A PROJECT REPORT ON

ANIMAL DISEASE PREDICTION USING MACHINE LEARNING

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

BACHELOR OF ENGINEERING

In

COMPUTER ENGINEERING

Of

SAVITRIBAI PHULE PUNE UNIVERSITY

By

AKSHAY SHEKADE	**71813888M**
AMIT WAKADE	**71814137H**
VISHAL TAYADE	**71814049E**
SHUBHAM MAKAR	**71813439H**

Under the guidance of

Prof. JAYANTI KAMALASEKARAN

DEPARTMENT OF COMPUTER ENGINEERING

SINHGAD COLLEGE OF ENGINEERING, PUNE-41

Accredited by NAAC

Sinhgad Technical Education Society,
Sinhgad College of Engineering , Pune-41
Department of Computer Engineering

Date:

CERTIFICATE

This is to certify that the project report entitled

Animal Disease Prediction Using Machine learning

Submitted by

AKSHAY SHEKADE	**71813888M**
AMIT WAKADE	**71814137H**
VISHAL TAYADE	**71814049E**
SHUBHAM MAKAR	**71813439H**

is a bonafide work carried out and is approved for the partial fulfillment of the requirements of Savitribai Phule Pune University, Pune for the award of the degree of Bachelor of Engineering (Computer Engineering) during the year 2019-20.

Prof. Jayanti Kamalasekaran Prof. M. P. Wankhade Guide Head
Department of Computer Engineering Department of Computer Engineering

Dr. S.D. Lokhande
Principal
Sinhgad College of Engineering

ACKNOWLEDGEMENT

We feel great pleasure in expressing our deepest sense of gratitude and sincere thanks to our guide **Prof. Jayanti Kamalasekaran** for her valuable guidance during the Project work, without which it would have been very difficult task. I have no words to express my sincere thanks for valuable guidance, extreme assistance and cooperation extended to all the **Staff Members** of our Department.

This acknowledgement would be incomplete without expressing our special thanks to **Prof. M.P. Wankhade Head of the Department (Computer Engineering)** for his support during the work.

We would also like to extend our heartfelt gratitude to our **Principal, Dr. S. D. Lokhande** who provided a lot of valuable support, mostly being behind the veils of college bureaucracy.

Last but not least we would like to thank all the Teaching, Non- Teaching staff members of our Department, our parents and colleagues those who helped us directly or indirectly for completing this Project successfully.

ABSTRACT

Major animal diseases pose a great threat to animal husbandry and human beings. With the deepening of globalization and the abundance of data resources .

The prediction and analysis of animal diseases by using machine learning are becoming more and more important. Our proposed model provides an automatic method to determine animal disease using a trained dataset of disease symptoms. The model is use to predict the disease by checking the symptoms entered by user by using random forest algorithm in supervised machine learning. This model provides a methodology that how the use of machine learning can detect animal diseases which can provide economical medical solution to place with scarce in medical facilities for farm animals.

In our model disease of animal is predicted and the prevention of disease is proposed automatically. Our model provides useful estimation and prediction of diseases causing agent with necessary precautions.

List of Figures

List of Tables

Abbreviations

- RFA : Random forest Algorithm
- NLP : Natural Language Processing
- GUI : Graphical User Interface

TITLE **PAGE NO**

Chapter No. Title Page No.

Chapter 1
Introduction

1.1. Background and Basics

1. Health issues in animals is one of the important factor which have a direct implication over the quality and production efficiency of farm production . Farm animals are prone to many pulmonary, infections, gastrointestinal diseases .Some of the diseases like mastitis can even lead to death of the animals.

2. The demand for the poultry is rising and with agriculture sector being the occupation of more than 40% in many developing nations use and growth of technology in daily practices for farm production should be a major concern.

3. Most of the dairy farmers of the developing countries are not moneyed enough to have regular diagnosis of their animals which can detect disease at an early stage and the medical infrastructure is not available at many places .Many farmers from villages have to take their cattle to very far distance for getting to a medical facility.

4. An early diagnosis with an automated system which would record all the vital monitoring information related to cattle which would be helpful in speeding up the healing process and can avoid the deaths due to unavailability or lack of veterinary specialist at the locality .

5. Speaking of India which is having a huge animal population is deprived of doctors in the rural sectors. A system is needed for spreading awareness in villages regarding the prevention of these animal diseases which would lead to reduction in dependency of medical expertise and would also reduce the expenditure of transportation of livestock in big cities or towns during a medical condition.

