



Ain Shams University
Faculty of Computer & Information Sciences
Bioinformatics Department

Detection of Respiratory Diseases

By:

Kareem Mohamed Wardany
Karim Tarek Emam
Omar Mostafa Mohamed
Mark Sameh William
Monica Rafik William
Mohamed Khaled Abd El Aziz

Under Supervision of:

Rasha Mohamed Ismail
Professor in Information Systems Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

Eslam Taher Sharshar
Assistant Lecturer in Bioinformatics Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

Abstract

The healthcare industry is rapidly evolving, and advancements in technology are transforming the way medical care is delivered. With the development of advanced medical systems, patients can now benefit from a host of features that make remote healthcare delivery more efficient and effective.

In addition to the existing features mentioned, patients can now access personalized health recommendations based on their medical history and lifestyle factors. This predictive analysis can help detect potential health risks early and prevent complications.

Doctors can now provide teleconsultations, allowing patients to access medical care from the comfort of their homes. This feature helps save time and resources while still providing patients with quality medical care. Additionally, doctors can access the patient's complete medical history, enabling them to make informed decisions regarding the patient's treatment plan.

Radiologists can now use advanced image recognition technology to analyze medical scans more accurately and efficiently. This technology can detect subtle changes in the scans that may not be visible to the human eye, leading to more accurate diagnoses and better patient outcomes.

Patient's data can now be securely stored and accessed by authorized healthcare providers. This ensures that patient privacy is protected, and only authorized personnel can access the patient's data.

Overall, the development of advanced medical systems has revolutionized the healthcare industry, making medical care more accessible, efficient, and effective. As technology continues to advance, we can expect to see even more innovative features and improvements in the future.

الملخص

تتطور صناعة الرعاية الصحية بسرعة ، وتؤدي التطورات في التكنولوجيا إلى تغيير طريقة تقديم الرعاية الطبية. مع تطور الأنظمة الطبية المتقدمة ، يمكن للمرضى الآن الاستفادة من مجموعة من الميزات التي تجعل تقديم الرعاية الصحية عن بعد أكثر كفاءة وفعالية.

بالإضافة إلى الميزات الموجودة المذكورة ، يمكن للمرضى الآن الوصول إلى التوصيات الصحية الشخصية بناء على تاريخهم الطبي وعوامل نمط حياتهم. يمكن أن يساعد هذا التحليل التنبئي في اكتشاف المخاطر الصحية المحتملة مبكراً ومنع المضاعفات.

يمكن للأطباء الآن تقديم الاستشارات عن بعد ، مما يسمح للمرضى بالوصول إلى الرعاية الطبية من منازلهم المريحة. تساعد هذه الميزة في توفير الوقت والموارد مع الاستمرار في توفير رعاية طبية عالية الجودة للمرضى. بالإضافة إلى ذلك ، يمكن للأطباء الوصول إلى التاريخ الطبي الكامل للمريض ، مما يمكنهم من اتخاذ قرارات مستنيرة بشأن خطة علاج المريض.

يمكن لأخصائي الأشعة الآن استخدام تقنية التعرف على الصور المتقدمة لتحليل الفحوصات الطبية بشكل أكثر دقة وكفاءة. يمكن لهذه التقنية اكتشاف التغيرات الطفيفة في عمليات المسح التي قد لا تكون مرئية للعين البشرية ، مما يؤدي إلى تشخيصات أكثر دقة ونتائج أفضل للمرضى.

يمكن الآن تخزين بيانات المريض بشكل آمن والوصول إليها من قبل مقدمي الرعاية الصحية المعتمدين. هذا يضمن حماية خصوصية المريض ، ويمكن للموظفين المصرح لهم فقط الوصول إلى بيانات المريض.

بشكل عام ، أحدث تطوير الأنظمة الطبية المتقدمة ثورة في صناعة الرعاية الصحية ، مما جعل الرعاية الطبية أكثر سهولة وكفاءة وفعالية. مع استمرار تقدم التكنولوجيا ، يمكننا أن نتوقع رؤية المزيد من الميزات والتحسينات المبتكرة في المستقبل.

Table of Contents

Abstract	1
List of Figures	3
List of Tables	4
List of Abbreviations	5
1. Introduction	6
1.1 Motivation	6
1.2 Problem Definition	8
1.3 Objective	9
1.4 Time Plan	10
1.5 Document Organization	11
2. Background	13
3. Analysis and Design	21
3.1 System Overview	21
3.1.1 System Architecture	21
3.1.2 Functional Requirements	22
3.1.3 Nonfunctional Requirements	23
3.1.4 System Users	24
3.2 System Analysis & Design	26
3.2.1 Use Case Diagram	26
3.2.2 Class Diagram	27
3.2.3 Sequence Diagram	29
3.2.4 Database Diagram	42
4. Implementation and Testing	44
5. User Manual	83
6. Conclusion and Future Work	106
6.1 Conclusion	106
6.2 Future Work	107
References	108

List of Figures

Figure 2.1 residual block	17
Figure 2.2 ResNET Model	19
Figure 2.3 224x224 Model confusion matrix	20
Figure 2.4 512x512 Model confusion matrix	20
Figure 3.1 System Architecture	21
Figure 3.2 System Use Case	26
Figure 3.3 System Class Diagram	27
Figure 3.4 Predict Scan sequence diagram.	29
Figure 3.5 Chat with doctor sequence diagram	30
Figure 3.6 Purchase VIP sequence diagram	31
Figure 3.7 Show All Prescriptions sequence diagram	32
Figure 3.8 Purchase Coins sequence diagram	33
Figure 3.9 Chat with patient sequence diagram	34
Figure 3.10 Assign Patient sequence diagram	35
Figure 3.11 Create Prescription sequence diagram	36
Figure 3.12 Report Patient sequence diagram	37
Figure 3.13 Close Chat sequence diagram	38
Figure 3.14 Predict Multiple X-ray Scan sequence diagram	39
Figure 3.15 Handle Reports sequence diagram	40
Figure 3.16 Verify Doctor sequence diagram	41
Figure 3.17 Database Diagram	42
Figure 4.1 Doctor Register Page	44
Figure 4.2 Patient Register Page	45
Figure 4.3 Radiology Register Page	45
Figure 4.4 Patient Main Page	46
Figure 4.5 Predict X-ray Scan Patient Feature	46
Figure 4.6 Chat with Doctor Patient Feature	47
Figure 4.7 Purchase VIP Patient Feature	47
Figure 4.8 Purchase Coins Patient Feature	48
Figure 4.9 Doctor Main Page	48
Figure 4.10 Chat with Patient Doctor Feature	49
Figure 4.11 Patients Request	49
Figure 4.12 Radiologist X-ray Prediction Feature	50

List of Tables

Table 1.1 Time Plan	10
Table 2.1 Dataset Split	16
Table 3.1 Description of System Classes	28
Table 3.2 Database Table Description	43
Table 4.1 User Class	51
Table 4.2 Patient Class	52
Table 4.3 Patient GUI Class	53
Table 4.4 Doctor Class	55
Table 4.5 Doctor GUI Class	56
Table 4.6 Radiologist Class	57
Table 4.7 Radiologist GUI Class	57
Table 4.8 Administrator Class	58
Table 4.9 Administrator GUI Class	59
Table 4.10 UserFactory Class	59
Table 4.11 Starter Class	60
Table 4.12 ResNet Model Class	61
Table 4.13 ImageProcessing Class	63
Table 4.14 Database Class	64
Table 4.15 Client Class	65
Table 4.16 Server Class	66
Table 4.17 Patient Class and Patient GUI Class Unit Testing	67
Table 4.18 Doctor Class and Doctor GUI Class Unit Testing	70
Table 4.19 Radiologist Class and Radiologist GUI Class Unit Testing	75
Table 4.20 Administrator Class and Administrator GUI Class Unit Testing	76
Table 4.21 Starter Class Register Part Unit Testing	79
Table 4.22 Starter Class Login Part Unit Testing	81

List of Abbreviations

Abbreviation	What the abbreviation stands for
ChexNet	CheXpert Neural Network
CNN	convolutional neural network
COCO	Common Objects in Context
conv	convolution
COPD	chronic obstructive pulmonary disease
CSV	Comma-Separated Values
CT	Computed Tomography
CXR	Chest X-ray
DenseNet	Densely Connected Convolutional Network
FC	Fully Connected
GUI	Graphical User Interface
PDF	Portable Document Format
ReLU	Rectified Linear Unit
ResNet	Residual Network
TB	Tuberculosis
UI	User Interface
Xception	Extreme Inception
VIP	Very Important Person

1. Introduction

1.1 Motivation

The healthcare industry is facing several challenges, including increasing demand for medical care, shortage of healthcare professionals, and limited resources. This project aims to address these challenges by providing a system that offers several features for patients, doctors, and radiologists.

For patients, the project provides a prediction model that can predict five diseases based on chest x-ray scans. This helps patients to detect potential health risks early and take necessary precautions. Additionally, patients can communicate with a chatbot to identify their health condition and access medical care remotely. Patients can also download prescriptions, making it easier to access medical care from the comfort of their homes. The project also offers a VIP level, which reduces the amount of coins required for medical consultations.

For doctors, the project offers a platform to provide medical care remotely. Doctors can view chat histories to better understand the patient's condition, enabling them to make informed decisions regarding the patient's treatment plan. Doctors can also report patients who violate the system's rules, ensuring that the system is safe for everyone. Doctors can earn coins for each chat, which they can exchange for real money.

For radiologists, the project offers a model that can analyze multiple scans simultaneously, making it easier to detect diseases accurately and efficiently. This feature helps radiologists to save time and resources, enabling them to focus on providing quality medical care.

Overall, this project's impact on the healthcare industry cannot be overstated, this project is needed to address the challenges faced by the healthcare industry. The project's potential to revolutionize the industry and improve healthcare outcomes is immense, making it an essential tool for the healthcare industry's continued evolution. By providing a system that offers several features for patients, doctors, and radiologists, the project helps to make medical care more accessible, efficient, and effective. The project also helps to reduce the burden on healthcare professionals and resources, enabling the industry to provide quality medical care to more people.

1.2 Problem Definition

The healthcare industry is facing numerous challenges, such as limited resources, high costs, and long wait times. In addition, with the current global pandemic, access to medical care has become even more challenging, with patients avoiding in-person appointments to reduce their risk of contracting the virus. These challenges have led to the development of various technological advancements that aim to provide better access to medical care for patients, and one such advancement is the development of a system that offers several features for patients, doctors, and radiologists.

By providing a system that offers various features for these groups, the project aims to make medical care more accessible, efficient, and effective. The system also reduces the burden on healthcare professionals and resources, enabling the industry to provide quality medical care to more people.

One of the significant benefits of the system is its ability to provide medical care remotely. This feature is particularly crucial in times of pandemics, such as the current COVID-19 pandemic, where people are avoiding in-person appointments to reduce their risk of contracting the virus. The system's prediction model and chatbot can provide preliminary diagnoses, reducing the need for patients to see a doctor in-person, and enabling medical care to be provided remotely.

The system's ability to provide medical care remotely also enables patients to access medical care from anywhere at any time. This feature is particularly beneficial to patients who live in remote areas, where access to medical care is limited, or those with mobility issues. With the system, patients can access medical care from the comfort of their homes, reducing the need for them to travel long distances to see a doctor.

1.3 Objective

1. Patient Objectives:

- Predict chest x-ray scan results for diseases like Covid-19, Fibrosis, bacterial or viral pneumonia, and tuberculosis using the prediction model.
- Engage in a conversation with the chat bot to assess their health condition and receive initial advice.
- Access doctors available on the system for further consultation by using system coins.
- Purchase a VIP level to reduce the number of coins required for predictions or doctor consultations.
- Download prescriptions written by doctors for their condition.

2. Doctor Objectives:

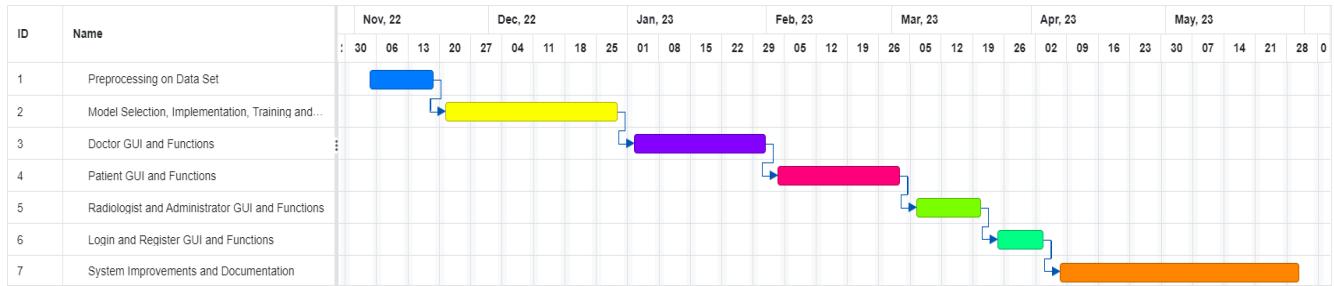
- Assist patients who are seeking medical advice and treatment for their health condition.
- Review the chat history between patients and the chat bot to gain insight into the patient's symptoms and condition.
- Prescribe appropriate medications and treatments for the patient's condition and save the prescription.
- Report patients who violate system rules and provide reasons for the report.
- Receive coins as compensation for their services, which can be exchanged for real money.

3. Radiologist Objectives:

- Utilize the prediction model to simultaneously predict results for multiple scans.
- Analyze and interpret the results of chest x-ray scans efficiently.
- Provide accurate and timely diagnoses based on the scan findings.
- Collaborate with doctors in providing comprehensive medical assessments and recommendations.

1.4 Time Plan

Table 1.1 Time Plan



The project began in November with the initial stage of searching for a suitable dataset and performing preprocessing techniques. In the last week of November, the second stage commenced, which involved searching for a Deep Learning model for the system. The selected model was then implemented, and the training, validation, and testing processes were carried out. The objective was to achieve a model accuracy of over 90%.

The third stage, which spanned from January to March, was the longest phase. It involved creating UI designs tailored to each user who would utilize our system. These designs were then transformed into an actual working system.

The fourth stage, taking place from April to May, focused on system improvements and the finalization of system documentation.

1.5 Document Organization

In the background section, we begin by providing a comprehensive overview of the project's field, elucidating the nature of each disease and its mode of transmission, as well as outlining potential remedies in case of detection. Furthermore, we present existing models that bear resemblance to our project, establishing a context for our work. Next, we delve into the process through which we obtained our dataset and describe the methodology we employed to partition it effectively. Subsequently, we introduce our model, offering pertinent details regarding its characteristics and the underlying architecture. By adopting this structured approach, we aim to foster a thorough understanding of the subject matter, laying the foundation for our project's subsequent development and analysis.

In the Analysis and Design section, we commence by meticulously examining the system architecture pertinent to the project. Subsequently, we delve into elucidating the intended users for the system, along with the requisite skills expected from these users. Following this, we present a comprehensive portrayal of the system through the Use Case Diagram, outlining its various functions and interactions. To further enhance our understanding, we proceed to illustrate the Class Diagram, which encapsulates the structure and relationships among the classes within our system. Subsequently, we meticulously describe each class, expounding upon their respective attributes and behaviors. Furthermore, we augment our comprehension by employing Sequence Diagrams, which meticulously delineate the sequence of interactions and behaviors for the system's crucial functions. To address the vital data management aspect, we elucidate the database schema and provide detailed descriptions for the tables within the database, meticulously outlining their attributes and relationships. Through this comprehensive and meticulous approach, we aim to establish a solid foundation for the subsequent phases of the project.

In the Implementation and Testing section, we begin by presenting the initial UI designs for the system, tailored to each user who will utilize it. These designs serve as a visual representation of the system's interface and provide a foundation for further development. Subsequently, we delve into a comprehensive description of each class within our system. Additionally, we showcase the outcomes of the testing phase, employing unit testing to validate the performance and reliability of each function in every class. Through rigorous testing, we ensure that each function operates as intended, identifying and rectifying any discrepancies or errors. This systematic approach to implementation and testing aids in the refinement and optimization of our system, enhancing its overall quality and usability.

In the User Manual's first section, the initial chapter provides a comprehensive guide on installing the system setup. Moving forward to the second chapter, users will be instructed on how to conveniently log into our system. The third chapter focuses on the creation of user accounts, offering step-by-step instructions for seamless account setup. As we progress to the fourth chapter, users assuming the role of patients will find an inclusive guide tailored to their specific needs, ensuring their smooth navigation and utilization of our system. Subsequently, the fifth chapter is dedicated to users taking on the role of doctors, furnishing them with detailed instructions to maximize their experience and effectively utilize the system's features. Lastly, the sixth chapter provides an extensive guide for users acting as radiologists, granting them comprehensive knowledge on leveraging our system optimally within their profession. Through this meticulously structured and organized manual, we aim to empower users with a clear understanding of the system's installation, login procedures, account creation, as well as tailored guidance for patients, doctors, and radiologists, allowing them to make the most of our system's capabilities.

2. Background

Respiratory diseases refer to a group of medical conditions that affect the lungs and the respiratory system, which includes the organs responsible for breathing. These diseases can range from minor illnesses like common colds to chronic, life-threatening conditions like COPD, lung cancer, and pulmonary fibrosis. Respiratory diseases are a significant public health concern globally, with millions of people affected by them each year.

The respiratory system is made up of several organs, including the nose, mouth, pharynx, larynx, trachea, bronchi, bronchioles, and alveoli. These organs work together to take in oxygen and expel carbon dioxide, which is a waste product of cellular metabolism. The lungs, which are the primary organs of the respiratory system, are responsible for carrying out these critical functions.

Pulmonary fibrosis is a chronic respiratory disease that causes scarring of the lung tissue, making it difficult to breathe. The exact cause of pulmonary fibrosis is unknown, although it is believed to be caused by long-term exposure to harmful substances like cigarette smoke, air pollution, and occupational dust and chemicals. Symptoms of pulmonary fibrosis can include shortness of breath, dry cough, and fatigue. While there is no cure for pulmonary fibrosis, treatments like oxygen therapy and medications can help manage the symptoms and slow the progression of the disease.

COVID-19 is a respiratory illness caused by the SARS-CoV-2 virus. It first emerged in late 2019 and quickly became a global pandemic. The virus spreads through respiratory droplets when an infected person talks, coughs, or sneezes. The symptoms of COVID-19 can range from mild to severe and can include fever, cough, shortness of breath, fatigue, and loss of taste or smell. Treatment for COVID-19 varies depending on the severity of the illness but can include supportive care like oxygen therapy, antiviral medications, and steroids. Vaccines are also available to prevent infection and reduce the severity of illness.

Pneumonia is a type of respiratory infection that affects the lungs. It can be caused by viruses or bacteria. The symptoms of pneumonia can include cough, fever, chest pain, and shortness of breath. Treatment for pneumonia depends on the underlying cause and can include antibiotics medications, and supportive care like oxygen therapy. In severe cases, hospitalization may be necessary.

Vaccines are also available to prevent certain types of pneumonia, particularly in vulnerable populations like young children and the elderly. Prompt diagnosis and treatment are essential for managing pneumonia and preventing complications.

TB is a bacterial infection that primarily affects the lungs but can also affect other parts of the body. It is caused by the bacteria *Mycobacterium tuberculosis*. The symptoms of TB can include cough, fever, night sweats, weight loss, and fatigue. Treatment for TB usually involves a combination of antibiotics taken over a long period, typically six to nine months. In some cases, drug-resistant TB may require more prolonged and intensive treatment. Preventing the spread of TB involves identifying and treating infected individuals, as well as promoting good respiratory hygiene practices like covering the mouth and nose when coughing or sneezing. Vaccines are also available to prevent TB, although their effectiveness varies.

Prevention and early detection are critical for managing respiratory diseases. This includes avoiding exposure to environmental pollutants, maintaining good respiratory hygiene practices like covering the mouth and nose when coughing or sneezing, and getting regular check-ups with a healthcare provider. Vaccines are also available to prevent certain respiratory illnesses, including influenza and pneumococcal pneumonia, and can be especially important for vulnerable populations like young children, the elderly, and individuals with compromised immune systems.

In a work by Singh et al. [1], a CNN model has been used to identify Covid-19 patients with the help of CT scan images. There are several more research works to detect the presence of the Covid-19 virus in the human lungs with the help of a CT scan [1, 2, 3, 4, 5]. In the work by Yan et al. [2], a multi-task, self-supervised AI model has been developed for the diagnosing of the Covid-19 virus in human lungs with the help of CT scan images, with an accuracy of 89%. Automatic segmentation and quantification of the lungs are done by Shan et. al. [3]. Li et al. [4] describe a fully automatic framework to detect coronavirus affected lungs from chest CT scan images and differentiated it from other lung diseases. However, Ng et al. [6] and Huang et al. [7] have concluded that CXR images are better than any other means in the detection of Covid-19 because of their promising results along with the availability of CXR machines and their low maintenance cost.

There have been multiple works done by researchers in the area of Covid-19 patient detection using CXR images [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]. In one such work by Makris et al.[8], transfer learning has been used with the Inception-v3 network to classify normal, pneumonia and Covid-19 patients using CXR images. Another work by Mangal et al. [19] has used the DenseNet with ChexNet architecture to segregate normal subjects, bacterial and viral pneumonia patients and Covid-19 patients. Rahimzadeh et al. [10] has used a concatenated Xception and ResNet50V2. Another work by Xu et al. [12] has used ResNet to detect viral pneumonia and Covid-19 patients. The unavailability of a large number of image data of Covid-19 positive patients is a challenge faced by most researchers working in this area. Development of COVIDGAN for the generation of data artificially has been done in a work by Waheed et al.[20] which in turn will help in improved Covid-19 detection.

As can be observed, most of the works related to COVID-19 detection from CXR images have utilized individual deep learning models e.g., DenseNet, ResNet, Xception, etc. [11, 13, 14]. None of the works have tried to combine the models to multiply their capability of classification. Various works done on Ensemble Learning with Deep Neural Networks show that ensembling learning methods are superior in prediction than an individual model and also helps in preventing overfitting [21]. In [22], a weighted average of the output probabilities has been introduced as a method for ensembling. It is found to be better than the unweighted average. In another work Cheng et al. [23], the relative performance of the different ensemble methods with Convolutional Neural Networks like unweighted average, majority voting, Bayes Optimal Classifier, and Super Learner has been compared. In this research, we have proposed a new method to ensemble three state-of-the-art CNN models-DenseNet201, Resnet50V2, and Inceptionv3 to classify Covid-19 positive patients from CXR images.

Alright, our goal is to create a prediction model capable of distinguishing between five diseases: Covid-19, Fibrosis, Tuberculosis, and either viral or bacterial pneumonia. To begin, we need to locate a suitable dataset containing images of these diseases. After conducting a thorough search, we have identified a dataset that will allow us to train our model from scratch.

Our dataset is composed of a combination of four other datasets: the NIH Chest X-ray dataset [24], the TB dataset [25], the COVID dataset [26, 27], and the Pneumonia dataset [28].

Table 2.1 Dataset Split

	Covid-19	Normal	Fibrosis	Pneumonia	Tuberculosis	Total
All Data	3616	5765	1686	4265	3500	18832
System Test	25	25	25	25	25	125
Train	2011	3215	930	2374	1946	10476
Model Test 30%	1077	1722	498	1272	1043	5612
Validation 20%	503	803	233	594	486	2619

To clarify, we have organized our dataset into several categories. 'All Data' represents all the images in our dataset. 'System Test' comprises all the images that will be utilized to test the entire system. 'Train' includes all the images used to train our model from scratch. 'Model Test' consists of the images that we will use after the model has been trained, and we will evaluate its accuracy according to our criteria that it should be above 90%. Images of 'Model Test' are calculated by removing the system test images from 'All Data', and the remaining images are split into (train + valid) and test categories with a 30% split for testing. 'Validation' is a group of images used to validate the model during the training process. We calculate it after removing the 'Model Test' images, and the remaining images are split into train and valid with a 20% split for validation images.

Our next step is to choose a deep learning model that can effectively predict from X-ray scans while also being able to handle variations in input. After much consideration, we have determined that ResNet is the ideal model for our purposes. Its capability to train much deeper networks results in improved accuracy, generalization, and robustness to variations in input.

Residual networks (ResNets) are a type of neural network architecture that were introduced in 2015 by Kaiming He [29]. ResNets are designed to solve the problem of vanishing gradients, which can occur in deep neural networks with many layers [30, 31, 32, 33, 34]. Vanishing gradients occur when the gradient of the loss function with respect to the weights in the lower layers of the network becomes very small, which can make it difficult to update the weights in those layers during training. ResNets address this problem by using skip connections, which allow the gradient to flow more easily through the network.

The main idea behind ResNets is to use residual connections, which are shortcut connections that allow information to bypass one or more layers in the network. A residual connection simply adds the input to the output of a layer, so that the output of the layer is the sum of the input and the transformed input. This way, the residual connection ensures that the output of the layer is at least as good as the input, and the network can learn to focus on refining the input rather than trying to reconstruct it from scratch. The residual connections also help to prevent vanishing gradients by providing a direct path for the gradient to flow through the network.

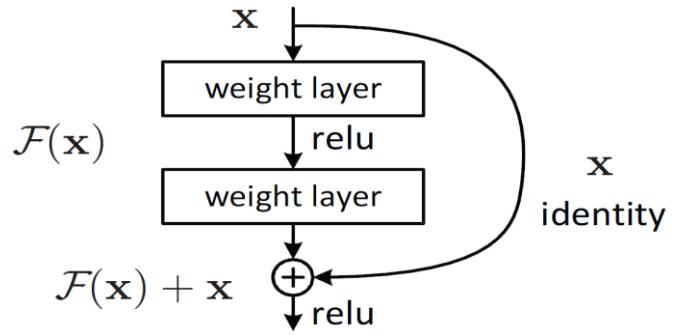


Figure 2.1 residual block

ResNets have been shown to achieve state-of-the-art performance on a wide range of computer vision tasks, including image classification, object detection, and semantic segmentation. They have also been used in other domains, such as natural language processing and speech recognition. ResNets are particularly effective when training very deep networks, with hundreds or even thousands of layers.

There are several different types of ResNets, including the original ResNet with residual connections, as well as variants such as ResNeXt, which uses grouped convolutions to reduce the number of parameters, and DenseNet, which uses dense connections between layers. Each of these architectures has its own strengths and weaknesses, and the choice of which architecture to use depends on the specific problem being addressed.

One of the key advantages of ResNets is their ability to enable the training of very deep networks. Prior to the development of ResNets, training deep neural networks was challenging because the gradients would become very small, making it difficult to update the weights in the lower layers. ResNets solve this problem by allowing the gradients to flow more easily through the network, making it possible to train much deeper networks without encountering the vanishing gradient problem.

Another advantage of ResNets is their ability to achieve high accuracy on a wide range of computer vision tasks. ResNets have been shown to outperform other state-of-the-art architectures on many benchmark datasets, including ImageNet, COCO, and Pascal VOC. This is partly due to the fact that ResNets are able to capture more complex features than previous architectures, thanks to their ability to train much deeper networks.

Despite their advantages, ResNets are not without their limitations. One of the main challenges with using ResNets is the increased complexity of the architecture, which can make them more difficult to train and optimize than simpler architectures. Additionally, ResNets require more computational resources than simpler architectures, which can be a limiting factor in some settings.

Very well, we are currently in the process of constructing our custom ResNet model for the prediction model, and we aim to adhere to the design choices established by He et al. (2016) as closely as possible. This approach should enable us to create a ResNet variant that is optimal for our purposes. In the few instances where we do deviate from He et al.'s design decisions, we will provide justifications for doing so.

