KINGDOM OF SAUDI ARABIA
Ministry of Education
Taibah University
College of Computer Science and Engineering
(Male Section)

المملكة العربية السعودية
وزارة التعليم
جامعة طيبة
كلية علوم و هندسة الحاسب الآلي
(قسم الطلاب)

A Self-Triage and Management System for Emergency Departments

Graduation Project (2)

by

Nawaf Dhaifallah Alharbi 4001781

Thamer Sultan Alrehily 3903334

Nawaf Muawwad Alharbi 4001109

Mohannad Omar Alahmadi 4028019

A project submitted in partial fulfilment of the requirements for the degree of Bachelor of Science (Computer Science)

Supervised by

Dr. Hosam Jaber S Aljihani

1st Semester - Academic Year 1445 (2023/2024)

Abstract

In this project we are aiming, emergency departments in providing efficient and timely patient care. The proposal introduces a Self-triage system that utilizes expert systems to analyze patient-provided information, including vital signs, to identify core symptoms faster and more accurately. The system can quickly refer patients who require immediate care to the appropriate departments, thus saving valuable time and resources. Furthermore, it can provide guidance to patients who do not need immediate care, reducing the burden on healthcare staff and optimizing their time utilization. Additionally, the system can offer simple prescriptions when necessary, alleviating pressure on doctors and ensuring basic medical care for patients. This project will discuss the technical aspects, functionality, and benefits of the proposed system, as well as explore the challenges, limitations, and potential future developments. By improving emergency patient care, this project our potential to make a significant impact on healthcare systems.

Keywords Vital signs; Medical service enhancement; Emergency department; Self-Triage

Acknowledgement

We would like to express our sincere gratitude to our fellow students and academic supervisor, who have provided invaluable guidance and support throughout the process. Their assistance has been instrumental in shaping the development and outcome of this endeavour.

we would like to extend our appreciation to our academic supervisor, Dr. Hosam Jaber S Aljihani, whose expertise, knowledge, and insightful feedback have been immensely valuable. Their guidance and encouragement have consistently motivated us as a group to push the boundaries of our research and strive for excellence.

We would also like to acknowledge the contributions of our fellow students at Computer Science and Engineering at Taibah University. Their engagement and collaboration have fostered a stimulating academic environment, allowing for fruitful discussions, exchange of ideas, and constructive feedback. The synergy created within our group has significantly enhanced the quality and depth of our work.

In conclusion, we are truly grateful for the support, guidance, and assistance received from our academic supervisor.

Table of Content

List of Figures

List of Tables

List of Abbreviations

ED	Emergency department

Chapter 1: Introduction

1.1 Introduction

Health care is a right for every patient, and the importance of health care lies in providing all the medical services that the patient needs, because this is considered a right for him and a human duty.

In today's fast-paced world, healthcare systems are constantly looking for ways to improve patient care, especially during emergencies. There are challenges facing emergency departments, one of these challenges is to identifying patients who need immediate care and referring them to the appropriate department also overcrowding of non-critical cases in the department and writing symptoms and vital signs manually, which delays the speed of work, all of this delays the process of triage patients and leads to increased pressure on the medical staff and the patient's lack of entitlement to his services.

To overcome this challenge, we propose a Self-triage system to assist patients incoming to emergency departments, The proposal system uses expert systems to analyze information provided by patients, including vital signs, to identify core symptoms faster and more accurately. By doing so, he can quickly refer patients who need immediate care to the appropriate departments, saving valuable time and resources. Moreover, the system can provide advice and guidance to patients who do not need immediate care, reducing the burden on healthcare staff and making more efficient use of their time.

In addition to the above, the system can also provide a simple prescription when needed, which can help relieve pressure on doctors and provide basic medical care for patients. This innovation in healthcare technology not only improves patient care, but also improves utilization of healthcare resources.

This project will discuss the technical system being used to assist patients in triage, how it works and the benefits it provides. It will also explore the challenges and limitations of using such a system, as well as potential future developments. Overall, this project has great potential to improve emergency patient care, and its impact on healthcare systems is worth exploring.

1.2 Problem Definition

Overcrowding in the emergency department (ED) during the triage process has wide-ranging effects that go beyond patient care, resource utilization, and efficiency. This complex issue impacts various aspects of healthcare delivery and requires comprehensive strategies to tackle its multifaceted challenges.

One of the main factors contributing to overcrowding in the ED triage process is the overall healthcare landscape. In many areas, the population has grown faster than healthcare services have expanded, leading to increased demand for emergency care. Additionally, seasonal outbreaks, such as flu, can strain EDs significantly, overwhelming the triage system with a sudden influx of patients.

Limited access to primary care also worsens overcrowding. When people lack adequate access to primary healthcare services or face barriers to timely care, they may turn to the ED for non-urgent conditions. These further burdens the triage process, as patients with conditions that could be effectively managed in a primary care setting end up using valuable ED resources.

Inadequate staffing and resource constraints present additional challenges during triage. EDs often struggle with shortages of trained triage nurses and physicians, leading to longer wait times for patients to be evaluated. Insufficient numbers of healthcare professionals can increase stress and burnout among the remaining staff, compromising their ability to provide high-quality care. Additionally, the availability of triage rooms, equipment, and supplies may not match the demand, creating bottlenecks and impeding the efficient flow of patients through the triage area.

The consequences of overcrowding during triage are significant and impact patients, healthcare providers, and the healthcare system. Delayed care for patients with urgent or life-threatening conditions increases the risk of adverse outcomes and can contribute to higher rates of morbidity and mortality. The inability to promptly identify and prioritize critical cases compromises the timely initiation of life-saving interventions.

Patients with non-urgent conditions, although not facing immediate life-threatening risks, may feel frustrated and dissatisfied due to long wait times. This can erode their trust in the healthcare system and negatively impact their overall patient experience. Disgruntled patients may seek alternative avenues for care or inappropriately utilize emergency services, further straining the already overwhelmed ED resources.

The overcrowding of the triage area also presents significant challenges for healthcare providers. Triage nurses and physicians are responsible for making quick and accurate assessments, but the high patient volumes can increase stress and cognitive load. These pressures may compromise the accuracy of their evaluations, potentially resulting in misdiagnoses or delayed interventions. Effective communication and collaboration among the triage team may also suffer due to overcrowding, increasing the likelihood of errors or misunderstandings.

1.3 Project Objectives

To develop a technical system that can analyze information provided by patients to identify basic symptoms faster and more accurately for quicker referral of patients in need of immediate care to appropriate departments.

To provide advice and guidance to patients who do not require immediate care, which will help to relieve pressure on doctors and provide basic medical care to patients.

To improve the accuracy and speed of triage assessments and reduce wait times for patients in emergency departments.

To enhance patient safety by ensuring that patients receive appropriate care in a timely manner.

To reduce healthcare costs associated with inappropriate use of emergency services and unnecessary hospital admissions.

To increase patient satisfaction by providing personalized care and attention, and reducing the time spent waiting for treatment.

To provide healthcare providers with a user-friendly system that is easy to navigate and can quickly and accurately determine the severity of a patient's condition.

To develop a system that is adaptable to different healthcare settings and can be customized to fit the specific needs of different hospitals and clinics.

To collaborate with healthcare providers and experts in the field to ensure that the system is evidence-based and grounded in best practices.

To evaluate the effectiveness of the system through rigorous testing and assessment, and continuously improve and update the system based on user feedback and changing healthcare needs.

To promote the use of technology in healthcare and foster a culture of innovation and continuous improvement in emergency departments.

To provide a control on emergency departments.

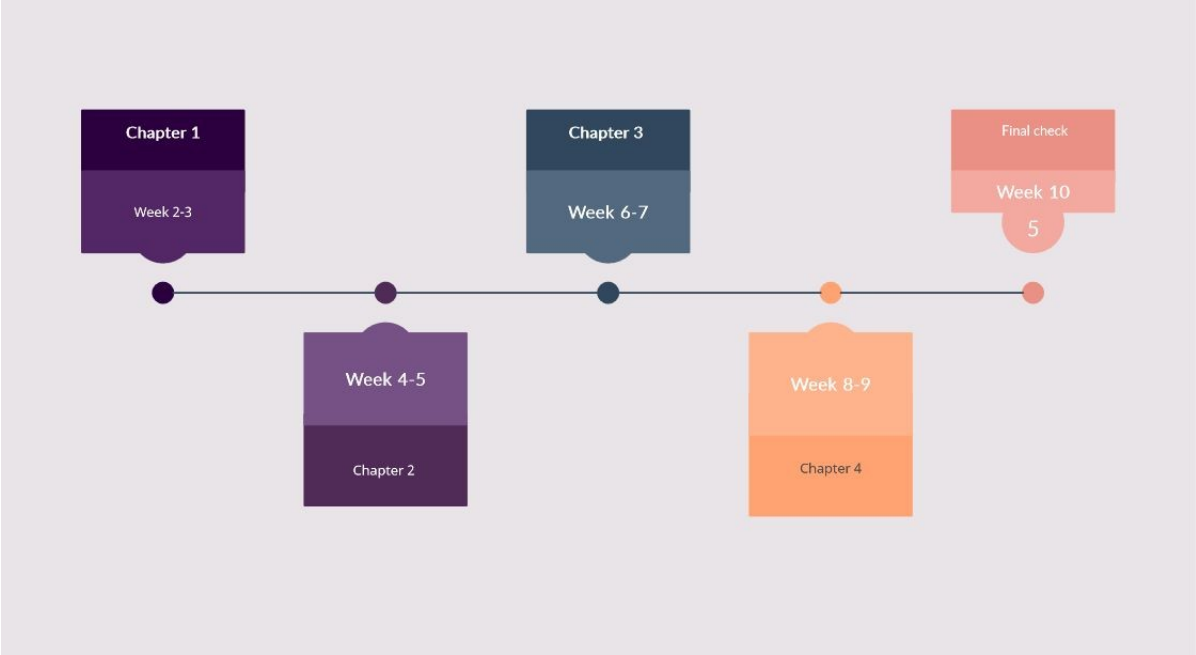
1.4 Project Scope

The proposed system is to be used in any emergency department. The target users of our system are the **patients**. This project is mainly emphasized on develop system is for Sorting the patient by vital signs and A series of procedures and algorithms are used by the system to aid healthcare practitioners in assessing and prioritizing patients based on their symptoms and clinical indicators.

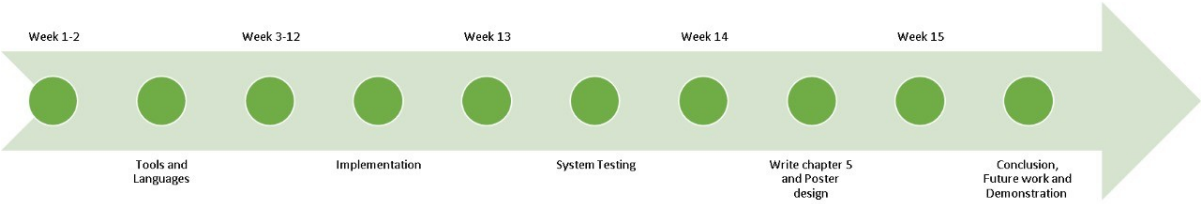
Which helps the patient in directing him while going to the hospital without the need for assistance from the health staff, and thus we have solved one of the waiting problems in the emergency.

1.5 Project Timeline

Third semester (1444):



First semester (1445):



1.6 Document Organization

This project consists of six chapters, organized as follows:

Chapter 1 introduces the project objectives, the Problem Definition of the project, the approach used in this project, the scope of the work, and project layout.

Chapter 2 provides the reader with an overview of the literature review.

Chapter 3 overview of system analysis and description and explaining methodology.

Chapter 4 overview of system design in many ways such as architectural and Object-Oriented Designs. And UI design.

Chapter 5 overview of system implementation and testing.

Chapter 6 overview of system future work and limitations.

Chapter 2: Literature Review

2.1 Introduction

Hospitals are a critical element in emergency management, and effective patient triage is necessary to ensure appropriate care in a timely manner. A Self-Triage system that can assist hospitals in evaluating triage processes and identifying areas for improvement can be an effective tool in improving emergency triage systems, so that cases can be quickly prioritized and the burden on medical staff can be reduced. This literature review aims to explore existing literature on self-evaluation systems for emergency triage in hospitals and the use of expert systems in triage.

2.2 Background

Emergency services are essential components of healthcare facilities as they allow for rapid care of patients with critical illnesses or urgent health concerns that require immediate attention. With the rapid development of technology, emergency services can now be further improved to ensure faster triage and more accurate diagnoses by using expert systems that analyze vital signs that are read by devices connected to the system for rapid and accurate assessment of emergency cases.

2.2.1 Vital signs devices overview

- **Digital sphygmomanometer:**
  are automated, consists of an electronic sensor to provide blood pressure reading without needing someone to operate the cuff or listen to the blood flow sounds.

Figure 2-2: a Digital sphygmomanometer.

- **Human Temperature Sensor (**MAX30205**):**

The MAX30205 temperature sensor offers a body temperature measurement solution. As a feature of the MAXREFDES100# Health Sensor Platform, it is easy to evaluate temperature sensing as part of a complete patient assessment system. We can collect temperature data using the Health Sensor Platform.

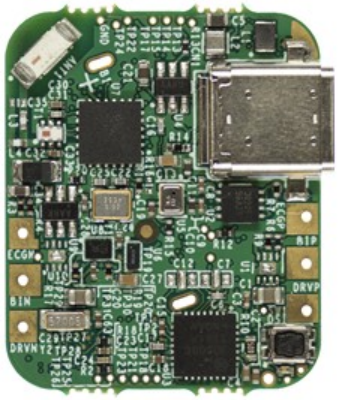


Figure 2-3: MAX30205 sensor

- **Fingertip Pulse Oximeter:**
  A pulse oximeter is a small, lightweight device used to monitor the amount of oxygen carried in the body. This noninvasive tool attaches painlessly to your fingertip, sending two wavelengths of light through the finger to measure your pulse rate and how much oxygen is in your system.

Figure 2-4: a Fingertip Pulse Oximeter.

2.2.2 Emergency Department (ED)

Emergency services are an essential component of healthcare facilities because they allow rapid care of patients with serious illnesses or urgent health concerns that require immediate attention.

When the patient enters hospitals, especially in emergency cases, the patient initially goes to the reception to collect his personal data. After that, the patient's vital signs are taken, in addition to a visual diagnosis. Based on this, his health condition is classified, and a specific color is chosen to symbolize the seriousness of the condition. The process of sorting patients into a color group according to the international classification in cases of disasters and major

accidents aims to facilitate the work of medical teams with medical cases that reach the triage area from the scene with ease. The red color symbolizes that the patient's condition is very critical, and he cannot bear any time to wait, and he needs immediate intervention without delay by the attending physician. As for the orange color, it is lighter than the previous color, but the high risk is present and indicates that the patient's condition is very urgent and can only bear 10 minutes of waiting. As for the yellow color, it symbolizes that the patient's condition is also urgent, but it bears a waiting period of 60 minutes. Green indicates that the patient's health condition is normal, and he can wait up to 120 minutes. The blue color symbolizes that the patient's health is fine, that his condition is not urgent, and that he can bear to wait. Time 240 minutes.

2.2.3 Medical triage protocols

- MTS (Manchester Triage System)

system that aims to address the issue of overcrowding in emergency departments is the Manchester Triage System (MTS).

MTS is a triage system that was developed in the late 1990s and has been widely used in emergency departments around the world.

MTS is a five-level triage system that categorizes patients based on the severity of their condition, ranging from category 1 (most urgent) to category 5 (least urgent). The system uses a set of protocols and algorithms to guide healthcare professionals in assessing and prioritizing patients based on their symptoms and clinical signs.

Our system Slightly different from the Manchester Triage System (MTS) in that to be a specific technology-based solution devised to automate triage and administer basic medical care, while the MTS is a comprehensive triage algorithm and a **manual** system utilized by healthcare Professionals to categorize patients based on the symptoms they display and their need for necessary care. The MTS considers various vital signs and clinical indications to determine the appropriate level of care required and the relevant department to which the patient should be referred.

- The Canadian Triage and Acuity Scale (CTAS)

The Canadian Triage and Acuity Scale (CTAS) is a widely used triage system in Canada, designed to help clinicians prioritize patients based on the urgency of their medical needs. The system was developed in response to a need for a standardized approach to triage that could be used across different healthcare settings and regions.

