COMSATS University, Islamabad Pakistan

Detection and Prediction of COVID-19

By

Muhammad Sajjad Ullah CUI/FA18-BCS-067/ISB

Muhammad Tayyab Paracha CUI/FA18-BCS-118/ISB

Supervisor

Mr. Muhammad Rashid Mukhtar

Co-Supervisor

Mr. Umar Nauman

Bachelor of Science in Computer Science (2018-2022)

The candidate confirms that the work submitted is their own and appropriate credit has been given where reference has been made to the work of others.

COMSATS University, Islamabad Pakistan

Detection and Prediction of COVID-19

A project presented to

COMSATS University, Islamabad

In partial fulfillment

of the requirement for the degree of

Bachelor of Science in Computer Science (2018-2022)

By

Muhammad Sajjad Ullah CUI/FA18-BCS-067/ISB

Muhammad Tayyab Paracha CUI/FA18-BCS-118/ISB

DECLARATION

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other, we will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Muhammad Sajjad Ullah   Muhammad Tayyab Paracha

--------------------------   --------------------------

CERTIFICATE OF APPROVAL

It is to certify that the final year project of BS (CS) “Detection and Prediction of Covid-19” was developed by **Muhammad Sajjad Ullah (CIIT/FA18-BCS-067)** and **Muhammad Tayyab Paracha (CIIT/FA18-BCS-118)** under the supervision of **“Mr. Muhammad Rashid Mukhtar”** and co supervisor **“Mr. Umar Nauman”** and that in their opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Sciences / Software Engineering.

Supervisor

External Examiner

Head of Department
(Department of Computer Science)

Executive Summary

As we all know, the biggest problem now a day in the world is COVID-19. The main issue is that the symptoms are identical to symptoms of flu. Both COVID-19 and flu can have varying degrees of signs and symptoms, ranging from no symptoms (asymptomatic) to severe symptoms. If a person has Covid and he does not take it seriously, he can end up in emergency condition. Many people mix both diseases and think they are suffering from normal fever or flu. Another problem is that doctors cannot also tell accurately just by checking chest x-rays and HRCT because the TB-patient's chest x-rays and Covid infected chest x-rays look similar. So just using chest x-rays or HRCT to see whether the patient has Covid can leads to the wrong result. So, we will also need the patient medical history and other blood reports for efficient results. For doctors, it is difficult for considering all the factors like medical history, report analysis, symptoms, chest X-Rays, HRCT of chests to tell accurately whether the patient is covid positive or not. The proper visualization of disease is also another problem. The reason why I am developing this system is to eliminate difficulties in prediction, suggestion, and visualization. COVID-19. This will help me enhance many skills such as problem-solving, image processing, 3D visualization, and grip on the web and mobile app development.

To the problem I described in the above section, I developed a system in which there is an AI-based Doctor it is a chatbot that will ask questions from a patient.. It includes the patient medical background such as name, sex, other diseases from which the patient is suffering in the past and present, and symptoms of the disease. After that, we will do image processing on chest x-rays and HRCT (uploaded by user) using different ML Algorithms. After that we will do a reports analysis and considering all the above things, we will predict the COVID-19 and severity of the disease. After predicting we will suggest a precaution to the patient according to the condition of the disease. So, the patient can overcome the disease by following these suggestions. This system also will make a 3D model of the lungs for the visualization of the disease. The COVID infected areas will be highlighted in the 3D model. The system consists of web applications and smartphone applications.

Acknowledgement

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor Mr. Muhammad Rashid Mukhtar and our Co-Supervisor Mr. Umar Nauman. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are grateful to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us with the values of honesty & hard work.

And we are also thankful to our batchmate Nabiha Zubair for helping us in our project.

Muhammad Sajjad Ullah

Muhammad Tayyab Paracha

Table of Contents

List of Figures

1 Introduction

Coronavirus Disease (COVID-19) is a respiratory disease that spreads from person to person. It has been more than a year but still, this disease is spreading and causing many deaths all around the world. Many scientists believe that this disease will last with us for a long time. The issue with this disease is that if you do not get the proper diet and care at the start of the disease, then it can be fatal for you. Many people do not even know that they have COVID-19 but they get into an emergency due to oxygen shortage. People are unaware of their medical conditions and don't give proper care to their health. Many people are so careless they do not even go to doctors even after showing symptoms. Our system will consist of a web application and a mobile application. The purpose of our system is to predict COVID-19 in our users using their chest x-rays, HRCT by considering other factors such as patient medical history, blood reports. After predicting the disease, the system will be able to suggest the necessary measures which should be taken depending on the severity of the disease. The system will also show the 3D visualization of the disease for better visuality of disease and the user can easily understand the condition of the lungs.

1.1 Vision Statement

For the People and Doctors who wants a seamless platform for the Prediction and Suggestions of Covid-19 of the patients. This will provide them with a better chance of diagnosing the disease and then suggest the patient on what to do next. The accuracy will be increased using multiple things such as Xray, HRCT, and other medical reports.

1.2 Related System Analysis/Literature Review

After studying Artificial intelligence and machine learning, we decided to do our final year project that would consist of a machine learning model and its application to real-world problems.

After deciding our major concern, we start working on the project ideas. Considering today's global problem, we decided to work on the prediction and suggestion of COVID-19. As many theories related to this exist but there is no such system exist that can predict and suggest the COVID-19.

1.3 Project Deliverables

The deliverables and developing requirements are providing a platform to let the users perform marketing with ought the hassle of time, effort, and money constraints. This application should provide a unified interface to market on many social media platforms, where the end user will only provide his target audience.

The deliverables and devolvement requirements of the project includes the following:

- Scope
- SRS
- SDS
- Final Report
- Presentation
- Source Code

1.4 System Limitations/Constraints

A dataset is a major factor on which the result depends so finding an accurate and good dataset may be difficult.

1.5 Tools and Technologies

Table 1: Tools and Technologies for the Targeted Project.

Tools And Technologies	**Tools**	**Version**	**Rationale**
	Jupyter Notebook	2020	IDE
	PyCharm	2021	IDE
	VS Code	2021	IDE
	MS Word	2019	Documentation
	MS PowerPoint	2019	Presentation
	MS Project	2019	Planning and scheduling
	Pencil	3.0.4	Mockup creation
	Technology	**Version**	**Rationale**
	Django/Flask	3.2.7/2.0.1	APIs

	Python	3.9.5	Programming language
	Firebase	4.2	DBMS
	Cloud		DBMS
	JavaScript	ECMA Script 2018	Programming language
	HTML	V5	Markup Language
	CSS	V3	Styling Language
	React	17.0.1	For Web Frontend
	React Native	2.0.6	Mobile Application Development

1.6 Relevance to Course Modules

- In this project, a large amount of knowledge from the courses that we studied throughout the course have been used.
- The course Mobile Application Development is used in making our Mobile Application for the project.
- Web Technologies and Topics in Computer Science I helped us in making a website for our project.
- Machine Learning and Topics in Computer Science I have also helped us in developing a machine learning model that will help us in prediction.
- Software Engineering Concepts has enabled us in making the documentations necessary for the project.

2 Problem Definition

2.1 Problem Statement

As we all know, the biggest problem now a day in the world is COVID-19. The main issue is that the symptoms are identical to symptoms of flu. Both COVID-19 and flu can have varying degrees of signs and symptoms, ranging from no symptoms (asymptomatic) to severe symptoms. If a person has Covid and he does not take it seriously, he can end up in emergency condition. Many people mix both diseases and think they are suffering from normal fever or flu. Another problem is that doctors cannot also tell accurately just by checking chest x-rays and HRCT because the TB-patient's chest x-rays and Covid infected chest x-rays look similar. So just using chest x-rays or HRCT to see whether the patient has Covid can leads to the wrong result. So, we will also need the patient medical history and other blood reports for efficient results. For doctors, it is difficult for considering all the factors like medical history, report analysis, symptoms, chest X-Rays, HRCT of chests to tell accurately whether the patient is covid positive or not. The proper visualization of disease is also another problem. The reason why I am developing this system is to eliminate difficulties in prediction, suggestion, and visualization. COVID-19. This will help me enhance many skills such as problem-solving, image processing, 3D visualization, and grip on the web and mobile app development.

2.2 Problem Solution

To the problem I described in the above section, I proposed a system in which there will be an AI-based Doctor it is a chatbot that will ask questions from a patient through NLP. It includes the patient medical background such as name, sex, other diseases from which the patient is suffering in the past and present, and symptoms of the disease. After that, we will do image processing on chest x-rays and HRCT (uploaded by user) using different ML Algorithms. After that we will do a reports analysis and considering all the above things, we will predict the COVID-19 and severity of the disease. After predicting we will suggest a precaution to the patient according to the condition of the disease. So, the patient can overcome the disease by following these suggestions. This system also will make a 3D model of the lungs for the visualization of the disease. The COVID infected areas will be highlighted in the 3D model. The system consists of web applications and smartphone applications.

2.3 Objectives of the Proposed System

- This system can also help any hospital to use it for accurate results.
- The main advantage of this system is no need to go to the doctor because a user can detect and predict the severity of disease using this system.
- The system will suggest the cure to the patient according to the severity of the disease.
- There will be a 3D model so the disease can easily be visualized.
- Easy to use, there will be a web and smartphone application.

2.4 Scope

It is an end-user product. In this system, the focus is on the accurate prediction of COVID-19 and after prediction the proper suggestion to the patient according to the severity of the disease and the 3D visualization of the disease. The system consists of the AI-based doctor which will ask some questions from the user related to the medical background of the user and the symptoms which the user has at the current time. According to this, the system will generate a user profile and after that, we will do the image processing on the chest x-rays and HRCT uploaded by the user. We will also do report analyses like blood reports, x-ray reports and HRCT reports. The main purpose of considering all the things is to accurately predict the disease because if we only consider the images like chest x-rays then it may lead to wrong results. After all, TB patient and COVID patient has the same x-ray because in both diseases the lungs are affected. When the disease is predicted then the system will suggest a cure according to the severity of the disease. After this, the 3D model of the lungs will be generated on which the disease and infected parts of the lungs are highlighted so the user can easily visualize the disease.

2.5 Modules

2.5.1 Module 1: AI-Based Chatbot

This is the major module of the system. The concept of using an AI-based chatbot is to get an answer to some questions from a user in an interactive way. These questions consist of the user's medical history background and current symptoms the user has. An AI-based chatbot is like a doctor which will ask questions from the users using NLP. In this way, we will get the patient's history and current symptoms and this information will also be used to predict the disease.

2.5.2 Module 2: Image Processing

This is the major module of the system. As we are also using chest x-rays and HRCT for the prediction of the disease, so this module will handle the uploading of images to the system and then perform dynamic segmentation according to our needs. The output of this module will be an image that we will use for a future process to predict the disease.

2.5.3 Module 3: Reports Analysis

For precise results, we also need patient reports such as x-ray report, HRCT report, and blood reports. By using only chest x-ray and HRCT images we cannot predict accurately, so we will also need the blood reports and x-ray reports to accurately predict the disease. We will do report analysis by image processing or by taking things from users in the type of form. So, these report results will also be used in the prediction of the disease.

2.5.4 Module 4: Detection and Prediction of COVID-19

This is the major module of the system. After the image processing and reports analysis and the patient history and symptoms taken from a user through a chatbot. This module will take all these results and then we will apply machine learning algorithms and checked them using our trained model to accurately detect and predict the COVID-19 and severity of the disease.

2.5.5 Module 5: Suggestion to overcome the disease

After the detection and prediction, the system will suggest the proper precautions to overcome the current situation of the disease. The system will suggest according to the severity and the impacts of the disease. The user can overcome the situation by following the suggestion given by our system. And if the condition is severe then our system will ask the user to go to the doctor for further recovery of the disease.

2.5.6 Module 6: 3D Visualization of Disease in the lungs

This is also the major module of the system. In this, we will generate a 3D model of the lungs on which the disease is visualized means that the infected part of the lungs will be highlighted so the user can easily see the infected parts of the lungs. In this way, the user can see the current situation of the lungs. As this system can be used by hospitals so it will be easy for doctors to visualize the disease.

2.5.7 Module 7: Notify the Patient

This module will handle sending the information to the patient. As this system can be used by the hospitals so if the doctor checks the patient through this system and want to send the information to the patient so by using this module the doctor can send the information to his patient. The doctor can also send the 3D Model to the patient using this module.

3 Requirement Analysis

3.1 User classes and characteristics

Table 2 User Classes

User class	Description
Doctor	The doctor will use this system for his assistance. The doctor will upload the HRCT and reports of the patient and the system will predict the Covid-19 and suggest the precautions according to the severity of the disease. The doctor can also see the 3D Model of the lungs on which the infected areas will be

	highlighted. The doctor can also check his patients’ records.
Patient	The patient can use this system directly. First, he will chat with a system using a chatbot after that the patient will upload his HRCT and reports and see his results. The system will also suggest precautions to the patient. The patient can also see the 3D model of lungs on which the infected area is visualized. The patient can also see his reports later.

3.2 Requirement Identifying Technique

2.2.1 Event-response table

Table 3 Event response table

Event	**System State**	**Exception**	**Response**	
			In case of exception (Yes)	In case of exception (No)
Upload CT scan	The system is displaying upload form	1. The wrong file is uploaded 2. Error in uploading the file	The system will ask to upload again	The system will store a CT scan on a server and its path in the database.
Upload CT Scan Report	The system is displaying upload form	1. The wrong file is uploaded 2. Error in uploading the file	The system will ask to upload reports again	The system will store the CT Scan report on the server and its path in the database.
Upload Chest X-Ray	The system is displaying upload form	1. The wrong file is uploaded 2. Error in uploading the file	The system will ask to upload again	The system will store Chest X-Ray on the server and its path in the database.

Upload Chest X-Ray report	The system is displaying upload form	1. The wrong file is uploaded 2. Error in uploading the file	The system will ask to upload reports again	The system will store the Chest X-Ray report on the server and its path in the database.
Upload Blood CP report	The system is displaying upload form	1. The wrong file is uploaded 2. Error in uploading the file	The system will ask to upload reports again	The system will store the Blood CP report on the server and its path in the database.
Upload D-Dimer report	The system is displaying upload form	1. The wrong file is uploaded 2. Error in uploading the file	The system will ask to upload reports again	The system will store the D-Dimer report on the server and its path in the database.
Upload CRP report	The system is displaying upload form	1. The wrong file is uploaded 2. Error in uploading the file	The system will ask to upload reports again	The system will store the D-Dimer report on the server and its path in the database.
Intake User History	The system is interacting with the user through a chatbot	The chatbot will not work if the connection is lost	Ask for the information again	Store user information in the database
Intake Current symptoms	The system is interacting with the user through a chatbot	The chatbot will not work if the connection is lost	Ask for the information again	Store user symptoms in the database
Extract information	The system will get reports from the server	Connection is lost	The system will get reports again	Extract the required data from the reports

				and store it.
Pre-Processing	The system will get images from the server	Connection is lost	The system will get images again	The system will perform image segmentation and do image processing on images
Apply ML Model	The system will take the images after pre-processing	Error in getting images	Try again to get images.	The system will analyze the images and detect and predict the disease.
Show Result	The system will get the results from ML Model	Connection is lost during getting the result	Try again to get results from Model	The System will display the result on the screen.
Suggest precautions	The system will get the suggestions from ML Model	Connection is lost during getting suggestions.	Try again to get suggestions from Model	The system will display the suggestions on the screen.
3D Model visualization	The system will highlight the infected areas on 3D-Model			The system will display the 3D Model on which the infected area is highlighted.
Focusing the model	The system is showing the model			The system will zoom in and zoom out the model according to the user.
Move the slider	The system is showing the			The system will predict the next

	model			condition of the lungs.

