Technical Documentation Requirement

Group No. 38

Early Edema Detection and its Solution using Deep Learning

BS Computer Science, Batch 2020S

Supervised by:
Dr. Muhammad Asif
Professor

Co-Supervised by:
Dr. Waleej Haider
Assistant Professor

Submitted by:

2020S-CS -097	**Muhammad Shayan**
2020S-CS-121	**Rose Ulfat**
2020S-CS-123	**Syeda Khizran Fatima**
2020S-CS-132	**Ilma Ameen**

Department of Computer Science & Information Technology
Sir Syed University of Engineering & Technology
University Road, Karachi 75300
http://www.ssuet.edu.pk

Chapter 1

Introduction

1.1 Project Background

The Area from where we have choose our project is from the field of medicine and the area of medical imaging and healthcare is constantly advancing with the help of technological advancements. Early detection of medical conditions is essential for effective treatment and improved patient outcomes. Edema, a medical condition characterized by the accumulation of fluid in the body's tissues, can lead to serious complications if left untreated. Therefore, early detection of edema is crucial for timely intervention and treatment.

The aim of our project is to develop a deep learning-based solution for the early detection of edema. The solution will be based on the analysis of medical images, such as X-rays, CT scans, or MRI scans, to detect early signs of edema. Deep learning algorithms, such as convolutional neural networks (CNNs), will be used to analyze these images and identify patterns that indicate the presence of edema.The solution will involve several stages, including the preprocessing of medical images, the development and training of the deep learning model, and the evaluation of the model's performance using a set of validation data. The project will also explore the use of transfer learning techniques to improve the performance of the model and reduce the amount of data required for training. The ultimate goal of this project is to develop an accurate and reliable deep learning-based solution for the early detection of edema, which can be integrated into clinical practice to improve patient outcomes and reduce healthcare costs.

1.2 Project Objectives

The objective is to develop a robust and accurate system application for automatic detection of edema regions in using image processing. The system should be able to accurately segment the edema regions and provide a quantitative measure of the extent and severity of edema type. The successful completion of the project will provide significant benefits for both clinicians and patients. Clinicians will be able to make more accurate and timely diagnoses, leading to improved treatment decisions and better patient outcomes. Patients will benefit from more personalized treatment plans and potentially reduced healthcare costs.

- The objective is to replace the manual method of analyzing the edema with an automated model based on deep learning.
- The desktop and mobile application will be developed for the practical use of our model.
- It will be a time and cost-efficient application.

1.3 Problem Statement

There is no conventional automated method in today's healthcare for diagnosing Edema. Edema is a medical condition that occurs when excess fluid builds up in the body's tissues, causing swelling and inflammation. Edema can be caused by a variety of factors, including heart failure, kidney disease, liver disease, and chronic venous insufficiency. Early detection allows for prompt intervention and treatment, which can prevent the condition from worsening and potentially causing more serious health problems. It will also help peoples living in rural areas to examine their selves through this application. Also, this application helps the doctors to examine the disease in the early stage without going through a lot of difficult procedure.

1.4 Product Scope

The product aims to develop an edema detection system using deep learning techniques. The system will analyze medical images and identify the presence of edema accurately. The scope of the product includes developing a user-friendly interface for inputting medical images, processing the images using deep learning algorithms to detect edema, and presenting the results in a clear and concise manner. The system will be designed to work with different types of medical images, including X-rays, CT scans, and MRI images. The goal is to provide healthcare professionals with a reliable tool for detecting edema quickly and accurately, which can improve patient outcomes and ultimately save lives.

Chapter 2

Literature Review

2.1 Existing Systems

Existing Systems Since no prior work in making an application having a deep learning model implemented in it to detect edema is done yet therefore, we cannot compare our solution with any application, but it could be compared with existing proposed deep learning models such as:

2.1.1 Title: Using deep learning models to analyze the cerebral edema complication caused by radiotherapy in patients with intracranial tumor. [1]

Description:

In this paper, researchers, due to GPU memory limitations, the 2D U-Net model can process complete slices at one time, while the 3D convolution system can only process small blocks that cover a small part of the 3D volume. Therefore, a 2D-based network is used. In addition, they proposed a fully automated brain tumor segmentation method, which was developed using a deep convolutional network based on U-Net. Their method was evaluated on Multimodal Brain Tumor Image Segmentation (BRATS 2015) dataset. The results of cross-validation show that the method can effectively obtain promising segmentation. Compared with our study, this result verifies that similar methods have the same excellent results. We still hope to have more cases in future studies, so that the explanation of the complications of cerebral edema caused by radiotherapy in patients with intracranial tumors can be clearer.