The CTAS system is based on a five-level scale, with each level corresponding to a different level of urgency and clinical complexity. Level 1 represents the most urgent cases, such as patients with cardiac arrest or severe trauma, while level 5 represents cases that are less urgent, such as patients with minor injuries or illnesses.

The CTAS system is designed to be flexible and adaptable to different clinical settings and patient populations. It considers a range of factors, including the patient's vital signs, symptoms, and medical history. It also considers the resources available in the healthcare setting, such as staffing levels and equipment.

- The Rapid Emergency Triage and Treatment System (RETTS):

The Rapid Emergency Triage and Treatment System (RETTS) is a medical triage system that is used to prioritize patient care in emergency departments. It was developed in Sweden in the late 1990s as a response to the need for a standardized and efficient approach to emergency triage. RETTS is now used in many countries around the world, including the United States, Canada, and several European countries.

The RETTS system uses a color-coded system to prioritize patients based on the severity of their condition. The system is designed to be easy to use and can be quickly implemented in any emergency department. It is based on a set of standardized protocols that guide clinicians in assessing a patient's symptoms and determining the appropriate level of care,

The RETTS system uses five different colors to categorize patients.

2.3 Related Work

2.3.1 Review of Relevant Work

at this moment, there are some hospitals that are having a huge problem with overcrowding patients in their emergency departments. This situation can lead to delays in providing urgent medical care and diagnosis to patients who require immediate medical attention. In such cases, it becomes necessary to quickly assess the severity of the patient's condition and determine the level of risk to which he or she is exposed. This information is vital in guiding the necessary actions and treatment plans to effectively manage a patient's condition. However, with overcrowding, it becomes difficult to prioritize patients based on the level of urgency, which can lead to patients waiting for long periods of time before receiving necessary medical care. Therefore, it is necessary to address the problem of overcrowding in emergency departments to ensure that patients receive timely and effective medical care. And after researching the topic, it turned out that there are some systems that went to solve the problem of congestion in emergency state and Developing emergency aid, such as:

- **ED system of King Khaled eye specialist hospital (KKESH):**

The Emergency Room in KKESH provides comprehensive care for acute ocular conditions that require prompt assessment and intervention as follows:

- ✓ KKESH Emergency staff is honored to provide urgent care to patients suffering from ocular emergencies.
- ✓ After registration, the patient must remain in the waiting area until he gets screen call to the examination rooms.
- ✓ Doctors and nurses are very keen to serve you as quickly as possible, noted that the waiting time in the Emergency Department cannot be specified.
- ✓ The treatment service is provided for cases classified medically as emergency cases only.
- ✓ Classification of case severity and its need for immediate intervention depends on the doctor's decision based on the initial assessment of the case. Also, it does not depend on the patient's age, his belief that his condition is considered an emergency, or the time of his registration. The priority is always for the most serious cases.
- ✓ The patient is responsible for his presence in the Emergency Department. If he leaves and his condition is getting worse, he is responsible for his behavior as he does not wait.
- ✓ Medical examinations, blood tests and x-rays results may take up to three hours.

- ✓ Cases acceptance for treatment at KKESH is through Medical Coordination and Eligibility Department only.

- **ED system of King Faisal specialist hospital & research center (KFSH&RC):**

Emergency Medicine is the medical specialty dedicated to the diagnosis and treatment of unforeseen illness or injury. It includes the initial evaluation, diagnosis, treatment, and disposition of any patient requiring expeditious medical, surgical, or psychiatric care.

As part of KFSH (Gen. Org.), the Department of Emergency Medicine's mission is to provide comprehensive, high quality, efficient, technically advanced "state of the art" emergency care to all patients.

This is achieved through the following:

- Integration of timely, pre-hospital care, rapid triage, expeditious resuscitation and stabilization, and appropriate disposition of all patients.
- The provision of supportive and consultation services to patients referred from other departments of KFSH as well as for patients referred from other hospitals.
- Ongoing improvement in patient management, communication, and health care delivery.
- Research in Emergency Medicine.
- The establishment and enhancement of specialized courses dedicated to continuing professional education for staff in the field of emergency medicine, as well as developing clinical pathways for different emergencies.
- Life support courses and continuous medical education for the medical staff at KFSH&RC and other medical institutions.

Scope of Services:

- The Department of Emergency Medicine is providing care for Acute, Cardiac, Respiratory, Neurological, Infectious and Gynecological Emergencies in tertiary care setting requiring immediate attention in both Adult and Pediatric age groups.
- DEM has a fast-track area for rapid turnover and for low acuity patients, core area equipped for Cardiovascular Monitoring and Resuscitation & a twenty-four (24 H) observation area for the patients requiring short stay or intermediate care.

- **Imam Abdul Rahman bin Faisal University-university hospital ED system (KAFU):**

Emergency Department is an academic clinical department accessible to patients of all age groups and genders who will be provided with the best quality of care and mostly will be treated at the ED and discharged with the proper instructions. If the patient's condition requires admission patient will be referred to the consulting service according to patient condition for further advanced management. The department functions 24 hours, 7 days a week.

ED staffs are continuously dedicated in improving the patient outcomes that are acutely ill and injured, through high quality, evidence-based practice, cost-effective and efficient care. ED embraces the quality improvement and patient safety processes then moving forward in raising the BAR of the standard in patient care through continuous education and research. ED staff working closely together in collaboration with other services to maintain the excellent service provided to patient and family.

In line with the KFHU mission the following are the ED goals and objectives towards attaining a center of excellence for the provision of emergency care:

- Level of Care based on Canadian Triage
- Promotion of Emergency Care Standards
- Quality Improvement and Patient Safety
- Educational Programs
- Research

Emergency Department provides resuscitation & stabilization of all critically ill and critically injured patients, handling all types of emergencies with threat to life, organs, limbs, or sight in all age groups. All emergency invasive and non-invasive procedures within the privilege of emergency medicine practice. The department is also equipped with resuscitation life-saving equipment and proper medications.

The department is run by proper staff members ranging from consultants with international board certification, to specialists and senior registrars and residents. Healthcare providers also include EMT's and paramedics, nursing staff and respiratory therapists, coordinators, and administrators.

2.3.2 Relationship Between the Relevant Work and Our Own Work

After we declared some related work to our proposal, now we will explain the basic Comparisons and how those systems are related and how our proposal is superior in many characteristics.

- **Auto ID verification**: refer to automated verification for patient identity by matching patient fingerprint with his information in hospital database or sending verification message to the patient phone number related to the patient ID in hospital database.

- **Self-reading of vital signs** is the patient's ability to measure his vital signs by devices connected on the system like a MAX30205 temperature sensor, pulse oximeter and digital sphygmomanometer.

- **Self-assessment**: is the patient's ability to assist himself without intervention of a medical practitioner.

- **Queue system:** existence of a queue system to arrange patients According to severity of patient.

- **Accept all cases**: The department accepts and serves all patients with different health conditions including critical conditions or non-critical.

- **Electronic Data recording**: the ability to Record patient information and cases information including symptoms and medical triage result electronically.

- **Auto prescription:** the ability to provide a prescription to the patient after Medical Examination.

Characteristic	Auto ID verification	self-reading of vital signs	Self-assessment	Queue system	Accept all cases	Electronic Data recording	auto Prescription
KKESH				✓		✓	
KFSH					✓	✓	
KAFU				✓	✓	✓	
Our proposal	✓	✓	✓	✓	✓	✓	✓

Table 2-1:compersion between systems characteristics.

2.4 Summary

in this chapter, we introduced our system and declare all resources we expect to use in the system implementation, and we discussed some relevant work and the points where our system is distinguished and characterized.

Chapter 3: System Analysis

3.1 Introduction

In this chapter we will analyze Self-Trige and Management System and we will describe the system and we will declare the basic functions in the system and what the system should do and how will react.

3.2 Requirements Elicitation

Self-Triage system is designed for both patients and emergency management department, from patient view the system should ask the patient for fingerprint or ID Number to identify the patient identity, after that the system will let the patient choose his symptoms from the symptoms list in the interface, and after asking the patient the device will read the vital signs that comes from the patient through the vital signs devices that connected to the self-triage devices and determine the state of the patient and assign him to one of three severity levels high-severity , medium- severity and low-severity, and in each level self-triage device will behave as following :

High severity: direct the patient to immediate care in high area.

medium severity: assign the patient in queue system in medium area.

Low severity: assign the patient in queue system in low area.

After determining severity, self-triage device will send patient information and his triage result to the hospital's database.

from emergency admin view, the ED Admin can view Statistical list about ED department including the number of cases for each severity level, total cases, number of clear and unclear cases, fullness percentage of each area and number of active and inactive doctor for each area.

Also, the admin can control the number of clinics for each area and insert a doctor or patient to the system and call a doctor.

3.2.1 Functional Requirements

- Identify the patient's identity via ID number.
- Send verification code to the related email with the patient when the patient enters ID number.
- Read the patient's vital signs.
- Ask patient about symptoms.
- Determine the severity level of the state in three levels: high, medium, and low.
- Direct patient to the appropriate department immediately if the patient's severity state is high.
- Put the patient in the queue if the severity state is medium.
- Send patient's information and situations to hospital's database.
- View Dashboard list about ED cases in admin interface including total cases for each severity level, total cases, number of active and inactive doctors, number of clear and unclear cases, fullness percentage of each area.
- Control the number of clinics in all areas by the admin.
- Insert a doctor by the admin.
- Insert a patient by the admin.
- Call a doctor via email by the admin.
- Change doctor Area by the admin.
- Remove doctor by the admin.
- report patient cases by the doctor.

- View patient history by both admin and doctor.

3.2.2 Non-Functional Requirements

Property	Measure
Availability	the system should Being accessible and available 24/7.
Usability	the system should provide a user-friendly and intuitive interface, ensuring a positive user experience and minimizing user errors.
Performance	the system functions should be finished in short time.
Security	the system must maintain information privacy, only authorized person can access the system
Internationalization and Localization	the system should support multiple languages (Arabic, English).

Table 3-1: non-functional requirements

3.2.3 User Requirements

User requirements specify what the user expects the system to be able to achieve. Our user requirements are as follows. The user should be able to:

- **Be identified from the system:** The system should have a mechanism in place to accurately identify and authenticate users, such as through fingerprint or ID number.
- **Self and fast triage:** The system should allow users to perform self-triage, where they can input their symptoms or vital signs, and the system should quickly analyse this information to determine the urgency of their condition.
- **Fair medical care:** "Ensure that all patients are consistently placed in a queue."
- **Organized ED:** The system should assist in organizing the emergency department (ED), possibly by categorizing and prioritizing patients based on their conditions or needs, ensuring a more efficient and effective workflow.

- **less pressure on ED:** The system should help alleviate the pressure on the emergency department by providing appropriate medical guidance or treatment to patients who do not require immediate care.

3.3 Requirements Specification

In requirements specification we will describe self-trige management system in UML use case diagram which is describe all intractions between usres in the system.

3.3.1 Use case diagram

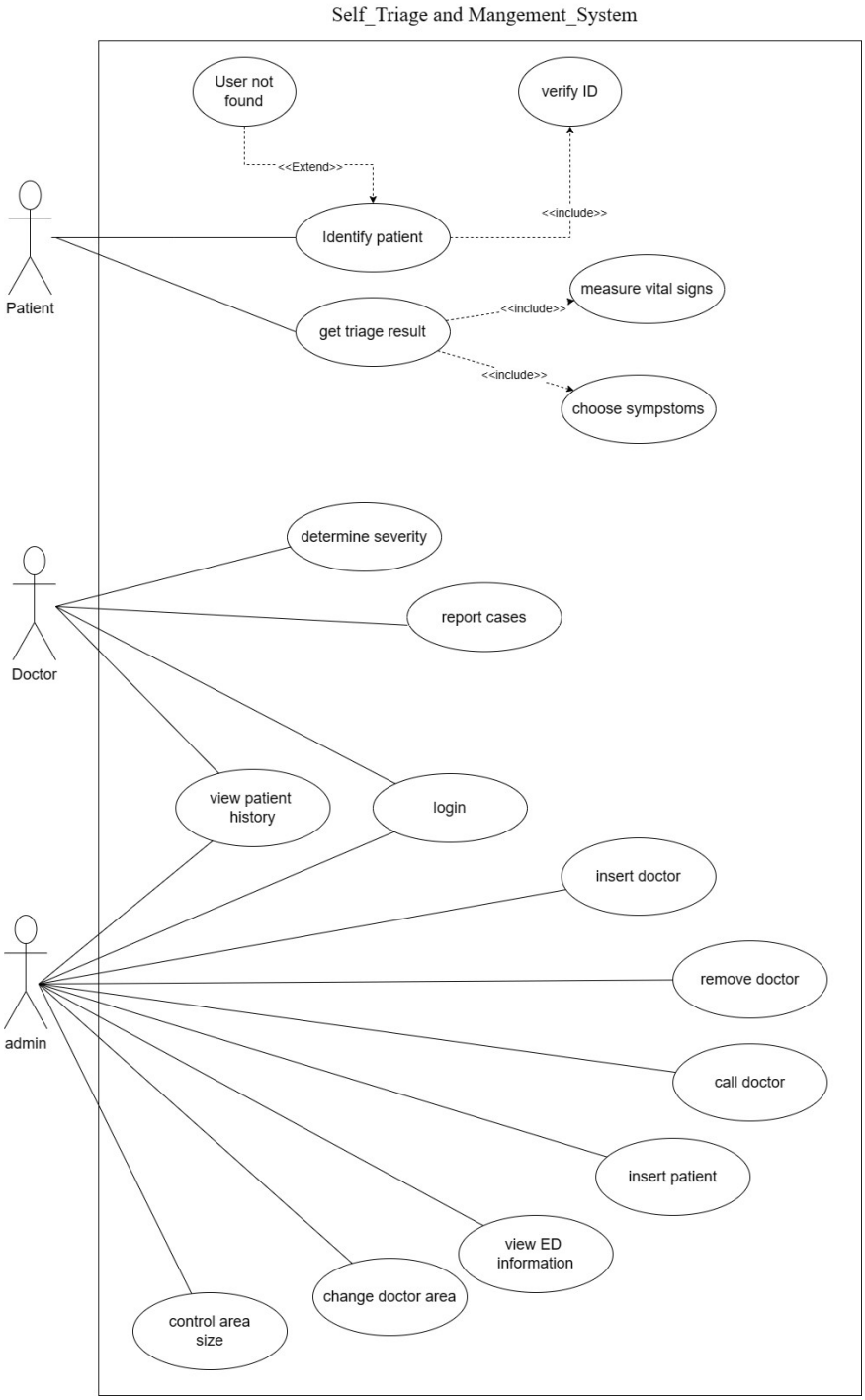


Figure 3.1: use-case diagram for self-triage management system.

3.3.2 Use Case Descriptions

Use case name	Identify patient
Use case ID	1

Actor	Patient
Description	the patient gets identified via fingerprint or ID number
Sequence of Events	-the patient enters register page -the patient must write his ID number or put his fingerprint -self-triage device will identify the patient.

Table 3-2: identify patient use case description.

Use case name	**Get triage result**
Use case ID	2
Actor	Patient
Description	Self-triage device will determine the severity level either high or low by reading patient's vital signs, if its high the patient directed to the appropriate department immediately else it will provide a simple medical advice.
Sequence of Events	-the Self-triage device determine severity level based on the current vital signs and symptoms. -if the severity level is high, Self-triage device will direct the state to the appropriate department immediately. -if the severity level is medium, -if the severity level is low, Self-triage device will provide a simple medical advice.

Table 3-3: get triage result use case description.

Use case name	**choose symptoms**
Use case ID	3
Actor	Patient

Description	the Self-triage device ask the patient about his symptoms
Sequence of Events	-Self-triage device will ask about symptoms. -the patient enters his symptoms -Self-triage device stores the symptoms

Table 3-4: choose symptoms use case description.

Use case name	**Masure vital signs**
Use case ID	4
Actor	Patient
Description	In this stage the patient should measures his vital signs using vital signs devices and Self-triage device will read these vital signs.
Sequence of Events	-vital sings devices use Instructions appears to the patient. -by following the Instructions, the patient can measure his vital signs by using vital signs devices that connected with Self-triage device. -self-triage device recognize patient vital signs.

Table 3-5: Masure vital signs use case description.