3.3 Functional Requirements

3.3.1 FR-1: Uploading CT scan

Identifier	FR-1
Title	Uploading CT scan
Requirement	The CT scan image shall be in DICOM or jpg format. User shall be able to upload a single file at a time. If format is in DICOM then first it is converted to jpg and then it shall be uploaded to server. If the format is not DICOM or jpg, then it shall display an error of wrong input format.
Source	Brainstorming
Rationale	The system depends on CT scan to work and predict the COVID symptoms.
Business Rule (if required)	N/A
Dependencies	N/A
Priority	High

3.3.2 FR-2: Storing CT scan

Identifier	FR-2
Title	Storing CT scan

Requirement	After uploading the CT scan to the server, the link of the CT scan image shall be stored in the database.
Source	Brainstorming
Rationale	CT Scan will be later used for prediction of our trained model
Business Rule (if required)	N/A
Dependencies	FR-1
Priority	High

3.3.3 FR-3: Retrieving CT scan

Identifier	FR-3
Title	Retrieving CT scan
Requirement	The address of patient's CT scan in the server shall be retrieved from the database.
Source	Brainstorming
Rationale	The CT scan will be used for prediction so it must be retrieved from the server
Business Rule (if required)	N/A
Dependencies	FR-1, FR-2
Priority	High

3.3.4 FR-4: Uploading CT Scan Report

Identifier	FR-4
Title	Uploading CT Scan Report
Requirement	Patient's CT Scan Report shall be in pdf format. If format is correct, then it shall be uploaded to server. If not, then it shall display an error of wrong input format
Source	Brainstorming
Rationale	If the user won't be able to upload a CT scan, then the system will not work
Business Rule (if required)	N/A
Dependencies	N/A
Priority	High

3.3.5 FR-5: Storing CT Scan Report

Identifier	FR-5
Title	Storing CT Scan Report
Requirement	After uploading the CT Scan report to the server, the address of the CT Scan report shall be stored in the database.
Source	Brainstorming
Rationale	CT Scan Report will be later used for prediction of our trained model

Business Rule (if required)	N/A
Dependencies	FR-4
Priority	High

3.3.6 FR-6: Retrieving CT Scan Report

Identifier	FR-6
Title	Retrieving CT Scan Report
Requirement	The address of patient's CT scan report in the server shall be retrieved from the database.
Source	Brainstorming
Rationale	The CT Scan Report will be used for prediction so it must be retrieved from the server
Business Rule (if required)	N/A
Dependencies	FR-4, FR-5
Priority	High

3.3.7 FR-7: Uploading X-Ray

Identifier	FR-7
Title	Uploading X-Ray
Requirement	The X-ray shall be in DICOM or jpg format. User can upload a single file at a time. If format is in DICOM then first it is converted to jpg and then it

	shall be uploaded to server. If the format is not DICOM or jpg, then it shall display an error of wrong input format.
Source	Brainstorming
Rationale	The system depends on X-ray to work and predict the COVID symptoms.
Business Rule (if required)	N/A
Dependencies	N/A
Priority	High

3.3.8 FR-8: Storing X-Ray

Identifier	FR-8
Title	Storing X-Ray
Requirement	After uploading the X-ray to the server, the link of the X-ray shall be stored in the database.
Source	Brainstorming
Rationale	The X-ray will be later used for the prediction of our trained model.
Business Rule (if required)	N/A
Dependencies	FR-7
Priority	High

3.3.9 FR-9: Retrieving X-Ray

Identifier	FR-9
Title	Retrieving X-Ray
Requirement	The address of patient’s X-Ray shall be retrieved from the database.
Source	Brainstorming
Rationale	The X-ray will be used for prediction so it must be retrieved from the server.
Business Rule (if required)	N/A
Dependencies	FR-7, FR-8
Priority	High

3.3.10 FR-10: Uploading X-Ray report

Identifier	FR-10
Title	Uploading X-Ray Report
Requirement	Patient’s X-ray Report shall be in pdf format. If format is correct, then it shall be uploaded to server. If not, then it shall display an error of wrong input format
Source	Brainstorming
Rationale	Using X-ray report we can accurately predict the COVID symptoms, so we need this report.
Business Rule (if required)	N/A

Dependencies	N/A
Priority	High

3.3.11 FR-11: Storing X-Ray report

Identifier	FR-11
Title	Storing X-Ray report
Requirement	After uploading the X-ray to the server, the link of the X-ray report shall be stored in the database.
Source	Brainstorming
Rationale	X-ray report will be later used for prediction of our trained model
Business Rule (if required)	N/A
Dependencies	FR-10
Priority	High

3.3.12 FR-12: Retrieving X-Ray report

Identifier	FR-12
Title	Retrieving X-Ray report
Requirement	The address of patient’s X-Ray report shall be retrieved from the database
Source	Brainstorming
Rationale	The X-ray report will be used for prediction so it must be retrieved from

	the server
Business Rule (if required)	N/A
Dependencies	FR-10, FR-11
Priority	High

3.3.13 FR-13: Uploading Blood CP test result

Identifier	FR-13
Title	Uploading Blood CP test result
Requirement	Patient's Blood CP test result shall be in pdf format. If format is correct, then it shall be uploaded to server. If not, then it shall display an error of wrong input format.
Source	Brainstorming
Rationale	If the user won't be able to upload the Blood CP test result, then the system will not work accurately
Business Rule (if required)	N/A
Dependencies	N/A
Priority	High

3.3.14 FR-14: Storing Blood CP test result

Identifier	FR-14
Title	Storing Blood CP test result

Requirement	After uploading the Blood CP test result to the server, the link of the Blood CP test result shall be stored in the database.
Source	Brainstorming
Rationale	Blood CP test result will be later used for prediction of our trained model
Business Rule (if required)	N/A
Dependencies	FR-13
Priority	High

3.3.15 FR-15: Retrieving Blood CP test result

Identifier	FR-15
Title	Retrieving Blood CP test result
Requirement	The address of patient’s Blood CP test result shall be retrieved from the database
Source	Brainstorming
Rationale	The Blood CP test result will be used for prediction so it must be retrieved from the server
Business Rule (if required)	N/A
Dependencies	FR-13, FR-14
Priority	High

3.3.16 FR-16: Uploading d-dimer test result

Identifier	FR-16
Title	Uploading d-dimer test
Requirement	Patient’s d-dimer test result shall be in pdf format. If format is correct, then it shall be uploaded to server. If not, then it shall display an error of wrong input format
Source	Brainstorming
Rationale	If the user won’t be able to upload the d-dimer test result, then the system will not work accurately
Business Rule (if required)	N/A
Dependencies	N/A
Priority	High

3.3.17 FR-17: Storing d-dimer test result

Identifier	FR-17
Title	Storing d-dimer test result
Requirement	After uploading the d-dimer test result to the server, the link of the d-dimer test result shall be stored in the database.
Source	Brainstorming
Rationale	Blood d-dimer result will be later used for the prediction of our trained model
Business Rule	N/A

(if required)	
Dependencies	FR-16
Priority	High

3.3.18 FR-18: Retrieving d-dimer test result

Identifier	FR-18
Title	Retrieving d-dimer test result
Requirement	The address of patient's d-dimer test result shall be retrieved from the database
Source	Brainstorming
Rationale	The Blood CP test result will be used for prediction so it must be retrieved from the server
Business Rule (if required)	N/A
Dependencies	FR-16, FR-17
Priority	Medium

3.3.19 FR-19: Intake Patient Information

Identifier	FR-19
Title	Patient Information
Requirement	User shall be able to chat with the AI bot, where multiple questions shall be asked. Patient information such as name, age, gender shall be input

	from the user through the Chabot and shall be stored in the database.
Source	Brainstorming
Rationale	The patient's information is required to generate the report for that patient.
Business Rule (if required)	N/A
Dependencies	N/A
Priority	Medium

3.3.20 FR-20: Intake patient medical history

Identifier	FR-20
Title	Patient medical History
Requirement	The patient's history of smoking, and other health issues shall be input from a user through a chat bot. Then it shall be stored in the database.
Source	Brainstorming
Rationale	N/A
Business Rule (if required)	N/A
Dependencies	N/A
Priority	Medium

3.3.21 FR-21: Intake Current Symptoms

Identifier	FR-21
Title	Current Symptoms
Requirement	Current symptoms of a patient shall be described through chat bot. These symptoms will be stored in the database.
Source	Brainstorming
Rationale	This will be required for better suggestions and prediction
Business Rule (if required)	N/A
Dependencies	N/A
Priority	Medium

3.3.22 FR-22: Image to Text conversion

Identifier	FR-22
Title	Image to Text conversion
Requirement	The images must be in jpg format. After applying OCR on the images, the data from the images must be converted to text.
Source	Brainstorming
Rationale	Data will be required from the image for predictions through the ML model so we will need to convert images to text first.
Business Rule (if required)	N/A
Dependencies	N/A

Priority	Medium

3.3.23 FR-23: Data Extraction

Identifier	FR-23
Title	Data Extraction
Requirement	After conversion of data from image to text, multiple string operations shall be applied on data to extract specific information.
Source	Brainstorming
Rationale	Data will be required from the image for predictions through the ML model so we will need to extract data first.
Business Rule (if required)	N/A
Dependencies	N/A
Priority	Medium

3.3.24 FR-24: Storing Data

Identifier	FR-24
Title	Storing Data
Requirement	The data is extracted from the reports and images. This must be stored in the database to be used later.
Source	Brainstorming
Rationale	Data such as blood cp value, d dimer values will be used in prediction. So, we need to store such values which are extracted
Business Rule (if required)	N/A
Dependencies	FR-23
Priority	Medium

3.3.25 FR-25: Grey scaling the Image

Identifier	FR-25
Title	Grey scaling the Image
Requirement	The image shall be converted to greyscale so that it must be used in machine learning
Source	Brainstorming
Rationale	This will improve our results of predictions
Business Rule (if required)	N/A
Dependencies	N/A

Priority	High

3.3.26 FR-26: Image Resizing

Identifier	FR-26
Title	Image Resizing
Requirement	The image shall be resized according to a specific size so that it must be used in machine learning
Source	Brainstorming
Rationale	This will improve our results in predictions
Business Rule (if required)	N/A
Dependencies	N/A
Priority	High

3.3.27 FR-27: Changing Orientation

Identifier	FR-27
Title	Changing Orientation
Requirement	The orientation of image must be straight.
Source	Brainstorming
Rationale	Orientation of images is different when we take images, but our ML model will work better if it gets the same orientation of images.

Business Rule (if required)	N/A
Dependencies	N/A
Priority	Medium

3.3.28 FR-28: Random Exposure

Identifier	FR-28
Title	Random Exposure
Requirement	Image brightness shall be adjusted so that image is a bit bright
Source	Brainstorming
Rationale	Sometimes an image is a bit dark which will cause our model to make predictions wrong.
Business Rule (if required)	N/A
Dependencies	N/A
Priority	Medium

3.3.29 FR-29: Random Noise

Identifier	FR-29
Title	Random Noise
Requirement	Noise shall be applied where needed in the image.

Source	Brainstorming
Rationale	Sometimes adding noise to the image helps the model to give better results
Business Rule (if required)	N/A
Dependencies	N/A
Priority	Medium

3.3.30 FR-30: Trained ML model

Identifier	FR-30
Title	Trained ML model
Requirement	There shall be a trained ML model from which we shall get our predictions. The ML model shall be trained from the dataset of CT Scans of Covid patients.
Source	Brainstorming
Rationale	When we get predictions, we always have a trained ML model from which we get our predictions by sending input of a sample data and getting prediction through the model
Business Rule (if required)	N/A
Dependencies	N/A
Priority	High

3.3.31 FR-31: Generate Result

Identifier	FR-31
Title	Generate Result
Requirement	There shall be ML models trained to give us the prediction of Covid in a patient. The system shall be able to generate the result of the prediction through that ML model.
Source	Brainstorming
Rationale	This is the basic purpose of the system to give predictions so we must have a way to generate a result
Business Rule (if required)	N/A
Dependencies	N/A
Priority	High

3.3.32 FR-32: View Result

Identifier	FR-32
Title	View Result
Requirement	User shall be able to view the result of the prediction.
Source	Brainstorming
Rationale	This is the basic purpose of the system to give predictions so we must have a way to show the result
Business Rule (if required)	N/A

Dependencies	FR-31
Priority	High

3.3.33 FR-33: Create Suggestions

Identifier	FR-37
Title	Create Suggestions
Requirement	The system shall be able to create suggestions from the result
Source	Brainstorming
Rationale	This is the basic purpose of the system to give suggestions according to the severity of the disease so we must have a way to generate suggestions
Business Rule (if required)	N/A
Dependencies	N/A
Priority	High

3.3.34 FR-34: View Suggestions

Identifier	FR-38
Title	View Suggestions
Requirement	Users shall be able to view suggestions from the system
Source	Brainstorming
Rationale	This is the basic purpose of the system to give suggestions according to the

	severity of the disease so we must have a way to show
Business Rule (if required)	N/A
Dependencies	FR-33
Priority	High

3.3.35 FR-35: Create a 3D model

Identifier	FR-39
Title	Create 3D model
Requirement	3d model shall be created according to the given reports and X-rays
Source	Brainstorming
Rationale	This will help the user identify which part of the lung is infected
Business Rule (if required)	N/A
Dependencies	N/A
Priority	High

3.3.36 FR-36: Display 3d model

Identifier	FR-36
Title	Display 3d model
Requirement	System shall be able to display 3d model

Source	Brainstorming
Rationale	This will help the user identify which part of the lung is infected
Business Rule (if required)	N/A
Dependencies	FR-35
Priority	High

3.3.7 FR-37: Zoom 3d model

Identifier	FR-37
Title	Zoom 3d model
Requirement	3d model shall be zoomed in and zoomed out if needed
Source	Brainstorming
Rationale	Users can easily see the lungs model if we have a way to zoom in and zoom out
Business Rule (if required)	N/A
Dependencies	FR-35, FR-36
Priority	High

3.3.38 FR-38: Slider

Identifier	FR-38
Title	Slider
Requirement	There shall be a slider through which you can see a prediction of 14 days on the 3d model
Source	Brainstorming
Rationale	This will help examine his condition in future days.
Business Rule (if required)	N/A
Dependencies	FR-36
Priority	High

3.3.39 FR-39: Move the Slider

Identifier	FR-39
Title	Move the Slider
Requirement	There shall be a pointer through which the user can move the slider.
Source	Brainstorming
Rationale	The user will be able to move the slider.
Business Rule (if required)	N/A
Dependencies	FR-38
Priority	High

3.3.40 FR-40: Predict lungs condition

Identifier	FR-40
Title	Predict lungs condition
Requirement	There shall be an ML Model which shall predict the condition of the lungs after the interval of time.
Source	Brainstorming
Rationale	This will help the user to see the future condition of his lungs.
Business Rule (if required)	N/A
Dependencies	N/A
Priority	High

3.3.41 FR-41: Show progress of lungs condition

Identifier	FR-41
Title	Show progress of lungs condition
Requirement	There shall be an ML Model which shall predict the condition of the lungs after the interval of time.
Source	Brainstorming
Rationale	The user can easily see the future condition of his lungs.
Business Rule (if required)	N/A
Dependencies	N/A

Priority	High

3.4 Non-Functional Requirements

3.4.1 Reliability

- The web app shall be able to run on all types of systems i.e., google chrome, edge, Firefox.
- The service shall be available 24/7.
- The system shall be able to provide full security to user data.

3.4.2 Usability

- Users need to have a basic understanding of how to navigate the application.
- All the icons used should easily depict what they do.
- Buttons in the system should be big enough so that users can press them easily.

3.4.3 Performance

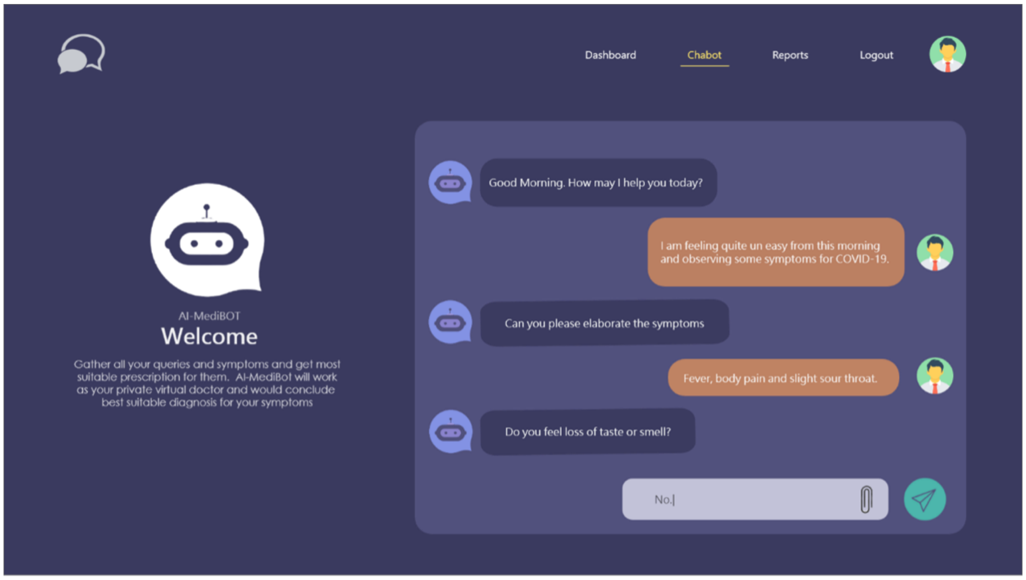
- It should be able to handle more than 5000 requests without crashing.
- The web app shall respond to any trigger within 1s.

3.4.4 Security

- Test results and images shall be sent to the concerned user.

3.5 External Interface Requirements

3.5.1 User Interfaces Requirements



AI-MediBOT

Welcome

Gather all your queries and symptoms and get most suitable prescription for them. AI-MediBot will work as your private virtual doctor and would conclude best suitable diagnosis for your symptoms

Good Morning. How may I help you today?

I am feeling quite un easy from this morning and observing some symptoms for COVID-19.

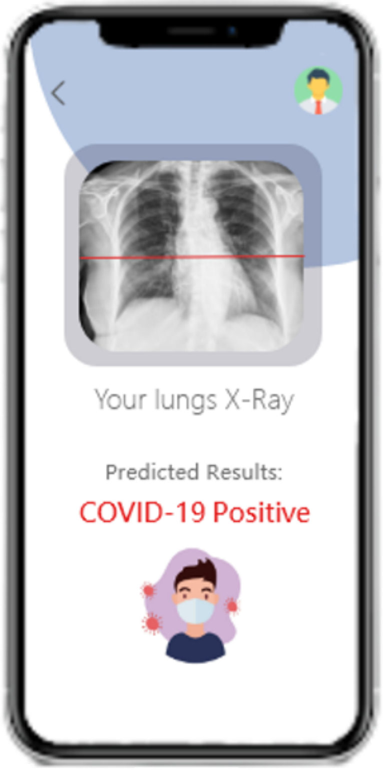
Can you please elaborate the symptoms

Fever, body pain and slight sour throat.

Do you feel loss of taste or smell?

No.|

Figure 1 AI Based Chatbot



Predicted Results:
COVID-19 Positive

Figure 2 Prediction of Disease

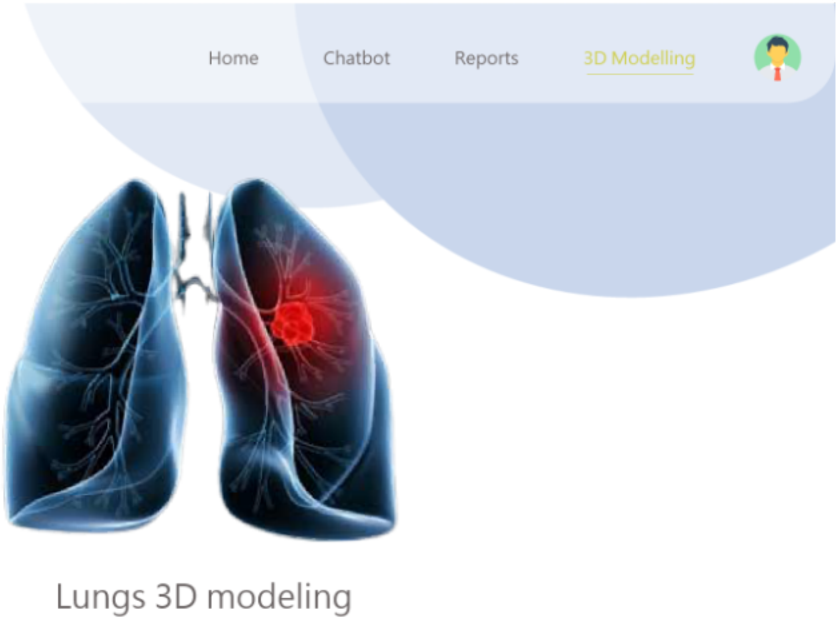


Figure 3 3D Visualization Web-App

3.5.2 Software interfaces

- The system user interface is independent of all operating systems.
- The system shall communicate with a cloud database system.
- The system shall allow the user to upload images in DICOM format

3.5.3 Communications interfaces

- The system shall send a report to user mail.
- The system supports all types of web browsers.

