Report For End Term Evaluation of Minor - I

HealthKit

Group Members :			**Supervisor :**
Shubh Gaur	-	17103036	Dr. Megha Rathi
Pranav Bahri	-	17103049	
Mayank Gupta	-	17803002	

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

TABLE OF CONTENTS

Chapter No.	Topics	Page No.
Chapter-1	Introduction	6
	1.1 General Introduction	
	1.2 Problem Statement	
	1.3 Brief description of the solution approach	
Chapter-2	Literature Survey	7-12
	2.1 Summary of papers studied	
	2.2 Integrated summary of the literature studied	
Chapter 3:	Requirement Analysis and Solution Approach	13-14
	3.1 Overall description of the project	
	3.2 Requirement Analysis	
	3.3 Solution Approach	
Chapter-4	Modeling and Implementation Details	15-22
	4.1 Design Diagram	
	4.2 Implementation details	
Chapter-5	Limitations	22
Chapter-6	Findings, Conclusion and Future Work	22-25
	6.1 Findings	
	6.2 Conclusion	
	6.3 Future Work	
Chapter-7	References	26-27

CERTIFICATE

This is to certify that the work titled “**HealthKit**” submitted by “**Shubh Gaur, Pranav Bahri, Mayank Gupta**” in partial fulfillment for the award of the degree of B. Tech of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor	……………………..
Name of Supervisor	Dr. Megha Rathi
Designation	Assistant Professor (Senior Grade)
Date	28 November 2019

ACKNOWLEDGEMENT

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma from a university or other institute of higher learning, except where due acknowledgment has been made in the text.

We express our sincere thanks to Dr. Megha Rathi(Assistant Professor(Senior Grade), Dept. of Computer Science), our project in-charge, who guided us through the project and also gave valuable suggestions and guidance for completing the project. She helped us to understand the intricate issues involved in project-making besides effectively presenting it. Our project has been a success only because of her guidance.

We are also thankful to the whole Computer Science Department for providing us the technical support to carry out the project work, to let us utilize all the necessary facilities of the institute and guidance at every step during the project work.

Signature of the Student(s)

Name of Students : Shubh Gaur, Pranav Bahri, Mayank Gupta

Enrollment Numbers : 17103036, 17103049, 17803002

Date : 28 November 2019

List of Figures

Figure Serial No.	Figure Description	Page No
Fig 4.1.1	Flow Chart: Disease Predictor - Workflow	15
Fig 4.1.2	Flow Chart : Reminder - Workflow	16
Fig 4.1.3	Flow Chart : Barcode Scanner - Workflow	17
Fig 4.1.4	Activity Diagram	18
Fig 4.2.1	Training the Model	19
Fig 4.2.2	Describing the Model	20
Fig 4.2.3	Testing the Model	20
Fig 4.2.4	Model Prediction	21
Fig 4.2.5	Custom-built API using Flask	21
Fig 6.2.1	MainActivity	22
Fig 6.2.2	Disease Predictor	22
Fig 6.2.3	Barcode Scanner	23
Fig 6.2.4	Medicine Reminder	23

1. Introduction

1.1 General Introduction

We are developing a Health-oriented application that aids the user in maintaining and tracking his/her health conditions properly. The aim is to build a one-stop solution to all health-related needs of a person. It automates the process of manually maintaining records and provides basic diagnostic features.

1.2 Problem Statement

With our fast-moving lives, we often take our health for granted. We start to overlook the symptoms we experience just because of the stigma of going to a doctor. People are unable to keep track of their medication, often forget to have their medications at the right time. But with the advancing era of Artificial Intelligence and automation, this task has become easier by the day. With the increasing dependence of humans on our mobile devices, our goal is to make a mobile application to automate all the things mentioned and improve upon that.

1.3 Brief description of the solution approach

The health application serves several purposes. To begin with, the app has a Disease Predictor, which takes symptoms as the user input and predicts the disease based on the same.

The application also implements a Barcode Scanner that scans the barcodes on different medicines via the camera and displays the details about the same like usage, effectiveness and side effects.

The app also lets the user input their daily medication including the number of doses and their timings. On adding, the app would send the user a notification at every dosage time.

2.Literature Survey

2.1 Summary of papers studied

Resource 1

Title of paper	Artificial Intelligence and global health:how can AI contribute to health in resource-poor settings?
Authors	Brian Wahl, Aline Cossy-Gantner, Stefan Germann, Nina R Shwalbe
Year Of Publication	2018
Publishing Details	BMJ Journals BMJ Global Health Volume 3, Issue 4
Abstract	The field of artificial intelligence (AI) has evolved considerably in the last 60 years. While there are now many AI applications that have been deployed in high-income country contexts, use in resource-poor settings remains relatively nascent. With a few notable exceptions, there are limited examples of AI being used in such settings. However, there are signs that this is changing. Several high-profile meetings have been convened in recent years to discuss the development and deployment of AI applications to reduce poverty and deliver a broad range of critical public services. We provide a general overview of AI and how it can be used to improve health outcomes in resource-poor settings. We also describe some of the current ethical debates around patient safety and privacy. Despite current challenges, AI holds tremendous promise for transforming the provision of healthcare services in

	resource-poor settings. Many health system hurdles in such settings could be overcome with the use of AI and other complementary emerging technologies. Further research and investments in the development of AI tools tailored to resource-poor settings will accelerate realising the full potential of AI for improving global health.

Resource 2

Title of paper	Deep Learning For Predictive Analytics in Healthcare
Authors	Anandhavalli Muniasamy, Sehrish Tabassam, Mohammad A. Hussain, Habeeba Sultana, Vasanthi Muniasamy, Roheet Bhatnagar
Year Of Publication	2019
Publishing Details	International Conference on Advanced Machine Learning Technologies and Applications Vol 921 pg. 32-42
Abstract	Despite a recent wealth of data and information, the healthcare sector is lacking in actionable knowledge. The healthcare industry faces challenges in essential areas like electronic record management, data integration, and computer-aided diagnoses and disease predictions. It is necessary to reduce healthcare costs and the movement towards personalized healthcare. The rapidly expanding fields of deep learning and predictive analytics has started to play a pivotal role in the evolution of large volume of healthcare data practices and research. Deep learning offers

	a wide range of tools, techniques, and frameworks to address these challenges. Health data predictive analytics is emerging as a transformative tool that can enable more proactive and preventative treatment options. In a nutshell, this paper focuses on the framework for deep learning data analysis to clinical decision making depicts the study on various deep learning techniques and tools in practice as well as the applications of deep learning in healthcare.

Resource 3

Title of paper	Predicting Heart Attack Through Explainable Artificial Intelligence
Authors	Mehrdad Aghamohammadi, Manvi Madan, Jung Ki Hong, Ian Watson
Year Of Publication	2019
Publishing Details	International Conference On Computational Science Vol 11537 pg.633-645
Abstract	A novel classification technique based on combined Genetic Algorithm (GA) and Adaptive Neural Fuzzy Inference System (ANFIS) for diagnosis of heart Attack is reported. Exploiting the combined advantages of neural networks, fuzzy logic and GA, the performance of the proposed system is investigated by evaluation functions such as sensitivity, specificity, precision, accuracy and Root Mean Squared Error (RMSE). Also, the efficiency of the algorithm is evaluated by employing 9-fold cross validation. To address the explainability of the predictions, explainable graphs are provided. The results show that the

	performance of the proposed algorithm is quite satisfactory. Furthermore, the importance of various symptoms in diagnosis of heart attack is investigated through defining and employing an importance evaluation function. It is shown that some symptoms have key roles in effective prediction of heart Attack.

