OmniDoc

Development and evaluation of a search and navigation tool for legal documents using ElasticSearch

Henrique Fernandes
Instituto Superior Técnico, Universidade de Lisboa
Lisbon, Portugal
henrique.fernandes@tecnico.ulisboa.pt

***Abstract* – The Portuguese Supreme Court is the most important part of the Portuguese judicial hierarchy. It is therefore essential that it operates efficiently at the highest level in terms of its tasks and the quality of its decisions. In order to improve the judges' workflows, the IRIS project was launched in cooperation with INESC-ID. The objective of this project is to improve the efficiency of the decision-making process and the publication of decisions to other judges of the Portuguese Supreme Court and the general public. This work is part of the first objective.**

To this end, a web-based system was developed using HTML, CSS, JavaScript and ElasticSearch, which allows users to search for and navigate through different decisions from the various existing courts based on search terms and other metrics such as author or date. The developed solution was tested with Portuguese Supreme Court judges, allowing for feedback and suggestions, which were subsequently implemented. In this usability test, the application had a *SUS* score of 80,75.

***Keywords* – Analysis; Portuguese Supreme Court; IRIS Project; Web-based system.**

I. INTRODUCTION

The Portuguese Supreme Court is the most important body in the Portuguese judicial hierarchy. In order to fulfil its duties, it must be at the highest level in terms of efficiency of its tasks and the quality of its decisions. As for the analysis of court proceedings, according to the judges of the Portuguese Supreme Court, some of them prefer to do it on paper and although there are tools to perform this task on the computer, they are very outdated making this process very ineffective. In cooperation with INESC-ID, the IRIS project was launched to improve the efficiency of the decision-making process and the publication of decisions to other judges of the Portuguese Supreme Court and the general public by updating digitisation techniques.

When a case is created, it is forwarded to a First Instance court. In this court, after analysing the evidence presented, a decision is made and a judgement is written. The lawyers involved can then appeal this decision if they disagree with it. If they do, then the case goes to a Second Instance court, the court of appeals. There, the appeal is examined and a new judgement is written stating whether the appeal is upheld or not. However, if the lawyers want to appeal against this new decision, the case goes to the Portuguese Supreme Court. However, only a few cases that meet a number of requirements reach this final stage. The Portuguese Supreme Court's decision is final. In order for the Portuguese Supreme Court's judges to write the new and final judgement, they must access the documents of the previous decisions and search and navigate the existing jurisprudence. In addition, the new decision must be made available to other judges so that it can be used in other cases.

Judges also have access to a number of websites with databases populated with various documents from different Portuguese courts. On these websites, they can search for specific terms or filter the results according to a number of criteria such as the date of the document or its author. The results are also ordered according to their relevance to the search or chronologically. The most used websites are DGSI[1] and ECLI[2]. These two websites offer the most complete searches for the magistrates' requirements. However, they have several flaws that judges would like to see addressed. DGSI is said to be very efficient in displaying results, but it is hard to search and filter the results. ECLI is the complete opposite of DGSI. This website has a very appealing interface and searching for documents is easy, but it takes a very long time to display results, so judges prefer to use other websites.

To achieve the above goal of improving the efficiency of the decision-making process, the IRIS project is divided into three different parts: Anonymisation, Summary and Analysis. This dissertation is part of the latter. In order to make it easier for judges to search and navigate through the numerous documents available, it is necessary to create a tool that resembles a digital library and locates relevant documents in an efficient and fast way, so that the process of decision-making is smoother.

A. Objectives

The aim of this dissertation is to **create an efficient and effective tool to find relevant judgments for judges when deciding on new cases**. To achieve this, a **web technology based tool was designed and implemented**. It provides users with a fast and efficient experience when searching for legal documents by using a powerful search engine tool in ElasticSearch[3] to quickly retrieve results and display a brief summary of each document in an intuitive way in order to help users understand the content and relevance to the search being performed. The user can enter a specific term and the solution will access a database filled with documents from several Portuguese courts and will find the documents in which the term or set of terms occur in several fields. It is also possible to filter the obtained results by specific metrics, e.g.

[1] http://www.dgsi.pt/
[2] https://jurisprudencia.csm.org.pt/
[3] https://www.elastic.co/pt/

author or descriptor. All versions and updates of the digital library development are stored in a GitHub repository[4].

In order to test the system, judges from the Portuguese Supreme Court performed usability tests. These tests allow us to determine whether the tool meets the intended objectives. Ten judges participated in these tests, in which they were asked to solve a series of seven tasks covering most of the functionalities of the digital library. Based on the results collected and analysed, as well as the feedback from the participants of the user tests, the interface is considered to be easy to learn and user-friendly, and is a helpful tool for the judges' work.

B. Document Organization

This document is divided into six chapters and several appendices. These chapters are Chapter 1 - Introduction, which presents the scope of the work and its involvement in the IRIS project as well as its objectives, Chapter 2 - State of the Art, which presents the research of relevant works for the design and development of the solution, Chapter 3 - Approach, which contains the decision regarding the approach for the development of the solution as well as its architecture, Chapter 4 - Implementation consists of the description of the implementation of the solution, Chapter 5 - Evaluation contains the description of the usability tests and the discussion of the results and finally Chapter 6 - Conclusion contains the final thoughts and discussion.

II. STATE OF THE ART

In this chapter, it is presented the research regarding relevant topics to the development of the interface. Since this project will be developed for the Portuguese Supreme Court, it makes sense to research tools and interfaces that allow to display and interact with legal information in an attractive and intuitive manner. The first topic researched was legal text visualization. However, due to the small sample size of interfaces found, it was necessary to widen the scope of the research. For this, text visualizations as well as document collection visualizations were also considered.