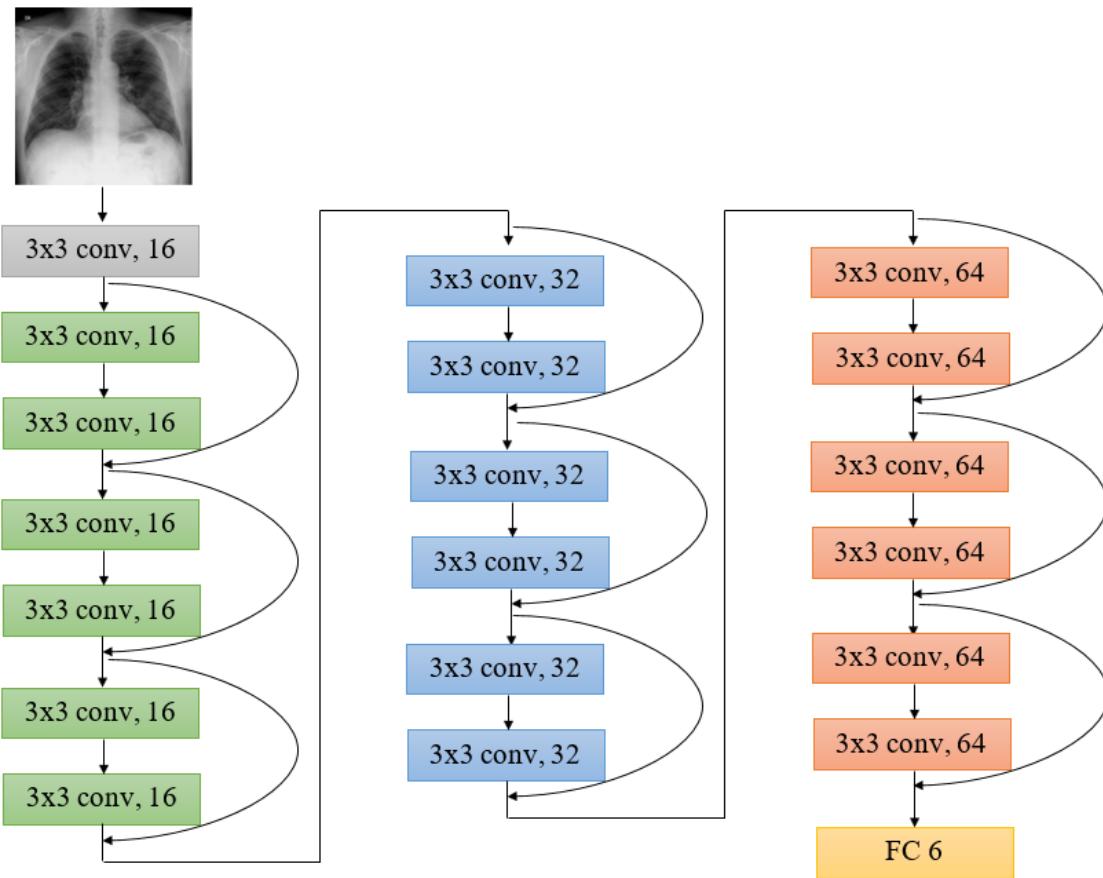


Figure 2.2 ResNET Model

Initially, the model receives an X-ray image which is then resized to a 512x512 format, ensuring that the model can analyze it accurately. Next, the image undergoes convolution using a layer with a 3x3 size and 16 filters. Batch normalization is then applied, followed by the activation function ReLU. The central technique of this model is the residual blocks, which serve as its core. We will apply this technique three times, with each time three blocks will be created.

The initial model was designed to process images of size 224x224, resulting in a test accuracy of 92%. Additional metrics such as precision, which measures the proportion of correctly classified positive instances out of all positive predictions made by the model, also yielded a value of 92%. Similarly, recall, which measures the proportion of correctly classified positive instances out of all actual positive instances in the dataset, had a value of 92%. Furthermore, a confusion matrix was created for the evaluation process.

		Confusion Matrix					
		Normal	COVID	Bacterial PNEUMONIA	Viral PNEUMONIA	Fibrosis	Tuberculosis
True labels	Normal	0.97	0.01	0.00	0.01	0.01	0.00
	COVID	0.01	0.96	0.00	0.00	0.00	0.02
	Bacterial PNEUMONIA	0.02	0.00	0.83	0.14	0.00	0.00
	Viral PNEUMONIA	0.03	0.00	0.35	0.61	0.00	0.00
	Fibrosis	0.01	0.00	0.00	0.00	0.98	0.00
	Tuberculosis	0.00	0.01	0.00	0.00	0.00	0.99

Figure 2.3 224x224 Model confusion matrix

On the other hand, the enhanced model utilizes larger images with a size of 512x512, leading to improved test accuracy of 99%. This improvement was also reflected in other metrics, such as precision and recall, which both achieved values of 99%. A confusion matrix was created to evaluate the enhanced model.

		Confusion Matrix					
		Normal	COVID	Bacterial PNEUMONIA	Viral PNEUMONIA	Fibrosis	Tuberculosis
True labels	Normal	1.00	0.00	0.00	0.00	0.00	0.00
	COVID	0.00	1.00	0.00	0.00	0.00	0.00
	Bacterial PNEUMONIA	0.00	0.00	0.99	0.01	0.00	0.00
	Viral PNEUMONIA	0.00	0.00	0.02	0.98	0.00	0.00
	Fibrosis	0.00	0.00	0.00	0.00	1.00	0.00
	Tuberculosis	0.00	0.00	0.00	0.00	0.00	1.00

Figure 2.4 512x512 Model confusion matrix

3. Analysis and Design

3.1 System Overview

3.1.1 System Architecture

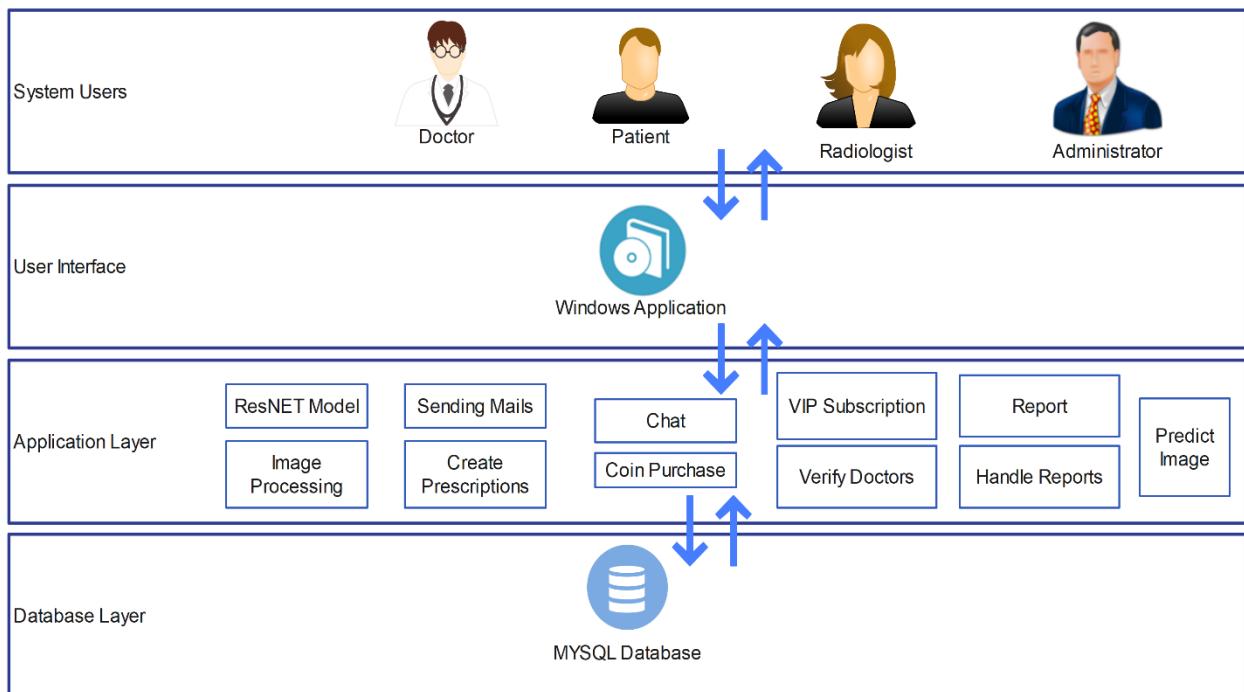


Figure 3.1 System Architecture

The diagram illustrates the different layers in our system. The first layer comprises various system users, such as doctors, patients, radiologists, and system administrators, who will interact with our system. The second layer is the user interface layer, through which these users will access the system. Specifically, they will use a Windows application, which will call all the necessary functions in the application layer based on the user's choices displayed on the screen.

When a function is executed and generates output that needs to be stored, it will be saved in the database layer, which is a MySQL database. This database will serve as a repository for all the data generated and manipulated by the system.

3.1.2 Functional Requirements

For Patients:

1. Chest X-ray Prediction: Patients can use the prediction model to predict the presence of diseases such as Covid-19, Fibrosis, bacterial or viral pneumonia, and tuberculosis based on their chest X-ray scans.
2. Chat Bot: Patients can interact with the chat bot, which will ask them relevant questions to identify their health condition. If the condition is severe, the bot will advise the patient to chat with doctors on the system.
3. Chat with Doctors: Patients can use the system's chat feature to communicate with doctors who are available on the platform. This may require using system coins, which can be purchased by the patient.
4. VIP Level: Patients can purchase a VIP level that provides benefits such as reduced coin requirements for predictions or chats with doctors.
5. Download Prescriptions: Patients can download prescriptions written by doctors for their specific health conditions.

For Doctors:

1. Patient Selection: Doctors can view and select patients who are waiting for a doctor's assistance with their health conditions.
2. Chat History: Doctors can access the chat history between patients and the chat bot to understand the patient's symptoms and previous interactions.
3. Prescription Management: Doctors can create prescriptions for patients, including recommended medications for their conditions, and save them in the system.
4. Reporting: Doctors can report patients if they violate the system's rules, providing a reason for the report. Temporary suspension of the patient from the system can occur until an administrator confirms or revokes the suspension.
5. Coins Conversion: After each chat, doctors can earn coins, which can be exchanged for real money.

For Radiologists:

1. Predictive Model: Radiologists can utilize the system's predictive model to predict multiple scans simultaneously, assisting in diagnosing chest conditions.

3.1.3 Nonfunctional Requirements

1. Performance:

- The prediction model provides accurate and timely results for chest x-ray scans with accuracy above 90%.
- The chat bot responds promptly to patient inquiries and provide relevant information.

2. Scalability:

- It supports a large volume of chest x-ray scans and user interactions without significant performance degradation.

3. Security:

- Patient data, including medical history, securely stored and protected from unauthorized access.
- Access to the system and sensitive patient information is protected through authentication and authorization mechanisms.

4. Reliability:

- The system is available and accessible to patients, doctors, and radiologists with minimal downtime.

5. Usability:

- The user interface is intuitive and user-friendly for patients, doctors, and radiologists.
- The prediction model results, chat interactions, and prescriptions are presented clearly and comprehensibly.
- The chat bot employs natural language processing techniques to understand and respond to patients' queries effectively.

6. Maintainability:

- The system is designed with modular and well-documented code to facilitate future updates and enhancements.

7. Compliance:

- The system adheres to relevant medical and healthcare industry regulations and standards.

3.1.4 System Users

A. Intended Users:

Patients are the key users of this system and stand to benefit greatly from it. Our prediction model can help patients anticipate the results of their chest x-ray scans. It can predict five diseases, including Covid-19, fibrosis, bacterial or viral pneumonia, and tuberculosis. Patients can also interact with our chatbot, which will ask them questions to identify their health condition. If the patient's condition is severe, the bot will advise them to consult with a doctor available on the system. Patients can purchase coins to use the service, and a VIP level is available that reduces the number of coins required for predictions or consultations with doctors. Additionally, patients can download prescriptions that have been written by our doctors.

Within the system, doctors are invaluable and irreplaceable as they offer some of the most significant benefits. Doctors can select patients who are waiting for medical assistance and review the chat history between the patient and our chatbot to understand the patient's health condition. Based on this information, the doctor can prescribe medications for the patient's condition and save it in the system. Additionally, doctors can report patients who violate the system's rules and provide a reason for the report. The patient may then be temporarily suspended from the system until an administrator confirms the suspension as permanent or revokes it. At the end of each chat, the doctor will receive coins that can be exchanged for real money.

We have also included features in the system specifically designed to assist radiologists in predicting patients' chest X-ray scans, ensuring that radiology centers can benefit from our technology.

B. User Characteristics

Patients who want to use this system do not require any special experience or skills. The system is designed to be user-friendly and accessible to everyone. Patients should be able to navigate the system easily. Patients who have experience using computers will find it easier to use the system, but it is not a requirement. The only requirement is that patients have access to an internet connection can connect to the system.

Doctors who want to use this system should have basic computer skills. They should also have the necessary medical training and expertise to review patient chat history and prescribe medications based on their medical conditions. Additionally, they should understand the system's rules and regulations and be able to report patients who violate them. They should be familiar with the system's interface and able to manage patient consultations efficiently. Finally, they should be comfortable exchanging the coins earned after each consultation for real money, which requires familiarity with financial transactions and security measures.

Radiologists who want to use this system should have basic computer skills. They should be able to use the prediction model provided by the system to review and interpret multiple scans simultaneously. Additionally, they should be able to use the features included in the system to assist them in analyzing and interpreting the scans accurately and efficiently.

3.2 System Analysis & Design

3.2.1 Use Case Diagram

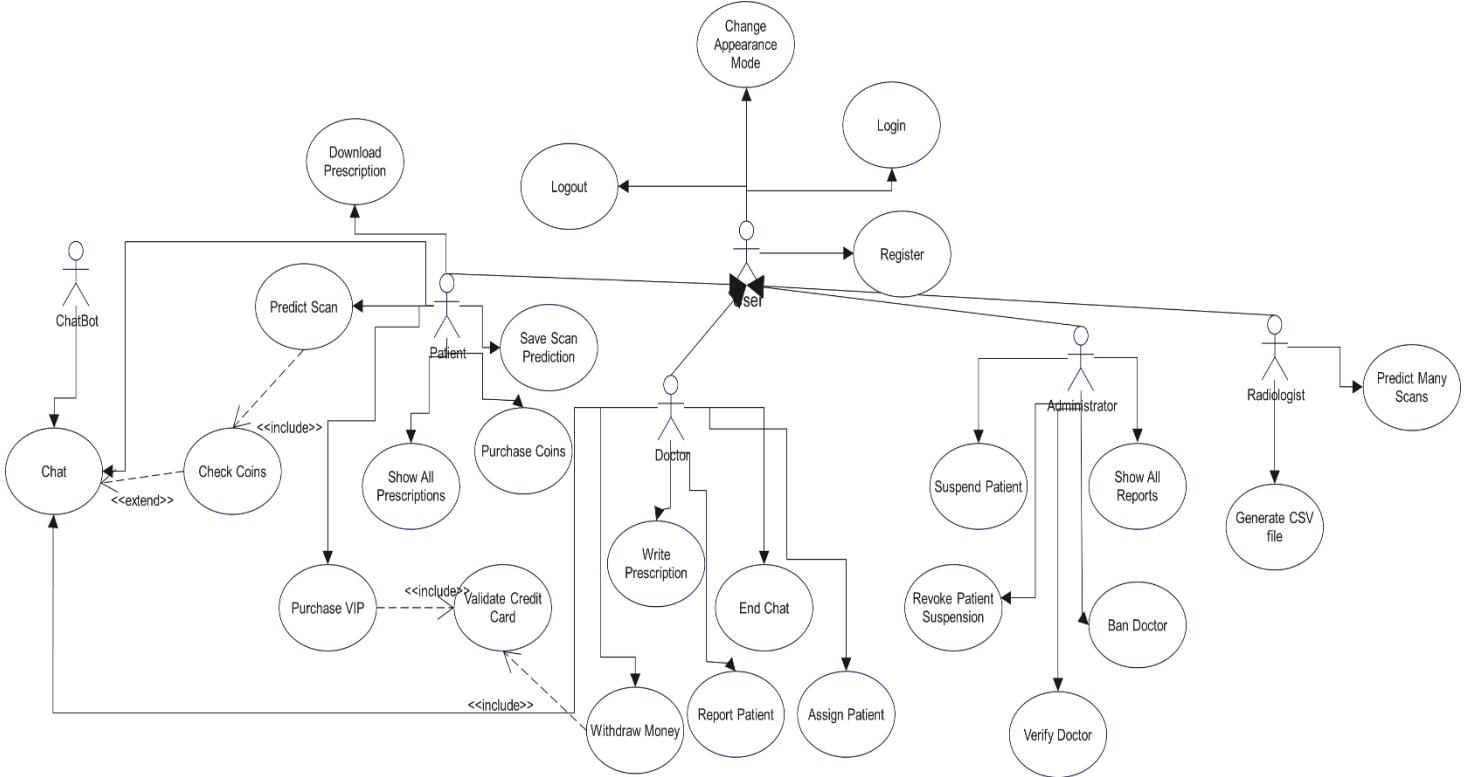


Figure 3.2 System Use Case

Patients can use our system to predict their chest X-ray scans using our prediction model. In addition, they can chat with our chat bot, which will help identify their health condition and provide advice to chat with a doctor available on the system. Patients can purchase Coins to use the system and access doctor consultations. They can also purchase a VIP level that reduces the number of Coins required for predicting their scan or chatting with a doctor. Patients can download prescriptions that were written by doctors on the system for their convenience.

Doctors can select patients who are waiting for assistance with their health condition. The doctor can review the chat history between the patient and our chat bot to better understand the patient's symptoms before making a prescription for the appropriate medication and saving it for the patient. Additionally, doctors can report patients who violate the system's rules and provide a reason for the report. If the patient is found to have violated the rules, their account will be temporarily suspended until an administrator confirms whether the suspension should be permanent or revoked. After each consultation, the doctor will receive Coins, which can be exchanged for real money, as compensation for their time.

Our system allows radiologists to use our prediction model to simultaneously predict multiple chest X-ray scans, making their work more efficient.

3.2.2 Class Diagram

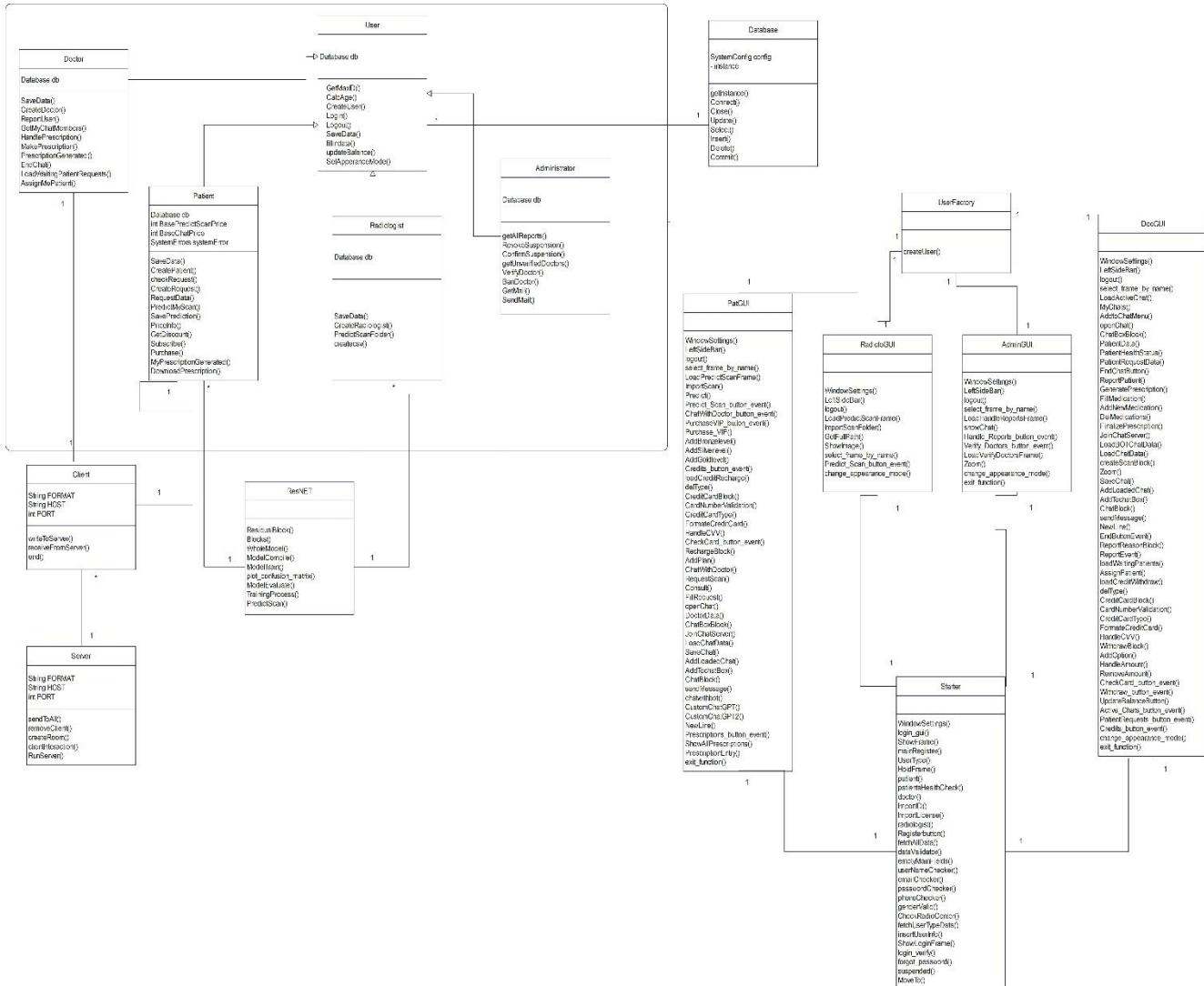


Figure 3.3 System Class Diagram

Table 3.1 Description of System Classes

Class	Description
User	It is the parent class for other user classes and contains the primary functionality for users
Administrator	This is the class that encompasses all the functionalities that an administrator can perform, such as reviewing reports, confirming the suspension of patients, revoking a patient's suspension, verifying doctors, and banning doctors.
Doctor	This class includes the functionalities that doctors can perform within the system, such as creating prescriptions, chatting with patients, reporting patients, and ending a chat with a patient.
Patient	This class comprises the functionalities that patients can perform within the system, such as chatting with our chatbot, consulting with a doctor, predicting an X-ray scan, purchasing a VIP subscription, buying coins, and downloading prescriptions written by doctors.
Database	This class involves the connection with our database to retrieve and store data.
Client	This class involves the client that connects to the server to initiate chat.
Server	This is the class that waits for client requests to connect to it, create a room, and engage in a chat with other users within the system.
ResNET	This is the class that contains the functions for creating our customized ResNet model and saving the model for future use in predictions.
Starter	This class contains the GUI for registering and logging in to our system.
RadioloGUI	This class contains the GUI for Radiologist to use our system.
PatGUI	This class contains the GUI for Patient to use our system.
DocGUI	This class contains the GUI for Doctor to use our system.
AdminGUI	This class contains the GUI for Administrator to use our system.

3.2.3 Sequence Diagram

Patient Sequence Diagrams

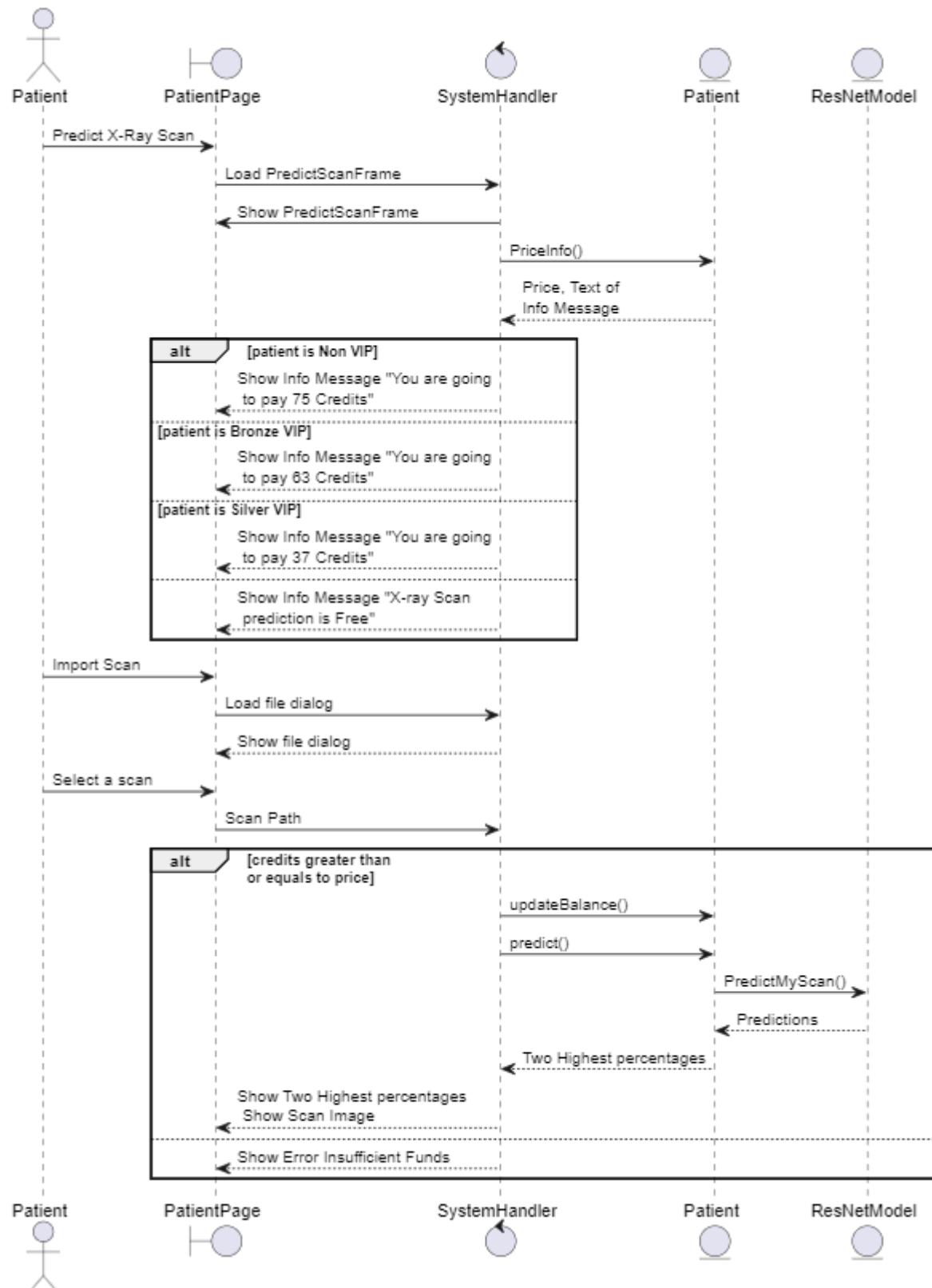


Figure 3.4 Predict Scan sequence diagram.

