System Design for Software Engineering

Team 17, Team RadiAIdance
Allison Cook
Ibrahim Issa
Mohaansh Pranjal
Nathaniel Hu
Tushar Aggarwal

January 18, 2024

1 Revision History

Date	Version	Notes
01/13/2024	0.0	Initial draft of all sections
01/15/2024	0.1	Edits to the component diagram and the connections
01/17/2024	0.2	Final edits and check for grammar and clarity

2 Reference Material

This section records information for easy reference.

2.1 Abbreviations and Acronyms

symbol	description
Software Engineering	Explanation of program name
AUC	Area Under Curve
DICOM	Digital Imaging and Communications in Medicine; technical standard for digital storage/transmission of medical images and related information
JPEG/JPG	Joint Photographic Experts Group; digital image compression standard, image format
M	Module
MG	Module Guide
MIS	Module Interface Specification
MVC	Model-View-Controller Software Architecture
NLP	Natural Language Processing
R	Requirement
ROC curve	Receiver Operating Characteristic curve, a graph to show the performance of a model, plots true positive rate and false positive rate
SRS	Software Requirement Specification
Software Engineering	The Process of Designing and Developing Software; a reference to the software application described in this document

Contents

1 Revision History **i**

2 Reference Material **ii**

2.1 Abbreviations and Acronyms ii

3 Introduction **1**

4 Purpose **1**

5 Scope **1**

6 Project Overview **2**

6.1 Normal Behaviour 2

6.2 Undesired Event Handling 3

6.3 Component Diagram 4

6.4 Connection Between Requirements and Design 4

7 System Variables **5**

7.1 Monitored Variables 5

7.2 Controlled Variables 5

7.3 Constants Variables 5

8 User Interfaces **5**

9 Timeline **5**

A Interface **7**

B Reflection **12**

List of Tables

1 Implementation Timeline 6

List of Figures

1 Context Diagram describing The Context of the Work 2
2 Component Diagram that shows the interaction of components 4
3 Login page, the first page seen by users 7
4 Main page with all main functions 8
5 Search page used to search through all existing patient records 9
6 Page that displays a selected patients records 10
7 Page which displays the results of a new chest-x-ray uploaded to the system 11
8 The page that allows new patients to be added to the system before conducting a new chest-x-ray review 12

3 Introduction

This document aims to outline the system design that meets the requirements defined in the project SRS (SRS Section 2). It will cover the general system with more details about the software architecture outlined in the MG document, and the detailed module interface specification in the MIS document.

4 Purpose

The purpose of this document is to communicate the overall system design. This document focuses on outlining how the system interacts with its environment, the expected operation and unexpected event handling, the explicit relation between design decisions and the requirements seen in the SRS (SRS Section 2), how the system is divided into components, user interface, and the timeline and any constraints for the project.

The purpose of the system is to aid medical professionals and specifically radiologists by reducing the amount of time they need to spend reviewing chest X-rays of patients by conducting a preliminary scan on a patient x-ray for four diseases, Atelectasis, Cardiomegaly, Pleural Effusion, and Pneumonia, with a greater AUC of the ROC than existing systems to create a higher level of confidence in the systems correct predictions.

5 Scope

The system is designed for use by medical professionals at hospitals, labs and offices to conduct a preliminary scan of a patient's chest X-ray and highlight the risk of four diseases identifiable from chest X-rays.

The diagram below outlines where our system interacts with the surrounding environment and the boundaries of our system.

