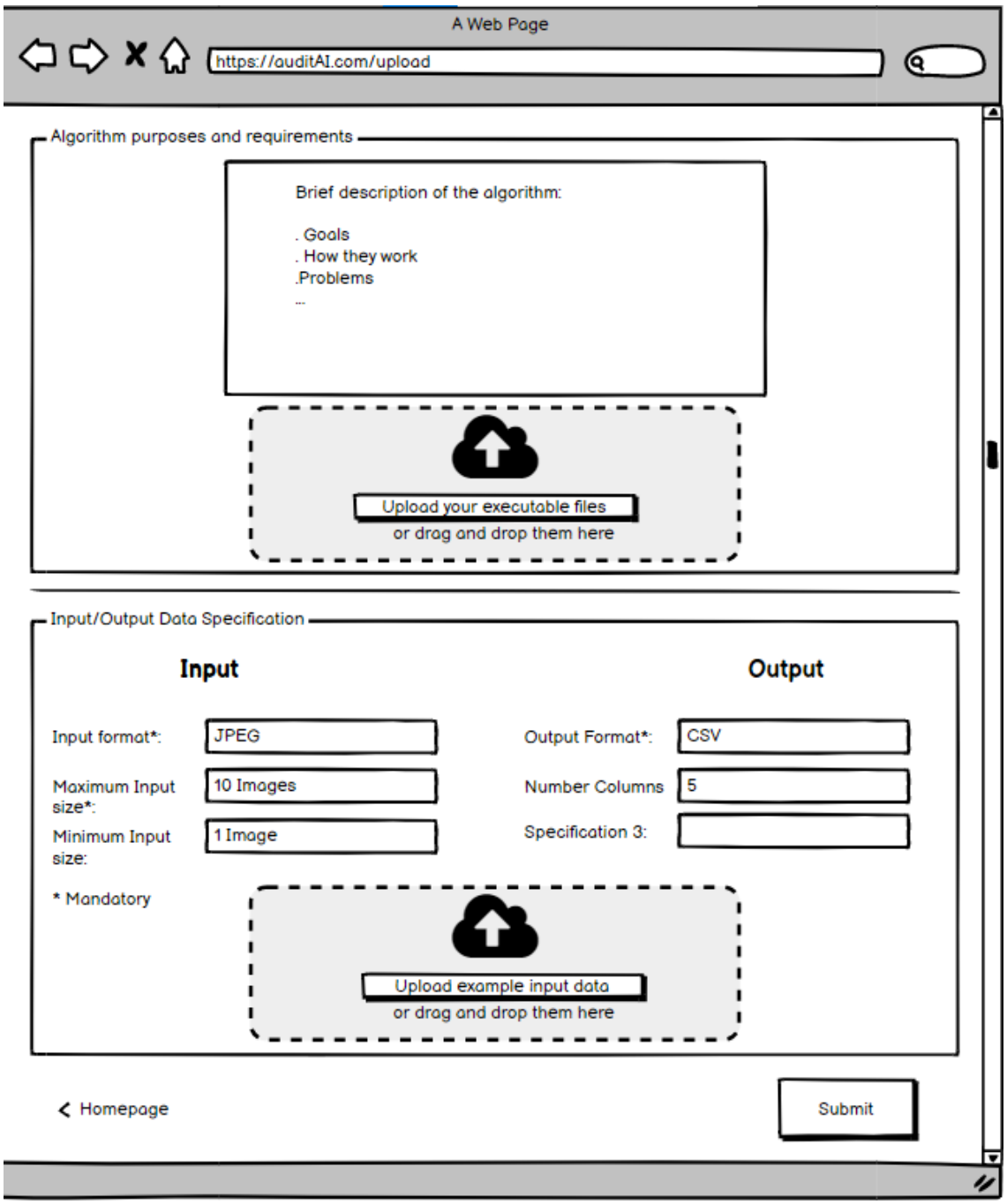

Figure 3.2: Screen of the algorithm upload page. Here it is illustrated the different fields for both the algorithm and input/output specification.

3.3.3 Algorithms list screen

In this screen of the prototype, we explained how users could visualize all the available algorithms to audit. These algorithms can be filtered by category and name and ordered by pages (Figure 3.3). We also demonstrated to interviewees how users could select the option to "view more +", which will redirect the screen to the visualization of all the details of the correspondent algorithm, including the input specification.

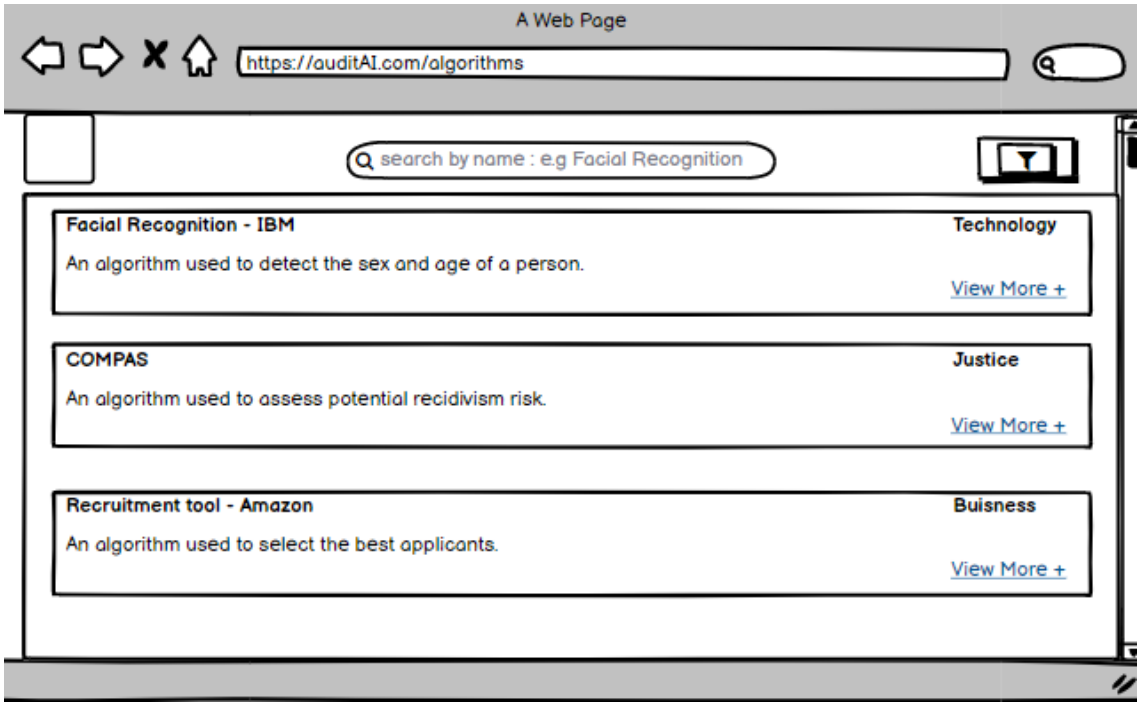

Figure 3.3: Screen of the algorithms list page. Here it is illustrated the different algorithms available to be audited.

3.3.4 Auditing request screen

In this screen, it was explained how the users could create an algorithm audit request. That is to say, the users can upload their private datasets for the algorithm to evaluate them and then return the response to the user. For this to happen, the user must click on the "Audit" button on the algorithm component (Figure 3.4), and then upload his datasets by dropping/selecting them.

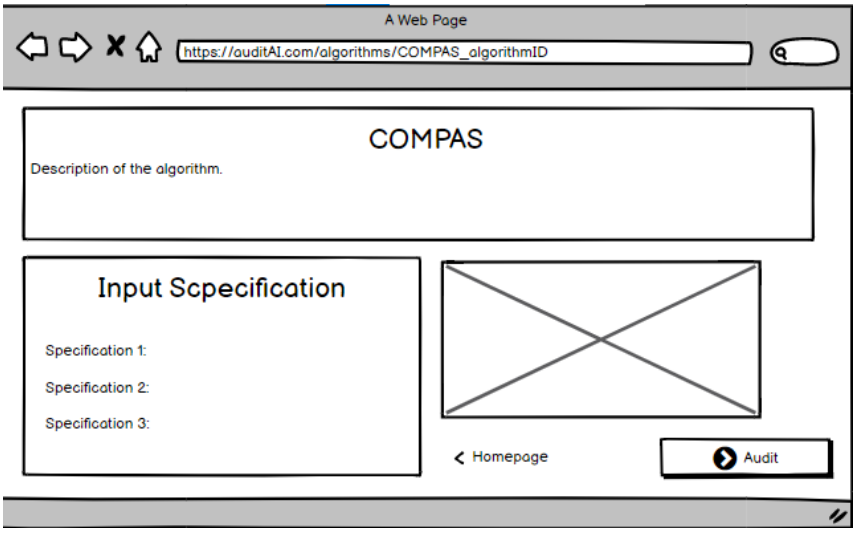

Figure 3.4: Screen of the algorithms details page. Here it is illustrated all the details of the algorithm selected. The user can also choose to audit this algorithm by selecting the button "Audit".

3.3.5 Result screen

In this final screen, we illustrated how the users could download the results of their auditing requests. When the result of an audit request is available, the user receives a notification that the result has arrived, which he can check in his list of audit requests. The user then can visualize in a more detailed way some of the specifics regarding the evaluation of the audit request and also download the result, which can be available in different formats (Figure 3.5).

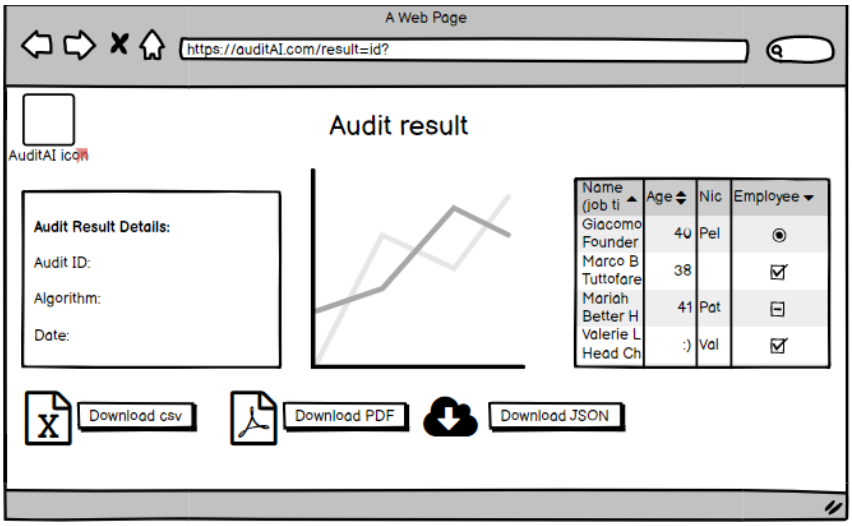

Figure 3.5: Screen of the audit request result. Here it is illustrated how the user can download the result of his auditing request. It is also possible to visualize some details regarding the evaluation of the auditing request.

3.4 Thematic Analysis

As a result of in-depth interviews, since every participant had their own ideas and opinions, each interview was different from one another. Hence the data gathered was not fully structured, so its analysis could not be performed using only objective methods. The thematic analysis method surged as a solution for this problem, allowing the analysis of the interviews' transcriptions.

According to the original authors, thematic analysis can be defined as a "method that helps a researcher to identify themes and patterns of meanings across a dataset in relation to a particular research question(s)" [Dam19]. This method allows researchers to discover significant patterns and relationships within the data, thus enabling the researchers to answer some questions of the study [BC06; Dam19]. The thematic analysis method comes with multiple advantages: it is easy and quick to learn, flexible, and does not require previous experience in qualitative research, which makes thematic analysis an attractive method to use for qualitative data analysis [BC06].

According to [BC06], the thematic analysis method involves six phases:

- **Phase 1 - Become familiar with data :** In this stage, the researcher starts transcribing the data collected, followed by its "repeated reading". This step is crucial to fully understand the data gathered, which will allow the discovery of patterns and meanings. During this stage, the researcher usually starts noting down some initial ideas [BC06].

- **Phase 2 - Generating initial codes :** In this next stage, the researcher identifies all the relevant portions of data within the dataset, designated as codes. These codes reflect the patterns and features found by the researcher, which will be further analyzed. It is recommended to code for as many potential themes/patterns as possible. [BC06; Dam19]

- **Phase 3 - Searching for themes :** At this stage, the researcher starts to break down the initial codes and evaluates how these codes can be congregated into potential themes. It is helpful to use some form of visual representation to assist the sorting of codes and themes, in our case we elaborated a thematic map. This is also the time when the researcher starts to think "about the relationship between codes, between themes, and between different levels of themes" [BC06].

- **Phase 4 - Reviewing themes :** At this stage, a refinement of the potential themes created in the previous stage needs to be done by the researcher. It will become clear that some potential themes are not themes, and others might collapse into each other. On the other hand, some themes might need to be converted into separate themes.

  This stage involves two levels of reviewing and refining potential themes: at the first level, the researcher needs to review all the coded data extracts and analyze

whether they "appear to form a coherent pattern". Level two involves a similar process, but in relation to the entire data set. The researcher evaluates the validity of the individual themes regarding the whole data set. At the end of this stage, the researcher must fully understand how the themes interact together and how they express the overall narrative present in the data [BC06; Dam19].

- **Phase 5 - Defining and naming themes :** At this stage, the researcher should refine the themes and analyze the data within them. It is crucial "not to try and get a theme to do too much, or to be too diverse and complex". The researcher should understand the narrative of each theme and how they fit within the data. [BC06]

- **Phase 6 - Producing the report :** At this final stage, the researcher should be able to demonstrate "the complicated story of your data in a way which convinces the reader". The ultimate analysis is supposed to be consistent and logical as well as clear and compact [BC06].

3.5 Iterative development of the web application for algorithmic auditing

This chapter briefly explains the different development stages of the dissertation.

3.5.1 Initial script and prototype

After completing the literature review, the next step was to develop the script that was used later in the in-depth interviews with the auditors and researchers. It is significant to notice that an initial prototype was built before the beginning of the interviews. The prototype was constructed using only the requirements extracted from the literature review and its analysis. It was developed using the Balsamiq Wireframes software [21a].

3.5.2 System development

After the conclusion of the in-depth interviews and its analysis (sections 3.2 and 3.4), we started the development of the back-end of the web application. As planned previously, the web application includes a security system and also multiple other functionalities, including a few that implement a certain degree of abstraction (more information about the functionalities of the system can be found later in section 5.3). In the final stages of the development of the backend, we started to build the frontend of the web application. We focused on creating an appealing interface that simultaneously allowed the users to interact efficiently with the platform. Finally, we recorded a video where we fully demonstrated and explained how the web application works. This video was sent to the interviewees in order for us to receive some feedback about the system. For recording the video we used the loom tool [21e].