2.1.2 Title: SwellFit: Developing A Wearable Sensor for Monitoring Peripheral Edema. [2]

Description:

Peripheral edema is a swelling of the legs, feet, or hands due to the accumulation of excessive fluid in the tissues.

As a systematic approach to assessing peripheral edema, we develop SwellFit, an experimental prototype of a novel wearable technology that monitors peripheral edema by tracking changes in ankle curvature. Through a series of proof-of-concept experiments, we demonstrate that SwellFit detects ankle swelling even in the presence of substantial noise in the raw sensor readings.

2.1.3 Title: Diabetic Macular Edema Detection. [3]

Description:

Diabetic macular edema (DME) is a common cause of vision impairment and blindness in patients with diabetes. We developed an end-to-end deep fusion model for DME classification and hard exudate (HE) detection. Based on the architecture of fusion model, we also applied a dual model which included an independent classifier and object detector to perform these two tasks separately. We used 35,001 annotated fundus images from three hospitals between 2007 and 2018 in Taiwan to create a private dataset. Clinically, this system could be applied to diabetic eye screening to improve the interpretation of fundus imaging in patients with DME.

2.2 Comparative Analysis:

Table 2-1 Comparative Analysis

FEATURES	CEREBRAL EDEMA	SWELLFIT	Diabetic Macular Edema Detection	EDEMA SENSE
Web Application				
CT-SCANS				
MRI-SCANS				
Dataset Conversion in 2D				
Convenience				
Time Efficient				
Mobile Application				

Chapter 3

Requirement Analysis

3.1 External Interface Requirements

3.1.1 User Interfaces

We are designing a user-friendly and intuitive interface using Flutter's widgets and UI components. It ensures responsiveness and adaptability across different screen sizes and orientations. Imp We can implement navigation between screens and components using Flutter's navigation framework., it works with data-driven applications with the algorithms of machine learning and deep learning which our use case is. This application is user-friendly and very easy to use as it will be used by public, doctors and health workers who are not as familiar with complex user interfaces so a basic design with prominent useable features will be deployed.

3.1.2 Hardware Interfaces

- Hard Disk: 1 GB or more to let the model run efficiently without any delay.
- RAM: 8 GB or more and SSD is recommended for quick results and less memory consumption.
- Basic computer components (mouse, keyboard, and a digital visualization screen)

3.1.3 Software Interfaces

The product is a cross platform application built on python programming language and for development Anaconda's Jupyter Notebook is used with different libraries and packages of python for model training Google Collaboratory is used for its built-in GPUs feature. The minimum operating system required for the development is Windows 7 and to host this application python framework streamlit is used.

The following are the software interfaces used:

- Anaconda

• Jupyter Notebook

• Flutter

• Tensorflow

• NumPy

• Matplotlib

Operating System

• Windows 7 onwards

Programming Language

• Python

3.1.4 Communications Interfaces

There are no special communication interface requirements.

3.2 Functional Requirements

ID	Requirement
REQ-1	To detect and analyze edema in medical images from various sources, including X-rays, CT scans, and MRIs.
REQ-2	Software is Compatible with multiple operating systems, including Windows, macOS, and Linux.
REQ-3	Software should be a user-friendly interface that allows users to upload, process, and view edema detection results for medical images.
REQ-4	The software shall provide real-time feedback on the edema detection process.
REQ-5	The software shall be able to run on both desktop and mobile devices, including tablets and smartphones.

3.3 Other Nonfunctional Requirements

3.3.1 Performance Requirements

The information is refreshed depending on whether some updates have occurred in the application. The system shall respond to the member in not less than five seconds from the time of the request submittal. The system shall be allowed to take more time when doing large processing jobs. Responses to view information shall take no longer than 10 seconds to appear on the screen.

- It must not crash.
- It should load in less than a minute.
- It should work very fast without any buffer.

3.3.2 Safety Requirements

- The data (MRI,X-Rays, CT) scanned by the system will not be saved anywhere within the system. Neither in external storage nor in the cloud.
- The system will only store the history of diagnoses and results obtained through the scans.