6. System presented in this model deals with an approach of having a hardware containing within a appropriate database which include symptoms and data of commonly occurring health care problems and can report and monitor the health condition. This can be accomplished with the help of machine learning algorithms which identifies the scenario and provides a pre diagnosis of the problem with the data collected from sensors which is matched from the internal database.

1.2. Literature Survey

S.No.	Paper Name	**Overview**	**Limitation**
1.	Multi-Criterion Disease Detection for Canines using Unsupervised Machine Learning.	The main purpose of this research is to diagnose the Canine diseases using Deep Learning. Deep Learning uses algorithms known as Neural Networks. The primary software tool of Deep Learning is Tensor Flow.	No instant medical precaution measures are provided.
2.	A Machine Learning Approach to Automated Gait Analysis for the Noldus Catwalk System.	The present work showed the ability of Machine Learning system to discriminate pharmacologically relevant animal diseases base on their walking Behaviour.	Disease detection primarily relies on a specific gait system. Other symptoms are not considered.
3.	A modelling of animal disease using artificial neural network.	The model is used to identify Animal disease by using large number of animal data with the help of ANN by using data set. The parameter identification under consideration is types of animal disease, types of species and locations of disease based on the Geographical Information System (GIS) data set.	Prediction of disease is mainly based on type of animal species found in specific location.

Table 1.Literature Survey

1.3 Project Undertaken 1.3.1. Problem Definition:

Prediction of different animal diseases using random forest algorithm by entering proper disease symptoms.

1.3.2. Scope Statement:

1. The recommendation for further study of this animal disease, we suggest to add more parameter to get a more accurate result, such as the parameter of time and temperature, which is we can predict the future time of disease will occur and to predict a better temperature for habitat, hence to maintain the population of the species.

2. Then, from the previous study of genetic algorithm, it is better if we use a huge data to obtain the optimal solution and it is desirable to maintain the population size as large as possible not only use a slight number of cases and parameter. Besides, the weakness GA method is using the limited data set in the previous study.

1.4. Organization of the Project Report

The Report is divided into Four Chapters. Chapter 2, discusses about Project Planning and Management that covers System Overview and functional and nonfunctional requirements. Chapter 3 discusses about Analysis and Design. Chapter 4 about Implementation and coding.

Chapter 5 about Testing.

