

**University of Mumbai**

**Covid-19 Disease Detection with Explainable  
Artificial Intelligence**

Submitted in partial fulfillment of requirements  
For the degree of

**Bachelors in Technology**

by

**Jai Mehta**

**1814036**

**Vishant Mehta**

**1814038**

**Vighnesh Naik**

**1814041**

**Pankti Nanavati**

**1814045**

Guide:

**Prof. Nilkamal More**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**K. J. Somaiya College of Engineering, Mumbai-77**

**(Autonomous College Affiliated to University of Mumbai)**

**Batch 2018-2022**

# **K. J. Somaiya College of Engineering, Mumbai-77**

(Autonomous College Affiliated to University of Mumbai)

## **Certificate**

This is to certify that the dissertation report entitled **Covid-19 Disease Detection with Explainable Artificial Intelligence** is bona fide record of the dissertation work done by **Jai Mehta, Vishant Mehta, Vighnesh Naik, Pankti Nanavati** in the year 2021-22 under the guidance of **Prof. Nilkamal More** of Department of Information Technology in partial fulfillment of requirement for Bachelors in Technology degree in Information Technology of University of Mumbai.

---

**Prof. Nilkamal More**  
Guide

---

**Dr. Ujwala Bhangale**  
Head of the Department

---

**Dr. Shubha Pandit**  
Principal

Date:

Place: Mumbai-77

**K. J. Somaiya College of Engineering, Mumbai-77**

(Autonomous College Affiliated to University of Mumbai)

**Certificate of Approval of Examiners**

We certify that the dissertation report entitled **Covid-19 Disease Detection with Explainable Artificial Intelligence** is bona fide record of Project work done by **Jai Mehta, Vishant Mehta, Vighnesh Naik, Pankti Nanavati**. This project is approved for the award of Bachelors in Technology Degree in Information Technology of University of Mumbai.

---

**Internal Examiner**

---

**External Examiner**

Date:

Place: Mumbai-77

# **K. J. Somaiya College of Engineering, Mumbai-77**

(Autonomous College Affiliated to University of Mumbai)

## **DECLARATION**

We declare that this written report submission represents the work done based on our and / or others' ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/ matter in our submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

Jai Mehta	Vishant Mehta
<b>Signature of the Student</b>	<b>Signature of the Student</b>
1814036	1814038
<b>Roll No.</b>	<b>Roll No.</b>
Vighnesh Naik	Pankti Nanavati
<b>Signature of the Student</b>	<b>Signature of the Student</b>
1814041	1814045
<b>Roll No.</b>	<b>Roll No.</b>

Date:

Place: Mumbai-77

## **Abstract**

There exist projects which take into consideration either textual data or image data to predict the presence of a Covid-19 disease in humans. In our project, we plan to include CT-Scan Images, to predict the disease. Since it many times happen that symptoms don't show the presence of disease but the CT-Scan shows a completely different picture. There is a high requirement to increase the transparency of the algorithms used in the prediction. In fields like healthcare, scientists often use various methods of predictions but often they also do not know the functions at the back end and thus the algorithms are referred to as enclosed within a black box. The aim of the explainable AI is to address this black box and help increase the transparency in the system, thereby increasing the trustworthiness of the system.

***Keywords : Prescription, Covid-19, Web Development, Machine Learning, Explainable Artificial Intelligence.***

# TABLE OF CONTENTS

<b>List of Figures</b>	<b>ii</b>
<b>List of Tables</b>	<b>iii</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Problem Definition . . . . .	2
1.2 Motivation of the thesis . . . . .	2
1.3 Scope of the thesis . . . . .	2
1.3.1 Description . . . . .	2
1.3.2 Criteria . . . . .	3
1.4 Salient Contribution . . . . .	3
1.5 Organization of the Thesis . . . . .	3
<b>2 Literature Survey</b>	<b>4</b>
<b>3 Software Project Management Plan</b>	<b>6</b>
3.1 Introduction . . . . .	6
3.1.1 Project Overview . . . . .	6
3.1.2 Project Deliverables . . . . .	6
3.2 Project Organization . . . . .	7
3.2.1 Software Process Models . . . . .	7
3.2.2 Roles and Responsibilities . . . . .	8
3.2.3 Tools and Techniques . . . . .	12
3.3 Project Management Plan . . . . .	13
3.3.1 Tasks . . . . .	13
3.3.1.1 Problem Definition and Scope Design . . . . .	14
3.3.1.2 System Requirements with SRS . . . . .	14
3.3.1.3 Algorithm development . . . . .	15
3.3.1.4 UI Design . . . . .	15
3.3.1.5 Database Design . . . . .	15
3.3.1.6 Frontend And Backend Development . . . . .	16
3.3.1.7 Development Version 1 . . . . .	16
3.3.1.8 Development Version 2 . . . . .	17

3.3.2	Assignment . . . . .	17
3.3.3	Timetable . . . . .	18
<b>4</b>	<b>Software Requirements Specification</b>	<b>19</b>
4.1	Product Overview . . . . .	19
4.2	Specific Requirements . . . . .	19
4.2.1	External Interface Requirements . . . . .	19
4.2.1.1	User Interfaces . . . . .	19
4.2.1.2	Hardware Interfaces . . . . .	23
4.2.1.3	Software Interfaces . . . . .	23
4.2.2	Software Product Features . . . . .	23
4.2.2.1	Authentication . . . . .	23
4.2.2.2	Upload CT Scan: . . . . .	23
4.2.2.3	Pre processing: . . . . .	23
4.2.2.4	Classify: . . . . .	23
4.2.2.5	Interpretable Results: . . . . .	23
4.2.2.6	Accuracy: . . . . .	23
4.2.3	Software System Attributes . . . . .	23
4.2.3.1	Reliability . . . . .	23
4.2.3.2	Availability . . . . .	24
4.2.3.3	Security . . . . .	24
4.2.3.4	Usability . . . . .	24
4.2.3.5	Portability . . . . .	24
4.2.3.6	Performance . . . . .	24
4.2.4	Dataset Requirements . . . . .	24
4.2.5	Database Requirements . . . . .	24
<b>5</b>	<b>Software Design Description</b>	<b>26</b>
5.1	Introduction . . . . .	26
5.1.1	Project Overview . . . . .	26
5.1.2	Purpose of Document . . . . .	26
5.2	System Architecture Design . . . . .	26
5.2.1	Chosen System Architecture . . . . .	26
5.2.2	System Interface Description . . . . .	27
5.3	Detailed Description of Components . . . . .	27
5.3.1	Authentication . . . . .	27
5.3.2	Exploring and Preprocessing Datasets . . . . .	27
5.3.3	Training Algorithm . . . . .	27
5.3.4	Testing and Deployment . . . . .	28

5.4	Description of the User Interface . . . . .	28
5.5	Detailed Description of Components . . . . .	32
5.5.0.1	ER Diagram . . . . .	32
5.6	System Architecture . . . . .	33
5.7	Data Flow Specifications . . . . .	35
5.7.0.1	DFD Level0 . . . . .	35
5.7.0.2	DFD Level1 . . . . .	36
<b>6</b>	<b>Implementation</b>	<b>37</b>
6.1	Technologies Used . . . . .	37
6.2	Languages Used . . . . .	37
6.3	Libraries Used . . . . .	37
6.4	Workflow . . . . .	38
<b>7</b>	<b>Software Test Document</b>	<b>39</b>
7.1	Introduction . . . . .	39
7.1.1	System Overview . . . . .	39
7.1.2	Test Approach . . . . .	39
7.2	Test Plan . . . . .	40
7.2.1	Test Plan Objectives . . . . .	40
7.2.2	Features to be Tested . . . . .	40
7.2.3	Features Not to be Tested . . . . .	40
7.2.4	Testing Tools and Environment . . . . .	40
7.2.4.1	Testing Tools . . . . .	40
7.2.4.2	Environment . . . . .	40
7.3	Test Cases . . . . .	41
7.3.1	Testing Login using Selenium . . . . .	41
7.3.2	Testing Register using Selenium . . . . .	41
7.3.3	Testing Input CT-Scan using Selenium . . . . .	42
<b>8</b>	<b>Conclusions and Scope for Future Work</b>	<b>1</b>
8.1	Conclusion . . . . .	1
8.2	Future scope . . . . .	1
<b>References</b>		<b>1</b>
<b>Acknowledgements</b>		<b>1</b>

# List of Figures

3.1	Gantt chart: Disease Detection . . . . .	18
4.1	Sign In Page . . . . .	20
4.2	Sign Up Page . . . . .	20
4.3	Home Page . . . . .	21
4.4	CT-Scan Submission Page . . . . .	21
4.5	Uploading the CT-Scan image . . . . .	22
4.6	Output Page . . . . .	22
5.1	Sign In Page . . . . .	28
5.2	Sign Up Page . . . . .	29
5.3	Home Page . . . . .	29
5.4	CT-Scan Submission Page . . . . .	30
5.5	Uploading the CT-Scan image . . . . .	30
5.6	Output Page . . . . .	31
6.1	Module Implementation Flow . . . . .	38

# List of Tables

3.1	Project Roles and Responsibilities . . . . .	11
3.2	Tools and Techniques . . . . .	13
3.3	Assignments . . . . .	18

# Chapter 1

## Introduction

*This chapter presents the brief description of the project - Covid-19 Disease Detection with Explainable Artificial Intelligence. It covers a number of points such as the Problem Definition, Motivation of the thesis, Scope of the thesis, Salient Contribution and the Organization of the thesis.*

### 1.1 Problem Definition

To design a system that aids the medical industry to make predictions of presence of covid-19 based on the patient's complaints to give an output with results that would be trustworthy and easy to interpret. The system would take symptoms(like: Sore throat,breathlessness, fever, cough etc...) and CT-Scan images of lungs as an input to predict the possibility of Covid-19 infection by using the concepts of Machine Learning and Deep Learning. The system should incorporate the concept of Explainable AI to make the interpretation of the results easier for the user.

### 1.2 Motivation of the thesis

There are several reasons due to which we chose the topic Covid-19 Disease Detection with Explainable Artificial Intelligence as our LY B.Tech Project. The most important reason being the tremendous increase in the number of Covid-19 cases across the world and the fatalities caused because of it. The world was hit with this pandemic, and people have suffered a lot due to various reasons. Due to the suddenness of the global problem, people were not exactly equipped with the resources required to deal with it, and the transition was not exactly smooth either. As moving out of the houses for testing was dangerous and practically impossible, we thought of developing a system which could detect the presence of Covid-19 in patients by eliminating the need of visiting hospitals.