3.3.3 Security Requirements

- Application may require users to create accounts to access applications that store
- information and display profiles. A security system typically grants access to accounts
- when users enter the correct username and password.
- Application may not grant access until the user creates a strong password. For example, a
- strong password might contain a certain number of characters and a capital letter.
- The behavior of the software must be correct and predictable.
- The software must be available and behave reliably even under any attacks.

- The software must ensure the integrity of the customer account information.

3.3.4 Software Quality Attributes

Reliability

The system has to be 100% reliable due to the importance of data and the damage caused by incorrect or incomplete data scanning. The system will run 7 days a week, 24 hours a day.

Portability

The probability is that the Software should perform effectively in every environment.

Availability

The system is available for the user 24 hours a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week. In case of any failure or server shutdown, the system should have an up-to-date backup available.

Usability

The user should easily navigate its interface. The system is user-friendly, which makes the system easy.

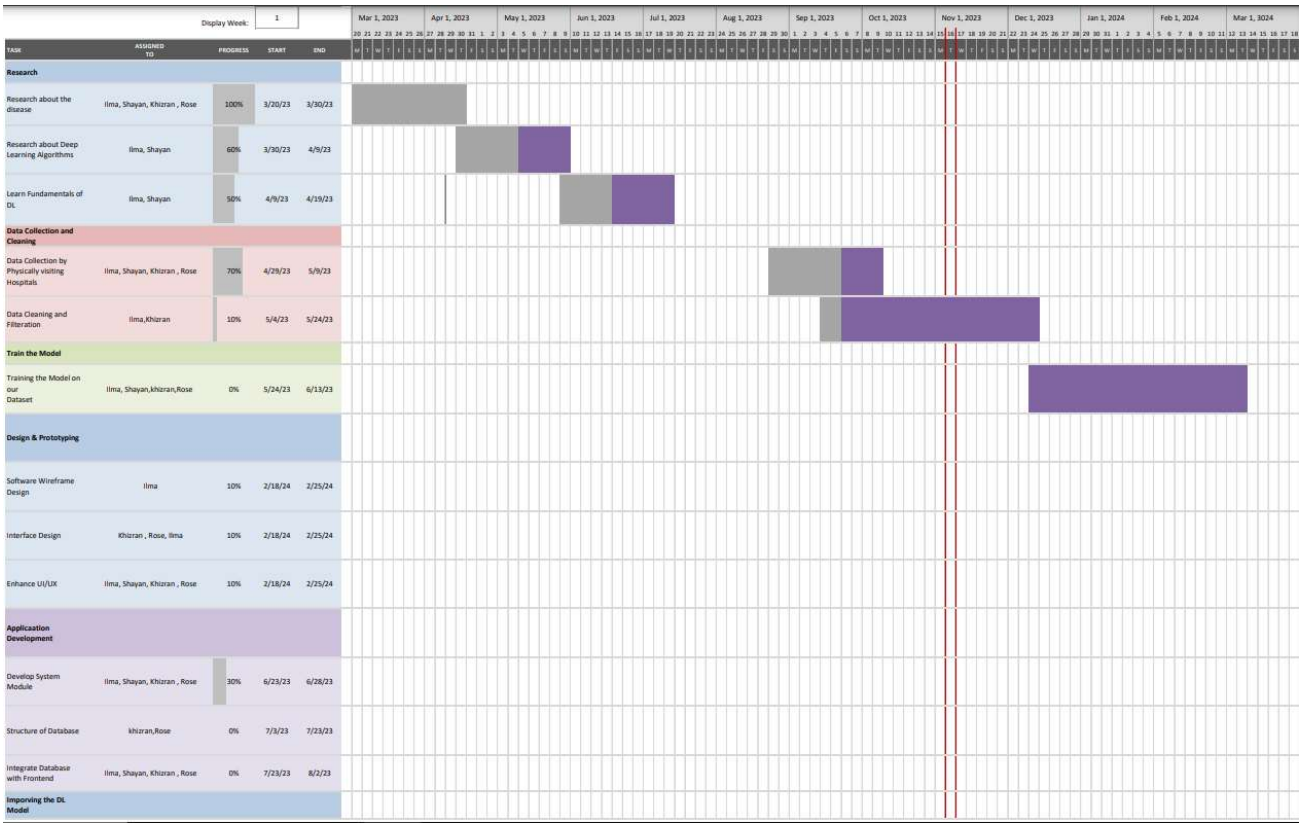
Timeless

The system will carry out all the operations in a short time.

