Université de Blida 1–Saad Dahlab

Faculté des sciences

Département d'Informatique

Mémoire présenté par :

Mr CHAGUETMI Yousr ELLAH Wassim

Pour l'obtention du diplôme de Master

Domaine : Mathématique et Informatique

Filière : Informatique

Spécialité : Traitement Automatique de la Langue

Suj

Ensemble Based Decision Making System from Vocal Hints.

Soutenu le : 07 Juillet 2020, devant le jury composé de :

Mme. N. BENBLIDIA	Université de Blida 1	Présidente
Mme. L. Ouahrani	Université de Blida 1	Examinatrice
Mme. M. MEZZI	Université de Blida 1	Promotrice

Abstract

A decision support system (DSS) is a software that provides information used to make decisions. The DSS is generally composed of a knowledge base and an inference engine that contains all the knowledge that is useful for problem solving. The inference engine is a program that uses the knowledge contained in the knowledge base to solve a particular problem described by facts.

Some fields, in particular the medical field, are considered huge and have so much information and experiences. As for the medical field it doesn’t only have a huge amount of information, but that information is diverse and wide spread covering many concepts.

In this regard, the aim of this work, is to benefit and show the value added by ensemble-based decision-making systems approach. Taking advantage of its separation of concerns and dividing the information and data to conquer by extracting the desired results in a form of decisions.

This system will assist the doctor in his reasoning in order to make a diagnosis and choose the proper treatment while making use of the widely used UML language in systems engineering.

Keywords: *Decision Support Systems, Knowledge Engineering, Medical Decision Support System, Ensemble-based Systems.*

Résumé

Un système d'aide à la décision (DSS) est un logiciel qui fournit des informations utilisées pour prendre des décisions. Le DSS est généralement composé d'une base de connaissances et d'un moteur d'inférence contenant toutes les connaissances utiles à la résolution de problèmes. Le moteur d'inférence est un programme qui utilise les connaissances contenues dans la base de connaissances pour résoudre un problème particulier décrit par des faits.

Certains domaines, en particulier le domaine médical, sont considérés comme énormes et ont tellement d'informations et d'expériences. Quant au domaine médical, il ne dispose pas seulement d’une énorme quantité d’informations, mais ces informations sont diverses et largement diffusées et couvrent de nombreux concepts.

A cet égard, le but de ce travail est de bénéficier et de montrer l'utilité de l'approche des systèmes décisionnels basés sur l'ensemble. Profiter de sa séparation des préoccupations et diviser les informations et les données à conquérir en extrayant les résultats souhaités sous forme de décisions.

Ce système aidera le médecin dans son raisonnement afin d'identifier un diagnostic et de choisir la thérapie adéquate tout en utilisant le langage UML largement utilisé en ingénierie des systèmes.

Mots-clés : *systèmes d'aide à la décision, ingénierie des connaissances, aide à la décision médicale, systèmes basés sur des ensembles.*

ملخص

نظام دعم القرار (ن.د.ق) هو برنامج يوفر المعلومات المستخدمة في اتخاذ القرارات. تتكون (ن.د.ق) بشكل عام من قاعدة معرفية ومحرك استنتاج يحتوي على كل المعارف المفيدة لحل المشكلات. محرك الاستدلال هو برنامج يستخدم المعرفة الموجودة في قاعدة المعرفة لحل مشكلة معينة موصوفة بالحقائق.

تعتبر بعض المجالات، ولا سيما المجال الطبي، ضخمة ولديها الكثير من المعلومات والخبرات. أما المجال الطبي فهو لا يحتوي فقط على كمية هائلة من المعلومات، ولكن هذه المعلومات متنوعة وواسعة الانتشار تغطي العديد من المفاهيم.

في هذا الصدد، الهدف من هذا العمل، هو الاستفادة وإظهار فائدة نهج أنظمة صنع القرار على أساس المجموعة. الاستفادة من فصل الاهتمامات وتقسيم المعلومات والبيانات للتغلب عليها من خلال استخراج النتائج المرجوة في شكل قرارات.

سيساعد هذا النظام الطبيب في استدلاله من أجل تحديد التشخيص واختيار العلاج المناسب مع الاستفادة من لغة UML المستخدمة على نطاق واسع في هندسة النظم.

الكلمات المفتاحية : *أنظمة دعم القرار، هندسة المعرفة، دعم القرار الطبي، الأنظمة القائمة على المجموعات.*

Dedications

I dedicate this work to my lovely mum.

Your Son.

Acknowledgments

I would like to thank Allah, my mom, my five sisters especially JiJi, my supervisor whom I consider as my sixth sister and everyone who helped me in my journey.

Table of content

List of figures

List of tables

List of acronymes

DSS:	Decision Support System
IE :	Inference Engine
KB :	Knowledge base
MDSS:	Medical DSS
PSM :	Problem Solving Method
MLP :	Multilayer Perceptron

General Introduction

1. General context

In the daily life we face so many choices, either while buying groceries, shopping, studying or any other daily routine. Every choice we make, no matter how random it may seem, it is considered a decision.

One of the daily routines that require as much decision as someone can imagine, is the life of a doctor at his job. He is faced with many different cases that at first glance seem similar, but in fact are very different, and these situations are very confusing in the work, where confusion is unacceptable. Because of this, he, most of the time, asks the opinion of other colleagues in order to elaborate a decision for a specific case. This is known as, DECISION MAKING.

2. Research challenges

Decision making systems started with the beginning of information technologies as theories on papers, and went through different stages. Either an inference engine, a classifier, a neural network...etc.

They all belong to the same family of THE DECISION SUPPORT SYSTEMS (DSS). To make a decision, someone needs to know the available alternatives and the possible consequences. Especially in medical field where every action is important and every minute counts.

The lack of assistants and time makes often impossible to ask other doctors advices, use medical reference book and medical archives to make a medical to make a diagnosis. Understanding the environment is crucial to make a good DDS to help doctors, especially beginners, in their daily professional life.

The main objectives are:

- Assist and help doctors in decision making.
- Reduce the repetition of tasks.
- Digitalize the archive of documents.
- Help beginner doctors with tough decisions, especially when they are alone.
- Saving time by generating documents and reports from vocal notes.

3. Research problems

Seeing the importance of DSS, and understanding the situations a doctor can face, the main issues regarding our project can be summed up in what follows:

- How to help the doctors?
- How can we provide the help to a doctor, with a decision aid, in situations where he is alone?
- How can we simplify and automate repetitive tasks?

4. Thesis outline

To meet the above-mentioned challenges, we have organized our thesis in 4 chapters:

- Chapter 1: The first chapter of this thesis will be dedicated to the notion of "Decision Making Systems" (DMS). First, we will give a definition and then talk about the main characteristics of DMS, focusing on medical field at the same time. After that, we will give an overview of the different DM techniques and focus more accurately, on Ensemble-based Systems that we choose to use in our project.
- Chapter 2: After understanding the field that we are dealing with in the first chapter. We will present the solution modeling and conceptualization. Particularly, we will begin by a reminder about the Medical field and problems it faces and how NLP can be a solution. After that we will specify and analyze the requirements. Then, we will expose our solution making use of ensemble-based expert systems, where we will describe the knowledge base and the inference engine to understand the logic of our solution.
- Chapter 3: focus on the technical and practical aspects of the solutions. In this latter, we will first present the tools and languages that have been used. We will then talk about our dataset, database, and give some back-end execution examples

of the essential steps of the solution. Finally, we will set forth our demonstration platform through its screenshot and its evaluation.

- Finally, the general conclusion of this thesis will summarize the theoretical and practical work that have been done to the aim of finding a solution decision making based upon audio records and settle some outlooks and future works.

Chapter I:

Decision Making Systems

I.1 Introduction

Matters of great importance tend to have great consequences. And before making a decision, we need to be sure that it is the right one. Especially in areas like healthcare and medicine, where we tend to ask for a second opinion, and sometimes many more. We weigh these individual opinions and combine them through to arrive to the final decision, which is supposedly the most informed and correct one.

This “multi-expert” consultation process can be often extensive. Thus, automated decision-making applications were only a matter of time.

These processes are known under various other names, such as Multiple Classifier Systems, Committee of Classifiers, or Mixture Of Experts. Ensemble based systems [1] have performed favorable results compared to those of single-expert systems for a broad range of applications and under a variety of scenarios.

In this chapter, we will present a brief state of the art about Decision Making Systems by introducing their general characteristics and some medical decision-making systems, while focusing on Ensemble –based Systems as a promising technique in Decision Making Systems.

I.2 Definition of Decision Making

Decision Making can be defined as [2]: “The process of choosing among two or more alternative courses of action for the purpose of attaining a goal or goals.”

It is the process of making a choice by gathering and organizing relevant information, then identifying and evaluating alternative solutions. Using this step-by-step process can help get more informed and thoughtful decisions.

I.3 Characteristics of Decision Making

The characteristics of decision making, can be summed up in what follows [2]:

a) From the decision maker perspective, decision making have the following characteristics:

- Individuals are involved; diversity in opinions makes the decision more informative.
- Decisions may involve risk; different people have different appetite for risk.
- Group think can lead to bad decision.
- Decision makers are interested in evaluating “what-if” scenarios.
- There may be several conflicting objectives.

b) From environment perspective:

- Changes in the decision-making environment may occur continuously, lending to invalidating assumptions about situations.
- Collecting information and analyzing a problem takes time and can be expensive.
- Past results may not be sufficient to predict future results.
- Experimentation with a real system-trial and error-may result in failure.

c) From decisions perspective:

- Decisions are interrelated, which means that one decision might lead to another.
- There may be 100s’ of alternatives.
- There is a need for data and analysis with understanding to make a good decision.
- Experimentation with a real system is possible only for one set of conditions at a time and can be disastrous.
- There may not be sufficient information to make an intelligent decision.

I.4 Decision Making Systems

“A decision making-*system (DMS) is a computer-based information system that supports business or organizational decision-making activities, typically resulting in ranking, sorting, or choosing from among Alternatives”* [3].

A decision-making system, or expert system, is software that imitates the behavior of a human expert in a specific field to perform a task that requires intelligence (experience). They are conceived to resolve, classification or decision problems (medical diagnosis, financial rules ...). And since they are used when there is no exact algorithmic method (solution) that can be applied to the problem, are considered as AI (Artificial Intelligence) tools.

In addition, DMSs serve the management, operations, and planning levels of an organization and help people make decisions about problems that can change quickly and are not easily specified in advance (i.e., unstructured and semi-structured decision problems). Decision-making systems can be fully computerized, human-controlled, or a combination of both.

I.4.1 Decision Support Systems types

There are a number of decision support systems that can be categorized into five types [4]: communications driven DSS, data driven DSS, document driven DSS, knowledge driven DSS and model driven DSS.

- **Communication driven DSS**: mostly targeted at internal teams, including partners. Its purpose is to help users collaborate. The most common technology used to deploy the DSS is a web or client server.
- **Data driven DSS**: focuses on the collected data, which is then processed according to the needs of the decision maker. Most data-driven DSSs are targeted at managers, staff and product/service suppliers. It is used to query a database or data warehouse to find answers for specific purposes. It is deployed via a main frame system, client/server link, or via the web.
- **Document driven DSS:** targeted at a broad base of user groups their main purpose is to search web pages and find documents on a specific set of keywords or search terms. It uses computer storage and processing technologies to provide document retrieval and analysis. The technologies used commonly to set this up are via web or a client / server system.
- **Knowledge driven DSS:** it encompasses a wide variety of user systems within an organization that customize it, but can also include other users who interact with the organization. It is mainly used to provide management advice or to select products or services.
- **Model driven DSS:** it is a complex system that helps in analyzing decisions or choosing between different options. It uses limited data

and parameters provided by decision makers to help them in analyzing a situation. It is used by managers and staff members, or people who interact with the organization, for a number of purposes depending on the model setup. Also it can be deployed via software/hardware in stand-alone PCs, client/server systems, or the web.

I.5 Medical decision making

Decision making in medicine[1] have been acquired tacitly in the course of training from mentors, instructors, and trainers [5]. It is actually the ultimate currency of our existence. In fact, bad decisions are the main cause of death. Medical decision-making is the process in which a diagnosis or treatment plan is formulated from the available information and tests results, often with incorporation of known patient preferences.

I.5.1 Main characteristics of medical decision making

The main characteristics of medical decision making has been summed up in [5] as follows:

- *Rationality:* the main characteristic of the perfect decision maker. Decision making should be logical, follow the laws of science and probability and lead to corresponding solutions.
- *Critical thinking:* as essential component of rationality. Learning the basic elements will mitigate some aspects mistakes of rationality, facilitate improvements and lead to better thinking.
- *Cognitive and affective biases:* Major problem for all decision-making in all areas of human behavior, medicine is no exception. Understanding and detecting bias is an important feature of good medical decision making.
- *Metacognition:* A broad strategy for thinking about thinking. Just as it is a sign of cognitive development and cognitive performance.
- *Reflection:* it requires a conscious looking and thinking about our actions and feelings and then making interpretations.
- *Mindfulness:* personal awareness of oneself, and in this case, the duties to a patient.
- *Communication:* the most important process in development decision making, a lot of specific biases can be involved in this exchange of information.
- *Ordering and interpretation of appropriate investigations*: this aspect of decision making is often underestimated. Sometimes, tests are

[1] Medicine is the science and practice of establishing the diagnosis, prognosis, treatment, and prevention of disease.

ordered without thinking. Yet test ordering is a critical part of the pre-analytic laboratory stage, non-judicious ordering may lead to lot of errors.

- *Patient preferences:* No decision about the patient should be made without the patient. The patients must be fully involved in decisions that are made about their health. Any decision about a patient needs active engagement and, whenever possible, informed input from the patient and/or their caregivers.

I.5.2 Types of Medical Decision Making

According to [6, 7], there are four types of medical decision-making according to CMS [2] and they are explained in the table down below (Table 1):

	Options	**Data**	**Risk of complications and/or morbidity or mortality**
Plain	Minimal	Minimal or none	Minimal
Low complexity	Limited	Limited	Low
Medium difficulty	Multiple	Multiple	Medium
High complexity	Extensive	Extensive	High

Table 1: Types of Medical Decision Making

Medical decision making refers to the difficulty of making a diagnosis and choosing right treatment option. This difficulty is measured by:

- The number of diagnoses and / or the number of treatment options to consider.
- The volume and / or complexity of medical records, diagnostic tests and / or other information to be obtained, reviewed and analyzed.
- Risk of serious complications, morbidity and / or mortality, and comorbidities associated with patient problems, diagnostic procedures, and / or potential treatment options.

The final level of medical decision-making results in the number of diagnoses / treatment options, the amount of data checked / ordered and the level of risk.

[2] CMS: The Centers for Medicare & Medicaid Services and it is part of the United States Department of Health and Human Services.

I.6 Ensemble Based System

In this section, we will focus on the Ensemble Based Systems as a promising technique in DMS and that we choose to use in our solution.

I.6.1 History of Ensemble Based Systems

According to [1], the Ensemble Based Systems starts with Dasarathy an al. paper [8] that talked about partitioning the feature space by using two or more classifiers.

Then in 1990, Hansen and Salamon showed in [9] that by using an ensemble of neural networks with a similar configuration, it is possible to improve the generalized characteristics of a neural network. While Schapire proved in [10] that by combining weak classifiers through boosting, a strong classifier can be generated in probably approximately correct (PAC) sense, the predecessor of the suite of AdaBoost algorithms[3].

Since the appearance of these foundational works, research into ensemble systems has expanded rapidly, appearing frequently in the literature under many creative names. The long list includes composite systems of classifiers, mixture of experts, stacked generalization, consensus aggregation, combination of multiple classifiers, change-glasses approach to classifier selection, dynamic classifier section, classifier fusion committees of neural networks, voting pool classifiers, classifier ensembles, and pandemonium system of reflective agents, among many others. It is just that the paradigms of these approaches usually differ from one another.

I.6.2 Reasons to use Ensemble Based Systems

According to [1], there are several theoretical and practical reasons why we may prefer an ensemble system:

- *Statistical Reasons:* When dealing with any kind of classifiers, the engineer should keep in mind that there is a generalization gap. That

[3] AdaBoost, short for “Adaptive Boosting”, is the first practical boosting algorithm proposed by Freund and Schapire in 1996. It focuses on classification problems and aims to convert a set of weak classifiers into a strong one.

is caused by the data that the classifier has not been trained on. This concludes in, even if the training performance of a classifier is good, the prediction of generalization performance will not be good. If there is not enough similarity between the training data and the future data (the data that the classifier will actually work on in the future). Then the classifier will not have a good performance in the field.

- *Large Volumes of Data:* In some cases, the amount of analyzed data may be too large for efficient processing, so dividing the data into smaller subsets, and training different classifiers with different data parts and combining their outputs using the smart combination rule is a more efficient expedient approach than training one classifier with a huge amount of data.
- *Too Little Data*: Ensemble systems can also solve the exact opposite problem - having too little data. To successfully learn the underlying data distribution, the availability of an adequate and representative set of training data is of paramount importance for a classification algorithm. Moreover, in the absence of adequate training data, resampling techniques can be used to draw overlapping random subsets of the available data, each of which can be used to train a different classifier, to create ensemble.
- *Divide and Conquer:* regardless of the amount of data available, some problems are too difficult to solve, because the decision boundary that separates the data from different classes may be too complex or lie outside the space of functions that can be implemented by the selected classifier model.
- *Data Fusion*: dealing with heterogeneous features, one cannot use one classifier to work with the information contained in all data. And some diagnostics may require several tests. Each test generates data with a different number and type of features that cannot be used together to train a single classifier. In these cases, data from each test can be used to train different classifier, the results of which can then be combined.

I.6.3 Diversity of Ensemble-Based Systems

According to [1], due to noise, outliers, and overlapping data distribution, it is impossible to have a classifier with perfect generalization performance. In this regard, a solution can be: creating many different classifiers and combining their outputs so that the combination improves the performance of single classifier and this is a strategy of ensemble based systems. Only this requires ensuring that each of them is unique. Once uniqueness is guaranteed, the errors made by each of them can be ignored because the errors will refer to different instances of the data, and therefore the strategic combination can reduce each error. In the end, combining their outputs of these classifiers will be much more performant than training a single classifier with all the diverse data available.

Such a set of classifiers is called diverse. The diversity of classifiers can be achieved in different ways. The one of the most popular methods is to use different training datasets for training individual classifiers. Obtained using resampling techniques such as bootstrapping or bagging, where subsets of training data are randomly selected, usually with replacement from all training data. This is illustrated in Figure 1, where random and overlapping subsets of the training data are selected to train three classifiers, which then form three different decision boundaries. These boundaries are combined to obtain a more accurate classification.

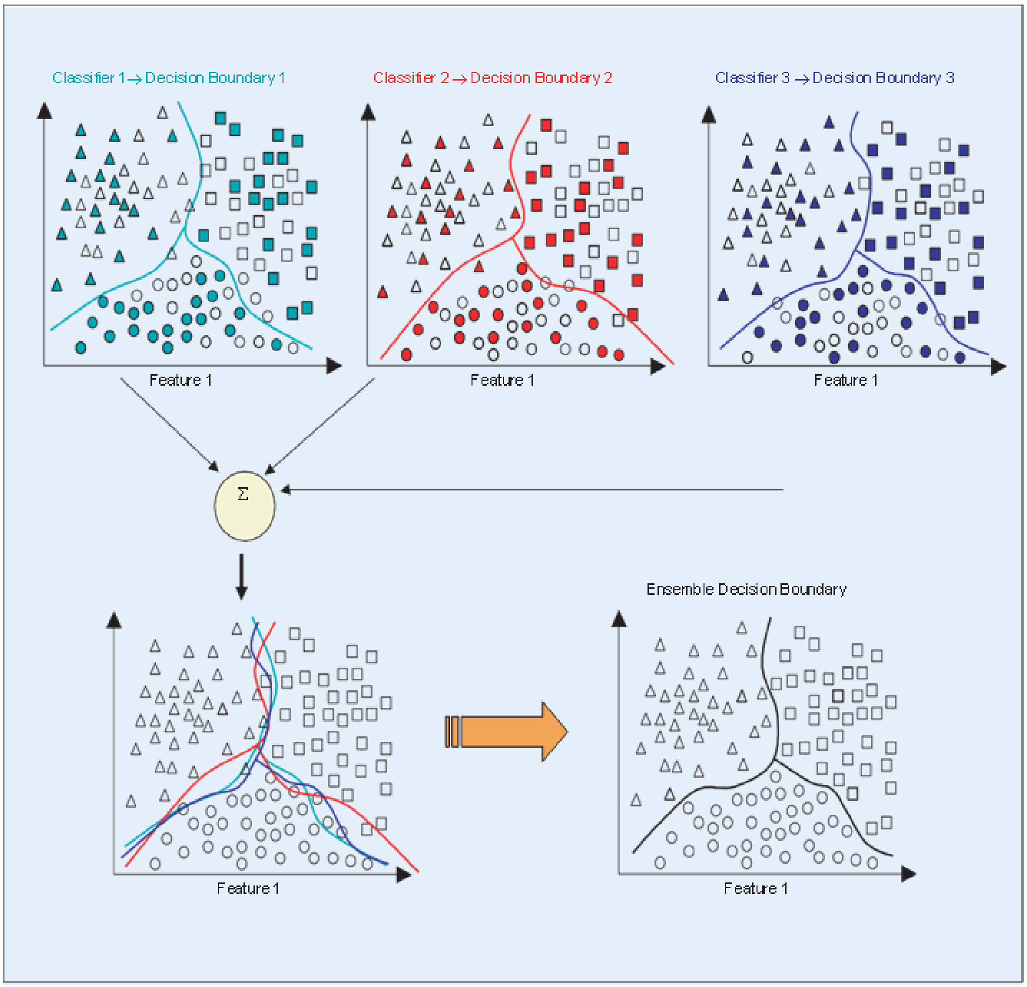

Classifier 1 → Decision Boundary 1
Classifier 2 → Decision Boundary 2
Classifier 3 → Decision Boundary 3
Feature 1
Feature 1
Feature 1
Σ
Ensemble Decision Boundary
Feature 1
Feature 1

Figure 1: Combining classifiers that are trained on different subsets of the training data [1].

