



***İstanbul Aydın University
Department of Computer Engineering
ENGINEERING PROJECT***

***HOSFACE
(HOSPITAL APPOINTMENT SYSTEM)***

Advisor: Prof. Dr. ÖGR Üyesi AHMET GÜRHANLI

Group Members:

NURŞAH DEMİRPOLAT – B1605.010037

ZEYNEP GİZEM ÇETİNCİ – B1605.010034

Table Of Contents

1. INTRODUCTION.....	6
2. SYSTEM ANALYSIS.....	8
2.1 EXISTING SYSTEM.....	8
2.2 PROPOSED SYSTEM.....	8
2.3 LIFE CYCLE MODEL	8
2.4 FEASIBILITY STUDY	9
2.4.1 ECONOMIC FEASIBILITY	9
2.4.2 TECHNICAL FEASIBILITY	9
2.4.3 BEHAVIOURAL FEASIBILITY	10
2.5 SOFTWARE REQUIREMENT SPECIFICATION	10
2.5.1 INTRODUCTION	10
2.5.2 OBJECTIVE.....	10
2.5.3 DOCUMENT CONVENTION	10
2.5.4 INTENDED AUDIENCE AND READING SUGGESTIONS	10
2.5.5 PROJECT FEATURES	11
2.5.6 USER CLASSES AND CHARACTERISTICS	11
2.5.7 OPERATING ENVIRONMENT	11
2.6 HARDWARE REQUIREMENTS.....	12
2.7 SOFTWARE REQUIREMENTS	12
2.8 GANTT CHART.....	13
2.9 COST ESTIMATION	13
3. SYSTEM DESIGN.....	16
3.1 INTRODUCTION.....	16
3.1.1 PURPOSE.....	16
3.1.2 SCOPE OF DESIGN	17
3.2 SPECIFICATION	17
3.2.1 MODULES	17
3.3 USE CASE DIAGRAM	18
3.3.1 PURPOSE.....	18
3.4 FACE RECOGNITION DIAGRAM	19
3.4.1 PURPOSE.....	19
3.5 DATA FLOW DIAGRAMS	20
3.5.1 PURPOSE.....	20
3.6 SQLITE DATABASE DIAGRAMS	21
3.6.1 PURPOSE.....	21

3.6.2 TABLE	22
4. COMPONENT DESCRIPTION	25
4.1 TYPE.....	27
4.2 PURPOSE	27
4.3 FUNCTION	28
4.4 SUBORDINATES	29
4.5 DEPENDENCIES	30
4.6 RESOURCES.....	32
4.7 PROCESSING	33
5. SYSTEM IMPLEMENTATION.....	35
5.1 LANGUAGE	35
5.1.1 FEATURES	35
5.1.2 ARCHITECTURE OF PYTHON.....	37
5.1.3 LIBRARIES.....	38
6. SCREENSHOTS	52
6.1 SCREENSHOTS FOR UNKNOWN REGISTER.....	52
6.1.1 HOSFACE WELCOME PAGE	52
6.1.2 HOSFACE WEBCAM PAGE – UNKNOWN REGISTER.....	53
6.1.3 HOSFACE TC IDENTITY CHECK PAGE	54
6.1.4 HOSFACE PATIENT REGISTRATION PAGE	55
6.1.5 HOSFACE SELECT POLICLINIC PAGE.....	57
6.1.6 HOSFACE QUEUE NUMBER PAGE	57
6.2 SCREEN SHOTS FOR KNOWN REGISTER.....	58
6.2.1 HOSFACE WELCOME PAGE	58
6.2.2 HOSFACE WEBCAM PAGE – KNOWN REGISTER	59
6.2.3 HOSFACE TC IDENTITY CHECK PAGE	60
6.2.4 HOSFACE SELECT POLICLINIC PAGE.....	60
6.2.5 HOSFACE QUEUE NUMBER PAGE	61
7. SYSTEM TESTING	63
7.1 MANUAL TESTING	63
7.1.1 TESTING FOR HARDWARE.....	63
7.1.2 TESTING FOR FACE RECOGNITION	66
7.2 UNIT TESTING.....	69
8. FUTURE SCOPE	71
9. CONCLUSION.....	73
10. REFERENCES	75

11.CODE	77
11.1 MAIN.PY	77
11.2 ACILISUİ.PY	90
11.3 ARAMAUİ.PY.....	90
11.4 HASTAARAMAUİ.PY	91
11.5 İKİNCİUİ.PY	92
11.6 İLKUİ.PY	93
11.7 UCUİ.PY	97

INTRODUCTION

1. INTRODUCTION

We have taken our project into consideration the current global problem of coronavirus (COVID-19). One of the ways to end this disease is to stay away from contact and since hospitals are one of the places where it is seen most, we decided to create a face recognition recording system for hospitals that is controlled only by voice and a queuing system from this recording system. In this way, we aim to reduce the crowded waiting order at the reception or to reduce the coronavirus that can be transmitted to the screen on touch screen tablets.

The name of our project is hosface.

- hos - hospital
- fa - face
- ce - voice

SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

Nowadays, when you go to any hospital, you need to go to the floor of that department if you want to make an appointment. Afterwards, you can make an appointment by saying the doctor you want to be examined in counseling. After these processes are finished, you can go to the doctor's room and you can be inspected when your turn comes to the doctor's room according to the number or name letters from the screens on the top.

2.2 PROPOSED SYSTEM

If you want to make an appointment with the hospital, you can make an appointment very quickly by using Hosface machines located at the entrance of the hospital.

In order to do this, the first thing you need to do is to start the system by saying the 'Open' command to the microphone of the hosface machine.

As a result of detecting your face in face scanning, you can select the section you want to go to and get a sequence number directly, and then you can see the appointments you have come before.

In addition, if you are not registered to the hospital, you can register from the registration department as a result of the face scan failing to recognize you. In this way, you can quickly make an appointment the next time you come.

2.3 LIFE CYCLE MODEL

The paradigm chosen for current project was iterative waterfall model. The model consists of six distinct stages, namely:

- Requirements analysis phase
- Specification phase
- System and software design phase
- Implementation and testing phase
- Integration and system testing phase
- Maintenance phase

Reasons for choosing Iterative Waterfall model:

- Testing is inherent to every phase of the model

- It is an enforced disciplined approach
- Work flow is very simple.
- It is documentation driven, that is, documentation is produced at every stage.

2.4 FEASIBILITY STUDY

The main objective of this study is to determine whether the proposed system is feasible or not. Mainly there are three types of feasibility study to which the proposed system is subjected as described below.

Three key considerations involved in the feasibility are are :

- Economic Feasibility
- Technical Feasibility
- Behavioral Feasibility

The proposed system must be evaluated from a technical viewpoint first, and if technically feasible, their impact on the organization must be assessed. If compatible, the behavioral system can be devised. Then those must be tested for economic feasibility.

2.4.1 ECONOMIC FEASIBILITY

Economic feasibility refers to the ability of proposed system is to provide cost effective data processing. In other words all the benefit of the proposed system is economically feasible.

One of the important factors for the operation of our project is a computer with an operating system. In addition, other important factors include camera and microphone devices. In order for the systems to work properly, each machine must have the same capacity and the same quality. In this case, not every hospital can meet the specified criteria economically.

2.4.2 TECHNICAL FEASIBILITY

The Feasibility center on the existing computer system (hardware, software) and to what extend it can support the proposed addition. The computers are sufficient to implement the project. Therefore the system is technically feasible.

2.4.3 BEHAVIOURAL FEASIBILITY

People are inherently resistant to change and computer has known to facilitate change. Technology not mastered many people may have a little difficulty in this case.

2.5 SOFTWARE REQUIREMENT SPECIFICATION

2.5.1 INTRODUCTION

The following subsections are an overview of the entire software requirements Specification document for 'Hospital Appointment System'. This SRS covers only the requirements of this project.

2.5.2 OBJECTIVE

The hospital appointment system gives the patient a sequence number of the desired department by accessing the patient's information as a result of detecting the patient's face. It keeps the information of each patient in the sqlite database. In this way, all information can be accessed simultaneously.

2.5.3 DOCUMENT CONVENTION

Here we include the specification of particular software product, program, or set of programs that performs certain functions in a specific environment. We have followed IEEE standards for preparing the software requirement specification.

2.5.4 INTENDED AUDIENCE AND READING SUGGESTIONS

This project was developed by two computer engineer candidates from Istanbul Aydın University Computer Engineering Department for hospitals and patients.

2.5.5 PROJECT FEATURES

The main features of hosface system which makes it different from others are:

- Thanks to face recognition, an appointment can be made quickly.
- There is a voice command system for people who do not know how to use a keyboard.
- He can easily see his old appointments.
- He can make an appointment quickly without the need for anyone.
- They don't have to share their Tc identification number with any other person.

2.5.6 USER CLASSES AND CHARACTERISTICS

Patients: Normal users who use the project to make an appointment at the hospital.

When the application starts, the gui interface will be displayed, which will directly inform you. Our application is a python platform and is based on windows operating system. It also includes a sqlite database to keep data.

2.5.7 OPERATING ENVIRONMENT

Our product is intended to work in a hospital environment. This device will be a windows operating system installed on the hosface device in the hospital.

2.6 HARDWARE REQUIREMENTS

In order for our application to work properly, it must have a hardware system that meets at least the following features:

- Processor: Intel® Core™ i7-7500U 2.7GHz Processor (Max Turbo Frequency 3.5GHz, 4MB, dual core, 25W TDP, with Intel® HD Graphics 620)
- System Memory: 8GB DDR4
- Max. Memory: Max. 16GB DDR4
- Hard Drive Capacity: 1TB 5400Rpm 2.5 "SATA
- Optical Drive: DVD +/- RW Super Multi
- Graphics: AMD Radeon 530 2GB VRAM
- Windows Type: Windows 10 and above
- Camera: 720p HD Camera
- Sound: Stereo sound system with Dolby® Audio™

2.7 SOFTWARE REQUIREMENTS

While developing our project, we used the following software languages and platforms.

- Operating System: Windows
- Language: Python (3.6)
- Front End: PyCharm
- Back End: SQLite

2.8 GANTT CHART

A Gantt chart is a type of bar chart that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project. Some Gantt charts also show the dependency (i.e., precedence network) relationships between activities.

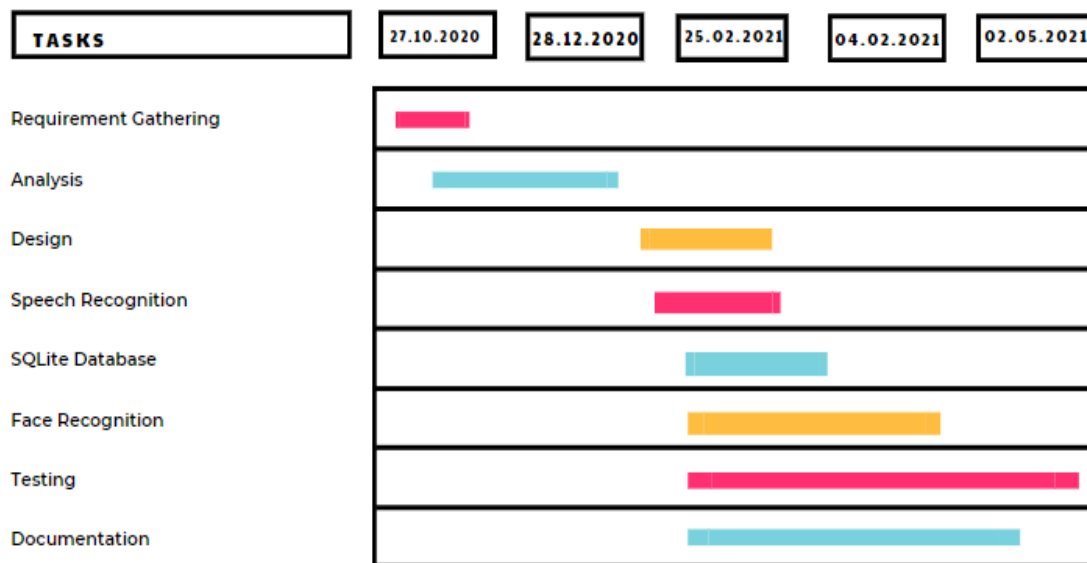


Fig 2.1 Gantt chart

2.9 COST ESTIMATION

Basic COCOMO computes software development effort (and cost) as a function of program size. Program size is expressed in estimated thousands of lines of code (KLOC).

COCOMO applies to three classes of software projects:

- Organic projects - "small" teams with "good" experience working with "less than rigid" requirements
- Semi-detached projects - "medium" teams with mixed experience working with a mix of rigid and less than rigid requirements
- Embedded projects - developed within a set of "tight" constraints (hardware, software, operational, ...)

Based on the information written above, we came to the conclusion that our project is an organic project. That's why we adjusted our calculations accordingly.

The basic COCOMO equations take the form

$$\text{Effort Applied} = 2.4 * (0.624)^{1.05} [2-3] = 1,4625$$

$$\text{Development Time} = 2.5 * (4,3848)^{0.35} [2-3] = 4,1937$$

$$\text{People required} = 4,3848 / 12,5812 * 3 = 0,11$$

The coefficients a_b , b_b , c_b and d_b are given in the following table.

Software project	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.35

Table 2.1 COCOMO Model Coefficients

Cost Estimation of hosface:

- Lines of Code (LOC) : 624
- Kilo lines of code : 0.624
- Effort for Development: 0.635 PM
- Time for development, 3 months
- No of members: 2
- Salary of each member: Rs 2000 /-
- Price of the Product: Rs 18000/-

SYSTEM DESIGN

3. SYSTEM DESIGN

3.1 INTRODUCTION

3.1.1 PURPOSE

Design of the system includes mainly two steps: System design and detailed design.

In system design a structural framework for the entire system is created. It is done in such a way that related part come under particular groups. Thus after the system design, a network of different groups is obtained. It is the high-level strategy for solving the problem and building a solution. It includes the decision about the organization of the system into subsystems, the allocation of subsystems to hardware and software components, and major conceptual and policy decisions that form the framework for the detailed design. System design is the first stage in which the basic approach to solving problem is selected. During system design, the overall style and structure are decided.

In detailed design, each group is studied in detail and the internal operations are decided. Based on this, the data structures and the programming language to be used are decided. Apart from detailed design, the system design can be grouped into physical design and structural design. The physical design maps out the details of the physical system and plans the system implementation and specifies the hardware and software requirements. Structured design is an attempt to minimize the complexity and make a problem manageable by subdividing into smaller segments, which is called modularization or decomposition.

This way structuring minimizes intuitive reasoning and promotes maintainable provable systems. The structured design partitions a program into small, independent modules. They are arranged in a hierarchy that approximates a model of the business are and is organized in a top-down manner. Logical design proceeds in a top-down manner. General features, such as reports and inputs are identified first. Then each is studied individually and in more detail. Hence the structured design is an attempt to minimize the complexity and make a problem.

3.1.2 SCOPE OF DESIGN

In our design, we include the following diagrams to illustrate the flow .

- Use Case diagram
- Face Recognition diagram
- Data Flow diagram
- SQLite Database diagram

3.2 SPECIFICATION

The following are the specifications that are used to develop the proposed system:

3.2.1 MODULES

3.2.1.1 Speech Recognition

The user can activate the voice command by pressing the buttons and print his / her information aloud without the need to use a keyboard.

3.2.1.2 SQLite Database

Patient information (Tc identification number, name, surname, date of birth, appointments) are kept in the SQLite database. Thanks to this section, TR ID number allows to identify each person one by one and allows to show patient information.

3.2.1.3 Face Recognition

Thanks to face recognition, it is ensured that the patient receives a sequence number directly without the need for any action. If the patient is not registered in the system, the system registration section can be opened and registered in the system.