3.4 Cost Estimation

S.No	Project Expenditure	Cost in Rupees
1	**Software Tools and API**	
	• Anaconda • Jupyter Notebook • Pandas • NumPy • Tensorflow • Matplotlib • Streamlit	**Sub Total: Rs. None**
	Sub Total	**None**
2	**Domain Name and Hosting & Printing**	
	Domain Name	2000
	Hosting	8000
	Printing Cost	1000
	Sub Total	**11000**
3	**Hardware and Networking Cost**	
	• GTX 1660 Super	50000
	• Power Supply 300w	5000
	• Developing cost and other	50000
	Sub Total	**105,000**
	Grand Total	**116,000**

3.5 Project Plan with Gantt Chart

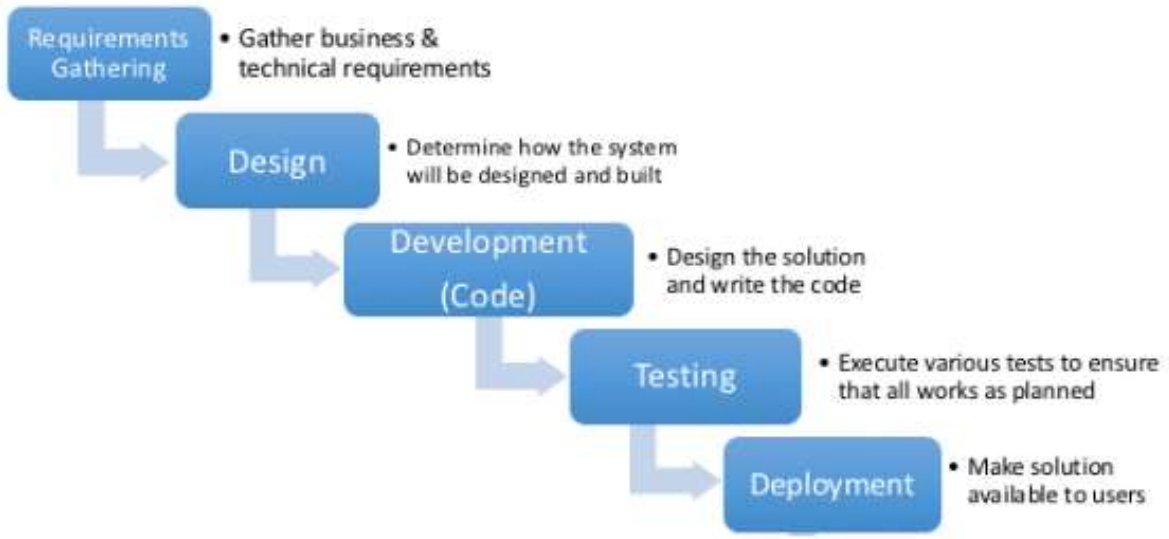


3.5-Project Plan with Gantt Chart

3.6 Software Development Life Cycle Model

3.6.1 Waterfall Methodology

The waterfall project management methodology lets us plan out our project in a linear manner where each subsequent phase is initiated after the last one ends. It's one of the most straightforward ways to manage a project and is the best suitable choice since we already have clearly outlined objectives and still have sufficient time for planning because deadlines are involved.