If the training data subsets are drawn without replacement, the procedure is also called jackknife or k-fold data split: the entire dataset is split into k blocks, and each classifier is trained only on k-1 of them. A different subset of k blocks is selected for each classifier as shown in Figure 2.

To ensure the difference between individual boundaries, unstable classifiers are used as base models. If subsets of the training data are drawn without replacement, the procedure is also called a jackknife or k-fold splitting of the data: the entire dataset is split into k blocks, and each classifier is trained on only k-1 of them. For each classifier, a different subset of k blocks is selected, as shown in Figure 2.

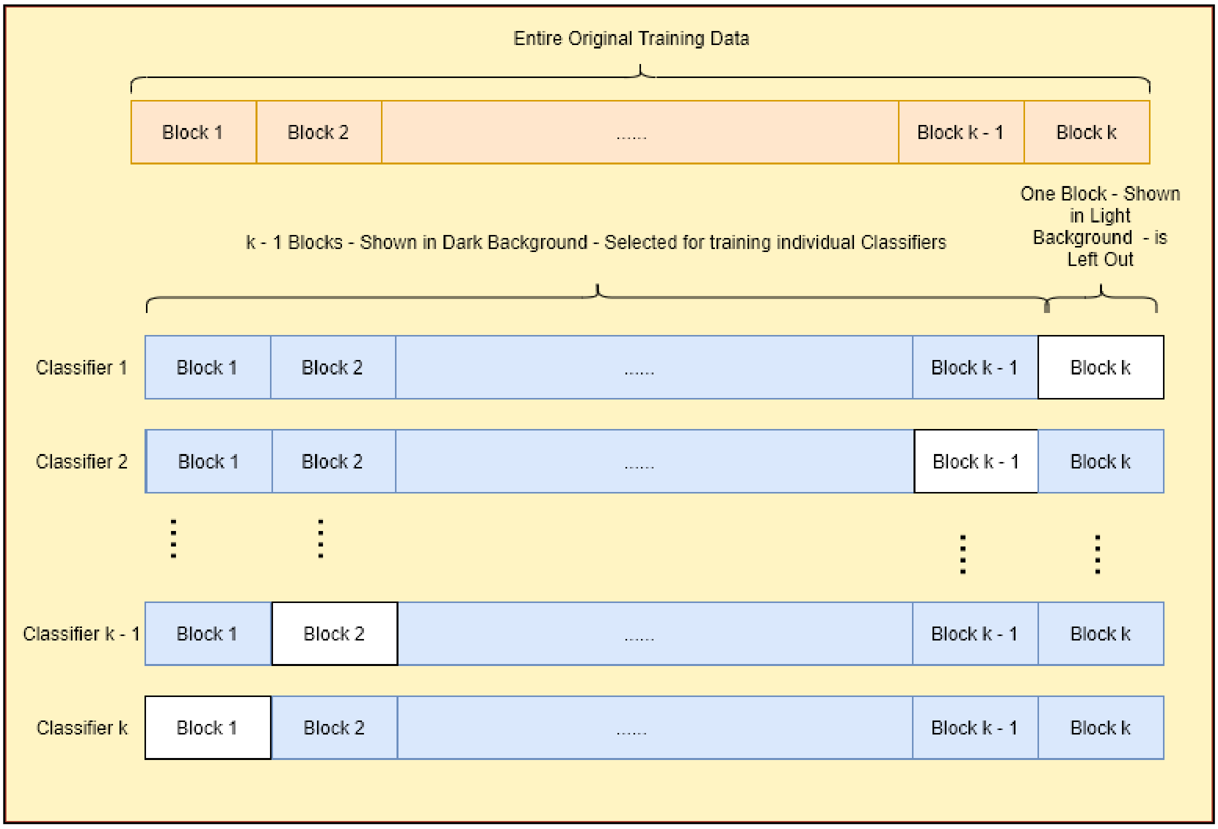

Entire Original Training Data
Block 1
Block 2
......
Block k - 1
Block k
k - 1 Blocks - Shown in Dark Background - Selected for training individual Classifiers
One Block - Shown in Light Background - is Left Out
Classifier 1
Classifier 2
Classifier k - 1
Classifier k

Figure 2: k-fold data splitting for generating different, but overlapping, training datasets [1].

Another approach to achieving diversity is to use different training parameters for different classifiers. As a series of multilayer perceptron (MLP) neural networks can be trained using different initializations of weights, number of layers / nodes, error goal, etc. Adjusting parameters allows to control the instability of individual classifiers and contributes to their diversity. The ability to control the instability of neural networks and decision tree classifiers makes them suitable candidates for use in ensemble setting. Alternatively, very different classifiers such as MLPs, decision trees, nearest neighbor classifiers, and support vector machines can also be combined.

Diversity is usually achieved by resampling the training data, since this procedure is simpler, however, the combination of different models is only used for specific applications that warrant them.

Finally, diversity also can be achieved through different features. In fact, generating different classifiers using subsets of random features is known as the random subspace method, and it has found widespread use in certain applications.

I.6.4 **Basic Algorithms of Ensemble Based Systems**

When it comes to creating an ensemble-based system, it is important to know its basic algorithms [1]. Since this determines the diversity of classifiers and, therefore, affects the performance of all the system.

a) **Bagging:**
Breimans' bagging is one of the first ensemble-based algorithms, short for bootstrap aggregation. That uses bootstrapped replicas of training data to achieved the diversity:

- Different training data subsets are randomly drawn—with replacement—from the entire training data.
- Each training data subset is used to train a different classifier of the same type.
- Individual classifiers are then combined by taking a majority vote of their decisions.

When the available data is limited in size this system works well. Because it ensures that there are enough training samples in each subset by drawing large portions of the sample (75% to 100%) in each subset. This significantly overlaps individual training subsets. A relatively unstable model is used to provide diversity. It gets different decision boundaries for small perturbations in different training datasets. Neural networks and decision trees are good candidates for this purpose, since their instability can be controlled by choosing their free parameters.

b) Pasting Small Votes

It is a variation of the bagging algorithm that is designed for to work with large dataset. This algorithm can be summed up in the following points:

- Trains the classifier with bites (smaller subsets of a larger dataset).
- Each classifier focuses on the most information-rich instances for the current ensemble member.
- Examples that improve diversity are important.
- The classifier is trained on a balanced dataset (contains simple and complex instances).
- A classifier is called non-standard if it has not used a specific instance in its training.
- If the instance x is classified correctly, it still fits into the training set, but only with probability εt / $(1 - \varepsilon t)$, where $0 <\varepsilon t<1/2$ is the classifier error.
- If instance x is incorrectly classified by a simple majority of the current ensemble, it is automatically placed in the training subset of the next classifier.
- In the opposite case, a probability is added to it and then it is placed in the training set.
- To ensure that each classifier can provide significant performance individual classifiers are expected to perform at least 50%.

There are two options for pasting small votes have emerged: one that creates subsets of data at random, called Rvotes, and one that creates consecutive datasets based on the importance of the instances, called "Ivotes."

c) Boosting

In 1990, Schapire proved that a weak learner [4]algorithms but can become like the strong learner [5]ones. And he proposed the boosting algorithm that can change a weak learner to a strong one. The similarity between bagging and boosting algorithms is that they both resample the data to create an ensemble of classifiers.

[4] An algorithm is called a weak learner, if his classifiers can slightly do better than random guessing.

[5] An algorithm is called a strong learner, if his classifier can correctly classify all but an arbitrarily small fraction of the instances.

And it works this according to this:

- Creates three classifiers: C1, C2 and C3. Who are weak.
- C1 is trained with a random subset of the available training data.
- The most informative subset of data is given to C2.
- C2 is trained on a training data only half of which is correctly classified by C1, and the other half is misclassified.
- The disagreements of C1 and C2, are collected and given to C3 to train on.
- The three classifiers are combined through a three-way majority vote.

Schapire showed that the error of this ensemble of three classifiers is bounded from above and less than the error of the best classifier in the ensemble, provided that each classifier has an error rate of less than 0.5. For a problem with two-classes, the error rate of 0.5 is the least expected from the classifier, since an error of 0.5 equals a random guess. Therefore, a stronger classifier is created from three weaker classifiers. A strong classifier in the strict sense of PAC learning can be created by recursive boosting applications.

d) Mixture-of-Experts

It is another conceptually similar technique, where the ensemble consists of a set of C1,.... ., CT classifiers, followed by a CT + 1 second level classifier used to assign weights to the serial combiner.

In this model, the combiner itself is usually not a classifier, but rather a simple combination rule such as random selection (from weight distribution), weighted majority, or weighted winner-take-all principle (Figure 3).

However, the weight distribution used for the combiner is determined by a second level classifier, usually a neural network called a gating network. The gating network is trained either by standard gradient descent backpropagation or, by the expectation maximization (EM) algorithm. In any case, the inputs to the gating network are the actual training data instances themselves (as opposed to the output of the first level classifiers for composite generalization), hence the weights used in the union rule are instance specific, creating a dynamic union rule.

A mixture of experts can be viewed as an algorithm for selecting a classifier. Individual classifiers are experts in some part of the feature space, and the combination rule selects the most appropriate classifier or classifiers, weighted by their experience, for each instance x.

The pooling system can use weights in several different ways: it can choose single classifier with the highest weight, or it can calculate a weighted sum of the output of the classifier for each class and choose the class that will receive the highest weighted sum. The latter approach requires the output of the classifier to have continuous values for each class.

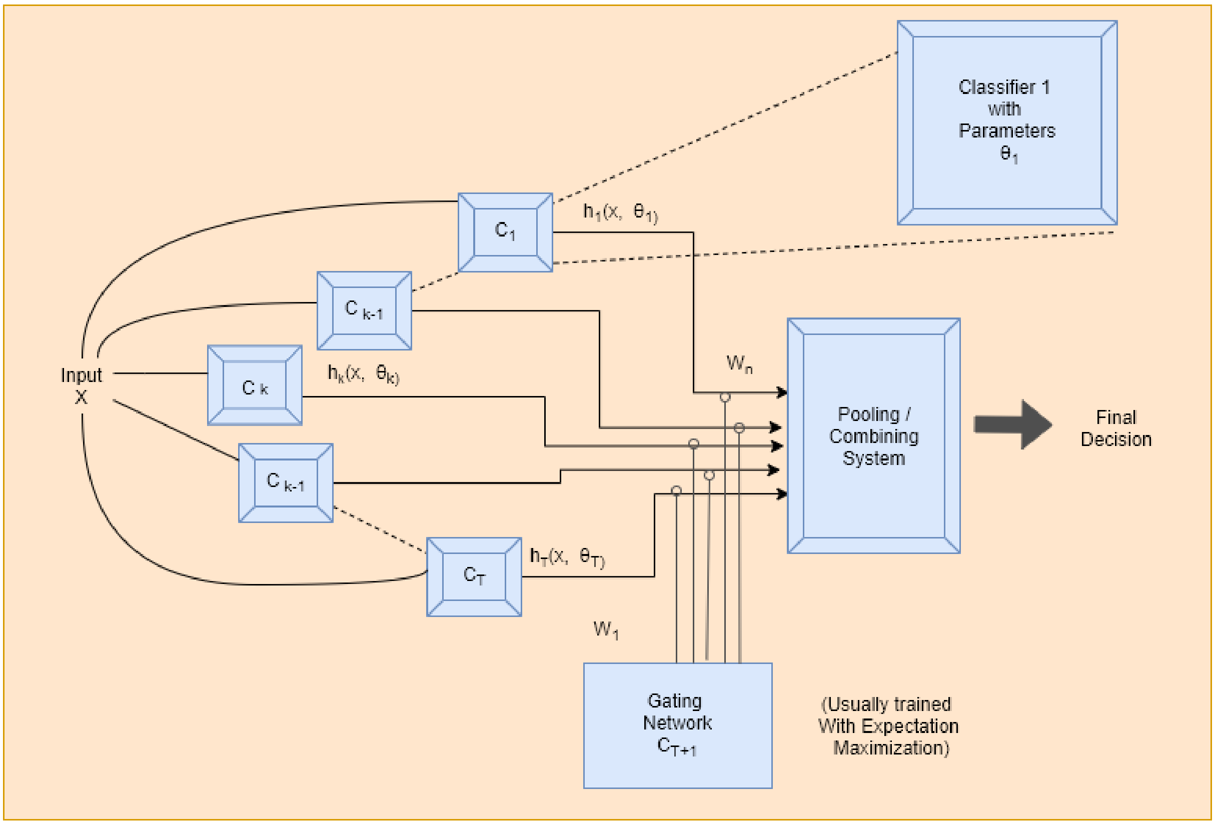


Figure3: Mixture-of-experts [1]

I.7 Decision making systems in medical field

In this section we will talk about some decision making systems in the medical field:

I.7.1 AI systems for complex decision-making in acute care medicine

Authors in [11] state that the integration of AI into the emergency care system opens up a new source of intelligent thinking and offers great potential for synergies between AI systems and human intelligence. This transition is inevitable due to advances in technology, the process of developing a formal plan to prevent the need to monitor patient care using computers is no longer delayed. AI assistance is important in the emergency department because it detects complex relational time series patterns in datasets, and at this level of analysis goes beyond the usual threshold-based analysis used in hospital protocols in use today.

And to get good results, first the experts must take over the oversight of AI, but there is a risk that the current role of doctors and nurses of emergency care could be overwhelmed by computers. But doctors and nurses need to oversee the shift to AI because there are significant risks, and much of that risk is unique to the medical field and outside the experience of AI designers. The addition of AI can bring new communication challenges associated with high error rates and patient safety requirements for handover, which include full transparency that must be maintained.

I.7.2 Data assimilation and multisource decision-making in systems biology based on unobtrusive Internet-of-Things devices

From the [12], we notice an interesting use of device based on Internet of Things (IoT) with continuous monitoring of human. And those devices can be very useful if they are based on a new data assimilation process that fuses multiple data scales from several sources to provide diagnoses. And those required technologies are ready to support the desired disease diagnosis levels, such as hypothesis test, multiple evidence fusion, machine learning, data assimilation, and systems biology.

But to complement the physical limitations of those portable devices advanced data processing technologies and interdisciplinary integration must be developed and new data assimilation processes allow multiple data scales from different sources to be combined to provide the desired accuracy for diagnosis or medical decision making.

However, different approaches to assimilating data in systems biology still need to be tried, and pervasive sensory research can help figure out the patients' physical condition through small, unobtrusive, long-term measurements.

I.7.3 An Ensemble based Decision Support Framework for Intelligent Heart Disease Diagnosis

In [13], we can find another experience on how to improve the accuracy of disease diagnosis in medical health. Researchers have used several statistical analyzes and data mining techniques because heart disease is considered the leading cause of death worldwide over the past 10 years. The proposed framework uses ensemble-based classifiers to combine different data mining classifiers. A standard dataset of heart disease from the UCI[6] repository was used for training and testing. Three classifiers such as Naïve Bayes, DT-GI and SVMs were used to predict a patients' heart disease and the proposed ensemble-based technique gave good and encouraging results.

Before the model was built, inconsistencies and missing values had to be corrected and the majority vote method was used to calculate and predict heart disease. Moreover, those techniques can be extended to determine the degree and intensity of heart disease, and the same framework can be used to for multiple diseases prediction too.

Table 2, down below, summarizes the most important points of the above-mentioned works:

[6] Union Cycliste Internationale (UCI) is the world governing body of cycling.

Work	Kind	Purpose	Synthesis
AI systems for complex decision-making in acute care medicine	AI system	Integration of AI into emergency medical care.	e) Doctors and nurses need to oversee the shift to AI because there are significant risk is unique to the medical field. f) Doctors and nurses could be overwhelmed by computer AI systems. g) Challenges associated with high error rates and patient safety requirements.
Data assimilation and multisource decision-making in systems biology based on unobtrusive Internet-of-Things devices	IOT	Use IOT devices for disease diagnosis, Data assimilation and multisource decision-making in systems biology.	h) Use different data assimilation process and technology that fuses multiple data scales from several sources to provide diagnoses. i) Physical limitations of IoT devices.
An Ensemble based Decision Support Framework for Intelligent Heart Disease Diagnosis	Clinical decisionsupportframework	Use Naive Bayes, DT-GI and SVMs classifiers to predict a patients' heart disease as an ensemble based technique.	j) Inconsistencies and missing values had to be corrected. k) Majority vote method was used to calculate and predict heart disease l) Need to use other techniques to determine the degree and intensity of heart disease and/ or for multiple diseases prediction as well.

Table 2: Decision making systems in medical field

I.8 Main Clinical Decision Support System (CDSS) Companies by Ambulatory, Inpatient Settings

To reduce patient harm and optimize health outcomes, ambulatory and inpatient providers are switching to Clinical Decision Support Systems technology. While the needs in ambulatory and inpatient settings differ and vary, some clinical decision support system providers demonstrate the capabilities to meet the demands of both branches of this industry.

According to [14], the top five clinical decision support system for ambulatory and inpatient settings are:

- **First Databank:**

  It provides informative messages through alerts in existing applications. Named the highest-rated drug database by KLAS, First Databank is currently in use at thousands of ambulatory care facilities around the world. The vendors' clinical decision support technology specializes in installations of e-prescribing systems and prioritizes delivering concise, immediate electronic messages offering up-to-date drug information.

- **Medispan**

  It offers point-of-care clinical decision support with fast, real-time online and mobile applications to help ensure patient safety and improve health outcomes. It provides detailed medication classifications that exceed mandatory industry standards to meet federal regulatory demands and market needs and reduce prescription errors.

- **Allscripts**

  Designed to make it easier to search for patient health records, also offers clinical decision support tools for various medical departments including acute, ambulatory, emergency and surgical care because it is adapted to fit any conditions and setting.

- **Cerner**

  Uses a vetted set of evidence-based standards and criteria to give reliable guidance to ensure patients receive the proper treatment for their specific needs. It offers clinical decision support from advanced imaging and radiology to mobility and provides with up-to-date information to ensure accurate ordering and prescribing.

- **Elsevier**

As a worldwide publisher of scientific, technical and medical information products and services, it offers a set of tools and provide answers to any clinical question to help clinicians at the point of care. Also Elsevier equips providers with a variety of tools, from drug information to learning and competency management to improve patient outcomes for pharmacists, doctors and nurses.

We summarized the above-mentioned information in Table 3:

Workname	Function	Domain
FirstDatabank	Deliver useful medication- and medical device-related information.	Information about medication- and medical device.
Medispan	Clinical decision support with real-time online and mobile applications.	Informationaboutmedication.
Allscripts	Management of patient's health record.	Management and electronic health record technology.
Cerner	Ensure that patients receive the proper treatment for their specific needs.	Health information technology services and devices.
Elsevier	Provide answers to any clinical question to help clinicians at the point of care	Information in scientific and medical content and competency management.