Also, since the application resembles a digital library, it is pertinent to research how these are developed and how users perform tasks on them. It is also relevant to understand how the results are displayed to the users and how they can filter such results. This research is focused on digital libraries for justice documents due to the scope of the project.

A. Legal Text Visualization

Since this work involves displaying legal documents, it is crucial that we understand what sort of work was done in this area. Below are some examples of tools that use different visualizations to explore and navigate documents to obtain or extract information from them. In the papers presented there are also described various interaction techniques.

The *Parallel Tag Clouds* [1] combines two different approaches: layout techniques from parallel coordinates and word-sizing. Keywords are organized alphabetically into columns, each representing a distinct topic and font size is used to encode frequency. It shows a bar chart (only one word selected) or a stacked bar chart with different colors (multiple words selected) representing the documents where those words appeared. The height of the bars is used to encode the number of occurrences of the term in each document. It also allows to show the document where the term is located. However, it can become difficult to understand if there are a lot of information.

TileBars [2] collects and displays a part of every document related to the searched terms and provides the frequency of the keywords allowing to analyze the documents in a better way. It is a visualization created to smooth the process of document retrieval based on a query. The user inputs a set of desired terms to be found in a collection and in the window is displayed the tiles representing the documents and the title as well as the initial words of said documents (on the right). Inside the tiles there are squares representing the text segments where those terms were found. The darkness of the square is used to encode the frequency. The downside is the fact that is impossible to analyze the content of the documents.

B. Text Visualization

As it was said before, due to the lack of number of legal text visualizations, it was necessary to expand the research. There are a lot of examples of useful text visualizations to help analyze short texts or long ones, ranging from simple interfaces like the work developed by Byrd [3] or others more complex.

Throughout the Internet, there is a multitude of interfaces and tools that use different visualizations to represent and analyze text [4]. These interfaces and visualizations use various techniques, such as word tags (e.g *Wordle* [5]) that combined with other features help users achieve their goals. These techniques can range from word highlighting to keyword searching, etc. Some visualizations also allow users to interact with the documents by adding annotations or comments to certain parts.

An example of the first feature is a simple tool [3] in which the user can input words and a window is displayed with the document content with the words highlighted. On the right side of the window, dots represent the highlighted keywords. Different words can be input, having different colors assigned to them.

Another tool was created in the context of cybersecurity for authorship analysis. The *AzAA* [6] portal uses crawlers to extract messages from different Web forums and they can be analyzed in two perspectives: author-level and message-level. The first perspective is used to identify which authors use specific stylometric features the most. From the author-perspective, the user can compare the authorship of two people using a radar chart to summarize authorship differences and similarities.

Both these tools are very useful to find keywords in documents. However, the navigation is quite difficult in extensive documents.

A tool was created to analyze text using visual fingerprinting that minimizes the problem mentioned above. *LiteratureVis* [7] allows to load multiple texts to compare them. To create the visualizations, the user can select the measures to be used such as sentence length, Simpsons Index,

[4]`https://github.com/HenrySmash/SAMA-IRIS`

parts of speech, etc. The color of the pixels is used to encode the information received from the algorithms. It is easy to analyze the information related to the text. However, it does not provide access to the text itself.

C. Digital Libraries for Justice Documents

There are various different digital libraries that are used by magistrates to search for decisions on a daily basis. In order to develop the interface, it is necessary to analyze the advantages and disadvantages of these libraries so that we can adapt and incorporate them in the newly created one. As it was said before, this project will be developed for the Portuguese Supreme Court, so it is required to understand how the magistrates use the existing libraries and what sort of tasks they perform on them.

ECLI is one of the most used digital libraries to search for decisions on par with DGSI. This website allows the user to search for terms on the Portuguese jurisprudence and sorts the results according to its relevance or chronologically as well as filter the results by the different Portuguese courts as well as by date by choosing a range of a time period..

DGSI is, according to a number of magistrates, the most used digital library for decision searching. Similarly to the ECLI website, it allows the user to search for terms from different courts. However, it does not have a filter system implemented, it only allows to search for terms, descriptors or field.

Another website to find Portuguese jurisprudence that is not as used is the Diário da República Eletrónico. This website allows to quickly search terms in the search box located in the front page or specify certain fields in the advanced search. This website is mainly used as as a source of information for Portuguese legislation. For decision searching, magistrates admitted to use the ECLI and DGSI websites.

III. DESIGNING THE SOLUTION

A. Requirements

After analysing the state of the art in different areas, a number of requirements can be identified that the interface should fulfil. It is also important to consider the scope of the IRIS project and its objectives. Several meetings were held with different judges of the Portuguese Supreme Court to draw up a list of requirements that the developed solution should meet. In these meetings, the judges described the different steps of a case until it reaches the Portuguese Supreme Court. A case contains all the decisions taken by the courts of first and second instance, so a single case can include documents adding up to thousands of pages. To write a new decision, judges also use documents from other cases that have reached the Portuguese Supreme Court. The judges showed the current applications used to search and analyse the different documents of each case. The judges explained that their work is currently very difficult because the applications used are very slow and outdated. The system must allow judges to search for different documents in a way that:

- **allows access to documents from other Portuguese courts** so that judges have access to decisions from similar cases;
- **displays results efficiently and quickly**, which is important to achieve the goal of this project, namely to improve the efficiency of judges' workflow;
- **it is easy to understand and use**;
- **it allows filtering the results by a number of metrics such as court or author** to get more precise results and find the right document faster;
- **allows quick access to the content of a document** to reduce the time spent searching for a specific term;
- **does not fill the screen with unnecessary information**, as the user interface must be readable and easy to use;
- **allows search results to be shared with different people**, which helps judges to share documents with each other;

B. Design Approach

A low-fidelity prototype was designed that incorporated some key features that the judges considered important. Taking inspiration from the ECLI website and the various existing search engines, an interface was designed to allow users to search for one or more specific terms, with results displayed in order of relevance to the search or by other metrics such as the date of the documents. Users are provided with a search box in which they can enter words, much like a search engine. The application then returns several documents divided by pages. Each page contains a maximum number of results.