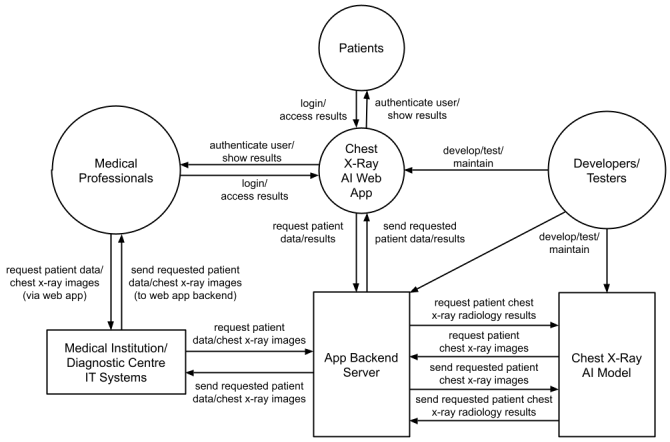

Figure 1: Context Diagram describing The Context of the Work

6 Project Overview

6.1 Normal Behaviour

The system is to be used through a web-application that requires a computer and stable access to the internet. This allows for the system to be used within any existing enterprise systems within hospitals, offices, and labs without the need to understand and integrate into each individual use.

1. **Chest X-Ray Read Module: Reads the JPEG of the X-ray, after conversion from DICOM.**
2. **Results Generation Module: Generates a disease risk classification for the X-ray input.**
3. **Report Component Generation Module: Generates a brief diagnosis report for the X-ray input.**
4. **Database Operations Module: Contains database management functions for the patient database.**
5. **User Authentication/Management Module: Responsible for authorizing user login credentials.**
6. **App GUI Module: The graphical user interface of the system.**
7. **Login Module: Login page UI view.**
8. **Perform Scan Module: UI view for submitting X-ray input to the system to perform a diagnosis.**

9. **View Results Module: View for the results of the scan and disease report.**
10. **AI Model Module: AI model for identifying diseases in the X-ray input.**
11. **NLP Model Module: Model for generating a breif diagnosis report using NLP.**
12. **Backend Module: Module that interfaces with DICOM server containing patient data.**
13. **App Controller Module: Controller for interacting with the different models and views.**
14. **Medical Institution Interface Module: Allows information upload, for DICOM files, between application and the medical institutions' IT system.**

6.2 Undesired Event Handling

The system handles undesired events, such as incorrect data, by first checking that any alterations to the data or system will not result in an unsafe state. Should the system receive improper data or experience a connection issue it will prevent further operation until the issue has deemed resolved by the system. The user will need to recommence their actions from login in order to prevent storage of incorrect data.

