UNIVERSITY PARTNER

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

Project and Professionalism

(6CS007)

A1: Project Report

< COVID-X: The COVID-19 detection system using Convolutional Neural Network (CNN)>

Student Id	: 2039208
Student Name	: Roshan Shrestha
Supervisor	: Mr. Dinesh Saud
Reader	: Mr. Rupak Koirala
Cohort/Batch	: 4

Submitted on : May 6, 2021

Declaration Sheet

Acknowledgement

I am really thankful for our supervisor Mr. Dinesh Saud for his helpful behavior and to the point feedbacks. Thanks to reader Mr. Rupak Koirala for his guidance. This project would not be possible without their help. Also, I would like to acknowledge my parents for supporting me and providing all the necessary essential/working environment.

Lastly, I really appreciate the amazing study environment provided by herald college and University of Wolverhampton for us students even in this global pandemic.

Abstract

Covid 19 was a deadly virus and is declared as a global pandemic in 2020. Hundreds of thousand people lost their lives due to this virus. There is still no sign of getting proper vaccination for people in developing countries like Nepal. People from all over the world is trying to overcome this problem using every means necessary. This project is a tiny effort to help people relieve from this deadly virus using deep learning. Covidx is a web app developed with the aim of detecting either the given x-ray image is infected or not. It classifies if the CXR image is “positive” or “negative”. Here the comparison of 3 different model based on custom architecture inspired by VGG16, ResNet50 and LeNet were illustrated. Best model with highest accuracy, f1-score and other performance matrix was integrated in a python django model.

Along with detecting COVID this web app also helps to digitalize the traditional approach of providing handwritten prescription by medical staffs. This report gives the detail information used to complete this project including its aims and objectives, methodology used, language, library, software, model comparison etc.

Contents

1. Introduction

1.1. Academic Question

How does this web app help medical staffs to diagnose COVID-19 along with the digitalization of prescription provided by them?

1.2. Aims

- To digitalize health care facility centers.
- To find most suitable architecture for covid image classification
- To build an automated COVID-19 detection system which is cheaper and more reliable than current tools used for diagnosis like (RT-PCR test and CT scans).

1.3. Objectives

- To digitalize the traditional approach of providing handwritten prescription by doctors.
- To prepare a User Management System to digitally manage doctor, receptionist and patient.
- To compare between different image classification architecture
- To implement an AI model that can accurately diagnose COVID-19 from CXR in a web app.

1.4. Introduction of the topic

COVID-X is a project that has two major aspects: detecting COVID-19 using chest X-Ray (CXR) images and digital platform for managing prescription provided by doctors. In this study, we build a web app that classifies the provided chest X-Ray images accordingly ('+ve' or '-ve'). Also, this web app helps medical staffs to digitally manage patients, doctors and the prescription provided by doctors. We can think of this app as a medical prescription management app which also has a feature to detect one of the deadliest viruses of decade "COVID-19".

COVID-19 is a virus which causes lung inflammation filled with fluids that reduces their ability to extract oxygen from the air we breathe. We need trained radiologist to diagnose the COVID-19 lung inflammation from CXR. However, we can create computer-aided diagnosis system using CNN which does not require knowledge of radiology for diagnosis. CNN is an artificial neural network used in Deep Learning for image processing and pattern recognition (technopedia , 2018). Even if a substantial number of COVID-19 patients are affected globally, the number of widely accessible online chest X-ray images is limited and scattered. So, to gain more accurate output with less data we use a method we can build our custom CNN architecture based on popular and well recognized network like ResNet, VGG, Inception etc and compare them to find the best model for detecting classifying COVID. (Donges, 2020)

1.5. Problem Domain and Solution

The deadly virus COVID-19 which was discovered in late 2019 was declared its outbreak a global pandemic by World Health Organization (WHO) in March of 2020. (MFMER, 2020) . The virus will move across the respiratory system and into the lungs of an individual. This induces inflammation which can fill with fluid in the air sacs, or alveoli. The development then restricts the capacity of a human to take in oxygen which in turn may lead to death (Healthline Media UK Ltd, 2020).

We can see that most of the medicals and health sectors in Nepal is following a traditional approach of taking prescription using pen and paper. If in case the doctor wants to checks one's medical history it has a really good chance that the patient has already lost the handwritten prescription given by doctor. But on storing those prescription in database, medical experts can easily recall the medical history of patient simply by searching the name of patient.

Currently, the go-to diagnostic test method for COVID-19 is RT-PCR test which uses specimens collected from the patient's upper and lower respiratory systems. Nasal swab is required to perform this test (Roston, 2020). However, due to the number of increasing cases daily, limited amount of test kits available and delayed test results. Using chest computed tomography (CT) scans or chest X-Rays (CXR) to diagnose COVID-19 is an excellent alternative diagnostic measure to prevent this virus from spreading (University, 2020). CXR are cheaper, reliable, more accessible and less harmful process than CT scans. CXR which are used to diagnose COVID-19 need expert radiotherapists for evaluation. Since, the virus is so new there are only limited amount of radiologist out there who can diagnose COVID using CXR especially in a developing country and remote areas (Dimpy Varshni, 2020). So, with the help of automated covid detection system one can diagnose covid without the help of expert radiotherapists.

1.6. Dataset

One of the major issues while building computer-based diagnose system is finding a considerable amount of dataset. These issues occur due to: (i) COVID-19 is a very new virus which makes it harder to find large number of CXR images of COVID positive patients. and (ii) Since medical information of an individuals are so vulnerable medical staffs simply does not want to share these kinds of information. Because of the small number of X-Ray samples available.

Thanks to Kaggle and its community, we found considerable amount of covid related dataset required for covid classification. We used database of Covid-19 RADIOGRAPHY DATABASE (Winner of the COVID-19 Dataset Award by Kaggle Community) for this project.

2. Literature Review

2.1. Potential End-User

As this web app is designed for medical staffs to automate prescription management and COVID-19 detection. So, the primary end-users of this applications are both medical professionals and amateurs.

2.2. Review on Similar Research

2.2.1. COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images

COVID-Net is a computer-aided diagnose system which was proposed by Linda Wang, Zhong Qiu Lin and Alexander Wong. This system classifies the given input into 3 categories i.e. Normal, Non-Covid19 and covid-19. It compares two architecture

(VGG19 and ResNet-50) with COVID-Net. COVID-Net and ResNet-50 are based on residual design pattern whereas VGG-19 is not. COVID-Net architecture is based on lightweight projection-expansion-projection-extension (PEPX) design pattern in which projection and expansion has 1x1 convolutions along with 3x3 depth-wise convolutions to minimize computational complexity. The 1x1 conv layers are used as a form of dimension reduction and restoration. The output is then flattened into fully connected layer and SoftMax activation layer as shown in Figure 1. COVID-Net is comprised of different convolutional layer with diversity of kernel (or filter) varying from (7x7 to 1x1). Unlike VGG-19 similar 1x1 conv layers are also used in ResNet50 (as bottleneck pattern). The COVID-Net was first pre-trained on ImageNet dataset (1.2 million images) and then on the collected data using Adam optimizer. The hyperparameters used in training are learning rate 2e-4, number of epochs = 22 and batch-size = 64. For pre-processing data augmentation techniques like translation, rotation, horizontal flip, zoom and intensity shift were used. The datasets used in this paper were combined from five open access data repositories which consists of 358 CXR images from 266 COVID-19 patients, 8066 images of normal patients and 5,538 images of non-COVID-19 pneumonia patients. After the completion of train, validate and testing process, Accuracy of COVID-Net and ResNet-50 were really high with 93.3% and 90.6% whereas VGG-19 lacks in accuracy percentage with 83% (Linda Wang, 2020).

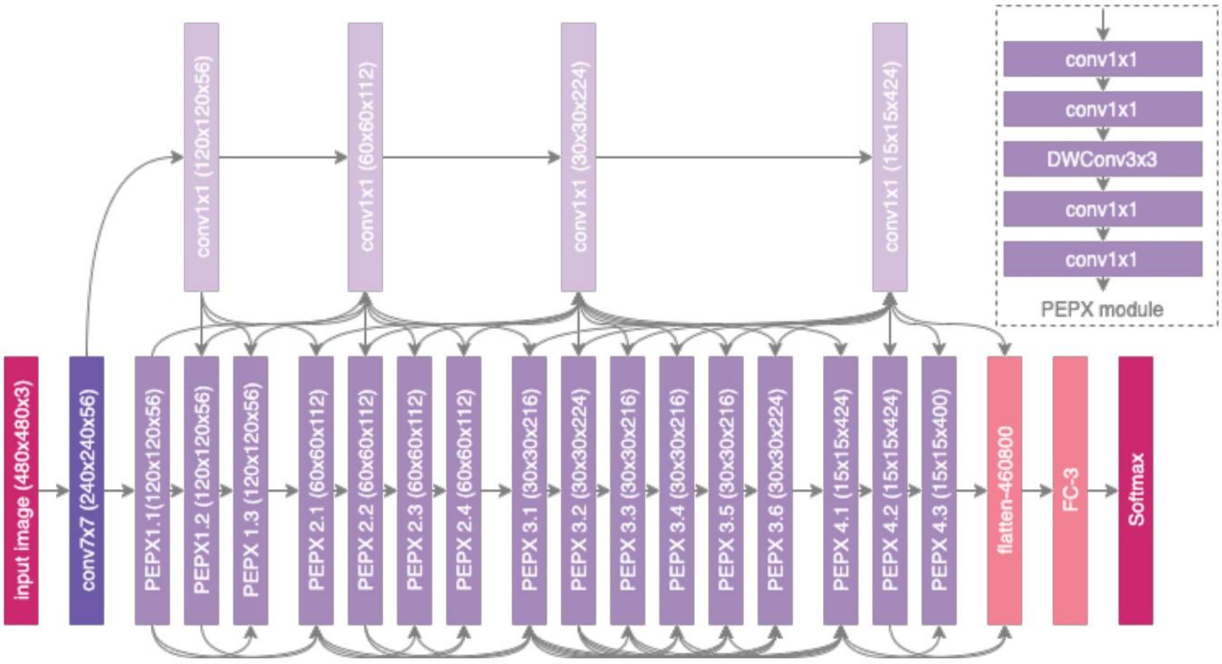


Figure 1 COVID-Net Architecture with heavy use of PEPX design pattern

Analysis:

From the paper we can see that skip connection/residual and 1x1 conv layer architectural pattern can make a significant improvement in our models. Also, the idea of classifying viral pneumonia along with COVID-19 and Non-pneumonia can also be implemented on my model. There were some new augmentation techniques like translation and intensity shift used in this paper which could be helpful for increasing amount of data and avoid overfitting. ResNet50 performed great in this paper also with 90% accuracy. In the case of ResNet50, since the dataset is small in size it would have been great if they could have also tried leaving more layers frozen to avoid overfitting or in extreme cases tried to freeze whole convolutional base due to very small dataset. We might get more accurate result if the ResNet50 is finetuned with manually tweaking hyper parameters like Learning Rates.

2.2.2. COVID-ResNet: A Deep Learning Framework for Screening of COVID19 from Radiographs

COVID-ResNet is a computer-aided diagnose system which was proposed by Muhammad Farooq and Abdul Hafeez. This system classifies the given input into 4 different classes i.e. bacterial, COVID-19, Normal and Viral Pneumonia. The datasets used in this system is same as that of COVID-Net in section 2.2.1. They were able to build a system with an insane rate of sensitivity, Positive Predictive Value and F-1 score of 100% for COVID-19 with accuracy percentage of 96.23% and the overall accuracy of the model for all the four classes with positive predictive values is above 90% among all other classes too. For augmentation, only training data were transformed using vertical flip, rotation with max 15-degree angle and changing lighting condition. In case of manually changing the learning rate, they use the Cyclical learning rate technique. It is an approach to LR adjustments where the value is cycled between a lower bound and upper bound. In case of training, First the head of ResNet50 was replaced with average/max pooling layer, batch-normalization (Batch normalization is a method of normalization which normalizes activations across the mini-batch in a network), dropout and linear layers. Then the training was taken place for 3 different stages (i) first input images were sized into 128x128x3 pixels and is trained for 3 epochs with le-3 then for other 5 epochs the model is again fine-tuned using discriminative learning rate: it is a technique in which instead of using the same learning rate on all model layers, discriminatory fine-tuning helps one to set separate learning rates for each layer. (Jeremy Howard, 2018).(ii) secondly, image was resized to 224x224x3 and is trained for 3 epochs with le-4 then for other 5 epochs with discriminative LR. (iii) thirdly, image was resized to 229x229x3 for 25 epochs with LR between le-6 an le-4 for earliest and last layer respectively. Between these layers the LR was equally distributed between le-6 and le-4 (Hafeez, 2020).

Analysis:

The primary strength of this method is its positive predictive value, in fact they were able to predict 100% of the COVID-19 patients correctly while testing. There are various techniques we could use from this paper for getting an optimum result to our model. The concept of batch-normalization and fine-tuning hyper parameter using discriminative learning rate and Cyclical learning technique can make a huge impact while training. Constantly changing hyper parameter and dropping out neurons will help to decrease problems like underfitting and overfitting. For pre-processing, augmenting data by changing the light condition would be very helpful for small amount of dataset. Also, changing the size of the model after certain epochs would also help in increasing positive predictive value.

2.2.3. Automatic COVID-19 Detection from X-Ray images using Ensemble Learning with CNN

Group of researchers from India Mr. Amit Kumar Das, Sayantani Ghosh, Samiruddin Thunder, Rohit Dutta, Sachin Agarwal And Amlan Chakrabarti proposed a method of detecting COVID following the architecture. The detection model in this project is composed of 3 different model (DenseNet201, ResNet50 and InceptionV3). DenseNet takes lesser parameters than any other traditional CNN approach, ResNet is excellent for addressing one of the most common problem while training called vanishing and exploding gradient by technique called skip connection and Inception v3 is well-known for feature extraction which has 11 inception modules consisting convolution filters, Pooling Layers and ReLU as activation function. At first, all of the model is trained individually to make independent prediction. After then they are combined together using the method called weighted average assembling technique. A modeling averaging ensemble combines the prediction from each model equally. The models were trained on using total of 771 images where 438 images are of COVID-positive patients and 333 images are of COVID-negative patients. Random data of 20% from total images are used for testing purpose. The image trained in each model is resized to 224X224 and all models have been trained for 60 epochs using Adam optimizer with initial learning rate of 0.0001, $\beta1=0.9$ and $\beta2=0.999$. On training all the three models they have come to the conclusion that ResNet50 has the most effective result with low validation error than DenseNet201 and InceptionV3. The accuracy percentage of ResNet50 was 95.3% with 98% sensitivity whereas the accuracy of combined model is 95.7% with sensitivity of 98% which is almost identical to that of ResNet50. The same datasets and optimizer were used for ensemble techniques which has somewhat similar results with that of individual models (Mr. Amit Kumar Das, 2020).

Analysis:

The major takeaway from this paper is that (i) ResNet50 architecture work better in comparison with any other models mentioned above, (ii) We can calculate the performance metrics using provided method. (iv) seems like ensemble techniques does not make much of difference in final results. Also, there were a lot of steps they could have followed to get more optimum results. For example, augmenting using common techniques (flip, crop, rotate and change perspective) to the existing datasets would be helpful. There were also small numbers of data sets for individual networks with millions of parameters, leading to issues such as increasing architectural complexity, computational complexity and overfitting. So, to overcome this issue, we could simply fine tune the hyper parameters or freeze the networks accordingly.

2.2.4. Automatic Detection of Coronavirus Disease (COVID-19) Using X-ray Images and Deep Convolutional Neural Networks

Recently, a group of researcher Ali Narin, Ceren Kaya and Ziynet Pamuk (2020) proposed a CNN based models using chest X-ray radiographs. This system is also based on three different pre-trained model. They are ResNet50, InceptionV3 and Inception-ResNetV2. Same as above system these models are also first trained independently and the performance result was 98% for ResNet50, 97% for InsceptionV3 and 87% accuracy for Inception-ResNetV2. 98% accuracy with datasets of only 50 COVID-19 and 50 Normal patients CXR images is pretty good. The image trained in each model is resized to 299X299 and is trained for 30 epochs to avoid overfitting with random initialization weights using Adam optimizer with learning rate of le-5 and batch-size of 2. Datasets were randomly distributed at the ratio of 5:1 for training and testing respectively. k-fold was used for cross-validation with k value of 5. K-fold cross validation is a method where datasets are divided in the form of k-folds as per the no. of k value (in this case 5) and one of the block of k-fold is used for testing until each of the k-folds has served as test data as shown in Figure 2 (Mujtaba, 2020).They also state that ResNe50 shows faster training process than other models. (Ali Narin, 2020)

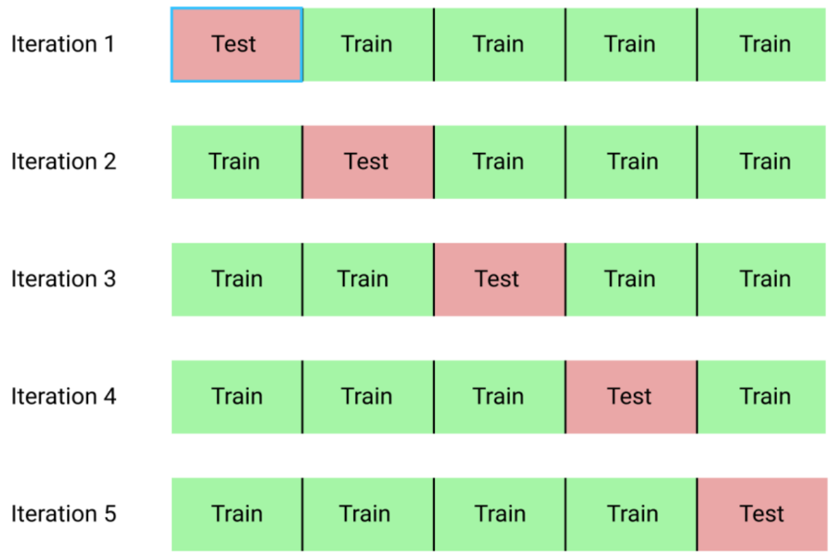


Figure 2 dataset distribution of k-fold cross validation with k=5 (Shaikh, 2018)

Both of these papers use followings for Performance metrics:

Accuracy = (TN + TP) / (TN + TP + FN + FP)

Sensitivity = TP / (TP + FN)

Precision = TP / (TP + FP)

F1-Score = 2x(Precision*Recall)/(Precision*Recall)

where TP, FP, FN and TN stand for True Positive, False Positive, False Negative and True Negative respectively.

Analysis:

From above system we can find that ResNet50 has higher precision than any other model used for COVID detection with low validation error and faster training processes. In my opinion, the k-fold cross validation technique in this paper is its primary strength as shown in Figure 1. As they were able to maintain really good accuracy with minimal amount of dataset. The major takeaway from these two papers is that ResNet50 work better and faster in comparison with any other models both while training and testing as mentioned above and we can calculate the performance metrics using provided method. However, there were a lot of steps they could have shadowed to get more optimum results. For example, augmenting using common techniques (flip, crop, rotate and change perspective) to the existing datasets would be helpful. Also, there were small number of datasets with millions of parameters for individual networks which leads to the problems like increasing architectural complexity, computational complexity and overfitting. So, we could just fine tune the hyper parameters or freeze the networks accordingly to overcome this problem.

2.3. Literature Review on Algorithm and Architecture

2.3.1. CNN

Convolutional Neural Network (CNN) is one of the most widespread deep neural networks due to its excellent performance in a field of ML. CNN has prominent results in a field related to pattern recognition. A standard neural network transforms data into a one-dimensional sequence, reducing the sensitivity of the qualified classifier to positional changes. But CNN processes the images considering its structure. In our project different layers are used to extract feature from CXR images. Here is a simple example on how CNN works theoretically, if there is an image of 'dog' we fed in as input then at first it might start with detecting the edges of the picture, then eyes, nose and eventually it can classify the given image as a dog using various layers including Convolution layer, Non-linearity or activation layers (ReLU), Pooling Layer and fully-connected layer. (Alabwi, 2017)

- Convolution Layer

The picture that we fed as input is first arranged in a matrix format and what this layer does is that it performs a dot product with a smaller sized matrix known as "Filter" or "Kernel" as shown in Figure 3. Filters are the hyperparameters in CNN which is especially used to detect some specific feature (like edges, eyes etc of dog images in above example). Each layer has its own filter or kernel which identifies different features from input. The filter is applied to each overlapping part systematically from left to right and top to bottom according to the stride value. Stride is a value that specify the number of jumps or steps to take while sliding in case of Figure 3 the stride value is 1.

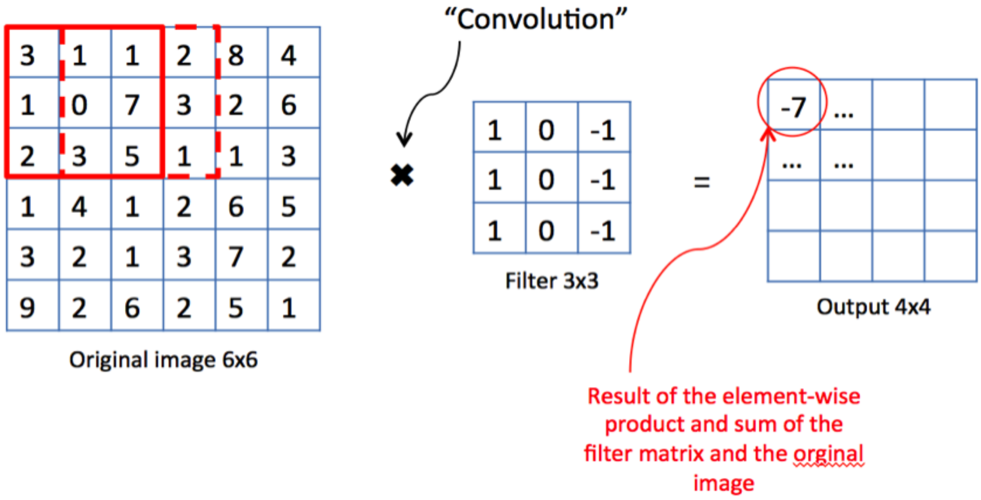


Figure 3 Dot product of images matrix and filter (Cavaioni, 2018)

- Padding

Padding is used if we don't want our image to lose the information of corners or edge of the given image. Also, when we pass input on a neural net it convolves the image in hundreds of layers which may shrink the image a little too much. So, to overcome this problem we can use padding technique called zero padding which add a layer of additional border in an input image as shown in Figure 4. Notice that in figure below the input and output size is same i.e. 6X6.

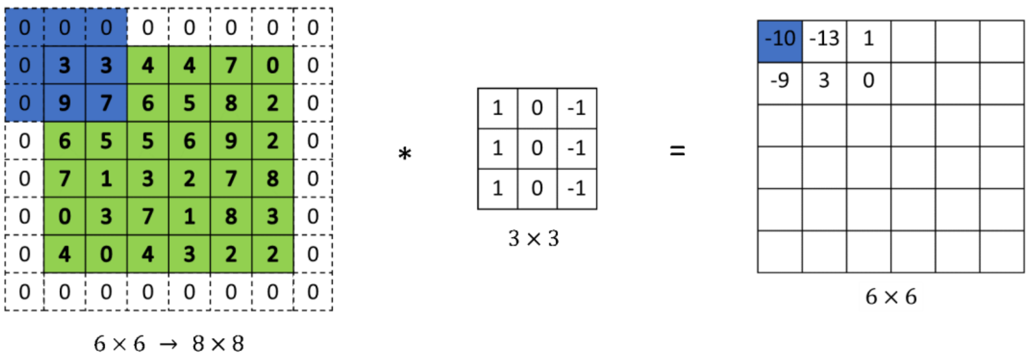


Figure 4 Zero Padding the input and using dot product with filter

- ReLU Layer (Activation Function)

ReLU stands for Rectified Linear Unit. The goal of ReLU is to incorporate non-linearity within our network. For each element of an input, a ReLU layer performs a threshold operation where any value less than zero is set to zero. Those neurons containing value less than 0 is deactivated. We denote ReLU as max(0,x). The only problem with ReLU is that totally blocks information of those values having less than 0 which may lead to information loss. In such case we can use Leaky ReLU. There are other nonlinear functions which can also be used instead of ReLU, such as softmax, leaky-ReLU, tanh or sigmoid which are discussed later. (Alabwi, 2017)

- Pooling Layer

If the picture is too large, a pooling layer is used to reduce the number of parameters. It is similar to reducing the resolution of an image to reduce the complexity for further layers. There are different types of Pooling: i. max pooling ii. Average pooling and iii. Sum Pooling. Max pooling is the most common which divides image into sub partition and return the max matrix value in that partition as shown in Figure 5.

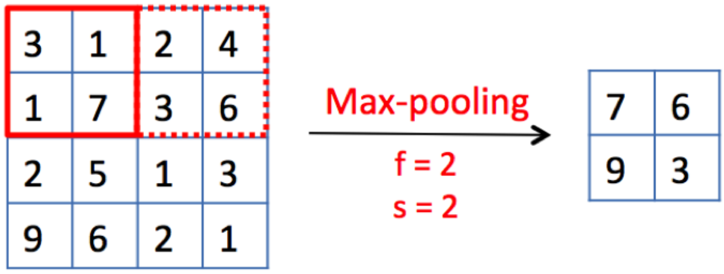


Figure 5 Max-pooling Example (Cavaioni, 2018)

- Fully Connected Layer

This layer is also known as Dense Layer. In this layer we flatten the matrix into a vector. This is where the training time takes the most because it includes a lot of parameters. We can use dropout technique to overcome this problem and also to avoid overfitting. This layer is similar to the normal ANN which consists of its own input, hidden and output layer. All the inputs are multiplied and added to their respective weights and biases which then pass through the activation function and provide input to hidden layer. This process continues until the output layer. Then, we have an activation function such as softmax or sigmoid to classify the outputs. Sigmoid which squeezes the values of input between 0 to 1 and is used mostly in binary classification problem.

Finally, we calculate the loss of the given network and some hyperparameters like weights or kernels, biases, learning rate etc are optimized to reduce the loss as much as possible. (Alabwi, 2017)

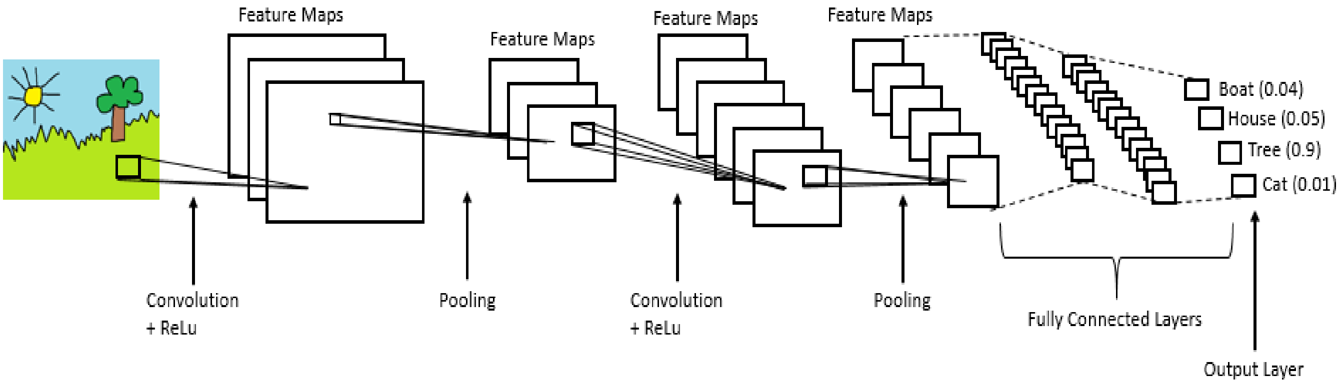


Figure 6 Complete architecture of CNN (Prabhu, 2018)

2.3.2. Activation Function:

In neural networks, activation functions are used to calculate input weights and biases, which help in determining whether a neuron can be activated or not. Gradient processing is used to manipulate the data, and then output is generated. AF can be linear or nonlinear, and it can be used to monitor outputs in a variety of fields, including object recognition and classification, speech recognition, segmentation, weather forecasting, and many others. The neural network input layer takes in data such as images, texts, videos, audio, or numeric data for training, and the output layer produces results that are then passed through AFs to perform classifications with their probabilities. If a certain threshold value is met, AF returns 1; otherwise, it returns 0.

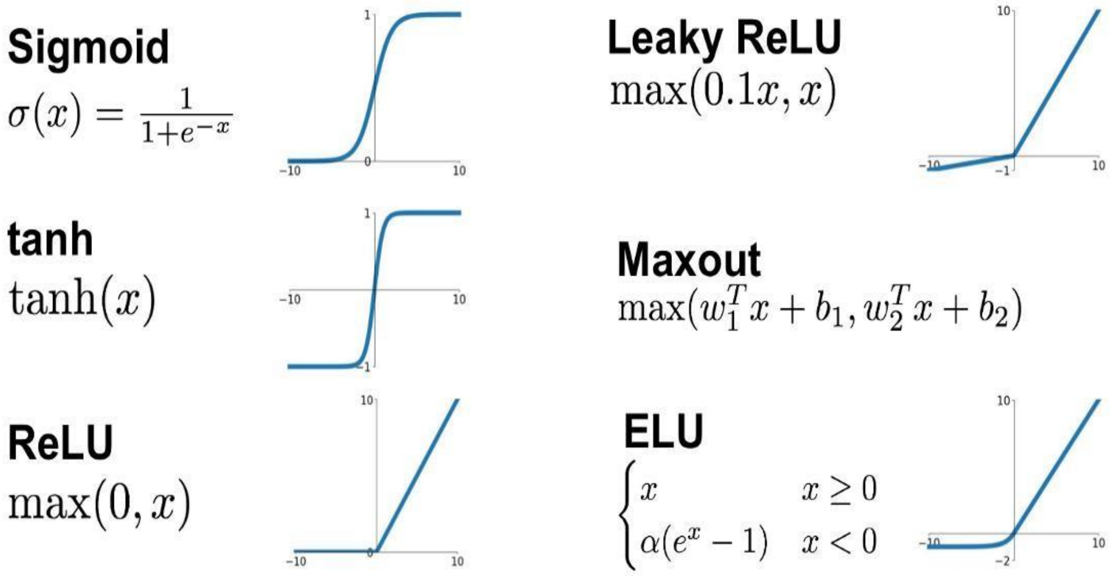


Figure 13: Activation Functions

AFs have a major impact on the training dynamics as well as task performance efficiency in deep networks. Rectified Linear Unit (ReLU) is the most common and commonly used AF at the moment because, unlike tanh and sigmoid, it does not destroy the gradient during backpropagation.