Each result displays information about each document, such as the author, date and court. This gives the user a brief description of the document before reading its contents. It also displays the title of the document and its relevance to the search. The title name is the ECLI link and opens a separate page to the document itself so that the user can read the full text. In order to analyse the text corpus without reading it in its entirety, it was decided to create a visualisation based on the related work, with a horizontal bar representing the content of the document and several vertical bars representing the number of occurrences of the searched words

Another feature incorporated into the prototype was the ability to hover over the vertical bars and see an excerpt of the text in which the word is located. This would give the user an idea of the content of the document and the context of the word in the text without the user having to read the whole corpus. The words searched for would be highlighted in the part of the text with the same colour that was assigned at the beginning. This way, the user could quickly find the words they were looking for and read the relevant paragraph. It would also be possible to press a button in the right-hand corner of each document to search for similar documents. This would provide a new set of results with documents that mention the searched words in several fields and are thematically related to the selected document.

C. Improvements

Several changes were made to improve the visual aspect of the user interface of the solution. The layout of each result was changed to make it clearer, and a document summary field was added. This would help the user to further analyse the content of the document without having to read the whole corpus. The title of the user interface and the button to find

similar documents were changed to a blue colour to be more aesthetically pleasing. The colour of the vertical bars in the visualisation was also changed to a more subtle colour scheme so that they do not distract the user from the rest of the information on the screen.

The search box has been centred and a filter button has also been added so that the user can perform a more specific search. The filters include the different courts, authors and time periods. The user can click on the button and a drop-down list of each filter will appear. The options in the list are check boxes and when the user selects them, the results are updated accordingly. The excerpt appears above the summary so that it is easier to hover over a new bar, and it minimises the extent to which the user loses context of the state of the user interface. The words being searched for are highlighted with the appropriate colour instead of being surrounded by a box, and the bar is also highlighted so that the user always knows that the bar is being analysed.

IV. SOLUTION

Following the design of the prototype, a solution was developed to meet the updated objectives of the project. In this section we describe how the interface works and how to use it.

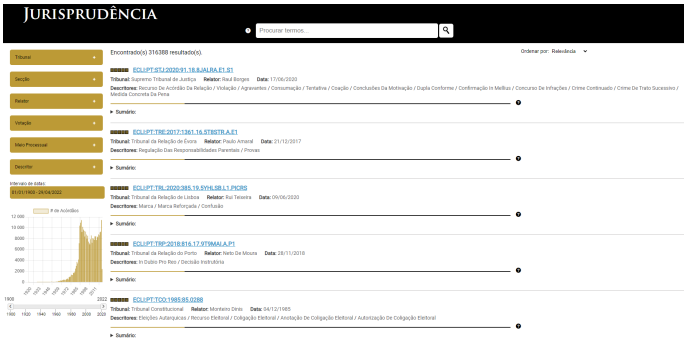

Fig. 1. Solution's initial page.

A. Perform a search

After opening the page, the user can search for terms by entering them in the search box at the top of the page. These terms are matched against the various fields indexed in the database and the results are sent to the client. There are several ways in which the user can perform the search. The first is to simply enter the words. They are then matched individually against the text of the summary and the decision itself. The second option is to search for words between inverted commas and these are matched together as an expression. The third method of searching is to specify the field in which the user wants to search. Several fields are available, e.g. descriptor, court, date, vote, etc. The different search methods can also be used together. The user only has to insert an operator e.g. *AND* or *OR* between the different methods.

B. Results

After the search is performed, the interface is updated with the results retrieved from the server. In addition to the documents that emerged from the search, the total number of documents related to the search is displayed. However, the maximum number of results displayed is 500, divided into 25 pages with 20 documents each because the relevance of the documents after this number was very low and therefore it did not make sense to display more. When the user changes pages, a new request is made to the database with the number of the currently displayed page, so that the next 20 documents are retrieved. The user can sort the documents by relevance and by date, from most recent to least recent by clicking on the drop-down list on the right-hand side of the interface.

1) Metadata: Each result is a different document relating to the words entered. It contains information that allows you to determine which court the document belongs to, who its author is and what descriptors it contains. It also has a summary of the document so that the user can read the topics to which the document refers.

The first line of a result contains an indicator that shows the relevance of the document to the search. Relevance is measured by five bars that are filled with the same colour, but vary in lightness depending on the relevance value. Next to the relevance indicator is the title of the document. It is a link that opens a separate tab in the browser with the full document information.

Below this line are the fields that allow the user to identify the document, e.g. the court, the author, the date of the document and the list of descriptors. If the latter contains words that have been searched for, the background colour is changed to gold. In this way, you can quickly find documents that cover specific topics.

The last section for a result is the summary of the document. This section works like a drop-down list. When the user clicks on it, the box expands to show the summary of the document. If it contains the words searched for, the background colour of these words changes to gold as well.

2) Visualisation: For each result, there is also a visualisation that allows the user to analyse the number of occurrences of the searched words and their distribution in the documents. This visualisation consists of two horizontal bars in different colours. The first gold-coloured bar represents the summary part of the document and the second black line represents the full text of the document.

