# Pneumonia Detection System Using Deep Learning

**Abdul Rahman Bin Salam [1], Ibaad Mohammed Hameeduddin [2], Mohammed Faizan Hussain [3], Hajira Sabuhi [4]**

[1, 2, 3] B.E. Student, [4] Assistant Professor,
Department of IT, Lords Institute of Engineering and Technology,
Hyderabad, Telangana, India.

Published in IJIRMPS (E-ISSN: 2349-7300), Volume 11, Issue 1, (January-February 2023)

## Abstract

Artificial intelligence and machine learning are increasingly being applied in medicine, particularly in biomedical imaging and diagnostic procedures. Machine learning algorithms are being used to process chest X-ray images, enhancing consistency and accuracy in reporting. The research focuses on using deep learning algorithms based on convolutional neural networks to build a processing model for detecting pneumonia-related changes in chest X-rays and classifying them into two groups based on detection results. This approach aims to improve decision-making and accuracy in medical imaging.

## Introduction

Artificial Intelligence (AI) gained prominence in the 1950s but was initially limited due to its limited practical feasibility. However, the development of AI has since accelerated due to the availability of processing power and the emergence of big data. The "AI Winter" period, from the 1970s to the 2000s, was a significant period for AI development. The rise of AI began with IBM's Deep Blue chess program, which beat world champion Gary Kasparov in 1997. AI has since developed new fields like Machine Learning (ML) and Deep Learning (Deep Learning), which require human input to classify data independently.

Deep Learning, based on Google's AlphaGo program, made a significant comeback in 2016 with its ability to beat the world champion in the board game Go. AI and Deep Learning are now used in various medical fields, such as gastroenterology, radiology, cardiology, and endoscopy. AI and ML are often used in fields that build large databases of medical data for training models. This paper focuses on pneumonia and the use of CNN-based algorithms to process chest X-ray images. Pneumonia is a leading cause of child mortality in developing countries, and X-rays are crucial for diagnosing respiratory disorders and diseases.

## Input Design

It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system.

https://doi.org/10.37082/IJIRMPS.v11.i1.230314

The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple.

The input is designed in such a way so that it provides security and ease of use with retaining the privacy.
The dialog to guide the operating personnel in providing input.
Methods for preparing input validations and steps to follow when error occur.

This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. When the data is entered it will check for its validity. It also provides record viewing facilities. Appropriate messages are provided as when needed so that the user will not be in maize of instant.

## Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly.
It is the most important and direct source information to the user.
Efficient and intelligent output design improves the system's relationship to help user decision-making.
Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively.
When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
Select methods for presenting information.
Crate document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives:

- Convey information about past activities, current status or projections of the future.
- Signal important events, opportunities, problems, or warnings.

## System Analysis

### 1. Existing System

In existing system, we have used machine learning algorithms those are SVM and random-forest. X-ray image-based diagnosis is not possible in existing system. Using these algorithms, it is time consuming process and less efficient.

### 1.1. Disadvantages of Existing System

- Image-based medical diagnosis is not possible.
- Using Machine Learning algorithms, we can predict proper accuracy.
- Not detected in early stages.

### 1.2. Algorithms Used

**SVM**

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

**Random-Forest**

Efficient and intelligent output design improves the system's relationship to help user decision-making.
Create document, report, or other formats that contain information produced by the system.
The output form of an information system should accomplish one or more of the following objectives.
Convey information about past activities, current status or projections of the future.
It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.
As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."
Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.
The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

## 2. Proposed System

In this project, we have focused on pneumonia and the use of CNN based algorithm to process chest X-ray images.
AI and ML are usually used in the fields that have the ability to build a large database of medical data, typically in a form of digital images, that can be used later for training models.
The use of AI in support tools for processing medical images has been suggested in order to improve accuracy and consistency, and time efficiency in reporting.
Before the analysis all the bad quality X-rays have been removed by the experts at the Medical Centre.
The dataset contains images of chest X-rays (JPEG).
It is divided into three folders, named train, Val and test, that are used as training, validation and testing data.
The image classification was done with the use of a CNN based machine learning algorithm.
The CNN is a class of deep learning neural networks.

### 2.1. Advantages of Proposed System

- Image-based medical diagnosis is possible the user has to give the X-ray image to the model, then our model will predict where the person viral infected or not.
- It has many features such as simple structure, less training parameters and adaptability.

