University Constantine 2
Faculty of New Technologies
IFA Department

Mémoire de Licence

Automatic Speech Recognition in Healthcare

Major : Computer Science

By :

Meribout Ahmed Yahia

Zouak Omar Elfarouk

Supervised by : Prof. Khaled NECIBI

Dedication

This work is dedicated to:

The sake of Allah, our Creator and my Master,My great teacher and messenger, Mohammed (May Allah bless and grant him),who taught us the purpose of life.

And to our parents who have never failed to give us fancial and moral support,for giving all our needs during the time we developed our system and for teaching us that even the largest task can be accomplished if it is done one step as a time.

Special thanks to our supervisor, Dr.Khaled Necibi,for your patience, guidance, and support.We are extremely grateful for the faith you had in us throughout the year.

Acknowledgement

Special thanks to our supervisor, Dr.Khaled Necibi, for your patience, guidance, and support.We are extremely grateful for the faith you had in us throughout the year.

ملخص

يهدف هذا المشروع إلى استكشاف حلول مبتكرة للتشخيص والعلاج الفعال من خلال تطوير تطبيق للرعاية الصحية يستخدم تحويل الكلام إلى نص للمساعدة في تشخيص المرضى ، ويدمج التطبيق قدرات التعرف على الكلام ، مما يمكّن المرضى من وصف أعراضهم. ومن خلال الاستعلام عن قاعدة بيانات شاملة للمعرفة الطبية نحصل على نتيجة التشخيص من شدة الأعراض إلى ما يجب القيام به للتخفيف منها.

الكلمات الرئيسية: التعرف التلقائي على الكلام ، الذكاء الاصطناعي ، خبير النظام ، الشبكة العصبية ، تحويل النص إلى كلام ، قواعد البيانات ، التشخيص ، المتطرفة.

Abstract

This project aims to explore innovative solutions for efficient diagnosis and treatment by developing a healthcare application that utilizes Speech to text to aid in the diagnosis of patients.The app integrates speech recognition capabilities, enabling patients to describe their symptoms. and by querying a comprehensive database of medical knowledge we get a result of the diagnose from the severeness of the symptoms to what to do to mitigate them.

Keywords : Automatic Speech Recognition, Artificial Intelligence, System Expert, Neural network, Google Speech API, Text to Speech, Databases, Diagnose, Severe, minor.

Résumé

Ce projet vise à explorer des solutions innovantes pour un diagnostic et un traitement efficaces en développant une application de soins de santé qui utilise la synthèse vocale pour faciliter le diagnostic des patients. L'application intègre des capacités de reconnaissance vocale, permettant aux patients de décrire leurs symptômes. et en interrogeant une base de données complète de connaissances médicales, nous obtenons un résultat du diagnostic de la gravité des symptômes à ce qu'il faut faire pour les atténuer.

Mots-clés : Reconnaissance automatique de la parole, Intelligence artificielle, Expert système, Réseau de neurones, API Google Speech, Synthèse vocale, Bases de données, Diagnostiquer, Sévère, Mineur.

Contents Table

Figures table

Tables list

General Introduction :

Even though e-mails and messenger services have changed human communication significantly, speech is still the most important method of communication between humans. Therefore, automatic speech recognition (ASR) is of particular relevance as well, providing a transcription of the spoken language that can be evaluated by automated systems considerably better than a raw audio signal. With smart assistants such as Google Home, Alexa or Siri, ASR is already an integral part of many households and is used to play music, answer questions or control other smart devices such as a home automation system.

Nevertheless, ASR can also be found in many other sectors such as healthcare, BFSI(Banking, financial services and insurance), IT telecommunications, automotive, government, legal, retail, travel and hospitality, and others. The demand for speech and voice recognition software has increased drastically among healthcare and BFSI, owing to the COVID-19 outbreak.

As technology continues to advance at an unprecedented pace, it has become increasingly evident that innovative solutions can revolutionize the healthcare industry. Among the transformative technologies, Automatic Speech Recognition (ASR) has emerged as a powerful tool with immense potential to enhance healthcare delivery and improve patient outcomes. This project aims to explore the application of ASR in healthcare, investigating its capacity to streamline the diagnostic process, we aim to identify areas where ASR can enhance efficiency, accuracy, and overall patient care.

Healthcare can be greatly aided by voice recognition. Using voice recognition, healthcare providers, such as doctors and nurses, can dictate notes into their computers without having to take time away from patient care. As a result, they can devote more time to personal interactions or other tasks while completing their work efficiently. Furthermore, patients can also benefit from this technology since it makes it easier than ever before to seek help if they are feeling unwell using an app on their phones where voice-dictation software will transcribe their words so someone else can read it back and see if anything needs to be addressed or an virtual assistant that can give treatment for small common illness like cold and basic headache and redirect patients to nearby doctors if necessery.

Chapter 1

Speech recognition in the field of healthcare?

1.1 Introduction:

What is AI?
Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. Specific applications of AI include expert systems, natural language processing, speech recognition and machine vision.
You can think of AI as being a form of intelligence that is used to solve problems, come up with solutions, answer questions, make predictions, or offer strategic suggestions.Because AI can do all these things, it's become incredibly important to modern businesses and other types of organizations.
How AI works in simple terms?
AI systems work by combining large sets of data with intelligent, iterative processing algorithms to learn from patterns and features in the data that they analyze.
Why AI?
Applications of AI can be seen in everyday scenarios such as financial services fraud detection, retail purchase predictions, and online customer support interactions. One of the biggest fields that AI is getting attention into is HealthCare where AI technology has been trained to provide personalized medicine, including giving reminders about when patients need to take their medicine and suggestions for specific exercises they should perform to improve their recovery from injuries and many more other solutions AI can give.

1.2 Automatic Speech Recognition :

Speech recognition is the capacity of a computer to convert human speech into written text. Also known as automatic/automated speech recognition (ASR) and speech to text (STT), it's a subfield of computer science and computational linguistics. Today, this technology has evolved to the point where machines can understand natural speech in different languages, dialects, accents, and speech patterns.
In terms of technological development, we may still be at least a couple of decades away from having truly autonomous, intelligent artificial intelligence.
The basic sequence of events that makes any Automatic Speech Recognition software, regardless of its sophistication, pick up and break down your words for analysis and response goes as follows:

1. You speak to the software via an audio feed.

2. The device you're speaking to creates a wave file of your words

3. The wave file is cleaned by removing background noise and normalizing volume

4. The resulting filtered wave form is then broken down into what are called phonemes. (Phonemes are the basic building block sounds of language and words. English has 44 of them, consisting of sound blocks such as “wh”, “th”, “ka” and “t”.

5. Each phoneme is like a chain link and by analyzing them in sequence, starting from the first phoneme, the ASR software uses statistical probability analysis to deduce whole words and then from there, complete sentences.