3.3 USE CASE DIAGRAM

3.3.1 PURPOSE

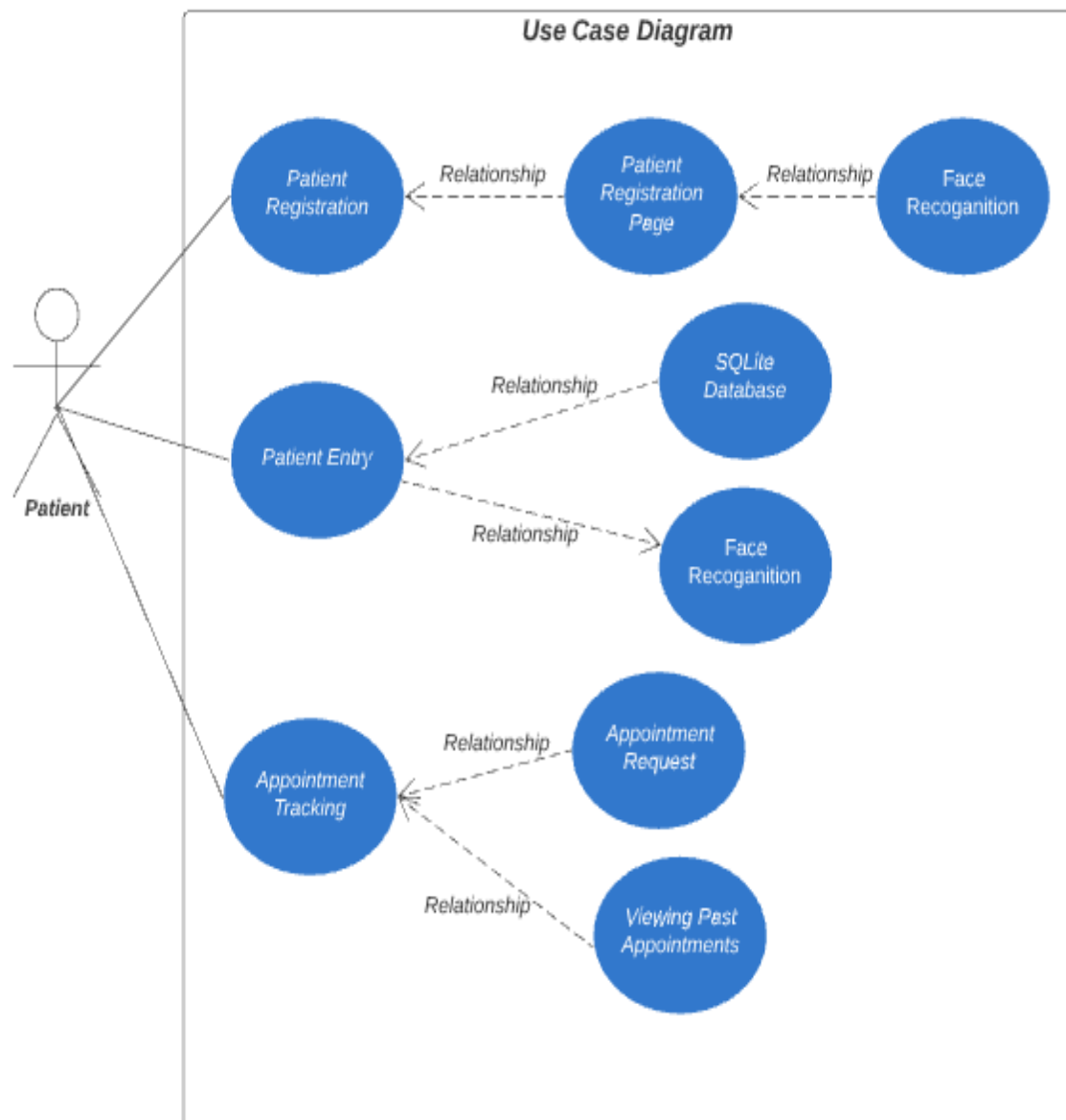


Fig 3.3 Use Case Diagram

3.4 FACE RECOGNITION DIAGRAM

3.4.1 PURPOSE

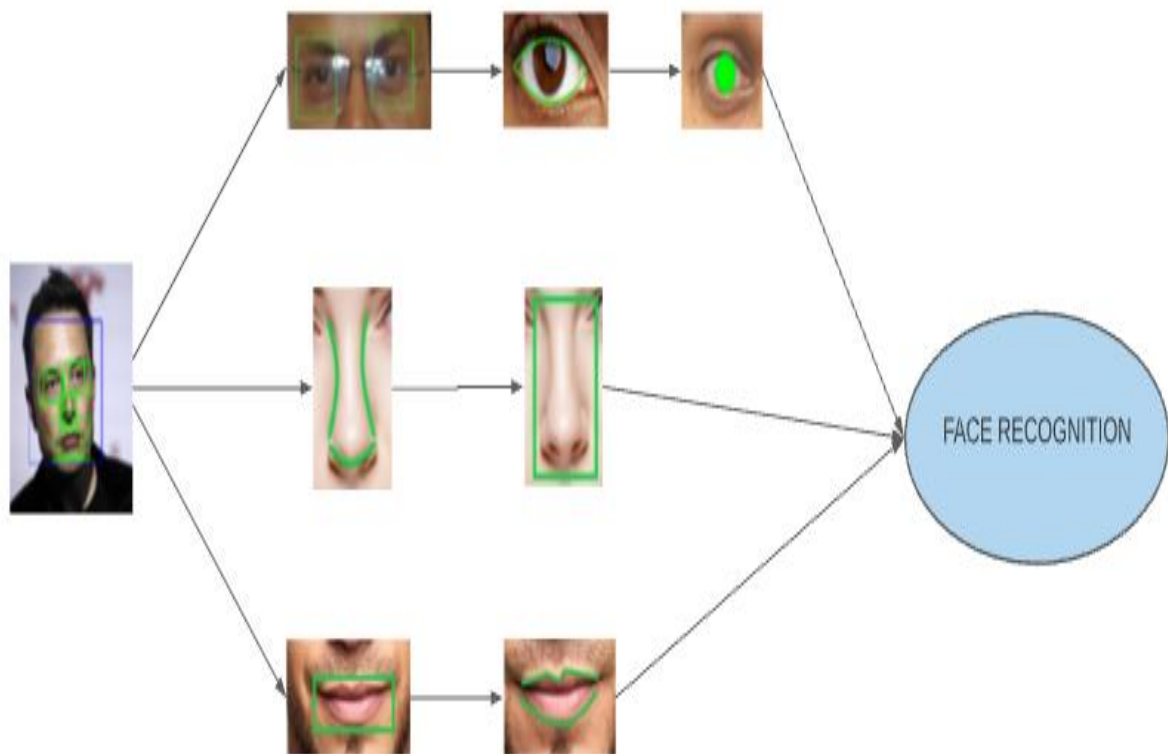


Fig 3.4 Face Recognition Diagram

3.5 DATA FLOW DIAGRAMS

3.5.1 PURPOSE

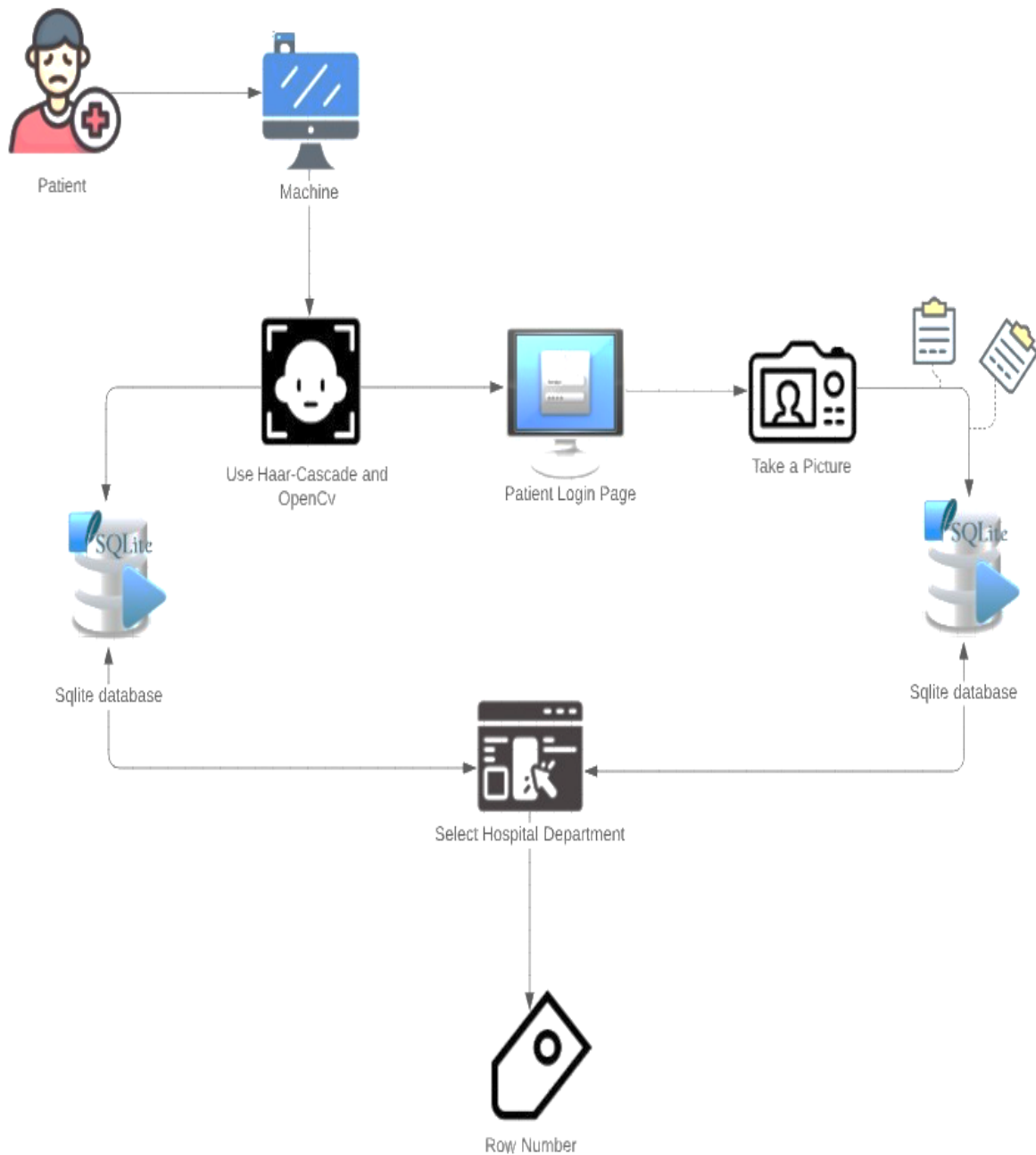


Fig 3.5 Data Flow Diagram

3.6 SQLITE DATABASE DIAGRAMS

3.6.1 PURPOSE

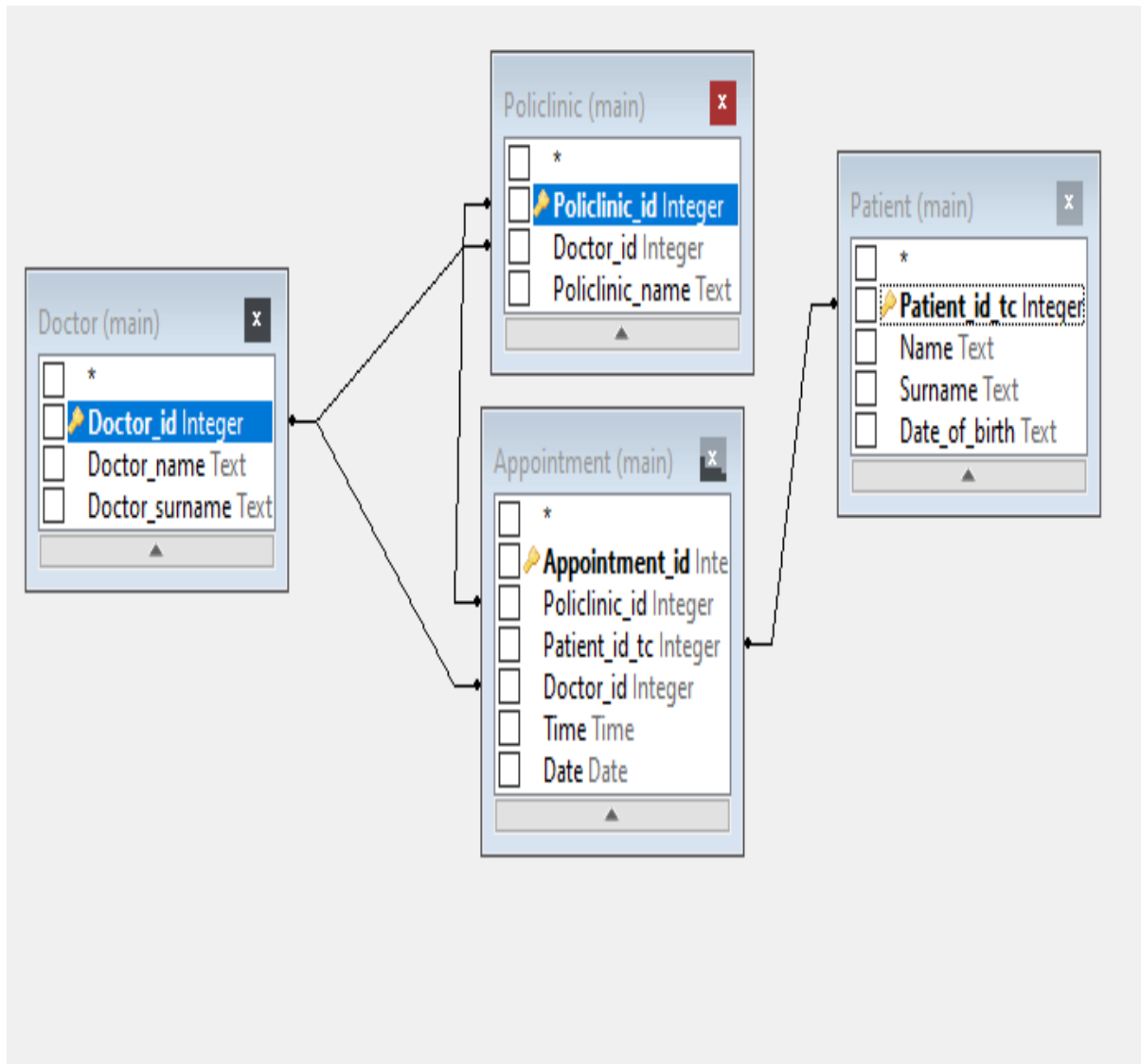


Fig 3.6 SQLite Database Diagram

3.6.1 TABLE**TABLE 1: DOCTOR**

FIELD	TYPE
Doctor_id	Int Primary Key
Doctor_name	Text
Doctor_surname	Text

Table 3.6.2.1 Doctor list

TABLE 2: APPOINTMENT

FIELD	TYPE
Appointment_id	Int Primary Key
Policlinic_id	Int
Patient_id_tc	Char (11)
Doctor_id	Int
Time	Time
Date	Date

Table 3.6.2.2 Appointment list

TABLE 3: POLICLINIC

FIELD	TYPE
Policlinic_id	Int
Doctor_id	Int
Policlinic_name	Text

Table 3.6.2.3 Policlinic list

TABLE 4 : PATIENT

FIELD	TYPE
Patient_id_tc	Char(11)
Name	Text
Surname	Text
Date_of_birth	Text

Table 3.6.2.4 Patient list

Component Description

4. COMPONENT DESCRIPTION

After installing the windows-based machine that we will run first in our project in our hospital, the necessary applications should be downloaded and run. The machine we are talking about is in the pictures below:



We do not need to have an admin interface because patients who come to the hospital can register themselves in the project we do. The machine we will use has a camera feature so that people can record their faces from the camera.

We only need a register interface. We create the interface we need thanks to the "Pyqt5" library. The reason it has to have a register interface is that the face of the incoming patient is actually taken from the camera during face recognition and the face is verified by our "Haar-Cascade" classification, and the face that is confirmed later is trying to find out who it belongs to by "OpenCv".

If the patient's face is detected later, the interface with the hospital departments will be displayed directly and the hospital department where the patient wants to go will be selected. When this selection is made, the sequence number to wait will be given thanks to our "sqlite3" data storage base.

If the patient looks at the camera and his face is not recognized, he will enter with the help of the voice commands of the TC, which stands for ID. In case the system fails or is confused with the face of another patient, we will examine whether there is any patient

registered with TC thanks to "sqlite3", which we use as storage operation with the TC entered in the error given, and as a result, we will have two options.

If the patient looks at the camera and his face is not recognized, he will enter with the help of the voice commands of the TC, which stands for ID. In case the system fails or is confused with the face of another patient, we will examine whether there is any patient registered with TC thanks to "sqlite3", which we use as storage operation with the TC entered in the error given, and as a result, we will have two options.

In case of presence in our "sqlite3" data storage base with the TC that the patient enters, verification is provided and the department interface that the patient wants to go to will appear and it will be provided to learn the sequence number by choosing from this interface.

Another option in our "sqlite3" data storage base, if the patient does not have a TC, the user registration interface will be opened. In this current section, the patient's name, the patient's surname, and the patient's TC will be requested to be entered through voice commands. All of the information about this patient entered is recorded in our "sqlite3" data storage base. After the registration process is completed, photos will be taken for face recognition by the camera and these photos will be used by "OpenCv" for later comparison, so it is recognized in case of coming back to the hospital and making an appointment. After the registration process is completed, the interface with the departments in the hospital will open and the patient will choose the department he wants to make an appointment with. After this selection, by going to the section list selected in "sqlite3", the sequence number will be given by looking at how many people there are.

The stages of our project are as above and the reason why we do it is that we try to minimize the contact in hospitals, which is one of the places where the covid-19 epidemic currently exists in the world, and we try to reduce the risk level that people will not be afraid to go.

4.1 TYPE

The face recognition system also has a lot of usage areas today. In particular, a face recognition system is used to enable users to enter the computer using simple camera systems on all newly released computers. In the face scanning system, just like in iris

scanning, it is based on taking certain reference points of the face, keeping them and then comparing them.

Face recognition systems also give much better results than before with the development of technology. Facial recognition as a result of storing only certain ratios and reference points by writing a small algorithm system rather than recording the entire face; It has also been used in places where high security is required such as airports and police stations.

The speech to text system is the process of converting an acoustic signal which is captured using a microphone to a set of words. The recorded data can be used for document preparation.

The speech recognition system is based on the past, just like the face recognition system. In the old days, this system used to convert numbers. However, studies have also been made with the text for a long time. Convert speech to text is software that allows the user to control computer functions and dictate text by voice. In particular speech-to-text technology has become a part of our lives in the last decade. Tools like Siri, Hey Google, and Alexa mean most people use speech recognition almost every day.

4.2 PURPOSE

Face recognition algorithms can be divided into two in terms of general structure. The first is the face recognition technique made on pictures and the other is the face recognition technique over a moving image. These two technologies are used in dozens of places today. These are banks and our own project, for example. As a security in the bank, pictures and signatures are taken while credit card is taken. When the customer wants to make a big withdrawal, a comparison is made with face recognition to make sure that the card owner is the same person. Likewise, in our project, we will make our appointment at the hospital faster and in a hygienic environment.

Voice is the basic, common and efficient form of communication method for people to interact with each other. Today speech technologies are commonly available for a limited but interesting range of task. This technologies enable machines to respond correctly and reliably to human voices and provide useful and valuable services. As communicating with computer is faster using voice rather than using keyboard, so people will prefer such system. One of our aims to make voice commands in our project is speed. Therefore we will use this algorithm.

4.3 FUNCTION

The operation steps in face recognition algorithms are as follows;
image input from an optical source such as web cam or traffic cameras,
preparing the picture for taking only the section rather than taking the whole face,
taking the face section and comparing it with the face section in the database,
repeating the steps from the picture entry in case of incompatibility
In case the face section matches the record in the database or if a new record is taken,
the biometric features of the face can be determined and registered directly in the database.

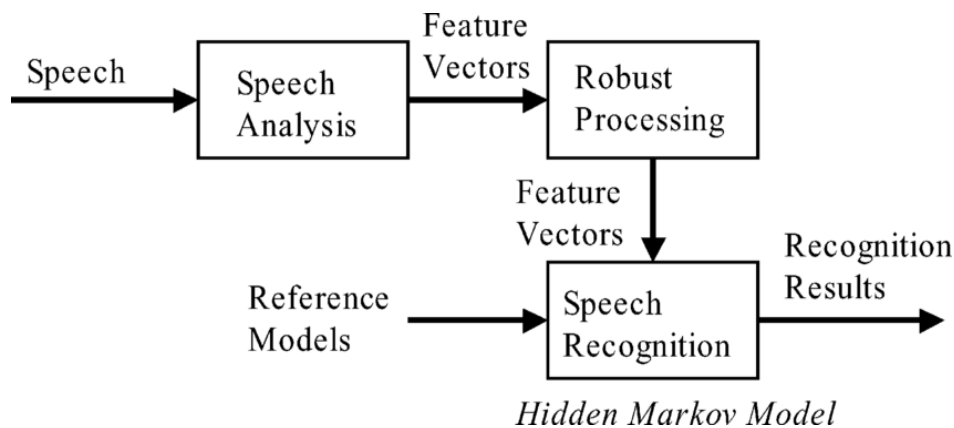
Speech to text algorithm consists of two components; The first component is for processing the acoustic signal captured by a microphone and the second component is interpreting the processed signal, then mapping the signal to words. The model for each letter will be created using the Hidden Markov Model (HMM). Feature extraction will be done using Mel Frequency Cepstral Coefficients (MFCC).

There are two alternatives to run this algorithm. First, you can dictate the project we have done to him and have him write the text while you speak. Second, you can load a recording made elsewhere into the app and let it convert to text.

However, we will ensure that we put the text in writing while speaking in a project. Because our project will be in a hospital environment.

The operation steps in speech to text algorithms are as follows;

- Command to start recording will be given.
- Words that will come out of the conversation momentarily after the conversation will be retained.
- These words will be analyzed letter by letter.
- The analyzed letters will be shared on the screen as text.

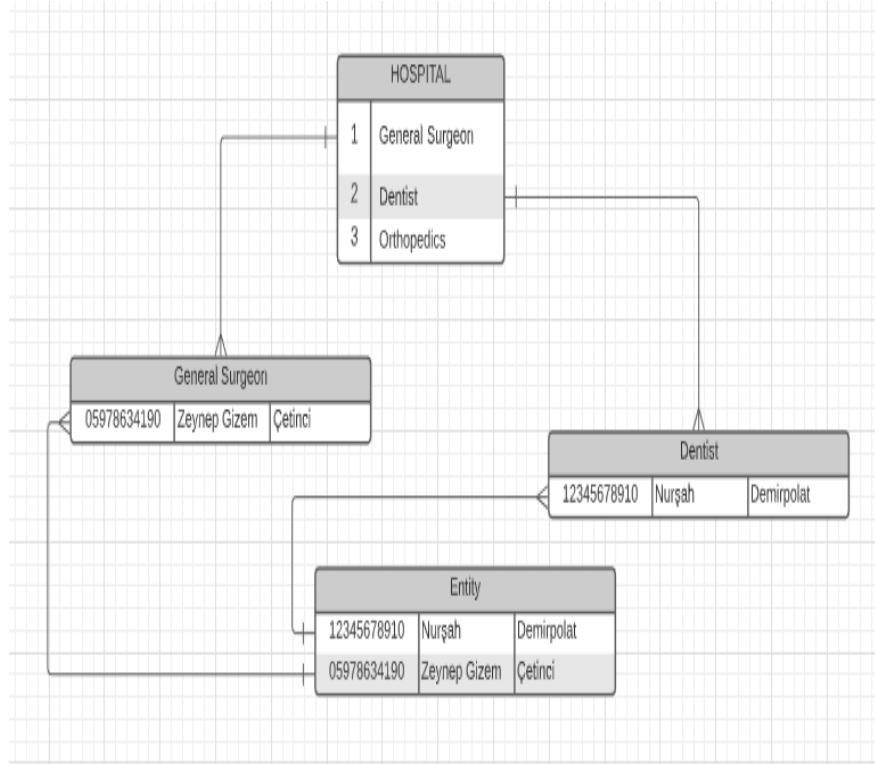


4.4 SUBORDINATES

We decided to use "Sqlite3" as a database in our project. The database we will use keeps the information of the patients who will use the application and stores the information of the hospitals that are available separately.

The patient's information includes the patient's TR ID number, patient name, patient surname, so we can access the patient's name and surname thanks to the TR ID number we will use as ID.

As the hospital information, we will keep the department names of the hospital and the people who come to each department. For example, if we consider the general surgeon as the hospital department, we will have a table called the general surgeon and this is the sub-table of the table called the general surgeon hospital, the Turkish ID number from our table in the name of the patients who will accommodate the general surgeon. Thanks to the system, a warning will appear on the screen in case of failure of the system or making an appointment again in similar minutes, which will reduce our appointment errors and this will then be given to people. Below is a small sample table we made.

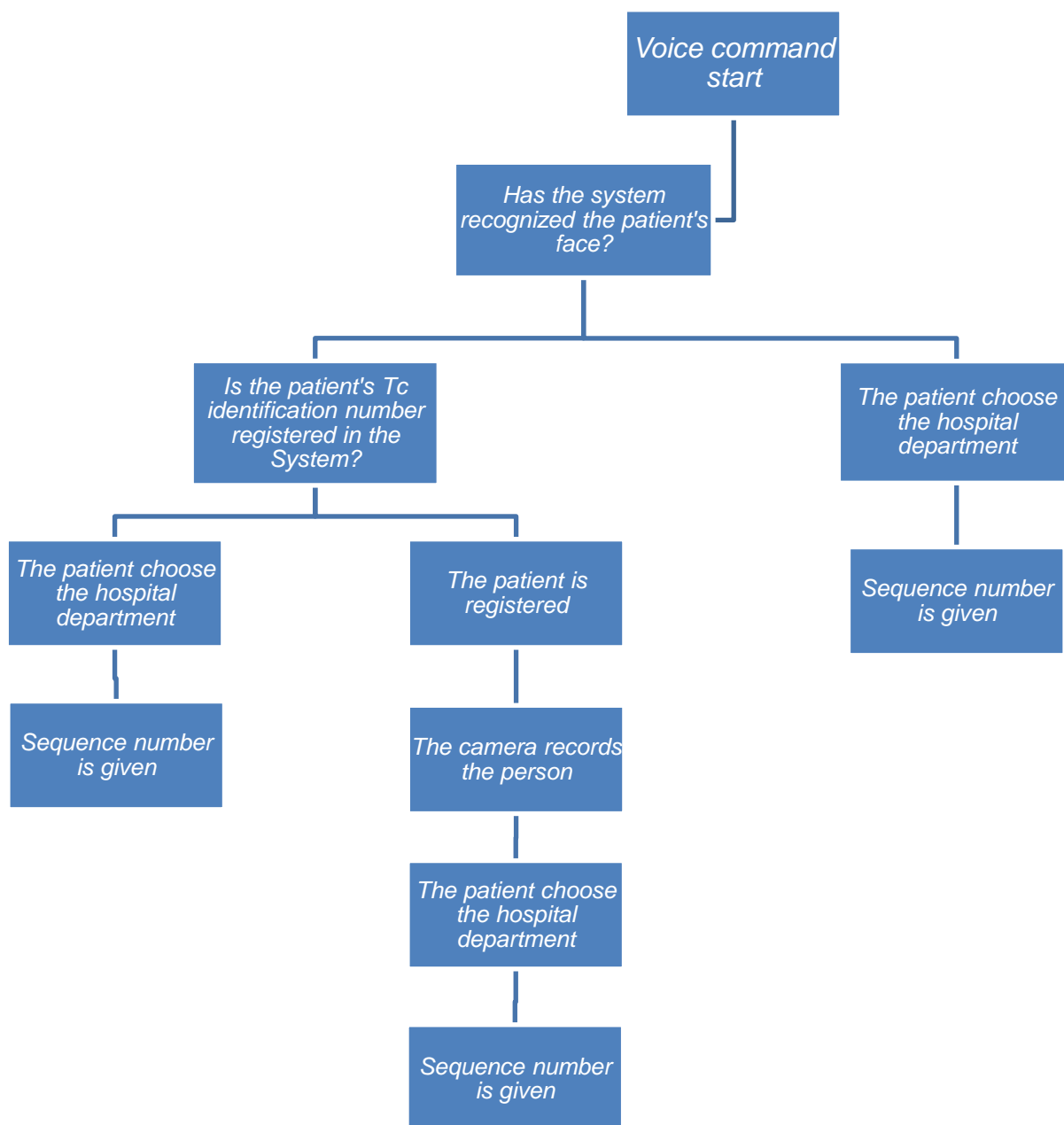


4.5 DEPENDENCIES

It is necessary to say the specified command in order to make the first registration in the project we will do. Later, the application started with this command will open the camera, find the face object with the Haar-Cascade classification and try to determine who it belongs to by using the "OpenCv" library. The determined name will be scanned in the database of "Sqlite3" and it will check whether it exists in the patients table. The determined name will be scanned in the database of "Sqlite3" and it will check whether it exists in the patients table. At this stage, two possibilities arise.

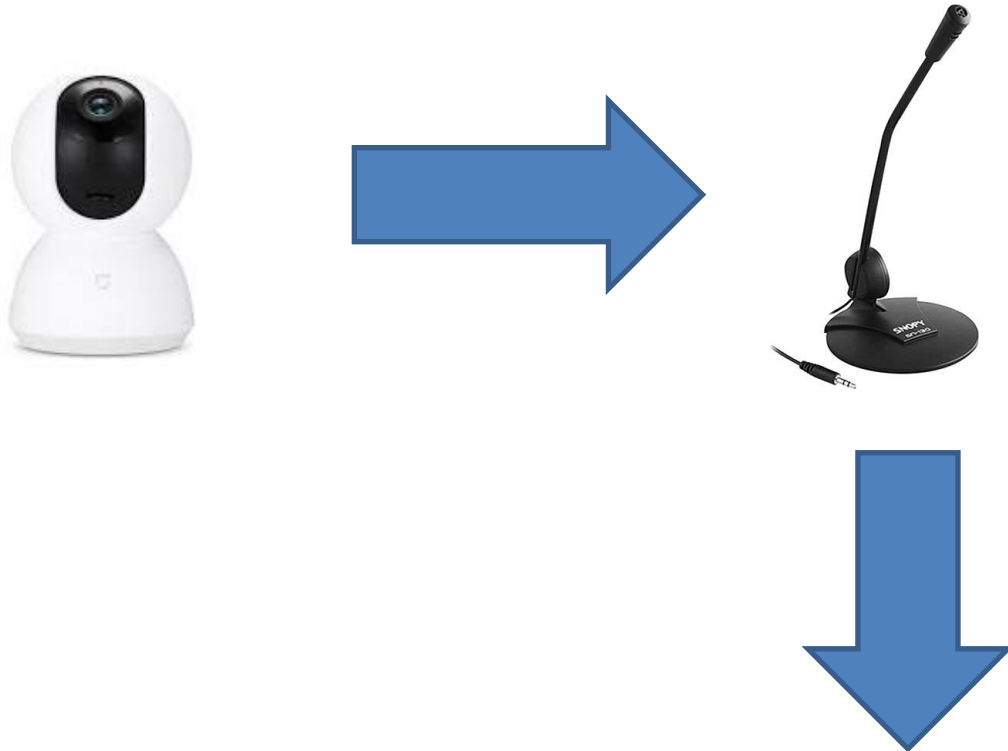
- As a result of finding the existing face on my system, the only thing that should be done by the patient is to select the hospital department where he / she wants to make an appointment, while the patient makes this stage, the system puts the selected appointment on the table of the department where the patient wants to go and returns a sequence number to the patient according to the order in the table.
- As a result of not finding the existing face in the system, an interface with the information that the patient needs to enter will appear. Pictures are taken to be added to the "OpenCv" library, which will be useful in face matching. After the registration to the system is completed, the hospital department selection screen will appear before the patient, while the patient makes this stage, the system puts the selected appointment on the table of the department the patient wants to go to and returns a sequence number to the patient according to the order in the table.

A short summary diagram is included in the table below. As can be seen, each process follows each other. For example, the Haar-Cascade face matching process cannot be performed with OpenCv without finding the face, or the patient registration page is opened directly and registration is not performed. In this ranking, we have been able to overcome the project we think will not work best and in most system crashes with minimal damage.