### 1.3 Scope of the thesis

#### 1.3.1 Description

The times of covid 19 pandemic struck us hard and made us realize that there should be an efficient system for detection of diseases, at our hands, i.e., online thereby minimizing the requirement to go to the hospitals for diagnosis. Our project aims to provide efficient method to detect the presence of Covid-19 on the basis of the dataset that includes CT-Scan images of lungs. We plan to create a model using machine

learning and deep learning algorithms to achieve highest possible accuracy and precision. To make the model more trustworthy we plan to include the technology Explainable AI in the scope of our project, which would help the users comprehend and interpret the results.

### 1.3.2 Criteria

- Providing contact-free and accurate Covid-19 diagnosis at home and further eliminating the requirement of visiting hospitals.
- No specific requirement of prior educational experience or technical expertise to use the application.
- Facility to upload the Chest Ct-Scan images of patients on the web application in a specified format.
- Evaluating the scans by taking Covid-19 positive and Covid-19 negative CT-Scan images present in a dataset as reference.
- Providing the users with the results generated and a detailed explanation on the same.

## 1.4 Salient Contribution

We have used Explainable AI for the implementation of our project. Explainable AI helps in understanding and interpreting predictions made by the machine learning model. It helps in improving the model's performance and also helps others understand the behaviour of the model. With Explainable AI, the patients can get a comprehensive report of their Covid-19 results. This concept has not been implemented by anyone in the world yet. We will be making a vital contribution to the ML, AI, and medical fields.

## 1.5 Organization of the Thesis

The report consists of abstract followed by the table of contents. The figures and table list are mentioned in the report after the table of content. The further work is divided into chapters, and then finally the references and acknowledgement.

The **Chapter 1** consists of Introduction. It has the Problem Definition of the project, Motivation of the thesis, Scope of the thesis, Salient Contribution and the Structure of the thesis.

The **Chapter 2** consists of Literature Survey.

The **Chapter 3** consists of Software Project Management Plan. This includes Project Overview, Project Deliverables, Project Organization, Software Process Model, Roles & Responsibilities, Tools & Techniques.

The **Chapter 4** consists of the Software Requirements Specification which includes Product Overview, External Interface Requirements, etc.

The **Chapter 5** has Software Design Description. It includes the System Architectural Design, Description on Components, User Interface design, etc.

The **Chapter 6** describes the Implementation process which includes Technologies, Libraries, Workflow.

The **Chapter 7** is the description of the Software Test Document. The test approach, features to be tested, features not to be tested, testing tools and environments, etc. are covered here.

The **Chapter 8** covers the Conclusion and Scope for Future work.

# Chapter 2

## Literature Survey

*This chapter presents the Literature Survey. It discusses the approach and the method used by various related papers. It throws light on the different algorithms and models implemented. This section also elaborates the idea of Explainable Artificial Intelligence and its applications in various fields.*

- There exists projects which take into consideration textual data to predict the presence of a covid 19 disease in humans. In our project we plan to include Chest CT Scan Images, to predict the disease. Since it many times happen that symptoms don't show the presence of disease but the chest CT Scan shows a completely different picture.
- In the papers we referred, it is mentioned that there is a high requirement to increase the transparency of the algorithms used in the prediction. In fields like healthcare, scientists often use various methods of predictions but often they also do not know the functions at the back end and thus the algorithms are referred to as enclosed within a black box. The aim of the explainable AI is to address this black box and help increase the transparency in the system, thereby increasing the trustworthiness of the system.
- The papers we reviewed and surveyed explained that the algorithms used in AI can be differentiated into white-box and black-box machine learning algorithms. White-box models provide results that are understandable to the people. Black-box models, on the other hand, are extremely hard to explain and can hardly be understood even by experts in this domain.
- The author explains that AI Explanations in AutoML Tables and Vertex AI Predictions provide data scientists with the insight needed to improve datasets or model architecture and debug model performance.
- The papers also elaborated on the idea that XAI algorithms are considered to follow the three principles transparency, interpretability and explainability. Transparency refers to the processes that extract model parameters from training data and generate labels from testing data. Interpretability describes the possibility to comprehend the ML model and to present the underlying basis for decision-making in a way that is understandable to humans. Explainability is a concept that is recognized as important.
- Chest CT Scans are valuable tool for clinical diagnosis of Covid-19. Although CT and RT-PCR are most commonly concordant, CT may also detect early COVID-19 in patients with a negative RT-

PCR examination, in patients without symptoms, before or after symptoms develop. LIME Method is used To determine which areas of CT Scans are more predictive leading to final decision. The author compares various XAI modules like CAM, LIME and SHAP for providing convincing explanation.

- In one of the paper, the author takes internet scans of web interface as the dataset to classify the device. He uses the explainable AI method, LIME framework to explain the results of Naive Bayes classifier algorithm. Neural Network classification lacks transparency and explainable classification methods not only provides high accuracy but also lets the expert interpret the information.

# Chapter 3

# Software Project Management Plan

*This chapter presents the Software Project Management Plan. It discusses various aspects such as the Project Overview, Project Deliverables, Project Organization which includes Software Process Models, Roles & Responsibilities and Tools & techniques, Project Management Plan, Assignment and Timetable.*

## 3.1 Introduction

### 3.1.1 Project Overview

The times of covid 19 pandemic struck us hard and made us realize that there should be an efficient system for detection of diseases, at our hands, i.e., online thereby minimizing the requirement to go to the hospitals for diagnosis. Our project aims to provide efficient method to detect the presence of Covid-19 on the basis of the data set that includes CT-Scan images of lungs. We plan to create a model using machine learning and deep learning algorithms to achieve highest possible accuracy and precision. To make the model more trustworthy we plan to include the technology Explainable AI in the scope of our project, which would help the users comprehend and interpret the results.

### 3.1.2 Project Deliverables

A project management term for the quantifiable goods or services that will be provided upon the completion of a project. Deliverables can be tangible or intangible parts of the development process and are often specified functions or characteristics of the project. Deliverables serves as a general term that encompasses the requirements of a project. A deliverable may be an object used in the greater scheme of the project.

Project Duration: 12 Months

Start Date: 30/08/2021

#### 1. Project Management Plan:

A plan of actions to be carried out in the due process until project completion.

Delivery Date- 1st October, 2021

#### 2. Data Collection Module:

Research various data sets on trusted platforms and selected the most appropriate data set.

Delivery Date- 15th November, 2021

### 3. Data Pre-Processing Module:

Pre-process and clean to transform the raw data in a useful and efficient format for training the model.

Delivery Date- 10th December, 2021

### 4. Algorithm Selection Module:

Researching, trying and evaluating the various classification algorithms to select the most suitable and best performing algorithm in terms of accuracy and loss factors.

Delivery Date- 15th January, 2022

### 5. Train and Validation Model Module:

Performing train, test and validation split on data set. Training the model using the selected algorithm with input as the training data set. Validation the model and tuning the parameters based on the evaluating matrix.

Delivery Date- 1st February, 2022

### 6. Testing and Prediction Module:

Performing final test on the model and making effective predictions of the image input by the user.

Delivery Date- 15th March, 2022

### 7. Creation of User Interface Module:

Creating a user interface for the users to easily access the application without looking at the complex back-end.

Delivery Date- 20th May, 2022

## **3.2 Project Organization**

### **3.2.1 Software Process Models**

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. The collaborative

agile methodology breaks down large projects into small, manageable increments or “sprints,” typically of two weeks’ duration. It encourages experimentation and the use of small projects and quick iterations to facilitate fast-paced problem solving. It contains six phases as follows:

1. Concept
2. Inception
3. Iterations
4. Release
5. Maintenance
6. Retirement

### **3.2.2 Roles and Responsibilities**

<b>Position</b>	<b>Team member</b>	<b>Responsibilities</b>
Client Manager	Vighnesh Naik	<ol style="list-style-type: none"><li>1. The interface between the project owner and everyone else, channeling communication both ways.</li><li>2. Responsible for communication and formal agreements between stakeholders</li><li>3. Primary stakeholder within the production team and must ensure that the goals are being constantly communicated and met.</li></ol>
Project Manager	Jai Mehta	<ol style="list-style-type: none"><li>1. Overall supervision of project and team.</li><li>2. Delegates requirements.</li><li>3. Maintains project plan.</li><li>4. Performs implementation.</li><li>5. Arbitrates issues that may arise.</li></ol>

Project Leader	Pankti Nanavati, Vishant Mehta	<ol style="list-style-type: none"> <li>1. Performs the tasks of the Project Manager when the Project Manager is unavailable.</li> <li>2. Analyzes software requirements.</li> <li>3. Manages all project documentation .</li> </ol>
UI/UX Designer	Pankti Nanavati, Vishant Mehta	<ol style="list-style-type: none"> <li>1. Gathering and evaluating user requirements, in collaboration with product managers and engineers.</li> <li>2. Illustrating design ideas using storyboards, process flows and sitemaps.</li> <li>3. Designing graphic user interface elements, like menus, tabs and widgets.</li> <li>4. Develop UI mockups and prototypes that clearly illustrate how sites function and look like.</li> <li>5. Create original graphic designs (e.g. images, sketches and tables).</li> <li>6. Prepare and present rough drafts to internal teams and key stakeholders.</li> <li>7. Identify and troubleshoot UX problems (e.g. responsiveness), and conduct layout adjustments based on user feedback.</li> </ol>

Frontend Developer	Vishant Mehta, Pankti Nanavati	<ol style="list-style-type: none"> <li>1. Develop new user-facing features.</li> <li>2. Build reusable code and libraries for future use.</li> <li>3. Ensure the technical feasibility of UI/UX designs.</li> <li>4. Optimize application for maximum speed and scalability.</li> <li>5. Assure that all user input is validated before submitting to back-end</li> <li>6. Proficient understanding of cross-browser compatibility issues and ways to work around them.</li> <li>7. Good understanding of SEO principles and ensuring that application will adhere to them.</li> </ol>
Backend Developer	Pankti Nanavati, Jai Mehta, Vighnesh Naik, Vishant Mehta	<ol style="list-style-type: none"> <li>1. Be involved and participate in the overall application lifecycle.</li> <li>2. Optimization of the application for maximized metrics(accuracy, recall, precision, etc) speed and scalability and implementation of security and data protection.</li> <li>3. Main focus on coding and debugging</li> <li>4. Develop functional and sustainable web applications with clean codes.</li> </ol>