6. Your ASR, now having “understood” your words, can respond to you in a meaningful way.

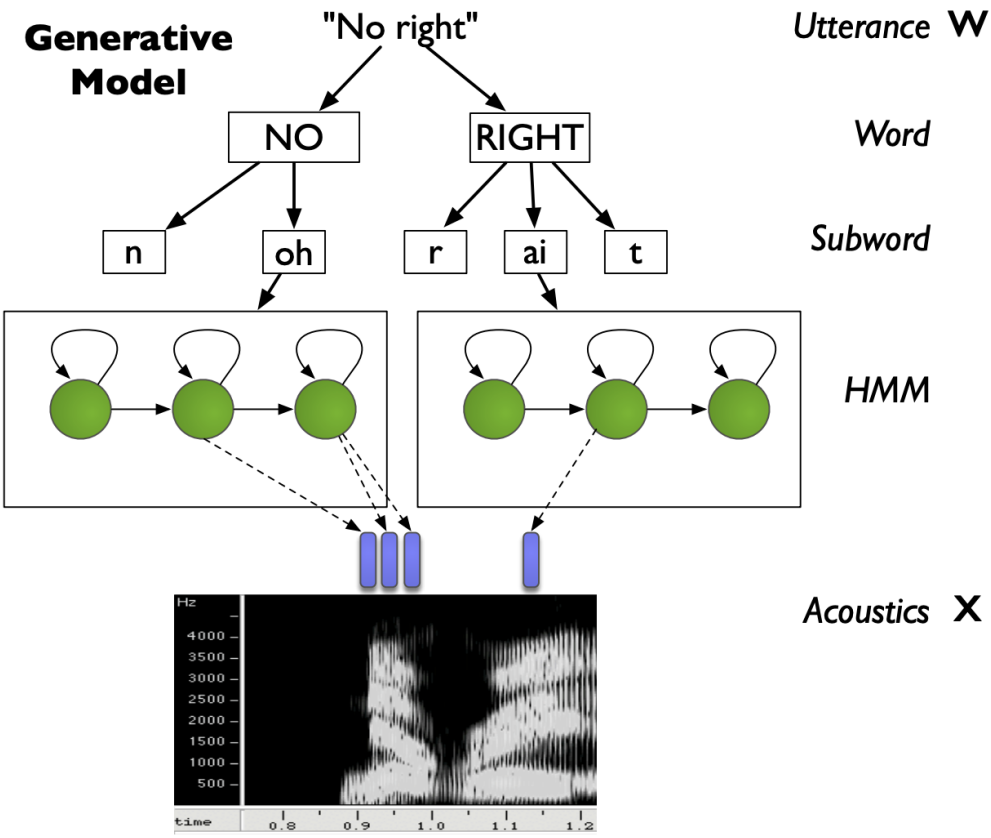

Figure 1.1: Figure of how ASR works.[4]

1.2.1 How it works under the hood:

“Automatic continuous speech recognition (CSR) has many potential applications including command and control, dictation, transcription of recorded speech, searching audio documents and interactive spoken dialogues. The core of all speech recognition systems consists of a set

of statistical models representing the various sounds of the language to be recognised. Since speech has temporal structure and can be encoded as a sequence of spectral vectors spanning the audio frequency range, the hidden Markov model (HMM) provides a natural framework for constructing such models . HMMs lie at the heart of virtually all modern speech recognition systems and although the basic framework has not changed significantly in the last decade or more, the detailed modeling techniques developed within this framework have evolved to a state of considerable sophistication . The result has been steady and significant progress and it is the aim of this review to describe the main techniques by which this has been achieved. The foundations of modern HMM-based continuous speech recognition technology were laid down in the 1970's by groups at CarnegieMellon and IBM who introduced the use of discrete density HMMs 197 198 Introduction, and then later at Bell Labs where continuous density HMMs were introduced. An excellent tutorial covering the basic HMM technologies developed in this period is given in. " [18]

1.2.2 What is the Hidden Markov Process in simple terms?

The Hidden Markov model is a probabilistic model which is used to explain or derive the probabilistic characteristic of any random process. It basically says that an observed event will not be corresponding to its step-by-step status but related to a set of probability distributions.

1.3 How is ASR used in healthcare:

Speech recognition technology has significant impacts on the healthcare profession. The most significant impacts and benefits are:

1.3.1 For Doctors:

1. **Time Savings and Better Productivity:**Medical professionals normally spend a large percentage of their day doing paperwork. That's where speech recognition technologies can have an impact. It takes time to write or type out notes, but it is quicker to speak them aloud. All they have to do is talk into a recording device and the speech recognition technology will translate the spoken word to written word.

2. **Better Quality of Care:**Saving time on paperwork allows medical practitioners to spend more time treating their patients, which improves overall patient care. Instead of spending time sorting through a medical chart or reviewing information on the computer, voice technologies can allow clinicians to truly be present with their patients.

1.3.2 For Patients:

Voice Recognition in medical services has been generally utilized as it gives an open door to patients to get care from home. There are many benefits of involving discourse acknowledgment innovation in the medical care industry. Health care providers and patients can communicate

using speech recognition technology, which helps reduce the need for written documentation. Here are some benefits and uses for Voice Recognition in healthcare:

1. **No need for Face-to-Face Interactions:** It permits the patient to stay associated without up close and personal contact.
2. **Language Dependability:**Assists patients who are unable to communicate traditionally.
3. **Variety in Communication Method:** Wearable monitoring devices permit real-time evaluation and a much faster response time. When a patient's cholesterol or glucose levels begin to rise, these devices can inform doctors immediately so that changes can be made.
4. **Flexibility:**Its flexibility is one of the best aspects of voice recognition technology. Voice recognition can be accessed via virtually any device or interface. Combining the power of the cloud with the ability to share patient information and notes, doctors and medical staff can easily collaborate. This is especially useful when patients need medical care from remote places.
5. **Accuracy and Speed:**Because voice recognition reduces the likelihood of errors during data entry, which can lead to misdiagnoses, voice recognition facilitates diagnosis.

1.4 Voice Assistants:

Voice assistants are digital assistants that use natural language processing (NLP) and automatic speech recognition (ASR) technologies to interact with users through voice commands. Here are some popular voice assistants:

1. **Amazon Alexa:**Alexa is a voice-controlled virtual assistant developed by Amazon. It can perform a variety of tasks, such as setting reminders, playing music, controlling smart home devices, and ordering products.
2. **Google Assistant:** Google Assistant is a virtual assistant developed by Google. It can perform similar tasks to Alexa, such as answering questions, setting reminders, and controlling smart home devices.
3. **Apple Siri:** Siri is a voice-controlled virtual assistant developed by Apple. It is integrated into Apple's ecosystem of products, such as iPhones, iPads, and Macs. It can perform tasks such as sending text messages, setting reminders, and making phone calls.
4. **Microsoft Cortana:** Cortana is a virtual assistant developed by Microsoft. It can perform similar tasks to other voice assistants, such as setting reminders and controlling smart home devices. It is integrated into Microsoft's ecosystem of products, such as Windows 10 and Microsoft Office.

5. **Samsung Bixby:**Bixby is a virtual assistant developed by Samsung. It is integrated into Samsung's ecosystem of products, such as Samsung smartphones and smart home devices. It can perform tasks such as setting reminders, playing music, and making phone calls.

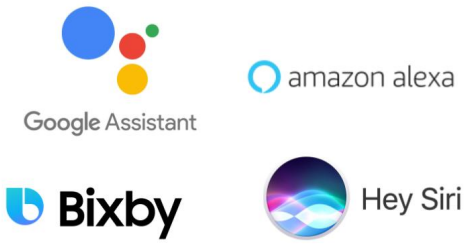

Figure 1.2: Figure showing different voice assistants logos.[8]

1.5 Voice assistant tools used in healthcare:

Voice assistants are being increasingly used in healthcare to help healthcare professionals, patients, and caregivers. Here are some examples:

1. **Orbita Voice:** Orbita Voice is a voice assistant that uses natural language processing and machine learning to interact with patients and caregivers. It can help patients manage chronic conditions, monitor their symptoms, and receive medication reminders.

2. **Aiva Health:** Aiva Health is a voice assistant that can be used in hospitals and long-term care facilities. It can help patients communicate with their caregivers, order meals, and control their environment.

3. **Suki.AI:** Suki.AI is a voice assistant designed for physicians. It can help physicians with clinical documentation, such as taking notes and filling out forms. It can also provide real-time clinical decision support based on patient data.

4. **Buoy Health:** Buoy Health is a voice assistant that can help patients determine their symptoms and receive personalized health recommendations. It uses natural language processing to understand patient input and provide appropriate guidance.

