**République Algérienne Démocratique et Populaire**
**Université Abou Bakr Belkaid– Tlemcen**
**Faculté des Sciences**
**Département d'Informatique**

**Mémoire de fin d'études**

**pour l'obtention du diplôme de Master en Informatique**

***Option:*** *Génie Logiciel (G.L)*

# Thème

# Développement d'une API éducatif pour partage de cas cliniques

**Réalisé par :**

- **Redouane LESHAF**
- **Abdelghani MAHAMMEDI**

*Présenté le 03 Juillet 2022 devant le jury composé de MM.*

- Mme. Asma BENMANSOUR *(Président)*
- Mr. Ilyas BAMBRIK *(Encadreur)*
- Mme. Asma BENMANSOUR *(Examinatrice)*
- Mme. Souad KHITRI *(Examinatrice)*

Année universitaire: 2021-2022

# Dedication and thanks

First, we are thankful to Allah the almighty for giving us the health, the will and the courage to accomplish this work.

We would like to express our sincere gratitude and thanks to our professor and supervisor: Dr. Ilyas BAMBRIK for their advice, orientations and help along the way of elaborating this work.

Our gratitude also goes to Mrs. Asma BENMANSOUR and Mrs. Souad KHITRI for accepting to review and examine our work.

We extend our thanks to all our teachers at computer science department for their efforts.

Of course, we do not forget to especially express our gratitude to our parents and thank them for their patience, help and support whether it was material or moral.

Finally, thanks to everyone who supported us from near or far, even morally.

## Table of Contents

**List of Figures**

**List of tables**

**Abbreviations**

| | |
|---|---|
| API | Application Programming Interface |
| AKT | Applied Knowledge Test |
| AWS | Amazon Web Services |
| CBOW | Continuous Bag-Of-Words |
| CK | Clinical Knowledge |
| CME | Continuing Medical Education |
| CSS | Cascading Style Sheets |
| GCP | Google Cloud Platform |
| GloVe | Global Vectors |
| HAML | HTML Abstraction Markup Language |
| HTML | HyperText Markup Language |
| IDF | Inverse Document Frequency |
| MCQ | Multiple Choice Question |
| MSD | Merck Sharp & Dohme |
| NBME | National Board of Medical Examiners |
| NPM | Node Package Manager |
| ODM | Object Data Modelling |
| SASS | Syntactically Awesome Style Sheets |
| SBA | Single Best Answer |
| SVM | Support Vector Machine |
| SoaML | Service-oriented-architecture Modelling Language |

| | |
|---|---|
| SysML | System Modelling Language |
| TF | Term Frequency |
| UCAT | University Clinical Aptitude Test |
| UML | Unified Modelling Language |
| USMLE | United States Medical Licensing Examination |
| VCS | Version Control System |

**Introduction**

This theme belongs to the context of research and development for the field of university students learning/studies.

Medical students often, if not always, seek to learn through the diagnosis of clinical cases because it is the best way to explore the acquired knowledge in a semi-applied way. However, the only current way is to go to the hospital, read books and handouts or use e-learning platforms. This will make the learning process harder and unorganized, especially with the big number of sources, since students will search for learning resources in different ways (books, e-learning platforms, etc). Thus, a place that regroups the majority of the content that can be found in these resources is required. The objective of this work is to develop an API [1] allowing in the first place the easy sharing of clinical cases. In addition, this API can be extended to allow searching by keywords or by semantic similarity in order to have more relevant results. The main purpose of the proposed API is that it enables the user to find semantically similar clinical cases to a provided query string. And thus, it helps the student search and locate relevant information quickly and easily.

**Dissertation organization**

- In chapter 1, we present some state-of-the-art mobile applications, websites and platforms available on the internet used by medical students for studying. Each application is described concisely. To conclude the chapter, a comparison between these applications' features is made.
- Chapter 2 summarises what automated document classification and document retrieval are. Also, we highlight the most popular machine learning methods related to Information Retrieval. Particularly, we discuss word2vec which is the Deep Learning algorithm that we use in order to obtain a semantic representation of the clinical cases.
- In chapter 3, we will showcase our contribution to this work. First, we give the conception of the API and an example client application that consumes our API. Thereafter, we present the tools and technologies that were used during the development. And at the end of the chapter, we explain how to use this API (API calls, etc).

To conclude this work, a general conclusion and some perspectives are given for future optimizations.

# Chapter I: State of the art

# 1 Introduction

In this chapter, we will summarize a non-exhaustive list of the different available applications used by medical students, starting with medical mobile applications and after that medical websites then medical platforms and provide a comparative study between their functionalities.

# 2 Mobile applications

### 2.1. Lecturio Medical Education

Lecturio Medical Education is an application that helps medical students and medical practitioners to study in an effective way but also to prepare for their board exams such as NBME, USMLE 1 & 2 CK, etc [2] [3].

The application provides many ways to learn, such as quizzes, video lectures that can be later added to a custom playlist but also an integrated medical concept library which summarizes different medical topics. Furthermore, the user can study on his own pace by downloading slides on his smart phone and reading them in offline mode. This application covers the medical part of the Lecturio platform which discussed under Medical platforms section.

### 2.2. Prognosis : Your Diagnosis

Prognosis is designed to improve the diagnostic skills and decision making by solving clinical case studies where each case is followed by a comprehensive discussion of the diagnostic reasoning, as well as key learning takeaways [4] [5]. The interactive clinical cases combined with a game-like design make learning new skills in short time easy.

### 2.3. AB-QCM (Résidanat)

AB-QCM is an Android application that helps medical students prepare for their externship and residency medical exams using MCQs. The student can have a general overview of his progress [6].

The MCQs as well as the clinical cases are classified by subject and course which allows the student to be more organized in his study. To get access to the whole content (more than 50,000 MCQ and more than 20,000 clinical cases) included in this application, an annual subscription is required.

# 3 Medical websites

## 3.1. ESPACE DES ETUDIANTS EN MEDECINE

The main objective of the site is to create a space for exchanging and sharing knowledge in the medical field [7]. Medical students are allowed to share university lectures, previous exams, practical courses, links to other medical applications or websites and any educational materials. Some courses are only dedicated to study clinical cases.

## 3.2. Medscape

Medscape is the leading online global destination for physicians and healthcare professionals worldwide, offering the latest medical news and expert perspectives; essential point-of-care drug and disease information; and relevant professional education and CME [8].

The platform provides a large set of well detailed clinical cases simulations, interactive diagnostic modules and MCQs to help the student improve his diagnostic and decision making skills. A network called Medscape Consult [9] is available for the members to share their knowledge or ask for help about a given case or topic.

## 3.3. MSD Manuals

The MSD Manuals is a medical information source covering thousands of topics in all fields of medicine. They are offered as a free public service to health care professionals and the general public [10].

The professional version [11] dedicated to healthcare professionals and medical students has a rich, well explained content such as 3D models, figures and video lectures but also quizzes, topics discussions, detailed case studies and other useful resources.

## 3.4. PassMedicine

PassMedicine is a website dedicated to both students in the early years and finals of medical studies as well as eligible doctors, to review subjects for their university exams or prepare for high-stakes medical exams such as USMLE, UCAT, AKT and others [12].

With thousands of configurable questions in the form of SBA questions accompanied by notes and explanations, and medical facts represented in the ANKI-style spaced repetitions or in flashcard style, the user can study the way he wants. Also, a histogram based on other candidates' results is available for each user so he can see his ranking and determine if he is ready to take his exam or not.

# 4 Medical platforms

## 4.1. AMBOSS

AMBOSS is a medical platform that provides powerful learning and clinical tools for those pursuing a medical career, whether they are still students or they are already in the field. It aims to empower all doctors to provide the best possible care [13] [14].

Among many tools, an extensive medical library of clinical articles covering all systems. This library has a rich set of functionalities including the high yield mode that helps getting info with the most high-yield in an article. The user can create custom question bank sessions that are based on clinical case studies based on many criteria such as difficulty, discipline and others. Afterwards, he can take these sessions in study mode or in test mode where the latter adds time constraints to simulate a real exam. Additionally, a performance analyzer which exploits the results of the user to give study recommendations is available.

## 4.2. Lecturio

Lecturio is a medical learning platform meant to enable users to achieve the mastery of medical/nursing concepts [15]. For medical students, Lecturio provides two learning paths with pre-configured schedules: board exams for specific exams like USMLE and subject exams that cover general medicine, pathology, pediatrics, microbiology, neuroscience, etc.

No matter if the student enrolled in one of the paths or not, a set of videos with note taking functionality accompanied by quizzes, spaced repetitions, articles from the concept library are available in addition to a question bank of clinical cases with text and video explanations. Among other useful tools we have the interactive 3D anatomy models and the bookmatcher which helps finding relevant video lectures faster.

### 4.3. Osmosis

Osmosis is a rich health education platform for students in the fields of medicine, nursing, dentistry, pharmacy and other healthcare professions but also for those who are not in the medical field, like patients [16]. The platform provides a library containing 2000+ videos covering a wide range of topics, in addition to flashcards powered by spaced repetitions and case based questions including detailed explanations of correct and incorrect answers written and reviewed by medical and health professionals [17].

The user can also build and share their custom playlists, flashcards, questions quizzes and high-yield notes and also assemble a deck of previously built flashcards or even create workspaces. Workspaces are used to store in an organized way quizzes, flashcards, documents uploaded by the user and private notes and it is possible to invite other peers to collaborate with them.

### 4.4. Radiopaedia

Radiopaedia is a rapidly growing open-edit educational radiology resource that has been primarily compiled by radiologists and radiology trainees from across the world. Its mission is to create the best radiology reference, and to make it available for free [18].

The platform focuses mainly on articles and cases but it also offers courses and case quizzes. Except from courses, all other resources can be shared. The user, following certain guidelines and rules, can create his own articles, cases, cases playlists and MCQs then share, edit or delete them. Cases created by the user can be linked to existing articles. Furthermore, since it is an open edit platform, it is possible to edit existing articles. The majority of Radiopaedia courses need to be purchased but individuals in specific low or middle-income regions can apply for free access to these courses.

### 4.5. Medmastery

Medmastery is a video based learning platform with the goal of helping medical professionals improve their clinical skills. In addition to doctors, Medmastery can be used by doctors in training, medical students, nurses, paramedics and other healthcare providers [19].

Plenty of courses are available with a content varying from mini/essentials to masterclass. Each course is divided into chapters and is accompanied by downloadable handbook(s). Each chapter of the course is presented by short videos and concluded by a quiz to test the acquired knowledge of the user. Since the platform uses a case-based approach [20], the user can practice with interactive quizzes and real world case workshops.

The user can also read the proposed articles to build his skills and improve his clinical decision making. Some of these articles are literature reviews while others are tutorials or clinical guides.