### 2.2. Algorithms Used

A convolutional neural network (CNN or ConvNet) is a subset of machine learning.

A CNN is a kind of network architecture for deep learning algorithms and is specifically used for image recognition and tasks that involve the processing of pixel data.

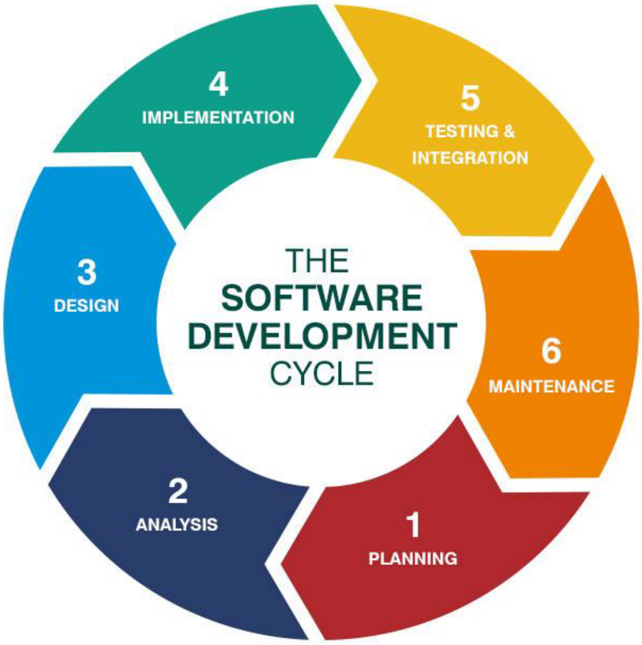
There are other types of neural networks in deep learning, but for identifying and recognizing objects, CNNs are the network architecture of choice.

The CNN is another type of neural network that can uncover key information in both time series and image data.

To identify patterns within an image, a CNN leverages principles from linear algebra, such as matrix multiplication.

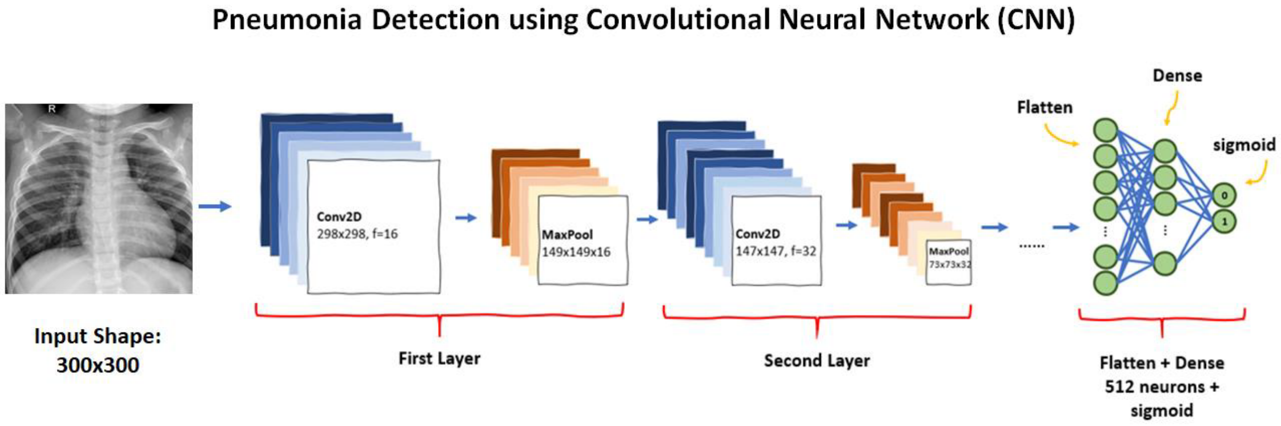
### 3. Process Model Used with Justification