5. **MedWhat:** MedWhat is a voice assistant that can help patients and physicians with medical questions. It uses natural language processing and machine learning to understand complex medical terms and provide accurate answers.

1.6 Conclusion:

We came to an end this chapter and we concluded the potential impacts of Automatic Speech Recognition in the field of healthcare and that it can increase the productivity of doctors and helps patient have access to some degree of medical care anywhere at any time.

Chapter 2

Tools necessary for the realization of the project

2.1 Introduction

Do you know Android has covered 72.26% of the mobile operating system market share across the globe? Yes, Android is consistently leading by keeping all of its competitors behind. Therefore, the demand for Android applications is constantly going up. The reason behind increasing demand is the benefits that Android offers. High-level customization and community support are two of the essential factors to consider. Even, using the same factors our Android developers build interactive apps effectively with the help of Android development IDEs. The IDE that google itself recommends is Android Studio.

2.2 What is 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.It is available for download on Windows, macOS and Linux based operating systems.It provides many excellent features that enhance productivity when building Android apps, such as:

1. A blended environment where one can develop for all Android devices Apply Changes to push code and resource changes to the running app without restarting the app
2. A flexible Gradle-based build system
3. A fast and feature-rich emulator
4. GitHub and Code template integration to assist you to develop common app features and import sample code
5. Extensive testing tools and frameworks
6. C++ and NDK support
7. Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine, and many more.
8. Provides GUI tools that simplify the less interesting parts of app development.
9. Easy integration with real time database 'firebase'.

Android Studio was announced on May 16, 2013, at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014.The first stable build was released in December 2014, starting from version 1.0.At the end of 2015, Google dropped support for Eclipse ADT, making Android Studio the only officially supported IDE for Android development.On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development.Java is still supported, as is C++.[**?**]

Figure 2.1: Figure of the Androids Studio logo.[6]

System Requirements for Android Studio are:

- Microsoft Windows 7/8/10 (32-bit or 64-bit)
- 4 GB RAM minimum, 8 GB RAM recommended (plus 1 GB for the Android Emulator)
- 2 GB of available disk space minimum, 4 GB recommended (500 MB for IDE plus 1.5 GB for Android SDK
- and emulator system image) 1280 x 800 minimum screen resolution

Installation : Head over to this link to get the Android Studio executable or zip file.

2.3 What is Google Speech Recognition?

Google Speech Recognition is a technology that allows users to convert spoken words into text. It is a part of Google's suite of machine learning and artificial intelligence products and services, and is available through various Google products such as Google Search, Google Assistant, and Google Translate.

Google Speech Recognition is powered by deep neural networks and machine learning algorithms, which work in the backend to process and analyze audio data. The technology uses complex algorithms to analyze the unique features of an individual's speech, such as their accent, tone, and intonation, to accurately recognize the words they are speaking. Google Speech Recognition has been trained on large datasets of speech data, allowing it to recognize spoken words with a high degree of accuracy. It is also constantly learning and improving, as it is exposed to more speech data and user feedback. Overall, Google Speech Recognition is a powerful tool that allows users to interact with technology using their voice, and has a wide range of applications such as voice search, dictation, and language translation.[10]

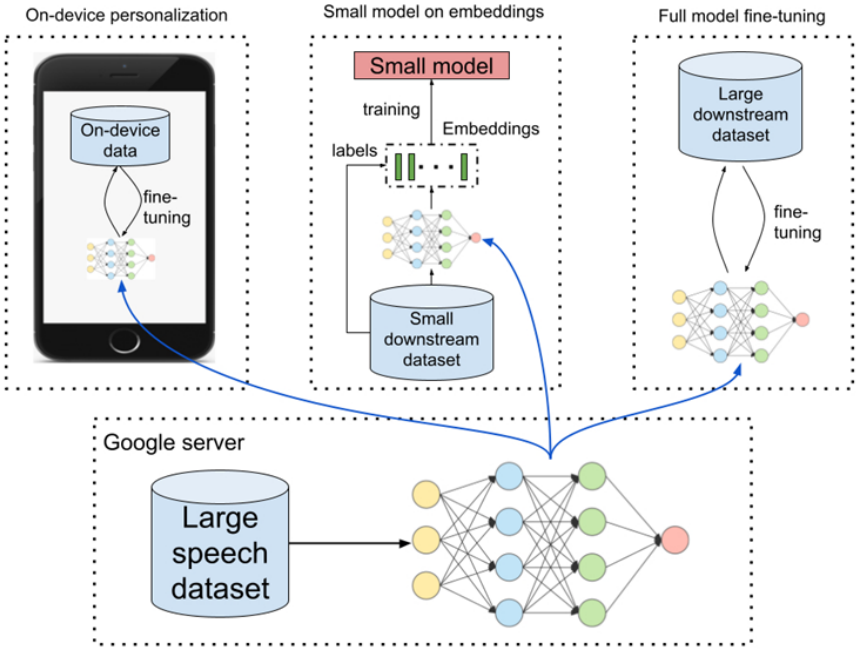

Figure 2.2: Figure show how gooogle speech api works under the hood.[5]

2.3.1 How to implement Google Speech Recognition in Android Studio:

1. First of all, create a new Android Studio project and in the manifest file add the following user-permissions:

2. Secondly, in the activity_main.xml file, we will add an EditText and an ImageView.

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.INTERNET" />
```

Figure 2.3: Screenshot showing how to implement the permisions.

3. Create an instance of the SpeechRecognizer class in your activity or fragment:

```
private SpeechRecognizer speechRecognizer;
```

Figure 2.4: Create a SpeechRecognizer variable.

4. Now here comes the important part first which will initialize the SpeechRecognizer object and then create the intent for recognizing the speech.

```
speechRecognizer = SpeechRecognizer.createSpeechRecognizer( context: this);
final Intent speechRecognizerIntent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
speechRecognizerIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
speechRecognizerIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());
```

Figure 2.5: Screenshot of code to Initialize the Speech recognizer.

5. In the onBeginningOfSpeeh() method we will add the following code to tell the user that his voice is being recognized

6. The onResults() method of the RecognitionListener will be called when the user has finished speaking, and will provide a list of possible transcription matches.

```
@Override
public void onBeginningOfSpeech() {
    editText.setText("");
    editText.setHint("Listening...");
}
```

Figure 2.6: Screenshot of OnBeginning Method.

```
@Override
public void onResults(Bundle bundle) {
    micButton.setImageResource(R.drawable.ic_mic_black_off);
    ArrayList<String> data = bundle.getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION);
        editText.setText(data.get(0));
}
```

Figure 2.7: Screenshot of OnResults Method

7. To be able to start the speech recognition in Arabic we simply change this line in the speechRecognizer initialisation.

```
speechRecognizer = SpeechRecognizer.createSpeechRecognizer( context: this);
final Intent speechRecognizerIntent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
speechRecognizerIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
speechRecognizerIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, value: "ar-AR");
```

Figure 2.8: Figure show how gooogle speech api works under the hood.

2.3.2 API Text to Speech in Android Studio:

Google Text-to-Speech (TTS) is a cloud-based service that converts text into spoken words. It uses Google's advanced deep learning neural network algorithms to produce natural-sounding speech.
Google TTS is available as a free service to Android developers, and can be easily integrated into Android applications using the Android TextToSpeech API. It supports many languages and voices, and can even be customized to sound like specific people or characters.[16] In order to implement it:

1. create an instance of the TextToSpeech.
2. Initialize the TextToSpeech in the onCreate() method:
3. To convert text to speech, call the speak() method of the TextToSpeech class:

```
ttp=new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener() {
    @Override
    public void onInit(int status) {
        if(status != TextToSpeech.ERROR) {
            ttp.setLanguage(Locale.UK);
        }
    }
});
```

Figure 2.9: Screenshot showing how to write the constructor of the text to speech.