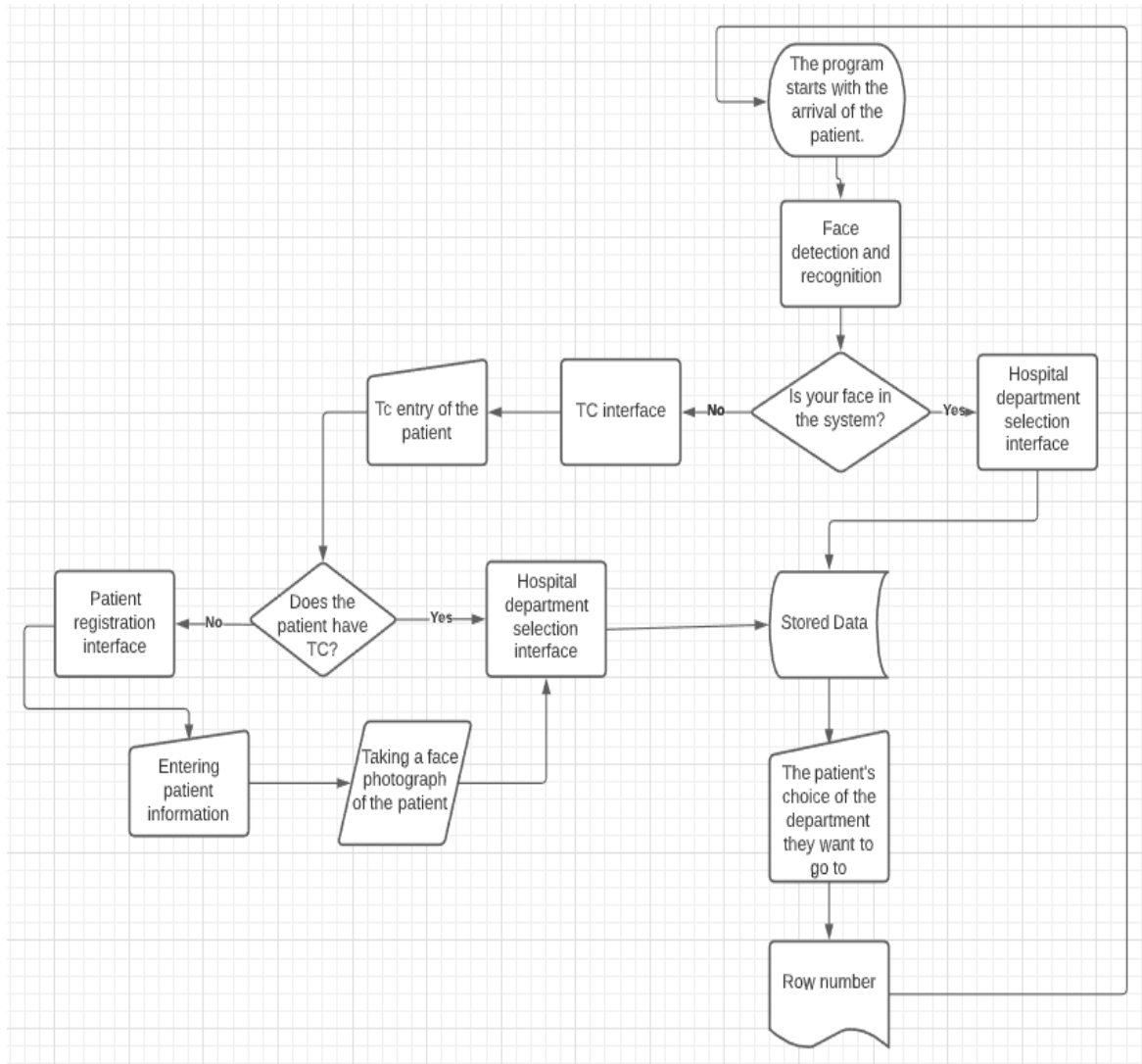
4.6 RESOURCES

A Windows-based touch screen computer and camera in a stand-alone stand. We will use the camera for face recognition. We will also use a microphone to translate the voice command into text. As seen below, we aim to operate a machine in a way that takes up less space and adds a stylish visual.



4.7 PROCESSING

In the photo below is a diagram of how our project works. We tried to show the working method of our diagram project we made in detail and clearly.



SYSTEM IMPLEMENTATION

5. SYSTEM IMPLEMENTATION

5.1 LANGUAGE

Python development started in 1990 by Guido van Rossum in Amsterdam. Contrary to popular belief, it was named not from a snake, but from the Monty Python's Flying Circus, a six-person British comedy group named Monty Python, loved by Guido van Rossum.

- Python reached its version 1.0 in January 1994.
- Version 2.0 released on October 16, 2000.
- 3.x series has been launched on December 3, 2008; however, 3.x series is not compatible with 2.x series.

5.1.1 FEATURES

Python provides many useful features which make it popular and valuable from the other programming languages. It supports object-oriented programming, procedural programming approaches and provides dynamic memory allocation. We have listed below a few essential features:

- **Easy to Learn and Use:** Python is easy to learn as compared to other programming languages. Its syntax is straightforward and much the same as the English language. There is no use of the semicolon or curly-bracket, the indentation defines the code block. It is the recommended programming language for beginners.
- **Expressive Language:** Python can perform complex tasks using a few lines of code. A simple example, the hello world program you simply type `print("Hello World")`. It will take only one line to execute, while Java or C takes multiple lines.
- **Interpreted Language:** Python is an interpreted language; it means the Python program is executed one line at a time. The advantage of being interpreted language, it makes debugging easy and portable.
- **Cross-platform Language:** Python can run equally on different platforms such as Windows, Linux, UNIX, and Macintosh, etc. So, we can say that Python is a portable language. It enables programmers to develop the software for several competing platforms by writing a program only once.

- **Free and Open Source:** Python is freely available for everyone. It is freely available on its official website www.python.org. It has a large community across the world that is dedicatedly working towards make new python modules and functions. Anyone can contribute to the Python community. The open-source means, "Anyone can download its source code without paying any penny."
- **Object-Oriented Language:** Python supports object-oriented language and concepts of classes and objects come into existence. It supports inheritance, polymorphism, and encapsulation, etc. The object-oriented procedure helps to programmer to write reusable code and develop applications in less code.
- **Extensible :** It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our Python code. It converts the program into byte code, and any platform can use that byte code.
- **Large Standard Library :** It provides a vast range of libraries for the various fields such as machine learning, web developer, and also for the scripting. There are various machine learning libraries, such as Tensor flow, Pandas, Numpy, Keras, and Pytorch, etc. Django, flask, pyramids are the popular framework for Python web development.
- **GUI Programming Support :** Graphical User Interface is used for the developing Desktop application. PyQt5, Tkinter, Kivy are the libraries which are used for developing the web application.
- **Integrated :** It can be easily integrated with languages like C, C++, and JAVA, etc. Python runs code line by line like C,C++ Java. It makes easy to debug the code.
- **Embeddable :** The code of the other programming language can use in the Python source code. We can use Python source code in another programming language as well. It can embed other language into our code.
- **Dynamic Memory Allocation :** In Python, we don't need to specify the data-type of the variable. When we assign some value to the variable, it automatically allocates the memory to the variable at run time. Suppose we are assigned integer value 15 to x, then we don't need to write `int x = 15`. Just write `x = 15`.

5.1.2 ARCHITECTURE OF PYTHON

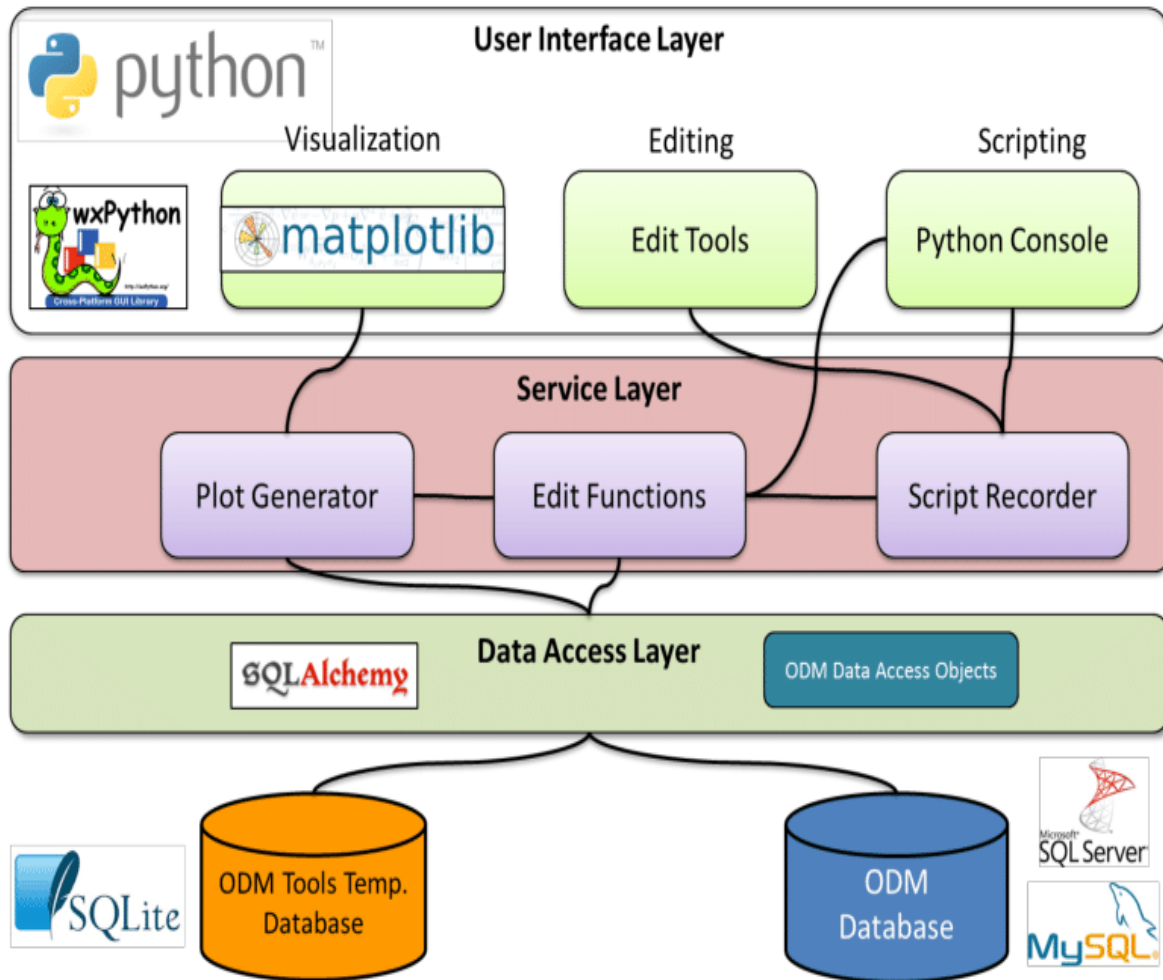


Fig 4.1.2 Architecture Of Python

5.1.3 LIBRARIES

For our 'Hosface' product that we have made in Python, we used 3 libraries of python. These libraries are as follows:

- **SQLite**
- **OpenCv**
- **Haar Cascade**
- **Speech Recognition**

5.1.3.1 SQLite

Sqlite is a database library. It is very easy to use and install. It doesn't require setup to use it. There is no setup procedure. There is no need to start, stop or configure any server processes to use it. An administrator does not need to create a new database instance or grant users access to use sqlite. It does not require any action to recover after a system crash or power failure. It just works. It is an application that makes life easier.

Sqlite does not require any main server operation. It is not necessary to establish intermediate process communication to access the database, send requests to the server and get results. This program reads and writes database files directly on disk. The main advantage of this situation is; There is no need to do a separate server operation to set up, configure, start, manage and troubleshoot. However, this situation also has a disadvantage. Databases using servers are generally more secure.

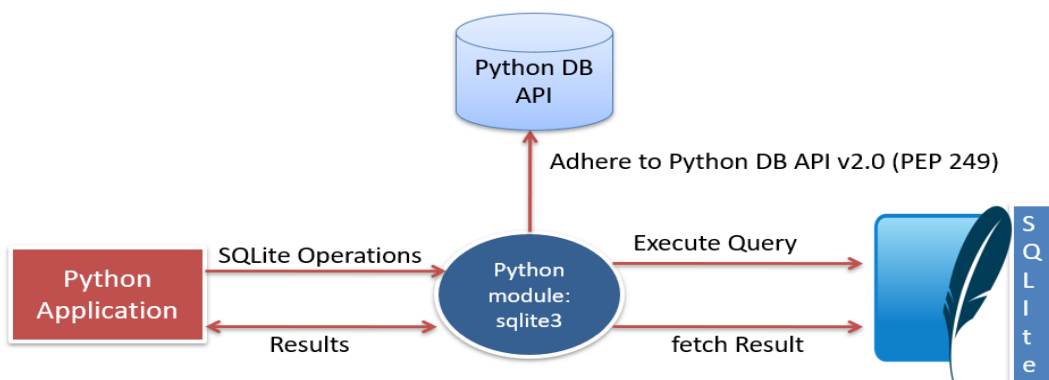
If sqlite can read the disk file, it can also read anything contained in the database. If the disk file and directory are writable, you can use this program to change anything in the database. You can copy database files to a USB or share them by e-mail. The files are extremely easy to access. Of course, while performing all these operations easily and quickly, it is extremely important to ensure the security of the data.

Sqlite is a cross-platform file format. In this way, a database file written on a device can be copied to a different device with a different architecture and used on this device. Considering that other database engines require you to dump and reload your database as you move from one platform to another, it can be better understood how much Sqlite makes things easier.

In addition, the file formats are fixed and backward compatible.

Considering the size, a Sqlite library is less than 500 KiB. Unnecessary features can be disabled to save more space. The size of the library can be reduced up to 300 KiB. Most other databases are larger than this. Manifest writing removes many restrictions. Thus, the user can store any value of any data type in any column regardless of the column type. The language specification is suitable for manifest writing. Sqlite becomes more reliable and easier to use, especially when used with dynamically typed programming languages such as Tc and Python. Although this feature seemed like a bug at first, it was later proven in practice by software developers.

Sqlite source codes are readable. It is designed to be accessible and readable by standard programs. Also, source code for Sqlite is in the public domain. Anyone can access and read. No copyright is claimed on any part of the core source code. All users contributing to the software can use the source code. The code can be freely used by a wide audience. Language extensions also allow these codes to be easily accessed and read by anyone. All of these made Sqlite very popular with programmers in a short time. Security is a software that makes life easier as long as it is properly managed.



5.1.3.1.1 Sqlite Usage Areas

- Websites
- Data analysis
- Data transfer format
- Application file format
- Experimental SQL language extensions

5.1.3.1.2 How To Create A Database?

We use the sqlite module to create a database. To use this module, we first need to import the module.

```
import sqlite3
```

Let's look at how to create a database with the help of this module. For this, the sqlite3 module We will use a method called connect ().

We use this method as follows:

```
>>> vt = sqlite3.connect('hastane.db')
```

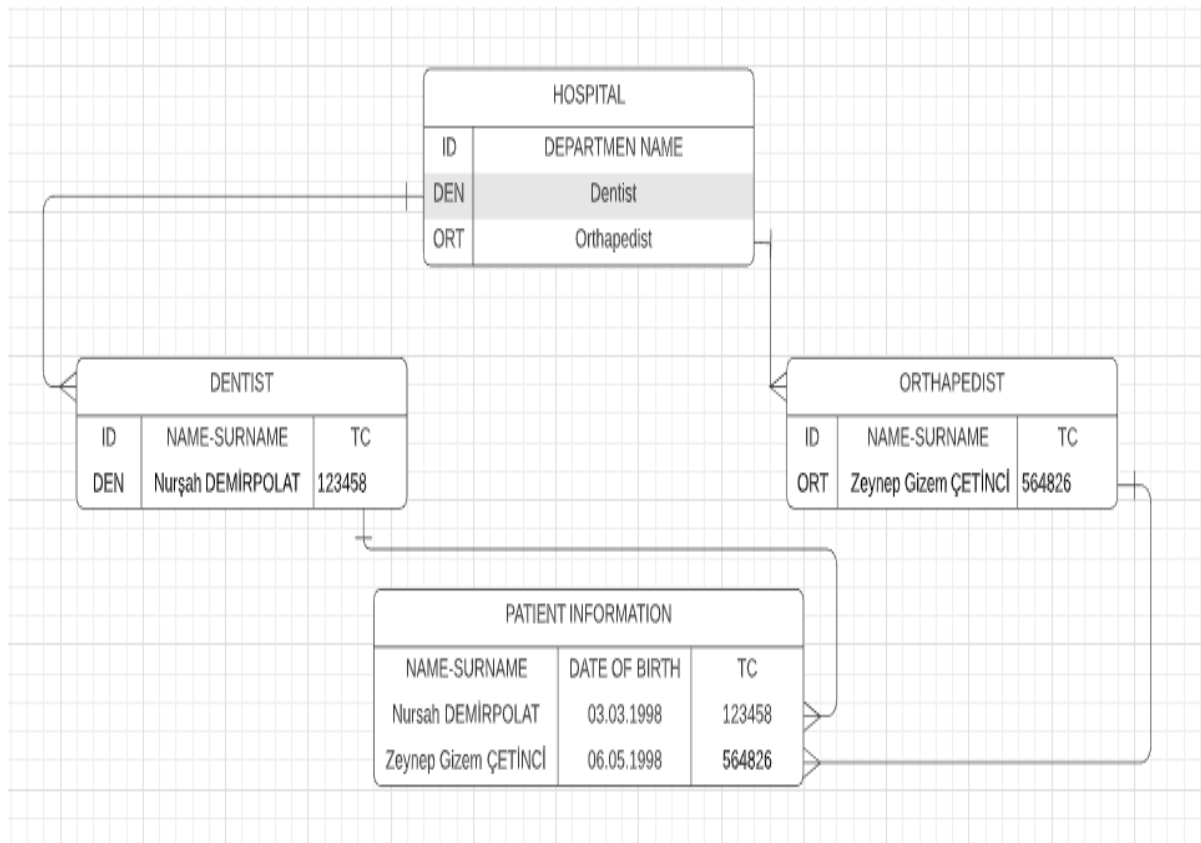
The as we give to the Connect () method is the name of the database we will use.

5.1.3.1.3 Why Do We Use Database?

The reason for using the database is that we will create a table that shows the departments in the hospital, the information of the patients and which patients go to a department. The reason we make the table with patients going to the department in the hospital is that a campaign will be made to the department that is preferred the most in 1 month. To be more detailed, we thought of creating 2 main tables in sqlite table;

- Patients
- Hospital Departments

Our patients table is variable and consists of adding the information of incoming patients. Hospital departments table is not variable but will be a table in which the hospital's departments are written. We will check whether there are patients in our patients table with the "TC" column. Each hospital department will have its own table, and we will keep a record of who came to which department from these tables, and we will arrange the incident of placing patients in the departments by asking our interface, so each department table will be variable. Thanks to the main table and sub-tables, we will have access to all information about who, how much, when.



5.1.3.2 OpenCv

Along with being a BSD-licensed free product, OpenCV (Open Source Computer Vision) appears as an image-vision based open-code library used in the production of real-time applications. It makes it easy for businesses to use codes by making changes on existing algorithms. Currently, it is seen that the software program is used very actively by very important companies such as Google, Yahoo, Microsoft, IBM, Intel, Sony, Honda.

If a retrospective assessment is required, it can be easily said that it is also used by companies, research groups and government bodies. OpenCV, which complies with the definition of "Computer Vision" library, which has an opensource code, has a download number of over 14 million by 2019.

It is also among the data that it is actively used by around 50,000 users. While it was a software program developed in Russia by the Intel company in 1999, it was introduced for the development of applications with CPU. Having a common infrastructure specifically for computer vision applications aimed its functionality.

OpenCV, which is a very comprehensive program, gives much more detailed results with its features. Thanks to "Computer Vision", it can now enable devices not only to record cameras, but also to read the license plates of vehicles autonomously

With its advanced technological structure, it can detect faces and objects. Then the identification process begins. While the classification of human actions in videos is concerned, it is beneficial to watch moving objects and to be able to sort them into 3-dimensional models. It is also useful in producing 3D point clouds. The OpenCV program, which also provides high resolution combining of images, is currently used in many different areas.

First, the coding process was done with C, OpenCV, and after the updates in the following years, it uses the C ++ language. It has a much more modern structure as well as Python, Matlab / Octave and Java interfaces. In terms of usage, it supports Linux, Windows, Android, Mac OS. It includes more than 500 algorithms and 10 times more functions that take part in the formation of these algorithms and continue to support them. It also appears to make use of MMX and SSE instructions.

OpenCV is a project that was first developed in Intel's laboratories in Russia in 1999. It was originally created to develop applications with CPUs, which were part of a number of projects including real-time ray tracing and 3D screen walls, providing a common infrastructure for computer vision applications and accelerating the use of machine perception in commercial products.

One of the biggest advantages of OpenCV is that it is an open source library. Because of this feature, it can be improved by making changes on the algorithms. It is also distributed under the BSD license. OpenCV, a BSD licensed product, makes it easy for businesses to use and modify code.

Except for well-established companies such as Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota, OpenCV is widely used; There are many new companies like Applied Minds, VideoSurf, and Zeitera.

OpenCV can be used in areas such as detecting and identifying faces and objects, classifying human actions in videos, tracking camera movements and moving objects, sorting objects into 3-dimensional models, generating 3D point clouds from stereo cameras, combining images in high resolution. This is the most important advantage of this algorithm for our project.

- **Core:** It contains the basic functions of OpenCV and data structures such as matrix, point, size. It also includes methods that can be used to draw on images and the necessary components for XML operations.
- **HighGui:** It contains methods that may be required for viewing images, managing windows and graphical user interfaces. Before 3.0, it had methods that perform image file reading and writing operations over the file system.
- **Imgproc:** A package that includes almost all functions such as filtering operators, edge detection, object detection, color space management, color management and thresholding. Although some functions have changed in versions 3 and later, many functions are the same in version 2 and 3.
- **Imgcodecs:** It contains methods that perform image and video reading / writing operations over the file system.
- **Videoio:** It includes the methods required to access cameras and video devices and to capture and write images. Before OpenCV 3 release, many methods in this package were in the video package.

5.1.3.2.1 Using OpenCv

Since we will use Python via PyCharm in our project, we can install opencv like this:

```
pip install opencv-python
```

This use above allows us to add the OpenCv library to our project.

5.1.3.3 Haar Cascade

Computer vision is a field of study which aims at gaining a deep understanding from digital images or videos. Combined with AI and ML techniques, today many industries are investing in researches and solutions of computer vision. Think about the following example: many studies are being carried on to implement security cameras with object detection capabilities. Our aim is to capture the snapshot when the patient comes across the web camera and compare it with the data in the database. Object detection is a powerful tool. For this purpose, We will use OpenCV (Open Source Computer Vision Library), an open source computer vision and machine learning software library that is easy to import in Python. We will specifically use the Haar Cascade algorithm.

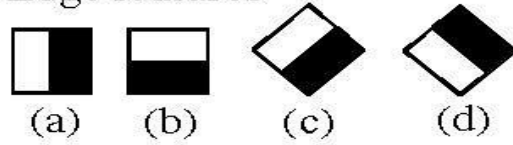
5.1.3.3.1 What Is The Haar Cascade?

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, “Rapid Object Detection using a Boosted Cascade of Simple Features” in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Fortunately, OpenCV offers pre-trained Haar cascade algorithms organized into categories (faces, eyes, etc.) depending on the images they've been trained on.

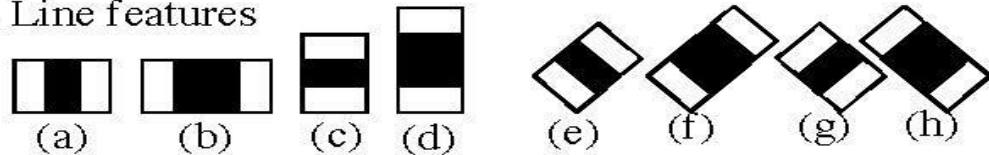
5.1.3.3.2 How Does Work Haar Cascade?

We will talk about how the Haar cascade algorithm works. The Haar cascade algorithm is to extract attributes from images using a type of 'filter' similar to the convolutional kernel concept. Initially, the algorithm needs a lot of positive images (face images) and negative images (faceless images) to train the classifier. Then we have to extract properties from it. For this, the Haar features shown in the image below are used. They are just like our convolutional core. Each property is a single value obtained by subtracting the sum of the pixels under the white rectangle from the sum of the pixels under the black rectangle.