4 Design and Architecture

4.1 Architectural Design

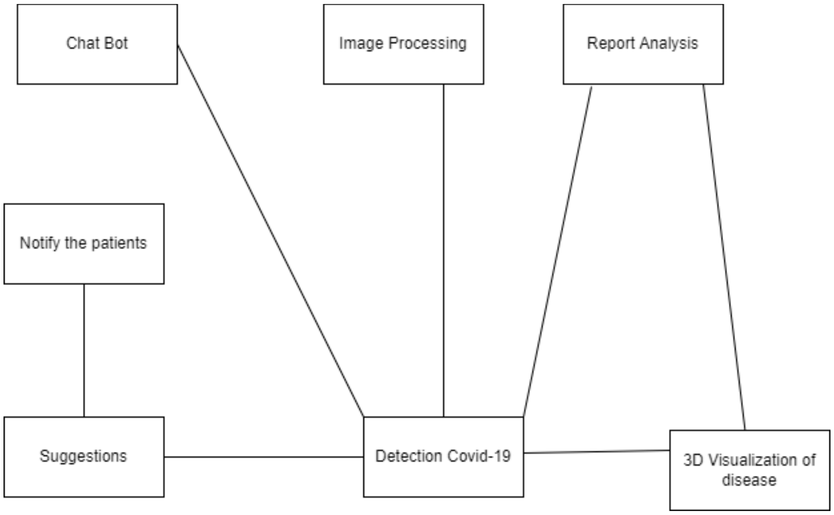


Figure 4: Box and line Diagram

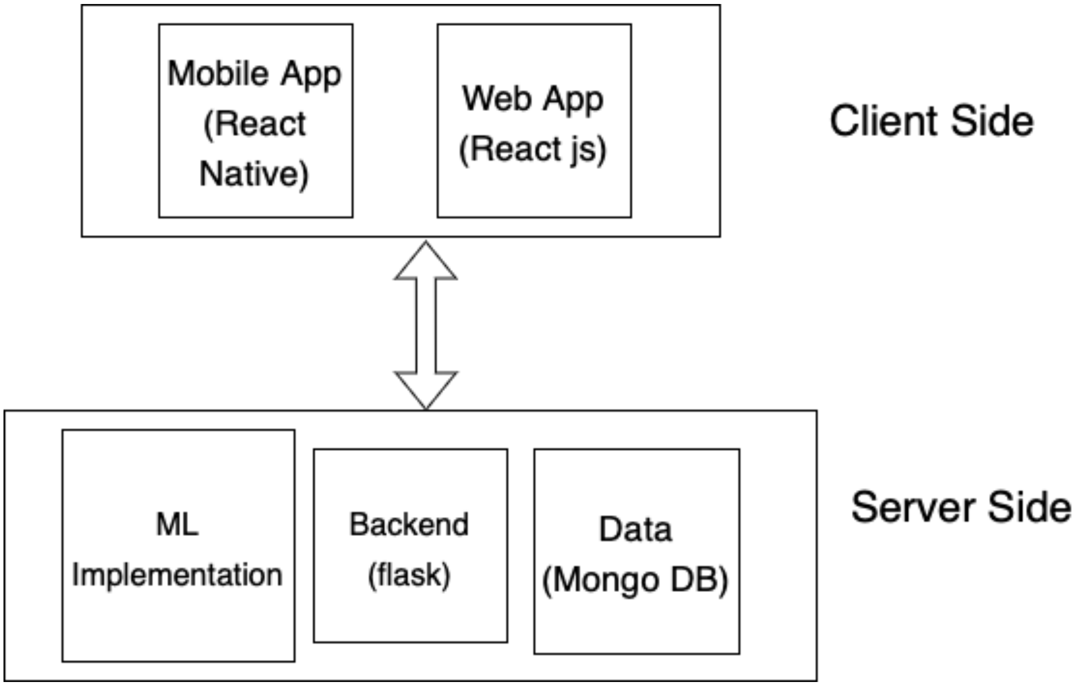


Figure 5: Architecture diagram general

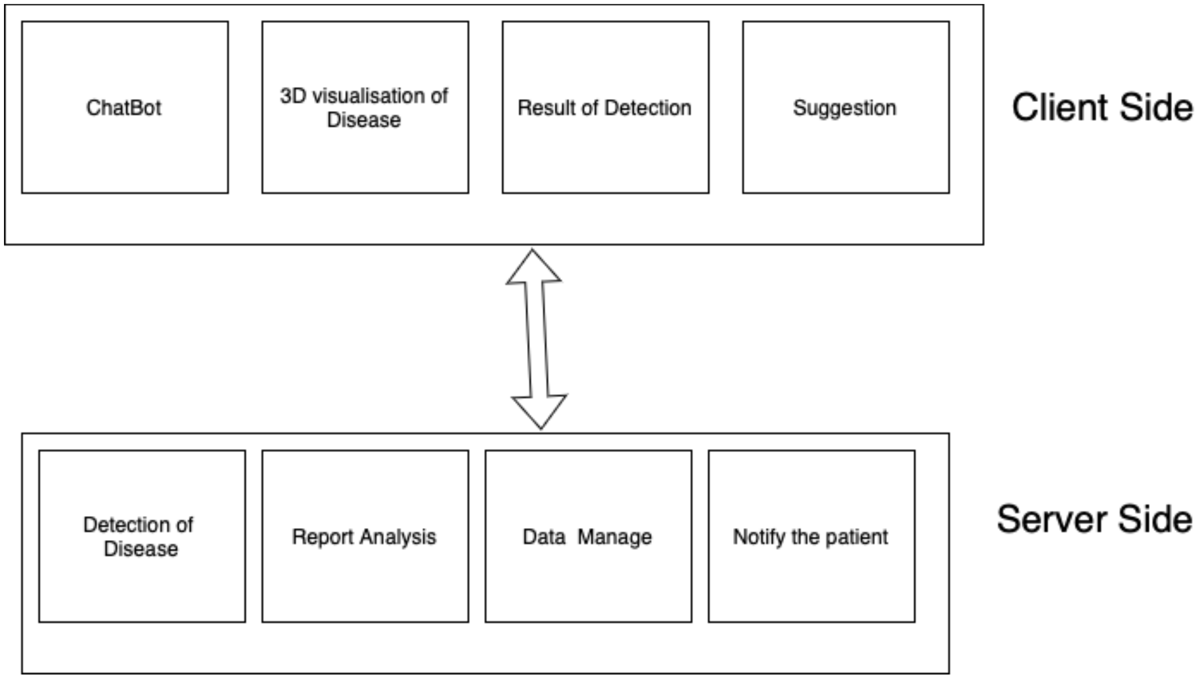


Figure 6 Architectural Diagram (Detailed)