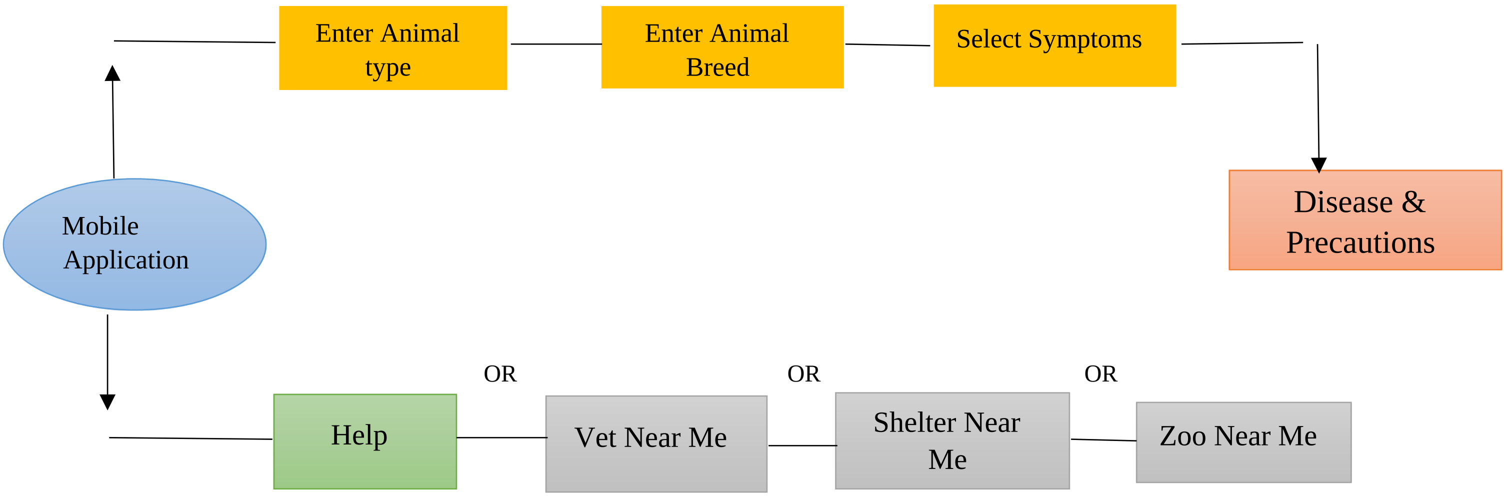


Fig 1. System Overview

- First of all we have to enter mobile number for verification purpose then otp will be get generated and will get transferred to the entered mobile number.
- After that user will have to enter the type of animal(eg. Cow, Buffelo, Goat, dog etc) and also have to select their breed.
- Then entered all the symptoms which your animal have then wait for a seconds model will work on the entered symptoms will give you a final result.
- After getting result you will see precautions and help option is also given for user.

2.2 Detailed Requirement Specification

2.2.1 Functional Requirements:

- As model works on the inputs model should properly processed inputs and should calculate correct result.
- For handling of large amount of data related to symptoms and their precautions efficient algorithms should be selected Algorithm should work fast and should consumed less time for showing Result.
- Proper server(Firebase, dJango etc) should be selected for storing datasets.

- Help option which is given in the option bar should work properly in case of emergency.

2.2.2. Non - Functional Requirements:

1. Performance Requirements:

   The application should be adaptable to various situations where different types of inputs provided by different users.

   The system must be robust enough to handle any data which is provided as the input to it and should be able to compute the correct output.

2. Safety Requirements:

   Users' credentials should be transferred to the server safely and securely. The result should not taken more time to display or more time to send the mail.

3. Security Requirements

   Authentication and privacy of the users must be maintained.

4. System Security

   Individual will have their separate login and their login credential is included in the database.

   User will have their login and password, so that their profile will not be accessible to any other user.

Software Requirements:

1. Visual Studio

2. Android Studio

3. Python(import required packages)

4.Active Firebase server with created account **Hardware**

Requirements:

1. Node component (PC/Laptop)

2.3 Project Process Modelling

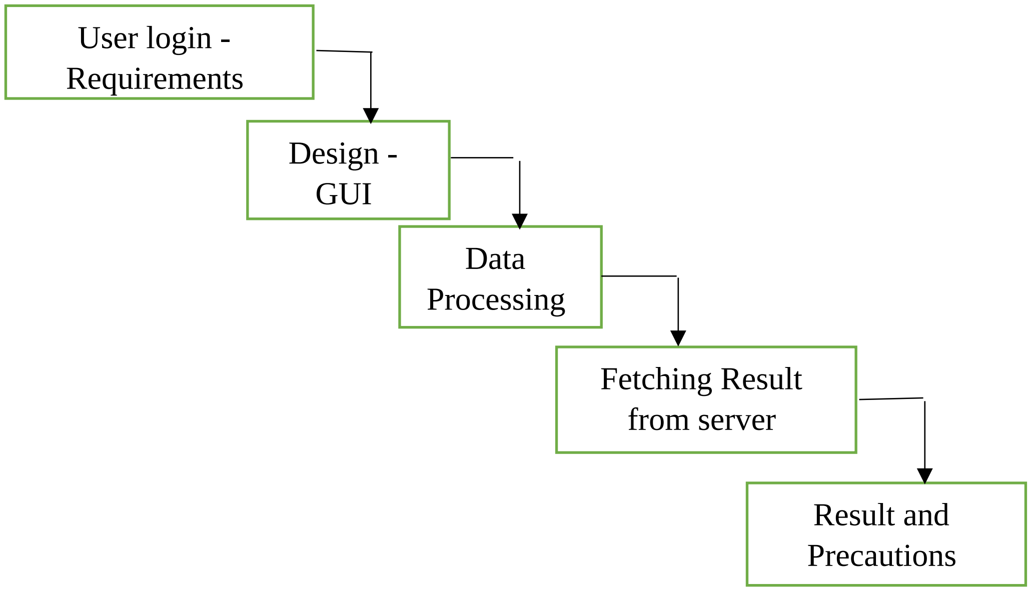


Fig 2.Project Processing Model

For making application more reliable and easy to use we should segregate different parts in different module such that we can work on them more freely and should think on how we can improve our system ,so step by step work should be done.

Machine learning is programming computers to optimize a performance using example data or past data. Machine learning is study of computer systems that learn from data and experience .Machine learning algorithm has two passes: Training, Testing .Prediction of a disease by using patient's symptoms and history machine learning technology is struggling from past decades. Machine Learning technology gives a good platform in medical field, so that a healthcare issues can be solved efficiently.