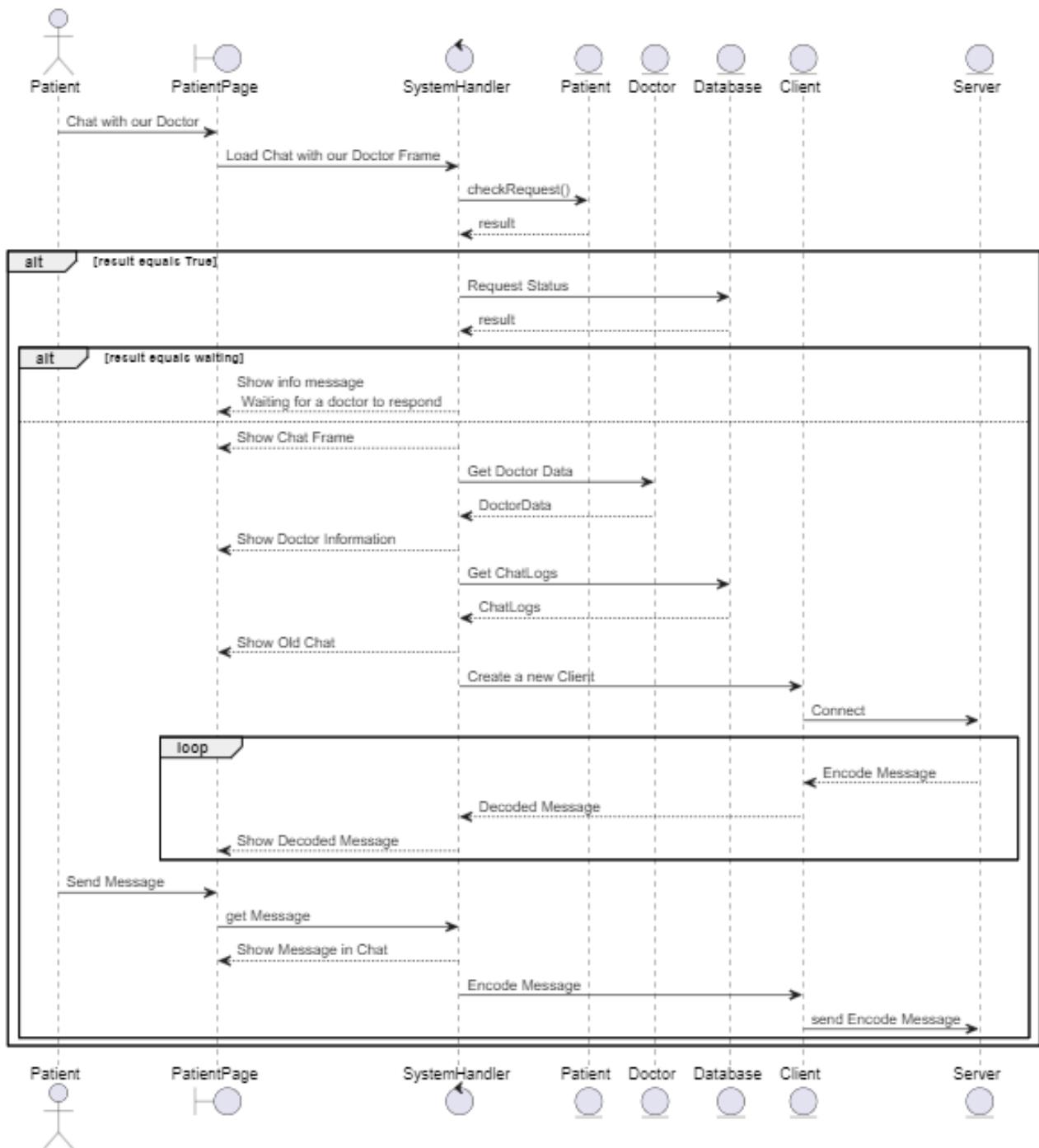


Figure 3.5 Chat with doctor sequence diagram

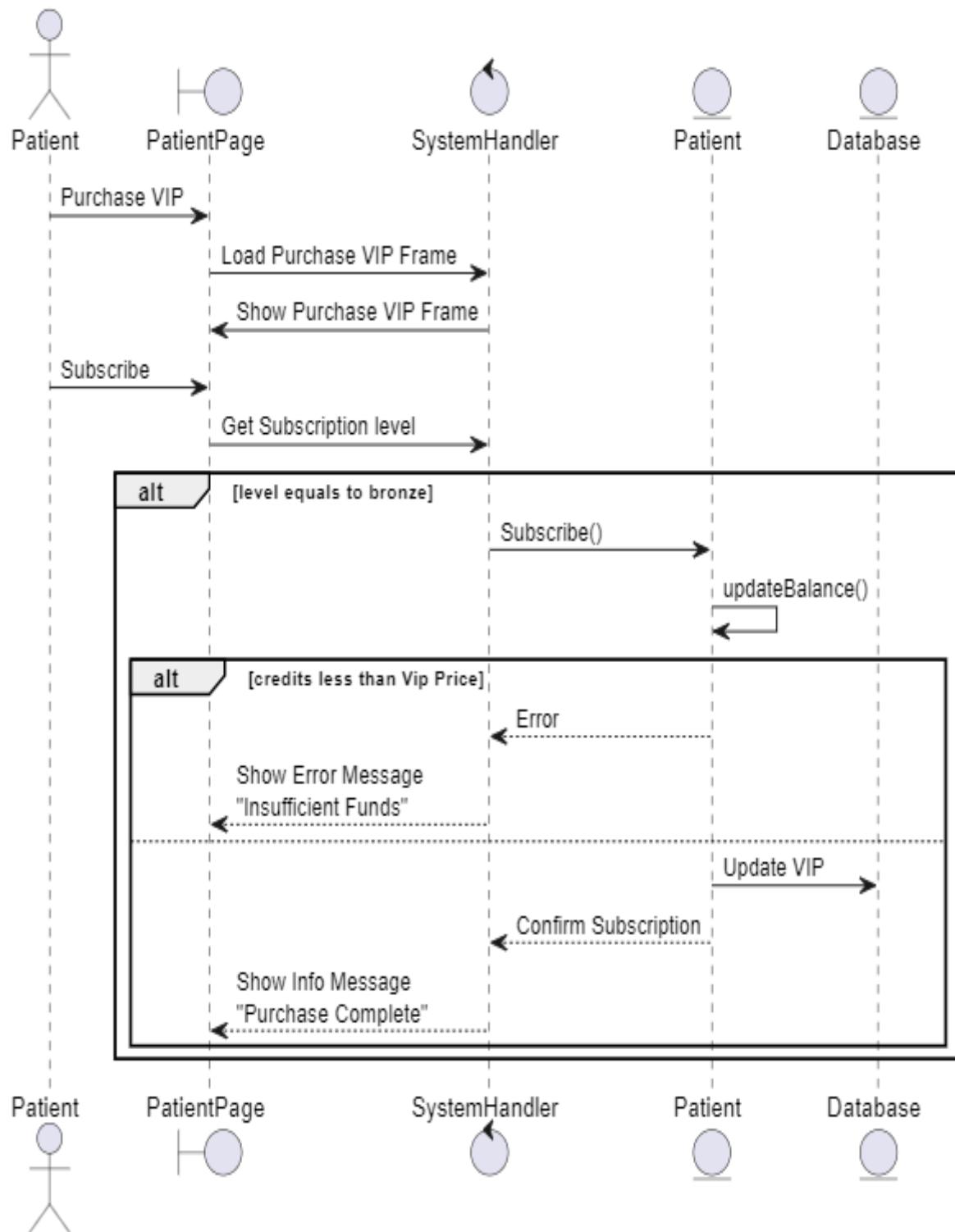


Figure 3.6 Purchase VIP sequence diagram

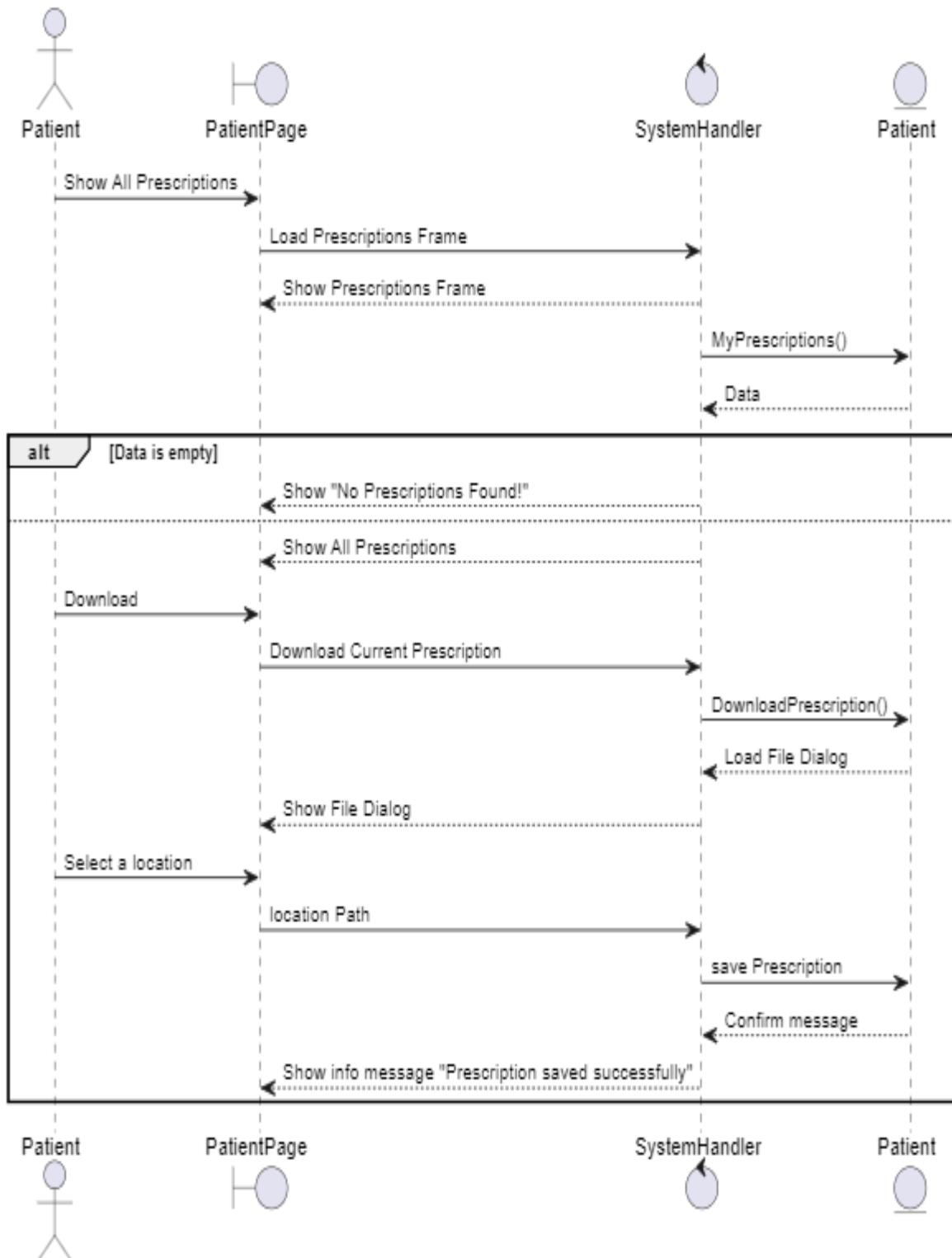


Figure 3.7 Show All Prescriptions sequence diagram

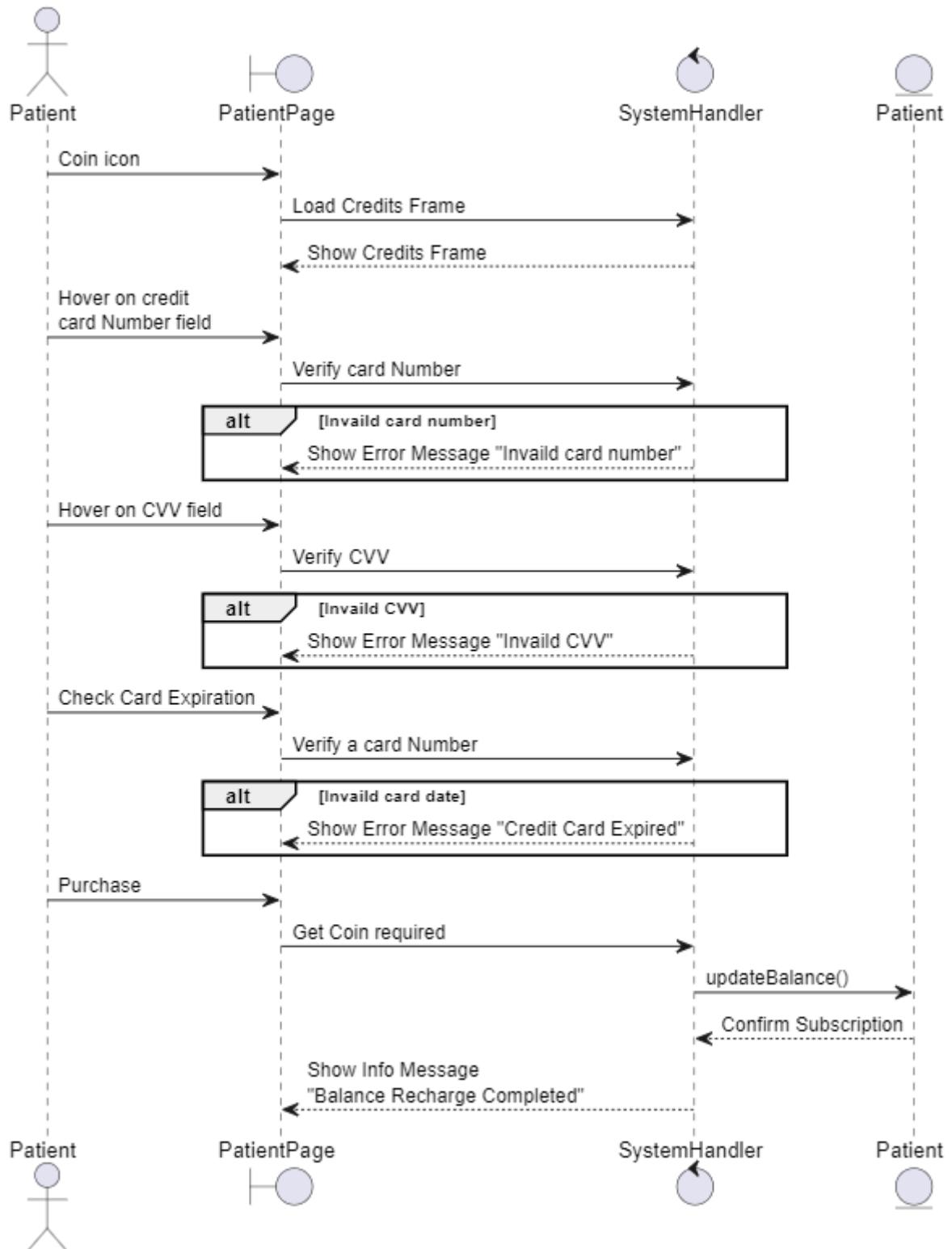


Figure 3.8 Purchase Coins sequence diagram

Doctor Sequence Diagrams

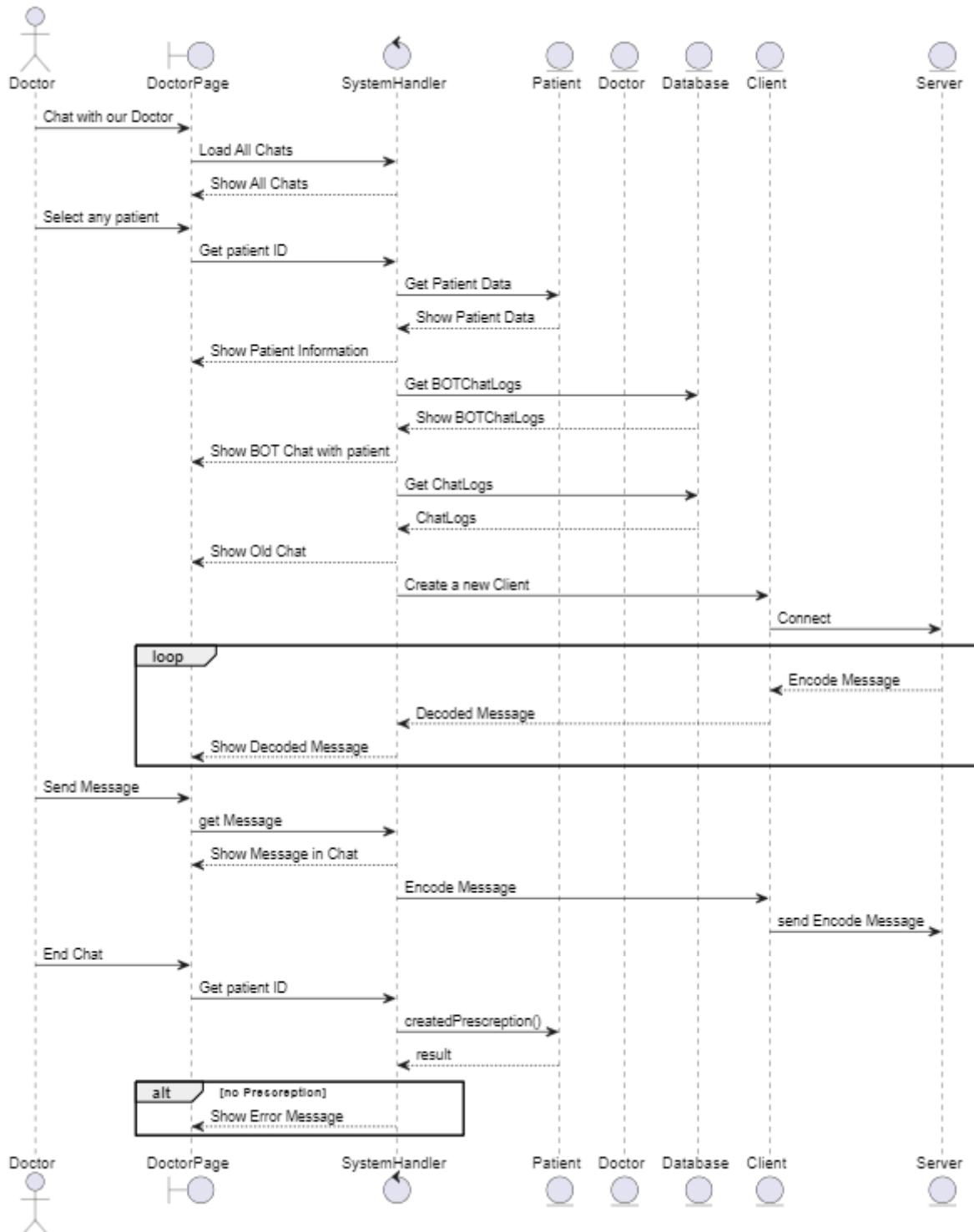


Figure 3.9 Chat with patient sequence diagram

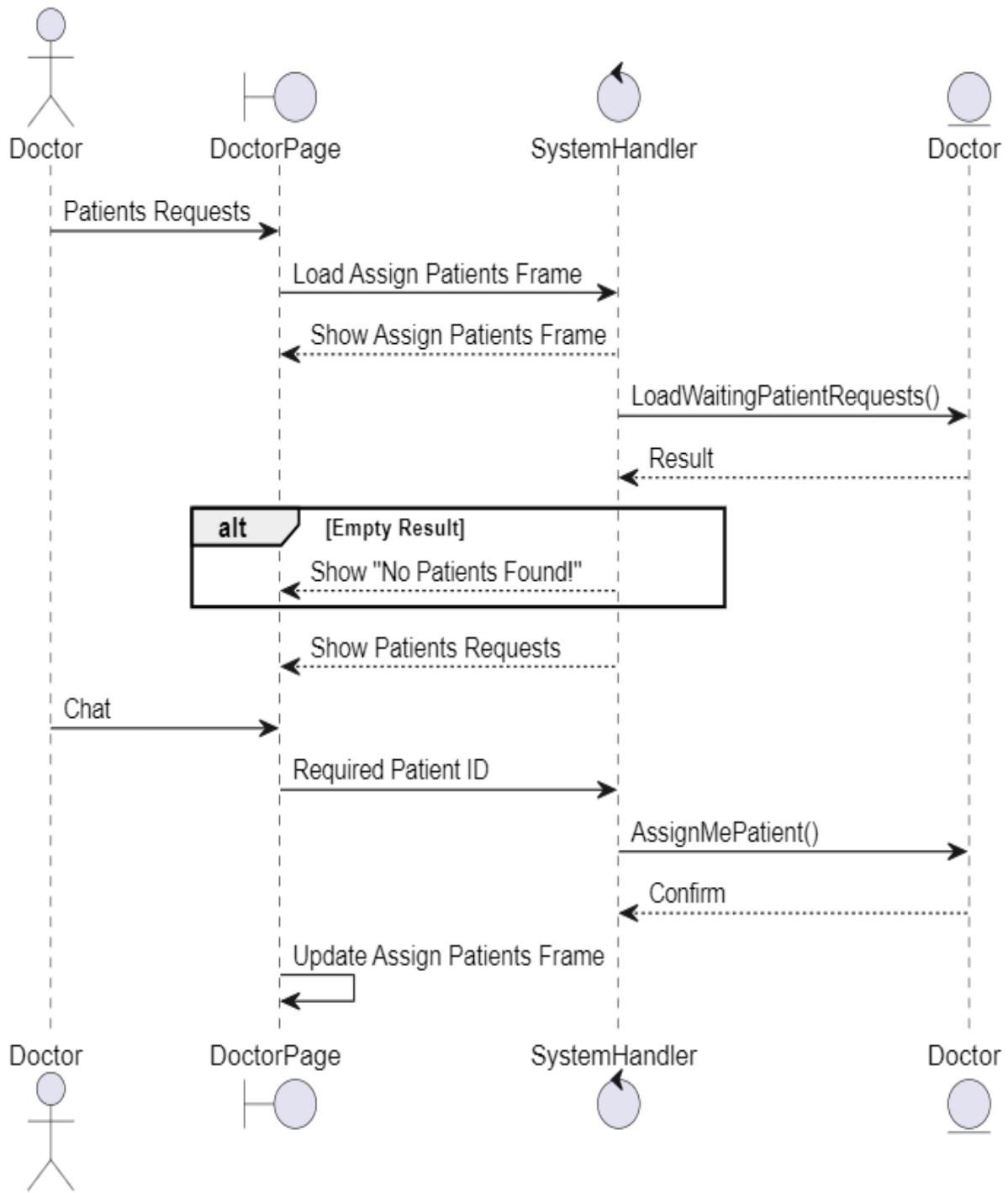


Figure 3.10 Assign Patient sequence diagram

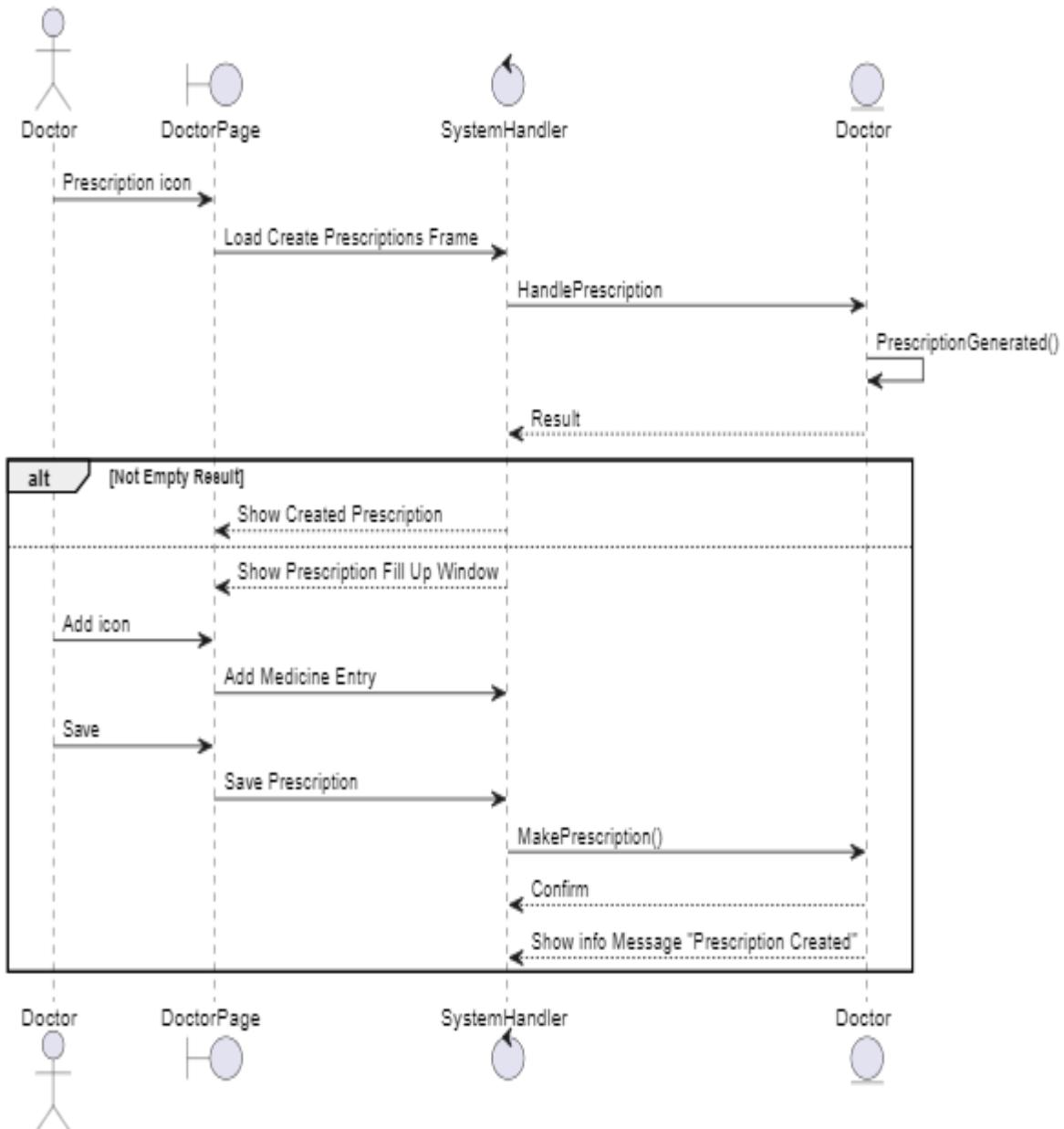


Figure 3.11 Create Prescription sequence diagram

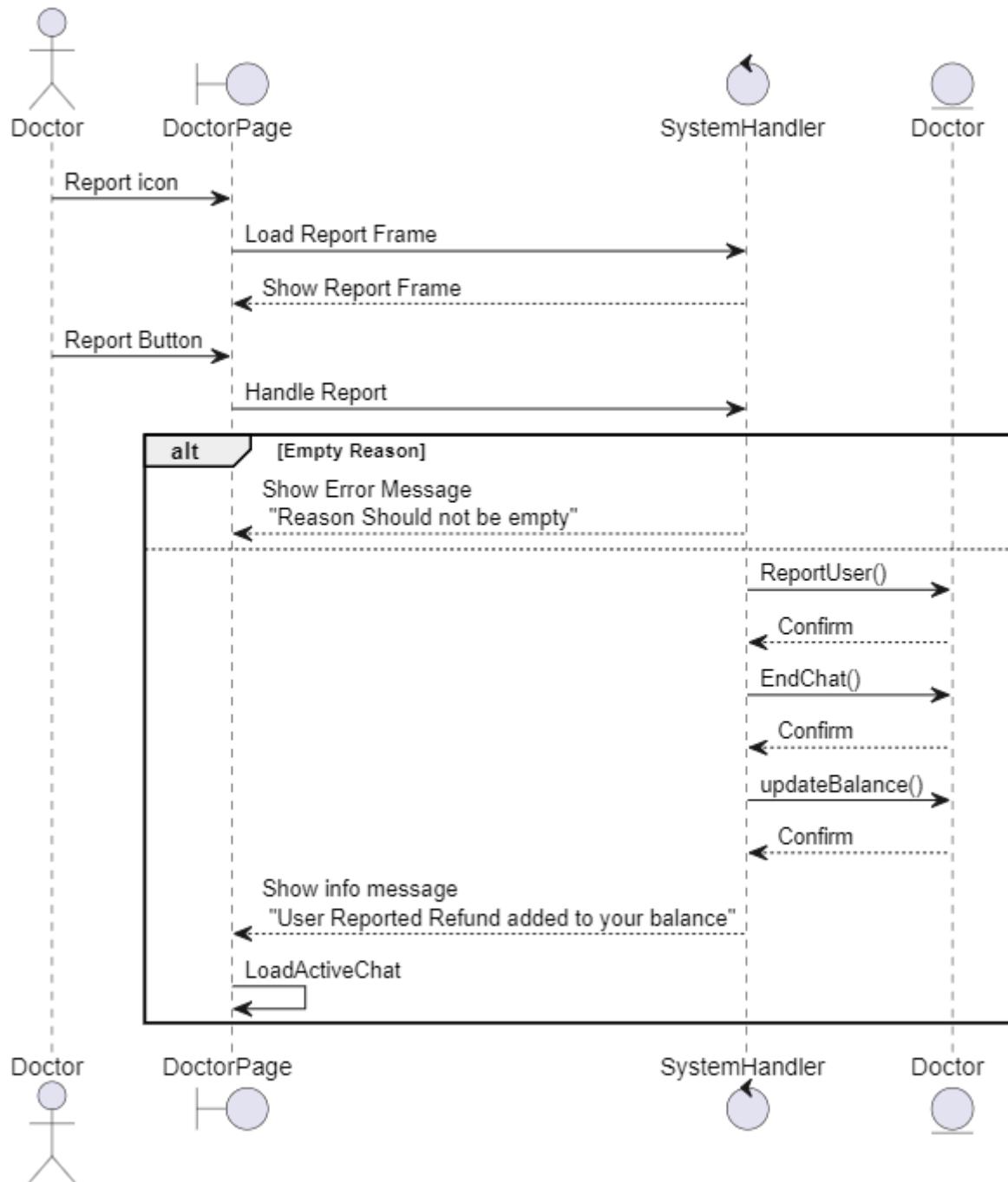


Figure 3.12 Report Patient sequence diagram

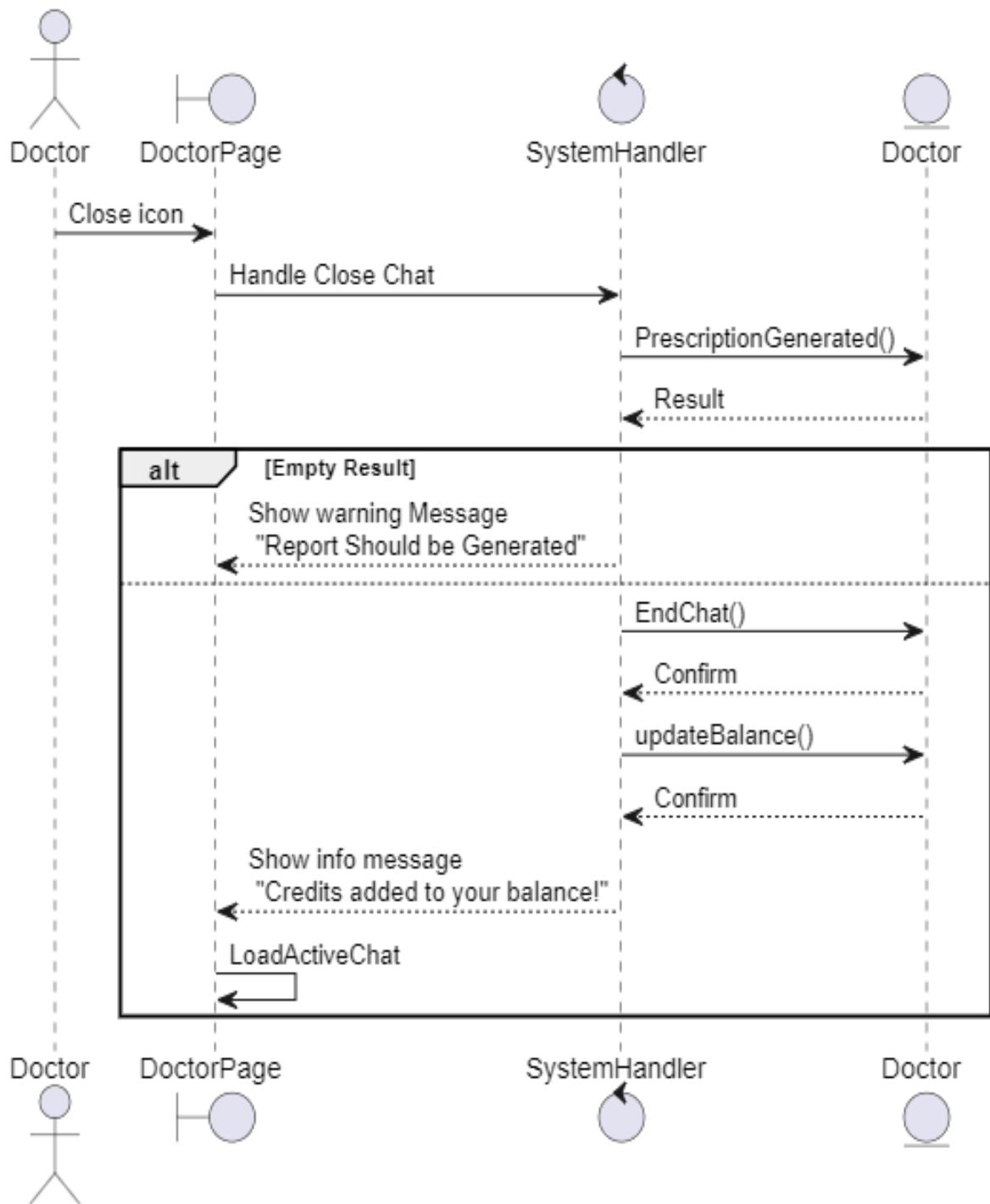


Figure 3.13 Close Chat sequence diagram

Radiologist Sequence Diagrams

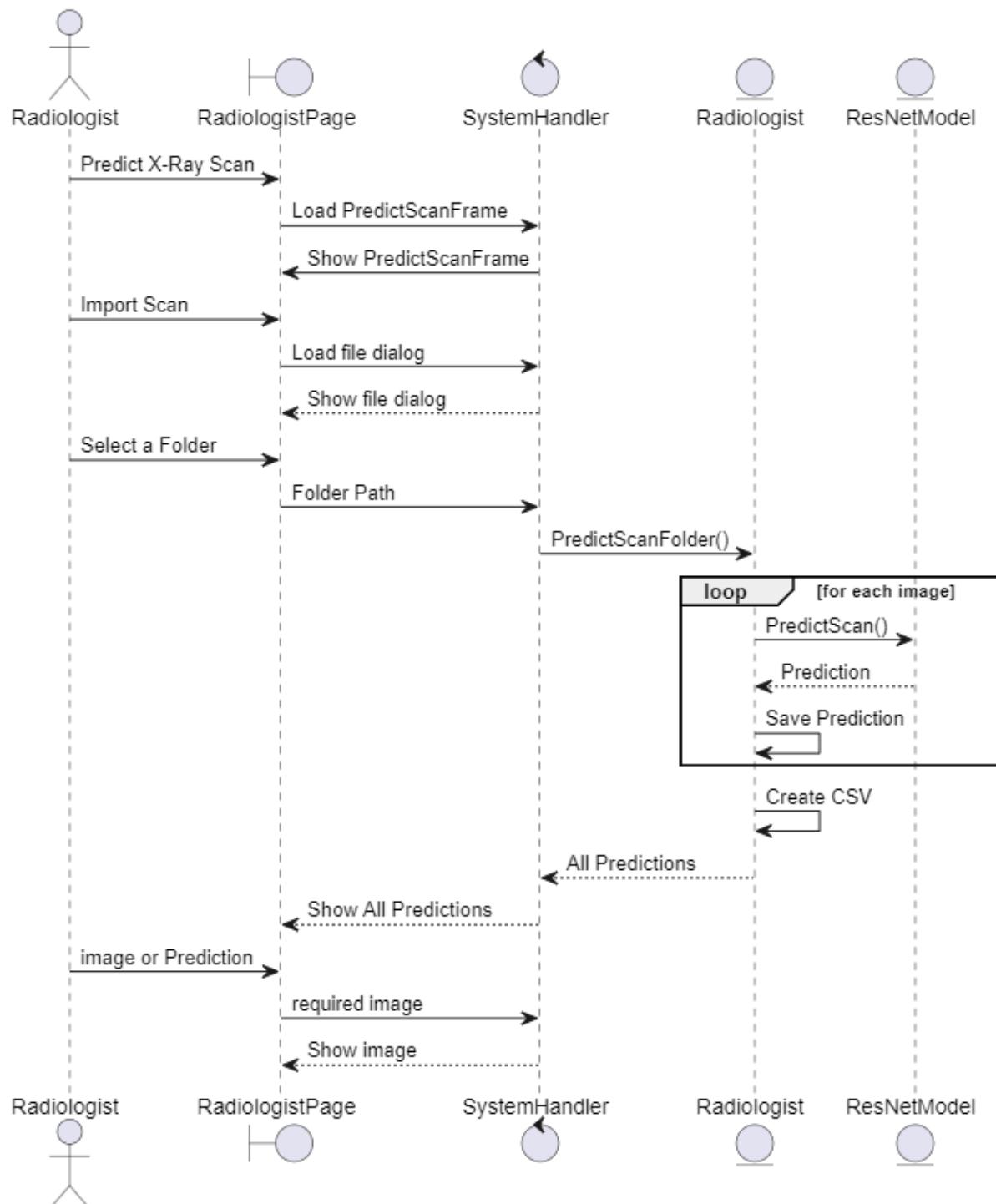


Figure 3.14 Predict Multiple X-ray Scan sequence diagram

Administrator Sequence Diagrams

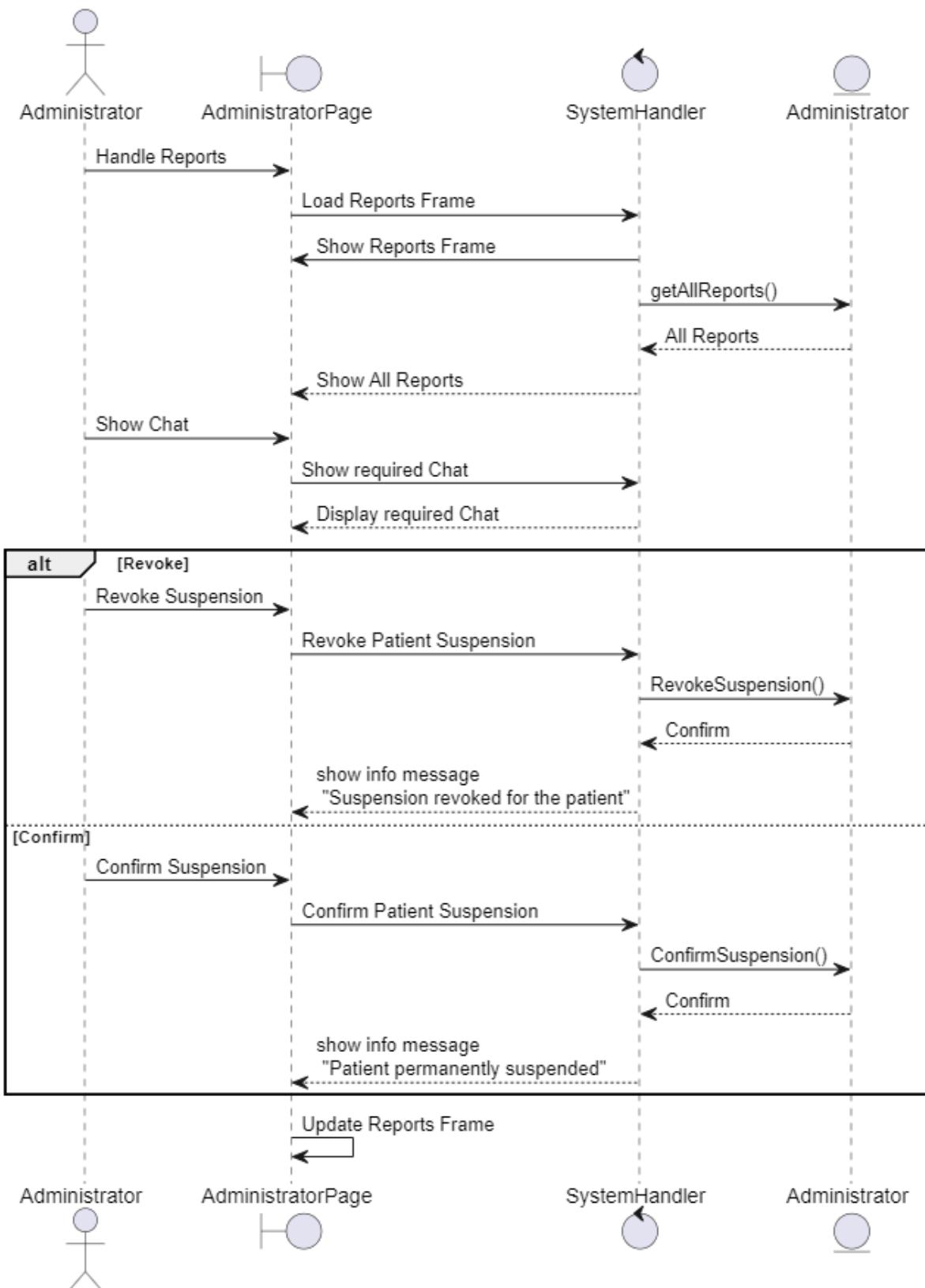
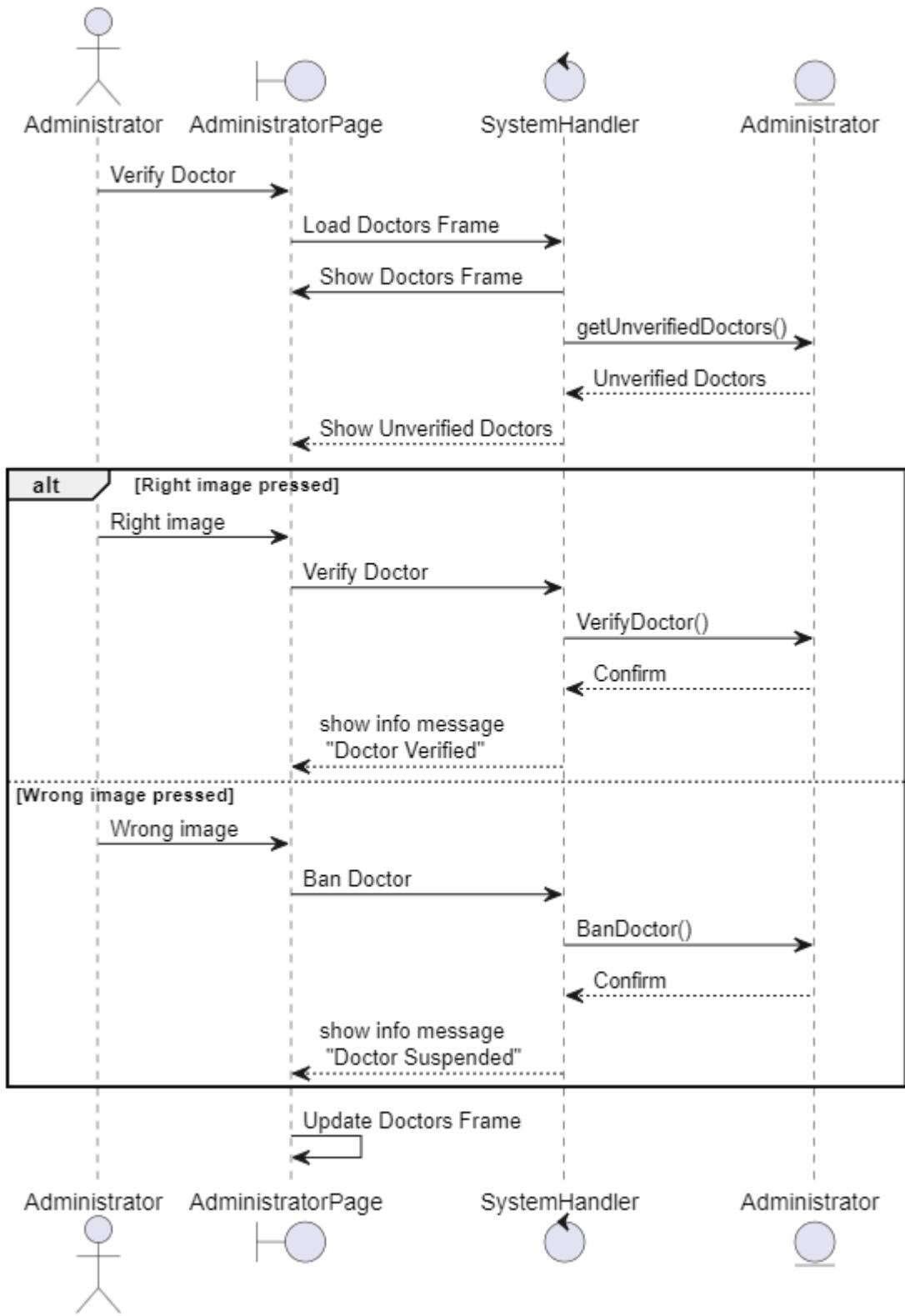


Figure 3.15 Handle Reports sequence diagram



3.16 Verify Doctor sequence diagram

Figure

3.2.4 Database Diagram

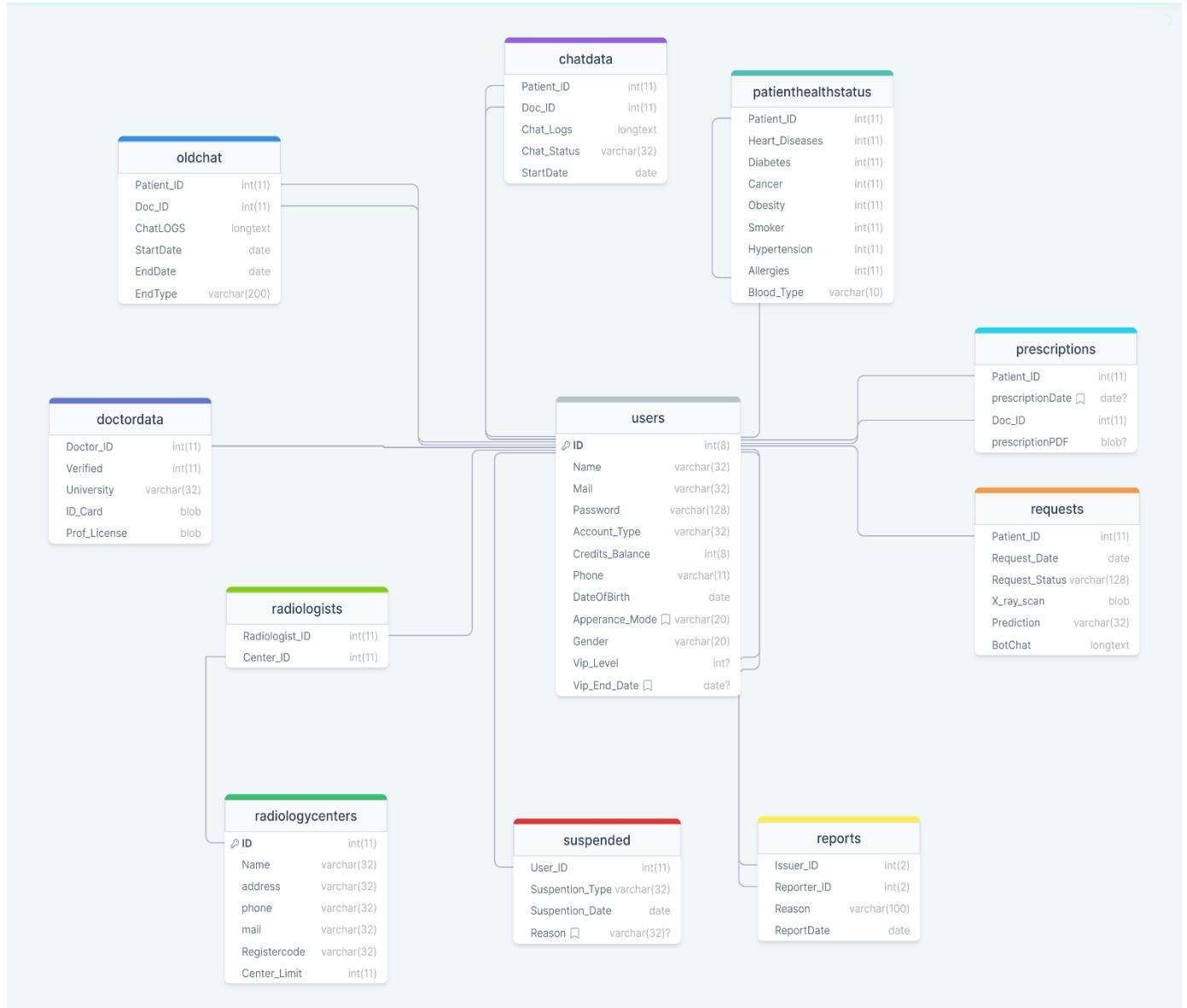


Figure 3.17 Database Diagram

Table 3.2 Database Table Description

Table	Description
users	This table contains all the user data, such as their ID, name, email, password, user type, coin balance, phone number, date of birth, system appearance mode, gender, VIP level, and VIP end date.
doctordata	This table will contain the doctor's ID, university, ID card image, professional license, and verification status.
patienthealthstatus	This table will contain all patient health status information, such as heart diseases, diabetes, cancer, obesity, smoking habits, hypertension, allergies, and blood type.