Software Development Life Cycle Model

Chapter 4

System Design

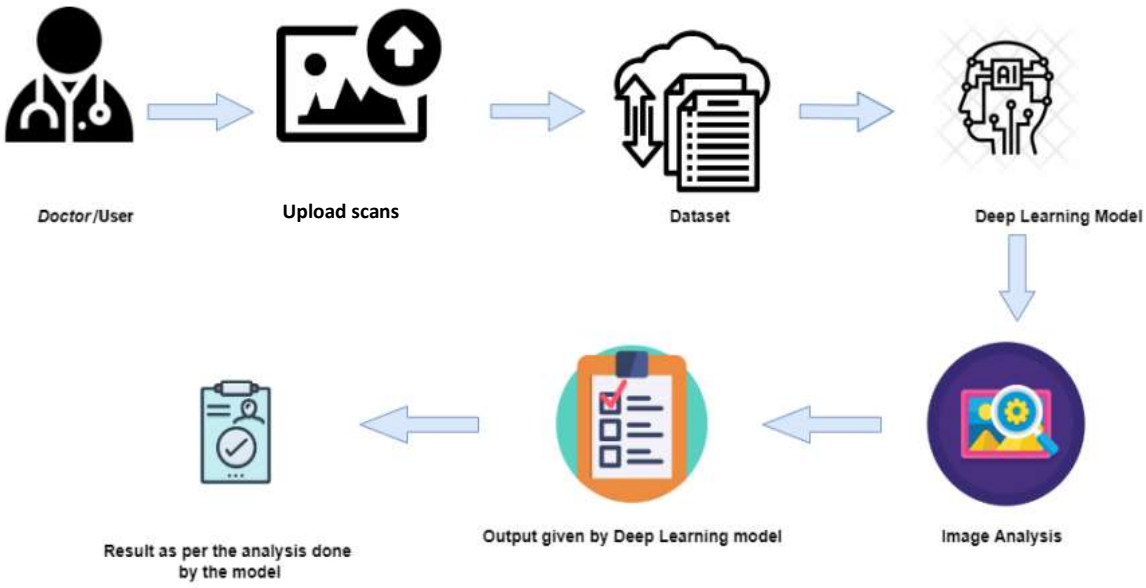


Figure 4-1 System Architecture Diagram

This architecture model shows the blueprint of our desktop/mobile application, which is quite visible, doctor/user is interacting with the upload feature for CT/MRI/Chest X-ray scans then that scan will reach in AI model for analysis. The next step is Image Analysis where the Segmentation will be performed. Now the Dice Score (Accuracy) will be tested. After that output based on AI model findings will be shown and on basis of that output user doctors get the result also they suggest the best doctor for the particular disease and type treatment.

4.2 System Operations

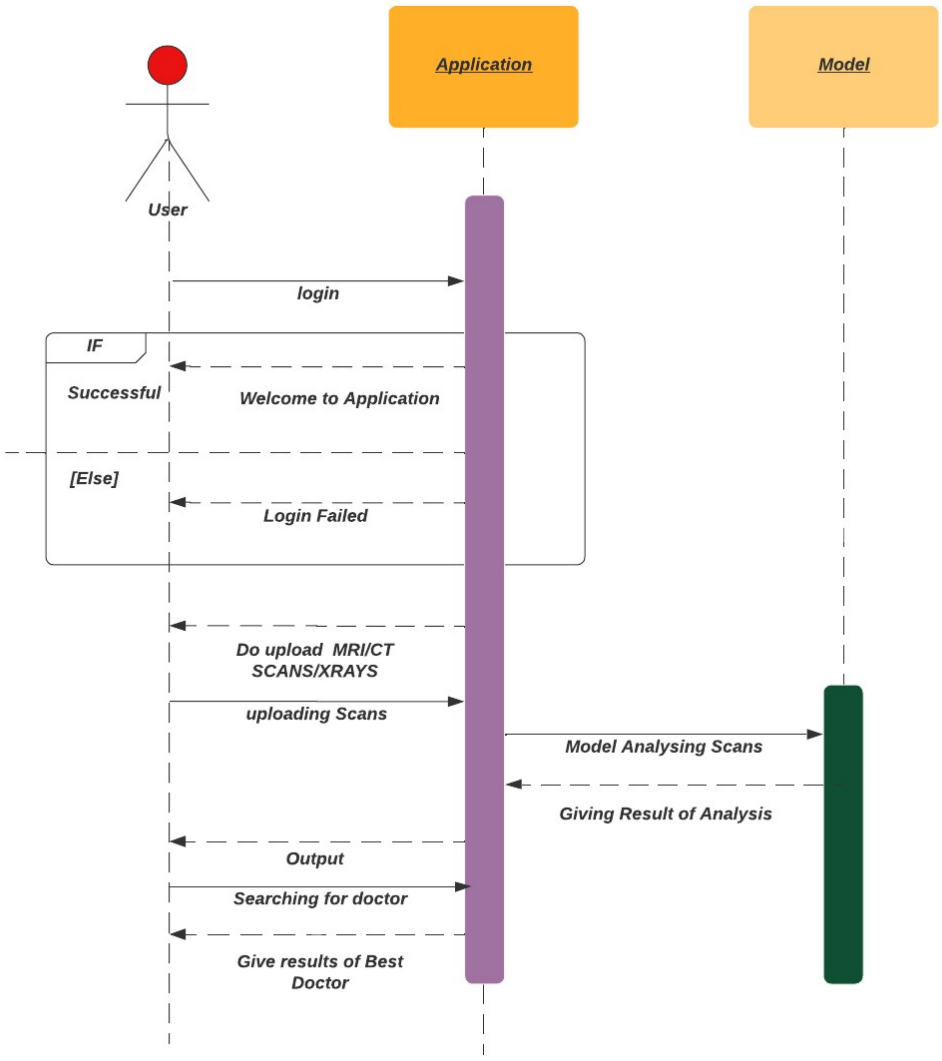


Figure 4-2 Sequence Diagram

In this sequence diagram every activity happening in the project will be demonstrated by a sequence that states first the login activity with an alternate if statement after successful login user will be able to upload

the MRI/CT/chest Xray scans which will be analyzed in the model and the output will be shown to the user and then show the suggestion for doctor. This is how the whole application will work in sequence.