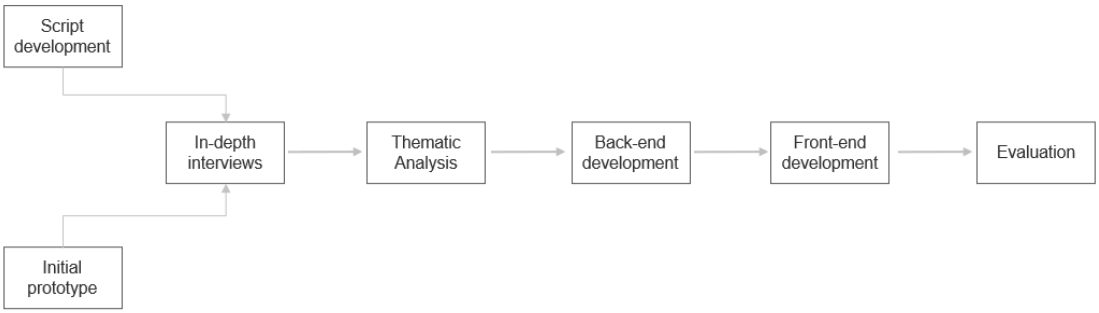

Figure 3.6: Illustration of the multiple web application' development stages.

3.6 Summary

In this chapter, we explained the methodology followed during the elaboration of this thesis. We pointed out the techniques and methods used during the development stages of the web application, illustrated above in figure 3.6.

4

In-depth Interviews Analysis

This chapter describes the results from the in-depth interviews, obtained through thematic analysis. Finally, it portrays the impact these results had on the design of the proposed solution.

4.1 Findings

As stated previously in section 3.2, the purpose of the interviews and respective thematic analysis was to understand the difficulties related to the algorithmic auditing processes, datasets, share of executable files, amongst others. It is significant to notice that we divided the interviewees into three different categories: developers of algorithms in research/industry context, algorithm auditors, and researchers of algorithm auditing. It is meaningful to mention this division for a better understanding of the responses of the different interviewees.

In order to better illustrate the challenges of the algorithmic auditing field and the prototype itself, we focused on the four themes generated from the in-depth analysis: the impact of datasets in the algorithms, societal and ethical questions associated with algorithms, difficulties in the algorithmic auditing field, and prototype design suggestions and challenges.

4.1.1 Impact of datasets in the algorithms

One of the main topics that most interviewees, in particular developers, mentioned during the interviews was the importance that datasets used in the training phase have on the global performance of the algorithms. As referenced above in subsection 2.3.1, among the principal causes of algorithmic bias is the use of biased datasets during the training phase. This problem was promptly noted by participants P1, P5, and P6, who consider that the "raw data can not be reliable". Hence, the lack of instances in the training dataset can lead to the bias of the algorithms, as quoted below by P5. P6 believes inclusively that the "fundamental causes of discriminatory outcomes created by the algorithms are due to

the use of biased datasets", reinforcing the importance that datasets have on the general functioning of the system.

> "We do not have many types of dark photo-types [of skin moles] , which end up biasing the algorithm." (P5-developer)

We have been given different responses regarding the prior validation of the datasets before their use in the training phase. Participants P1, P4, and P5 confessed that they were careful when using the datasets in the training phase, applying a layer of verification to the dataset before using it. However, as quoted below, P6 reported to us that there are still developers that do not practice any type of verification to the datasets used in the training phase, despite all of the existing knowledge on why that is important.

> "To my best knowledge that preoccupation (verifying the dataset) does not exist. I have the expertise to ensure that the datasets are not biased, and I recognize that there are people who have that awareness, however, it does not mean that most of them practiced it ." (P6-developer)

Finally, when it comes to the tools available that allow the verification of the datasets, it was pointed out to us, by P4 and P5, that these tools were scarce. They also briefed us that almost all of the work done concerning the validation of the datasets was manual, as observed by the quote below by P5. The manual work implies that engineers/developers have to evaluate all the datasets used for bias and discrimination without the use of appropriate tools, which end up making this process more time-consuming and more subject to errors.

> "It is all manual work, there are not many tools that can help us with this task (verifying datasets). " (P5-developer)

4.1.2 Societal and Ethical questions associated with algorithms

Another theme approached by the interviewees was related to the complex ethical and societal issues that the algorithms might encode, which are often not discussed but can have a significant impact on the outcomes of the algorithms, as previously described in section 2.3. This topic was broadly discussed during the in-depth interviews, where most of the interviewees agreed that these questions are of extreme importance. Participants P1, P4, and P5 confessed taking into consideration these topics when developing the algorithms, either by analyzing the real impacts the algorithms have on particular populations (as mentioned in the quote below by P4) or just by extrapolating how the algorithms being developed can affect the users (as referred in the quote by P5).

> "As we started interacting with the technology, we began to realize the impact that it is having on the users. " (P4-developer)

> "Sometimes we have to think outside the data and deliberate how the system can affect its users. " (P5-developer)

Nevertheless, and despite the increasing awareness on these matters, which is still "clearly not enough" as P2 stated, engineers and developers continue to lack the knowledge or the training to eradicate the ethical and societal risks involved when developing an algorithm, as mentioned below in the quote by P2, and so, just the awareness by itself is not enough. This means that despite the increasing conscienceness of why it is significant to audit algorithms or allow the algorithms to be audited by entities and developers, this awareness is still very ineffective if not put into practice.

> "Engineers are not trained to mitigate societal risks, so awareness is not that useful in practice. " (P2-auditor/researcher)

Nonetheless, as defended (and quoted below) by P7, the increasing awareness of the societal and ethical questions during the algorithm development allows engineers and developers to think ahead in these questions before starting the development of the algorithms, instead of just rectifying them after being deployed.

> "I still think that developers being aware of these issues is also important, so you are not just fixing issues after the models have been deployed, but instead, thinking about these questions from the beginning. " (P7-auditor)

To conclude this theme, it is significant to notice that despite the present regulations linked with these matters, as the GDPR for instance, there is still a lack of compliance with those regulations, as claimed and quoted below by P2. Thus, this lack of conformity with the existent regulations has a negative effect on the improvement of the algorithmic systems since they do not verify if the correct procedures, which were created to reduce the cases of bias and discrimination, are applied.

> "There is regulation, there is no enforcement of the regulation, but it exists... so it is not a legal issue, it is a compliance issue. " (P2-auditor/researcher)

4.1.3 Difficulties in the algorithmic auditing field

The goal of this dissertation was to create a tool for algorithmic auditing which could facilitate the tasks of both auditors and engineers. Therefore, in this theme, we discussed the importance of auditing algorithms and some of the main challenges and issues present in the algorithmic auditing field, thus allowing us to understand the problems that needed to be solved and how our prototype can help mitigate these problems.

As mentioned previously in section 2.3, we have seen an increase in the use of algorithmic systems in recent years. These systems have a substantial impact on society, however, when it comes to ascertaining if these systems are performing correctly and

according to regulations, we observed a lack of enforcement and compliance with those guidelines. Hence, algorithmic systems left unchecked can easily amplify the bias and discriminatory outcomes that already exist, which can occur, among other reasons, due to the bad design of the algorithms (as quoted by P2) or even engineers' errors, as claimed (and quoted below) by P3.

> "So the way that algorithms have being developed have been so messy... they just did some statistics and math. Some of them have so many problems that we often have to rebuild them. " (P2-auditor/researcher)

> "We find that new developers that are new to C++ still make some rookie mistakes ... during the development of the algorithms . " (P3-researcher)

Algorithmic auditing surges as an essential method to fight the existing inequality and bias present in algorithmic systems. Nonetheless, algorithmic auditing faces multiple obstacles which constitute a barrier to the improvement of the algorithmic systems. More information about this topic can be found in section 2.4.

One of the main obstacles referred to (and quoted below) by P7 during the interviews was the monetary and temporary cost of performing an external audit. During an external audit, it is necessary to interact with the algorithm without the entity that owns the algorithm knowing that the algorithm is part of an auditing process, meaning the scientists have to repeat the experiments multiple times in order to gather the results. This translates into high monetary costs since these experiments charge a certain amount of money by iteration. They are also temporarily costly since many of these interactions are done manually, and it may be needed to repeat them inclusively.

> "The cost of auditing was also high... we ended up spending close to 10K USD to run the experience [referring to an audit of a social network advertisement algorithm], and it is not just in terms of money but time as well ." (P7-auditor)

Another barrier discussed with the interviewees was the cultural differences that are often present during the algorithmic audits. There are different regulations related to data privacy or algorithm transparency according to the region of the globe where the algorithms are applied. Moreover, the definition of some concepts like bias or discrimination, for instance, may differ from region to region (as mentioned and quoted by P2). Thus, it is difficult for auditors to evaluate the fairness of the algorithms since the concept of bias can vary depending on the culture they are inserted.

> "I think that it need to be seen a spark of a longer or broader process of assessing what is fair and what is ethical, there is a massive cultural difference, for instance the dynamic of discrimination in the data are different in the US and in Europe. " (P2-auditor/researcher)

Finally, the last obstacle is related to the initial prototype, where it would be possible to upload the algorithms executables. In this particular topic, P2 and P4 pointed out that there is still a lot of reluctance concerning the share of the algorithm's data, even in the form of executable files, mainly due to intellectual property matters, as mentioned in the quote by P2. So, there is very little encouragement and motivation to share the algorithms to be audited. In addition, P4 defended that the motivation for allowing the algorithms to be audited is dependent on the clients (as stated in the quote below by P4) and their need to provide some layer of verification to the algorithms.

> "I think that with algorithms there is a huge issue with intellectual property, so I think that there is very little encouragement and motivation for sharing your algorithms. " (P2-auditor/researcher)

> "(the willingness to let the algorithms be audited) ... depends a lot on the client and their need to do some kind of job in that sense. " (P4-developer)

4.1.4 Prototype design suggestions and challenges

In this final theme, we discuss with the interviewees some of the main obstacles the web application could face, as well as some suggestions and feedback on how to improve the overall design and functionalities of the system.

4.1.4.1 Prototype challenges

One of the challenges provided to us was the possible **abusive use of the system**. P5 pointed out (as quoted below) that one of the obstacles of the prototype may be the excessive use of the system by some users, hence allowing these users to benefit from the system in a way that is beyond the goal of algorithmic auditing.

> "One of my concerns is to know how you will prevent the abusive use of an algorithm from part of certain entities. " (P5-developer)

Additionally, the interviewees referred that the **impossibility to upload the dataset used in the training phase** might be an obstacle. P6 stated, as quoted below, that this impossibility might be a limitation to the system since there might be an overlap of the data used in the training phase with the data used to audit the algorithms.

> "One of the obstacles I noticed is that the upload of the dataset used in the training phase is not possible, which can be a huge limitation. " (P6-developer)

Moreover, another possible barrier in the development of the web application was the **use of "fake" versions of algorithms .** P7 mentioned that one of the challenges related to the prototype development is the mission to ensure that the algorithms provided by

the companies are the ones that are actually being used in production. This might be a problem since auditors would audit these versions of algorithms, which are different from the ones used in production, thus misleading the auditors regarding the overall results provided by the systems.

> "We are auditing their (entities that own the algorithms) algorithms because we don't trust them to be fair, and to ensure that they are doing the best for society's interest, but my question is: do we trust them enough to upload the executable file of the right source code that they actually use in production? " (P7-auditor)

Finally, the last challenge indicated by the interviewees was the fact that **the system cannot be considered an external auditing tool.** This issue was reported by P9, as observed in the quote below, and it is related to the definition of external algorithmic auditing. P9 noted that technically the prototype can not be considered a tool for external auditing since the concept of external auditing implies the total discretion of the auditing, so companies should not be aware that their algorithms are being audited, which is not the case in our platform.

> "This prototype sure helps a lot, but is not totally a tool for external auditing, this is because external audits should be performed without companies/governments knowing that they are being audited, which is not the case in this, however, it is also not an Internal audit, is somewhat in between. " (P9-researcher)

4.1.4.2 Prototype suggestions

The first suggestions provided by the interviewees were related to the **input and output issues**. Regarding these questions, one of the most concordant suggestions we have received from the interviewees was the restriction of possible formats for both inputs and outputs. Participants P1, P4, P5, and P6 suggested focusing on three/four forms of inputs/outputs in order to implement the prototype more efficiently, as mentioned in the quote below by P4.

> "I recommend you reduce the spectrum of input and output formats, hence focusing on a specific type. " (P4-developer)

Following this discussion, P1 proposed in the quote below that we normalize the input received by the algorithms, so in sum, auditors will pre-process the data, and so all the algorithms would receive the same input format. This alteration would simplify the system regarding the inputs received. Nonetheless, it would require auditors to have some knowledge in processing the data, which could be a limiting factor.

> "(The input format)... could be like a CSV file used by all the algorithms. If the input were the features, then it would basically be the same for everyone, however, it would require developers and auditors to make some adjustments and data processing to arrive at that structure of features CSV. " (P1-developer)

Still regarding this topic, and contrarily to the advice provided by P1, both interviewees P4 and P5 pointed out (as mentioned in the quote below by P4) that it is better to leave the pre-processing of the data to the algorithm and developers, for the reason that auditors can make some mistakes during the pre-processing of the data, thus misleading the developers.

It is significant to notice that these were the first notable opposite suggestions, which we already expect to encounter during the in-depth interviews. Despite being different, both feedbacks are valid, each having its pros and cons, but overall it was a very engaging topic to discuss with the interviewees.

> "It is preferable to be the developer and the algorithm to have control of these issues (pre-data processing) since I could think that the algorithm has some problem when in reality, it could be the pre-data processing that is being badly done." (P4-developer)

The recommendation we received next was to **specify who is going to use the system and how.** P3 (as quoted below) and P7 pointed out that one of the flaws they encountered in the prototype was the lack of description of how the system would work and by whom. Thus, they suggested explaining in a more detailed way how the system would be integrated, who is going to use it, and how they will use it.

> "I think that you could specify in a more detailed way how the algorithms will be used and by whom for example." (P3-researcher)

Another suggestion made by participants P1, P4, and P5 (as quoted below) was for us to supply some **statistics about the results and datasets**, if possible. According to the interviewees, this measure would provide the auditors a better context of the results, thus improving the explainability of the global evaluation.

> "When presenting the results, I think it would be interesting to provide some histograms and statistics about the output in order to help the auditors take their conclusions. " (P5-developer)

To conclude this topic, we end up with the most significant recommendation provided by the interviewees, **to use an API instead of algorithms' executable files**. This suggestion was supplied to us by P8 (as quoted below) during the in-depth interview, where he defended that for companies, and inclusively for us, it would be better to interact with the algorithms through an API. P8 pointed out that most companies would feel safer to let

the algorithms be audited through an API rather than having to share the executable files of the algorithm since it would allow them to have more control over their intellectual property, considering that they would not need to share their intellectual property with outside organizations anymore.

Conclusively, the web application turned into a third-party application that allows communication between auditors and developers/companies, hence contributing to the improvement of the algorithmic auditing experience.

> "I think that it would be better for you to use a type of API to interact with the algorithms instead of asking for the upload of executable files since most of the companies will feel more secure and more willing to let the algorithms be audited when they do not have the need to share their intellectual property." (P8- developer/researcher)

4.2 Implications on the solution design

In this section, it is described all the adopted suggestions made by the interviewees (4.1.4.2) and explained how these recommendations influenced the final design of the prototype. In the end, it would be possible to understand the significant impact these suggestions had on the overall improvement of the system.

4.2.1 Improved the description of the system

One of the suggestions we adopted was the improvement in the specification of the system on the homepage. In the initial prototype, we had very little information about how the system would work and who was going to use it. After hearing the suggestion by P3 and P7, we added some useful data that helped the users to understand the importance of the system and how easily they could use it. We also highlighted how significant the external auditing practices are in the fight for fairness and to reduce the number of discriminatory outcomes present in the algorithmic systems.