Resource 4

Title of paper	A Study On Machine Learning Algorithm in Medical Diagnosis
Authors	D.Pavithra , Dr. A.N. Jayanthi
Year Of Publication	2018
Publishing Details	International Journal of Advanced Research in Computer Science Vol 9, No 4
Abstract	Machine learning is a method of optimizing the performance criterion using the past experience. It builds the mathematical model by using the theory of statistics, as the main task is to infer from the samples provided. The algorithm uses computational methods to get the information directly from the data. They are mainly used in medical diagnosis for making critical decisions, as the data in the medical field is huge and the accuracy of the diagnosis depends on considering the huge data of the patients. ML improves the accuracy of the diagnostic of the disease. It also provides automatic learning techniques for

	predicting the common patterns from the realistic data. There are different ML algorithms, the appropriate method has to be chosen based on their performance. This paper focuses on the use of different machine learning algorithms like Support Vector Machine, Naïve Bayesian, J48, Random Forest etc. for accurate medical diagnosis.

Resource 5

Title of paper	Artificial Intelligence based Smart Doctor using Decision Tree Algorithm
Authors	Rida Sara Khan, Asad Ali Zardar, Zeeshan Bhatti
Year Of Publication	2017
Publishing Details	Journal of Information & Communication Technology Vol 11 Issue 2
Abstract	Artificial Intelligence (AI) has already made a huge impact on our current technological trends. Through AI developments, machines are now given power and intelligence to behave and work like human mind. In this research project, we propose and implement an AI based health physician system that would be able to interact with the patient, do the diagnosis and suggest quick remedy or treatment of their problem. A decision tree algorithm is implemented in order to follow a top down searching approach to identify and diagnose the problem and suggest a possible solution. The system uses a questionnaire based approach to query the user (patient) about various Symptoms,

	based on which a decision is made and a medicine is recommended.

2.2 Integrated Summary Of The Literature Studied

Healthcare is a domain in which the prediction and prevention are more important than explanation, considering the cost in diagnosis and treatment. There are many cases where when a person gets information about the disease he/she is affected, it is too late to cure and early detection might have saved his/her life. With the rise in pollution and sedentary lifestyle and also the decreasing amount of nature on the globe many new diseases are affecting humans.

Development of applications which can monitor their health data, recommend different ways of prevention and treatments from diseases and ailments, predict disease based on the symptoms listed by the user is to take care of the user at every second and to help them in emergency situations. Such applications are useful for both the patient and the doctor as it can provide very detailed information about the patient’s health,lifestyle and whether or not the medicines are taken at the correct time.

Many different artificial intelligence algorithms and techniques can be implemented to detect disease based on the user’s input of symptoms. The best way to detect disease is to make a neural network and train it to a sample data. The model will then predict based on the user’s input. This implementation can vary based on the context they are used. For example, to detect cancer input variables will be different than that of those to detect asthma.

Many patients with Alzheimer's forgets to take the medicine at the time prescribed. This affects the treatment and of course, their health. An application which reminds the user to take a particular at a particular time can be a very good way to help them.

3. Requirement Analysis and Solution Approach

3.1 Requirement Analysis

Python - It is an interpreted, high-level, general-purpose programming language. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large scale projects.

Tensorflow - It is an open-source artificial intelligence library for python. It is used to create major scale machine learning and deep learning models. It allows developers to create large-scale neural networks with many layers. It is mainly used for: Classification, Perception, Understanding, Discovering, Prediction and Creation.

Keras - It is a high-level API built on top of TensorFlow to create machine learning and deep learning models.Keras is an API designed for human beings, not machines. It follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear and actionable feedback upon user error.

Pandas - A python module for handling large amounts of data for testing and training purposes. Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

Flask - A python framework for hosting a local server to help the android application communicate with the model. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

Android Studio - Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. A Java/Kotlin IDE for developing android applications with the help of Gradle.

Firebase - Firebase provides a realtime database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored in Firebase's cloud.

Custom-Built API - For connecting the android application(written in Java) with the server(written in Python).

Kaggle - It is a dataset hosting platform. It allows users to find and publish data sets, explore and build models in a web-based data-science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges.

3.2 Solution Approach

For the disease prediction model, we have taken a dataset from Kaggle which includes Symptoms and co-related diseases. The data is then used to train the Deep Learning ANN model made in Tensorflow using Keras API. The model is then hosted on the Custom-Built API that lets android applications interact with models written in Python using Tensorflow. This makes the Application more modular as the changes can be directly made with the server and the same can be reflected in the application without any changes in the app.

The android application also implements a Barcode scanner which is implemented using Firebase’s native ML Kit API. The API lets us use the camera of the mobile device and read the Barcode present on drugs. This then connects to a database and displays the available details of the drug.

4. Modeling and Implementation Details

4.1 Design Diagrams

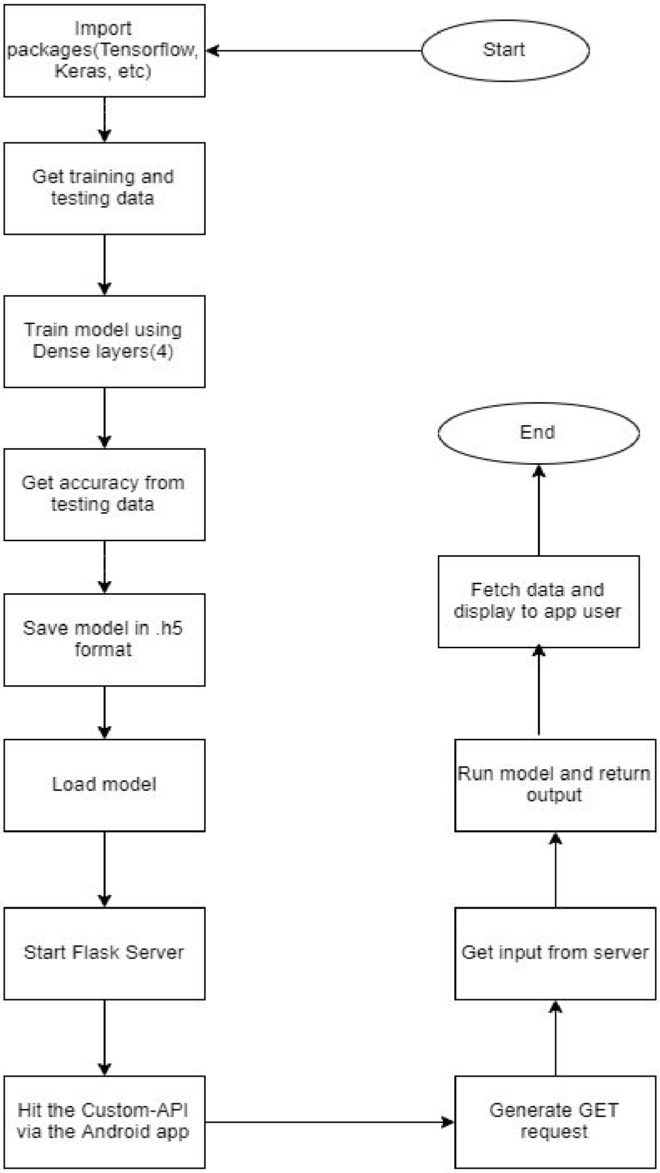

Start
Import packages(Tensorflow, Keras, etc)
Get training and testing data
Train model using Dense layers(4)
Get accuracy from testing data
Save model in .h5 format
Load model
Start Flask Server
Hit the Custom-API via the Android app
Generate GET request
Get input from server
Run model and return output
Fetch data and display to app user
End