Figure: The Software Development Life Cycle



### 4. System Design

### 4.1. System Architecture



### 4.2. Data Flow Diagram

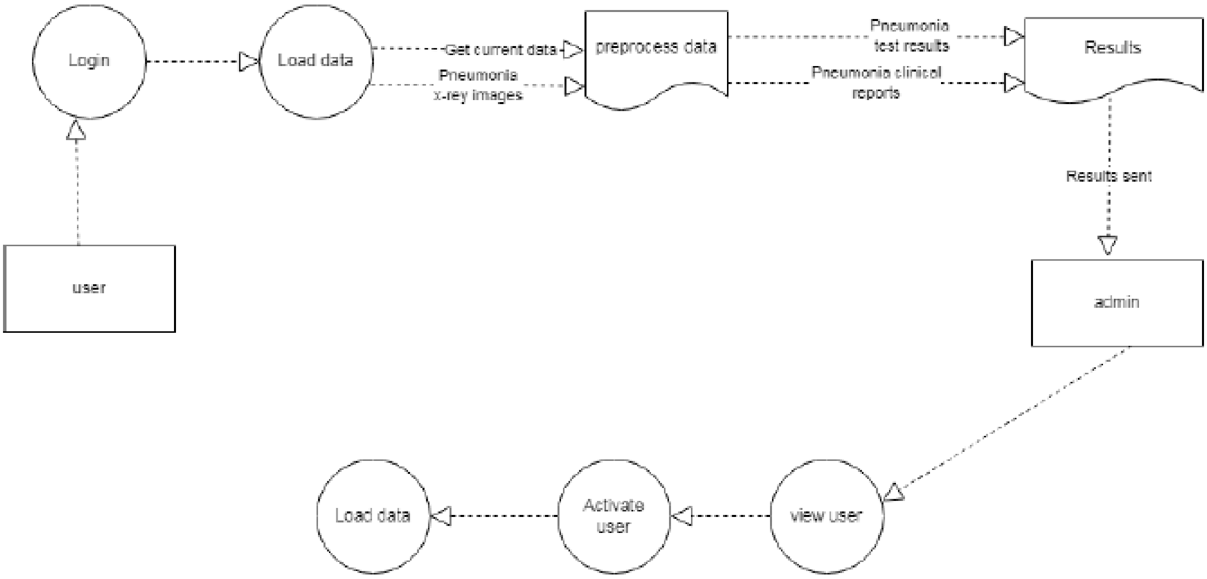
Data Flow Diagrams (DFDs) are one of the most important modeling tools.

DFD shows how information moves in the system and how information is changed by a series of transformations.

It is a graphical technique that depicts the flow of information and the transformations applied as data moves from input to output.

DFDs can be broken down into levels that represent an ever-increasing flow of information and functional details.

Figure: Data Flow Diagram



### 4.3. UML Diagrams

UML is a standardized modeling language for general use in the field of object-oriented software engineering.

The goal is for UML to become a common language for creating object-oriented computer software models.

In its current form, UML is made up of two main components: a meta-model and a symbol.

UML represents a set of proven engineering best practices for modeling large and complex systems.

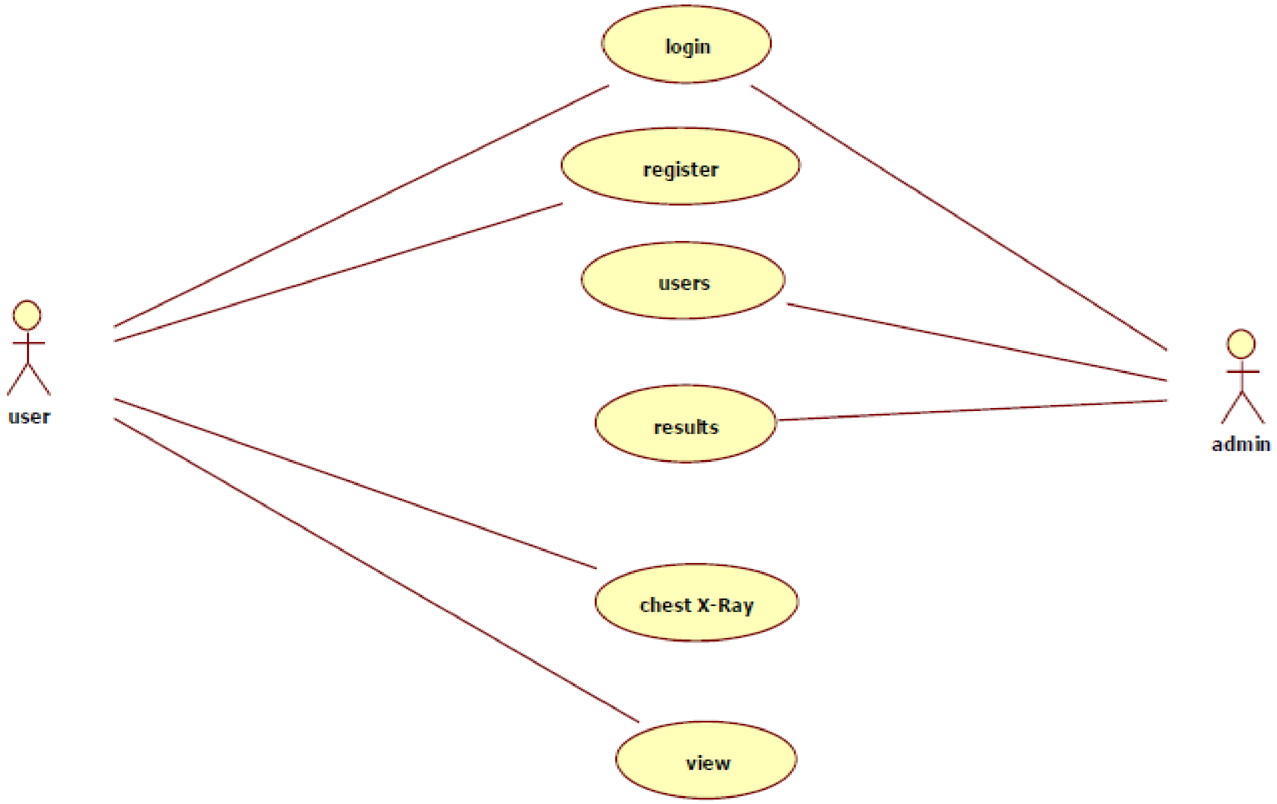
UML mainly uses graphical symbols to represent the design of software projects. Provide users with a superior expressive visual modeling language so they can develop and communicate meaningful models.