4.2.2 Restricted the input/output formats

Another suggestion taken into consideration was the restriction of formats of both inputs and outputs. One of the main difficulties that we encountered during the designing phase of the system was how to include all the different input and output formats that exist and if that would be feasible. To face this challenge, most interviewees advised us to focus on the most used formats of both input and output. This suggestion will allow us to implement the system in a way that could work more efficiently and also be demonstrated more clearly.

Our system supports four formats of input: CSV, IMAGE, JSON, TXT, and also four formats of output: CSV, PDF, JSON, XLSX. The formats available for both inputs and

outputs were selected according to the advice provided by interviewees and also through the reading of statistics regarding the most used file formats in algorithmic development [20c].

4.2.3 Transmitted the unprocessed datasets

An additional proposal provided by the interviewees was to not pre-process the datasets used in the auditing. As stated above in subsubsection 4.1.4.2, we received different opinions regarding the processing of the datasets used in the auditing practices. Nonetheless, both positions were extremely valid, each having its pros and cons.

We choose not to require auditors to pre-process their datasets since it would imply auditors to have a certain degree of knowledge in the field of data pre-processing, which would limit the global volume of users that would be able to use the system. Even for engineers and developers, demanding users to pre-process the datasets would also mean that the algorithms will most likely need to be modified in order to accept the normalized input, which requires further changes to the algorithmic systems. Furthermore, this requirement would create some uncertainty in developers and engineers since they will no longer have control of this stage, and so, in case of a system failure, they will not know if the root causes of the incident are intrinsic with the algorithm or if it was a mistake made by the auditors during the pre-processing of the datasets.

In sum, our system allows auditors to upload their datasets in the formats available (CSV, IMAGE, JSON, TXT) without having the need to perform any additional task, thus leaving the pre-processing of the datasets to the algorithm.

4.2.4 Used APIs instead of executable files

The final and probably most significant recommendation we have received was to audit the algorithms using an API rather than executable files. This proposal was provided to us by P8, who pointed out that companies and other entities will feel safer and more in control of their algorithms (intellectual property), thus making them more receptive to allow their algorithms to be audited.

We have adopted this recommendation since we agreed in full with the interviewee, and so, we changed the way of uploading the algorithm to be audited. We removed the field to upload the executable file, and instead, we added a section for the developer or engineer to note the endpoint to audit the algorithm.

In summary, and despite this change being relatively small, we consider that it had a significant impact on the system since it now allows more entities to share their algorithms to be audited. This receptivity increases not only due to security questions involved but also with the fact that the system has become simpler.

4.3 Discussion and Conclusions

In this chapter, we have presented diverse questions and topics retrieved from the analysis of the in-depth interviews, from issues related to the algorithmic auditing fields to suggestions of the design of the prototype.

It has become clear to us that despite the slight increase in awareness of the algorithmic auditing field, there is still a long way to go when it comes to achieve full non-discriminatory and fair algorithmic systems. It was highlighted that there is still a lot of algorithms that are being badly constructed, either by the use of biased datasets in the training phase or just by the lack of formation of engineers involved. All these factors contribute a lot to the discriminatory outcomes that are often present in our society. Nonetheless, there is still a high level of reluctance when it comes to allowing the algorithmic systems to be audited, primarily due to the fear of share valuable intellectual property.

Regarding the prototype, we acquired numerous suggestions and challenges that we should address during the development stages of the system. When it comes to the issues of the prototype, many were related with some ethical questions that are often present in this form of tools. Furthermore, the interviewees also pointed out multiple challenges concerning the processing of the datasets and the numerous formats of inputs/outputs available. The final recommendation was related to the issues of sharing executable files. Consequently, interviewees suggested we use APIs rather than executable files to increase the trustworthiness of our system since entities will have full control over their intellectual property.

The thematic map retrieved through the analysis of in-depth interviews, and where it is possible to find the principal findings of each theme, is located in figure 4.1.

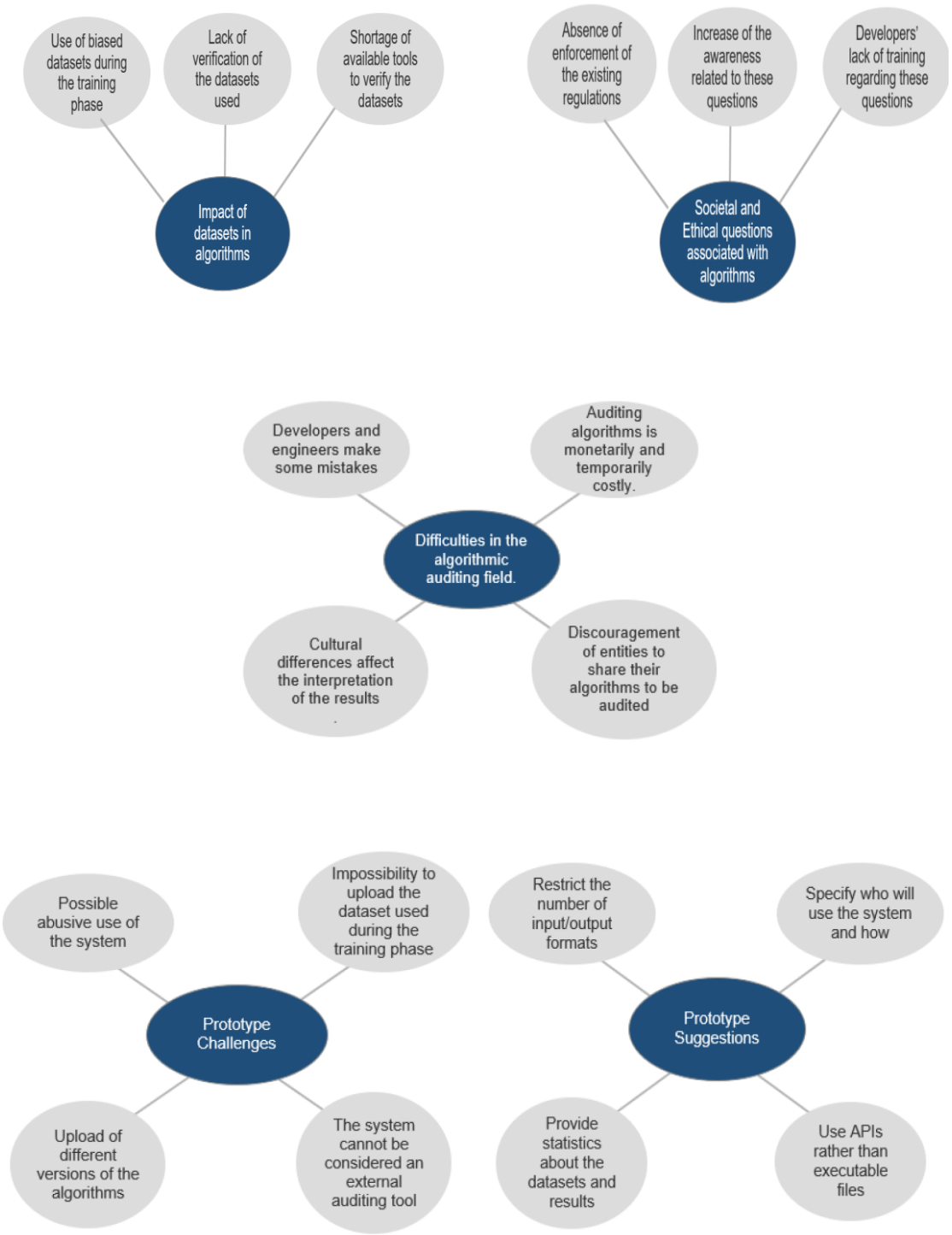

Figure 4.1: Thematic map with the principal findings of the in-depth interviews.

5

Solution Proposal: A Web application for external algorithm auditing

This chapter details the design and functionalities of the developed system. This system was deeply inspired by the feedback received from the interviewees during the in-depth interviews.

5.1 System Requirements

In this section, it is possible to visualize all the system requirements implemented by the system, which were defined based on the literature review and from the responses provided by the interviewees during the in-depth interviews.

- **Enable developers to upload the algorithms to be audited:** The web application must allow the users (developers) to upload the details of their algorithms, including the endpoint and the input/output specification, and then permitting the algorithms to be audited by the other users (auditors).
- **Be scalable:** The web application should accept the existent input and output formats and be able to handle new ones. The system should also be prepared to receive a high load of auditing requests and algorithms introduced.
- **Enable auditors to audit the algorithms with their own datasets:** The users (auditors) should be able to visualize all the algorithms available to be audited and choose any algorithm to audit using their own datasets.
- **Support asynchronous evaluation of the auditing requests:** Regarding the evaluation of the auditing requests, the system should allow async communication with the algorithms' APIs with the purpose of not "blocking" the system while waiting for the evaluation to be completed, thus allowing the system to be more efficient.
- **Provide the results of the auditing requests:** The web application should notify the users when the results of their auditing requests arrive. Then, the users must be

able to download the results in the initially specified format (during the algorithm upload).

- **Provide personal records of auditing requests and algorithms uploaded:** The web application should implement a personal area containing the statistics of the algorithms uploaded by the users and another personal area with the records of the auditing requests.
- **Be secure:** The web application must be secure for both developers and auditors. The system should implement mechanisms that hide the sensitive details of the algorithm, like the algorithm endpoint, for instance. The system should also implement processes that ensure that the auditing requests and corresponding results are only visible to the users that created them.

5.2 Proposed architecture

The Audit AI web application is composed by two core modules, the front-end, and the back-end. Additionally, we also implemented an external module: an algorithm API that interacts with a functioning algorithm. Further down, it is explained the specifics of each module alongside the technologies used to implement them. The complete visualization of the modules used and the technologies associated with them can be found below in Figure 5.1.

5.2.1 Front-end

This component interacts with the back-end of the web application, either through the transmission of the users' data or just by showing the information received to the user. Thus, the front-end allows the user to interact with the complete system, and so, it was significant to ensure that the module was highly responsive, simple to use, and with a user-friendly design. For the development of the front-end, we used React JS plus a CSS framework (UIkit) [21c; 21i]. We chose to use React due to the existent familiarity we had with the tool and also because it is a tool easy to learn and use, among all the other advantages like the ability to reuse some components, for instance [21f]. The UIkit framework allowed us to improve the aspect of some elements, thus making the overall design of the web application much more appealing.

5.2.2 Back-end

The back-end of the web application interacts with the front-end of the web application and with all the algorithms' endpoints. This component is responsible for providing the requested information to the users and for processing all the data sent by the users. For the development of the back-end, we used (Spring + Kotlin), the H2 database (local database), JSON Web Token (JWT) (security tokens), and Azure containers.

We have implemented some mechanisms to ensure that this component would be safe, efficient, and scalable. Firstly, regarding security questions, we used JWT for authenticating the users [aut]. With these tokens, we were able to verify who was the user that was making the requests alongside their permissions, and so, it was possible to provide the users their personal information (their auditing requests and the statistics of their algorithms). Moreover, we used Azure containers to store the files introduced by the users. We needed to store the datasets provided by each user, and so we chose to save them in Azure containers since these containers provided us the safety and scalability we required (they allowed the storage of a large number of files safely) [dle]. It is important to notice that given the context of the system (a web application prototype), we opted to store all the data, except the files introduced by the users, in the H2 database [bae19].

To finish the security theme it is significant to mention that we store the algorithms endpoints internally, and we do not provide them to the users, which is an additional safety measure we took to protect the interests of the entities that upload the algorithms.

Regarding the communication with the algorithms' APIs, we implemented an async communication with these APIs since we needed to prevent the back-end from "blocking". In other words, the back-end of our web application could not wait for the evaluation of the algorithms to finish considering that this would cause the system to stay "locked" until the evaluation has been completed. Thus, an async communication with the algorithms' APIs makes the overall application a lot more efficient since the system would not have to wait for the results to arrive.

Finally, it is significant to mention that we chose to develop the back-end of the web application with Spring+Kotlin due to the familiarity we had with the technology and the advantages it includes [21g].

5.2.3 Algorithm API

To implement the algorithm API, we also used Spring+Kotlin [21h]. While this API does not make part of the core modules of the web application, it was of great significance to implement it, to completely test and interact with the platform. This algorithm API is isolated from the core modules and interacts with both the functioning algorithmic system, in this case, the Microsoft Computer Vision algorithm, and with the back-end of the web application [Pat].

As mentioned above in subsection 5.2.2, the communication between the back-end and the algorithm API must be asynchronous since the back-end cannot wait for the evaluation to be completed considering that this would cause the back-end to stay "blocked" while waiting for the results, which could make the system extremely inefficient. To solve this obstacle, we implemented the algorithm API using Kotlin suspending functions which allowed the async communication with the web application back-end [21b]. These functions were one of the main arguments for the use of Kotlin+Spring in the implementation of the algorithm API, alongside all the other reasons already mentioned

above.

Finally, it is significant to mention that the results received by the algorithmic system are uploaded into an Azure container due to the scalability and safety reasons we have already noted [dle]. The algorithm API then transmits to the back-end of the web application the location of the results.

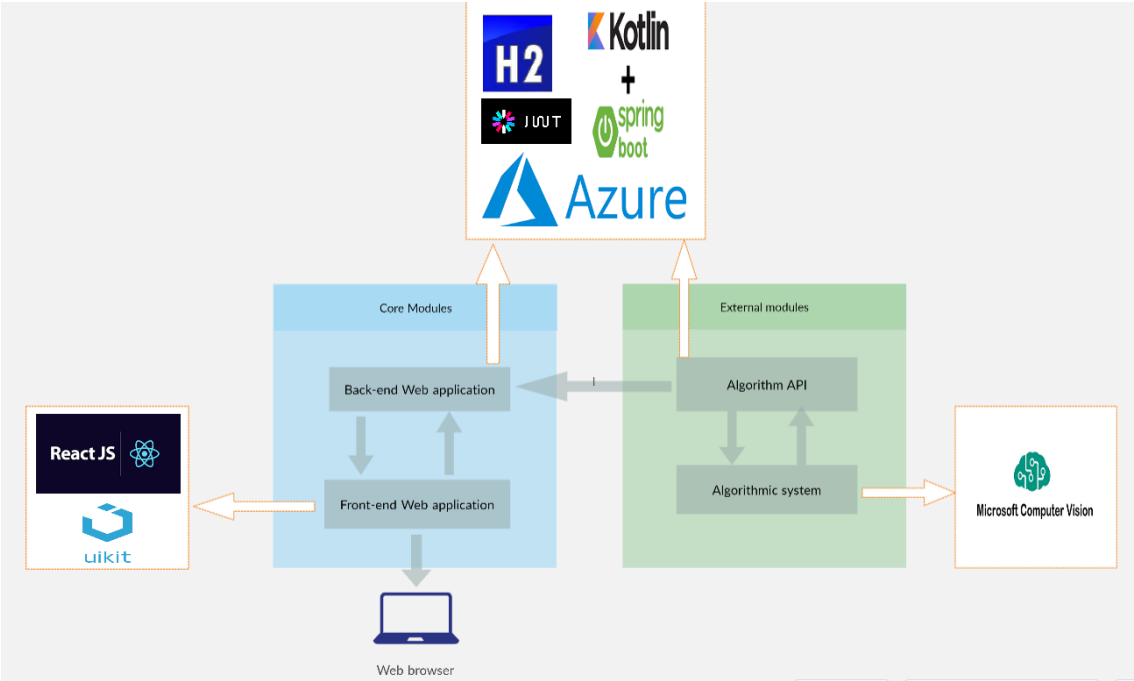

Figure 5.1: Technologies architecture diagram

5.3 Web application functionalities

During this section, we demonstrate all the functionalities of the web application. In order to illustrate these functionalities, we used images of the final version of the web application alongside their respective explanations.