4.3 System Model

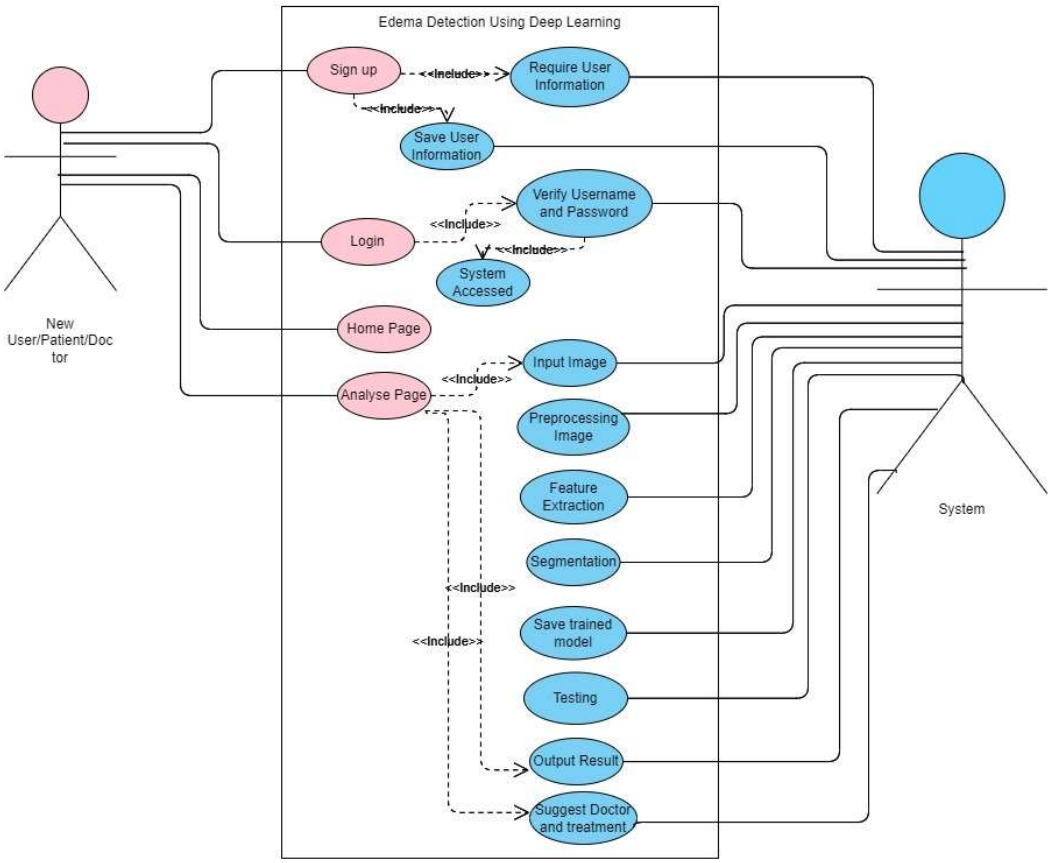


Figure 4-3 UML Diagram

In this UML use case diagram, the doctor is an actor who is interacting with different activities and on other side system is another actor who's performing the task. The first activity is the log-in the user will log in and then he/she can upload the MRI/CT/chest X-ray scans which will be analyzed by the deep learning model. The next step is Image Analysis and Preprocessing Image where the Segmentation will be performed after that system saved the trained model and perform testing then after that user will have the

output of the result coming from the analysis of the model and suggest doctor which suite best for the treatment of particular result.