6.3 Component Diagram

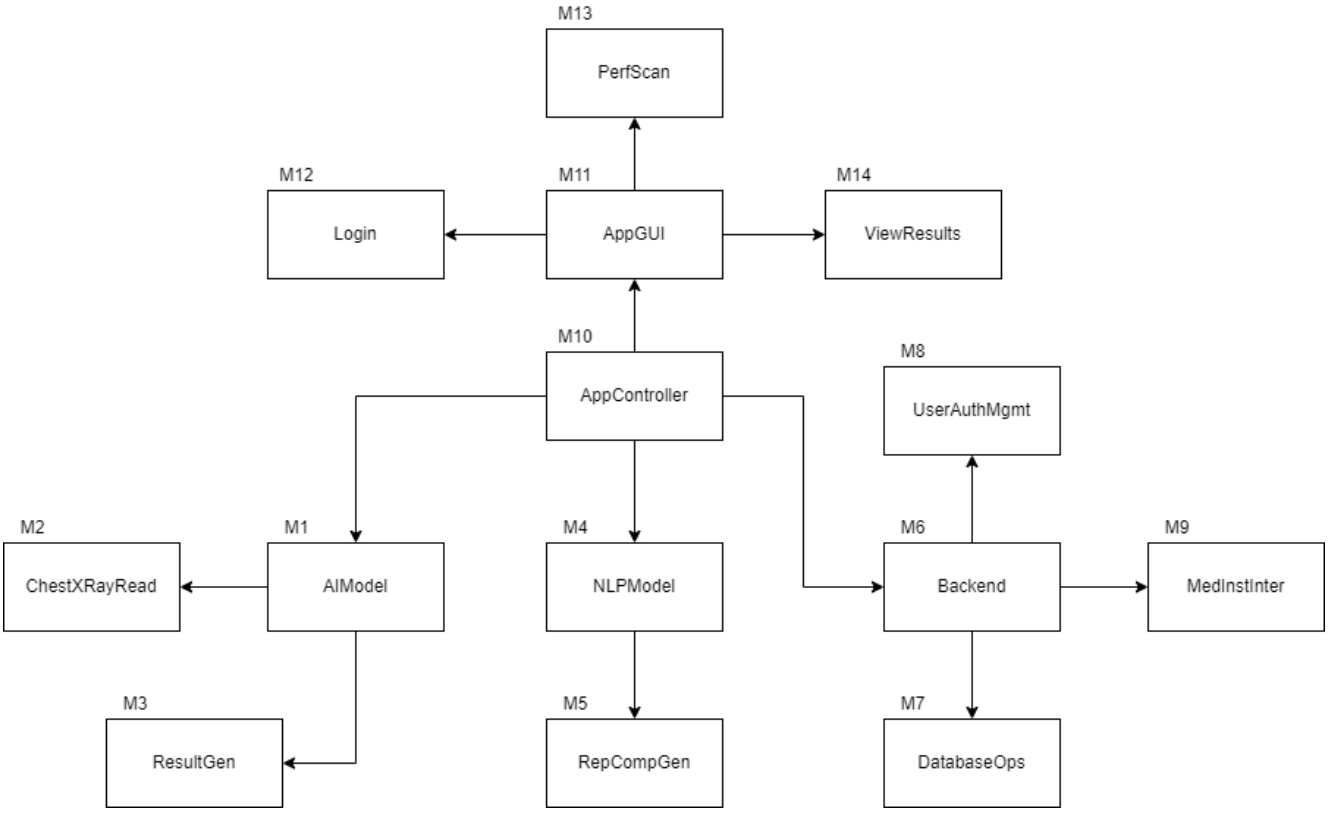

Figure 2: Component Diagram that shows the interaction of components

6.4 Connection Between Requirements and Design

The design of the system is intended to satisfy the requirements developed in the SRS.

1. FR4: We decided that the best way to display a structured set of data about a patient was to display all findings in table format.

2. FR11: To allow for multiple different key searches we decided to have one search entry field but allow the user to select an option on what data is used in the search. (ie. ID or name)

3. NFAR0: Our choice for 'clean' design meant minimal colour changes and minimal feature other than those necessary.

4. NFLR0: The system follows similar design to other systems already used in medical offices/hospitals, so there is minimal learning required for operation.

5. NFSR0: A description of the main features and functionality of the system will be available on the main page.

6. NFAR0: Users will require a username and password to verify authorization to patient data.

7 System Variables

7.1 Monitored Variables

N/A

7.2 Controlled Variables

N/A

7.3 Constants Variables

N/A

8 User Interfaces

Our system only has a software component for user interaction, and in following the non-functional requirements defined in SRS sections 3.1 and 3.2, uses a simple, consistent colour theme and only the necessary elements for operation. It also follows common design principles and patterns to be conducive to intuitive understanding and quick comprehension of the system. The appendix contains all the current web page designs.

9 Timeline

Deliverable	Assigned to	Deadline
Model Module		
ChestXRayread	Tushar, Mohaansh, Ibrahim	January 21, 2024
ResultsGen	Ibrahim	January 21, 2024
ResCompgen	Tushar	January 21, 2024
DatabaseOps	Nathaniel	January 22, 2024
UserAuthMgmt	Nathaniel	January 22, 2024
MedInstInter	Allison	January 21, 2024
View Module		
Login	Allison	January 24, 202
PerfScan	Allison	January 24, 202
ViewResults	Allison	January 24, 2024
Controller Module		
AIModel	Tushar, Mohaansh, Ibrahim	January 20, 2024
NLPModel	Allison	February 1, 2024
Backend	Nathaniel	January 21, 2024
AppController	Nathaniel	January 21, 2024
AppGUI	Allison	January 21, 2024
Testing		
Model Module	Mohaansh with participants	February 3, 2024
View Module review	Allison with participants	January 26, 2024
Controller Module	Allison, Nathaniel	February 2, 2024
System	Allison, Ibrahim, Mohaansh, Nathaniel, Tushar	February 5, 2024

Table 1: Implementation Timeline

A Interface

The interface appendix includes all Figma images that illustrate the design of the user interface.