2.3.3. Loss Function

Loss functions can be used to determine whether a machine learning model's prediction is good or bad. After being multiplied by their individual weights, the input layer's neurons are summed and passed on to the hidden layers. After the summation function with respective weights, the output layer receives its input from the hidden layer. The loss layer, which handles weight changes across the network, comes after the fully-connected layer. The weights in the convolution and fully-connected layers are given random values before the network is trained. During preparation, the loss layer compares the fully-connected layer's estimates to the actual values as much as possible, with the intention of minimizing the discrepancy between the estimate and the real value. The weights in both the convolution and dense layers are adjusted by the loss layer. The loss function is chosen in accordance with the desired output. If the output is a projection of a real-value quantity, using Mean Squared Error (MSE) as the loss function is a better option, whereas Cross-Entropy is used for Binary and multi-class classification problems. (Simon Kornblith, 2020).

2.3.4. Optimizer

Loss functions are used in deep learning models to calculate the errors that occur during the model's prediction. Each iteration, minor adjustments to the model's parameters, such as weights, biases, and other hyper-parameters, are made to minimize these errors. Optimizers are used to determine when and by how much to adjust the parameters. Choosing the right optimizer and parameters for neural network models will dramatically improve their performance. The optimizer is escorted by the loss feature as it marches in the right direction. The most popular optimization techniques for deep learning models are gradient descent algorithms.

Due to the restricted computing capacity and large-scale dataset, gradient descent forms such as batch gradient descent and stochastic gradient descent are unable to perform the training effectively. Aside from that, the phase size is difficult to adjust, resulting in a high computation expense. Various optimization algorithms have been developed to solve this. Adam is the most common optimization algorithm among various optimization algorithms such as gradient descent, adagrad, Adadelta, and RMSprop. Adam is a first-order gradient-based stochastic objective function optimization algorithm based on adaptive estimates of lower order moments that can adjust learning rates for different parameters. (Diederik P. Kingma, 2015)

2.3.5. Confusion matrix

Confusion matrices also known as error matrix is the tabular format visualization means especially used for classification problems. It measures the amount of accurate and incorrect assumptions made by a classifier/model.

- true positives (TP): When both actual and predicted value is positive.
- true negatives (TN): When both actual and predicted value is false.
- false positives (FP): Also known as a “Type 2 error”, where the actual value is negative but predicted value is positive

- false negatives (FN): Also known as a "Type 1 error", where the real value is positive but the expected value is negative.

  The followings are the list of rates that are often calculated from a binary classifier's confusion matrix:

- Accuracy: Accuracy essentially tests how much the right approximation is made by the classifier. It's the ratio between the number of predictions that are correct and the total number of predictions

  Frac {TP + TN} {TP + TN + FP + FN}

- Precision: It refers to as from all the positive predicted value how much is actually positive

  Frac {TP} {TP + FP}

- Recall or True Positive Rate: Out of the total actual positive, what percentage are predicted positive

  Frac {TP} {FN + TP}

- f1-score: This is a harmonic mean of both recall and precision. The best f1-score is considered to be 1 and worst is 0.

  Fraction of 2 * {Precision * Recall} {Precision + Recall}

Predicted → Actual ↓	0	1
0	TP	FN
1	FP	TN

2.3.6. Deep Residual Learning for Image Recognition (ResNet)

ResNet which is also known as Residual Network is a winner of ILSVRC 2015 and as well as MS COCO 2015. The model was trained on roughly 1.2 million images form ImageNet (tsang, 2018). It was proposed by 4 Microsoft Researchers: Kaiming He, Xiangyu Zhang, Shaoquing Ren and Jian Sun. In theory there is a belief that the more neural net is deep the more well it performs. But, on their official paper (Kaiming He, Dec, 2015) they mentioned that in practical that it is not the case. If the layer is too deep the test percentage starts increasing because of Vanishing and Exploding Gradient. But it can be fixed with a technique called “Skip/shortcut Connections” or “Residual Learning Block” which adds a shortcut connection that skips the layer as shown in Figure 7 which allows to help train a very deep network. Also, they use the 1x1 conv layers as a form of dimension reduction and restoration called “bottleneck design” which makes the architecture more efficient as shown in Figure 8. By this method, they were able to train up-to 152 layers following the similar architecture as shown in Figure 9 (34-layer).

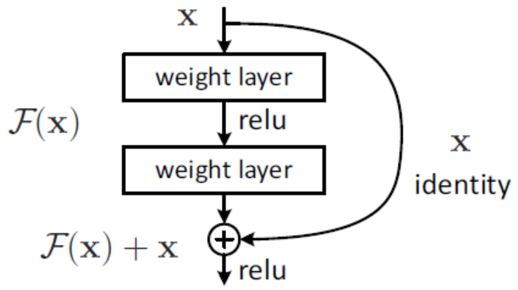


Figure 7 Residual learning: a building block (tsang, 2018)

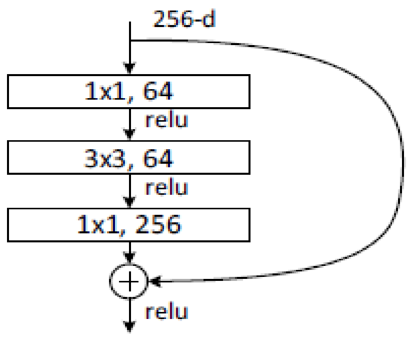


Figure 8 A "bottleneck" building block for ResNet50 (Kaiming He, Dec, 2015).

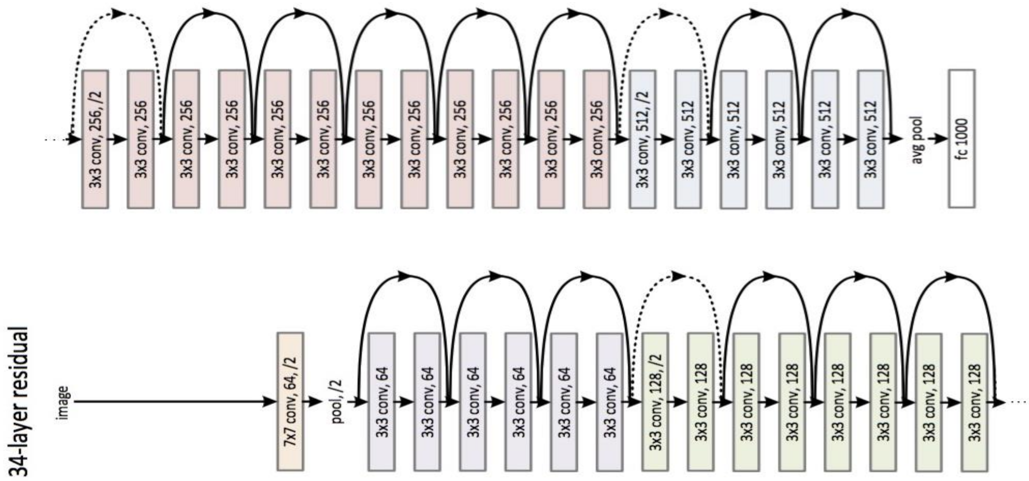


Figure 9: 34-layer ResNet Architecture (TitoOrt, 2018)

This architecture is excellent if there are limited amount of dataset available. As Covid-19 is very new and due to limited amount of test kits available and delayed test results there are only few amounts of data available online which may lead system to a problem called under fitting. So, to avoid these kind of problem ResNet50 can be used.

2.3.7. Very Deep Convolutional Networks for Large-Scale Image Recognition

In their paper "Very Deep Convolutional Networks for Large-Scale Image Recognition," K. Simonyan and A. Zisserman from the University of Oxford proposed the VGG16 convolutional neural network model. The model achieves 92.7 percent top-5 test accuracy on ImageNet datset, a dataset of over 14 million images belonging to 1000 categories. It was a well-known model that was submitted to the ILSVRC-2014.

In this architecture the image of size 224X224X3 is first passed to stack of cov1 layer. Filter size of 3X3 were used with 64 channels and stride value of 1. The channel size is doubled after max pooling layer until it becomes 512. 5 max-pooling layers were used with window size of 2X2 and stride value of 2. Three Dense layers adopt a stack of convolutional layers (which vary in depth depending on architecture): the first two have 4096 channels each, whereas the third performs 1000-way ILSVRC classification and thus has 1000 channels (one for each class). The soft-max layer is the final layer. In all networks, the completely connected layers are configured in the same way. (Parashar, 2020)

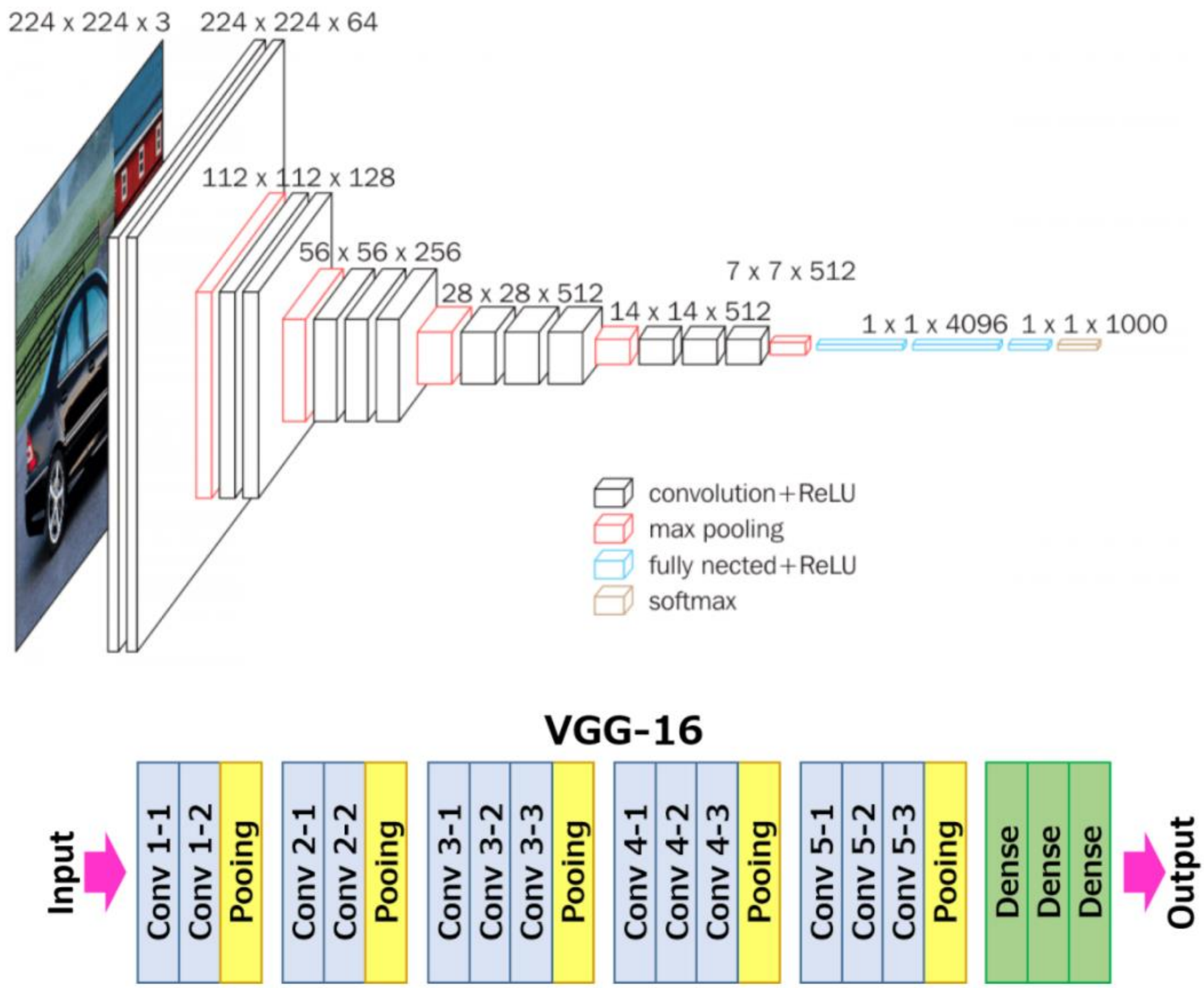


Figure 10 VGG16 architecture

3. Probable issues during the project

3.1. Possible Risk Factors and fallback plan

Possible Risk Factors	Fallback Plan
Hardware Failure	Important project file should be stored in the cloud. Important data should be backed up in an external hardware.

Change in unexpected project planning due to current pandemic.	Update the requirement according to the situation
Operational Risks: Lack of communication with stakeholder/supervisor	Have communication plan. Fix appointment before meeting to supervisor. Track supervisors schedule if possible. (Windsor, 2019)
Internet Unavailability	Notify ISP's to provide smooth internet during your working hours
Lack of resources or resource availability.	Make a strong Resource Management Plan. (Bridges, 2019)
The project is behind schedule.	Work overtime. Extend the Deadline. Redefine the Scope.

3.2. Social, Legal and Ethical issues and its solution.

- Plagiarism
  - Plagiarism is one of the main issues that can occur after completing a project. To avoid this, you should always paraphrase the piece of information that you have found from the research paper. Also, one of the most important way to avoid plagiarism is by including a reference page or provide citation. Following the document formatting guidelines is also a good way to avoid plagiarism.

- Accountability
    - When things start to get messed up. Its human nature to avoid mistakes and consequences that they have done. People tend to blame or try to shift those problems to others which can create some serious issues. To avoid this, one should understand the importance of making their own mistakes and try to handle by themselves rather than blaming it to others.

- Criticism
    - Criticism is the practice of judging the merits and demerits of something. This may occur after you publicize the app. People reviewing your app might criticize you in both ways: constructive and destructive. Most criticism are not meant to be hurtful so try to acknowledge them and appreciate the people for their review.

4. Project Methodology

Here, I have gathered 2 methodologies that is suitable for my project and accepted only one of them.

4.1. Waterfall

The Waterfall Model is a sequential model for software development that separates the process into pre-defined phases. Each process must be finished before the next one can begin, and the phases must not overlap. During the SDLC process, each phase is programmed to execute a particular task.

The main reason for this model to be in my list is that the environment of this model is stable and Application is not big and complicated. But this model needs solid planning because requirements should not be changed after you defined it once. There is no overlapping in the phases. Due to the single cycle strategy, there is no room for evaluating, revising and modifying in a Waterfall project once you've done it. When you're in the development stage, going back and modifying something that wasn't well planned in the prototype stage is very challenging. This is the reason that I rejected this methodology as my project is very new to me there is a high possibility that there will be changes here and there while project is ongoing. (Aston, 2019)

4.2. Scrum

Because Waterfall model does not let you reconstruct your requirement this methodology is not the best for my project. I need to find a methodology which is very similar to waterfall model but unlike waterfall model I can re-construct my requirements at any point of time which is where Scrum model shines. This is one of the most popular methodology which inherits the features of agile methodology. The main objective of this methodology is to improve communications and speed of development. What makes this methodology so good for my project is its versatility and flexibility. The main disadvantage of scrum is that it is difficult to accurately determine the amount of cost that will be needed to complete the project due to constantly changing requirements which in case of my project is not a problem as there is no cost involved in this project. I already have necessary tools on building my project. There is also a term called “Retrospective meeting” in this framework/methodology which is mainly based for team project, as my project is an individual project it is really a viable option. (Aston, 2019)

Thus, according to the framework the whole system is divided into 3 different sub systems and each sub-system is then carried out to the phase called sprint planning where tasks

The different stages involved in my project according to scrum methodology are:

1. Specification
    - Various fact-finding methods are used relating to covid classification, such as analysis and paper sampling, are used to identify requirements.
    - Different functional, non-functional and usability requirements are identified first for our web app. Then, the document resulting from this is called SRS (System Requirement Specification).

2. Design

- We designed wireframes and different UML diagrams required like WBS, Functional decomposition, use case, activity and sequence.

3. Implementation

- Translating the design into executable programs
- Also, Model Training and Testing are also done in this section along with data visualization.

4. Merging

- Since, there are two components in this project we need to merge the AI model with a web app we created.

5. Testing

➢ Checking if the system is what the end-user really wants

4.3. Steps to build model

Following steps are to be maintained to build an AI-model.

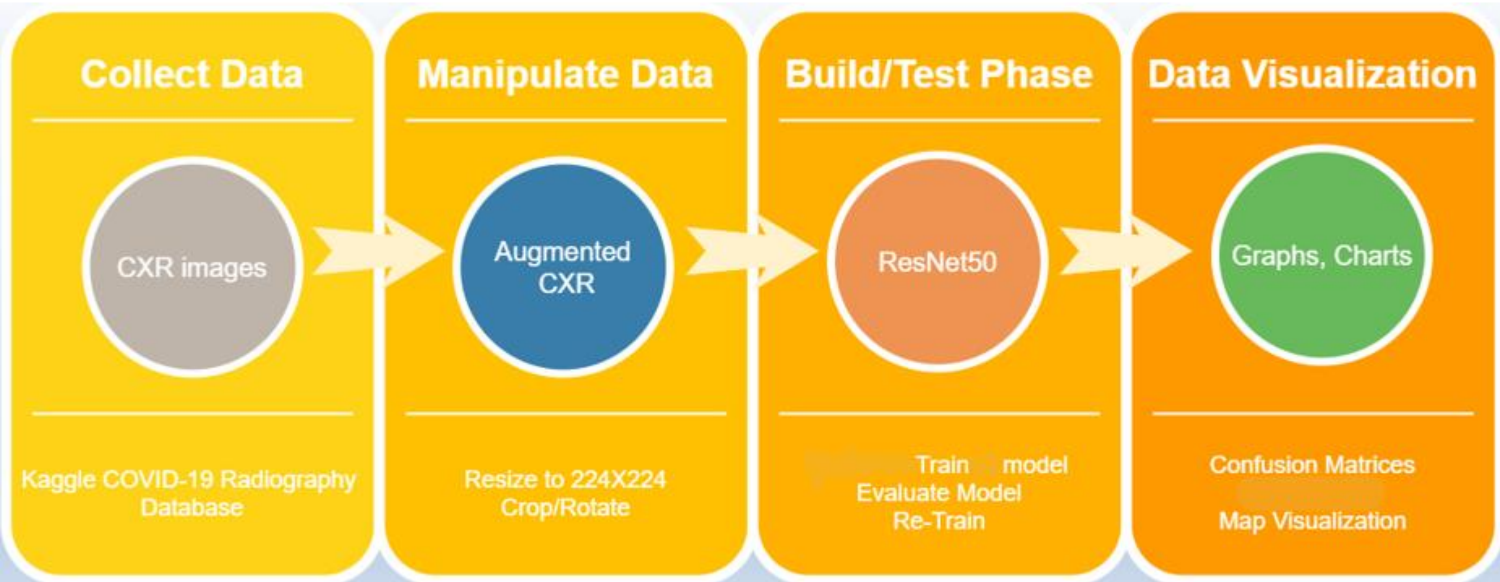


Figure 11 Steps to build our model

Required images for building a model is first collected, there are various open-source data available online. In our case, we are using dataset provided by Kaggle. Then the CXR images are manipulated in a way that we can get better output. Then all the augmented data are to be processed on custom models (ResNet, Vgg and Lenet). If the result is not satisfying after the evaluation and testing, we can re-train our model by tuning the parameters. We can use various tables, graphs and charts like Confusion matrices and train vs validation on loss and accuracy graphs to visualize our data and make comparison between these architecture's.

5. Tools and Techniques

5.1. Frameworks and Libraries

- Django

Python Django framework is used for developing the web app. This is a high-level python framework based on model-view-template (MVT) architectural pattern.

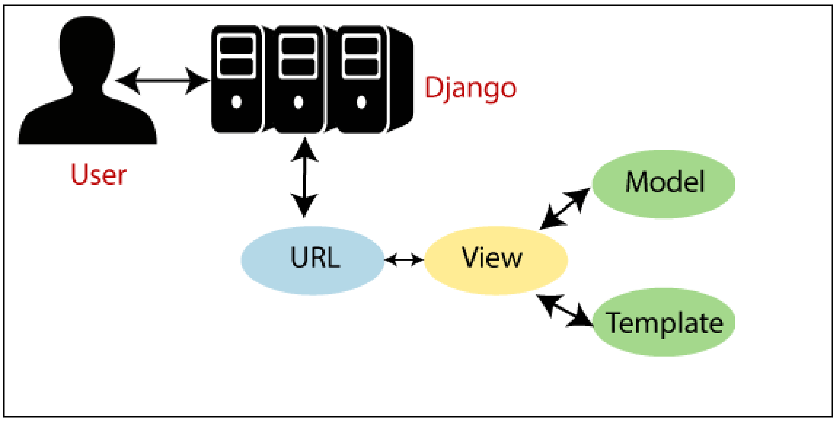


Figure 12 MVT pattern in Django (Tutorail and Example, 2020)

The "MODEL" in MVT helps in maintaining the data in database, "VIEW" acts as the link between model data and templates. It sees the user request (URL), retrieves appropriate data from the database, then renders back the template accordingly and "TEMPLATES" are responsible for the entire User Interface. The architecture of MVT pattern is shown in Figure 12 (javatpoint, 2018)

- Model train, save and integration:

Keras library on top of TensorFlow as a backend was implemented in model training. Keras was also utilized for integrating our saved model into Django web app.

➢ Data Preprocessing and Visualizing libraries:

Numpy and Pandas: Numpy and Pandas along with matplotlib and seaborn was used for visualization of data.
These libraries were also used for image preprocessing.

Matplotlib, Seaborn: These are very reliable python library used for visualizing the model performance. I have used this library for graphically representing Confusion matrix, train vs validation loss, train vs valid accuracy etc.

Tensorboard: This library is used for the visual comparison of the 3 custom architectures (Vgg/ ResNet/ lenet) used in this project.

5.2. Tools

a. Modeling tools
- For UML**:** Draw.io is a free online web based diagraming app where I created most of my diagrams like flowcharts, UML, ER Diagrams, use-case etc.
- For Wireframe: Balsamiq is easy drag and drop UI design tool which is very easy to use.

b. Programming Tools:
- Jupyter Notebook – The Jupyter Notebook is a server client program that allows the editing and running of notebook documents using a web browser. It is used to create a model of AI in my project. (Antonino Ingargiola and contributors., 2015)
- Programming Language/framework:
  - Python – Python is the go-to programming language for Building AI model.

c. Project Management Tool:

- GitHub: It helps to manage or backup constant changes and modifications in the code.

d. Communication Tools and others:

- Gmail – To communicate with supervisor
- Facebook Messenger and Viber – To communicate with Supervisor
- MS office – To prepare report (Word, Excel).
- Mozilla Firefox - Browser to research.
- Model is trained on NVIDIA GeForce MX150.

6. Artefact Designs

6.1. Gantt Chart

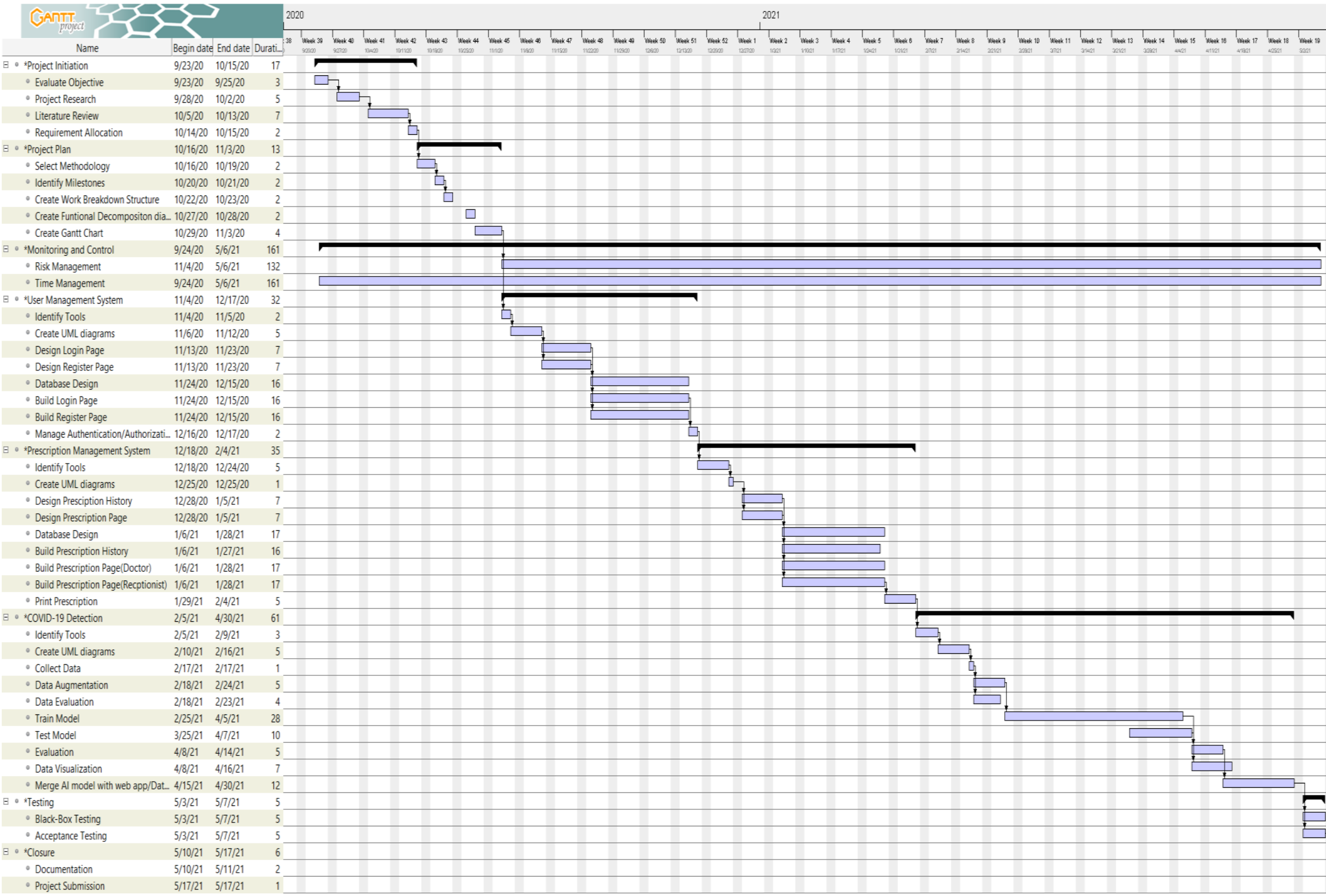


Figure 13 Gantt Chart of project COVID-X

7. Artefact

7.1. WBS (Work Breakdown Structure)

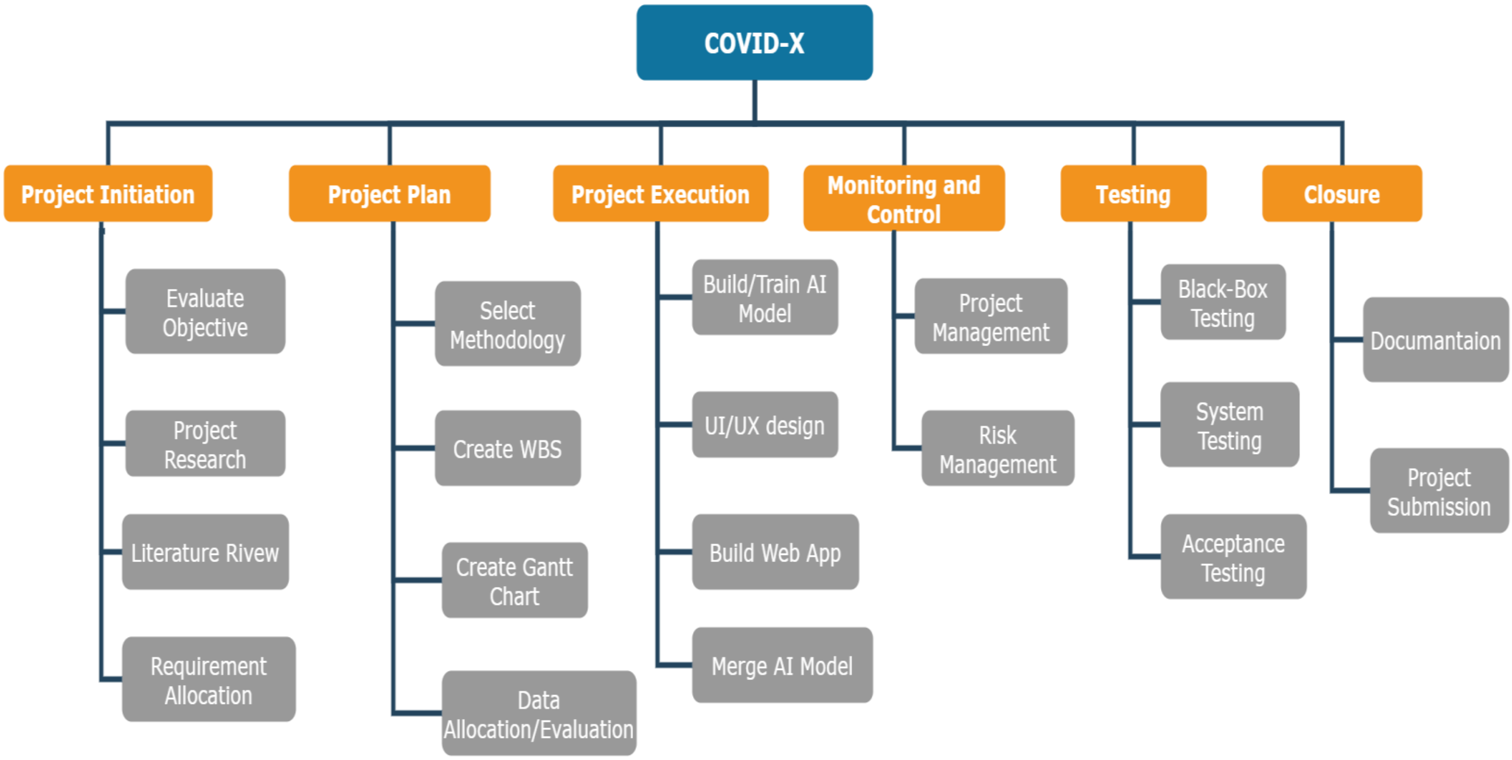


Figure 14 Work Breakdown Structure of COVID-X

The Figure 14 shows the three-level deep WBS of COVID-X. The first level determines our system, second level is the project management process groups in PMBOK (The project management body of knowledge) and the third level is an actual work divided into high level subtask.