We are applying machine learning to diagnose a different diseases of different dairy and farm animals which allows building models to get quickly analyse data and deliver results faster, with the use of machine learning technology doctors can make good decision for patient diagnose and treatment options, which leads to improvement of patient healthcare services. Healthcare is the most prime example of how machine learning is use in medical field. To improve the accuracy from a large data, the existing work will be done on unstructured and textual data .For prediction of diseases the existing will be done on linear, KNN, Decision Tree algorithm.

2.4 Cost and Efforts Estimates

Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e. number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the variou parameters associated with making a project such as size, effort, cost, time and quality. The key parameters which define the quality of any software products, which are also an outcome of the Cocomo are primarily Effort and Schedule:

Effort: Amount of labor that will be required to complete a task. It is measured in person-months units.

Schedule: Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months.

Different models of Cocomo have been proposed to predict the cost estimation at different levels, based on the amount of accuracy and correctness required. Boehm's definition of organic, semidetached, and embedded systems.

Organic

A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has been solved in the past and also the team members have a nominal experience regarding the problem.

Semi detached

A software project is said to be a Semi-detached type if the vital characteristics such as team-size, experience, knowledge of the various programming environment lie in between that of organic and Embedded. The projects classified as Semi-Detached are comparatively less familiar and difficult to develop compared to the organic ones and require more experience and better guidance and creativity.

Embedded

A software project with requiring the highest level of complexity, creativity, and experience requirement fall under this category. Such software requires a larger team size than the other two models and also the developers need to be sufficiently experienced and creative to develop such complex models.

All the above system types utilize different values of the constants used in Effort Calculations.

As from the characteristic stated above, we can draw our project to be Organic.

Chapter 3
Analysis and Design

Introduction

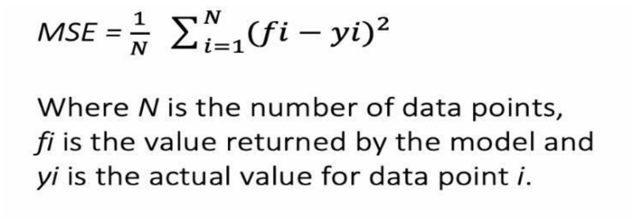
This chapter covers the analysis and design of the considered system.

3.2 Mathematical Modeling

The Mathematics behind Random Forest Algorithm

Regression Problems

When using the Random Forest Algorithm to solve regression problems, you are using the Mean Square Error(MSE) to how your data branches from each node.



$$MSE = \frac{1}{N} \sum_{i=1}^{N}(fi - yi)^2$$

Where N is the number of data points,
fi is the value returned by the model and
yi is the actual value for data point *i*.

This formula calculates the distance of each node from the predicted actual value, helping to decide which branch is the better decision for your forest. Here, *yi* is the value of the data point you are testing at a certain node and *fi* is the value returned by the decision tree.

Classification Problems

When performing Random Forests based on classification data, you should know that you are often using the Gini Index or the formula used to decide how nodes on a decision tree branch.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

This formula uses the class and probability to determine the Gini of each branch on a node, determining which of the branches is more likely to occur. Here, *pi* represents the relative frequency of the class you are observing in the dataset and *c* represents the number of classes. You can also use entropy to determine how nodes branch in a decision tree

$$Entropy = \sum_{i=1}^{C} -p_i * \log_2(p_i)$$

Entropy uses the probability of a certain outcome in order to make a decision on how the node should branch. Unlike the Gini index, it is more mathematical intensive due to the logarithmic function used in calculating it.

3.3 IDEA Matrix

IDEA	Deliverable	Parameter Affected
Increase	• Increase the accuracy of predicting correct animal disease with given symptoms. • Increase the efficiency of the Random Forest Algorithm.	• Accuracy • Performance • Efficiency • Time (Search)
Improve	• Improve possible precautions, information about diseases and veterinary. • Improve request and response time. • Improve correctness by using appropriate datasets.	• Requesting and Response of information. • Time • Dataset • Performance
Ignore	• Ignored the attributes not affecting the disease prediction in animals.	• None
Invent	• Inventing systematic operation which gets correct symptoms and gives correct solution.	• Invention • Operation
Deliver	• Application for the user, which can be used to predict the animal diseases.	• Distribution • Application • Prediction
Decrease	• Decrease the request and response time. • Decrease the error rate close to zero.	• Time • Error
Evaluate	• Evaluate that the system works smoothly without any inconvenience. • Evaluate that any necessary functionalities do not fail.	• Testing • Transparency • Accuracy