Table 3: Main Clinical Decision Support System Companies by Ambulatory, Inpatient Settings

I.9 Conclusion

Over the last decade, the ensemble-based systems have enjoyed a growing attention and popularity due to their many desired properties, and the broad spectrum of applications that can benefit from them. In this chapter, we discussed the fundamental aspects of these ensemble systems, including the need to ensure —and ways to measure— diversity in the ensemble.

Whereas there is no single ensemble generation algorithm or combination rule that is universally better than others, all of the approaches have been shown to be effective on a wide range of real world and benchmark datasets, provided that the classifiers can be made as diverse as possible.

One last example of an expert system is MYCIN, an application to perform medical diagnosis. In the MYCIN example, the domain experts were medical doctors and the knowledge represented was their expertise in diagnosis, in other words this expert system is the representation of the doctors' knowledge, implementing knowledge engineering techniques, and this will be the core of our next chapter.

Chapter II:

Solution Modeling

I.1 Introduction

Until now, we have seen and understood general (and some detailed) concepts about: decision making systems, there is only one concept remaining: “Clinical Aid” and it will be the core of this chapter.

Indeed, the purposes of this chapter is to model and design our solution (mobile platform) that helps at Medical Decision Support from Audio Instructions. In this chapter, we will go over some of the main concepts in the medical field, starting by “the sign” and finishing by “the diagnoses”.

I.2 Medical Concepts

In this section, we will present some definitions about the medical concept as found in [15].

I.2.1 The Sign

It is an objective manifestation of a pathology noted by a doctor in human medicine and called "sign" or "a set of signs". In other words, the description of the state of a patient that can appear at different levels of observation and there are two types of signs:

1. **Positive sign:** A positive sign is the sign whose presence is significant as: high fever, nasal pain and cough are flu-like illness.
2. **Negative sign:** A negative sign also called missing sign is all the signs expressed by negation like: no dizziness, no fever, etc.

I.2.2 The Symptom

A symptom is: “*a physical feeling or problem that shows that you have a particular illness*”[7]. A symptom is one of the subjective manifestations of a disease or a pathological process expressed by a patient. And for a given pathology there can be multiple symptoms. The symptoms differs from the signs since they are the subjective expression of the effects that a patient

[7] According to the Cambridge dictionary.

experiences, while a sign is an objective expression inferred by the person making the diagnosis like a doctor.

I.2.3 The Disease

The disease refers to a set of changes due to which the poor body functions. A patient is a person who suffers from a disease that requires treatment. The disease should not be confused with a disability, syndrome, or injury. Many factors can cause disease in humans and they can be internally or externally related to the body.

I.2.4 The Syndrome

The syndrome is: "*a combination of medical problems that shows the existence of a particular disease or mental conditions*"[8]. In other words, it is a collection of signs or symptoms that have arisen simultaneously.

I.2.5 The diagnosis

Examination of the symptoms, the identification process and search of the nature of a disease that the patient may have. The medical diagnosis is the examination designed to determine the state of health of a person.

I.2.6 The diagnostic process

"*little attention has been given to the diagnostic process itself*" [16], this was said about the several papers which talked about diagnosis, it does not apply to current project, since it is all about modeling the diagnosis process.

It is an approach based on research of the causes (etiology) and consequences (signs and symptoms) of the condition. It is practiced by the domain experts (doctors and medical practitioners in this particular case), when facing a set of sign/symptoms. But the term itself, is found in two

[8] According to Cambridge dictionary

different fields, in engineering and in medicine, both fields have developed systematic, formal diagnostic processes.

According to [16], (Figure 4) the engineering diagnostic process, which is equated to problem solving, can be summed up in:

- Collection of data.
- Development of casual inference(s).
- Formation of hypotheses.
- Testing of hypotheses.
- Confirmation of diagnosis.

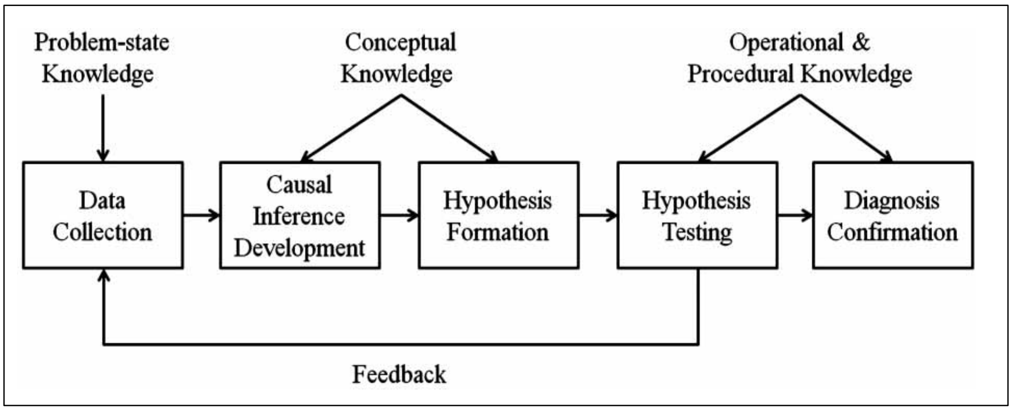


Figure 4: Model of the engineering diagnostic process

But in the medical field, it is described in the following steps:

- Data acquisition.
- Accurate problem representation.
- Differential diagnosis.
- Prioritization of differential diagnoses.
- Tests of hypotheses.
- Review and reprioritization of the differential diagnosis on the basis of the new data.
- Revision of the differential diagnosis.
- Testing the revised differential diagnosis.
- Confirming final diagnosis.

This has been simplified (Figure 5), by combining related steps into five-step process [16] :

- Data collection.
- Data interpretation.
- Differential diagnosis formation and prioritization.
- Diagnostic tests.

- Final diagnosis with feedback.

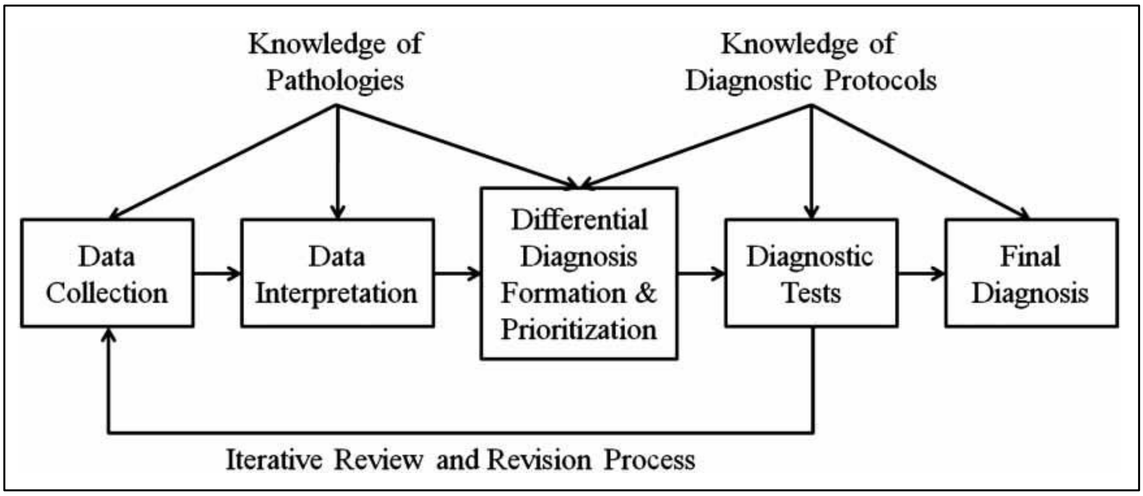

Knowledge of Pathologies
Knowledge of Diagnostic Protocols
Data Collection
Data Interpretation
Differential Diagnosis Formation & Prioritization
Diagnostic Tests
Final Diagnosis
Iterative Review and Revision Process

Figure 5: Model of the medical diagnostic process 39

I.3 The field of Natural Language Processing

Natural Language Processing (NLP) is a branch of artificial intelligence and linguistics dedicated to make computers understand human languages. And it came to existence in order to ease the users' work and satisfy the desire to communicate with the computer in natural language [17].

A language can be defined as a set of symbols and rules. Symbols are combined and tyrannized by the rules to transmit information. And linguistics is the science of language, which includes phonology, which refers to sound, morphology of word formation, sentence structure of syntax, syntax of semantics, and pragmatics, which refers to understanding.

Verily, it all started in in the late 1940s even though the term didn't even exist, but it was a period where the work on machine translation began [17]. Russian and English were the dominant languages. And in 1960, influenced by AI began work with the BASEBALL Q-A system[9], LUNAR[10] and

[9] BASEBALL Q-A system: System is a Chatterbot Question Answering System that answered questions about the US baseball league developed by Green Jr et al in 1961.

[10] LUNAR: One of the first natural language question answering systems built by William Aaron Woods.

Winograd SHRDLU[11] were natural successors of these systems, but they were considered more advanced in terms of their language and task processing capabilities.

In the early 1980s, computational grammar theory became a very active area of research related to the logic of meaning and the ability of knowledge to deal with the beliefs and intentions of the user, as well as functions such as emphasis and themes.

Towards the end of the decade, powerful general purpose sentence processors such as the Core Language Engine [12]and Discourse Representation Theory [13]that offered a means of solving more extended discourse in grammatical -logical framework. This period was a period of community growth. Practical resources, grammars, tools and analyzers became available.

Speech Recognition and Message Understanding (Information Extraction) conferences were designed not only for the task they were solving, but with an emphasis on heavy evaluation, which started a trend that became a major feature in the 1990s. The work on user modeling was one of the directions in research work and same for using rhetorical schemes to create both linguistically coherent and communicatively effective text.

Some NLP research has highlighted important topics for the future works, such as word disambiguation, probabilistic networks and vocabulary work.

As for recent research, they are mainly focused on unsupervised and supervised learning algorithms.

[11] Winograd SHRDLU:An early natural language understanding computer program, developed by Terry Winograd at MIT in 1968–1970.

[12] Core Language Engine: It is a domain independent system for translating natural language (English) sentences into formal representations of their literal meanings which are capable of supporting reasoning.

[13] Discourse Representation Theory: It is a framework for exploring meaning under a formal semantics approach.

I.4 Natural Language Processing and medicine

NLP is also used in medicine. The Linguistic String Project-Medical Language Processor is a good example of the large-scale NLP projects in the medical field. LSP-MLP helps extract and summarize information about any signs or symptoms, drug dosages, and response data in order to identify possible side effects of any drug by highlighting or flagging data items. Also the National Library of Medicine is developing a System of Specialists that expected to function as an information extraction tool for biomedical knowledge bases such as Medline abstracts [14] [17].

Moreover, the Hospital Center of the Geneva Canton Hospital is working on an electronic archiving environment with NLP functions. In the first step, medical records were archived and a proper NLP system called RECIT was developed with using a technique called proximity handling. The challenge was to implement a robust and multilingual system capable of analyzing / understanding medical sentences and storing knowledge as free text in a language-independent representation of knowledge. Columbia University of New York has developed an NLP system called MEDLEE (Medical Language Extraction and Coding System), which identifies clinical information in descriptive reports and converts textual information into structured representation.

I.5 Problematic

Usually, experienced doctors do not have any difficulties dealing with a patient’s condition, no matter how complex it looks, since they can consult each other to combine their knowledge (Figure 6).

[14] MEDLINE (Medical Literature Analysis and Retrieval System Online, or MEDLARS Online) is a bibliographic database of life sciences and biomedical information.

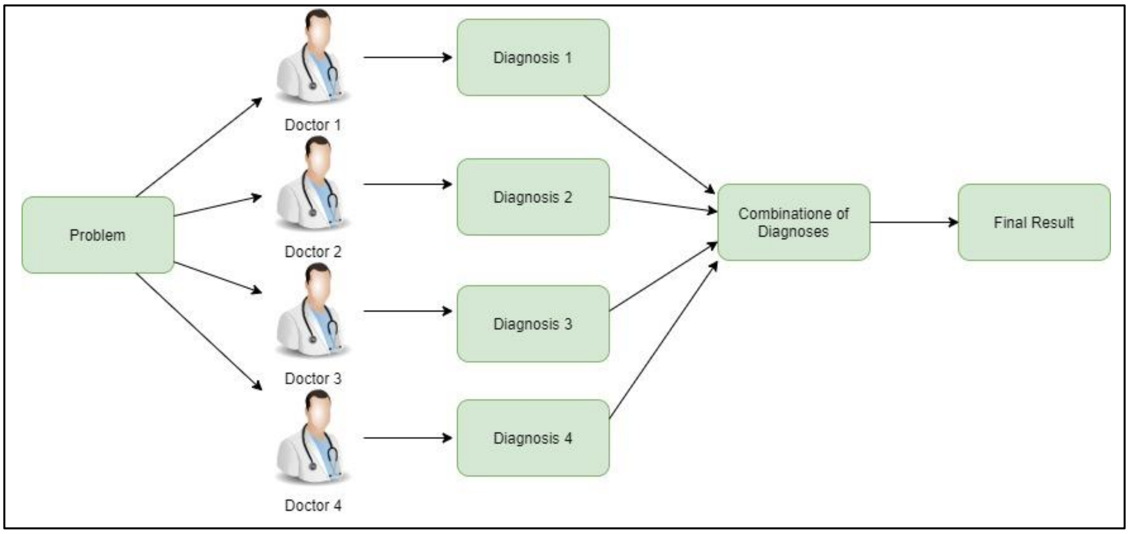


Figure 6: Diagnoses Combination.

In the same conditions, in the case of making a report either of an x-ray image result, or a diagnosis brief, the process can be summed up in Figure 7.

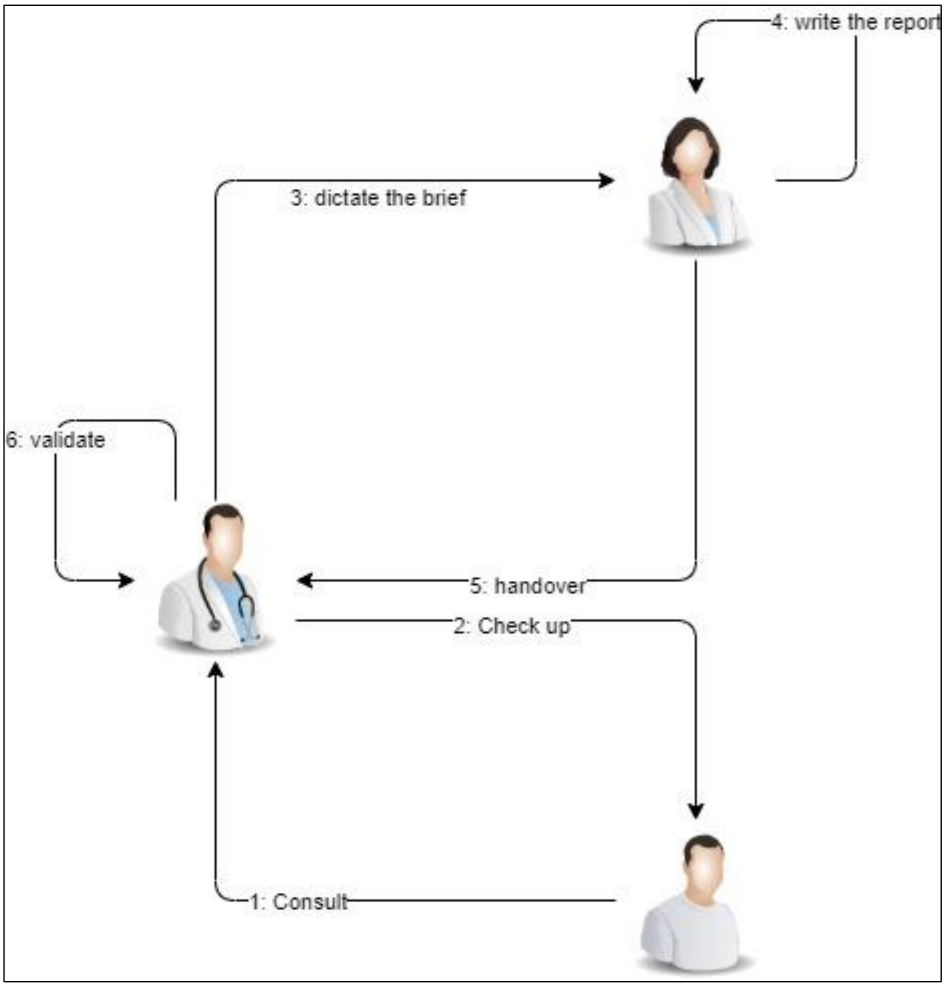


Figure 7: Old Report Making Process

After looking at the situation of the medical section in Algeria, on the lack of assistants and humble in-field-experience that most doctors may have as they will be at the beginning of their careers (Civil service) it was all as a result of the doctor being the only actor in the whole process (Figure 7).

I.6 Proposed solution

As we said previously, our solution is inspired by Ensemble-based expert systems, before a detailed explanation of our solution, we introduce hereafter, the general architecture of the system.

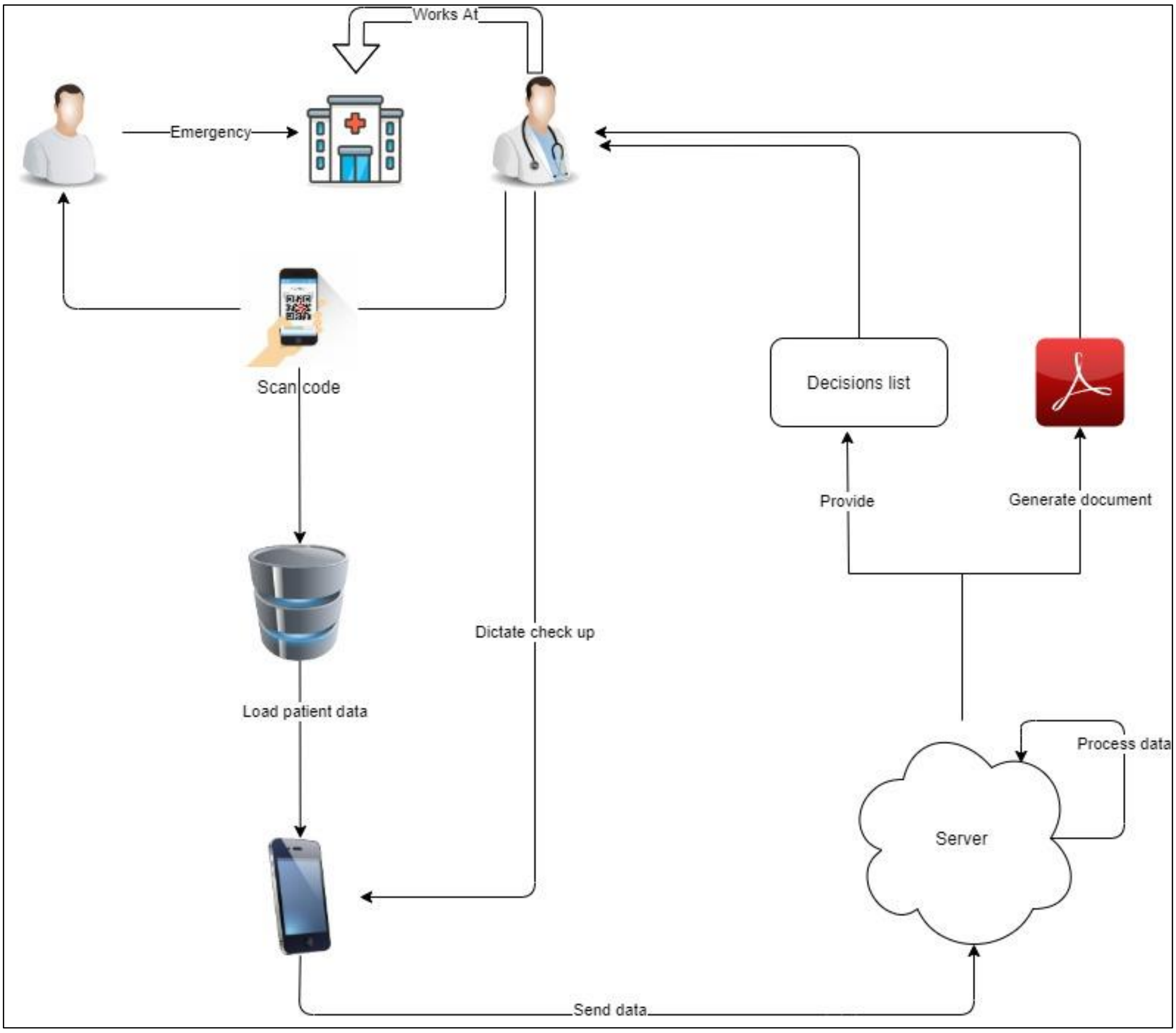


Figure 8: General System Architecture.