Database Manager	Vighnesh Naik, Jai Mehta	<ol style="list-style-type: none"> <li>1. Maintains database results by setting and enforcing standards and controls.</li> <li>2. Prepares for database expansion by studying plans and requirements; advising senior technical management; coordinating design and programming.</li> <li>3. Maintains database performance by troubleshooting problems.</li> <li>4. Secures database by developing policies, procedures, and controls.</li> </ol>
Software Tester	Jai Mehta	<ol style="list-style-type: none"> <li>1. analyzing users images inputs and/use cases/requirements for validity and feasibility.</li> <li>2. execute all levels of testing (System, Integration, and Regression) and design and develop automation scripts when needed.</li> <li>3. Detect and track software defects and inconsistencies and provide timely solutions.</li> <li>4. Apply quality engineering principals throughout the Agile product lifecycle and provide support and documentation.</li> </ol>

Table 3.1: Project Roles and Responsibilities

### 3.2.3 Tools and Techniques

Tool Category	Tool Used
Wireframing Tool	Marvelapp
Testing tool	Selenium
Gantt Chart	Lucidchart
UML diagram tool	Lucidchart
LaTeX editor	Overleaf

Machine Learning	Keras, Sci-Kit Learn, Tensorflow
Technologies	Languages/Libraries used
Front End	Flask, HTML, CSS, Bootstrap
Back End	Python, PHP, MySQL
Explainable AI	LIME, SHAP, Grad-CAM

Activity	Tools Required	Technique to be used
Documentation	MS Word, LaTeX	The procedure would be given in steps wherever necessary with labelled diagrams to be used.
Algorithm development	Keras, Sci-Kit Learn, Tensorflow, LIME, SHAP, Grad-CAM	After the documentation, the team will be working on the ML Algorithm, which forms the crux of the application.
UI/UX Designing	Adobe XD	The prototype will be first prepared and shared with the client. After the design is approved implementation of the same will be performed.

Frontend Development	HTML, CSS, BOOTSTRAP, JS	After the approval of the prototype by the clients, the frontend developers start building the user interface of the application.
Backend Development	PHP, MySQL	The backend developers work parallelly with the frontend developers to make the static website functional.
Project Storage	Github	The developers working on the application will be added to the github organization and will be given access to the repository which will have the project code.
Meetings	Microsoft Teams, Zoom	All the meetings regarding the project with various groups and sub-groups will be carried out on these platforms.
Testing	Mobiles, Desktops, screens of different widths	To check the responsiveness of the application and to make it browser friendly, it will be tested on these screens.

Table 3.2: Tools and Techniques

### 3.3 Project Management Plan

#### 3.3.1 Tasks

The tasks that are involved in this project are:

1. Problem Definition and Scope Designing
2. Getting Feedback from user, requirement analysis, develop an exploratory prototype
3. Developing the SRS document
4. Developing the Algorithm
5. System UI design

6. Database and Schema Designing
7. System Design- Frontend and Backend Development
8. Developing the Version 1
9. Developing the Version 2
10. Software Testing

#### **3.3.1.1 Problem Definition and Scope Design**

Description Formulation of problem statement, done along the guidelines of client requirements to be done along with the scope that the project can achieve. Keeping in mind about how useful the product will be to its client and to develop a user-friendly product with minimum use of Tech Jargon. Major and minor functionalities also to be specified. Resources are required to carry out the project tasks. They can be people, equipment, facilities, funding, or anything else need for completion of a project activity.

Deliverables and Milestones Deliverable:

Problem statement and scope document.

Milestone:

Delivering the document before the scheduled deadline with including everything as per the requirement.

Resources needed Data set research and collection.

Dependencies and constraints Client requirements and needs must be known for the team to devise the problem Statement. Modify the Problem Statements as per the need and requirement of the Product in the open Market.

Contingencies Client might add/update needs and requirements which might render the current version of the problem statement obsolete.

#### **3.3.1.2 System Requirements with SRS**

Description The SRS clearly and precisely describes each of the essential requirements (functions, performances, design constraints, and attributes) of the software and the external interfaces. Each requirement is defined such that its achievement is capable of being objectively verified and validated by a prescribed method, for example, inspection, analysis, demonstration, or test. Preparation of the Software Requirements Specification document that will outline all the requirements of the project. It must encompass functional requirements, non-functional requirements, software requirements, hardware requirements and database requirements to be assumed as a complete SRS document. The cost estimation and the schedule of the product is planned on the basis of the SRS document. The SRS document can be also used as a reference for end user testing A typical SRS includes: A purpose, an overall description, specific requirements, system features and non-functional requirements. This SRS will give a brief information about the Prototype to be developed (It will contain some assumptions).

Deliverables and Milestones Deliverable: Complete SRS document. Milestone: Delivering the document before the scheduled deadline.

Resources needed Problem statement and scope document, Technical limitations, Budget allotment for the entire prototype.

**Dependencies and constraints** Client requirements and needs must be known for the team to prepare SRS. Also as per the technical aspect, some assumptions will be made in the SRS document.

**Contingencies** Failure to provide the allocated budget/resources once the project has been finalized and SRS hasn't been prepared can cause the project to be delayed/terminated.

#### **3.3.1.3 Algorithm development**

**Description** This part forms the main crux of the application. The algorithm's main function will be to classify the image based on the CT Scan as Covid Positive or negative. The output must be displayed in an efficient and explanatory manner. **Deliverable and Milestones** Deliverable: Completing the algorithm and displaying the explainable output

Milestone: Completing the algorithm with high accuracy Resources needed Keras, Sci-Kit Learn, Tensorflow, LIME,SHAP,Grad-CAM **Dependencies and constraints** Good literature survey must be conducted and the algorithm must be viable for all versions of CT Scan **Contingencies** Poor performance of the algorithm may lead to huge consequences.

#### **3.3.1.4 UI Design**

**Description** The user interface (UI) is the point of human-computer interaction and communication in a device. This can include display screens, keyboards, a mouse, and the appearance of a desktop. It is also the way through which a user interacts with an application or a website.

**Deliverables and Milestones** Deliverable:

The final version of User Interface.

Milestone:

Delivery of a full-fledged user interface for users before the deadline.

Resources needed Clients expected delivery date, Access to Adobe XD or Figma.

**Dependencies and constraints** UI design has to start with user research and task analysis.

**Contingencies** Poor/Complex design of user interface may lead to difficulties in accessing the product.

#### **3.3.1.5 Database Design**

**Description** Database Design is defined as a collection of steps that help with designing, creating, implementing, and maintaining a business's data management systems. The main purpose of designing a database is to produce physical and logical models of designs for the proposed database system.

**Deliverables and Milestones** Deliverables:

The design of the database schema.

Milestones:

Delivery of a optimized database with appropriate columns specific to the project.

Resources Needed

MySQL, Draw.io for designing the schema.

Dependencies and constraints

Database design has to start by enlisting the entities and attributes and further deciding the tables in the database that will be required according to the modules provided by the clients.

#### Contingencies

Poor design of the database will result in an inefficient system and will reduce the performance measure of the application.

#### **3.3.1.6 Frontend And Backend Development**

Description Systems design which includes the frontend and backend development is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. The SDD describes the major components of the software design including databases and internal interfaces. The major components in this system are:

1. Login and Register module for users.
2. Proper verification of the Details provided by the customer.
3. Entry of all the records in the database

#### Deliverables and Milestones Deliverable:

The final version of the system design.

#### Milestone:

Delivery of a high-level design of the system with all the necessary details and structure of servers, load balancers, databases, etc. before the deadline.

Resources needed A technical team with experience of performance, modifiability, availability, scalability, reliability, etc. Allocation of task to a specific member of the team like frontend, backend, design, database, testing, etc.

Dependencies and constraints System design will highly depend on the client requirements, target audience, and client budget.

Contingencies Poor system design may give rise to serious problems like compromised security, increased latency, increased cost, or may even bring the entire system to a complete halt. If client requirement changes then system design might have to be changed drastically.

#### **3.3.1.7 Development Version 1**

Description Implementing the functionalities required in the application through coding. As this is the most important part it will be given utmost importance. There are members in the team who will be entirely responsible for this part. Also, through backup we can ensure that no code is lost and the system is always in recoverable state.

#### Deliverables and Milestones Deliverable:

First stage of Prototype Development.

#### Milestone:

Delivery of 75-80% prototype implementation before the deadline.

Resources needed A proper SRS and SDD along with the SPMP with explanation in brief.

Dependencies and constraints This part will totally be dependent on the technical team, how and when each individual does the task as from here, one part will be connected with another. Backups will be an important constraint, as one person's work should be saved in a backup copy and then another person should proceed further with his/her job and also the PC specifications for implementation.

Contingencies Proper understanding of the Problem and requirements and then start the implementation, as if there's some misunderstanding by any of the Technical Team member, then the prototype needs to be developed from scratch.

### **3.3.1.8 Development Version 2**

Description After showing the first development version to the client and receiving the feedback, he might suggest some changes or modifications. So taking his feedback into consideration, the Technical Team will have to modify the prototype and complete it.

Deliverables and Milestones Deliverable:

Final Prototype.

Milestone:

Delivery of entire perfect prototype implementation before the deadline.

Resources needed A proper SRS and SDD along with the SPMP with explanation in brief along with the modification/feedback from the end user(client).

Dependencies and constraints It will be dependent about how big are the modifications suggested by the client. Backups will be an important constraint, as one person's work should be saved in a backup copy and then another person should proceed further with his/her job and also the PC specifications for implementation.

Contingencies Proper understanding of the changes/modifications by the client and then start with the changes as if there's some misunderstanding by any of the Technical Team member, then the prototype needs to be developed from scratch.