radiologycenters	This table will contain all the data of radiology centers that use our system, including their name, address, phone number, email, registration limit within the system, and registration code.
Radiologists	This table will contain data for each radiologist and the center in which they work.
Suspended	This table will contain data for users who have been suspended from using our system, including the type of suspension, date, and reason for suspension.
Reports	This table will contain data for the reports created by doctors for patients, including the reason for the report and the date it was created.
Request	This table will contain data for when a patient requests to chat with a doctor, including the request status, request date, X-ray scan image (if applicable), prediction results for the image (if applicable), and chat logs with the chatbot.
Prescriptions	This table will contain data for the prescriptions created for each patient, including the PDF file for the prescription and the date it was created.
Chatdata	This table will contain data for the chat between a patient and a doctor, including the chat status and the start date of the chat.
Oldchat	This table will contain logs for previous chats between a patient and a doctor, including the start and end dates of the chat and the reason for ending the chat

4.Implementation and Testing

In this section, we are discussing the primary functions of our system, including their descriptions and the algorithms, techniques, and design patterns used to implement them. Prior to delving into the technical details, we aim to develop initial designs for the system's user interface by crafting UI designs for every page of each user. Our goal is to ensure that the interface is not only visually appealing but also user-friendly, allowing users to interact with the system seamlessly without encountering any issues.

UI Designs:

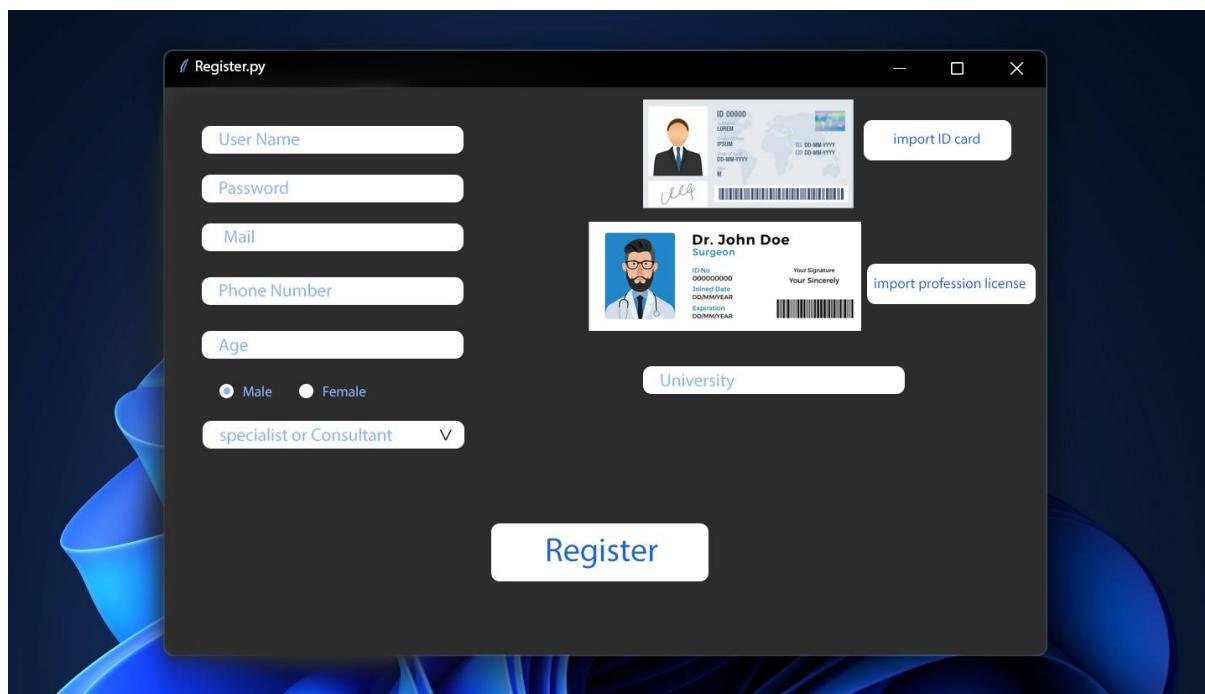


Figure 4.1 Doctor Register Page

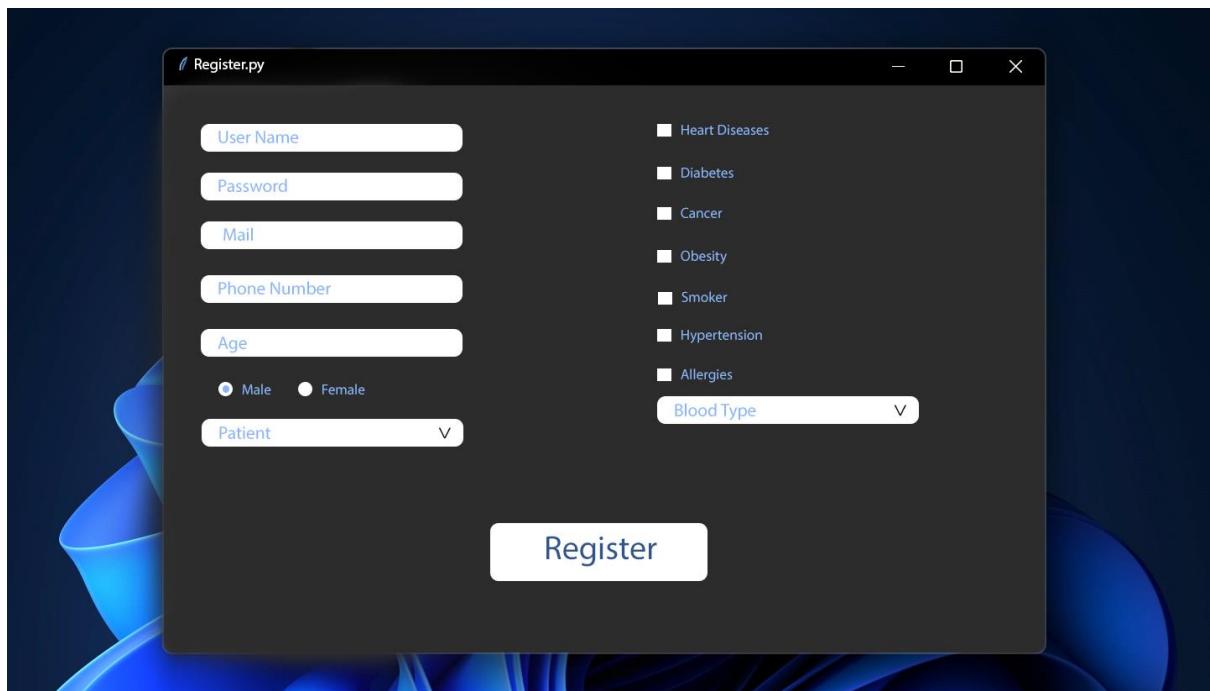


Figure 4.2 Patient Register Page

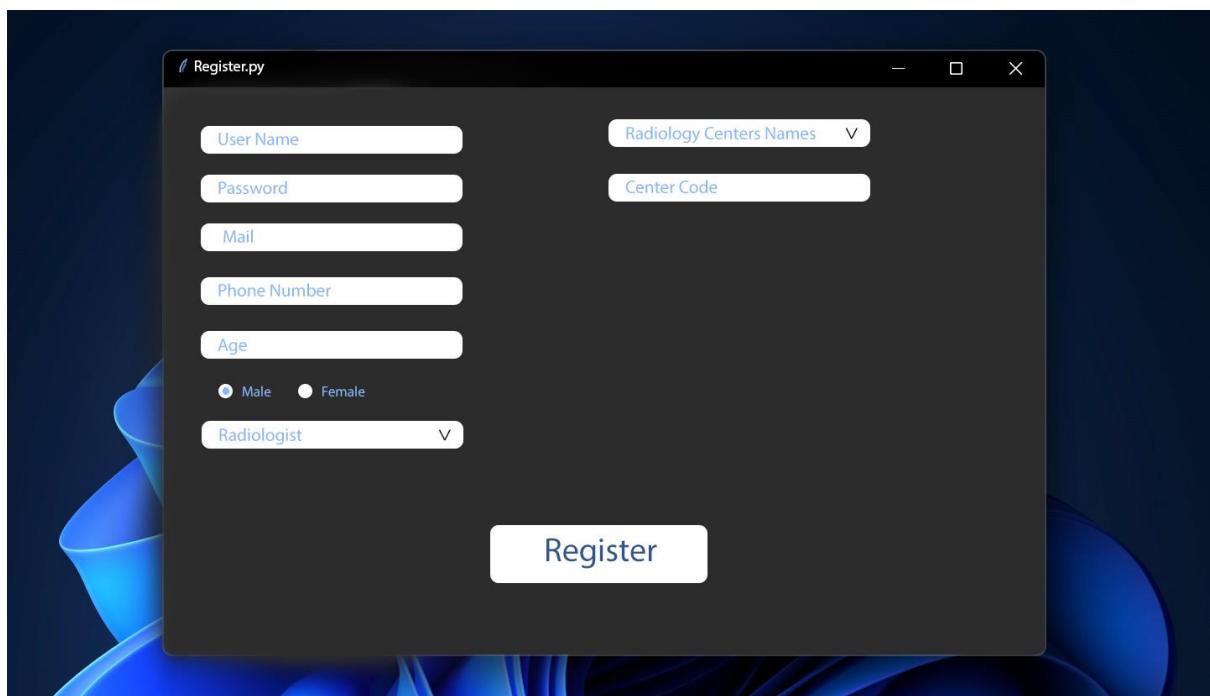


Figure 4.3 Radiology Register Page

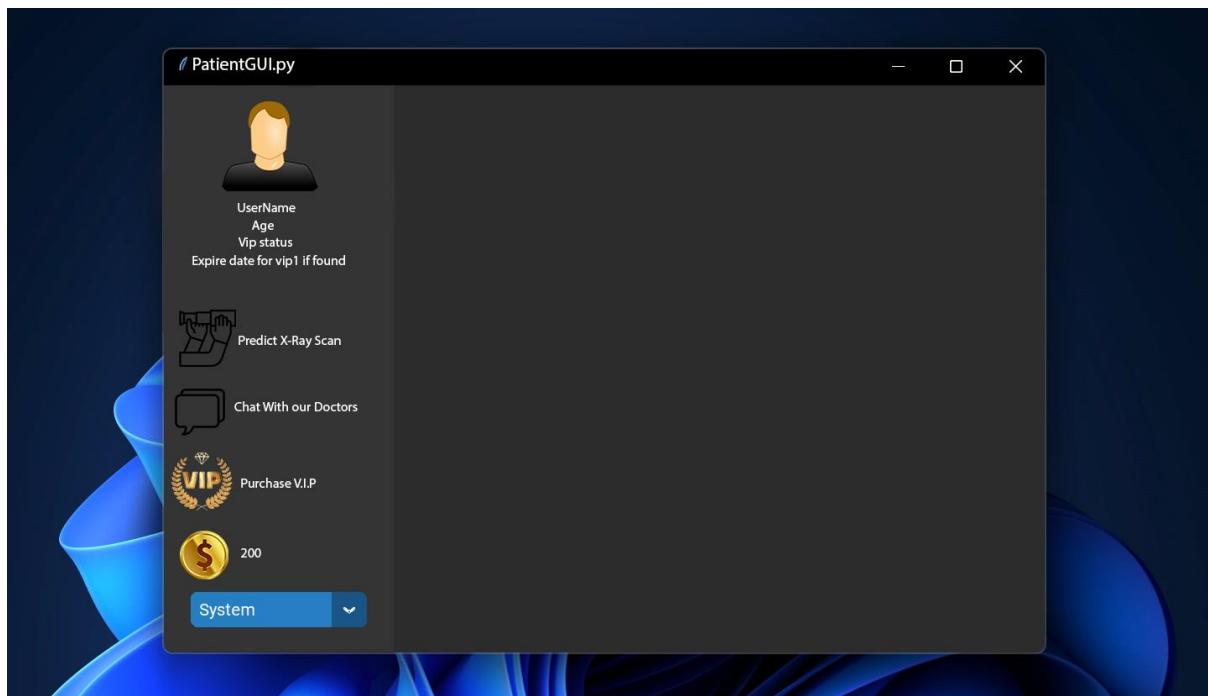


Figure 4.4 Patient Main Page

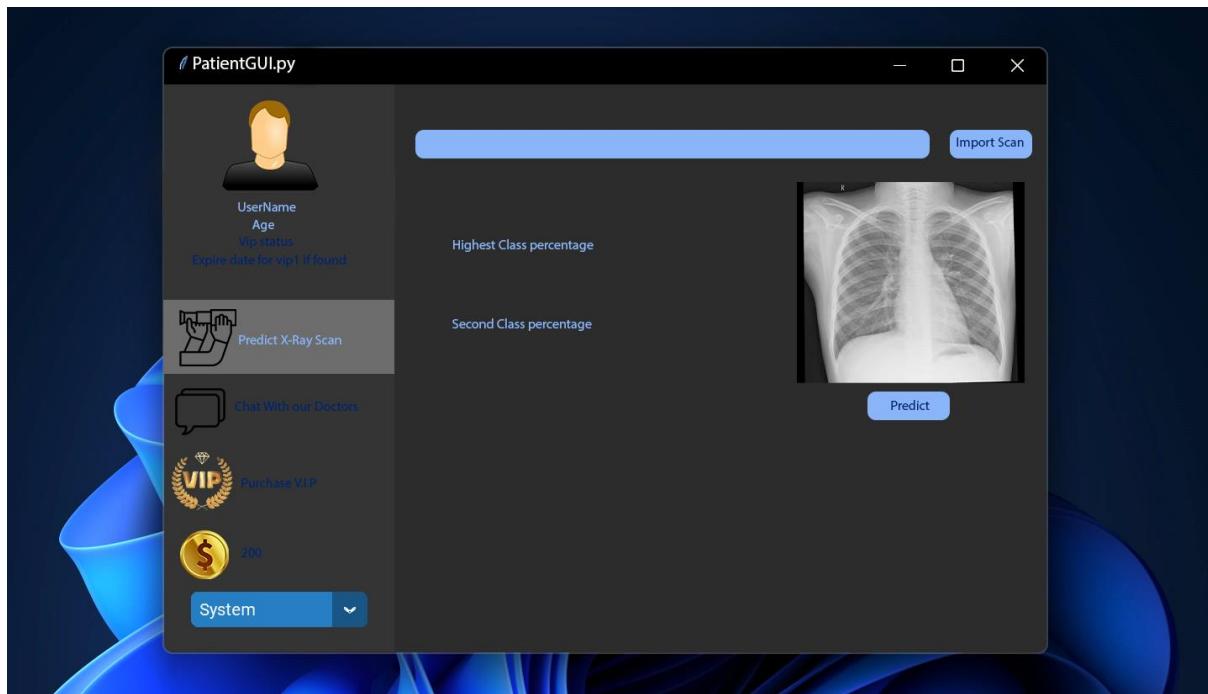


Figure 4.5 Predict X-ray Scan Patient Feature

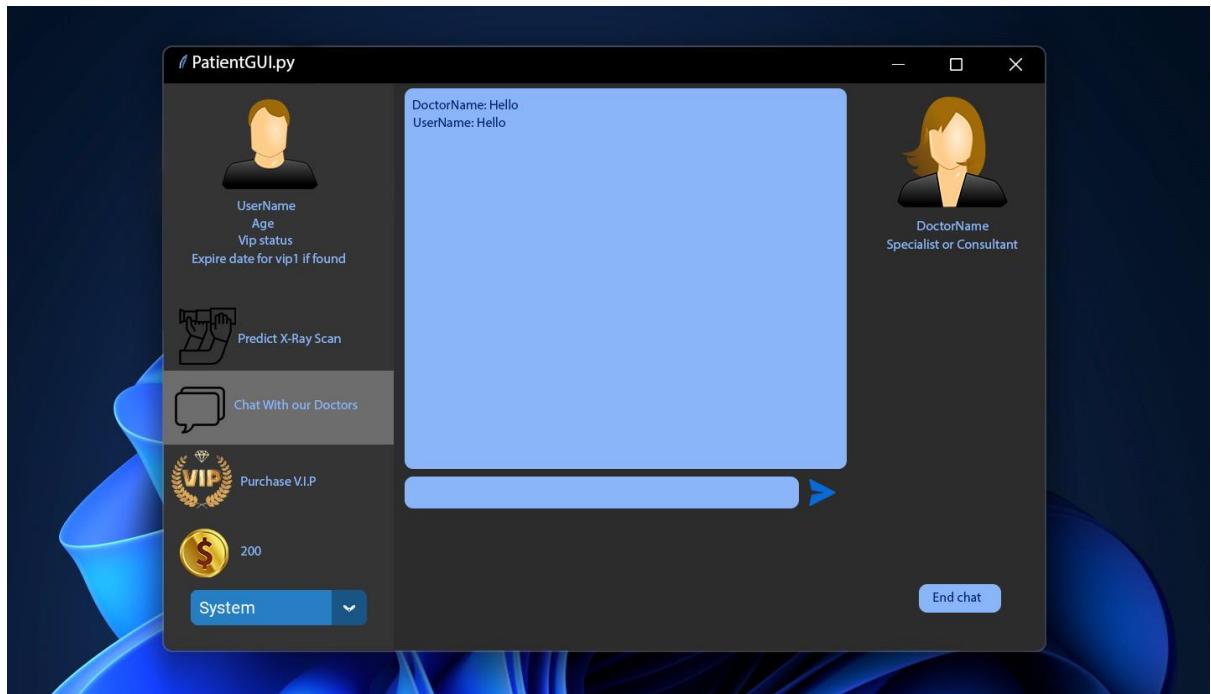


Figure 4.6 Chat with Doctor Patient Feature

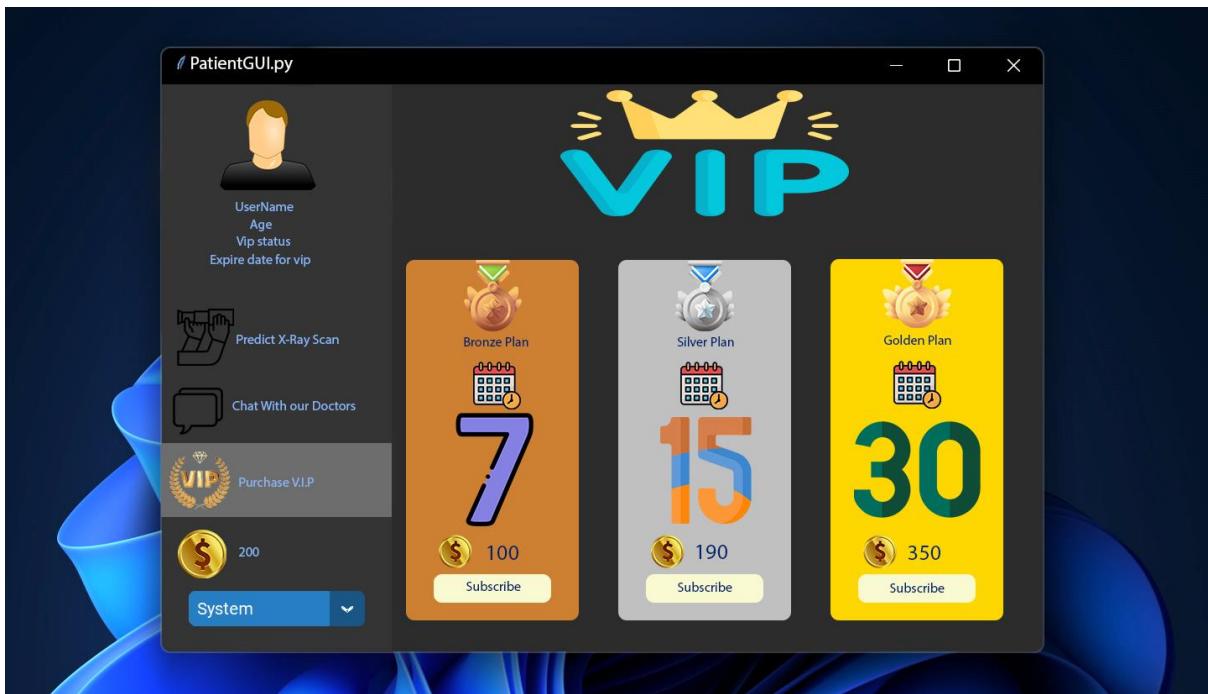


Figure 4.7 Purchase VIP Patient Feature

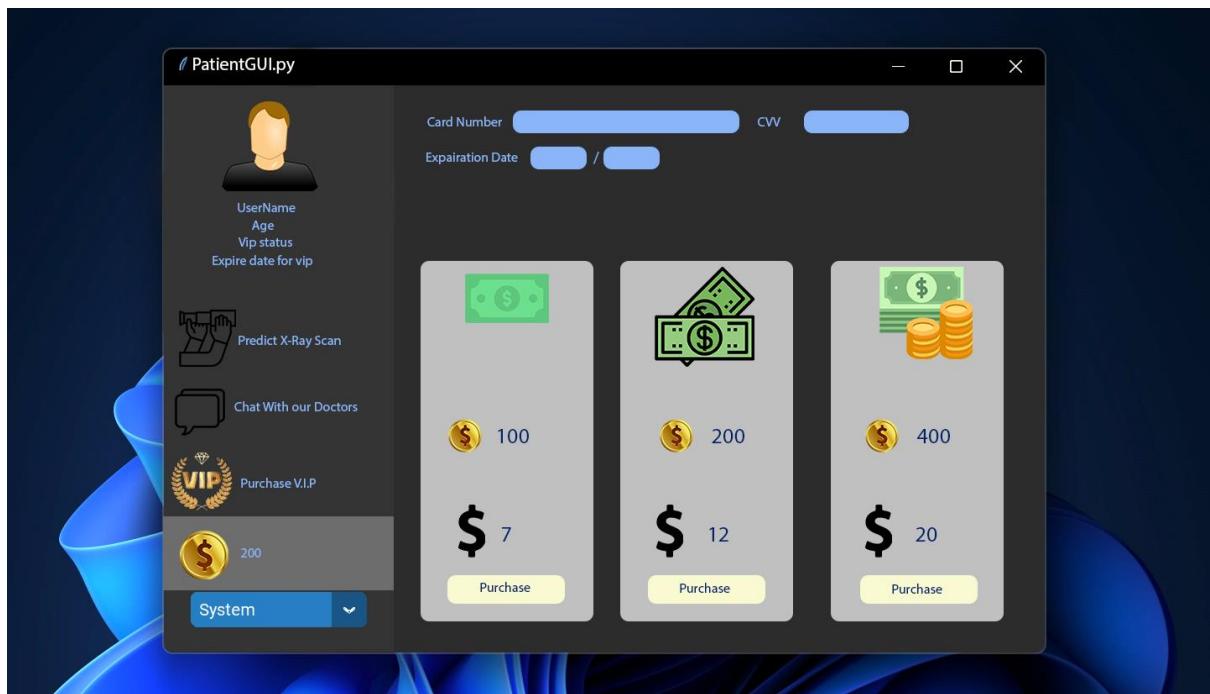


Figure 4.8 Purchase Coins Patient Feature

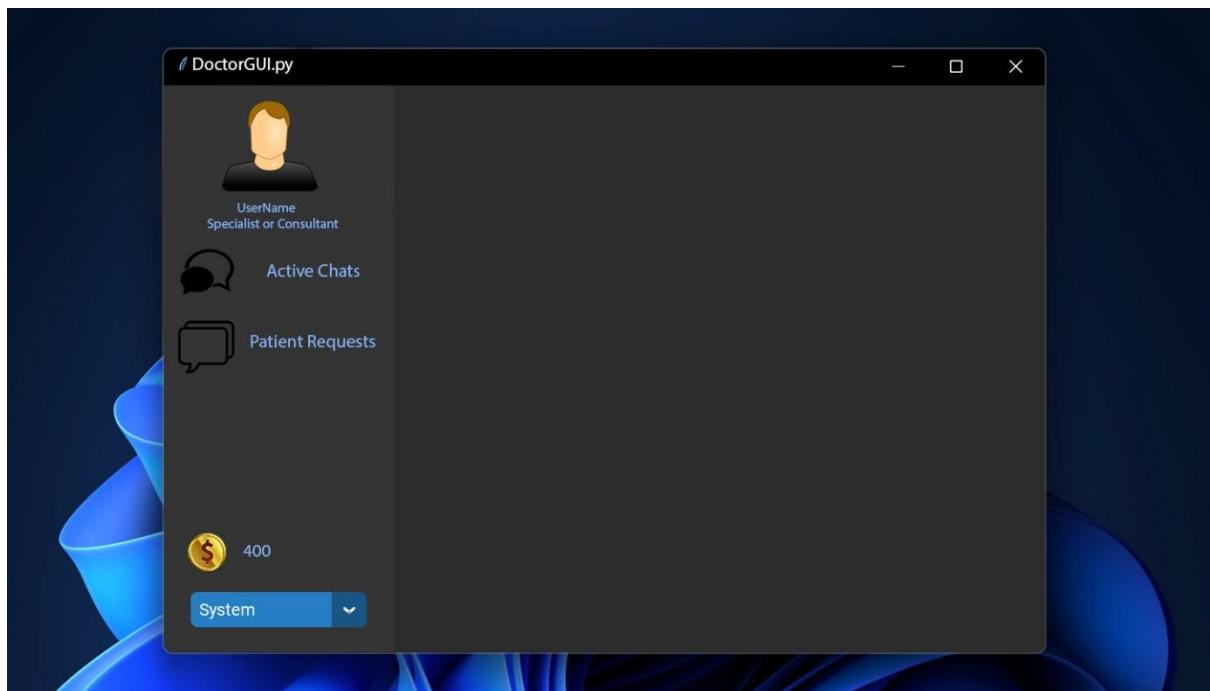


Figure 4.9 Doctor Main Page

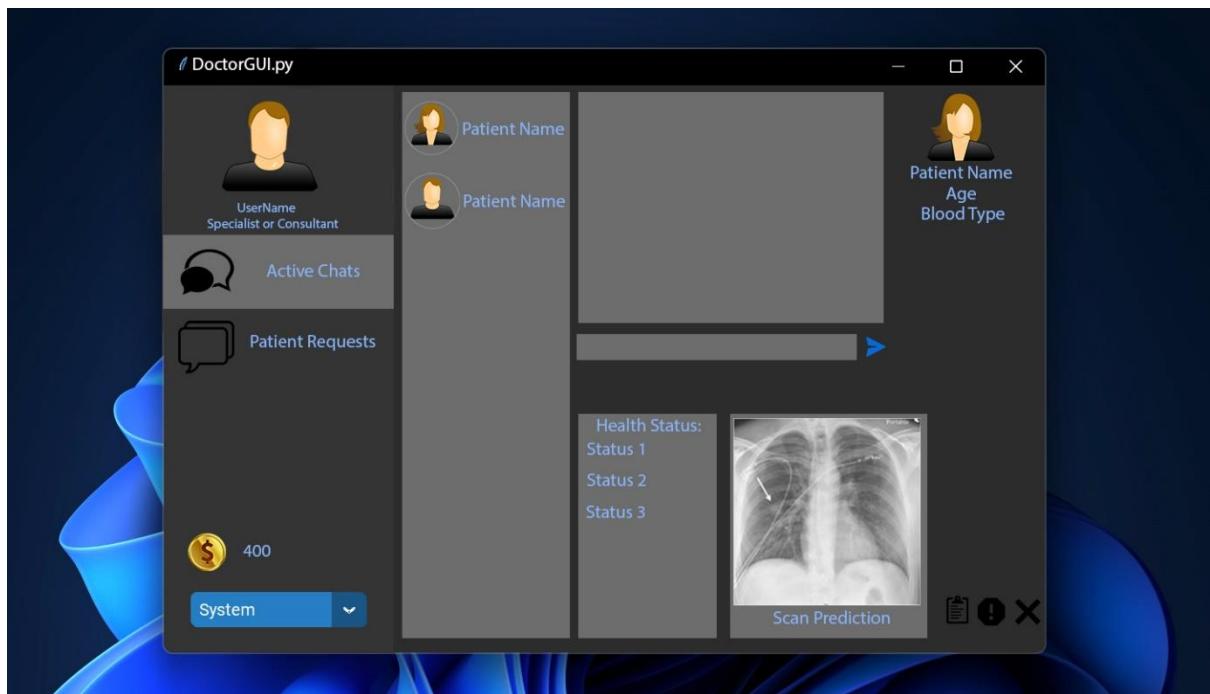


Figure 4.10 Chat with Patient Doctor Feature

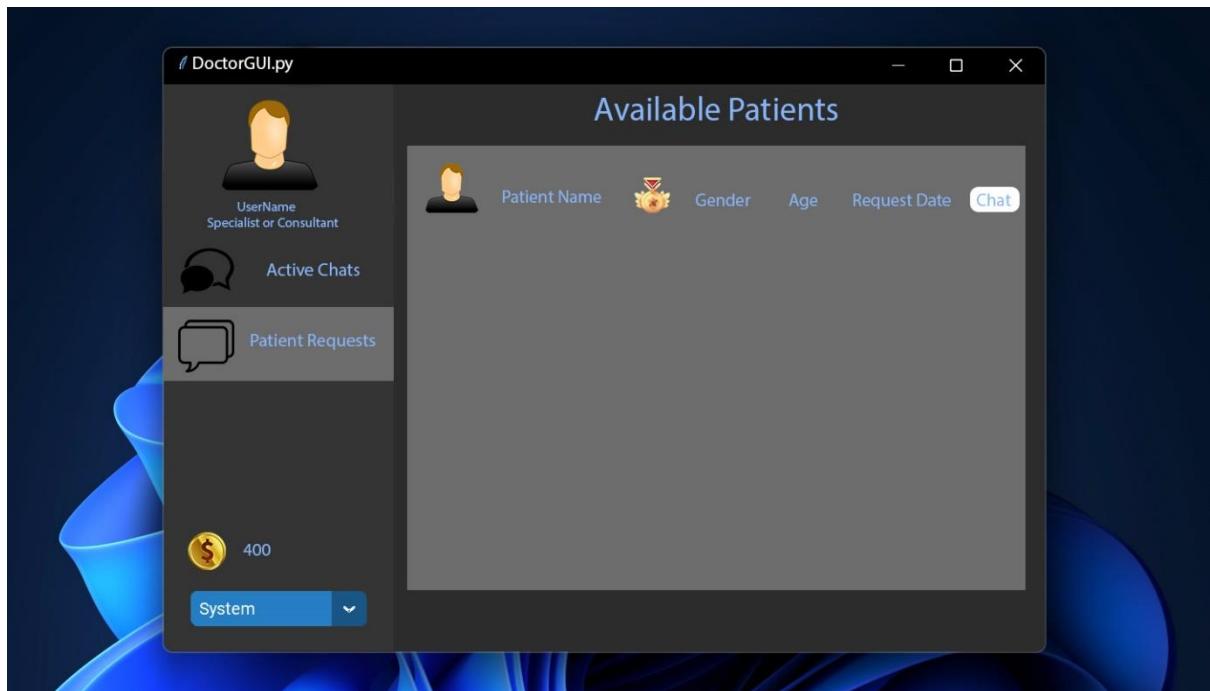


Figure 4.11 Patients Request

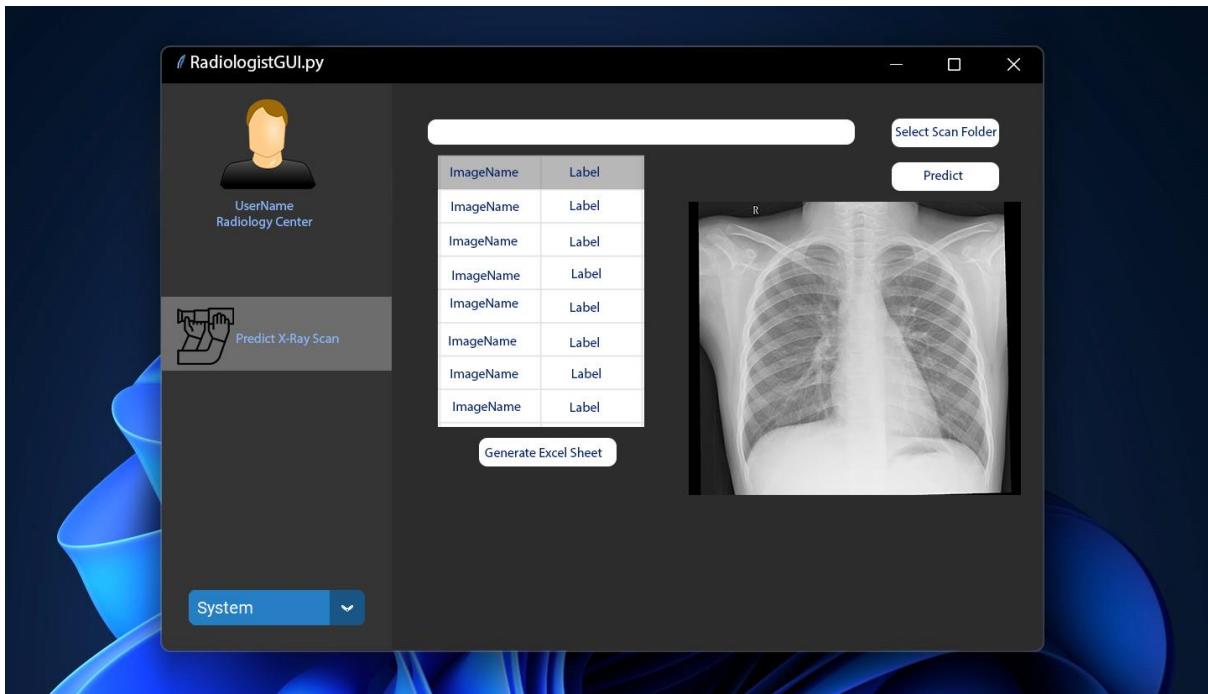


Figure 4.12 Radiologist X-ray Prediction Feature

These UI designs were our preliminary versions that aided us in formulating the fundamental theme of the system. However, most of these designs were further refined to make the system more user-friendly, ensuring seamless usage without any obstacles. Additionally, the system's color scheme was applied to the overall theme, providing a cohesive and visually appealing experience for the users.