5.3.1 Allow algorithms to be audited

One of the requirements for using our web application is the creation of a user account due to performance and security questions already mentioned in section 5.1, like the providence of the users' requests information, for instance. After creating an account, or after the sign-in, the users are presented on the homepage (Figure 5.2) with a description of both the importance of allowing algorithms to be audited as well as the importance of auditing algorithms. In the section where it is described the significance of allowing algorithms to be audited, the users can press the button located at the end of the section (green button with a withe arrow), which will then redirect them to the page where they will introduce the details of their algorithms (Figures 5.3 - 5.4).

The first details to be introduced are related to the algorithm itself. These fields will allow the auditor to get more information about the purpose of the algorithm and possible problems the algorithm might have. The parameters in this first section (Figure 5.3) are: the name of the algorithm, the category where it fits, an email for auditors to provide some feedback to the developers, a description of the purposes and goals of the algorithm, and finally, the endpoint where to send the datasets, thus allowing the auditing of the algorithm.

The second part of this component (Figure 5.4) is where the users will specify the input and output parameters of the algorithm. As mentioned previously, the system allows the users to upload algorithms that receive more than one input parameter, and so, it is possible to find in the "Input Specification" section a field where the user can select the number of input parameters. The fields of each input parameter are: the name of the input parameter, the input format, the minimum and the maximum number of samples of the respective parameter, a description of the input itself, and finally, an image of an input example. Regarding the output specification, the user only has to select the format of the output.

At last, the users just had to click on the "Upload your algorithm !" button, and then the algorithm will be available for other users to audit.

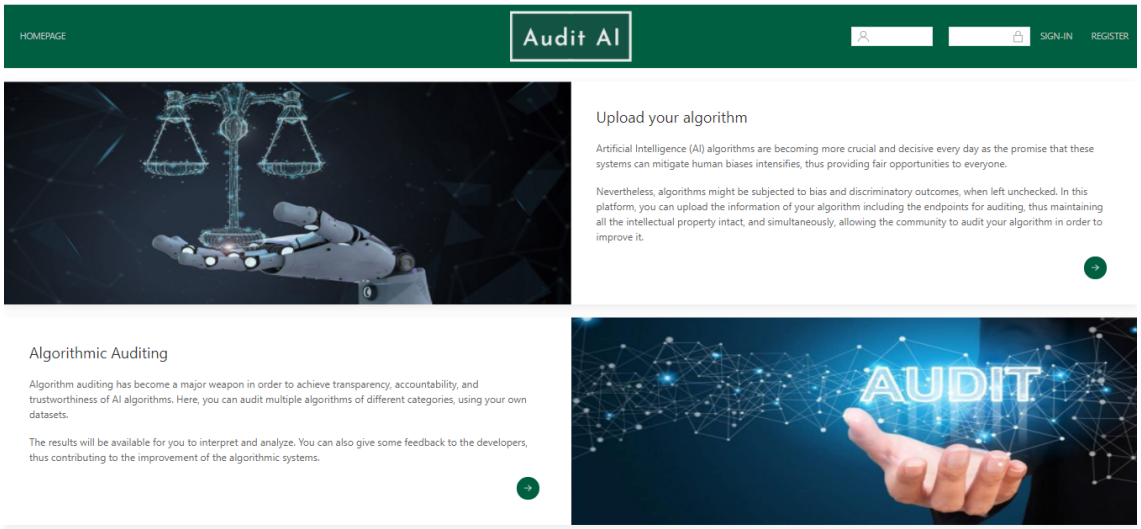

Figure 5.2: Home page of the web application

5.3.2 Create an auditing request

On the homepage (Figure 5.2), alongside the description of how important it is to allow the algorithms to be audited by other auditors, there is also a description explaining the significance of auditing algorithms. In this section, the users can press the button located at the end of the division, which will then redirect them to the component where it is possible to visualize all the algorithms available to audit (Figure 5.5). In this component, it is possible to filter the algorithms available by their category and even search them

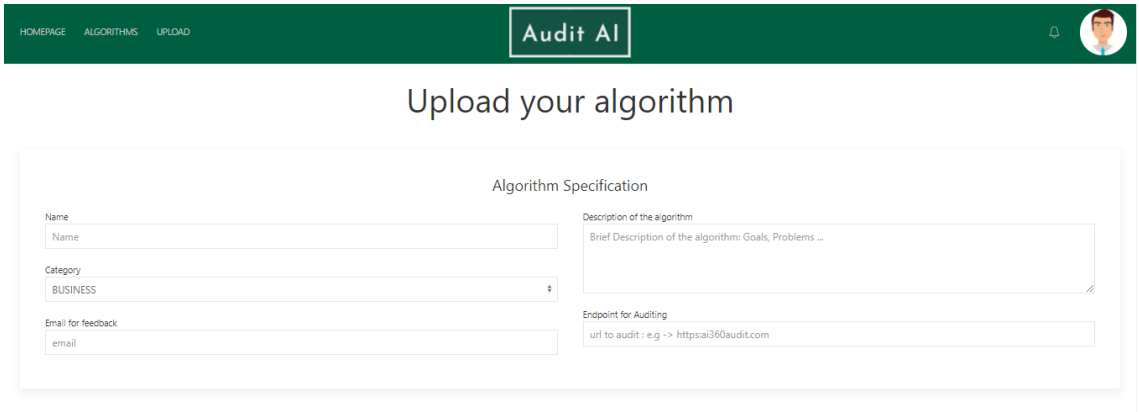

Figure 5.3: Algorithm upload page part 1

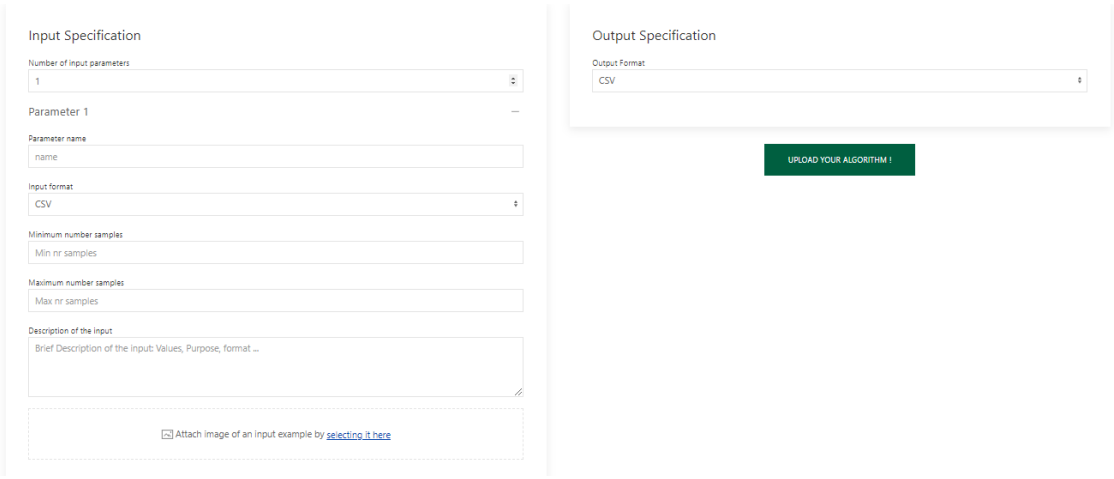

Figure 5.4: Algorithm upload page part 2

by their name. We use pages to list all the algorithms, which we found to be the most efficient way to organize them.

At the list of available algorithms (Figure 5.5), the users can choose to see additional information about a specific algorithm by pressing the button located in the "More info" column of the respective algorithm. The users then will be redirected to the page where all the algorithm's details are shown, including the input specifics (Figure 5.6).

Here, the users can visualize all the algorithm's data provided by the users that uploaded the algorithm, with particular detail to the input specifics, which show the users the type of dataset required to create an audit request. Once more, the algorithm used to illustrate the component is the Microsoft Computer Vision algorithm. It is significant to notice that this is not the official email nor the official description of the algorithm API.

Still in the component of the specifics of a particular algorithm (Figure 5.6), it is located a button that allows the users to create an auditing request, the users need to press the "Audit Now !" button, which will then redirect them to the page where they can insert their private dataset(s), according to the input specification, thus allowing the creation of an auditing request. In this new section (Figure 5.7), the users can choose

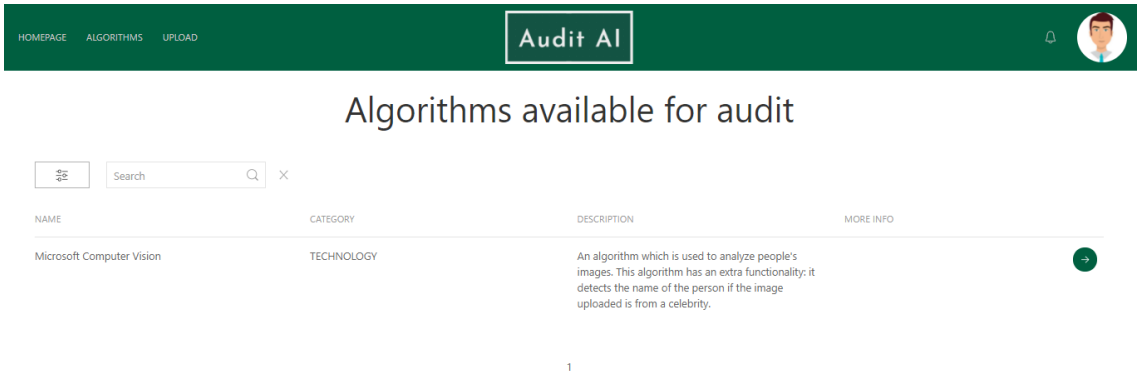

Figure 5.5: Page with the list of algorithms available for auditing

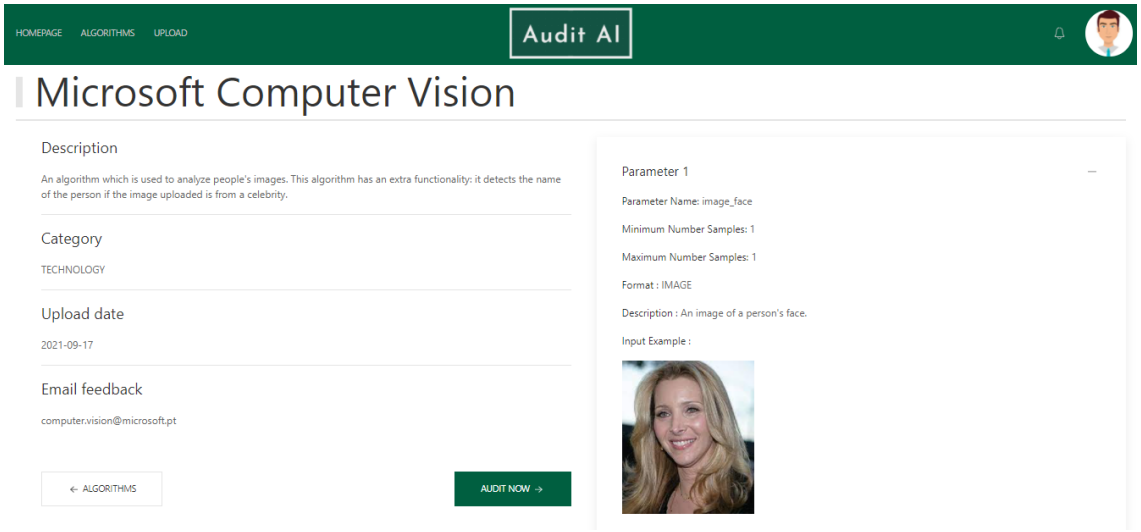

Figure 5.6: Page with the details of the Microsoft Computer Vision algorithm

their private datasets by selecting them in the "selecting the file" link. After selecting the file(s) to use, the users can then create an auditing request by pressing the button "Create Request". Ultimately, after the auditing request creation, the users can observe the status of all of their requests in the list of auditing requests (Figure 5.10).

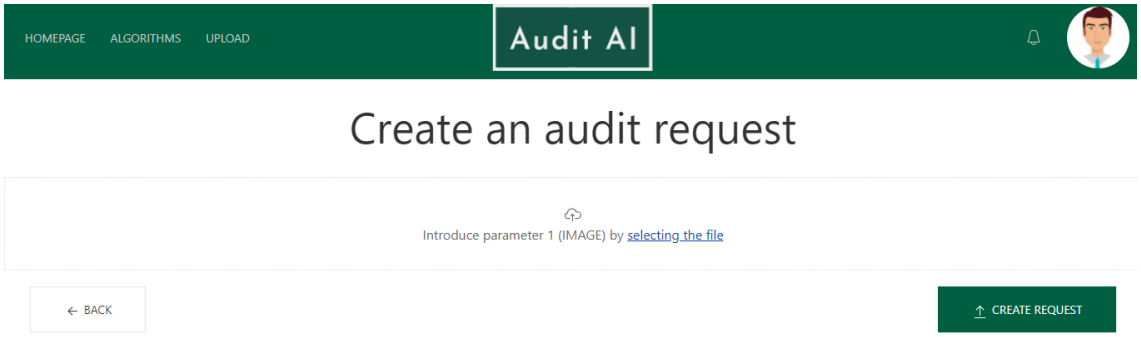

Figure 5.7: Page where users upload their datasets

Creating an audit request is simple in the eyes of the user, nonetheless, it is a little bit more complex to understand the whole process, and so, we made two diagrams to help the visualization of the step-by-step in the creation of an auditing request: a summary diagram (Figure 5.8) and an activity diagram (Figure 5.9). Listed below there are enumerated (according to the summary diagram) and explained all the stages in the formation of an auditing request:

1. **Transmitting the datasets to the back-end:** This is the first phase in the creation of an auditing request. As mentioned above, the users choose the dataset(s) intended to be audited and then press the button to create the request. It is significant to mention that all the files introduced at this stage will be uploaded to an Azure container, being accessible via an URL. The front-end then sends all the information required to create an auditing request to the back-end. The transmitted parameters are: the URL(s) of the file(s) uploaded and the id of the algorithm that is going to be audited.

2. **Creating an algorithm request in the algorithm API:** When the back-end receives the data to create an auditing request, it will produce a REST request to interact with the selected algorithm through the endpoint previously provided by the user that uploaded it. At this stage, the back-end of the web application will send to the algorithm endpoint the following parameters: the URL(s) of the dataset(s) that will be audited, and also the auditing request id. At last, the algorithm API returns the response confirming it has received the request, and so the web application continues its functioning.

3. **Evaluating the datasets:** In this stage, the algorithm API transmits the datasets received (from the back-end of the web application) to the algorithmic system that will evaluate them.

   One of the main issues we had to face during the development of the algorithm API was the fact that we could not wait for the algorithm to evaluate the datasets

to transmit the response back to the back-end of the web application, as explained above in subsection 5.2.2. To solve this issue, as noted in subsection 5.2.3, we implemented the algorithm API using suspending functions, which allows the algorithm API to receive the information and to return the response confirming that it has received the specifics of the datasets almost immediately, without having to wait for the algorithmic system to evaluate the datasets first. Thereby, the back-end of the web application does not stay locked while the algorithm evaluates the datasets.