Fig 4.1.1

Disease Predictor Workflow

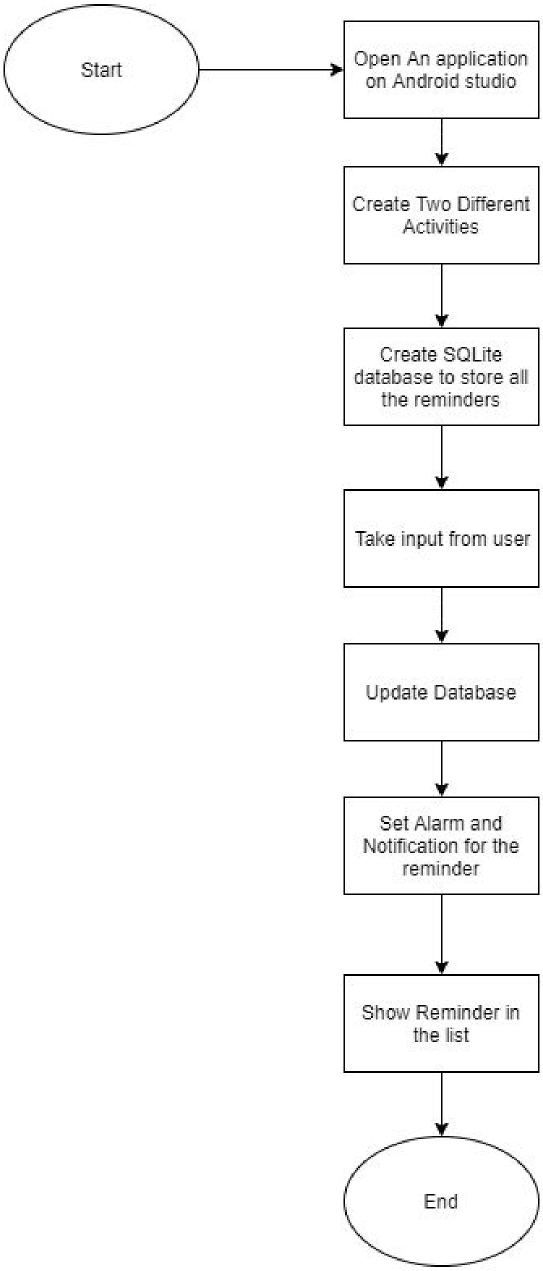

Start
Open An application on Android studio
Create Two Different Activities
Create SQLite database to store all the reminders
Take input from user
Update Database
Set Alarm and Notification for the reminder
Show Reminder in the list
End

Fig 4.1.2

Reminder - Workflow

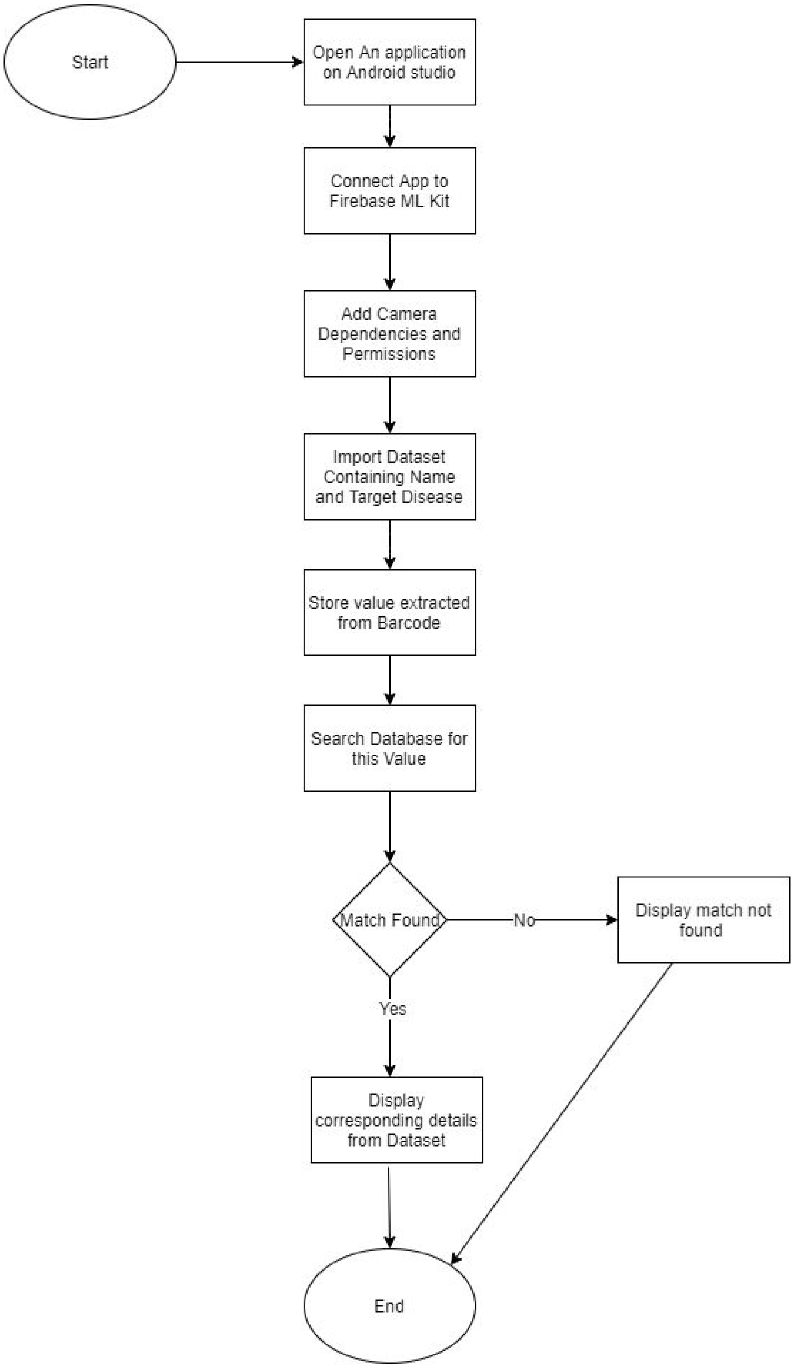

Start
Open An application on Android studio
Connect App to Firebase ML Kit
Add Camera Dependencies and Permissions
Import Dataset Containing Name and Target Disease
Store value extracted from Barcode
Search Database for this Value
Match Found
No
Display match not found
Yes
Display corresponding details from Dataset
End