```
ttp.speak( text: "hello", TextToSpeech.QUEUE_FLUSH, params: null);
```

Figure 2.10: Screenshot showing the call of the speak method.

2.4 CRUD manipulation:

We are going to use the one and only XAMPP for the backend service.

2.4.1 XAMPP:

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends,consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages,Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.[17]

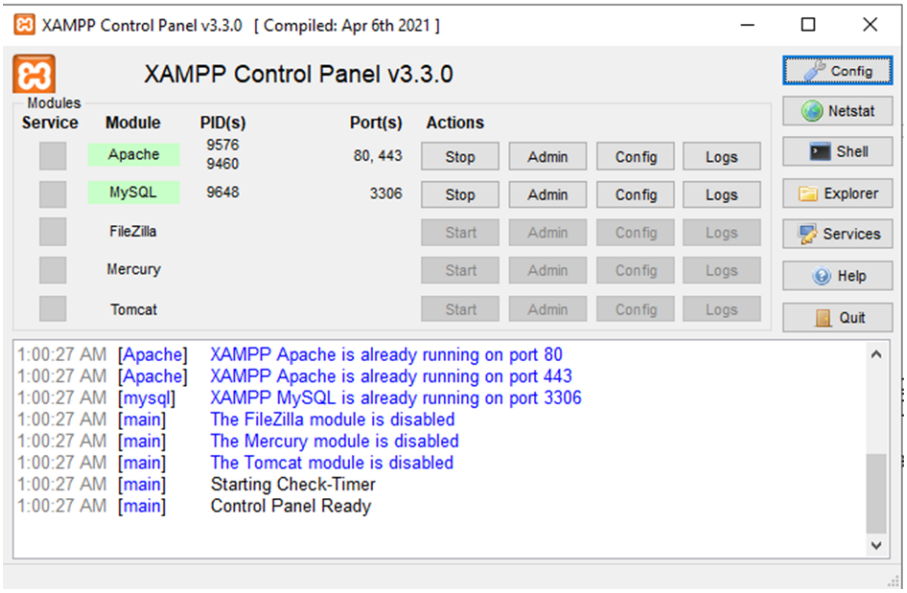

Figure 2.11: Screenshot of XAMPP interface.

And to send the http requests to interact with the backend we used Volley Library.

2.4.2 Volley:

Volley is an HTTP library that makes networking very easy and fast, for Android apps. It was developed by Google and introduced during Google I/O 2013. It was developed because there is an absence in Android SDK, of a networking class capable of working without interfering with the user experience. Although Volley is a part of the Android Open Source Project(AOSP), Google announced in January 2017 that Volley will move to a standalone library. It manages the processing and caching of network requests and it saves developers valuable time from writing the same network call/cache code again and again. Volley is not suitable for large download or streaming operations since Volley holds all responses in memory during parsing. [14]

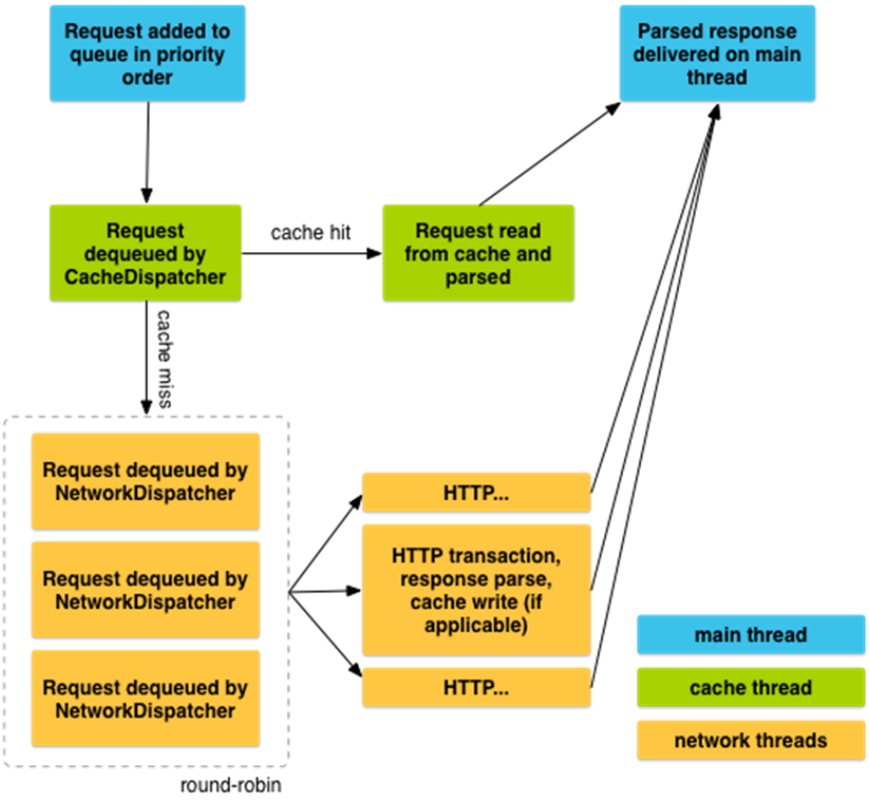

Figure 2.12: Figure shows how Volley Threads words.[15]

To start using volley we should add it to the dependencies like so:

```
implementation 'com.android.volley:volley:1.2.1'
```

Figure 2.13: Implementation for volley in the dependencies

```
RequestQueue queue = Volley.newRequestQueue( context: MainActivity.this);
StringRequest strRequest = new StringRequest(Request.Method.POST, url,
        new Response.Listener<String>()
        {
            @Override
            public void onResponse(String response)
            {
                Toast.makeText(getApplicationContext(), response, Toast.LENGTH_SHORT).show();
                editText.setText(response);
                ttp.speak(response, TextToSpeech.QUEUE_FLUSH,  params: null);
            }
        },
        new Response.ErrorListener()
        {
            2 usages
            @Override
            public void onErrorResponse(VolleyError error)
            {
                Toast.makeText(getApplicationContext(), error.toString(), Toast.LENGTH_SHORT).show();
            }
        })
{
    @Override
    protected Map<String, String> getParams() throws AuthFailureError
    {
        Map<String, String> datasent = new HashMap<~>();
        datasent.put( k: "speech", data.get(0));
        return datasent;
    }
};
queue.add(strRequest);
}
```

Figure 2.14: In this block of code we send the result of the recognition in the http request to the backend.

2.4.3 PHP:

PHP is a programming language used to script websites that are dynamic and interactive. You'll find it in various types of web applications, from e-commerce websites to CRM systems like HubSpot and Salesforce.
The term PHP stands for PHP Hypertext Preprocessor.
We used PHP for its popularity and to create a fully customizable backend.[1]

```php
<?php
    $servername = "localhost:3306";
    $username = "root";
    $password = "";
    $dbname = "si";

    // Create connection
    $connection = new mysqli($servername, $username, $password, $dbname);

  //fetch table rows from mysql db
  $id=$_POST['id'];
  $sql = "select p.id,filename,d.name,time FROM pdf_data p,doctors d WHERE p.doctor_id=d.id and p.user_id=$id";
  $result = mysqli_query($connection, $sql) or die("Error in Selecting " . mysqli_error($connection));

  //create an array
  $emparray = array();
  while($row =mysqli_fetch_assoc($result))
  {
      $emparray[] = $row;
  }
  echo json_encode($emparray);