Provides extensibility and specialization mechanisms to extend core concepts.

### 4.4. Use-case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavior diagram that is defined and generated from a use case analysis. Its purpose is to present a graphical overview of the functionality provided by a system of the actors, their goals (represented as use cases), and any dependencies between them. these use cases. The main purpose of a use case diagram is to show which system functions are performed for which actors. The role of system actors can be represented.

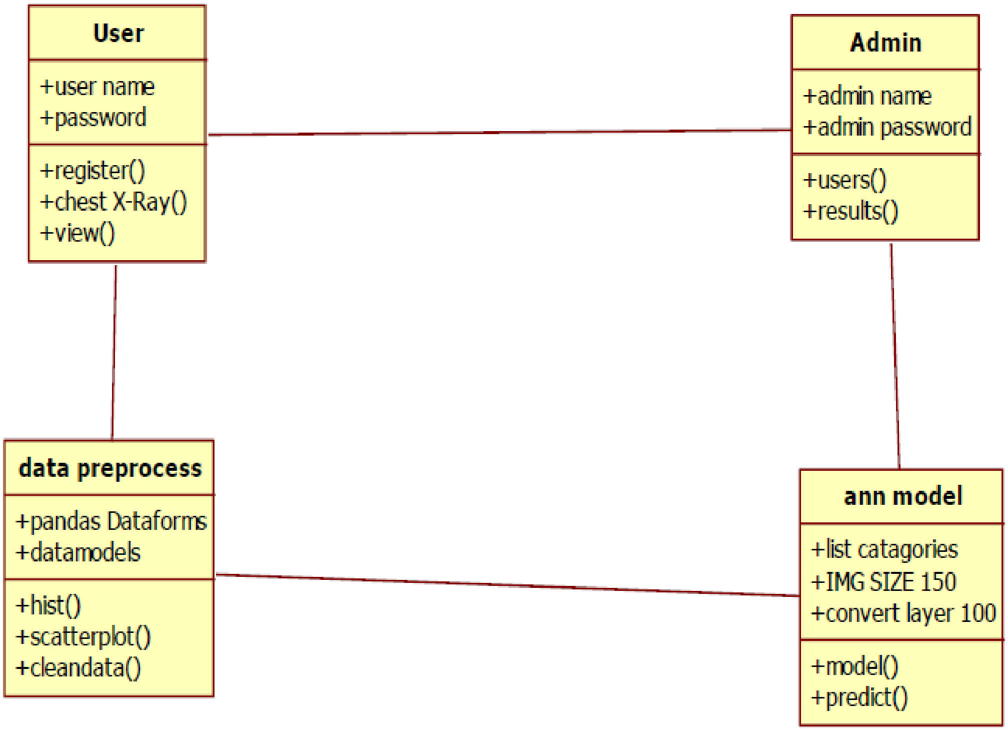
Figure: Use-case Diagram



### 4.5. Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, properties, operations, etc. their (or methods) and the relationships between them. It explains which class contains the information.