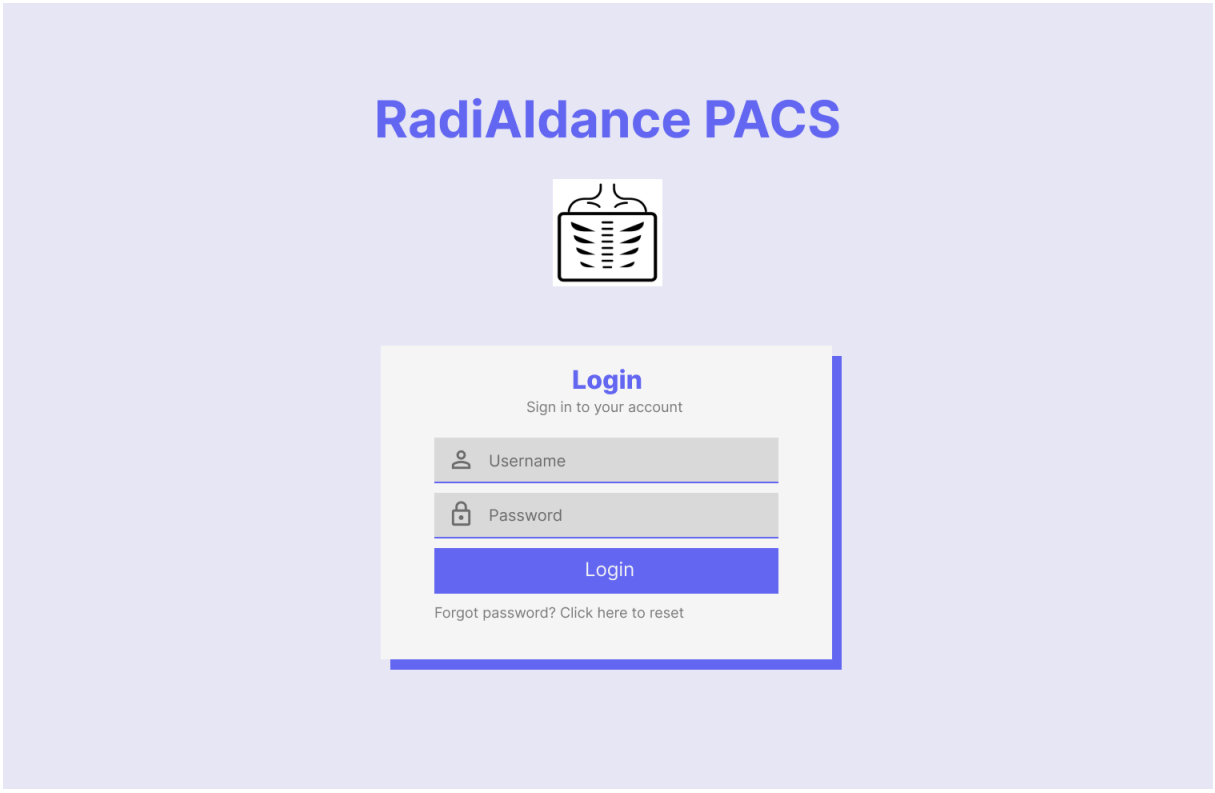

Figure 3: Login page, the first page seen by users

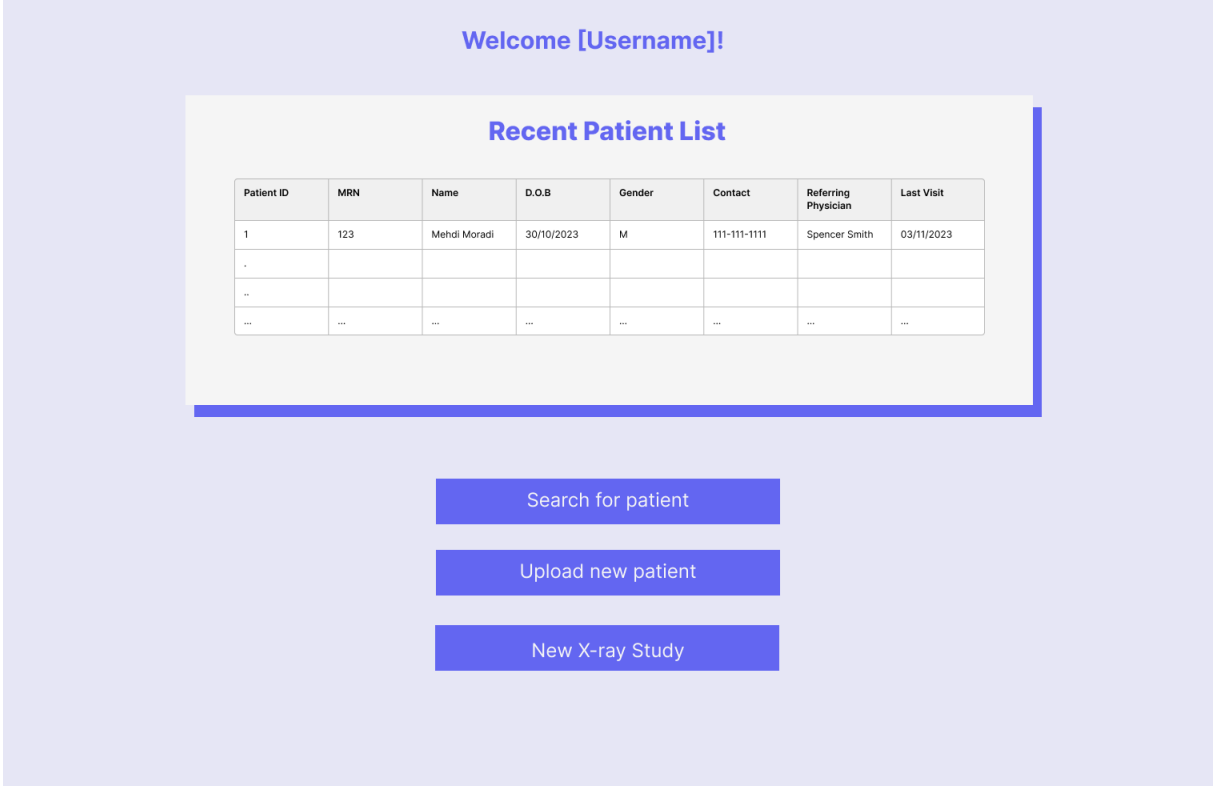