  //close the db connection
  mysqli_close($connection);
?>
```

Figure 2.15: Screenshot showing and example of a php code used in the project.

2.4.4 MariaDB:

MariaDB is an open-source relational database management system (RDBMS) that is a fork of the popular MySQL database system. It was developed by the original creators of MySQL after concerns arose over the acquisition of MySQL by Oracle Corporation. MariaDB aims to provide a compatible, drop-in replacement for MySQL while offering additional features, performance enhancements, and better community-driven development. We are using for the following reasons:
Compatibility: MariaDB is designed to be a drop-in replacement for MySQL, which means that most applications and systems that work with MySQL can seamlessly switch to MariaDB without any major modifications.
Performance: MariaDB incorporates several performance improvements and optimizations over MySQL. It includes enhanced storage engines, query optimizations, and parallel processing capabilities, resulting in improved query execution and overall database performance.
Security: MariaDB offers various security features to protect data and ensure database integrity. It supports secure connections using SSL/TLS encryption, provides authentication mechanisms, and has advanced access control features to manage user privileges and permissions.

2.5 Conclusion:

In order to build a successful project we need a set of tools from IDEs to libraries to infrastructure. STT and TTS technologies have transformed the way we communicate with computers, and their continued development will undoubtedly lead to even more innovative and exciting applications in the future.
Down the line we can add even more libraries and features to make the application as interactive as possible, we are only limited by our imagination .

Chapter 3

Implementation and results

3.1 Introduction

Throughout this chapter, we will explore various UML diagrams, such as use case diagrams, class diagrams, sequence diagrams, and activity diagrams. Each diagram type serves a distinct purpose, offering unique insights into the system's behavior, structure, and interactions. We will discuss their construction, interpretation, and practical application, allowing you to harness the full potential of UML in your software development endeavors. Additionally, we will discuss application execution and the testing done to insure the requirements are met.

3.2 Part 1: Conception

3.2.1 UML definition

The unified modeling language (UML) is a general-purpose modeling language that is intended to provide a standard way to visualize the design of a system.
UML provides a standard notation for many types of diagrams which can be roughly divided into 3 main groups: behavior diagrams, interaction diagrams, and structure diagrams.