After the client side has received the documents returned by the server, the search is filtered so that stop words are not shown in the visualisation or in the colour legend. Then, depending on the relative position of the words in the document, vertical bars are superimposed on the horizontal ones. The vertical bars are also coloured differently so that they are easy to recognise. There are five different colours, but it is possible to have even more vertical bars in the visualisation. However, they will then be coloured black. The reason for this is that users do not usually look for long sentences in the text, so it did not make sense to use more than five colours.

Another feature of this visualisation is the possibility to display an excerpt from the text in which the word occurs. The user only needs to move the mouse pointer over a vertical bar and a box with the extract is displayed.

C. Filters

To enable a more specific search, the user can apply filters before and after searching for documents. These filters consist of drop-down lists that correspond to the various existing fields in the database. The filter for the date is divided in three parts: a chart showing the number of documents for a given year, a slider that allows you to select an interval of years, and a date picker that allows you to select a specific interval of days, months or years.

1) Drop-down Filters: The drop-down filters allow you to select multiple values from different fields, e.g. court, section of the court, author, voting decision, procedural means and descriptor. When the user selects one of these dropdown lists, it expands to show the different options for filtering the search. Each list contains twenty options corresponding to the number of results per page. At the end of these options is a field with the number of options that were not displayed. Each option contains the number of documents for each value of the filters. When the user selects a specific value, the number of documents for each option is updated with the number of documents available for the new search. All drop-down lists except the court filter also have a search field where the user can enter terms. The list is then updated with the options that match the input value entered.

2) Date filters: The date filters' interface is different from the others. The three filters use different methods, but they update each other and the other filters according to the new results received. The first method is a date picker. The user can select a specific day, month and year for a start and end date.

The next method has two features. It displays the number of documents for a given interval of years as a bar chart and allows the user to request documents for a given year by clicking on a single bar, updating the information on the other filters of the page.

The final method of filtering by date is a range slider. The user can move two handles on opposite sides of the slider and the results will update according to the years selected. The smallest year is 1900 and the largest year is the current year. Above each handle is a tooltip that shows which year is selected.

D. Document's page

On this page, the user can see all the metadata associated with the particular document, as well as the summary and full text. On the left side of the summary and full text is a column with the rest of the metadata, i.e. court, author, etc. Each value of these fields is a link and acts as a filter, as the user can click on it and the page returns to the search interface, where a new search is performed with the specific filter. There is also a link below these fields that takes the user to the document's page on the DGSI website.

E. Architecture

Using the list of requirements, it was possible to design the architecture of the application. The solution is divided into two main components: Front-End and Back-End.

1) Back-End: The back-end is divided into two sections: Server and Database. The database contains all the information that needs to be displayed to the user, while the server is responsible for handling requests from the client and communicating with the database to get the required information and return it to the user. The server uses ElasticSearch's API to execute the queries to the database and receives a JSON object from it, which is sent to the client. ElasticSearch[5] is a powerful search engine that allows users to perform several tasks such as search for documents, aggregations and document count or even log analysis. It works with several programming languages such as Python, JavaScript, PHP, C#, etc. It has an API that allows users to accomplish the tasks mentioned before. To access said API, we need to create an instance of ElasticSearch in the server using a node with the URL where the database is located.

2) Front-End: The front-end processes the events performed by the user (e.g. clicking on a filter or searching for a term) and displays the information received from the server. The communication between the client and the server is done through REST API requests and the interface is built based on the JSON object mentioned above. REST API is used because it is easy to implement and works well with the JSON format.

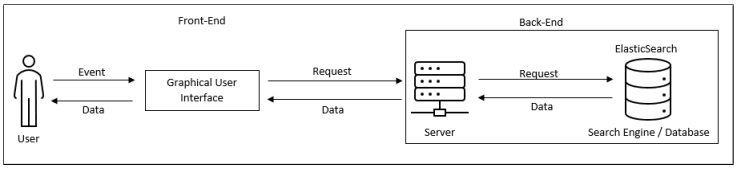

Fig. 2. Digital library's architecture.

F. Implementation

The front-end component of the digital library was implemented using HTML, CSS and JavaScript. In this way, a simple but effective interface could be created. Although a JavaScript framework such as React.js or Vue.js could have been used, I had trouble implementing the filters and document count for each option, because when an option was selected, the remaining ones with zero documents were hidden. We did not want this to happen, since the goal of the filters was to always display their options even if their document count was zero to prevent the user from losing context of the existing filters. The backend component was developed using Node.js[6] and the Express.js[7] Framework. This framework is suitable for simple solutions and is very powerful. It provides the necessary tools to efficiently process requests and return the required information, facilitating the communication between the server and the database. EJS[8] was also used to display the page of documents. This tool allows us to have a predefined structure of an HTML page (called a template) and fill it with the required information when a document is found. This was helpful to create the page template and serve it directly from the server without overloading the frontend. We chose EJS because it is a simple tool that is easy to learn and use. There

[5] https://www.elastic.co/pt/
[6] https://nodejs.org/en/
[7] https://expressjs.com/
[8] https://ejs.co/

was another option, *Pug*[9], but it has a steeper learning curve. A database was used to index all documents retrieved from the DGSI website. It was implemented using ElasticSearch as it provides powerful and efficient search tools that can be useful in the context of this work, such as its API used to make calls to the database and get the desired results returned to the frontend.

1) Back-End: During the development and testing phase, the server was hosted at INESC-ID. This is a temporary solution to share the progress of the development between the group members while the digital library is not yet fully developed. The goal is to host the application on the Portuguese Supreme Court servers so that it is only available to the judges.

To better organise the project, the back-end was split into two components: `server.js` and `search.js`. The `server.js` component receives the requests from the client and forwards them to the `search.js` component. In order to make the queries to the ElasticSearch database, several parameters were required, such as the search term or the filters to apply. The server analyses the parameters sent in the client's request by checking whether they are present or not, and then passes them on to the API as a JSON object. This process is used as well to check the current page of results.