7.2. Functional Decomposition Diagram

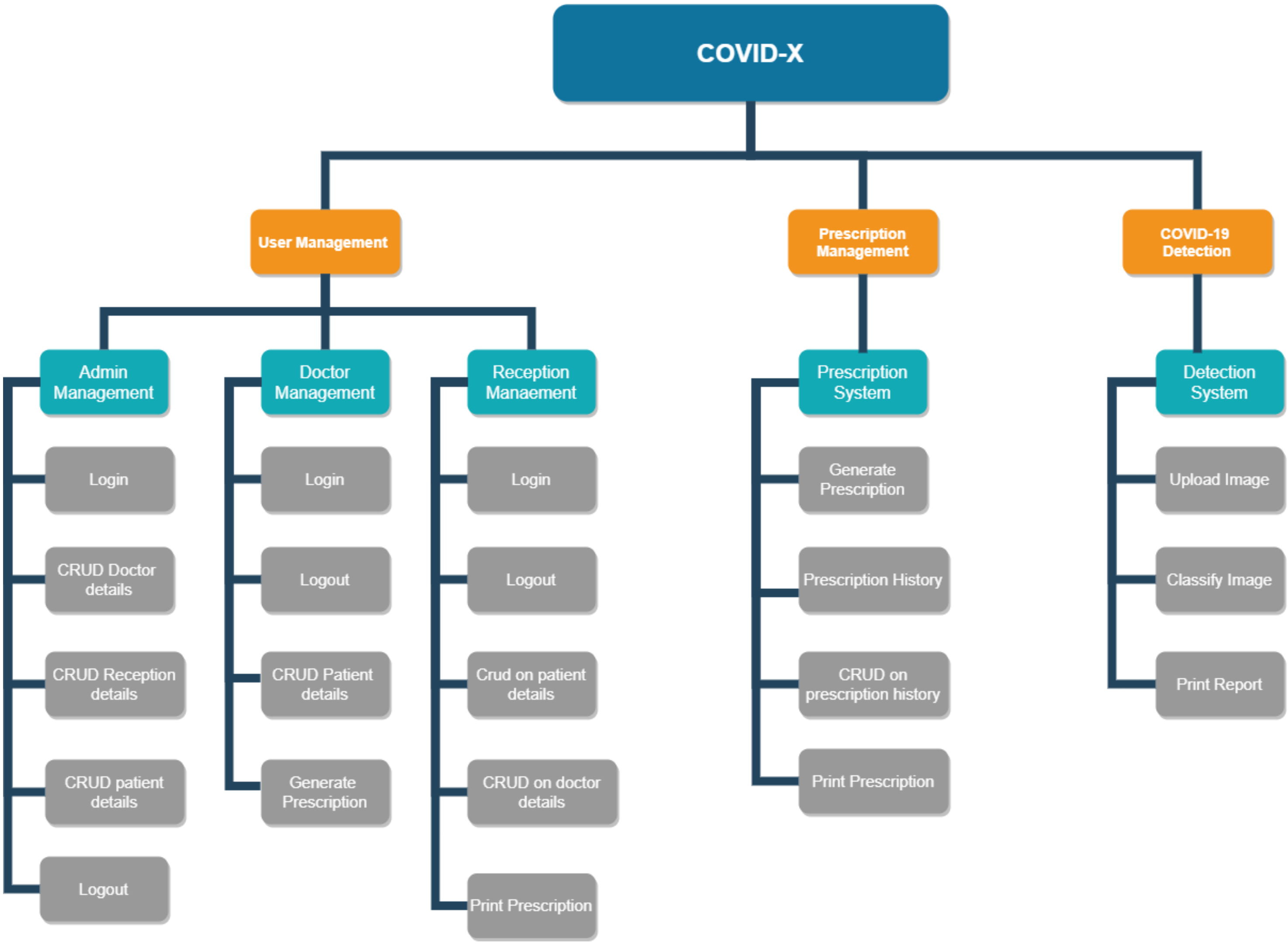


Figure 15 Functional Decomposition Diagram of COVID-X

<table>
<tr><th colspan="6"><u>COVID-X</u></th></tr>
<tr><th>Steps
Sub-system</th><th>Project Planning</th><th>Project Backlog</th><th>Modeling/Visualization</th><th>Implementation</th><th>Testing</th></tr>
<tr><td>User Management</td><td>Login
a. All users
Register (view/create)
b. Patient
c. Doctor
d. Reception
Logout
a. All users</td><td rowspan="3">SRS Table</td><td rowspan="2">• Use Case Diagram
• Activity Diagram
• Wireframe
• ERD
• Class Diagram
• Sequence Diagram</td><td rowspan="3">Execution</td><td rowspan="3">Testing/Testing Documents</td></tr>
<tr><td>Prescription Management</td><td>➢ Generate Prescription
➢ Prescription history</td></tr>
<tr><td>COVID Detection</td><td>➢ Build AI Model
➢ Compare different architectures
➢ implement Model
➢ Upload Image
➢ Classify Image</td><td>➢ Use Case Diagram
➢ Activity Diagram
➢ Wireframe
➢ ERD
➢ Class Diagram
➢ Sequence Diagram
➢ Confusion Matrices
➢ Comparison graphs between Custom VGG, ResNet and LeNet.
➢ Performance matrices
➢ (Accuracy, Sensitivity, f1-score, Positive Predictive value)</td></tr>
</table>

Table 1 Decomposition of each system in COVID-X.

7.3. SRS Table

DM = Doctor Management

RM = Reception Management

AM = Admin Management

PM = Prescription Management

CD = Covid-19 Detection system

Requirement Id	Requirement Specification	Moscow
DM-Functional	System should allow doctors to login into the dashboard through login portal	Must Have
	Doctor should be able to view, edit and delete patient.	Must Have
	System should allow doctors to search patient	Could Have
	System should allow user to logout from the dashboard.	Must Have
DM-Usability	Doctor should be Provided with the opinion for the password visibility.	Should Have
	System should show error message to user just beside the button.	Could Have
DM-Non-Functional	Password should contain alpha numeric, symbols.	Could Have
	Receptionist should not be able to login to Doctor’s dashboard with their credentials.	Must Have
	Doctors must match Login credentials from database before logging into the dashboard.	Must have
RM-Functional	System should allow receptionist to login into the dashboard through login portal	Must Have
	System should allow receptionist to be able to view, edit and delete patient.	Must Have
	System should allow receptionist to view details of doctors.	Must Have

	System should allow receptionist to search patient and doctor.	Could Have
	System should allow user to logout from the dashboard.	Must Have
RM-Usability	Receptionist should be Provided with the opinion for the password visibility.	Could Have
	System should show error message to user just beside the button.	Could Have
RM-Non-Functional	Password should contain alpha numeric, symbols	Could Have
	Doctor should not be able to login to Receptionist's dashboard with their credentials	Must Have
	Receptionist must match Login credentials from database before logging into the dashboard.	Must Have
AM-Functional	System should allow admin to login into the dashboard through login portal	Must Have
	System should allow admin to add Receptionist, doctor and patient.	Must Have
	System should allow admin to view patient, doctor and Reception details	Must Have
	System should allow admin to search patient, doctor and Receptionist.	Must Have
	System should be able to delete and edit the records of patient, doctor and receptionist.	Must Have
AM-Usability	Admin should be Provided with the opinion for the password visibility.	Should Have
AM-Non-Functional	Admin must match Login credentials from database before logging into the dashboard.	Must Have
	Password should contain alpha numeric, symbols	Could Have
	System should allow doctor to save prescription given to patient.	Must Have
	System should allow reception to view prescription provided by doctors.	Must have

PM-Functional	System should allow doctor to view patient's prescription history.	Must Have
	System should allow doctor to search prescription history.	Should Have
	System should allow reception to print prescription.	Could Have
PM-Non-Functional	Prescription history should be managed in alphabetical order (patients name).	Could have
CD-Functional	System should allow doctor to add CXR image from PC.	Must Have
CD-Non-Functional	System should provide detection result in less than 5 second.	Must have

7.4. Artefact Design

7.4.1. User Management System

PS: Note that the keyword "Manage" in the diagrams below represents Create, Read, Update, Delete, Search, and Restore Feature. For example, in the use case diagram below "Doctor" is associated with "Manage Patient" which means doctor can Create, Read, Update, Delete, Search and restore patient details.

Use Case diagrams

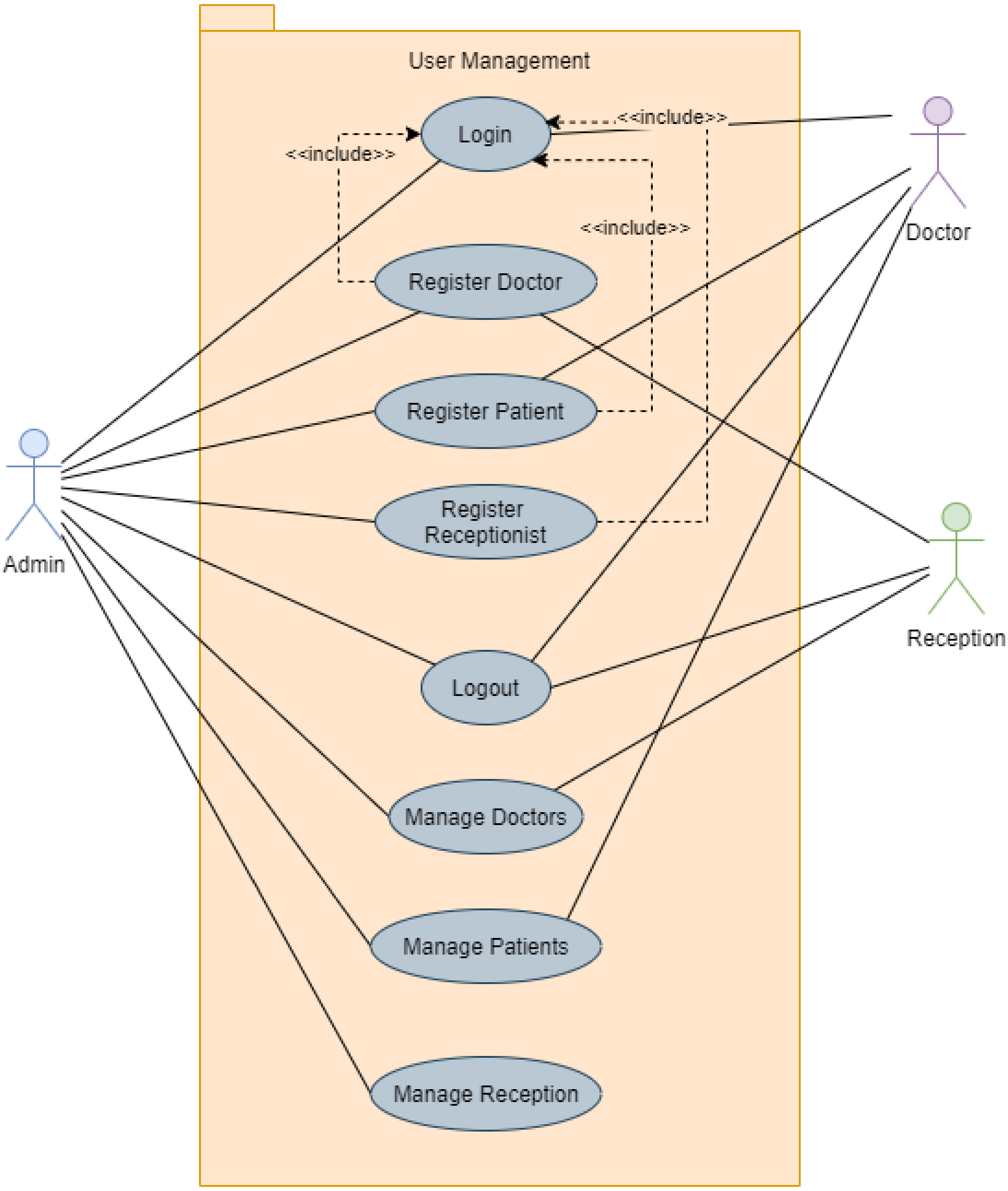


Figure 16 Use Case Diagram of UMS

Activity Diagram

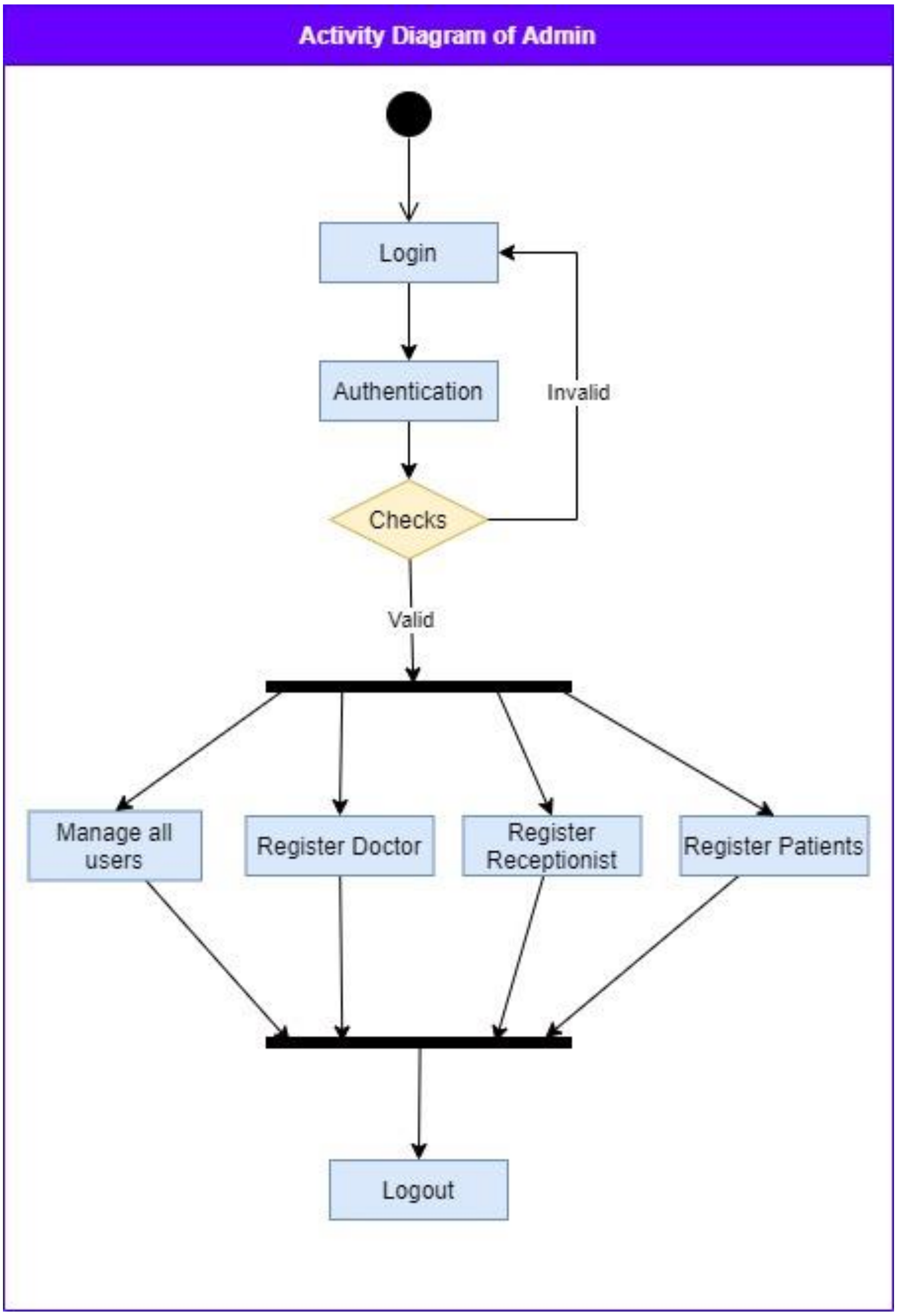


Figure 17 Activity Diagram of Admin in UMS

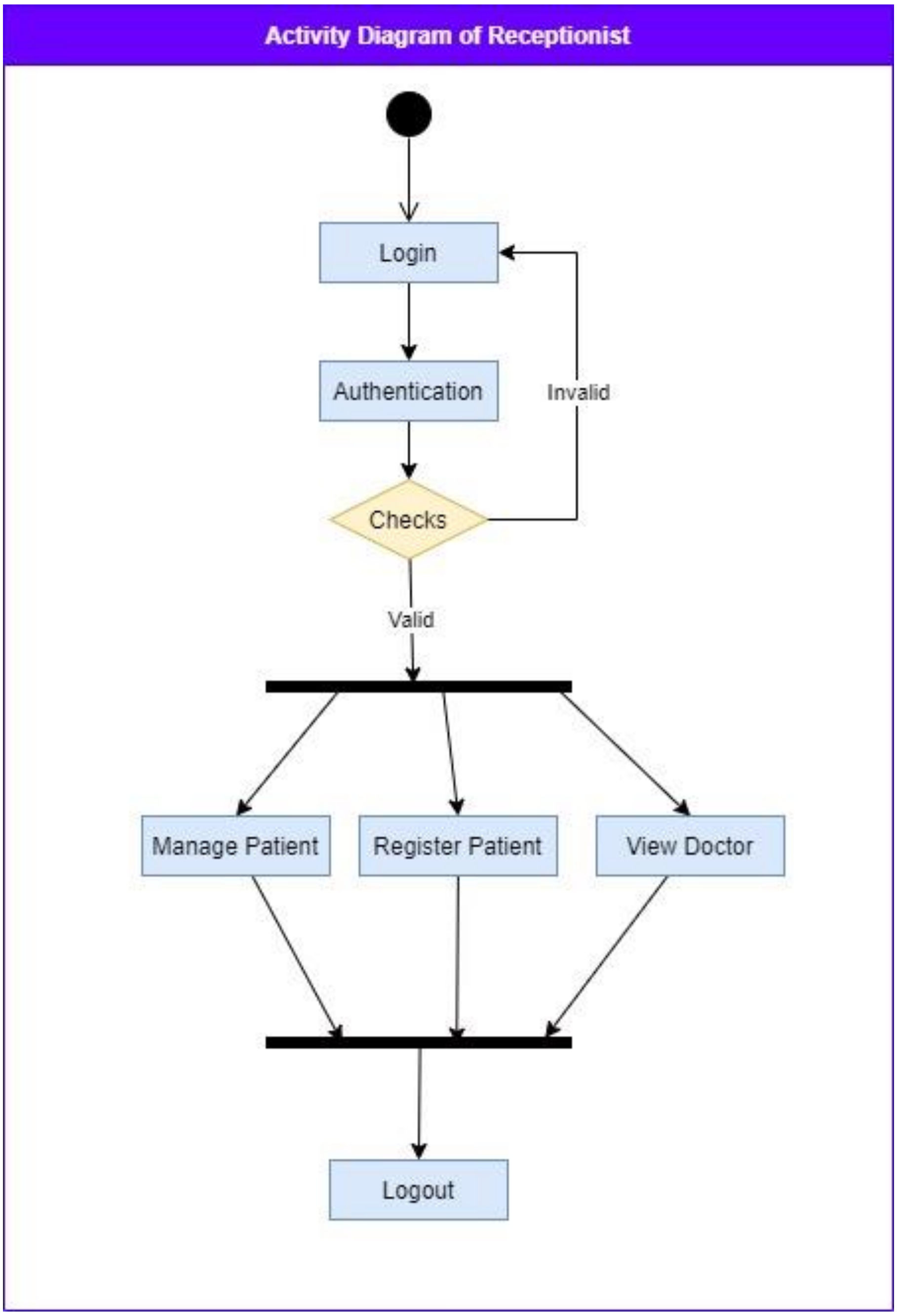


Figure 18 Activity Diagram of Receptionist in UMS

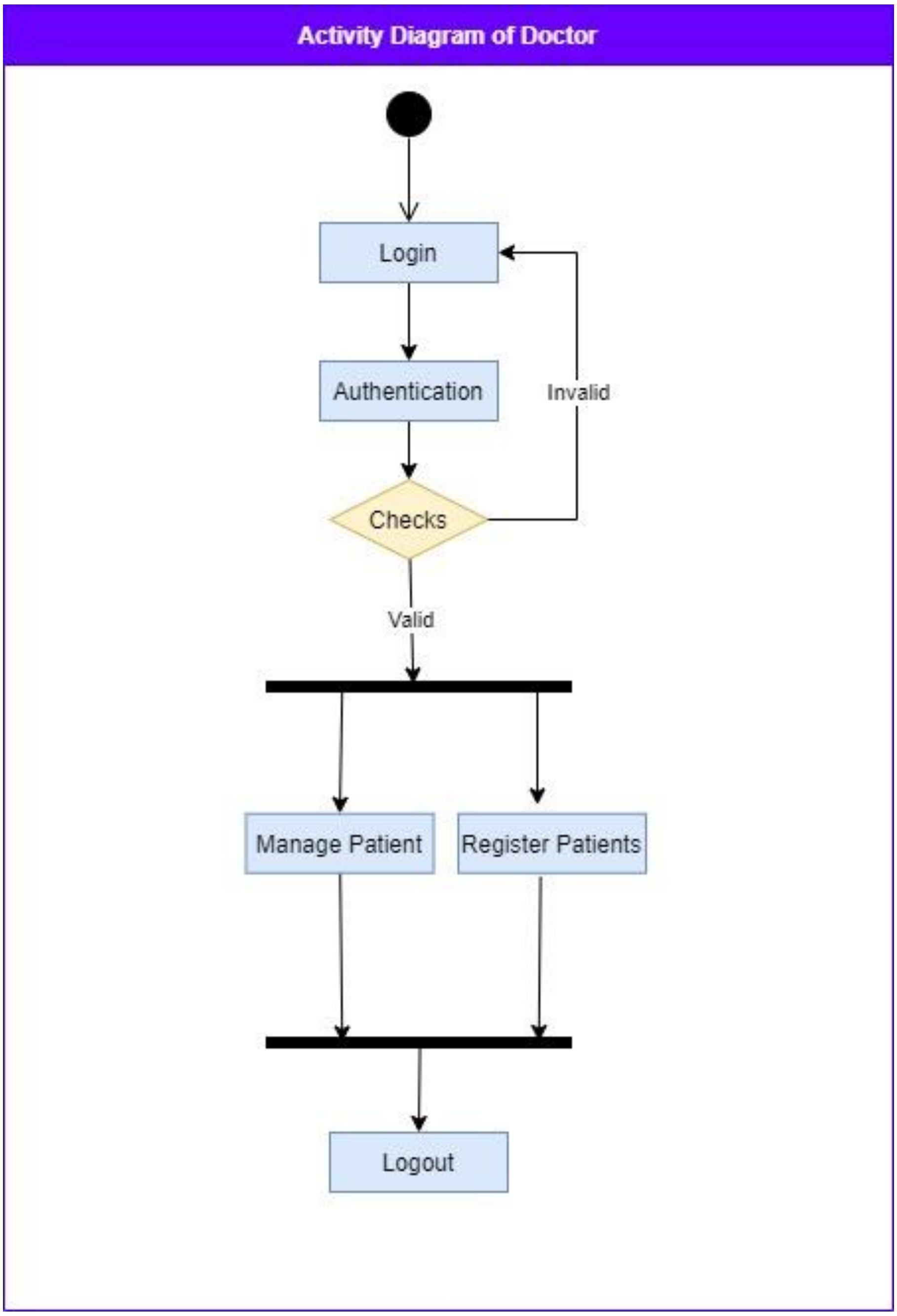


Figure 19 Activity Diagram of Doctor in UMS

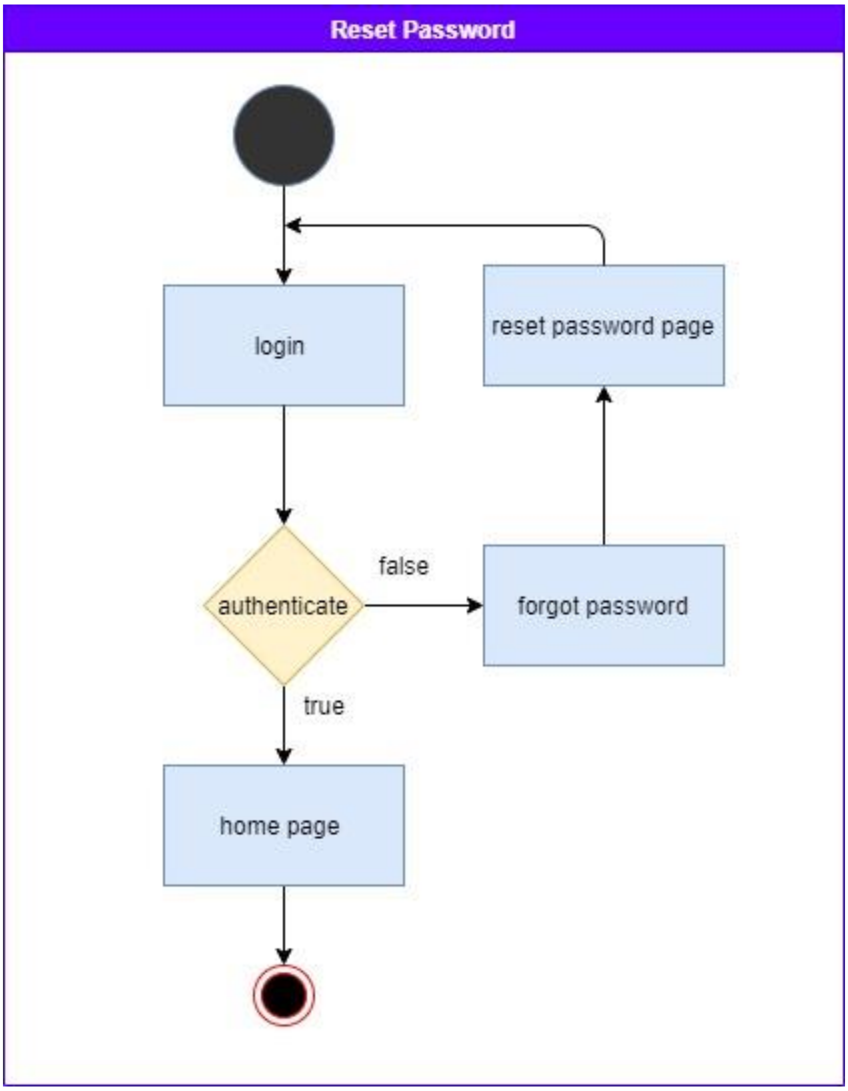


Figure 20 Activity diagram for reset password

ERD (Crowfoot)