The system coordinates the interactions between doctor and patient, and facilitates (barely eliminate) the bureaucracy process (a patient’s data is loaded with a simple QR code scan).

It also simplifies the document (report/prescription) generation process to a single tap (dictation). And when it comes to decision making, the server will do the required job, with the combination of the implemented inference engines.

As it can be seen from the figure above (Figure 8), the general process of our solution can be broken up to six steps:

- Step 1: patient getting in the emergency room.
- Step 2: Scanning QR Code.
- Step 3: Loading patient Data.
- Step 4: Checking up the patient.
- Step 5: Dictating the check-up.
- Step 6: Sending the data to the server so he can process the data.
- Step 7: The server will provide the decision list and generate pdf document.

Finally the doctor will receive these results that will help him in his decision making process.

We will go in the details of those steps further in this chapter.

I.6.1 Knowledge Base

"*A knowledge base (KB) is a technology used to store complex structured and unstructured information used by a computer system".*

The knowledge base, is a repository that has information stored, organized and prepared to be shared. Among the 6 types of knowledge bases: Internal, Hosted, self-hosted, customer, open-source, external. An internal knowledge base is going to be used for the future work. As it offers speed access, security, privacy and gives more control.

The first KB (Figure13), will contain the rules between symptoms and pathologies, in other words, it will describe the set of symptoms that a pathology has.

The second KB (Figure 14), will contain the criteria (age, gender and region) of a pathology.

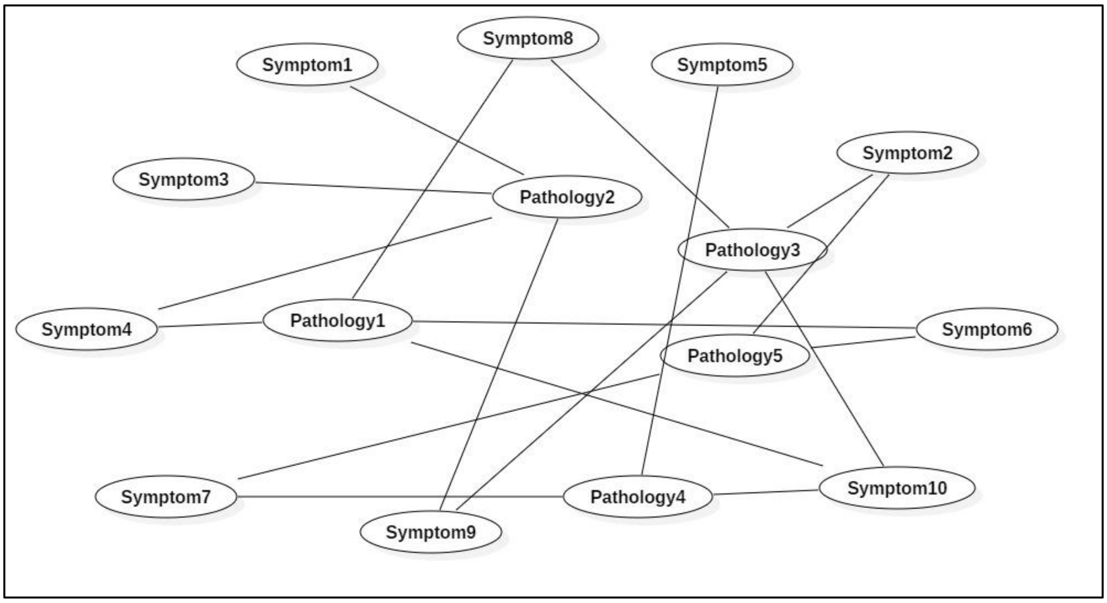


Figure 9. Knowledge Base 1

To simplify the concept of this KB, it cannot be imagined as a network of pathologies and symptoms interconnected. As an example, let's take the symptoms: fever, headache, nausea, muscle pain vomiting, chills and skin rash. As for the pathologies, the most common cases for such a set of symptoms would be:

- Influenza connected to: fever, headache, nausea and muscle pain.
- Leptospirosis connected to: fever, headache, nausea, muscle pain and skin rash.
- Malaria connected to: fever, headaches, vomiting and chills.
- The third KB (sub part of Figure14), is the archive database, it contains the past diagnoses cases.

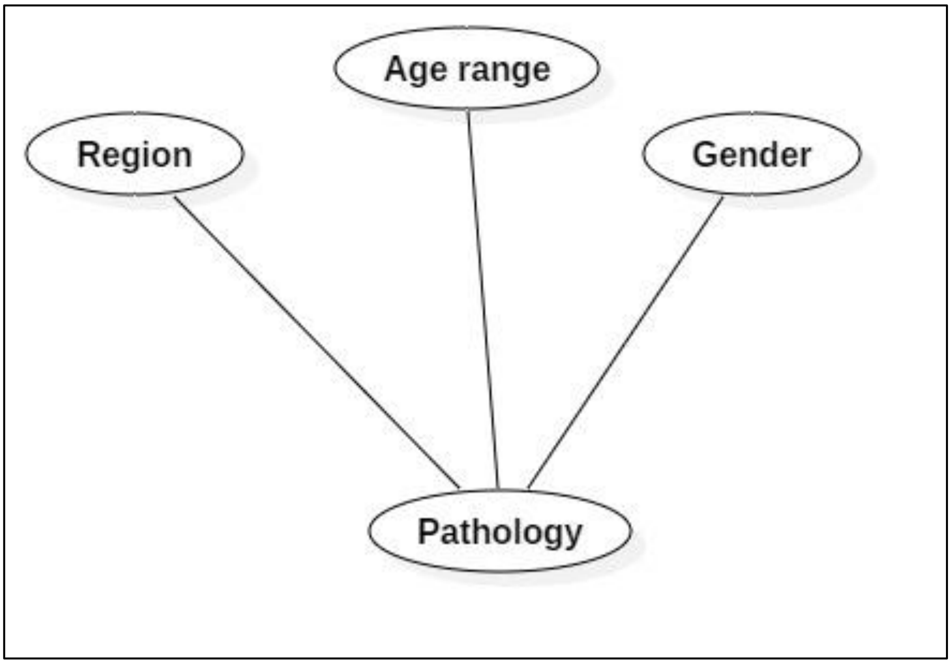


Figure 10: Knowledge Base 2

From the previous diagnosed cases and approved AI generated diagnoses and decisions, This KB is generated. Taking breast cancer as an example, the record inferenced would be: gender = female ; age range = >40 ;region = unspecified.

I.6.2 Expert System

Expert systems (Figure 13) are composed of two parts, Inference Engine (IE) (like in Table. 4) and Knowledge Base (like in Figure 15) (KB).

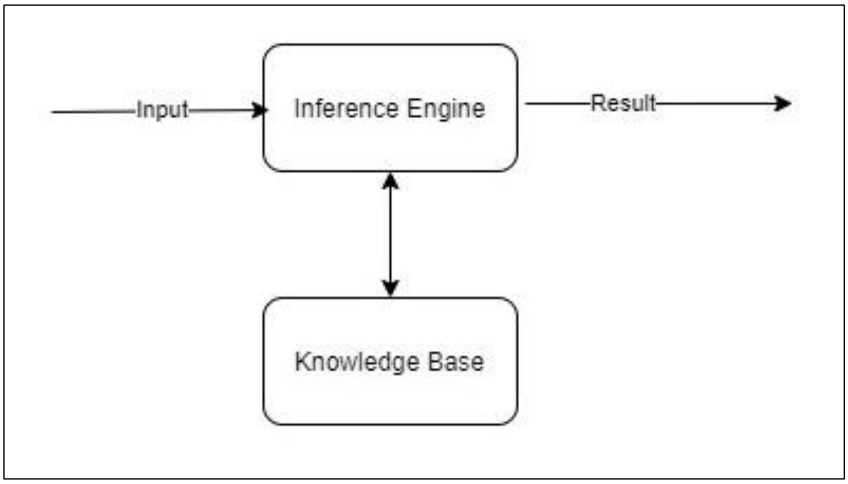


Figure 11: Expert System, adapted from [15]

I.6.3 Inference Engine

In the artificial intelligence field, inference engines provide results from inputs, "it is the component that applies the logical rules to the knowledge base in order to deduce new information".

The first IE, that we will be introducing (Table 4), will manage the symptom-pathology graph, it will go through the list of symptoms mentioned by the observer (doctor), and extract corresponding related pathologies, as described by the rules of its KB, then calculate the pathologies' score, represented by the count of present symptoms, the pathology that has the maximum present symptoms will be the top of the table:

	Fever	**Headache**	**Nausea**	**Muscle pain**
Influenza	True	True	True	True
Leptospirosis	True	True	False	True
Malaria	True	True	True	False

Table 4: Inference Engine 1

As it can be seen from the examples from the table above (Table 4), a pathology can present one or many symptoms.

When the symptoms are mentioned by the observer, they are mixed with some out of context words, since he would be describing a report of a patient. For this issue, a preprocessing operation goes over the transcribed observation text and it:

1. Step 1: Replace punctuations with empty spaces. Example: "Fever, Headache and Nausea." Will become: "Fever Headache and Nausea"
2. Step 2: Tokenize it. Example: "Fever Headache and Nausea" will become "[' Fever ', ' Headache ', ' and ', ' Nausea ']"

3. Step 3: Convert every letter to lower case. Example: “[' Fever ', ' Headache ', ' and ', ' Nausea ']” will become “[' fever ', ' headache ', ' and ', ' nausea ']”
4. Step 4: Remove the stop words. Example: “[' fever ', ' headache ', ' and ', ' nausea ']” will become “[' fever ', ' headache ', ' nausea ']”

As a final step in this preprocessing, we extract the symptoms mentioned in the cleaned data, which the KB already knows about, and then they are passed to the IE, as mentioned above.

The dataset of pathologies, has been adapted from “Ortolang”, it is a thesaurus from the medical vocabulary used until the end of 2014, it was made to index the PASCAL database[15], produced by Inist. Table 4 is a very simple subset of it.

For this project, we have cleaned the corpus manually, to extract the information concisely for our needs, which resulted in having (Figure 12):

- 4 abstract domain concepts (Anatomy, Disease, organism, Biological substance).
- Pathologies names (over 6000).
- Over 20,000 relation between pathologies (a pathology can be the broader concept of a narrower concept pathology).
- Over 4500 leaf specification. (A specification that does not have a narrower specification).

[15] PASCAL is a scientific bibliographic database, which is maintained by INIST (CNRS). PASCAL covers the core scientific literature in science, technology and medicine with special emphasis on European literature.

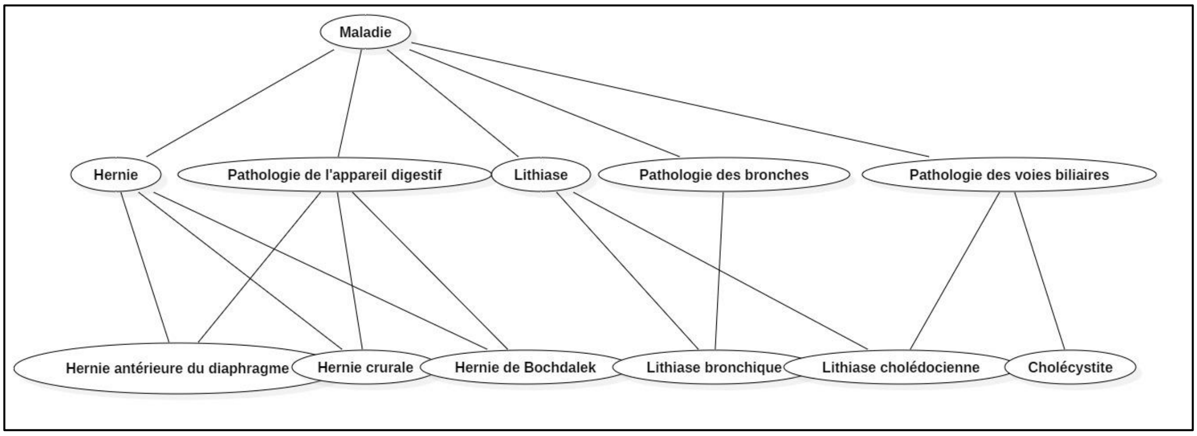


Figure 12: A subset of pathologies dataset

The second IE (Table. 5), will manage other important diagnostic criteria, which are: *gender*, *age* and *region*. It will go through the ordered list of pathologies resulted from the first IE, and apply the rules of its KB (Figure12), each criteria has a coefficient, any coefficient can be mandatory (this is configurable), so after this process if any pathology does not meet the logic of the rules will be eliminated.

The Third IE, will extract the similar archived cases and make a comparison of the results. As for its dataset, it always grows, since the history of patient and diagnoses keep getting updated, therefore it keeps learning new rules each time.

	Patient Age: xx	Patient Gender: x	Patient Region: x
Pathology 3	True	True	True
Pathology 1	False	True	True
Pathology 2	True	False	True

Table 5: Inference Engine 2.

I.7 Requirement specification and analysis

For the issues discussed above, the proposed solution can be broken up in two sub-solution: An Ensemble Based Expert System (EBES) for the clinical aid decision, and a report generator, both of which will be based on voice memos.

For the requirements specification, we will use, UML[16]'s use case diagrams, which are the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation. A key concept of use case modeling is that it helps us design a system from the end user's perspective.

The task's distribution use case to sum up the ideas (Figure 13).

[16] Short for Unified Modeling Language, is a standardized modeling language consisting of an integrated set of diagrams, it has a collection of best engineering practices used in the modeling of large and complex systems.

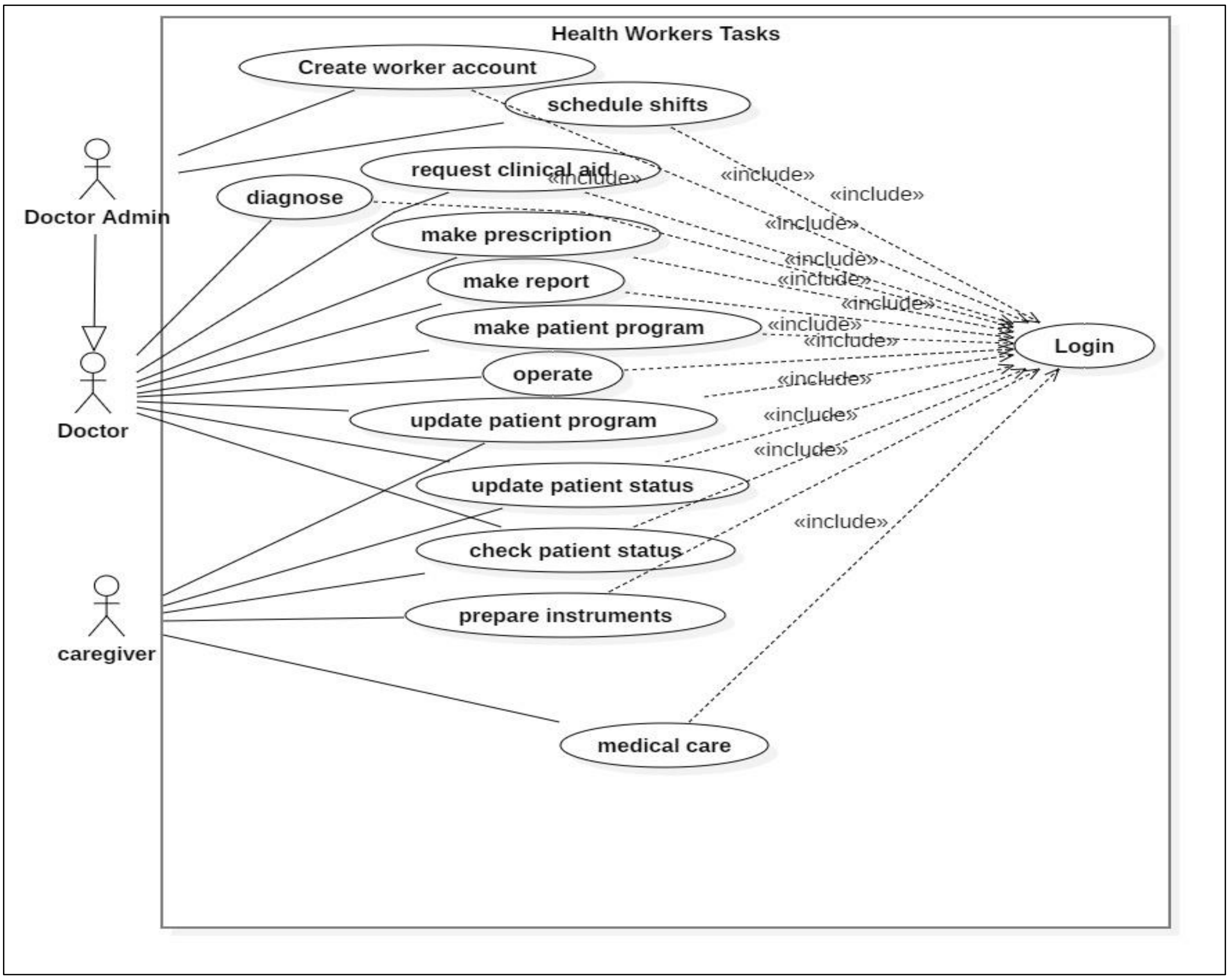


Figure 13: Tasks' Distribution use case.

Table 6 hereafter gives a description of the above diagram.

Use Case	Details
Create worker account	For security purposes the accounts of other workers (doctors, staff, caregivers…) are to be created by a doctor-admin (chief of service).
Schedule shifts	Distribution of working times between different workers, this will make the target not being able to connect to the system out of his shift time.
Login	Access to the given credentials' user's account.
diagnose	The process of diagnosing a patient, and saving it to his history, (the data later will be used anonymously to help with other automatic diagnoses).

Make prescription	The process of generating a prescription document from voice memos.
Make report	The process of generating other types of reports than voice memos (x-rays…etc.).
Request clinical aid	The process of describing the current patient emergency situation over a voice memo, to request an urgent decision aid from the expert system.
Make patient program	The process of describing the treatment a patient should have/follow in the hospital (medicines, injections…etc.).
Update patient program	Updating the patient program (described by a doctor) changing, adding or removing…etc.
Operate	The process of describing the operation received by a patient, to be archived in his history.
Check patient status	See if a patient has any treatments left or have any coming soon.
Update patient status	Update the patient status in the system (no need to complete the treatments, the need of another treatment…)
Prepare instruments	Request/use instruments for different tasks/operations.
Medical care	Provide different medical care services.

Table 6: Use case description.

After analyzing the specifications, we used UML's sequence diagrams, which are interaction diagrams that detail how operations are carried out.

They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent, between whom, and when.

The two main tasks are decision making (summed up in Figure 10) and document generation (summed up in Figure 11) both tasks rely on the transcription process (Figure 14).

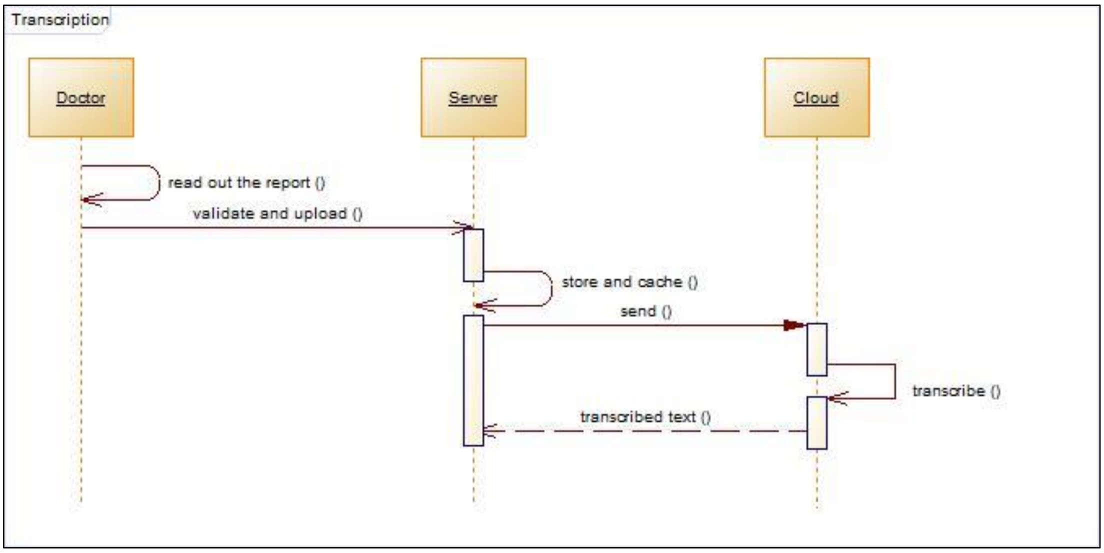


Figure 14: The Transcription Process.