Fig 4.1.3

Barcode Scanner - Workflow

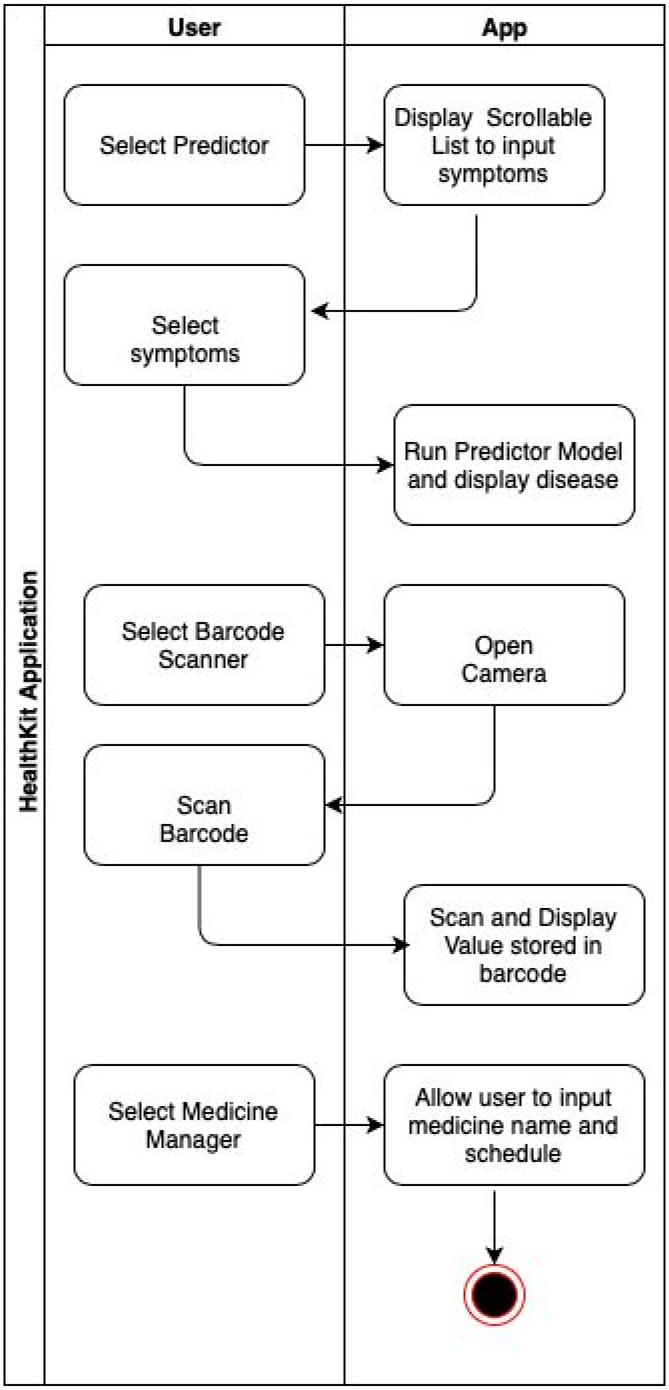
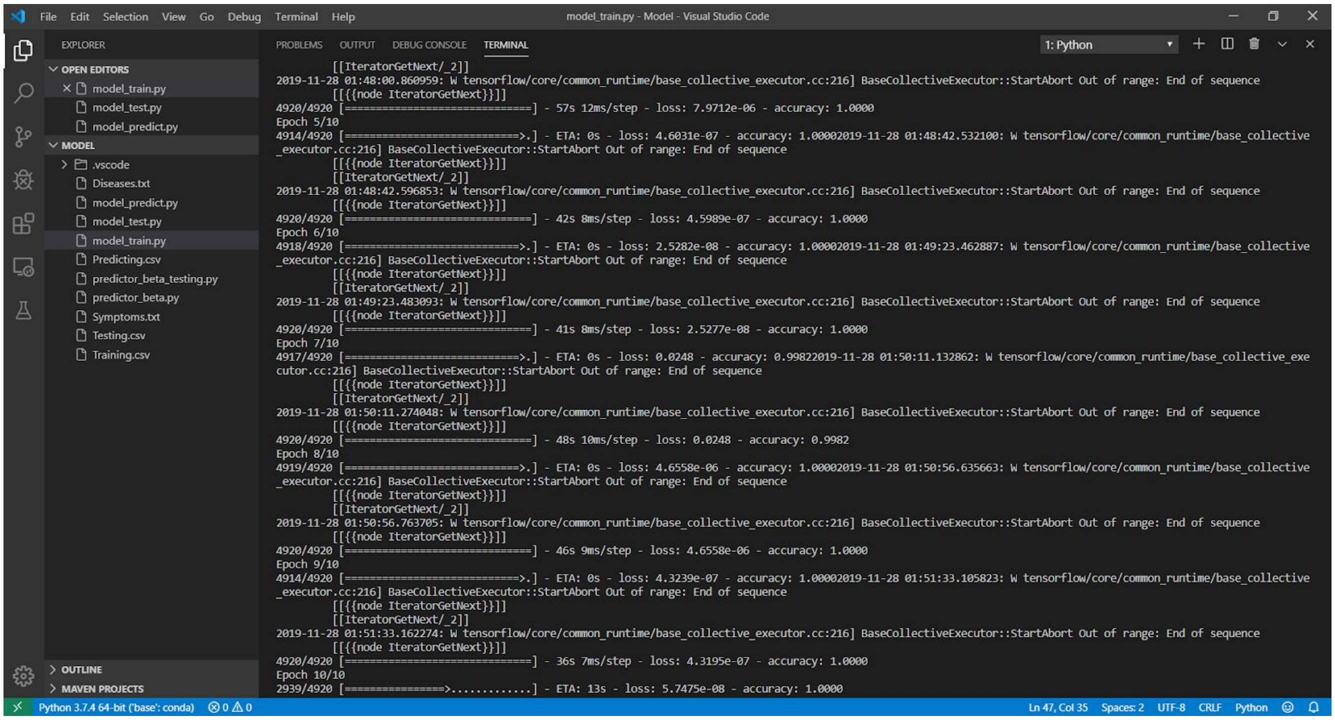

HealthKit Application
User
App
Select Predictor
Display Scrollable List to input symptoms
Select symptoms
Run Predictor Model and display disease
Select Barcode Scanner
Open Camera
Scan Barcode
Scan and Display Value stored in barcode
Select Medicine Manager
Allow user to input medicine name and schedule

Fig 4.1.4

Activity Diagram

4.2 Implementation Details

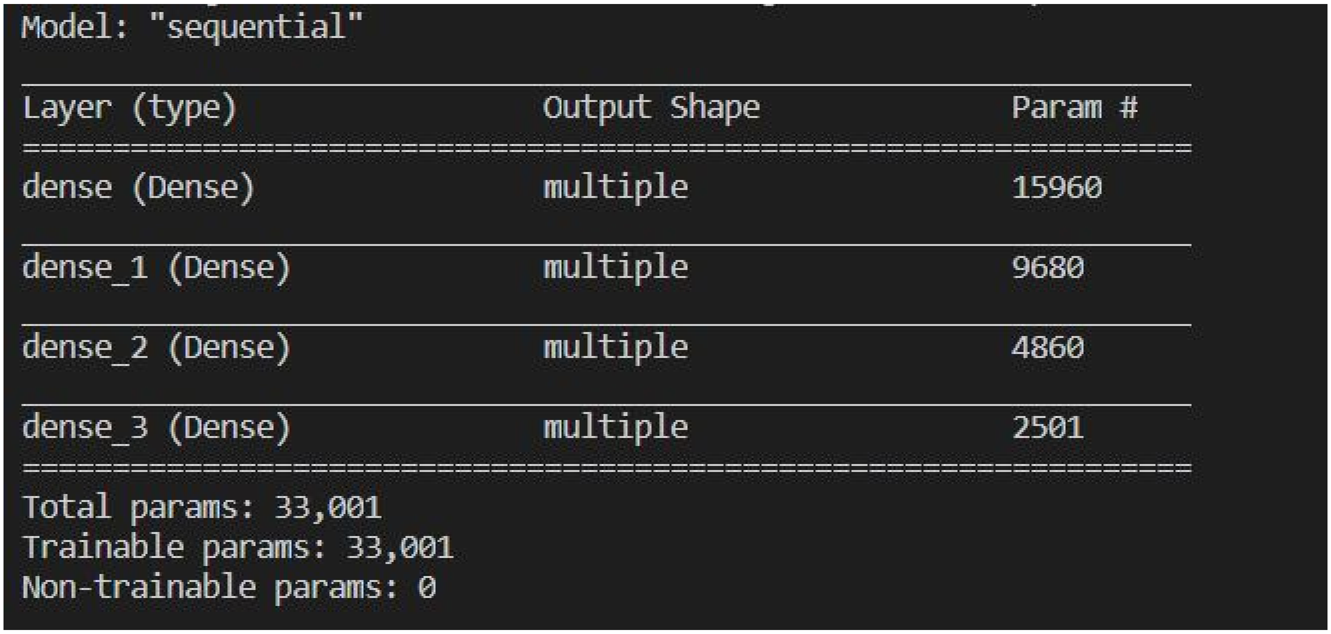
For the Disease Prediction, we have taken a dataset from Kaggle which includes a total of 132 Symptoms and 41 diseases. The total instances of the symptom-disease mapping in the training dataset is 4920. The data is converted in a readable format for the Tensorflow Library using Pandas. The model is then made using Keras. It includes 4 dense layers with a reducing number of perceptrons at each layer. The first three layers have Relu as their activation function and the output layer's activation function is Softmax. The optimizer used is 'adam'.