Figure 4: Main page with all main functions

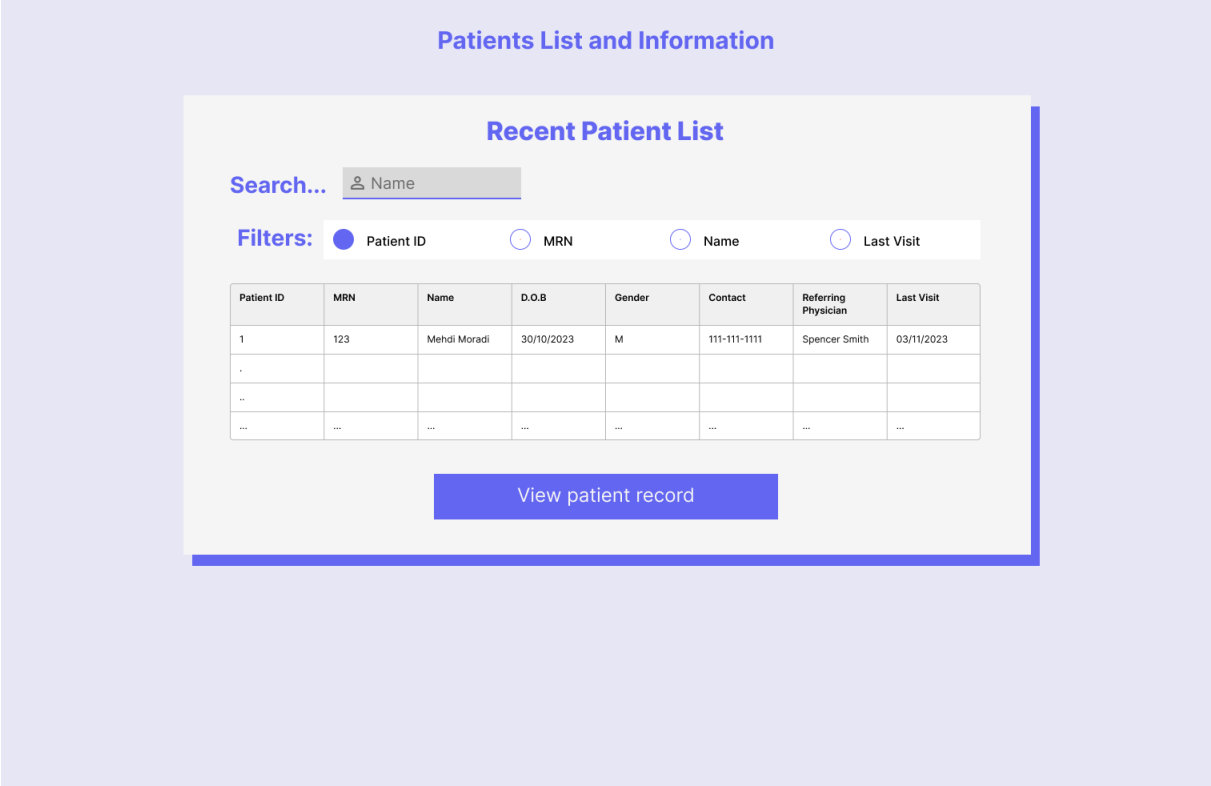

Figure 5: Search page used to search