Table 1.1: Comparison of features of mobile applications, websites and platforms

| | **Authenticate to access cases** | **Share cases** | **Full free access** |
|---|---|---|---|
| **Lecturio Medical Education** | Yes | No | No |
| **Prognosis : Your Diagnosis** | Yes | No | No |
| **AB-QCM (Résidanat)** | Partially Yes | No | No |
| **ESPACE DES ETUDIANTS EN MEDECINE** | No | Yes | Yes |
| **Medscape** | Yes | No | Yes |
| **MSD Manuals** | No | No | Yes |
| **PassMedicine** | Yes | No | No |
| **AMBOSS** | Yes | No | No |
| **Lecturio** | Yes | No | No |
| **Osmosis** | Yes | No | No |
| **Radiopaedia** | No | Yes | Yes |
| **Medmastery** | Yes | No | No |

As we can see in Table 1.1 above, illustrating the features of the mobile applications, websites and platforms we have mentioned. The majority of these sources

do not allow their users to share new cases. In addition, the users are invited to authenticate to access certain existent cases and to make a paid subscription to get access to the rest of the cases. Of course, a paid subscription is not possible for all students.

## 5 Conclusion

In this chapter, we presented some of the most useful online resources available for medical students. However, these resources are limited for multiple reasons. Particularly, searching for similar clinical cases is either unavailable or minimal. Furthermore, students are usually unable to share valuable clinical cases. In the next chapter, we discuss machine learning algorithms useful for text summary that we will use for our API to enable searching for clinical cases.

# Chapter II: Automated document classification and document retrieval

# 1 Introduction

The automated process of document classification is used to aggregate documents that are similar based on criteria such as date, topic, author, etc. by assigning them a category, also known as label, to make their retrieval easier later. As an example, classification can be accomplished using machine learning algorithms that can be divided into four families: supervised, semi-supervised, unsupervised learning and reinforcement learning. We are interested in supervised and unsupervised learning algorithms and how they can be used to classify documents and measure similarity between them. First, we give a brief explanation of supervised learning and unsupervised learning. Then, we explain some machine learning algorithms.

# 2 Supervised Learning

Supervised learning is an approach of machine learning where the training of the model is done via a labeled dataset (the inputs and the correct outputs are know beforehand) [21]. The training datasets help the algorithm to adjust the parameters of the model, known as weights, by calculating the difference between the output we get from the algorithm and the expected result provided beforehand. Some common algorithms relying on this approach are: neural networks, support vector machines (SVM) and linear regression, etc.

# 3 Unsupervised Learning

In this approach, unlike supervised learning, an unlabelled dataset is passed to the algorithm and the observations are grouped by the algorithm based on the structure of the dataset [22]. Usually, the observations are classified based on some similarity measurement such that observations belonging to the same category are close to each other.

One of the common and popular unsupervised learning algorithms is the K-means algorithm. The clustering process using K-means is explained below:

At the beginning, the number of clusters is set to K with K roughly the number of expected categories in the dataset. Afterwards, if we have N observations, there will be K clusters and K cluster heads. To do so, the following steps are followed:

1. Choose the initial K cluster centers randomly.

2. Assign the observation to the cluster with the nearest cluster head based on the Euclidean distance between the observation and the cluster head.

3. For each cluster, compute its gravity center and assign it as the new cluster head.

4. Repeat step 2 and 3

Steps 2 and 3 are repeated until no observation is assigned to a new cluster. If all observations remain in the same cluster between the iteration x and the iteration x+1 (i.e. the centroids of iteration x and the centroids of iteration x+1 are the same) we consider that the algorithm converged and the cluster heads given by the algorithm are the model that will be used to determine the how to classify new observations. In document retrieval, the cluster heads are subjects in the document dataset.

# 4 Machine learning algorithms

## 4.1. Term Frequency Inverse Document Frequency

TF-IDF is a technique of data modeling that computes a weight for each word which indicates the importance of the word for a given document within a corpus [24].

**Terminology**:

- **t** : term (word)
- **d** : document (set of words)
- **N** : total number of documents in the corpus
- **corpus** : the complete document set

**Term frequency**: local frequency of the term. In other words, the count of the term t in the document d.

- **TF(t,d) =** count of **t** in **d**.

**Inverse Document Frequency**: it measures the rarity of the term in the document set.

- **DF(t) =** count of documents containing **t** in the corpus.

- **IDF(t)** = $log\left(\frac{N}{DF(t)+1}\right)$ : inverse document frequency in the corpus.

By multiplying the **TF(t, d)** by **IDF(t)**, we get the TF-IDF score represented below:

**TF-IDF(t, d)** = $TF(t,d) \times IDF(t) = count\ of\ t\ in\ d \times log\left(\frac{N}{df(t)+1}\right)$

The higher the **TF-IDF(t, d)** score, the more important **t** becomes in **d**. Otherwise, **t** is less important in **d**. Another way of putting this is that a term **t** is important in a document **d** if it is repeated a lot in that document **d** and rarely encountered elsewhere in the corpus. Thus, words that appear with a high frequency in all documents of the corpus are down-weighted. The model produced by **TF-IDF** is each term and its importance in the document represented by a weight:

**[**{t1, tf-idf(t1, d1)}**,** {t2, tf-idf(t2, d1)}**,** …**,** {ti, tf-idf(ti, d1)}**]**

Once the documents are represented by a vector of weights, to measure the similarity between two documents **d1** and **d2** represented by the vectors **v1** and **v2** respectively, we compute the dot product of **v1** and **v2**. Essentially, we get a scalar that shows how similar two documents are. The higher the value of the dot product, the more similar they are. In other words, there are terms in common that are important in both documents.

### 4.2. Word2vec

Word2vec is a Deep Learning algorithm that takes a text corpus as input and gives a vector representation for each word in the text corpus. The produced vector representation is a semantic description of the word such that words that are similar have a cosine similarity close to 1. Similarly, words that are semantically opposite to each other have a cosine similarity close to -1. Compared to other word vector models, such as one-hot-encoding, the word2vec representation relies on the word context and produces a meaningful representation. First, the text corpus must have been pre-processed before passing it as input to the algorithm. The pre-processing involves

lowercasing words, removing punctuation marks such as commas, removing stop words[1], etc.

For example, if we want to measure the similarity between two words **w1** and **w2** represented by 2 vectors **v1** and **v2**, we will calculate the cosine of the angle between the two vectors **v1** and **v2.** The two words **w1** and **w2** are more similar if the cosine of the angle between **v1** and **v2** is higher.

Word2vec can utilise either of two model architectures proposed by Mikolov et al. in [25] to produce a distributed representation of words:

- **Continuous bag-of-words model**: predicts the middle word based on surrounding context words. This architecture is called a bag-of-words model as the order of words in the context is not important [26].
- **Continuous skip-gram model**: Predicts, in the same sentence, words before and after the current word [26].

These two models are illustrated in Figure 2.1.

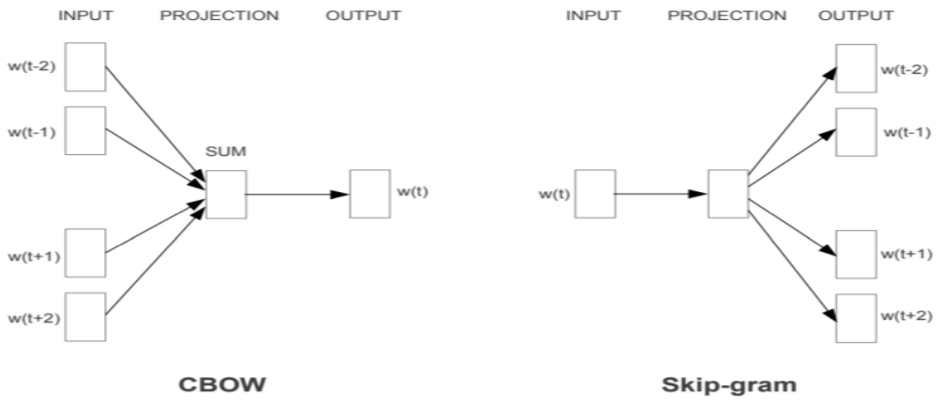


Figure 2.1: Graphical representation of the CBOW model and Skip-gram model

*Note*. From "Exploiting Similarities among Languages for Machine Translation," by MIKOLOV Tomas, LE Quoc V., and SUTSKEVER Ilya., 2013, *arXiv preprint arXiv:1309.4168,* p.3 [27].

[1] Common words that typically appear in any text are called stop words. They tell us little about the data we have, so we remove them. E.g.: coffee or tea is fine for me-> coffee, tea, fine.

### 4.3. GloVe

GloVe (Global Vectors) is a count-based unsupervised learning algorithm. It takes a matrix of word-word co-occurrence as input and uses the non-zero entries to train and give meaningful word representations as output. Each entry represents the co-occurrence counts of a word with another word in a given context/corpus [28].

## 5 Conclusion

In this chapter, we review relevant machine learning algorithms that can be used for text summary. We will particularly use word2vec[2] and clustering[3] for clinical case retrieval in our implementation. In the next chapter, we proceed to document the specific details of our implementation.

[2] Because word2vec gives a semantic representation of the document that can be used to later to retrieve similar documents

[3] Because user queries and cases inserted by the user are not labelled/classified previously

# Chapter III: Contribution

# 1 Introduction

After summarizing the machine learning algorithms relevant to our project, this chapter presents and describes the API we proposed in response to the problematic we are interested in. First, we start with mentioning the requirements. Next, we present the conception. After that, we pass to the implementation. Then, there is the API explanation in addition to some screenshots of an example client application. In the following sections, by doc or document, we generally refer to the clinical case or to its description (according to the context). For example in the expression: “document representation”, document refers to the description of the clinical case.

# 2 Requirements

In this section, we will mention functional and non functional requirements.

## 2.1. Functional requirements

- Sign up.
- Sign in.
- Search for clinical cases.
- Access clinical cases.
- Share new clinical cases.
- Manage clinical cases.
- Manage users (admin only).

## 2.2. Non functional requirements

- Performance.
- Availability.
- Usability.

# 3 Conception

In this section, we will use UML [29] diagrams such as use case diagram to present the most important use cases of the actors interacting with the client application that consumes the API. Moreover, we use sequence diagrams to illustrate the chronology of some use cases and class diagram to give the classes participating in this system and the relations between them.

### 3.1. Use case diagram

As shown in Figure 3.1, there are three actors: **User**, **Authorized user** and **Admin**. where **Authorized user** and **Admin** are actors with more privileges than the simple user represented by the actor **User**.

1. User can:
   - Sign up.
   - Sign in.
   - Search case(s).
   - Access case.
2. In addition to the use cases of User, Authorized user can:
   - Share new case.
   - Manage case (update or delete case).
3. In addition to the use cases of Authorized user, Admin can manage the users accounts (update user account or delete user account).

In the use case diagram, by *case*, we mean *clinical case*. Furthermore, the use case 'update user account' of the actor Admin doesn't mean that the admin can update every setting in the account. It means that it can allow or deny the user to share new cases.