The creation of UML was originally motivated by the desire to standardize the disparate notational systems and approaches to software design. It was developed at Rational Software in 1994–1995, with further development led by them through 1996.[12]

Figure 3.1: UML logo.[7]

3.2.2 Use case diagram

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.[13]
Figure below shows the Use case diagram for our project:

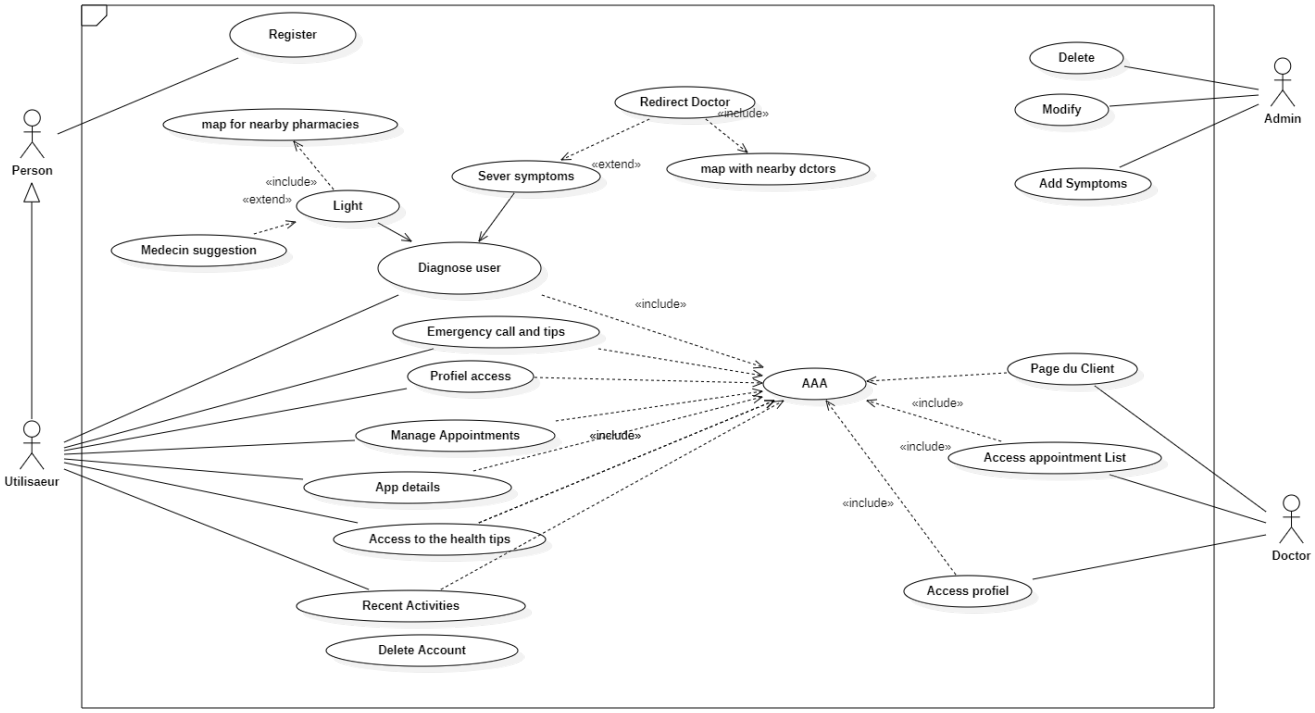

Figure 3.2: Use case diagram

3.2.3 Sequence Diagram

A sequence diagram or system sequence diagram (SSD) shows process interactions arranged in time sequence in the field of software engineering. It depicts the processes and objects involved and the sequence of messages exchanged between the processes and objects needed to carry out the functionality.[11]
The figure below shows our project use case diagram:

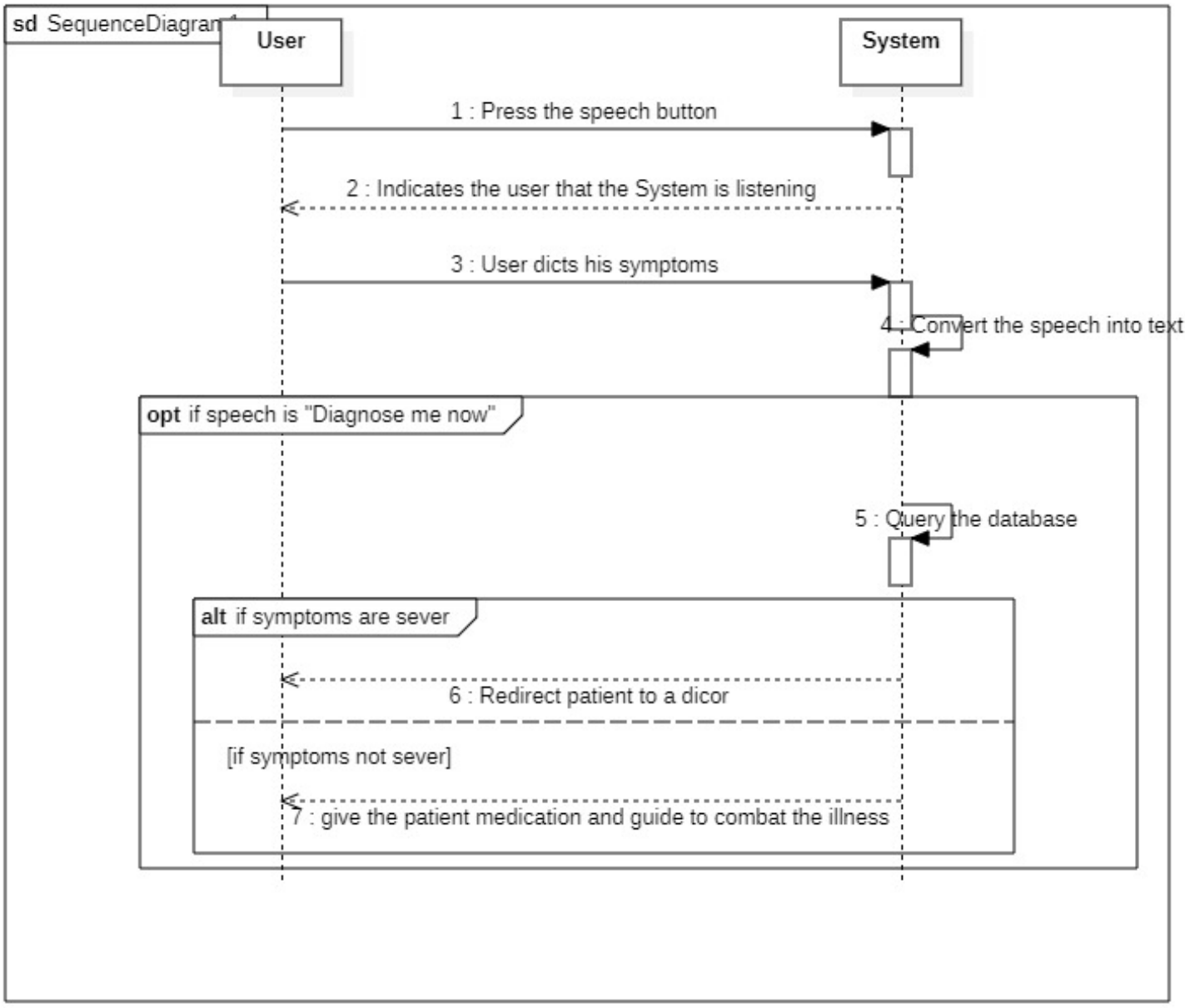

Figure 3.3: Sequence Diagram

3.2.4 Class Diagram

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling, translating the models into programming code.[3]
The figure below shows the class diagram for our project:

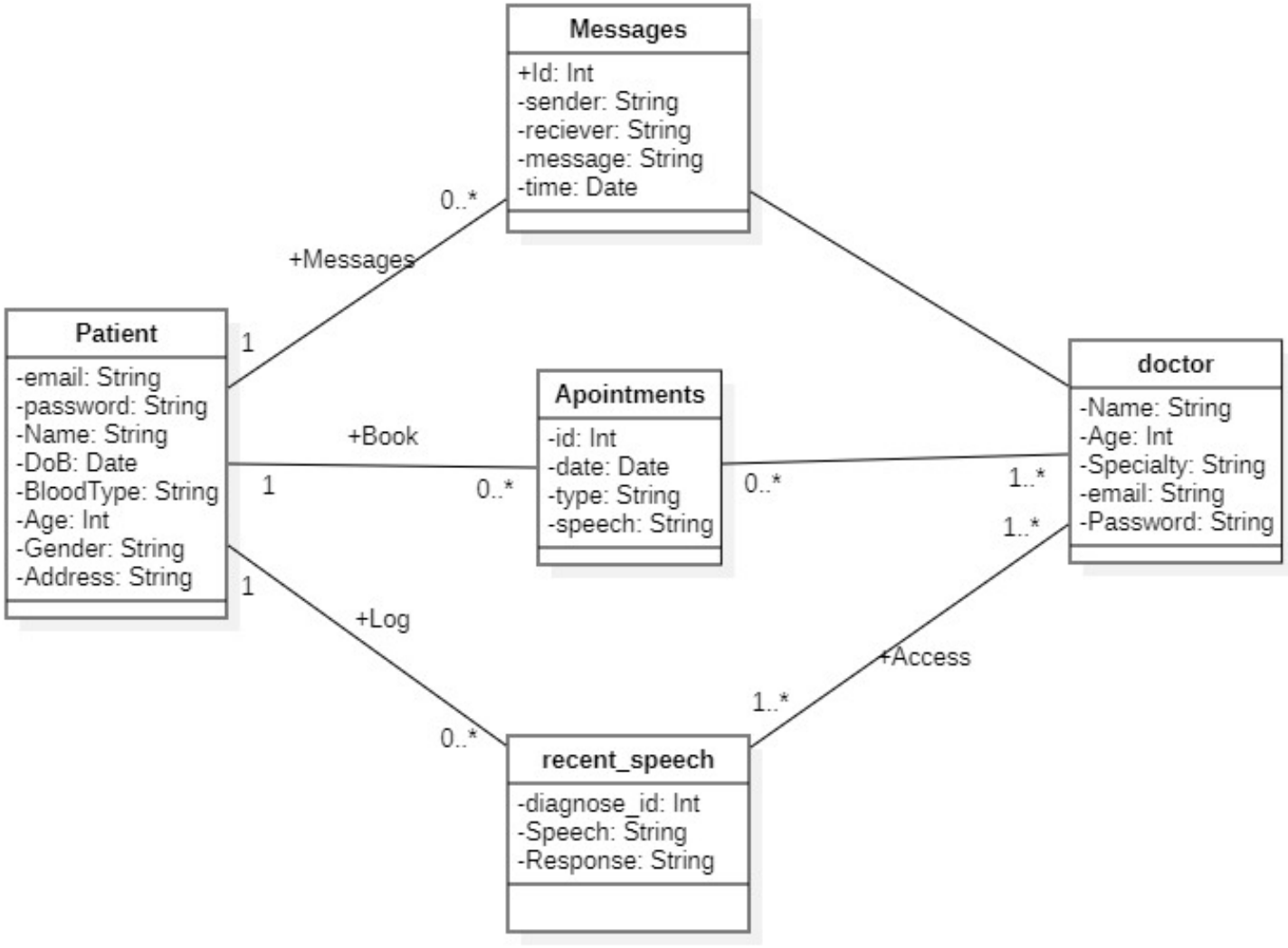

Figure 3.4: Class Diagram

3.2.5 Activity Diagram

Activity diagrams are graphical representations of workflows of step wise activities and actions with support for choice, iteration and concurrency.[2]
Figure below shows the activity diagram for our project:

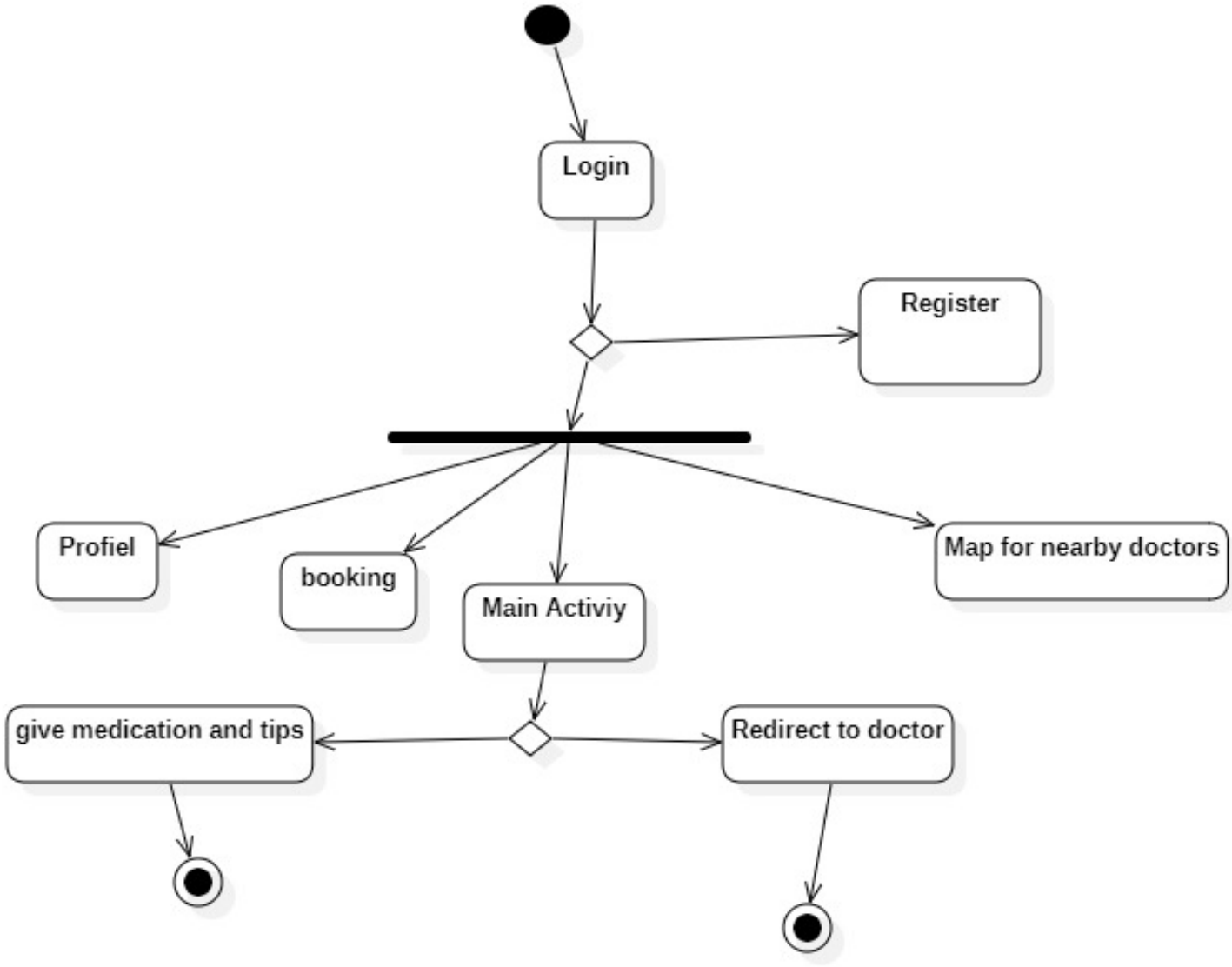

Figure 3.5: Activity Diagram

3.3 Part 2: Application execution

Table of the steps:

1	Launch the app from your mobile phone
2	Authenticate yourself by inputting your email and password
3	Now you are in the main screen now press the + symbol at the button to open up a list of options
4	Press Diagnose to get directed to the diagnose screen.
5	Press the microphone icon and dict your symptoms one by one
6	after you are all set press the microphone button again and say diagnose me to start the dianose process
7	if there is a result the app will direct you to another page with the illness name and the tip you should do to combat it if not te app simply tells you to contact a doctor

3.3.1 Application Screenshots:

Additional features not shown in the screenshots:
Integration of wikipedia api .
Record of previous sessions.
Booking with nearby doctors.
A New Health tips every time you load the app.
Weather Tracking.
Google Maps with easy access to nearby hospitals and pharmacies.
Traffic over the internet with encrypted traffic.(the app is ready to be deployed).
Add funds to be able to Diagnose and book an appointment with a doctor.
Download and access medical files sent my doctors.
Able to receive messages for doctors.

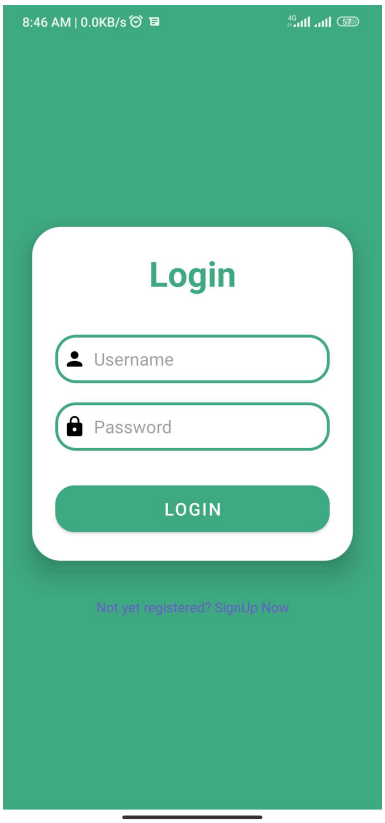

Figure 3.6: Authentication page

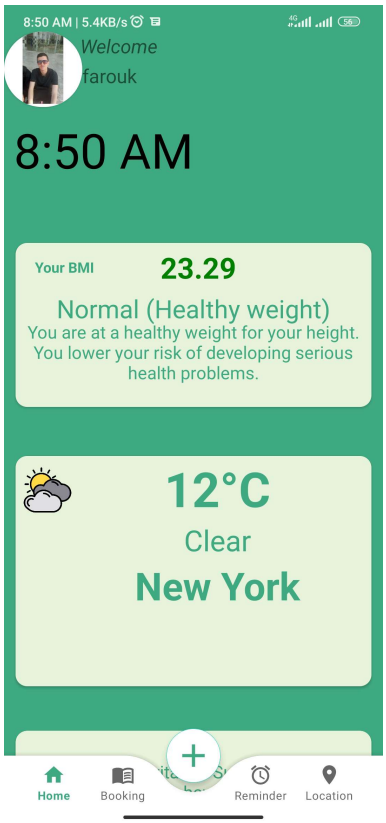

Figure 3.7: Main page

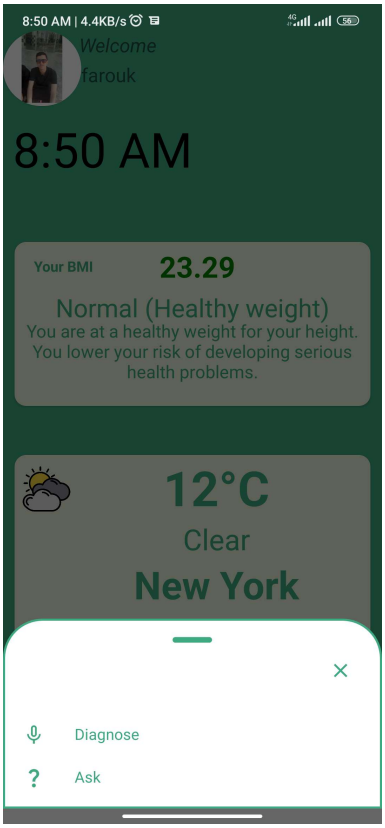

Figure 3.8: After pressing the main button

Figure 3.9: Diagnose page

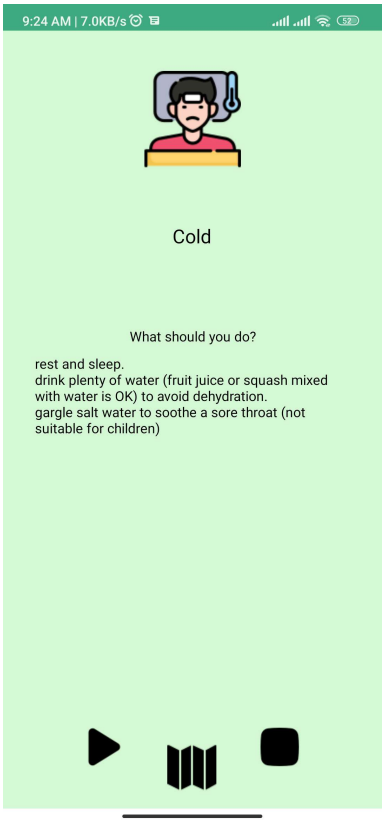

Figure 3.10: if symptoms are light