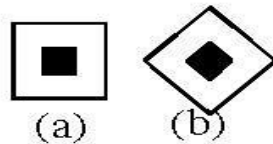
1. Edge features



2. Line features



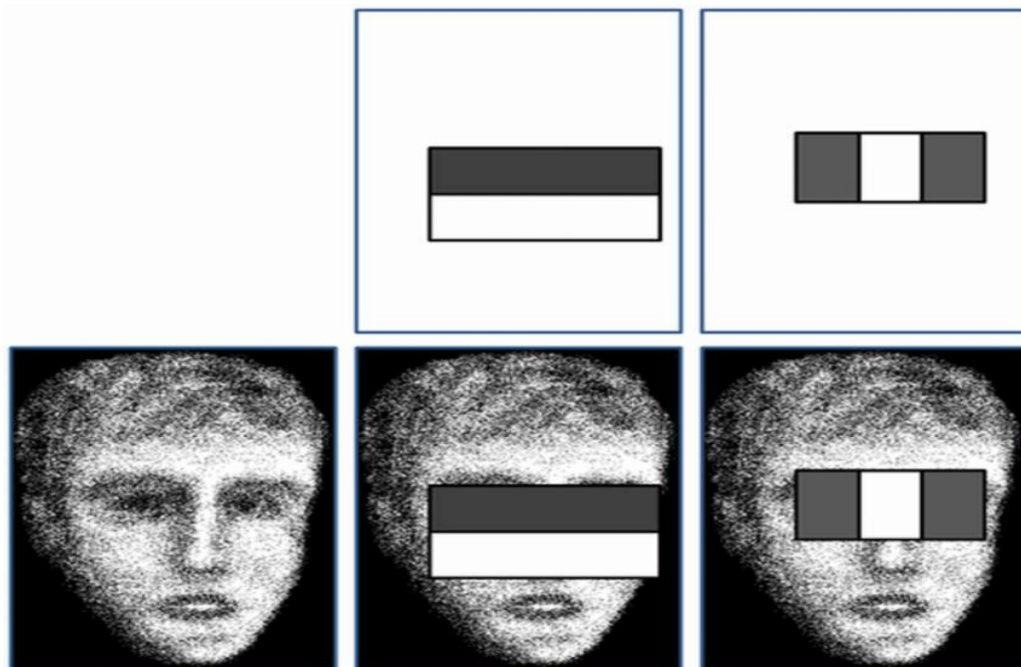
3. Center-surround features



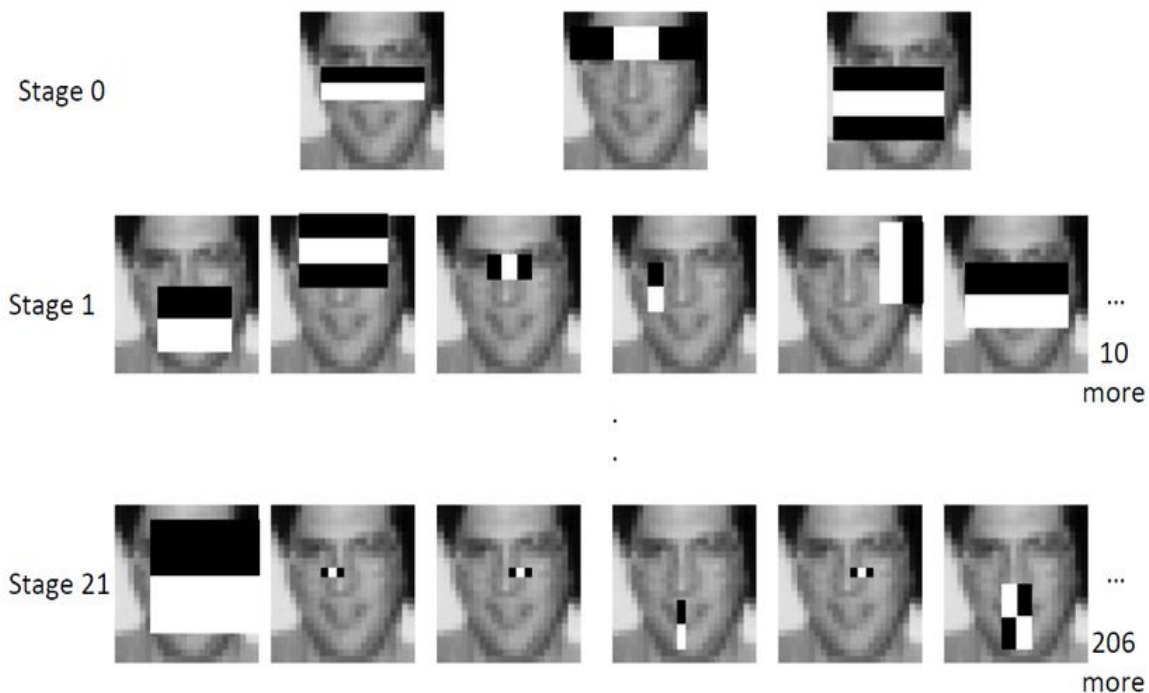
4. Special diagonal line feature used in [3,4,5]



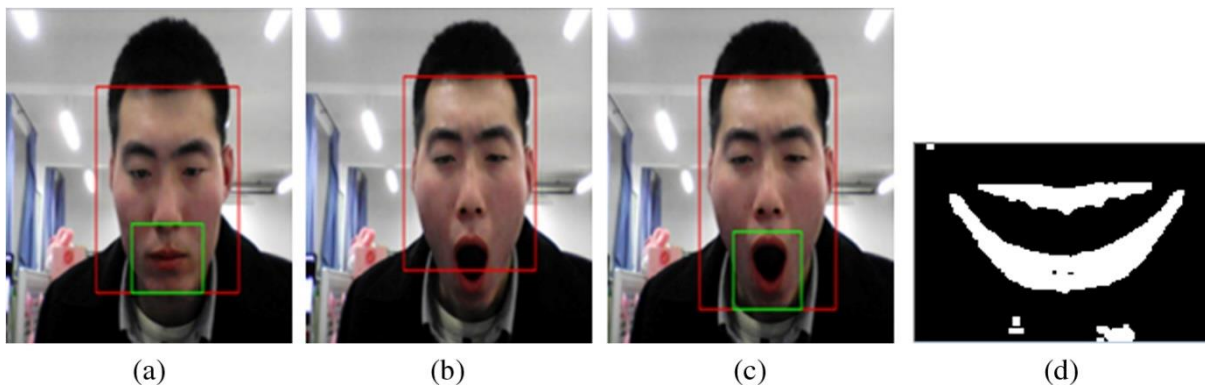
Ideally, a great value of a feature means it's relevant. Namely, if we take the Edge property (a) and apply it to the Black and White image below:



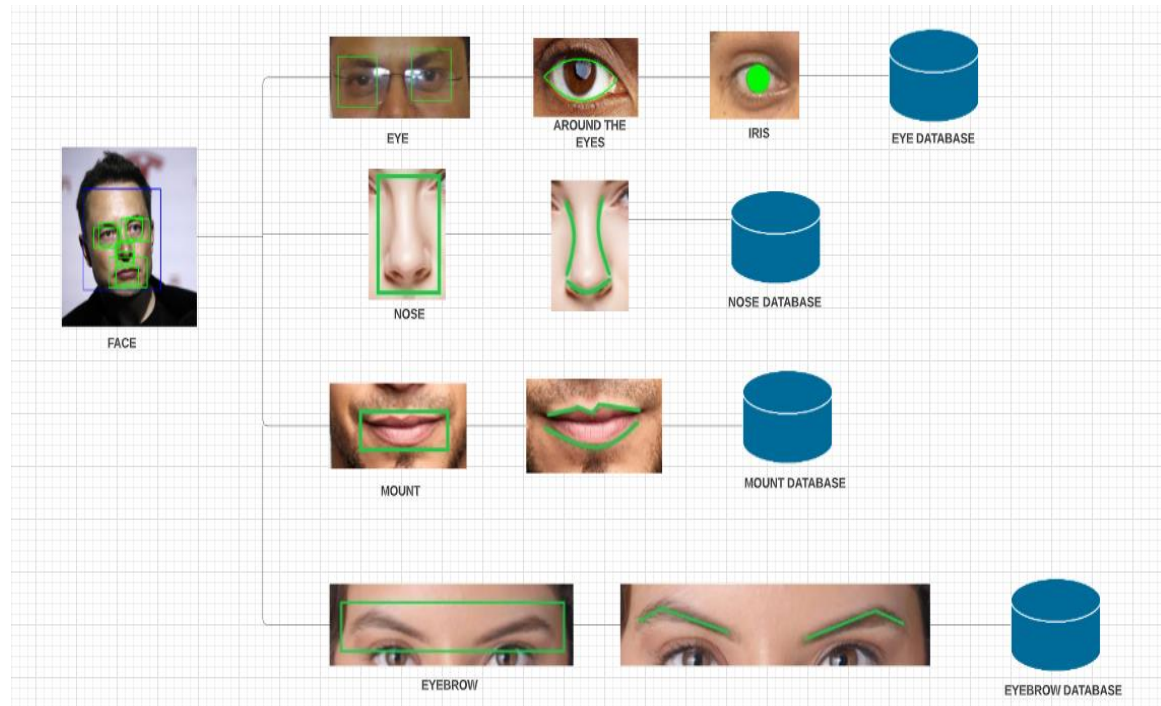
.We will obtain a significant value, hence the algorithm will return an edge feature with high probability. Of course, the real intensities of pixels is never equal to white or black, and we will often face a similar situation.



Let's take our mouth to give an example. Not everyone's mouth size is the same. Therefore, adjustments should be made to determine the mouth. Errors or incorrect classifications will be made for the image. First, we select the features with the minimum error rate. These are the features that classify the mouth and non-mouth images most accurately. Each image is given equal weight initially. After each classification, the weights of misclassified images are increased. New error rates are calculated.



The process is continued until the required accuracy or error rate is achieved or the required number of features are found. The last classifier is the weighted sum of these poor classifiers. It is called weak because it alone cannot classify the image, but together with others it creates a strong classifier.



Face recognition algorithm will be made according to the chart we have seen above. There will be data for each region for the face. It will proceed step by step to detect the eye. First, the general eye area is found. Then the second step is to detect the eye area. Finally, the pupil is removed. And eye analysis is done. In this way, eye detection is provided for the first recording person. In order to perform a face scan for the registrant, it is also provided to recognize the person's eye.

5.1.3.3 Expressing By Formula

In order to increase the efficiency of the algorithm, the integral image is first produced. The integral image is denoted by $ii(x, y)$. Here, (x, y) refers to the point where the image is

$$ii(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i(x', y'), \quad (1)$$

In the equation, $i(x, y)$ is the image given as input. The integral image contains the sum of the pixel values above and to the left of the (x, y) coordinate.

The decision-making mechanism of the Haar-Cascade classifier algorithm is given in equations 2 and 3.

$$C_m = \begin{cases} 1, & \sum_{i=0}^{J_m-1} F_{m,i} > \theta_m \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$F_{m,i} = \begin{cases} \alpha_{m,i}, & \text{if } f_{m,i} > t_{m,i} \\ \beta_{m,i}, & \text{otherwise} \end{cases} \quad (3)$$

Here, $f_{m,i}$ is the weighted sum of 2-dimensional integrals and also i is the decision threshold for the attribute extractor. $\alpha_{m,i}$ and $\beta_{m,i}$ are constant values associated with the attribute extractor. $\theta_{m,i}$ is the decision threshold for the classifier.

5.1.3.4 Speech Recognition

5.1.3.4.1 Type

In our project, the patient who comes to the hospital will record with a voice command, even if you have not been registered before. We will use the SpeechRecognition library to translate the voice command into text.

Speech Recognition covers the sampling of audio signals and making them meaningful by methods such as artificial neural networks and machine learning.

In fact, applications such as Apple Siri, Google Assistant, Amazon Alexa have this command on most of our phones. NLP (Natural Language Processing) algorithm is also used to make the SpeechRecognition command more meaningful.

5.1.3.4.2 Purpose

One of our purposes of using the SpeechRecognition library will be to fill in the patient information with voice command. Because the hospital is crowded, it can also analyze the voices of other speaking people while recording. For this reason, it may cause mistakes in understanding the words of the patient and also turn the voice into false texts. For this reason, we use the SpeechRecognition library to prevent these errors. Because with the SpeechRecognition library, we can limit the noise value so that it does not detect outside sounds. Generally, the noise value is given as 300.

5.1.3.4.3 Function

In order to convert the voice to text, we first download the SpeechRecognition library via python.

```
pip install SpeechRecognition
```

To be able to run it in Python code, we have to type

```
import speech_recognition as sr
```

Of course, this library alone is not very useful. That's why we should use APIs to support the library. These;

- 1 CMU Sphinx (works offline)
- 2 Google Speech Recognition
- 3 Google Cloud Speech API
- 4 Wit.ai
- 5 Microsoft Bing Voice Recognition
- 6 Houndify API
- 7 IBM Speech to Text
- 8 Snowboy Hotword Detection (works offline)

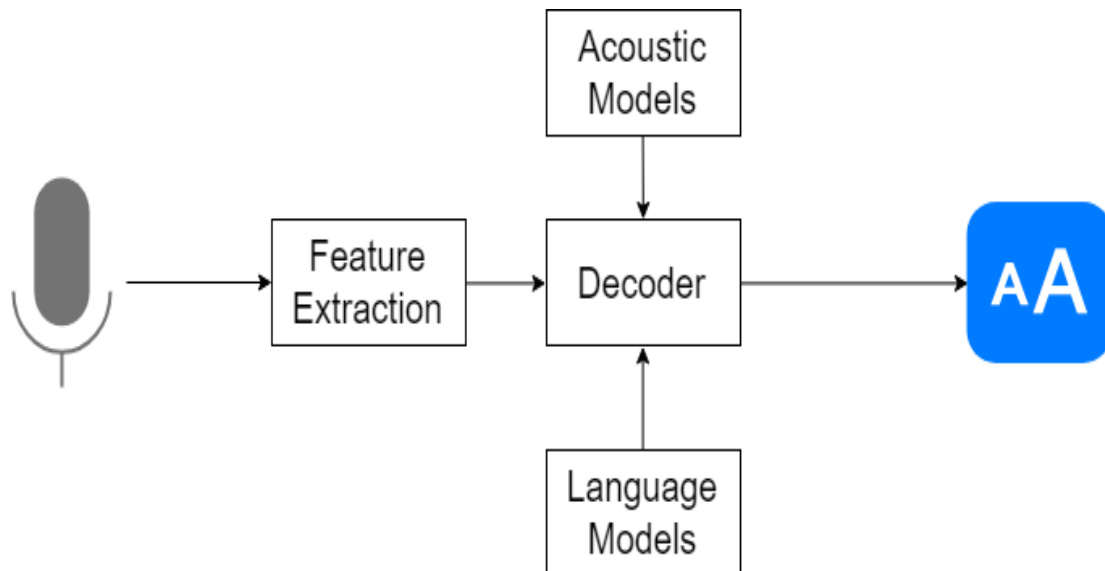
5.1.3.4.4 How To Use The Speech Library?

In order to record the sound from the microphone and transcribe it, we first need to include the library in our project and define a microphone and recognizer from the package.

- `import speech_recognition as sr`
- `mic = sr.Microphone ()`
- `r = sr.Recognizer ()`

5.1.3.4.5 How It Is Work?

The picture below shows the working principle of Speech Recognition very clearly, now let's understand the concept behind it.



It is based on acoustic and language modeling algorithm. Now the question is what is acoustic and language modeling?

- Acoustic modeling represents the relationship between linguistic speech units and audio signals.
- Language modeling matches sounds with word strings to help distinguish words that sound similar.

Any speech recognition program is evaluated using two factors:

- Accuracy (percentage error when converting spoken words into digital data).
- .Speed (the degree to which the program can keep up with a human speaker).



SCREENSHOTS

6. SCREENSHOTS

6.1 SCREENSHOTS FOR UNKNOWN REGISTER

6.1.1 HOSFACE WELCOME PAGE

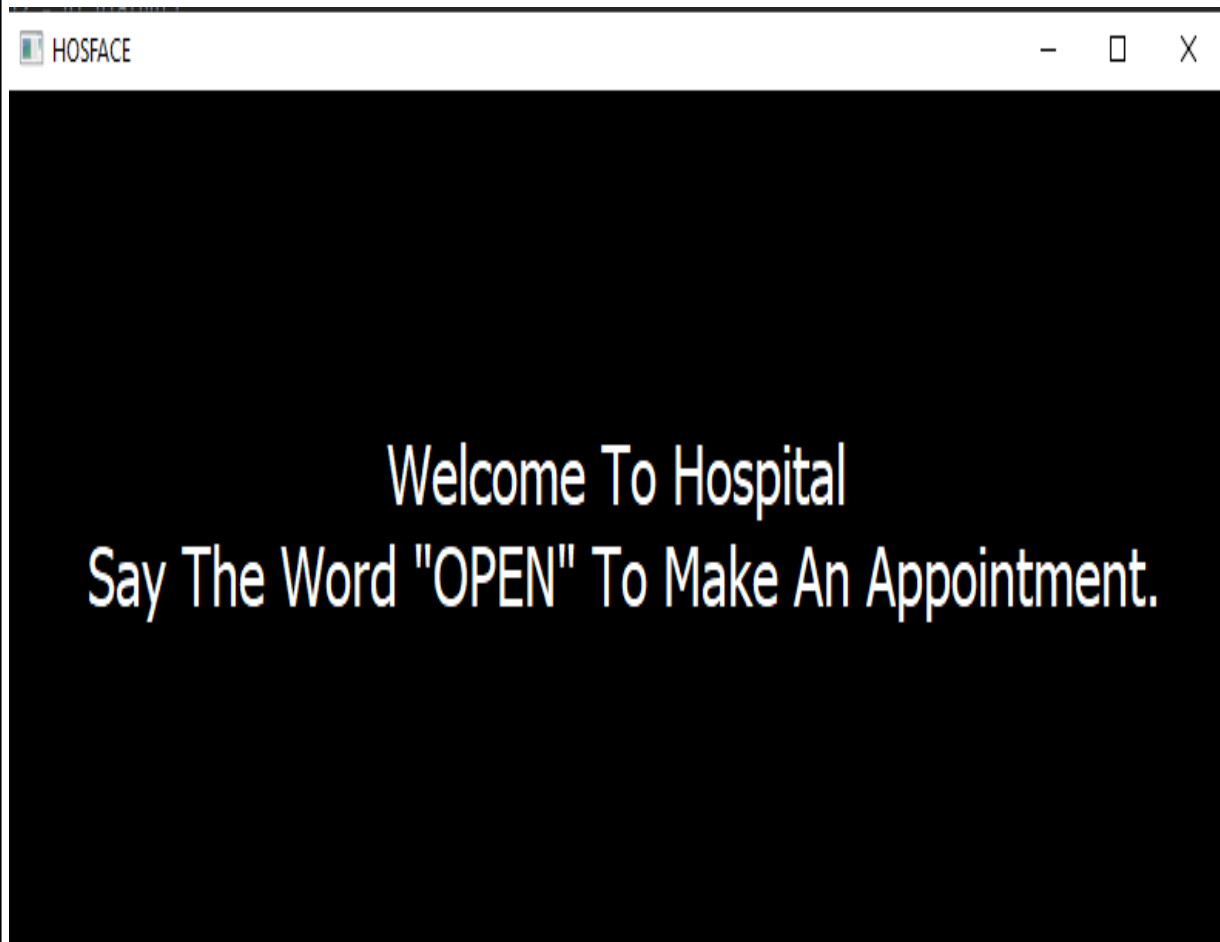


Fig 5.1 HOSFACE Welcome Page

This page will appear when you first start our project. On this page, there is a post that tells you the action you need to take to start the project.

6.1.2 HOSFACE WEBCAM PAGE – UNKNOWN REGISTER

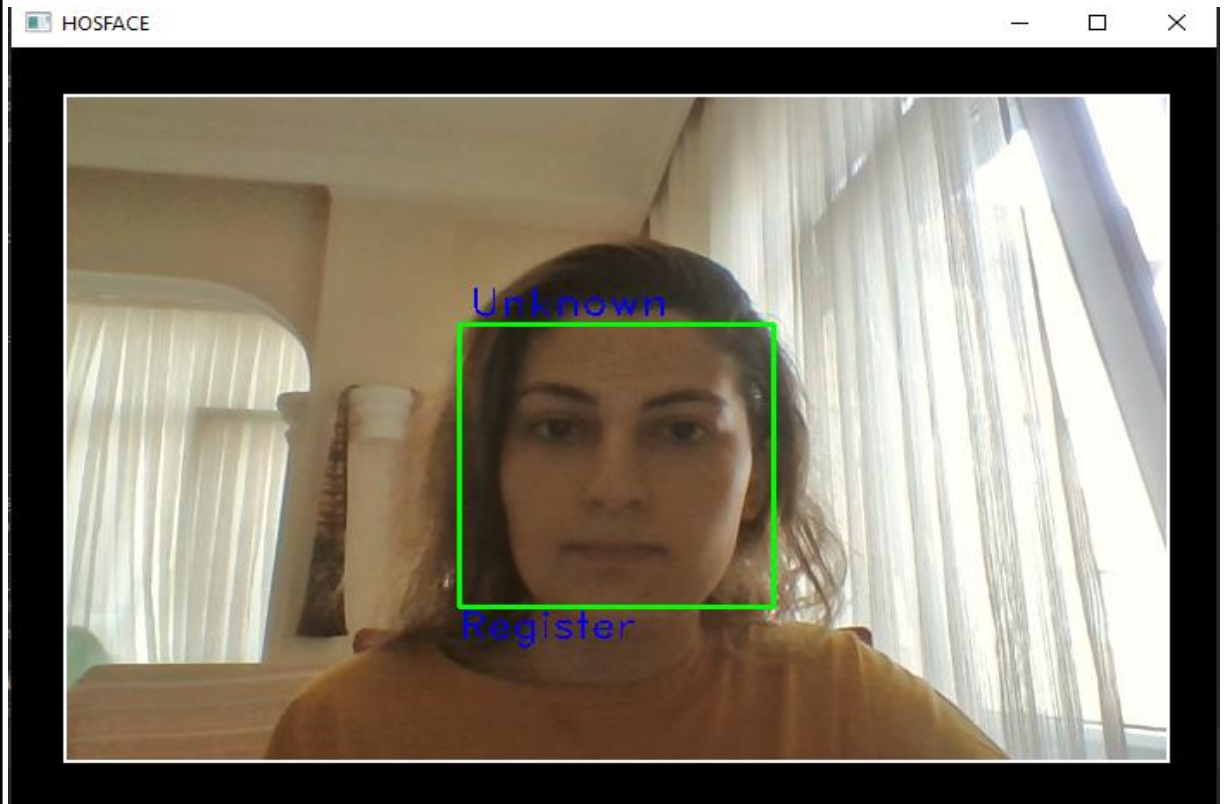


Fig 5.2 HOSFACE Webcam Page – Unknown Register

For the unknown register, it will find your face live on the webcam page and frame it, then the unknown user will appear on the screen in the green area framing your face, as the unknown register is not registered in the system. The system here works with similarity. A face scan is made from previously recorded pictures and the face in the live camera is compared with the recorded pictures. If the similarity is 80 and above, it is determined that the person is not registered in the hospital system.

```
id Unknown
uyum Uyum= 87.0%
id Unknown
uyum Uyum= 88.0%
id Unknown
```