Use case name	**Insert doctor**
Use case ID	5
Actor	Admin
Description	The admin can insert new doctor to the database.
Sequence of Events	-First, the admin should add doctor's personal information include his ID, name, email, area number and generate a password for the doctor. -Then the generated password will sent to the doctor' email and the will be inserted to the database.

Table 3-6: Insert doctor use case description.

Use case name	**Remove doctor**
Use case ID	6
Actor	admin

Description	The admin can remove an existing doctor from the database.
Sequence of Events	When the admin enters doctor's ID, the doctor will be deleted from the database.

Table 3-7: Remove doctor use case description.

Use case name	**Change doctor area**
Use case ID	7
Actor	Admin
Description	The admin can move the doctor between areas (low, medium, high).
Sequence of Events	-first, admin enters doctor's ID. -then the admin choose which area to change doctor to it.

Table 3-8: Change doctor area use case description.

Use case name	**Control area size**
Use case ID	8
Actor	Admin
Description	The admin can control the size of all areas (low, medium, high).
Sequence of Events	-the admin chooses which area to change its size. -enter the new size of the chosen area.

Table 3-9: Change doctor area use case description.

Use case name	**Insert patient**
Use case ID	9
Actor	Admin
Description	The ED admin can view all ED information and patients information that comes from admin system via hospital database.
Sequence of Events	-First, the admin should add patient's personal information include his ID, name, email, sex and date of berth. -Then the patient will be inserted to the database.

Table 3-10: insert patient use case description.

Use case name	**Call doctor**
Use case ID	10
Actor	Admin

Description	The admin can call a doctor via email.
Sequence of Events	-the admin should enter the email address that belong to the wanted doctor and the subject and massage body. -the mail will be sent to the doctor.

Table 3-11: call doctor use case description.

Use case name	Patient history
Use case ID	11
Actor	Admin, doctor
Description	The ED admin can view a specific patient's information and triage history.
Sequence of Events	-the admin enters the patient's ID. -then the patient information will appear with his triage history.

Table 3-12: patient history use case description.

Use case name	Report patient
Use case ID	12
Actor	doctor
Description	The doctor can write a reports about patients cases.
Sequence of Events	-the doctor enters the patient's queue number. -write a report for his case. The report will be saved in the database.

Table 3-13: report patient use case description.

3.4 Developmental Methodology

In this section, we will outline the Developmental Methodology utilized in the self-triage and ED Management systems. The chosen software development approach for this system follows.

a waterfall model, consisting of requirements gathering, design, implementation, testing, and maintenance phases, as depicted in the accompanying **Figure 3-2**.

The aim of this methodology **Figure 3-2** is to establish a structured and predictable framework for project management and software development. Its primary objectives include clearly defining requirements from the outset, ensuring a sequential and linear progression of project phases, and ultimately delivering a final product that aligns with the specified requirements. This approach emphasizes adhering to a predetermined plan, minimizing the need for significant changes during development.

Figure 3.2: Waterfall Method

3.4.1 System Methodology

The system methodology consists of five sections (Requirements, Design, implementation, Testing, Deployment)

3.4.1.1 Requirements Gathering

The first step is to collect project requirements. This includes identifying and documenting specific needs and functionality of the system.

- **Medical Patient Identification Requirements.**

Determine patient identification methods, such as using identification numbers or fingerprints.

Decide what data should be collected during the identification process, such as personal information and unique identifiers.

- **Determining the tools for checking vital signs.**

Determine vital signs to be collected from patients such as heart rate, blood pressure, temperature, and respiratory rate.

Determine compatible devices and instruments required to accurately capture vital signs.

Determine data parameters and units to be collected for each vital sign.

- **Determine a set of questions to analyse the situation.**

Select a set of questions to be asked of the patient to assess their condition.

Identify relevant factors to consider when analysing a patient's condition, such as symptoms, severity.

Select the desired format for capturing and storing patients' responses to questions.

- **Define results printing and database storage.**

Determine how the result will be presented to the patient, through a printed document and electronic presentation.

Determine the format and content of the score to ensure clarity and understanding for the patient.

Define database requirements for storing patient information, vital sign data, and case analysis results.

Outline any security and privacy considerations for data storage and access.

3.4.1.2 System Design

Once the requirements have been gathered, the next step is to design. This phase focuses on creating the overall architecture and user interfaces, defining the logic for data processing and analysis, and designing the database structure for data storage.

- **User Interface Design**

Design the user interface for the patient identification module, allowing patients to input their ID numbers or fingerprints.

Design the interface for vital signs monitoring.

Create a user-friendly interface for the severity analysis module, presenting the set of questions and capturing patient responses.

Design a result generation interface that displays the result to the patient, offering clear instructions or medical guidance.

Develop an intuitive interface for ED management personnel to access patient information and states data.

- **Logic Design**

Define the logic for the patient identification module, including the algorithms for verifying ID numbers or fingerprints.

Establish the logic for vital signs monitoring, enabling the system to read and process the incoming data accurately.

Design the logic for the state analysis module, incorporating algorithms to analyze the vital signs data and patient responses.

Determine the logic for generating the final result based on the severity assessment and patient information.

Establish the logic for ED management personnel to access and retrieve patient information and situational data.

- **Database Design**

Determine the structure of the database for storing patient information, including fields for ID numbers, personal details, and contact information.

Design the database schema to accommodate vital signs data, including tables or collections for different types of vital signs and their corresponding parameters.

Create a structure within the database to store states analysis results, such as severity assessments and recommended actions.

Establish the necessary relationships between different tables or collections to ensure efficient data retrieval and management.

3.4.1.3 Implementation

In the implementation phase, the designed components and functionalities are translated into actual code. This step involves writing the necessary programming logic, integrating external tools or libraries, and building the system's functionality according to the defined requirements.

- **Patient Identification Implementation**

Developing the code to handle patient identification using ID numbers or fingerprints.

Integration relevant APIs or libraries for fingerprint recognition or ID verification, if required.

Implementation error handling and validation to ensure accurate and secure patient identification.

- **Severity Analysis Implementation**

Developing the code to present the set of questions to the patient and capture their responses.

Implementation the algorithms for analysing the vital signs data and patient responses to assess the severity of the situation.

Ensuring proper error handling and validation to provide accurate situational analysis results.

- **Vital Signs Monitoring Implementation**

Writing code to interface with the vital signs monitoring devices and retrieve real-time data.

Implementation data processing algorithms to filter and analyse the vital signs readings.

Handle data synchronization and ensure accurate data transmission between the monitoring devices and the system.

- **Result Generation Implementation**

Writing code to generate the result based on the severity assessment and patient information.

Implement the logic to present the result to the patient through printing or electronic display.

Incorporate appropriate formatting and instructions to ensure clarity and ease of understanding for the patient.

- **Database Management Implementation**

Creating the necessary code to establish a connection with the database system.

Implementation data storage and retrieval functionalities for patient information, vital signs data, and situational analysis results.

Handle database operations, such as inserting, updating, and querying data, according to the defined database design.

- **ED Management Interface Implementation**

Developing the code to provide ED management personnel with access to patient information and states data.

Implementation functionalities to retrieve and display patient data based on ED management requests.

3.4.1.4 Testing

Testing plays a crucial role in ensuring functionality, reliability, and performance. The testing phase involves verifying the system's behavior as a whole and validating its compliance with the specified requirements. The testing process consists of the following steps:

- **Functional Testing**

form comprehensive functional testing of the entire system to validate that it meets the functional requirements.

Test user interface component, including patient identification, vital signs monitoring, situational analysis, result generation, and ED management interfaces, to ensure they function as intended.

Verifying the flow of data and information between different modules, ensuring seamless integration and accurate results.

- **Usability Testing**

Involve end-users, such as patients and ED management personnel, to evaluate the usability of the system.

Conduct user-centric testing to assess the system's ease of use, navigation, and overall user experience.

Gathering feedback on the intuitiveness of the interfaces, clarity of instructions, and overall satisfaction with the system's usability.

3.4.1.5 Deployment

Deployment is the process of making the system available for use by patients and ED management. It involves preparing the system for production and ensuring its smooth integration into the healthcare environment.

3.5 Summary

In this chapter, we analyzed self-triage and management system for emergency departments, and we described how it works and interact, in addition to the basic functions that must be available in the system, non-functional requirements, and what is the modeling that suits the system.

Chapter 4: System Design

4.1 Introduction

This chapter will provide an architectural design, static/dynamic models, sequence diagrams, ER diagrams, and the design of the user interface (UI) for the self-triage and management system.

4.2 Architectural Design

The architectural design gives us an overview of how the basic components of the system interact, **Figure4-1** represents the architectural design of the self-triage and ED management system.

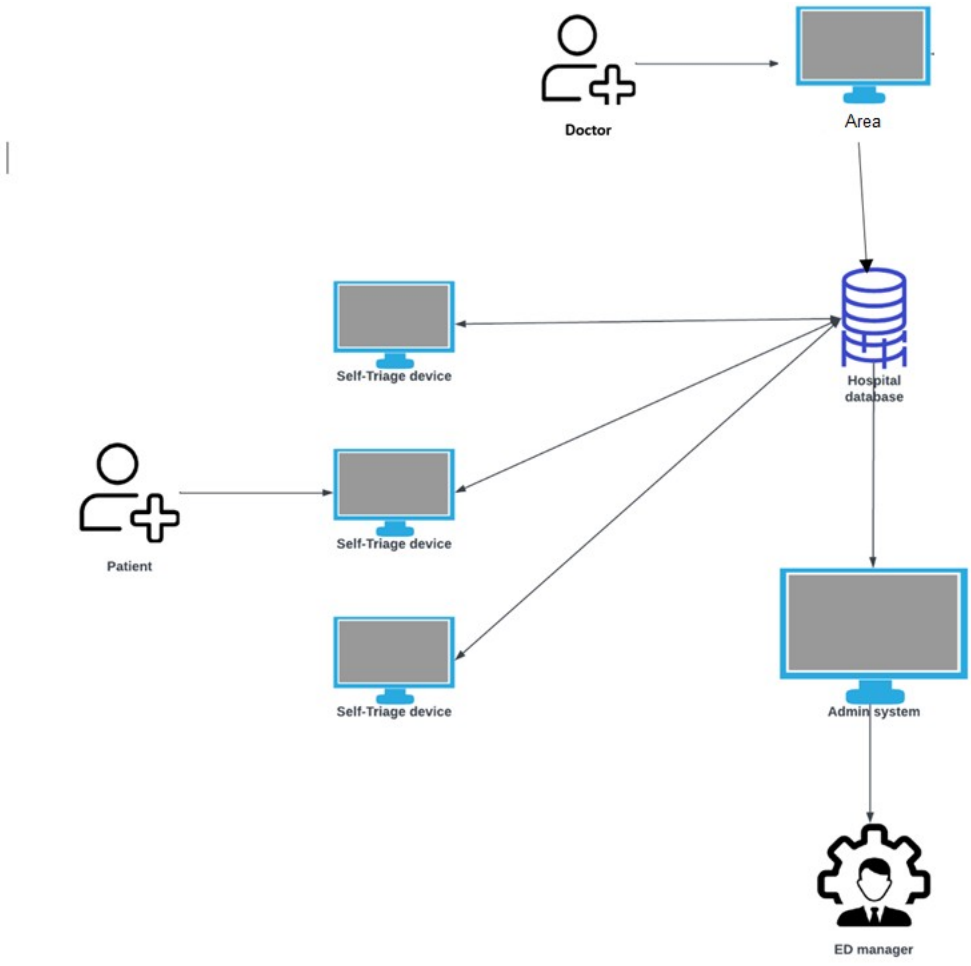


Figure4-1: Architectural Design

Self-triage device: the interface that interacts with patients and assist them to get triaged by themselves.

Admin system: the interface that view the control panel to the ED admin.

Hospital database: is a general database belong to the hospital that manage the ED.

Area: the interface that view the Doctor information of the patient.

4.3 Object Oriented Design

Object Oriented design is concerned with defining the modules in the system and how they interact with each other.

In this section we modeled the system in structural static model using UML class diagram and dynamic models using UML state chart diagram and sequence diagram.

4.3.1 Structural Static Models

Class diagram is we will describe the structure of our proposal and we will define the classes, attributes, operations, and the relationships between classes.

We will implement our proposal using JavaScript, so we designed a class diagram that Compatible with JavaScript language datatypes and objects.

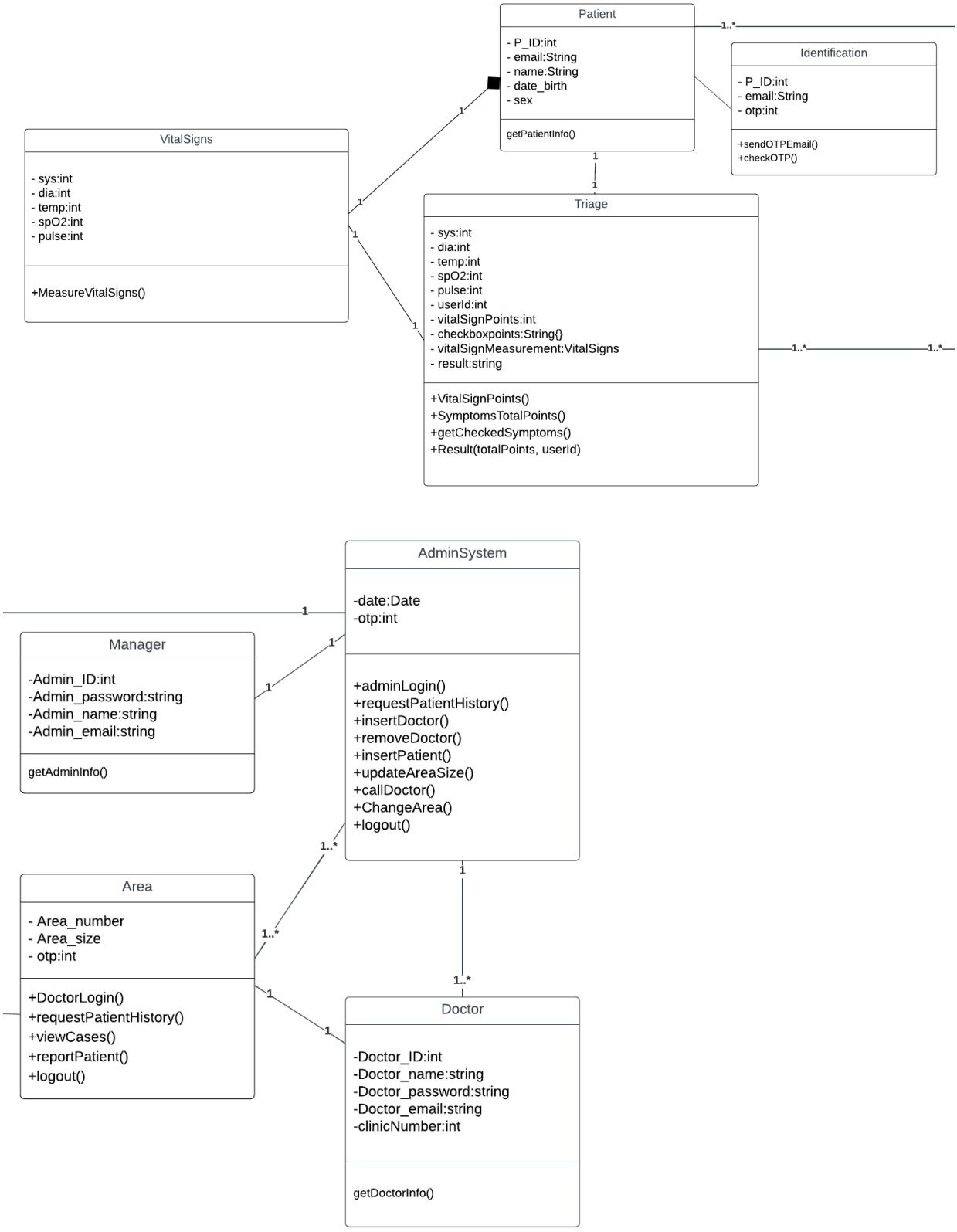


Figure4-2: Class diagram

4.3.2 Dynamic Models

The dynamic model is used to express and represent the behavior and interactions between the system components over the time.

We have designed two dynamic models to represent our proposal which is State chart diagram and sequence diagram.

4.3.2.1 State chart Diagram

State chart diagram describes the flow of different states in the system starting from first state until end state.

- Patient state chart diagram.