### **3.3.2 Assignment**

Sr.No.	Tasks	Team Members
1.	Requirements Gathering	Pankti Nanavati, Vishant Mehta
2.	Planning and Project Flow discussion	Jai Mehta, Vighnesh Naik

3.	Authentication	Pankti Nanavati, Vighnesh Naik
4.	Work on Algorithm	Pankti Nanavati, Vishant Mehta, Vighnesh Naik, Jai Mehta
5.	Database Integration	Vishant Mehta, Jai Mehta
6.	Testing	Vishant Mehta, Vighnesh Naik
7.	Completion and Maintenance	Jai Mehta, Pankti Nanavati

Table 3.3: Assignments

### 3.3.3 Timetable



Figure 3.1: Gantt chart: Disease Detection

# Chapter 4

# Software Requirements Specification

*This chapter presents the Software Requirements Specification. It gives a brief description of Product Overview, External Interface Requirements, Software Product Features, Software System Attributes, Dataset Requirements and Database Requirements.*

## 4.1 Product Overview

An online disease detection system to provide accurate diagnosis at home and to eliminate the need of visiting hospitals for health checkups and assessments. The web application provides the users a contact-free method to detect the presence of Covid-19 on the basis of CT-Scan images of lungs. The users can monitor their health and can find out if they have contracted the disease without exposing themselves to their surroundings. The user-friendly application can be used by anyone and does not require any educational level, experience or technical expertise.

## 4.2 Specific Requirements

### 4.2.1 External Interface Requirements

#### 4.2.1.1 User Interfaces

User lands on the login page first. The user gets an option to sign up. Clicking on sign up leads the user to the sign up page. If login is successful, the user lands up on the home page. User can see the catalogue on the home page and is provided a menu at the top. All functionalities can be accessed through the menu. The user can upload the X-Ray images of the patient through the X-Ray submission page.

On Submitting the images, the user will be directed to the output page where the results will be displayed. The user will be provided with analysed output of the X-Ray images followed by a statement confirming or rejecting the presence of Covid-19.

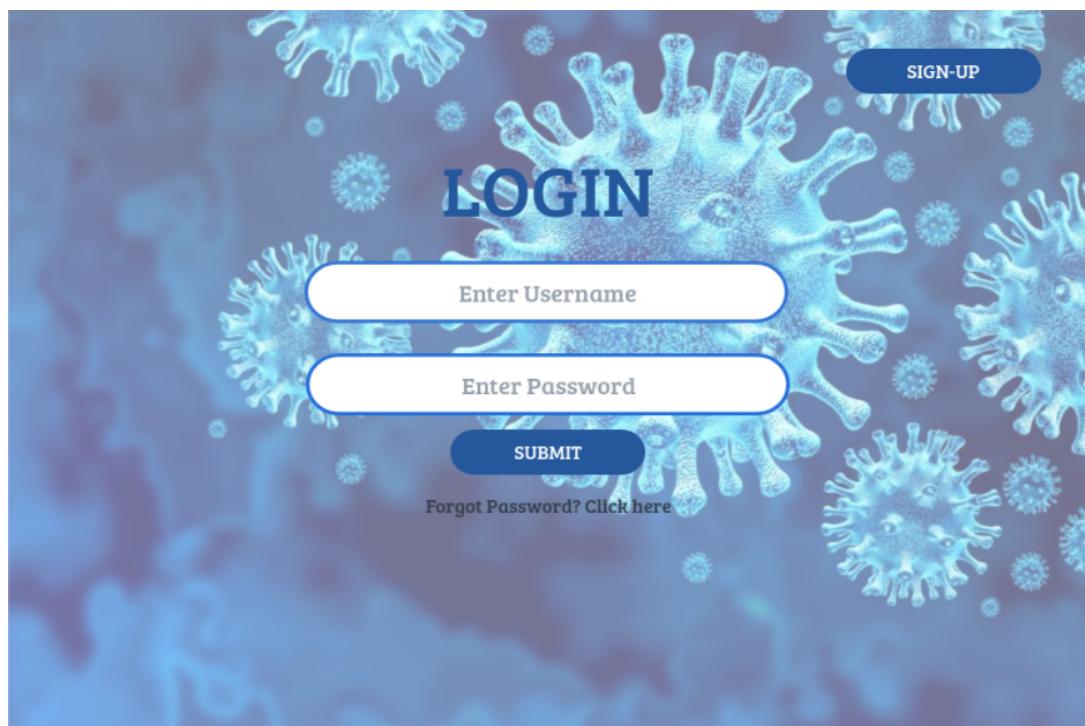


Figure 4.1: Sign In Page

The image shows a sign-up page with a background of blue COVID-19 virus particles. The title "Sign-Up" is centered at the top in large, bold, blue capital letters. Below the title are seven input fields, each preceded by a label in blue text: "First Name:", "Last Name:", "Email-id:", "Contact No:", "Aadhar Card No:", "Password:", and "Confirm Password:". To the right of each label is a white input field with a blue border. At the bottom center is a blue rounded rectangle button labeled "Register".

Figure 4.2: Sign Up Page

If you are ready to run Covid-19 prediction test on your reports, please click on the button below!

[RUN TESTS !](#)

We have made our tests more accurate and trustworthy by adding the new feature, Explainable AI.

Results Made Easy To Understand!

Figure 4.3: Home Page



We hope you are ready with CT Scan image of the patient  
Please upload clear image for accurate result

No file chosen

(jpeg,png)

Figure 4.4: CT-Scan Submission Page

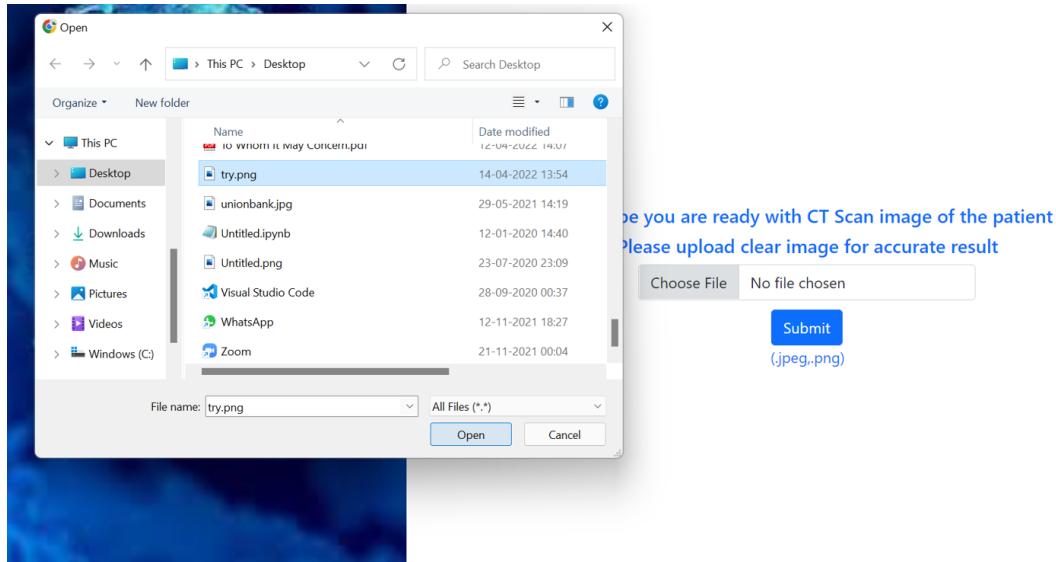


Figure 4.5: Uploading the CT-Scan image

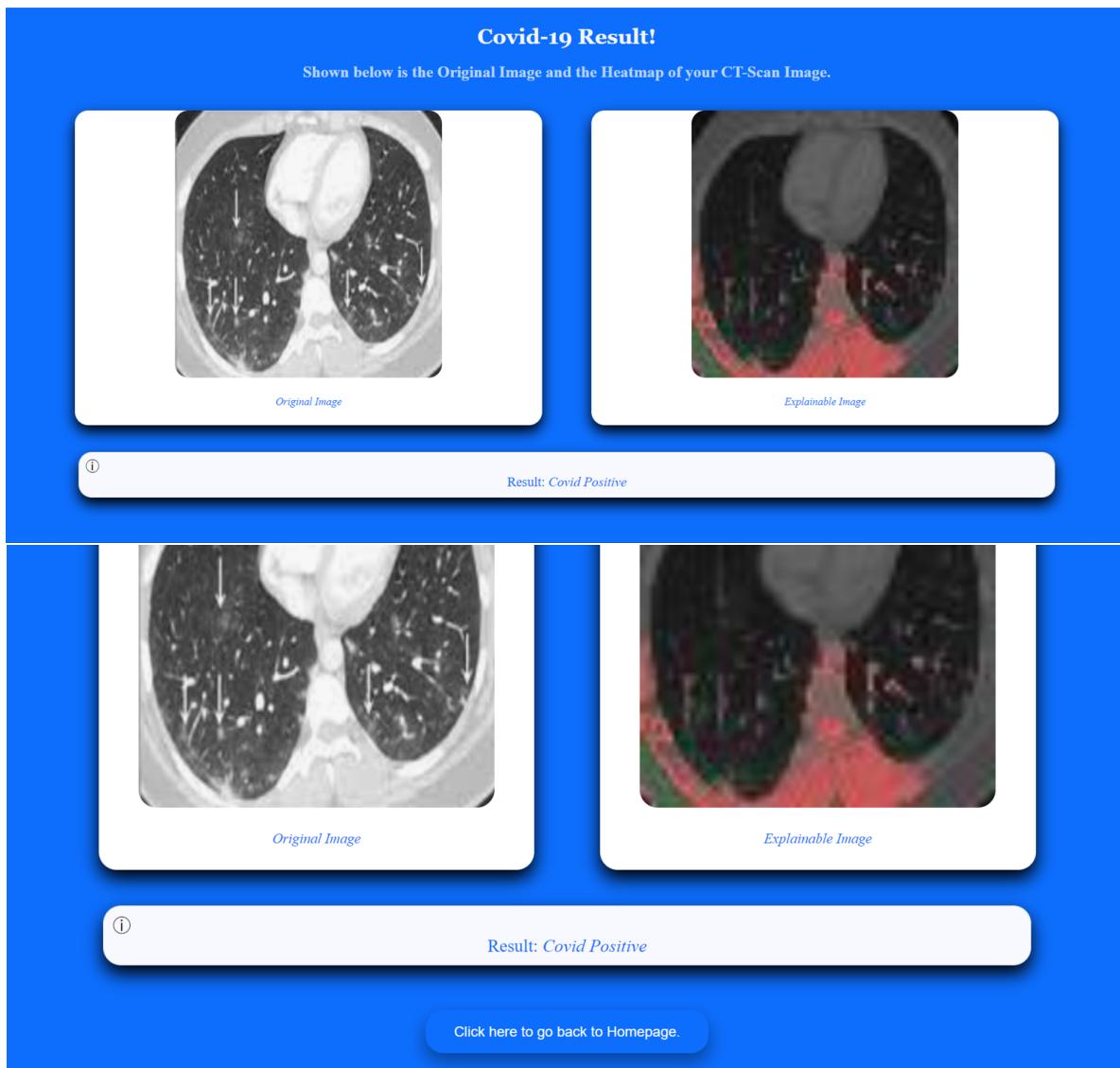


Figure 4.6: Output Page

#### **4.2.1.2 Hardware Interfaces**

A device with at least 8GB of RAM and a CPU with 7th generation configuration is needed.