through all existing patient records

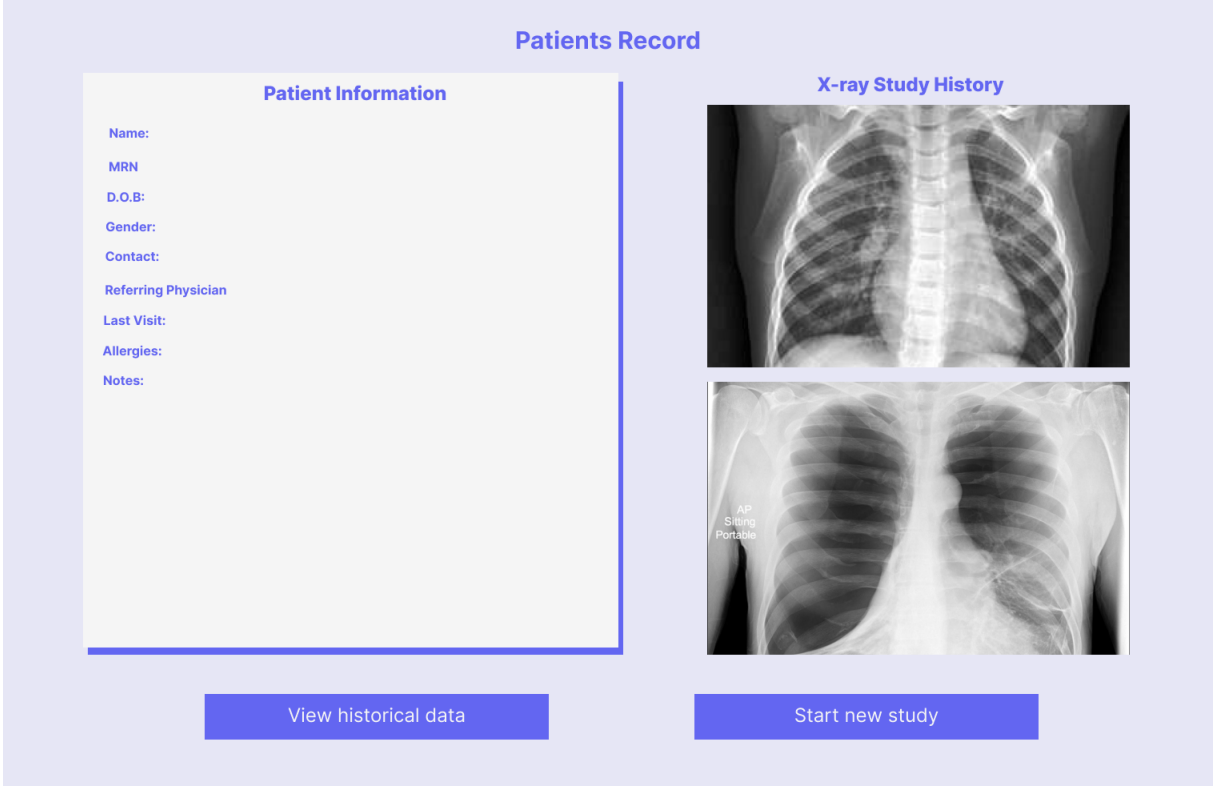

Figure 6: Page that displays a selected patients records