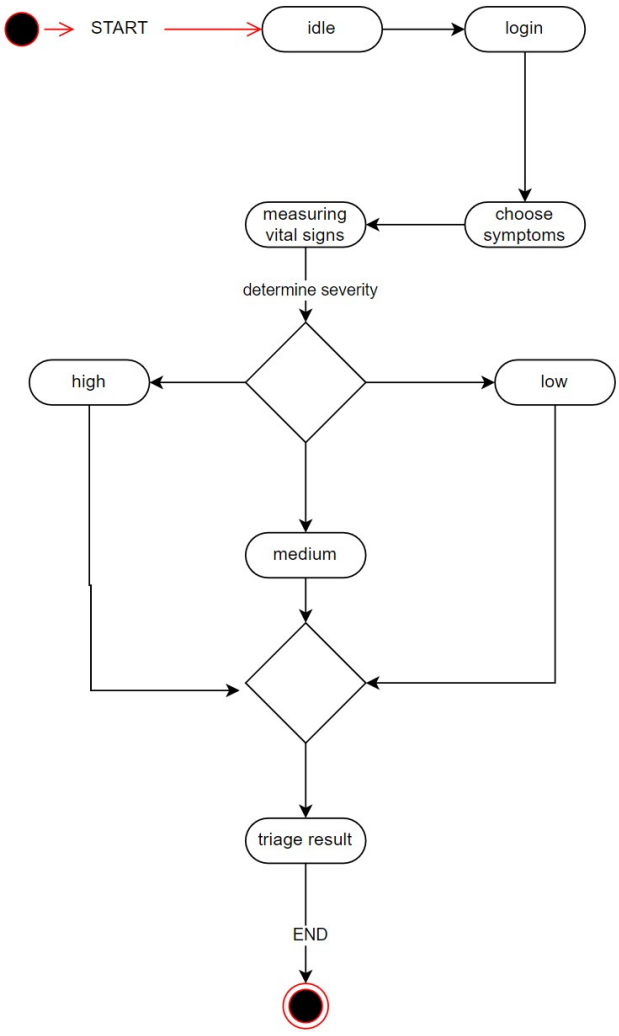


Figure 4-3: patient state chart diagram

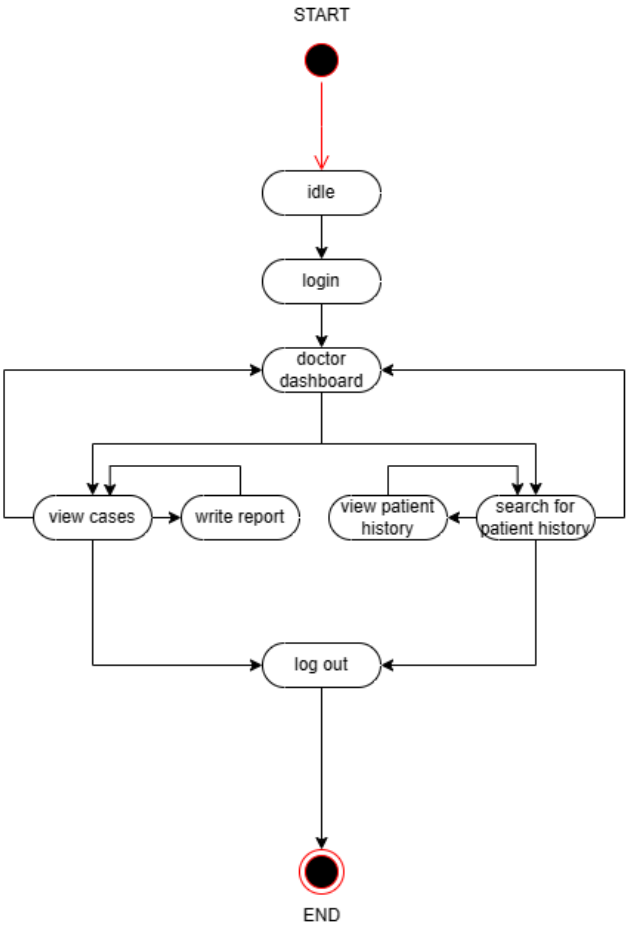


Figure 4-4: doctor state chart diagram.

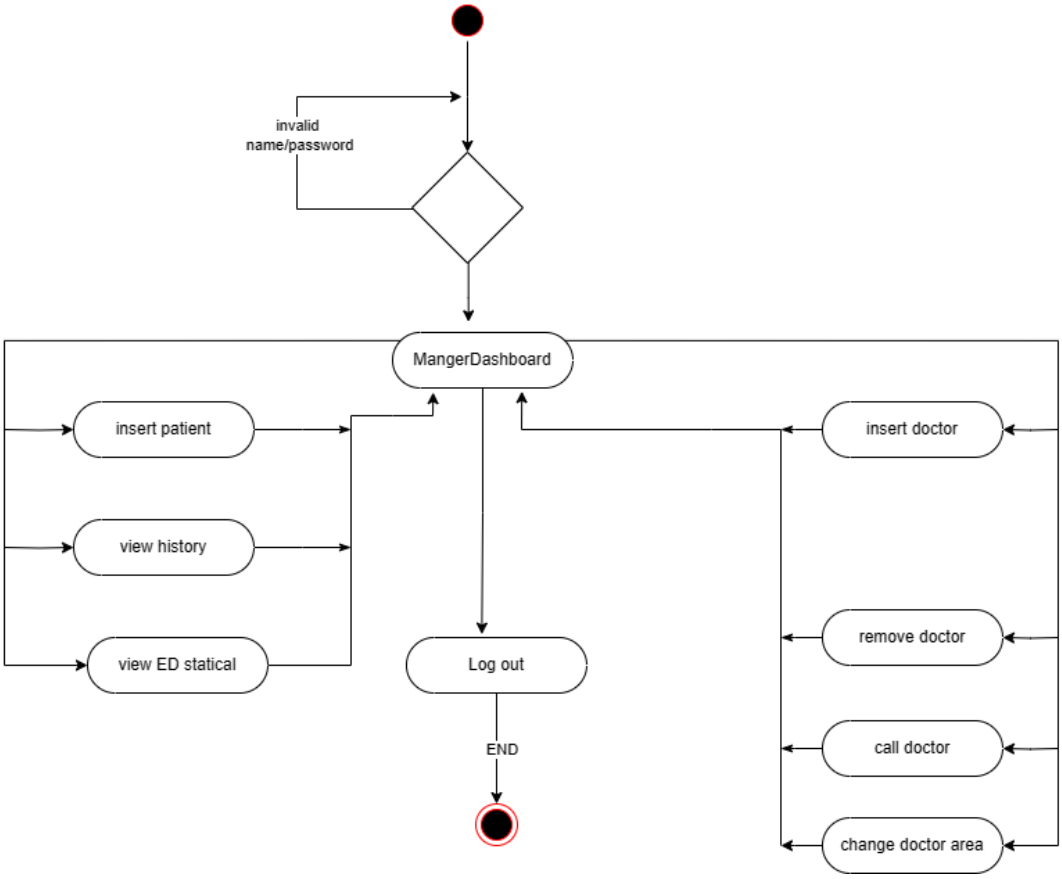


Figure 4-5: admin state chart diagram.

4.3.2.2 Sequence diagram

Sequence diagram in Figure 4-4, Figure4-5, Figure4-6 and Figure4-7 list the Sequence of events that will occurs during time in our proposal.

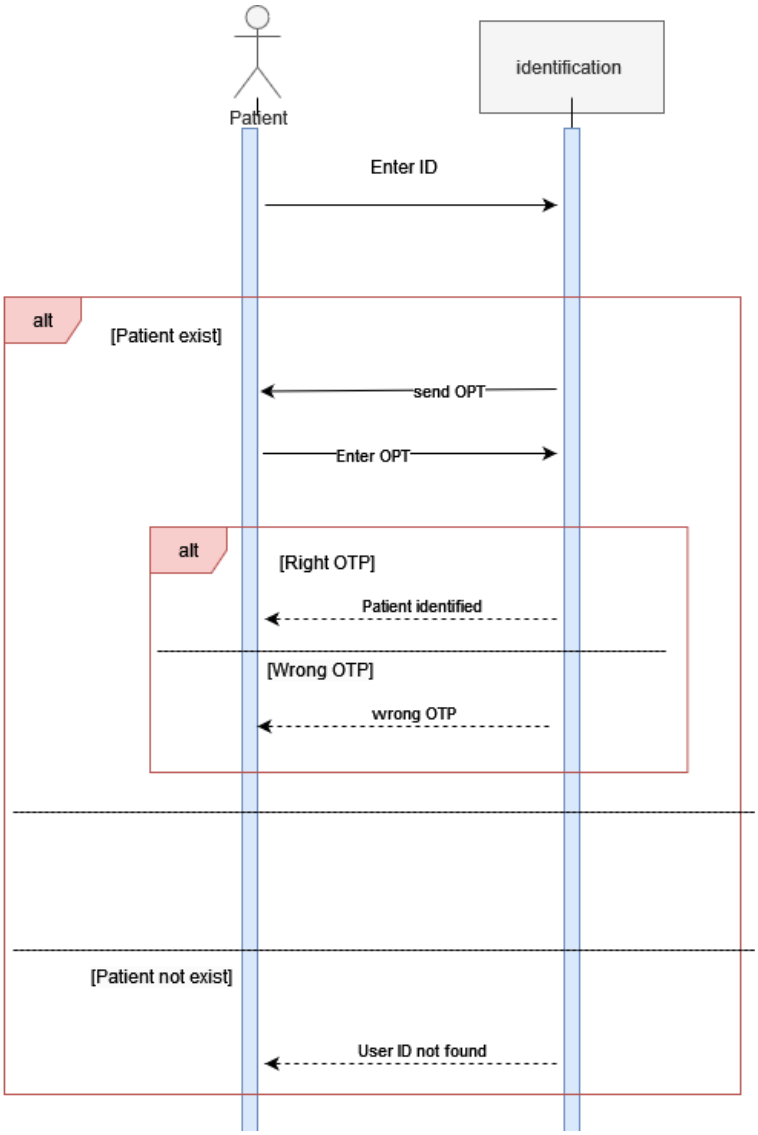


Figure 4-6: identify patient sequence diagram

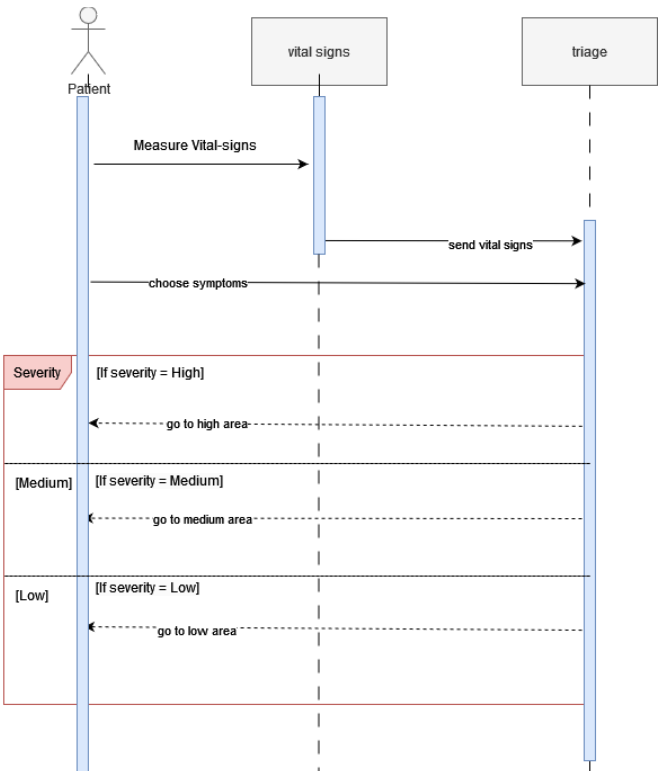


Figure 4-7: self-triage sequence diagram

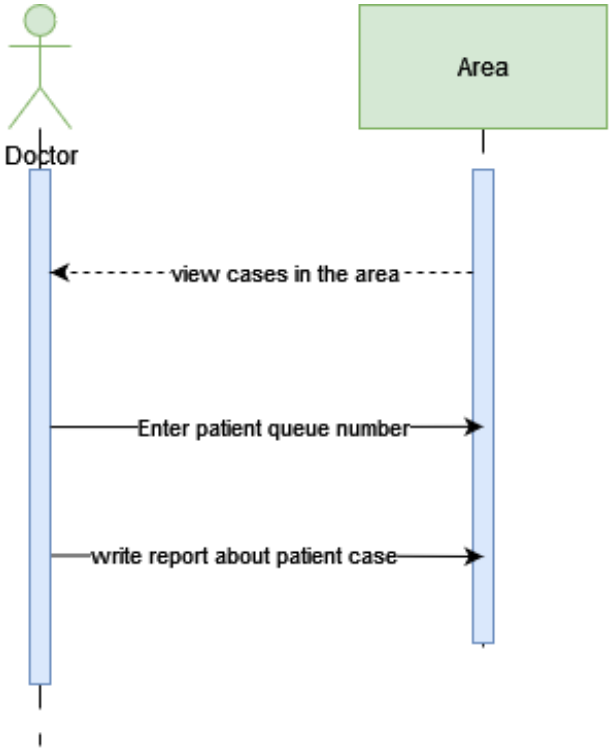


Figure4-8: doctor sequence diagram

4.4 Data Modeling

In data modeling we represented the structure of the entities and the relationship between them in the database using ER diagram in **Figure 4-9** and relational model schema 4-8.