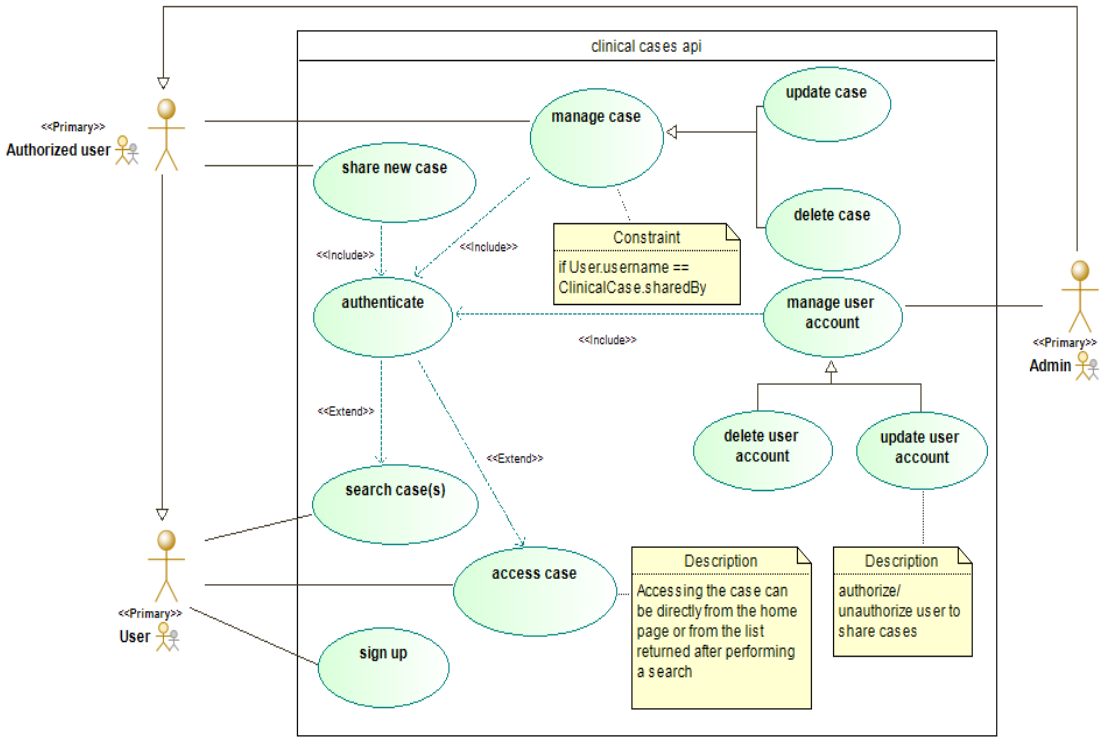


Figure 3.1: Use case diagram for the actors of the system

### 3.2. Sequence diagrams

We are interested in representing the sequences for the major use cases: ***authenticate***, ***search case(s)***, ***access case*** and ***share new case*** respectively when using the client application that consumes the API. As we can see in Figure 3.4 and Figure 3.6, the authentication is optional since the user will not edit any information related to the case.

In Figure 3.2, the user will authenticate using his username and password. If the inputs are invalid, an error message will be displayed. Otherwise, we will try to find the user corresponding to the given inputs. If a user is found then we will return their information and redirect them to the index view ("/" route). Else, an error message will be returned and shown to the user.

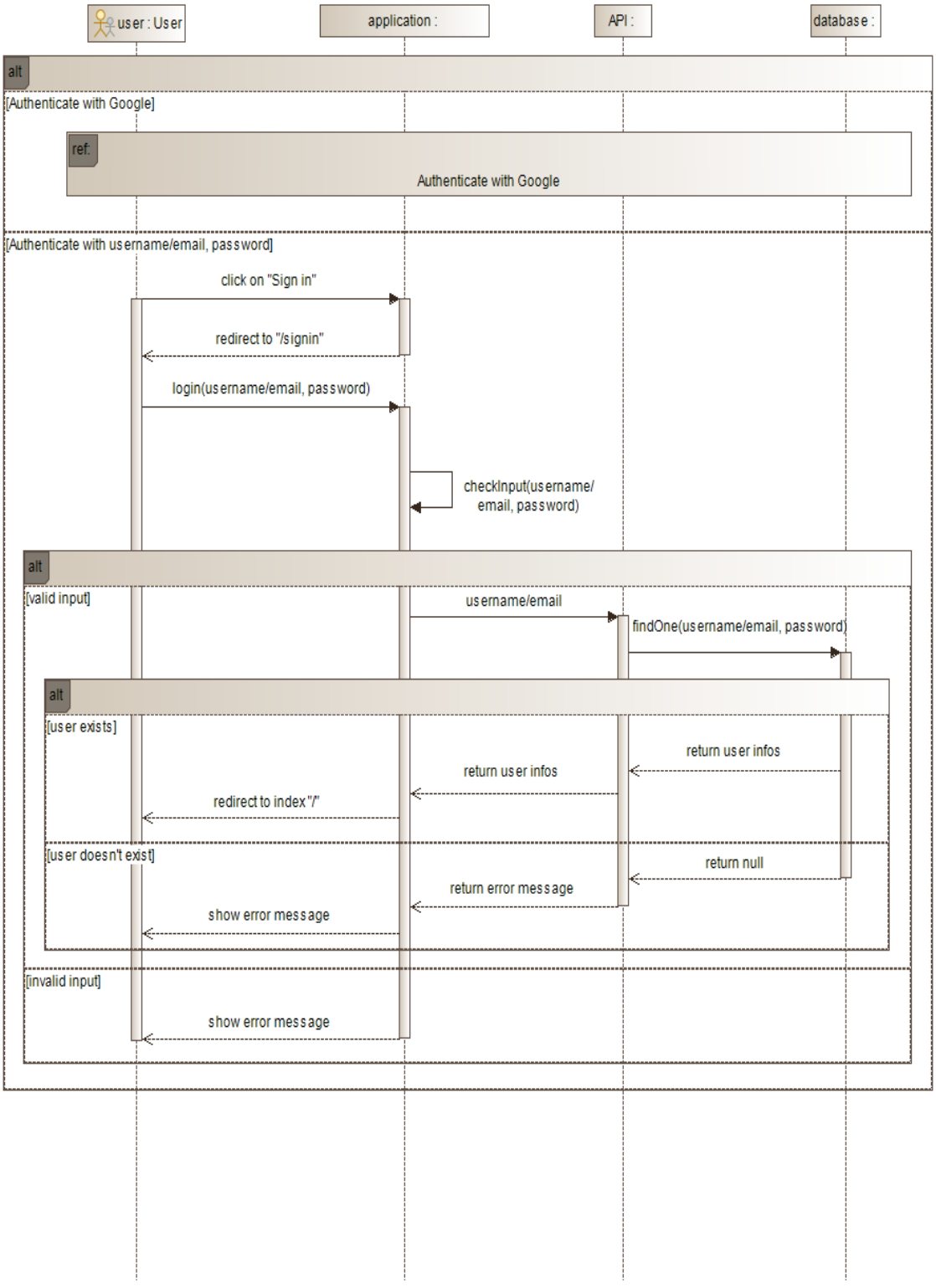


Figure 3.2 : Authenticate sequence diagram

The Google authentication process referenced by “Authenticate with Google” interaction use is explained in Figure 3.3.

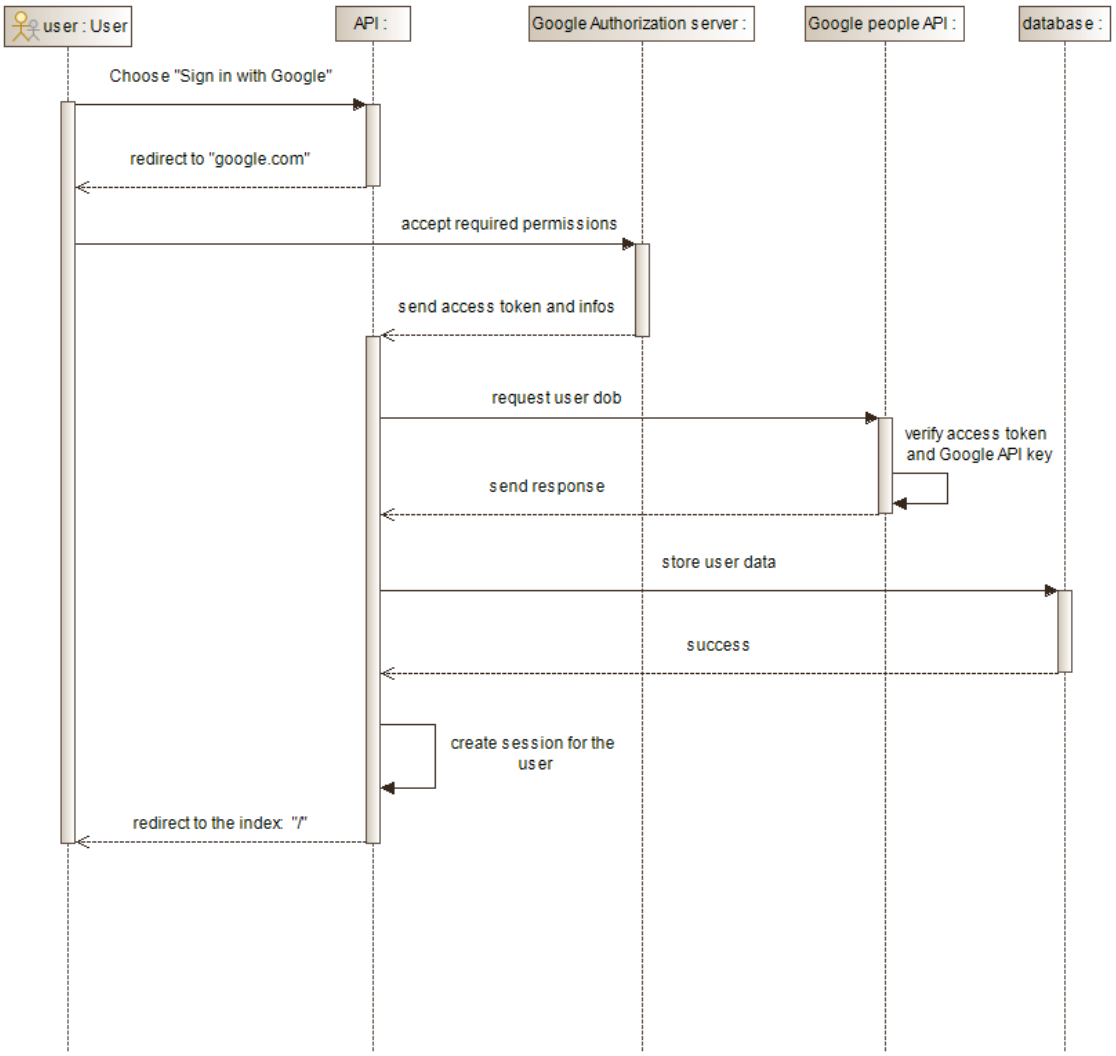


Figure 3.3: Google Authentication sequence diagram

As shown in Figure 3.4, to search for specific case(s), the user should provide the description of the case as input. Afterwards, while retrieving the existing clusters, the description is pre-processed before we can use it to generate its vector representation by calculating the average of the representations of the terms within the description. Then, for each one of the clusters, we compute the cosine similarity between the cluster head/centroid and the vector representation of the description. Next, we take the **_id** of the cluster which is the closest to the description in terms of representations and use it to retrieve the correspondent document(s) and return them to the user. If no similar document(s) was/were found, an error message will be returned instead.