The only type of request used is the GET request, so it is only possible to retrieve information and not add it to the database. When you made a request to the server, sometimes the results came back empty. This was because the server sent the results before ElasticSearch had finished retrieving the documents. To solve this problem, the queries were made asynchronously. This means that the system waits until the data has been retrieved from the database before sending it. This improves efficiency and ensures that the results are always ready to be sent to the client.

As mentioned earlier, ElasticSearch was used to create the database. ElasticSearch is a distributed search and analytics engine built on Apache Lucene[10]. It accepts JSON queries and returns the results in JSON format as well. We chose this tool because it is fast, scalable and provides the necessary tools to efficiently search for documents and analyse their information. This system contains all the documents retrieved from the DGSI website and indexes multiple fields for each document.

At the beginning of the development phase, the documents were divided into several indexes representing the different Portuguese courts. However, this was difficult to handle when several courts were selected. Therefore, in order to facilitate the search for these documents, they were combined into a single index.

After we had set up the database, we had to figure out how queries were made to it. As mentioned earlier, ElasticSearch receives queries in the form of a JSON object. Within this object there are several fields that indicate where the search needs to take place and what to search for. We needed to ensure that the search would be performed, but at the same time guarantee that the filters would be applied when selected. The only solution we found was the boolean query. In this case, the field `must` is used to ensure that the searched terms occur in the document. It also ensures that when searching for two or more terms, they all occur in the same document. This field is the equivalent of the operator "*AND*".

In order to search for the terms entered, we had to specify the fields in which to search for them and ensure that they all occur in the document if possible. Initially, the "`multi_match`"[11] query was used. However, since no search operators could be specified in this query, the search was performed with the "`query_string`"[12]. This query provides a field called "`default_operator`" where you can specify the default search operator that the query uses to find the words. In this case, the operator "*AND*" has been defined.

After searching for the terms, we need a way to refine the search. In earlier stages of development, the field "*filter*" was not used. Instead, we tried to do multiple "`match`" queries within the main query. However, since this method did not give good results from the database and returned errors most of the time, the field "*filter*" was used within the bool query. This field accepts two types of filters: the "*range*" and the "*terms*". The first is used to filter the results by a specific date interval and the second is used for the other type of filters. This field allows you to search for multiple terms within a specific database field.

Ordering the results, either by relevance or by date, is an important issue in the drafting of decisions according to Portuguese Supreme Court judges. It is important to check the date of the document to understand if it is really relevant to the case being analysed. ElasticSearch offers only one option to sort the results, namely the `sort` field. In the first phase of development, there was no option to sort the documents, as the use of this ElasticSearch feature prevented the score of the documents from being sent to the client. However, after researching the ElasticSearch documentation, the flag `track_scores` was discovered which, when set to true, allows the score to be sent.

Since the database retrieves hundreds or even thousands of documents from a single search, it does not make sense to display them all on a single page as it would be too long and take a long time to load. There was the possibility of setting up an infinite scrolling feature, similar to the social media applications. However, this would not work in this case as it would be extremely slow to make multiple queries and the front end would be overwhelmed in creating the interface, so it was important to create the pagination feature. Pagination means dividing the data into pages. In this case, we divide the 500 results into 25 pages. To do the pagination, the server requests 20 documents per request because the performance is limited and it is easier to display on the screen. This is done using the *from* and *size* fields. The first allows the selection of the index number from which the database should start searching and the second specifies the number of documents to be sent to the client. When the user changes the page by clicking on the buttons at the bottom of the page, a new request is made to the database with the new number of the current

[9] `https://pugjs.org/api/getting-started.html`
[10] `https://lucene.apache.org/`
[11] `https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-multi-match-query.html`
[12] `https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-query-string-query.html`

page.

The number of documents is also an important topic to display when performing a search and using the filters, as it provides context to the user. This is where ElasticSearch's aggregations come in. They consist of several groups of values based on the different fields and related to the search performed, with a document count associated with each value. Originally, the number of values for each aggregated group was 1000, but due to performance limitations when entering terms in the filters' search field, the number had to be reduced to 500. When a new query is made to the database, the aggregations are calculated according to the terms and filters entered.

2) Front-End: As mentioned above, the digital library was implemented using HTML, CSS and JavaScript. After a request is made to the server, the frontend creates all the elements displayed on the page. All elements are created using native JavaScript, except for the time filters. Since there was no easy way to create these filters natively, the jQuery UI[13] library was used. This library provides the necessary tools to create the filters and process the various events as their state changes. The bar chart was created using the *Chart.js*[14] library. This allowed to quickly create a chart with the information available. The requests to the server are made through the axios[15] API. This API was implemented because it has proven to be easy to use and provides backward compatibility for browsers. Requests to the server are always asynchronous to ensure that results are retrieved before they are displayed to the user on the screen. To achieve this, JavaScript's *async* and *await* functions were used.

The interface needed to be visually appealing similar to the ECLI website. For this purpose, the color scheme and fonts of this website were initially used. However, the group members of the IRIS project felt that since the interface will be used by judges of the Portuguese Supreme Court, it should be similar in color and font to the Portuguese Supreme Court website. Therefore, the fonts TrajanPro[16] is used for titles and Lato[17] is used for the rest of the page's body. For the colors, gold and black are used. This ensures that consistency is maintained between the websites.