File Edit Selection View Go Debug Terminal Help
model_train.py - Model - Visual Studio Code
EXPLORER
OPEN EDITORS
model_train.py
model_test.py
model_predict.py
MODEL
.vscode
Diseases.txt
model_predict.py
model_test.py
model_train.py
Predicting.csv
predictor_beta_testing.py
predictor_beta.py
Symptoms.txt
Testing.csv
Training.csv
OUTLINE
MAVEN PROJECTS
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: Python
Python 3.7.4 64-bit ('base': conda) ⊗ 0 ⚠ 0
Ln 47, Col 35 Spaces: 2 UTF-8 CRLF Python

```
         [[IteratorGetNext/_2]]
2019-11-28 01:48:00.860959: W tensorflow/core/common_runtime/base_collective_executor.cc:216] BaseCollectiveExecutor::StartAbort Out of range: End of sequence
         [[{{node IteratorGetNext}}]]
4920/4920 [==============================] - 57s 12ms/step - loss: 7.9712e-06 - accuracy: 1.0000
Epoch 5/10
4914/4920 [============================>.] - ETA: 0s - loss: 4.6031e-07 - accuracy: 1.00002019-11-28 01:48:42.532100: W tensorflow/core/common_runtime/base_collective
_executor.cc:216] BaseCollectiveExecutor::StartAbort Out of range: End of sequence
         [[{{node IteratorGetNext}}]]
         [[IteratorGetNext/_2]]
2019-11-28 01:48:42.596853: W tensorflow/core/common_runtime/base_collective_executor.cc:216] BaseCollectiveExecutor::StartAbort Out of range: End of sequence
         [[{{node IteratorGetNext}}]]
4920/4920 [==============================] - 42s 8ms/step - loss: 4.5989e-07 - accuracy: 1.0000
Epoch 6/10
4918/4920 [============================>.] - ETA: 0s - loss: 2.5282e-08 - accuracy: 1.00002019-11-28 01:49:23.462887: W tensorflow/core/common_runtime/base_collective
_executor.cc:216] BaseCollectiveExecutor::StartAbort Out of range: End of sequence
         [[{{node IteratorGetNext}}]]
         [[IteratorGetNext/_2]]
2019-11-28 01:49:23.483093: W tensorflow/core/common_runtime/base_collective_executor.cc:216] BaseCollectiveExecutor::StartAbort Out of range: End of sequence
         [[{{node IteratorGetNext}}]]
4920/4920 [==============================] - 41s 8ms/step - loss: 2.5277e-08 - accuracy: 1.0000
Epoch 7/10
4917/4920 [============================>.] - ETA: 0s - loss: 0.0248 - accuracy: 0.99822019-11-28 01:50:11.132862: W tensorflow/core/common_runtime/base_collective_exe
cutor.cc:216] BaseCollectiveExecutor::StartAbort Out of range: End of sequence
         [[{{node IteratorGetNext}}]]
         [[IteratorGetNext/_2]]
2019-11-28 01:50:11.274048: W tensorflow/core/common_runtime/base_collective_executor.cc:216] BaseCollectiveExecutor::StartAbort Out of range: End of sequence
         [[{{node IteratorGetNext}}]]
4920/4920 [==============================] - 48s 10ms/step - loss: 0.0248 - accuracy: 0.9982
Epoch 8/10
4919/4920 [============================>.] - ETA: 0s - loss: 4.6558e-06 - accuracy: 1.00002019-11-28 01:50:56.635663: W tensorflow/core/common_runtime/base_collective
_executor.cc:216] BaseCollectiveExecutor::StartAbort Out of range: End of sequence
         [[{{node IteratorGetNext}}]]
         [[IteratorGetNext/_2]]
2019-11-28 01:50:56.763705: W tensorflow/core/common_runtime/base_collective_executor.cc:216] BaseCollectiveExecutor::StartAbort Out of range: End of sequence
         [[{{node IteratorGetNext}}]]
4920/4920 [==============================] - 46s 9ms/step - loss: 4.6558e-06 - accuracy: 1.0000
Epoch 9/10
4914/4920 [============================>.] - ETA: 0s - loss: 4.3239e-07 - accuracy: 1.00002019-11-28 01:51:33.105823: W tensorflow/core/common_runtime/base_collective
_executor.cc:216] BaseCollectiveExecutor::StartAbort Out of range: End of sequence
         [[{{node IteratorGetNext}}]]
         [[IteratorGetNext/_2]]
2019-11-28 01:51:33.162274: W tensorflow/core/common_runtime/base_collective_executor.cc:216] BaseCollectiveExecutor::StartAbort Out of range: End of sequence
         [[{{node IteratorGetNext}}]]
4920/4920 [==============================] - 36s 7ms/step - loss: 4.3195e-07 - accuracy: 1.0000
Epoch 10/10
2939/4920 [================>.............] - ETA: 13s - loss: 5.7475e-08 - accuracy: 1.0000
```

Fig 4.2.1

The model has been trained for a total of 10 epochs. The accuracy score and loss is as given above.

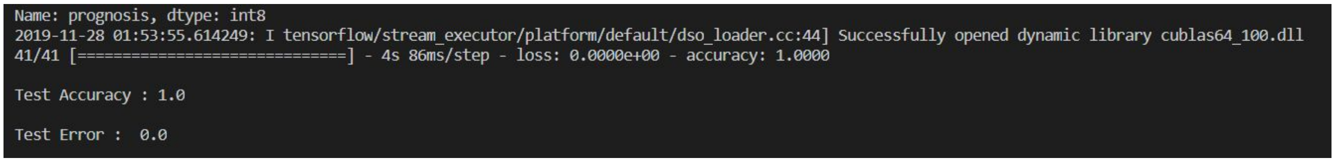


```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                multiple                  15960
_________________________________________________________________
dense_1 (Dense)              multiple                  9680
_________________________________________________________________
dense_2 (Dense)              multiple                  4860
_________________________________________________________________
dense_3 (Dense)              multiple                  2501
=================================================================
Total params: 33,001
Trainable params: 33,001
Non-trainable params: 0
```

Fig 4.2.2

Describing the Model

The model is tested using the testing database which includes a set of 41 different diseases.



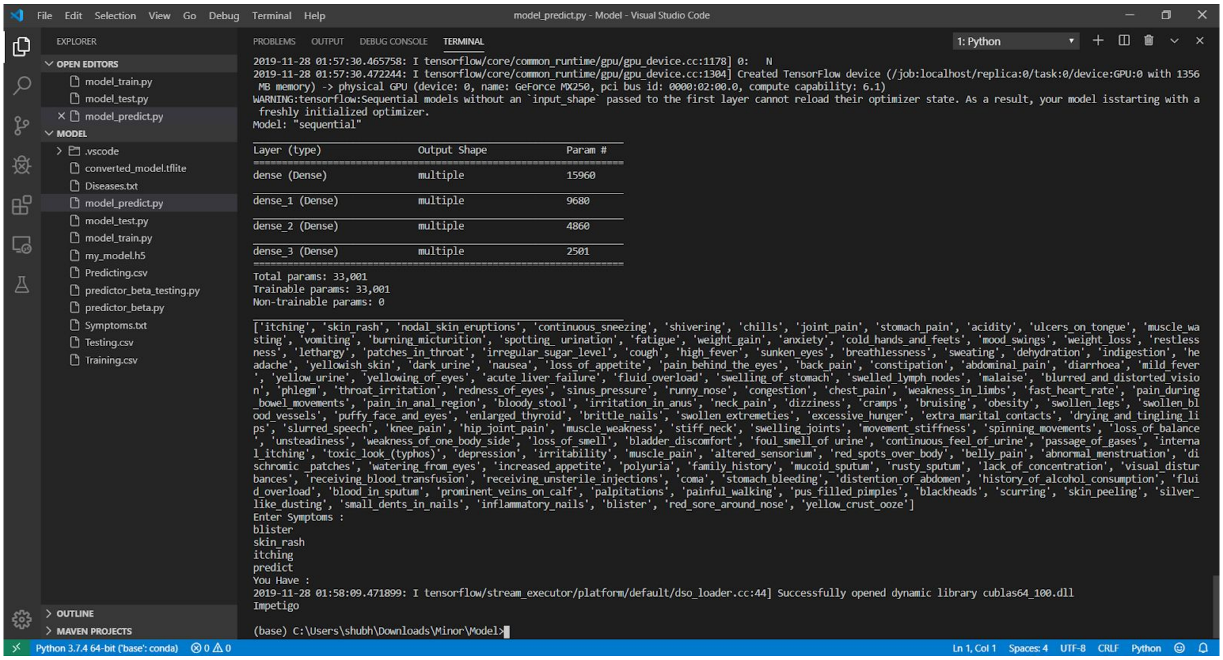
```
Name: prognosis, dtype: int8
2019-11-28 01:53:55.614249: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library cublas64_100.dll
41/41 [==============================] - 4s 86ms/step - loss: 0.0000e+00 - accuracy: 1.0000

Test Accuracy : 1.0

Test Error :  0.0
```