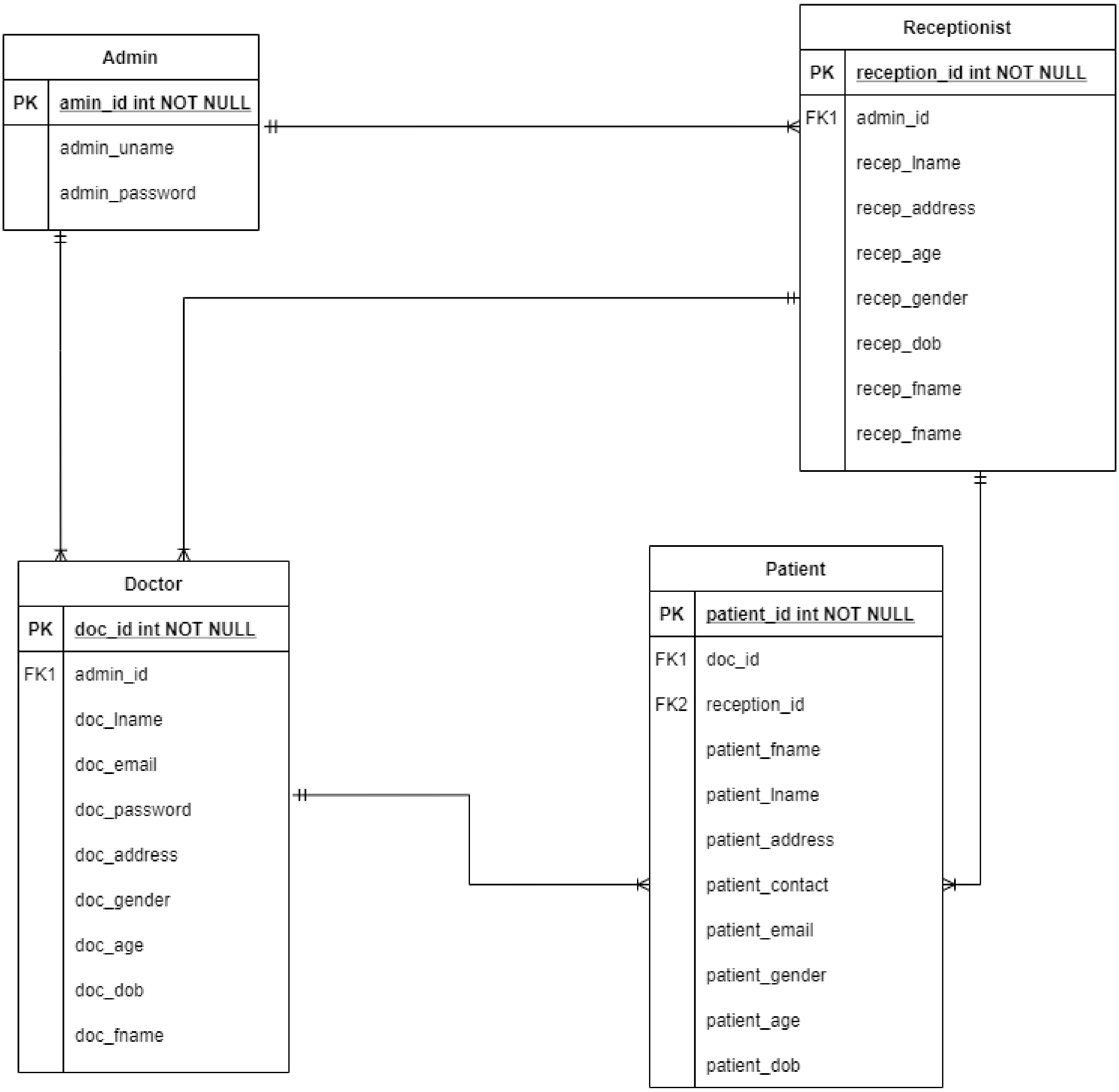


Figure 21 ERD of UMS

Class Diagram

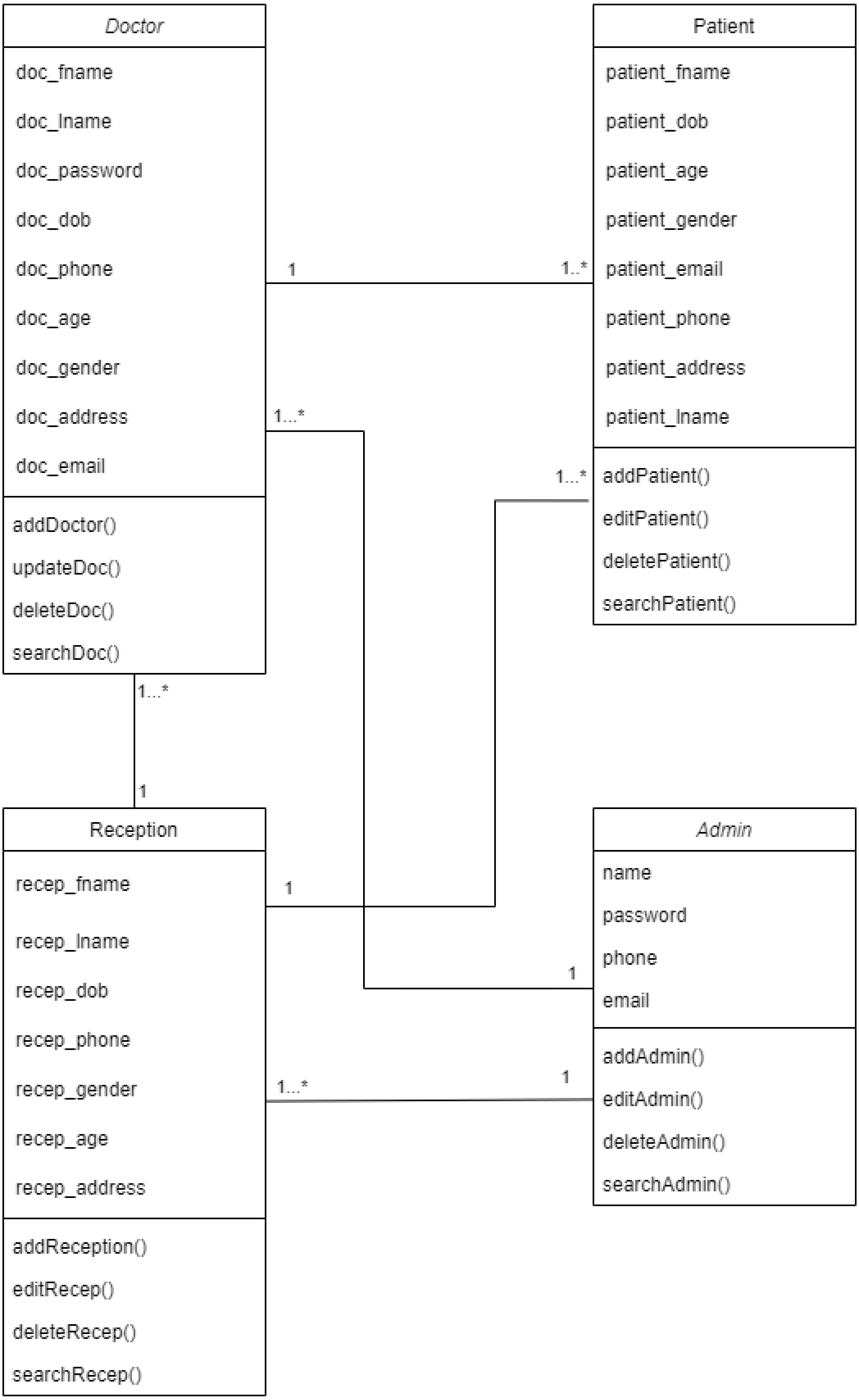


Figure 22 Class diagram for UMS

Sequence Diagram

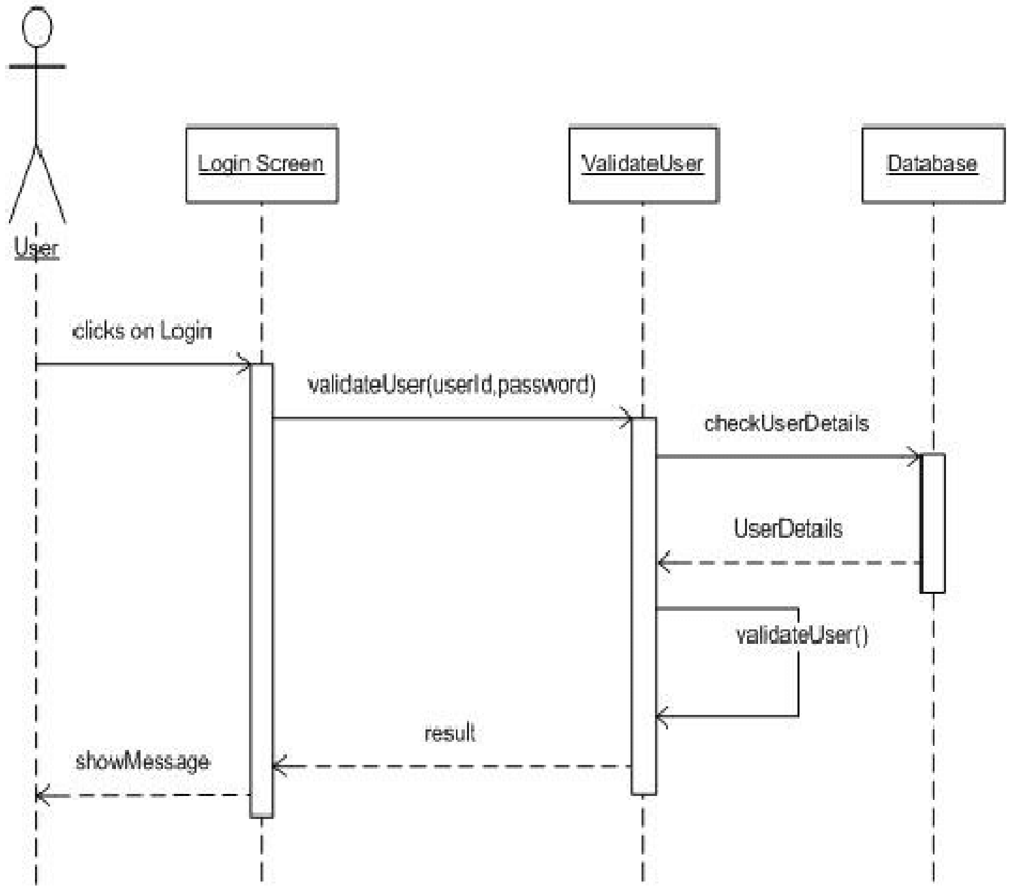


Figure 23 Sequence diagram for user Login

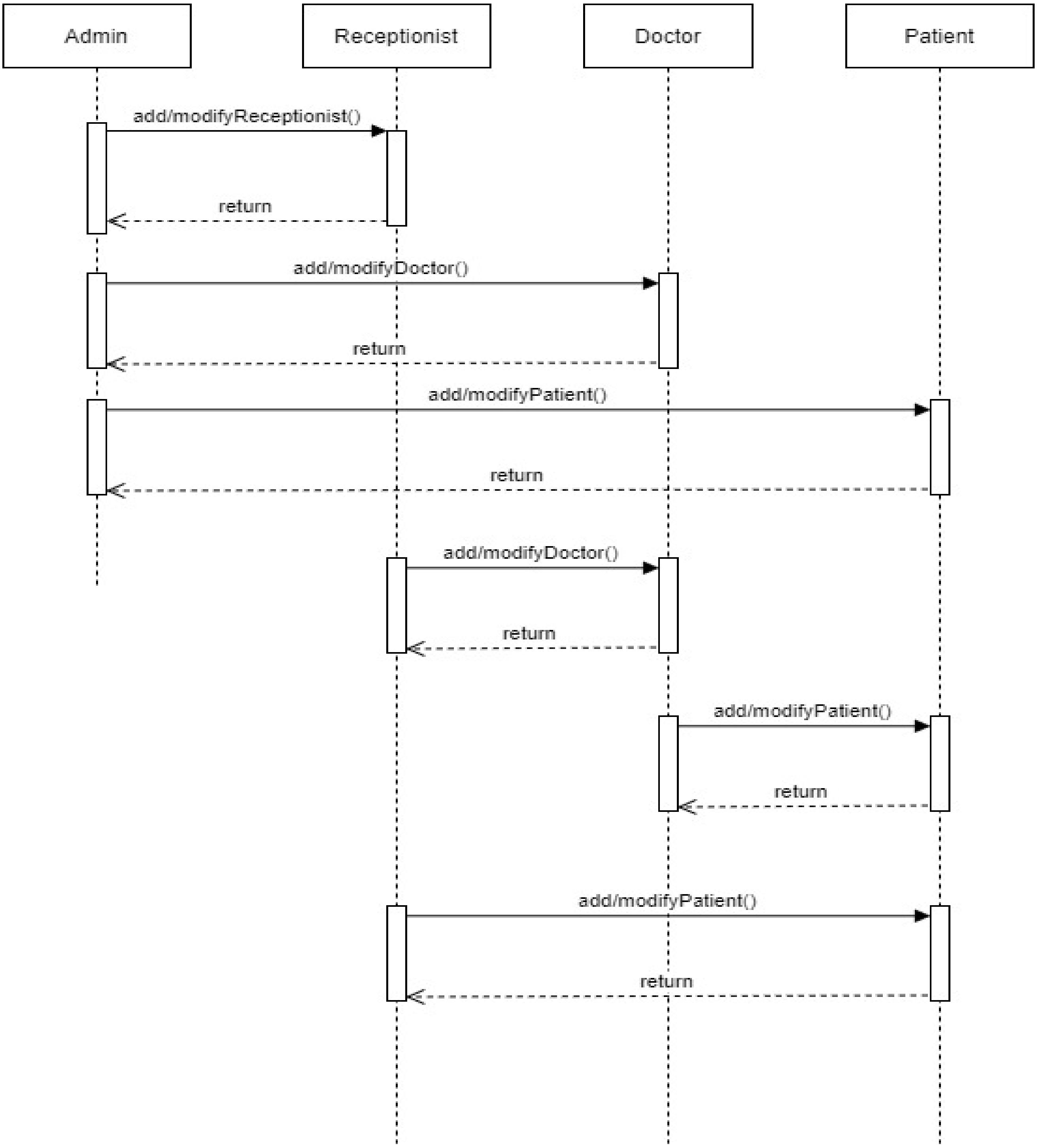
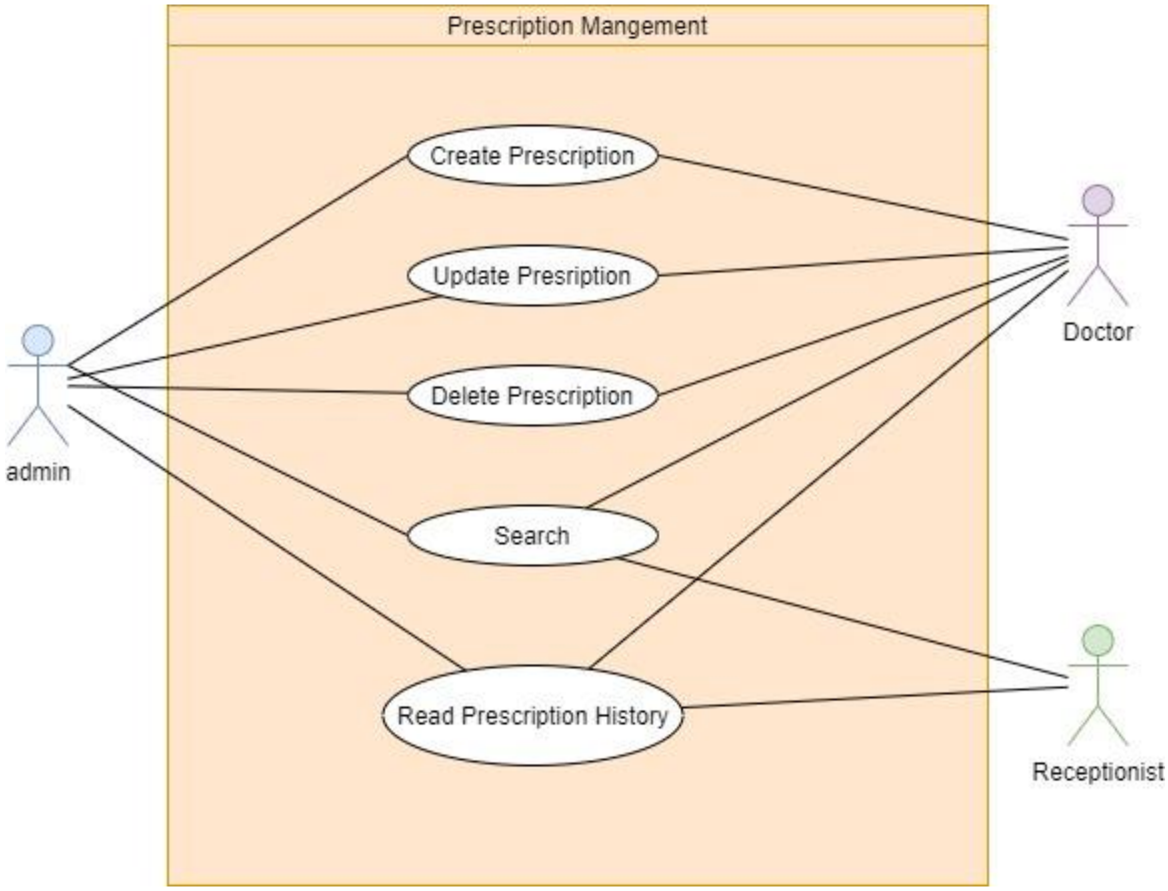


Figure 24 Sequence diagram of UMS

7.4.2. Prescription Management System

Use Case diagram



Activity Diagram

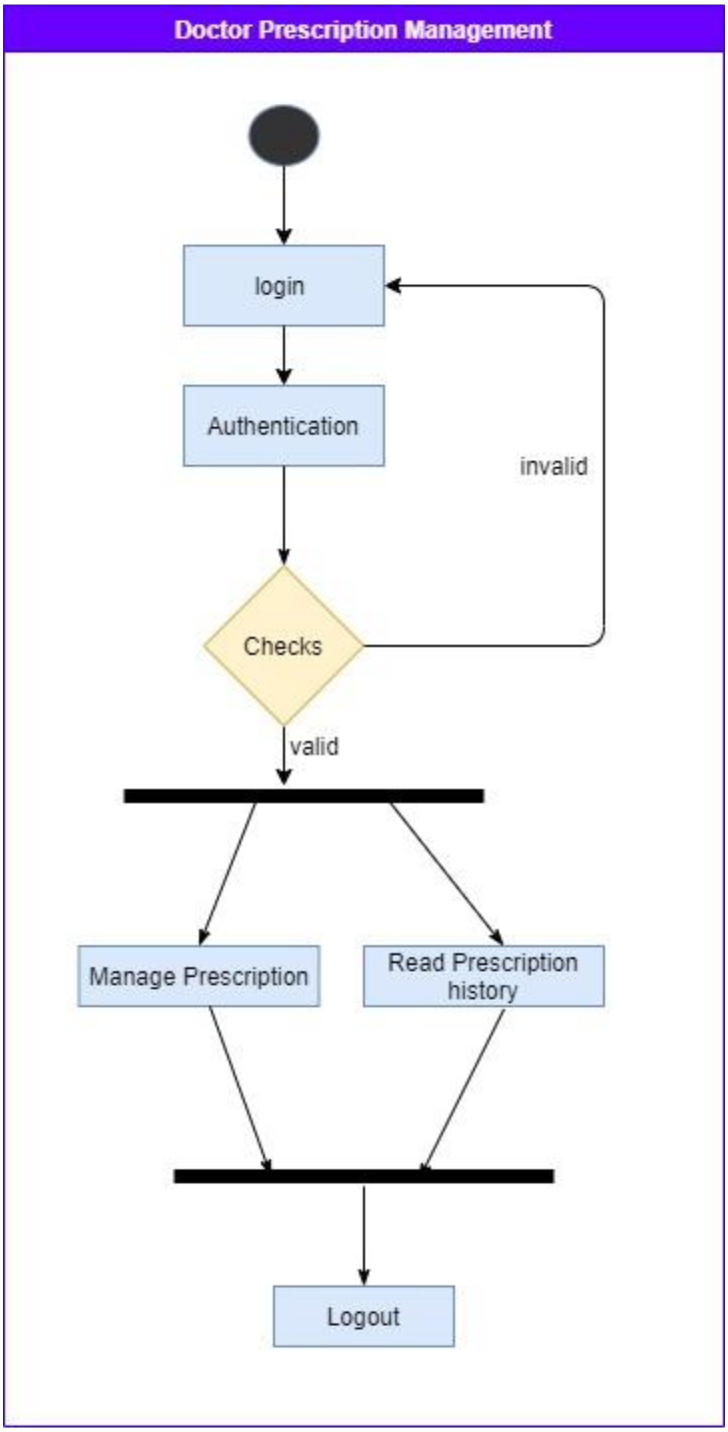


Figure 25 Activity Diagram of Doctor Prescription Management

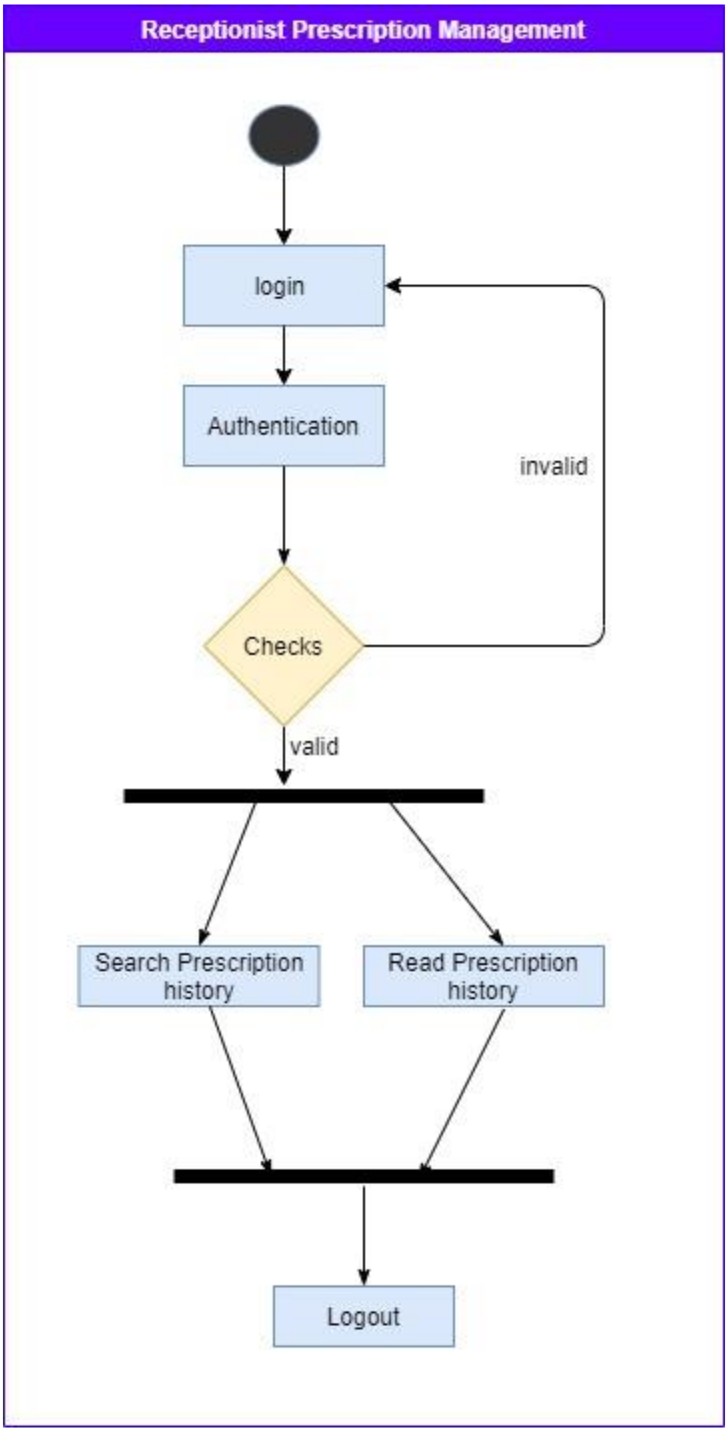


Figure 26 Activity Diagram of Receptionist Prescription Management

ER- Diagram
Class Diagram
Sequence Diagram

7.4.3. Data Dictionary

Entity name	Attributes	Description	data type and length	Nulls
Patient	id (pk) name email address contact dob gender blood	ID of patient Name of Patient Email of Patient Address of Patient Contact of Patient Date of Birth of Patient Gender of Patient Blood group of Patient	varchar(10) varchar(100) varchar(100) varchar(100) int DateField varchar(20) varchar(20)	No No Yes Yes Yes No No No
Doctor	id (pk) uname name email password address contact dob specialty gender blood	ID of doctor Username of doctor Name of Doctor Email of Doctor Login Password of Doctor Address of Doctor Contact of Doctor Date of Birth of Doctor Specialty of doctor Gender of Doctor Blood group of Doctor	varchar(10) varchar(100) varchar(100) varchar(100) varchar(100) varchar(100) int DateField varchar(50) varchar(20) varchar(20)	No No No No No Yes Yes Yes No No No
Receptionist	id (pk) uname name email password address contact dob gender blood	ID of receptionist Username of receptionist Name of receptionist Email of receptionist Login Password Address of receptionist Contact of receptionist Date of Birth of recept Gender of receptionist Blood groups of receptionist	varchar(10) varchar(100) varchar(100) varchar(100) varchar(100) varchar(100) int DateField varchar(20) varchar(20)	No No No No No Yes Yes Yes No No
	paymentID payMethod discount VAT totalAmount	Id of payment method of payment discount amount VAT amount total amount of order	varchar(10) varchar(20) int int int	No No Yes Yes Yes
prescription	id (pk) p_id (fk) name (fk)	Prescription id Patient id Name of Patient	varchar(25) varchar(25) varchar(100)	No No No

	prescription date	Prescription given to patient Date of prescription	varchar(100) date	No No
Patient_trash	id (pk) name email address contact dob gender blood	ID of deleted patient Name of deleted Patient Email of deleted Patient Address of deleted Patient Contact of deleted Patient Date of Birth of deleted Patient Gender of deleted Patient Blood group of deleted Patient	varchar(10) varchar(100) varchar(100) varchar(100) int DateField varchar(20) varchar(20)	No No Yes Yes Yes No No No

7.4.4. Covid-19 Detection

The continuous development of training and building model was carried out in three different custom architectures based on ResNet, VGG and LeNet. Comparison of these models is in section 7.4.5.

Data pre-processing:

All the models were trained using similar dataset and settings. All of them were trained for 10 epochs with Adam as an optimizer, loss function as binary_crossentropy, ReLU activation function were implemented to bring non-linearity in the network. Sigmoid was the go-to Activation function for every architecture before the output which classifies the image into either 1 (Covid) or 0 (Normal).

Similar data-preprocessing were carried out in all the architecture, the image of training and test dataset was resized to 224 X 224 and all the dataset were split into X_train, y_train, X_test, y_test as train image, train label, test image and test label respectively before training.

All of these three architectures were train on same dataset provided on Kaggle COVID-19 Radiography database: Dataset from the winner of COVID-19 dataset award by Kaggle Community. For training, 5019 images were used from which

15% of dataset were used for validation. 1400 separate images, 700 for each class in every network which were not used in train or valid was allocated for testing purpose. All the trained model was saved to '.h5' format. Since the accuracy from ResNet is better so far in comparison to other algorithms. Therefore, ResNet saved model has been integrated in Django. In Django, user clicks the button and the system records image for and outputs the result showing either the given image is of class 0(Normal) or 1(Covid).

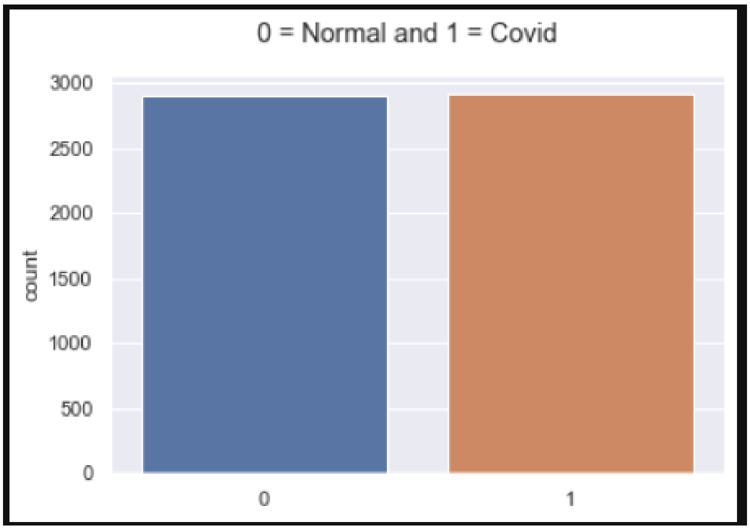


Figure 27 Data distribution of class 0 and 1 in train set

```
(Left) - Normal (0) Vs (Right) - COVID (1)
--------------------------------------------------------------------------------
----
```

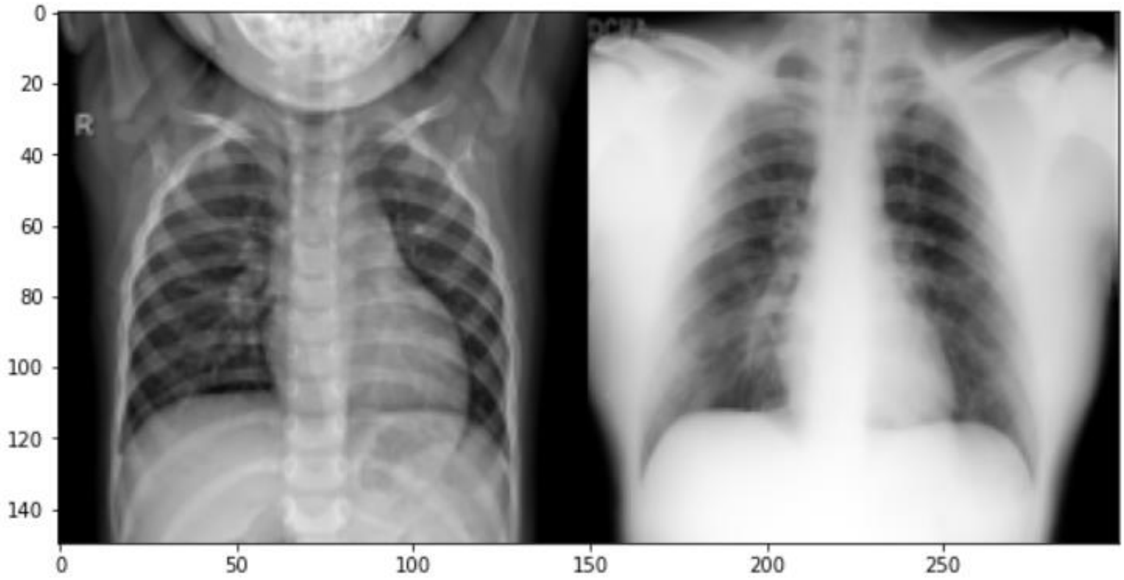


Figure 28 Sample image from training dataset

```
(Left) - Normal (0) Vs (Right) - COVID (1)
--------------------------------------------------------------------------------
----
```

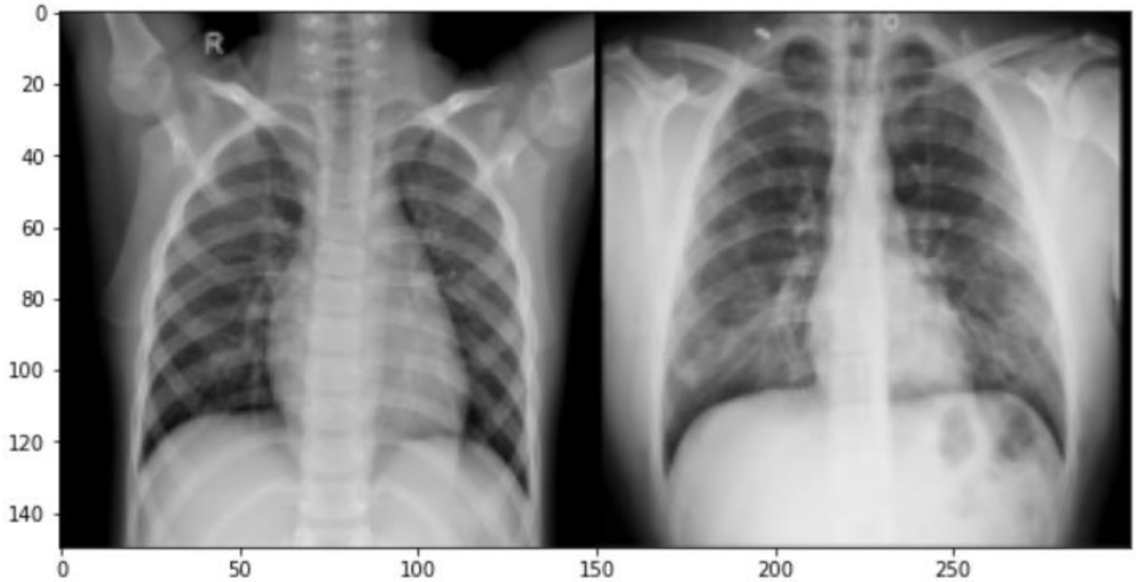


Figure 29 sample image from training dataset

Custom ResNet:

This architecture is based on a functional model which is inspired by ResNet50 architecture. The architecture used here consists of 56 total layer including input, batch normalization and activation layer as shown in Figure 31. Among the three, this architecture gave the highest accuracy of 97.35% on test set. F1- score for both classes were 97%. The model was trained for 10 epochs taking the average time of 7.8 minutes per epoch making it the 2nd fastest architecture despite being the largest model because of the skip connection technique.