4.2 Design Models

4.2.1 Activity Diagram

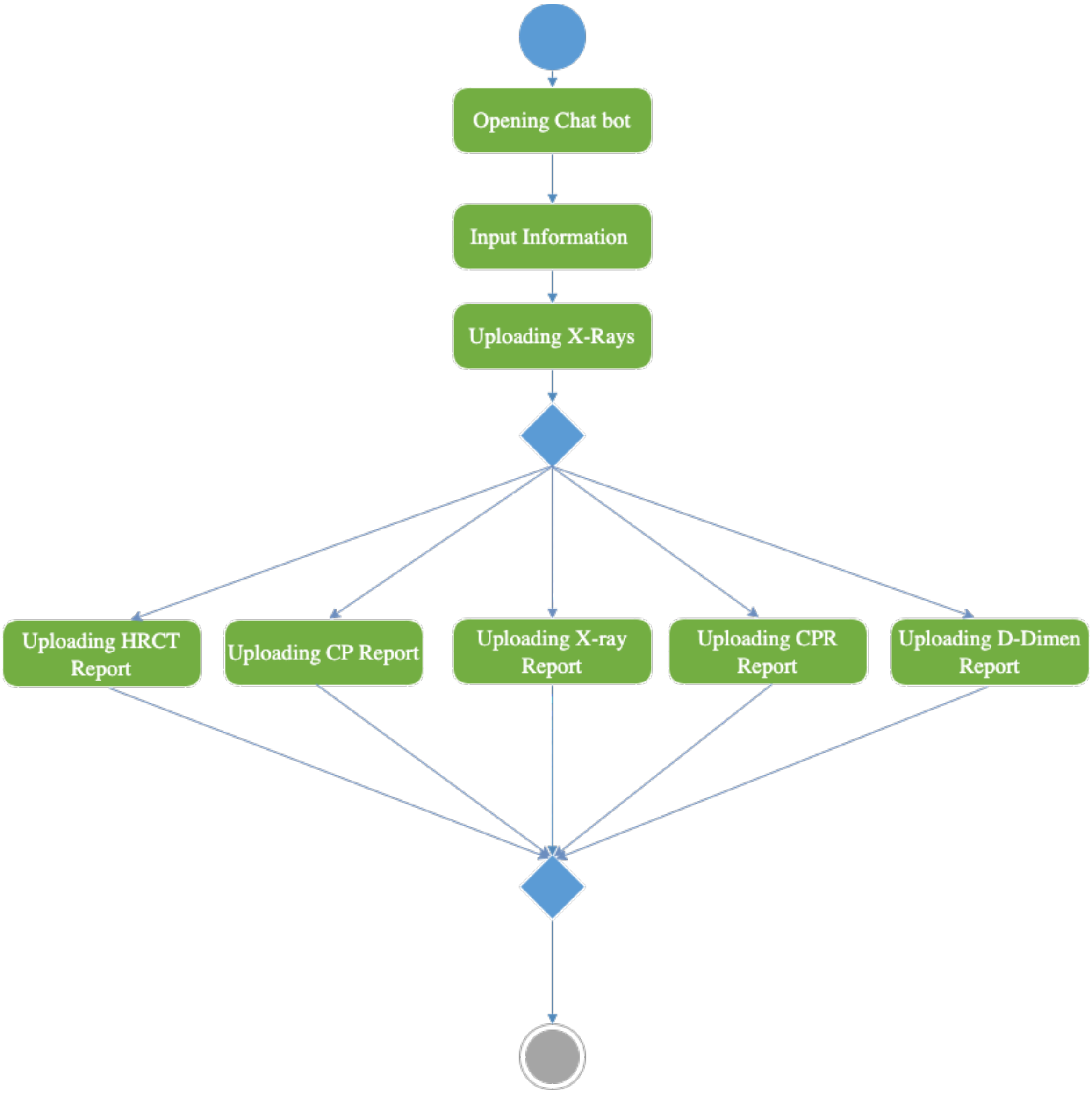


Figure 7 Uploading Data

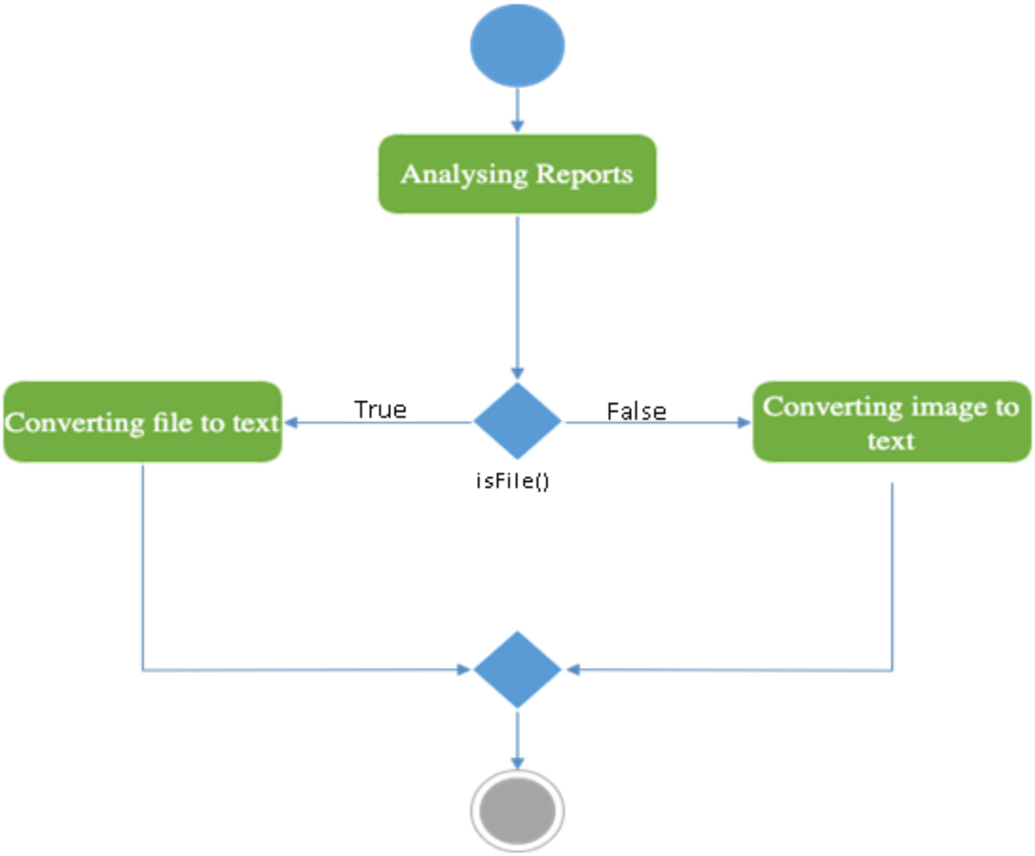


Figure 8 Report Analysis

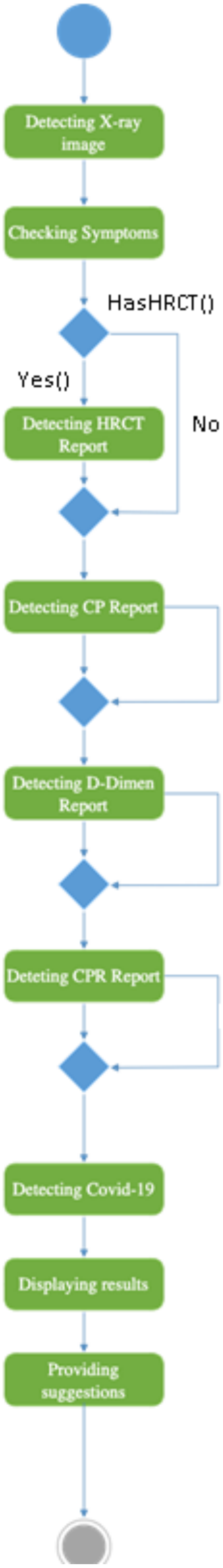


Figure 9 COVID-19 Detection

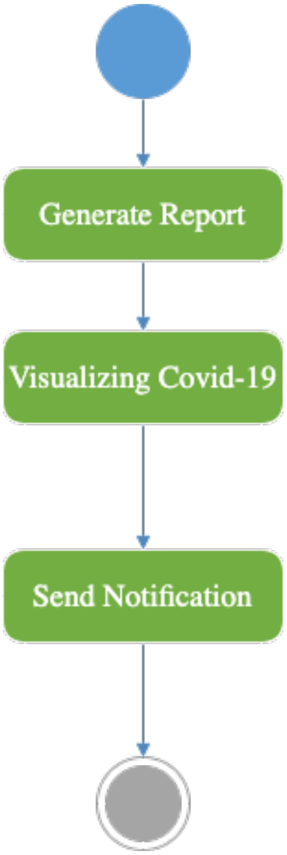


Figure 10 Sending Reports

4.2.2 Data Flow Diagram

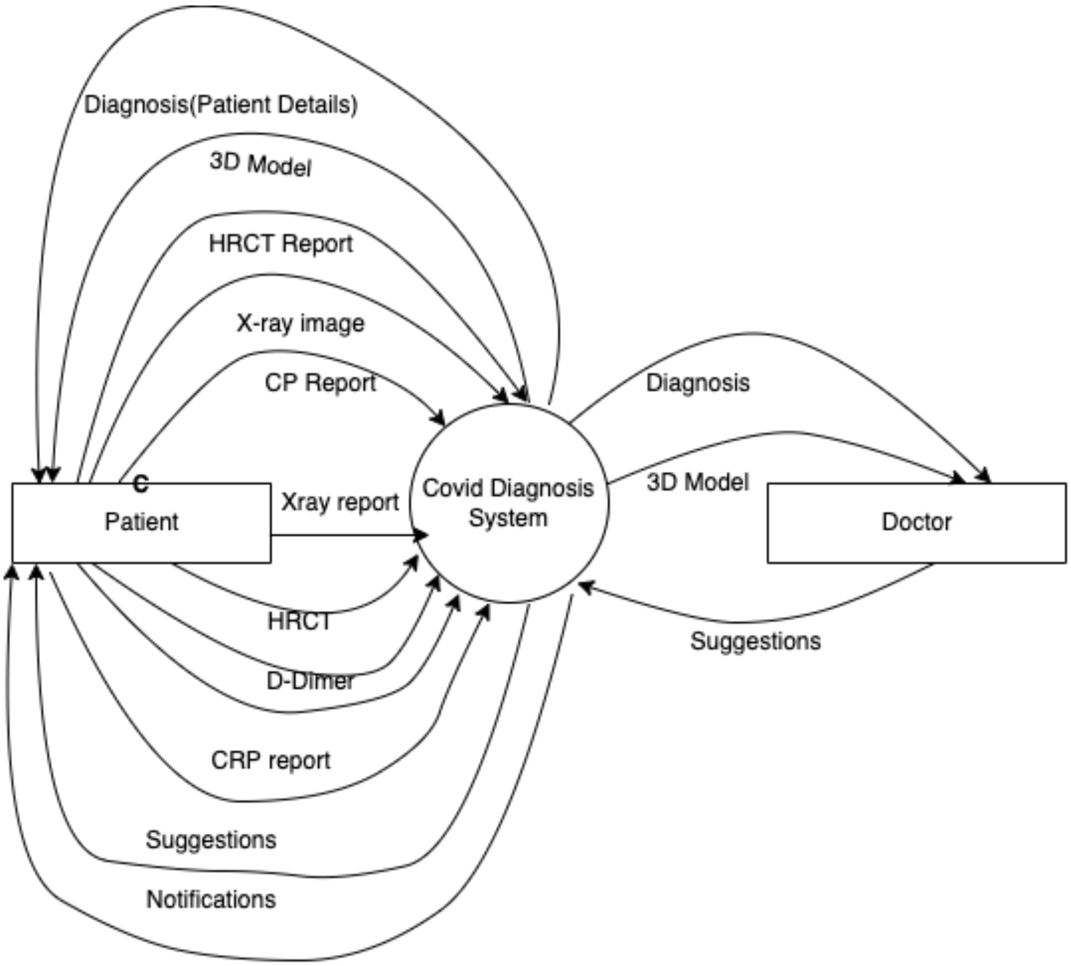


Figure 11 DFD Level 0

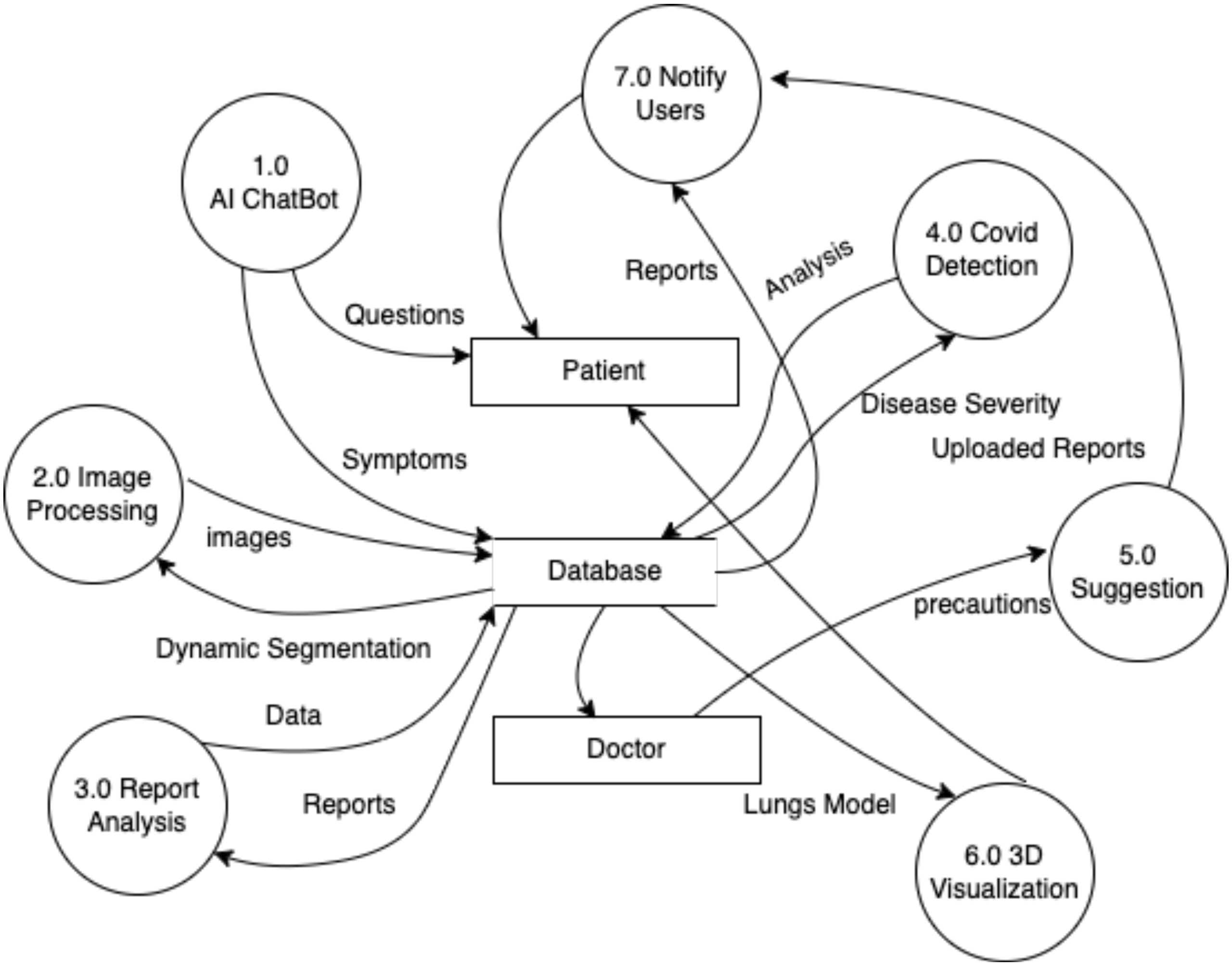


Figure 12: DFD level 1

x

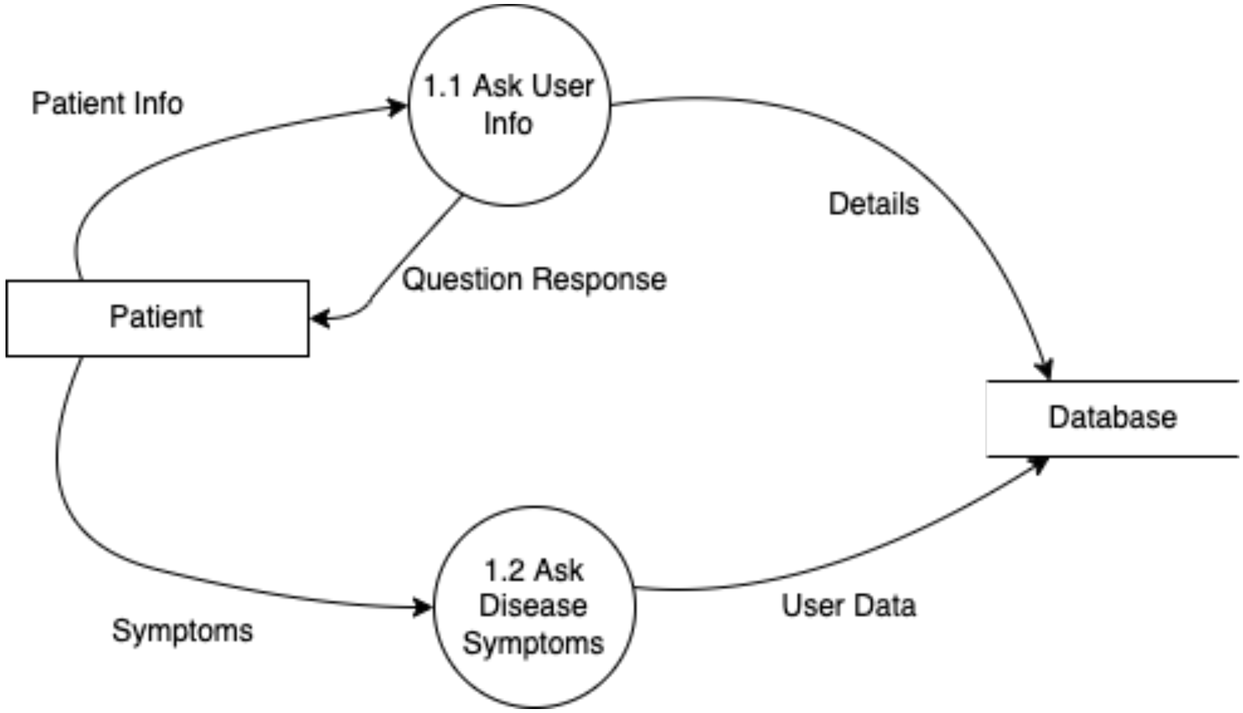


Figure 13: Module 1 level 2 DFD

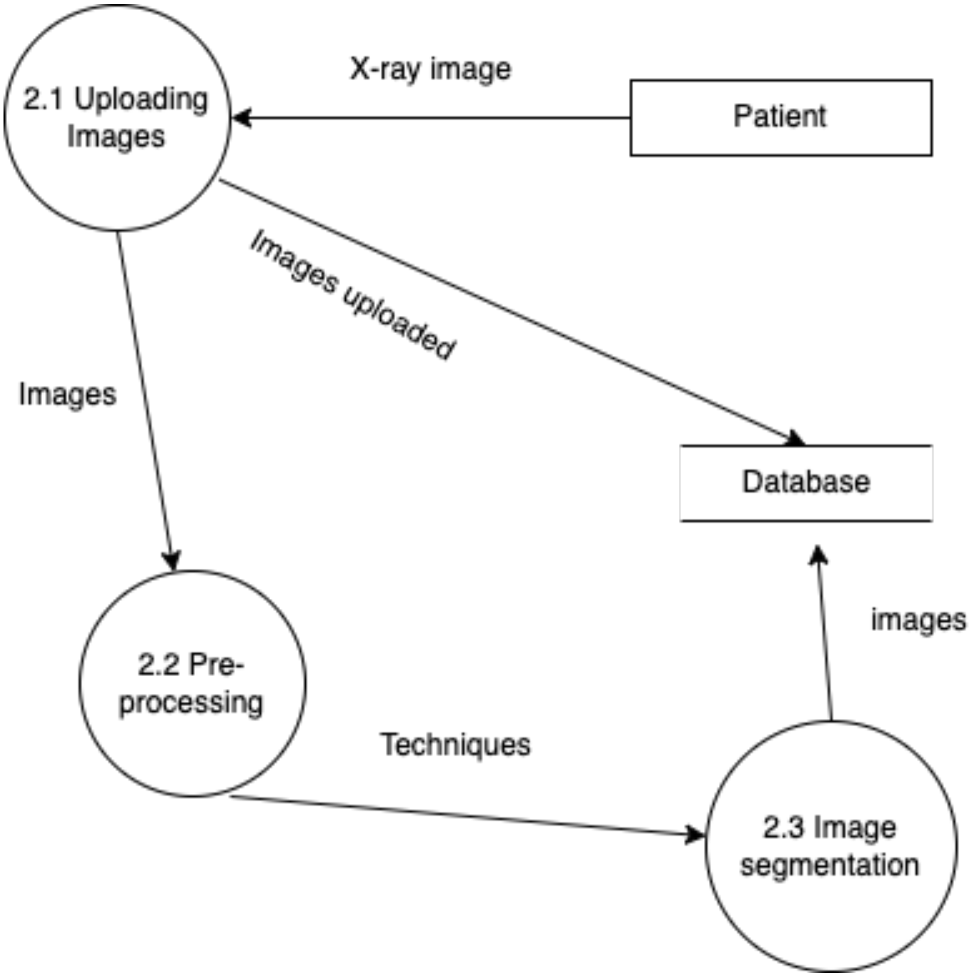


Figure 14: Module 2 Level 2 DFD

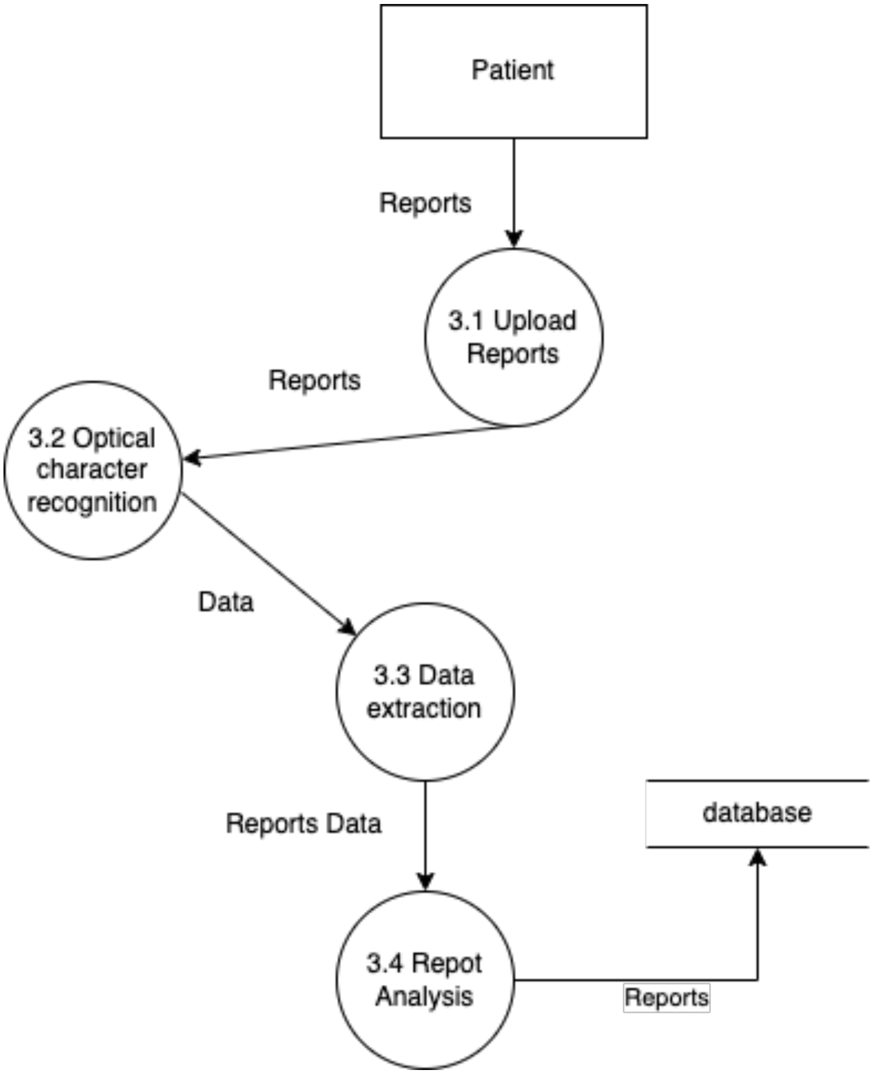


Figure 15: Module 3 Level 2 DFD

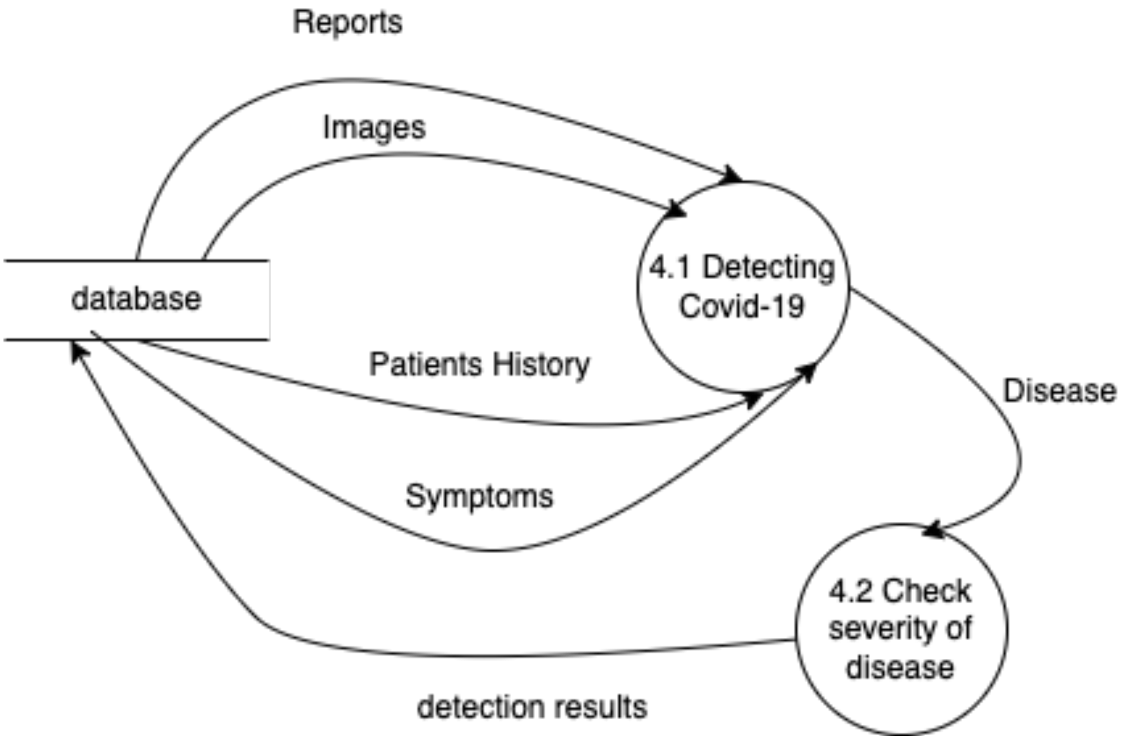


Figure 16: Module 4 Level 2 DFD

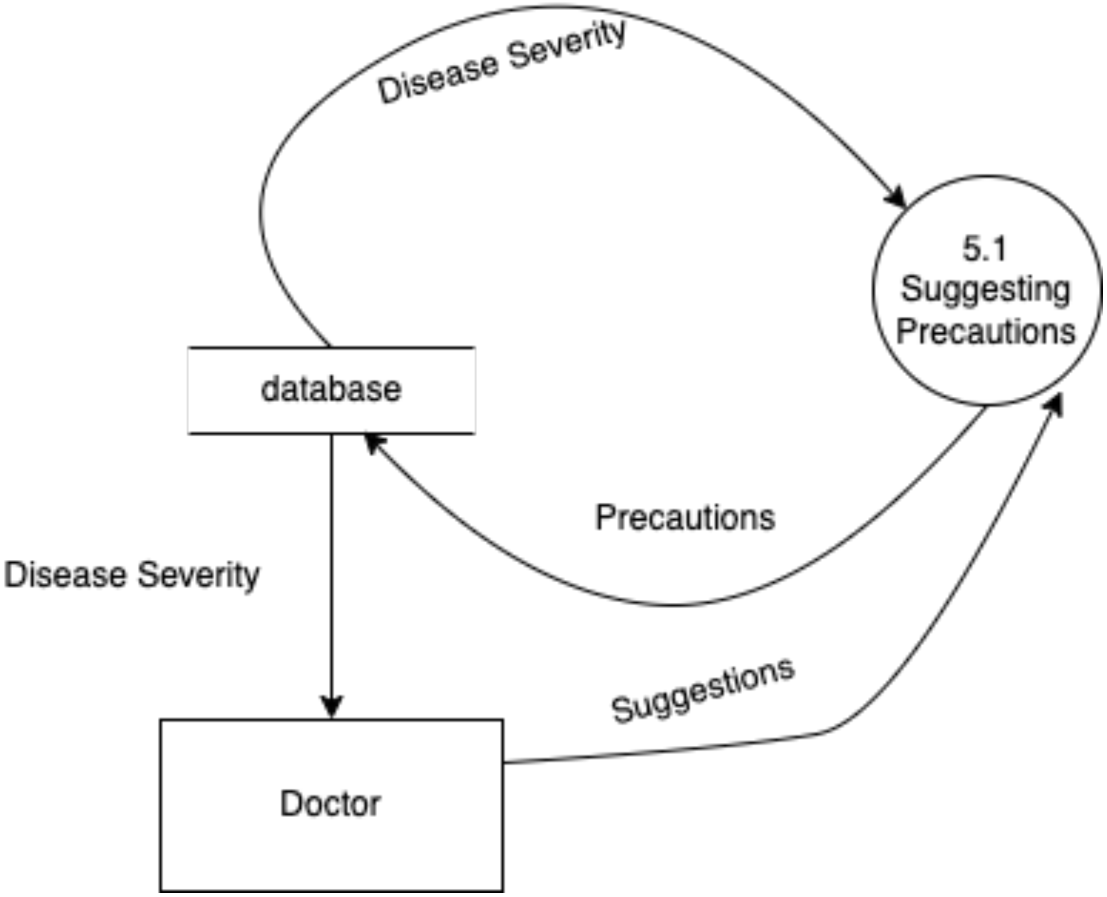


Figure 17: Module 5 Level 2 DFD

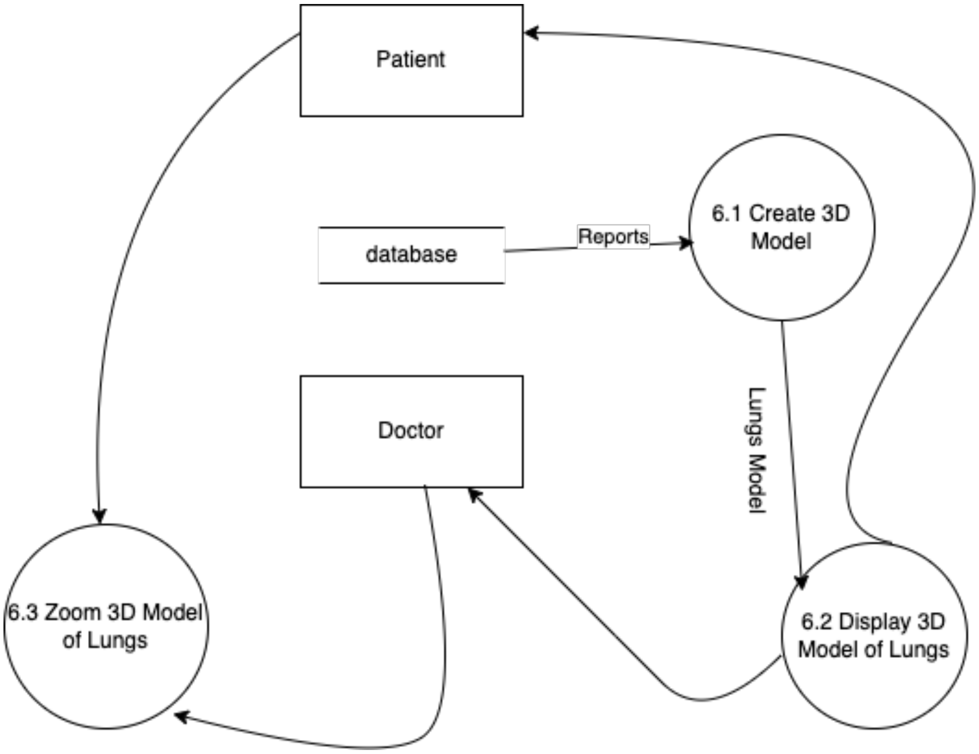


Figure 18: Module 6 Level 2 DFD

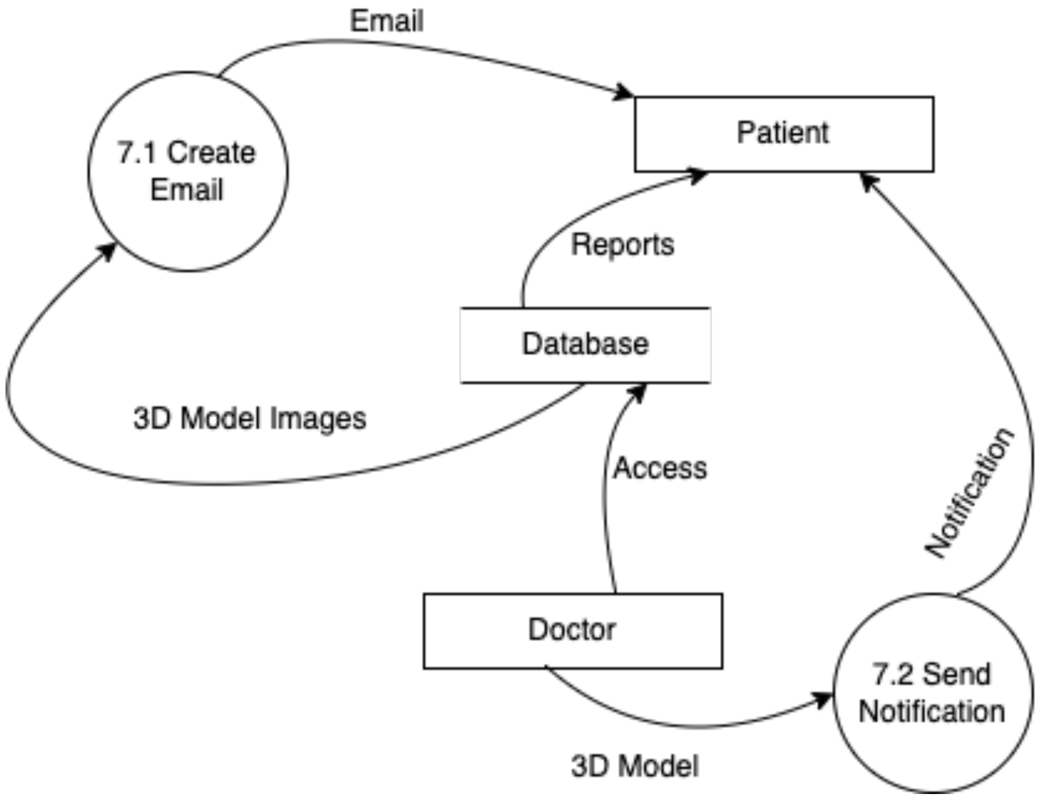
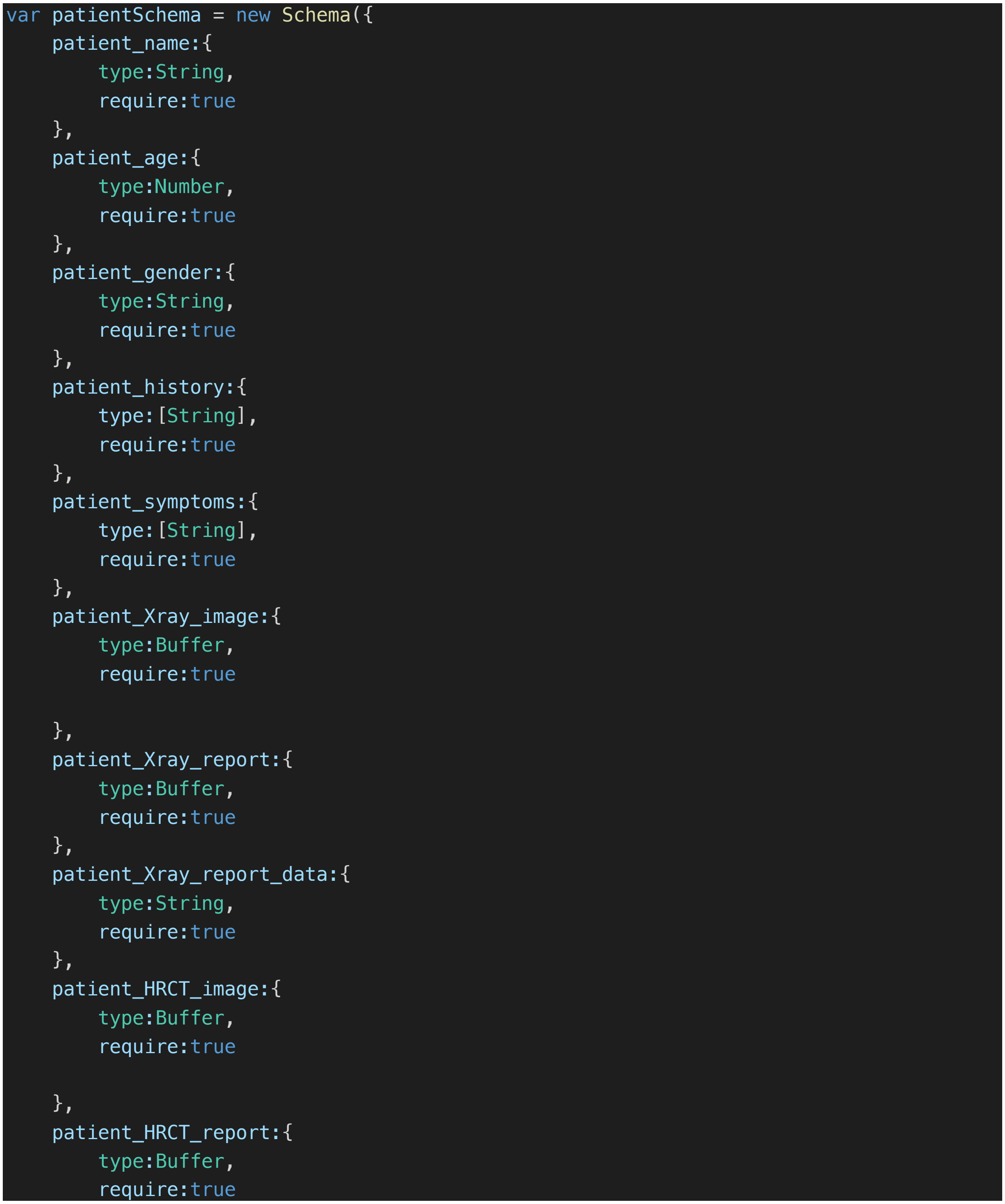


Figure 19 Module-7: Level 2 DFD

4.3 Data Design

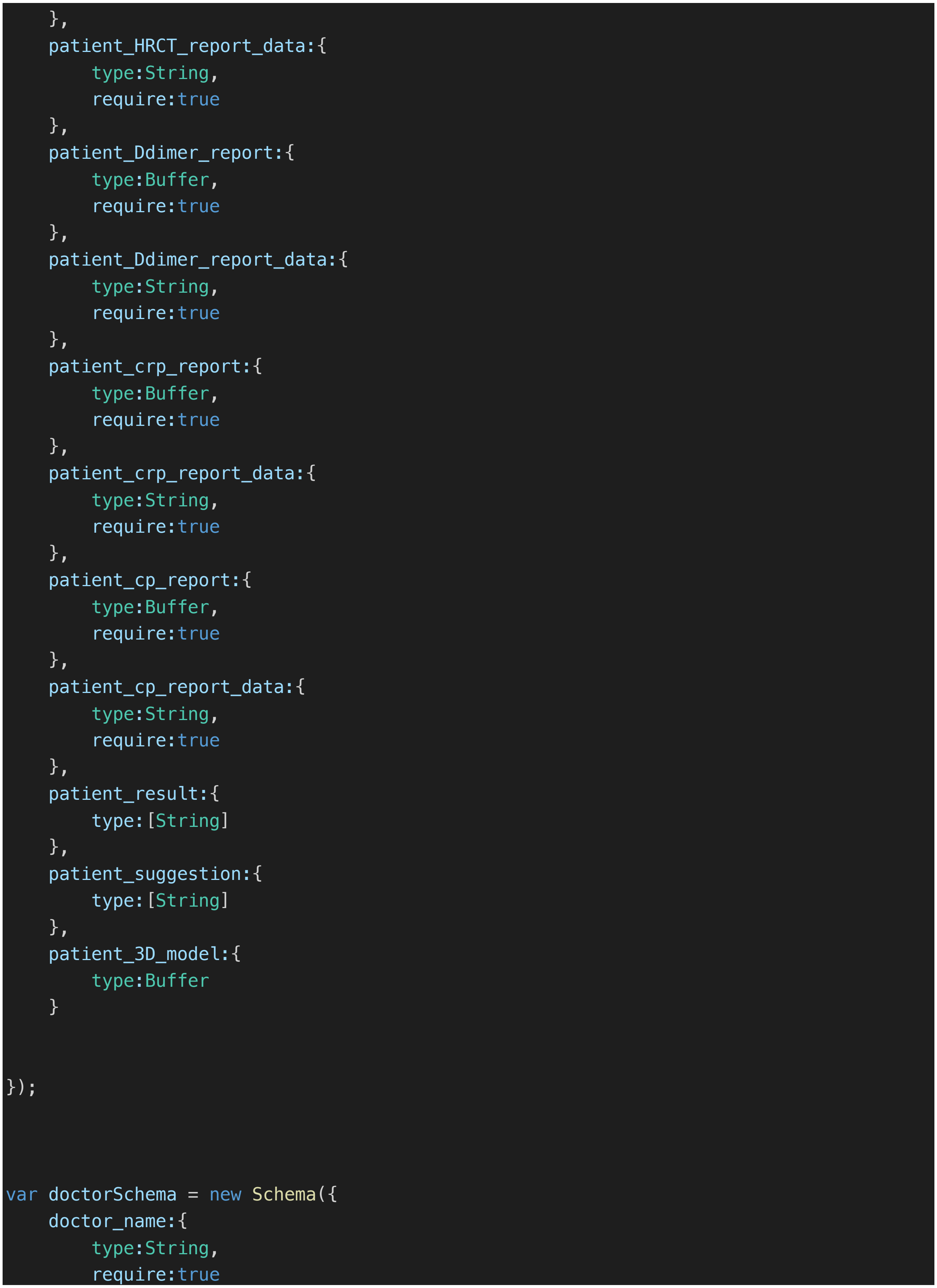
4.3.1 JSON Schema



```
var patientSchema = new Schema({
    patient_name:{
        type:String,
        require:true
    },
    patient_age:{
        type:Number,
        require:true
    },
    patient_gender:{
        type:String,
        require:true
    },