6.1.3 HOSFACE TC IDENTITY CHECK PAGE

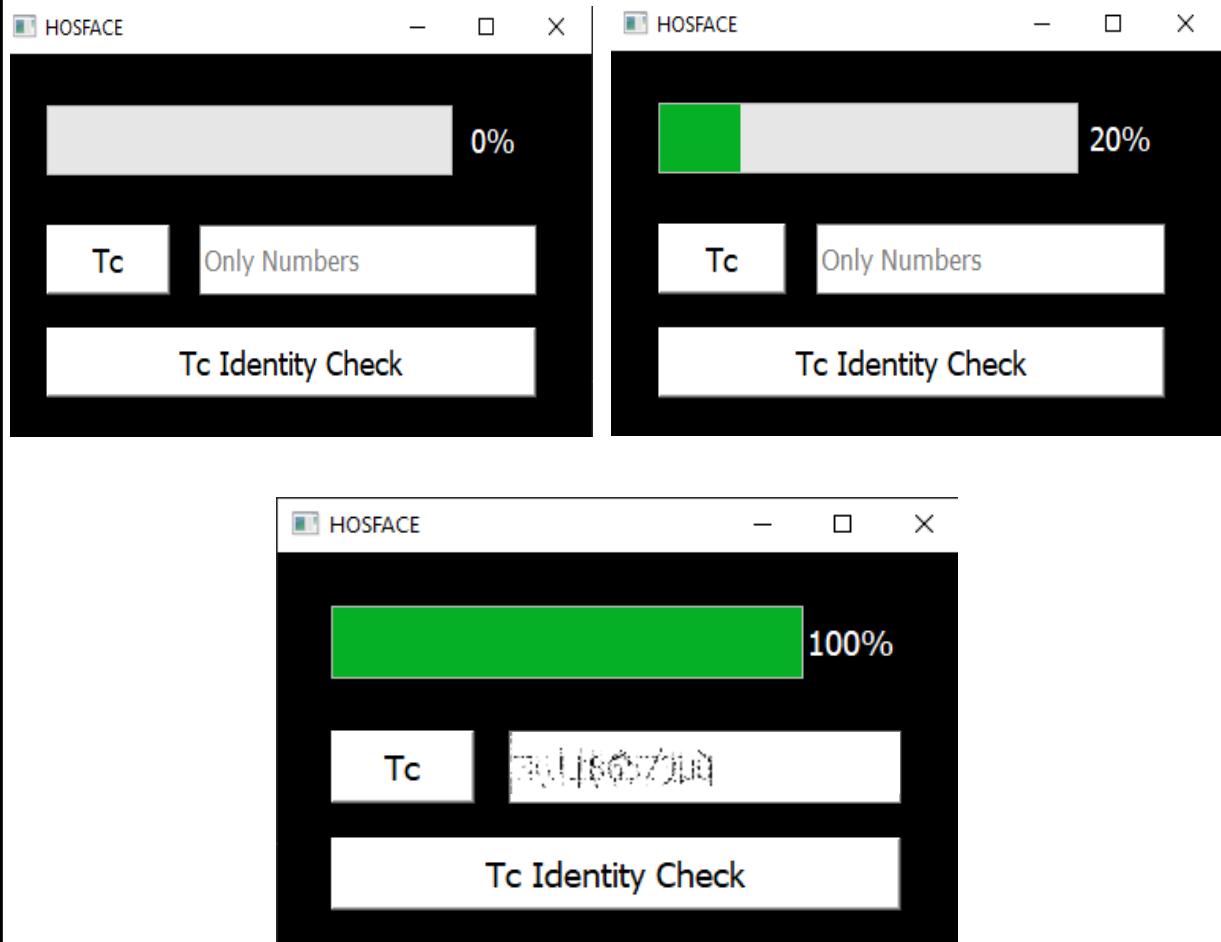
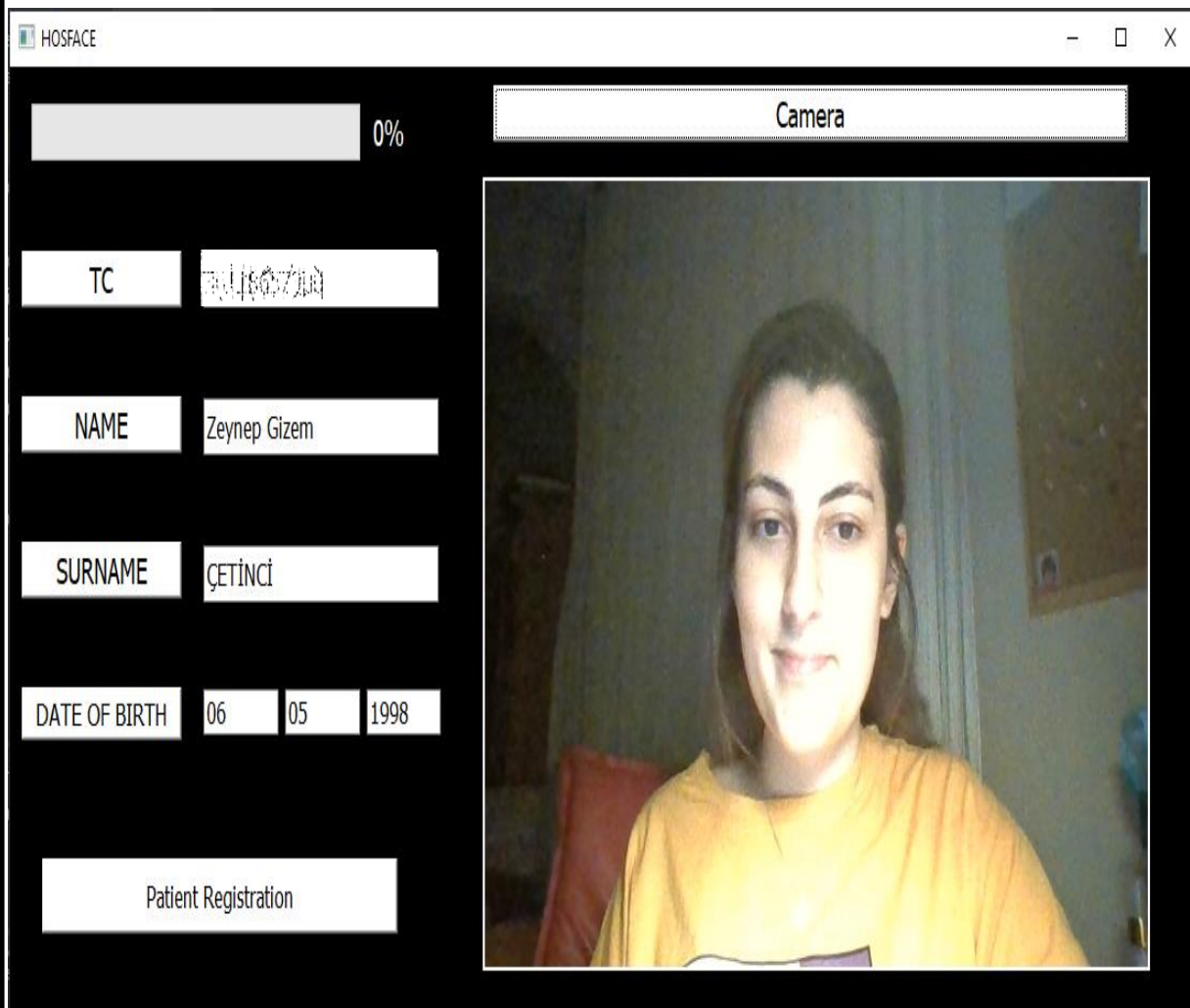


Fig 5.3 HOSFACE TC Identity Check Page

This page is a page we have made in order to write your TC ID number on the screen and check it in case you press your TC ID number and tell your TC ID number to the microphone within 10 seconds, considering the possibility of any malfunction in the webcam machine. If your TR identity number is not registered in the system, it will be directed to the registration page.

6.1.4 HOSFACE PATINT REGISTRATION PAGE



The screenshot displays the HOSFACE Patient Registration interface. It features a black background with white text and input fields. At the top left, there is a progress bar indicating 0% completion. Below this, the registration form includes fields for TC (71160700), NAME (Zeynep Gizem), SURNAME (ÇETİNCİ), and DATE OF BIRTH (06/05/1998). A 'Patient Registration' button is located at the bottom left. On the right side, there is a 'Camera' button and a video feed showing a woman with dark hair wearing a yellow shirt.

Fig 5.4 HOSFACE Patient Registration Page

The TC identification number entered on the TC identification page on the patient registration page is automatically written on the screen. On this page, name, surname, date of birth parts are written when the name, surname, date of birth names are pressed into the microphone by pressing the buttons. By pressing the camera button, a picture of the face is taken in the system and recording is made. When the patient record button is pressed, the information is written to the SQLite database and the captured picture is trained.


Fifty pictures of the patient mentioned above are taken for registration, and each picture is recorded with the patient's id number. Thanks to the id numbers here, it becomes easier for us to access the information of the patient. The pictures taken are then trained and converted into an yml file in the name of 'eğitim.yml'

```

main %
text: 50796402146
^
|
[INFO] yüzler eğitiliyor. Birkaç saniye bekleyin ...
id= 1
id= 1
id= 1
id= 1
id= 1
id= 1

```



Ad	Değiştirme tarihi	Tür	Boyut
 eğitim.yml	2.05.2021 18:32	YML Dosyası	19.360 KB

6.1.5 HOSFACE SELECT POLICLINIC PAGE

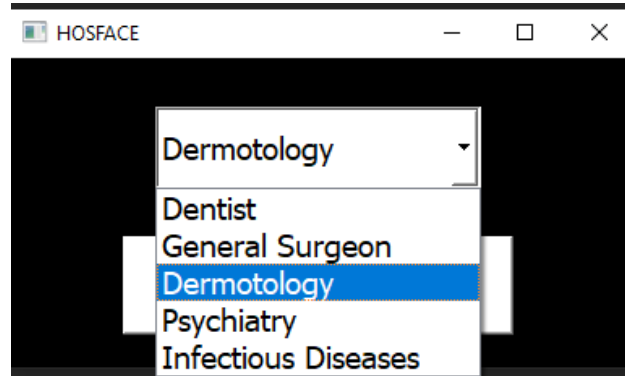


Fig 5.5 HOSFACE Select Polyclinic Page

On the polyclinic selection page that opens, the polyclinic to go to is selected and continued.

6.1.6 HOSFACE QUENCE NUMBER PAGE

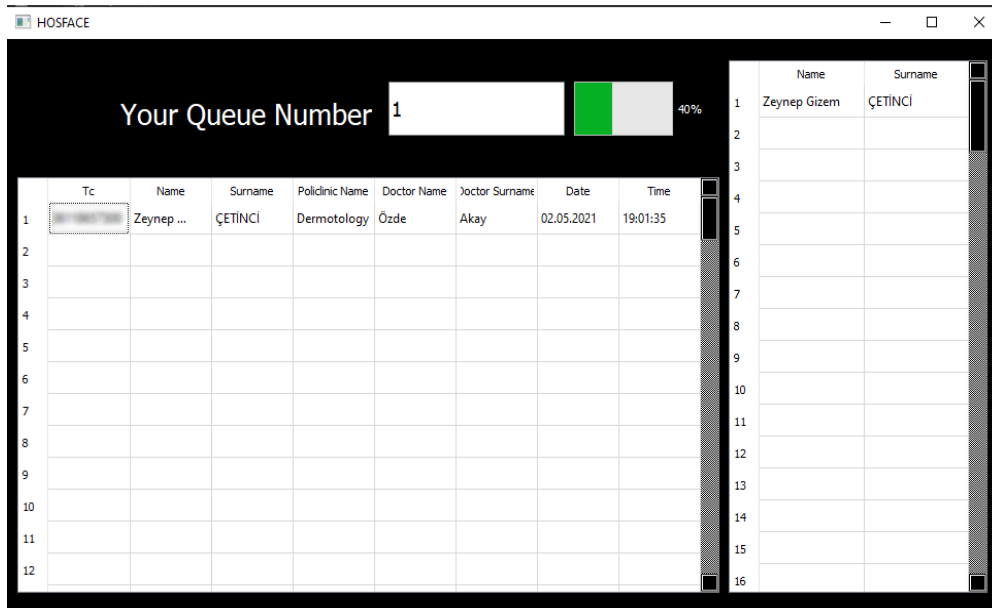


Fig 5.6 HOSFACE Quence Number Page

The page of seeing the sequence number is the page that appears last. No action is taken here, you can see your sequence number and the appointments you have previously made for 10 seconds on this screen. After 10 seconds, the screen turns off automatically and the welcome page opens.

6.2 SCREEN SHOTS FOR KOWN REGISTER

6.2.1 HOSFACE WELCOME PAGE

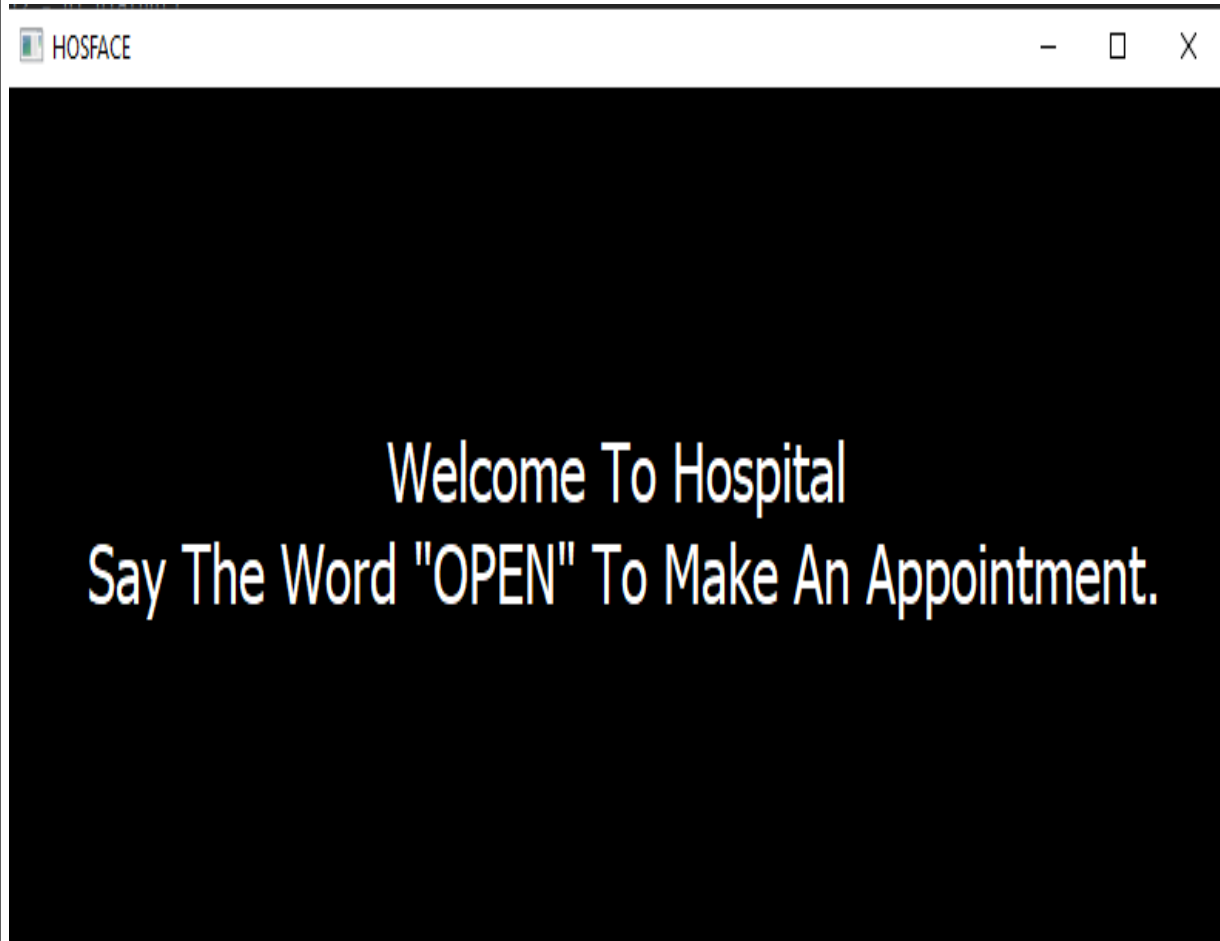


Fig 6.1 HOSFACE Welcome Page

This page will appear when you first start our project. On this page, there is a post that tells you the action you need to take to start the project.

6.2.2 HOSFACE WEBCAM PAGE – KNOWN REGISTER

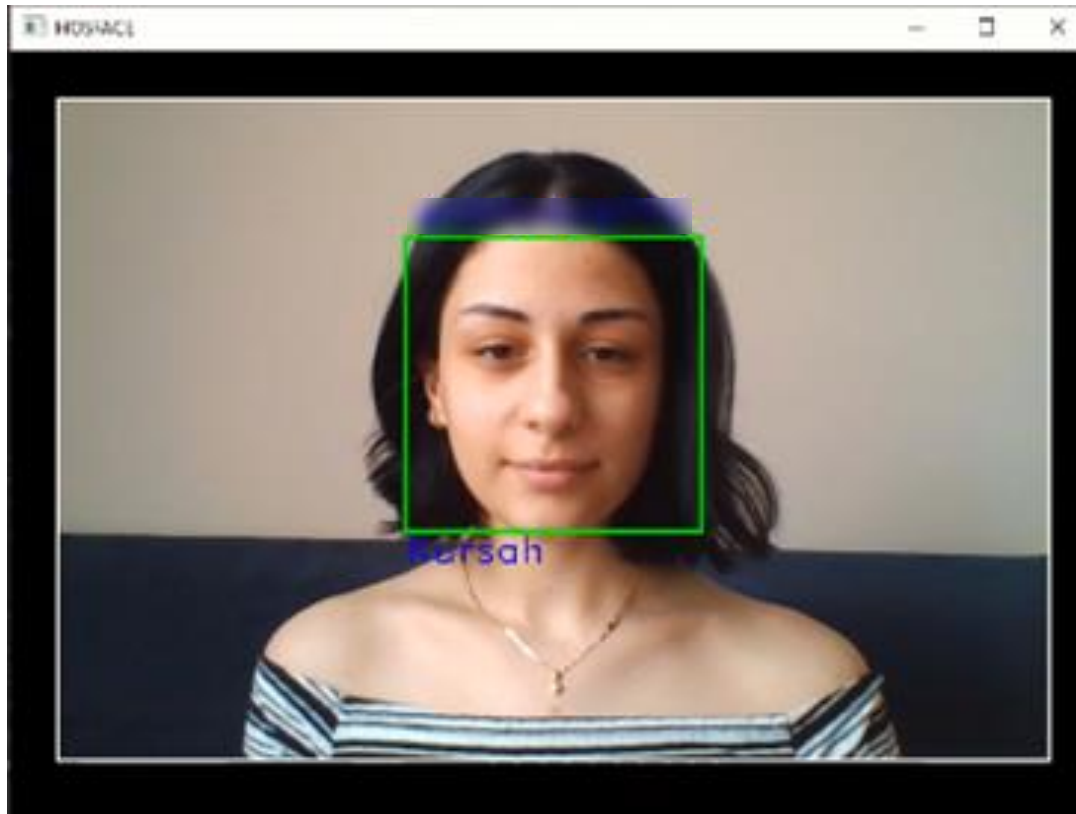


Fig 6.2 HOSFACE Webcam Page – Known Register

On the webcam page, if you have been registered to the hospital before, your face is compared with the photos of the registered users and its compatibility is checked. If this compatibility is below 80, the patient's id in the photo is compared with the id in the database, and the patient's Tc identification number and name are written in the frame framing his face in green.

```
1d46552424778  
4c942424778  
Mersah  
uyumUyum= 79.0%  
1d46552424778  
4c942424778  
Mersah  
uyumUyum= 79.0%  
1d46552424778
```

6.2.3 HOSFACE TC IDENTITY CHECK PAGE

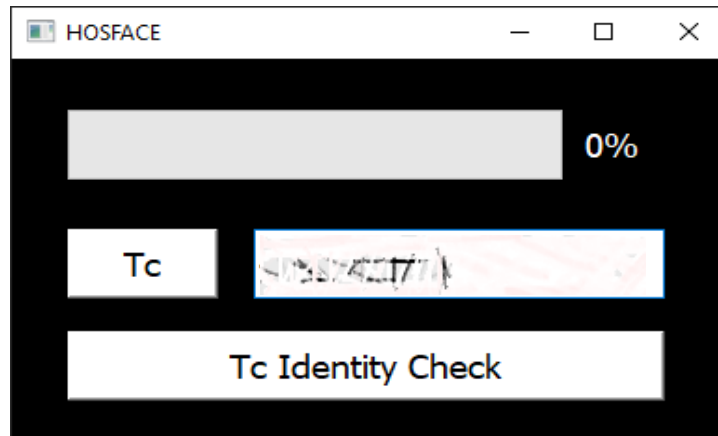


Fig 6.3 HOSFACE Tc Identity Check Page

On the Tc Identity check page, the patient's Tc identification number is written. If it is correct, it is continued by pressing the Tc Identity Check button, if not, the correct Tc identification number is told to the microphone by pressing the Tc button.

6.2.4 HOSFACE SELECT POLICLINIC PAGE

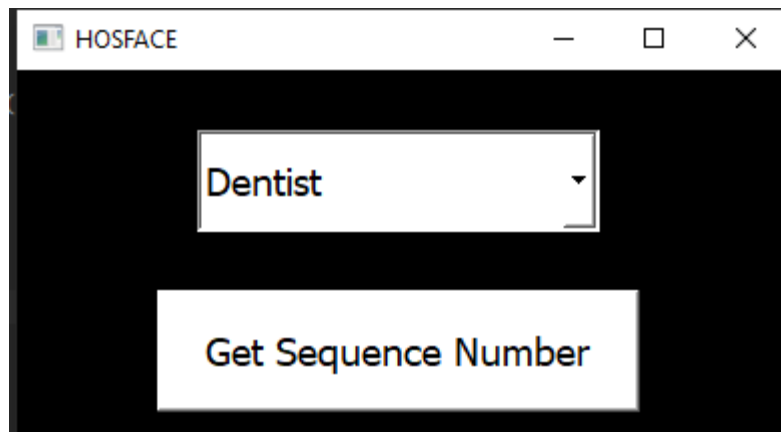


Fig 6.4 HOSFACE Select Polyclinic Page

On the polyclinic selection page that opens, the polyclinic to go to is selected and continued.

6.2.5 HOSFACE QUENCE NUMBER PAGE

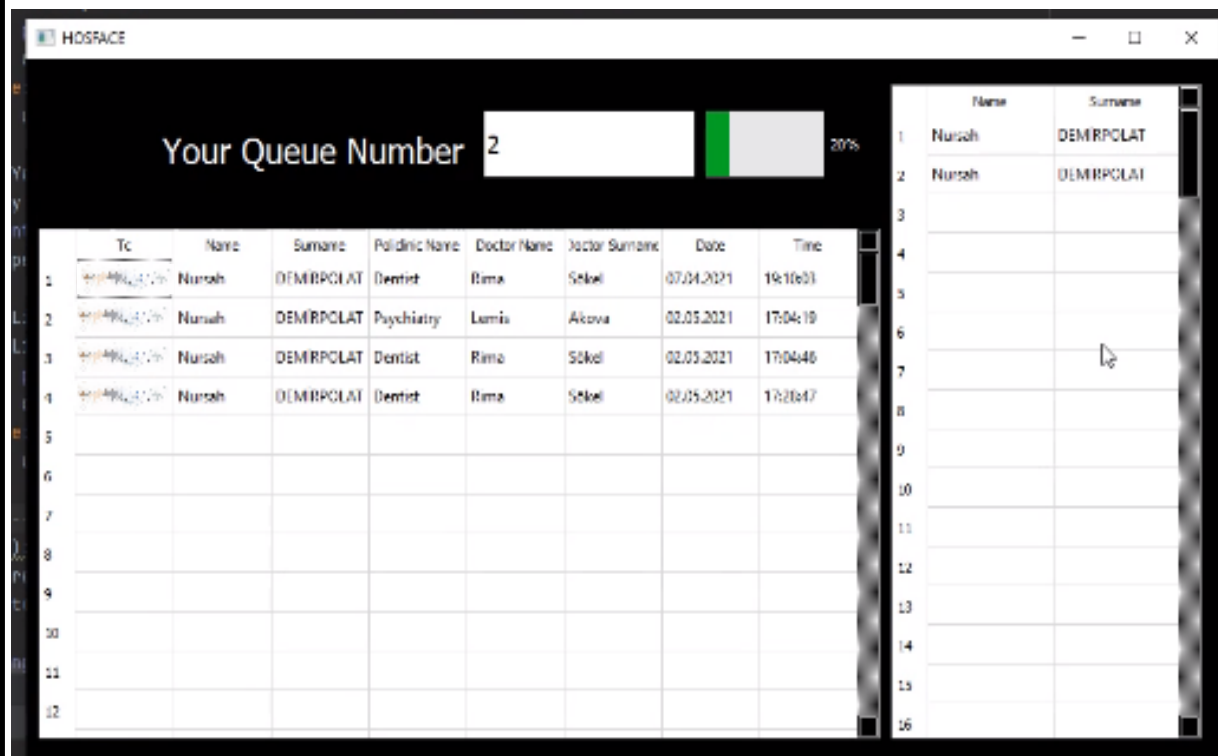


Fig 6.6 HOSFACE Quence Number Page

The page of seeing the sequence number is the page that appears last. No action is taken here, you can see your sequence number and the appointments you have previously made for 10 seconds on this screen. After 10 seconds, the screen turns off automatically and the welcome page opens.

SYSTEM TESTING

7.SYSTEM TESTING

7.1 MANUEL TESTING

7.1.1 TESTING FOR HARDWARE

Manual testing requires human intervention to perform the test. Each type of application can be tested manually, some types of tests, such as special tests and dummy tests, are more convenient to use manually. We can experience these tests as repetitive.

In the manual test, we paid attention to the working speed of the code, the working speed of the camera, the speed of the camera taking the picture, the speed of the microphone activation, the correct detection of the sound of the microphone, the speed of face recognition and its accuracy.

In order to test the running speed and efficiency of the code, we ran our program separately 20 times on 4 different windows-based laptops. According to the data we have obtained, the average running speed of our code is 01,29 seconds.

In order to test the working speed of the camera, we also experimented with 4 different laptops with different camera features. The reason we tried it on the laptop is that we will use our cameras on the webcam. In this way, we tested the working speed and accuracy by testing different camera features. The average opening speed of the camera is 10,86 seconds.

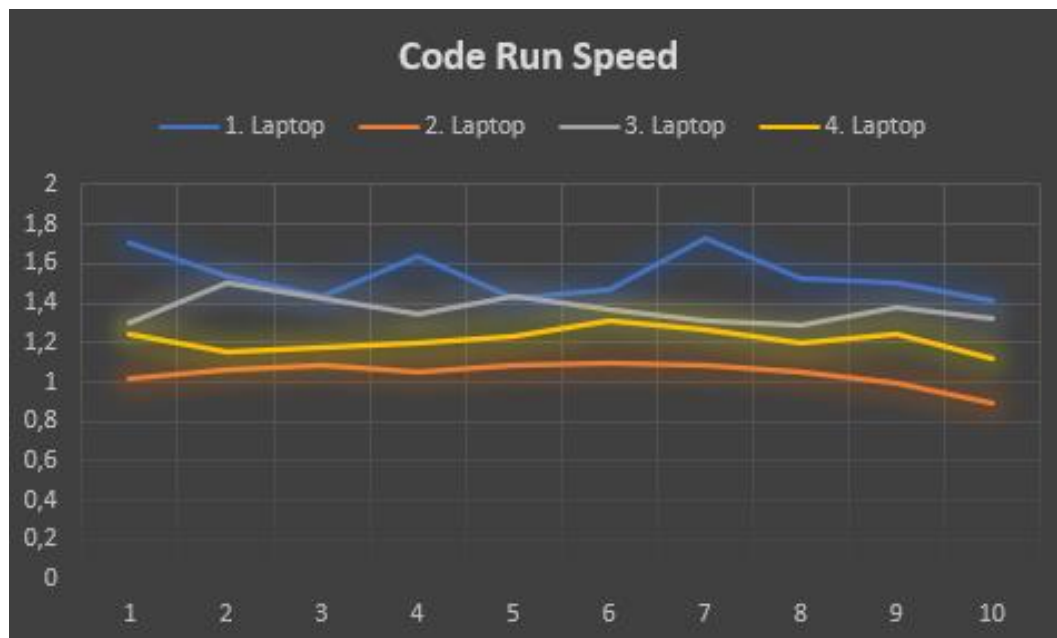
The speed at which the microphone is active is directly proportional to the running speed of the code. Because the moment we run our program is to be active on our microphone. However, this situation is only for the command we will say in the first opening panel. In this case, the average time for the microphone to be active for the first opening command of the microphone is 01,29 seconds. Afterwards, the average activation time of the microphone working by pressing the button is 00.47 seconds. The recording time of the microphone is 10 seconds. We set this situation in our code.