Transcription Process: the process of transcribing audio generated by the user (doctor) in a form of text, using a cloud service.

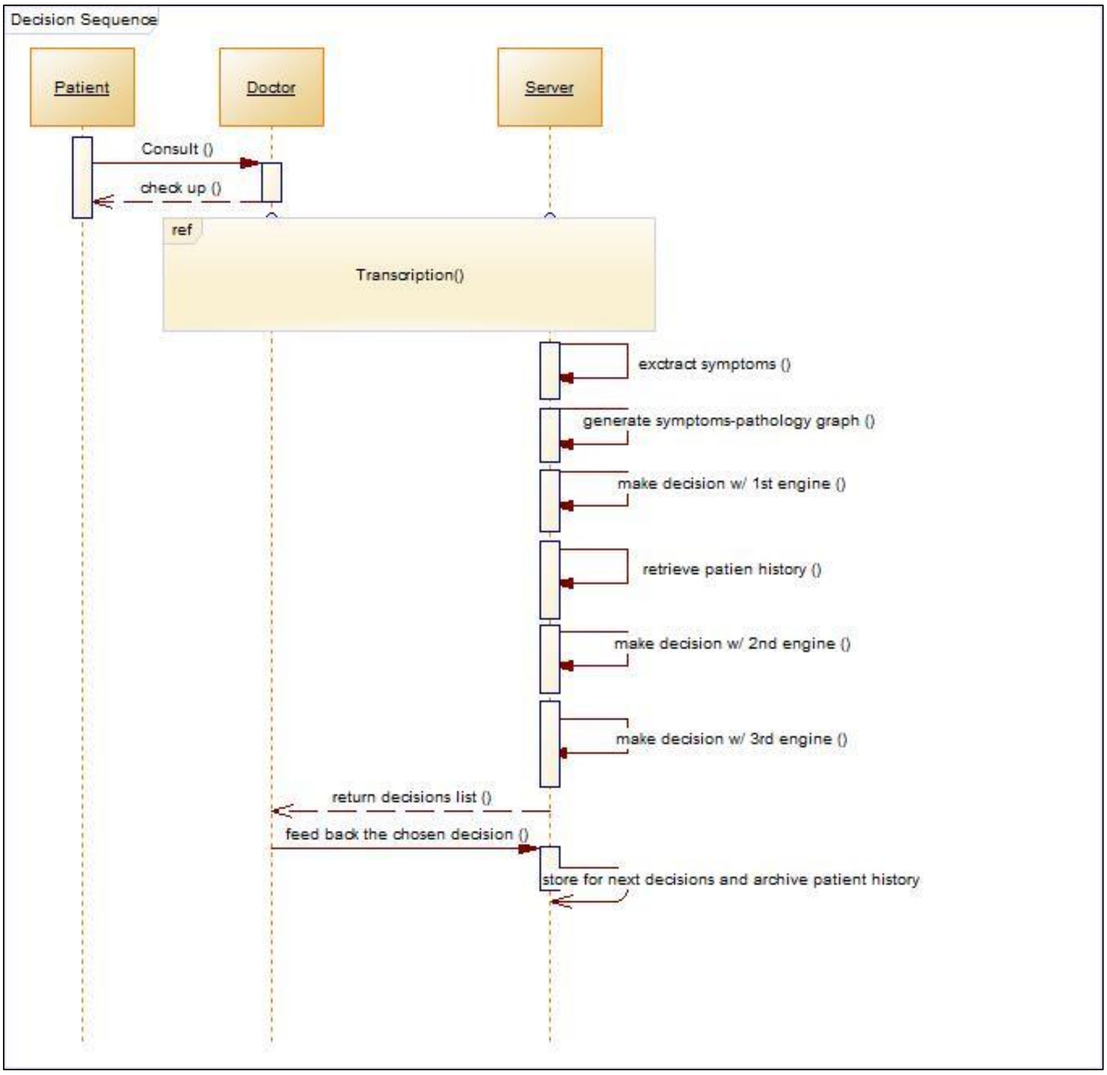

Decision Sequence
Patient
Doctor
Server
Consult ()
check up ()
ref
Transcription()
exctract symptoms ()
generate symptoms-pathology graph ()
make decision w/ 1st engine ()
retrieve patien history ()
make decision w/ 2nd engine ()
make decision w/ 3rd engine ()
return decisions list ()
feed back the chosen decision ()
store for next decisions and archive patient history

Figure 15: The Decision Process.

Decision Making Process: this process includes the transcription process at the start, of course the patient is considered as the triggering action, then goes as follows:

- Symptom extraction from the resulted text.
- Generate Symptom-Pathology graph, from the KB (Fig. 18).
- Provide the graph as input data to the IE (Table. 4).
- Retrieve patient history.
- Provide the first result and the patient’s history as input data to the second IE.
- Compare the current result to similar old diagnoses, to enhance the final result.

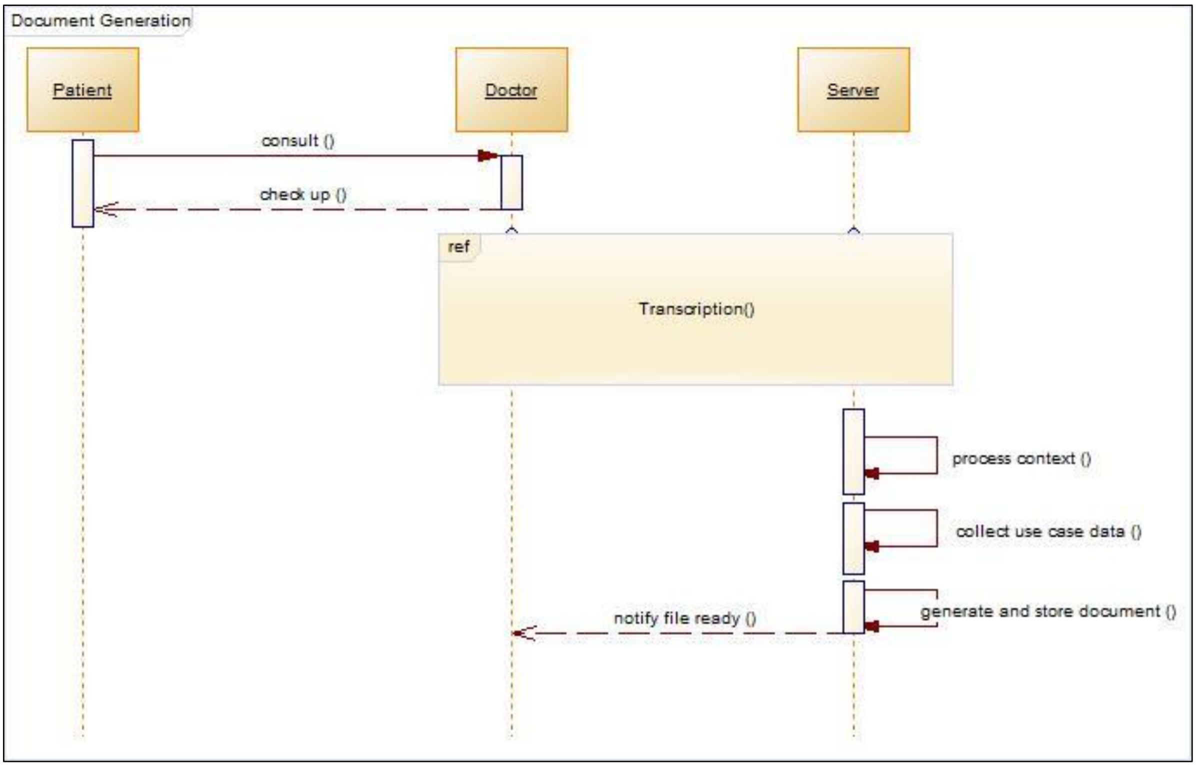


Figure 16: Document generation process.

Document Generation Process: same start at the previous process and then goes as:

- Context[17] processing by extracting domain keywords.
- Updating the existing context scores.
- Updating the existing out of context scores.
- Collect the doctor's data, patient data.
- Generate the corresponding document.
- Store the document's meta-data.
- Notify document is ready.

I.8 System design

To present our system design, we will make use of the well-known class diagram of UML. At the system design stage, the most widely used UML tool is class diagram. It is a type of static structure diagram that describes the

[17] Context analysis in NLP involves breaking down sentences into n-grams and noun phrases to extract the themes and facets within a collection of unstructured text documents.

structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed. The diagram hereafter gives a global view of our project (Figure 17).

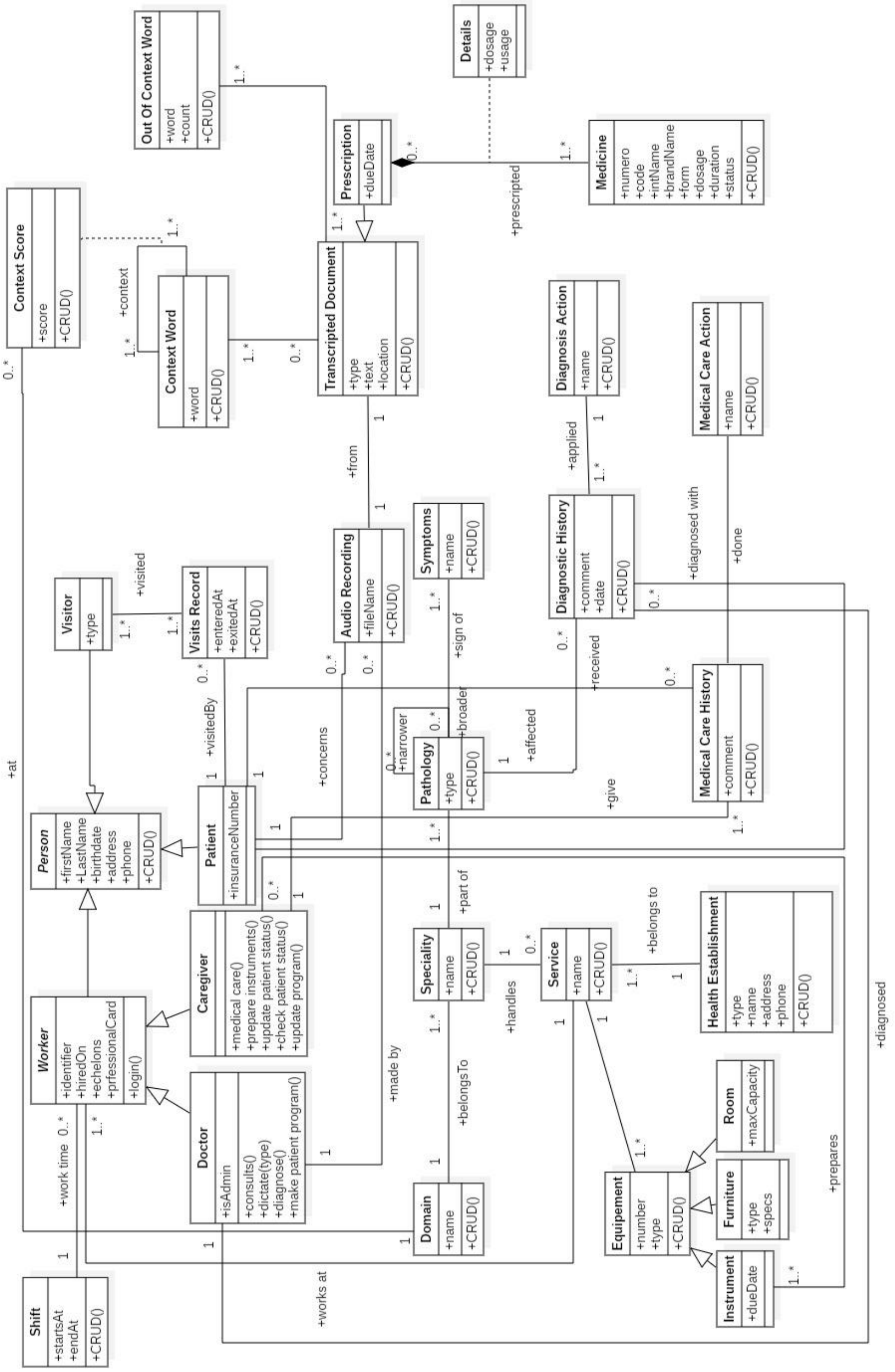


Figure 17.The System's Class Diagram.

Classe	Attributs	Description
Person	First name, last name, birth date, address, phone.	The abstract description of a person in the system.
Worker	Identifier, hired on, echelons, professional card, service.	The abstract description of a worker in the system, specified from the Person class.
Shift	Starts at, ends at	The definition of a work shift, starts and ends at a specified time, which is related in a many to many relationships with the worker. Thanks to this class the users access time to the system can be controlled.
Doctor	Is admin	The specification of a worker as a doctor, if the attribute isAdmin is True, the user will acquire other permission. (Fig.15)
Caregiver		The specification of a worker as a caregiver.
Visitor	type	The visitor type attribute is to distinguish between a regular visitor and a guardian, it is specified from a person.
Patient	Insurance number	The patient is a specification from a person, it will receive different health care and diagnostic actions.
Visits Record	Entered at, exited at	This class relates the patient with his visitors, to manage and controls the flow of visits.
Audio recording	File name	This class archives the recording concerning patients generated by doctors.
Transcripted text	Type, text, location	This class archives the results of transcribing audio recordings, per type (prescription, reports…), and the resulted text.

Prescription	Due date	The specification from a transcribed text, since is has a due date, also it is related to the prescribed medicines.
Medicine	Numero, code, international name, brand name, form, dosage, duration, status	This class holds the medicines records, acquired from the medicine's nomenclature.
Context word	word	The class holds all the words related to the medical domains, these words are used and acquired from/in the transcribed texts.
Context score	score	This class holds the score of the context domain related words, which are the words who appeared together in a transcribed text of the same domain, for each time they appear together, the score is incremented.
Out of context word	Word, count	This class is a specification of the context word class, as the words contained in this one, are standalone.
Diagnosis action	Name	This class's objects are types of the actions that can be performed during diagnostic.
Diagnostic history	comment	This class holds the history of diagnoses of a patient, so it relates all of: diagnosis action, patient, doctor, pathology, symptoms. This represents the KB of the third IE.
Medical care action	Name	This class's objects are the types of action that can be performed by caregivers to patients.

Medical care history	comment	This class holds the history of actions applied by a caregiver to a patient at a specified time (insulin injection, blood pressure…etc.).
Pathology	Name	This class describes the concept of a Pathology.
Symptom	Name	This class stands as a representation of a Symptom.
Domain	Name	This class describes the concept of a medical domain.
Specialty	Name	This class describes the concept of a medical specialty.
Service	Name	This class represent instances of different hospital’s services.
Equipment	Number, type	This class is the abstraction of an equipment definition.
Instrument	Due date	The specification of an equipment to represent instances of different medical operational instruments.
Furniture	-	The specification of an equipment to represent the furniture inside the hospital’s blocs.
Room	Max capacity	The specification of an equipment to represent the room of a hospital, the main purpose is to keep track of the flow of patients.

Table 7: Description of Class Diagram.

To sum up all the information contained in the above-introduced class diagram, we can say that:

- The system has multiple users, which are: Patient, Caregiver and Doctors.
- All of the users, require a Doctor with admin privileges to register them.

- Only doctors can request clinical aid from the system, therefore they are the only users who can: make reports and prescriptions.
- Only caregivers can prepare instruments.
- Each time a doctor diagnoses a patient with either a symptom or a pathology via a diagnosis action, the process is saved to the patient's history (same applies for medical care history).
- The score of a context is incremented each time two words appear in a transcription for a specific domain.
- The domain of a context score is the same of the doctor who is making the transcription. Which his domain is deduced from the service he is working at.

I.9 Conclusion

In this chapter, we have seen some of the basics of medical diagnosis, and of the medical field in general. We learned more about the problem we are facing, studied and analyzed requirements for its solution, and finally, we elaborated on the components of the system and their relationships.

The final chapter will be dedicated of the technical and practical sides of the solution.

Chapter III:

Solution Implementation and Testing

I.1 Introduction

Now that we have seen the different theoretical aspects of clinical decision-making system, their architecture, and logic and presented the conception of our proposed solution, this chapter will be devoted to the implementation of our platform. We will then talk about the tools that we used, the programming languages and then we will present our platform.

I.2 Environments

In this section, we will introduce the languages, environments, and the tools we have used to develop our platform.

I.2.1 Android

Android[18] is a mobile operating system based on a modified version of the Linux kernel and other open source software, Android is developed by a consortium of developers known as the Open Handset Alliance and commercially sponsored by Google. It was unveiled in 2007.

In this project, android will be responsible for the UI to communicate with the server.

[18] https://www.android.com/

Figure 18. Android's Logo.

I.2.2 Python

Python[19] is an interpreted, high-level, general-purpose programming language[20]. Created by Guido van Rossum and first released in 1991.

To the best we know, it is the most dynamic flexible language, it allows and ensures the extension of the applications and easy maintenance. In this project, it will be used for the server side's business layer, as it has the needed libraries of NLP and Data Science required by this project.

Figure 19.Python's Logo.

I.3 Tools and Libraries

In this section we will talk about the tools and libraries we used to develop our solution:

I.3.1 Django

"The web framework for perfectionists with deadlines" The Django Slogan.

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers,

[19] https://www.python.org/

[20] Number 1 in demand programming languages to learn in 2020 according tohttps://towardsdatascience.com/

it takes care of much of the hassle of Web development, so that the focus will be on writing the app without needing to reinvent the wheel [18]. Moreover, it is free and open source.

It will be responsible for managing and maintaining the web server application.

Figure 20.The Django's Logo.

I.3.2 Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. Android Studio was announced on May 16, 2013 at the Google I/O conference [19].

Android Studio provides the fastest tools for building apps on every type of Android device that is why it is the best choice for us.

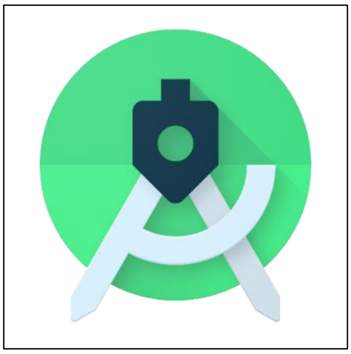


Figure 21: Android Studio's Logo.

I.3.3 PyCharm

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as Data Science with Anaconda.

“*PyCharm knows everything about your code. Rely on it for intelligent code completion, on-the-fly error checking and quick-fixes, easy project navigation, and much more.”* [19].

Figure 22: The PyCharm's Logo.

I.3.4 NLTK

NLTK (Natural Language Toolkit), is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries [20].

In this project, NLTK was used for the tokenization part in preprocessing.

Figure 23: The NLTK's Logo.

I.3.5 Pandas

Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language. In 2008, *Pandas* development began at AQR Capital Management. By the end of 2009 it had been open sourced, and is actively supported today by a community of like-minded individuals around the world who contribute their valuable time and energy to help make open source *pandas* possible [21].

In this project, it will help the inference engine to handle the available information.

Figure 24. Panda's Logo

I.3.6 MySQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data.

A Relational database is the best choice for this project, as it is based on relations in every aspect, either between words, contexts, symptoms and pathologies, doctors and patient…

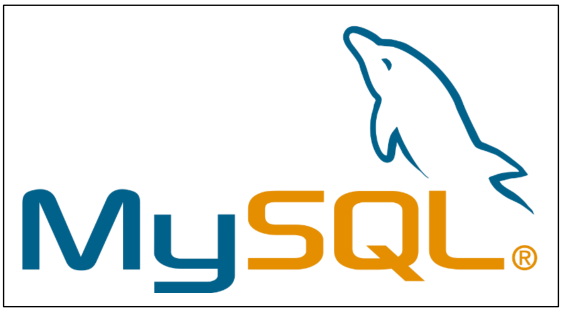


Figure 25.MySQL's Logo.

I.4 Architectures

In the following part, we will talk about the architectures of our solution:

I.4.1 Server side

The server side will be following the Model-View-Template architecture (Figure 26), as it is one of the best architectures to ensure the separation of concerns of different application layers.