4.4.1 ER diagram

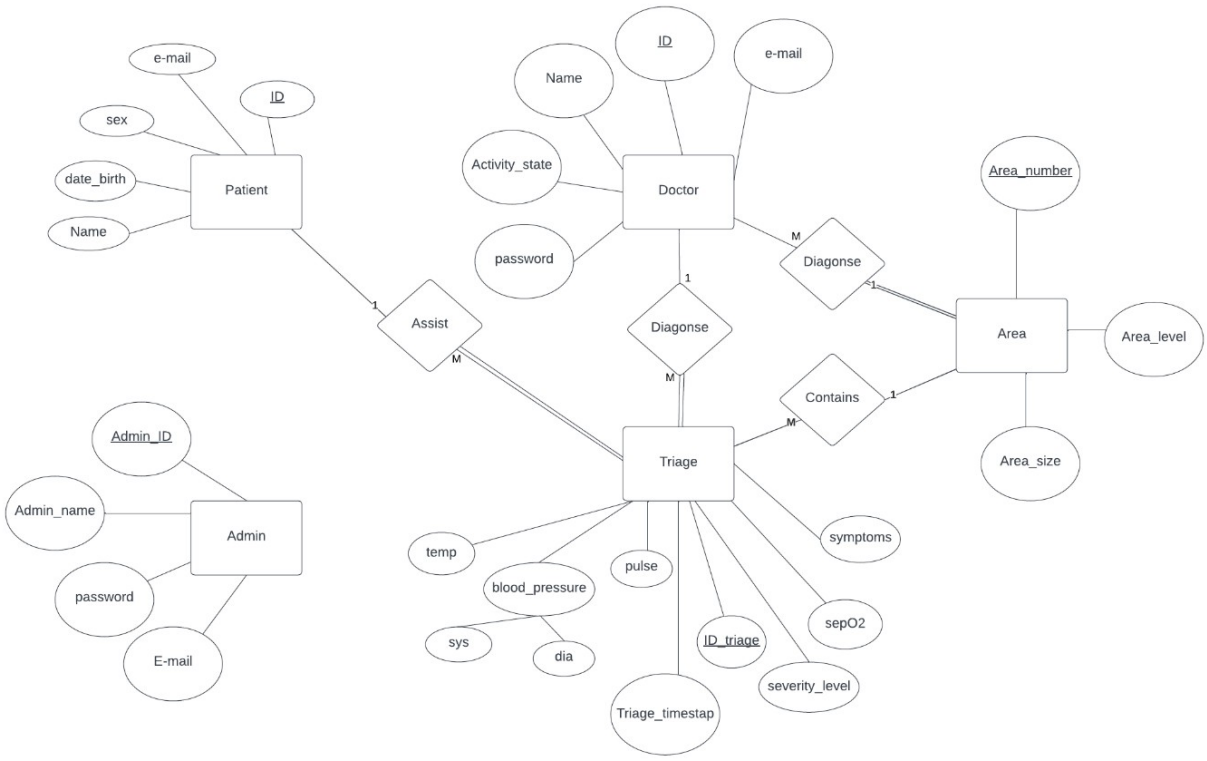


Figure 4-9: ER diagram

4.4.2 Relational model (Schema)

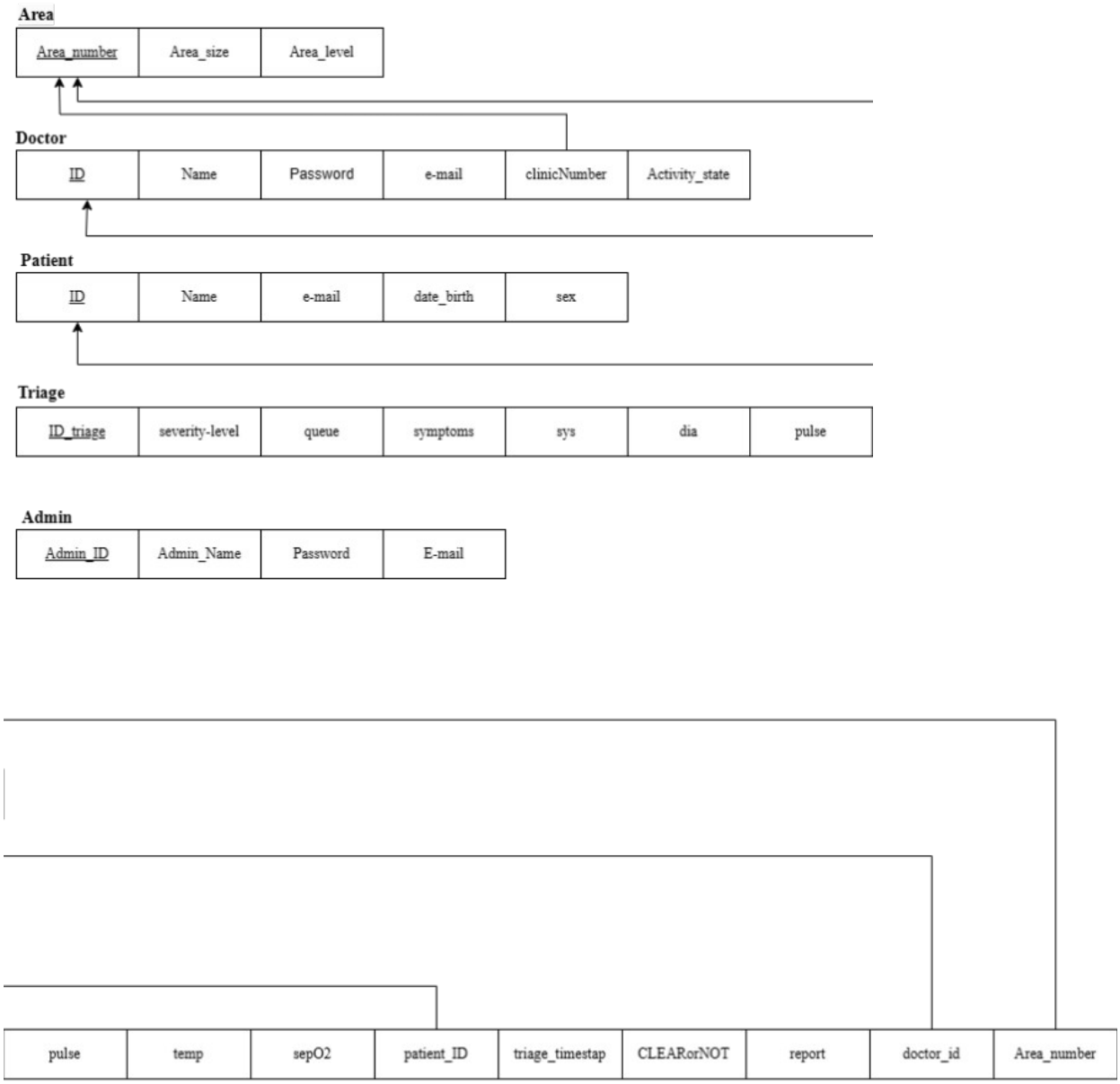


Figure 4-10: Schema

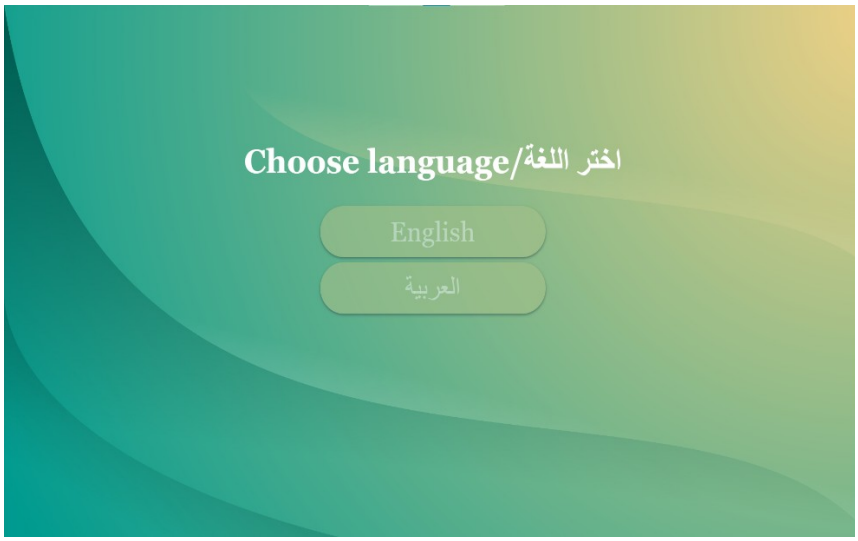
4.5 User Interface Design

The user interface (UI) design phase of the project focuses on creating an intuitive, visually appealing, and user-friendly interface for the self-triage system and ED management.

In this section, we will present the visual representation of the user interface (UI), providing a comprehensive view of how patients and ED management will interact with the interface to accomplish their objectives seamlessly and efficiently.

4.5.1 self-triage device UI

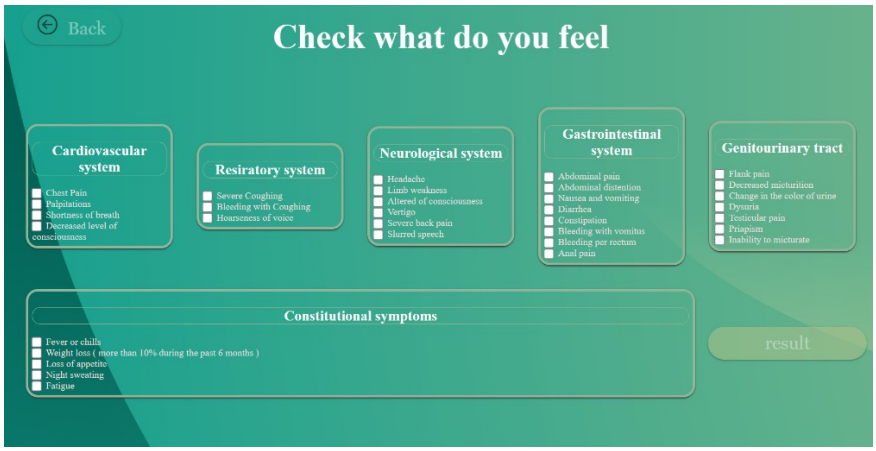
- choosing language frame



- Verify ID number frame.

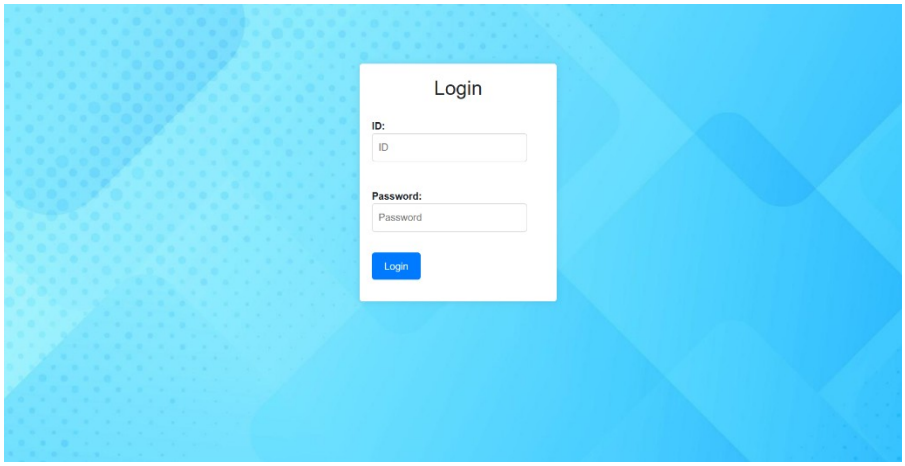
- Masure vital signs frame

- Choosing symptoms frame

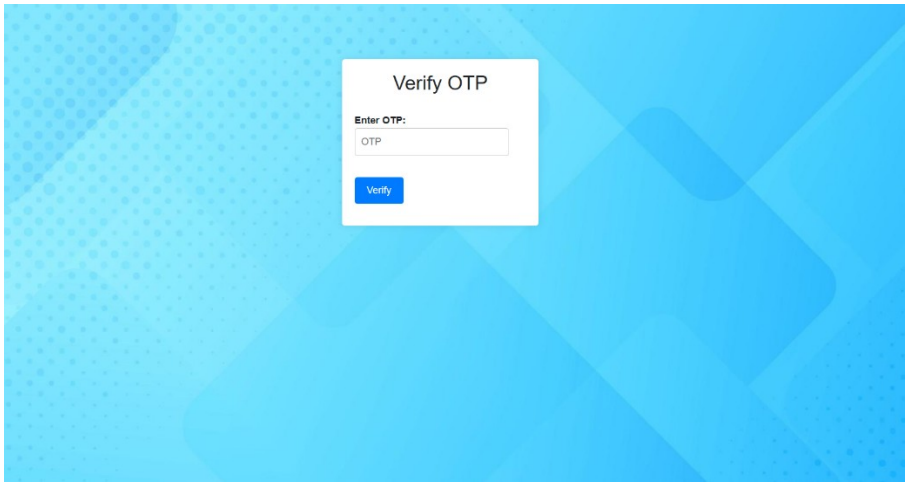


- Result frame

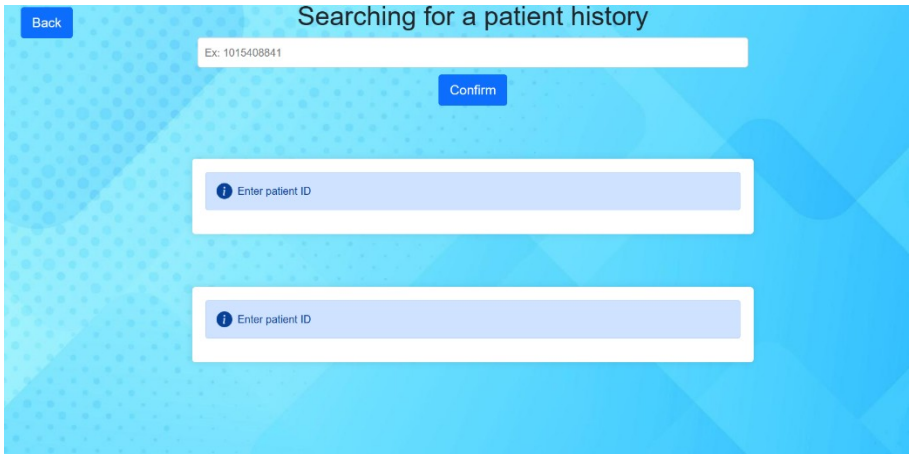
- Admin and doctor login frame



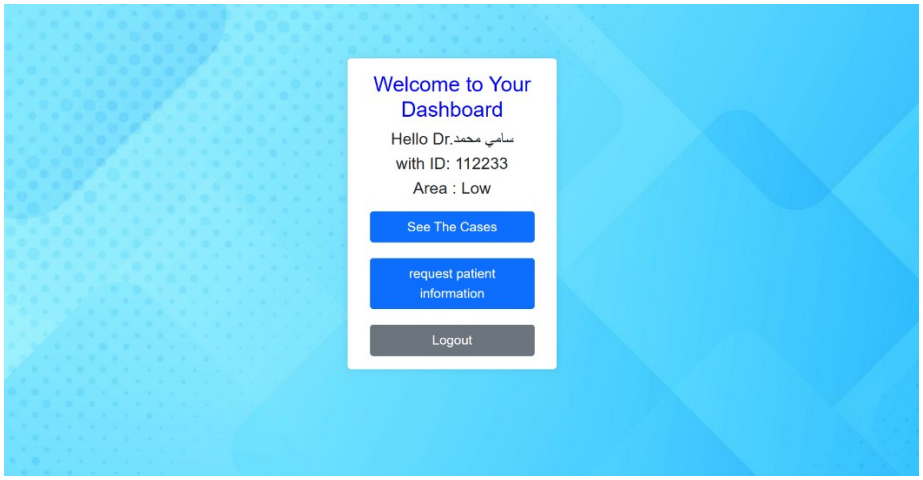
- Verify OTP frame



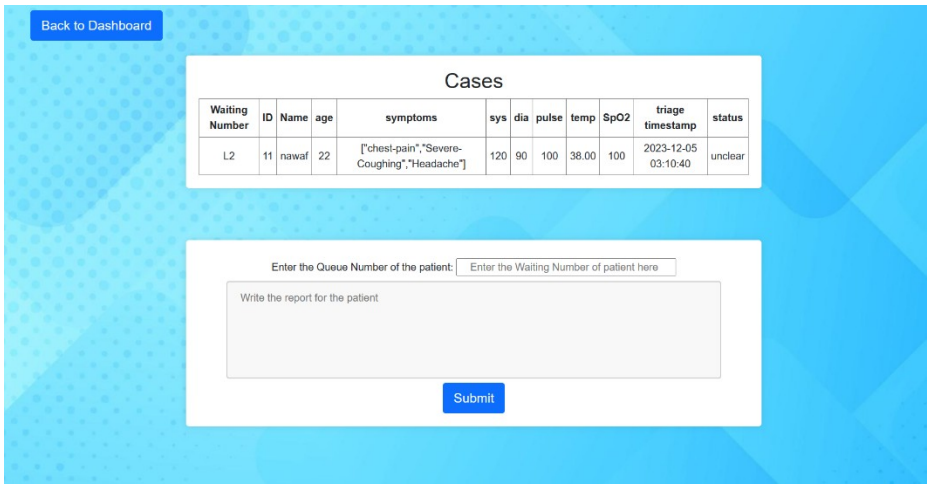
- Patient history frame



- Doctor dashboard frame



- Reporting patient frame



4.6 Summary

The design of a self-triage and ED management system is covered in this chapter, along with the architecture, object-oriented design, data modeling, and user interface design. The system's components and their interactions are described in general terms by the architectural design.

Chapter 5: System Implementation

5.1 Introduction

In this chapter we are going to discuss how we are turning the proposed system into a functional reality and implement all functions that we have introduced in previous chapters.

5.2 Tools and Languages

In this section we use these tools and languages to implement our proposal.

Tool/Language	Logo	Use description
HTML		We used HTML language to build the structure for Self-Triage system and run the system in the browser.
JavaScript		We used JavaScript to interact with HTML structure that we built and make it dynamic and usable.
CSS		To give the users a good impression and improve their experience, we used CSS language to create a special layout that goes along with our proposal
MySQL		To acquire our data, we built a database using MySQL language.
phpMyAdmin		We used phpMyAdmin to handle our MySQL database server.
PHP		We used PHP as backend language to communicate with our database.

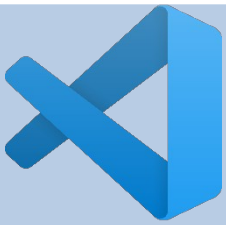
Visual Studio Code		Visual Studio Code is our code editor that we used to write all the functions of the system.
XAMPP		We used XAMPP as development environment for PHP and as backend server.
Bootstrap 5		We used bootstrap to import some basic CSS layout.

Table 5-1: used tools and languages.

5.3 Mapping Design to Implementation

The mapping of the system design to the actual implementation is a critical step in the system implementation process. It involves translating the architectural and object-oriented design into executable code that brings the system to life.

To begin with, the architectural design, The design outlines the overall structure of the system, including the different components, modules, and their interactions.