Displaying the relevance of a document to the search is considered important because it can be a deciding factor in navigating the results list. Originally, relevance was displayed as a percentage. However, after a discussion with the group members of INESC-ID, it was agreed that the different percentages do not make a difference when selecting a document. The relevance of each document consists of five $<div>$ tags, which are painted according to the value of the score returned by the database. In the first iteration of this approach, the colour scale used was red (least relevant), orange, yellow, dark green and green (most relevant). However, as there were problems with distinguishing the colours and several people from the INESC-ID group, who are colour blind, stated that it was difficult to distinguish the colours. So the colour was changed to the same golden colour used throughout the user interface and the lightness of the colour represents the relevance of the document. The higher the lightness, the higher the relevance.

The choice of colour scheme for the visualisation was also difficult, as there were not many colours that together made the visualisation easy to interpret. Originally, there was only one horizontal bar in black, representing the summary and full text of the document. This was changed after group members from INESC-ID explained that it was better to split the bars in order to distinguish where the words were. So the horizontal bar was divided into two and the bar representing the summary was given a golden colour.

When performing a search, the visualisation may contain a large number of vertical bars. To calculate their position, the relative position of the word in the text is used. This is not the most efficient way to create a visualisation. However, this was the best option found.

Each time a word occurs, the excerpt of the text in which it occurs must also be displayed. Originally, the excerpt was only displayed when a vertical bar was clicked, but this method was not very intuitive as there was no indication on the user interface that the bars could be clicked. Therefore, the approach was changed so that the excerpt is displayed when the mouse is hovered over a bar. This is a more intuitive way of displaying the text because the first thing users do is hover the mouse over elements, as found in the usability tests.

The filters are one of the most important parts of the user interface and it is important that they are visible and easily accessible to users. Originally, the filters were arranged in a drop-down list next to the search box, which displayed their options when the user hovered over it. However, this made users lose track of what was already selected and forced them to hover over the list again to review it. With the current approach, each filter is an expandable $<div>$ that displays the options when the user clicks on it. This way, the filter can always be open so that the user can see which options have been selected. Each filter consists of a list of twenty options to keep the consistency between the number of results per page. This number of options also helps to keep the user interface clear and easy to understand. Below these options is a field that shows the number of options not displayed, so that the user knows how many have been retrieved from the server.

In addition to these twenty options, each filter also has an input field, as mentioned earlier. This was done to avoid having too much information on the screen and to make it easier to access the different values of the filters. When a user starts typing in the field, the values of the filters that match the input appear. The first iteration of these feature consisted of performing queries to the server using the ElasticSearch search API with a *wildcard* for each character typed by the user, but this was quickly discarded due to performance issues. The second phase was to create an HTML $<datalist>$. After the first twenty values were added to the filters, the rest were added to this list, which was displayed with the options that matched the input when the user started typing. However, when the user started entering terms, the context of the filters was lost and they took up a lot of space on the screen.

[13] `https://jqueryui.com/`
[14] `https://www.chartjs.org/`
[15] `https://axios-http.com/docs/intro`
[16] `https://fontsgeek.com/fonts/Trajan-Pro-Regular`
[17] `https://fonts.google.com/specimen/Lato`

Therefore, the last stage was implemented.

The document page contains all the information about a single document, therefore a lot of information is displayed on the screen. The layout of this page was also based on the ECLI website. The data is presented in an attractive way and allows the user to quickly find information about the document. In the area next to the summary and the full text, the user can perform another search by clicking on the link of the respective filter.

Judges also found it useful to have an option to find similar documents to those found with a search, so that they have access to documents that are related to each other or to a particular topic. Originally this was done using the *more_like_this* function of ElasticSearch. If a term occurs frequently in a large number of documents, it is considered similar by the function. However, this function did not give good results and no solution to this problem was found, so the feature had to be removed from the digital library.

3) Responsive Design: To ensure that the user interface is suitable for different devices such as computer, tablet, mobile phone, etc., the *<meta>* tag was added along with the *viewport* method. This helps in adapting the information to the size of the screen. Also, the *@media* rule for multiple screen widths has been added in the CSS file. This allows you to show or hide multiple elements on the user interface depending on the size of the screen.

V. EVALUATION

To evaluate the application, meetings with several judges were held in the Portuguese Supreme Court building. The aim of these tests was to find out whether the digital library meets the requirements and to collect the judges' opinions about the user interface and the general experience. These tests were also useful for finding bugs to fix at a later date, and for getting feedback and suggestions on specific features that the judges might like to see implemented. The tests were conducted using the most recent version of the project. After the tests were completed, some changes were made according to user feedback.

A. User Evaluation

These tests were conducted with ten different judges of the Portuguese Supreme Court. The judges were asked to perform a series of seven tasks and the time it took them to complete the task and whether the answer was correct or not were collected. These tasks were to cover all the functionalities of the digital library user interface.

At the beginning of each test session, there was a brief explanation of the digital library and how it was integrated into the scope of the IRIS project. After that, the judges had five minutes to use the functions of the digital library and familiarise themselves with how it worked. The list of tasks is given below.

- **Task 1** - Indicate the date of the oldest document that refers to the word "Contract".
- **Task 2** - For the first document in which the word "Debt" appears, indicate in which section of the court the decision was made.
- **Task 3** - Identify the author who has the most documents in which the word "Forfeiture" occurs.
- **Task 4** - For the word "Contract" and documents of the Supreme Administrative Court between January and February 2015, indicate the last descriptor of the second document.
- **Task 5** - For the word "Debt" in the second occurrence of the word of the first document, indicate the value of the debt in euros.
- **Task 6** - How many documents refer to the word "Guilt" for the year 2021?
- **Task 7** - Indicate to which court the most recent document referring to the word "Pawn" belongs.

Each task is independent, which means that the result of one task does not depend on the result of the previous task. For each individual test session, the order of the tasks was randomised and each task started on the home page of the user interface because we did not want all users to perform the tasks in the same order since this could alter the results (learning effect[18]).