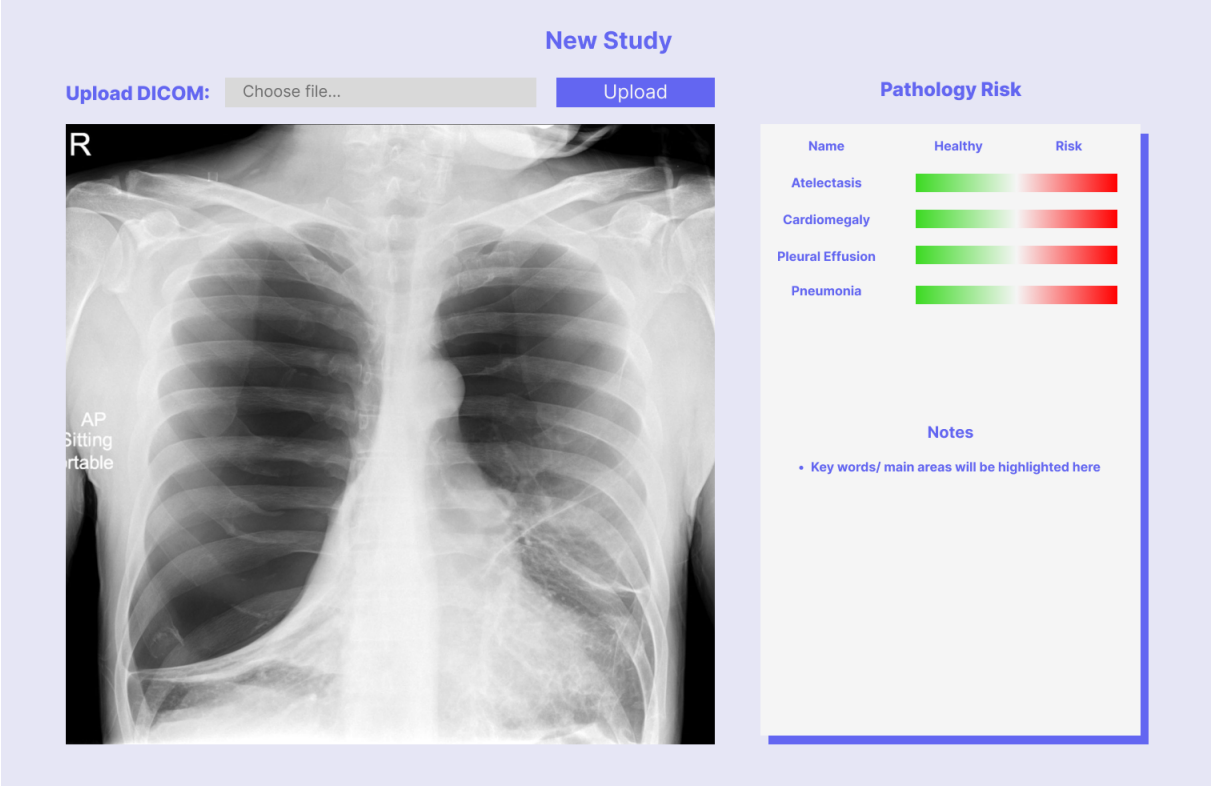

Figure 7: Page which displays the results of a new chest-x-ray uploaded to the system