The object-oriented design, presented in Section 4.3, provides further details on the internal structure and behavior of the system's objects. This design specifies the classes, their attributes, methods, and relationships.

By successfully mapping the design to the implementation, the system begins to take shape, and its functionalities become accessible for testing and evaluation. The mapping process serves as a bridge between the design and the actual implementation, ensuring that the envisioned system is accurately translated into executable code.

5.4 Main/Most Important Codes

This section highlights the most important codes that our proposal is Based on it.

5.4.1 Patient Identification

Getting patient email from the database based on his patient id that he entered it:

```php
<?php

$servername = "127.0.0.1";
$username = "root123";
$password = "root1234";
$dbname = "sleftriage";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$userId = isset($_GET["userId"]) ? $_GET["userId"] : "";

$sql = "SELECT email FROM patient WHERE ID = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("s", $userId);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows > 0) {
    $row = $result->fetch_assoc();
    $email = $row["email"];
    echo json_encode(["exists" => true, "email" => $email]);
} else {
    echo json_encode(["exists" => false]);
}

$stmt->close();
```

After that we fetch above data (include email) and send an OTP via email to the patient email to verify his identity:

```
fetch(`get_email.php?userId=${userId}`)
  .then((response) => response.json())
  .then((data) => {
    if (data.exists) {
      const userEmail = data.email;

      emailjs
        .send("service_y93wki4", "template_rpsy8o1", {
          from_email: "selftraige22@gmail.com",
          to_email: userEmail,
          subject: "ID Verify",
          message: "Hello, Your OTP is: " + otp,
          Body: "Your OTP is: " + otp,
        })
        .then(
          function (response) {
            console.log("Email sent successfully:", response);

            otpVerify.style.display = "block";
            const otpInput = document.getElementById("otp_inp");
            const button2 = document.getElementById("otp1");

            button2.addEventListener("click", () => {
              if (otpInput.value == otp) {
                emailResultDiv.innerHTML = "";
                const redirectURL = `frame5.html?userId=${
                  document.getElementById("userId").value
                }`;
                window.location.href = redirectURL;
              } else {
                otpDiv.innerHTML = "<div class='alert alert-danger d-fle
              }
```

5.4.2 Triage process

After identifying the patient, the triage process begins. The first step is to measure the patient's vital signs. We create a variable named "vitalSignsPoints" to calculate the severity of the case. The following code explains how the vital signs calculation method works and assigns points virtually for all possible measurements of vital signs (Note: sys, dia, pulse, temp and spO2 all these variables are referring to a vital sign that measured by the patient, sys ->Systolic pressure etc.).

```
function VitalSignPoints() {
    let vitalSignPoints = 0;

    if (sys < 90) {
        vitalSignPoints += 3;
    }
    if (sys >= 140 && sys < 180) {
        vitalSignPoints += 1;
    }
    if (sys >= 180 || dia >= 120) {
        vitalSignPoints += 11;
    }
    if (dia < 60 && sys >= 90) {
        vitalSignPoints += 1;
    }
    if (dia >= 90) {
        vitalSignPoints += 1;
    }
    if (pulse > 100 && sys < 90) {
        vitalSignPoints += 3;
    }
    if (pulse > 100 && sys >= 90) {
        vitalSignPoints += 1;
    }
    if (temp > 38 && temp < 40) {
        vitalSignPoints += 5;
    }
    if (temp >= 40) {
        vitalSignPoints += 11;
    }
    if (temp < 35) {
        vitalSignPoints += 3;
    }
    if (spO2 < 96) {
        vitalSignPoints += 3;
    }
```

Below object contains the common symptoms and we assign one points for each symptom to calculate the total with vital signs

```
const checkboxPoints = {
    'chest-pain': 1,
    'palpitations': 1,
    'shortness-of-breath': 1,
    'decreased-level-of-consciousness': 1,
    'Severe-Coughing': 1,
    'Bleeding-with-Coughing': 1,
    'Hoarseness-of-voice': 1,
    'Headache': 1,
    'Limb weakness': 1,
    'Altered of consciousness': 1,
    'Vertigo': 1,
    'Severe back pain': 1,
    'Slurred speech': 1,
    'Abdominal pain': 1,
    'Abdominal distention': 1,
    'Nausea and vomiting': 1,
    'Diarrhea': 1,
    'Constipation': 1,
    'Bleeding with vomitus': 1,
    'Bleeding per rectum': 1,
    'Anal pain': 1,
    'Flank pain': 1,
    'Decreased micturition': 1,
    'Change in the color of urine': 1,
    'Dysuria': 1,
    'Testicular pain': 1,
    'Priapism': 1,
    'Inability to micturate': 1,
    'Fever or chills': 1,
    'Weight loss': 1,
    'Loss of appetite': 1,
    'Night sweating': 1,
    'Fatigue': 1,
};
```

Here we check the symptoms that the user has choose from the symptoms box and we calculate the total pointes:

```
function SymptomsTotalPoints() {
    let totalPoints = 0;

    for (const checkboxId in checkboxPoints) {
        const checkbox = document.getElementById(checkboxId);
        if (checkbox.checked) {
            totalPoints += checkboxPoints[checkboxId];
        }
    }

    totalPoints += VitalSignPoints();

    return totalPoints;
}
```

Now after we calculated the total points we can determine the triage result, below method show how triage result determined based on the total points.

```
function Result(totalPoints, userId) {
    let result = '';
    if (totalPoints === 0) {
        result = 'No Risk';
    } else if (totalPoints <= 5) {
        result = 'Low';
    } else if (totalPoints <= 10) {
        result = 'Medium';
    } else {
        result = 'High';
    }

    let triage_result;
    switch (result) {
        case 'No Risk':
        case 'Low':
        case 'Medium':
        case 'High':
            triage_result = 'triage.php';
            break;
        default:
            triage_result = 'triage.php';
    }

    let redirectURL;
if (result === 'No Risk') {
    redirectURL = `noRisk.html`;
    window.location.href = redirectURL;

} else {
    redirectURL = `frame7.html?result=${result}&userId=${userId}`;
}
```

Finally, we will insert the triage result in the database.

```php
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die(json_encode(['status' => 'error', 'message' => 'Connection failed: ' . $conn->connect_error]));
}
$data = json_decode(file_get_contents('php://input'), true);

$severityLevels = [
    'Low', 'Medium', 'High'
];
if (in_array($data['Severity_level'], $severityLevels)) {
    $triageSql = "CALL InsertTriage(?, ?, ?, ?, ?, ?, ?, ?, ?)";
    if ($stmt = $conn->prepare($triageSql)) {
        date_default_timezone_set('Asia/Riyadh');
        $currentTimestamp = date('Y-m-d H:i:s');
        $stmt->bind_param("ssdddddis", $data['Severity_level'], $data['symptoms'], $data['sys'], $data['dia'], $data['pulse'], $data['temp'], $data[

        if ($stmt->execute()) {
            $response = ['status' => 'success'];
        } else {
            $response = ['status' => 'error', 'message' => $conn->error];
        }
        $stmt->close();
    } else {
        $response = ['status' => 'error', 'message' => $conn->error];
    }
} else {
    $response = ['status' => 'unknown'];
}

$conn->close();
```

5.4.3 Admin system

Below is log in code for the admin, we request admin login information from the database, and we validate this information with the entered ID and password if the validation succeed then we send an OTP to the admin email that related to him in the data base using PHPMailer library.

```php
$query->bind_param('s', $ID_Ofadmin);
$query->execute();
$query->bind_result($db_ID, $db_password, $db_name, $db_Email);
$query->fetch();

if ($db_ID && $hash === $db_password) {
    $otp = mt_rand(1000, 9999);

    $_SESSION['Admin_ID'] = $db_ID;
    $_SESSION['Admin_Name'] = $db_name;
    $_SESSION['otp'] = $otp;
    $_SESSION['Admin_password'] = $db_password;


    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->SMTPAuth = true;
    $mail->Username = 'selftriage22@gmail.com';
    $mail->Password = 'icrziehnektjrejl';
    $mail->SMTPSecure = 'tls';
    $mail->Port = 587;

    $mail->setFrom('selftriage22@gmail.com', 'Admin View');
    $mail->addAddress($db_Email, $db_name);

    $mail->Subject = 'Your OTP for Login';
    $mail->Body = 'Hello ' . $db_name . ', This is your OTP : '. $otp .'';
```

After sending OTP to the admin's email, the admin should verify his identity by entering the OTP.

Below code shows how the system check the OTP:

```php
<?php
session_start();

if (!isset($_SESSION['Admin_ID']) || !isset($_SESSION['otp'])) {
    header("Location: Admin_login.php");
    exit;
}

$error = "";

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $userEnteredOTP = $_POST['otp'];
    $_SESSION['verify_otp'] = $userEnteredOTP;

    if ($userEnteredOTP == $_SESSION['otp']) {

        header("Location: dashboard.php");
        exit;
    } else {
        $error = '<div class="container">
        <div class="alert alert-danger d-flex align-items-center" role="alert">
      <svg class="bi flex-shrink-0 me-2" width="24" height="24" role="img" aria-label="Danger:">
      <use xlink:href="#exclamation-triangle-fill"/>
      </svg>
      <div>
      Invalid OTP. Please try again</div>
      S</div>
      </div></div>';
    }
}
?>
```

The admin can see patient history by entering his ID, so this code contains the SQL query that serves this function this function.

```php
if ( $_SERVER[ 'REQUEST_METHOD' ] === 'POST' ) {
    $patientID = $_POST[ 'patientID' ];

    $servername = '127.0.0.1';
    $username = 'root123';
    $dbPassword = 'root1234';
    $dbname = 'sleftriage';

    $conn = new mysqli( $servername, $username, $dbPassword, $dbname );

    if ( $conn->connect_error ) {
        die( 'Connection failed: ' . $conn->connect_error );
    }

    $query = $conn->prepare( "SELECT *, TIMESTAMPDIFF(YEAR, patient.date_birth, CURDATE()) AS age
    FROM triage
    JOIN patient ON triage.patient_ID = patient.ID
    WHERE patient.ID = ?" );
    $query->bind_param( 's', $patientID );
    $query->execute();
    $result = $query->get_result();

    $conn->close();
}
```

Here we can see how to insert a doctor by the admin.

We made a proceduer in phpMyAdmin that insert doctor and we call it in the query below and we assign the information that entered by the admin to the query except the password its generated randomly and the system send it to the doctor email.

```php
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
if (isset($_POST['doctorid'], $_POST['doctorname'], $_POST['doctorpassword'], $_POST['doctoremail'], $_POST['doctorAreanumber'])) {
    $doctorid = $_POST['doctorid'];
    $doctorname = $_POST['doctorname'];
    $doctorpassword = $_POST['doctorpassword'];
    $doctoremail = $_POST['doctoremail'];
    $doctorAreanumber = $_POST['doctorAreanumber'];
    $query = $conn->prepare("CALL `InsertDoctor`(?,?,?,?,?);");
    $query->bind_param('ssssi', $doctorid, $doctorname, $doctorpassword, $doctoremail, $doctorAreanumber);
    if ($query->execute()) {
        $mail = new PHPMailer;
        $mail->isSMTP();
        $mail->Host = 'smtp.gmail.com';
        $mail->SMTPAuth = true;
        $mail->Username = 'selftriage22@gmail.com';
        $mail->Password = 'icrziehnektjrejl';
        $mail->SMTPSecure = 'tls';
        $mail->Port = 587;

        $mail->setFrom('selftriage22@gmail.com', 'Admin');
        $mail->addAddress($doctoremail, $doctorname);
        $mail->Subject = 'Welcome to our ED';
        $mail->Body = 'Dear ' . $doctorname . ', Welcome to our ED!. Your ID: ' . $doctorid . ' with password: ' . $doctorpassword;

        if ($mail->send()) {
            echo 'Doctor Insert successfully';
        } else {
            echo 'Email could not be sent. Mailer Error: ' . $mail->ErrorInfo;
        }
    } else {
        echo 'Failed to execute query.';
    }
```

Below code is responsible about updating the size of each area:

```
if(isset($_POST['New_low_size'])){
$Area=1;
$newSize=$_POST['New_low_size'];
$query =$conn->prepare( "UPDATE Area SET Area_size =? WHERE Area_number =?");
$query->bind_param('ii', $newSize,$Area);
$query->execute();
$query->close();

}

if(isset($_POST['New_med_size']) ){
    $Area=2;
    $newSize=$_POST['New_med_size'];

    if ($newSize != 0){
        $query =$conn->prepare( "UPDATE Area SET Area_size =? WHERE Area_number =?");
    $query->bind_param('ii', $newSize,$Area);
    $query->execute();
    $query->close();
    }


    }

if(isset($_POST['New_high_size'])){
    $Area=3;
    $newSize=$_POST['New_high_size'];

   if ($newSize != 0){
        $query =$conn->prepare( "UPDATE Area SET Area_size =? WHERE Area_number =?");
    $query->bind_param('ii', $newSize,$Area);
    $query->execute();
    $query->close();
```

5.4.4 Doctor Interface

below code is responsible about doctor basic functionality which is clearing cases and sending report and save it in the data base, so first we post the waiting number for the patient number that entered by the doctor, and we post the report, then we update the triage with the new state "clear", report from the doctor and the doctor ID.

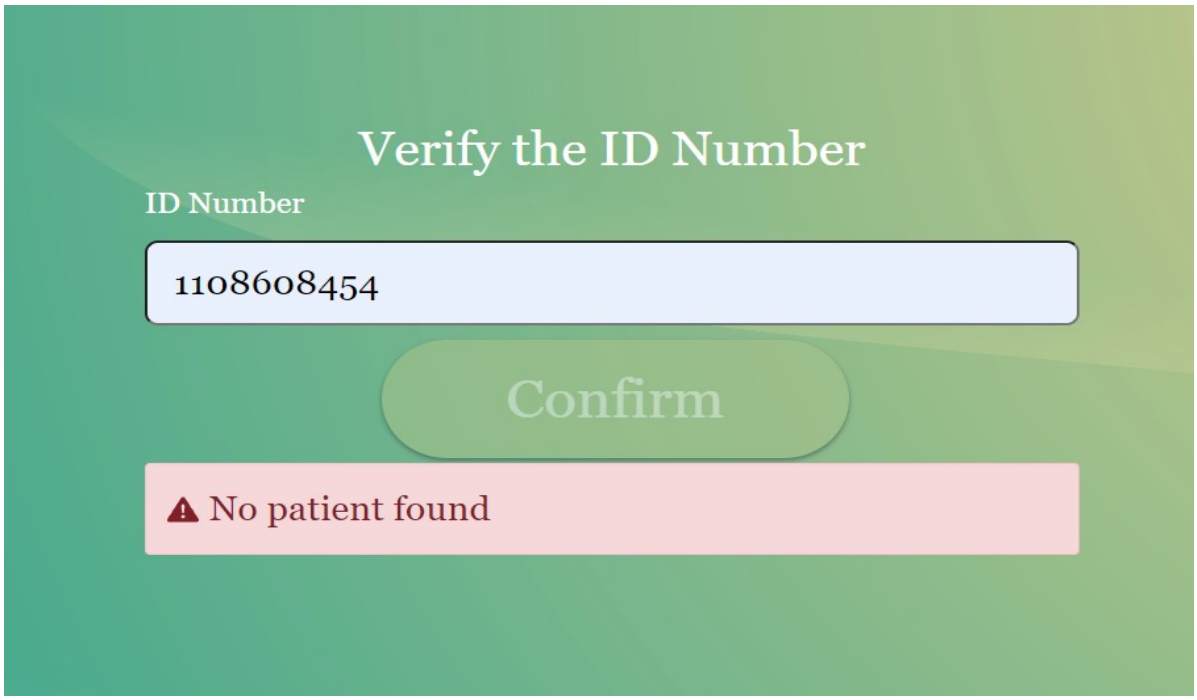
```php
$query = $conn->prepare($sql);
$query->bind_param('s', $clearStatus);
$query->execute();
$result = $query->get_result();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (isset($_POST['waiting_number']) && isset($_POST['report'])) {
        $waitingNumber = $_POST['waiting_number'];
        $report = $_POST['report'];

        $insertSql = "UPDATE triage SET doctor_id = ? ,report = ?, CLEARorNOT = 'CLEAR' WHERE queue = ?";
        $insertQuery = $conn->prepare($insertSql);
        $insertQuery->bind_param('iss', $DoctorID, $report, $waitingNumber);

        if ($insertQuery->execute()) {
            echo "<script>redirectURL = `http://localhost/doctor/Cases.php`;
            window.location.href = redirectURL;

            </script>";

        } else {
            echo "Error inserting report: " . $conn->error;
        }
    }
}
```