At this juncture, we elucidate all the system functions that have been employed to implement these UI designs into an operational system, which is now fully functional and ready for user utilization. Our team has meticulously examined and tested each function, ensuring optimal performance and error-free operation. Moreover, we have taken significant measures to ensure that the user interface remains intuitive and user-friendly, allowing users to interact with the system with ease.

To achieve this, we have utilized various algorithms, design patterns, and techniques to implement each system function effectively. Our team has left no stone unturned to ensure that the system is robust, efficient, and user-friendly. With all the essential functions in place and the user interface optimized, we are confident that the system will provide a seamless and satisfying user experience, meeting the needs and expectations of all its users.

In our software application, we have implemented a class hierarchy for the different types of users who interact with the system. This includes patients, doctors, radiologists, and administrators. All of these classes have a common parent class called "User". This allows us to define common properties and methods for all types of users in one place and avoid code duplication.

To create instances of these classes, we used a design pattern called "Class factory". This pattern provides a centralized way to create objects of different classes. By using this design pattern, we can achieve a high level of code modularity and scalability. It also helps to improve code readability and maintainability by reducing complexity and avoiding repetitive code. Overall, this approach ensures that our software is flexible and can adapt to changing requirements in the future.

Table 4.1 User Class

Function Name	Description
__init__	This is our Python constructor, which takes a user ID as a parameter and retrieves their basic information from the database, such as name, email, password, account type, coin balance, phone number, age, system appearance mode, gender, VIP level, and VIP end date.
GetMaxID	This parameter less function returns the maximum ID stored in the database.
CalcAge	This Function takes Birthdate as a parameter and return age.
CreateUser	This is a class method that serves as a secondary constructor. It takes parameters such as name, email, password, account type, phone, age, and gender, and uses them to create a new user with the provided data.
Login	This is a class method that takes an email and password as parameters and returns the user ID and account type associated with that user.
Logout	This function takes the current page as a parameter, closes it, and opens the login page
SaveData	This function does not take any parameters and simply adds the newly created user to the database.
updateBalance	This function takes a value that needs to be added or removed from the coin balance. It then checks if the balance is sufficient and updates and saves the new balance to the database.
SetApperanceMode	This function takes a new system mode as input and applies it to the system. It then saves the mode to the database so that it can be used later.

Table 4.2 Patient Class

Function Name	Description
__init__	This is the constructor method in Python, which takes the user ID as a parameter. It retrieves the user's basic information from the database, such as name, email, password, account type, coin balance, phone, age, system appearance mode, gender, VIP level, VIP end date, as well as their health status data, such as heart diseases, diabetes, cancer, obesity, smoker status, hypertension, allergies, and blood type.
CreatePatient	This is a class method that serves as a secondary constructor. It takes parameters such as name, email, password, account type, phone, age, gender, heart diseases, diabetes, cancer, obesity, smoker status, hypertension, allergies, and blood type, and uses them to create a new user with the provided data.
checkRequest	This function does not take any parameters, but it returns whether the patient has created a request to chat with a doctor or not yet.
CreateRequest	This function takes X-ray scan path, prediction made by our system, and chat logs with our chatbot as parameters. It then saves this data into the requests table in our database and sets the request status to 'waiting'.
requestData	This function takes no parameters and return the request created by the patient.
PredictMyScan	This function takes a scan path and an output type as parameters. If the output type is equal to two, the model will be called to predict the X-ray scan and return the two highest predictions. If the output type is equal to one, only the highest-class prediction will be returned
SavePrediction	This function takes a scan path as a parameter. It takes that path, creates a text file, and writes the two highest classes for the prediction to the file.
PriceInfo	This function takes a context type as a parameter and checks if the context type is 'chat'. If so, it checks the VIP level of the patient to determine the price they shall pay for the chat. The same process occurs if the context type is 'predict'.
GetDiscount	This function takes a value and a percentage as parameters to calculate the discount for the patient.
Purchase	This function takes the amount of coins that the patient wants to buy as a parameter. It then checks if the credit card is valid, and if so, adds the coins to his balance.

Subscribe	This function takes the VIP level that the patient wants to buy as a parameter. It then checks the type of VIP and reduces his coins to the price of the VIP from the patient's coins if he has the money. If the patient already has a VIP membership, 30 days are added to his end date. If the patient does not have a VIP membership, 30 days are added to today's date, and the VIP level is set to the required one.
MyPrescriptions	This function returns all prescriptions written for the patient.
DownloadPrescription	This function allows the patient to download a selected prescription by selecting a location on their computer to download the prescription at.

Table 4.3 Patient GUI Class

Function Name	Description
__init__	This is our Python constructor that will include all the functions required to construct the patient page in a more comprehensive and organized manner.
WindowSettings	This function will apply some custom adjustments to the page, such as applying a specific color theme and setting the width to 1280 and the height to 720.
LeftSideBar	This function will include the Patient Name, Age, VIP level, and End Date (if available) along with buttons for each feature that the patient can utilize.
LoadPredictScanFrame	This function will display a textbox and button that will allow the patient to upload their X-ray scan for prediction.
ImportScan	This function will be called when the patient presses the Import Scan button to open a popup that allows him to select the location of the X-ray scan image he want to import and predict. It will then call the function responsible for predicting the X-ray scan.
Purchase_VIP	This function will display the available VIP plans for the patient to purchase, along with their respective prices and benefits. The patient can select a plan and proceed with the payment process if they wish to upgrade their VIP status.
CreditCardType	This function will display the type of credit card being used in our system to the patient.

CardNumberValidation	This function is responsible for validating credit card numbers. It checks the format and structure of the provided credit card number to ensure it is valid. If the credit card number is valid, the function returns true; otherwise, it returns false.
FormatCreditCard	This function will split the credit card number into groups of four digits, separated by a space, to enhance its readability.
HandleCVV	This function will check the validity of the CVV (Card Verification Value) entered by the patient.
change_appearance_mode	This function will change the system's appearance mode to the option selected by the patient from the combo box.
Consult	This function checks if the patient has had a conversation with the chatbot, and if so, enables the patient to start a chat session with a doctor.
FillRequest	This function will send the patient's chat request to the system and save it in the database until a doctor helps him.
openChat	This function will encapsulate all the necessary functions that will be invoked during the patient's chat session with either the chat bot or the doctor. It will serve as a centralized hub that manages the different actions that can be performed during the chat session.
DoctorData	This function will display the doctor's name and position within the system to the patient during their chat session.
JoinChatServer	This function will establish a connection between the patient and the doctor by connecting to the server.
ShowAllPrescriptions	This function will display to the patient a list of all his prescriptions that have been created by doctors.

Table 4.4 Doctor Class

Function Name	Description
<u>__init__</u>	This is our constructor in Python. It takes user ID as a parameter, retrieves the user's basic information from the database, such as their name, email, password, account type, coin balance, phone number, age, system appearance mode, gender, verification status, university, ID card, and professional license.
CreateDoctor	This is a class method that serves as a secondary constructor. It takes parameters such as the user's name, email, password, account type, phone number, age, gender, verification status, university, ID card, and professional license. Using this information, it creates a new user account with the provided data.
ReportUser	This function will create a report based on the patient's behavior and save it into the database. Additionally, it will suspend the patient's access to the system.
GetMyChatMembers	This function will retrieve all the patients who currently have an active chat with the doctor and return a list of their names.
HandlePrescription	This function checks if a prescription has been created for a patient or not. If a prescription has not been created, it prompts the doctor to create one.
MakePrescription	This function will get the data doctor added to the prescription and fill the data in our prescription formatted PDF
EndChat	This function will end the chat between the patient and the doctor, save the chat logs in the database, and delete the chat data and patient request from the chatdata table.
LoadWaitingPatientRequests	This function will display a list of all patients who are waiting for doctors' help, ordered by VIP level and the date their request was created.
AssignMePatient	This function sets the patient's request to ongoing and initiates a chat session with the patient. It first updates the status of the patient's request to ongoing, indicating that a doctor is now attending to it. It then creates a new chat session with the patient, allowing the doctor to communicate with the patient.

Table 4.5 Doctor GUI Class

Function Name	Description
__init__	This is our Python constructor that will include all the functions required to construct the doctor page in a more comprehensive and organized manner.
WindowSettings	This function will apply some custom adjustments to the page, such as applying a specific color theme and setting the width to 1280 and the height to 720.
LeftSideBar	This function will display the name of the doctor and their position in the system, as well as provide access to various features that the doctor can use through buttons on the interface.
LoadActiveChat	This function will display a list of all patients who are currently in a chat session with the doctor.
openChat	This function is triggered when the doctor clicks on a patient's name to open their chat window. It loads the Chatbot logs and chat logs, as well as the patient's information and health status, including their X-ray scan with its prediction (if available).
FillMedication	This function will display a page that allows the doctor to add or remove medicines from the patient's prescription. The doctor can choose to show the prescription or save it.
JoinChatServer	This function will establish the connection between the doctor and the patient by connecting to the server. It will handle the socket connection and enable the doctor to send and receive messages in real-time with the patient.
CardNumberValidation	This function is responsible for validating credit card numbers. It checks the format and structure of the provided credit card number to ensure it is valid. If the credit card number is valid, the function returns true; otherwise, it returns false.
CreditCardType	This function will display the type of credit card being used in our system to the patient.
FormateCreditCard	This function will split the credit card number into groups of four digits, separated by a space, to enhance its readability.
HandleCVV	This function will check the validity of the CVV (Card Verification Value) entered by the patient.
change_appearance_mode	This function will change the system's appearance mode to the option selected by the patient from the combo box.

Table 4.6 Radiologist Class

Function Name	Description
<u>__init__</u>	This is our Python constructor which takes the user ID as a parameter and retrieves the user's basic information from the database, such as their name, email, password, account type, coins balance, phone number, age, system appearance mode, and gender.
CreateRadiologist	This is a class method that serves as a secondary constructor. It takes parameters such as name, email, password, account type, phone number, age, gender, and radiology center name to create a new user with the provided data.
PredictScanFolder	This function allows the radiologist to import a folder containing X-ray scans and obtain predictions for each scan.
createcsv	This function will take a folder path as input which contains X-ray scans, and it will use a pre-trained machine learning model to predict each scan. It will then create a CSV file in the same folder location as the scans, containing each image with its corresponding prediction in each row.

Table 4.7 Radiologist GUI Class

Function Name	Description
<u>__init__</u>	This is our Python constructor that will include all the functions required to construct the doctor page in a more comprehensive and organized manner.
WindowSettings	This function will apply some custom adjustments to the page, such as applying a specific color theme and setting the width to 1280 and the height to 720.
LeftSideBar	This function will display the radiologist's name, the radiology center he works in, and buttons for each feature that the radiologist can perform.
LoadPredictScanFrame	This function will display a button in the user interface. When clicked, it will open a popup dialog box allowing the radiologist to select the location of the folder containing the X-ray images.
ShowImage	This function will show each image and its prediction in one row if any image or its prediction pressed on the image will be shown inside the system

Table 4.8 Administrator Class

Function Name	Description
__init__	This is our Python constructor which takes a user ID as a parameter and retrieves the user's basic information from the database, such as their Name, Email, Password, Account Type, Coins Balance, Phone, Age, System Appearance Mode, and Gender.
getAllReports	This function will return every report in reports table that is not handled yet.
RevokeSuspension	This function will delete the report from the database, lift the suspension of the patient's account, and send an email notification to inform the patient that their account has been re-activated.
ConfirmSuspension	This function will delete the report and make patient's suspension to be permanent and send mail to patient that his account has been permanently suspended.
getUnverifiedDoctors	This function will return every unverified doctor that has registered.
VerifyDoctor	This function will update doctor's data and verify his account and send mail to him that his account has been verified.
BanDoctor	This function will update doctor's data and unverify his account, suspend the doctor from using the system and send mail to him that his account has been suspended.
GetMail	This function will get the email of the user that admin will perform his action on him.
SendMail	This function will allow the system to send the emails to the users.

Table 4.9 Administrator GUI Class

Function Name	Description
__init__	This is our Python constructor that will include all the functions required to construct the administrator page in a more comprehensive and organized manner.
WindowSettings	This function will apply some custom adjustments to the page, such as applying a specific color theme and setting the width to 1280 and the height to 720.
LeftSideBar	This function will contain the buttons for each feature the administrator can do.
LoadHandleReportsFrame	This function will display a frame that contains the information of a doctor's report, such as the doctor's name, university name, ID card, and professional license. The frame will also have two buttons, one with a right icon to verify the doctor and the other with a wrong icon to ban the doctor.
showChat	This function will display the chat logs between a patient and their doctor. It will include two buttons for the administrator to either confirm the suspension of the patient's account or revoke the suspension.
LoadVerifyDoctorsFrame	This function will show frame holding the information of unverified doctors like Doctor Name, university name, ID card, and professional license.

Table 4.10 UserFactory Class

Function Name	Description
createUser	This is a class method that can create an instance of any class, such as Patient, Doctor, Radiologist, or Administrator, by providing the ID and account type as input. The method will then create and return the instance.

Table 4.11 Starter Class

Function Name	Description
__init__	This is our Python constructor that will include all the functions required to construct landing page in a more comprehensive and organized manner.
WindowSettings	This function will apply some custom adjustments to the page, such as applying a specific color theme and setting the width to 1280 and the height to 720.
login_gui	This function displays a login form which includes two text fields for entering email and password, a button for submitting the form, and a forget password button.
login_verify	This function checks whether the entered email and password are valid, it also checks whether the doctor's account is verified or not.
forgot_password	This function will check if the email entered is valid and if it belongs to an existing account. If it does, it will send the password of the account to that email.
mainRegister	This function displays a registration form for users to create an account in our system. It includes fields for the user's name, email, password, account type.
fetchAllData	This function retrieves the data entered by the user in the register page and initiates the process of validating the input.
dataValidator	This function validates the user input in the registration page by calling corresponding functions for each input field.
insertUserInfo	This function creates a new user using the data entered by the user in the registration page.
MoveTo	This function closes the login page and navigates the user to the required page according to their account type.

As mentioned in the background section, the prediction model is a crucial component of the system. Our custom version of ResNet is a testament to this fact. In this regard, it is essential to highlight the process of transforming the theoretical concept into a practical model that can predict diseases with an impressive accuracy rate of 99%.

Table 4.12 ResNet Model Class

Function Name	Description
ResidualBlock	This function represents a residual block. It takes an input, number of filters, a Boolean variable to check whether the filter size matches, and saves the input in a variable for later use. If the filter size matches, it applies convolution with a size of 3x3 and stride equals 2 to the input. If not, it applies convolution with a size of 3x3 and stride equals 1 to the input. Then, it applies batch normalization to the output of the previous operation, followed by the activation function ReLU. Next, it applies convolution with a size of 3x3 and stride equals 1 to the output of the previous operation, followed by batch normalization. The function then performs matching of filter numbers if necessary on the variable that stores the input, adds the input and output, applies the activation function ReLU to the output of the previous operation, and returns it.
Blocks	This function will call the ResidualBlock function multiple times to complete our model. First, we will create two nested loops. Each inner loop will iterate three times. If the inner loop index equals zero and the outer loop index is greater than zero, we will double the filter size and call the ResidualBlock function. Otherwise, we will call the ResidualBlock function without doubling the filter size. After completing the two loops, the function will return the output.
WholeModel	This function will build the entire model consisting of residual blocks and output layers. First, it takes the input image and applies a convolution with a size of 3x3 and stride equals 1. Then, it applies Batch normalization on the output of the previous operation and applies the activation function ReLU on the output. After that, it calls the Blocks function with the input as the output of the previous operation. Then, it applies Average pooling on the output of the previous operation, followed by flattening the output. After that, it applies a Dense layer to the output and returns the final output.

ModelCompile	This function will call the WholeModel function to create the model and compile it by adding the loss function, optimizer, and metrics. After that, it will return the compiled model.
ModelTrain	This function is responsible for training our model. It calls the model.fit function with the training and validation data, as well as the batch size and number of epochs specified as input arguments. During the training process, the model is optimized based on the specified loss function, optimizer, and metrics.
ModelEvaluate	This function will evaluate the performance of the trained model using test data. It will call the model.evaluate function with the test data and return the evaluation metrics such as loss and accuracy.
TrainingProcess	This function is called to start the training process. It also checks if the evaluation is active or not.
PredictScan	The function will enable the system user to make predictions on their own X-ray images.

In order to optimize the performance of our custom ResNet model on X-ray images, we have conducted some preprocessing. Our analysis of the images revealed the presence of noise in some of them. We recognized that this noise could potentially enable our model to accurately predict noisy images, so we made a decision to include these images in our training set.

To ensure consistency and facilitate efficient processing, we resized all the images to 512x512. Given that the images had varying light intensities and rotations, we decided against any augmentation techniques. Our goal was to maintain the original characteristics of the images as much as possible while still ensuring they were in a standardized format that our model could work with.

By applying these preprocessing steps, we believe that our custom ResNet model will be better equipped to accurately predict the presence of abnormalities in X-ray images. We recognize the importance of ensuring that our model is able to identify and work with noisy images, as these are commonly encountered in clinical practice.

Table 4.13 ImageProcessing Class

Function Name	Description
GenerateDataSet	This function generates the dataset in the form of a numpy file if it is not found. It first reads all the images in the train folder for each disease, converts them to grayscale, and resizes them to 512x512. It then creates a label for each image with the name of its disease, converts this label to an integer, and finally converts that integer to a binary class matrix.
load_images_from_folder	This function is called by GenerateDataSet to read images and resize them to 512x512. It takes a path to the folder containing the images and their labels. It reads all images in the folder, converts them to grayscale, resizes them to 512x512, and saves them along with their corresponding labels in a list. The list is then returned.
create_label	The purpose of this function is to create a label for each image based on the folder it is located in, where the folder name corresponds to the disease name. It takes in the folder path as an argument, reads the images in the folder, and assigns a label to each image based on the folder name. The function then returns a list of tuples, where each tuple contains the image array and its corresponding label.
load_image	This function is used to load an X-ray image imported by the system's users and prepare it for use with our prediction model. It reads the image and resizes it to 512x512 pixels so that the model will be able to make predictions on it.

Storing all data within the system is critical, and as such, a database is an essential component. Therefore, we have implemented a database using a design pattern called the Singleton design pattern. This pattern ensures that only one instance of the database connection is created.

The Singleton design pattern is an effective way of managing resources and ensuring that there are no unnecessary instances of the database connection. It ensures that the system operates efficiently and effectively without any performance issues.

By using the Singleton design pattern, we can maintain a consistent and reliable connection to the database, which is crucial for the proper functioning of the system. This ensures that the data is stored correctly, and it is easily accessible to all components of the system that require it.

Table 4.14 Database Class

Function Name	Description
getInstance	This function will check if there is no instance for database then create one and return that instance.
__init__	This is the constructor check if there is no instance for database then call Connect function.
Connect	This function will establish the connection with the database
Update	This function is made to perform the update queries with database.
Select	This function is made to perform the select queries with database.
Insert	This function is made to perform the insert queries with database.
Delete	This function is made to perform the delete queries with database.
Commit	This function is made to apply any change made to the database.

The chat feature is a critical component of our system as it enables patients to communicate with doctors in different locations. This feature is implemented using advanced communication technologies, which provide real-time communication capabilities between patients and doctors.

To implement the chat feature, we first had to identify the best communication technologies that could support real-time messaging. We selected a combination of technologies, including Socket, to ensure that the chat feature was stable, reliable, and provided an excellent user experience.

Socket provides a two-way, real-time communication channel between the server and the client. We integrated this technology into the system, and the chat feature was made available for patients to communicate with doctors.

Table 4.15 Client Class

Function Name	Description
__init__	This is our constructor, which initializes a socket and connects it to the server using the provided username and channel that the user wants to connect to.
writeToServer	The function will encode the user message and send it to the server.
receiveFromServer	This function will constantly listen to the server to receive any messages sent to the user and decode the received messages to make them readable.
end	This function will end the connection to the server.

Table 4.16 Server Class

Function Name	Description
__init__	This is the constructor of our chat server class. It initializes an empty dictionary to hold the chat rooms and lists to keep track of connected clients and their names. It also binds the server to a given address and port number and waits for any incoming connections from clients.
sendToAll	This function will send the message of a user to all other users in the same room that the message was sent to.
removeClient	The function will remove the user from all the rooms they are in if they disconnect from the server.
createRoom	This function will create a room with the id of patient to let both doctor and patient to connect to this room
clientInteraction	This function will wait for any encoded message from any client, decrypt the message, and call the sendToAll function to send it to all clients in the same room as the sender.
RunServer	This function handles all the functionalities of the Server Class. It accepts a new connection from a client and asks the client to provide their name and the desired room to join. The function then adds the client to the requested room if it exists. If the room does not exist, it creates the room for the client.

After the implementation of our system, it is crucial to test its functionality to ensure that it operates according to its intended design. To achieve this, various testing techniques are employed.

Unit testing is one of the most critical and widely used testing techniques employed during the development of a system. It is a type of software testing that focuses on testing individual components or units of code in isolation from the rest of the system. In this approach, each function or module of the system is tested independently to ensure that it performs its intended task accurately and meets the specifications and requirements outlined during the development phase.

Unit testing can help to identify defects in the code early in the development process, allowing developers to fix them before the system is integrated and tested as a whole. This approach also makes it easier to locate and isolate any issues that may arise, minimizing the time required to debug and resolve problems.

Table 4.17 Patient Class and Patient GUI Class Unit Testing

Test Case #	Function Name	description	Input	Expected output	Output	State
1	LoadPredictScanFrame	aim to show page for the patient with textfield and button to import his scan	press Predict X-Ray Scan button	info Message appears with the price for prediction according to patient VIP level then frame will open with textfield and Import button for the scan	info Message appears with the price for prediction according to patient VIP level then frame will open with textfield and Import button for the scan	Pass
2	ImportScan	aim to import Patient Scan to the system and predict it	Press Import Scan Button and then navigate to the image and select it	Scan Will be shown in the system with the highest two classes and a button to save the prediction as a textfile in the same path of image	Scan Will be shown in the system with the highest two classes and a button to save the prediction as a textfile in the same path of image	Pass
3	ImportScan	aim to show Error Message with insufficient funds	Press Import Scan Button and then navigate to the image and select it but the patient has no enough coins	Error Message will be Shown with "insufficient funds"	Error Message will be Shown with "insufficient funds"	Pass
4	ChatWithDoctor	aim to chat with our chat bot	Press Chat button	Chat will be shown for the patient to start talking with our bot with Import button and Chat with doctor button	Chat will be shown for the patient to start talking with our bot with Import button and Chat with doctor button	Pass

5	ChatWithDoctor	aim to show info message with Waiting for a doctor to respond	Press Chat button but when Patient has pressed before on Chat with doctor and no doctor chosen him yet	info message with "Waiting for a doctor to respond"	info message with "Waiting for a doctor to respond"	Pass
6	ChatWithDoctor	aim to show Error Message with insufficient funds	press chat with doctor while having no coins in balance	Error Message will be Shown with "insufficient funds"	Error Message will be Shown with "insufficient funds"	Pass
7	ChatWithDoctor	aim to show Error Message with Please chat first with our assistant	press chat with doctor while patient did not talked with our bot	Error Message with "Please chat first with our assistant"	Error Message with "Please chat first with our assistant"	Pass
8	ChatWithDoctor	aim to show that chat with doctor started	Press Chat button but when Patient has pressed before on Chat with doctor and a doctor selected him	Chat will be shown for the patient to start talking with doctor with information about doctor	Chat will be shown for the patient to start talking with doctor with information about doctor	Pass
9	Purchase_VIP	aim to show frame that will have VIP Plans for the patient to get	press Purchase VIP Button	Frame will be shown with 3 different plans	Frame will be shown with 3 different plans	Pass
10	Subscribe	aim to set patient with silver VIP	press Subscribe button of Silver plan	coins will decrease by the amount written in the plan and Silver VIP logo will be added under patient Age under it will be the expire date of VIP	coins will decrease by the amount written in the plan and Silver VIP logo will be added under patient Age under it will be the expire date of VIP	Pass
11	Subscribe	aim to show Error Message with insufficient funds	press Subscribe button of any plan without having required amount of coins	Error Message will be Shown with "insufficient funds"	Error Message will be Shown with "insufficient funds"	Pass

12	ShowAllPrescriptions	aim to show All Patient's Prescriptions	press All My Prescriptions Button	table will be shown with doctor that wrote this prescription and its rank and the date also a button to download it	table will be shown with doctor that wrote this prescription and its rank and the date also a button to download it	Pass
13	ShowAllPrescriptions	aim to show that patient has no prescription yet	press All My Prescriptions Button while patient has no prescriptions yet	"No Prescription s Found!" in the middle of the frame	"No Prescription s Found!" in the middle of the frame	Pass
14	CardNumberValidation	aim to check if card number is only 16 digit and format it	Enter credit card as 5471462613718510	card number will be shown as 5471 4626 1371 8510 and added mastercard logo	card number will be shown as 5471 4626 1371 8510 and added mastercard logo	Pass
15	CardNumberValidation	aim to check if card number is only 16 digit and format it	Enter credit card as 7612reahgsdv76123 51	warning message with Credit Card is not 16 digit as a text shows for the user	warning message with Credit Card is not 16 digit as a text shows for the user	Pass
16	CardNumberValidation	aim to check if card number is only 16 digit and format it	Enter Credit card as 7612	warning message with Credit Card is not 16 digit as a text shows for the user	warning message with Credit Card is not 16 digit as a text shows for the user	Pass
17	HandleCVV	aim to valid CVV for credit card	Enter CVV as 123	Nothing will be shown	Nothing will be shown	Pass
18	HandleCVV	aim to valid CVV for credit card	Enter CVV qw1	warning message with CVV is not 3 digit as a text shows for the user	warning message with CVV is not 3 digit as a text shows for the user	Pass
19	CheckCard_button_event	aim to valid credit card expire date	select month 1 and year 21	error message with Credit Card Expired as a text shows for the user	error message with Credit Card Expired as a text shows for the user	Pass
20	CheckCard_button_event	aim to valid credit card expire date	select month 1 and year 25	Nothing will be shown	Nothing will be shown	Pass

21	Purchase	aim to show error of invalid card	Enter invalid card	error message with Credit Card is not checked	error message with Credit Card is not checked	Pass
22	Purchase	aim to add amount of coins from one plan	Enter valid card and select any offer	coins of the offer will be added to the balance and info message with "Balance Recharge Completed" will be shown	coins of the offer will be added to the balance and info message with "Balance Recharge Completed" will be shown	Pass
23	logout	aim to check that logout button will open Login Page	Press Logout Button	Closing the current page and open the login page	Closing the current page and open the login page	Pass