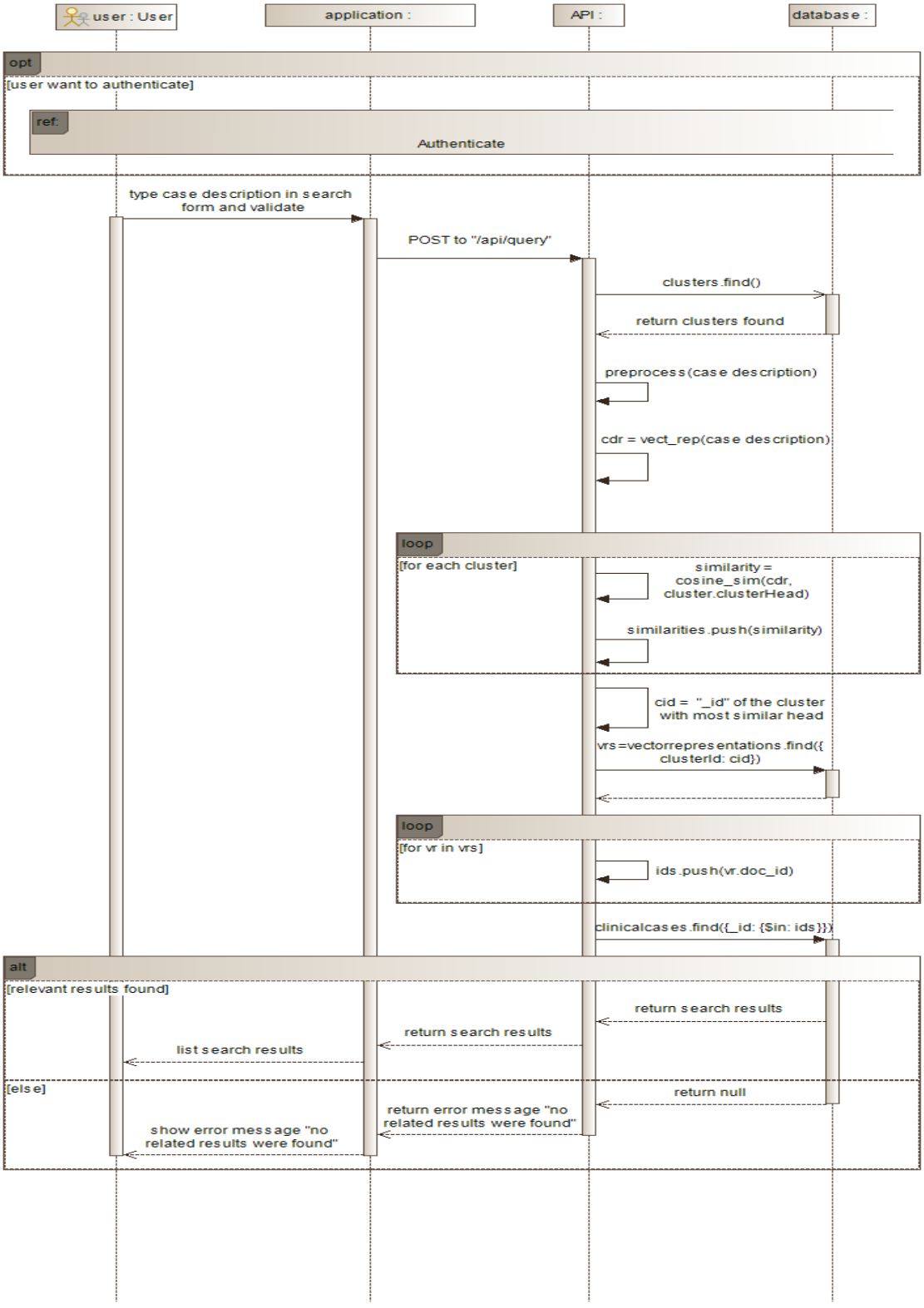


Figure 3.4: Search case(s) sequence diagram

Figure 3.5 shows that to share a new case, user must authenticate first for two main reasons:

- To check, when trying to share new cases, if he is authorized to do that or no.

- To add the username to the new case information. This will help other users to determine the credibility of the case.

After authenticating, if a user tries to share a new case and it is confirmed that he is authorized to share, he will be redirected to another view where he will be invited to fill and submit the clinical case information so that we store them in the database. Among the submitted information, we have the clinical case description that will be pre-processed before generating then storing its vector representation for a later use. The addToCluster method describes the process of classifying the new case and that is by measuring the similarity between cluster heads and the vector representation of the case.

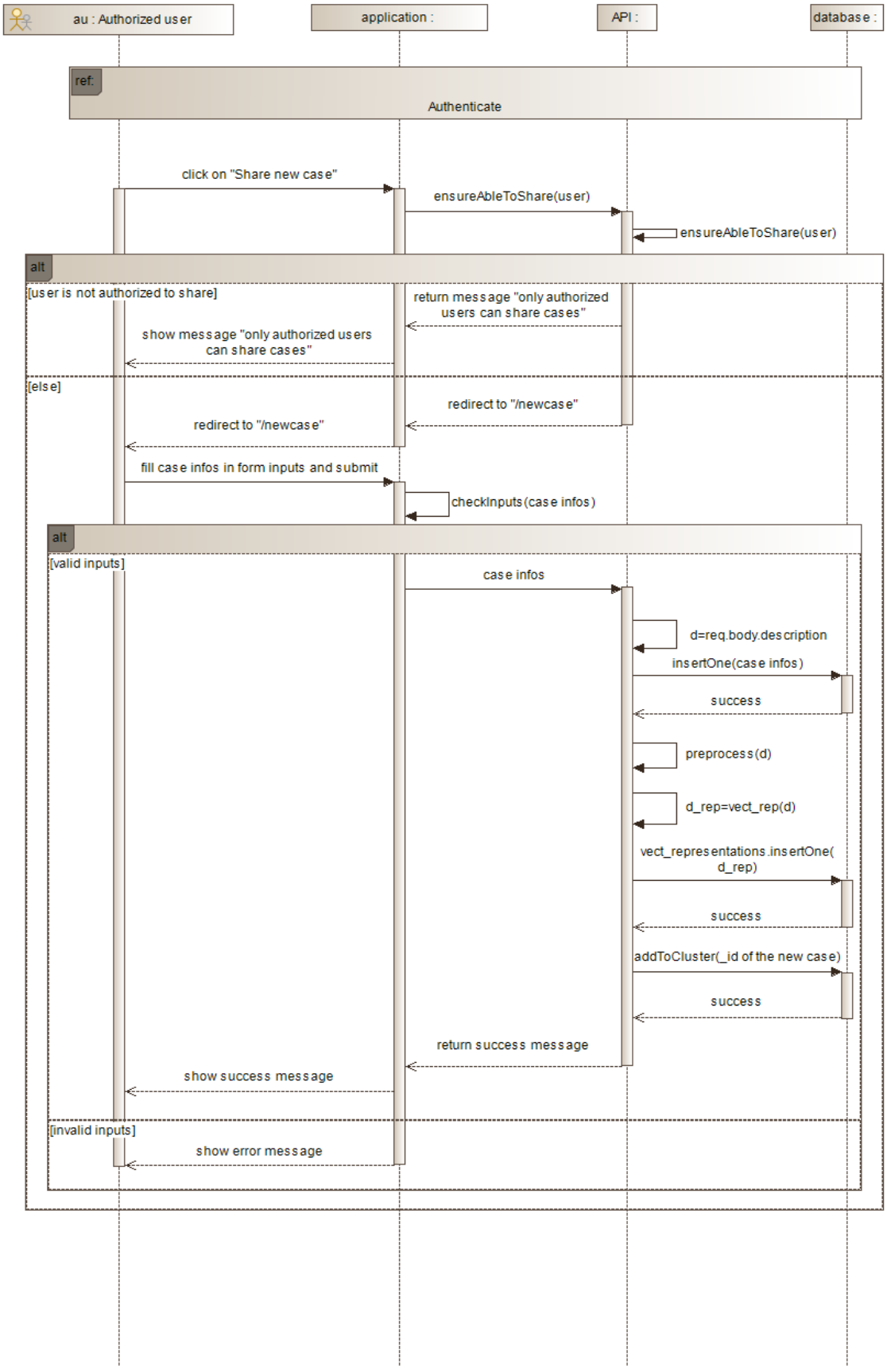


Figure 3.5: Share new case sequence diagram

In Figure 3.6 we see that the user can directly access a case from the ones already listed (in ***/cases*** view) or, make a search first if he is looking for a specific topic then choose from search results a case to access using its **_id**.

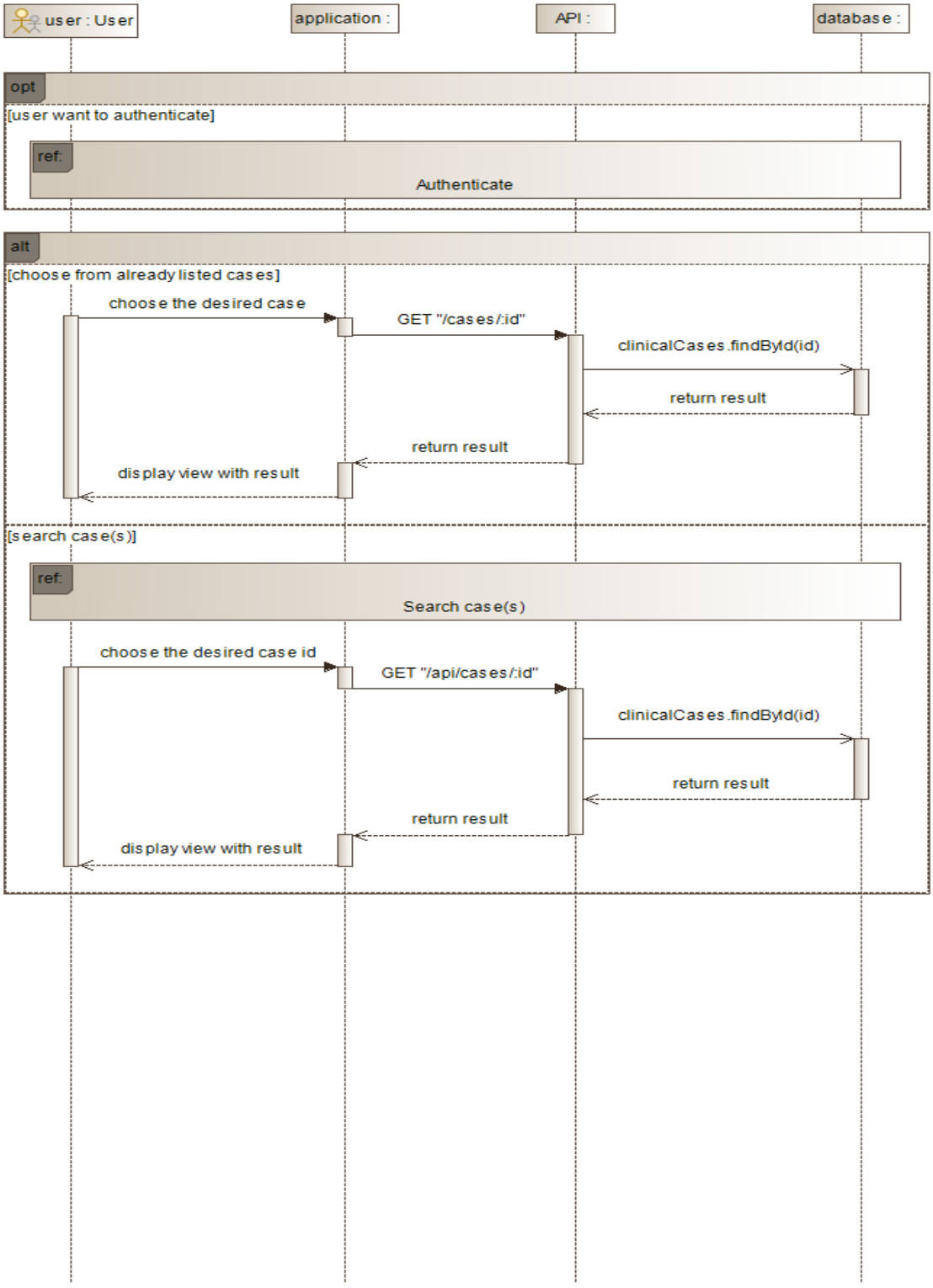


Figure 3.6 : Access case sequence diagram

### 3.3. Class diagram

The classes in Figure 3.7 describe the structure of the entities of the system and the relations between these entities.