Once the algorithmic system has evaluated the datasets of the auditing request, it will return the result to the algorithm API, which will then upload it to a container (in our case, in an Azure container), thus being accessible through an URL.

4. **Transmitting the algorithm result to the back-end:** After uploading the outcome received by the algorithmic system, the algorithm API will send a REST request to the back-end of the web application in order to transmit the location of the result. The back-end of the web application will then update the status of the auditing request to "Evaluated".

5. **Informing the user that the result has arrived:** When the back-end of the web application changes the status of the auditing request to "Evaluated" it also sends a notification to the users informing the result of the auditing request has arrived. The users can then visualize in the list of requests that the status of the auditing request has changed to "Evaluated" (Figure 5.10).

   Finally, the users can press the "Evaluated" link, which redirects them to the result component where they are capable to visualize the evaluation details, and more importantly, download the result (Figure 5.11).

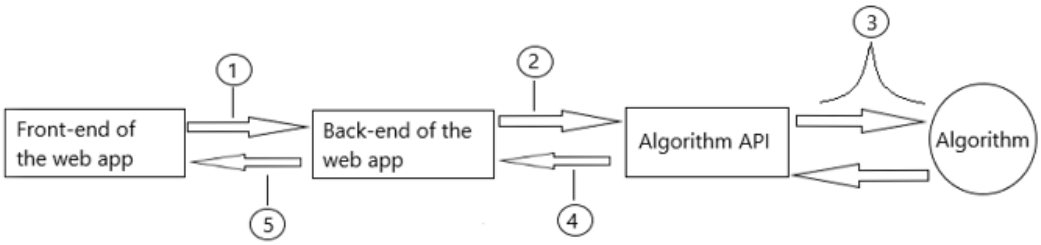

Figure 5.8: Summary diagram of the creation of an auditing request

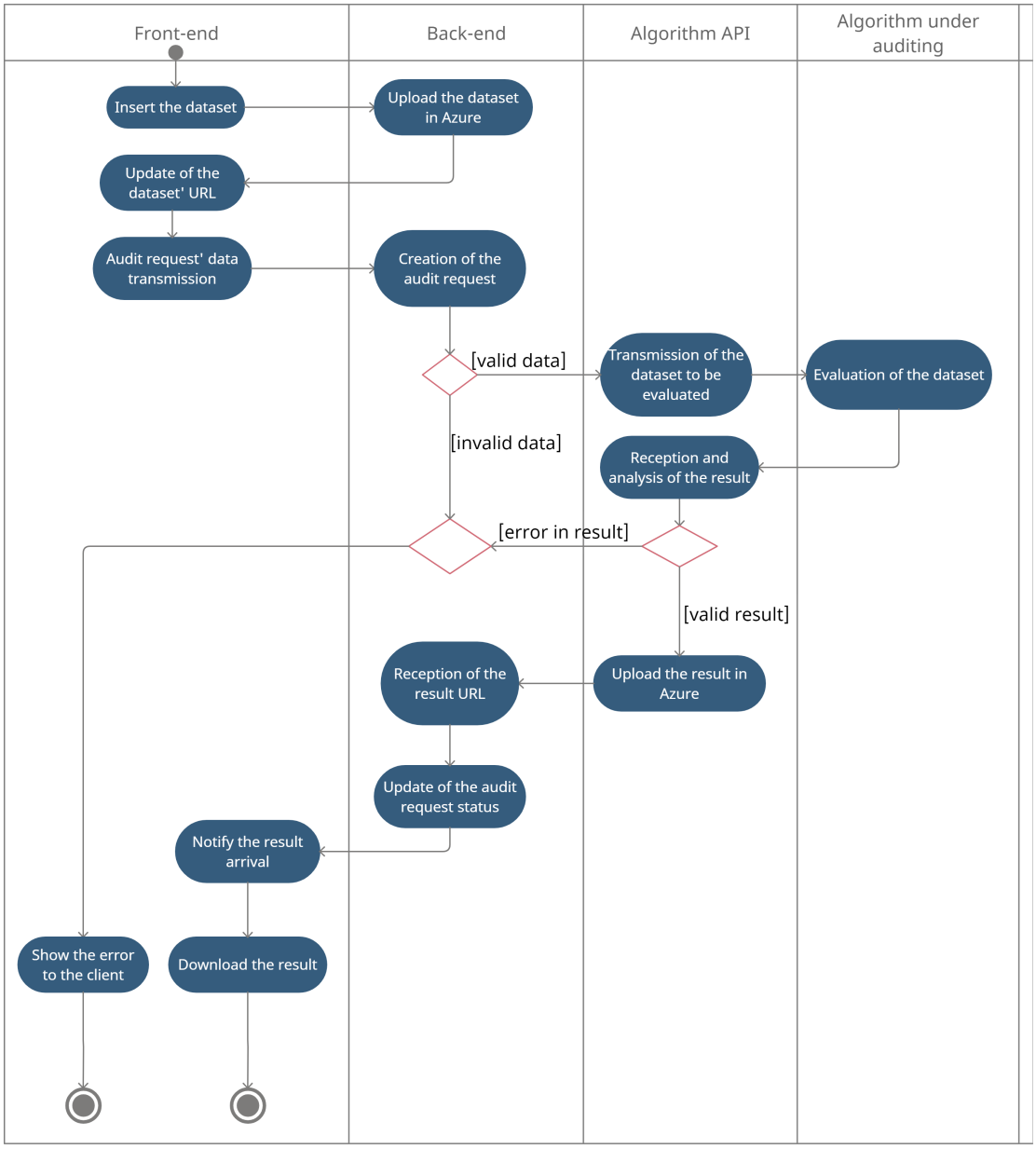

Figure 5.9: Activity diagram of the creation of an auditing request

5.3.3 Download the auditing requests' results

When the results of the auditing requests arrive at the back-end of the web application, the users also receive a notification informing that the results of the auditing requests are now available and can be downloaded. By pressing the notification button, the users are redirected to the list of their auditing requests (Figure 5.10). In this component, the users can then press the "Evaluated" link associated with the specific auditing request in order to access its result.

In this section (Figure 5.11), the users can observe some additional details regarding the evaluation, such as the evaluation date, but most importantly, download the results by pressing the "Download Results!" button.

Figure 5.10: Page with the users' list of auditing requests

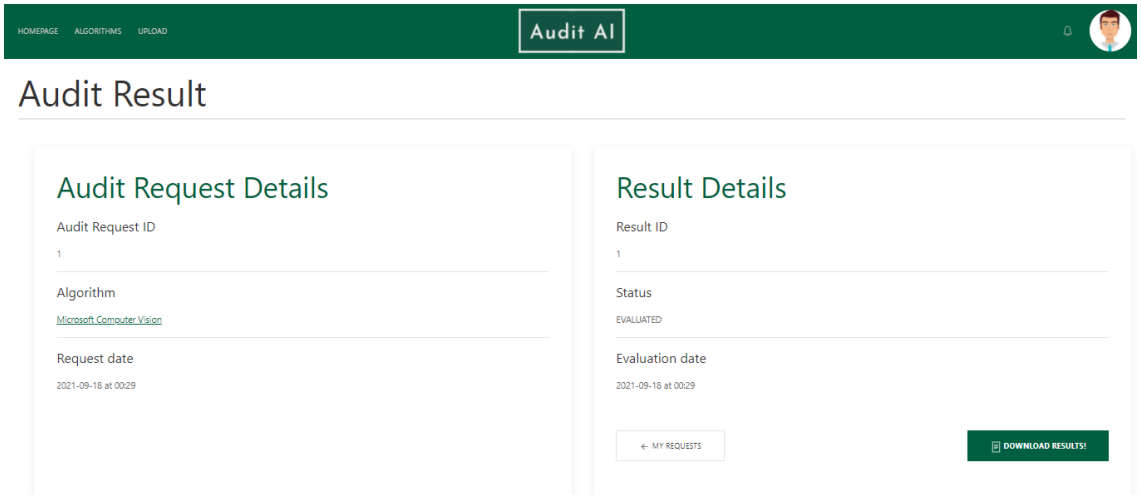

Figure 5.11: Page of the result of an auditing request

5.3.4 Visualize the uploaded algorithms' statistics

Lastly, the users are able to observe some statistics of their own algorithms, mainly the number of auditing requests that each algorithm has received. In order to access this component (Figure 5.12), the users just need to click on the profile picture at the superior right corner of the screen, and in that menu, click on the link "my-algorithms".

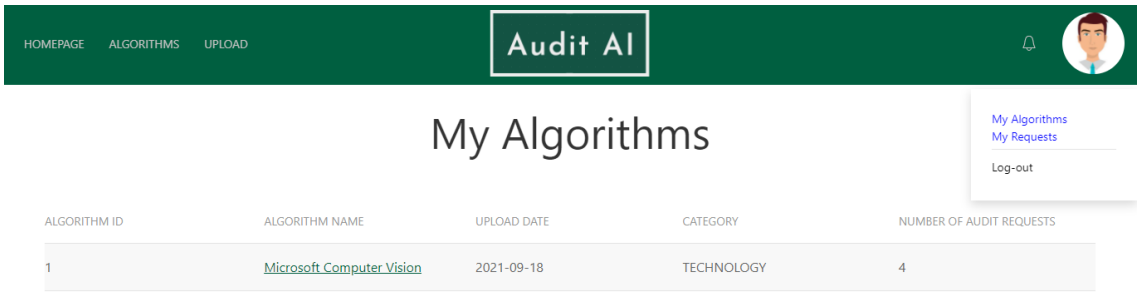

Figure 5.12: Page with the list of the users' uploaded algorithms

5.4 Initial Evaluation

Finished the development of the web application, we had to thought how we could receive some feedback about the final version of the system. For that purpose, we decided to record a video demonstrating the complete functioning of the web application using a tool named loom [21e]. When we finished recording the video, we sent it to the interviewees that participated in the in-depth interviews. We then asked them to provide us some final considerations about the global state of the system, if possible.

Globally, all the interviewees that provided us some feedback regarding the final version of the web application (P1, P6, P8, P9) considered that the overall status of the web application was appropriate and well designed, which was very glad to read. Nonetheless, and despite all the positive responses received, the interviewees also gave us some suggestions that could have been interesting to implement and others that might be considered later for future work.

One of the suggestions given to us by participants P1, P8, and P9 (as it is visible in the quote below by P8), was to support the insertion of a batch of inputs per auditing request. Our web application allows the creation of auditing requests that require more than one input file (it enables the upload of multiple input files, however, they have to be of different input parameters). Nevertheless, despite the limitation our web application has to enable the upload of batches of inputs per input parameter, we can not entirely solve this issue as the number of instances per input parameter is defined by the user that uploads the algorithm, as it is exemplified in the case of the API of the Microsoft Computer Vision that only evaluates one input file at a time. In sum, this suggestion has

great significance, and it is one of the aspects to improve in the web application. However, we are not able to ensure that the insertion of batches of input will always be possible since the number of input files per parameter depends on the specification made by the user that uploads the algorithm.

> "Maybe providing a way to audit with a batch of inputs could be useful as that is what an auditor would do in practice? " (P8-auditor)

Moreover, another proposal provided by the interviewees (P8 and P9) was to introduce a section that enables the users (auditors) to visualize the results of the auditing requests within the website (as mentioned in the quote below by P8). Additionally, P9 also referred to us, as quoted below, that this section could have the functionality of quantifying the disparities in results based on input characteristics. Although this suggestion seemed very relevant to implement, it was relatively out of the scope of our prototype. We tried to provide a web application that facilitated the communication between auditors and algorithms, and so, we did not focus on creating an analysis section within the website. Nonetheless, we think this suggestion could be a valuable addition to the prototype as an area of future work.

> "In addition to a way for downloading the results, it would be useful if you can display a summary of the results within the website. " (P8-auditor)

> "The functionality of quantifying the disparities in results based on input characteristics seems to be missing and it would be an important area of future work. " (P9-researcher)

Finally, the last suggestions were provided to us by P6, and they were related to some technical questions of the system. P6 pointed out that access to the algorithm endpoint should be safe and that we should clarify what security measures were implemented to ensure the security of the interaction with the endpoints. The other suggestion was to change the status of the auditing request from "evaluated" to "processed" since the term evaluated could mislead the users and should only be used after the validation of the results has been done.

In sum, the feedback we received regarding the global performance of the web application was highly positive. All the feedback and suggestions we have received were also considerably relevant, and they are significant fields to improve in the future work of the web application.

5.5 Summary

Through this chapter, we have illustrated and explained how the system works and all of its functionalities. It is never enough to highlight the significant impact that the feedback received during the in-depth interviews had on the development of the system and in its final version.

Our proposed solution intended to help the tasks of both auditors and developers, either by facilitating the auditing of algorithms or simply by allowing the users' algorithms to be audited. All the functionalities incorporated into the system had the goal of adding value to the algorithmic auditing field, and we really believe that this system can help the users to improve the existing algorithmic systems through auditing practices the platform permits.

6

Discussion

Algorithmic systems are becoming crucial in the decision-making processes that have a significant impact on our society, from deciding who gets financial credit to domains as the judicial system, for instance. One of the main arguments for the use of algorithmic systems to make all of these significant decisions rely on the concept that algorithms are unbiased, contrary to humans since humans can have their own bias when they are deciding, and algorithms do not. Nonetheless, it was demonstrated in subsection 2.3.1 that "bias-free" systems do not exist and that algorithmic systems, when left unchecked, not only can replicate the existing bias and discriminatory outcomes but even augment them. Thus, our dissertation focused on creating a web application that eased and improved the external auditing of algorithmic systems.

In this chapter, we discussed all the key findings we were able to obtain through the performance of in-depth interviews and literature review. Moreover, we also addressed the implications and limitations of our developed solution. Finally, we mentioned several recommendations, which can be considered later in future work on the field.

6.1 Key Findings

In order to understand all the requirements, suggestions, and obstacles our web application might face, we performed multiple in-depth interviews with scientists and researchers related to the area alongside the literature review. Below, it is listed the key findings related to the algorithmic auditing research field and also associated with our web application. It is relevant to mention that this data had a significant impact on the development of our solution.

- **The significance of auditing algorithms:** During the research process, which includes both the literature review and in-depth interviews, it was mentioned that despite the increasing tendency in the usage of algorithmic systems during decision-making processes, there has not been a follow-up regarding the monitoring and verification of those systems [Gus+18].

  The literature analysis displayed the two main factors responsible for introducing bias in the systems: the use of biased datasets and errors during the development stages of the algorithm (subsection 2.3.1). The usage of discriminatory and biased systems during decision-making processes can harm the users of those systems since these decisions impact the users' lives, from judicial sentences to government funds allocation (fields of great relevance), for instance. So, with this in mind, the data suggested that algorithmic auditing can be a crucial weapon in the fight for more fair and non-discriminatory systems by improving the overall performance of the algorithms.

- **Challenges in the algorithmic auditing field:** Through the literature analysis and in-depth interviews, it was retrieved multiple obstacles, which were related to the execution of algorithm audits, from both the point of view of the auditors and entities responsible for the algorithms.

  Regarding the entities that possess the intellectual property of the algorithms, the data suggested there was a high level of reluctance of these companies to allow their algorithms to be audited, in an attempt to protect their property (section 2.3).