5.5 System Testing

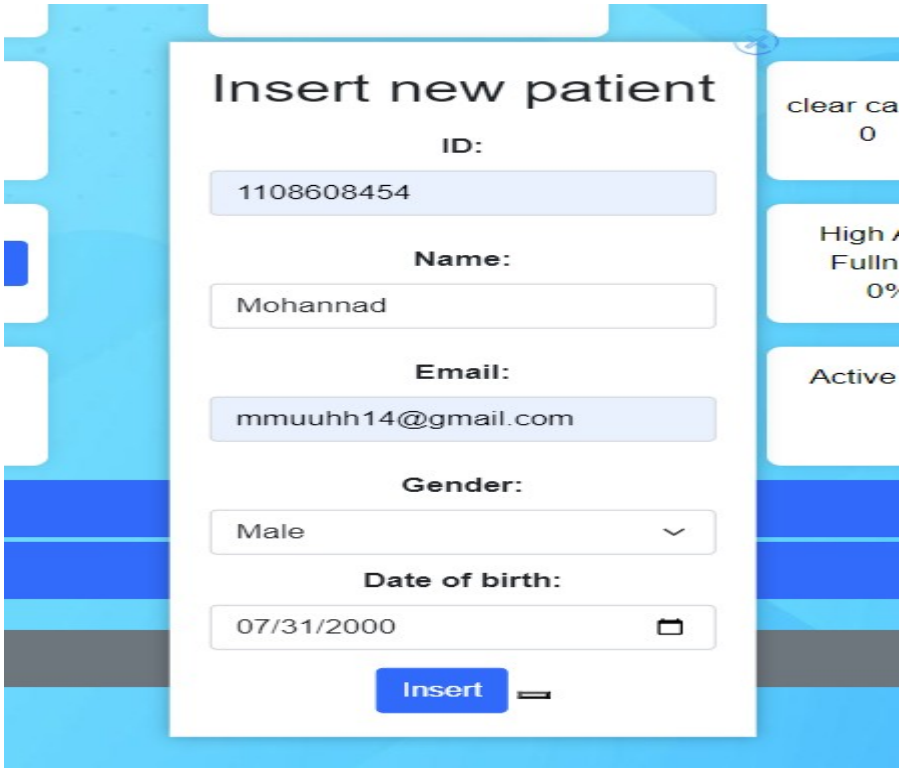
In this section, we presented visual representations showcasing the interaction between the administrator, doctor, and patient within the system. Through these pictures, we depicted the various functionalities and actions that can be performed by the administrator as well as the doctor and patient users. These images provide a clear visual demonstration of how different users engage with the system and utilize its features to manage medical information and facilitate communication. The inclusion of these visuals enhances the understanding and appreciation of the overall system and its user interfaces.

The initial interface presented to the patient:

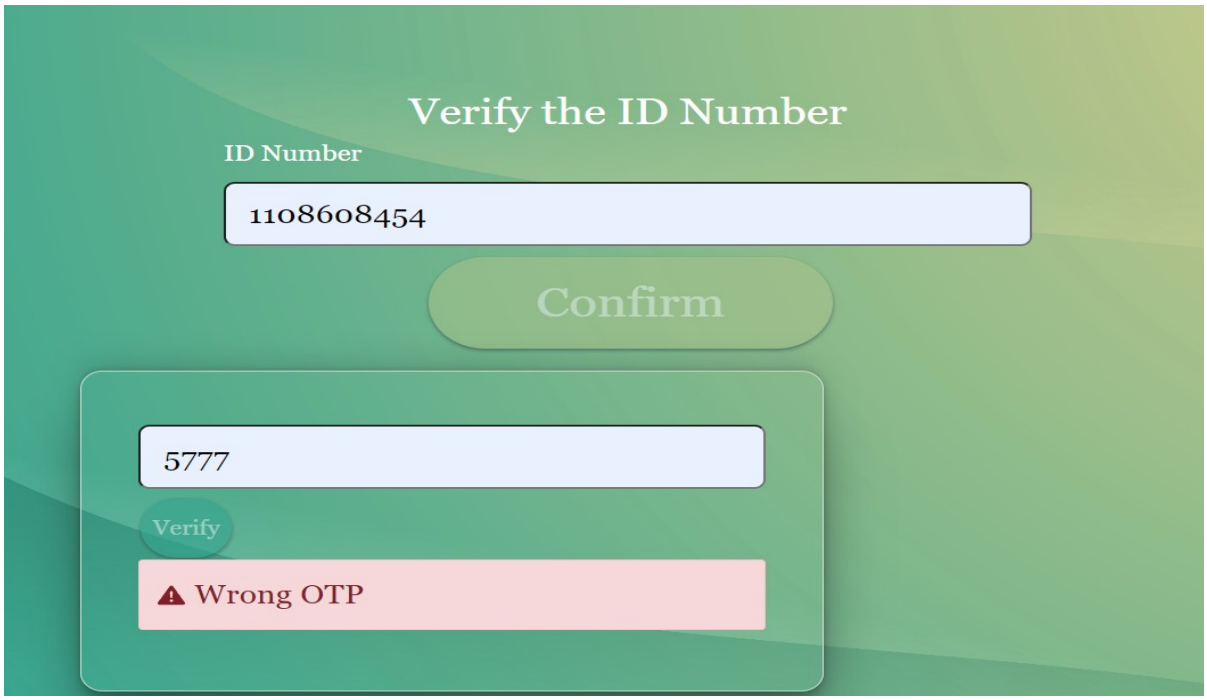
Access to self-triage is only granted to the patients that exist in the database.



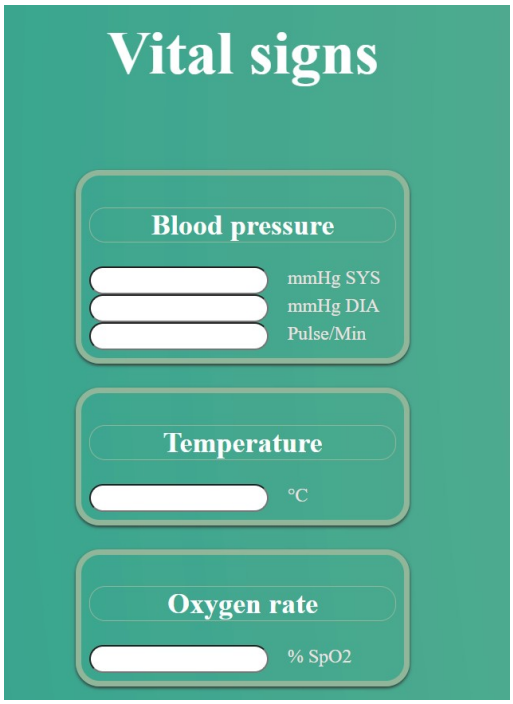
From the admin: admin has the capability to add new patient entries through the administrative dashboard.



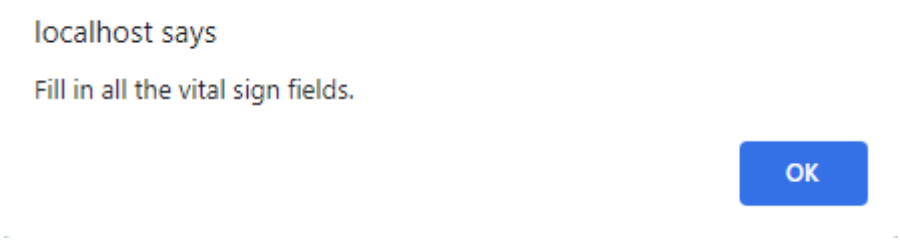
After adding the patient from the admin, it will appear OTP filed, If the patient enters an incorrect OTP (One-Time Password), an error message will be displayed.



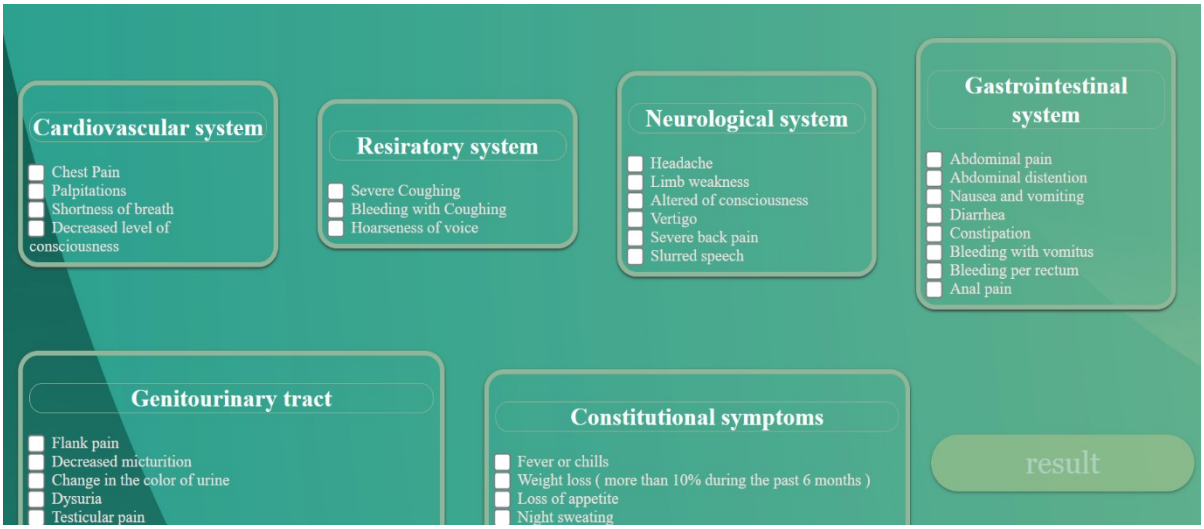
correctly entering the OTP that sent to their email address, the patient gains access to self-triage for making a diagnosis:



Failure to input complete vital signs will result in the patient being denied access to the system and unable to finalize their assessment:

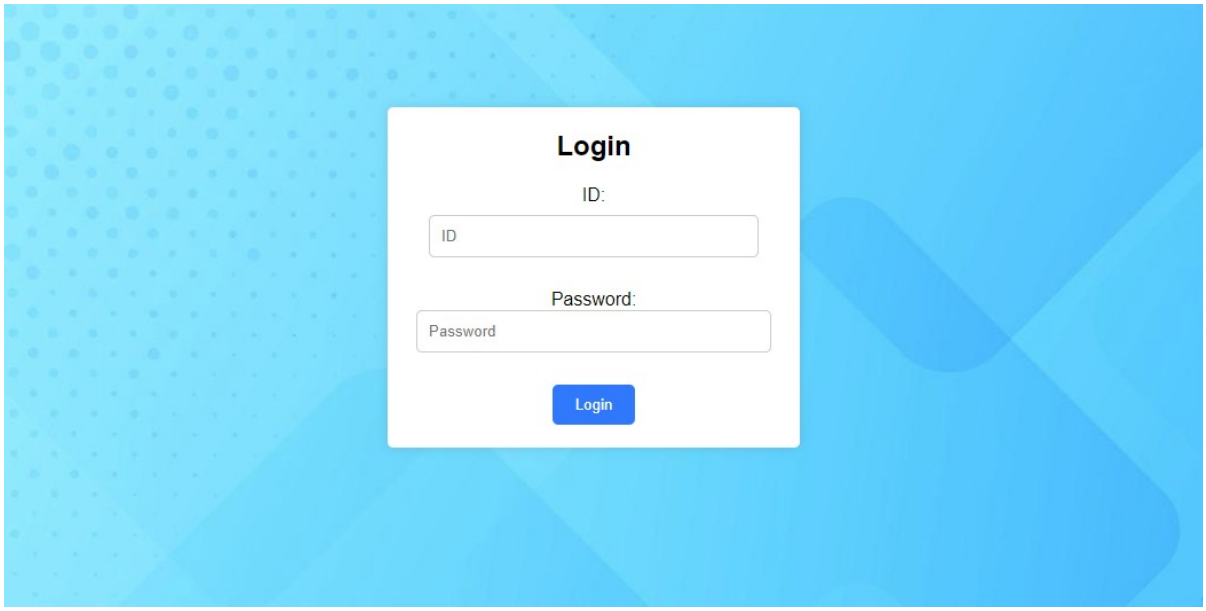


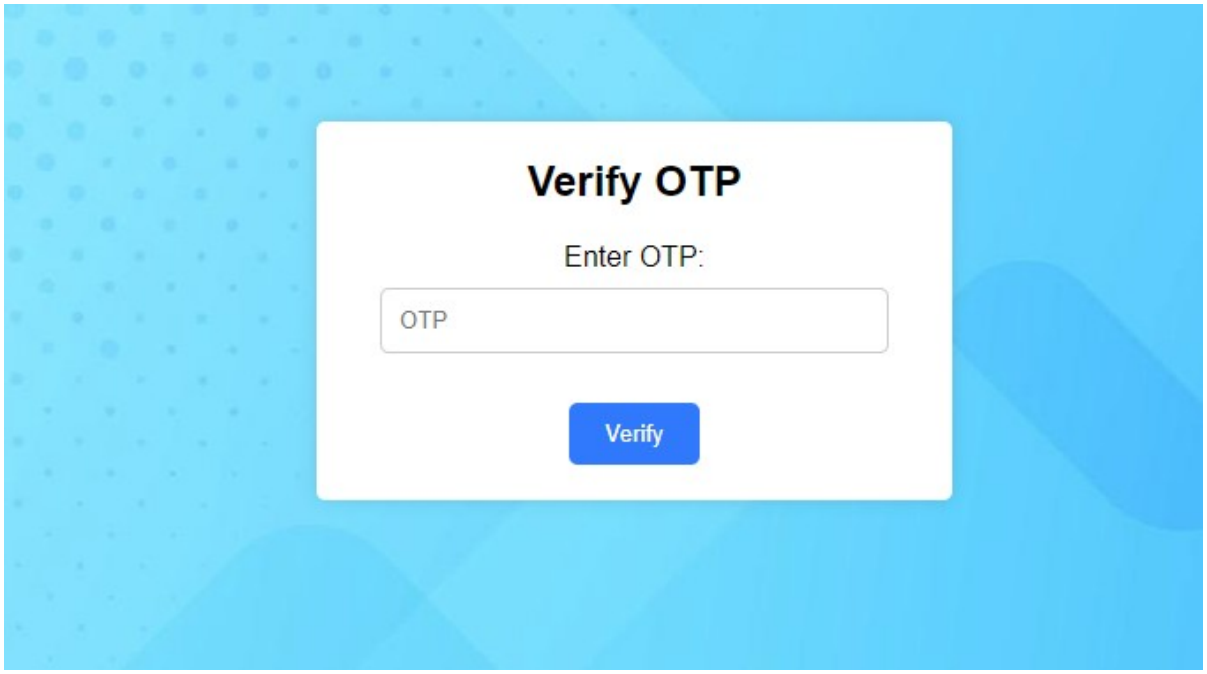
After measuring vital signs, the patient proceeds to the subsequent section to choose symptoms; it's possible for the patient to be symptom-free even if their vital signs are abnormal.



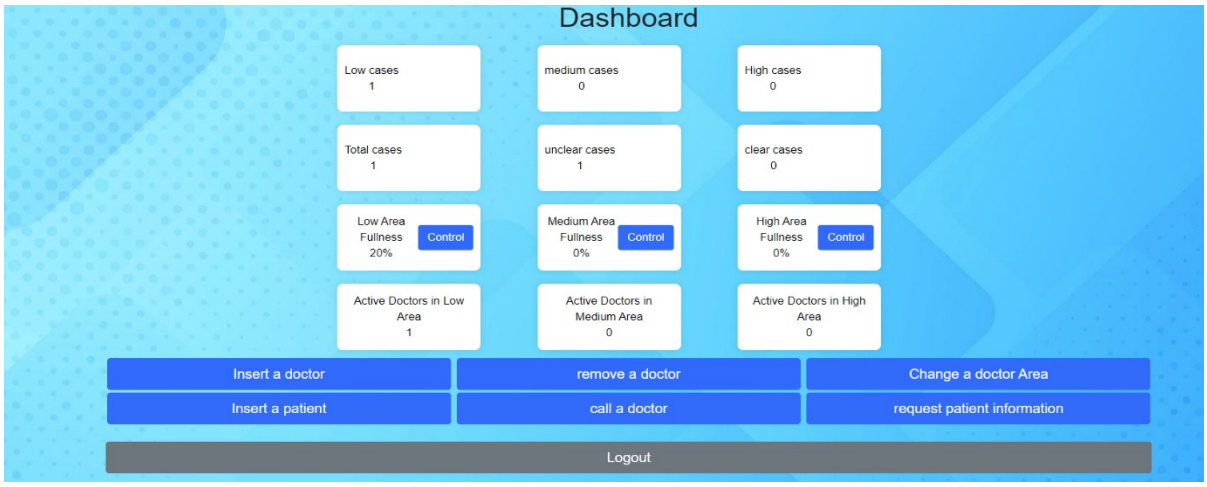
After all, the patient will get his result including condition, waiting number and area number.