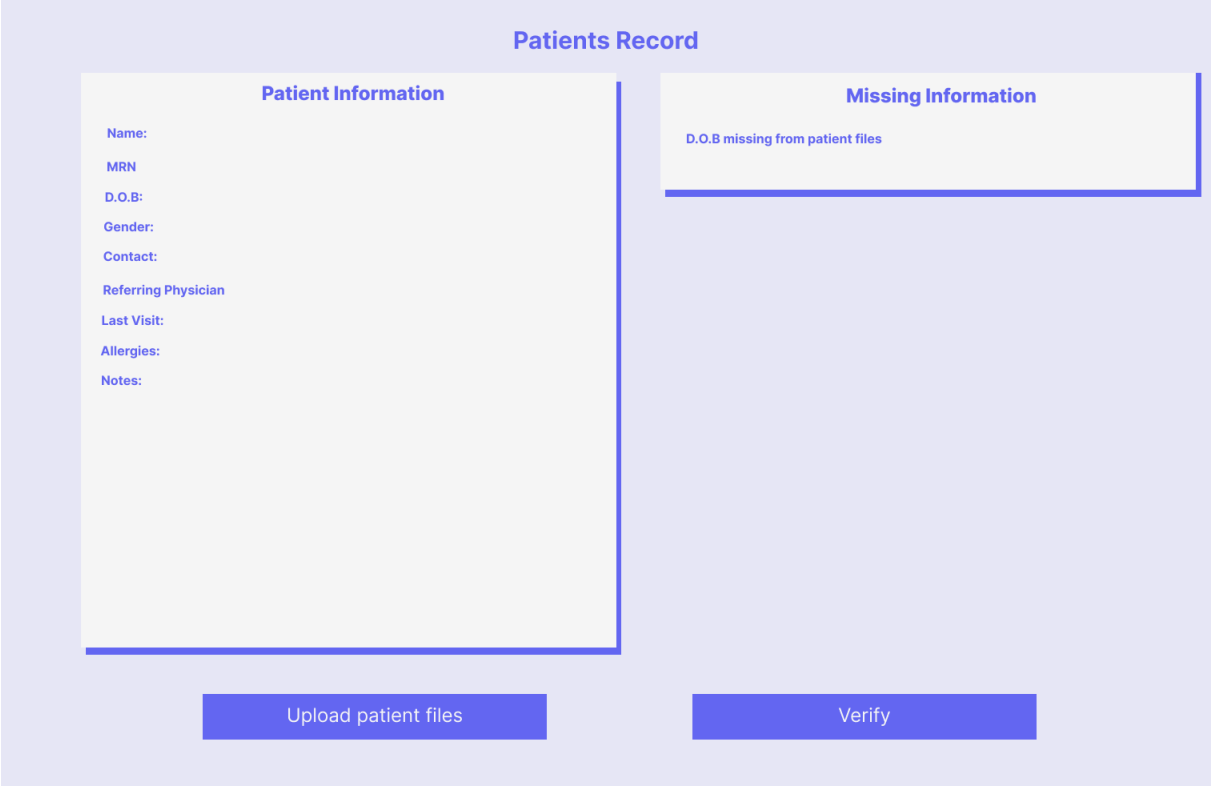

Figure 8: The page that allows new patients to be added to the system before conducting a new chest-x-ray review

B Reflection

1. Some limitations of the solution is the number of diseases, we are focusing on improving the AUC of the ROC for only 4 diseases, with more time, computational power, access to better dataset and/or additional time to conduct our own data creation with material professional (doctors & radiologists), we could aim to improve the AUC of ROC for more diseases, allowing the solution to give more time to the limited doctors to be able to save more people and improve the medical system.

2. Some other design solutions we considered were making the application as a local desktop app. Instead, the data is stored on a cloud based system (web app).

3. The alternate design choice of making a desktop application would pose storage limitations, and there would need to be special considerations for device and system specifications (e.g. development for macOS, windows, etc.), The software was chosen to be a web app because it would be easier to use a cloud service to store data, and web pages are a more universal and easier to maintain interface. A trade-off for this is that specific data security protocols would need to be implemented as it is not local.