  Concerning the auditors, the data indicated that the access to the algorithms for the auditors to test them was difficult since entities do not allow their algorithms to be easily accessible. Additionally, it was mentioned to us by the interviewees that the cost, monetary and temporary, of the auditings was high and that it did not exist many tools that could support the auditing of algorithms (subsection 4.1.3).

- **Obstacles in the implementation of the web application:** One of the most crucial stages of the development of our system was to understand the difficulties that may have arisen during its implementation. We were able to explore this topic in a more broad context during the in-depth interviewees since we had more liberty to discuss the specifics of our prototype, thus understanding more fully the feedback we were receiving about the system. Nonetheless, it is significant to mention that the literature review was also relevant in the definition of the challenges of the system.

  During the in-depth interviews, the feedback we received indicated the existence of serious obstacles (subsubsection 4.1.4.1): firstly, there might be an abusive use of the system on the part of some users. Second, by definition, the system can not be considered an external auditing tool. Finally, there might exist the possibility that the entities do not share the algorithms they are using in production.

- **How should the system be designed and implemented:** Lastly, our research (literature analysis and in-depth interviews) allowed us to receive numerous suggestions about the implementation of the system as well as the definition of the requirements needed (section 4.2).

  From all of the recommendations received, it is significant to highlight those who had a greater impact on the final version of the system: the first recommendation adopted was the restriction of the input/output formats available for the datasets and results. The second was the improvement of the details of the system on the homepage. The third and final one, which had the higher impact, was the usage of algorithms' APIs rather than algorithms' executables.

6.2 The solution

The main goal of our web application was to support external auditing practices by allowing users (auditors) to interact with algorithms that otherwise would not be possible and then analyze the results obtained. To achieve our objective, we had to fulfill the multiple requirements the system needed to have, and so, the system had to: **enable developers to upload the algorithms to be audited**, **be scalable**, **enable auditors to audit the algorithms with their own datasets**, **provide the results of the auditing requests**, **provide personal records of auditing requests and algorithms uploaded**, and **be secure**. The complete list and explanation of the system's requirements can be found in section 5.1. It is significant to mention that during the implementation of the web application, we completed various measures to achieve all of the system requirements needed. The full specification of the architecture used in the development of the system is available in section 5.2.

In the final version of the prototype, the users (auditors) could interact with the available algorithm (Microsoft Computer Vision), thus auditing the algorithms with their own datasets. The users (developers) could also introduce the specifics of their algorithms for them to be audited.

Comparing the final version of the prototype with the tools already available in the market, we believe that the system developed complements the existing tools (like aequitas or AI Fairness 360, which only allowed the auditing of datasets) considering that our platform enables the auditing of the algorithms without its source code. Thus, it is our opinion that the developed solution achieved the goals initially proposed and that can add value to the field of algorithmic auditing.

6.3 Limitations

Despite our opinion that the purposes of our web application had been fulfilled, it is significant to notice that during the research, and development stages, we faced different challenges and obstacles. The first main challenge we had was in the recruitment of interviewees. Due to the COVID situation and lack of papers regarding external algorithmic auditings, it was tough to gather interviewees that had previous experiences in performing external algorithmic audits. To solve this issue, we chose to interview not only authors that performed external audits but also scientists and researchers that have done studies within the algorithmic auditing field.

The other significant limitation is related to the web application, more specifically, with the number of files per input parameter. One of the limitations our system has is the impossibility to upload multiple datasets (input files) within the same input parameter. Our platform allows the introduction of different files. However, they have to be related to distinct input parameters. Despite this limitation within the system, it is also important to mention that this feature can not always be ensured since the number of datasets (files) introduced per input parameter is defined by the users (developers) that upload the specifics of the algorithms.

6.4 Recommendations

Finally, we have a few recommendations for the future improvement of both the platform and the algorithmic auditing field. The first suggestion we have is the continuation of the development of tools to help developers and scientists to identify and analyze the bias within the datasets used in the training stages since the use of biased datasets is one of the main factors responsible for the discriminatory results present in the algorithmic systems.

The next recommendations we have are related with the improvement of the web application. Firstly, the possibility to upload multiple files within the same input parameter can be an advantageous measure since this would be what auditors would do in practice. Nonetheless, once again, it is significant to refer that this improvement is also dependent on the users (developers) that introduce the specifics of the algorithm in the platform. Moreover, it would be interesting to have a section where the results could be shown to the user within the platform and perhaps even an area where it would be possible to compare the results of the different algorithms.

In addition to our recommendations, it is crucial to highlight the excellent work that has been done in the field of algorithmic auditing and reinforce the significance this field has in the fight for less discriminatory and biased systems. This work should be maintained, and if possible, increased.

7

Conclusion

Algorithmic systems are becoming crucial in the decision-making processes that have a significant impact on our society, from deciding who gets financial credit to domains as the judicial system, for instance. Our research revealed that one of the main arguments for the use of algorithmic systems in these significant decisions relies on the concept that algorithms are unbiased, contrary to humans since humans can have their own bias when they are deciding, and algorithms do not. Nonetheless, it also revealed that algorithmic systems, when left unchecked, not only can replicate the existing bias and discriminatory outcomes but even augment them (section 2.3). So, with this in mind, algorithmic auditing practices surges as a tool to achieve more transparent and accountable systems. Algorithmic auditing does not focus only on comparing inputs and outputs, but also on the foundations of the algorithm, fields that technical tools alone cannot ascertain.

This dissertation had the goal to create a platform, a web application, that would allow the external auditing of algorithms. The system intended not only to ease the auditing practices but also to encourage developers and entities to share their algorithms to be audited. In order to develop the system in a more sustained and complete way, we decided to perform multiple in-depth interviews, which allowed us to receive some feedback about the challenges we could face as well as possible design suggestions. In the end, it was visible the significant impact these interviews had on the final version of the web application (section 4.2). It is considerable to mention that we encountered some difficulties regarding the gathering of interviewees, not only due to the COVID situation but also due to the lack of engineers that have performed external algorithmic audits. Notwithstanding, the interviewees we had were extremely helpful and were a crucial piece in the systems' development.

We consider that our system achieved its purposes, even though it's still in an early stage in order to be available for the public in general. Nonetheless, we were able to demonstrate the global functioning of the web application by auditing the Microsoft Computer Vision algorithm. So, we could display how the system would work for other algorithms too. We believe that algorithmic auditing is and will be a significant method in order to achieve more just and fair algorithms. Platforms that help auditors execute

the algorithmic audits, like ours, for instance, are extremely valuable and can contribute immensely to the improvement of this field.

At last, it is visible that there is still space for the development of additional tools which could help auditors, and also other versions of instruments to assist developers and scientists in the analysis of datasets used in the training phase. The work done regarding the instruction of developers to societal and ethical questions associated with algorithms should be maintained, and if possible, increased.

Bibliography

[22] 2022. URL: https://www.aicos.fraunhofer.pt/en/home.html (cit. on p. 3).

[Agg20] N. Aggarwal. *Biases in Machine Learning*. en. June 2020. URL: https://towardsdatascience.com/biases-in-machine-learning-61186da78591 (cit. on pp. 10–12).

[20a] *AI Auditing Framework*. en. Publisher: ICO. July 2020. URL: https://ico.org.uk/about-the-ico/news-and-events/ai-auditing-framework/ (cit. on pp. 6, 16).

[20b] *An internal auditing framework to improve algorithm responsibility*. en-US. Oct. 2020. URL: https://hellofuture.orange.com/en/auditing-ai-when-algorithms-come-under-scrutiny/ (cit. on pp. 1, 9, 10, 13, 16).

[aut] auth0.com. *JWT.IO*. en. URL: http://jwt.io/ (visited on 10/16/2021) (cit. on p. 41).

[bae19] baeldung. *Spring Boot With H2 Database | Baeldung*. en-US. Apr. 2019. URL: https://www.baeldung.com/spring-boot-h2-database (visited on 10/16/2021) (cit. on p. 41).

[21a] *Balsamiq Wireframes - Industry Standard Low-Fidelity Wireframing Software | Balsamiq*. 2021. URL: https://balsamiq.com/wireframes/ (cit. on pp. 21, 26).

[BL94] W. E. Bijker and J. Law. *Shaping Technology / Building Society: Studies in Sociotechnical Change*. en. Google-Books-ID: 7i1hO90ZDHUC. MIT Press, Sept. 1994. ISBN: 978-0-262-26043-5 (cit. on pp. 2, 10).

[Boy17] G. A. Boy. *The Handbook of Human-Machine Interaction: A Human-Centered Design Approach*. en. Google-Books-ID: t2kQEAAAQBAJ. CRC Press, Nov. 2017. ISBN: 978-1-317-02947-2 (cit. on p. 18).

[BC06] V. Braun and V. Clarke. "Using thematic analysis in psychology". In: *Qualitative Research in Psychology* 3.2 (Jan. 2006). Publisher: Routledge _eprint: https://www.tandfonline.com/doi/pdf/10.1191/1478088706qp063oa, pp. 77–101. ISSN: 1478-0887. DOI: 10.1191/1478088706qp063oa. URL: https:

//www.tandfonline.com/doi/abs/10.1191/1478088706qp063oa (visited on 08/20/2021) (cit. on pp. 25, 26).

[BG18] J. Buolamwini and T. Gebru. "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification". In: *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*. Ed. by S. A. Friedler and C. Wilson. Vol. 81. Proceedings of Machine Learning Research. New York, NY, USA: PMLR, Feb. 2018, pp. 77–91. URL: http://proceedings.mlr.press/v81/buolamwini18a.htm (cit. on p. 5).

[CIM] J. Chou, R. Ibars, and O. Murillo. "In Pursuit of Inclusive AI". en. In: (), p. 21 (cit. on p. 11).

[Cla+20] G. G. Clavell et al. "Auditing Algorithms: On Lessons Learned and the Risks of Data Minimization". In: *AIES '20: AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA, February 7-8, 2020*. Ed. by A. N. Markham et al. ACM, 2020, pp. 265–271. DOI: 10.1145/3375627.3375852. URL: https://doi.org/10.1145/3375627.3375852 (cit. on p. 12).

[20c] *Commonly used file formats in Data Science*. en-us. Section: Python. July 2020. URL: https://www.geeksforgeeks.org/commonly-used-file-formats-in-data-science/ (visited on 09/13/2021) (cit. on p. 36).

[21b] *Composing suspending functions | Kotlin*. en-US. 2021. URL: https://kotlinlang.org/docs/composing-suspending-functions.html (visited on 09/17/2021) (cit. on p. 41).

[Dam19] S. Damayanthi. *Thematic Analysis of Interview Data in the Context of Management Controls Research*. 1 Oliver's Yard, 55 City Road, London EC1Y 1SP United Kingdom: SAGE Publications, Ltd., 2019. ISBN: 978-1-5264-7485-8. DOI: 10.4135/9781526474858. URL: http://methods.sagepub.com/dataset/thematic-analysis-management-controls (visited on 08/23/2021) (cit. on pp. 25, 26).

[Das18] J. Dastin. "Amazon scraps secret AI recruiting tool that showed bias against women". en. In: *Reuters* (Oct. 2018). URL: https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G (cit. on p. 1).

[dle] dlepow. *Azure Container Instances documentation - serverless containers, on demand*. en-us. URL: https://docs.microsoft.com/en-us/azure/container-instances/ (visited on 09/17/2021) (cit. on pp. 41, 42).

[20d] *Examining the Black Box: Tools for assessing algorithmic systems*. en-GB. Apr. 2020. URL: https://www.adalovelaceinstitute.org/report/examining-the-black-box-tools-for-assessing-algorithmic-systems/ (cit. on pp. 4–8).

[GEB14] E. S. P. Gangadharan, W. V. Eubanks, and S. Barocas. "DATA AND DISCRIMINATION: COLLECTED ESSAYS". en. In: *NEW AMERICA* (2014), p. 6 (cit. on p. 2).

[21c] *Getting Started – React*. en. 2021. URL: https://reactjs.org/docs/getting-started.html (visited on 09/16/2021) (cit. on p. 40).

[GF17] B. Goodman and S. Flaxman. "European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation"". en. In: *AI Magazine* 38.3 (Oct. 2017). Number: 3, pp. 50–57. ISSN: 2371-9621. DOI: 10.1609/aimag.v38i3.2741. URL: https://ojs.aaai.org/index.php/aimagazine/article/view/2741 (cit. on p. 2).

[21d] *google/ml-fairness-gym*. original-date: 2019-08-06T14:03:17Z. Feb. 2021. URL: https://github.com/google/ml-fairness-gym (cit. on p. 14).

[GDM11] L. A. Guion, D. C. Diehl, and D. McDonald. "Conducting an In-depth Interview". en. In: *EDIS* 2011.8 (Aug. 2011). ISSN: 2576-0009. DOI: 10.32473/edis-fy393-2011. URL: https://journals.flvc.org/edis/article/view/127025 (cit. on p. 19).

[Gus+18] J. Guszcza et al. "Why We Need to Audit Algorithms". In: *Harvard Business Review* (Nov. 2018). Section: Economics & Society. ISSN: 0017-8012. URL: https://hbr.org/2018/11/why-we-need-to-audit-algorithms (cit. on pp. 1, 2, 10, 54).

[HDP19] A. Haddad, R. Doherty, and R. Purtilo. *Health Professional and Patient Interaction (Ninth Edition)*. en. W.B. Saunders, Jan. 2019, pp. 267–278. ISBN: 978-0-323-53362-1. DOI: 10.1016/B978-0-323-53362-1.18001-6. URL: https://www.sciencedirect.com/science/article/pii/B9780323533621180016 (visited on 10/02/2021) (cit. on p. 11).

[10] "Human Computer Interaction". In: ISSN: 2157-0485. Nov. 2010, pp. 1–4. DOI: 10.1109/ICETET.2010.85 (cit. on p. 18).

[18] *Introducing AI Fairness 360, A Step Towards Trusted AI - IBM Research*. en-US. Sept. 2018. URL: https://www.ibm.com/blogs/research/2018/09/ai-fairness-360/ (visited on 02/15/2021) (cit. on pp. 14, 15).

[KCL20] A. A. Kennedy, D. Coates, and K. Lindquist. "Auditing Government AI: How to assess ethical vulnerability in machine learning". en. In: (2020), p. 7 (cit. on p. 1).

[Kim17] P. Kim. *Auditing Algorithms for Discrimination*. en. SSRN Scholarly Paper ID 3093982. Rochester, NY: Social Science Research Network, Dec. 2017. URL: https://papers.ssrn.com/abstract=3093982 (cit. on pp. 1, 2, 4, 6, 9, 12).

[Koe19] A. Koene. "A governance framework for algorithmic accountability and transparency". en. In: (Apr. 2019), p. 4 (cit. on pp. 7, 10).

[LLL95] J. Lofland, L. H. Lofland, and P. L. H. Lofland. *Analyzing Social Settings: A Guide to Qualitative Observation and Analysis*. en. Google-Books-ID: lN_qAAAAIAAJ. Wadsworth, 1995. ISBN: 978-0-534-24780-5 (cit. on p. 19).

[21e] *Loom: Video Messaging for Work*. 2021. URL: https://www.loom.com/ (visited on 09/12/2021) (cit. on pp. 26, 50).

[Oal+20] L. Oala et al. "ML4H Auditing: From Paper to Practice". en. In: *Machine Learning for Health*. ISSN: 2640-3498. PMLR, Nov. 2020, pp. 280–317. URL: http://proceedings.mlr.press/v136/oala20a.html (cit. on p. 13).