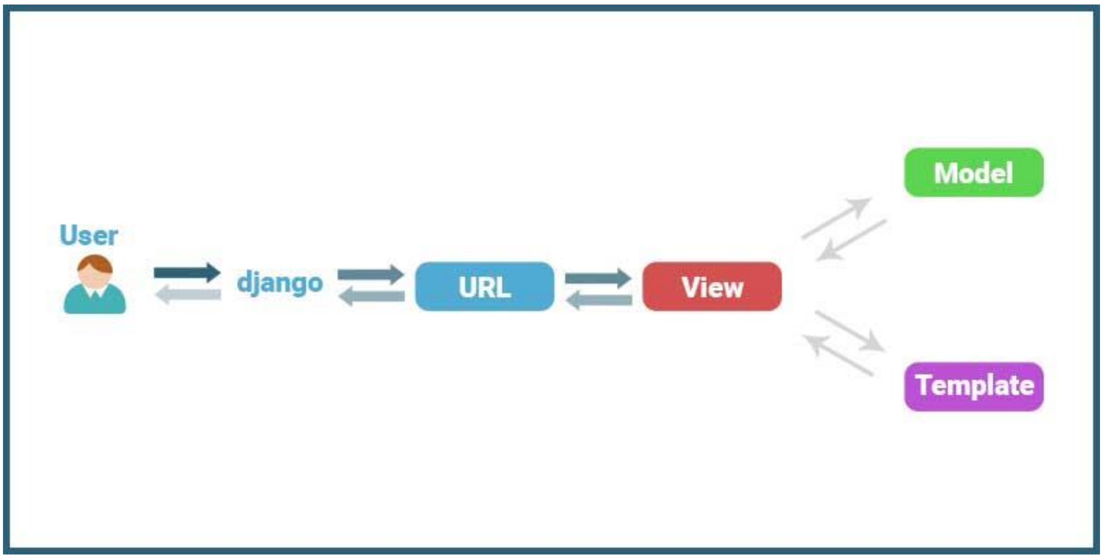


Figure 26: MVT Architecture

- The model layer, will be responsible for the database’s communication and the representation of its data.

- The view layer, in this special project, will be based on REST API, as it will have only a mobile interface, it will be the best choice to exchange data back and forth between the Model and the Template layers.

- The template layer in this project, will be represented as JSON, it will be responsible for presenting the data provided by the view layer.

I.4.2 Mobile Side

The client side will be implemented using the Model-View-View-Model (MVVM) architecture (Figure 27), it guarantees real time interactions and provides the best use of the OBSERVER design pattern. it is the recommended by GOOGLE.

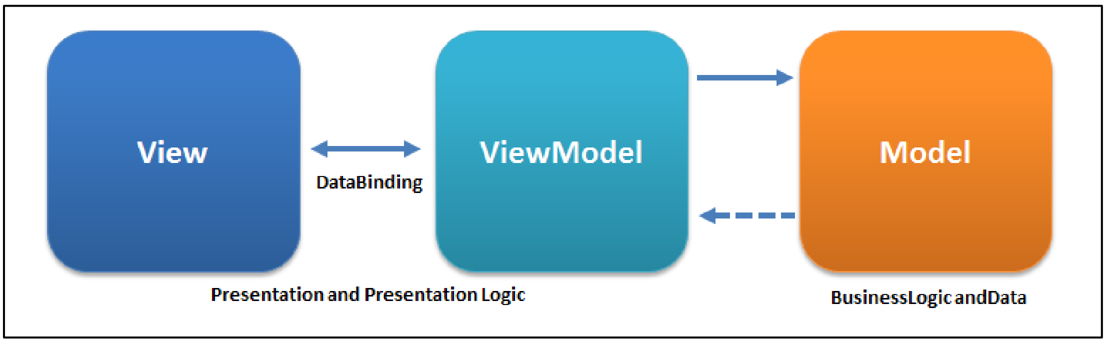


Figure 27: MVVM architecture

- The Model layer, is responsible for communicating with the server over REST API, through separated repositories (each handles a concern).
- The View layer, will be responsible for the data presentation, provided by the Model layer, through the interface.
- The View-Model layer, is responsible for communication and data transfer between the two previous layers, on top of that it guarantees data caching for a better user experience.

I.5 Datasets

I.5.1 Pathologies Dataset

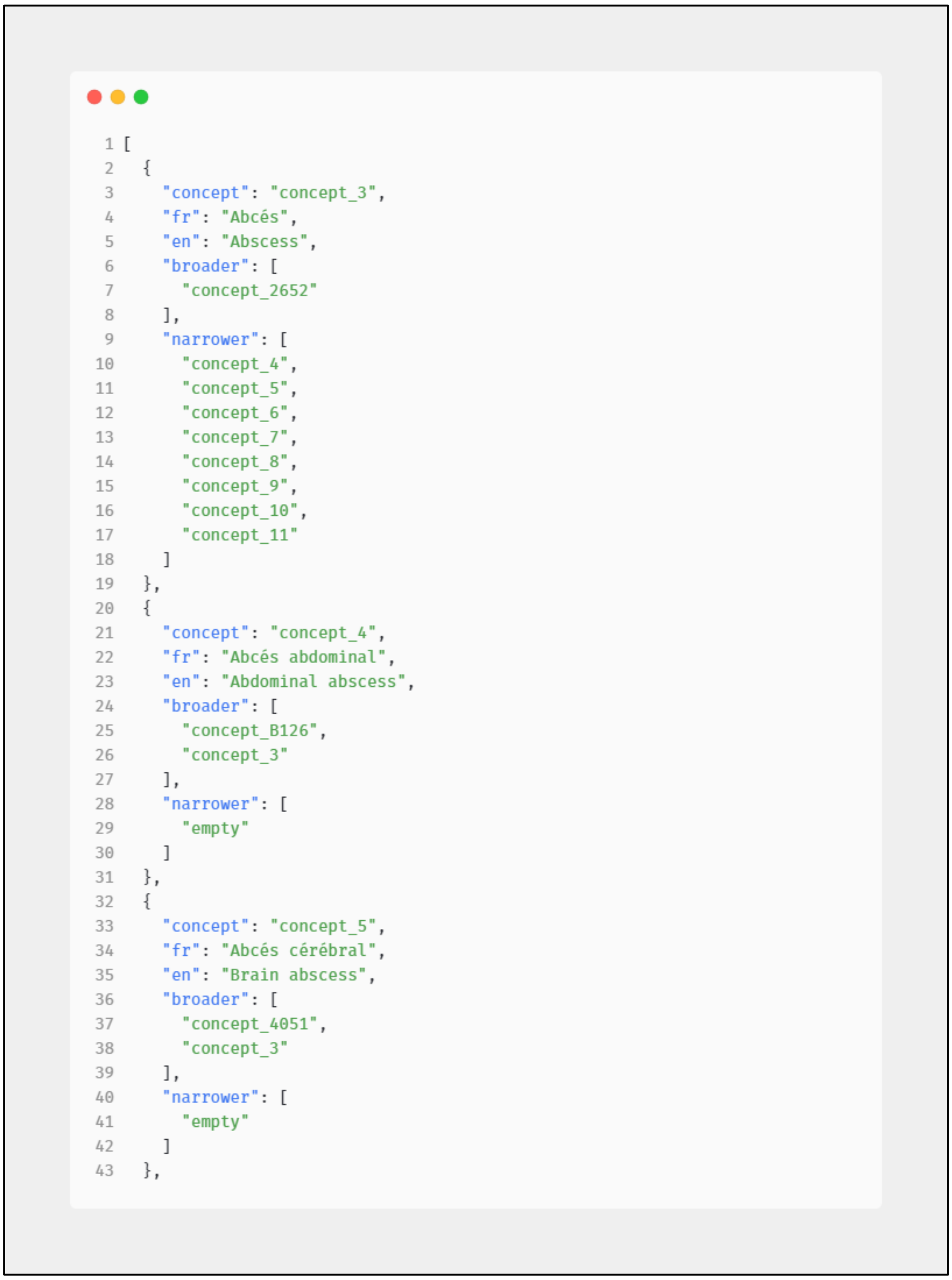
This dataset[21] was used to build the knowledge base about pathologies (Figure 28), it is provided by "Ortolang" (Outils et Ressources pour un Traitement Optimisé de la LANGue).

It is a thesaurus presenting in a hierarchical form the main pathologies that can affect the human species [22].

Thanks to this, the knowledge base has been filled with more than 6000 pathology, and more than 20,000 relation between pathologies.

It should be mentioned that the dataset has been adapted manually from RDF to JSON, for a better integration with the APIs.

[21] https://www.ortolang.fr/market/terminologies/vocabulaire-de-pathologies-humaines



```
[
  {
    "concept": "concept_3",
    "fr": "Abcés",
    "en": "Abscess",
    "broader": [
      "concept_2652"
    ],
    "narrower": [
      "concept_4",
      "concept_5",
      "concept_6",
      "concept_7",
      "concept_8",
      "concept_9",
      "concept_10",
      "concept_11"
    ]
  },
  {
    "concept": "concept_4",
    "fr": "Abcés abdominal",
    "en": "Abdominal abscess",
    "broader": [
      "concept_B126",
      "concept_3"
    ],
    "narrower": [
      "empty"
    ]
  },
  {
    "concept": "concept_5",
    "fr": "Abcés cérébral",
    "en": "Brain abscess",
    "broader": [
      "concept_4051",
      "concept_3"
    ],
    "narrower": [
      "empty"
    ]
  },
```

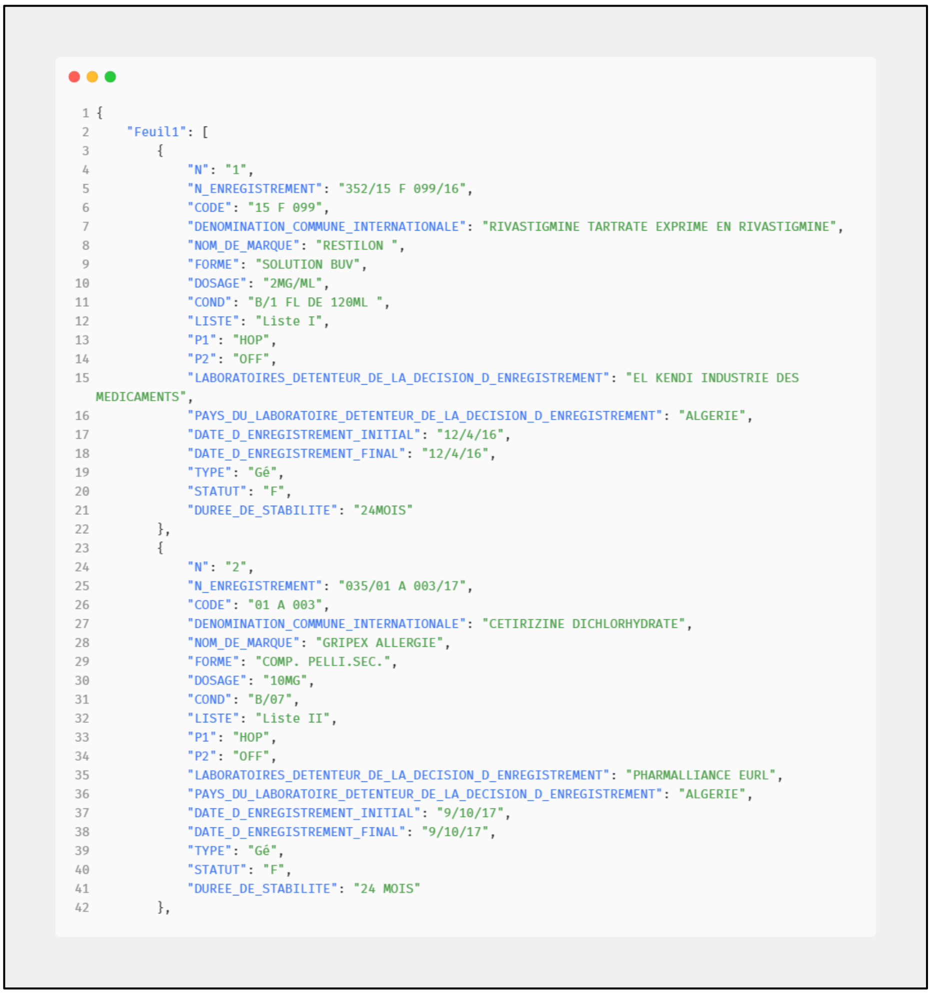
Figure 28: Subset of the Pathologies' Dataset.

Figure 29:Ortolang's Logo.

I.5.2 **Medicines Nomenclature**

Figure 28 shows a subset of the official list of medicines and drugs used in Algeria[22], it has so many details including the medicine name, dosage, scientific name, commercial name, dosage…etc. Its main purpose is to train the transcriber for generating prescription documents.

[22] http://www.sante.gov.dz/pharmacie/407-la-nomenclature.html



```json
{
    "Feuil1": [
        {
            "N": "1",
            "N_ENREGISTREMENT": "352/15 F 099/16",
            "CODE": "15 F 099",
            "DENOMINATION_COMMUNE_INTERNATIONALE": "RIVASTIGMINE TARTRATE EXPRIME EN RIVASTIGMINE",
            "NOM_DE_MARQUE": "RESTILON ",
            "FORME": "SOLUTION BUV",
            "DOSAGE": "2MG/ML",
            "COND": "B/1 FL DE 120ML ",
            "LISTE": "Liste I",
            "P1": "HOP",
            "P2": "OFF",
            "LABORATOIRES_DETENTEUR_DE_LA_DECISION_D_ENREGISTREMENT": "EL KENDI INDUSTRIE DES MEDICAMENTS",
            "PAYS_DU_LABORATOIRE_DETENTEUR_DE_LA_DECISION_D_ENREGISTREMENT": "ALGERIE",
            "DATE_D_ENREGISTREMENT_INITIAL": "12/4/16",
            "DATE_D_ENREGISTREMENT_FINAL": "12/4/16",
            "TYPE": "Gé",
            "STATUT": "F",
            "DUREE_DE_STABILITE": "24MOIS"
        },
        {
            "N": "2",
            "N_ENREGISTREMENT": "035/01 A 003/17",
            "CODE": "01 A 003",
            "DENOMINATION_COMMUNE_INTERNATIONALE": "CETIRIZINE DICHLORHYDRATE",
            "NOM_DE_MARQUE": "GRIPEX ALLERGIE",
            "FORME": "COMP. PELLI.SEC.",
            "DOSAGE": "10MG",
            "COND": "B/07",
            "LISTE": "Liste II",
            "P1": "HOP",
            "P2": "OFF",
            "LABORATOIRES_DETENTEUR_DE_LA_DECISION_D_ENREGISTREMENT": "PHARMALLIANCE EURL",
            "PAYS_DU_LABORATOIRE_DETENTEUR_DE_LA_DECISION_D_ENREGISTREMENT": "ALGERIE",
            "DATE_D_ENREGISTREMENT_INITIAL": "9/10/17",
            "DATE_D_ENREGISTREMENT_FINAL": "9/10/17",
            "TYPE": "Gé",
            "STATUT": "F",
            "DUREE_DE_STABILITE": "24 MOIS"
        },
```

Figure 30: A subset of Medicines' Dataset.

I.6 Google speech recognition technologies:

When it comes to the transcription process, that consists of converting Speech to Text. The platform relies on google's cloud service: Speech-to-Text Cloud service [23]. It is a service that accurately converts speech into text using an API powered by Google's AI technologies. It allows to transcribe the audio either in real time or from stored files (the platform's case).

The key feature for choosing this technology, is "speech adaptation", it can customize the speech recognition to transcribe every domain-specific terms or rare words.

I.7 Implemented Design Patterns

I.7.1 Multi-threading

It is the ability of executing multiple tasks on different threads (in a single process) at the same time (hypothetically), also known as pseudo-parallelism, it benefits from the hardware (CPU) ability to switch between multiple tasks and execute a chunk of instructions in each. The context switching happens so fast that it seems like if the tasks are running in parallel, but actually they are not.

We should keep in mind that multi-threading does not make a program or an application run faster, but it makes it more performant and smoother rather (Figure 31).

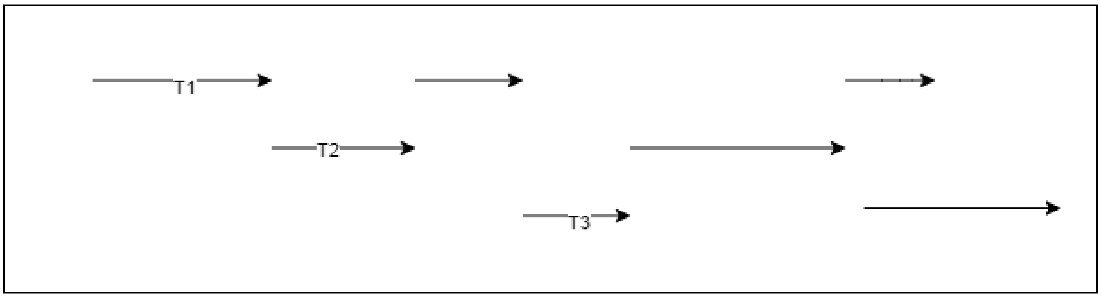


Figure 31: Threads explained

I.7.2 Async Programming

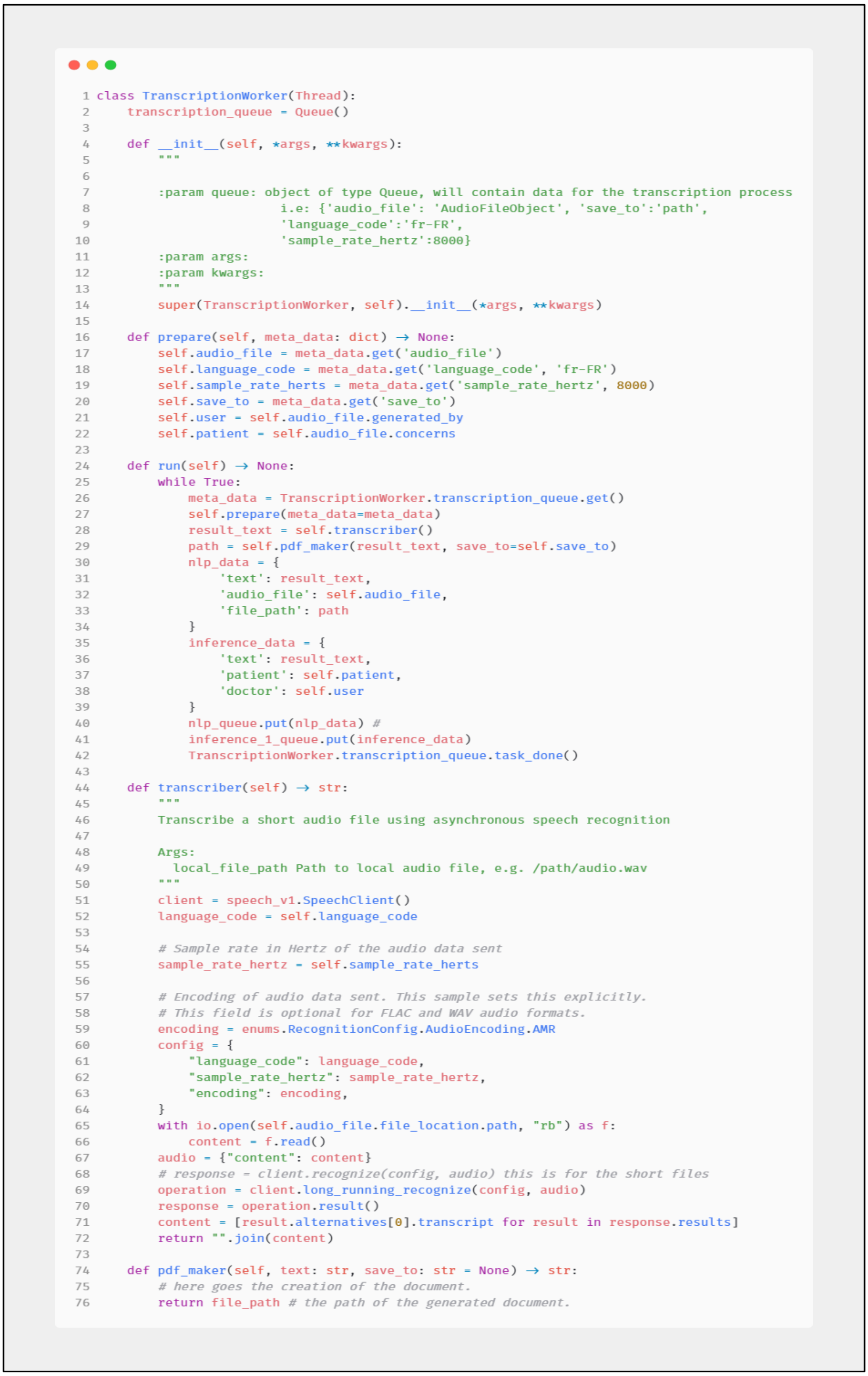
Based on multi-threading, the async programming pattern, focuses on giving the heavy tasks to worker threads, keeping the main thread focused on display and communicating with the user in general. This pattern relies on callbacks to handle the results obtained from the worker thread.