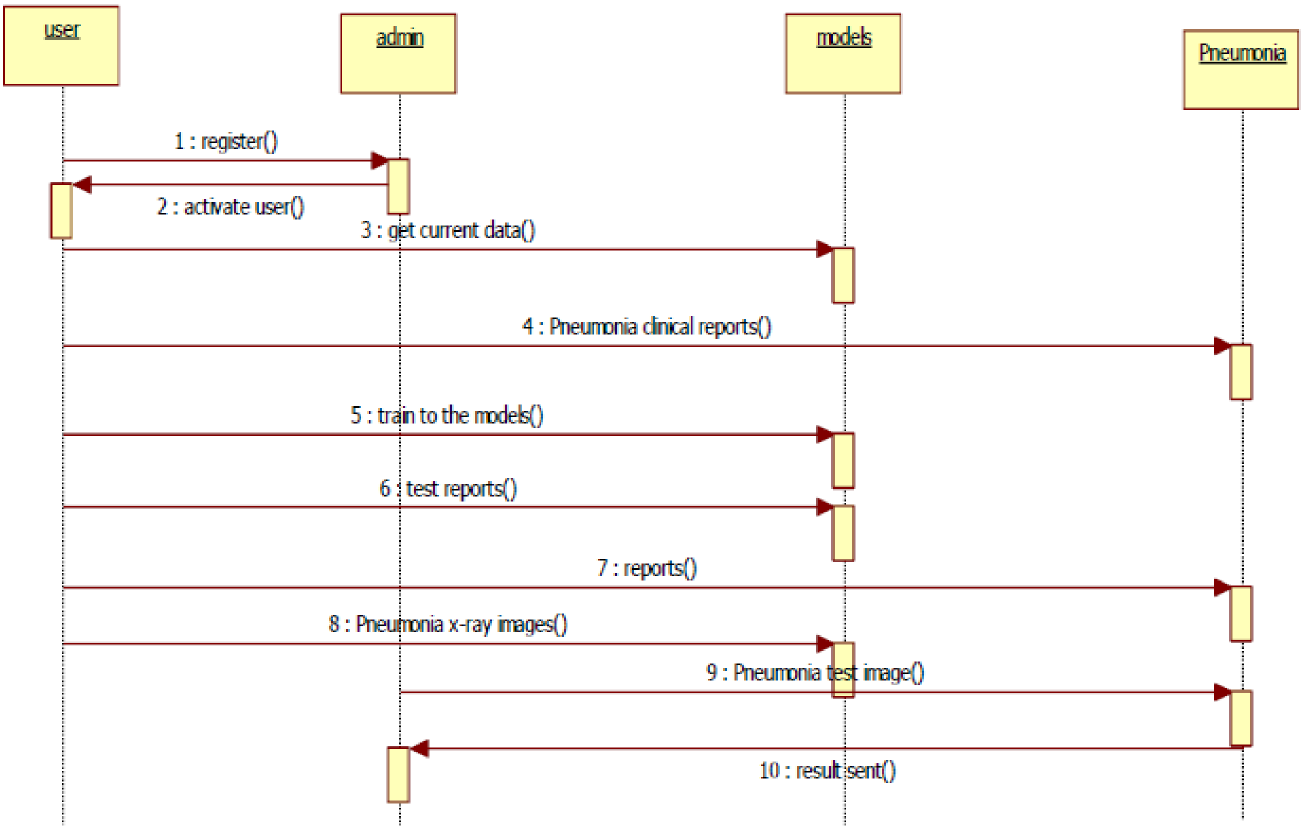
Figure: Class Diagram



### 4.6. Sequence Diagram

A sequence diagram in the Unified Modeling Language (UML) is a type of interaction diagram that shows how and in what order processes work together. It is a structure of the message sequence diagram. Sequence diagrams are sometimes called event, event scenario, and timeline diagrams.