#### **4.2.1.3 Software Interfaces**

As we are using python programming language for the implementation of the algorithm, a python developing environment should be available. Latest python version and various python library such as LIME, Tensor flow etc. must be also installed.

### **4.2.2 Software Product Features**

#### **4.2.2.1 Authentication**

User will first have to Register to use the Application. Once the user is registered, then he can be authenticated and use the application.

#### **4.2.2.2 Upload CT Scan:**

The user will have to upload and submit the CT Scan images/Chest X ray of the lungs. The extension of the images should be in specified format and an image as clear as possible.

#### **4.2.2.3 Pre processing:**

The program should be able to take the data input by the user and pre-process it to extract the important features, for the algorithm to work using the data efficiently.

#### **4.2.2.4 Classify:**

The algorithm should be able to classify the information fed in to classify whether the person is infected with covid 19 or not.

#### **4.2.2.5 Interpretable Results:**

The results must be displayed in a simple and explainable approach. It provides valuable information on how, the model reached that result

#### **4.2.2.6 Accuracy:**

The model must have low values of false negatives. As we do not want some infected person to be given a negative report.

### **4.2.3 Software System Attributes**

#### **4.2.3.1 Reliability**

For making the system more reliable, we aim to keep the prediction accuracy as high as possible. One more feature we have decided to add is the Explainable AI part, which would increase the trustworthiness of the system, hence making it more reliable.

#### **4.2.3.2 Availability**

The system should be available to the users for a maximum amount of time. So that the users may feel free to visit the application anytime they want. Also, the system should require a less amount of maintenance work which will ensure more availability of the application to its users.

#### **4.2.3.3 Security**

The date of the user on whose health the prediction of presence of Covid-19 virus is being made will not be disclosed to any unauthorized person.

#### **4.2.3.4 Usability**

The application must be easy to use such that they do not need to read an extensive number of manuals to use this website. The system must be quickly accessible by the users. The system must be intuitive and simple in the way it displays all relevant data and relationships. The features of the system must be easily navigable by the users with buttons that are easy to understand

#### **4.2.3.5 Portability**

The web application is portable to any device-mobile or desktop. The only things the device requires are-stable internet connection,browser and sufficient memory requirement.

#### **4.2.3.6 Performance**

Mainly the system must predict with an appropriate accuracy and reduce false positives and false negatives. The system must not lag, because the users using it don't have down-time to wait for it to complete an action. The system must complete updating the databases, adding the details successfully every time the user requests such a process. All the functions of the system must be available to the user every time the system is turned on. If for any reason some operation cannot be performed or has failed to execute, then the user must be alerted with appropriate messages.

### **4.2.4 Dataset Requirements**

The data set should be selected from a trustworthy source since it will influence really important healthcare decisions. The data set should consist of at least 1000 images of each class. In our case there are two classes, positively affected and negatively affected people. So, there our data set will consists of minimum of 2000 images. If a data set does not consists of equivalent amount of images of each class, there will be an imbalance and there will be higher influence of the larger class on the final output i.e the output will be biased leading to inaccurate predictions. Hence, it is really important for the data set to be balanced.

### **4.2.5 Database Requirements**

All data is stored in the MySQL database. Users can access their own information only. Users cannot know any other users' details. Email ID is unique to a user and needs to be given during registration. Only admin accounts are privileged to access the data of all users. Users can neither see nor modify data

of other users. Admins can access data of users. One user can have only one account under his email address.

# Chapter 5

# Software Design Description

*This chapter presents the Software Design Description. It discusses the Project Overview, Purpose of Document, System Architecture Design, Chosen System Architecture, Detailed Description of Components, System Architecture, Description of User Interface and Data Flow Specifications.*

## 5.1 Introduction

### 5.1.1 Project Overview

The times of covid 19 pandemic struck us hard and made us realize that there should be an efficient system for detection of diseases, at our hands, i.e., online thereby minimizing the requirement to go to the hospitals for diagnosis. Our project aims to provide efficient method to detect the presence of Covid-19 on the basis of the data set that includes CT-Scan images of lungs. We plan to create a model using machine learning and deep learning algorithms to achieve highest possible accuracy and precision. To make the model more trustworthy we plan to include the technology Explainable AI in the scope of our project, which would help the users comprehend and interpret the results.

### 5.1.2 Purpose of Document

Software design documents are an important way of looping everyone into the process who is involved in the product. Design documents are created to coordinate efforts of a large team, give them a stable reference point, and describe all parts of the software and how they will operate. Software design documents not only help others understand your system and provide documentation for future projects, but it also forces you to think through the entire system architecture. This ensures you go through every possible roadblock or challenge you might face, thereby exposing all gaps in your thinking.

## 5.2 System Architecture Design

### 5.2.1 Chosen System Architecture

The system is defined using a component diagram and a data flow diagram. The data flow diagram is attached at the end of the document. The different components of this system are as follows:

1. Authentication - On user side it is ensured that only registered users can login and get tested.
2. Data Exploration and Pre-processing - This allows the engineers to select the data and alter it according

to the requirements to extract the highest amount of information possible.

3. Training Algorithm - This step allows the engineers to select the most efficient algorithm to classify the input as healthy or infected.

4. Testing and Deployment - This step allows the engineers to validate on the designed algorithm and perform the tests to tune the hyperparameters to achieve the metrics required for output. Finally the testing is to be done on the application.

### **5.2.2 System Interface Description**

The Product will be made on Google Collab, Flask, HTML, CSS, PHP and SQL as database. The system will be able to run on Windows and Linux OS Platforms. The system will run on a web server using different graphics and an interface which makes it easy for the user to perform different actions.

## **5.3 Detailed Description of Components**

### **5.3.1 Authentication**

Responsibility - Vishant K. Mehta

Constraints - User must sign up or sign into the system to be able to detect the presence of Covid-19.

Composition - The user must provide details like user name or Email ID and sign in with their password.

Interactions - Client server interaction using the web browser of user. Resources - Database and Validation.

### **5.3.2 Exploring and Preprocessing Datasets**

Responsibility - Jai Mehta

Constraints - Dataset should be neither too small nor be too large. Composition - Dataset should be balanced i.e., it should have equivalent number of images of healthy and infected people.

Interactions - Client server interaction using the web browser of user.

Resources - Kaggle and Database.

### **5.3.3 Training Algorithm**

Responsibility - Pankti Nanavati

Constraints - Select the algorithm that is best suited for the dataset chosen as well as the one that gives the targeted metric combination.

Composition - The engineer performs trials on various algorithms and selects the most suitable.

Interactions - Client server interaction using the web browser of user.

Resources - Google Collab, Database and Stack Overflow .

#### **5.3.4 Testing and Deployment**

Responsibility - Vighnesh Naik

Constraints - The algorithm should be validated with all the test cases to perform the hyperparameter tuning. And the final testing should be done using the test set.

Interactions - Client server interaction using the web browser of user.

Resources - Test Set and Google Collab.