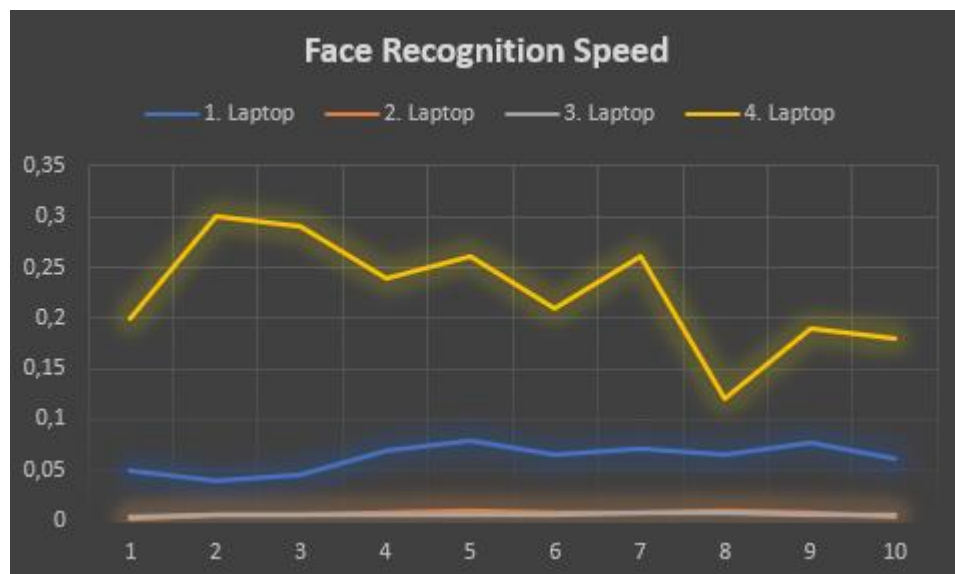
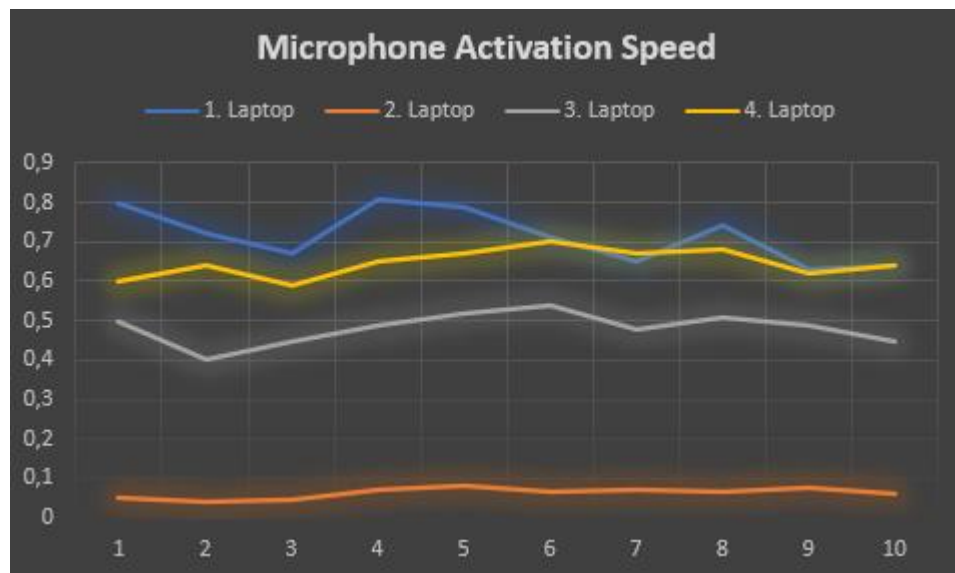
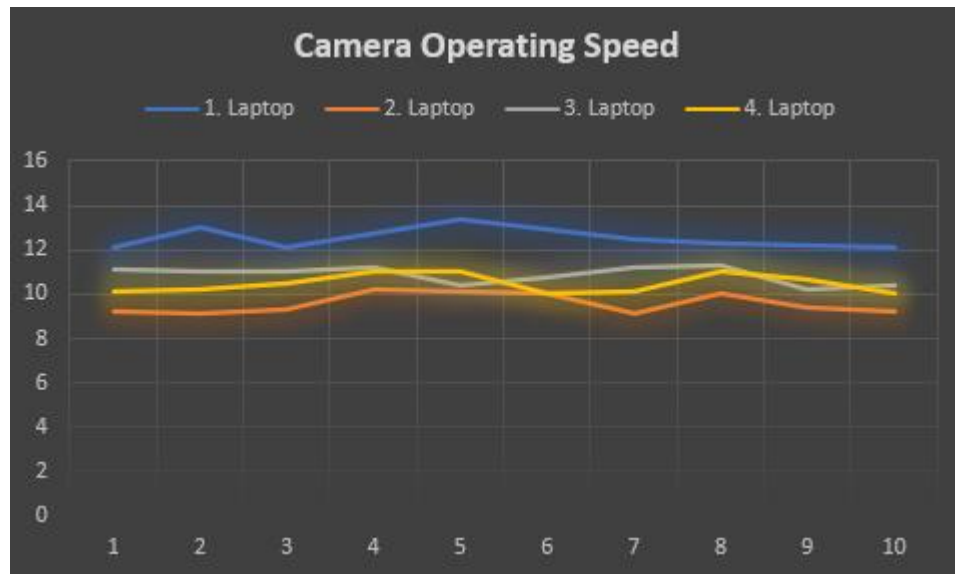
The problem of misperception of the sound of our microphone was misunderstood by 20 different people, some of the words of only 4 people. Our reasons for experiencing this situation stems from people's accents. However, there is no problem of making an incorrect recording since the error can be done by pressing the microphone again and again by pressing the microphone. Likewise, when there is an extra number in the patient's TC, our program sends an error message and the patient is asked to say his / her information again.

In the face recognition test of the camera, 4 different laptops were also used. And at the same time, face recognition detection was made in 3 different environments in these laptops. These have been experimented in a dim, very bright and sunny environment. After recording in a very bright environment, it can sometimes not be recognized when face recognition is detected in a dim environment. However, we did not have a problem when we recorded in the same environment and then a face recognition process was performed in the same environment. For this reason, our project will always have a constant light and environment in the hospital environment, so there will be no problem in facial recognition. However, we have created a tc inquiry section for problems that may occur. In this way, we have minimized the problems that may occur.

Finally, face recognition speed detection is 0,075 seconds on average detection speed in 4 different windows-based laptops.

7.1.1.1 TESTING TABLE





7.1.2 TESTING FOR FACE RECOGNITION

We registered 20 different people in the database to test the facial recognition system. Later, we trained the images of these 20 people in the database folder.

Apart from the 20 people we recorded, we put 10 people to face recognition test at 4 different light settings.

1.Test:

We did it under dim light, as a result of which only 3 out of 10 people matched with a different person.

2.Test:

We did it under sunlight, as a result of which only 2 out of 10 people were matched with a different person.

3.Test:

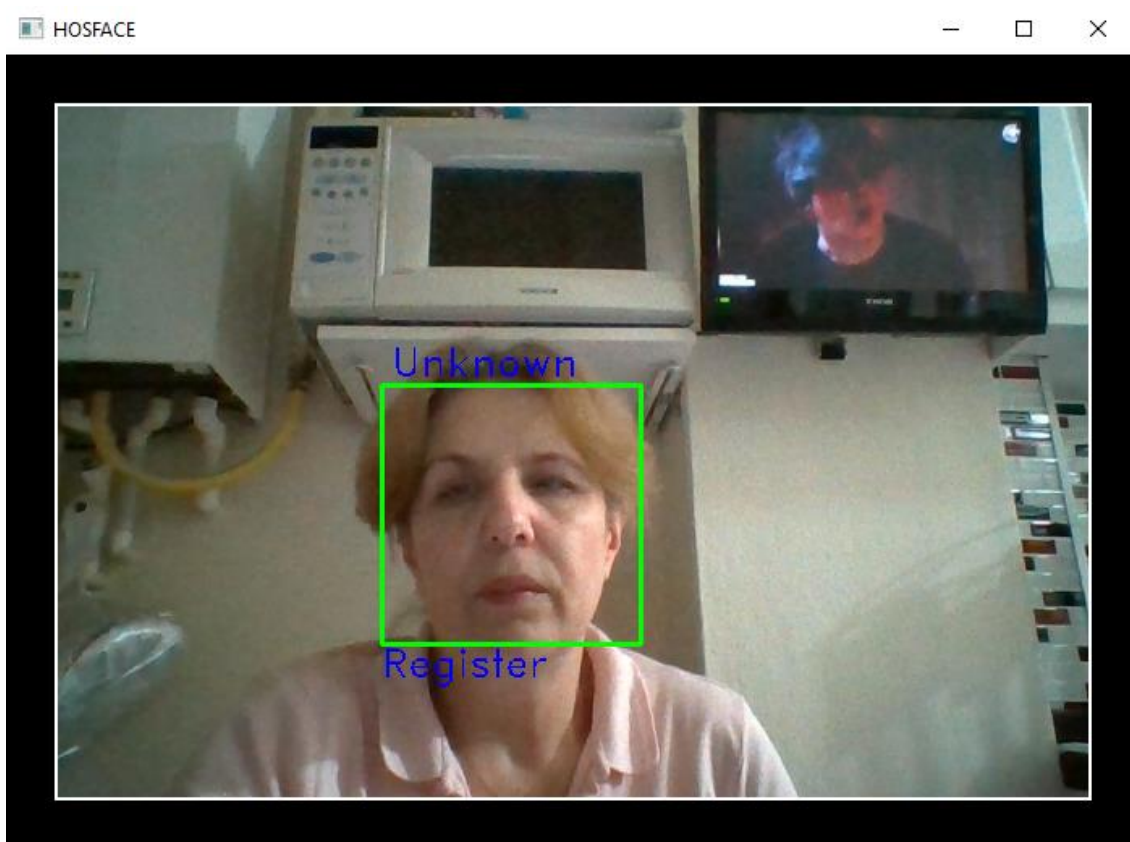
We did it under white light, as a result of which only 1 out of 10 people matched a different person.

4.Test:

We did it under yellow light, as a result of which only 2 out of 10 people were matched with a different person.

As a result of our tests, we achieved the best efficiency under white light. Since white light is used a lot in hospitals, we concluded that our project will not cause any problems in the hospital environment.

7.1.2.1 SCREEN SHOTS



9. user not registered in a dim environment.

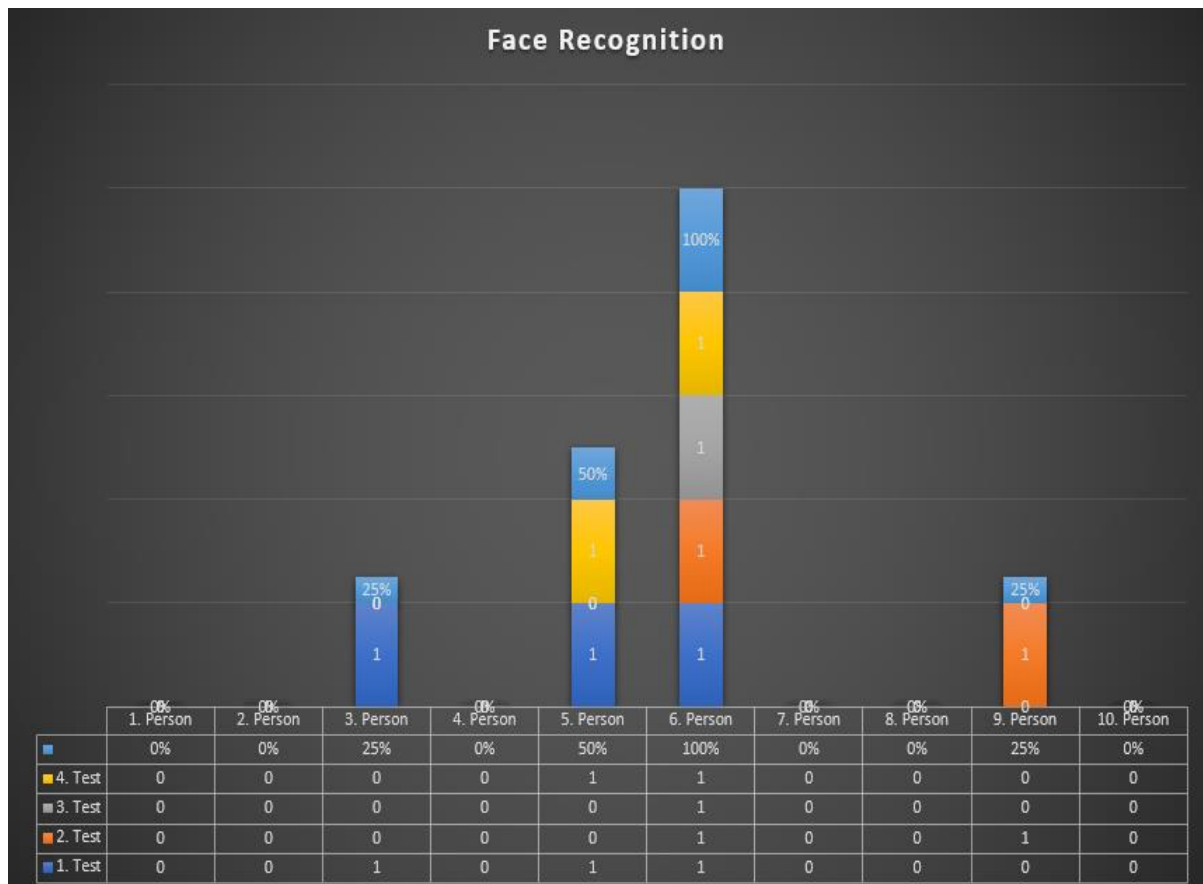
7.1.2.2 TESTING TABLE

This test has been tested on unregistered individuals.

0: Do not be like someone else

1: Comparison to someone else

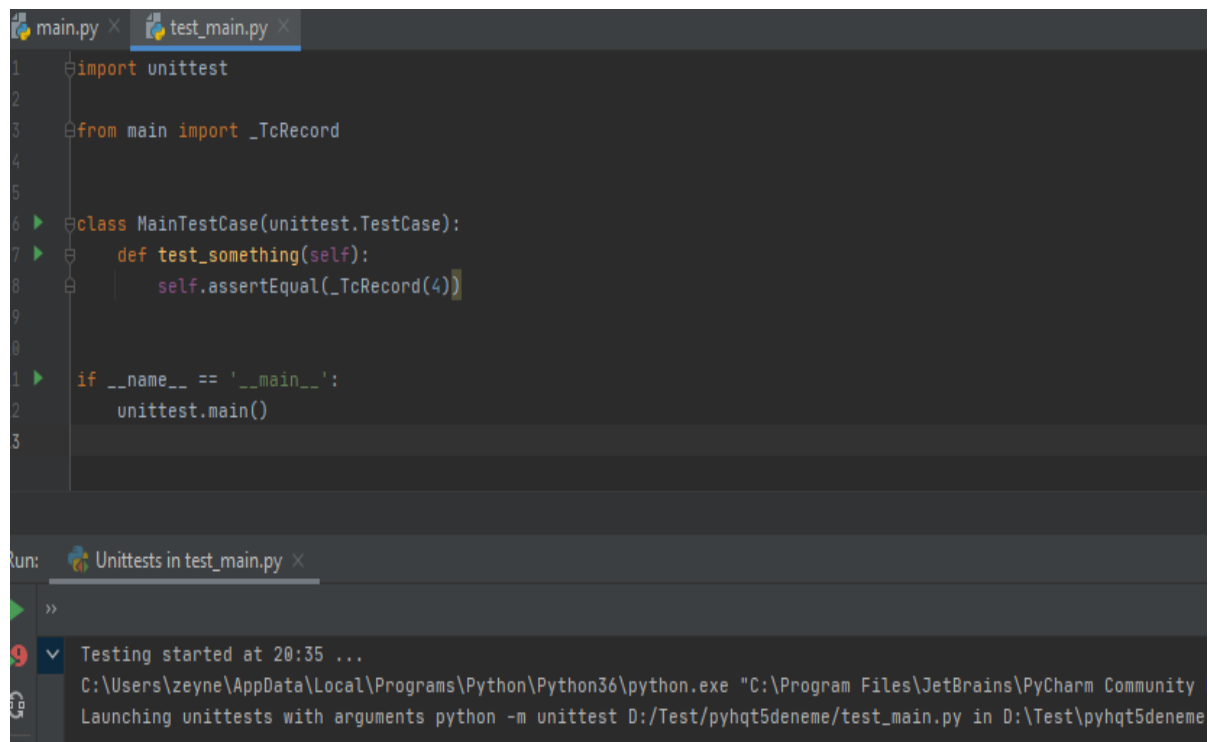
	A	B	C	D	E	F
1		1. Test	2. Test	3. Test	4. Test	
2	1. Person	0	0	0	0	0%
3	2. Person	0	0	0	0	0%
4	3. Person	1	0	0	0	25%
5	4. Person	0	0	0	0	0%
6	5. Person	1	0	0	1	50%
7	6. Person	1	1	1	1	100%
8	7. Person	0	0	0	0	0%
9	8. Person	0	0	0	0	0%
10	9. Person	0	1	0	0	25%
11	10. Person	0	0	0	0	0%
12		30%	20%	10%	20%	20%



7.2 UNIT TESTING

Unit testing approaches the system as a set of modules and examines the interactions and proper connectivity between them. Unit testing is a software testing method in which individual source code units are subjected to various tests to determine whether they are suitable for use. It determines and detects the quality of code.

Before doing unit testing, we created an init test page named 'test_main.py' that contains the name of the code we will test. We used the '_TcRecord' function in the main code to test it afterwards. We tested it according to the way it works as shown in the picture. We did not encounter any problems.



The screenshot displays the PyCharm IDE interface. The top editor window shows the code for 'test_main.py':

```
1 import unittest
2
3 from main import _TcRecord
4
5
6 class MainTestCase(unittest.TestCase):
7     def test_something(self):
8         self.assertEqual(_TcRecord(4))
9
10
11 if __name__ == '__main__':
12     unittest.main()
```

The bottom panel shows the 'Run' output for 'Unit tests in test_main.py'. It indicates that testing started at 20:35 and provides the command used to launch the unittests:

```
C:\Users\zeyne\AppData\Local\Programs\Python\Python36\python.exe "C:\Program Files\JetBrains\PyCharm Community
Launching unittests with arguments python -m unittest D:/Test/pyhqt5deneme/test_main.py in D:\Test\pyhqt5deneme
```

FUTURE SCOPE

8.FUTURE SCOPE

There may be improvements in the program in the future. Because products such as cameras and microphones are renewed and their quality increases day by day.

➤ **Image Operations:**

When the enhanced camera is displayed as an image, the camera image and camera shooting quality can be changed. Pictures taken in the recording for face recognition may be of better quality.

➤ **Microphone Operations:**

It can perceive the voice more clearly in the process of translating the voice made using the microphone into text. Thanks to the newly released better quality microphones, people's words can be perceived more clearly and accurately despite their accents.

➤ **Processor Speed:**

In Windows-based laptops, the speed of the program can be increased with the increase of the processor.

➤ **Database:**

Patient information records are made in the database. Thanks to the increased SSDs in the future, patient registration can be made more and faster than now.

CONCLUSION

9.CONCLUSION

Hosface is a hospital appointment system. It is designed so that you do not have to share your TC identity number with others when you go to the hospital, you can access your previous appointment information more easily, and you do not have to wait in a queue. Our aim is to reduce the covid -19 disease spread through speech during the pandemic period and to reduce the loss of time lost due to crowding in hospitals. The information on Hosface is never shared with third parties.

You can start the program with the command written on the welcome page to the hosface machine at the entrance of the hospital, and after the face scan, you can easily get a polyclinic appointment.

If you do not have any registration to the hospital, you can use the commands on the welcome page again, after the face scan, you can easily record with the voice conversation from the registration window and if you come again, you can make your appointment alone without any hassle.

REFERENCES

10. REFERENCES

- [1] Cuimei, Li, Qi Zhiliang, Jia Nan, and Wu Jianhua. 2017. "Human Face Detection Algorithm via Haar Cascade Classifier Combined with Three Additional Classifiers." In *2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, IEEE, 483–87. <http://ieeexplore.ieee.org/document/8265863/>.
- [2] GELEN, Mine Büşra, and Alparslan Serhat DEMİR. 2019. "Selection Of Information Technology Personnel For An Enterprise In The Process Of Industry 4.0 With The MultiMoora Method." *Sakarya University Journal of Science* 24(4): 1–1. <http://www.saujs.sakarya.edu.tr/tr/download/article-file/1210062>.
- [3] Das, Pranab, and Vijay Prasad. 2015. "VOICE RECOGNITION SYSTEM: SPEECH-TO-TEXT." *Journal of Applied and Fundamental Sciences* 169(2): 580.
- [4] Khan, Maliha, Sudeshna Chakraborty, Rani Astya, and Shaveta Khepra. 2019. "Face Detection and Recognition Using OpenCV." *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) 2019-Janua*: 116–19. <https://ieeexplore.ieee.org/document/8974493/>.
- [5] Padilla, R, Cff Costa Filho, and Mgf Costa. 2012. "Evaluation of Haar Cascade Classifiers Designed for Face Detection." *Journal of WASET* 6(4): 323–26. <http://www.waset.org/publications/2910>.
- [6] Bhosale, S T, Tejaswini Patil, and Pooja Patil. 2015. "SQLite: Light Database System." *International Journal of Computer Science and Mobile Computing* 44(4): 882–85.
- [7] Bi, Chunyue. 2009. *8 WSEAS Transactions on Computers Research and Application of SQLite Embedded Database Technology*.
- [8] Goyal, Kruti, Kartikey Agarwal, and Rishi Kumar. 2017. "Face Detection and Tracking: Using OpenCV." In *2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA)*, IEEE, 474–78. <http://ieeexplore.ieee.org/document/8203730/>.
- [9] Kalas, Mamata S. 2014. "Real Time Face Detection And Tracking Using Opencv." (1): 41–44.
- [10] Goyal, Kruti, Kartikey Agarwal, and Rishi Kumar. 2017. "Face Detection and Tracking: Using OpenCV." In *2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA)*, IEEE, 474–78. <http://fivedots.coe.psu.ac.th/~ad/jg/nui07/>.

- [11] Bashar, Dr. Abul. 2019. "SURVEY ON EVOLVING DEEP LEARNING NEURAL NETWORK ARCHITECTURES." *Journal of Artificial Intelligence and Capsule Networks* 2019(2): 73–82.
- [12] Tavanaei, Amirhossein et al. 2019. "Deep Learning in Spiking Neural Networks." *Neural Networks* 111: 47–63. <https://doi.org/10.1016/j.neunet.2018.12.002>.
- [13] Deng, Li, Geoffrey Hinton, and Brian Kingsbury. 2013. "New Types of Deep Neural Network Learning for Speech Recognition and Related Applications: An Overview." *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*: 8599–8603.
- [14] Ngiam, Jiquan et al. 2011. "Multimodal Deep Learning." *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*: 689–96. https://icml.cc/2011/papers/399_icmlpaper.pdf.
- [15] SATO, Yoshiharu. 1995. "Artificial Neural Networks and Statistical Model." *Japanese journal of applied statistics* 24(2): 77–88. <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=D4438D9BC64A51F7E3814833CB55F7DD?doi=10.1.1.27.699&rep=rep1&type=pdf>.