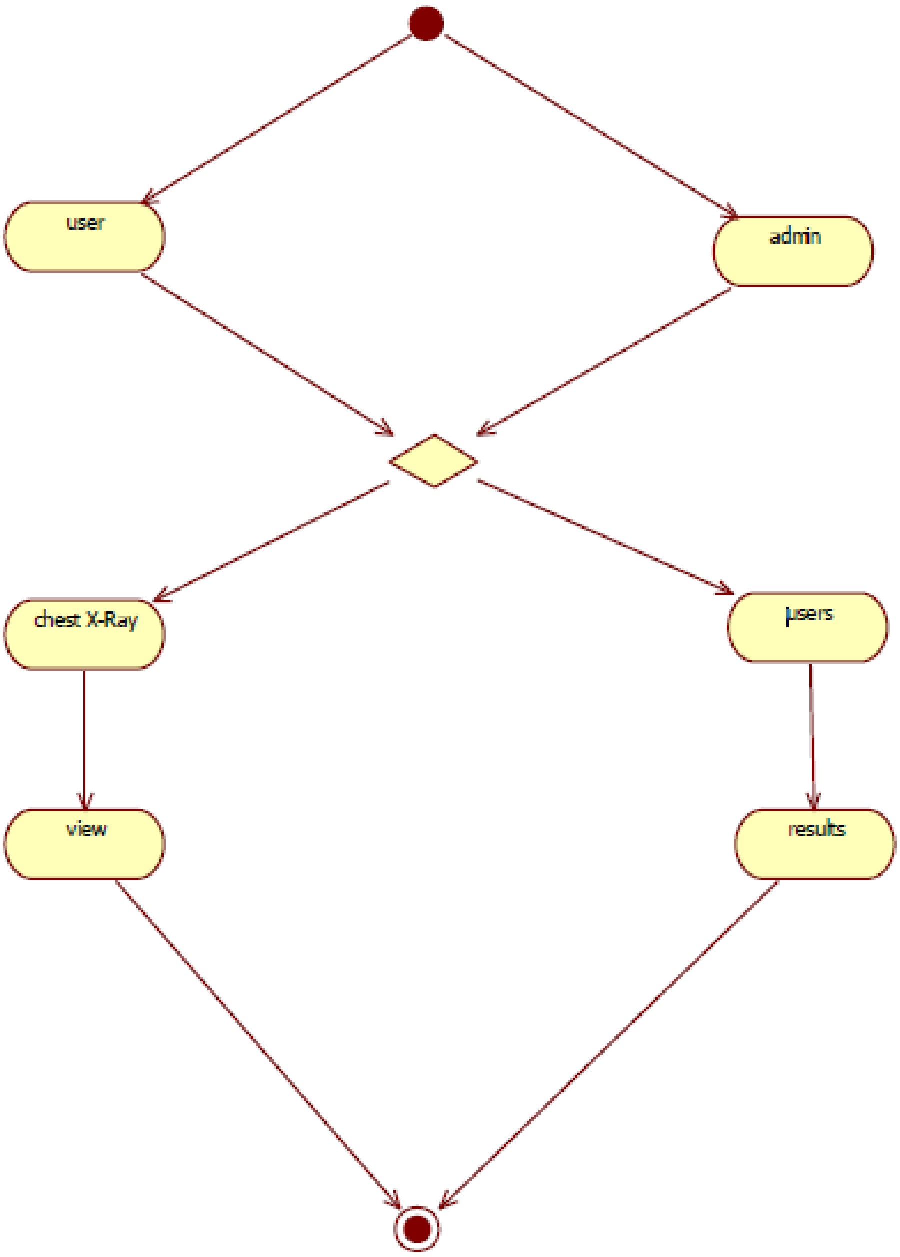
Figure: Sequence Diagram



### 4.7. Activity Diagram

An activity diagram is a graphical representation of a step-by-step operation and action process with support for selection, iteration, and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe step-by-step the workflow and business operations of a system's components. An activity chart showing the overall flow of control.