- A user, if he is an admin, can manage many users.
- A user can access or search for case(s). Also, he can share new cases if he is authorized to do that and manage the cases he shared.
- A clinical case has a vector representation which is actually the representation of its description.
- The vector representation of a clinical case belongs to one cluster, while a cluster may include many vector representations.
- A cluster[4] has one or many clinical cases as elements (Cluster.clusterElements contains the **_id**s of the clinical cases belonging to the cluster).

*Note:*

- VectorRepresentation.doc_id corresponds to the **_id** of the clinical case.

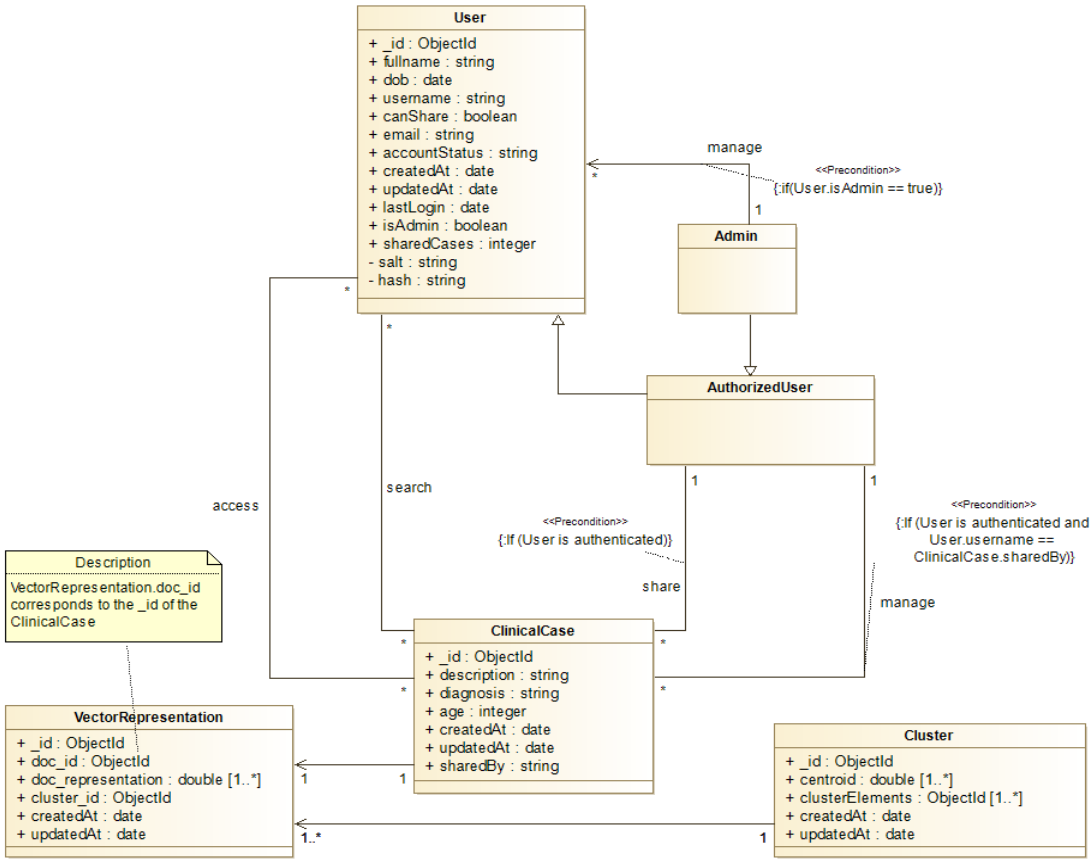


Figure 3.7: Entities class diagram

[4] In our case, a cluster represents the domain/specialization/subject to which a clinical case belongs. E.g.: cardiology, paediatrics, etc.

# 4 Implementation

## 4.1. Tools

- **Visual Studio Code**: Visual Studio Code is a free, multiplatform and built on open source code editor from Microsoft. In addition to the built-in support for JavaScript, Typescript and Node.js, it supports other languages such as Python, PHP, etc [30].
- **Modelio Open Source**: Modelio Open Source is a free, open source modeling environment supporting many standards/languages like UML, SysML, SoaML, etc [31]. It can be extended using modules whether they already exist or developed by the user. The version we used in our work is 3.8.
- **MongoDB Compass**: MongoDB Compass is the free, source-available and multi platform MongoDB client used for managing MongoDB data [32].
- **MongoDB Atlas**: MongoDB Atlas is a fully-managed cloud database that handles all the complexity of deploying, managing, and healing deployments on the cloud service provider of your choice (AWS, Azure, and GCP) [33].
- **Git**: Git is a free and open source distributed version control system designed to manage projects with speed and efficiency [34]. It is mainly used for source code management in software development.

  An important difference between Git and other VCSs is the way of storing data. Every time you commit, or save the state of your project in Git, it basically takes a picture of what all your files look like at that moment and stores a reference to that snapshot [35].
- **Github**: Github is a platform where developers and companies collaborate to build and maintain software [36]. Users can manage their code on this platform via Git.
- **Anaconda**: Anaconda Distribution is an open-source Python distribution platform that makes Python/R packages management simple [37].

## 4.2. Frontend technologies:

- **HTML: (**for HyperText Markup Language) is a descriptive language to define webpage structure [38]. The version we used is HTML5 and it is the latest version of HTML.
- **CSS:** (Cascading Style Sheets) is a declarative language that controls how webpages look in the browser [39]. The version we used is CSS3 (current version of CSS).

- **JavaScript:** JavaScript is a client-side programming/scripting language used to create dynamic webpages [40].
- **jQuery:** jQuery is a fast and small JavaScript library. It helps writing clear and simple JavaScript code and makes manipulating with HTML elements easier [41].
- **Sass:** Sass is a stylesheet language that's compiled to CSS. It uses some programming ideas like conditions, loops, etc, all with CSS-compatible syntax. Sass helps keep large stylesheets well-organized and makes their maintenance easy [42].

### 4.3. Backend technologies:

- **Node.js:** is an open-source and multi-platform asynchronous event-driven JavaScript runtime environment [43]. It is a popular tool for almost any kind of projects especially microservices.
- **Express.js:** Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications [44].
- **Pugjs:** is a template engine heavily influenced by Haml [45] and implemented with JavaScript for Node.js and browsers [46]. Pug code will be later compiled to HTML code.
- **Passport:** Passport is an authentication middleware for Node.js that can be integrated in applications that use Express.js [47]. It offers a variety of authentication strategies for a wide range of platforms, including social networks.
- **Mongoose:** is a MongoDB ODM (object modeling tool) designed to work in an asynchronous environment [48]. One of the powerful features of Mongoose is schemas that make model abstraction so the user can deal with data as simple objects.

### 4.4. Other third party tools:

- **Google-translate:** a Node.js library to consume Google Translate API and available for free [49].
- **Word2vec-api:** a simple web service providing a word embedding model [50]. This API accepts models using other architectures and/or trained using other algorithms such as GloVe.
- **Node-kmeans:** an NPM package providing an implementation of the k-means clustering algorithm [51].

From the model choice perspective, we have used in our work a pre-trained model from GloVe with a vocabulary size of 400,000 most frequent words and 50 dimensions for each word vector. This model was trained using a combination of two corpora (6 billion tokens): the 2014 Wikipedia dump (1.6 billion tokens) and Gigaword5 (4.3 billion tokens).

## 5 System architecture

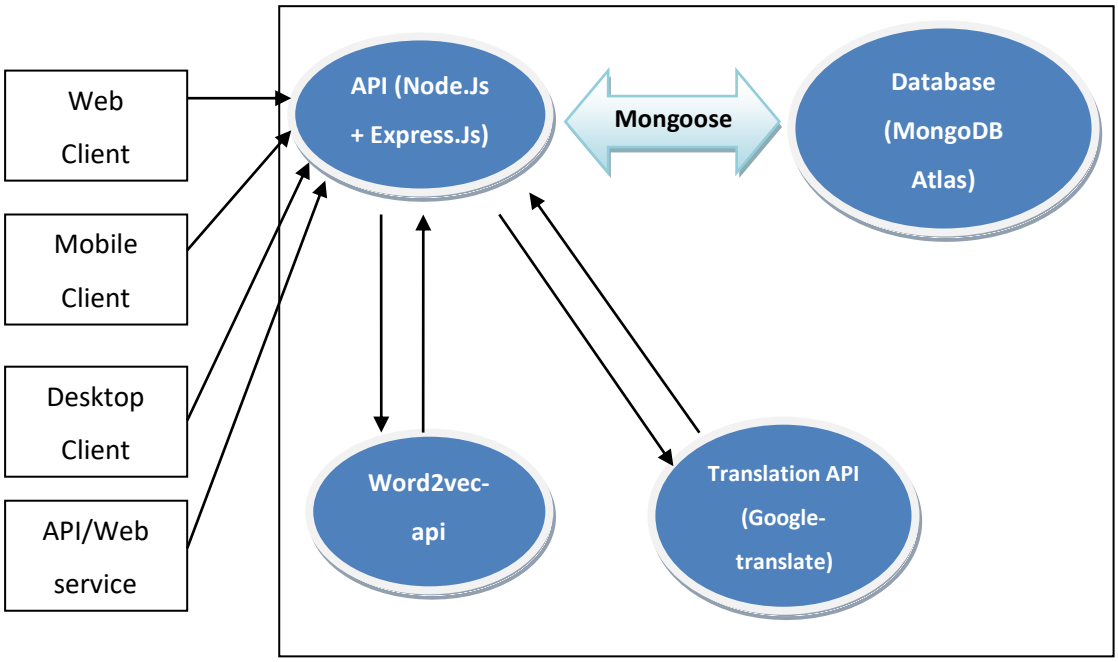


Figure 3.8 : System Architecture

## 6 API explanation

### 6.1. Routes

Table 3.1: API routes and their description

| Route | HTTP method(s) | Description |
|---|---|---|
| **/api/cases** | GET | Get all the clinical cases |
| **/api/cases/:id** | GET | Get a specific clinical case using its id |
| **/api/query** | POST | Search for clinical case(s) |
| **/newcase** | POST | Share a new clinical case |

The Table 3.1 shows the main routes of the API in addition to the HTTP methods used to access each one of these and a brief description of their usage. Some additional routes are represented in table below.