#### **5.4 Description of the User Interface**

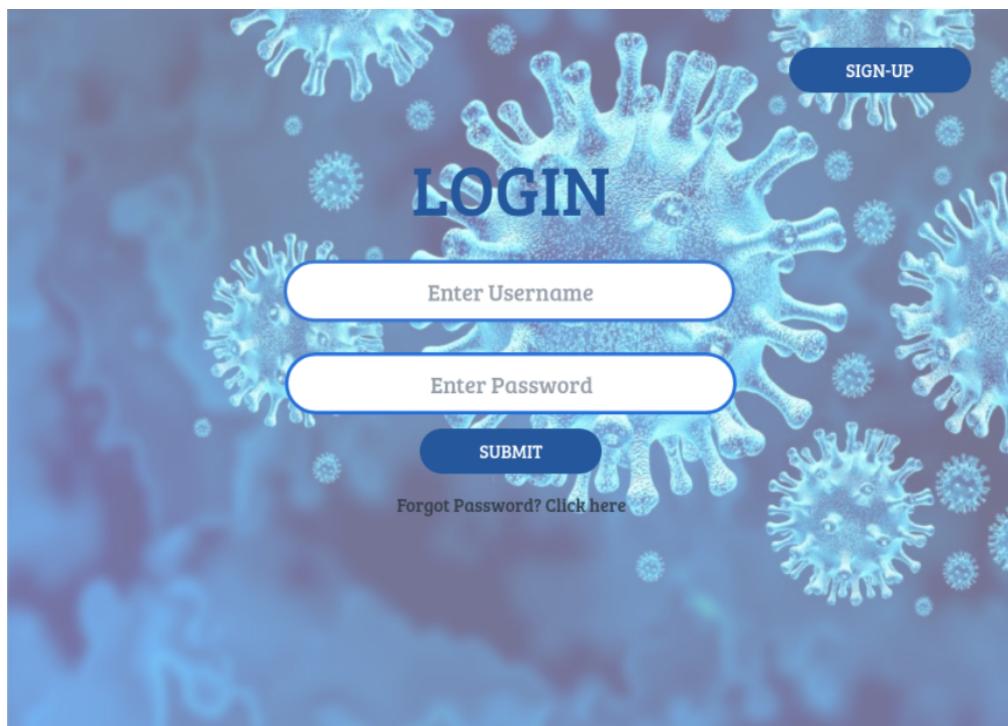


Figure 5.1: Sign In Page

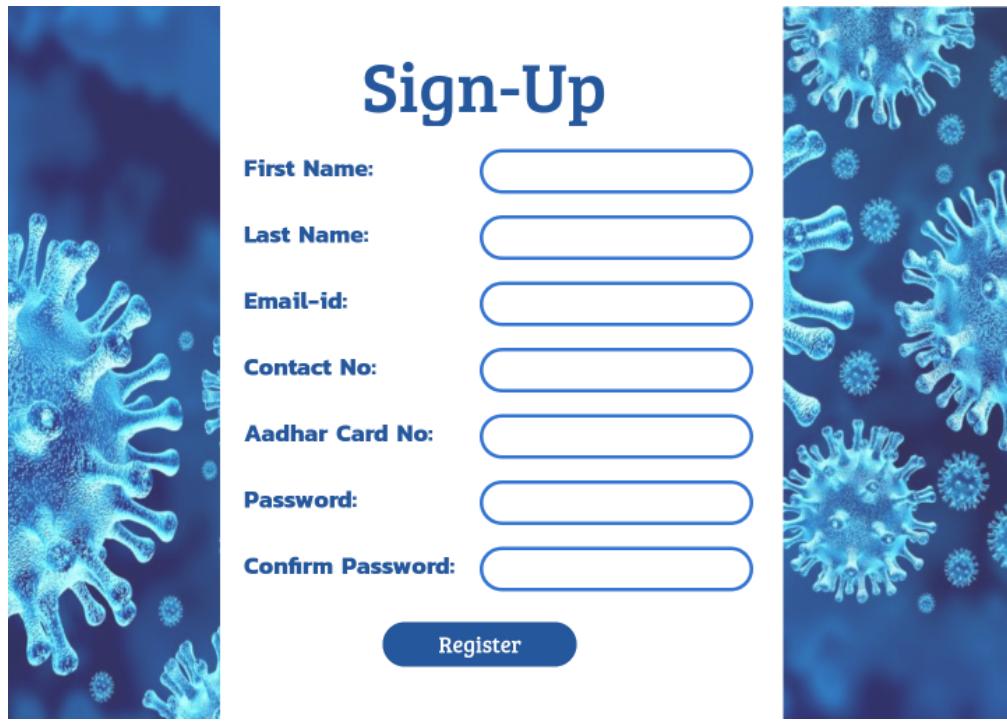


Figure 5.2: Sign Up Page

The image shows the home page of the application. At the top, there is a blue header bar with the text "Covid-19 Disease Detection" on the left and a three-line menu icon on the right. The main content area has a white background. It contains a message: "If you are ready to run Covid-19 prediction test on your reports, please click on the button below!". Below this message is a blue rounded rectangular button with the text "RUN TESTS !". Underneath the button, there is a small explanatory text: "We have made our tests more accurate and trustworthy by adding the new feature, Explainable AI." and "Results Made Easy To Understand!".