    patient_history:{
        type:[String],
        require:true
    },
    patient_symptoms:{
        type:[String],
        require:true
    },
    patient_Xray_image:{
        type:Buffer,
        require:true

    },
    patient_Xray_report:{
        type:Buffer,
        require:true
    },
    patient_Xray_report_data:{
        type:String,
        require:true
    },
    patient_HRCT_image:{
        type:Buffer,
        require:true

    },
    patient_HRCT_report:{
        type:Buffer,
        require:true
```

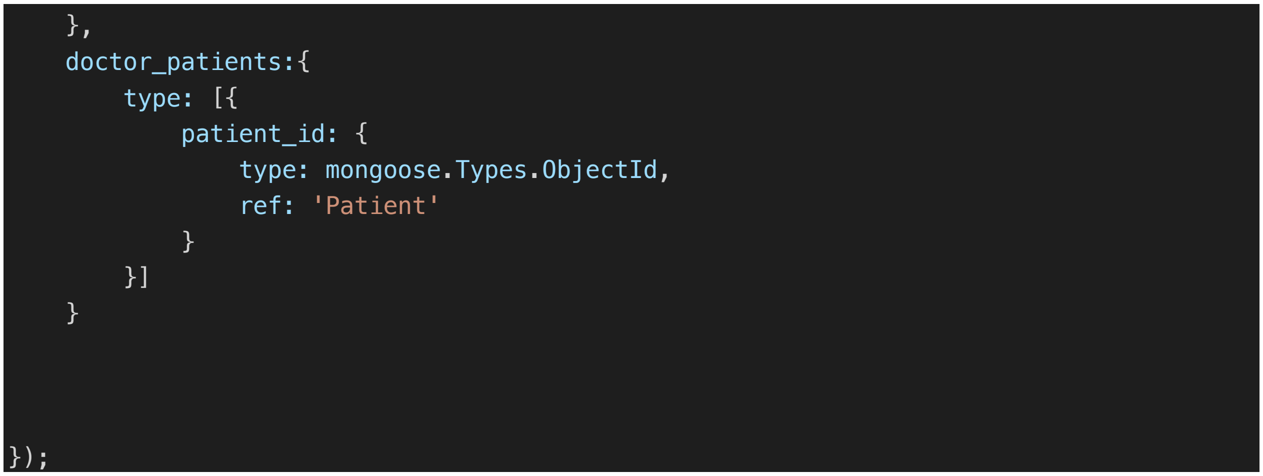


```
    },
    patient_HRCT_report_data:{
        type:String,
        require:true
    },
    patient_Ddimer_report:{
        type:Buffer,
        require:true
    },
    patient_Ddimer_report_data:{
        type:String,
        require:true
    },
    patient_crp_report:{
        type:Buffer,
        require:true
    },
    patient_crp_report_data:{
        type:String,
        require:true
    },
    patient_cp_report:{
        type:Buffer,
        require:true
    },
    patient_cp_report_data:{
        type:String,
        require:true
    },
    patient_result:{
        type:[String]
    },
    patient_suggestion:{
        type:[String]
    },
    patient_3D_model:{
        type:Buffer
    }


});




var doctorSchema = new Schema({
    doctor_name:{
        type:String,
        require:true
```