Table 4.18 Doctor Class and Doctor GUI Class Unit Testing

Test Case #	Function Name	description	Input	Expected output	Output	State
1	select_frame_by_name	Aim is to check if Hover shows on right selection and open corresponding frame for it	select Active Chat button	shows hover on Active chat button and open active chat frame	shows hover on Active chat button and open active chat frame	Pass
2	select_frame_by_name	Aim is to check if it changes hover to the new selected button and remove old frame and open corresponding frame for that button	select patient requests	hover shows on the new button and old frame deleted and new one opens of patient requests	hover shows on the new button and old frame deleted and new one opens of patient requests	Pass
3	MyChats	Aim to show if doctor has patients to chat with them or no	login with a doctor that has patients assigned to him	the scrollable frame shall has data of patient like name	the scrollable frame shall has data of patient like name	Pass

4	MyChats	Aim to show that doctor has no patients to chat with	login with a doctor that has no patients assigned to him	the scrollable frame shall show no patients found	the scrollable frame shall show no patients found	Pass
5	openChat	Aim to show chat between doctor and selected patient with patient health status and scan	select a patient to open its chat	frame shall appears and with chatbox with old chat and patient health status and Scan with its prediction	frame shall appears and with chatbox with old chat and patient health status and Scan with its prediction	Pass
6	EndChat	Aim to show warning message when trying to end chat while Prescription was not created yet	select a patient where prescription was not created yet and press end chat button	Warning box with Report Should be Generated as a text for it	Warning box with Report Should be Generated as a text for it	Pass
7	EndChat	Aim to show info message after ending chat successfully	select a patient where prescription was created and press end chat button	info box with Credits added to your balance! as a text for it and adding 100 credit to his balance and delete patient from doctor chat	info box with Credits added to your balance! as a text for it and adding 100 credit to his balance and delete patient from doctor chat	Pass
8	ReportPatient	Aim to show new frame to enter Reason for report and then patient will be suspended and doctor will be given credits	select a patient and then press report button and Enter a reason	info box with User Reported Refund added to your balance as a text for it and adding 100 credit to balance and delete him from doctor chat and suspend that patient	info box with User Reported Refund added to your balance as a text for it and adding 100 credit to balance and delete him from doctor chat and suspend that patient	Pass

9	ReportPatient	Aim to show error message when leaving Reason empty	select a patient and then press report button and don't enter a reason	error message with Reason Should not be empty as a text shows for the user	error message with Reason Should not be empty as a text shows for the user	Pass
10	GeneratePrescription	Aim to show new frame to enter medications for patient and save it in database	select a patient and then press generate prescription and fill in some medicine and press save button	show a new frame with ability to add medicines and then save them	show a new frame with ability to add medicines and then save them	Pass
11	GeneratePrescription	Aim to show prescription while adding medicines	select a patient and then press generate prescription and fill in some medicine and press show button	show a new frame with ability to add medicines and info with prescription created	show a new frame with ability to add medicines and info with prescription created	Pass
12	GeneratePrescription	Aim to show error message when no medicine added	select a patient and then press generate prescription and press show or save button	show error message with No Medicine to be added to Prescription	show error message with No Medicine to be added to Prescription	Pass
13	GeneratePrescription	Aim to add medicine without explain for it	select a patient and then press generate prescription and fill in some medicine without explain and press show or save button	Prescription will be shown or saved	Prescription will be shown or saved	Pass
14	JoinChatServer	Aim to show error When Server is offline	Select patient with server is offline	show error message with Problem With Chat Server	show error message with Problem With Chat Server	Pass
15	JoinChatServer	Aim to load chat to chatbox and send and receive chat	Select any patient	old Chat will be loaded to Chatbox and can send, receive messages	old Chat will be loaded to Chatbox and can send, receive messages	Pass
16	loadWaitingPatients	Aim to show patients that waiting for doctors to	select patient requests with patients found	shows patients listed according to vip level	shows patients listed according to vip level	Pass

		chat with them with patients found		and request date in frame	and request date in frame	
17	loadWaitingPatients	Aim to show No Patients Found	select patient requests with no patients found	shows No Patients Found in Frame	shows No Patients Found in Frame	Pass
18	AssignMePatient	Aim to Assign patient and remove it from patient requests and add him to doctor chat	select patient requests and press chat button of any patient	patient will be removed from patient requests and added to doctor chat	patient will be removed from patient requests and added to doctor chat	Pass
19	CardNumberValidation	aim to check if card number is only 16 digit and format it	Enter credit card as 5471462613718510	card number will be shown as 5471 4626 1371 8510 and added mastercard logo	card number will be shown as 5471 4626 1371 8510 and added mastercard logo	Pass
20	CardNumberValidation	aim to check if card number is only 16 digit and format it	Enter credit card as 7612rehagsdv7612351	warning message with Credit Card is not 16 digit as a text shows for the user	warning message with Credit Card is not 16 digit as a text shows for the user	Pass
21	CardNumberValidation	aim to check if card number is only 16 digit and format it	Enter Credit card as 7612	warning message with Credit Card is not 16 digit as a text shows for the user	warning message with Credit Card is not 16 digit as a text shows for the user	Pass
22	HandleCVV	aim to valid CVV for credit card	Enter CVV as 123	Nothing will be shown	Nothing will be shown	Pass
23	HandleCVV	aim to valid CVV for credit card	Enter CVV qw1	warning message with CVV is not 3 digit as a text shows for the user	warning message with CVV is not 3 digit as a text shows for the user	Pass
24	CheckCard_button_event	aim to valid credit card expire date	select month 1 and year 21	error message with Credit Card Expired as a text shows for the user	error message with Credit Card Expired as a text shows for the user	Pass

25	CheckCard_button_event	aim to valid credit card expire date	select month 1 and year 25	Nothing will be shown	Nothing will be shown	Pass
26	Withdraw_button_event	aim to withdraw required amount	Enter valid card and select required amount (5)	infobox with Credits will be added to you bank account Soon! And remove 5 credits from doctor balance	infobox with Credits will be added to you bank account Soon! And remove 5 credits from doctor balance	Pass
27	Withdraw_button_event	aim to show error of invalid card	Enter invalid card	error message with Credit Card is not checked	error message with Credit Card is not checked	Pass
28	Withdraw_button_event	aim to show error of invalid amount	Enter valid card and don't select amount	error message with Invalid Amount	error message with Invalid Amount	Pass
29	logout	aim to check that logout button will open Login Page	Press Logout Button	Closing the current page and open the login page	Closing the current page and open the login page	Pass

Table 4.19 Radiologist Class and Radiologist GUI Class Unit Testing

Test Case #	Function Name	description	Input	Expected output	Output	State
1	ImportScanFolder	aim to check that function show filedialog to choose folder that contains scan(s) to be predicted	press on Predict X-ray Scan button then press Import Folder	Filedialog will be shown for the user then after user selects his folder then info Message will be shown with "Don't Panic!" then model will start prediction at the end, table will be show with image name and prediction for it also a CSV file will be generated with two columns one for image name and second for prediction	Filedialog will be shown for the user then after user selects his folder then info Message will be shown with "Don't Panic!" then model will start prediction at the end, table will be show with image name and prediction for it also a CSV file will be generated with two columns one for image name and second for prediction	Pass
2	ShowImage	aim to show Scan for user if he pressed on image name or its prediction in the table	Press on Name of Scan or its prediction in the table	Pressed Image will be shown for the user near the table	Pressed Image will be shown for the user near the table	Pass
3	logout	aim to check that logout button will open Login Page	Press Logout Button	Closing the current page and open the login page	Closing the current page and open the login page	Pass

Table 4.20 Administrator Class and Administrator GUI Class Unit Testing

Test Case #	Function Name	Description	Input	Expected output	Output	State
1	LoadHandleReportsFrame	aim to show chat reports made by doctors if found	Press on Handle Reports	Table shown with Doctor Name, Patient Name, Report Reason, Date of Report, Button For Chat history	Table shown with Doctor Name, Patient Name, Report Reason, Date of Report, Button For Chat history	Pass
2	showChat	aim to show chat history for the admin to revoke or confirm suspension	press on Show Chat	New page will be opened with chat history and two buttons one for Revoke suspension and other to Permanent Suspend the patient	New page will be opened with chat history and two buttons one for Revoke suspension and other to Permanent Suspend the patient	Pass
3	RevokeSuspension	aim to remove any patient suspension made by wrong Report	Press Revoke Suspension in Chat History page	Report will be removed as well as patient suspension will be removed also a mail will be sent to patient's mail with subject "Account Activation" and body "Your account has been re-activated successfully" then Info Message will be shown for the admin with "Suspension revoked for the patient"	Report will be removed as well as patient suspension will be removed also a mail will be sent to patient's mail with subject "Account Activation" and body "Your account has been re-activated successfully" then Info Message will be shown for the admin with "Suspension revoked for the patient"	Pass

4	ConfirmSuspension	aim to permanent suspend the patient and remove the report	Press Permanent Suspension in Chat History page	Report will be removed and patient suspension type will be updated to permanent and then an email will be sent to patient with subject "Account Activation" and body "Your account has been permanently suspended" and an info message will be shown for the admin "Patient permanently suspended"	Report will be removed and patient suspension type will be updated to permanent and then an email will be sent to patient with subject "Account Activation" and body "Your account has been permanently suspended" and an info message will be shown for the admin "Patient permanently suspended"	Pass
5	LoadVerifyDoctorsFrame	aim to show unverified doctors if found	Press on Verify Doctors	Table shown with Doctor Name, University Name, ID card, Profession License and verify button and unverify button	Table shown with Doctor Name, University Name, ID card, Profession License and verify button and unverify button	Pass
6	VerifyDoctor	aim to verify a doctor to be able to login to the system		doctor will be verified and an email will be sent to him to use his account and info message will be shown for admin with "Doctor Verified"	doctor will be verified and an email will be sent to him to use his account and info message will be shown for admin with "Doctor Verified"	Pass

7	BanDoctor	aim to ban doctor from our system	Press on Unverify icon	doctor will be unverified and added to suspended accounts of the system and an email will be sent to doctor with subject "Account Activation" and body "Your account has been suspended" and an info message will be shown for admin with "Doctor Suspended"	doctor will be unverified and added to suspended accounts of the system and an email will be sent to doctor with subject "Account Activation" and body "Your account has been suspended" and an info message will be shown for admin with "Doctor Suspended"	Pass
8	logout	aim to check that logout button will open Login Page	Press Logout Button	Closing the current page and open the login page	Closing the current page and open the login page	Pass

Table 4.21 Starter Class Register Part Unit Testing

Test Case #	Function Name	description	Input	Expected output	Output	State
1	dataValidator	aim to check if when register button pressed without filling in the fields should return an error for the user	Press Register Button	Error message with 'Please fill all the fields!' shows up for user	Error message with 'Please fill all the fields!' shows up for user	Pass
2	userNameChecker	aim to check if that username should be only alpha not digits	Enter First name as test123 and second name as test	Error message with 'Invalid Name!' shows up for user	Error message with 'Invalid Name!' shows up for user	Pass
3	userNameChecker	aim to check if that username should be only alpha not digits not special Characters	Enter First name as test@# and second name as test	Error message with 'Invalid Name!' shows up for user	Error message with 'Invalid Name!' shows up for user	Pass
4	emailChecker	aim to check if that email should be in its format	Enter mail as test	Error message with 'Invalid email address!' shows up for user	Error message with 'Invalid email address!' shows up for user	Pass
5	passwordChecker	aim to check if password equals to Confirm password	Enter password as test and confirm as test1	Error message with 'Password Mismatch!' shows up for user	Error message with 'Password Mismatch!' shows up for user	Pass
6	passwordChecker	aim to check if password is greater than or equal to 8 characters	Enter password as test and confirm as test	Error message with 'Password should be longer than 8 characters!' shows up for user	Error message with 'Password should be longer than 8 characters!' shows up for user	Pass

7	phoneChecker	aim to check if phone number is only digit not alpha	Enter phone number as 01test	Error message with 'Invalid phone number!' shows up for user	Error message with 'Invalid phone number!' shows up for user	Pass
8	phoneChecker	aim to check if phone number is between 11 and 15	Enter phone number as 01234123	Error message with 'Invalid phone number!' shows up for user	Error message with 'Invalid phone number!' shows up for user	Pass
9	phoneChecker	aim to check if phone number is between 11 and 15	Enter phone number as 01241234615 24167	Error message with 'Invalid phone number!' shows up for user	Error message with 'Invalid phone number!' shows up for user	Pass
10	genderValid	Aim to check if user does not select any gender	Don't select any gender	Error message with 'Please select your gender' shows up for user	Error message with 'Please select your gender' shows up for user	Pass
11	CheckRadioCenter	Aim to check that Radiologist can not register with invalid Code for his center	Enter code test for Alpha Scan center	Error message with 'Invalid radiology center code!' shows up for user	Error message with 'Invalid radiology center code!' shows up for user	Pass
12	CheckRadioCenter	Aim to check that Radiologist can not register with code that its limit is 0	Enter code ygah183665hj sbf for Cairo Scan center	Error message with 'Your radiology center has reached its maximum amount of users!' shows up for user	Error message with 'Your radiology center has reached its maximum amount of users!' shows up for user	Pass
13	fetchUserTypeData	Aim to check if Doctor did not import his ID	Don't Import ID but Import prof lic	Error message with 'Please check your ID' shows up for user	Error message with 'Please check your ID' shows up for user	Pass
14	fetchUserTypeData	Aim to check if Doctor did not import his prof lic	Import ID but Don't Import prof lic	Error message with 'Please check your Profession License' shows up for user	Error message with 'Please check your Profession License' shows up for user	Pass

15	fetchUserTypeData	Aim to check if Doctor did not import both ID and prof lic	Don't Import ID or prof lic	Error message with 'Please check your ID and Profession License' shows up for user	Error message with 'Please check your ID and Profession License' shows up for user	Pass
16	fetchUserTypeData	Aim to check if Doctor did not enter his university	Import ID or prof lic and don't enter university	Error message with 'Please enter your university' shows up for user	Error message with 'Please enter your university' shows up for user	Pass

Table 4.22 Starter Class Login Part Unit Testing

Test Case #	Function Name	description	Input	Expected output	Output	State
1	login_verify	aim to check if user can not login without filling fields	press login button without filling data	Error message with 'Please fill all the fields!' shows up for user	Error message with 'Please fill all the fields!' shows up for user	Pass
2	login_verify	aim to check if user can not login if enter only email	enter email and press login	Error message with 'Please fill all the fields!' shows up for user	Error message with 'Please fill all the fields!' shows up for user	Pass
3	login_verify	aim to check if user can not login if enter only password	enter password and press login	Error message with 'Please fill all the fields!' shows up for user	Error message with 'Please fill all the fields!' shows up for user	Pass
4	login_verify	aim to check if user can not login if user is suspended	enter mail and password of suspended user then press login	Error message with 'You are suspended' shows up for user	Error message with 'You are suspended' shows up for user	Pass
5	login_verify	aim to check if doctor can not login until get verified	enter email and password of unverified doctor and press login	Error message with 'Please wait until you are verified' shows up for user	Error message with 'Please wait until you are verified' shows up for user	Pass

6	login_verify	aim to check if doctor can login if verified	enter email and password of verified doctor and press login	Login page will be closed and open Doctor's page	Login page will be closed and open Doctor's page	Pass
7	login_verify	aim to check if patient can login	enter email and password of patient and press login	Login page will be closed and open patient's page	Login page will be closed and open patient's page	Pass
8	login_verify	aim to check if radiologist can login	enter email and password of radiologist and press login	Login page will be closed and open radiologist's page	Login page will be closed and open radiologist's page	Pass
9	login_verify	aim to check if admin can login	enter email and password of admin and press login	Login page will be closed and open admin's page	Login page will be closed and open admin's page	Pass
10	forgot_password	aim to check if user can not get his password without entering email	press on forget password without entering mail	Error message with Please enter your email address' shows up for user	Error message with Please enter your email address' shows up for user	Pass
11	forgot_password	aim to check if user can not get his password with entering email	enter mail and then press on forget password	An email will be sent to the user with his password to the email he registered with it	An email will be sent to the user with his password to the email he registered with it	Pass

5. User Manual

Live Healthy



User Manual

Table of Contents

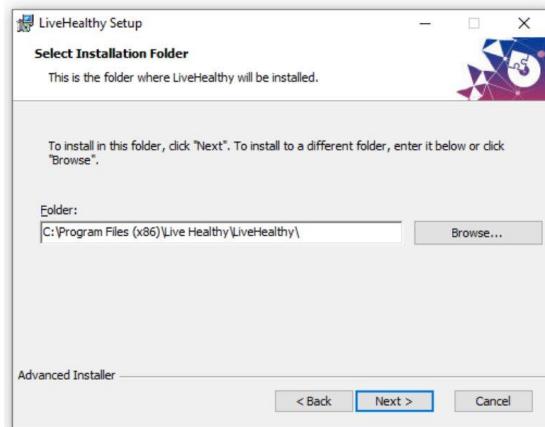
1. Installation Guide	2
2. Login Page	4
3. Register Page	5
4. Patient Guide	9
5. Doctor Guide	15
6. Radiologist Guide	20

1. Installation Guide

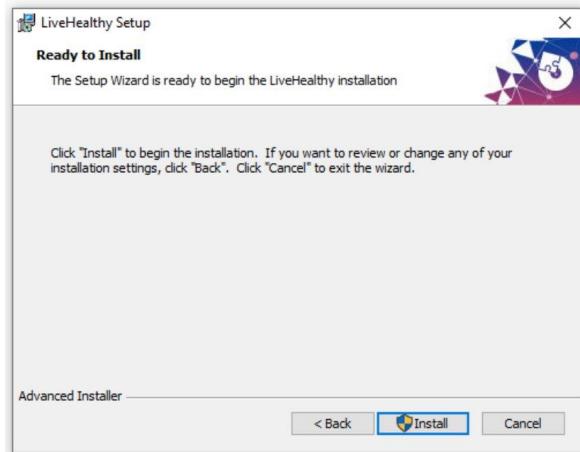
Click on the setup "LiveHealthy.msi"



Click on "Next"

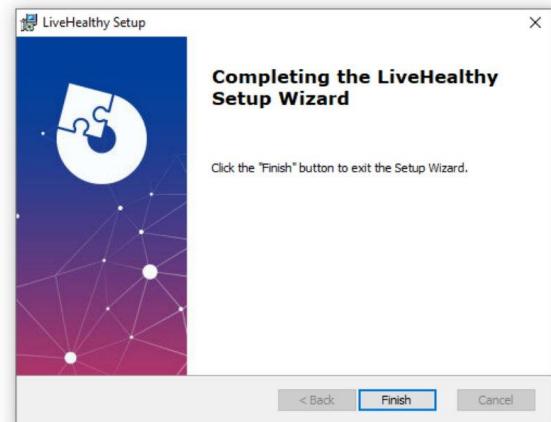


Click on "Next"



Click on "Install"

Then wait until the install finish

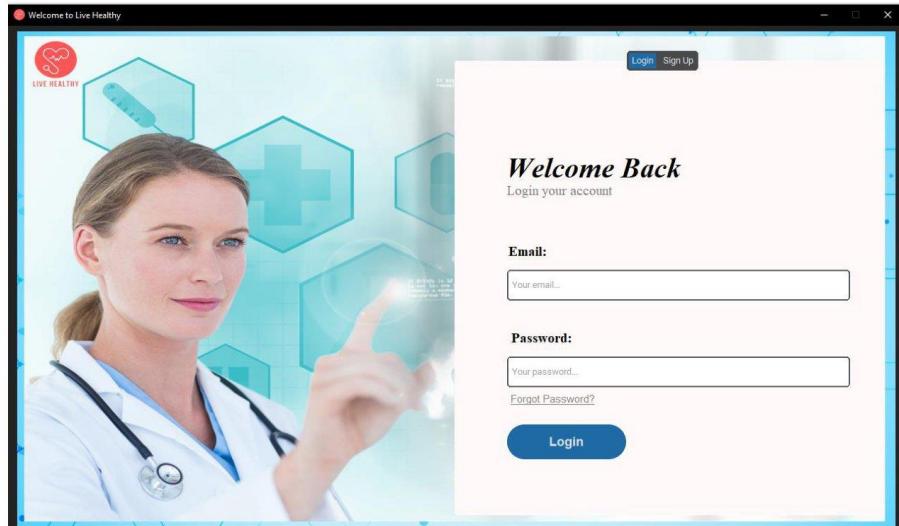


Click on "Finish"



Click on this icon found on your desktop

2. Login Page



This is the initial screen that appears when you run the system executable file. Here are the available options:

- "Sign Up" takes you to the registration page where you can create a new account for the system.
- "Forgot Password?" can be used if you have entered an existing email to receive an email with your password.
- "Login" directs you to the necessary page to access our system if you've entered an existing account.

3. Register Page

Welcome to Live Healthy

Full Name*
Input Your Full Name...

Email*
Input Your Email...

Password*
Input Your Password...

Confirm Password*
Confirm Your Password...

Phone Number*
Input Your Phone Number...

Date of Birth*
January 2001

Gender*
 Male Female

User Type*
Patient

Patient's Health Status

Heart Diseases Diabetes
 Cancer Obesity
 Smoker Hypertension
 Allergies

Blood Type: Unknown

Register

After clicking on the "Sign Up" button, the following screen will appear. Please provide your basic information, and then choose one of the following options for the "User Type": "Patient," "Radiologist," "Specialist," or "Consultant," depending on how you wish to access our system.

If you select "Patient," you will need to choose any chronic diseases that apply from the "Patient's Health Status" options. Additionally, if you know your blood type, please select it. If you are unsure of your blood type, you can leave it as "Unknown."

After filling in your information and selecting the appropriate options, click on the "Register" button to create your account. Once your account has been successfully created, click on the "Login" button. Enter your email and password on the login page to access the system.



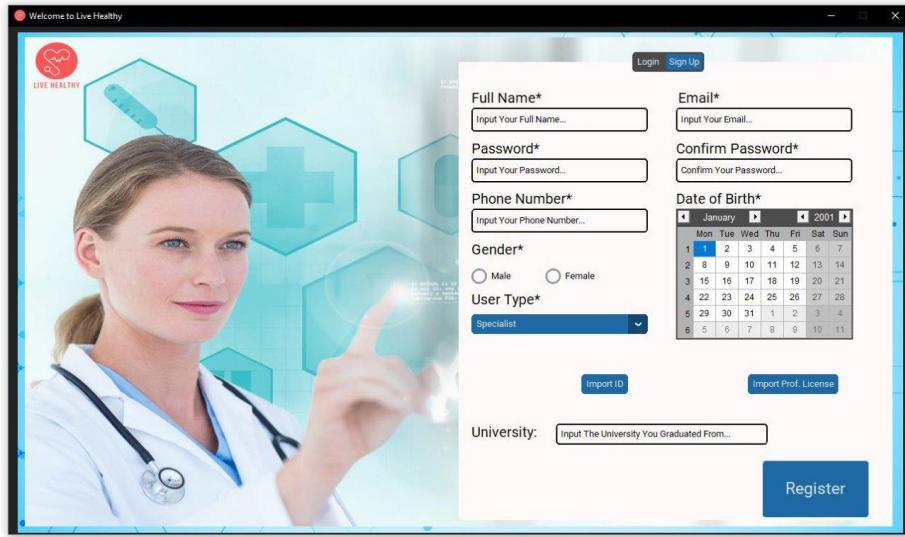
If you choose the "Radiologist" option, you will be required to select the Radiology Center that utilizes our system. Additionally, you need to enter the verification code that has been assigned to the center to complete the registration process.

After filling in your information and selecting the appropriate options, click on the "Register" button to create your account. Once your account has been successfully created, click on the "Login" button. Enter your email and password on the login page to access the system.



If you select the "Consultant" option, you will need to upload your ID card, Profession License, and provide the name of your university. It is important to provide genuine information as our administrators will review it to confirm or reject your access to the system.

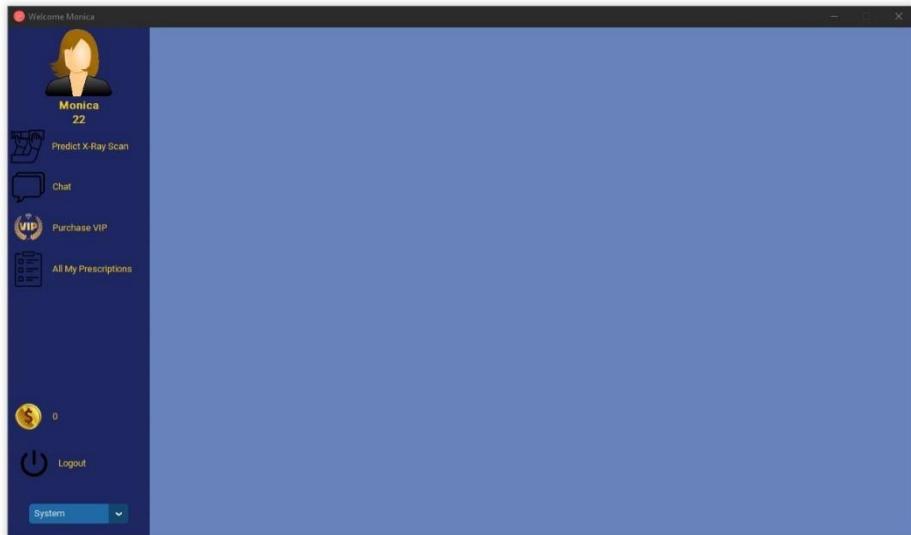
After entering all the required information and selecting the appropriate options, click on the "Register" button to create your account. Once your account has been successfully created, you will need to wait for an administrator to review your information. The administrator will then either confirm or reject your access. If your access is confirmed, you will receive an email that you can access system.



If you select the " Specialist" option, you will need to upload your ID card, Profession License, and provide the name of your university. It is important to provide genuine information as our administrators will review it to confirm or reject your access to the system.

After entering all the required information and selecting the appropriate options, click on the "Register" button to create your account. Once your account has been successfully created, you will need to wait for an administrator to review your information. The administrator will then either confirm or reject your access. If your access is confirmed, you will receive an email that you can access system.