As mentioned earlier, for each task, the time taken to complete it and whether the answer was correct or not was recorded. Each task had a time limit of five minutes. At the end of each task, users were asked to fill in a Single Ease Question (SEQ) form. This form asked how difficult the task was on a scale of 1 (Very Difficult) to 5 (Very Easy). They were also asked to give feedback/suggestions on the particular task. After all tasks were completed, the judges were asked to fill out a System Usability Scale form (SUS).

B. Results

The test sessions were conducted with ten assistant judges of the Portuguese Supreme Court. Of the ten participants, one was a man and nine were women. All the judges use their computers on a daily basis and are moderately comfortable with technology. From the results collected, only two tasks were answered correctly in each test session.

1) Time to Complete Tasks and Judges' Answers: Although some tasks are easy to perform, some judges may find it difficult to perform others. This is because judges are not yet familiar with the user interface and some of its functionalities. As you can see from the boxplot below, tasks number one, four and six are the most time-consuming to complete. The first two are considered the most challenging tasks, so it is normal that they take the longest to complete. There is also the number of correct and incorrect answers given for each task. As you can see, task four has a high number of wrong answers, while easier tasks like two and seven were answered correctly by all judges in a short amount of time. Task number five also has a high number of wrong answers. However, judges were quicker to indicate the desired answer in this task than in others.

[18] https://sportscienceinsider.com/what-is-the-learning-effect/

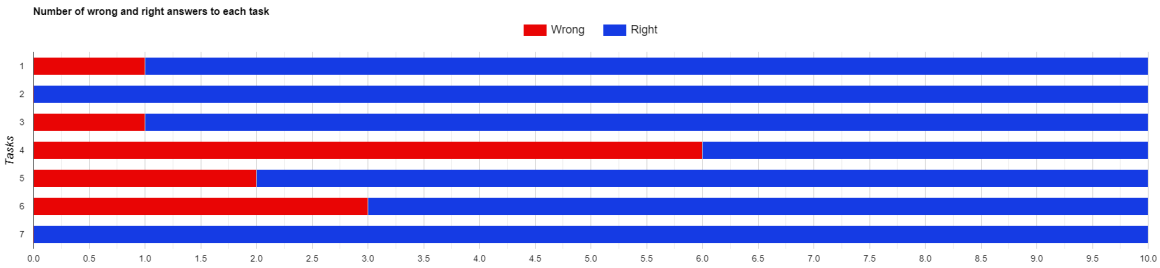

Fig. 3. Number of right and wrong answers for each task.

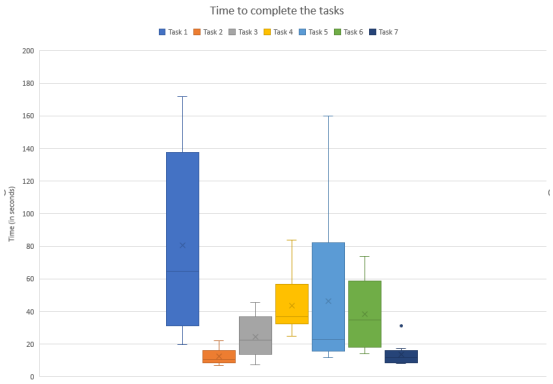

Fig. 4. Time to complete the tasks.

2) User Comments: As mentioned earlier, judges were asked to fill out a *SEQ* form indicating how difficult a task was to complete on a scale of one (Very Hard) to five (Very Easy). This helped to understand the judges' thoughts about the interface.

Judges found the first task the most difficult to solve. Tasks five and six were also considered the most difficult. For the first task, judges suggested that the tooltips for the years should not overlap, as this makes it difficult to read the selected numbers. For tasks four and five, the judges suggested implementing a tooltip explaining the different ways of searching and how the visualisation works. This would help in understanding how the digital library works and help new users get used to the interface quickly. One judge also suggested that excerpts should be clickable to jump to where the excerpt is on the document page. For task six, some judges suggested adding a filter to sort the documents from older to newer.

Some tasks were quick to complete and scored highly in the *SEQ* questionnaire as the judges had time to use the interface before the tests began. For example, tasks two and seven had a low completion time due to this. After completing the tests, judges were asked to fill in a *SUS* form in which they rated how they felt about using the interface and the overall experience. The *SUS* score of the interface was 80.75, so the application is considered *excellent*, as any system with a rating above 80.3 is considered excellent[19].

In the *SUS* questionnaire, low ratings are expected for the odd questions, while high ratings are expected for the even questions. This is the case with this system, except for the last question, where there were a number of high-scoring answers. This means that the judges felt that they needed a function that was not available in the digital library to do a particular task. The scores in the remaining answers are expected for each question.

C. Changes to the Interface

After the usability tests were completed, the judges gave their feedback on how the digital library's user interface and overall experience could be improved. Some features were implemented after the tests, but there are more that can be developed in future versions of the system.

The biggest complaint from the judges was the lack of an option to sort documents from oldest to newest. Since there was an option to sort from newest to oldest, it made sense to have the other option as well. Therefore, an option was added to the drop-down list on the right side of the user interface.

Another feature strongly requested by almost all judges were two tooltips explaining how to perform a search and how to use visualisation. Two icons, taken from the *Font Awesome* website[20], were placed next to the search box and each horizontal bar of the visualisation. Hovering the mouse pointer over these icons displays the tooltips.

The tooltip explaining how to perform the search consists of four bullet points describing the different searches (normal search, expression search and super search) and how to use the operators *AND* or *OR* . The second tooltip also consists of a list of bullet points and it describes what the horizontal bars represent (the golden bar is the summary of the document and the black bar is the full text) as well as the vertical bars (occurrences of the searched term or terms). It is also explained to the user that it is possible to move the mouse pointer over the vertical bars to display an excerpt.