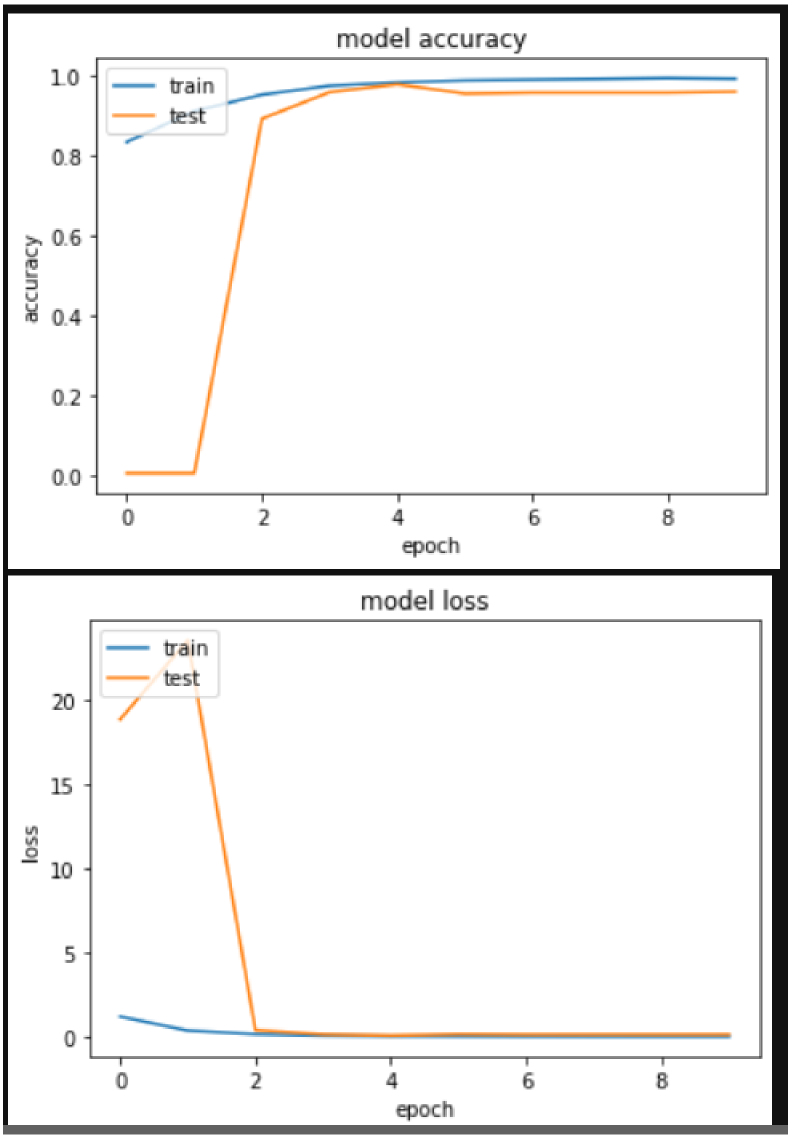
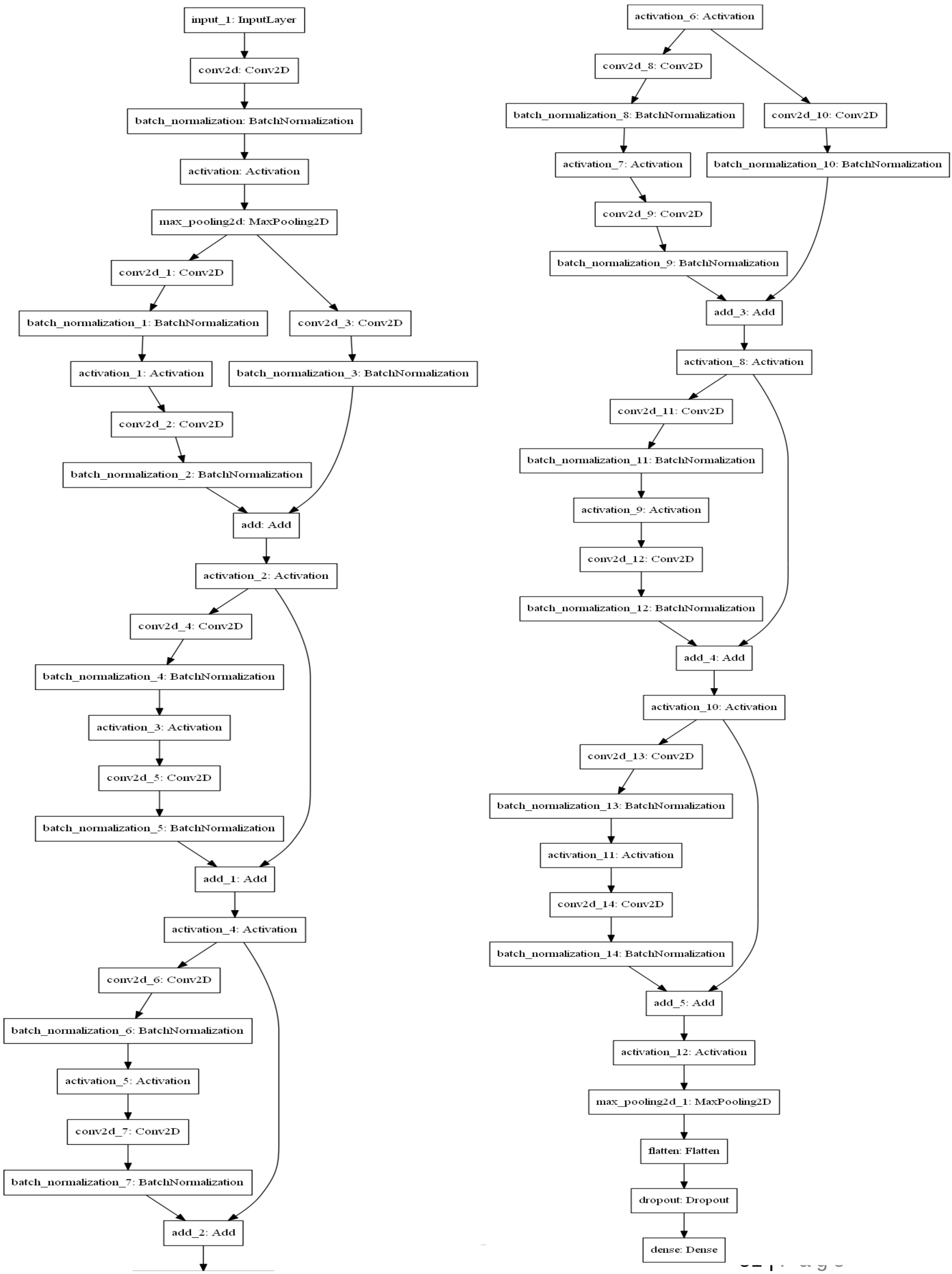


Figure 30 Train vs Test (loss and accuracy)



input_1: InputLayer
conv2d: Conv2D
batch_normalization: BatchNormalization
activation: Activation
max_pooling2d: MaxPooling2D
conv2d_1: Conv2D
batch_normalization_1: BatchNormalization
conv2d_3: Conv2D
activation_1: Activation
batch_normalization_3: BatchNormalization
conv2d_2: Conv2D
batch_normalization_2: BatchNormalization
add: Add
activation_2: Activation
conv2d_4: Conv2D
batch_normalization_4: BatchNormalization
activation_3: Activation
conv2d_5: Conv2D
batch_normalization_5: BatchNormalization
add_1: Add
activation_4: Activation
conv2d_6: Conv2D
batch_normalization_6: BatchNormalization
activation_5: Activation
conv2d_7: Conv2D
batch_normalization_7: BatchNormalization
add_2: Add
activation_6: Activation
conv2d_8: Conv2D
batch_normalization_8: BatchNormalization
conv2d_10: Conv2D
activation_7: Activation
batch_normalization_10: BatchNormalization
conv2d_9: Conv2D
batch_normalization_9: BatchNormalization
add_3: Add
activation_8: Activation
conv2d_11: Conv2D
batch_normalization_11: BatchNormalization
activation_9: Activation
conv2d_12: Conv2D
batch_normalization_12: BatchNormalization
add_4: Add
activation_10: Activation
conv2d_13: Conv2D
batch_normalization_13: BatchNormalization
activation_11: Activation
conv2d_14: Conv2D
batch_normalization_14: BatchNormalization
add_5: Add
activation_12: Activation
max_pooling2d_1: MaxPooling2D
flatten: Flatten
dropout: Dropout
dense: Dense

Figure 31 Flow of every layer based on Resnet architecture

On the first try of training on ResNet for 10 epochs the model was overfitted as the validation accuracy was really low while training accuracy was above 98%. After adding a “ReduceLrOnPlateu” which reduces the learning rate when a validation accuracy has stopped improving and “Dropout layer” while training, the problem of overfit was solved and we were able to build a model with training and validation accuracy of 99% and 95% respectively. Upon further evaluating the model on test set the accuracy and loss of the model is found to be 97.35% and 0.144 respectively as shown in Figure 32. Also, the result on confusion matrix were also pretty good with only 46 images being predicted falsely among 1400 images. There is still a room of improvement in this model as we can see out of 46 falsely predicted images, 45 were False Positive which is crucial in medical sector. The final model use in a web app is based on this architecture as it gives the most prominent results.

```
print("Loss of the model is - " , resnet.evaluate(X_test, label_test)[0])
print("Accuracy of the model is - " ,resnet.evaluate(X_test,label_test)[1]*100 , "%")

44/44 [==============================] - 19s 427ms/step - loss: 0.1444 - accuracy: 0.9736
Loss of the model is -  0.14442287385463715
44/44 [==============================] - 20s 445ms/step - loss: 0.1444 - accuracy: 0.9736
Accuracy of the model is -  97.35714197158813 %
```

Figure 32 Test set evaluation

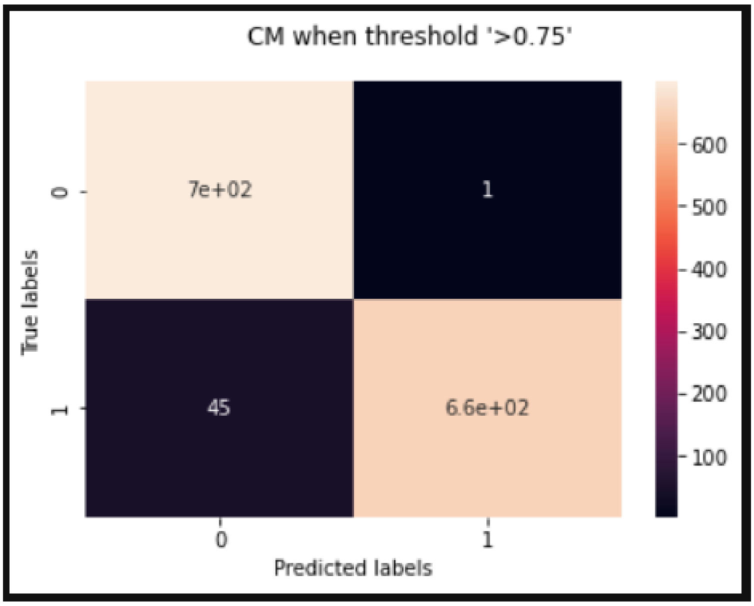


Figure 33 Confusion Matrix of Resnet when threshold is set to '0.75'

VGG:

This architecture is inspired from VGG16 and is similar to vgg16 architecture except it has dropout after the first dense layer and Batch-Normalization after each layer to avoid overfitting and potentially faster training time. It is 27 layers deep including input layer and batch-normalization. This is sequential based model and was able to output the peak of 95% validation accuracy. F1- score for both classes were 95%. The model was trained for 10 epochs taking the average time of 55.9 minutes per epoch making it the slowest architecture among the three.

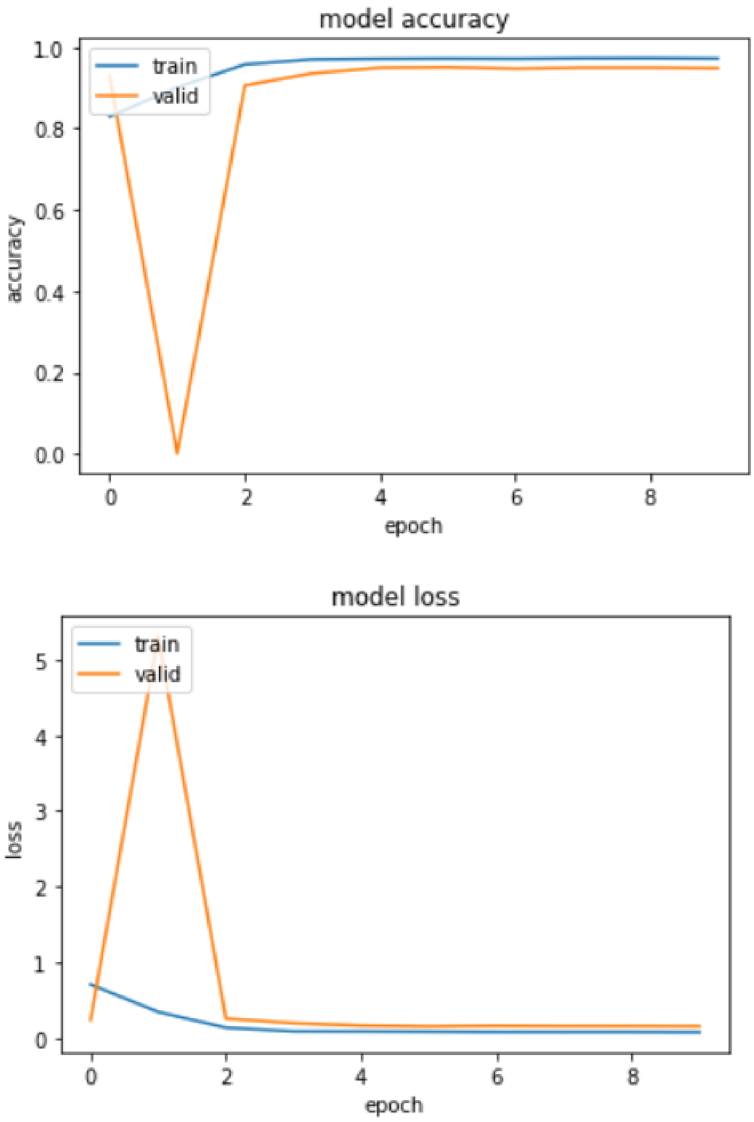


Figure 34 train vs validation of vgg (loss and accuracy)

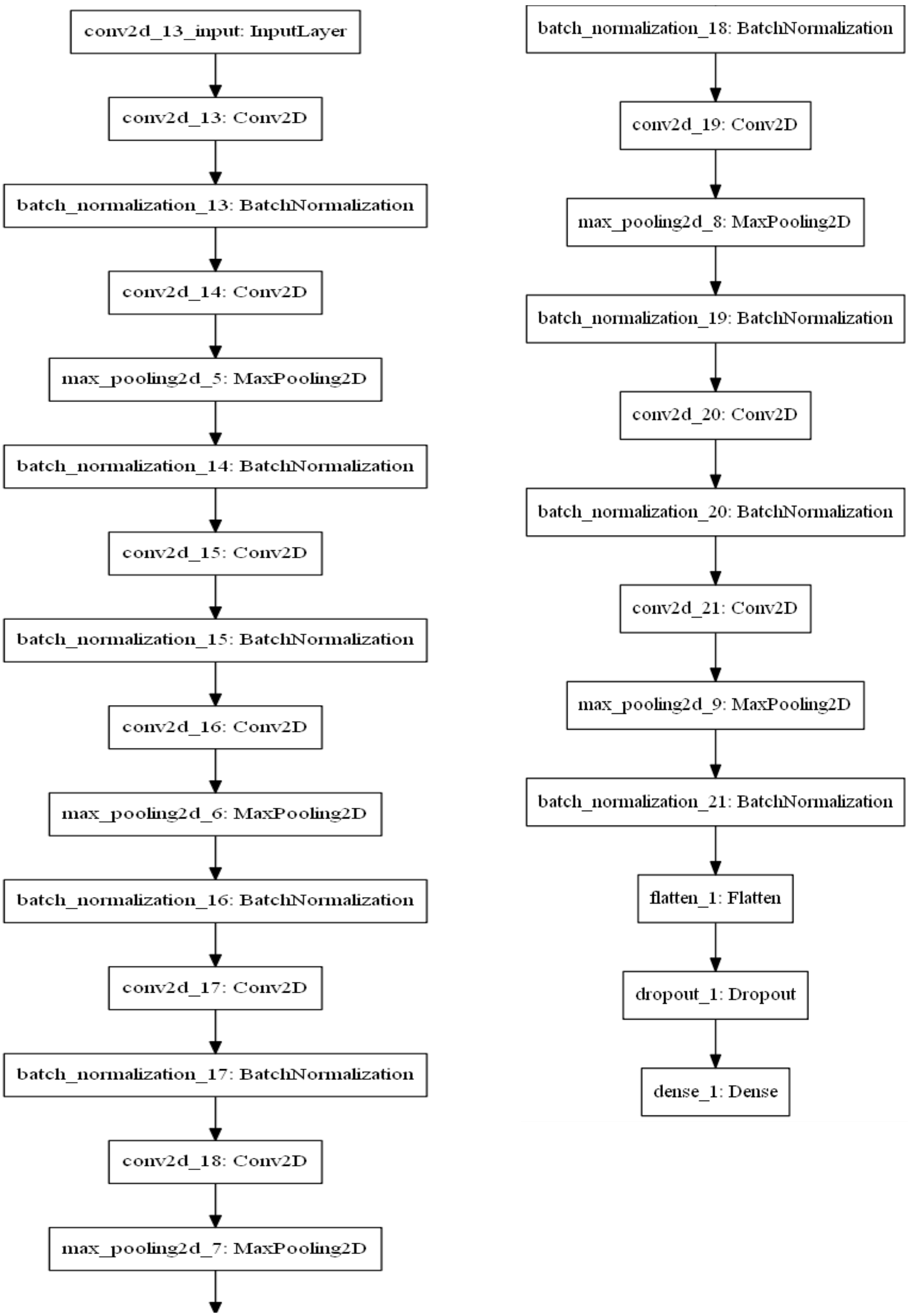


Figure 35 Model architecture based on Vgg16

Model was trained on same settings as ResNet i.e., using ReduceLRonPlateu, Adam optimizer, loss as binary_crossentropy and sigmoid as an activation function at Fully connected layer. This architecture was right below ResNet with training and validation accuracy of 99% and 95% respectively and loss of 0.07 on train and 0.15 on validation as represented in graph Figure 34. Upon further evaluating the model on test set the accuracy and loss of the model is found to be 96.57% and 0.142 respectively as shown in Figure 36. 66 images had been predicted falsely among 1400 images. This model was slightly less reliable than ResNet and can be improved as out of 66 falsely predicted images, 64 were False Negative when the threshold is set to 0.7.

```
44/44 [==============================] - 181s 4s/step - loss: 0.1422 - accuracy: 0.9657
Loss of the model is -  0.14220811426639557
44/44 [==============================] - 182s 4s/step - loss: 0.1422 - accuracy: 0.9657
Accuracy of the model is -  96.57142758369446 %
```

Figure 36 Evaluation of VGG on test set

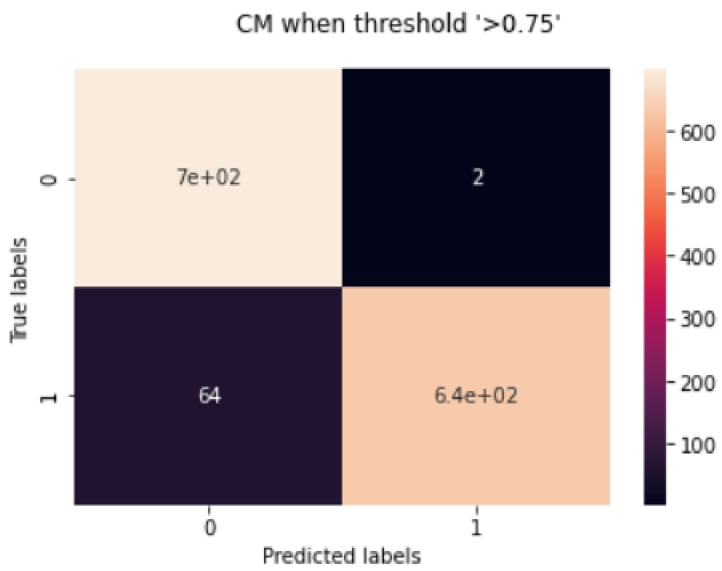


Figure 37 CM of VGG on test set (threshold 0.75)

LeNet:

LeNet is a very small network which contains the basic modules of deep learning Convolutional layer, Pooling layer and FC layer. This architecture performs the worst out of the three chosen architecture with validation accuracy of 77 percent. The average training time was 7.4 minutes each epoch which is the fastest among the 3 which was expected as it is the smallest network.

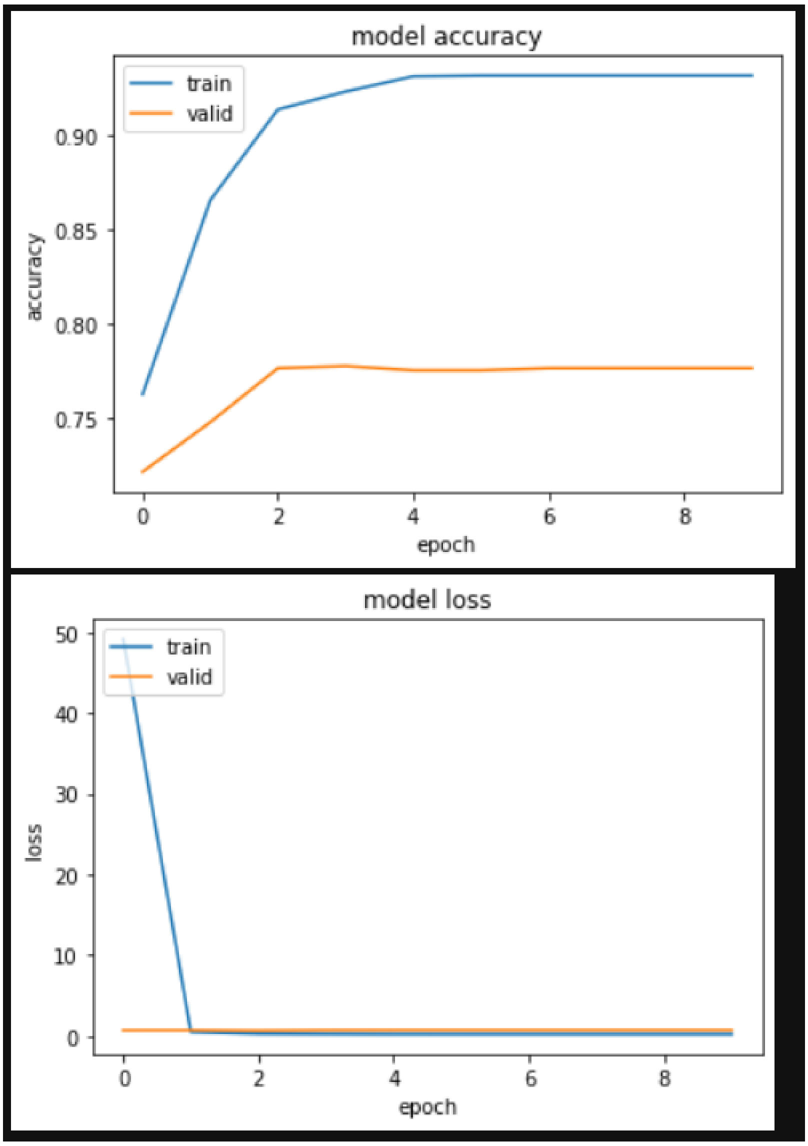


Figure 38 train vs valid of lenet while training (accuracy and loss)

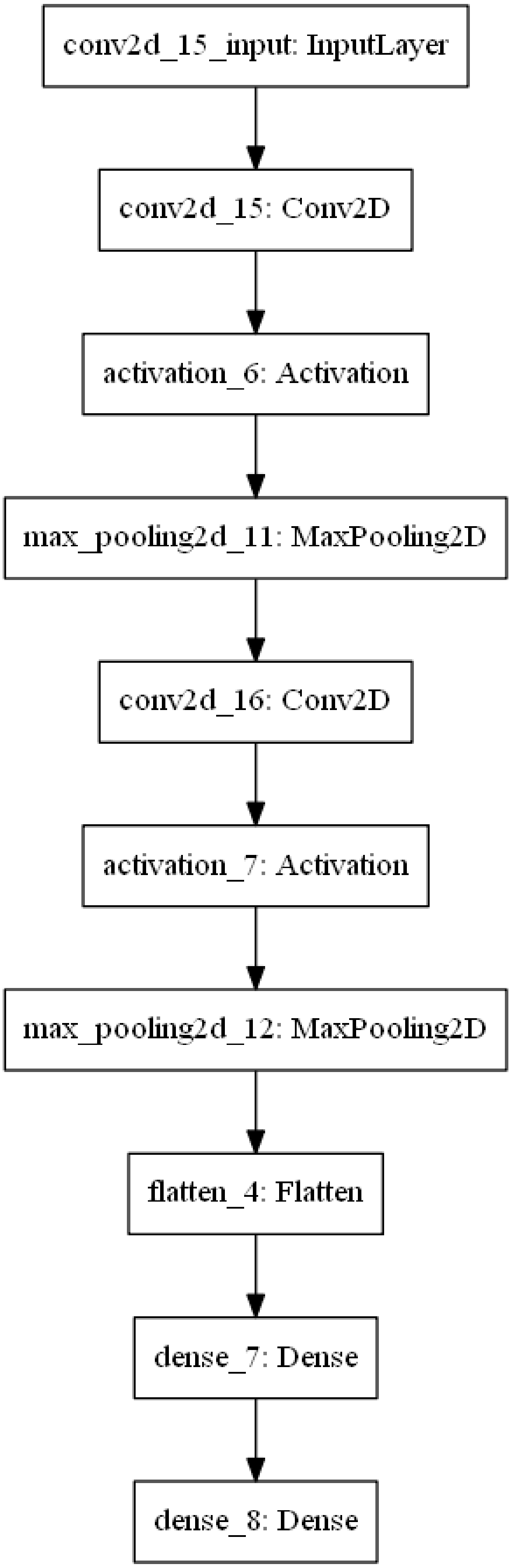


Figure 39 lenet architecture

7.4.5. Comparison

While testing, Threshold of 0.75 was set to evaluate test set in every network meaning that the model will only predict the image as 0 (Normal) if and only if the model is at least 75% sure that the given image is Normal otherwise it will predict it as 1 (Covid). This is very crucial step if you are dealing with medical domains like covid detection.

Train Vs Validation (Accuracy):

Model	Valid Accuracy	Train Accuracy
ResNet	96 %	99 %
VGG	95 %	97 %
LeNet	78 %	93 %

Table 2 Peak valid and train accuracy of each model

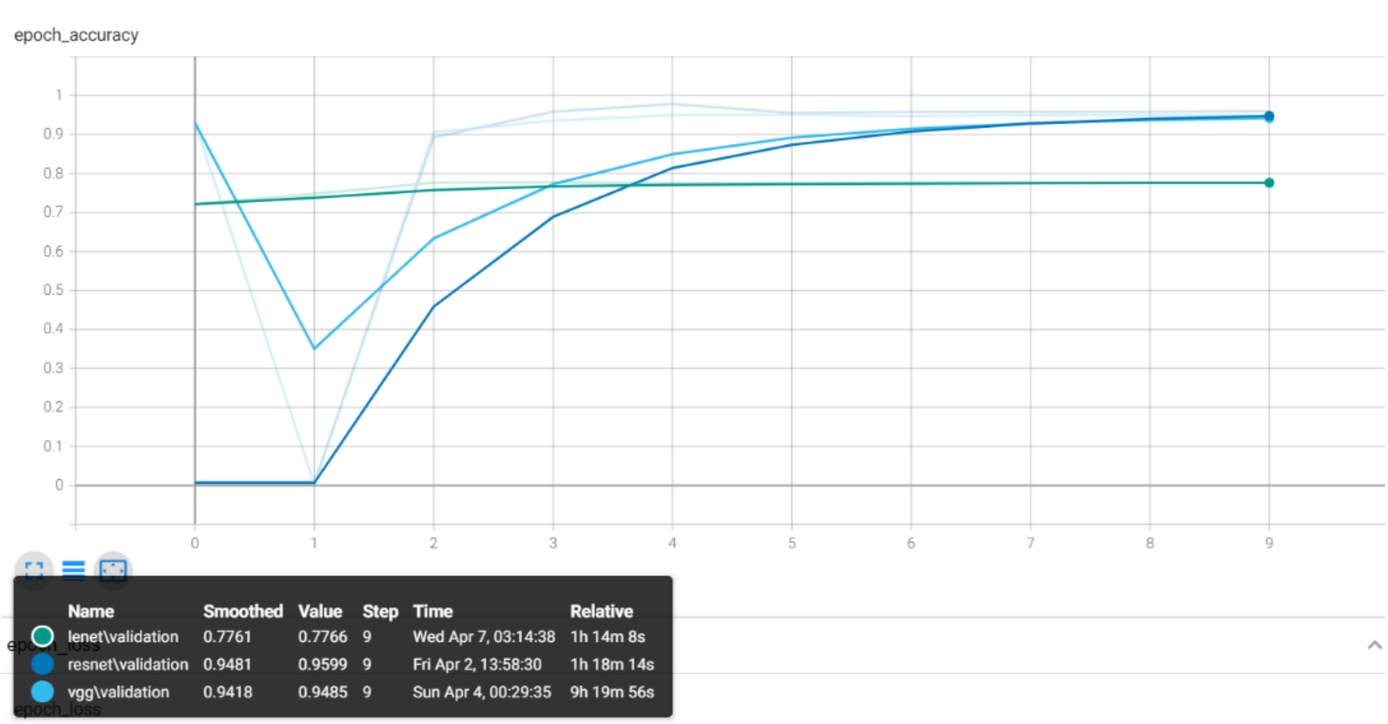


Figure 40 Time taken and Validation accuracy of each model during training

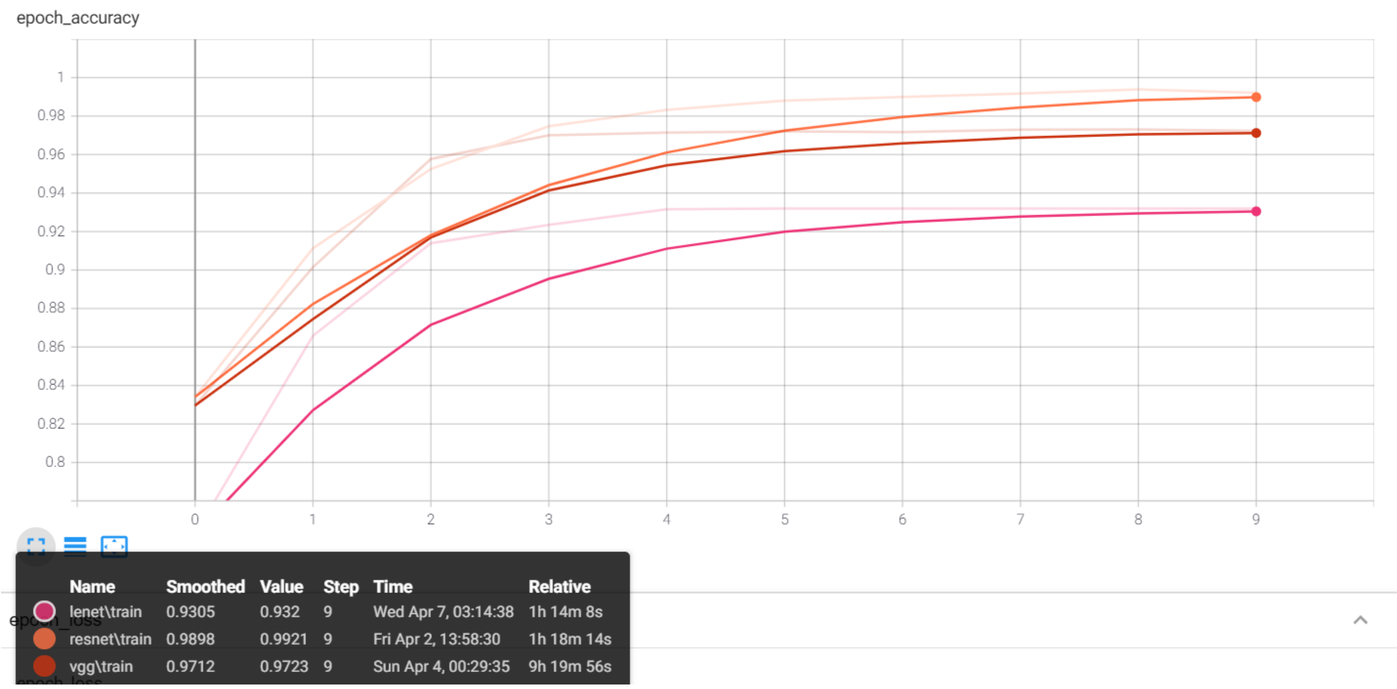


Figure 41 Training Accuracy in for each model during training

Both Validation and Train accuracy of ResNet network was found to be the highest with 96 and 99 percent respectively whereas LeNet has lowest with 78 % valid and 93 % training accuracy. Training time for VGG was the highest with 9 hour and 19 minutes to complete 10 epochs where training time for ResNet and LeNet was almost similar as shown in Figure 42.