Figure 5.3: Home Page

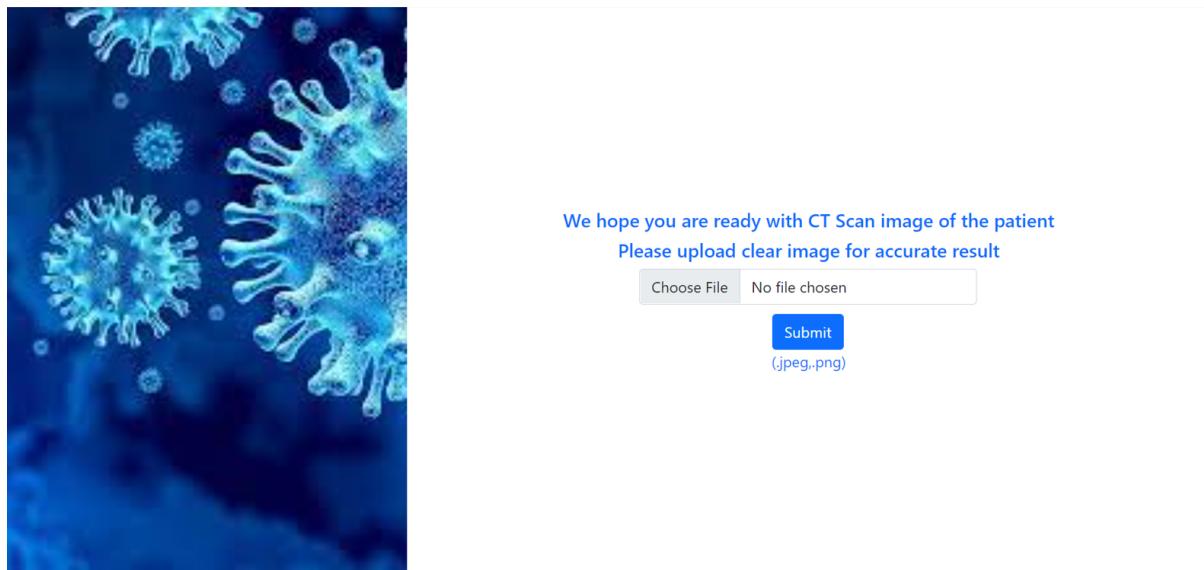


Figure 5.4: CT-Scan Submission Page

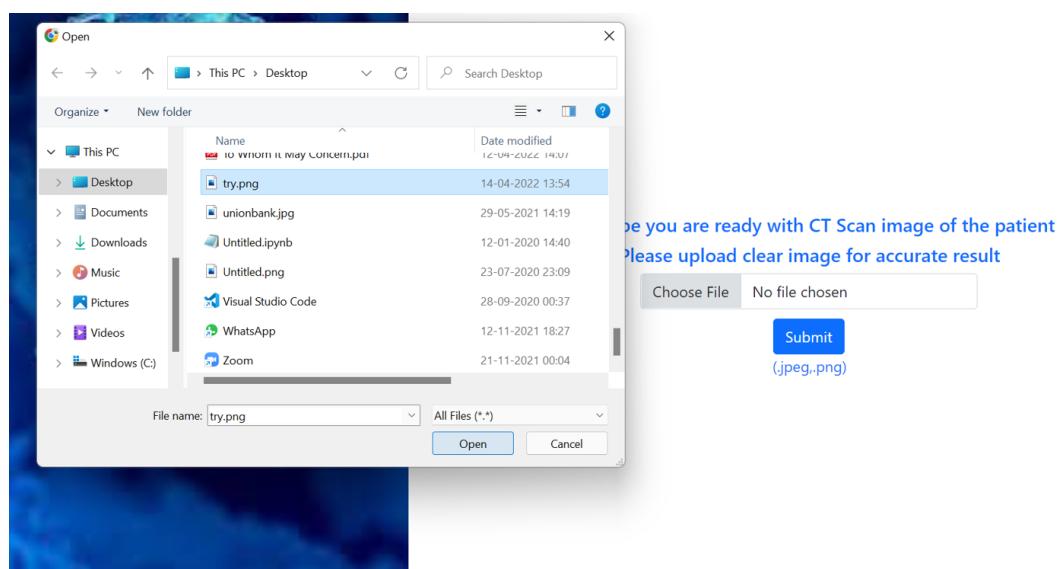


Figure 5.5: Uploading the CT-Scan image

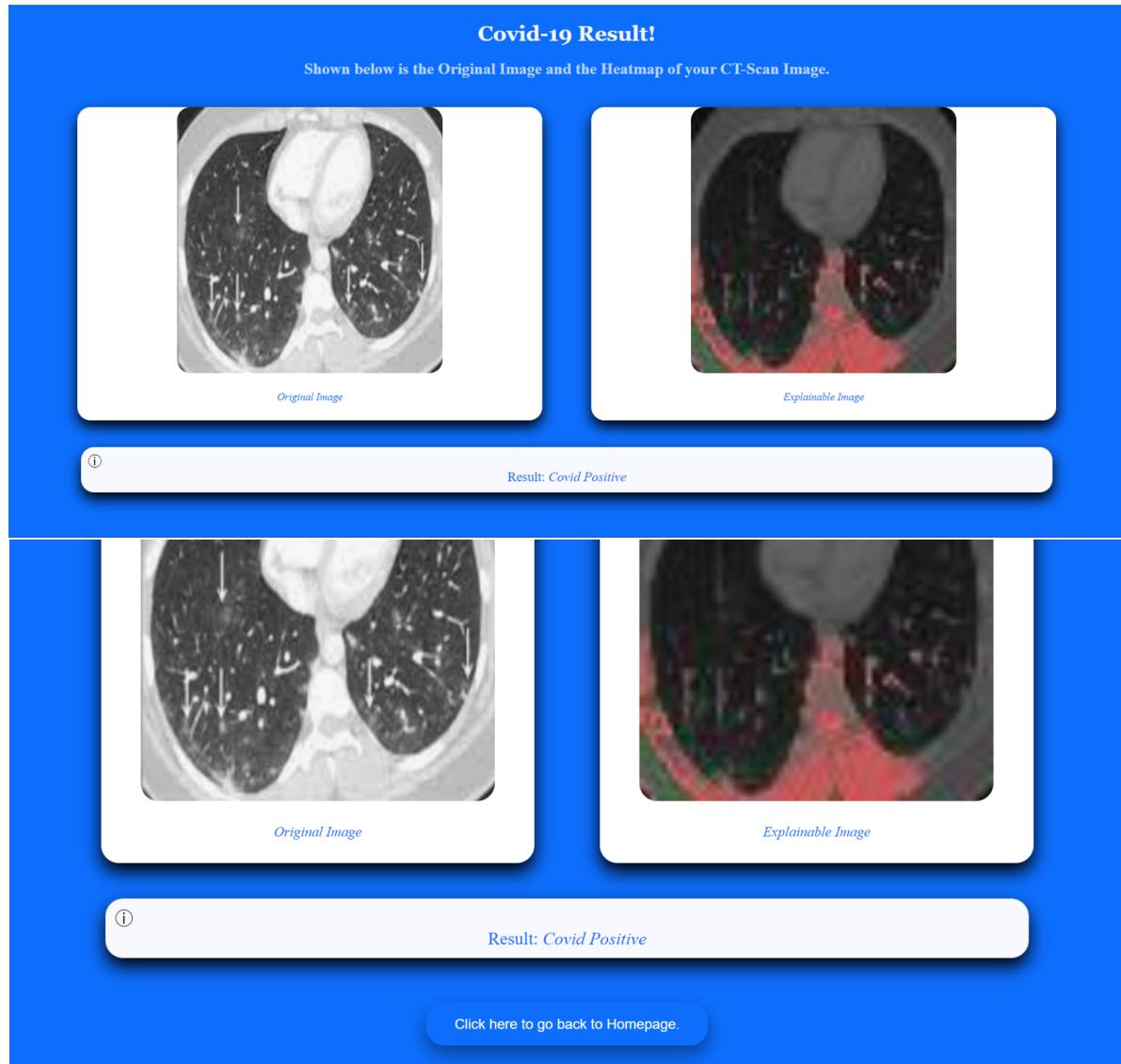
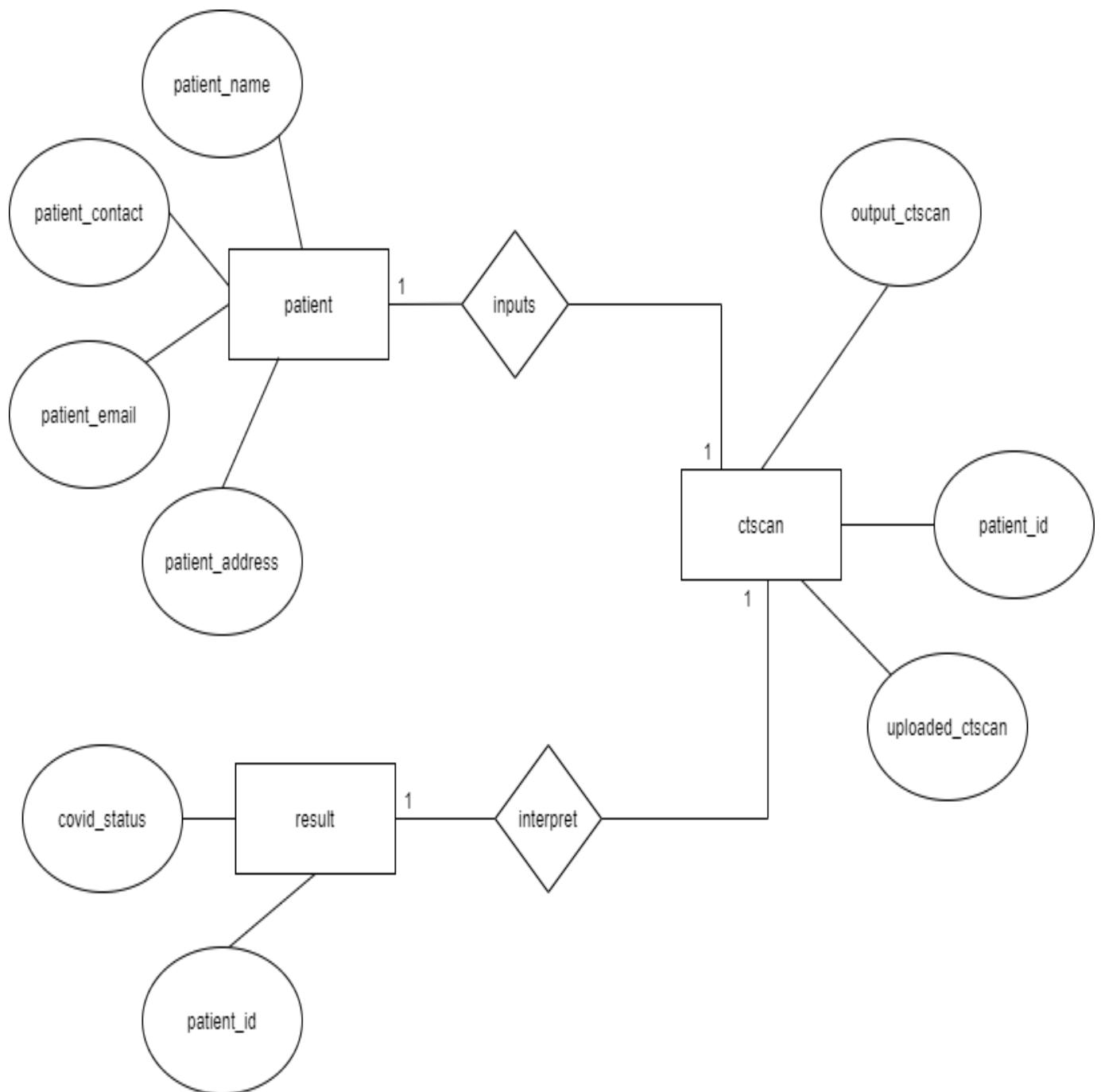


Figure 5.6: Output Page

## 5.5 Detailed Description of Components

### 5.5.0.1 ER Diagram



## 5.6 System Architecture

<b>User Case ID:</b>	1		
<b>Use Case Name:</b>	<b>Log In or Sign Up</b>		
Created By:	Vishant Mehta	Last Updated By:	-
Date Created:	14 <sup>th</sup> October, 2021	Date Last Updated:	-

Primary Actors:	User
Secondary Actors:	Admin
Descriptions:	This use case describes how the user Signs Up or Logs into the system. The actors starting this use case are admin who are the secondary actors.
Triggers:	When the user clicks on the Sign In or Sign-Up button.
Preconditions:	-
Postconditions:	-
Normal Flow:	<ol style="list-style-type: none"> <li>1. User clicks the Sign In or Sign-Up bottom on homepage</li> <li>2. A Login or Sign Up page will appear according to the choice</li> <li>3. The user has to fill in the details of the corresponding form</li> <li>4. User clicks on the Sign In or Sign-Up button</li> </ol>
Alternative Flow:	<ol style="list-style-type: none"> <li>1. Sign In/Sign UP via Google</li> <li>2. Sign In/Sign UP via Facebook</li> </ol>
Exceptions:	-
Includes:	Upload symptoms (X Ray Images of lungs), Detection of Covid-19
Priority:	High
Frequency of Use:	High

<b>User Case ID:</b>	2		
<b>Use Case Name:</b>	<b>Exploring and Pre-processing Datasets</b>		
Created By:	Jai Mehta	Last Updated By:	Jai Mehta
Date Created:	14 <sup>th</sup> October, 2021	Date Last Updated:	15

Primary Actors:	User
Secondary Actors:	Admin
Descriptions:	This use case defines the flow from dataset research to splitting the dataset into three parts. Most important component of this use case is the pre-processing that is to be done on the dataset.
Triggers:	-
Preconditions:	Dataset is in a rudimentary stage which can be difficult to extract information from.
Postconditions:	The data is pre-processed such that the highest amount of information is extracted.
Normal Flow:	<ol style="list-style-type: none"> <li>1. We carry research for finding dataset most suitable for our project.</li> <li>2. After finding the dataset, we clean and transform the dataset to reduce noise and the remove inconsistent data.</li> <li>3. Next, split the dataset into training, testing and validation sets.</li> </ol>
Alternative Flow:	-
Exceptions:	-
Includes:	-
Priority:	High
Frequency of Use:	High

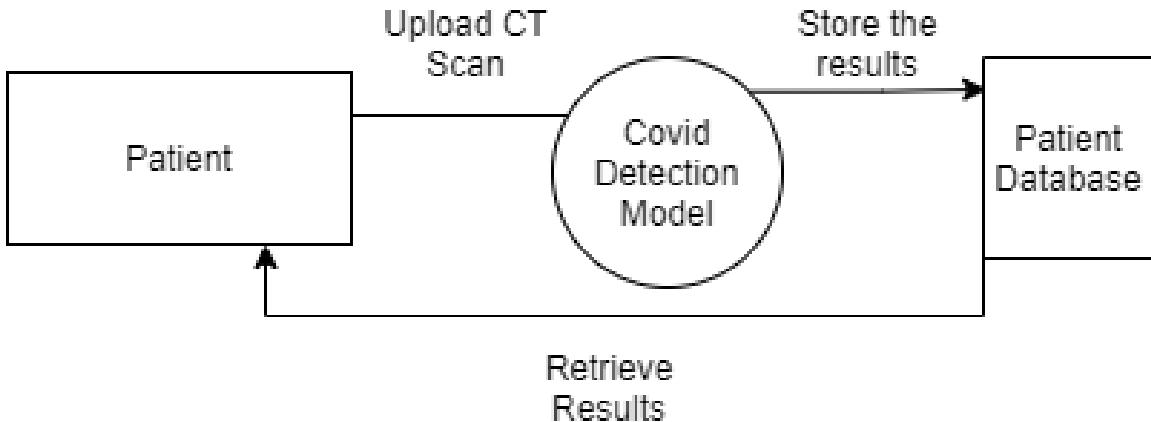
<b>User Case ID:</b>	<b>3</b>		
<b>Use Case Name:</b>	<b>Input and Results</b>		
Created By:	Pankti Nanavati, Vighnesh Naik	Last Updated By:	-
Date Created:	14 <sup>th</sup> October, 2021	Date Last Updated:	-

Primary Actors:	User
Secondary Actors:	Admin
Descriptions:	This use case defines how the user will input the X-Ray image of the lungs and how the system will process the input, make prediction based on it and show the results to the user.
Triggers:	When the user clicks on the Upload button.
Preconditions:	The uploaded image should be in the correct format.
Postconditions:	The user will land on the results page.
Normal Flow:	<ol style="list-style-type: none"> <li>1. The user will upload the X-Ray image of lungs.</li> <li>2. The system will proceed to make the predictions on the input if the inputted image is in the correct format otherwise the user will be prompted to enter the file in the correct format.</li> <li>3. Next, the system will make the prediction based on the fed data and show the results on the next page.</li> </ol>
Alternative Flow:	-
Exceptions:	Incorrect format of file.
Includes:	-
Priority:	High
Frequency of Use:	High

## 5.7 Data Flow Specifications

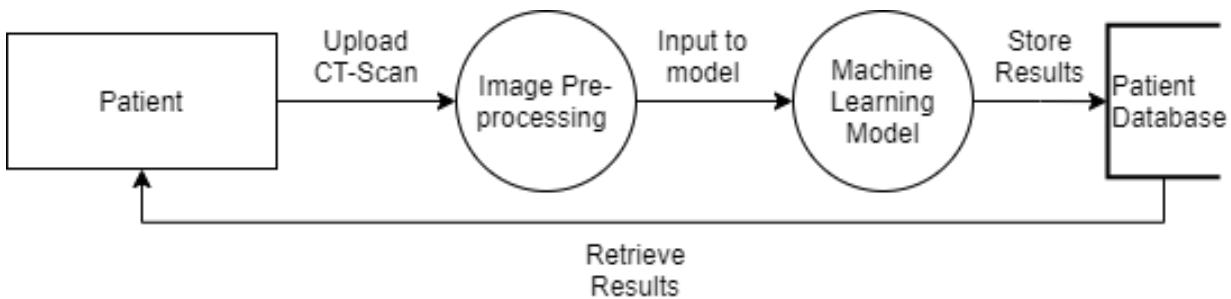
### 5.7.0.1 DFD Level0

A context diagram is a top level (also known as "Level 0") data flow diagram. It only contains one process node ("Process 0") that generalizes the function of the entire system in relationship to external entities. Draw data flow diagrams can be made in several nested layers. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities.