11.CODE

11.1 MAIN.PY

```

import ctypes
import sys
import re
from PyQt5.QtWidgets import *
from ilkui import *
from ikinciui import *
from Ucui import *
from HastaAramaui import *
from Aramaui import *
from Acilisui import *
import speech_recognition as sr
import time
import threading
import cv2
from PyQt5.QtWidgets import QApplication
from PyQt5.QtGui import QImage
from PyQt5.QtGui import QPixmap
import numpy as np
from PIL import Image
import os
# ----- Interface -----
Uygulama = QApplication(sys.argv)
KayıtPen = QMainWindow()
ui = Ui_MainWindow()
ui.setupUi(KayıtPen)
SecmePen = QMainWindow()
ui2 = Ui_Dialog()
ui2.setupUi(SecmePen)
SıraPen = QMainWindow()
ui3 = Ui_Form()
ui3.setupUi(SıraPen)

WelcomeHomePage = QMainWindow()
ui4 = Ui_HomePage()
ui4.setupUi(WelcomeHomePage)

HastaArama = QMainWindow()
ui5 = Ui_HastaArama()
ui5.setupUi(HastaArama)

Kamera = QMainWindow()
ui6 = Ui_Cam()
ui6.setupUi(Kamera)
WelcomeHomePage.show()

```

```

# ----- DATABASE -----
import sqlite3
import datetime
global curs
global conn
conn = sqlite3.connect('Hospital.db') # db veri tabanı yoksa oluşturur
curs = conn.cursor() # veri tabanı ile dialog kurmayı sağlıyor.

# ----- SPEECH RECOGNITION -----
def _record():
    deneme = threading.Thread(target=showTime)
    deneme.start()
    liste = []
    Zaman = datetime.datetime.now()
    Day = int(Zaman.strftime("%d"))
    Mount = int(Zaman.strftime("%m"))
    Year = int(Zaman.strftime("%Y"))
    for x in range(3):
        recognizer1 = sr.Recognizer()
        with sr.Microphone() as source:
            recognizer1.adjust_for_ambient_noise(source, duration=1)
            print("Recording for 5 seconds")
            recorded_audio = recognizer1.listen(source, timeout=5, phrase_time_limit=5)
    print("Done recording")
    try:
        print("Recognizing the text")
        text1 = recognizer1.recognize_google(recorded_audio, language='tr')
        print("Decoded Text : {}".format(text1))
        liste.append(text1)

    except Exception as ex:
        print(ex)

    LisDay = int(liste[0])
    LisAy = int(liste[1])
    LisYıl = int(liste[2])

    if LisYıl > Year:
        print("Error Year Big")
        ctypes.windll.user32.MessageBoxW(0, "Year Big", "YEAR", 0)
    elif LisYıl == Year:
        ui.LineYear.setText("{} {}".format(int(re.sub('[\s+|\\W+]', '', liste[2]))))
        if LisAy > Mount:
            print("Mistake Month Is Bigger Than This Month")
            ctypes.windll.user32.MessageBoxW(0, "The Month You Entered Is Greater Than The Current Month", "MONTH", 0)
        elif LisAy > 12:
            print("Error Month Greater Than 12")
            ctypes.windll.user32.MessageBoxW(0, "Month Greater Than 12", "MONTH", 0)

```

```

else:
    ui.LineMounth.setText("{} ".format(int(re.sub('[\s+|\\W+]', '', liste[1]))))
    if LisDay > Day:
        print("Error Day Greater Than Today")
        ctypes.windll.user32.MessageBoxW(0, "The Day Is Greater Than Current
Time", "DAY", 0)
    elif LisDay > 31:
        print("Error Day Greater Than 31")
        ctypes.windll.user32.MessageBoxW(0, "Day Greater Than 31", "DAY", 0)
else:
    ui.LineDay.setText("{} ".format(int(re.sub('[\s+|\\W+]', '', liste[0]))))
else:
    ui.LineYear.setText("{} ".format(int(re.sub('[\s+|\\W+]', '', liste[2]))))
    if LisAy > 12:
        print("Error Month Is Greater Than 12")
        ctypes.windll.user32.MessageBoxW(0, "Month Is Greater Than 12", "MONTH",
0)
    else:
        ui.LineMounth.setText("{} ".format(int(re.sub('[\s+|\\W+]', '', liste[1]))))
        if LisDay > 31:
            print("Error Day Greater Than 31")
            ctypes.windll.user32.MessageBoxW(0, "Day Greater Than 31", "DAY", 0)
        else:
            ui.LineDay.setText("{} ".format(int(re.sub('[\s+|\\W+]', '', liste[0]))))

# -----TC SPEECH RECOGNITION -----
def _TcRecord(a):
    deneme = threading.Thread(target=showTime)
    deneme.start()
    liste = []
    for x in range(1):
        recognizer1 = sr.Recognizer()
        with sr.Microphone() as source:
            recognizer1.adjust_for_ambient_noise(source, duration=1)
            print("Recording for 10 seconds")
            recorded_audio = recognizer1.listen(source, timeout=10, phrase_time_limit=10)
            print("Done recording")
        try:
            print("Recognizing the text")
            text1 = recognizer1.recognize_google(recorded_audio, language='tr')
            print("Decoded Text : {}".format(text1))
            liste.append(text1)

        except Exception as ex:
            print(ex)
    if a == "1":
        ui5.LineTcGiris.setText("{} ".format(int(re.sub('[\s+|\\W+]', '', liste[0]))))
        text = ui5.LineTcGiris.text()
    elif a == "2":
        ui.LineTc.setText("{} ".format(int(re.sub('[\s+|\\W+]', '', liste[0]))))
        text = ui.LineTc.text()

```

```
elif a == "3":
    ui.LineAd.setText("{} ".format(liste[0].title()))
    text = ui.LineAd.text()
elif a == "4":
    ui.LineSoyad.setText("{} ".format(liste[0]).upper())
    text = ui.LineSoyad.text()
print("text: " + text)
print(a)

# ----- Welcome -----
def _WelcomeRecord():
    deneme = threading.Thread(target=showTime)
    deneme.start()
    for x in range(1):
        recognizer1 = sr.Recognizer()
        with sr.Microphone() as source:
            recognizer1.adjust_for_ambient_noise(source, duration=1)
            print("Recording for 10 seconds")
            recorded_audio = recognizer1.listen(source, timeout=10, phrase_time_limit=10)
            print("Done recording")
        try:
            print("Recognizing the text")
            textWel = recognizer1.recognize_google(recorded_audio, language='tr')
            print("Decoded Text : {}".format(textWel))
        except Exception as ex:
            _WelcomeRecord()
        textOpen = str(textWel)
        print(textOpen)
    while True:
        if textWel == "open":
            WelcomeHomePage.hide()
            Kamera.show()
            acilis()
            break
        elif textWel == "Open":
            WelcomeHomePage.hide()
            Kamera.show()
            acilis()
            break
        elif textWel == "OPEN":
            WelcomeHomePage.hide()
            Kamera.show()
            acilis()
            break
        elif ' open ' in textOpen:
            WelcomeHomePage.hide()
            Kamera.show()
            acilis()
            break
```



```
elif ' Open ' in textOpen:
    WelcomeHomePage.hide()
    Kamera.show()
    acilis()
    break
elif ' OPEN ' in textOpen:
    WelcomeHomePage.hide()
    Kamera.show()
    acilis()
    break
elif 'open ' in textOpen:
    WelcomeHomePage.hide()
    Kamera.show()
    acilis()
    break
elif 'Open ' in textOpen:
    WelcomeHomePage.hide()
    Kamera.show()
    acilis()
    break
elif 'OPEN ' in textOpen:
    WelcomeHomePage.hide()
    Kamera.show()
    acilis()
    break
elif ' open' in textOpen:
    WelcomeHomePage.hide()
    Kamera.show()
    acilis()
    break
elif ' Open' in textOpen:
    WelcomeHomePage.hide()
    Kamera.show()
    acilis()
    break
elif ' OPEN' in textOpen:
    WelcomeHomePage.hide()
    Kamera.show()
    acilis()
    break
else:
    _WelcomeRecord()
```

```
# -----
def arayuz(a):
    t2 = threading.Thread(target=_TcRecord(a))
    t2.daemon = True
    t2.start()
    t2.join()
```

```

def arayuz2():
    t1 = threading.Thread(target=_record)
    t1.daemon = True
    t1.start()
    t1.join()

# -----CAMERA -----
def _kamera():
    ui.cap = cv2.VideoCapture(0)
    a = int(0)
    sor = "SELECT COUNT(Patient_id_tc) FROM Patient "
    curs.execute(sor, )
    data = curs.fetchall()
    c = str(data)
    b = c.strip("[ ( , )]")
    print(b)
    i = int(b)
    i = i + 1
    print(i)
    n = str(i)
    while True:
        face_detector =
cv2.CascadeClassifier('Cascades/haarcascade_frontalface_default.xml')
        ret, img = ui.cap.read()
        img = cv2.flip(img, 1)
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        height, width, channel = img.shape
        step = channel * width
        qImg = QImage(img.data, width, height, step, QImage.Format_RGB888)
        ui.label.setPixmap(QPixmap.fromImage(qImg))
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        yuzler = face_detector.detectMultiScale(gray, 1.3, 5)
        a = a + 1
        for (x, y, w, h) in yuzler:
            cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
            cv2.imwrite('VeriDepo/%s.%s.jpg' % (n, a), gray[y:y + h, x:x + w])
        if cv2.waitKey(1) & a == 50:
            break
    ui.cap.release()
    cv2.destroyAllWindows()

# -----
def Tanımlama():
    path = 'VeriDepo'
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    detector = cv2.CascadeClassifier("Cascades/haarcascade_frontalface_default.xml")
    # imajların alınması ve etiketlenmesi için fonksiyon
    def getImagesAndLabels(path):
        imagePath = [os.path.join(path, f) for f in os.listdir(path)]
        ornekler = []
        ids = []

```

```

for imagePath in imagePaths:
    PIL_img = Image.open(imagePath).convert('L') # gri
    img_numpy = np.array(PIL_img, 'uint8')
    id = int(os.path.split(imagePath)[-1].split(".")[0])
    print("id= ", id)
    yuzler = detector.detectMultiScale(img_numpy)
    for (x, y, w, h) in yuzler:
        ornekler.append(img_numpy[y:y + h, x:x + w])
        ids.append(id)
    return ornekler, ids

print("\n [INFO] yuzler eğitiliyor. Birkaç saniye bekleyin ...")
yuzler, ids = getImagesAndLabels(path)
recognizer.train(yuzler, np.array(ids))
print(recognizer.train(yuzler, np.array(ids)))
# Modeli egitim/egitim.yml dosyasına kaydet
recognizer.write('egitim/egitim.yml') # Dikkat! recognizer.save() Raspberry Pi
üzerinde çalışmıyor
# Eğitilen yüz sayısını göster ve kodu sonlandır
print(f"\n [INFO] {len(np.unique(ids))} yüz eğitildi. Betik sonlandırılıyor.")

# ----- Adding a Record -----
def Kayit():
    _tc = ui.LineTc.text()
    _name = ui.LineAd.text()
    _surname = ui.LineSoyad.text()
    _dateDay = ui.LineDay.text()
    _dateMonth = ui.LineMounth.text()
    _dateYear = ui.LineYear.text()
    _date = (_dateDay + "." + _dateMonth + "." + _dateYear)
    curs.execute("INSERT INTO Patient VALUES (?, ?, ?, ?)", (_tc, _name, _surname,
    _date,))
    conn.commit()
    deneme = threading.Thread(target=Tanımlama)
    deneme.start()
    SecmePen.show()
    KayıtPen.close()

# ----- Combobox -----
def Secim():
    deneme = threading.Thread(target=showTime3)
    deneme.start()
    _tc = ui5.LineTcGiris.text()
    Zaman = datetime.datetime.now()
    _Date1 = Zaman.strftime("%d.%m.%Y")
    secim = ui2.comboBox.currentText()
    print("text :", secim)
    if secim == "Dentist":
        _BolumID = "1"
        _DoctorId = "3"
        tc = ui5.LineTcGiris.text()

```

```

Zaman = datetime.datetime.now()
_Date = Zaman.strftime("%d.%m.%Y")
_Time = Zaman.strftime("%H:%M:%S")
curs.execute(
    "INSERT INTO
Appointment(Policlinic_id,Patient_id_tc,Doctor_id,Time,Date) VALUES (?, ?, ?,? ,?)",
    (_BolumID, _tc, _DoctorId, _Time, _Date))
elif secim == "General Surgeon":
    _BolumID = "2"
    _DoctorId = "5"
    _tc = ui5.LineTcGiris.text()
    Zaman = datetime.datetime.now()
    _Date = Zaman.strftime("%d.%m.%Y")
    _Time = Zaman.strftime("%H:%M:%S")
    curs.execute("INSERT INTO
Appointment(Policlinic_id,Patient_id_tc,Doctor_id,Time,Date) VALUES (?, ?, ?,? ,?)",
    (_BolumID, _tc, _DoctorId, _Time, _Date))
elif secim == "Dermatology":
    _BolumID = "3"
    _DoctorId = "1"
    _tc = ui5.LineTcGiris.text()
    Zaman = datetime.datetime.now()
    _Date = Zaman.strftime("%d.%m.%Y")
    _Time = Zaman.strftime("%H:%M:%S")
    curs.execute("INSERT INTO
Appointment(Policlinic_id,Patient_id_tc,Doctor_id,Time,Date) VALUES (?, ?, ?,? ,?)",
    (_BolumID, _tc, _DoctorId, _Time, _Date))
elif secim == "Psychiatry":
    _BolumID = "4"
    _DoctorId = "2"
    _tc = ui5.LineTcGiris.text()
    Zaman = datetime.datetime.now()
    _Date = Zaman.strftime("%d.%m.%Y")
    _Time = Zaman.strftime("%H:%M:%S")
    curs.execute("INSERT INTO
Appointment(Policlinic_id,Patient_id_tc,Doctor_id,Time,Date) VALUES (?, ?, ?,? ,?)",
    (_BolumID, _tc, _DoctorId, _Time, _Date))
elif secim == "Infectious Diseases":
    _BolumID = "5"
    _DoctorId = "4"
    _tc = ui5.LineTcGiris.text()
    Zaman = datetime.datetime.now()
    _Date = Zaman.strftime("%d.%m.%Y")
    _Time = Zaman.strftime("%H:%M:%S")
    curs.execute("INSERT INTO
Appointment(Policlinic_id,Patient_id_tc,Doctor_id,Time,Date) VALUES (?, ?, ?,? ,?)",
    (_BolumID, _tc, _DoctorId, _Time, _Date))
else:
    print("HATA")

```

```

print("_BolumID :", _BolumID)
print("_DoctorId :", _DoctorId)
print("_tc :", _tc)
print("_Date :", _Date)
print("_Time :", _Time)
conn.commit()
SıraPen.show()
SecmePen.close()
secim = ui2.comboBox.currentText()
print("text :", secim)
Tc = ui5.LineTcGiris.text()
print("text :", Tc)
sor = "SELECT COUNT(Appointment.Policlinic_id) FROM Patient INNER JOIN
Appointment ON Patient.Patient_id_tc = Appointment.Patient_id_tc INNER JOIN
Doctor ON Appointment.Doctor_id = doctor.Doctor_id INNER JOIN Policlinic ON
Appointment.Policlinic_id = policlinic.Policlinic_id WHERE policlinic.Policlinic_id = ?
AND Appointment.Date = ? ; "
curs.execute(sor, (_BolumID, _Date1,))
data = curs.fetchall()
a = str(data)
ui3.lineEdit.setText("{} ".format(a.strip('[(,)]')))
ui3.tableWidget.clear()
ui3.tableWidget.setHorizontalHeaderLabels(
    ('Tc', 'Name', 'Surname', 'Policlinic Name', 'Doctor Name', 'Doctor Surname',
'Date', 'Time'))
ui3.tableWidget.horizontalHeader().setSectionResizeMode(QHeaderView.Stretch)
# Tablonun taşmasını önler (sıkıştırır.)
sor2 = "SELECT
Patient.Patient_id_tc,Patient.Name,Patient.Surname,Policlinic.Policlinic_name,doctor.D
octor_name,Doctor.Doctor_surname,Appointment.Date,Appointment.Time FROM
Patient INNER JOIN Appointment ON Patient.Patient_id_tc =
Appointment.Patient_id_tc INNER JOIN Doctor ON Appointment.Doctor_id =
doctor.Doctor_id INNER JOIN Policlinic ON Appointment.Policlinic_id =
policlinic.Policlinic_id WHERE Appointment.Patient_id_tc = ? ; "
curs.execute(sor2, (Tc,))
for satirIndex, satirVeri in enumerate(curs):

# 2 parametreyi enumerate tutar
for sutunIndex, sutunVeri in enumerate(satirVeri):
    ui3.tableWidget.setItem(satirIndex, sutunIndex,
QTableWidgetItem(str(sutunVeri)))

ui3.tableWidget_2.clear()
ui3.tableWidget_2.setHorizontalHeaderLabels(('Name', 'Surname'))
ui3.tableWidget_2.horizontalHeader().setSectionResizeMode(
    QHeaderView.Stretch)

# Tablonun taşmasını önler (sıkıştırır.)
sor3 = "SELECT Patient.Name,Patient.Surname FROM Patient INNER JOIN
Appointment ON Patient.Patient_id_tc = Appointment.Patient_id_tc INNER JOIN
Doctor ON Appointment.Doctor_id = doctor.Doctor_id INNER JOIN Policlinic ON

```

```

Appointment.Policlinic_id = polyclinic.Policlinic_id WHERE polyclinic.Policlinic_id
= ? AND Appointment.Date = ? ; "
    curs.execute(sor3, (_BolumID, _Date1,))
    for satirIndex, satirVeri in enumerate(curs):

# 2 parametreyi enumerate tutar
    for sutunIndex, sutunVeri in enumerate(satirVeri):
        ui3.tableWidget_2.setItem(satirIndex, sutunIndex,
        QTableWidgetItem(str(sutunVeri)))

# ----- Patient Inquiry -----
def HastaSorgula():
    tc = ui5.LineTcGiris.text()
    ui.LineTc.setText("{} ".format(ui5.LineTcGiris.text()))
    sor = "SELECT COUNT(Patient_id_tc) FROM Patient WHERE Patient_id_tc = ?"
    curs.execute(sor, (tc,))
    data = curs.fetchall()
    a = str(data)
    b = a.strip('[(),]')
    print(b)
    if b == '0':
        print("kayıt Ol")
        HastaArama.close()
        KayıtPen.show()
    else:
        SecmePen.show()
        HastaArama.close()

# ----- Sura Notation -----
def showTime():
    minute = 0
    while True:
        if minute == 10:
            break
        else:
            pass
        minute = minute + 1
        time.sleep(1)
        QtCore.QCoreApplication.processEvents()
        ui.progressBar.setValue(minute)
def showTime2():
    minute = 0
    while True:
        if minute == 10:
            break
        else:
            pass
        minute = minute + 1
        time.sleep(1)
        QtCore.QCoreApplication.processEvents()
        ui5.progressBar.setValue(minute)

```

```

def showTime3():
    minute = 0
    while True:
        if minute == 10:
            break
        else:
            pass
        minute = minute + 1
        time.sleep(1)
        QtCore.QCoreApplication.processEvents()
        ui3.progressBar.setValue(minute)
        print(ui3.progressBar.text())
        if ui3.progressBar.text() == "100%":
            SıraPen.hide()
            WelcomeHomePage.show()
            break
    again()
# -----
def SesKayıtbaslama():
    deneme = threading.Thread(target=showTime2)
    deneme.start()
    a = "1"
    arayuz(a)
# -----Sinyal - Slot (tıklanınca yapılacaklar)-----
def tc():
    deneme = threading.Thread(target=showTime)
    deneme.start()
    a = "2"
    arayuz(a)
def name():
    deneme = threading.Thread(target=showTime)
    deneme.start()
    a = "3"
    arayuz(a)

# -----
def surname():
    deneme = threading.Thread(target=showTime)
    deneme.start()
    a = "4"
    arayuz(a)
# -----
def day():
    deneme = threading.Thread(target=showTime)
    deneme.start()
    arayuz2()
# -----
def acılıs():
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    recognizer.read('egitim/egitim.yml')

```

```

face_detector =
cv2.CascadeClassifier('Cascades/haarcascade_frontalface_default.xml')
ui6.cap = cv2.VideoCapture(0)
timer = 0
timer2 = 0
id = 0
while (True):
    ret, img = ui6.cap.read()
    img = cv2.flip(img, 1)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    gri = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    yuzler = face_detector.detectMultiScale(img, 1.1, 5, minSize=(100,100))
    for (x, y, w, h) in yuzler:
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
        id, uyum = recognizer.predict(gri[y:y + h, x:x + w])
        if (uyum < 80):
            sor = "SELECT Patient_id_tc FROM Patient WHERE rowid = ?"
            curs.execute(sor, (id,))
            data = curs.fetchall()
            c = str(data)
            b = c.strip("[ ( , )]")
            print(b)
            id = b
            n = b
            sor = "SELECT Name FROM Patient WHERE Patient_id_tc = ?"
            curs.execute(sor, (id,))
            data = curs.fetchall()
            dataname = str(data)
            name = dataname.strip("[ ( , )]")
            name
            print(name)
            uyum = f"Uyum= {round(uyum, 0)}%"
            print("uyum " + uyum + "\nid " + id)
            timer = timer + 1
            timer2 = 0
        else:
            id = "Unknown"
            name= "Register"
            n = ""
            uyum = f"Uyum= {round(uyum, 0)}%"
            print("uyum " + uyum + "\nid " + id)
            timer = 0
            timer2 = timer2 + 1
    cv2.putText(img, str(id), (x + 6, y - 6), cv2.FONT_HERSHEY_DUPLEX, .8, (0, 0,
255))
    cv2.putText(img, str(name), (x,y+h+20), cv2.FONT_HERSHEY_DUPLEX, .8, (0,
0, 255))
    height, width, channel = img.shape
    step = channel * width
    qImg = QImage(img.data, width, height, step, QImage.Format_RGB888)
    ui6.label.setPixmap(QPixmap.fromImage(qImg))

```



```

        ui5.LineTcGiris.setText(n)
        a = len(yuzler)
        if cv2.waitKey(1) & timer == 100:
            Kamera.hide()
            HastaArama.show()
            break
        elif cv2.waitKey(1) & timer2 == 100:
            Kamera.hide()
            HastaArama.show()
            break

# -----
if Kamera.isActiveWindow() == True:
    acilis()
# -----
def again():
    print("Hello Open again")
    ui4.pushButton.click()

def welcomeAcılma():
    _WelcomeRecord()
    ui.LineAd.setText("")
    ui.LineDay.setText("")
    ui.LineYear.setText("")
    ui.LineSoyad.setText("")
    ui.LineMounth.setText("")
    ui.label.clear()

if WelcomeHomePage.isActiveWindow() == True:
    print("Hello Open")
    ui4.pushButton.animateClick(100)

ui.BKayit.clicked.connect(Kayit)
ui.BKayit_2.clicked.connect(_kamera)
ui.pushButton.clicked.connect(day)
ui.pushButton_2.clicked.connect(tc)
ui.pushButton_3.clicked.connect(surname)
ui.pushButton_4.clicked.connect(name)
ui2.BtnKayit2.clicked.connect(Secim)
ui4.pushButton.clicked.connect(welcomeAcılma)
ui5.pushButton.clicked.connect(HastaSorgula)
ui5.pushButton_2.clicked.connect(SesKayıtbaslama)

sys.exit(Uygulama.exec_())

```

11.2 ACILISUI.PY

```

from PyQt5 import QtCore, QtGui, QtWidgets
class Ui_Cam(object):
    def setupUi(self, Cam):
        Cam.setObjectName("Cam")
        Cam.resize(698, 496)
        Cam.setStyleSheet("background-color: rgb(0, 0, 0);")
        self.label = QtWidgets.QLabel(Cam)
        self.label.setGeometry(QtCore.QRect(30, 30, 641, 431))
        self.label.setStyleSheet("color: rgb(255, 255, 255);")
        self.label.setFrameShape(QtWidgets.QFrame.WinPanel)
        self.label.setText("")
        self.label.setObjectName("label")
        self.retranslateUi(Cam)
        QtCore.QMetaObject.connectSlotsByName(Cam)

    def retranslateUi(self, Cam):
        _translate = QtCore.QCoreApplication.translate
        Cam.setWindowTitle(_translate("Cam", "HOSFACE"))

```

11.3 ARAMAUI.PY

```

from PyQt5 import QtCore, QtGui, QtWidgets
class Ui_HomePage(object):
    def setupUi(self, HomePage):
        HomePage.setObjectName("HomePage")
        HomePage.resize(789, 331)
        HomePage.setStyleSheet("background-color: rgb(0, 0, 0);")
        self.LTc = QtWidgets.QLabel(HomePage)
        self.LTc.setGeometry(QtCore.QRect(20, 20, 751, 291))
        self.LTc.setMinimumSize(QtCore.QSize(31, 31))
        font = QtGui.QFont()
        font.setPointSize(24)
        font.setItalic(False)
        self.LTc.setFont(font)
        self.LTc.setStyleSheet("Color: rgb(255, 255, 255)")
        self.LTc.setAlignment(QtCore.Qt.AlignCenter)
        self.LTc.setObjectName("LTc")
        self.pushButton = QtWidgets.QPushButton(HomePage)
        self.pushButton.setGeometry(QtCore.QRect(20, 260, 751, 51))
        font = QtGui.QFont()
        font.setPointSize(16)
        self.pushButton.setFont(font)
        self.pushButton.setStyleSheet("")
        self.pushButton.setText("")
        self.pushButton.setObjectName("pushButton")
        self.retranslateUi(HomePage)

```

```

QtCore.QMetaObject.connectSlotsByName(HomePage)
def retranslateUi(self, HomePage):
    _translate = QtCore.QCoreApplication.translate
    HomePage.setWindowTitle(_translate("HomePage", "HOSFACE"))
    self.LTc.setText(_translate("HomePage", "Welcome To Hospital \n"
" Say The Word \"OPEN\" To Make An Appointment."))