[OS18] L. Ortolano and A. Shepherd. "ENVIRONMENTAL IMPACT ASSESSMENT: CHALLENGES AND OPPORTUNITIES". en. In: *Impact Assessment* 13.1 (Apr. 2018), pp. 3–30. ISSN: 0734-9165. DOI: 10.1080/07349165.1995.9726076. URL: http://www.tandfonline.com/doi/abs/10.1080/07349165.1995.9726076 (cit. on p. 7).

[Pat] PatrickFarley. *Computer Vision documentation - Quickstarts, Tutorials, API Reference - Azure Cognitive Services*. en-us. URL: https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/ (visited on 09/12/2021) (cit. on p. 41).

[21f] *Pros and Cons of ReactJS - javatpoint*. en. 2021. URL: https://www.javatpoint.com/pros-and-cons-of-react (visited on 10/16/2021) (cit. on p. 40).

[21g] *Pros and Cons of Using Spring Boot*. en-US. Section: Community. July 2021. URL: https://bambooagile.eu/insights/pros-and-cons-of-using-spring-boot/ (visited on 10/16/2021) (cit. on p. 41).

[Rah20] F. Rahman. *COMPAS Case Study: Fairness of a Machine Learning Model*. en. Sept. 2020. URL: https://towardsdatascience.com/compas-case-study-fairness-of-a-machine-learning-model-f0f804108751 (cit. on p. 1).

[Raj+20] I. D. Raji et al. "Closing the AI Accountability Gap: Defining an End-to-End Framework for Internal Algorithmic Auditing". In: *arXiv:2001.00973 [cs]* (Jan. 2020). arXiv: 2001.00973. URL: http://arxiv.org/abs/2001.00973 (cit. on pp. 4, 9, 13, 16).

[ray] rayid. *Aequitas*. en-US. URL: http://www.datasciencepublicpolicy.org/projects/aequitas/ (cit. on pp. 13, 14).

[20e] *Responsible AI Toolkit*. en. 2020. URL: https://www.tensorflow.org/responsible_ai (cit. on p. 16).

[20f] *Responsible AI with TensorFlow*. en. 2020. URL: https://blog.tensorflow.org/2020/06/responsible-ai-with-tensorflow.html (cit. on p. 16).

[San+14] C. Sandvig et al. "Auditing Algorithms: Research Methods for Detecting Discrimination on Internet Platforms". en. In: (2014), p. 23 (cit. on p. 5).

[Sec20] T. B. o. C. Secretariat. *Algorithmic Impact Assessment (AIA)*. eng. guidance. July 2020. URL: https://www.canada.ca/en/government/system/digital-government/digital-government-innovations/responsible-use-ai/algorithmic-impact-assessment.html (cit. on p. 7).

[Ske20] S. K. Skelton. *Auditing for algorithmic discrimination*. en. 2020. URL: https://www.computerweekly.com/feature/Auditing-for-algorithmic-discrimination (cit. on pp. 1, 2, 10, 12, 13).

[21h] *Spring | Home*. 2021. URL: https://spring.io/ (cit. on p. 41).

[Syd21] I. Sydorenko. en. Apr. 2021. URL: https://labelyourdata.com/articles/what-is-dataset-in-machine-learning (cit. on p. 9).

[US92] S. (: U.S.), ed. *ACM SIGCHI curricula for human-computer interaction*. en. New York: Association for Computing Machinery, 1992. ISBN: 978-0-89791-474-1 (cit. on p. 18).

[21i] *UIKit*. 2021. URL: https://getuikit.com/ (cit. on p. 40).

[21j] *What is User Centered Design?* en. 2021. URL: https://www.interaction-design.org/literature/topics/user-centered-design (cit. on pp. 18, 19).

[WD14] C. R. Wilkinson and A. De Angeli. "Applying user centred and participatory design approaches to commercial product development". en. In: *Design Studies* 35.6 (Nov. 2014), pp. 614–631. ISSN: 0142694X. DOI: 10.1016/j.destud.2014.06.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S0142694X14000507 (visited on 02/17/2021) (cit. on pp. 18, 19).

A

Participant Informed Consent

Fraunhofer PORTUGAL

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

www.fraunhofer.pt
Rua Alfredo Allen 455/461
4200-135 Porto, PORTUGAL

CONSENT FOR PARTICIPATION IN RESEARCH STUDY

Association Fraunhofer Portugal Research and *NOVA School of Science and Technology* are working on a research project entitled Co-design of a web platform for external algorithm auditing. The main goal of the project is to allow auditors to perform external auditing to black-box algorithms through a web application, for that purpose it is important to understand how the auditors perform the algorithm auditing and how the researchers use and develop the algorithms. We would like to count on your participation to help us reach these goals. You would participate in an interview where we will collect demographic data, insights on how the algorithm auditing is performed or how the algorithm was developed, and also your perspective in the matter of the importance of algorithm auditing. Moreover, with your permission, the conversation will be recorded to be analyzed later.

The participation in this study does not present any risk to participant's physical or mental integrity, does not involve any loss or material damage, and does not involve any payment whatsoever.

The data being collected is confidential and shall only be used by the researchers towards the originally intended ends. The data collected cannot be traced back to the owner. *Association Fraunhofer Portugal Research* and *NOVA School of Science and Technology* will take all the necessary measures to safeguard and protect the collected data in order to avoid access from unauthorized third parties.

Your participation in this study is voluntary and you are free to withdraw from it at any time with no consequence whatsoever.

We very much appreciate your contribution, which is crucial to our research.

The participant:

I declare to have read and understood this document, as well as the oral information that I was given, and I accept to take part in this research. I allow the use of the data which I provide on a voluntary basis, trusting that it will only be used for this research and with the guarantees of confidentiality and anonymity that I am being given by the researcher. I authorize the anonymous communication of the data and test results to other entities who establish partnerships for research purposes with Association Fraunhofer Portugal Research and NOVA School of Science and Technology, as well as in meetings, scientific journals, and academic papers.

Name: _____

Signature: _____

Date ___ / ___ / _____

Main researcher:

Name: Francisco Nunes

Phone: 220430328

E-mail: francisco.nunes@fraunhofer.pt

B

Interview Scripts: Auditors and Developers

Auditors Interview script

1. Introduction

Hello, my name is Rui Fernandes (Portugal), and I am currently attending my last year of my MSc in Informatics at FCT-NOVA with Fraunhofer Portugal AICOS.
My thesis project aims to develop a web application that will improve the auditing of black-box algorithms, by allowing programmers to upload an executable file of the algorithm (thus maintaining the "invisibility"-> no access to the source code) and consequently allowing auditors to audit these algorithms.
Following this concept, it is essential to understand better all the requirements needed to perform the algorithms auditing, thus your feedback and ideas about the platform (and how the platform should work) will be crucial for me to design the solution that better fits the needs of the users.
I would like to thank you all in advance for this opportunity and for your time and willingness to participate in this study.

2. Questions about algorithm auditing

2.1 Introductory Questions

- Can you please tell me about you? What is your work? Why did you start working on algorithm auditing?
- Can you please tell me about your algorithm auditing activities? Do you always work in a specific context (e.g., healthcare)? Is the purpose always the same?
- Can you give me an example of an algorithm auditing that you've recently performed? What did you do? How did you do it?
- Do you use any tools to support auditing?

2.2 Technical questions about auditing algorithms

- What are the main difficulties and obstacles when auditing algorithms?

- Through the auditing process and given the results, how is the search for discrimination/bias done?

- Which prior knowledge of the algorithm is essential to be present in order to perform efficient auditing?
  For example, aspects like, input format, description of the algorithm, etc.
  To illustrate, imagine you are in the platform and you observe an algorithm that you have never heard before, available to be audited. To perform the auditing, what do you need to know (the aspects)?

- Which features do you think should/must be present/implemented in this web-application? (E.g., different output formats like excel, graphics?)

- Before I start presenting the initial prototype mockup, is there something that you would like to say or ask?

3. Mockup design presentation

Now I will present you the initial prototype of the solution that I have made simply with knowledge of the literature analysis. This is supposed to be an interactive presentation, so feel free at any time to give ideas, feedback, or just tell me if some feature will not work out for any reason.

- During the specification of the output and input by the developers, there should be added some additional field to complete the specification?

- Given your knowledge of the algorithmic auditing field, what are the most common formats of the outputs by the algorithm?

- And, what are the best type of files (pdf, excel) that should be available to download the results?

- Are there some features, from what you have seen, that should be added?

- What would you change in the web application?

- Finally, do you have any other comments or suggestions?

4. Conclusion

Thank you once more for your contribution to this project, it was very important to me to get all these ideas and feedback from you in order to develop, in the best way possible, this solution.

Developers Interview script

1. Introduction

Hello, my name is Rui Fernandes (Portugal), and I am currently attending my last year of my MSc in Informatics at FCT-NOVA with Fraunhofer Portugal AICOS.
My thesis project aims to develop a web application that will improve the auditing of black-box algorithms, by allowing programmers to upload an executable file of the algorithm (thus maintaining the "invisibility"-> no access to the source code) and consequently allowing auditors to audit these algorithms.
Following this concept, it is essential to understand better all the requirements needed to perform the algorithms auditing, thus your feedback and ideas about the platform (and how the platform should work) will be crucial for me to design the solution that better fits the needs of the users.
I would like to thank you all in advance for this opportunity and for your time and willingness to participate in this study.

2. Questions about algorithm developing

2.1 Introductory Questions

- Can you please tell me about you? What is your work? When did you start developing algorithms?
- Do you always develop algorithms within a specific field? Or multiple fields?
- Can you give me an example of a recent algorithm or model that you've developed?

2.2 Technical questions about auditing algorithms

- Before using the dataset in the training phase, do you perform any verification (outliers, under-representation of some groups, etc) to it?

- Do you think that you understand the complexity (social, ethical) of the issues that the algorithms try to address? What do you think about these questions?

- Given your knowledge in the algorithmic developing field, what are the most common formats of the outputs by the algorithm? Does your model offer multiple output formats?

- **When introducing the details of the algorithm in the platform, what specifications do you think that should be added in order to allow the correct communication with the algorithm, thus allowing the auditing (Input and Output)? Is there any specification that should be given additional attention?**

- Before I start presenting the initial prototype mockup, is there something that you would like to say or ask?

3. Mockup design presentation

Now I will present you the initial prototype of the solution that I have made simply with knowledge of the literature analysis. This is supposed to be an interactive presentation, so feel free at any time to give ideas, feedback, or just tell me if some feature will not work out for any reason.

- Looking as a researcher/ person that develops algorithms, what would you change in the field of specification of the algorithm (input and output)?

- Do you think that should be added some field in order to better describe the algorithm to the auditors?

- Are there some features, from what you have seen, that should be added?

- What would you change in the web application?

- Finally, do you have any other comments or suggestions?

4. Conclusion

Thank you once more for your contribution to this project, it was very important to me to get all these ideas and feedback from you in order to develop, in the best way possible, this solution.

I

Initial Prototype

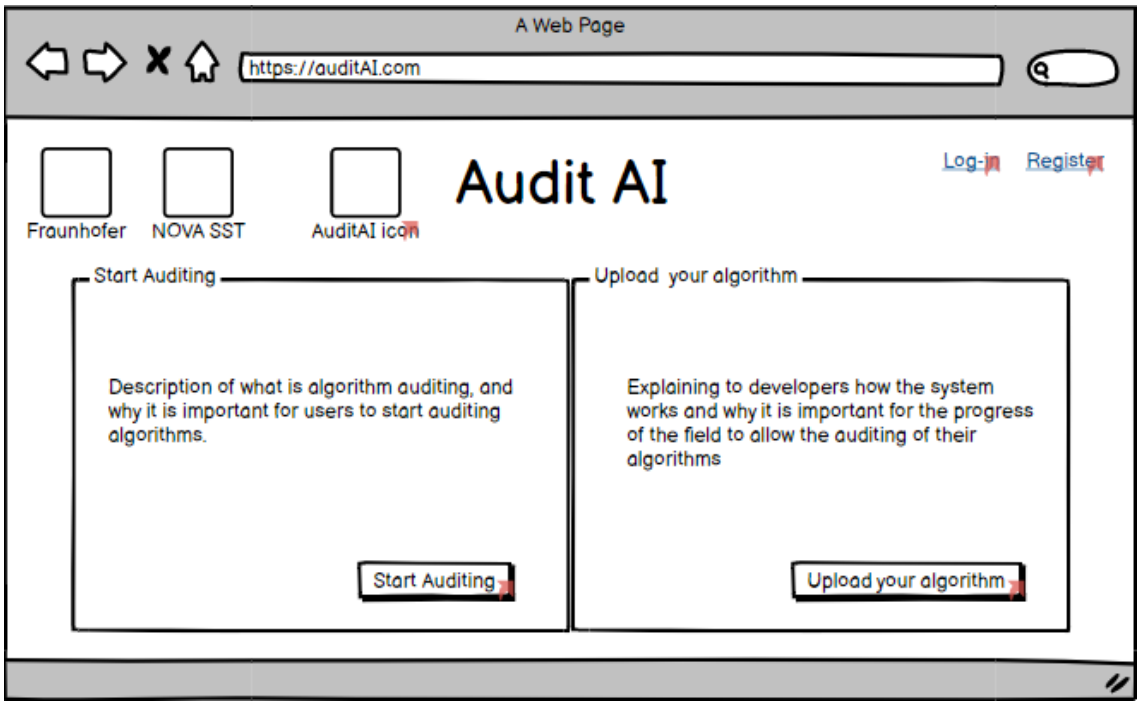

Figure I.1: Screen of the system home page.