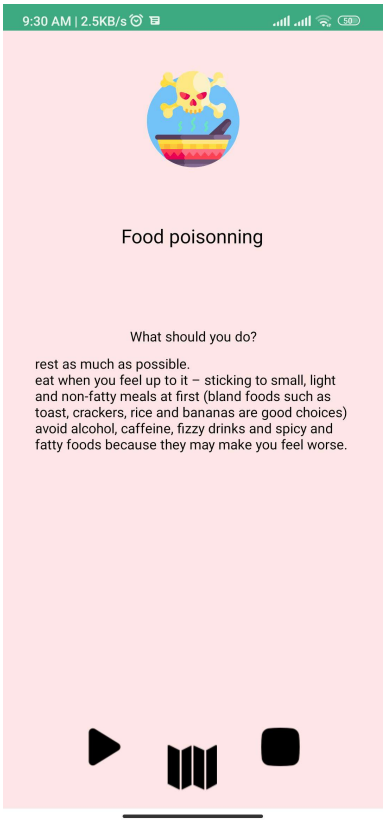

Figure 3.11: if symptoms are severe

3.3.2 Evaluating the performance of the application:

Word error rate (WER): is the ratio of errors in a transcript to the total words spoken. A lower WER in speech-to-text means better accuracy in recognizing speech,its calculated as followed:

$$\text{WER} = \frac{S + D + I}{N}$$

S is the number of substitutions,

D is the number of deletions,

I is the number of insertions,

N is the number of words in the reference ($N = S + D + C$),

True Positive Rate (TPR): refers to cases where the diagnostic app correctly identifies a person as sick, and that person is indeed sick,its calculated as followed: TPR = TP / (TP+FN)
True Negative Rate(TNR): refers to cases where the diagnostic app correctly identifies a person as sick, and that the person is actually healthy,its calculated as followed: TNR = TN / (TN+FP))
False Positive Rate(FPR): refers to cases where the diagnostic app correctly identifies a person as healthy, and that person is actually healthy,its calculated as followed: FNR = FN / (FN+TP))
False Negative Rate(FNR): refers to cases where the diagnostic app correctly identifies a

person as healthy, and that person is indeed healthy,its calculated as followed: TPR = TP / (TP+FN)

Table of results

To obtain a more precise calculation, we would ideally need a larger and diverse set of volunteers, including both sick and healthy individuals, to test the rates. However, due to limitations, we were only able to stimulate the testing process with a group of 8 people. We provided them with various symptoms and allowed them to interact with the app. Based on their interactions, we obtained the following results.

Word Error Rate	WER = (S + D + I)/N	WER=8 / 60 = 0.13
True Positive Rate	TPR = TP / (TP+FN)	TPR=14/14+2=0.875
True Negative Rate	TNR = TN / (TN+FP)	TNR=4/4+12=0.25
False Positive Rate	FPR = FP / (FP+TN)	FPR=12/12+4=0.75
False Negative	FNR = FN / (FN+TP)	FNR=2/2+14=0.125
Recall	R=TP / (TP+FN)	R=14/14+2=0.875
Precision	P= TP / (TP+FP)	P=14/12+14=0.5384

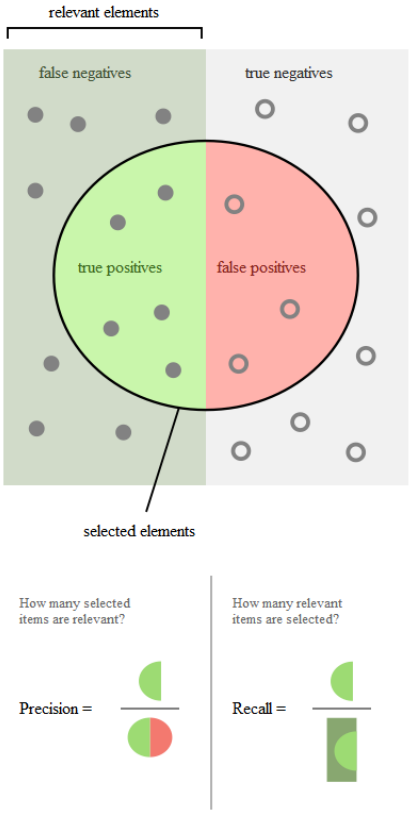

Figure 3.12: Sequence Diagram[9]

3.4 Conclusion:

We got to the end of the last chapter and conclusion, conceptions diagrams play a crucial role in the development process by providing a visual representation of complex ideas, relationships, and concepts. These diagrams help to clarify and organize information, making it easier for developers to understand and contribute to development initiatives.Overall, conceptions diagrams are an essential tool in the development toolkit, aiding in knowledge sharing, problem-solving, and effective decision-making.

After the app has been materialized, it enters a new phase of the development process where it is deployed and made available to users,after testing of all the features and the accuracy of the results.

General Conclusion and Prespectives