Train vs Validation (Loss):

Model	Train Loss	Validation Loss
ResNet	0.02	0.14
VGG	0.07	0.15
LeNet	0.18	0.69

Table 3 Lowest valid and train Loss of each model

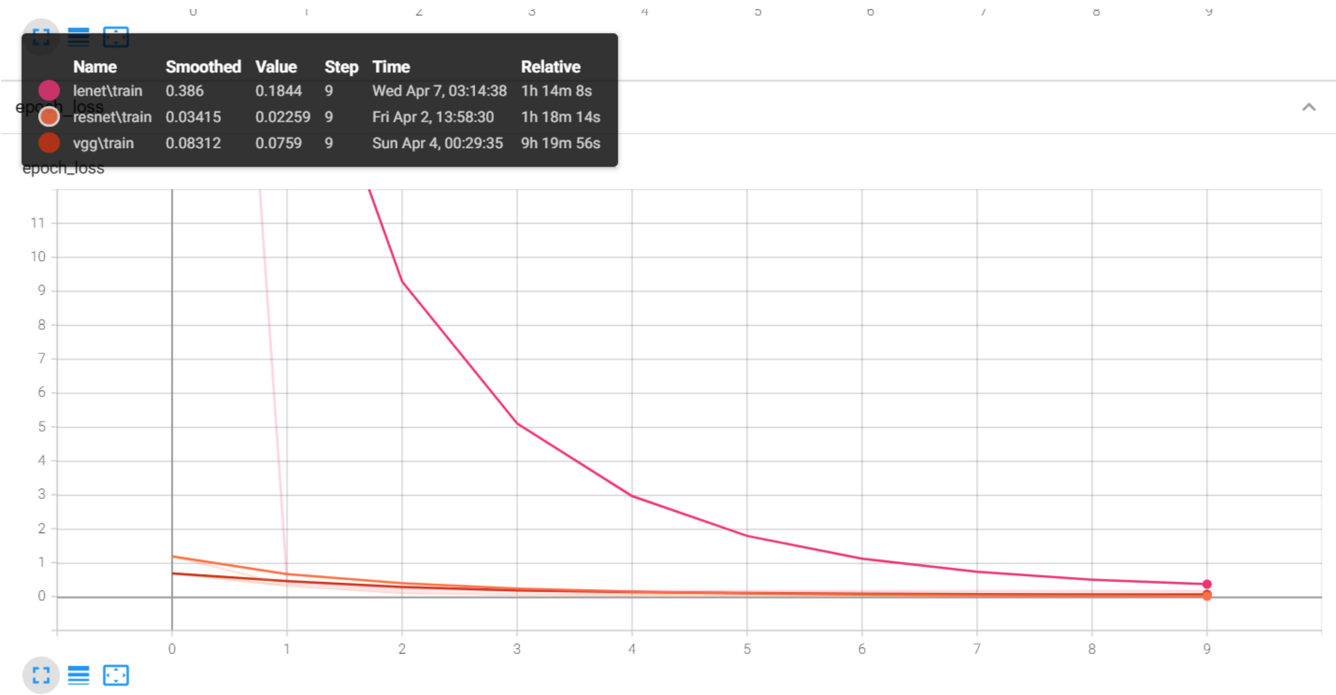


Figure 42 Time taken and Train Loss of each model during training

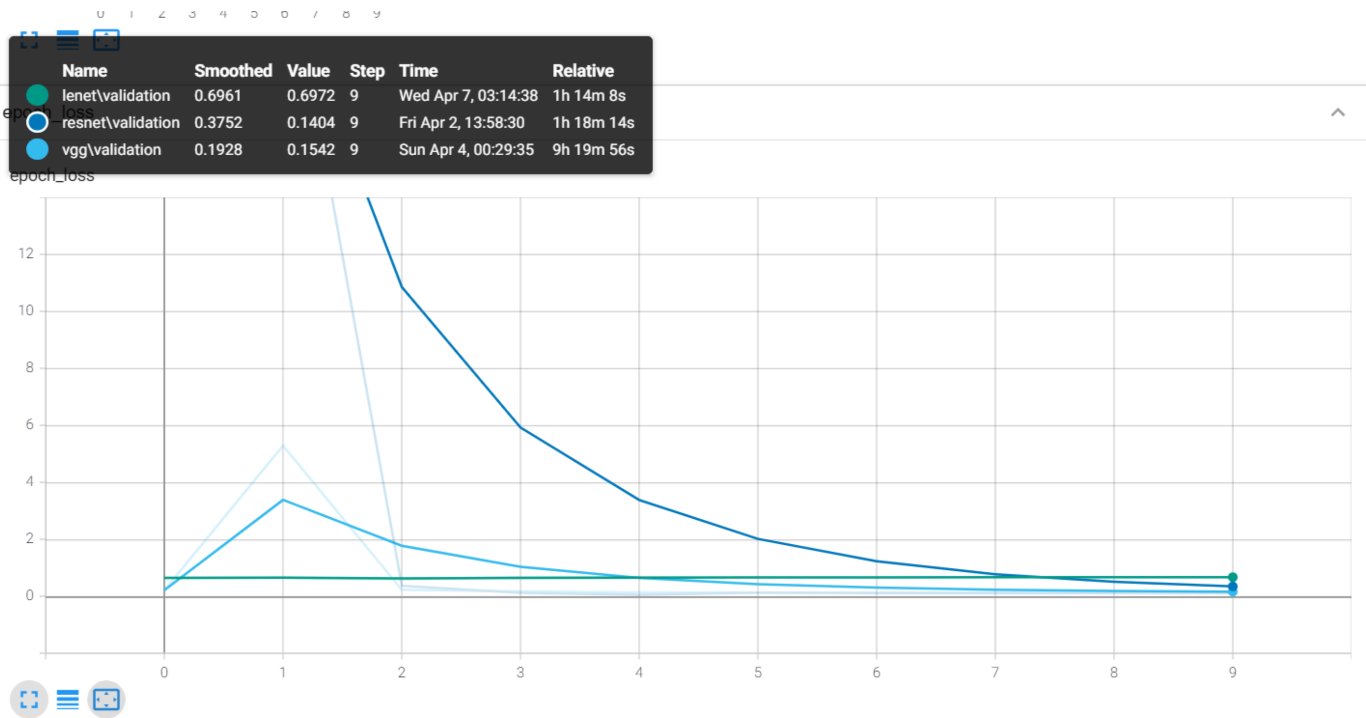


Figure 43 Validation loss in each model during training

Similarly, in case of loss ResNet performs slightly better than VGG with 0.02 loss on training and 0.14 on validation. LeNet being the fastest to train but while taking away the accuracy and loss percentage. From the graphs and tables, we can observe that ResNet based structure was able to perform really better in really short training time as compare to VGG.

Test Set Evaluation:

Model	Accuracy	Precision	Recall	F1-score
ResNet	98	97	97	97
VGG	96	96	95.5	96
LeNet	82	85	60	70.34

Figure 44 Test Set Evaluation

Further evaluating the model into test set, we observe that the ResNet was able to beat VGG and LeNet in evaluating test set also. LeNet performed worst with the accuracy of 82 percent and f1-score of 70 while ResNet has 98 percent accuracy and 97% accuracy being the best among 3.

7.5. Wireframe

7.5.1. Home page:

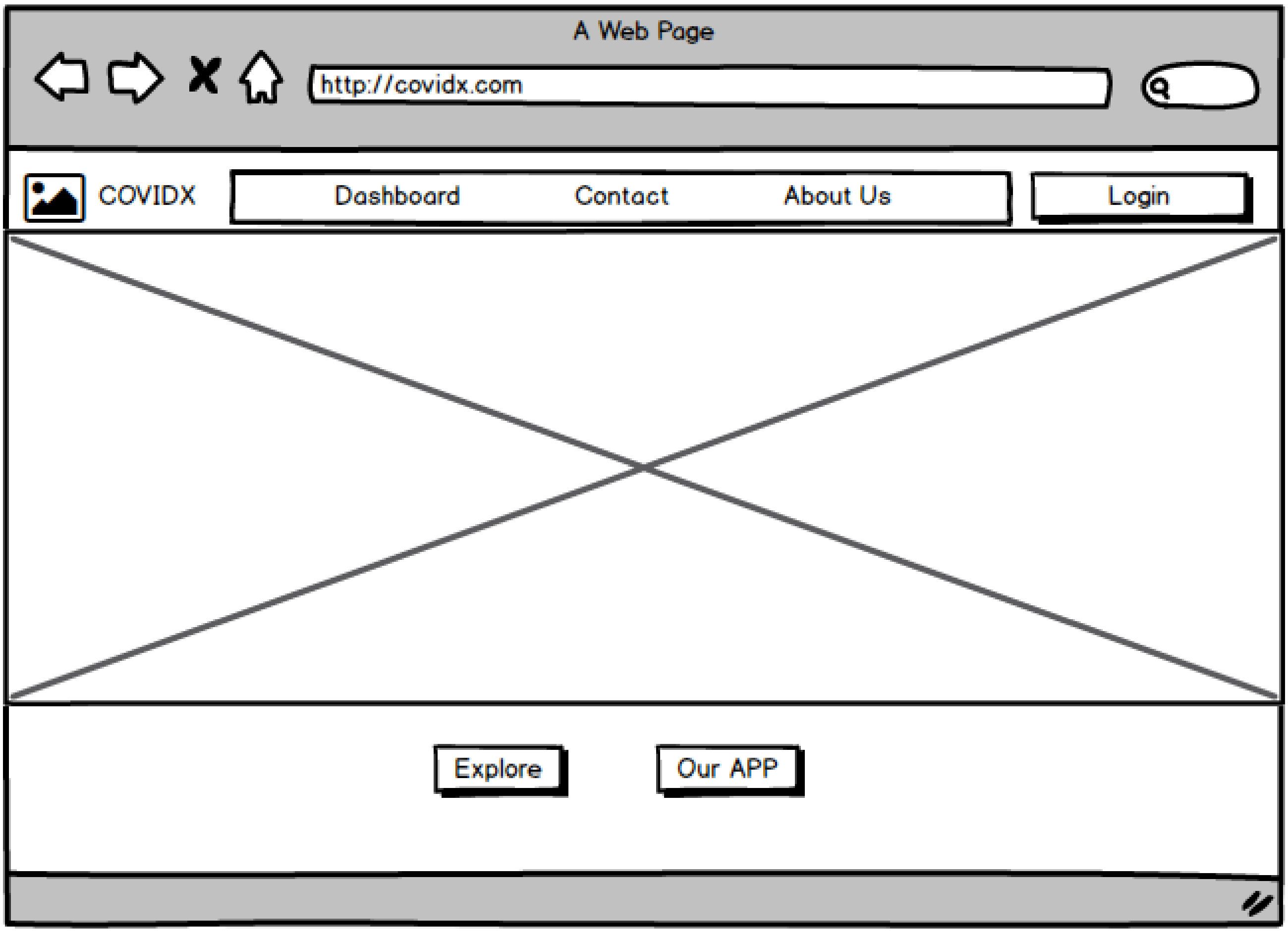


Figure 45 Home Page

7.5.2. Login

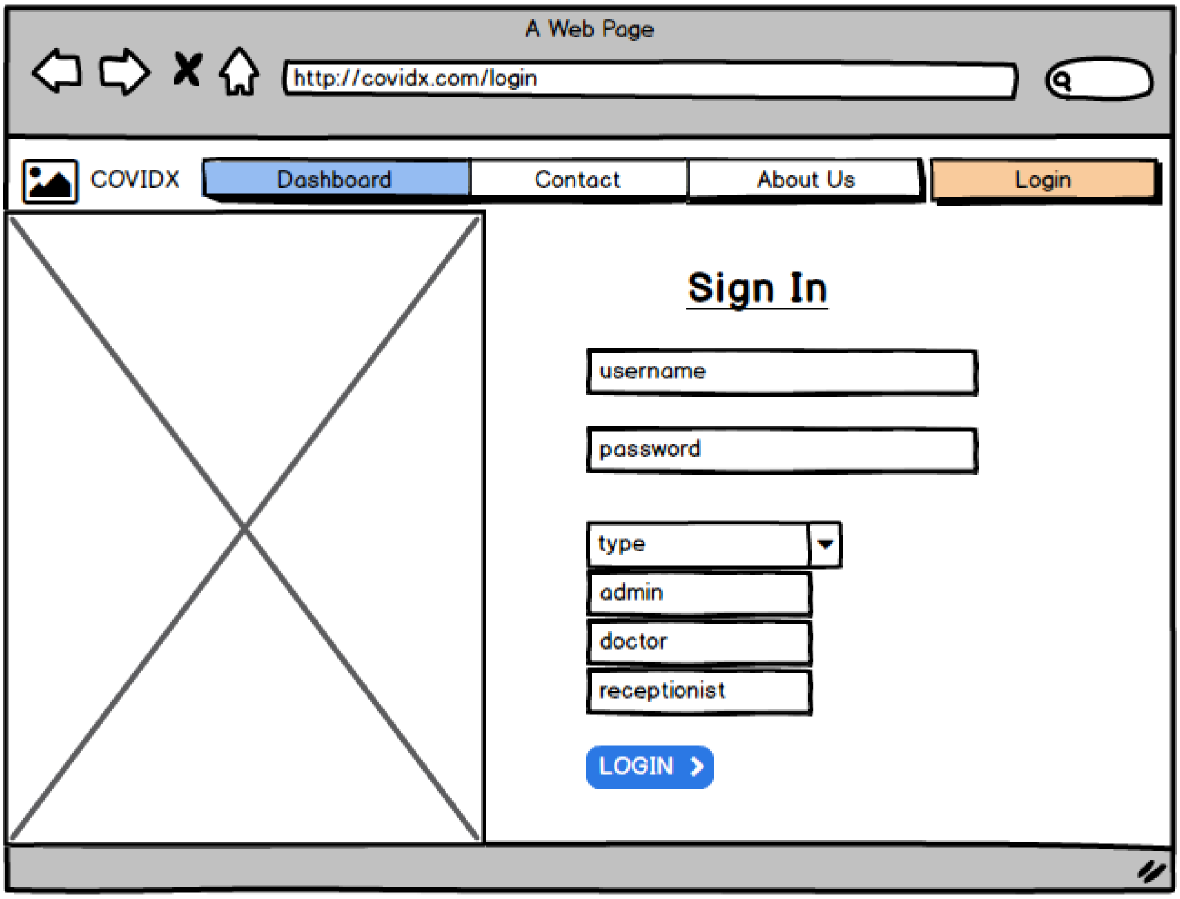


Figure 46 Wireframe for Login page

7.5.3. Doctor

Doctor/dashboard:

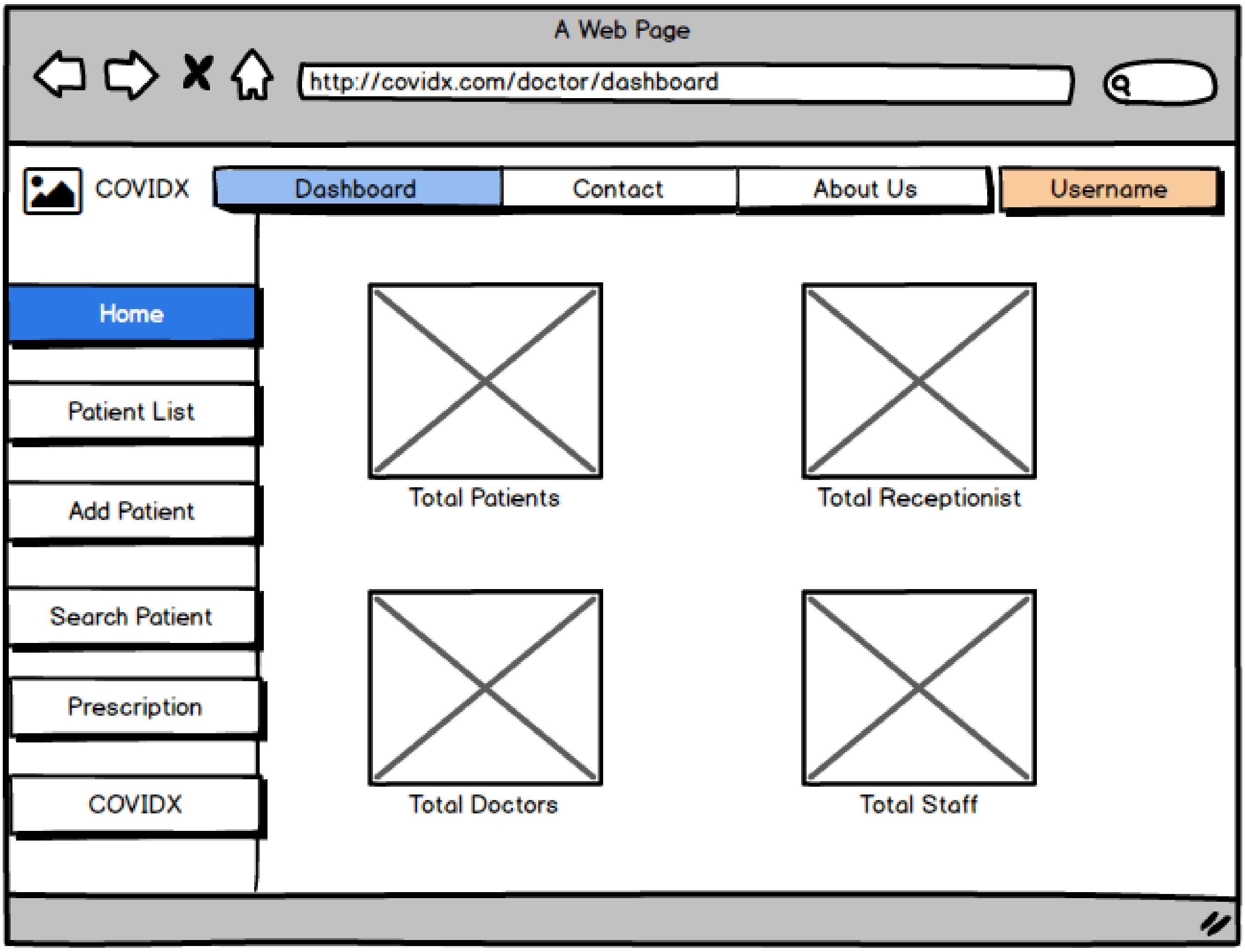


Figure 47 Wireframe for Doctor Dashboard page

Doctor/view_patient:

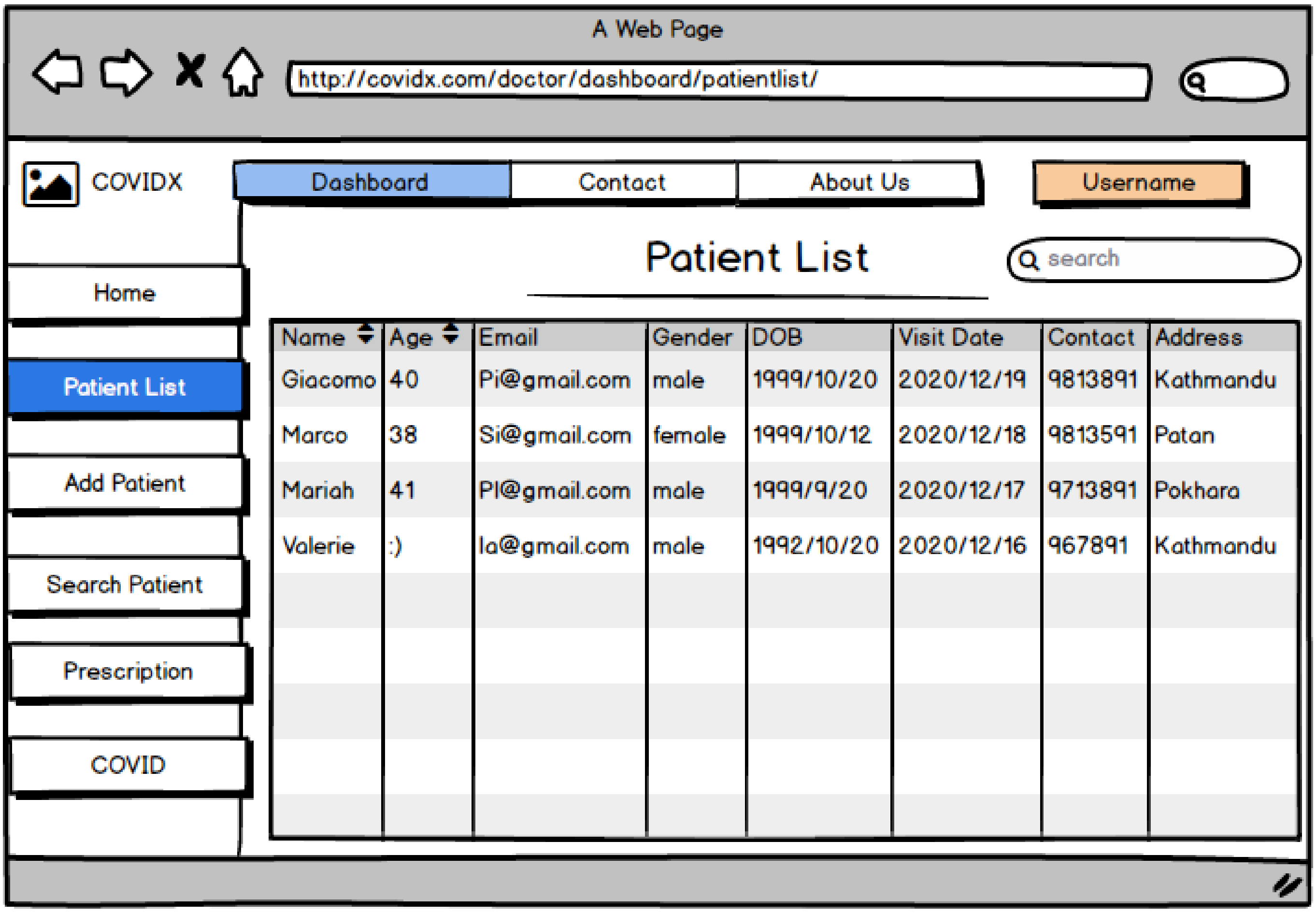


Figure 48 Wireframe for view patient page(doctor)

Doctor/add_patient

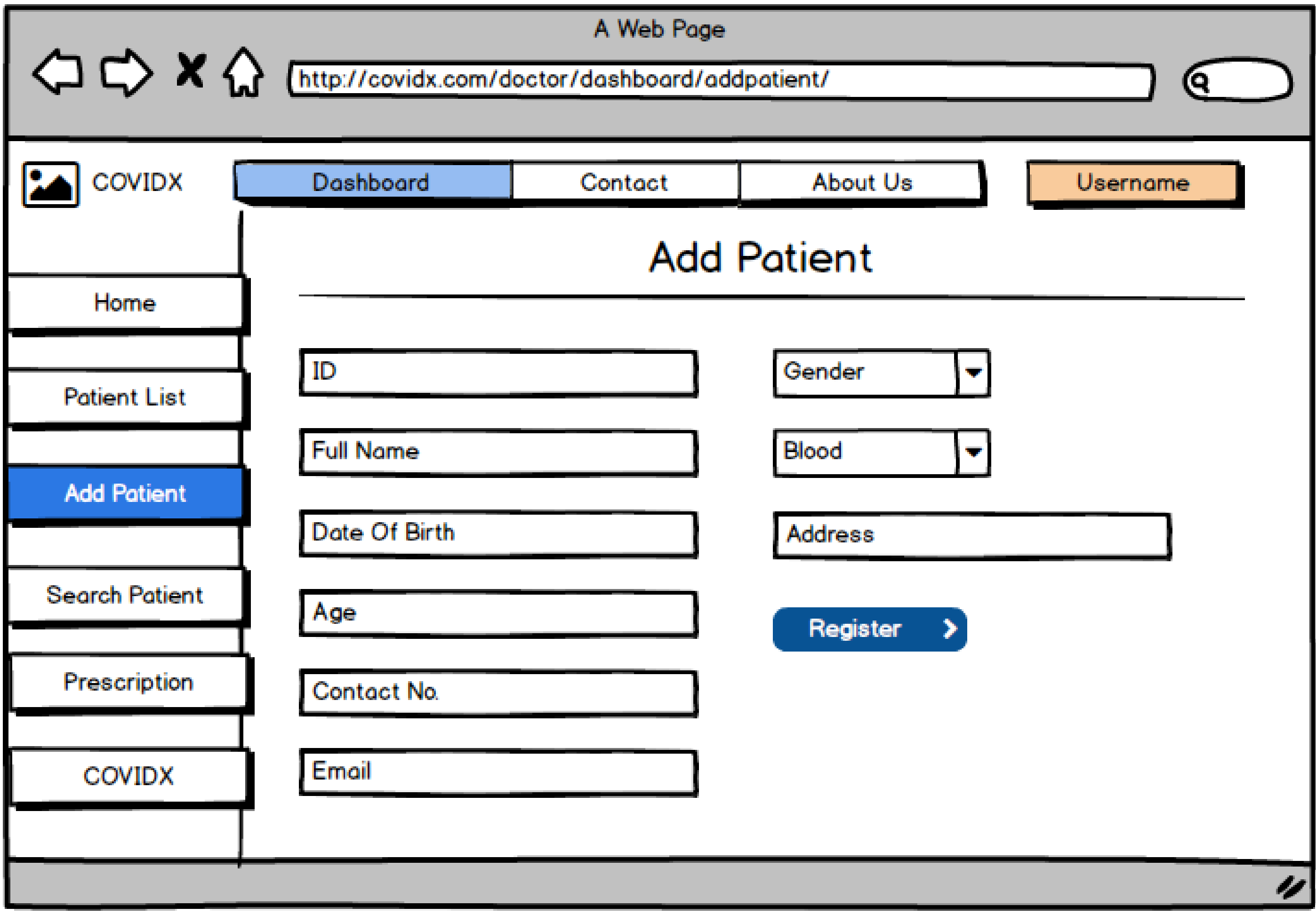


Figure 49 Wireframe for Add Patient page (Doctor)

Doctor/give_prescription:

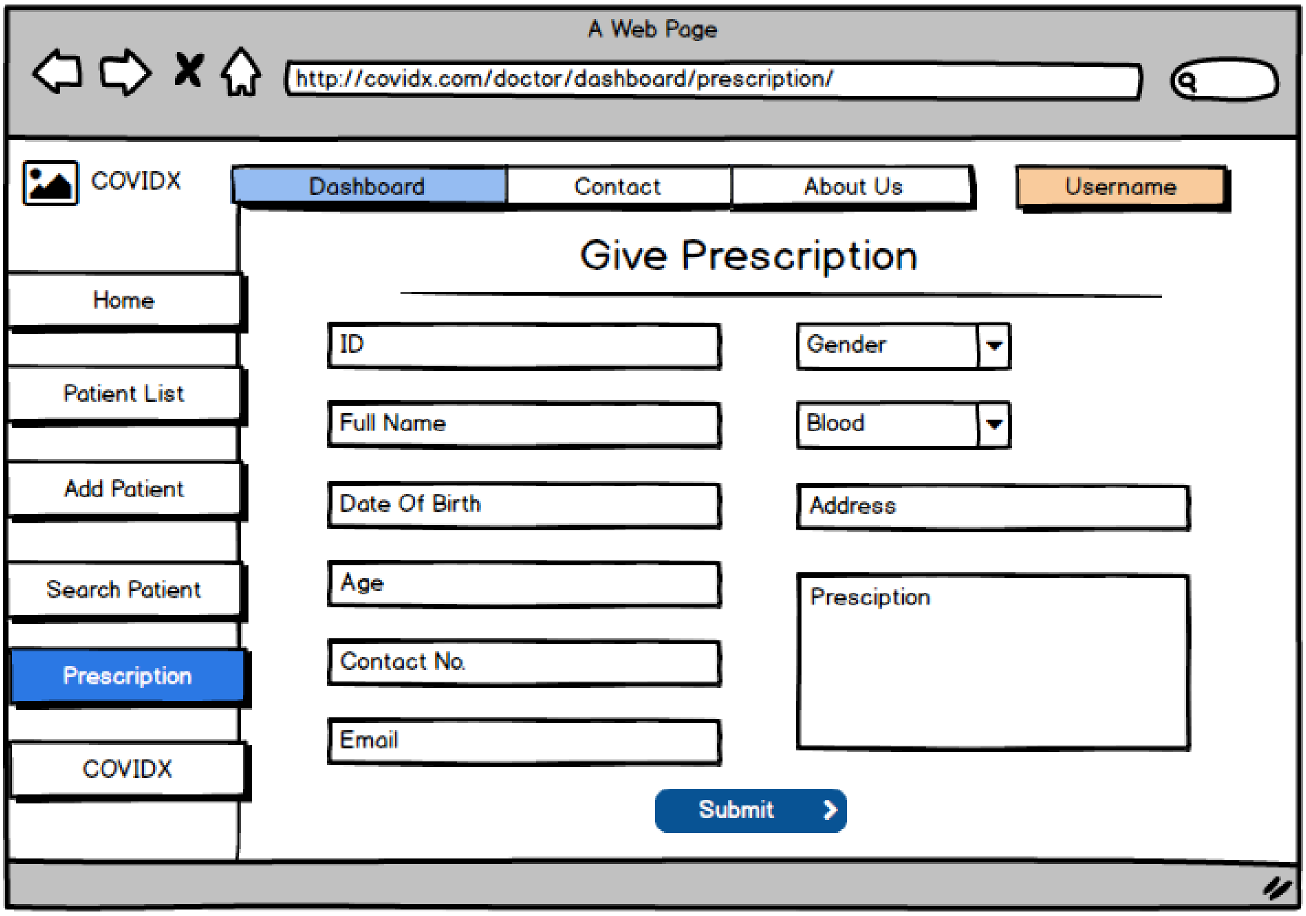


Figure 50 Wireframe for Prescription page (Doctor)