When implementing the application, we thought of three core and heavy tasks: Transcription, document generation, context classification and decision making. It is the reason why they have been assigned to background worker threads, since they are dependent on each other, and share the same resources.

The communication between them happens through queues. That will not only guarantee the communication and well sharing of resources, but it also keeps the order of the received requests.

I.7.3 Transcription and Document generation worker

The receiver of the incoming data (audio) from the end user (doctor), this first worker prepares the configurations for the googles' cloud transcription service, triggers the transcription and then generates the corresponding document, to send the results to the next workers queues (Figure 32).



```python
class TranscriptionWorker(Thread):
    transcription_queue = Queue()

    def __init__(self, *args, **kwargs):
        """


        :param queue: object of type Queue, will contain data for the transcription process
                        i.e: {'audio_file': 'AudioFileObject', 'save_to':'path',
                        'language_code':'fr-FR',
                        'sample_rate_hertz':8000}
        :param args:
        :param kwargs:
        """
        super(TranscriptionWorker, self).__init__(*args, **kwargs)

    def prepare(self, meta_data: dict) -> None:
        self.audio_file = meta_data.get('audio_file')
        self.language_code = meta_data.get('language_code', 'fr-FR')
        self.sample_rate_herts = meta_data.get('sample_rate_hertz', 8000)
        self.save_to = meta_data.get('save_to')
        self.user = self.audio_file.generated_by
        self.patient = self.audio_file.concerns

    def run(self) -> None:
        while True:
            meta_data = TranscriptionWorker.transcription_queue.get()
            self.prepare(meta_data=meta_data)
            result_text = self.transcriber()
            path = self.pdf_maker(result_text, save_to=self.save_to)
            nlp_data = {
                'text': result_text,
                'audio_file': self.audio_file,
                'file_path': path
            }
            inference_data = {
                'text': result_text,
                'patient': self.patient,
                'doctor': self.user
            }
            nlp_queue.put(nlp_data) #
            inference_1_queue.put(inference_data)
            TranscriptionWorker.transcription_queue.task_done()

    def transcriber(self) -> str:
        """
        Transcribe a short audio file using asynchronous speech recognition

        Args:
          local_file_path Path to local audio file, e.g. /path/audio.wav
        """
        client = speech_v1.SpeechClient()
        language_code = self.language_code

        # Sample rate in Hertz of the audio data sent
        sample_rate_hertz = self.sample_rate_herts

        # Encoding of audio data sent. This sample sets this explicitly.
        # This field is optional for FLAC and WAV audio formats.
        encoding = enums.RecognitionConfig.AudioEncoding.AMR
        config = {
            "language_code": language_code,
            "sample_rate_hertz": sample_rate_hertz,
            "encoding": encoding,
        }
        with io.open(self.audio_file.file_location.path, "rb") as f:
            content = f.read()
        audio = {"content": content}
        # response = client.recognize(config, audio) this is for the short files
        operation = client.long_running_recognize(config, audio)
        response = operation.result()
        content = [result.alternatives[0].transcript for result in response.results]
        return "".join(content)

    def pdf_maker(self, text: str, save_to: str = None) -> str:
        # here goes the creation of the document.
        return file_path # the path of the generated document.
```

Figure 32: TranscriptionWorker

I.7.4 Natural Language Processing worker

Transcribed data (text) received from the previous worker. This worker is responsible for preprocessing the text, extracting all the available words, group them into "medical words" and out of context words.

This medical word group will be iterated over, to build a bi-gram context and flag them with a medical domain.

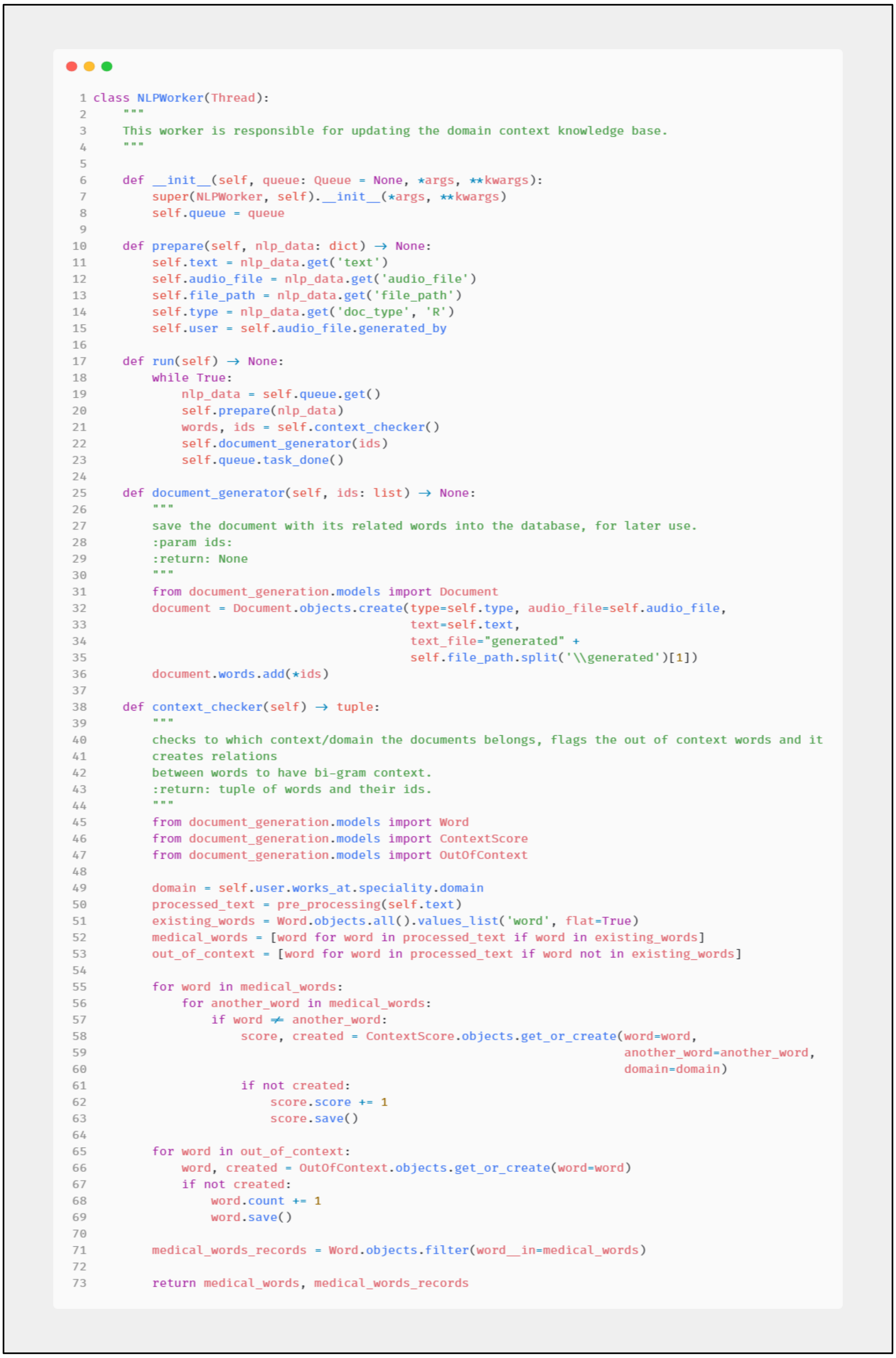
A bi-gram defines a subsequence of 2 items from a given sequence. Used in various fields of natural language processing, it defines a method for finding a set of n-gram[23]words from a given document. The commonly used model is the bigrams model and the following example will help better understand the concept:

Medical word group: “fever headache nausea chills

Bigrams: “fever headache”, “headache nausea”, “nausea chills”

In the end, the context knowledge base of the platform would be updated on each request. This can help later on the highlight medical words inside the documents (Figure 33).

[23] N-grams, a concept found in Natural Language Processing, an N-gram is simply a sequence of N words. it is the main process for context analysis



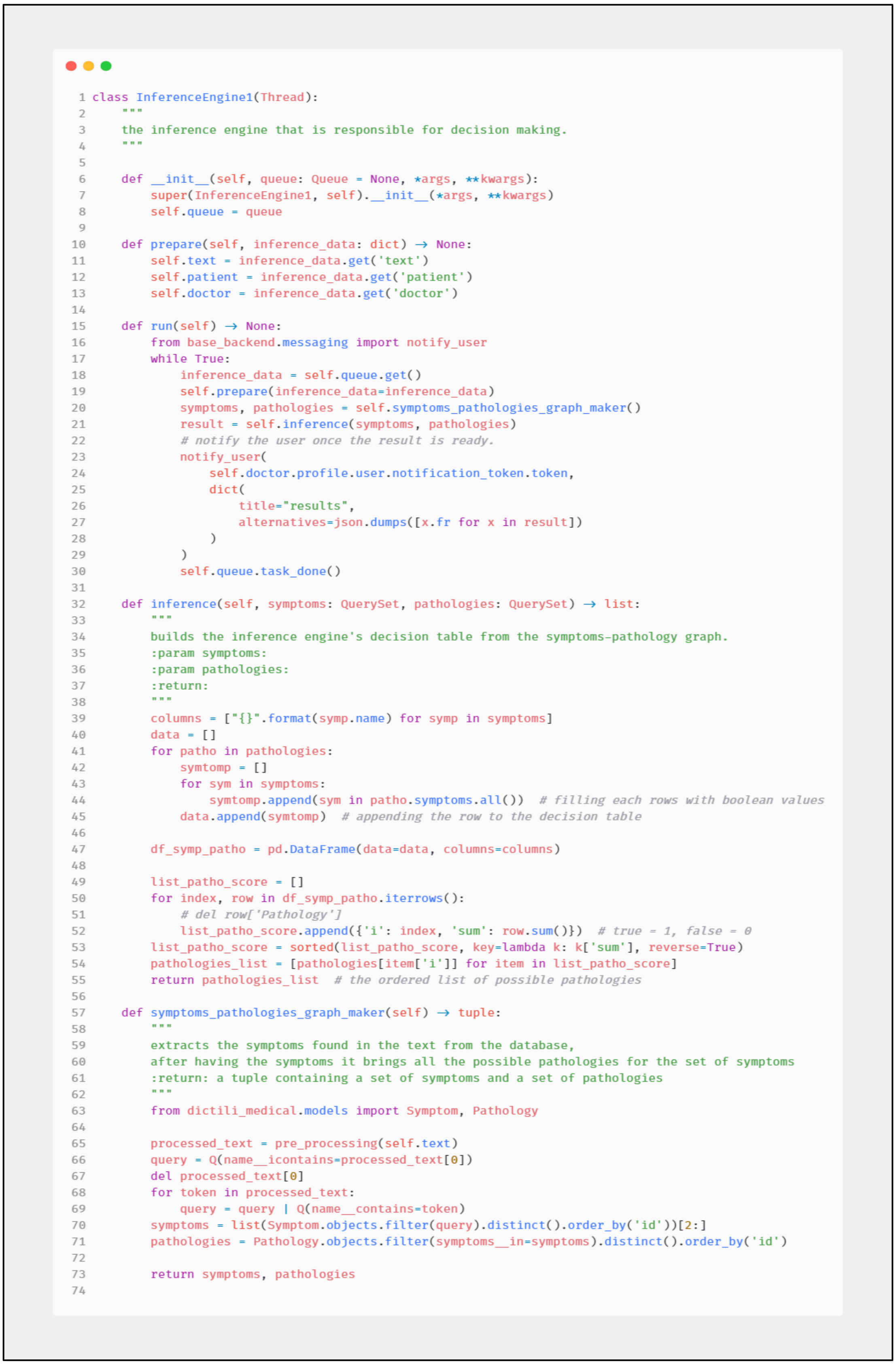
```python
class NLPWorker(Thread):
    """
    This worker is responsible for updating the domain context knowledge base.
    """

    def __init__(self, queue: Queue = None, *args, **kwargs):
        super(NLPWorker, self).__init__(*args, **kwargs)
        self.queue = queue

    def prepare(self, nlp_data: dict) → None:
        self.text = nlp_data.get('text')
        self.audio_file = nlp_data.get('audio_file')
        self.file_path = nlp_data.get('file_path')
        self.type = nlp_data.get('doc_type', 'R')
        self.user = self.audio_file.generated_by

    def run(self) → None:
        while True:
            nlp_data = self.queue.get()
            self.prepare(nlp_data)
            words, ids = self.context_checker()
            self.document_generator(ids)
            self.queue.task_done()

    def document_generator(self, ids: list) → None:
        """
        save the document with its related words into the database, for later use.
        :param ids:
        :return: None
        """
        from document_generation.models import Document
        document = Document.objects.create(type=self.type, audio_file=self.audio_file,
                                           text=self.text,
                                           text_file="generated" +
                                           self.file_path.split('\\generated')[1])
        document.words.add(*ids)

    def context_checker(self) → tuple:
        """
        checks to which context/domain the documents belongs, flags the out of context words and it
        creates relations
        between words to have bi-gram context.
        :return: tuple of words and their ids.
        """
        from document_generation.models import Word
        from document_generation.models import ContextScore
        from document_generation.models import OutOfContext

        domain = self.user.works_at.speciality.domain
        processed_text = pre_processing(self.text)
        existing_words = Word.objects.all().values_list('word', flat=True)
        medical_words = [word for word in processed_text if word in existing_words]
        out_of_context = [word for word in processed_text if word not in existing_words]

        for word in medical_words:
            for another_word in medical_words:
                if word ≠ another_word:
                    score, created = ContextScore.objects.get_or_create(word=word,
                                                                        another_word=another_word,
                                                                        domain=domain)

                    if not created:
                        score.score += 1
                        score.save()

        for word in out_of_context:
            word, created = OutOfContext.objects.get_or_create(word=word)
            if not created:
                word.count += 1
                word.save()

        medical_words_records = Word.objects.filter(word__in=medical_words)

        return medical_words, medical_words_records
```

Figure 33: NLPWorker

I.7.5 Inference Engine

The responsible for the decision-making part, after receiving the preprocessed text, this worker extracts the symptoms. Then it builds the symptom-pathology graph.

From this generated graph, the first decision table will be generated (explained earlier in Chapter II). Then it sorts the pathologies according to the found symptoms and in the end, it provides the result back to the user.



```python
class InferenceEngine1(Thread):
    """
    the inference engine that is responsible for decision making.
    """

    def __init__(self, queue: Queue = None, *args, **kwargs):
        super(InferenceEngine1, self).__init__(*args, **kwargs)
        self.queue = queue

    def prepare(self, inference_data: dict) -> None:
        self.text = inference_data.get('text')
        self.patient = inference_data.get('patient')
        self.doctor = inference_data.get('doctor')

    def run(self) -> None:
        from base_backend.messaging import notify_user
        while True:
            inference_data = self.queue.get()
            self.prepare(inference_data=inference_data)
            symptoms, pathologies = self.symptoms_pathologies_graph_maker()
            result = self.inference(symptoms, pathologies)
            # notify the user once the result is ready.
            notify_user(
                self.doctor.profile.user.notification_token.token,
                dict(
                    title="results",
                    alternatives=json.dumps([x.fr for x in result])
                )
            )
            self.queue.task_done()

    def inference(self, symptoms: QuerySet, pathologies: QuerySet) -> list:
        """
        builds the inference engine's decision table from the symptoms-pathology graph.
        :param symptoms:
        :param pathologies:
        :return:
        """
        columns = ["{}".format(symp.name) for symp in symptoms]
        data = []
        for patho in pathologies:
            symtomp = []
            for sym in symptoms:
                symtomp.append(sym in patho.symptoms.all())  # filling each rows with boolean values
            data.append(symtomp)  # appending the row to the decision table

        df_symp_patho = pd.DataFrame(data=data, columns=columns)

        list_patho_score = []
        for index, row in df_symp_patho.iterrows():
            # del row['Pathology']
            list_patho_score.append({'i': index, 'sum': row.sum()})  # true = 1, false = 0
        list_patho_score = sorted(list_patho_score, key=lambda k: k['sum'], reverse=True)
        pathologies_list = [pathologies[item['i']] for item in list_patho_score]
        return pathologies_list  # the ordered list of possible pathologies

    def symptoms_pathologies_graph_maker(self) -> tuple:
        """
        extracts the symptoms found in the text from the database,
        after having the symptoms it brings all the possible pathologies for the set of symptoms
        :return: a tuple containing a set of symptoms and a set of pathologies
        """
        from dictili_medical.models import Symptom, Pathology

        processed_text = pre_processing(self.text)
        query = Q(name__icontains=processed_text[0])
        del processed_text[0]
        for token in processed_text:
            query = query | Q(name__contains=token)
        symptoms = list(Symptom.objects.filter(query).distinct().order_by('id'))[2:]
        pathologies = Pathology.objects.filter(symptoms__in=symptoms).distinct().order_by('id')

        return symptoms, pathologies

```

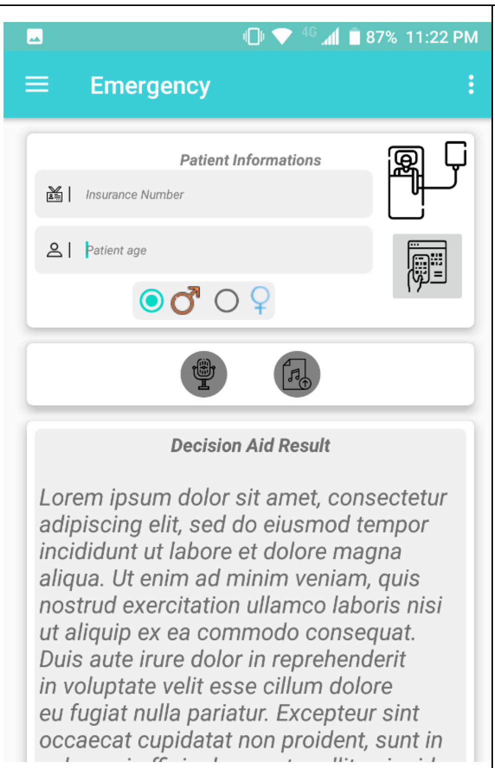
Figure 34: Inference Engine

I.8 Implementation and Testing

In this section, we will present and explain a test scenario of the platform.

The testing scenario would follow the sequence, so it will go as follows:

1. The patient enters the emergencies and presents his insurance card (if he has one).
2. If the patient has an insurance card, the doctor will scan (Figure 35) to load the patient’s information directly into the app (Figure 36). In the opposite case, the information has to be provided manually.



Patient Informations

Insurance Number

Patient age

◉ ♂ ○ ♀

Decision Aid Result

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in

Figure 35: Emergency Interface (before scanning).

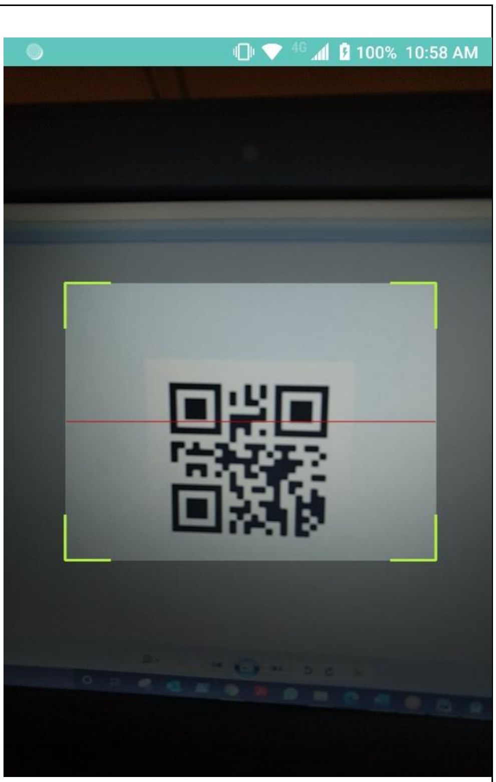
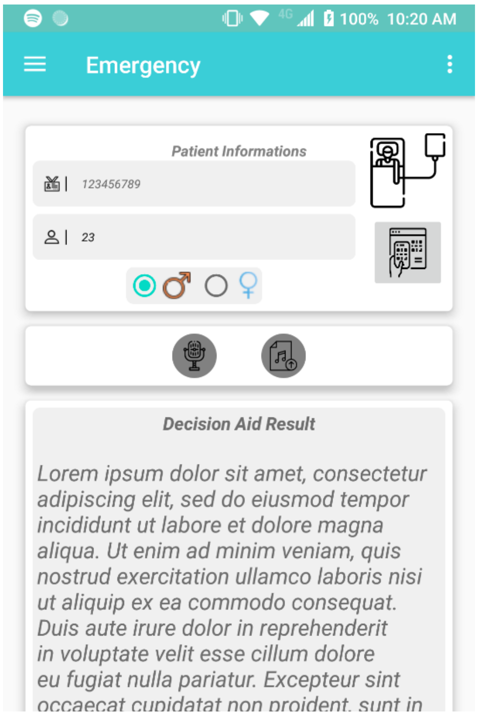


Figure 36: Code scanning interface.