Fig 4.2.3

Testing the Model



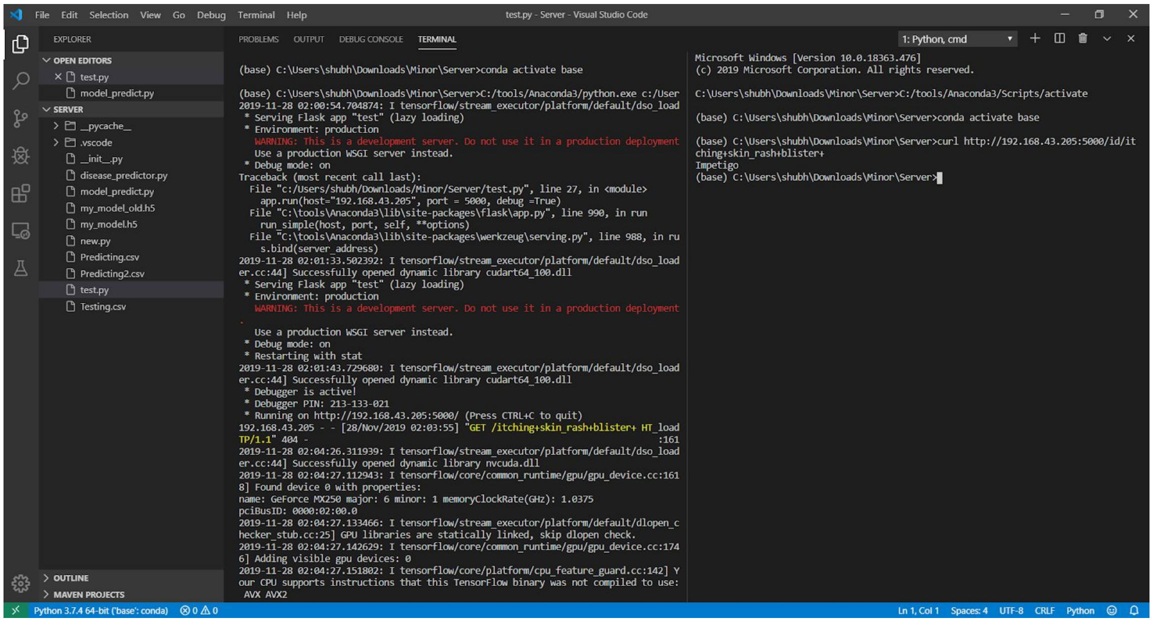
```
2019-11-28 01:57:30.465758: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1178] 0:   N
2019-11-28 01:57:30.472244: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1304] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 1356
 MB memory) -> physical GPU (device: 0, name: GeForce MX250, pci bus id: 0000:02:00.0, compute capability: 6.1)
WARNING:tensorflow:Sequential models without an `input_shape` passed to the first layer cannot reload their optimizer state. As a result, your model isstarting with a
 freshly initialized optimizer.
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                multiple                  15960
_________________________________________________________________
dense_1 (Dense)              multiple                  9680
_________________________________________________________________
dense_2 (Dense)              multiple                  4860
_________________________________________________________________
dense_3 (Dense)              multiple                  2501
=================================================================
Total params: 33,001
Trainable params: 33,001
Non-trainable params: 0
_________________________________________________________________
['itching', 'skin_rash', 'nodal_skin_eruptions', 'continuous_sneezing', 'shivering', 'chills', 'joint_pain', 'stomach_pain', 'acidity', 'ulcers_on_tongue', 'muscle_wa
sting', 'vomiting', 'burning_micturition', 'spotting_ urination', 'fatigue', 'weight_gain', 'anxiety', 'cold_hands_and_feets', 'mood_swings', 'weight_loss', 'restless
ness', 'lethargy', 'patches_in_throat', 'irregular_sugar_level', 'cough', 'high_fever', 'sunken_eyes', 'breathlessness', 'sweating', 'dehydration', 'indigestion', 'he
adache', 'yellowish_skin', 'dark_urine', 'nausea', 'loss_of_appetite', 'pain_behind_the_eyes', 'back_pain', 'constipation', 'abdominal_pain', 'diarrhoea', 'mild_fever
', 'yellow_urine', 'yellowing_of_eyes', 'acute_liver_failure', 'fluid_overload', 'swelling_of_stomach', 'swelled_lymph_nodes', 'malaise', 'blurred_and_distorted_visio
n', 'phlegm', 'throat_irritation', 'redness_of_eyes', 'sinus_pressure', 'runny_nose', 'congestion', 'chest_pain', 'weakness_in_limbs', 'fast_heart_rate', 'pain_during
_bowel_movements', 'pain_in_anal_region', 'bloody_stool', 'irritation_in_anus', 'neck_pain', 'dizziness', 'cramps', 'bruising', 'obesity', 'swollen_legs', 'swollen_bl
ood_vessels', 'puffy_face_and_eyes', 'enlarged_thyroid', 'brittle_nails', 'swollen_extremeties', 'excessive_hunger', 'extra_marital_contacts', 'drying_and_tingling_li
ps', 'slurred_speech', 'knee_pain', 'hip_joint_pain', 'muscle_weakness', 'stiff_neck', 'swelling_joints', 'movement_stiffness', 'spinning_movements', 'loss_of_balance
', 'unsteadiness', 'weakness_of_one_body_side', 'loss_of_smell', 'bladder_discomfort', 'foul_smell_of urine', 'continuous_feel_of_urine', 'passage_of_gases', 'interna
l_itching', 'toxic_look_(typhos)', 'depression', 'irritability', 'muscle_pain', 'altered_sensorium', 'red_spots_over_body', 'belly_pain', 'abnormal_menstruation', 'di
schromic _patches', 'watering_from_eyes', 'increased_appetite', 'polyuria', 'family_history', 'mucoid_sputum', 'rusty_sputum', 'lack_of_concentration', 'visual_distur
bances', 'receiving_blood_transfusion', 'receiving_unsterile_injections', 'coma', 'stomach_bleeding', 'distention_of_abdomen', 'history_of_alcohol_consumption', 'flui
d_overload', 'blood_in_sputum', 'prominent_veins_on_calf', 'palpitations', 'painful_walking', 'pus_filled_pimples', 'blackheads', 'scurring', 'skin_peeling', 'silver_
like_dusting', 'small_dents_in_nails', 'inflammatory_nails', 'blister', 'red_sore_around_nose', 'yellow_crust_ooze']
Enter Symptoms :
blister
skin_rash
itching
predict
You Have :
2019-11-28 01:58:09.471899: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library cublas64_100.dll
Impetigo

(base) C:\Users\shubh\Downloads\Minor\Model>
```