3.4 Object Model

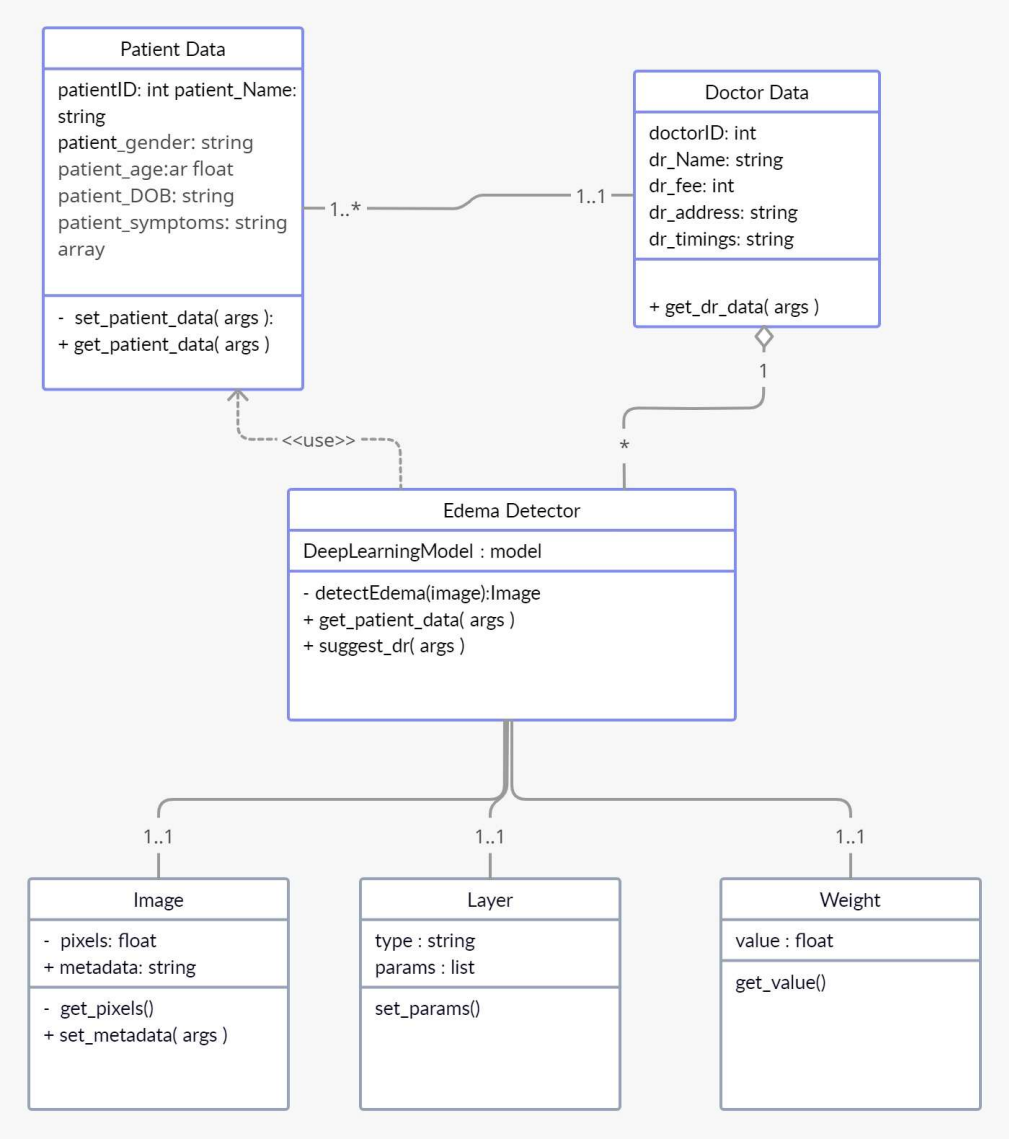


Figure 4-4 Object Model

The class diagram shows the four main classes in the system: Patient, Doctor, EdemaDetectionModel, and Edema_Snapshot. These classes represent the core entities and functionality of the system. The Patient class represents a patient in the system and contains

information such as their patient_id, patient_name, edema_status, and the doctor_id they are assigned to. The Patient class also has a method for creating a new Edema_Snapshot object, which is used to store data about the patient's edema status at a specific point in time. The Doctor class represents a doctor in the system and contains information such as their doctor_id, doctor_name, specialty, and location. The Doctor class has methods for getting a list of doctors with a specific specialty, as well as retrieving a doctor's information based on their doctor_id. The EdemaDetectionModel class represents the deep learning models used in the system for edema detection and doctor suggestion. It has a method for suggesting a doctor based on a given Edema_Snapshot object, which takes into account the patient's edema status, the doctor's specialty, and their availability. The Edema_Snapshot class represents a snapshot of a patient's edema status at a specific point in time. It contains information such as the snapshot_id, edema_percentage, timestamp, and model_id, as well as the patient_id it is associated with.