Table 3.2: Additional routes

| Route | HTTP method(s) | Description |
|---|---|---|
| **/** | GET | Basic route |
| **/signin** | GET, POST | Route to handle user sign in |
| **/signup** | GET, POST | Route to handle user sign up |
| **/signout** | GET | Route to handle user sign out |
| **/users** | GET | Get all users |
| **/user** | PUT, DELETE | Route accessed when updating or deleting a user |
| **/case** | PUT, DELETE | Route accessed when updating or deleting a clinical case |

## 7 API call examples

### GET /api/cases/:id

Parameters: id of the clinical case.

Response type: “application/json”

Response example:

```
{
    "result": {
    "_id": "6299d12b04636639825e45c7",

    "description": `Mr B.A, 45 ans, hypercholestérolémique (3 g/l),
                    hypertendu (160/90 mm Hg) et tabagique (30 paquets/années)
                    est hospitalisé en raison d'une dyspnée stade Ila ayant
                    débuté il y a quelques mois et qui s'aggrave progressivement,
                    actuellement en une dyspnée de repos.
                    \r\nL'examen clinique retrouve une tachycardie à
                    l'auscultation associée à un bruit de galop et des oedèmes
                    des membres inférieurs.`,

    "diagnosis": `Insuffisance cardiaque globale
                  chronique évoluée stade 4 basé sur:\r\nDyspnée de
                  repos\r\nTachycardie, galop\r\nOedèmes des MI\r\nFDR
                  cardio-vasculaire; HTA, Dyslipidémie, tabagisme`,
    "age": 45,
    "createdAt": "2022-06-03T09:01:56.358Z",
    "updatedAt": "2022-06-03T09:01:56.358Z",
    "sharedBy": "redouane_18",
    "__v": 0
    }
}

```


Figure 3.9: API response to GET /api/cases/:id request

**GET /api/cases**

Response type: “application/json”

Response example:

```
{
    "results": [
        {
            "_id": "6299262793da5e6dff060250",
            "description": `une femme âgée de 33 ans, se présente aux UMC pour
            un érythème noueux des membres inférieurs dans un cadre fébrile.`,
            "diagnosis": `Quel diagnostic probable évoquez-vous ?\r\n
            1. Varicelle.\r\n
            2. Infection streptococcique.\r\n
            3. Ulcère de la jambe.\r\n
            4. Érysipèle.\r\n
            5. Rougeole.\r\n\r\n
            Réponses: 2.`,
            "age": 33,
            "createdAt": "2022-06-02T21:56:56.784Z",
            "updatedAt": "2022-06-02T21:56:56.784Z",
            "sharedBy": "amine"
        },
        {
            "_id": "6299289e93da5e6dff06027e",
            "description": `Un jeune patient de 30 ans, toxicomane présentant une
            toux productive chronique, céphalées fébriles évoluant depuis un mois
            avec une opacité hétérogène apicale droite.À l'admission, l'examen
            clinique retrouve une paralysie faciale périphérique et un ptosis
            gauche.
            L'examen du LCR montre une pléioytose lymphocytaire avec
            hypoglycorachie
            et hyperalbuminorachie.`,
            "diagnosis": `Quel est le diagnostic le plus probable ?\r\n
            1. Une méningite listérienne.\r\n
            2. Une méningite tuberculeuse.\r\n
            3. Une méningite à pneumocoque.\r\n
            4. Une méningite herpétique.\r\n
            5. Une méningite à VIH.\r\n\r\n
            Réponses: 2.`,
            "age": 30,
            "createdAt": "2022-06-02T21:56:56.784Z",
            "updatedAt": "2022-06-02T21:56:56.784Z",
            "sharedBy": "hicham"
        },
    .
    .
  ],
        "nbCases": 36
}
```


Figure 3.10: API response to GET /api/cases request

**POST /api/query**

Parameters: a description of the clinical case as input. Response type: “application/json”

Response example:

```
{
    "results": [
        {
            "_id": "62a27d53d70481f02164094f",
            "description": `Madame B.S âgée de 35 ans, enseignante, mère
            de 2 enfants, mariée depuis 10 ans, vient consulter seule pour
            une anxiété majeure liée à des hallucinations.\r\nActuellement
            souffrant d'une asthénie qui la rendait incapable d'assurer ses
            responsabilités maternelles depuis presque un mois.Elle est
            obsédée par l'idée d'une rechute ainsi que des idées de
            damnation.\r\nEn effet après ces deux accouchements une instabilité
            psychomotrice s'est installée nécessitant une hospitalisation
            dans un service spécialisé avec rémission totale entre les
            épisodes.Aujourd'hui elle présente une intolérance à la
            frustration.Elle ne supporte pas le moindre bruit.\r\nDe plus,
            une indifférence à l'égard de sa famille et des idées de suicide
            ainsi que des hallucinations visuelles.Une tentative de suicide
            médicamenteuse lors du premier épisode.L'examen Somatique est
            sans particularité. Absence de prise médicamenteuse ou toxique.`,
            "diagnosis": `1. L'examen psychiatrique de madame B.S retrouve
            tous les éléments suivants sauf:\r\n
            1. Anxiété massive.\r\n
            2. Anesthésie affective.\r\n
            3. Irritabilité.\r\n
            4. Anhédonie.\r\n
            5. Incurabilité.\r\n\r\n
            Réponses: 4.\r\n\r\n2.
            Le diagnostic le plus probable ce jour étant: \r\n
            1. Trouble psychotique du post partum.\r\n
            2. Trouble dépressif majeur chronique.\r\n
            3. Trouble dépressif majeur sévère avec caractéristique psychotique.
\r\n
            4. Accès hypomaniaque.\r\n
            5. Trouble dépressif majeur atypique.\r\n\r\nRéponses: 3.`,
            "age": 35,
            "sharedBy": "redouane_18",
            "createdAt": "2022-06-10T00:08:03.037Z",
            "updatedAt": "2022-06-10T00:08:03.038Z"
        },
        {
            "_id": "62a27daed70481f02164095f",
            "description": `Monsieur A.N âgé de 23 ans s'est présenté seul
            aux consultations psychiatriques sans rendez vous et dès son
            entrée au bureau de consultation commence à rire puis à pleurer
            sans aucune raison puis au cours de l'entretien, le médecin lui
            demande sa raison pour être venu au service de psychiatrie alors
            le patient déclare « je suis venu aujourd'hui pour renouveler
            mon traitement car les oiseaux chantent tous les jours » et puis
            tout à coup sans aucune raison ni explication il sort du bureau.
            Monsieur A.N est suivi en psychiatrie depuis 4 ans pour une
            schizophrénie mais ne prends pas correctement son traitement et
            fait beaucoup d'arrêts thérapeutiques d'où ses rechutes et
            actuellement ne prend aucun traitement alors qu'il était sous
            Rispéridone / Risperdal à raison de 8 mg par jour.`,
            "diagnosis": `Que présente monsieur A.N?\r\n
            1. Un néologisme.\r\n
            2. Un paralogisme.\r\n
            3. Un rationalisme morbide.\r\n
            4. Une pensée désorganisée.\r\n
            5. Un affect désorganisé.\r\n\r\n
            Réponses: 3, 4, 5.`,
            "age": 23, "sharedBy": "redouane_18",
            "createdAt": "2022-06-10T00:09:34.119Z",
            "updatedAt": "2022-06-10T00:09:34.120Z"
        }
    ],
        "nbCases": 2
}
```


Figure 3.11: API response to POST /api/query request

## 8 Client web application

This application' screenshots (From Figure 3.12 to Figure 3.17) show a usage example of the API