Fig 4.2.4

Model Prediction

The model is then hosted on a server and a custom-built API built using Flask helps the android application interact with the model.



```
(base) C:\Users\shubh\Downloads\Minor\Server>conda activate base

(base) C:\Users\shubh\Downloads\Minor\Server>C:/tools/Anaconda3/python.exe c:/User
2019-11-28 02:00:54.704874: I tensorflow/stream_executor/platform/default/dso_load
 * Serving Flask app "test" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment
   Use a production WSGI server instead.
 * Debug mode: on
Traceback (most recent call last):
  File "c:/Users/shubh/Downloads/Minor/Server/test.py", line 27, in <module>
    app.run(host="192.168.43.205", port = 5000, debug =True)
  File "C:\tools\Anaconda3\lib\site-packages\flask\app.py", line 990, in run
    run_simple(host, port, self, **options)
  File "C:\tools\Anaconda3\lib\site-packages\werkzeug\serving.py", line 988, in ru
    s.bind(server_address)
2019-11-28 02:01:33.502392: I tensorflow/stream_executor/platform/default/dso_load
er.cc:44] Successfully opened dynamic library cudart64_100.dll
 * Serving Flask app "test" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment
.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
2019-11-28 02:01:43.729680: I tensorflow/stream_executor/platform/default/dso_load
er.cc:44] Successfully opened dynamic library cudart64_100.dll
 * Debugger is active!
 * Debugger PIN: 213-133-021
 * Running on http://192.168.43.205:5000/ (Press CTRL+C to quit)
192.168.43.205 - - [28/Nov/2019 02:03:55] "GET /itching+skin_rash+blister+ HT_load
TP/1.1" 404 -                                                               :161
2019-11-28 02:04:26.311939: I tensorflow/stream_executor/platform/default/dso_load
er.cc:44] Successfully opened dynamic library nvcuda.dll
2019-11-28 02:04:27.112943: I tensorflow/core/common_runtime/gpu/gpu_device.cc:161
8] Found device 0 with properties:
name: GeForce MX250 major: 6 minor: 1 memoryClockRate(GHz): 1.0375
pciBusID: 0000:02:00.0
2019-11-28 02:04:27.133466: I tensorflow/stream_executor/platform/default/dlopen_c
hecker_stub.cc:25] GPU libraries are statically linked, skip dlopen check.
2019-11-28 02:04:27.142629: I tensorflow/core/common_runtime/gpu/gpu_device.cc:174
6] Adding visible gpu devices: 0
2019-11-28 02:04:27.151802: I tensorflow/core/platform/cpu_feature_guard.cc:142] Y
our CPU supports instructions that this TensorFlow binary was not compiled to use:
 AVX AVX2
```

```
Microsoft Windows [Version 10.0.18363.476]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\shubh\Downloads\Minor\Server>C:/tools/Anaconda3/Scripts/activate

(base) C:\Users\shubh\Downloads\Minor\Server>conda activate base

(base) C:\Users\shubh\Downloads\Minor\Server>curl http://192.168.43.205:5000/id/it
ching+skin_rash+blister+
Impetigo
(base) C:\Users\shubh\Downloads\Minor\Server>
```

Fig 4.2.5

Custom-built API using Flask

The Barcode scanner and the medicine reminder is implemented using Firebase ML kit API. The drug barcodes are of many formats. The ones majorly used on drugs are implemented. This helps the user get the details about the drug just by scanning the barcode. The database is hosted on Firebase which makes the updation and management of the database quite easy.

5. Limitations

- Concerning the Disease Prediction Model, the model could be better trained if the symptoms had a gradation scale rather than binary data corresponding to a particular disease. This though is a limitation of the dataset.
- The dataset could include more diseases and symptoms.
- For the barcode reader, the work could be done in real-time rather than taking a picture first and then decoding it. Also, some drugs may not have a barcode.

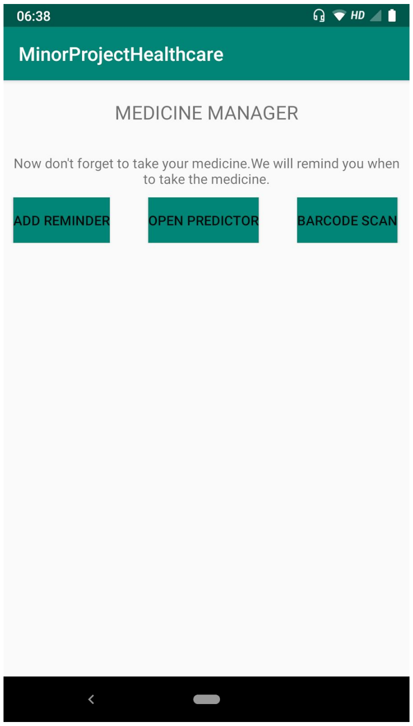
6.Findings, Conclusion and Future work

6.1 Findings

- The application is successful in predicting disease correctly from the given symptoms with nearing 100 percent accuracy.
- The app also accomplishes the task of reading the barcode from the label via the camera and shows details about the scanned drug.
- Medicine Reminder is working properly and reminding to take the medicine at the correct time.

6.2 Conclusion

Using neural network model to detect disease by taking input from user a list of symptoms and predicting using them. Scanning Barcode to find out the information about the scanned item from the database. Reminding the user to take the medicine based on the time and date entered by the user.

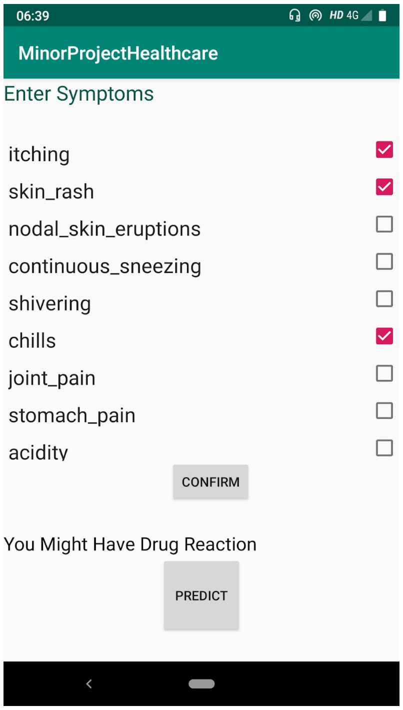


MEDICINE MANAGER

Now don't forget to take your medicine.We will remind you when to take the medicine.