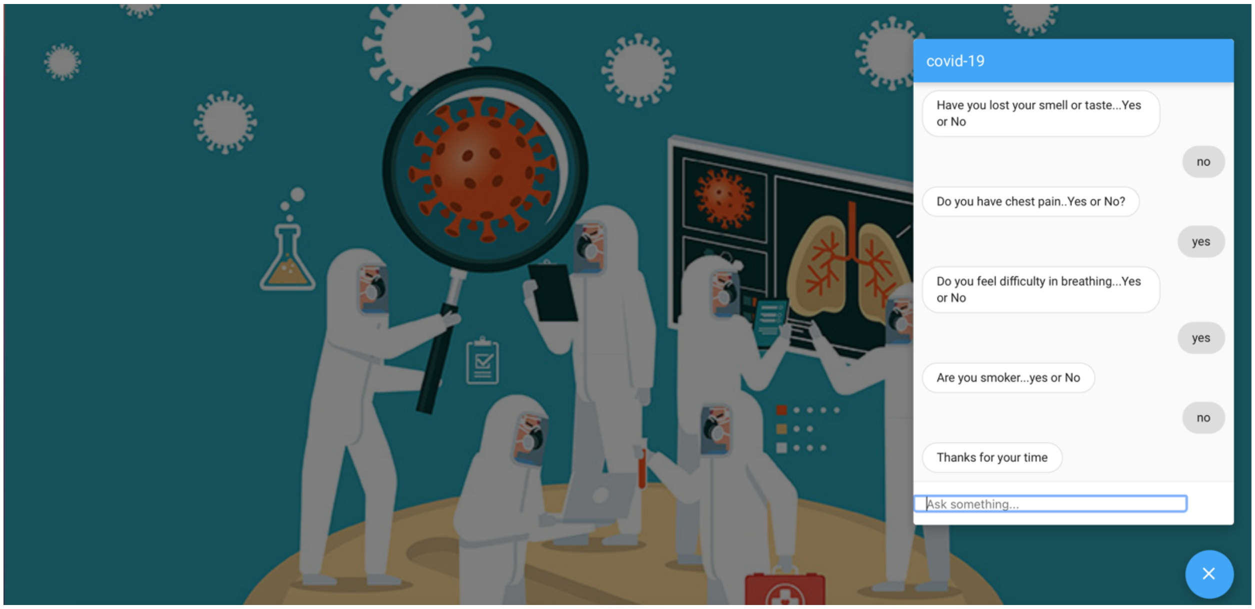
```
    },
    doctor_patients:{
        type: [{
            patient_id: {
                type: mongoose.Types.ObjectId,
                ref: 'Patient'
            }
        }]
    }



});
```

4.4 Human Interface Design

4.4.1 Screen Images



Have you lost your smell or taste...Yes or No

no

Do you have chest pain..Yes or No?

yes

Do you feel difficulty in breathing...Yes or No

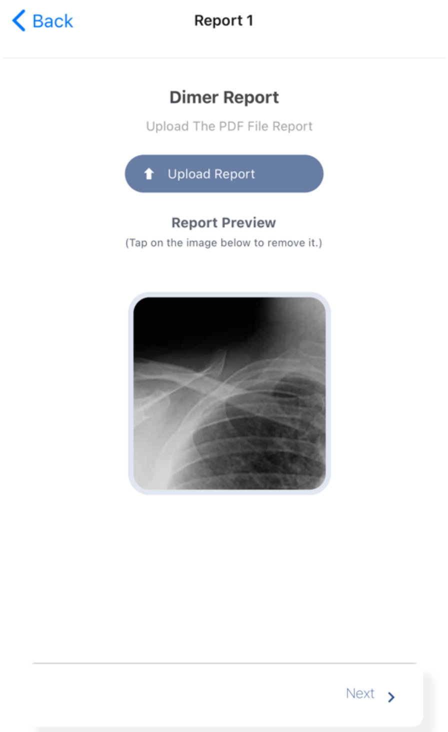
yes

Are you smoker...yes or No

no

Thanks for your time

Figure 20: Web App Chatbot



Report 1

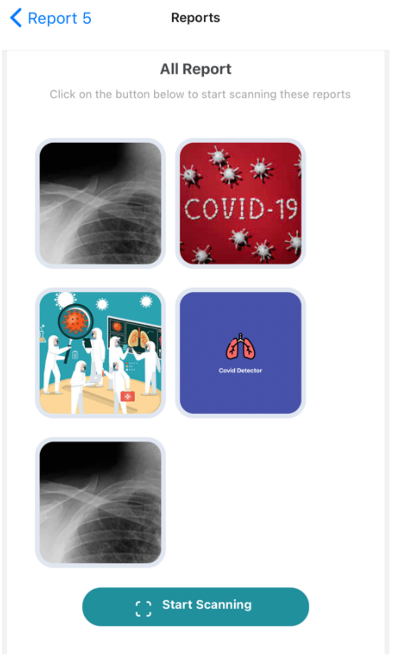
Dimer Report

Upload The PDF File Report

Report Preview

(Tap on the image below to remove it.)

Figure 21 Upload Screen Mobile App

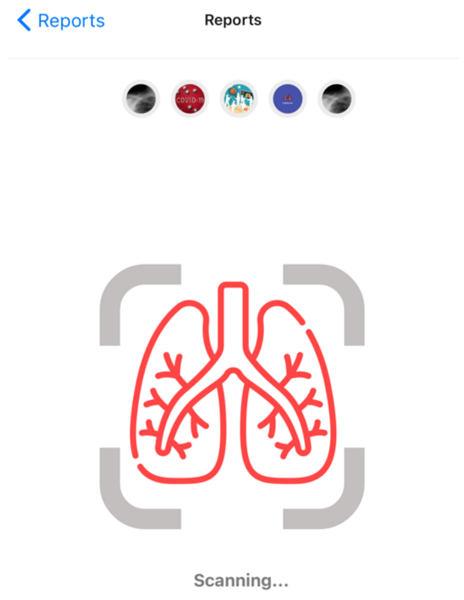


Reports

All Report

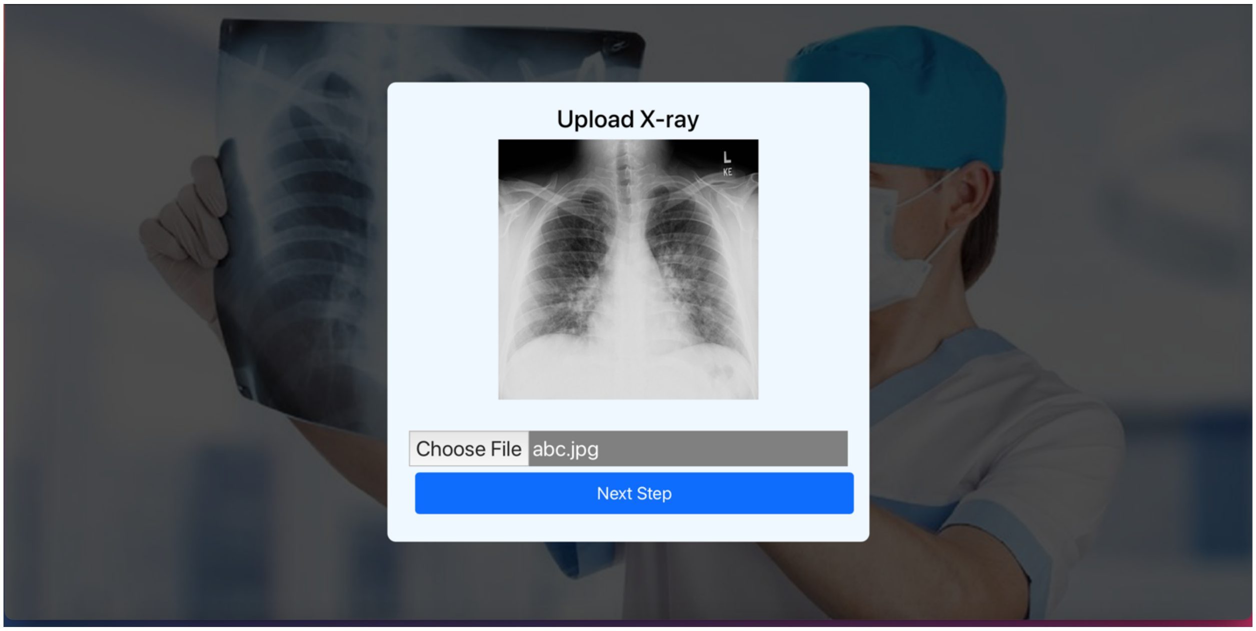
Click on the button below to start scanning these reports

Figure 22 All Reports View Mobile App



Scanning...

Figure 23: Result Screen Mobile App


Upload X-ray

abc.jpg

Figure 24: Upload Screen Web App

4.4.2 Screen Objects and Actions

4.4.2.1 Chatbot Screen:

In this screen the user will interact with the chatbot and give his details like name, age and gender etc. and give their current symptoms. And in the end the information is stored in database.

4.4.2.2 Upload X-ray image Screen:

In this screen the user clicks on the upload button and upload his chest x-ray image.

4.4.2.3 Upload HRCT Screen:

In this screen the user clicks on the upload button and upload his HRCT.

4.4.2.4 Upload X-ray report Screen:

In this screen the user clicks on the upload button and upload his X-ray report.

4.4.2.5 Upload HRCT report Screen:

In this screen the user clicks on the upload button and upload his HRCT report.

4.4.2.6 Upload D-dimer report Screen:

In this screen the user clicks on the upload button and upload his D-dimer report.

4.4.2.7 Upload CRP report Screen:

In this screen the user clicks on the upload button and upload his CRP report.

4.4.2.8 Upload Blood CP report Screen:

In this screen the user clicks on the upload button and upload his blood cp report.

4.4.2.9 Suggestion Screen:

In this screen the suggestion of the disease will be given according to the severity of the disease

4.4.2.10 3d-Visualization Screen:

In this screen the 3D model of the lungs will be displayed on which the infected areas will be displayed.

5 Implementation

5.1 Algorithm

Table 4 OCR to Fetch WBC Count

Algorithm 1 OCR to Fetch WBC count from IDC Lab report

Input: Image of blood report

Output: Returns the value of WBC count

```
def convertImage(filename):
    doc = fitz.open(filename)
    for i in range(len(doc)):
        for img in doc.get_page_images(i):
            xref = img[0]
            pix = fitz.Pixmap(doc, xref)
            if pix.n < 5:  # this is GRAY or RGB
                pix.save("p%s-%s.png" % (i, xref))
            else:  # CMYK: convert to RGB first
                pix1 = fitz.Pixmap(fitz.csRGB, pix)
                pix1.save("p%s-%s.png" % (i, xref))
                pix1 = None
            pix = None

def convertText2():
    return(pytesseract.image_to_string(Image.open("/content/p0-19.png")))

def hematology(filename):
    convertImage(filename)
    res = convertText2()
    wbc = res.split("WBC Count ")[1].split(" ")[0]
    return(wbc)

print("WBC Count: "+wbc)
```

Table 5 OCR to Fetch D-Dimer Count

Algorithm 2 OCR to Fetch D-dimer count from IDC Lab report

Input: Image of D-dimer report

Output: Returns the value of D-dimer count

```
def convertImage(filename):
    doc = fitz.open(filename)
    for i in range(len(doc)):
        for img in doc.get_page_images(i):
            xref = img[0]
            pix = fitz.Pixmap(doc, xref)
            if pix.n < 5:  # this is GRAY or RGB
                pix.save("p%s-%s.png" % (i, xref))
            else:  # CMYK: convert to RGB first
                pix1 = fitz.Pixmap(fitz.csRGB, pix)
                pix1.save("p%s-%s.png" % (i, xref))
                pix1 = None
            pix = None

def convertText():
    pipeline = keras_ocr.pipeline.Pipeline()
    images = [
        keras_ocr.tools.read(images) for images in [
            "p0-19.png"
        ]
    ]
    prediction = pipeline.recognize(images)
    text = ""
    for i in range(len(prediction[0])):
        k = prediction[0][i]
        text += k[0] + " "
    return(text)

def dDimer(filename):
    convertImage(filename)
    res = convertText()
    dDimer=res.split("ddimer")[1].strip().split(" ")[0]
    return(dDimer)

print("D-Dimer: "+dimer)
```

Table 6 OCR to Fetch CRP Count

Algorithm 3 OCR to Fetch CRP count from IDC Lab report

Input: Image of CRP report

Output: Returns the value of CRP count

```
def convertImage(filename):
    doc = fitz.open(filename)
    for i in range(len(doc)):
        for img in doc.get_page_images(i):
            xref = img[0]
            pix = fitz.Pixmap(doc, xref)
            if pix.n < 5:  # this is GRAY or RGB
                pix.save("p%s-%s.png" % (i, xref))
            else:  # CMYK: convert to RGB first
                pix1 = fitz.Pixmap(fitz.csRGB, pix)
                pix1.save("p%s-%s.png" % (i, xref))
                pix1 = None
            pix = None

def convertText():
    pipeline = keras_ocr.pipeline.Pipeline()
    images = [
        keras_ocr.tools.read(images) for images in [
            "p0-19.png"
        ]
    ]
    prediction = pipeline.recognize(images)
    text = ""
    for i in range(len(prediction[0])):
        k = prediction[0][i]
        text += k[0] + " "
    return(text)

def CRP(filename):
    convertImage(filename)
    res = convertText()
    res = res.split("rpy")[1].strip().split("negative")[0].split(" ")
    if(len(res)==1):
      crp = res[0]
    else:
      crp = res[0]+"."+res[1]
    return(crp)

print("C-Reactive Protein: "+crp)
```

Table 7 OCR to Fetch CRP Count

Algorithm 4 OCR to Fetch CRP count from CitiLab report

Input: Image of CRP report

Output: Returns the value of CRP count

```
def convertImage(filename):
    doc = fitz.open(filename)
    for i in range(len(doc)):
        for img in doc.get_page_images(i):
            xref = img[0]
            pix = fitz.Pixmap(doc, xref)
            if pix.n < 5:  # this is GRAY or RGB
                pix.save("p%s-%s.png" % (i, xref))
            else:  # CMYK: convert to RGB first
                pix1 = fitz.Pixmap(fitz.csRGB, pix)
                pix1.save("p%s-%s.png" % (i, xref))
                pix1 = None
            pix = None

def convertText():
    pipeline = keras_ocr.pipeline.Pipeline()
    images = [
        keras_ocr.tools.read(images) for images in [
            "p0-19.png"
        ]
    ]
    prediction = pipeline.recognize(images)
    text = ""
    for i in range(len(prediction[0])):
        k = prediction[0][i]
        text += k[0] + " "
    return(text)

def CRP(filename):
    convertImage(filename)
    res = convertText()
    res = res.split("rpy")[1].strip().split("negative")[0].split(" ")
    if(len(res)==1):
      crp = res[0]
    else:
      crp = res[0]+"."+res[1]
    return(crp)

print("C-Reactive Protein: "+crp)
```

Table 8 Training an ML-Model for Prediction of COVID via X-Ray

Algorithm 5 Training an ML model for prediction of Covid through X-Ray

Input: X-ray images of Covid and Normal patients

Output: ML model .h5 file

```
levels = ['Normal', 'COVID']
path = PATH OF THE DATA FOLDER
data_dir = os.path.join(path)
data = []
for id, level in enumerate(levels):
    for file in os.listdir(os.path.join(data_dir, level)):
        data.append(['{}/{}'.format(level, file), level])

data = pd.DataFrame(data, columns = ['image_file', 'corona_result'])

data['path'] = path + '/' + data['image_file']
data['corona_result'] = data['corona_result'].map({'Normal': 'Negative', 'COVID': 'Positive'})

data.head()


df = pd.DataFrame()
df['corona_result'] = ['Positive', 'Negative']
df['Count'] = [len(data[data['corona_result'] == 'Positive']), len(data[data['corona_result'] == 'Negative'])]
df = df.sort_values(by = ['Count'], ascending = False)


data['image'] = data['path'].map(lambda x: np.asarray(Image.open(x).resize((75,75))))
data.head()

all_data = []

# Storing images and their labels into a list for further Train Test split

for i in range(len(data)):
    image = cv2.imread(data['path'][i])
    image = cv2.resize(image, (70, 70)) / 255.0
    label = 1 if data['corona_result'][i] == "Positive" else 0
    all_data.append([image, label])

x = []
y = []
```

```
for image, label in all_data:
    x.append(image)
    y.append(label)

# Converting to Numpy Array
x = np.array(x)
y = np.array(y)

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 42)
x_train, x_val, y_train, y_val = train_test_split(x_train, y_train, test_size = 0.1, random_state = 4
2)

print(x_train.shape, x_test.shape, x_val.shape, y_train.shape, y_test.shape, y_val.shape)


cnn_model = models.Sequential()
cnn_model.add(layers.Conv2D(filters = 128, kernel_size = (3, 3), activation = 'relu', input_shape
 = (70, 70, 3)))
cnn_model.add(layers.MaxPooling2D((2, 2)))
cnn_model.add(layers.Dropout(0.3))

cnn_model.add(layers.Conv2D(filters = 64, kernel_size = (3, 3), activation = 'relu'))
cnn_model.add(layers.MaxPooling2D((2, 2)))
cnn_model.add(layers.Dropout(0.5))

cnn_model.add(layers.Conv2D(filters = 64, kernel_size = (3, 3), activation = 'relu'))
cnn_model.add(layers.Flatten())
cnn_model.add(layers.Dense(units = 16, activation = 'relu'))
cnn_model.add(layers.Dropout(0.2))

cnn_model.add(layers.Dense(units = 2))

cnn_model.compile(optimizer = 'adam',
          loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits = True),
          metrics = ['accuracy'])

cnn_model.summary()


es = tf.keras.callbacks.EarlyStopping(monitor = 'val_loss', mode = 'min', verbose = 1, patience =
4)

#tf.random.set_seed(42)
history = cnn_model.fit(x_train, y_train,
                    epochs = 50, batch_size = 256,
                    validation_data = (x_val, y_val),
                    callbacks = [es])


yp_train = cnn_model.predict(x_train)
yp_train = np.argmax(yp_train, axis = 1)
yp_val = cnn_model.predict(x_val)
yp_val = np.argmax(yp_val, axis = 1)
yp_test = cnn_model.predict(x_test)
yp_test = np.argmax(yp_test, axis = 1)


cnn_model.save('/content/Xray Model.h5')
```