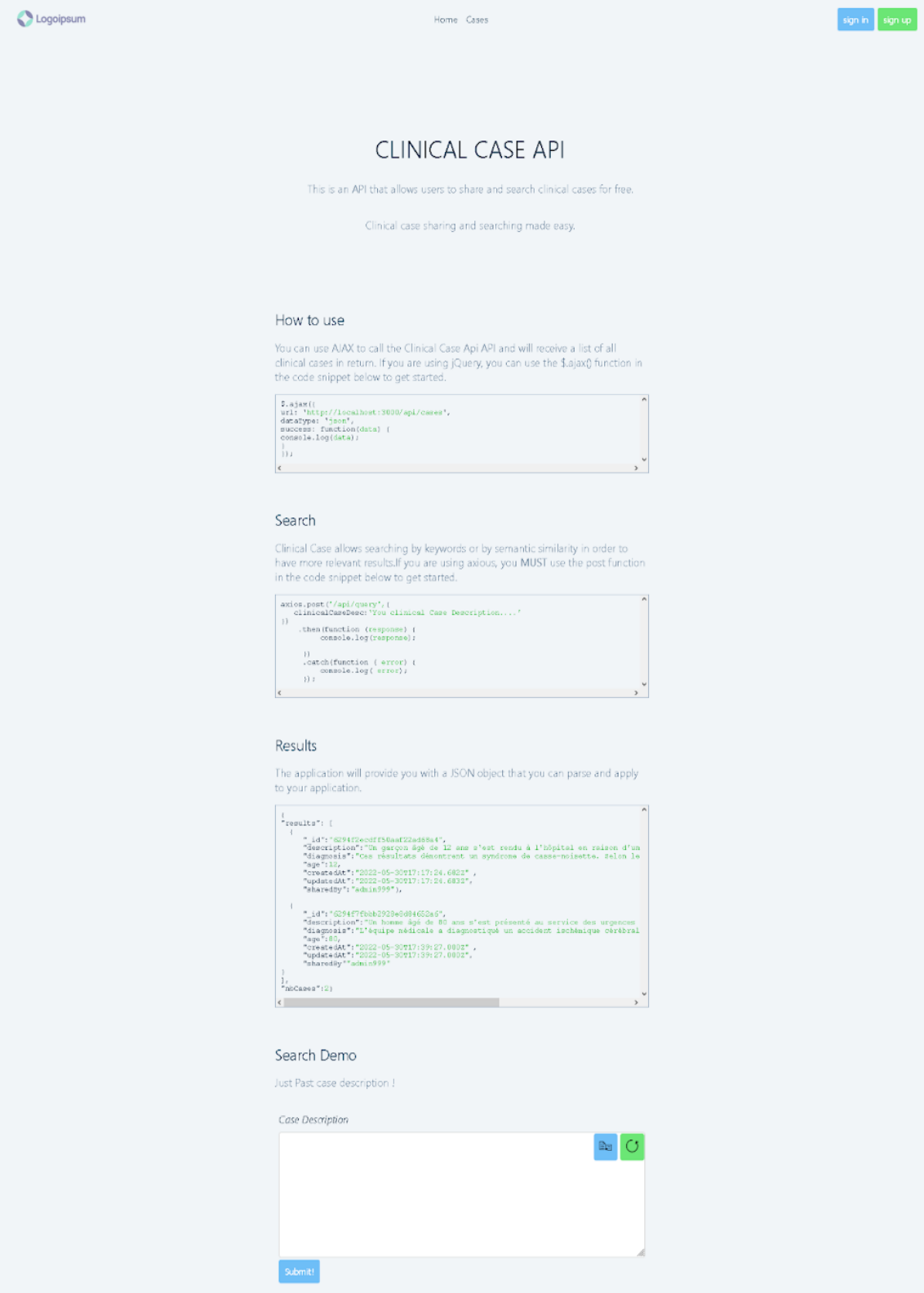


Figure 3.12 : Application home page

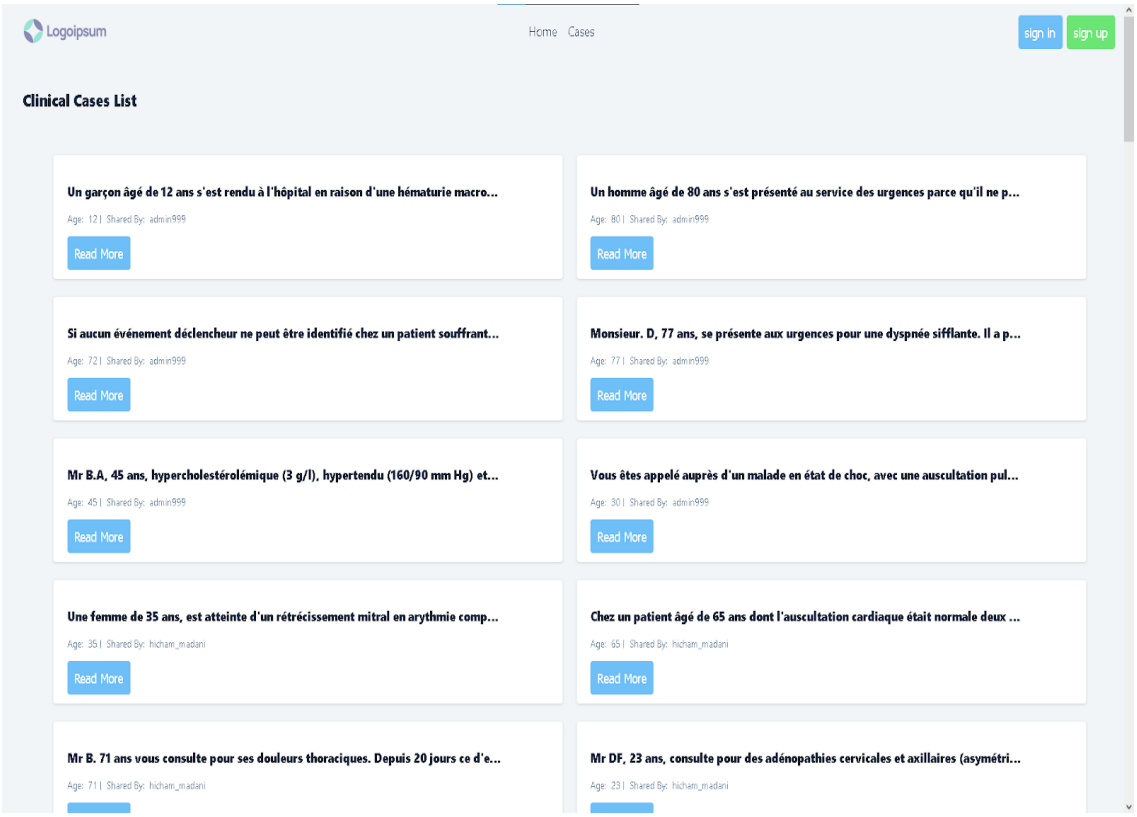


Figure 3.13 : Cases list

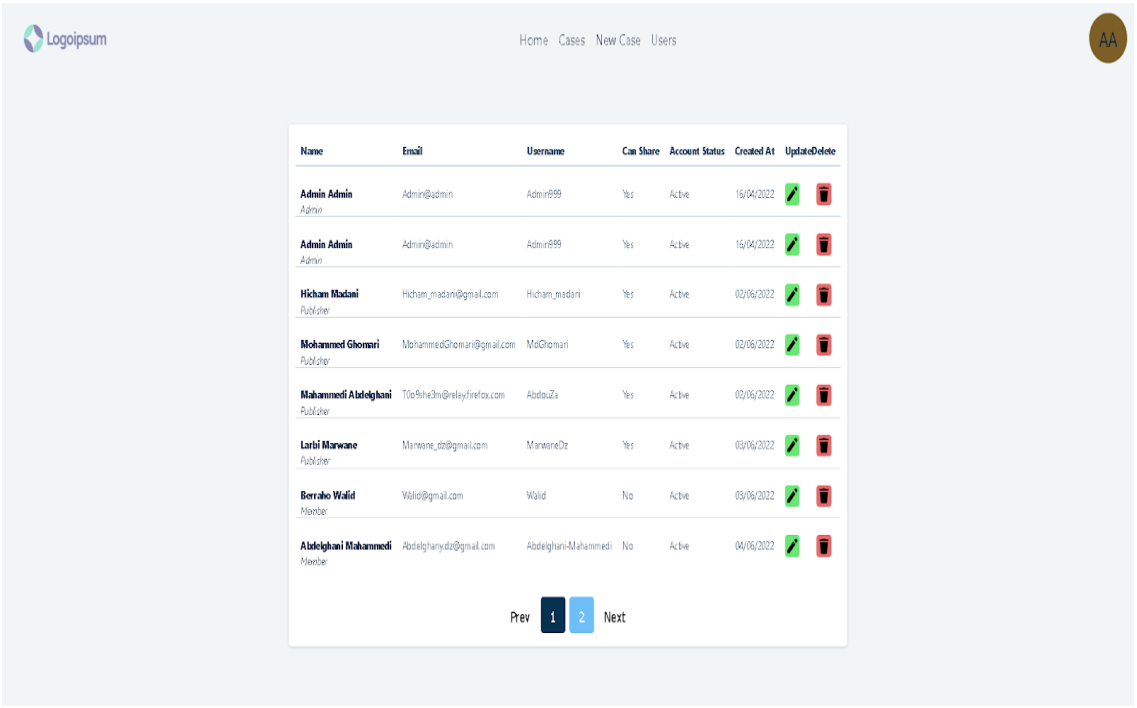


Figure 3.14 : Users list

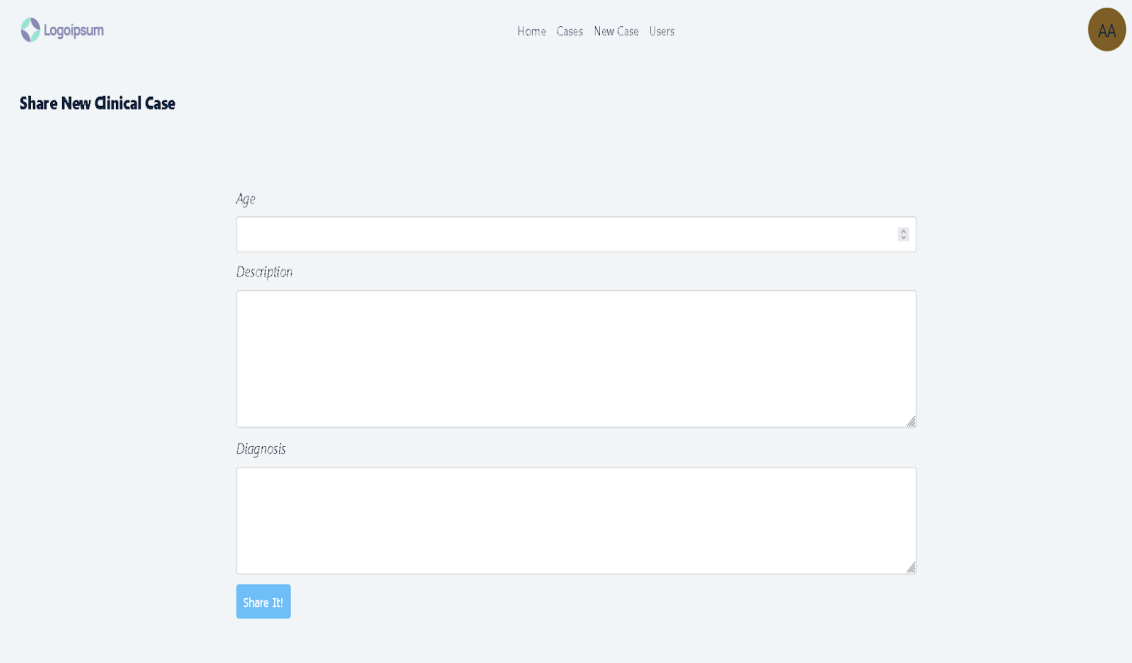


Figure 3.15 : Share new case view

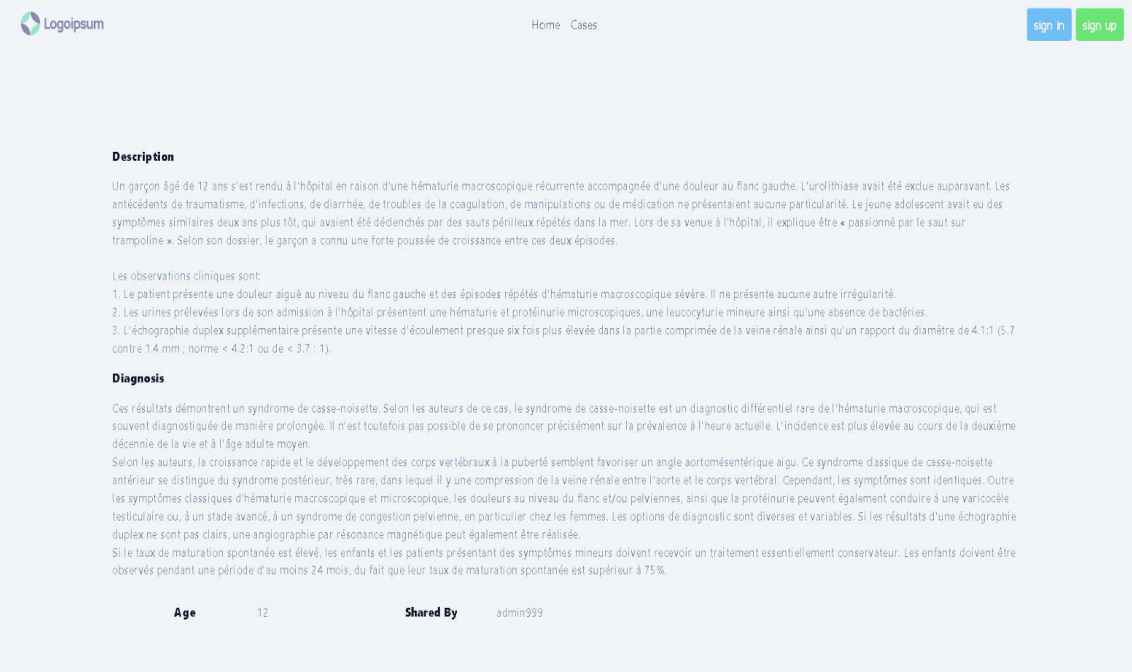


Figure 3.16 : Clinical case details view

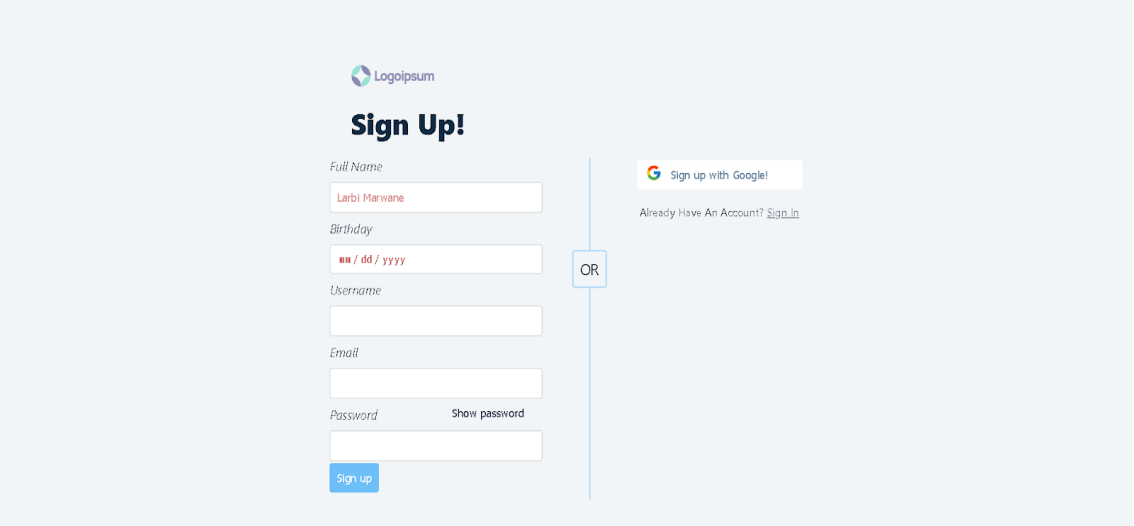


Figure 3.17 : Sign up view

## 9 Conclusion

In this final chapter, we present our API and its UML diagrams. The initial objectives of our API are met in addition to the benefit of using word2vec for a semantic representation of the clinical cases. Although our use of word2vec is simplistic, it enables the application to run fast while requiring less processing power. However, it is possible to have a more meaningful semantic representation and hence, more accurate query results.