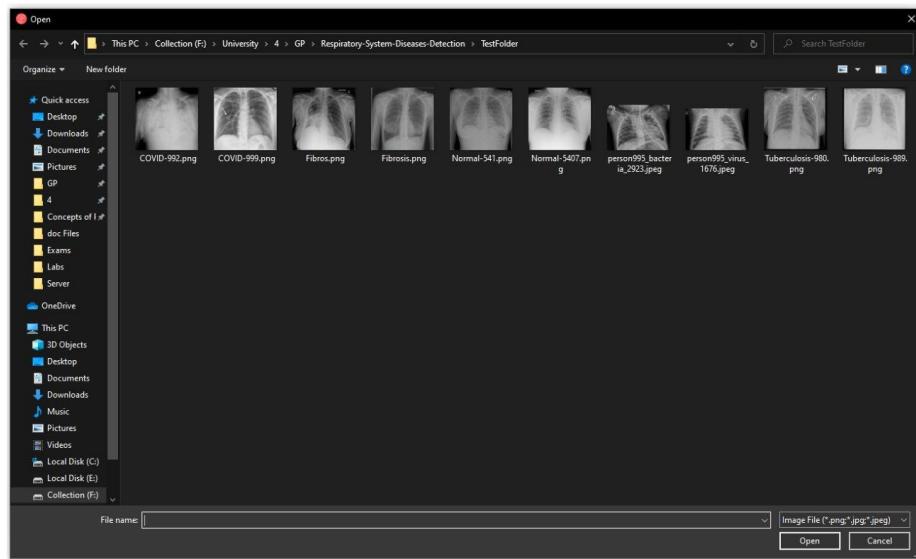
4. Patient Guide

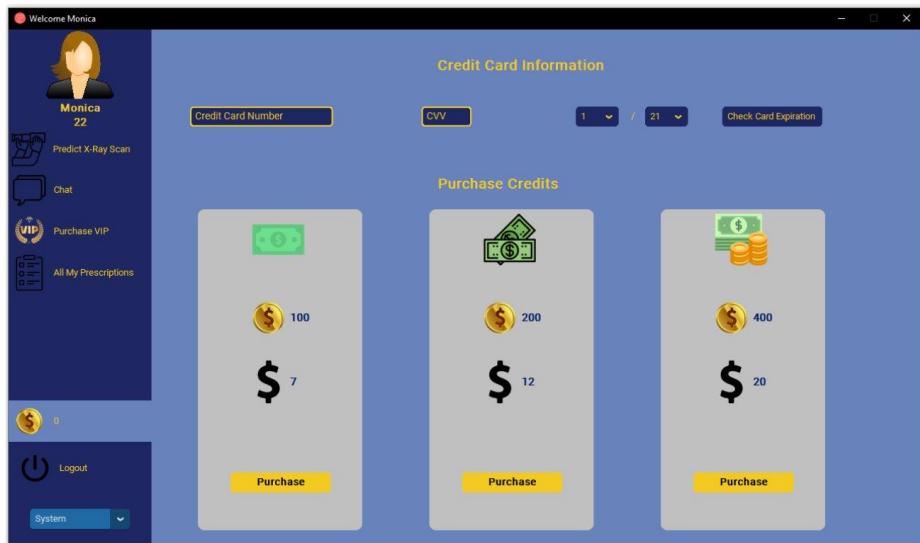


Once you login as a patient, the first screen you will encounter displays your name, your age, VIP level and its end date if found in the system. Additionally, you will find four buttons: "Predict X-ray Scan", "Chat", "Purchase VIP", "All My Prescriptions", "Coin" and "Logout".



After clicking on the "Predict X-Ray Scan" button, you will be presented with an "Import Scan" button. Use this button to select the scan you want to predict using the system model.

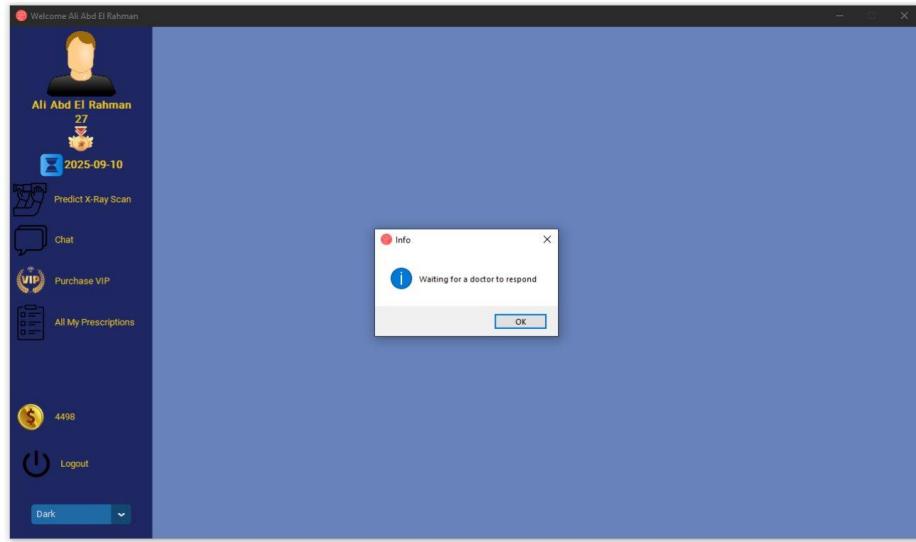




After clicking on "Coin" button, you enter your credit card information to get some coins to use system features.



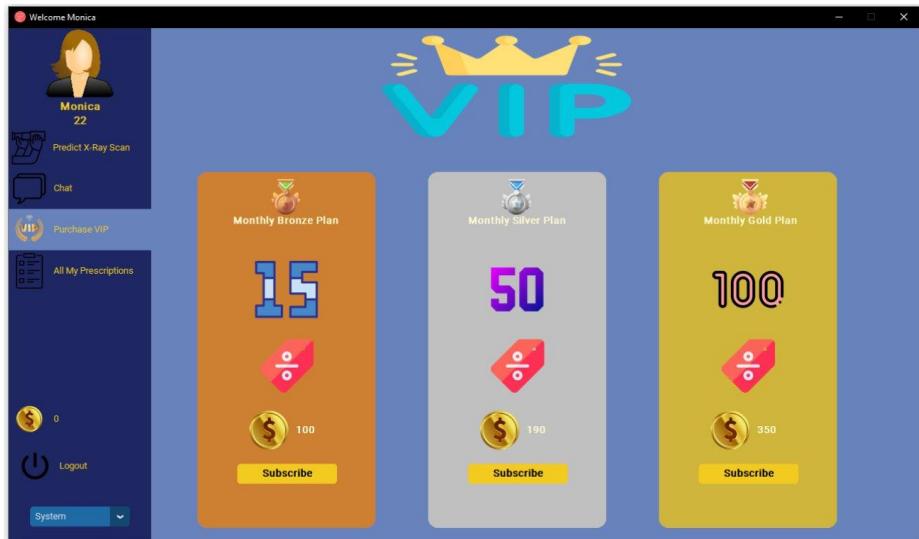
After clicking on "Chat" button, you can start chatting with our chat bot to try to figure out the disease you have.



After chatting with chat bot, you can press on "Chat with Doctor" to make a request to talk with one of our doctors so you will wait for a doctor to answer you to help you.



After a doctor start chatting with you so you can see his name and his position in our system

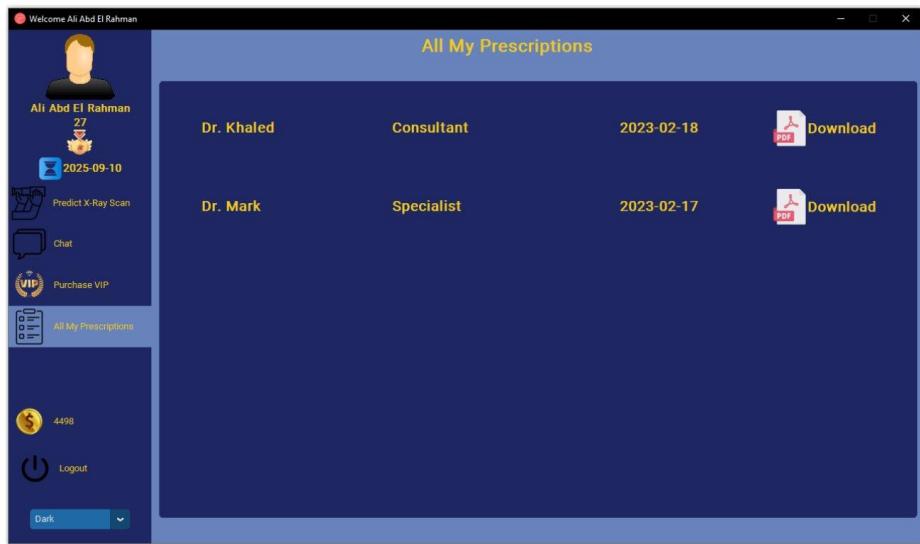


After clicking on "Purchase VIP" button, you can purchase VIP level for 30 days to get some discount for using system features.



	Gold VIP	Silver VIP	Bronze VIP	Non-VIP
Discount	100%	50%	15%	-
Scan Prediction	free	37	63	75
Chat with doctor	free	75	127	150

Price of each feature in coins



After clicking on "All My Prescriptions" button, you can see all prescriptions created for you by doctors in the system for each prescription doctor name, position in system, date of creation and download to download it to your computer.

5. Doctor Guide



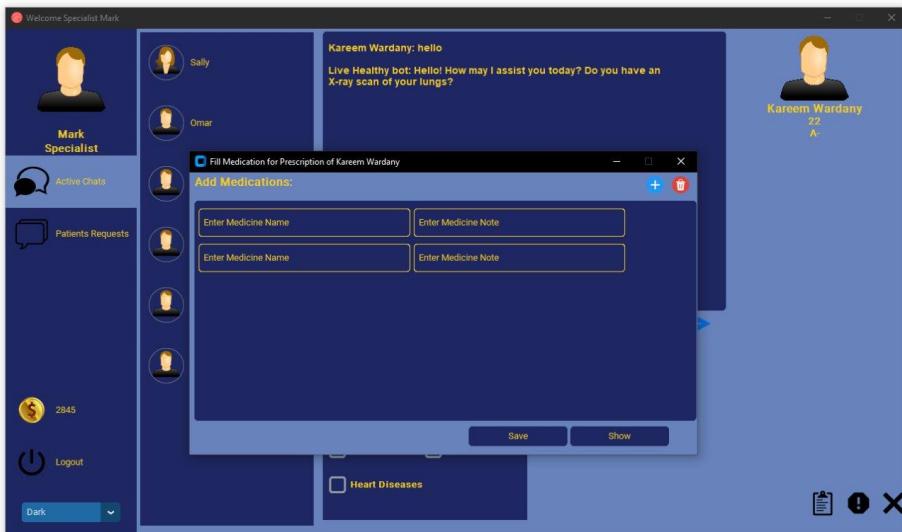
Once you login as a specialist or consultant, the first screen you will encounter displays your name and your position in the system. Additionally, you will find four buttons: "Active Chats", "Patients Requests", "Coin" and "Logout".



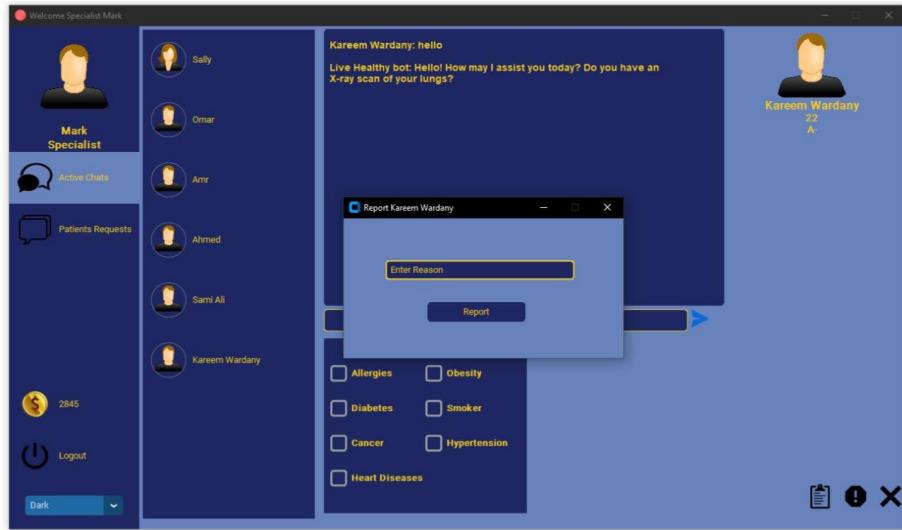
When you select "Active Chats", you will be able to view a list of all patients with whom you have an ongoing chat conversation.



When you select any patient the chat window will be opened with his name, age, Blood type, chronic diseases if found, Chat will our chat bot and his X-ray scan with its prediction if found.



When you click on a new window will be opened to enter medicine for patient to his prescription and you can save it for him to let him be able to download it.

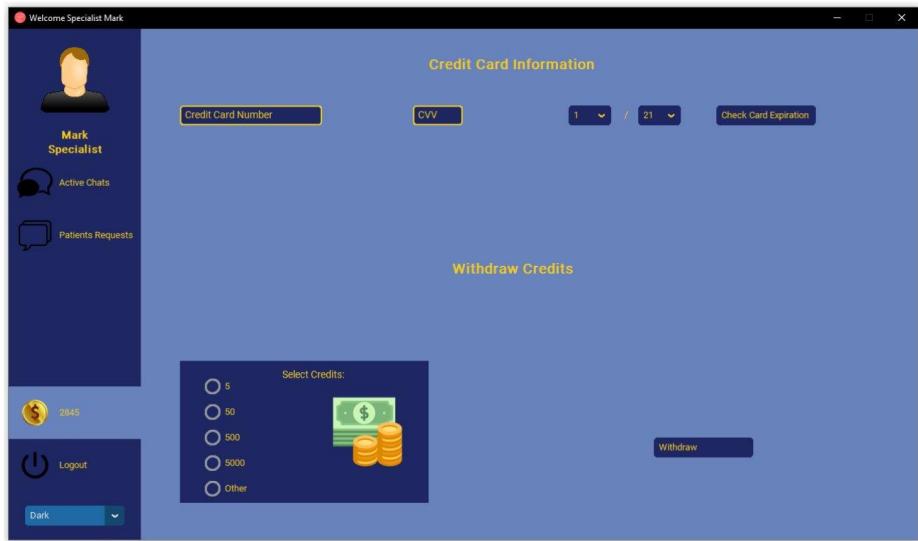


When you click on a new window will be opened to enter a reason for reporting patient for his bad behavior.

When you click on the chat between patient and you will be closed.



When you select "Patient Requests," you will have access to a list of all patients who are awaiting assistance from a doctor. In this list, you will see the patient's name, VIP level, gender, age, and the date the request was created. Each patient will have a chat button next to their information, allowing you to initiate a conversation with them.



When you click on the "Coin" button, you will be directed to a page where you can exchange system coins for real money. On this page, you can enter your credit card information and select the desired amount for the exchange.

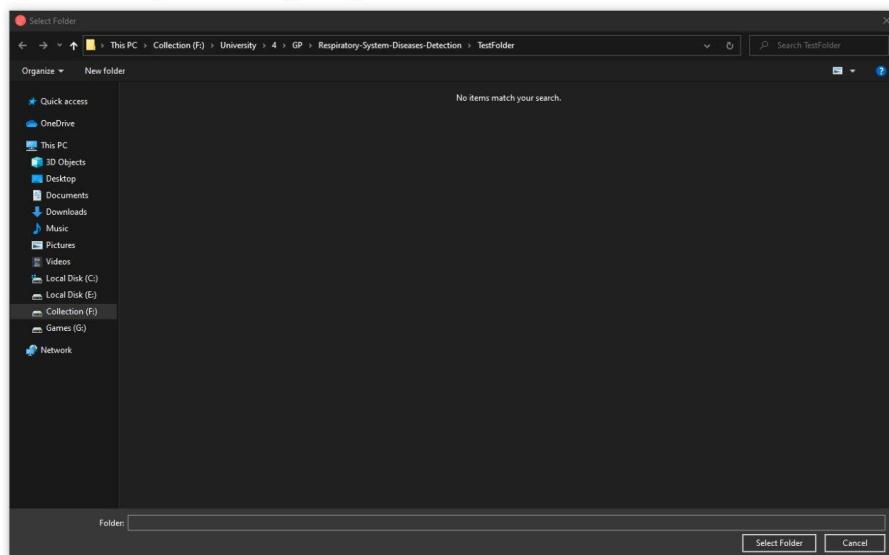
6. Radiologist Guide



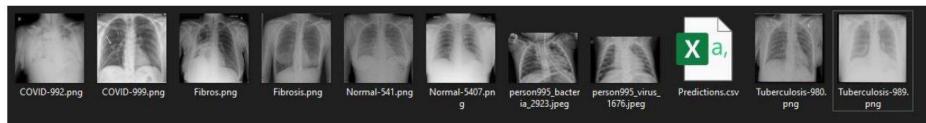
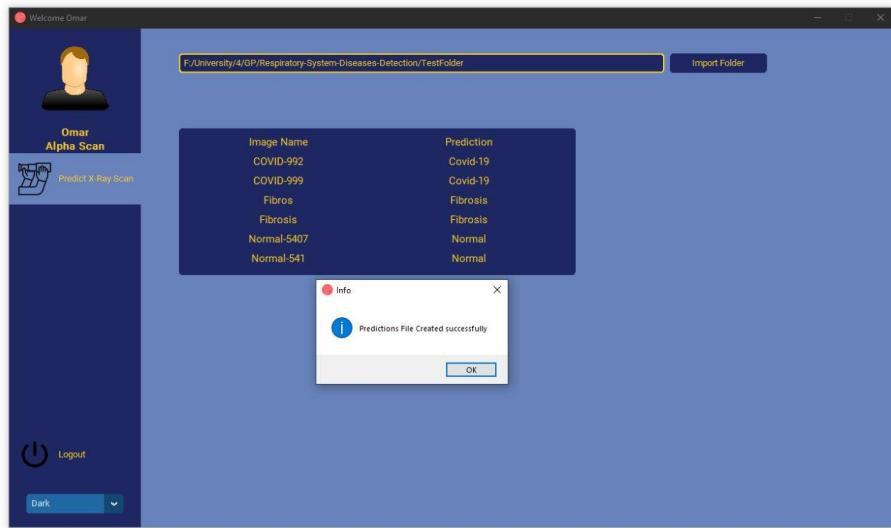
Once you login as a radiologist, the first screen you will encounter displays your name and the radiology center where you work. Additionally, you will find two buttons: "Predict X-Ray Scan" and "Logout".



After clicking on the "Predict X-Ray Scan" button, you will be presented with an "Import Folder" button. Use this button to select the folder that contains the scans you want to predict using the system model.



After clicking the "Import Folder" button, a file dialog will appear. Use this file dialog to navigate to the location where the folder containing the scans you want to predict using the system model.



Once you have selected the folder, the system model will proceed to predict the X-ray scans. The predictions will be displayed, and a CSV file will be created in the same folder. Each row in the CSV file will contain the image name and its corresponding prediction.

6. Conclusion and Future Work

6.1 Conclusion

The system described is a healthcare platform that offers features for patients, doctors, and radiologists. One of the main features available for patients is the ability to predict their chest x-ray scan for five diseases, including COVID-19, fibrosis, bacterial or viral pneumonia, and tuberculosis. This prediction is made using a prediction model that is built into the system. Additionally, patients can consult with a chatbot that is available on the system. The chatbot is designed to identify the patient's health condition by asking them a series of questions. If the patient's condition is deemed severe, the chatbot will advise them to chat with one of the doctors available on the system.

The system utilizes a coin-based system for payment, with patients able to purchase coins to use for various features. Patients can also purchase a VIP level, which reduces the amount of coins required for prediction or consultation. The platform also allows patients to download prescriptions written by doctors.

Doctors on the platform can select patients waiting for assistance, review the chat history with the chatbot to understand the patient's symptoms, and provide prescriptions and medication for the patient's condition. The doctor can also report a patient if they break the system's rules, with the ability to provide a reason for reporting the patient. If a patient is reported, they will be temporarily suspended from the system until an administrator confirms the suspension to be permanent or revokes it. After each chat, the doctor will receive coins that can be exchanged for real money.

The system also offers features for radiologists, who can use the prediction model to predict multiple scans simultaneously. This feature is designed to be efficient and save time for radiologists, as they can process many scans at once without having to manually review each one.

Overall, the system is designed to provide convenient and efficient healthcare services. The chatbot feature allows patients to easily identify their health condition and connect with doctors when necessary, while the prediction model for radiologists offers a time-saving solution for processing multiple scans.

6.2 Future Work

There are several important improvements that can be made to prediction model of the system to enhance its functionality, usability and accessibility. These improvements can have a significant impact on the system's ability to diagnose and treat diseases, and ultimately improve patient outcomes.

One of the key improvements that can be made to the system is to include more diseases in the prediction model. This can be achieved by collecting more data and incorporating it into the system. By doing so, the system will become more accurate and effective in diagnosing a wider range of diseases. The addition of more diseases will also make the system more versatile, allowing it to be used in a variety of healthcare settings.

Another important improvement that can be made to the system is to create a web-based application for the system. This will allow patients and healthcare professionals to access the system from any device with an internet connection. A web-based application will also make it easier to update and maintain the system, as changes can be made centrally and deployed instantly.

A mobile application for the system is another important improvement that can be made. This will allow patients to use the system on-the-go, and healthcare professionals to access the system from their mobile devices. A mobile application will also make it easier to collect data from patients, as they can input their symptoms and other relevant information directly into the application.

Improving the system to help disabled people use it easily is also an important consideration. This can be achieved by incorporating accessibility features such as text-to-speech and voice recognition. By doing so, disabled people can use the system without any hindrances or difficulties, making it more inclusive and accessible.

Another feature that can be added to the system is video chat between the patient and the doctor. This can improve communication between patients and healthcare professionals and make it easier for doctors to diagnose and treat diseases remotely. Video chat can also save patients time and money, as they will not have to travel to a healthcare facility for a consultation.

References

- [1] Singh D, Kumar V, Vaishali, Kaur M (2020) Classification of COVID-19 patients from chest CT images using multi-objective differential evolution based convolutional neural networks. *Eur J Clin Microbiol Infect Dis* 39:1379–1389
- [2] Yan Q, Wang B, Gong D, Luo C, Zhao W, Shen J, Shi Q, Jin S, Zhang L, You Z (2020) COVID-19 Chest CT image segmentation—a deep convolutional neural network solution.
- [3] Shan F, Gao Y, Wang J, Shi W, Shi N, Han M, Xue Z, Shen D, Shi Y (2020) Lung infection quantification of COVID-19 in CT images with deep learning. *arXiv:2003.04655*
- [4] Li L, Qin L, Xu Z, Yin Y, Wang X, Kong B, Bai J, Lu Y, Fang Z, Song Q, Cao K, Liu D, Wang G, Xu Q, Fang X, Zhang S, Xia J, Xia J (2020) Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT. *Radiology*
- [5] Dansana D, Kumar R, Bhattacharjee A, Hemanth DJ, Gupta D, Khanna A, Castillo O (2020) Early diagnosis of COVID-19-affected patients based on X-ray and computed tomography images using deep learning algorithm. *Soft Comput*, pp 1–9
- [6] Ng M-Y, Lee EY, Yang J, Yang F, Li X, Wang H, Lui M-M-S, Lo CS-Y, Leung B, Khong P-L, Hui CK-M, Yuen K-Y, Kuo MD (2020) Imaging prole of the COVID-19 infection radiologic endings and literature review. *Radiol Cardiothorac Image* 2(1):e200034. Accessed 29 Mar 2020
- [7] Huang C et al (2020) Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. *Lancet* 395:497–506. Accessed 29 Mar 2020
- [8] Makris A, Kontopoulos I, Tserpes K (2020) COVID-19 detection from chest X-ray images using deep learning and convolutional neural networks. *medRxiv*. Accessed 29 Mar 2020
- [9] Wang L, Lin ZQ, Wong A (2020) COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. *Scientific Reports*, Vol. 10

- [10] Rahimzadeh M, Attar A (2020) A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2. *Inf Med Unlock* 19:100360
- [11] Apostolopoulos ID, Bessiana T (2020) COVID-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. *Phys Eng Sci Med*, pp 1–6
- [12] Xu X, Jiang X, Ma C, Du P, Li X, Lv S, Y L, Chen Y, Su J, Lang G, Li Y, Zhao H, Xu K, Ruan L, Wu W (2020) Deep learning system to screen Coronavirus disease 2019 pneumonia. *arXiv:2002.09334*
- [13] Jain R, Gupta M, Taneja S, Hemanth DJ (2020) Deep learning based detection and analysis of COVID-19 on chest X-ray images. *Appl Intell*, pp 1–11. Accessed 29 Mar 2020
- [14] Minaee S, Kafieh R, Sonka M, Yazdani S, Soufi GJ (2020) Deep-COVID: predicting COVID-19 from chest X-ray images using deep transfer learning. *Med Image Anal* 65:101794
- [15] Khan AI, Shah J, Bhat M (2020) CoroNet: a deep neural network for detection and diagnosis of COVID-19 from chest x-ray images. *Comput Methods Programs Biomed* 196:105581. Accessed 29 Mar 2020
- [16] Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O, Acharya UR (2020) Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Comput Biol Med* 121:103792
- [17] Mahmud T, Rahman MA, Fattah SA (2020) CovXNet: a multidilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization. *Comput Biol Med* 122:103869
- [18] Duran-Lopez L, Dominguez-Morales JP, Corral-Jaime J, Vicente-Diaz S, Linares-Barranco A (2020) COVID-XNet: a custom deep learning system to diagnose and locate COVID-19 in chest X-ray images. *Appl Sci* 10(16):5683
- [19] Mangal A, Kalia S, Rajgopal H, Rangarajan K, Namboodiri V, Banerjee S, Arora C (2020) COVIDAID: COVID-19 detection using chest X-ray.

- [20] Waheed A, Goyal M, Gupta D, Khanna A, Al-Turjman F, Pinheiro PR (2020) COVIDGAN: data augmentation using auxiliary classifier GAN for improved COVID-19 detection. *IEEE Access* 8:91916–91923
- [21] Tao S (2019) Deep neural network ensembles.
- [22] Frazao X, Alexandre LA (2014) Weighted convolutional neural network ensemble. In: Iberoamerican Congress on pattern recognition, Springer, Cham
- [23] Ju C, Bibaut A, van der Laan M (2018) The relative performance of ensemble methods with deep convolutional neural networks for image classification. *J Appl Stat* 45(15):2800–2818
- [24] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammad Bagheri, Ronald Summers, ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases, IEEE CVPR, pp. 3462-3471, 2017.
- [25] Tawsifur Rahman, Amith Khandakar, Muhammad A. Kadir, Khandaker R. Islam, Khandaker F. Islam, Zaid B. Mahbub, Mohamed Arselene Ayari, Muhammad E. H. Chowdhury. (2020) "Reliable Tuberculosis Detection using Chest X-ray with Deep Learning, Segmentation and Visualization". *IEEE Access*, Vol. 8, pp 191586 - 191601. DOI. 10.1109/ACCESS.2020.3031384.
- [26] M.E.H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M.A. Kadir, Z.B. Mahbub, K.R. Islam, M.S. Khan, A. Iqbal, N. Al-Emadi, M.B.I. Reaz, M. T. Islam, "Can AI help in screening Viral and COVID-19 pneumonia?" *IEEE Access*, Vol. 8, 2020, pp. 132665 - 132676.
- [27] Rahman, T., Khandakar, A., Qiblawey, Y., Tahir, A., Kiranyaz, S., Kashem, S.B.A., Islam, M.T., Maadeed, S.A., Zughaiier, S.M., Khan, M.S. and Chowdhury, M.E., 2020. Exploring the Effect of Image Enhancement Techniques on COVID-19 Detection using Chest X-ray Images. *arXiv preprint arXiv:2012.02238*.
- [28] Kermany, Daniel; Zhang, Kang; Goldbaum, Michael (2018), "Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification", Mendeley Data, V2, doi: 10.17632/rscbjbr9sj.2

- [29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770-778, 2016.
- [30] Sergey Zagoruyko and Nikos Komodakis. "Wide Residual Networks." In Proceedings of the British Machine Vision Conference (BMVC), pages 87.1-87.12, 2016.
- [31] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. "Densely Connected Convolutional Networks." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2261-2269, 2017.
- [32] Xie Saining, Girshick Ross, Dollár Piotr, Tu Zhuowen, and He Kaiming. "Aggregated Residual Transformations for Deep Neural Networks." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1492-1500, 2017.
- [33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Identity Mappings in Deep Residual Networks." In Proceedings of the European Conference on Computer Vision (ECCV), pages 630-645, 2016.
- [34] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. "AutoAugment: Learning Augmentation Strategies From Data." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 113-123, 2019.