Table 9 Checking Prediction with Trained ML-Model

Algorithm 6 Checking Prediction with trained ML model

Input: X-Ray Image of Patient

Output: Returns 0 for Normal patient or 1 for Covid patient

```
model = keras.models.load_model('/content/drive/MyDrive/FYP/X-
Ray Model CNN/Xray Model.h5')

def Prediction(i):
  z_img = cv2.imread(i)
  z_img = cv2.resize(z_img, (70, 70)) / 255.0
  z_img = z_img.reshape(1, z_img.shape[0], z_img.shape[1], z_img.shape[2])
  z = model.predict(z_img)
  z = np.argmax(z, axis = 1)
  return z[0]
```

Table 10 Training an ML-Model for Prediction of COVID through HRCT

Algorithm 7 Training an ML model for prediction of Covid through HRCT

Input: HRCT images of Covid and Normal patients

Output: ML model .h5 file

```
levels = ['Normal', 'COVID']
path = PATH OF THE DATA FOLDER
data_dir = os.path.join(path)
data = []
for id, level in enumerate(levels):
    for file in os.listdir(os.path.join(data_dir, level)):
        data.append(['{}/{}'.format(level, file), level])

data = pd.DataFrame(data, columns = ['image_file', 'corona_result'])
```

```
data['path'] = path + '/' + data['image_file']
data['corona_result'] = data['corona_result'].map({'Normal': 'Negative', 'COVID': 'Positive'})

data.head()


df = pd.DataFrame()
df['corona_result'] = ['Positive', 'Negative']
df['Count'] = [len(data[data['corona_result'] == 'Positive']), len(data[data['corona_result'] == 'Negative'])]
df = df.sort_values(by = ['Count'], ascending = False)


data['image'] = data['path'].map(lambda x: np.asarray(Image.open(x).resize((75,75))))
data.head()

all_data = []

# Storing images and their labels into a list for further Train Test split

for i in range(len(data)):
    image = cv2.imread(data['path'][i])
    image = cv2.resize(image, (70, 70)) / 255.0
    label = 1 if data['corona_result'][i] == "Positive" else 0
    all_data.append([image, label])

x = []
y = []

for image, label in all_data:
    x.append(image)
    y.append(label)

# Converting to Numpy Array
x = np.array(x)
y = np.array(y)

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 42)
x_train, x_val, y_train, y_val = train_test_split(x_train, y_train, test_size = 0.1, random_state = 42)

print(x_train.shape, x_test.shape, x_val.shape, y_train.shape, y_test.shape, y_val.shape)


cnn_model = models.Sequential()
```

```
cnn_model.add(layers.Conv2D(filters = 128, kernel_size = (3, 3), activation = 'relu', input_shape
 = (70, 70, 3)))
cnn_model.add(layers.MaxPooling2D((2, 2)))
cnn_model.add(layers.Dropout(0.3))

cnn_model.add(layers.Conv2D(filters = 64, kernel_size = (3, 3), activation = 'relu'))
cnn_model.add(layers.MaxPooling2D((2, 2)))
cnn_model.add(layers.Dropout(0.5))

cnn_model.add(layers.Conv2D(filters = 64, kernel_size = (3, 3), activation = 'relu'))
cnn_model.add(layers.Flatten())
cnn_model.add(layers.Dense(units = 16, activation = 'relu'))
cnn_model.add(layers.Dropout(0.2))

cnn_model.add(layers.Dense(units = 2))

cnn_model.compile(optimizer = 'adam',
        loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits = True),
        metrics = ['accuracy'])

cnn_model.summary()


es = tf.keras.callbacks.EarlyStopping(monitor = 'val_loss', mode = 'min', verbose = 1, patience =
4)

#tf.random.set_seed(42)
history = cnn_model.fit(x_train, y_train,
                  epochs = 50, batch_size = 256,
                  validation_data = (x_val, y_val),
                  callbacks = [es])


yp_train = cnn_model.predict(x_train)
yp_train = np.argmax(yp_train, axis = 1)
yp_val = cnn_model.predict(x_val)
yp_val = np.argmax(yp_val, axis = 1)
yp_test = cnn_model.predict(x_test)
yp_test = np.argmax(yp_test, axis = 1)


cnn_model.save('/content/HRCT Model.h5')
```

Table 11 Checking Prediction with Trained ML-Model

Algorithm 8 Checking Prediction with trained ML model

Input: HRCT Image of Patient

Output: Returns 0 for Normal patient or 1 for Covid patient

```
model = keras.models.load_model('/content/drive/MyDrive/FYP/HRCT Model -
 CNN/HRCT Model.h5')

def Prediction(i):
  z_img = cv2.imread(i)
  z_img = cv2.resize(z_img, (70, 70)) / 255.0
  z_img = z_img.reshape(1, z_img.shape[0], z_img.shape[1], z_img.shape[2])
  z = model.predict(z_img)
  z = np.argmax(z, axis = 1)
  return z[0]
```

6 Testing and Evaluation

6.1 Unit Testing

Unit Testing 1: Optical character recognition for reports

Testing Objective: to ensure correct input file for OCR

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Report uploaded in pdf format	Filename: bloodcp.pdf	Report information in text form	Pass
2	Report Uploaded in image format	Filename: bloodcp.png	System will display wrong input file message	Pass
3	Report path null	Filename: blooddcp.pdf	System will display “no file found”	Pass

Unit Testing 2: Detection of COVID-19
Testing Objective: To ensure correct input for accurate results

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Image uploaded in (jpeg,png)	Filename: CTScan.png	COVID Prediction results	Pass
2	Image Uploaded in wrong format	Filename: X-ray.docx	System will display message "Enter correct file format"	Pass
3	Image path null	Filename: CTScaan.png	System will display "no file found"	Pass

Unit Testing 3: Gathering patient symptoms
Testing Objective: to gather patient symptoms for prediction

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Patient Enter Symptoms	Symptoms: cough, fever, shortness of breath	System will add the symptoms to database	Pass
2	Patient have no symptoms	Symptoms: no symptoms	System will add information to database	Pass
3	Symptoms not added	Symptoms: ""	System will prompt a message "Please enter symptoms"	Pass

Unit Testing 4: Gathering patient information
Testing Objective: for providing patient with their respective information

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Patient Enter personal information	Email: tayyab@gmail.com Contact: 0345-2682982	System will add the information to database	Pass

2	Patient does not enter information	Email: “”	System will show message “please add necessary details”	Pass
3	Patient Provides invalid Email	Email: “tayyab123gmail.com”	System will display “Invalid email format”	Pass
4	Patient Provides invalid contact	Email: 1262-2182671	System will display “Invalid Phone Number”	Pass

Unit Testing 5: Upload Blood Cp report
Testing Objective: To check correct upload of CP report

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Report uploaded in pdf format	Filename: bloodcp.pdf	Report successfully uploaded	Pass
2	Report Uploaded in image format	Filename: bloodcp.png	System will display wrong input file message	Pass
3	Report path null	Filename: blooddcp.pdf	System will display “no file found”	Pass
4	No file Uploaded	Filename: “null”	System will display “please select a file”	Pass

Unit Testing 6: Upload D Dimer report
Testing Objective: To check correct upload of D Dimer report

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Report uploaded in pdf format	Filename: D_Dimer.pdf	Report successfully uploaded	Pass
2	Report Uploaded in image format	Filename: D_Dimer.png	System will display wrong input file message	Pass
3	Report path null	Filename: D_Diiimer.pdf	System will display “no file found”	Pass
4	No file Uploaded	Filename: “null”	System will display “please select a file”	Pass

Unit Testing 7: Upload CRP report
Testing Objective: To check correct upload of CRP report

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Report uploaded in pdf format	Filename: CRP.pdf	Report successfully uploaded	Pass
2	Report Uploaded in image format	Filename: CRP.png	System will display wrong input file message	Pass
3	Report path null	Filename: CRRP.pdf	System will display "no file found"	Pass
4	No file Uploaded	Filename: "null"	System will display "please select a file"	Pass

Unit Testing 8: Upload CT scan
Testing Objective: To check correct upload of CT Scan

No.	Test case/Test script	Attribute and value	Expected result	Result
1	CT Scan uploaded in DICOM format	Filename: CTi1.DICOM	CTScan successfully uploaded	Pass
2	CT Scan uploaded in jpg format	Filename: CTi1.jpg	CTScan successfully uploaded	Pass
3	One File uploaded	Filename: CTi1.jpg	CTScan successfully uploaded	Pass
4	Multiple Files uploaded	Filename:CTi1.jpg,CTi2.jpg	CTScan uploaded fail	Pass
5	CT Scan uploaded in DICOM format	Filename: CTi1.DICOM	CTScan successfully converted to jpg	Pass
6	CT Scan uploaded in not DICOM or jpg format	Filename: CTi1.png	CTScan upload fail	Pass

Unit Testing 9: Upload CT scan Report
Testing Objective: To check correct upload of CT Scan Report

No.	Test case/Test script	Attribute and value	Expected result	Result
1	CT Scan Report uploaded in pdf format	Filename: CTi1report.pdf	Report successfully uploaded	Pass
2	CT Scan Report uploaded in wrong format	Filename: CTi1report.docx	Report upload fail	Pass

Unit Testing 10: Upload Xray
Testing Objective: To check correct upload of Xray

No.	Test case/Test script	Attribute and value	Expected result	Result
1	X-Ray uploaded in DICOM format	Filename: CTi1.DICOM	X-Ray successfully uploaded	Pass
2	Xray uploaded in jpg format	Filename: CTi1.jpg	X-Ray successfully uploaded	Pass
3	One File uploaded	Filename: CTi1.jpg	X-Ray successfully uploaded	Pass
4	Multiple Files uploaded	Filename:CTi1.jpg,CTi2.jpg	X-Ray uploaded fail	Pass
5	Xray uploaded in DICOM format	Filename: CTi1.DICOM	X-Ray successfully converted to jpg	Pass
6	X-Ray uploaded in not DICOM or jpg format	Filename: CTi1.png	X-Ray upload fail	Pass

Unit Testing 11: Upload X-Ray Report
Testing Objective: To check correct upload of X-Ray Report

No.	Test case/Test script	Attribute and value	Expected result	Result
1	X-Ray Report uploaded in pdf format	Filename: CTi1report.pdf	Report successfully uploaded	Pass
2	X-Ray Report uploaded in wrong format	Filename: CTi1report.docx	Report upload fail	Pass

Unit Testing 12: Gathering patient Medical history
Testing Objective: To gather patient Medical for prediction

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Patient Enter History	History: BP,Heart	System will add the history to database	Pass
2	Patients have no History	History: no history	System will add history to database	Pass
3	History not added	Symptoms: “”	System will prompt a message “Please enter history”	Pass

Unit Testing 13: Data extraction after OCR
Testing Objective: To extract data from image converted text

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Fully detailed converted report	String[…..data]	System will apply string operations to the string	Pass
2	Converted report data	String[null]	System will	Pass

	null		prompt an error message after applying string operation	

Unit Testing 14: Storage of Extracted data
Testing Objective: To store data from the data extraction

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Data is extracted from the string operations	Data: crp,2.34,full	System will add data to database	Pass
2	No data extracted	Data: undefined	System will display a message	Pass

Unit Testing 15: Gray Scaling
Testing Objective: To ensure the Gray scaling of image for future use

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Basic image is provided to system	Data: i1.jpg	System will produce a grey scale image	Pass
2	Filtered image	Data:i2.jpg	System will produce a grey scale image	Pass

Unit Testing 16: Image Resizing
Testing Objective: To ensure the Image resizing of image for future use

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Basic image is provided to system	Data: i1.jpg	System will produce a resized image	Pass
2	Filtered image	Data:i2.jpg	System will produce a resized	Pass

			image	

Unit Testing 17: Image Orientation
Testing Objective: To ensure the Image is in one correct orientation

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Image is provided in portrait	Data: i1.jpg	System will return the same image back	Pass
2	Image is provided in landscape	Data:i2.jpg	System will return the portrait image	Pass

Unit Testing 18: Trained ML Model
Testing Objective: To ensure the Model for detection is working.

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Image with COVID is provided	Data: i1.jpg	System will Return Yes answer	Pass
2	Image with no COVID is provided	Data:i2.jpg	System will return No answer	Pass

Unit Testing 19: Create suggestions
Testing Objective: To ensure the system is providing suggestions for the result.

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Prediction and detection results provided with COVID	Detected: YES Prediction: 24/60	Suggestions: Need to rest, Take medicine and nebulize	Pass
2	Prediction and detection results provided with no COVID	Detected: NO	No suggestions	Pass

Unit Testing 20: View suggestions

Testing Objective: To ensure the User can view suggestions for the result.

No.	Test case/Test script	Attribute and value	Expected result	Result
1	User Checks for suggestions from portal	Clicks on Report suggestion tab	Suggestions: Need to rest, Take medicine and nebulize	Pass
2	No COVID report	Clicks on Report Suggestion Tab	No suggestions	Pass

Unit Testing 21: Generate 3D Model

Testing Objective: To ensure the 3D Model is generated according to the data and images

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Report and Image provided	Ctreport.pdf, cti1.jpg	System generates a 3D model	Pass
2	Only image Provided	Cti2.jpg	System generates a 3D model	Pass

Unit Testing 22: Display 3D Model

Testing Objective: To ensure the 3D Model can be successfully viewed according to the data and images

No.	Test case/Test script	Attribute and value	Expected result	Result
1	3D model uploaded on patients portal	Model.obj	System display 3D model to patient	Pass

Unit Testing 23: Prediction of lungs condition
Testing Objective: To ensure that prediction is generated after detection

No.	Test case/Test script	Attribute and value	Expected result	Result
1	COVID Detection	Cti1.jpg	System predicts using prediction function	Pass
2	No COVID detected	Cti2.jpg	System does not call the prediction functions	Pass

Unit Testing 24: lungs condition progress
Testing Objective: To ensure that system is able to detect lung progress after some interval of time

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Image provided to Model	cti1.jpg	System displays new prediction along with the old one	Pass

6.2 Functional Testing

Functional Testing 1: Image to text conversion
Testing Objective: to ensure correct input file for OCR

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1	Report uploaded in pdf format	Filename: bloodcp.pdf	Report information in text form	Report information	Pass
2	Report Uploaded in image format	Filename: bloodcp.png	System will display wrong input file message	Message: “wrong input file”	Pass
3	Report path null	Filename: blooddcp.pdf	System will display “no file	Message: “no file	Pass

			found”	found”	

Functional Testing 2: Generate results
Testing Objective: To ensure correct input for accurate results

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1	Image uploaded in (jpeg,png)	Filename: CTScan.png	COVID Prediction results	Prediction result	Pass
2	Image Uploaded in wrong format	Filename: X-ray.docx	System will display message “Enter correct file format”	Message: “wrong input file”	Pass
3	Image path null	Filename: CTScaan.png	System will display “no file found”	Message: “no file found”	Pass

Functional Testing 3: Gathering patient symptoms
Testing Objective: to gather patient symptoms for prediction

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Patient Enter Symptoms	Symptoms: cough, fever, shortness of breath	System will add the symptoms to database	“Symptoms added to database”	Pass
2	Patient have no symptoms	Symptoms: no symptoms	System will add information to database	“Symptoms added to database”	Pass
3	Symptoms not added	Symptoms: “”	System will prompt a message “Please enter symptoms”	Message: “Please enter symptoms”	Pass

Functional Testing 4: Gathering patient information
Testing Objective: for providing patient with their respective information

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1	Patient Enter personal information	Email: tayyab1314@gmail.com Contact: 0345-2682982	System will add the information to database	"Information added to database"	Pass
2	Patient does not enter information	Email: ""	System will show message "please add necessary details"	Message: "Please add necessary details"	Pass
3	Patient Provides invalid Email	Email: "tayyab123gmail.com"	System will display "Invalid email format"	Message: "Invalid email format "	Pass
4	Patient Provides invalid contact	Email: 1262-2182671	System will display "Invalid Phone Number"	Message: "Invalid phone number"	Pass

Functional Testing 5: Upload Blood Cp report
Testing Objective: To check correct upload of CP report

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1	Report uploaded in pdf format	Filename: bloodcp.pdf	Report successfully uploaded	Report uploaded successful	Pass
2	Report Uploaded in image format	Filename: bloodcp.png	System will display wrong input file message	Message: "wrong input file"	Pass
3	Report path null	Filename: blooddcp.pdf	System will display "no file found"	Message: "no file found"	Pass

4	No file Uploaded	Filename: “null”	System will display “please select a file”	Message: “Please select a file”	Pass

Functional Testing 6: Upload D Dimer report
Testing Objective: To check correct upload of D Dimer report

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1	Report uploaded in pdf format	Filename: D_Dimer.pdf	Report successfully uploaded	Report uploaded successful	Pass
2	Report Uploaded in image format	Filename: D_Dimer.png	System will display wrong input file message	Message: “wrong input file”	Pass
3	Report path null	Filename: D_Diiimer.pdf	System will display “no file found”	Message: “no file found”	Pass
4	No file Uploaded	Filename: “null”	System will display “please select a file”	Message: “Please select a file”	Pass