Below the login page and OTP page for the admin and doctor interface are same for identical. But admin have access to this page. The admin can add doctors and patients to the system. Additionally, the admin can contact a doctor via email and request detailed patient information.

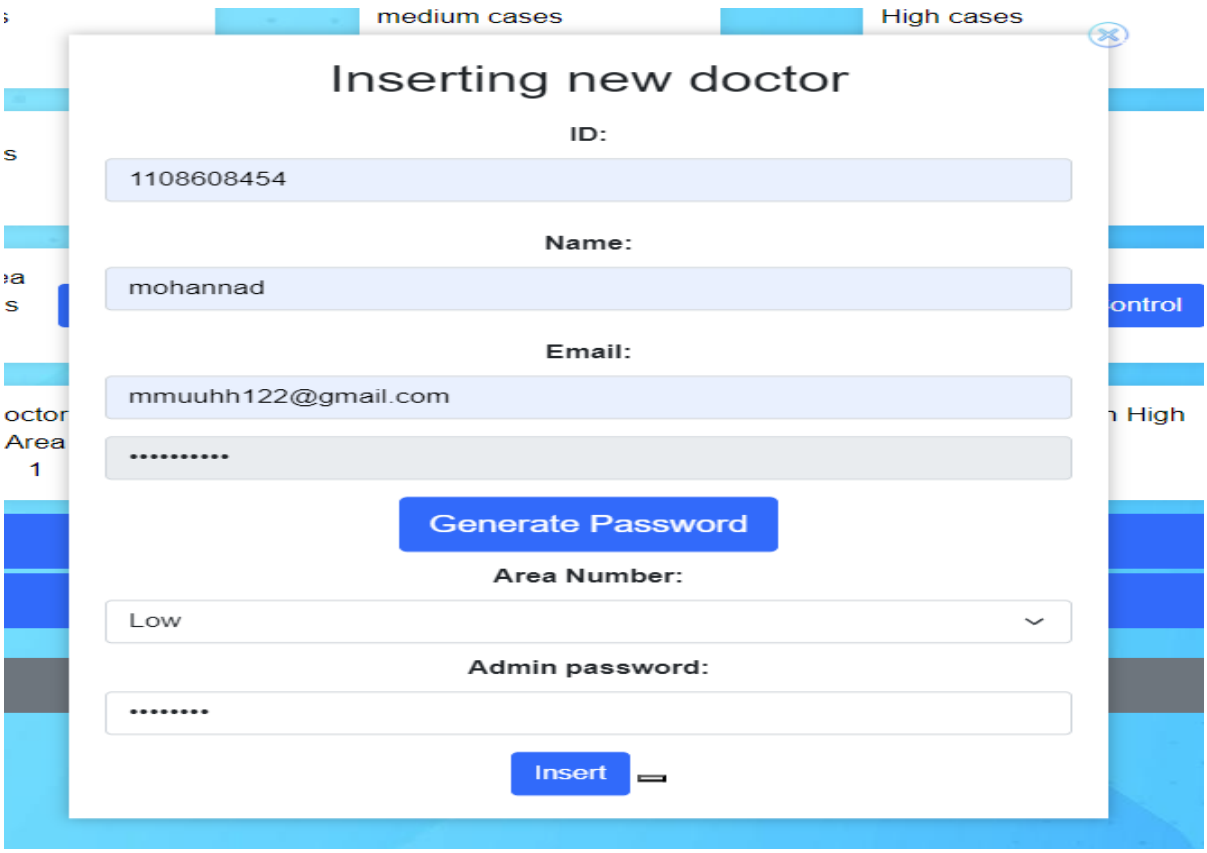




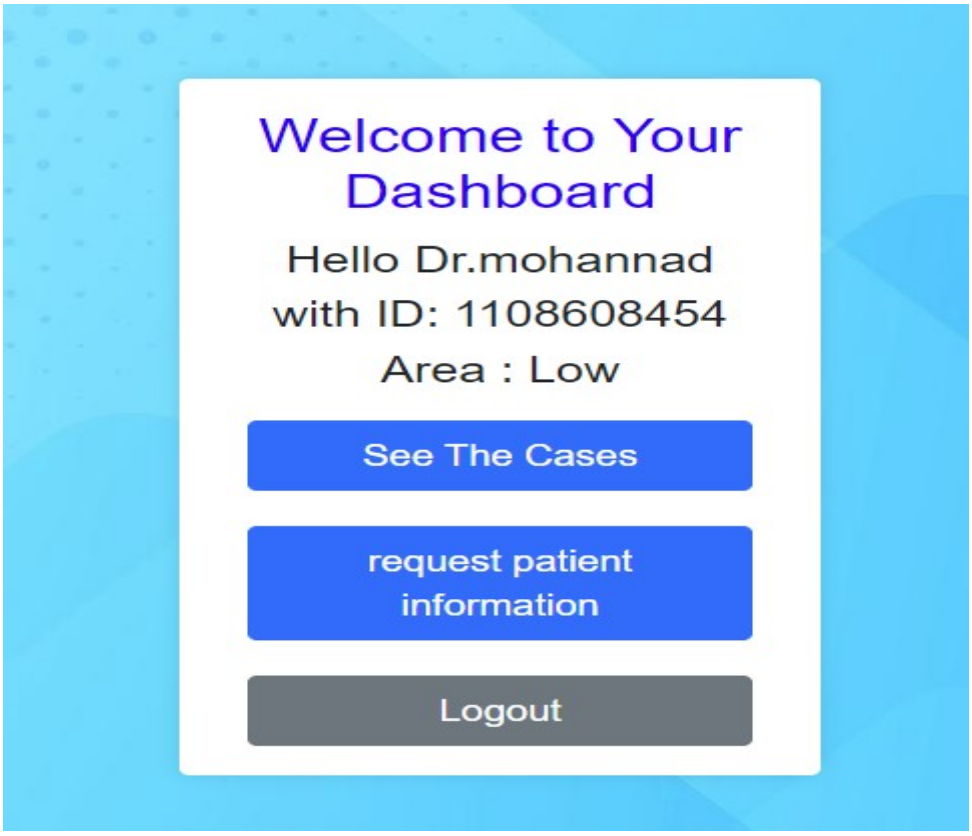
The admin has access to comprehensive information regarding the Emergency Department (ED).



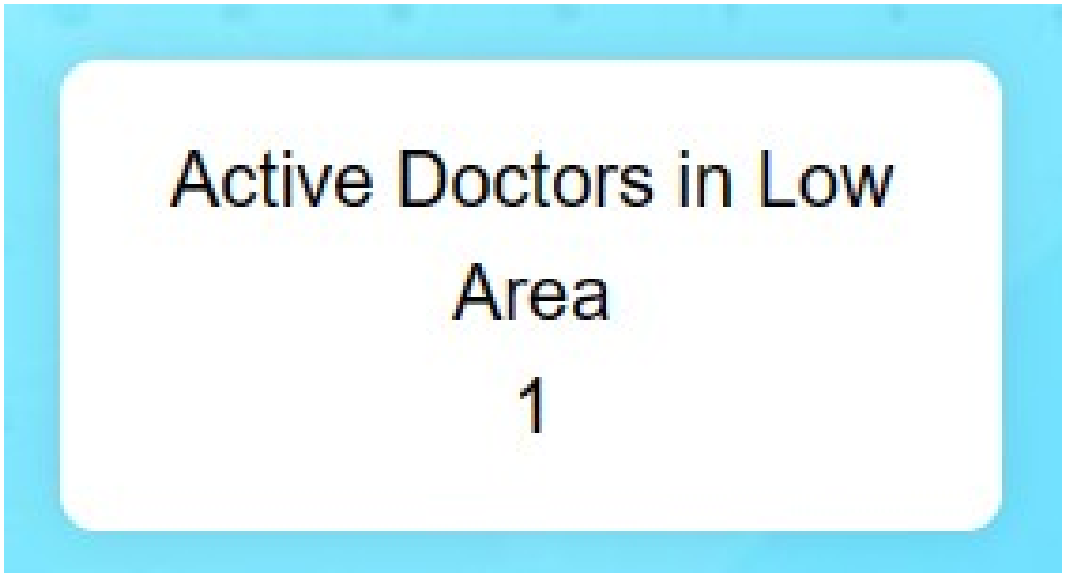
Here, the ability to insert a new doctor in database with a randomly generated password that will be sent to the doctor's email as a security measure.



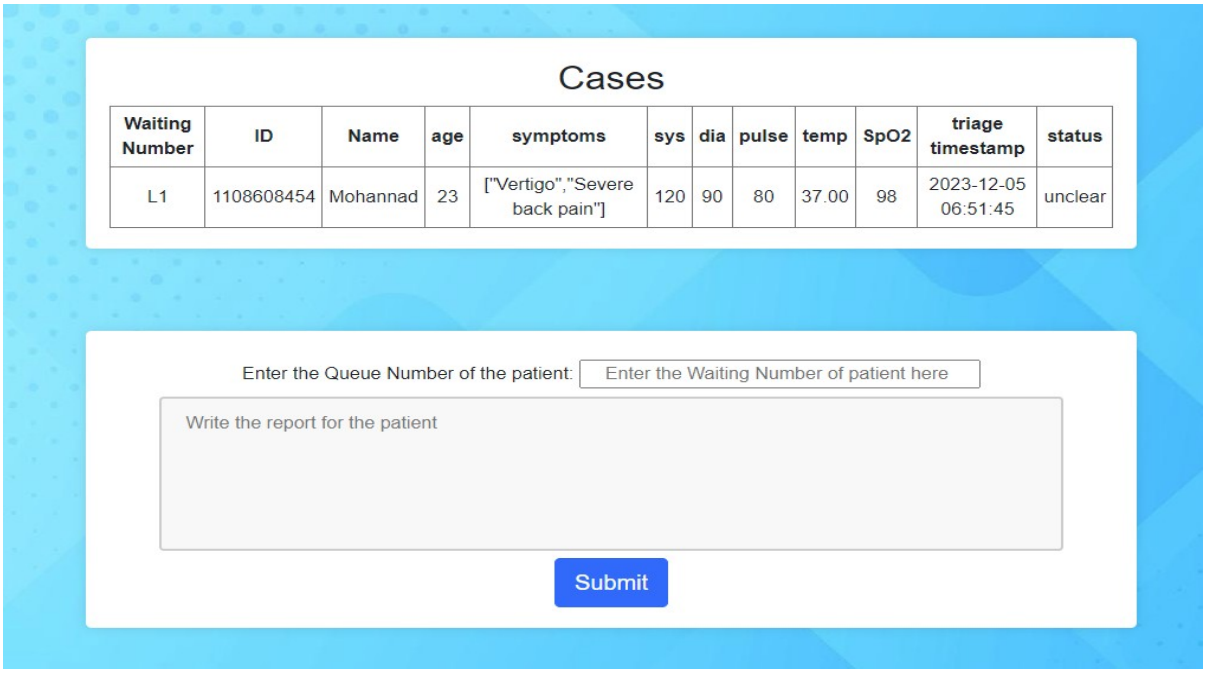
The doctor can access two functionalities: see the cases, requesting patient information.



Once a doctor logged in, the number of active doctors in admin dashboard will increase.



The doctor can access cases and assign reports about patients.



5.6 Results and Discussion

The testing phase of the self-triage and management system for emergency departments yielded positive results. The system effectively analyzed patient-provided information, including vital signs and symptoms, accurately identifying core symptoms and categorizing patients based on severity. The user interface was user-friendly, allowing for convenient data entry and navigation. Overall, the system demonstrated its ability to expedite the triage process, prioritize critical cases, and reduce waiting times in emergency departments. It also addressed overcrowding by preventing non-critical cases from occupying valuable resources, optimizing healthcare staff's focus on more urgent needs. The system's capability to provide simple prescriptions for non-emergency cases alleviated pressure on doctors and ensured basic medical care. However, improvements could be made by integrating additional data sources and expanding diagnostic capabilities for more comprehensive assessments and recommendations.

In conclusion, the implementation of the self-triage and management system showcased its potential to significantly enhance emergency patient care.

5.7 Summary

The self-triage and management system for emergency departments presented in this project offers a comprehensive solution to improve patient care and optimize resource utilization.

Future developments could focus on integrating additional data sources and expanding the system's diagnostic capabilities.

Chapter 6: Conclusion and Future Work

6.1 Conclusion

The emergency department experiences the highest level of strain within hospitals as it serves as the primary point for receiving and categorizing patients and the best way to increase emergency department efficiency is to automate the triage process by using Self-Triage and management system.

Self-Triage and emergency management system is developed to avoid ED crowding and to make a control on patients and doctors in the department.

6.2 Goals Achieved

We have developed many functions that serve Self-Triage and emergency management system to improve ED efficiency, so we have set some goals to achieve these functions.

the main goals that we have achieved is:

Patient goals:

- ✓ Identify patients.
- ✓ Automate the triage process.
- ✓ Categorizing patients according to their severity level.
- ✓ Distribute patients to an appropriate area according to their severity level.

Admin goals:

- ✓ View ED dashboard.
- ✓ Control the size of each Area.
- ✓ Add patients and doctors.
- ✓ View patients' history.

Doctor goals:

- ✓ Recording report on patient's state.
- ✓ View patients' history.

By achieving Those goals, system users can use all the functions that exist in the system.

6.3 Limitations

This system was proposed although being some limitations in this area, the most important of which are:

Error rate occurs by devices: Errors in accurately assessing vital signs during self-triage have the potential to significantly impact an individual's placement within a queue or the prioritization system. When vital signs such as heart rate, blood pressure, respiratory rate, and temperature are misread or misinterpreted, the resulting misinformation may lead to an incorrect estimate of the severity or urgency of a health concern.

Unexpected cases: Numerous real health conditions exist, and while our system addresses several, it's unable to comprehend every case presently. However, there is potential for future development to encompass a broader range of conditions.

Privacy Concerns: While our system ensures security for administrators and doctors, there's a gap in safeguarding patient privacy concerning their personal data. Encrypting the data stands as the solution, but due to time constraints, this issue remains unresolved.

6.4 Future Work

In the future, we will improve our system by improving the functionality and giving

additional features as follows:

AI-driven Decision Support Tools: Develop additional tools or modules within the system that provide healthcare practitioners with decision support, aiding in diagnosis or treatment recommendations based on triage assessments.

Security and Compliance Upgrades: While our system currently maintains strong security measures for administrators and doctors, there is a need to enhance security specifically concerning patient records.

User Experience (UX) Optimization: extensive user feedback sessions and usability studies to refine the system's interface and workflow. Enhancements based on user input can significantly improve the overall user experience.

References

[1] Canadian Journal of Emergency Medicine, https://www.cambridge.org/core/journals/canadian-journal-of-emergency-medicine/article/revisions-to-the-canadian-emergency-department-triage-and-acuity-scale-ctas-guidelines-2016/E2CB3E2063C54E11259313FA4FEAE495 , accessed (2023/4/3)

[2] validity study of the rapid emergency Triage and treatment system, 2021, https://sjtrem.biomedcentral.com/articles/10.1186/s13049-021-00832-2 , accessed (2023/4/5).

[3] KKESH Emergency,https://www.kkesh.med.sa/en-us/Professionals/Patient-Care-Departments/Emergency-Room, accessed (2023/5/5).

[4] KFSH&RC, https://www.kfshrc.edu.sa/en/home/hospitals/jeddah/emergencymedicinej , accessed (2023/5/5).

[5] KAFU, https://www.iau.edu.sa/en/university-hospitals/king-fahd-hospital-of-the-university/medical-services/emergency-room , accessed (2023/5/6).