In conclusion, this thesis has examined the application of Automatic Speech Recognition (ASR) technology in healthcare, specifically focusing on its potential in the diagnostic process. We found that ASR holds great promise in enhancing the efficiency, accuracy, and effectiveness of diagnosing medical conditions.

However, it is important to acknowledge that challenges exist in the successful implementation of ASR technology in diagnosing medical conditions.Such as the lack of medical information in the database, this challenge can be addressed by incorporating medical knowledge and experience into the database through the expertise of a team of medical staff.The second challenge is not taking into consideration the previous diagnostic sessions and the patients' health records, This challenge can be solved by incorporating Electronic Health Records (EHR) into the algorithm used for diagnosis.

And other features we can use to improve the diagnosis accuracy.Like the use of sensors in smart gadgets to monitor in real time the patient health such as heart rate,oxygen levels and temperature.and the use of the latest deep neural network methods with a continuous improvement overtime and learning form past mistake so the diagnose can be as much accurate as possible.And the addition of medical terms in the speech recognition.

ASR has the capacity to significantly improve diagnostic outcomes, enhance patient care, and shape the future of healthcare.

Bibliography

[1] Code academy. https://www.codecademy.com/resources/blog/what-is-php-used-for/#:~:text=PHP.

[2] Wikipidia Activity. https://en.wikipedia.org/wiki/Activity_diagram.

[3] Wikipidia Class diagram. https://en.wikipedia.org/wiki/Class_diagram.

[4] Maël Fabien. https://maelfabien.github.io/machinelearning/speech_reco_1/#introduction-to-hmm-gmm-acoustic-modeling.

[5] googlespeechapi. https://ai.googleblog.com/2020/06/improving-speech-representations-and.html.

[6] Android Studio Logo. https://en.wikipedia.org/wiki/Activity_diagram.

[7] Uml logo. https://en.wikipedia.org/wiki/File:UML_logo.svg.

[8] Voice Assistant Logos. https://www.slanglabs.in/think-voice/voice-assistant-for-apps.

[9] Wikipedia Precision and recall. https://en.wikipedia.org/wiki/Precision_and_recall.

[10] Abhinav Singh. https://medium.com/voice-tech-podcast/android-speech-to-text-tutorial-8f6fa71606ac.

[11] Wikipidia Sequence System. https://en.wikipedia.org/wiki/Sequence_diagram.

[12] Wikipidia UML. https://en.wikipedia.org/wiki/Unified_Modeling_Language.

[13] Wikipidia Usecase. https://en.wikipedia.org/wiki/Use_case_diagram.

[14] Volley. https://www.geeksforgeeks.org/volley-library-in-android/.

[15] volley figure. https://google.github.io/volley/simple.html.

[16] www.tutorialspoint.com. https://www.tutorialspoint.com/android/android_text_to_speech.htm.

[17] XAMPP. https://en.wikipedia.org/wiki/XAMPP.

[18] Steve J Young. Statistical modeling in continuous speech recognition (csr). 1:562–571, 2001.