Fig 6.2.1

MainActivity



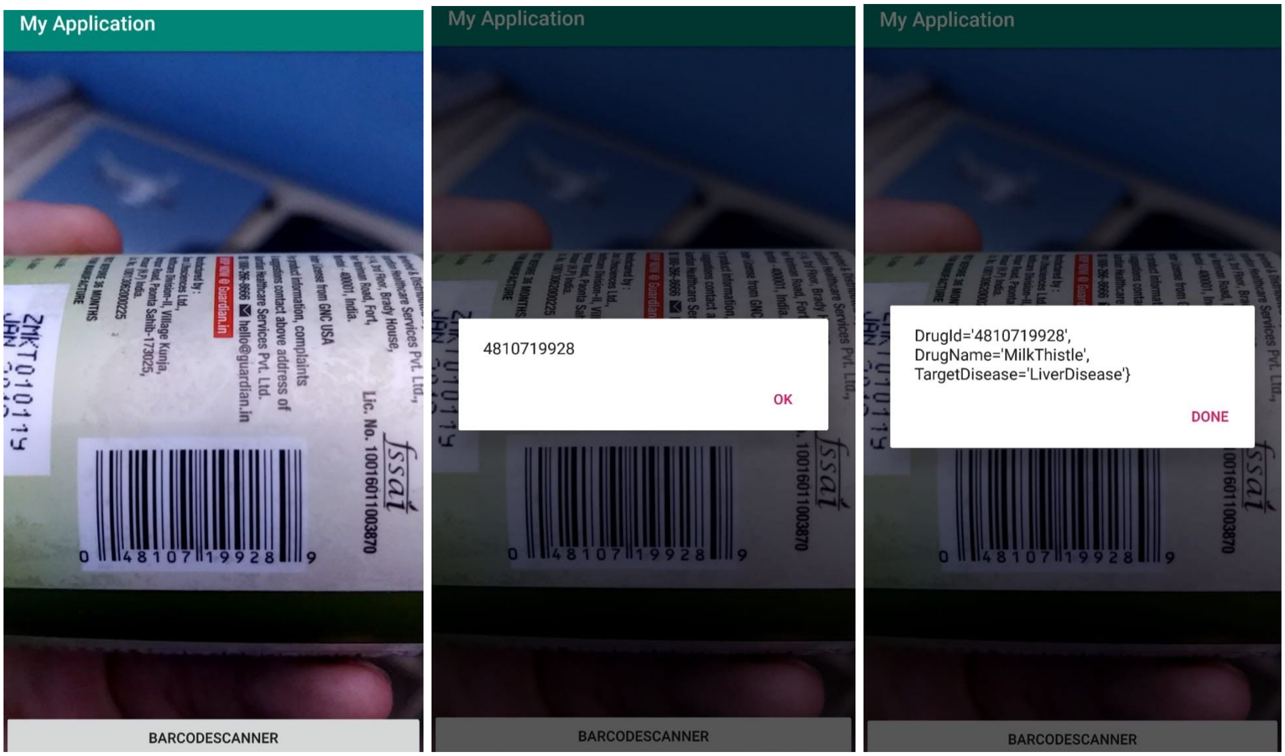
Enter Symptoms

itching ☑
skin_rash ☑
nodal_skin_eruptions ☐
continuous_sneezing ☐
shivering ☐
chills ☑
joint_pain ☐
stomach_pain ☐
acidity ☐

You Might Have Drug Reaction

Fig 6.2.2

Disease Predictor

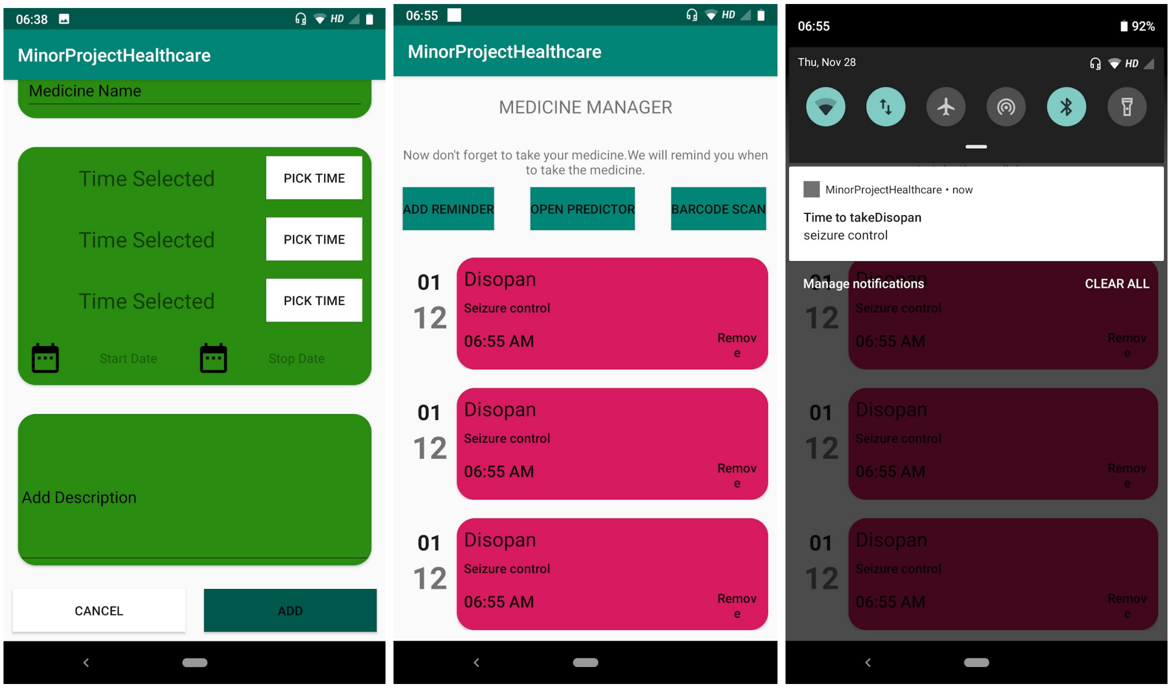


4810719928

DrugId='4810719928',
DrugName='MilkThistle',
TargetDisease='LiverDisease'}

Fig 6.2.3

Barcode Scanner



Medicine Name

Time Selected

Time Selected

Time Selected

Start Date

Stop Date

Add Description

MEDICINE MANAGER

Now don't forget to take your medicine.We will remind you when to take the medicine.

01 12 Disopan
Seizure control
06:55 AM

01 12 Disopan
Seizure control
06:55 AM

01 12 Disopan
Seizure control
06:55 AM

MinorProjectHealthcare • now

Time to takeDisopan
seizure control

Fig 6.2.4

Medicine Reminder

6.3 Future Work

A few improvements and updations that we would like to make to our project, given an opportunity, are-

- The barcode scanning could be done in real-time instead of taking a picture.
- The prediction model used could be improved as more refined and detailed data becomes public. The dataset could be larger.
- The app could also include a daily statistics module which displayed data like steps taken, distance walked, heart-rate, etc. using the relevant sensors and Google fit API.
- The user interface could always be improved upon.
- Login and SignIn Activity and personal information page

7. References

Guides and articles :

[1] https://developer.android.com/guide

[2] https://www.tensorflow.org/guide

[3] https://www.curiousily.com/posts/

[4] https://towardsdatascience.com/

[5] https://colab.research.google.com/notebooks

[6] https://firebase.google.com/

Research Papers for Literature Survey :

[7] Brian Wahl, Aline Cossy-Gantner, Stefan Germann, Nina R Shwalb,"Artificial Intelligence and global health:how can AI contribute to health in resource-poor settings?",*BMJ Journals BMJ Global Health Vol 3, Issue 4,* 24 February 2018

[8] Anandhavalli Muniasamy, Sehrish Tabassam, Mohammad A. Hussain, Habeeba Sultana, Vasanthi Muniasamy, Roheet Bhatnagar,"Deep Learning For Predictive Analytics in Healthcare", *International Conference on Advanced Machine Learning Technologies and Applications Vol 921*,pg. 32-42,2019

[9]Mehrdad Aghamohammadi, Manvi Madan, Jung Ki Hong, Ian Watson,"Predicting Heart Attack Through Explainable Artificial Intelligence", *International Conference On Computational Science Vol 11537,* pg.633-645, 2019

[10] D.Pavithra , Dr. A.N. Jayanthi,"A Study On Machine Learning Algorithm in Medical Diagnosis", *International Journal of Advanced Research in Computer Science Vol 9, No 4,*2018

[11] Rida Sara Khan, Asad Ali Zardar, Zeeshan Bhatti, ”Artificial Intelligence based Smart Doctor using Decision Tree Algorithm” In *Journal of Information & Communication Technology Vol 11 Issue 2,*2017

Side Papers we have read:

[12]Mallikarjuna B., Arun Kumar Reddy D. (2019) Healthcare Application Development in Mobile and Cloud Environments. In: *Internet of Things and Personalized Healthcare Systems. SpringerBriefs in Applied Sciences and Technology*. Springer, Singapore

[13]N. Weiskopf and C. Weng, "Methods and Dimensions of electronic health record data quality assessment: enabling reuse for clinical research," *Journal of the American Medical Informatics Association, vol. 20, no. 1*, pp. 144-151, 2013.

Resource and debugging :

[14] https://stackoverflow.com/

Database :

[15] https://www.kaggle.com/data/55703

[16] https://www.kaggle.com/data/49688