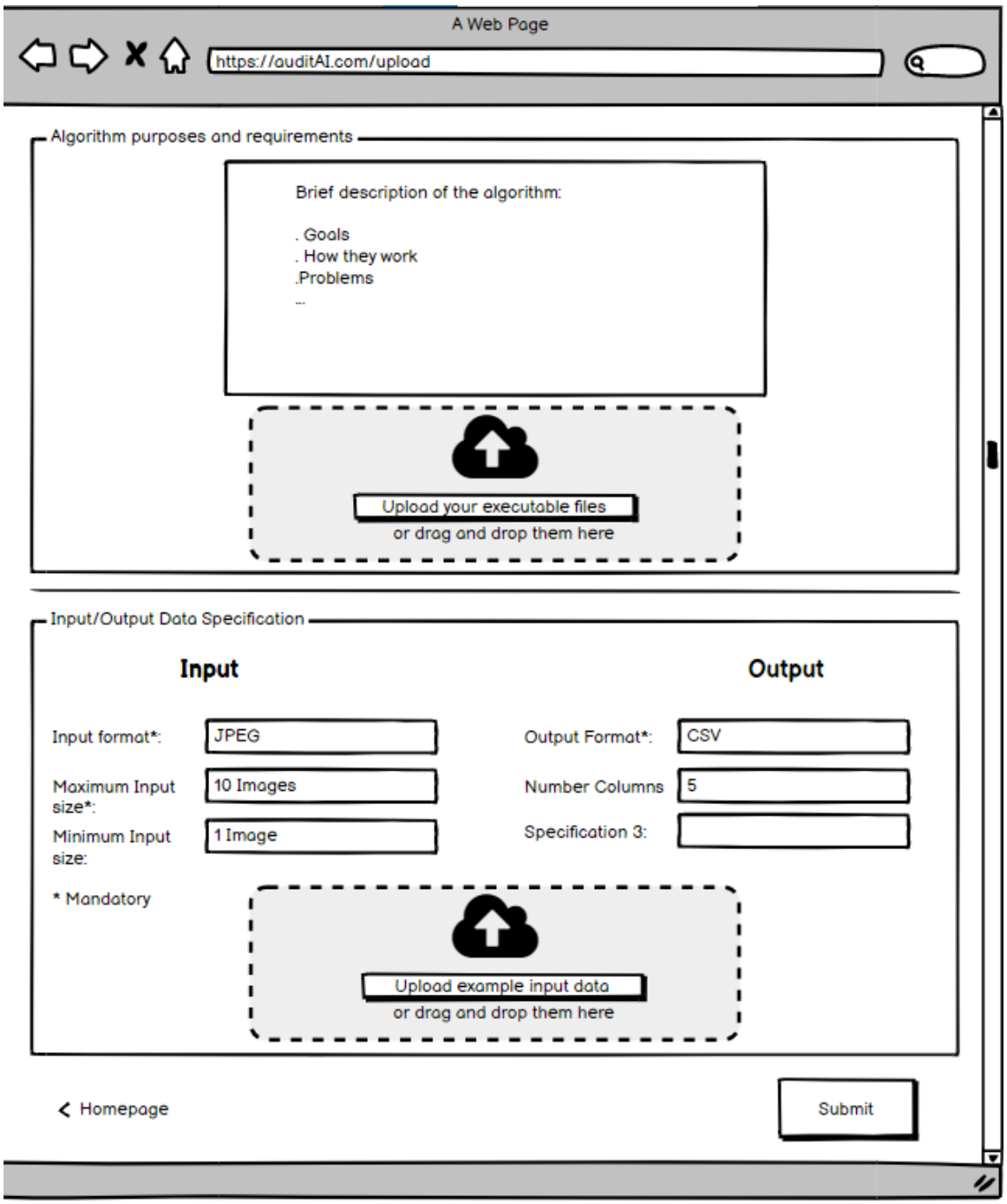

Figure I.2: Screen of the algorithm upload page. Here it is illustrated the different fields for both the algorithm and input/output specification.

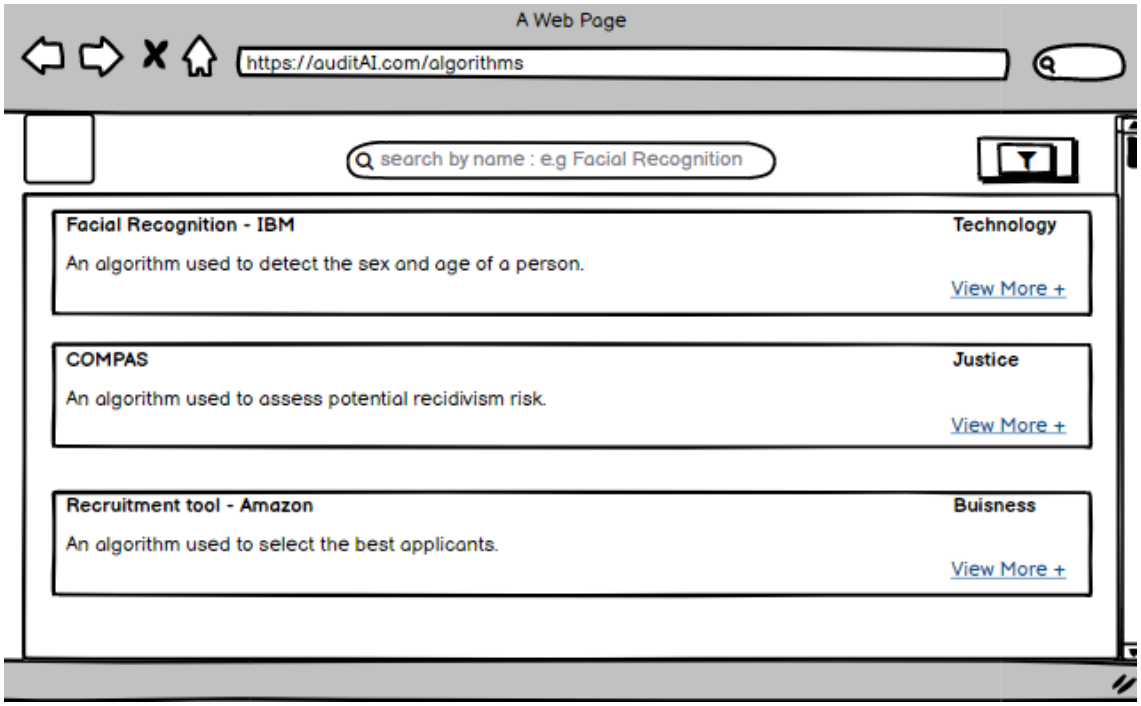

Figure I.3: Screen of the algorithms list page. Here it is illustrated the different algorithms available to be audited.

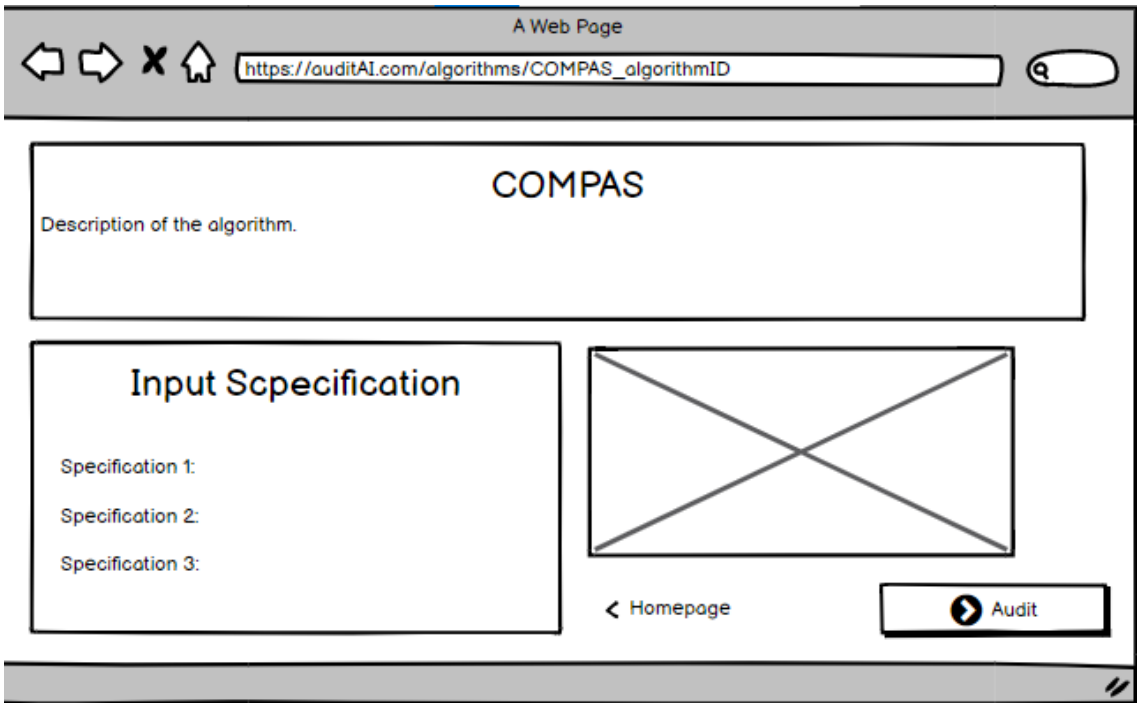

Figure I.4: Screen of the algorithms details page. Here it is illustrated all the details of the algorithm selected. The user can also choose to audit this algorithm by selecting the button "Audit".

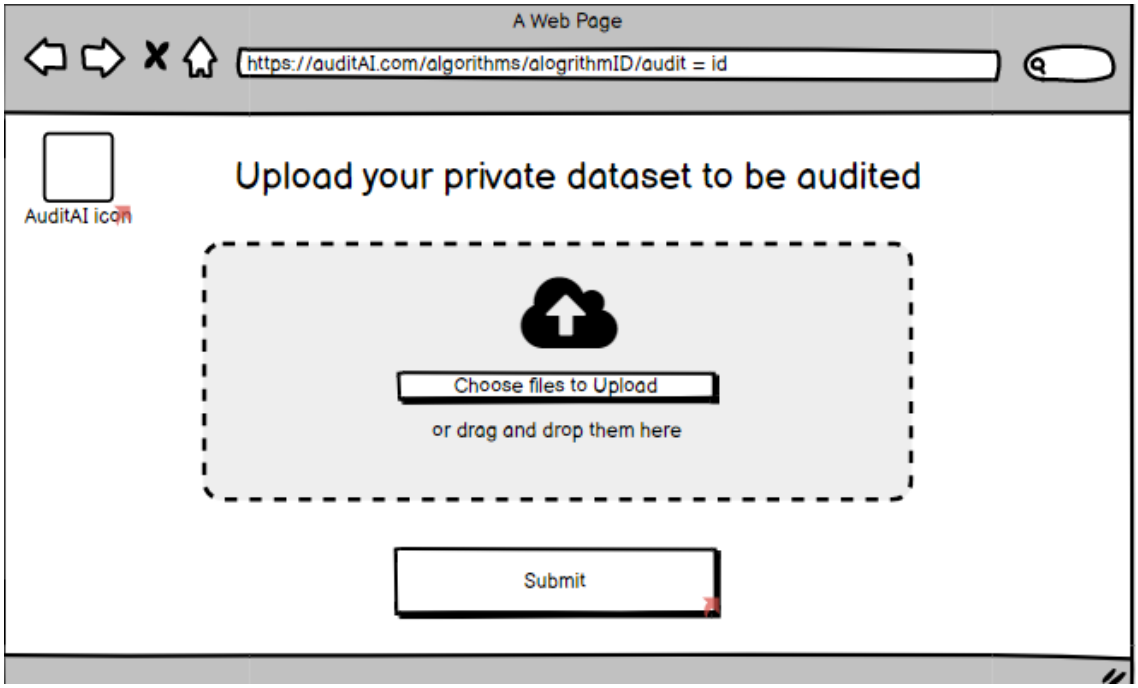

Figure I.5: Screen of the dataset upload page.

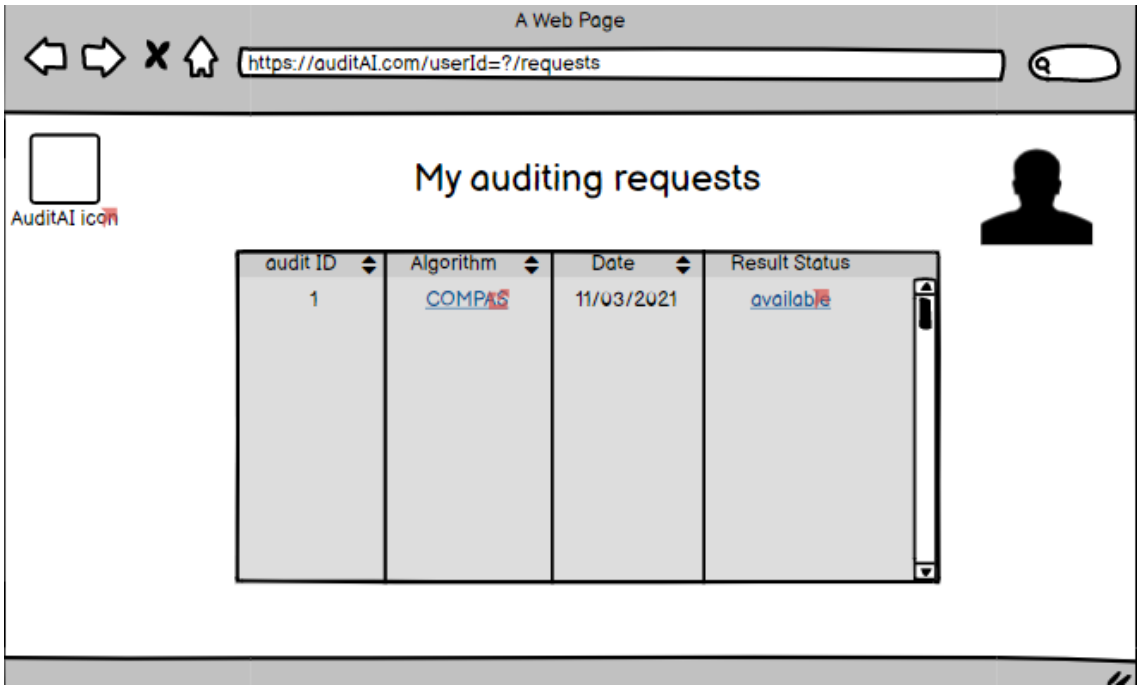

Figure I.6: Screen with the users audit requests.

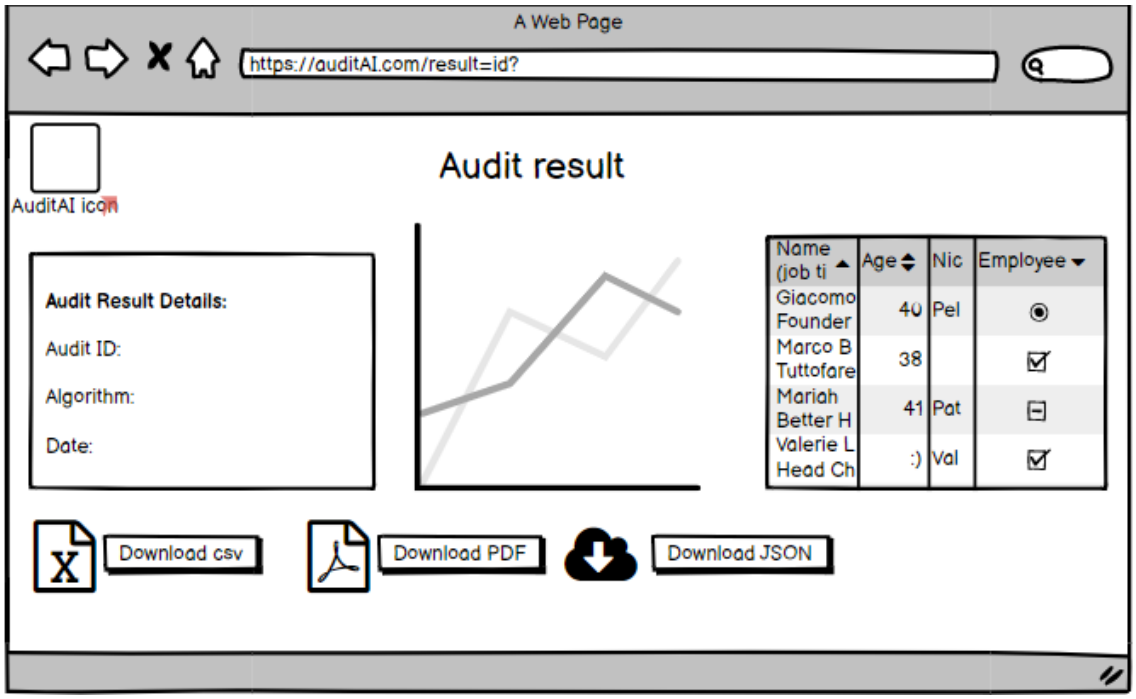

Figure I.7: Screen of the audit request result. Here it is illustrated how the user can download the result of his auditing request. It is also possible to visualize some details regarding the evaluation of the auditing request.