Although it was not necessary, the judges used pagination during the usability tests. They stated that they wished there was a way to jump to the beginning and end of the results. Therefore, these two new options were added. When a user goes to the beginning or the end, a new request is made to the server with the current page of results.

A new index has been created in the database with new fields. These new fields provide more information to the user and are therefore more helpful to judges when searching for a document. All these new fields are displayed on the documents page if they are available for the document in question.

After analysing the information gathered during the usability tests, it can be confirmed that the judges found the experience of the digital library user interface pleasant. The

[19] `https://shorturl.at/beloG`

[20] `https://fontawesome.com/icons`

system is well integrated, does not have many inconsistencies and for the most part correctly fulfilled the objectives of the tasks. Although the judges found some tasks more difficult than originally thought, they will get used to the features over time as they use the system more frequently.

When asked if the functions are well integrated, the judges responded with an average score of 4,2 in the *SUS* questionnaire. In addition, the judges felt that the system is not unnecessarily complex (average score 1,5), that it is easy to use (average score 4,2) and that they would quickly learn to use it (average score 4,5). However, some judges felt that they would need the help of experts (average score of 3,8) and that they would need to learn a lot before they could use the system (average score of 3,7). The rating of the overall *SUS* questionnaire (80,75) also shows that the interface is user-friendly and understandable. Therefore, it can be said that the digital library is able to fulfil the objective of facilitating the process of searching and analysing documents.

VI. CONCLUSIONS

It is important that the Portuguese Supreme Court works as efficiently as possible. To do so, its judges must have access to tools that facilitate their work. In this work, a digital library was designed and developed that allows judges to search and navigate legal documents from all Portuguese courts.

A series of interviews were conducted with several Portuguese Supreme Court judges to find out what tools and systems are currently available to them and how they use them in their daily work, and to obtain a list of requirements.

Once the list of requirements was finalised, designing the interface was the next step in the development process. The architecture of the system was also determined. The application was designed to be fast and efficient and to match the judges' existing tools. To this end, the solution was based on the interface of existing search tools such as the ECLI and DGSI websites, as these were considered by judges to be the best systems available and were the most widely used.

After the implementation phase, the system was tested with ten different judges of the Portuguese Supreme Court. These usability tests made it possible to understand how the judges perceived the user interface and to identify errors and areas for improvement. From the results of these tests, it could be concluded that the interface was easy to use and user-friendly. The judges liked their experience with it and the overall aesthetics of the system. They also gave their feedback and made suggestions on certain features they would like to see implemented. Some changes were made, but due to lack of time, others may be implemented in future versions.

A. System Limitations and Future Work

There are several features that can be added to further improve the system. One feature that should be added is the ability to find similar documents, as the originally implemented approach did not work as expected. One issue that was identified during user testing is the size of the bars of the bar chart. On smaller screens, the bars corresponding to a small number of documents are very small, making it difficult for users to see and click on them.

The first feature mentioned by a judge would be the ability to click on the vertical bars and open the page of the document at the location of the excerpt. This would make it easier to find a particular extract, so you do not waste time searching for it.

There are also features that can be implemented that were not suggested by the judges. They were considered important by the members of the INESC-ID group to make the judges' workflow even easier. At the moment, the tool is available to any user, but as this project is intended for the Portuguese Supreme Court, it is important to ensure the security of the users. Therefore, in the future, it would be important to implement an authentication system to access the tool. Another feature is the possibility to create bookmarks. This allows judges to save several documents that could be related to each other and access them quickly. There is also the option to implement a new filter that allows documents to be selected based on their relevance. If you want to ignore less relevant documents, this would be a good option. To increase efficiency in creating the visualisation, you could also use the ElasticSearch function for term vectors. This allows you to store the offset and position of each word and use it in a query.

REFERENCES

[1] C. Collins, F. Viegas, M. Wattenberg, "Parallel Tag Clouds to explore and analyze faceted text corpora", pp. 91 – 98, 11 2009, doi:10.1109/VAST.2009.5333443.

[2] M. A. Hearst, "TileBars: Visualization of Term Distribution Information in Full Text Information Access", *in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, p. 59–66, ACM Press/Addison-Wesley Publishing Co., USA, 1995, doi:10.1145/223904.223912, URL: `https://doi.org/10.1145/223904.223912`.

[3] D. Byrd, "A Scrollbar-Based Visualization for Document Navigation", *in Proceedings of the Fourth ACM Conference on Digital Libraries*, DL '99, p. 122–129, Association for Computing Machinery, New York, NY, USA, 1999, doi:10.1145/313238.313283, URL: `https://doi.org/10.1145/313238.313283`.

[4] K. Kucher, A. Kerren, "Text visualization techniques: Taxonomy, visual survey, and community insights", *in 2015 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 117–121, 2015, doi: 10.1109/PACIFICVIS.2015.7156366.

[5] F. B. Viegas, M. Wattenberg, J. Feinberg, "Participatory Visualization with Wordle", *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1137–1144, 2009, doi:10.1109/TVCG.2009.171.

[6] V. Benjamin, W. Chung, A. Abbasi, J. Chuang, C. A. Larson, H. Chen, "Evaluating text visualization for authorship analysis", *Security Informatics*, vol. 3, no. 1, p. 10, Sep 2014, doi:10.1186/s13388-014-0010-8, URL: `https://doi.org/10.1186/s13388-014-0010-8`.

[7] D. A. Keim, D. Oelke, "Literature Fingerprinting: A New Method for Visual Literary Analysis", *in 2007 IEEE Symposium on Visual Analytics Science and Technology*, pp. 115–122, 2007, doi:10.1109/VAST.2007.4389004.