#### 5.7.0.2 DFD Level1

In the level 1 DFD, the processes are further broken down and the flow can be understood in further detail. The database can also be seen in the level 1 DFD and we have a better idea about the processes regarding the Patient and Model entities.



# Chapter 6

## Implementation

*This chapter presents the Implementation of the project. It gives an account of Technologies Used, Languages Used, Libraries Used and also includes the workflow of the model.*

### 6.1 Technologies Used

- Wire framing tool- Marvelapp, a collaborative design platform is used.
- UML diagram tool: Lucidchart is used to make flowcharts and diagrams easily in an online environment.
- Documentation tool: Overleaf LaTeX editor is used for documentation. Templates available are useful.
- Gantt chart tool: Trello and TeamGantt are used to schedule and keep track of activities in the project.
- Code editor: VSCode is used as a code editor.
- Testing tool: Selenium is used for white box testing.
- Database: MySQL database is used.

### 6.2 Languages Used

- Python: Used for implementation of exploratory data analysis and coding the machine learning algorithm.
- Django: Frontend and backend development.
- CSS: Used for styling documents. This enhances the user experience.
- Bootstrap: Adds interactive behaviour to the web application.

### 6.3 Libraries Used

- Keras
- Tensor Flow
- Sci-kit learn
- Pandas

- Numpy
- LIME (local interpretable model-agnostic explanations)

## 6.4 Workflow

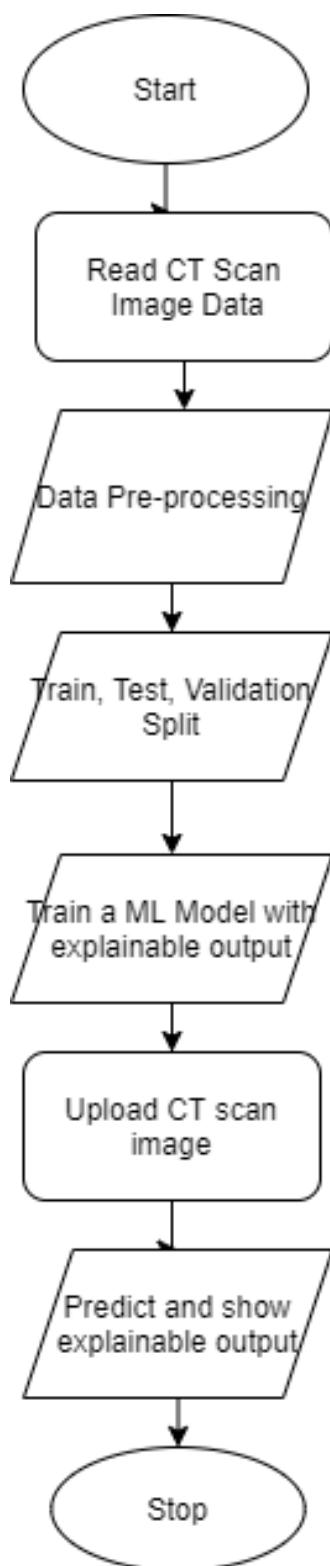


Figure 6.1: Module Implementation Flow

# Chapter 7

## Software Test Document

*This chapter gives a brief idea about the Software Test Document. It discusses the System Overview, Test Approach, Test Plan, Features to be tested, Features not to be tested, Test plan objectives, Testing tools and Environments and also shows various Test Cases.*

### 7.1 Introduction

#### 7.1.1 System Overview

The times of covid 19 pandemic struck us hard and made us realize that there should be an efficient system for detection of diseases, at our hands, i.e., online thereby minimizing the requirement to go to the hospitals for diagnosis. Our project aims to provide efficient method to detect the presence of Covid-19 on the basis of the data set that includes CT-Scan images of lungs. We plan to create a model using machine learning and deep learning algorithms to achieve highest possible accuracy and precision. To make the model more trustworthy we plan to include the technology Explainable AI in the scope of our project, which would help the users comprehend and interpret the results.

This document is proposed to define the test plan and strategy to test Covid-19 Prediction Model. It describes the overall approach to testing for each major group of features or feature combinations, the major activities, techniques, and tools which will be used to test the designated groups of features.

#### 7.1.2 Test Approach

A test approach is the test strategy implementation of a project, defines how testing would be carried out. This section defines the overall test approach to be taken for the project. We will focus on the below two types of testing :

- PRO-ACTIVE TESTING: we would be testing the system as we are developing it so any bugs or inconsistencies would be found out at an earlier stage hence reducing the risk of building a system that does not work at all. It is also easier to find the flaws of the system as the system grows increasingly complex rather than testing a complex system all along.
- BLACK BOX TESTING (also known as Behavioral Testing): is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

- **MANUAL TESTING:** It will be done to ensure the correctness of various parts of the code using test cases generated by the tester. Manual Testing is a type of software testing in which test cases are executed manually by a tester without using any automated tools. The purpose of Manual Testing is to identify the bugs, issues, and defects in the software application.

## 7.2 Test Plan

### 7.2.1 Test Plan Objectives

- Testing is the process of executing the program the program with the intention of finding an error.
- A good test is one that has high probability of finding an as yet undiscovered error/bug.
- A successful test is that which uncovers as-yet-undiscovered error.

### 7.2.2 Features to be Tested

1. Login Page:

- Testing on valid Credentials
- testing on Invalid Credentials

2. Sign Up Page:

- Testing On valid Credentials into database
- testing on invalid Credentials

3. Input CT-Scan :

- Valid CT-Scan Image format
- Invalid image format

### 7.2.3 Features Not to be Tested

1. Results Page: The prediction based on the input will be displayed with the Explainable AI feature and the user will be able to view it. No testing to be done on this page.

### 7.2.4 Testing Tools and Environment

#### 7.2.4.1 Testing Tools

1. Selenium IDE

#### 7.2.4.2 Environment

Hardware Configuration:

1. 16 GB Ram
2. 120 GB Hard Disk
3. 2.6 GHZ Processor

Software Configuration:

1. Google Chrome
2. Microsoft Visual Studio Code
3. Microsoft Edge
4. Xampp Server

## 7.3 Test Cases

### 7.3.1 Testing Login using Selenium

Test case : Login Page				
Test Case	Description	Input	Expected Output	Actual Output
Login 1	To Test if the website proceeds as expected on valid credentials	Valid credentials	Login Success	-
Login 2	To Test if the website proceeds as expected on invalid credentials	Invalid credentials	"Invalid Credentials Entered"	-

### 7.3.2 Testing Register using Selenium

Test case : Register Page				
Test Case	Description	Input	Expected Output	Actual Output
Register 1	To register a user to access website features	Valid Credentials	Register Success(User added to database)	-
Register 2	To register a user to access website features	Invalid credentials	Error occurred User not created	-

### 7.3.3 Testing Input CT-Scan using Selenium

Test case : Checkout Page				
Test Case	Description	Input	Expected Output	Actual Output
Input CT-Scan Page 1	To test whether the form proceeds on uploading Valid CT-Scan image format	Valid CT-Scan format	Form Submits successfully	-
Input CT-Scan Page 2	To test whether the form proceeds on uploading invalid CT-Scan image format	Invalid CT-Scan format	Form shows invalid format	-

## Chapter 8

# Conclusions and Scope for Future Work

*This chapter presents the Conclusion and the scope for future work. It gives a brief description about the outcomes and results of the project. It also throws light on the scope of this application and discusses various features that can be implemented in the future.*

### 8.1 Conclusion

We have built a Covid-19 disease detection web application and have made the software documentation for the same. The application satisfies the requirements mentioned in the SRS document. The user interface is simple and can be used by any naïve user. No technical knowledge is required to use the application. The risks associated with the project have been mentioned in the RMMM plan. We were able to take care of these risks while completing the project. However, we understand that some of the risk mitigation techniques need cooperation from the side of the client and those risks have been mentioned appropriately in the RMMM plan. Agile process model was followed to make sure that the changes needed are incorporated in successive releases.

### 8.2 Future scope

A few features can be added to the existing application.

- The following algorithm can be extended to many other diseases which require CT-Scans to detect the anomaly.
- User can book an appointment to the hospitals upon receiving the positive report.

## References

1. <https://www.w3schools.com/>
2. <https://stackoverflow.com/>
3. <https://www.lucidchart.com/blog/types-of-UML-diagrams>
4. <https://dev.mysql.com/doc/>
5. <https://trello.com/power-ups/5970d4298c14fdf691c95a76/teamgantt>
6. <https://health.economictimes.indiatimes.com/news/policy/health-ministry-comes-out-with-draft-rules-on-sale-of-drugs-by-e-pharmacy/65641297>
7. <https://creately.com/blog/diagrams/uml-diagram-types-examples/>
8. <https://www.tutorialspoint.com/bootstrap/index.htm>
9. <https://cloud.google.com/explainable-ai/>
10. <https://ieeexplore.ieee.org/document/9299286>
11. <https://www.geeksforgeeks.org/introduction-to-explainable-ai-xai-using-lime/>

## Acknowledgement

It gives immense pleasure in bringing out this synopsis of the project entitled "**Covid-19 Disease Detection with Explainable Artificial Intelligence**". We whole heartedly express our sincere gratitude to **Prof. Nilkamal More** who encouraged us to work on this project and gave us her valuable suggestions and ideas when we were in need of them. We are also grateful to our college, **K. J. Somaiya College of Engineering**, for giving us this wonderful opportunity to work on this project and all the respected faculties for intimate cooperation and providing us the necessary resources and instructions throughout the period of project completion. We would also like to thank our friends who helped us with the doubts and gave their valuable suggestions for the project as it would not have been possible to develop the project within the prescribed timeline.