4.5 Data Model

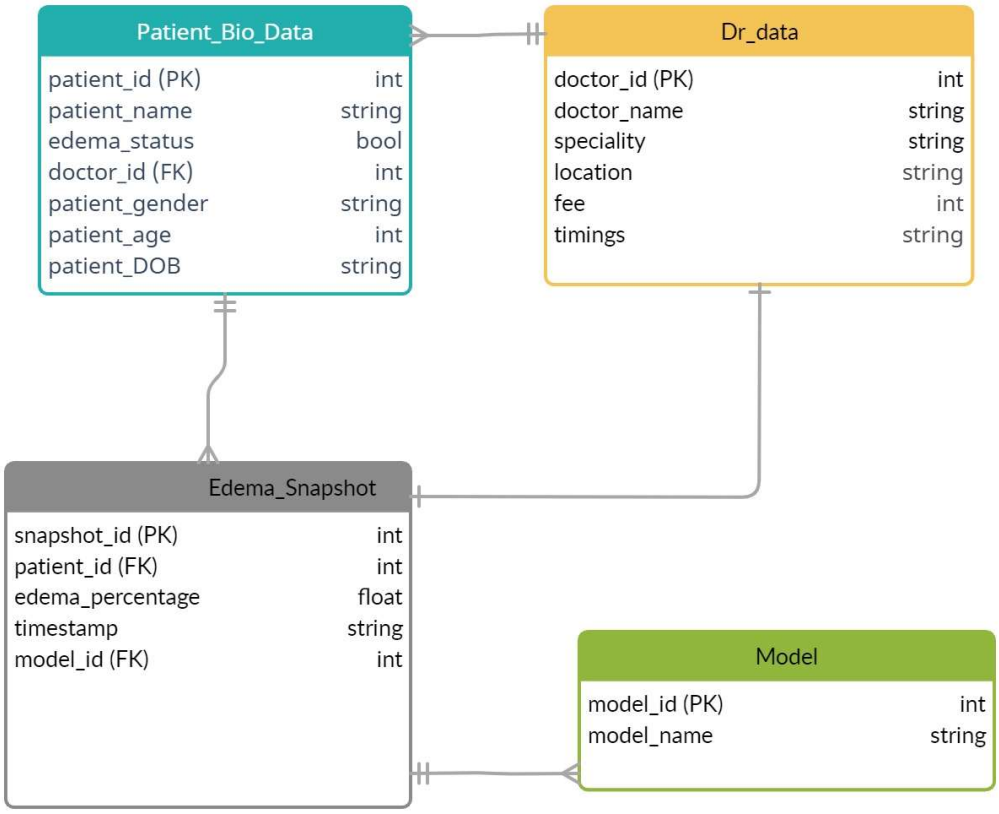


Figure 4-5 Data Model

The database diagram shows the main entities and relationships in the system, which are stored in a relational database. There are six main tables: Patient, Doctor, Specialty, Edema_Snapshot, Model, and Snapshot_Doctor. The Patient table stores information about the patients in the system, including their patient_id, patient_name, and edema_status. Each patient can be assigned to one doctor, which is represented by a foreign key doctor_id that references the Doctor table. The Doctor table stores information about the doctors in the system, including their doctor_id, doctor_name, location, and specialty_id. Each doctor has a specialty, which is represented by a foreign key specialty_id that references the Specialty table.The Specialty table stores information

about the different specialties available in the system, including the specialty_id and specialty_name. The Edema_Snapshot table stores information about the edema status of each patient at a specific point in time, including the snapshot_id, patient_id, edema_percentage, timestamp, and model_id. Each snapshot is associated with a patient, which is represented by a foreign key patient_id that references the Patient table. Each snapshot is also associated with a model, which is represented by a foreign key model_id that references the Model table. The Model table stores information about the different deep learning models available in the system, including the model_id and model_name. The Snapshot_Doctor table stores information about the doctor assigned to each snapshot, including the snapshot_id and doctor_id. This is a many-to-many relationship, as each snapshot can be assigned to multiple doctors, and each doctor can be assigned to multiple snapshots.

4.6 User Interface Design

The following is the wireframe workflow of Edema Detection. The user interface is visible, Flexible, and simple enough for the user to learn.