Doctor/covid:

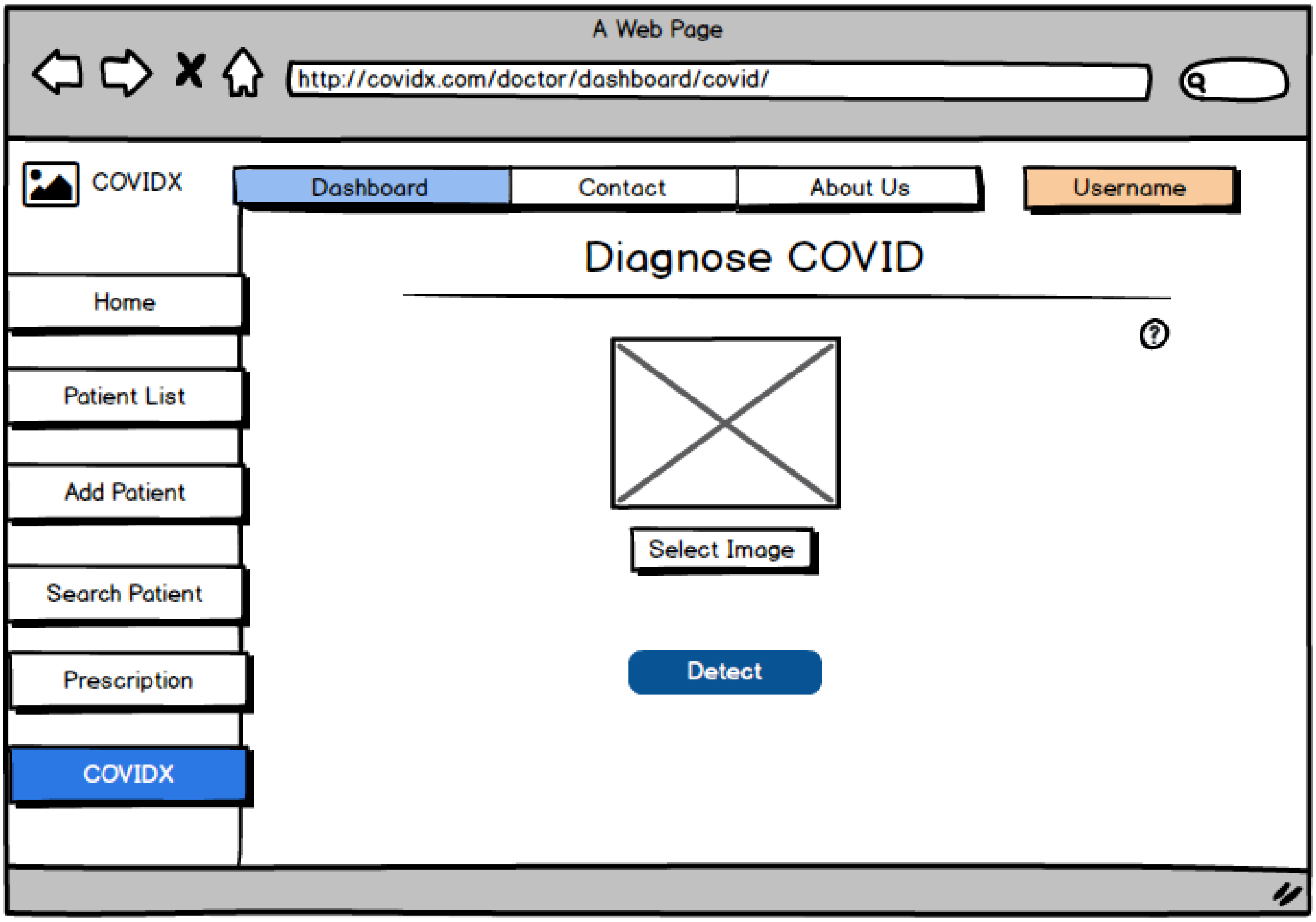


Figure 51 Wireframe for COVID detection page

Doctor/covid/result:

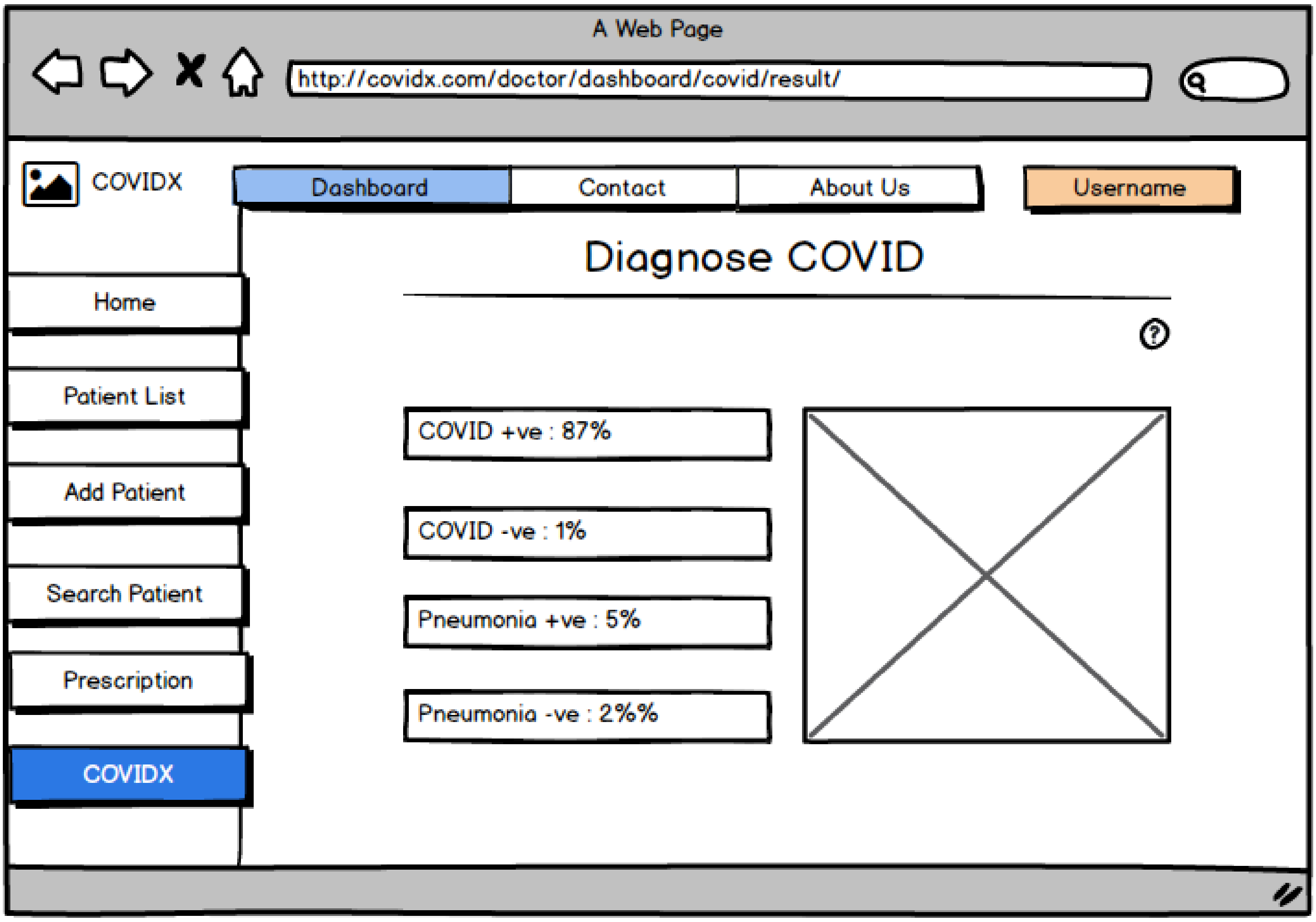


Figure 52 COVID result page

7.5.4. Receptionist

Receptionist/dashboard:

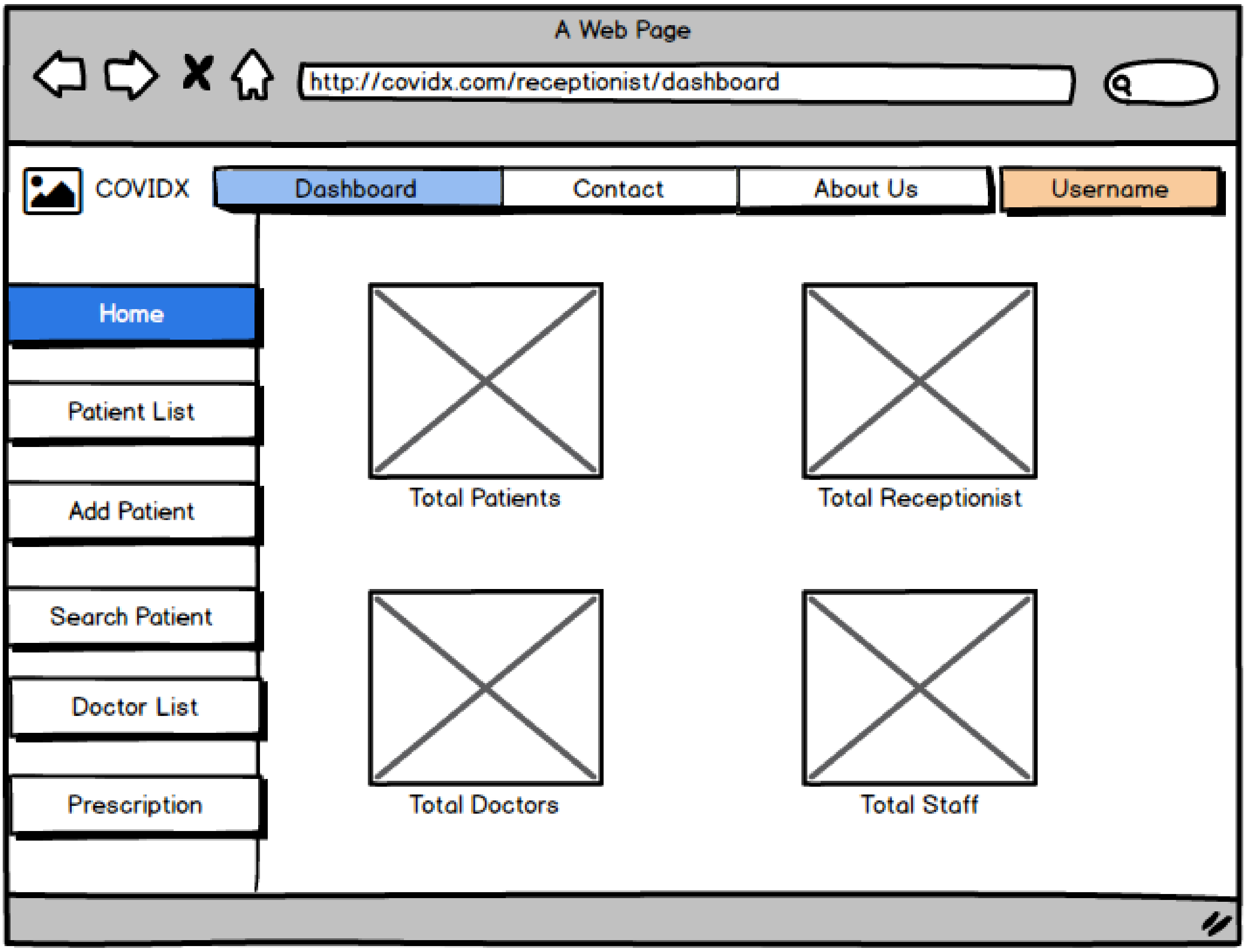


Figure 53 Receptionist dashboard page

Receptionist/view_patient

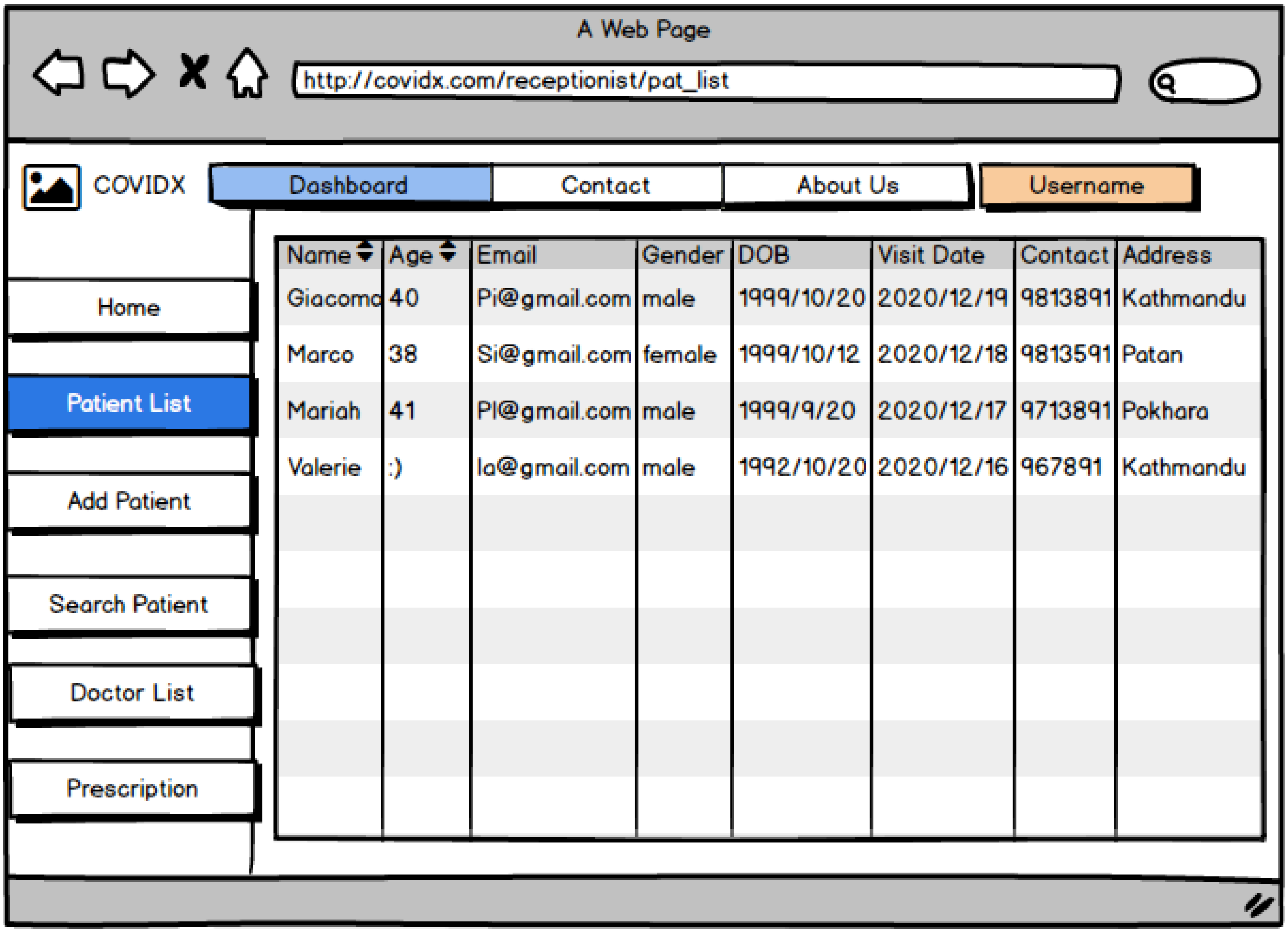


Figure 54 View Patient page(Receptionist)

Receptionist/add_patient

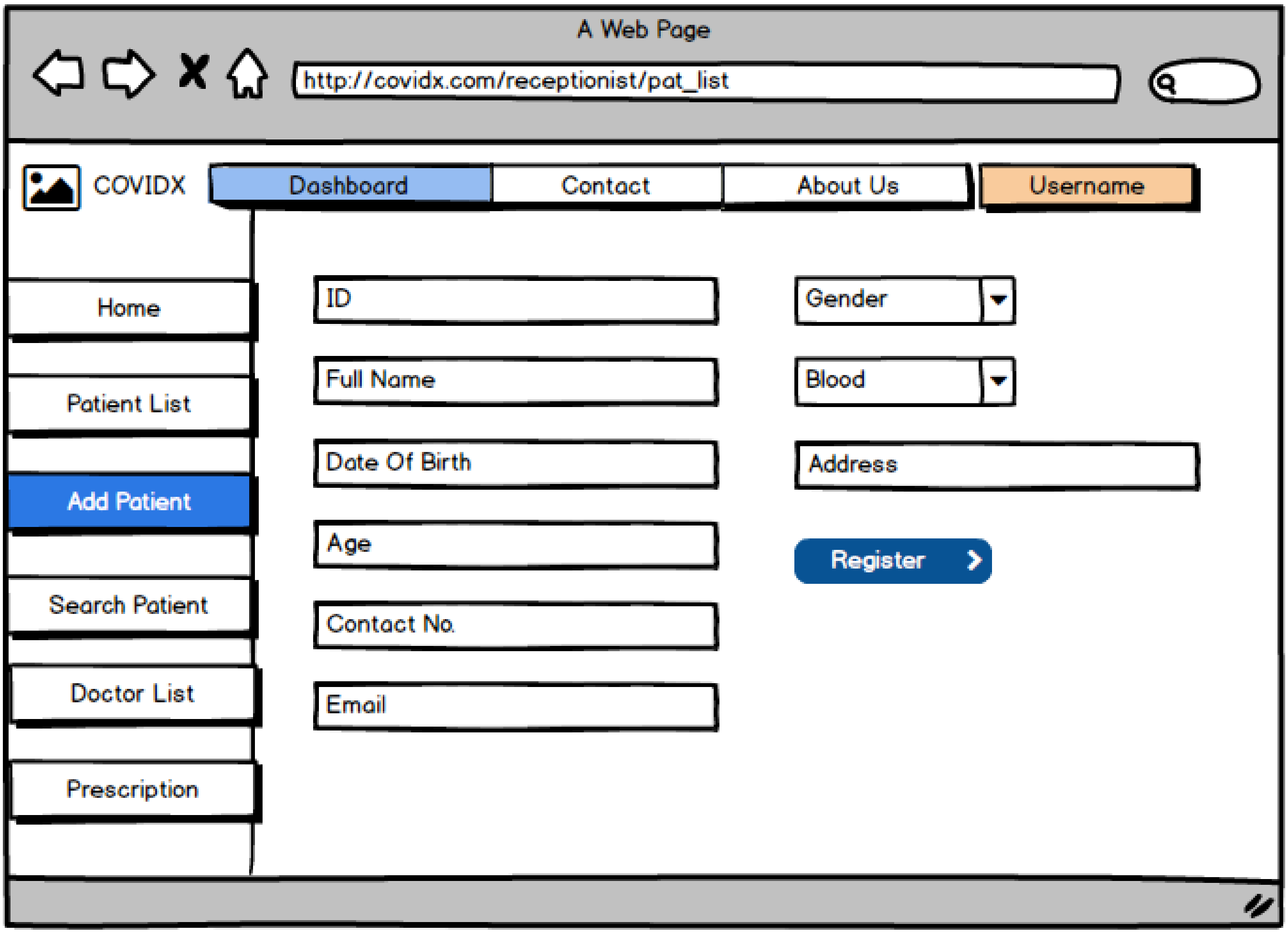


Figure 55 Add Patient page (Receptionist)

Receptionist/view_doctor:

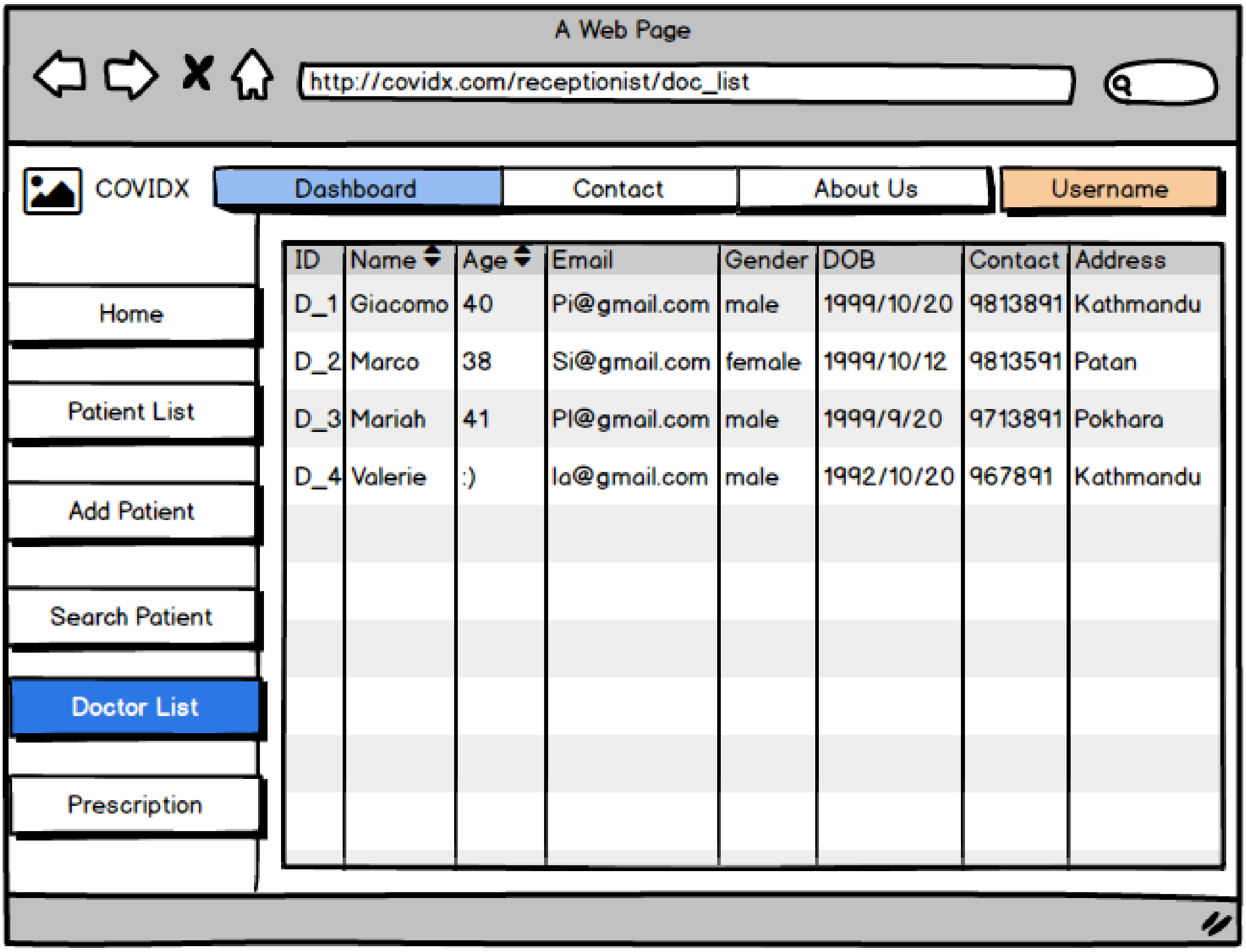


Figure 56 View Doctor page (Receptionist)

Receptionist/prescription:

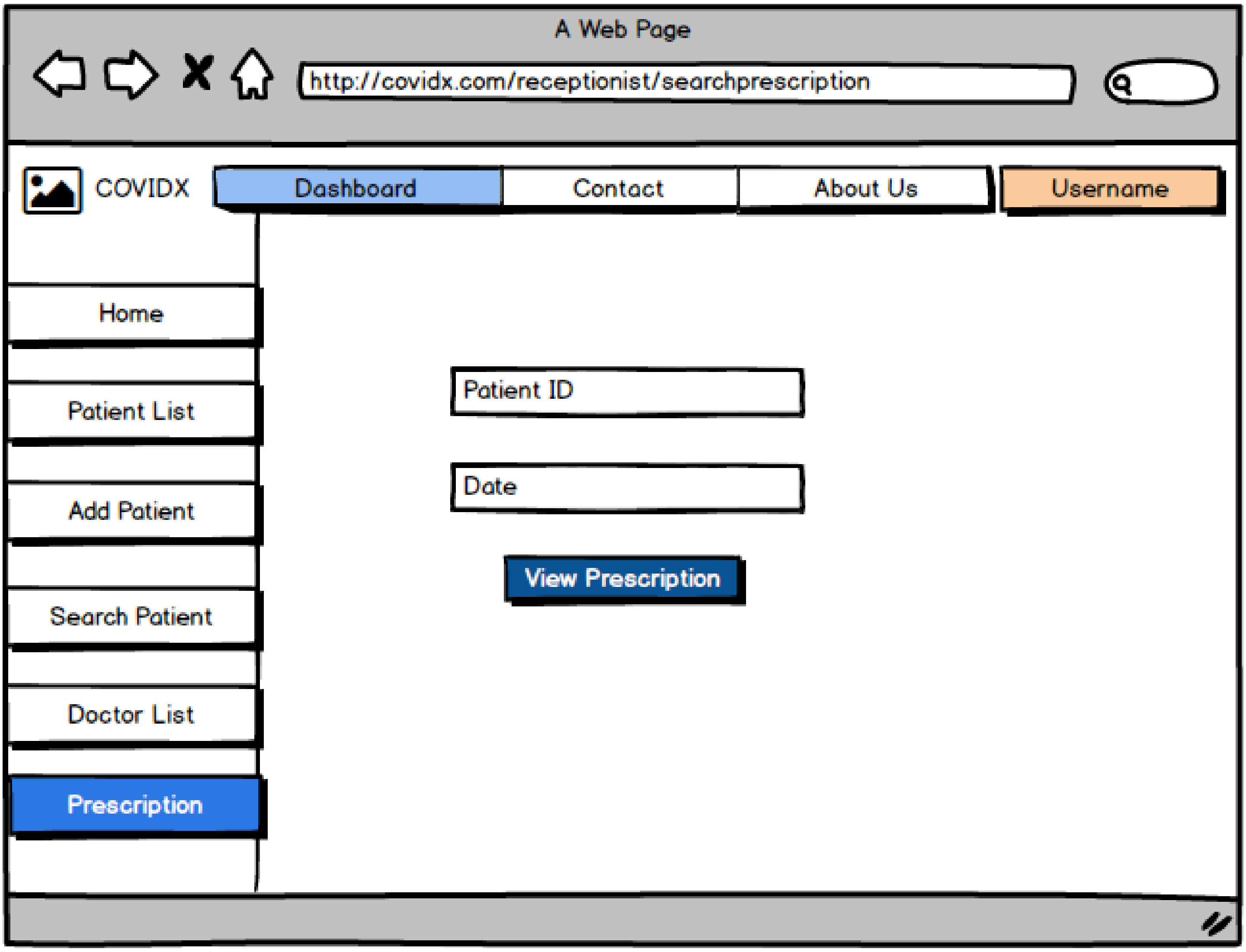


Figure 57 Prescription page (receptionist)

7.5.5. Admin

Admin/dashboard:

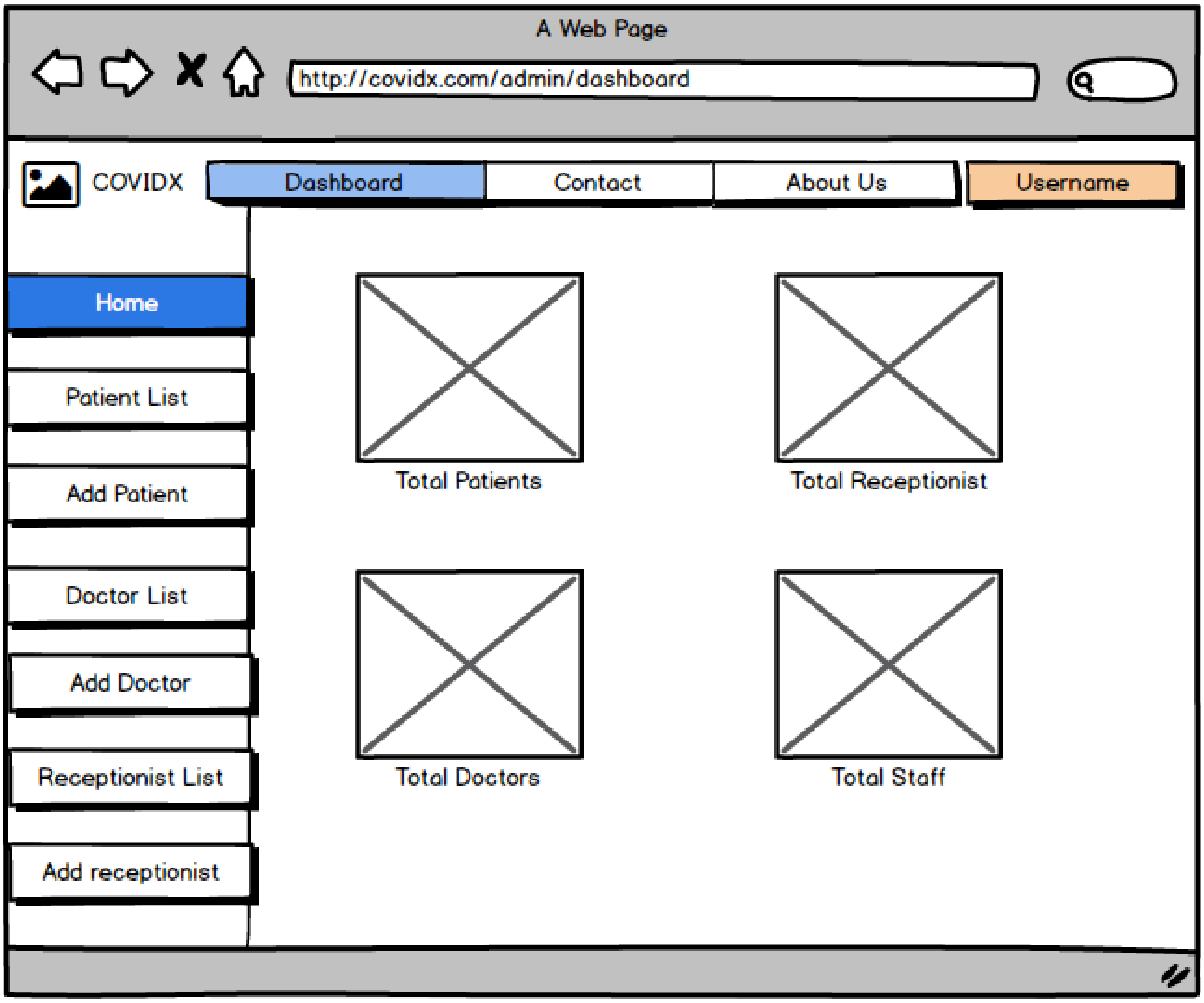


Figure 58 Admin Dashboard

Admin/add_patient:

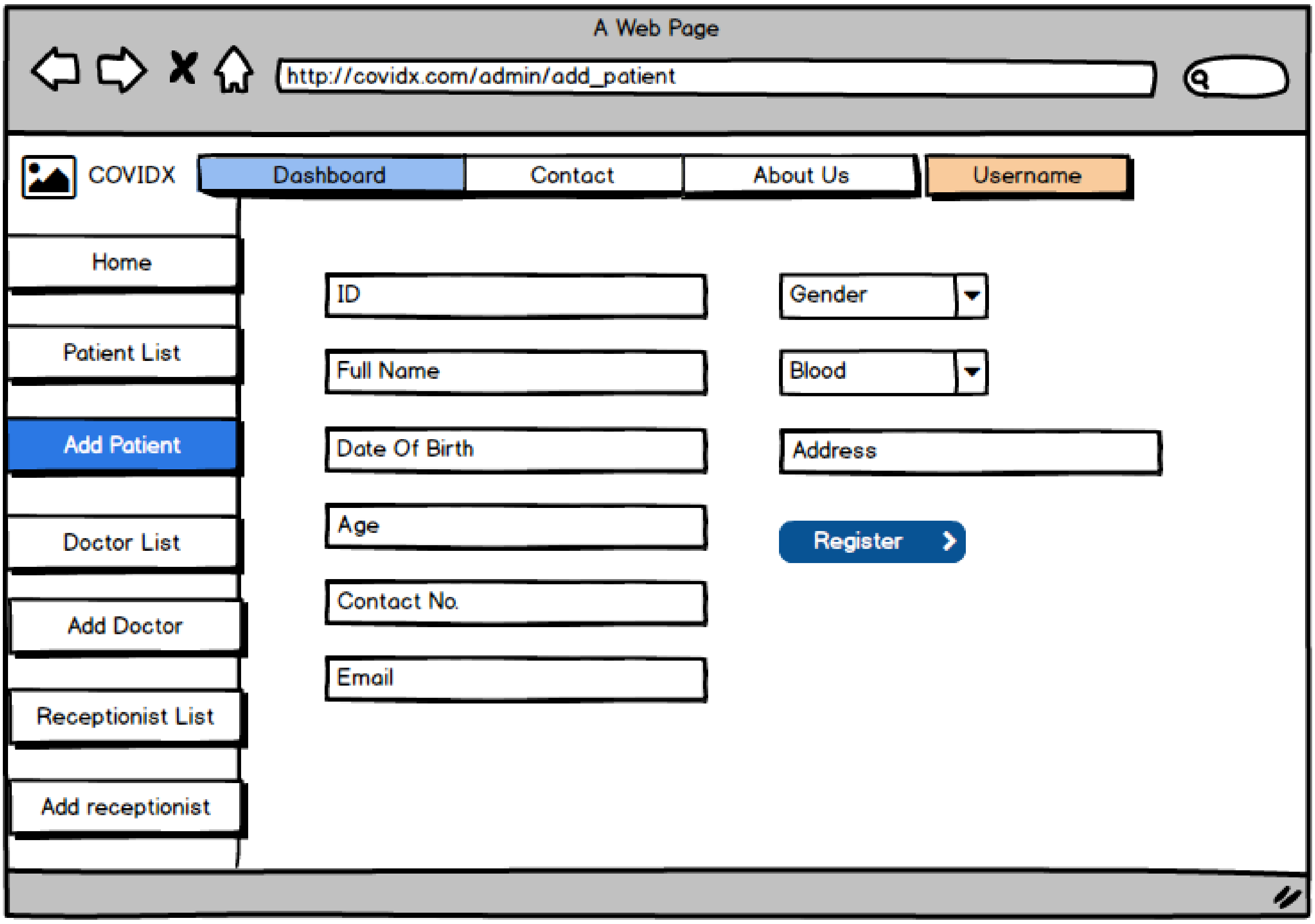


Figure 59 Add Patient page (Admin)

Admin/view_doctor

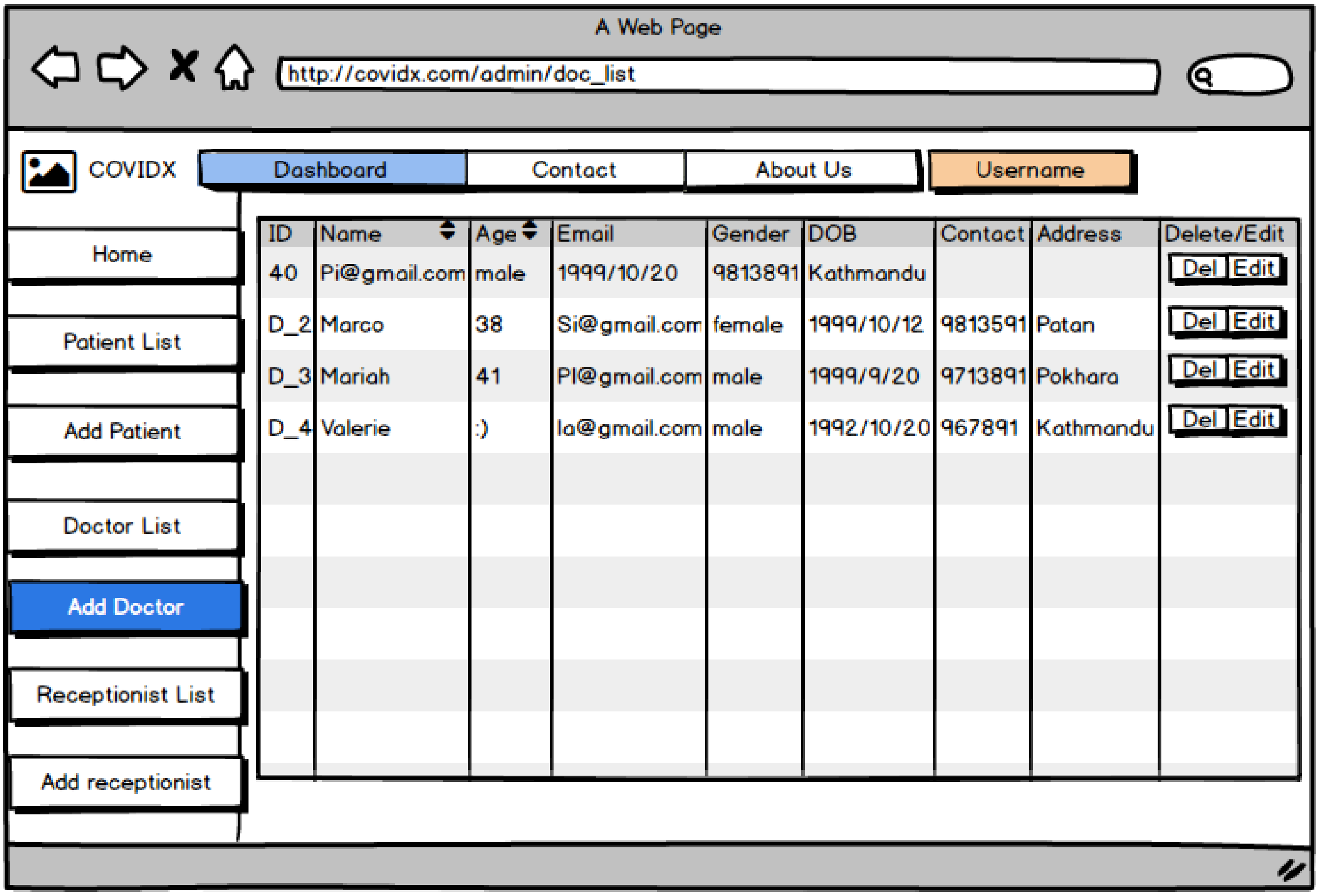


Figure 60 View Doctor page (Admin)

Admin/add_doctor

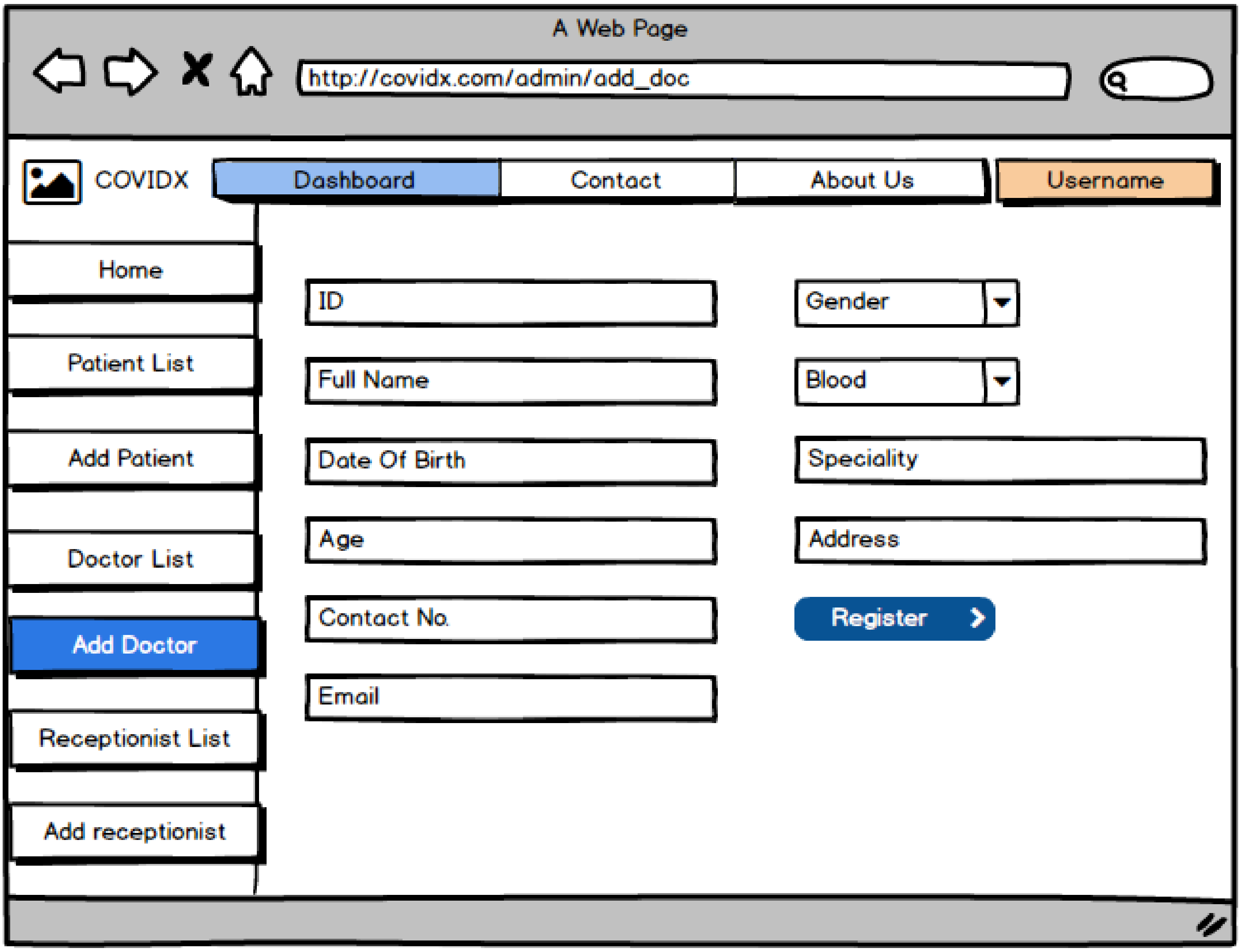


Figure 61 Add Doctor Page (Admin)

Admin/view_receptionist

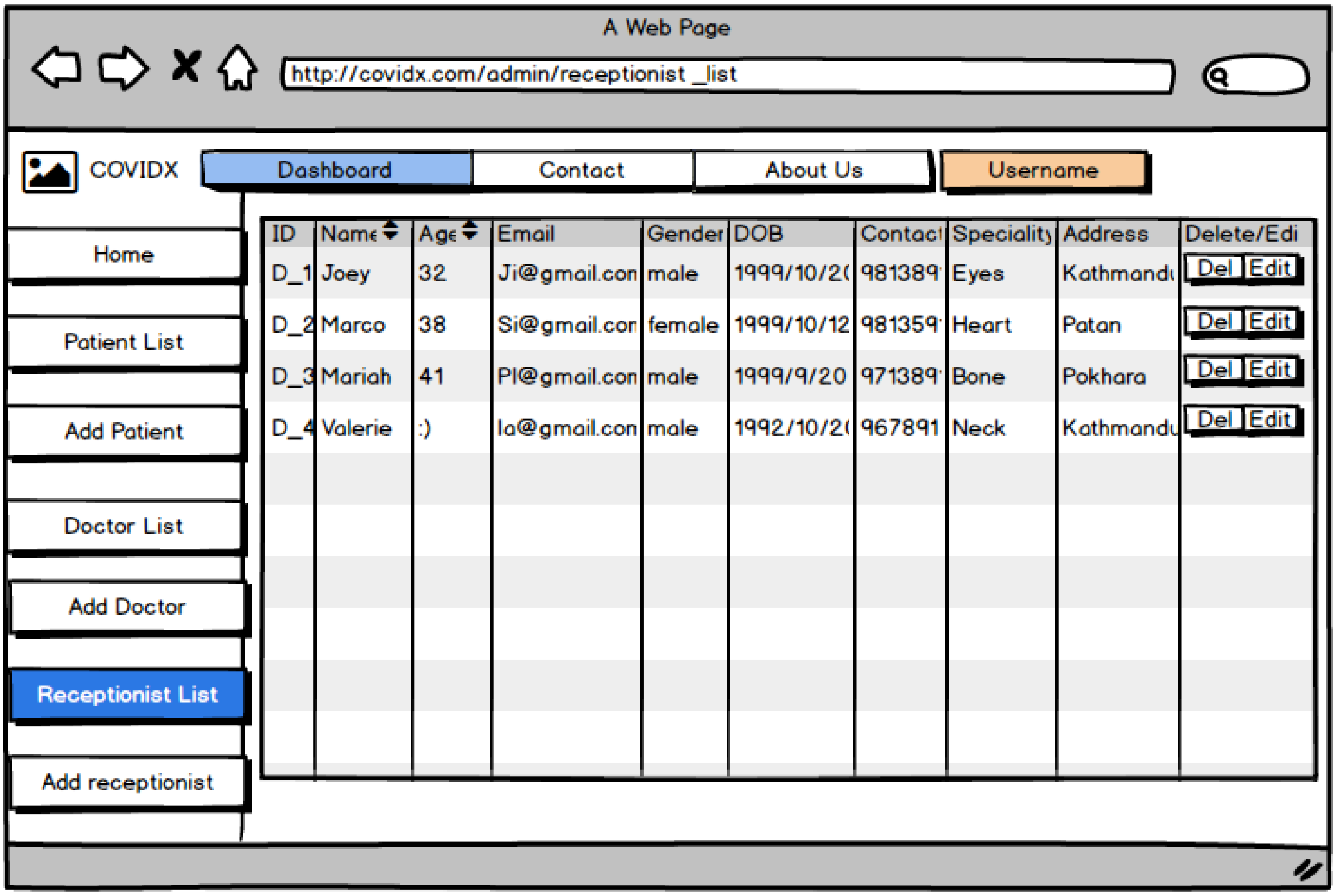


Figure 62 Add Receptionist page (Admin)

Admin/add_receptionist

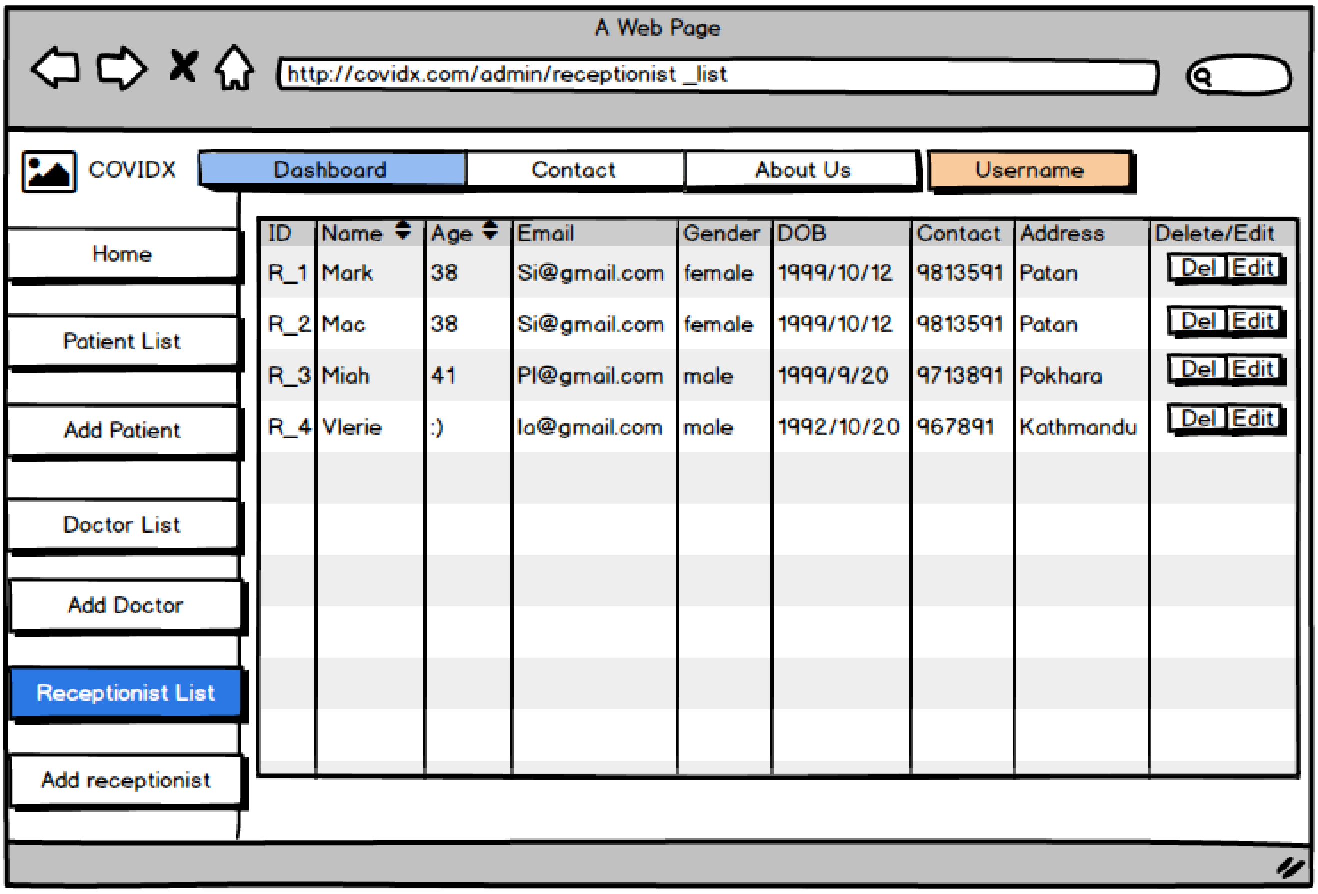


Figure 63 View Receptionist Page (Admin)

8. Black box Testing (Equivalence Partitioning Testing)

D = Doctor, R = Receptionist and A = Admin

TC_ID	Name	Login As	Steps	Data Set	Expected	Result
TC_01	Add Patients	D, R, A	Go to Add patients on nav bar > add patients	Name = "Steve", Email = happy@gmail.com etc.	Patient added to the database	Pass
TC_02	Update Patient	D, R, A	Go to View Patient > Click update button >update patient details	Updated patient name ="Happy"	Patient updated	Updated
TC_03	Delete patient	D, R, A	Go to View patient > Click delete button	Patient _name="Happy"	Patient deleted	Deleted
TC_04	Register Doctor	R, A	Go to Add Doctor on nav bar > add doctor	Name = "Stephen", Email = mike@gmail.com etc	Doctor Added	Doctor added
TC_05	Update Doctor	R, A	Go to View Doctor > Click update button >update patient details	Updated doctor name ="Mike"	Doctor name updated	Updated
TC_06	Delete Doctor details	R, A	Go to View Doctor > Click delete button	Doctor _name="Mike"	Doctor deleted	Deleted
TC_07	Register Receptionist	A	Go to Add Receptionist > add Receptionist	Name = "Pradi", Email = sudip@gmail.com etc	Doctor Added	Doctor added

TC_O 8	Update Receptionist	A	Go to View Doctor > Click update button >update patient details	Updated Receptionist name ="Sudip"	Reception name updated	Updated
TC_09	Delete Receptionist details	A	Go to View Receptionist > Click delete button	Reception _name="Sudip"	Reception deleted	Deleted
TC_10	Search patient	D, R, A	View Patient > Search	Search Value="Happy"	Detail of patient "Happy"	Pass
TC_11	Search Doctor	R, A	View Doctor > Search	Search Value="Mike"	Detail of patient "Mike"	Pass
TC_12	Search Receptionist	A	View Reception > Search	Search Value="Sudip"	Detail of patient "Sudip"	Pass
TC_13	Login Doctor (with correct credentials but "incorrect login as")	-	Go to login page > Enter login details (Choose login as Receptionist from dropdown)	Login details from database	Fail because login as is incorrect	Failed to login
TC_13	Login Doctor (with correct details")	-	Go to login page > Enter login details (Choose login as Doctor from dropdown)	Login details from database	Redirect to doctor dashboard	Redirect ed to doctor dashboar d
TC_14	Login Receptionist (with correct credentials	-	Go to login page > Enter login details (Choose login as	Login details from database	Fail because "login as" is incorrect	Failed to login

	but “incorrect login as”)		Doctor from dropdown)			
TC_15	Login Receptionist (with correct details”)	-	Go to login page > Enter login details (Choose login as Receptionist from dropdown)	Login details from database	Redirect to Receptionist dashboard	Success
TC_16	Patient Trash	D, R, A	Go to View Patient> Click delete button	Patient _name="Happy "	Add to too trash/restore table in database	Added
TC_17	Patient Restore	D, R, A	Go to Patient Restore > Click Restore Button	Patient name= "Happy"	Restore Patient details	Restore d
TC_18	Covid detect	D	Go to Covid detection in nav bar > Upload image	Image = “covid100.png”	Detect positive	success

9. Conclusion

COVID-X is a project with two main components: detecting COVID-19 using chest X-Ray (CXR) images and providing a digital interface for doctors to manage prescriptions. The web app assists medical professionals with managing patients, physicians, and prescriptions in a digital format. It also has a feature to detect covid-19.

The lethal virus COVID-19, which was detected in late 2019 was announced as a global pandemic. To diagnose Covid-19 lung inflammation from CXR, we'll need a professional radiologist. We developed a CNN-based computer-aided diagnostic method that does not require radiology expertise to diagnose. New virus makes it harder to find large number of CXR images of COVID positive patients. We found considerable amount of covid related dataset required for covid classification from Kaggle.

We build a custom architecture based on ResNet-50, VGG-16 and LeNet. All the three models were trained for 10 epochs with parameters: ReduceLRonPlateu, Adam optimizer, loss as binary_crossentropy and sigmoid as an activation function at Fully connected layer. In comparing, three different architecture we found that ResNet with a technique called “skip-connection” works best with accuracy of 98%. VGG comes second with 96% and last LeNet with only 82%. ResNet model was saved and is implemented into our webapp using Keras and python Django.

In case of web app, all the frontends and backends for web app were developed using bootstrap and Django. Ultimately, Users like Doctor, Reception and admin can add patient, view patient, delete patient, restore patient, add patient’s prescription, view prescription history and diagnose covid within a app.

Future Escalation:

This project can be further expanded in near future adding more features to it. Currently, there are only 3 users for this web app Doctor, receptionist and Admin. Upon adding appointment management system, Patient could fix an appointment with doctor digitally. I was also thinking to add feature like printing prescription, but due to time constraints I was not able to add it. In case of our classification model, we can use various strategy like ensemble technique, k-fold cross validation techniques etc. to make the model more precise in future.

Bibliography

Alabwi, S., 2017. *Understanding of a Convolutional Neural Network.* Antalya, Turkey, Research Gate.

Ali Narin, C. K. a. Z. P., 2020. Automatic Detection of Coronavirus Disease (COVID-19) Using X-ray Images and Deep Convolutional Neural Networks. pp. 1-17.

Antonino Ingargiola and contributors., 2015. *Jupyter notebook begineer guide.* [Online]
Available at: https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#jupyter-notebook-app
[Accessed 20 September 2020].

Aston, B., 2019. *dpm.* [Online]
Available at: https://thedigitalprojectmanager.com/project-management-methodologies-made-simple/#xp
[Accessed 19 September 2020].

Bridges, J., 2019. *Project Manager.* [Online]
Available at: https://www.projectmanager.com/training/overcoming-lack-project-resources
[Accessed 19 September 2020].

Brown, K., 2019. *How to Geek.* [Online]
Available at: https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/
[Accessed 20 September 2020].

Cavaioni, M., 2018. *Machine Learning Bites.* [Online]
Available at: https://medium.com/machine-learning-bites/deeplearning-series-convolutional-neural-networks-a9c2f2ee1524
[Accessed 21 September 2020].

Diederik P. Kingma, J. B., 2015. Adam: A Method for Stochastic Optimization. *arxiv,* 9(2021).

Dimpy Varshni, K. T., 2020. Pneumonia Detection Using CNN based Feature Extraction. *IEE.*

Donges, N., 2020. *builtin.* [Online]
Available at: https://builtin.com/data-science/transfer-learning
[Accessed 17 September 2020].

Hafeez, M. F. a. A., 2020. COVID-ResNet: A Deep Learning Framework for Screening of COVID19 from Radiographs. *arXiv:2003.14395,* p. 5.

Healthline Media UK Ltd, 2020. *Meical news today.* [Online]
Available at: https://www.medicalnewstoday.com/articles/pneumonia-and-covid-19
[Accessed 16 September 2020].

javatpoint, 2018. *javaTpoint.* [Online]
Available at: https://www.javatpoint.com/django-mvt
[Accessed 17 September 2020].

Jeremy Howard, S. R., 2018. Universal Language Model Fine-tuning for Text Classification. *arXiv:1801.06146v5,* Issue 23, p. 12.

Kaiming He, X. Z. S. R. J. S., Dec, 2015. Deep Residual Learning for Image Recognition. *arXiv:1512.03385,* Issue Oct, 2020, p. 12.
Linda Wang, Z. Q. L. a. A. W., 2020. COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images. *arXiv:2003.09871v4,* Issue 13, p. 12.
MFMER, 2020. *Mayo Clinic.* [Online]
Available at: https://www.mayoclinic.org/diseases-conditions/coronavirus/symptoms-causes/syc-20479963
[Accessed 15 September 2020].
Mr. Amit Kumar Das, S. G. S. T. R. D. S. A. A. A. C., 2020. Automatic COVID-19 Detection from X-Ray images using Ensemble Learning with CNN. p. 9.
Mujtaba, H., 2020. *great learning.* [Online]
Available at: https://www.mygreatlearning.com/blog/cross-validation/
[Accessed 13 October 2020].
Parashar, A., 2020. Vgg 16 Architecture, Implementation and Practical Use. *Medium.*
Prabhu, 2018. *Medium.* [Online]
Available at: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148
[Accessed 21 September 2020].
Roston, B. A., 2020. *Slash Gear.* [Online]
Available at: https://www.slashgear.com/coronavirus-tests-understanding-pcr-antigen-and-antibody-02615374/
[Accessed 15 Septemeer 2020].
Shaikh, R., 2018. *Medium.* [Online]
Available at: https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85
[Accessed 14 October 2020].
Simon Kornblith, H. L. T. C. M. N., 2020. What's in a Loss Function for Image Classification?. *arxiv,* 1(2021).
technopedia , 2018. *technopedia.* [Online]
Available at: https://www.techopedia.com/definition/32731/convolutional-neural-network-cnn
[Accessed 17 09 2020].
TitoOrt, 2018. *Stack Exchange.* [Online]
Available at: https://datascience.stackexchange.com/questions/33022/how-to-interpert-resnet50-layer-types
[Accessed 21 September 2020].
tsang, S.-H., 2018. *Medium.* [Online]
Available at: https://towardsdatascience.com/review-resnet-winner-of-ilsvrc-2015-image-classification-localization-detection-e39402bfa5d8
[Accessed 21 September 2020].
Tutorail and Example, 2020. *Tutorail and Example.* [Online]
Available at: tutorialandexample.com/django-mvt/
[Accessed 21 September 2020].

University, L. S., 2020. *Medical press.* [Online] Available at: https://medicalxpress.com/news/2020-09-radiologists-chest-x-rays-highly-covid-.html
[Accessed 15 September 2020].
Windsor, G., 2019. *BrightWork.* [Online] Available at: https://www.brightwork.com/blog/tackle-poor-project-communication
[Accessed 19 September 2020].