Figure: Activity Diagram



## 5. Implementation

### 5.1. Python

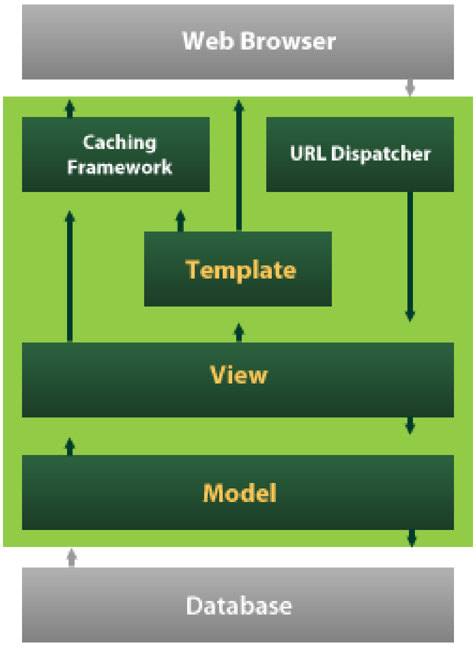
Python is a high-level, universally interpreted, interactive, object-oriented programming language. As an interpreted language, Python has a design philosophy that prioritizes code readability (specifically, using whitespace indentation rather than curly braces or keywords to delimit blocks of code) and a way for programmers to avoid traditional There are syntaxes that allow concepts to be expressed in fewer lines of code. Languages such as C++ and Java. It provides a structure that allows for clean programming both on the small and large scale. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and, like nearly all variant implementations, has a community-based development model. CPython is maintained by the non-commercial Python Software Foundation. Python has a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional, and procedural, and has a large and comprehensive standard library.

### 5.2. Django

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes a lot of your web development effort and lets you focus on building your app without reinventing the wheel. It's free and open source.

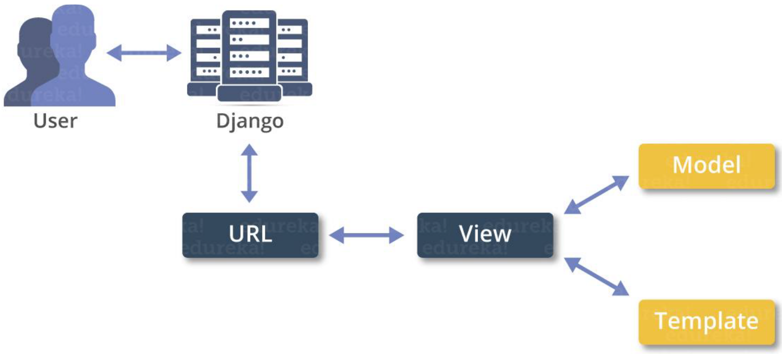
Django's main goal is to simplify the creation of complex database-driven he websites. Django emphasizes reusability and pluggability of components, rapid development, and the "never repeat" principle. Python is used throughout, including configuration files and data models.

Figure: Django Framework



Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Figure: Working of Django



## 6. System Testing

The purpose of testing is to find bugs. Testing is the process of trying to discover all possible flaws and weaknesses in a product. It provides a way to verify the functionality of components, subassemblies, assemblies and finished goods. It is a software application process aimed at ensuring that a software system meets its requirements and user expectations and does not fail unacceptably. There are different types of tests. Each test type addresses specific test requirements.

### 6.1. Unit Tests

Unit testing is the design of test cases that verify that the internal program logic is working correctly and that the program inputs produce valid outputs. All decision making branches and internal code flow should be verified. This includes testing individual software units of the application. This occurs upon completion of a single unit prior to consolidation. This is a structural test based on structural knowledge and is invasive. Unit tests perform basic component-level testing to test specific business processes, applications, and system configurations. Unit testing ensures that all paths in a business process adhere exactly to a documented specification, with well-defined inputs and expected results.

### 6.1.1. Integration Test

Integration tests are designed to test built-in software components to determine whether they actually run as programs. Tests are event-driven and deal more with underlying screen or field results. Integration testing shows that the combination of components is correct and consistent, while individually satisfying, as indicated by successful unit tests. Integration tests are specifically aimed at uncovering problems that arise from combining components.

### 6.1.2. Functional Test

Functional testing systematically demonstrates that the tested functionality is available according to business and technical requirements, system documentation, and user manuals.

Functional testing focuses on:
**Valid Input:** The identified class of valid input should be accepted.
**Invalid Input:** Identified invalid input classes MUST be rejected.
**Feature:** It must perform the specified function.
**Exit:** The identified class of application problems should be addressed.
**System / Procedure:** You need to call an interface system or procedure.

Functional test configuration and preparation focuses on requirements, key features, or special test cases. Additionally, systematic coverage to identify business process flows. Data fields, pre-defined processes, and follow-up processes should be considered when testing. Before functional testing is complete, additional tests are identified and effective values for the current tests are determined.