Functional Testing 7: Image Processing
Testing Objective: to Ensure correct image input is given to model for training

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1	Image uploaded in wrong orientation	Filename: x-ray.png Orientation: landscape	Covert orientation to portrait	Image conversion successful	Pass
2	Images uploaded in different size	Filename: X-ray.png Filename: x-ray1.png	Convert image size to (224,224)	Image size conversion successful	Pass
3	Image noise	Filename: CT_scan.png	Noise from image reduced	Image noise reduced	Pass

4	Adjust Exposure	Filename: CT-Scan.png Filename: Ct_scan1.png	Exposure adjusts to 2	Image exposure adjusted	Pass

Functional Testing 8: Upload CT scan
Testing Objective: To check correct upload of CT Scan

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	CT Scan uploaded in DICOM format	Filename: CTi1.DICOM	CTScan successfully uploaded	CTScan successfully uploaded	Pass
2	CT Scan uploaded in jpg format	Filename: CTi1.jpg	CTScan successfully uploaded	CTScan successfully uploaded	Pass
3	One File uploaded	Filename: CTi1.jpg	CTScan successfully uploaded	CTScan successfully uploaded	Pass
4	Multiple Files uploaded	Filename:CTi1.jpg,CTi2.jpg	CTScan uploaded fail	CTScan uploaded fail	Pass
5	CT Scan uploaded in DICOM format	Filename: CTi1.DICOM	CTScan successfully converted to jpg	CTScan successfully converted to jpg	Pass
6	CT Scan uploaded in not DICOM or jpg format	Filename: CTi1.png	CTScan upload fail	CTScan upload fail	Pass

Functional Testing 9: Upload CT scan Report
Testing Objective: To check correct upload of CT Scan Report

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	CT Scan Report uploaded in pdf format	Filename: CTi1report.pdf	Report successfully uploaded	Report successfully uploaded	Pass

2	CT Scan Report uploaded in wrong format	Filename: CTi1report.docx	Report upload fail	Report upload fail	Pass

Functional Testing 10: Upload Xray
Testing Objective: To check correct upload of Xray

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	X-Ray uploaded in DICOM format	Filename: CTi1.DICOM	X-Ray successfully uploaded	X-Ray successfully uploaded	Pass
2	Xray uploaded in jpg format	Filename: CTi1.jpg	X-Ray successfully uploaded	X-Ray successfully uploaded	Pass
3	One File uploaded	Filename: CTi1.jpg	X-Ray successfully uploaded	X-Ray successfully uploaded	Pass
4	Multiple Files uploaded	Filename:CTi1.jpg,CTi2.jpg	X-Ray uploaded fail	X-Ray uploaded fail	Pass
5	Xray uploaded in DICOM format	Filename: CTi1.DICOM	X-Ray successfully converted to jpg	X-Ray successfully converted to jpg	Pass
6	X-Ray uploaded in not DICOM or jpg format	Filename: CTi1.png	X-Ray upload fail	X-Ray upload fail	Pass

Functional Testing 11: Upload X-Ray Report
Testing Objective: To check correct upload of X-Ray Report

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	X-Ray Report uploaded in pdf	Filename: CTi1report.pdf	Report successfully	Report successfully	Pass

	format		uploaded	uploaded	
2	X-Ray Report uploaded in wrong format	Filename: CTi1report.docx	Report upload fail	Report upload fail	Pass

Functional Testing 12: Gathering patient Medical history
Testing Objective: To gather patient Medical for prediction

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Patient Enter History	History: BP,Heart	System will add the history to database	System will add the history to database	Pass
2	Patients have no History	History: no history	System will add history to database	System will add history to database	Pass
3	History not added	Symptoms: “”	System will prompt a message “Please enter history”	System will prompt a message “Please enter history”	Pass

Functional Testing 13: Data extraction after OCR
Testing Objective: To extract data from image converted text

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Fully detailed converted report	String[…..data]	System will apply string operations to the string	System will apply string operations to the string	Pass
2	Converted report data null	String[null]	System will prompt an error message after applying string operation	System will prompt an error message after	Pass

				applying string operation	

Functional Testing 14: Storage of Extracted data
Testing Objective: To store data from the data extraction

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1	Data is extracted from the string operations	Data: crp,2.34, full	System will add data to database	System will add data to database	Pass
2	No data extracted	Data: undefined	System will display a message	System will display a message	Pass

Functional Testing 15: Gray Scaling
Testing Objective: To ensure the Gray scaling of image for future use

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Basic image is provided to system	Data: i1.jpg	System will produce a grey scale image	System will produce a grey scale image	Pass
2	Filtered image	Data:i2.jpg	System will produce a grey scale image	System will produce a grey scale image	Pass

Functional Testing 16: Image Resizing
Testing Objective: To ensure the Image resizing of image for future use

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Basic image is provided to system	Data: i1.jpg	System will produce a	System will produce a	Pass

			resized image	resized image	
2	Filtered image	Data:i2.jpg	System will produce a resized image	System will produce a resized image	Pass

Functional Testing 17: Image Orientation
Testing Objective: To ensure the Image is in one correct orientation

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Image is provided in portrait	Data: i1.jpg	System will return the same image back	System will return the same image back	Pass
2	Image is provided in landscape	Data:i2.jpg	System will return the portrait image	System will return the portrait image	Pass

Functional Testing 18: Trained ML Model
Testing Objective: To ensure the Model for detection is working.

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Image with COVID is provided	Data: i1.jpg	System will Return Yes answer	System will Return Yes answer	Pass
2	Image with no COVID is provided	Data:i2.jpg	System will return No answer	System will return No answer	Pass

Functional Testing 19: Create suggestions

Testing Objective: To ensure the system is providing suggestions for the result.

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Prediction and detection results provided with COVID	Detected: YES Prediction: 24/60	Suggestions: Need to rest, Take medicine and nebulize	Suggestions: Need to rest, Take medicine and nebulize	Pass
2	Prediction and detection results provided with no COVID	Detected: NO	No suggestions	No suggestions	Pass

Functional Testing 20: View suggestions

Testing Objective: To ensure the User can view suggestions for the result.

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	User Checks for suggestions from portal	Clicks on Report suggestion tab	Suggestions: Need to rest, Take medicine and nebulize	Suggestions: Need to rest, Take medicine and nebulize	Pass
2	No COVID report	Clicks on Report Suggestion Tab	No suggestions	No suggestions	Pass

Functional Testing 21: Generate 3D Model

Testing Objective: To ensure the 3D Model is generated according to the data and images

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Report and Image provided	Ctreport.pdf, cti1.jpg	System generates a 3D model	System generates a 3D model	Pass

2	Only image Provided	Cti2.jpg	System generates a 3D model	System generates a 3D model	Pass

Functional Testing 22: Display 3D Model

Testing Objective: To ensure the 3D Model can be successfully viewed according to the data and images

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	3D model uploaded on patients portal	Model.obj	System display 3D model to patient	System display 3D model to patient	Pass

Functional Testing 23: Prediction of lungs condition

Testing Objective: To ensure that prediction is generated after detection

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	COVID Detection	Cti1.jpg	System predicts using prediction function	System predicts using prediction function	Pass
2	No COVID detected	Cti2.jpg	System does not call the prediction functions	System does not call the prediction functions	Pass

Functional Testing 24: Lungs condition progress

Testing Objective: To ensure that system is able to detect lung progress after some interval of time

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Image provided to Model	cti1.jpg	System displays new prediction along with the old	System displays new prediction	Pass

			one	along with the old one	

Functional Testing 25: Storing CT Scan
Testing Objective: To ensure that system stores Uploaded CT Scan in database

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Image uploaded on System	cti1.jpg	System stores Ct scan on connected database	System stores Ct scan on connected database	Pass

Functional Testing 26: Retrieving CT scan
Testing Objective: To ensure that system retrieves CT scan from database

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Image Exists in System	cti1.jpg	System retrieves image path from database	System retrieves image path from database	Pass

Functional Testing 27: Storing CT Scan Report
Testing Objective: To ensure that system stores Uploaded CT Scan Report in database

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Report uploaded on System	cti1.pdf	System stores Ct scan report on connected database	System stores Ct scan report on connected database	Pass

Functional Testing 28: Retrieving CT scan Report
Testing Objective: To ensure that system retrieves CT scan Report from database

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Reports Exists in System	cti1.jpg	System retrieves report from database	System retrieves report from database	Pass

Functional Testing 29: Storing X ray
Testing Objective: To ensure that system stores Uploaded Xray in database

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Image uploaded on System	cti1.jpg	System stores X ray on connected database	System stores Xray on connected database	Pass

Functional Testing 30: Retrieving Xray
Testing Objective: To ensure that system retrieves X ray from database

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Image Exists in System	cti1.jpg	System retrieves image path from database	System retrieves image path from database	Pass

Functional Testing 31: Storing Xray Report
Testing Objective: To ensure that system stores Uploaded X ray Report in database

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Report uploaded on System	cti1.pdf	System stores Xray report on	System stores Xray report on	Pass

			connected database	connected database	

Functional Testing 32: Retrieving Xray Report
Testing Objective: To ensure that system retrieves CT scan Report from database

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Reports Exists in System	cti1.jpg	System retrieves report from database	System retrieves report from database	Pass

Functional Testing 33: Storing blood CP test Result
Testing Objective: To ensure that system stores blood CP test Result in database

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	blood CP test Result uploaded on System	Result: 3.14	System stores blood CP test Result link on connected database	System stores blood CP test Result link on connected database	Pass

Functional Testing 34: Retrieving blood CP test Result
Testing Objective: To ensure that system retrieves blood CP test Result from database

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	blood CP test Result Exists in System	Result: 3.14	System retrieves blood CP test Result from database	System retrieves blood CP test Result from database	Pass

Functional Testing 35: Storing d-dimer test Result

Testing Objective: To ensure that system stores d-dimer test Result in database

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	d-dimer test Result uploaded on System	Result: 3.14	System stores d-dimer test Result link on connected database	System stores d-dimer test Result link on connected database	Pass

Functional Testing 36: Retrieving d-dimer test Result

Testing Objective: To ensure that system retrieves d-dimer Result from database

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	d-dimer test Result Exists in System	Result: 3.14	System retrieves d-dimer test Result from database	System retrieves d-dimer test Result from database	Pass

Functional Testing 37: Adjusting Random Exposure

Testing Objective: To ensure that system Adjusts random exposure on system

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Image provided to function	CTi1.jpg	Random exposure output image return to system	Random exposure output image return to system	Pass

Functional Testing 38: Adjusting Random Noise
Testing Objective: To ensure that system Adjusts random noise on system

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Image provided to function	CTi1.jpg	Random noise output image return to system	Random noise output image return to system	Pass

Functional Testing 39: Zoom 3D model
Testing Objective: To ensure that system allow user to zoom model

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	User use zoom button to zoom on model	CTi1.jpg	System zooms model display	System zooms model display	Pass

Functional Testing 40: Move Slider
Testing Objective: To ensure that system allow user to slide through model

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	User use slider to slide on model page	CTi1.jpg	System slides model display	System zooms model display	Pass

6.3 Integration Testing

Integration Testing 1: Prediction and notification through user information
Testing Objective: To ensure that data received from chatbot can be used for detection of COVID-19

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Patient information uploaded from chatbot	Email: sajjad@gmail.com Contact: 0312-1234632	System will notify the prediction results to user through email and number	Predict results sent to Sajjad@gmail.com and 03121234632	Pass
2.	Patient symptoms uploaded	Symptoms: cover, fever, shortness of breath	System will predict COVID-19 through report results	COVID-19 predicted	Pass

Integration Testing 2: Report Analysis for prediction
Testing Objective: To analysis reports for helping in prediction of COVID-19

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Reports Uploaded	Blood_cp: bloodcp.pdf CRP: CRP.pdf D_Dimer: D_Dimer.pdf	System will predict COVID-19 by Report information extracted through OCR	COVID-19 successfully predicted	Pass

Integration Testing 3: AI Based Chatbot
Testing Objective: To Gather patient information for prediction purposes

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result

1.	Gathering patient information	Patient personal information: Sajjad,21,0300-21892887 Patient history: Asthma,sugar Patient symptoms: Shortage of breath, blackout	Patient information will be stored in the database	Patient information will be stored in the database	Pass

Integration Testing 4: Image processing
Testing Objective: To process image to be used for detection

No.	**Test case/Test script**	**Attribute and value**	**Expected result**	**Actual result**	**Result**
1.	Processing uploaded picture	Cti2.jpg	Processed image will be stored in database	Processed image will be stored in database	Pass

Integration Testing 5: Suggestions for overcoming
Testing Objective: To ensure system suggests patient for healthy recovery

No.	**Test case/Test script**	**Attribute and value**	**Expected result**	**Actual result**	**Result**
1.	Prediction results uploaded for suggestions	Cti2.pdf	Suggestions: need to take rest, nebulize daily	Suggestions: need to take rest, nebulize daily	Pass

Integration Testing 6: 3D visualizing lung disease
Testing Objective: To ensure system provides 3D image of lungs

No.	**Test case/Test script**	**Attribute and value**	**Expected result**	**Actual result**	**Result**
1.	COVID detected image results	Cti2.obj	System displays 3D image of lung condition	System displays 3D image of lung condition	Pass

7 Conclusion and Future Work

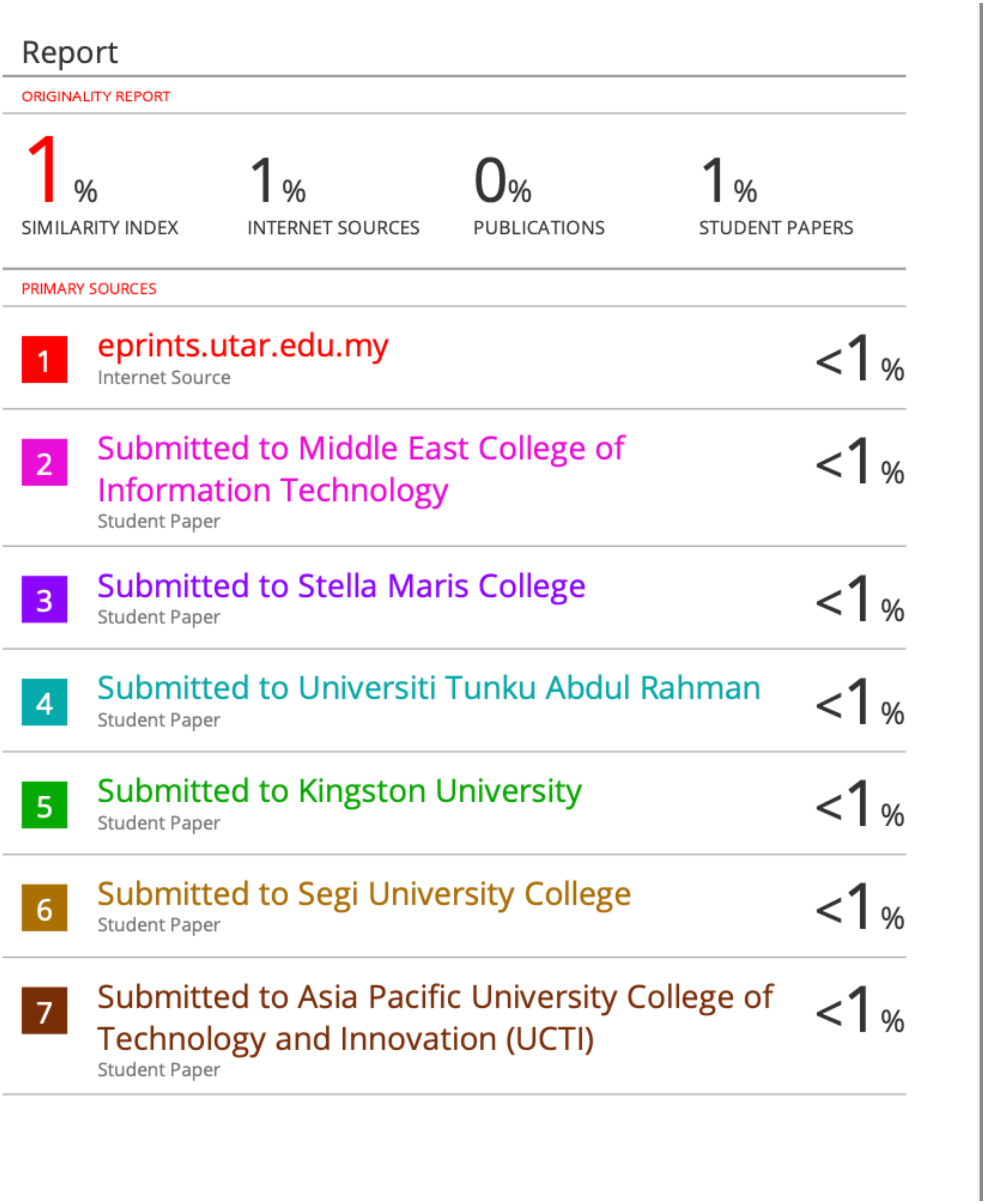
7.1 Conclusion

This document provides an overview of the project's implementation in progress. It briefly explains the software; motivation behind it, its design, functional and non-functional requirements, user interface and external APIs used.

For the People and Doctors who wants a seamless platform for the Prediction and Suggestions of Covid-19 of the patients. This will provide them with a better chance of diagnosing the disease and then suggest the patient on what to do next. The accuracy will be increased using multiple things such as Xray, HRCT, and other medical reports.

7.2 Future Work

In future we will develop this idea into a proper product and try to make it more efficient. Since for now our scope is limited and so is our datasets, we will expand its usability by training for more extensive and multiple datasets.



Report

ORIGINALITY REPORT

1%	1%	0%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.utar.edu.my Internet Source	<1%
2	Submitted to Middle East College of Information Technology Student Paper	<1%
3	Submitted to Stella Maris College Student Paper	<1%
4	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1%
5	Submitted to Kingston University Student Paper	<1%
6	Submitted to Segi University College Student Paper	<1%
7	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%