```

11.4 HASTAARAMAUI.PY

```

from PyQt5 import QtCore, QtGui, QtWidgets
class Ui_HastaArama(object):
    def setupUi(self, HastaArama):
        HastaArama.setObjectName("HastaArama")
        HastaArama.resize(387, 226)
        HastaArama.setStyleSheet("background-color: rgb(0, 0, 0);\n"
"alternate-background-color: rgb(0, 0, 0);")
        self.pushButton = QtWidgets.QPushButton(HastaArama)
        self.pushButton.setGeometry(QtCore.QRect(30, 160, 321, 41))
        font = QtGui.QFont()
        font.setPointSize(14)
        self.pushButton.setFont(font)
        self.pushButton.setStyleSheet("background-color: rgb(255, 255, 255);")
        self.pushButton.setObjectName("pushButton")
        self.LineTcGiris = QtWidgets.QLineEdit(HastaArama)
        self.LineTcGiris.setGeometry(QtCore.QRect(130, 100, 221, 41))
        font = QtGui.QFont()
        font.setPointSize(12)
        self.LineTcGiris.setFont(font)
        self.LineTcGiris.setLayoutDirection(QtCore.Qt.RightToLeft)
        self.LineTcGiris.setStyleSheet("background-color: rgb(255, 255, 255)")
        self.LineTcGiris.setInputMethodHints(QtCore.Qt.ImhNone)
        self.LineTcGiris.setText("")
        self.LineTcGiris.setMaxLength(11)
        self.LineTcGiris.setEchoMode(QtWidgets.QLineEdit.Normal)
        self.LineTcGiris.setCursorPosition(0)
        self.LineTcGiris.setAlignment(QtCore.Qt.AlignLeading|QtCore.Qt.AlignLeft|QtCore.
QtCore.AlignVCenter)
        self.LineTcGiris.setDragEnabled(False)
        self.LineTcGiris.setReadOnly(False)
        self.LineTcGiris.setCursorMoveStyle(QtCore.Qt.LogicalMoveStyle)
        self.LineTcGiris.setClearButtonEnabled(False)
        self.LineTcGiris.setObjectName("LineTcGiris")
        self.pushButton_2 = QtWidgets.QPushButton(HastaArama)
        self.pushButton_2.setGeometry(QtCore.QRect(30, 100, 81, 41))
        font = QtGui.QFont()
        font.setPointSize(14)
        self.pushButton_2.setFont(font)
        self.pushButton_2.setStyleSheet("alternate-background-color: rgb(255, 255,
255);\n"
"background-color: rgb(255, 255, 255);")

```

```

self.pushButton_2.setObjectName("pushButton_2")
self.progressBar = QtWidgets.QProgressBar(HastaArama)
self.progressBar.setEnabled(True)
self.progressBar.setGeometry(QtCore.QRect(30, 30, 321, 41))
font = QtGui.QFont()
font.setPointSize(14)
self.progressBar.setFont(font)
self.progressBar.setCursor(QtGui.QCursor(QtCore.Qt.ArrowCursor))
self.progressBar.setStyleSheet("background-color: rgb(255, 255, 255);\n"
"color: rgb(255, 255, 255);")
self.progressBar.setMaximum(10)
self.progressBar.setProperty("value", 0)
self.progressBar.setInvertedAppearance(False)
self.progressBar.setObjectName("progressBar")
self.retranslateUi(HastaArama)
QtCore.QMetaObject.connectSlotsByName(HastaArama)
def retranslateUi(self, HastaArama):
    _translate = QtCore.QCoreApplication.translate
    HastaArama.setWindowTitle(_translate("HastaArama", "HOSFACE"))
    self.pushButton.setText(_translate("HastaArama", "Tc Identity Check"))
    self.LineTcGiris.setPlaceholderText(_translate("HastaArama", "Only Numbers"))
    self.pushButton_2.setText(_translate("HastaArama", "Tc"))

```

11.5 İKİNCİUI.PY

```

from PyQt5 import QtCore, QtGui, QtWidgets
class Ui_Dialog(object):
    def setupUi(self, Dialog):
        Dialog.setObjectName("Dialog")
        Dialog.setEnabled(True)
        Dialog.resize(387, 191)
        font = QtGui.QFont()
        font.setFamily("MS Shell Dlg 2")
        font.setPointSize(14)
        font.setBold(False)
        font.setItalic(False)
        font.setWeight(50)
        Dialog.setFont(font)
        Dialog.setStyleSheet("font: 14pt \"MS Shell Dlg 2\";\n \"background-color: rgb(0, 0, 0);")
        self.BtnKayit2 = QtWidgets.QPushButton(Dialog)
        self.BtnKayit2.setGeometry(QtCore.QRect(70, 110, 241, 61))
        self.BtnKayit2.setStyleSheet("background-color: rgb(255, 255, 255);")
        self.BtnKayit2.setObjectName("BtnKayit2")
        self.comboBox = QtWidgets.QComboBox(Dialog)
        self.comboBox.setGeometry(QtCore.QRect(90, 30, 201, 51))
        self.comboBox.setFocusPolicy(QtCore.Qt.ClickFocus)

```

```

self.comboBox.setContextMenuPolicy(QtCore.Qt.ActionsContextMenu)
self.comboBox.setAcceptDrops(True)
self.comboBox.setToolTipDuration(-1)
self.comboBox.setLayoutDirection(QtCore.Qt.LeftToRight)
self.comboBox.setAutoFillBackground(False)
self.comboBox.setStyleSheet("background-color: rgb(255, 255, 255);\n"
"alternate-background-color: rgb(0, 170, 255);")
self.comboBox.setEditable(False)
self.comboBox.setDuplicatesEnabled(False)
self.comboBox.setObjectName("comboBox")
self.comboBox.addItem("")
self.comboBox.addItem("")
self.comboBox.addItem("")
self.comboBox.addItem("")
self.comboBox.addItem("")
self.comboBox.addItem("")
self.retranslateUi(Dialog)
QtCore.QMetaObject.connectSlotsByName(Dialog)
def retranslateUi(self, Dialog):
    _translate = QtCore.QCoreApplication.translate
    Dialog.setWindowTitle(_translate("Dialog", "HOSFACE"))
    self.BtnKayit2.setText(_translate("Dialog", "Get Sequence Number"))
    self.comboBox.setCurrentText(_translate("Dialog", "Dentist"))
    self.comboBox.setItemText(0, _translate("Dialog", "Dentist"))
    self.comboBox.setItemText(1, _translate("Dialog", "General Surgeon"))
    self.comboBox.setItemText(2, _translate("Dialog", "Dermatology"))
    self.comboBox.setItemText(3, _translate("Dialog", "Psychiatry"))
    self.comboBox.setItemText(4, _translate("Dialog", "Infectious Diseases"))

```

11.6 İLKUI.PY

```

from PyQt5 import QtCore, QtGui, QtWidgets
class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.setWindowModality(QtCore.Qt.ApplicationModal)
        MainWindow.resize(1103, 513)
        MainWindow.setCursor(QtGui.QCursor(QtCore.Qt.ArrowCursor))
        MainWindow.setMouseTracking(False)
        MainWindow.setLayoutDirection(QtCore.Qt.LeftToRight)
        MainWindow.setAutoFillBackground(False)
        MainWindow.setStyleSheet("background-color: rgb(0, 0, 0);")
        self.BKayit = QtWidgets.QPushButton(MainWindow)
        self.BKayit.setGeometry(QtCore.QRect(30, 430, 331, 41))
        font = QtGui.QFont()
        font.setPointSize(12)
        self.BKayit.setFont(font)

```

```
self.BKayit.setStyleSheet(" background-color:rgb(255, 255, 255)")
self.BKayit.setObjectName("BKayit")
self.verticalLayoutWidget = QtWidgets.QWidget(MainWindow)
self.verticalLayoutWidget.setGeometry(QtCore.QRect(10, 70, 151, 331))
self.verticalLayoutWidget.setObjectName("verticalLayoutWidget")
self.verticalLayout = QtWidgets.QVBoxLayout(self.verticalLayoutWidget)
self.verticalLayout.setContentsMargins(1, 0, 1, 3)
self.verticalLayout.setSpacing(18)
self.verticalLayout.setObjectName("verticalLayout")
self.pushButton_2 = QtWidgets.QPushButton(self.verticalLayoutWidget)
font = QtGui.QFont()
font.setPointSize(14)
self.pushButton_2.setFont(font)
self.pushButton_2.setStyleSheet("background-color: rgb(255, 255, 255);")
self.pushButton_2.setObjectName("pushButton_2")
self.verticalLayout.addWidget(self.pushButton_2)
self.pushButton_4 = QtWidgets.QPushButton(self.verticalLayoutWidget)
font = QtGui.QFont()
font.setPointSize(14)
self.pushButton_4.setFont(font)
self.pushButton_4.setStyleSheet("background-color: rgb(255, 255, 255);")
self.pushButton_4.setObjectName("pushButton_4")
self.verticalLayout.addWidget(self.pushButton_4)
self.pushButton_3 = QtWidgets.QPushButton(self.verticalLayoutWidget)
font = QtGui.QFont()
font.setPointSize(14)
self.pushButton_3.setFont(font)
self.pushButton_3.setStyleSheet("background-color: rgb(255, 255, 255);")
self.pushButton_3.setObjectName("pushButton_3")
self.verticalLayout.addWidget(self.pushButton_3)
self.pushButton = QtWidgets.QPushButton(self.verticalLayoutWidget)
font = QtGui.QFont()
font.setPointSize(13)
self.pushButton.setFont(font)
self.pushButton.setStyleSheet("background-color: rgb(255, 255, 255);")
self.pushButton.setObjectName("pushButton")
self.verticalLayout.addWidget(self.pushButton)
self.horizontalLayoutWidget = QtWidgets.QWidget(MainWindow)
self.horizontalLayoutWidget.setGeometry(QtCore.QRect(180, 330, 221, 41))
self.horizontalLayoutWidget.setObjectName("horizontalLayoutWidget")
self.horizontalLayout = QtWidgets.QHBoxLayout(self.horizontalLayoutWidget)
self.horizontalLayout.setContentsMargins(0, 0, 0, 0)
self.horizontalLayout.setObjectName("horizontalLayout")
self.LineDay = QtWidgets.QLineEdit(self.horizontalLayoutWidget)
self.LineDay.setEnabled(True)
self.LineDay.setBaseSize(QtCore.QSize(0, 0))
font = QtGui.QFont()
font.setPointSize(12)
self.LineDay.setFont(font)
self.LineDay.setToolTip("")
```

```

self.LineDay.setToolTipDuration(-1)
self.LineDay.setStatusTip("")
self.LineDay.setWhatsThis("")
self.LineDay.setStyleSheet(" background-color:rgb(255, 255, 255)")
self.LineDay.setInputMethodHints(QtCore.Qt.ImhFormattedNumbersOnly)
self.LineDay.setInputMask("")
self.LineDay.setText("")
self.LineDay.setMaxLength(2)
self.LineDay.setAlignment(QtCore.Qt.AlignLeading|QtCore.Qt.AlignLeft|QtCore.Qt.AlignVCenter)
self.LineDay.setCursorMoveStyle(QtCore.Qt.LogicalMoveStyle)
self.LineDay.setClearButtonEnabled(False)
self.LineDay.setObjectName("LineDay")
self.horizontalLayout.addWidget(self.LineDay)
self.LineMounth = QtWidgets.QLineEdit(self.horizontalLayoutWidget)
font = QtGui.QFont()
font.setPointSize(12)
self.LineMounth.setFont(font)
self.LineMounth.setStyleSheet(" background-color:rgb(255, 255, 255)")
self.LineMounth.setInputMethodHints(QtCore.Qt.ImhPreferNumbers)
self.LineMounth.setMaxLength(2)
self.LineMounth.setAlignment(QtCore.Qt.AlignLeading|QtCore.Qt.AlignLeft|QtCore.Qt.AlignVCenter)
self.LineMounth.setClearButtonEnabled(False)
self.LineMounth.setObjectName("LineMounth")
self.horizontalLayout.addWidget(self.LineMounth)
self.LineYear = QtWidgets.QLineEdit(self.horizontalLayoutWidget)
font = QtGui.QFont()
font.setPointSize(12)
self.LineYear.setFont(font)
self.LineYear.setStyleSheet(" background-color:rgb(255, 255, 255)")
self.LineYear.setInputMethodHints(QtCore.Qt.ImhPreferNumbers)
self.LineYear.setMaxLength(4)
self.LineYear.setAlignment(QtCore.Qt.AlignLeading|QtCore.Qt.AlignLeft|QtCore.Qt.AlignVCenter)
self.LineYear.setClearButtonEnabled(False)
self.LineYear.setObjectName("LineYear")
self.horizontalLayout.addWidget(self.LineYear)
self.progressBar = QtWidgets.QProgressBar(MainWindow)
self.progressBar.setGeometry(QtCore.QRect(20, 20, 361, 31))
font = QtGui.QFont()
font.setPointSize(14)
self.progressBar.setFont(font)
self.progressBar.setCursor(QtGui.QCursor(QtCore.Qt.ArrowCursor))
self.progressBar.setStyleSheet("background-color: rgb(255, 255, 255);\n"
"color: rgb(255, 255, 255);")
self.progressBar.setMaximum(10)
self.progressBar.setProperty("value", 0)
self.progressBar.setInvertedAppearance(False)
self.progressBar.setObjectName("progressBar")

```

```

self.BKayit_2 = QtWidgets.QPushButton(MainWindow)
self.BKayit_2.setGeometry(QtCore.QRect(450, 10, 591, 31))
font = QtGui.QFont()
font.setPointSize(14)
self.BKayit_2.setFont(font)
self.BKayit_2.setStyleSheet(" background-color:rgb(255, 255, 255)")
self.BKayit_2.setObjectName("BKayit_2")
self.LineTc = QtWidgets.QLineEdit(MainWindow)
self.LineTc.setEnabled(True)
self.LineTc.setGeometry(QtCore.QRect(180, 100, 219, 31))
font = QtGui.QFont()
font.setPointSize(12)
self.LineTc.setFont(font)
self.LineTc.setStyleSheet(" background-color:rgb(255, 255, 255)")
self.LineTc.setInputMethodHints(QtCore.Qt.ImhPreferNumbers)
self.LineTc.setText("")
self.LineTc.setMaxLength(11)
self.LineTc.setEchoMode(QtWidgets.QLineEdit.Normal)
self.LineTc.setCursorPosition(0)
self.LineTc.setAlignment(QtCore.Qt.AlignLeading|QtCore.Qt.
AlignLeft|QtCore.Qt.AlignVCenter)
self.LineTc.setDragEnabled(False)
self.LineTc.setReadOnly(False)
self.LineTc.setCursorMoveStyle(QtCore.Qt.LogicalMoveStyle)
self.LineTc.setClearButtonEnabled(False)
self.LineTc.setObjectName("LineTc")
self.LineAd = QtWidgets.QLineEdit(MainWindow)
self.LineAd.setGeometry(QtCore.QRect(180, 180, 219, 31))
font = QtGui.QFont()
font.setPointSize(12)
self.LineAd.setFont(font)
self.LineAd.setStyleSheet(" background-color:rgb(255, 255, 255)")
self.LineAd.setAlignment(QtCore.Qt.AlignLeading|QtCore.Qt.AlignLeft|QtCore.Qt.Align
VCeTter)
self.LineAd.setClearButtonEnabled(False)
self.LineAd.setObjectName("LineAd")
self.LineSoyad = QtWidgets.QLineEdit(MainWindow)
self.LineSoyad.setGeometry(QtCore.QRect(180, 260, 219, 31))
font = QtGui.QFont()
font.setPointSize(12)
self.LineSoyad.setFont(font)
self.LineSoyad.setStyleSheet(" background-color:rgb(255, 255, 255)")
self.LineSoyad.setAlignment(QtCore.Qt.AlignLeading|QtCore.Qt.AlignLeft|QtCore.Qt.Al
ignVCenter)
self.LineSoyad.setClearButtonEnabled(False)
self.LineSoyad.setObjectName("LineSoyad")
self.label = QtWidgets.QLabel(MainWindow)
self.label.setGeometry(QtCore.QRect(440, 60, 219, 431))
self.label.setStyleSheet("color: rgb(255, 255, 255);")
self.label.setFrameShape(QtWidgets.QFrame.WinPanel)

```



```

self.label.setText("")
self.label.setObjectName("label")
self.retranslateUi(MainWindow)
QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "HOSFACE"))
    self.BKayit.setText(_translate("MainWindow", "Patient Registration"))
    self.pushButton_2.setText(_translate("MainWindow", "TC"))
    self.pushButton_4.setText(_translate("MainWindow", "NAME"))
    self.pushButton_3.setText(_translate("MainWindow", "SURNAME"))
    self.pushButton.setText(_translate("MainWindow", "DATE OF BIRTH"))
    self.LineDay.setPlaceholderText(_translate("MainWindow", "Day-gg"))
    self.LineMounth.setPlaceholderText(_translate("MainWindow", "Mount-aa"))
    self.LineYear.setPlaceholderText(_translate("MainWindow", "Year-Y"))
    self.BKayit_2.setText(_translate("MainWindow", "Camera"))
    self.LineTc.setPlaceholderText(_translate("MainWindow", "Only Numbers"))
    self.LineAd.setPlaceholderText(_translate("MainWindow", "Your Name"))
    self.LineSoyad.setPlaceholderText(_translate("MainWindow", "Your Surname"))

```

11.7 UCUI.PY

```

from PyQt5 import QtCore, QtGui, QtWidgets
class Ui_Form(object):
    def setupUi(self, Form):
        Form.setObjectName("Form")
        Form.resize(966, 540)
        Form.setStyleSheet("background-color: rgb(0, 0, 0);")
        Form.setInputMethodHints(QtCore.Qt.ImhNone)
        self.tableWidget = QtWidgets.QTableWidget(Form)
        self.tableWidget.setGeometry(QtCore.QRect(10, 130, 681, 391))
        self.tableWidget.setMinimumSize(QtCore.QSize(561, 371))
        font = QtGui.QFont()
        font.setPointSize(9)
        self.tableWidget.setFont(font)
        self.tableWidget.setStyleSheet("background-color: rgb(255, 255, 255);\n"
"alternate-background-color: rgb(255, 255, 255);")
        self.tableWidget setFrameShape(QtWidgets.QFrame.StyledPanel)
        self.tableWidget setFrameShadow(QtWidgets.QFrame.Sunken)
        self.tableWidget.setLineWidth(1)
        self.tableWidget.setMidLineWidth(1)
        self.tableWidget.setVerticalScrollBarPolicy(QtCore.Qt.ScrollBarAsNeeded)
        self.tableWidget.setSizeAdjustPolicy(QtWidgets.QAbstractScrollArea.AdjustIgnored)
        self.tableWidget.setAutoScrollMargin(10)
        self.tableWidget.setEditTriggers(QtWidgets.QAbstractItemView.NoEditTriggers)

```

```

self.tableWidget.setTabKeyNavigation(True)
self.tableWidget.setProperty("showDropIndicator", True)
self.tableWidget.setDragEnabled(False)
self.tableWidget.setDragDropMode(QtWidgets.QAbstractItemView.NoDragDrop)
self.tableWidget.setAlternatingRowColors(True)
self.tableWidget.setHorizontalScrollMode(QtWidgets.QAbstractItemView.ScrollPerPixel)
    self.tableWidget.setGridStyle(QtCore.Qt.CustomDashLine)
    self.tableWidget.setWordWrap(True)
    self.tableWidget.setCornerButtonEnabled(True)
    self.tableWidget.setRowCount(100)
    self.tableWidget.setColumnCount(8)
    self.tableWidget.setObjectName("tableWidget")
    item = QtWidgets.QTableWidgetItem()
    self.tableWidget.setHorizontalHeaderItem(0, item)
    item = QtWidgets.QTableWidgetItem()
    self.tableWidget.setHorizontalHeaderItem(1, item)
    item = QtWidgets.QTableWidgetItem()
    self.tableWidget.setHorizontalHeaderItem(2, item)
    item = QtWidgets.QTableWidgetItem()
    self.tableWidget.setHorizontalHeaderItem(3, item)
    item = QtWidgets.QTableWidgetItem()
    self.tableWidget.setHorizontalHeaderItem(4, item)
    item = QtWidgets.QTableWidgetItem()
    self.tableWidget.setHorizontalHeaderItem(5, item)
    item = QtWidgets.QTableWidgetItem()
    self.tableWidget.setHorizontalHeaderItem(6, item)
    item = QtWidgets.QTableWidgetItem()
    self.tableWidget.setHorizontalHeaderItem(7, item)
    self.tableWidget.horizontalHeader().setCascadingSectionResizes(False)
    self.label_2 = QtWidgets.QLabel(Form)
    self.label_2.setGeometry(QtCore.QRect(110, 30, 251, 81))
    font = QtGui.QFont()
    font.setPointSize(20)
    self.label_2.setFont(font)
    self.label_2.setStyleSheet("color: rgb(255, 255, 255);")
    self.label_2.setObjectName("label_2")
    self.lineEdit = QtWidgets.QLineEdit(Form)
    self.lineEdit.setGeometry(QtCore.QRect(370, 40, 171, 51))
    font = QtGui.QFont()
    font.setPointSize(14)
    self.lineEdit.setFont(font)
    self.lineEdit.setStyleSheet("background-color: rgb(255, 255, 255);")
    self.lineEdit.setObjectName("lineEdit")
    self.tableWidget_2 = QtWidgets.QTableWidget(Form)

```

```

self.tableWidget_2.setGeometry(QRect(700, 20, 251, 501))
self.tableWidget_2.setStyleSheet("background-color: rgb(255, 255, 255);\n"
"alternate-background-color: rgb(255, 255, 255);")
self.tableWidget_2.setMidLineWidth(1)
self.tableWidget_2.setAlternatingRowColors(True)
self.tableWidget_2.setGridStyle(QtCore.Qt.CustomDashLine)
self.tableWidget_2.setRowCount(100)
self.tableWidget_2.setObjectName("tableWidget_2")
self.tableWidget_2.setColumnCount(2)
item = QtWidgets.QTableWidgetItem()
self.tableWidget_2.setHorizontalHeaderItem(0, item)
item = QtWidgets.QTableWidgetItem()
self.tableWidget_2.setHorizontalHeaderItem(1, item)
self.tableWidget_2.horizontalHeader().setSortIndicatorShown(True)
self.progressBar = QtWidgets.QProgressBar(Form)
self.progressBar.setGeometry(QRect(550, 40, 131, 51))
self.progressBar.setStyleSheet("color: rgb(255, 255, 255);")
self.progressBar.setMaximum(10)
self.progressBar.setProperty("value", -1)
self.progressBar.setTextVisible(True)
self.progressBar.setInvertedAppearance(False)
self.progressBar.setObjectName("progressBar")
self.retranslateUi(Form)
QtCore.QMetaObject.connectSlotsByName(Form)
def retranslateUi(self, Form):
    _translate = QtCore.QCoreApplication.translate
    Form.setWindowTitle(_translate("Form", "HOSFACE"))
    self.tableWidget.setSortingEnabled(True)
    item = self.tableWidget.horizontalHeaderItem(0)
    item.setText(_translate("Form", "Tc"))
    item = self.tableWidget.horizontalHeaderItem(1)
    item.setText(_translate("Form", "New Column"))
    item = self.tableWidget.horizontalHeaderItem(2)
    item.setText(_translate("Form", "Name"))
    item = self.tableWidget.horizontalHeaderItem(3)
    item.setText(_translate("Form", "Surname"))
    item = self.tableWidget.horizontalHeaderItem(4)
    item.setText(_translate("Form", "Polinilic Name"))
    item = self.tableWidget.horizontalHeaderItem(5)
    item.setText(_translate("Form", "Doctor Name"))
    item = self.tableWidget.horizontalHeaderItem(6)
    item.setText(_translate("Form", "Date"))
    item = self.tableWidget.horizontalHeaderItem(7)
    item.setText(_translate("Form", "Time"))

```

```
self.label_2.setText(_translate("Form", "Your Queue Number"))
self.tableWidget_2.setSortingEnabled(True)
item = self.tableWidget_2.horizontalHeaderItem(0)
item.setText(_translate("Form", "Name"))
item = self.tableWidget_2.horizontalHeaderItem(1)
item.setText(_translate("Form", "Surname"))
```