## General conclusion

Today, it is becoming increasingly common to use APIs since they can be exploited by the majority of applications (web, mobile, and even desktop apps).

As part of this work, we developed an API that enables users (often medical professionals or students) to browse previously created clinical cases or search through them by providing a description that will be processed and then used by the API to search either by keywords or by semantic similarity to retrieve the most relevant results. Furthermore, authorized users can share new cases through the web application we have proposed.

As a perspective, we would like to improve this solution by:

- Enhance the search system to get more relevant results.
- Add another authentication mechanism (use access token strategy) in case we decide to change the sharing policy (for example, give all users the right to share).

## References


1. **ROBILLARD, Martin P.** What makes APIs hard to learn? Answers from developers. *IEEE software*, 2009, vol. 26, no 6, p. 27-34.

2. Lecturio Medical Education. [Online] March 7, 2016. [Cited: February 26, 2022.] 22.0.0, https://play.google.com/store/apps/details?id=de.lecturio.android.LecturioMed.

3. Lecturio Medical Education. [Online] 2015. [Cited: April 15, 2022.] 22.7.0, https://apps.apple.com/us/app/lecturio-medical-education/id1067957933.

4. Prognosis : Your Diagnosis. [Online] January 26, 2011. [Cited: April 24, 2022.] 6.0.4, https://play.google.com/store/apps/details?id=com.medicaljoyworks.prognosis.

5. Prognosis : Your Diagnosis. [Online] 2010. [Cited: April 15, 2022.] 6.0.4, https://apps.apple.com/us/app/prognosis-your-diagnosis/id392489854.

6. AB-QCM (Résidanat). [Online] February 16, 2021. [Cited: February 16, 2022.] 1.0.1, https://play.google.com/store/apps/details?id=com.ab_app.ab_qcm.

7. *ESPACE DES ÉTUDIANTS EN MÉDECINE.* [Online] September 26, 2017. [Cited: April 15, 2022.]. https://fmed13.weebly.com/.

8. *Medscape*. [Online] May 22, 1995. [Cited: April 15, 2022.] https://www.medscape.com/public/about.

9. *Medscape*. [Cited: April 16, 2022.] https://www.medscape.com/consult/.

10. *MSD MANUALS.* [Online] August 14, 2015. [Cited: April 16, 2022.] https://www.msdmanuals.com/.

11. *MSD MANUALS.* [Cited: April 16, 2022.] MSD MANUALS Professional Version. https://www.msdmanuals.com/professional.

12. *Passmedicine*..[Online] December 23, 2007. [Cited: April 24, 2022.] a proven revision resource that helps you pass high-stakes medical exams. https://www.passmedicine.com/.

13. *AMBOSS.* [Online] February 23, 2017. [Cited: April 24, 2022.] Medical knowledge platform for doctors and students. https://www.amboss.com/int.

14. *AMBOSS.* [Online] February 23, 2017. [Cited: April 20, 2022.] Medical knowledge platform for doctors and students. https://www.amboss.com/us/about.

15. *Lecturio.* [Online] December 02, 2014. [Cited: April 22, 2022.] https://www.lecturio.com/.

16. *Osmosis.* [Cited: April 23, 2022.] https://www.osmosis.org/.

17. *Osmosis*. [Cited: April 23, 2022.] https://www.osmosis.org/plans/md.

18. *Radiopaedia*. [Online] 2005. [Cited: April 23, 2022.] a rapidly growing open-edit educational radiology resource. https://radiopaedia.org/.

19. *Medmastery*. [Cited: April 24, 2022.] https://www.medmastery.com/help/faqs.

20. *Medmastery*. [Cited: April 24, 2022.] https://www.medmastery.com/.

21. What is Supervised Learning? *IBM Cloud Learn Hub.* [Online] August 19, 2020. [Cited: March 29, 2022.] Education about cloud topics, technologies, and strategies. https://www.ibm.com/cloud/learn/supervised-learning.

22. What is Unsupervised Learning? *IBM Cloud Learn Hub.* [Online] September 21, 2020. [Cited: March 29, 2022.]. Education about cloud topics, technologies, and strategies. https://www.ibm.com/cloud/learn/unsupervised-learning.

23. **Liddy, Elizabeth D.** Automatic Document Retrieval. *Encyclopedia of Language and Linguistics.* 2nd Edition. s.l. : Elsevier Press, 2005.

24. **Dessi, Danilo, et al.** Tf-IDF vs word embeddings for morbidity identification in clinical notes: An initial study. *arXiv preprint arXiv:2105.09632.* 2021.

25. **Mikolov, Tomas, et al.** Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781.* 2013a.

26. word2vec | TensorFlow Core. *TensorFlow.* [Cited: April 12, 2022.] https://www.tensorflow.org/tutorials/text/word2vec.

27. **Mikolov, Tomas, Le, Quoc V and Sutskever, Ilya.** Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168.* 2013b.

28. **Jeffrey, PENNINGTON, Richard, SOCHER and Christopher D, MANNING.** GloVe: Global Vectors for word representation. 2014.

29. **QUATRANI, Terry and EVANGELIST, U. M. L.** Introduction to the Unified modeling language. *A technical discussion of UML,* 2003, vol. 6, no 11, p. 03

30. *Visual Studio Code.* [Cited: May 10, 2022.] https://code.visualstudio.com/.

31. *Modelio Open Source.* [online] 2011. [Cited: May 10, 2022.] https://www.modelio.org/.

32. MongoDB Compass. *MongoDB* [Cited: May 11, 2022.] The GUI for MongoDB. https://www.mongodb.com/products/compass.

33. MongoDB Atlas Database. *MongoDB.* [Cited: May 14, 2022.] https://www.mongodb.com/basics/mongodb-atlas-tutorial.

34. *Git.* [Cited: May 14, 2022.] https://git-scm.com/.

35. **Chacon, Scott and Straub, Ben.** *Pro Git.* [ed.] Apress. Berkeley, CA : Springer Nature, 2014. pp. 1-13.

36. *Github.* [Online] April, 2008. [Cited: May 14, 2022.] https://github.com/about.

37. Anaconda Distribution. *ANACONDA.* [Cited: May 14, 2022.] https://www.anaconda.com/products/distribution.

38. HTML – MDN Web Docs Glossary. *MDN Web Docs.* [Cited: May 14, 2022.] https://developer.mozilla.org/en-US/docs/Glossary/HTML.

39. CSS – MDN Web Docs Glossary. *MDN Web Docs.* [Cited: May 14, 2022.] https://developer.mozilla.org/en-US/docs/Glossary/css.

40. JavaScript – MDN Web Docs Glossary. *MDN Web Docs.* [Cited: May 14, 2022.] https://developer.mozilla.org/en-US/docs/Glossary/javascript.

41. *jQuery.* [Cited: May 14, 2022.] https://jquery.com/.

42. *Sass: Syntactically Awesome Style Sheets.* [Cited: May 14, 2022.] https://sass-lang.com/documentation.

43. *Node.js.* [Cited: May 14, 2022.] https://nodejs.org/en/about/.

44. *Express.* [Cited: May 14, 2022.] https://expressjs.com/.

45. *Haml.* [Cited: May 20, 2022.] https://haml.info/.

46. Pug. *Github.* [Cited: May 14, 2022.] https://github.com/pugjs/pug.

47. *Passport.js.* [Cited: May 14, 2022.] https://www.passportjs.org/.

48. Mongoose. *Github.* [Cited: May 14, 2022.] https://github.com/Automattic/mongoose.

49. [Cited: May 14, 2022.] https://github.com/iamtraction/google-translate.

50. word2vec-api. *Github.* [Online] July 15, 2014. [Cited: May 14, 2022.] https://github.com/3Top/word2vec-api.

51. node-kmeans. *Npm.* [Cited: June 06, 2022.] https://www.npmjs.com/package/node-kmeans.

**Abstract:**

It is common for medical students to learn by practice and this is largely achieved by diagnosing clinical cases. In this work, we present the development of an educational API that enables users to browse, search and share clinical cases. Moreover, the proposed API uses word2vec which is a Deep Learning model for semantic word embedding alongside the clustering algorithm k-means. By doing so, we enable the user to search for clinical cases similar to a given query string.

**Keywords:** API, online sharing of clinical cases, Deep Learning, clustering, k-means

**Résumé:**

Il est courant que les étudiants en médecine apprennent par la pratique et cela est largement réalisé en diagnostiquant des cas cliniques. Dans ce travail, nous présentons le développement d'une API éducative qui permet aux utilisateurs de parcourir, rechercher et partager des cas cliniques. De plus, l'API proposée utilise word2vec qui est un modèle Deep Learning pour l'intégration sémantique des mots aux côtés de l'algorithme de clustering k-means. Ainsi, nous permettons à l'utilisateur de rechercher des cas cliniques similaires à une chaîne de requête donnée.

**Mots-clés:** API, partage en ligne de cas cliniques, Apprentissage Approfondie, clustering, k-means

**ملخص:**

من الشائع أن يتعلم طلاب الطب بالممارسة و يتحقق ذلك إلى حد كبير من خلال تشخيص الحالات السريرية. في هذا العمل، نقدم واجهة برمجة تطبيقات تعليمية تمكن المستخدمين من تصفح الحالات السريرية و البحث عنها و مشاركتها. علاوة على ذلك، تستخدم واجهة برمجة التطبيقات المقترحة word2vec وهو نموذج التعلم العميق لتضمين الكلمات الدلالية جنبا إلى جنب مع خوارزمية التجميع k-means. من خلال القيام بذلك، يمكننا تمكين المستخدم من البحث عن الحالات السريرية المشابهة لسلسلة استعلام معينة.

**الكلمات الرئيسية:** API، المشاركة عبر الإنترنت للحالات السريرية، التعلم العميق ، التجميع، k-means