#### 6.1.3. System Test

System testing ensures that the overall integrated software system meets requirements. Test your configuration to see known and predictable results. An example of system testing is configuration-oriented system integration testing. System testing is based on process descriptions and flows, with a focus on pre-driven process links and integration points.

#### 6.1.4. White-box Testing

White-box testing is testing in which the software tester has knowledge of the inner workings, structure, language, or at least its purpose of the software. its purpose. Used to check areas not reachable from the blackbox level.

#### 6.1.5. Black-box Testing

Black-box testing tests software without requiring knowledge of the inner workings, structure, or language of the module under test. Black box tests, like most other types of tests, should be created from final source documents such as specifications and requirements documents. A test that treats the software under test like a black box that you cannot "peek into".

### 6.2. Field Tests

#### 6.2.1. Test Strategy and Approach

Field testing will be performed manually and functional tests will be written in detail.

#### 6.2.2. Purpose of Testing

All field inputs should work correctly.

Page must be activated via the identified link

- Input masks, messages, and responses must not be delayed.

#### 6.2.3. Functionality under Test

- Make sure your entries are in the correct format.
- Duplicate entries should not be allowed
- All links should direct the user to the correct page.

### 6.3. Integration Test

Software integration testing is incremental integration testing of two or more integrated software components on a single platform, generating errors caused by interface errors.

The task of integration testing is to verify that components or software applications (for example, components of a software system, or even enterprise-level software applications) interact correctly.

### Test Results

All the above test cases passed successfully. There were no flaws.

### 6.4. Acceptance Test

User acceptance testing is a critical stage of any project and requires significant end-user involvement. It also ensures that the system meets functional requirements.

### Test Results

All the above test cases passed successfully. There were no flaws.

## Conclusion and Future Improvements

### 1. Conclusion

This article describes the use of deep learning to classify digital images of chest radiographs for the presence or absence of pneumonia-related changes. The implementation was based on the CNN model. Initial experiments have shown promising results, but further investigation is needed. A relatively high model accuracy of almost 90% is expected, but there is a possibility of overfitting due to the size of the dataset. 90°Curacy also means that predictive models can potentially be used as decision support tools, but there is still work to be done. Proper diagnosis of any kind of disease still requires the involvement and presence of a specialist physician. Collecting as much data as possible is critical to creating good and reliable disease classification models.

### 2. Future Improvements

Collecting as much data as possible is critical to creating good and reliable disease classification models. Further research steps include experimenting with different preprocessing and CNN configurations, data augmentation techniques, and using additional X-ray datasets with additional data labels indicative of other pathologies.

## References

[1] M. Haenlein, A. Kaplan, “A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence”, California Management Review, 61(4), 2019, pp 5-14.

[2] V. Kaul, S. Enslin, S.A. Gross, “The history of artificial intelligence in medicine”, Gastrointestinal Endoscopy, In Press, 2020.

[3] M. Minsky, S.A. Papert, “Perceptrons: An Introduction to Computational Geometry”, MIT Press, Cambridge, MA, 1969.

[4] A. Esteva et al., “A guide to deep learning in healthcare”, Nature Medicine, 25(1), 2019, pp 24-29.

[5] D. Ravi et al., "Deep Learning for Health Informatics", IEEE Journal of Biomedical and Health Informatics, 21(1), 2017, pp. 4-21.

[6] Y.J. Yang, C.S. Bang, “Application of artificial intelligence in gastroenterology”, World Journal of Gastroenterolology, 2019(25), 2019, pp. 1666-1683.

[7] A. Hosny, C. Parmar, J. Quackenbush, L.H. Schwartz, H.J.W.L. Aerts, “Artificial intelligence in radiology”, Nature Reviews Cancer, 18(8), 2018, pp. 500-510.

[8] K.W. Johnson, et al., “Artificial Intelligence in Cardiology”, Journal of the American College of Cardiology, 71(23), 2018, pp. 2668-2679.

[9] V.N. Perisic, B. Jankovic, “Pedrijatrija za studente medicine”, Medicinski fakultet, Univerziteta u Beogradu, Beograd, 2010.

[10] N. Chen, et al., “Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in Wuhan, China: a descriptive study”, The Lancet, 395(10223), 2020, pp. 507-513.

[11] D. Kermany, K. Zhang, M. Goldbaum, “Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification”, Mendeley Data, v2, 2018. [Online Accessed On: July 2020.