3. The doctor then, will start recording (Figure 37) and describes the patient state (symptoms), once finished the note will be uploaded to the server.
4. Once on the server, the audio file will be archived and represented with an instance of Audio Recording Class. Following that, the same file will be uploaded to the cloud for transcription (Figure 38).



100% 10:20 AM

Emergency

Patient Informations

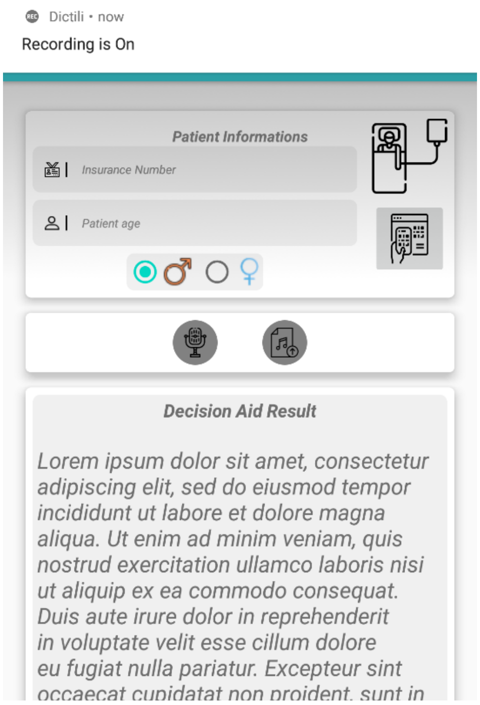
123456789

23

Decision Aid Result

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in

Figure 37: Emergency Interface (after scanning).



Dictili • now

Recording is On

Patient Informations

Insurance Number

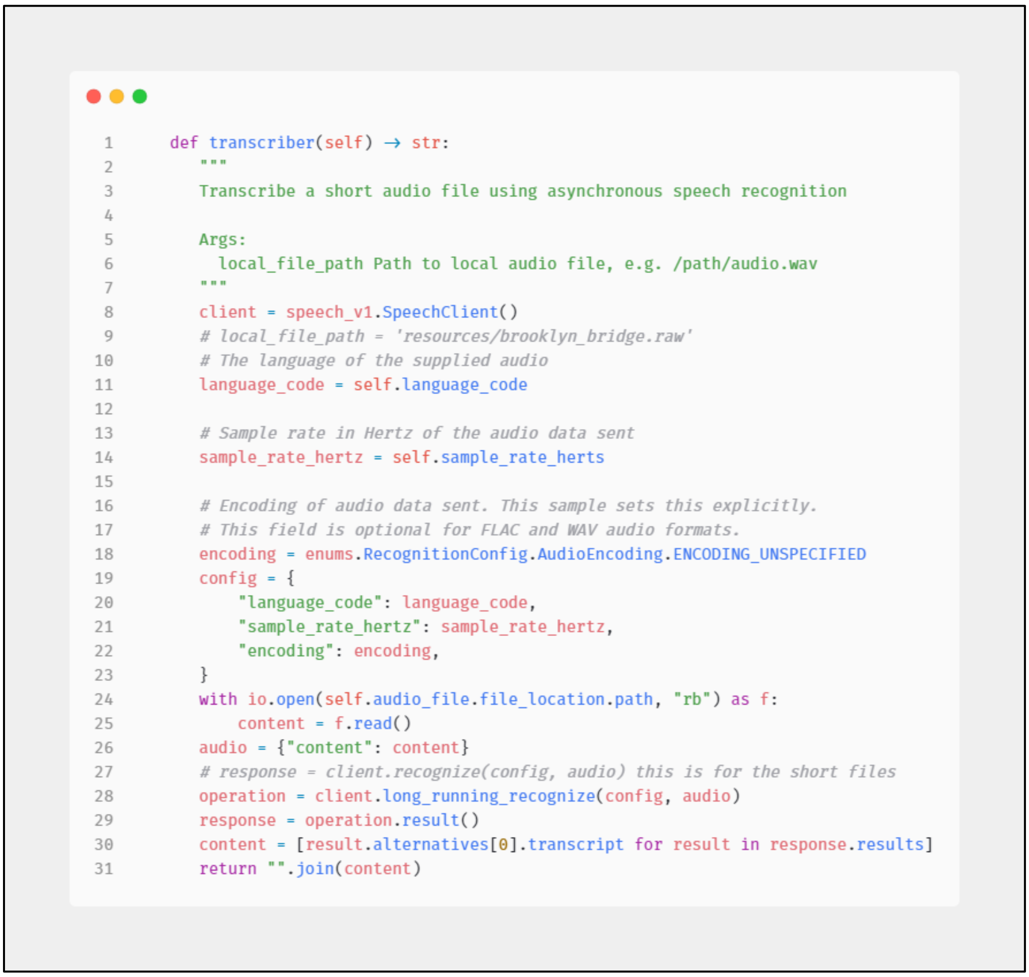
Patient age

Decision Aid Result

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in

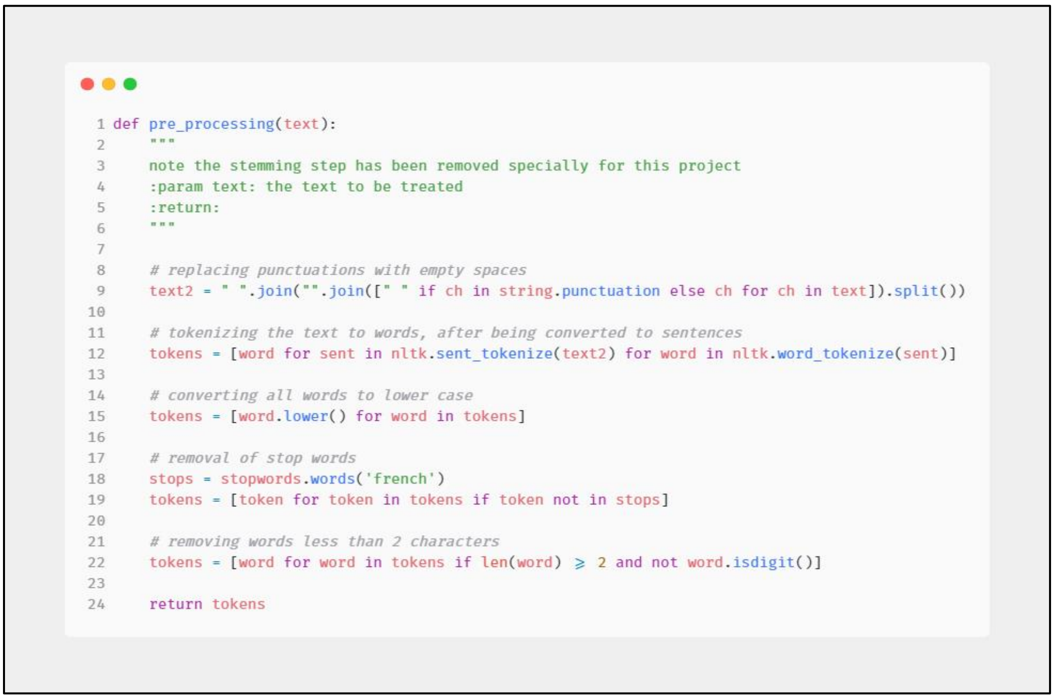
Figure 38: Recording Action.

5. The result of the transcription is : « La température du patient est de 38 degrés, il a mal au ventre. Mal et une sécheresse de gorge. Il respire difficilement et il a une toux.".
6. Next, the previous transcription result will be preprocessed (Figure 37), resulting in: “temperature patient 38 degrés mal au ventre mal séchresse à la gorge respire difficilement toux”, and then provided to the IE 1 (Table. 4).



```
def transcriber(self) -> str:
    """
    Transcribe a short audio file using asynchronous speech recognition

    Args:
      local_file_path Path to local audio file, e.g. /path/audio.wav
    """
    client = speech_v1.SpeechClient()
    # local_file_path = 'resources/brooklyn_bridge.raw'
    # The language of the supplied audio
    language_code = self.language_code

    # Sample rate in Hertz of the audio data sent
    sample_rate_hertz = self.sample_rate_herts

    # Encoding of audio data sent. This sample sets this explicitly.
    # This field is optional for FLAC and WAV audio formats.
    encoding = enums.RecognitionConfig.AudioEncoding.ENCODING_UNSPECIFIED
    config = {
        "language_code": language_code,
        "sample_rate_hertz": sample_rate_hertz,
        "encoding": encoding,
    }
    with io.open(self.audio_file.file_location.path, "rb") as f:
        content = f.read()
    audio = {"content": content}
    # response = client.recognize(config, audio) this is for the short files
    operation = client.long_running_recognize(config, audio)
    response = operation.result()
    content = [result.alternatives[0].transcript for result in response.results]
    return "".join(content)
```

Figure 39: Uploading A File for Transcription.



```python
def pre_processing(text):
    """
    note the stemming step has been removed specially for this project
    :param text: the text to be treated
    :return:
    """

    # replacing punctuations with empty spaces
    text2 = " ".join("".join([" " if ch in string.punctuation else ch for ch in text]).split())

    # tokenizing the text to words, after being converted to sentences
    tokens = [word for sent in nltk.sent_tokenize(text2) for word in nltk.word_tokenize(sent)]

    # converting all words to lower case
    tokens = [word.lower() for word in tokens]

    # removal of stop words
    stops = stopwords.words('french')
    tokens = [token for token in tokens if token not in stops]

    # removing words less than 2 characters
    tokens = [word for word in tokens if len(word) >= 2 and not word.isdigit()]

    return tokens
```

Figure 40.Preprocessing Function.

7. The IE 1, generated the results shown in (Table 8) and then provided them to IE 2.

	Temperature 38	**Mal au ventre**	**Mal à la gorge**	**Respire difficilement**	**Toux**	**Sécheresse de la gorge**
CoVid-19	True	True	True	True	True	True
Grippe saisonniére	True	False	True	True	True	False
Grippe aviaire	True	False	True	True	True	False

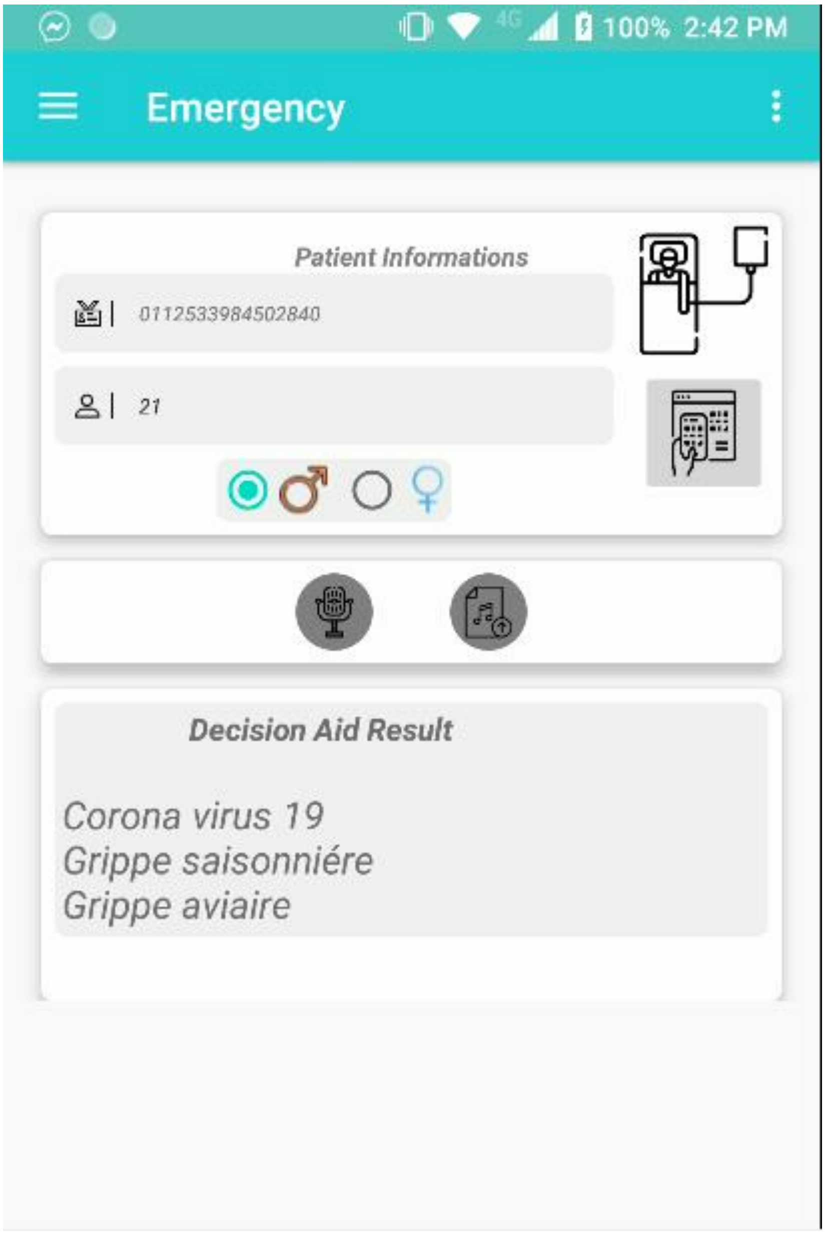
Table 8: Inference Engine 1 Result.

8. The IE 2, extracted the patient’s data (age, region, gender) and resulted in (Table 9).

	23	**Afrique du nord**	**Male**
CoVid-19	True	True	True
Grippe saisonniére	True	True	True
Grippe aviaire	True	False	True

Table 9: Inference Engine 2 Results.

9. The final result provided to the doctor wil then be: 1- CoVid-19; 2- Grippe saisonniére; 3-Grippe Aviaire.



Patient Informations

0112533984502840

21

● ♂ ○ ♀

Decision Aid Result

Corona virus 19
Gripple saisonniére
Grippe aviaire

Figure 41. Emergency Interface After Receiving the Results.

I.9 Conclusion

In this chapter, we have seen the development phase and its different steps. Also talked briefly about the tools which helped in the realization of this project. The different languages, platforms and the interoperability between them.

And finally, we have seen one of the successful test cases, actually it is the core solution this project proposed.

General conclusion

I.1 Summary

In life everything comes to an end, and this is the final part of this thesis, but not the solution. The problem we were trying to solve was how to help the doctors, especially beginners at the field in their daily professional life. Since the decisions of the diagnosis can be taught, especially when they are alone. In addition, our solution is designed to help in other situations like the lack of assistants.

In this work, we implemented a medical diagnostic assistance system, and a documents/prescriptions generator. Both were based on voice notes, in order to facilitate and accelerate the process on the user's side. Our application provides the basic functionalities: medical diagnosis, clinical examination which makes it possible to identify the diseases corresponding to the signs observed in the patient as well as other secondary functionalities such as the management of the diseases, the patients and the documents.

The inference engines were designed using Python and the Pandas library. They have been grouped as an ensemble using threads, and a queue for communication. The accuracy of the transcription was good. It helped in achieving the desired results during the tests.

I.2 Outlooks

However, our work is far from being perfect, and it is not even complete. As the fields of decision making and knowledge engineering are huge. We were hoping to collect more data about: symptoms and their relations with pathologies, pathologies' characteristics, and relationship between caregivers and their tools. Also, we could not interview doctors to collect more of their experiences, differences and methodologies, because of the situation the world is facing. Even so, we are glad that we had the chance to put our hands on a real solution that can improve the health section and its practitioners. It is necessary to mention, that our project can have many extensions. Each specialized for a specific issue. Like archiving the medical records of patients digitally, and that would eliminate the fear of losing paper documents. Another extension would be medical consultations' scheduler, and of course reminders.

Even if the first test result achieved the desired behavior, the ensemble system is still lacking data and experience of doctors. It still has a long way to go, starting by obtaining more pathologies and symptoms, analyzing real medical reports/prescriptions and finishing by having the approval of doctors about the decisions it makes.

Bibliography

[1] R. Polikar, *Ensemble based systems in decision making,* Circuits and Systems Magazine.6. 21 - 45. 10.1109/MCAS.2006.1688199, 2006.

[2] S. Jaiswal, “Decision making systems,” 1 Nov 2014. [Online]. Available: https://www.slideshare.net/shwetabhjaiswal/decision-making-systems..

[3] T. Ö. e. al., *Predictive Intelligence Using Big Data and the Internet of Things,* 2018.

[4] G. MARIN, *Decision support systems,* Romanian American University, Bucharest,: Faculty of Computer Science for Business Management..

[5] P. Croskerry, *A Model for Clinical Decision-Making in Medicine,* Medical Science Educator. 27. 10.1007/s40670-017-0499-9, 2017.

[6] “Understanding the Importance of Medical Decision Making,” 2018. [Online]. Available: https://www.healthcarecompliancepros.com/blog/understanding-the-importance-of-medical-decision-making.

[7] “MEDICAL DECISION MAKIN,” 2011. [Online]. Available: https://safemt.com/medical-decision-making.

[8] Belur V. Dasarathy. a. Belur V. Sheela, *Composite classifier system design.*

[9] L. H. a. P. Salamon, *Neural network ensembles,* Transactionson Pattern Analysis and Machine Intelligence, vol. 12, no. 10, pp.993–1001, 1990.

[10] R. Schapire, *The strength of weak learnability,* Machine Learning,vol. 5, no. 2, pp. 197–227, 1990.

[11] L. A. Lynn, “Artificial intelligence systems for complex decision-making in acute care medicine: a review,” Patient Safety in Surgery, 2019. [Online]. Available: https://doi.org/10.1186/s13037-019-0188-2..

[12] T. e. al., “ Data assimilation and multisource decision making in systems biology based on unobtrusive Internet of Things devices,” BioMed Eng OnLine, 2018. [Online]. Available: https://doi.org/10.1186/s12938-018-0574-5.

[13] S. &. Q. U. &. J. M. Bashir, *An Ensemble based Decision Support Framework for Intelligent Heart Disease Diagnosis,* International Conference on Information Society, i-Society 2014. 10.1109/i-Society.2014.7009056, 2014.

[14] K. Monica, “Top Clinical Decision Support System (CDSS) Companies by Ambulatory, Inpatient Settings,” ehrintelligence, 07 April 2017. [Online]. Available: https://ehrintelligence.com/news/top-clinical-decision-support-system-cdss-companies-by-ambulatory-inpatient.

[15] GHERZOULI Imene, *Système d'aide au diagnostic médical à base d'ontologie,* 2016/2017.

[16] J. &. O. D. &. B. W. &. V. G. McFillen, *Organizational Diagnosis: An Evidence-based Approach.,* Journal of Change Management. 13. 1-24. 10.1080/14697017.2012.679290., 2012.

[17] D. &. K. A. &. K. K. &. S. S. Khurana, *Natural Language Processing: State of The Art, Current Trends and Challenges.,* 2017.

[18] “django project,” [Online]. Available: https://www.djangoproject.com 18/10/2020.

[19] “jet brains,” [Online]. Available: www.jetbrains.com 18/10/2020.

[20] “nltk,” [Online]. Available: www.nltk.org 18/10/2020.

[21] “pandas,” [Online]. Available: www.pandas.pydata.org 18/10/2020.

[22] “ortolang,” [Online]. Available: www.ortolang.fr 18/10/2020.

[23] “Speech-to-Text,” [Online]. Available: https://cloud.google.com/speech-to-text 18/10/2020.