Eliminate	• Eliminate unnecessary attributes, to give greater accuracy. • Eliminate time required by manual tasks by automating.	• Accuracy • Intervention • Time • Space
Accelerate	• Right algorithm and libraries accelerate the performance of the system. • Removing the redundancy. • Reduce the operation time.	• Efficiency • Simplicity • Redundancy • Complexity
Avoid	• Avoid the system crashing. • Handle the Exceptions properly.	• System Performance • Malfunction • Software quality
Associate	• Improve performance between the user and the UI. • Application should be in synchronization with reality.	• Recognition • Reliability • Association

Table 3.1 IDEA Matrix

3.4 Architecture Diagram

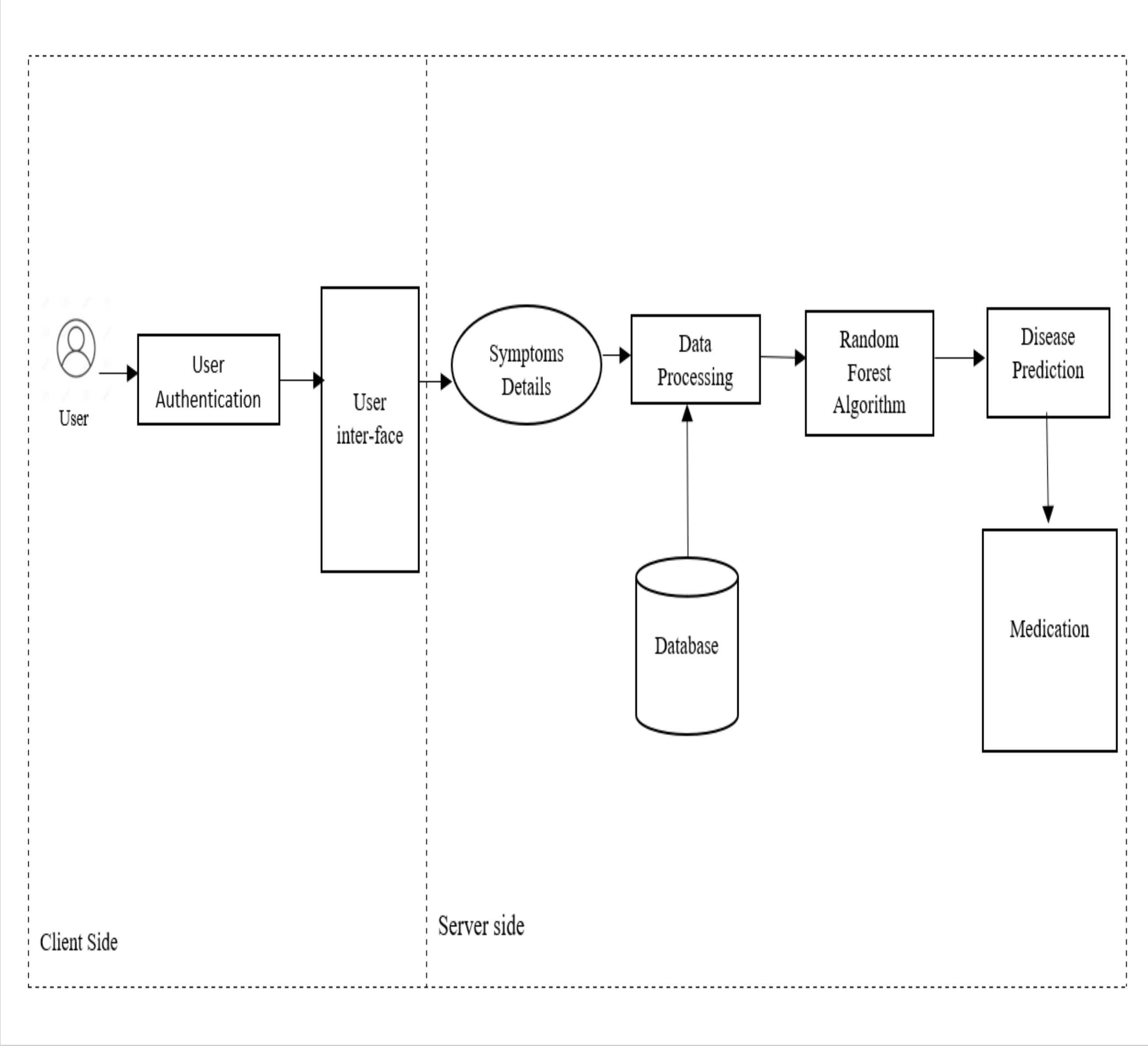


Fig. Architecture Diagram

System flow:-

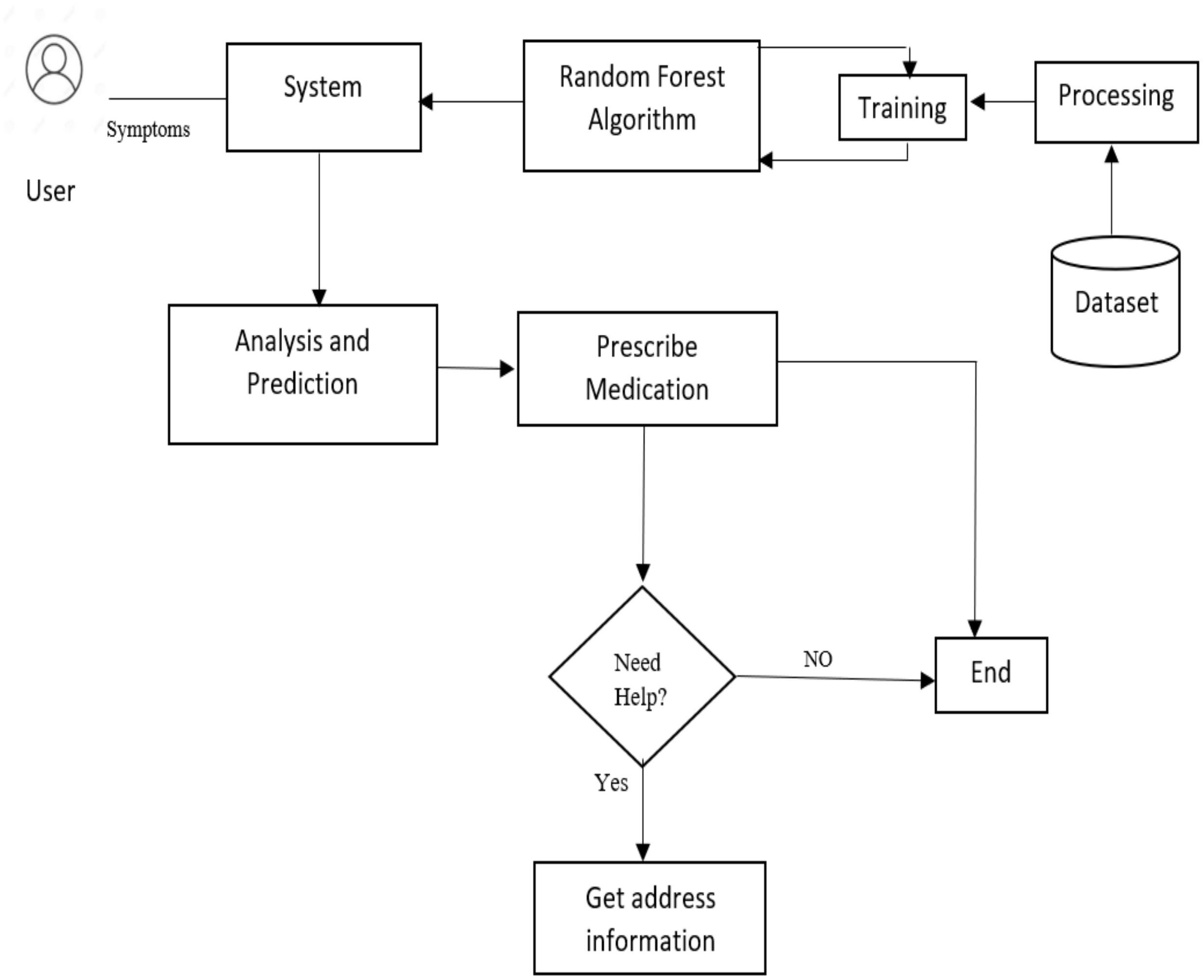


Fig. System flow

3.5 UML Diagrams

3.5.1 Use Case Diagram

Use-Case diagram represents the functional requirements of the system. This use case diagram covers following functional requirements:

1. Candidate Login and Logout

   1.1 Checking Login credentials

   1.2 If credentials are not as expected, display login error.

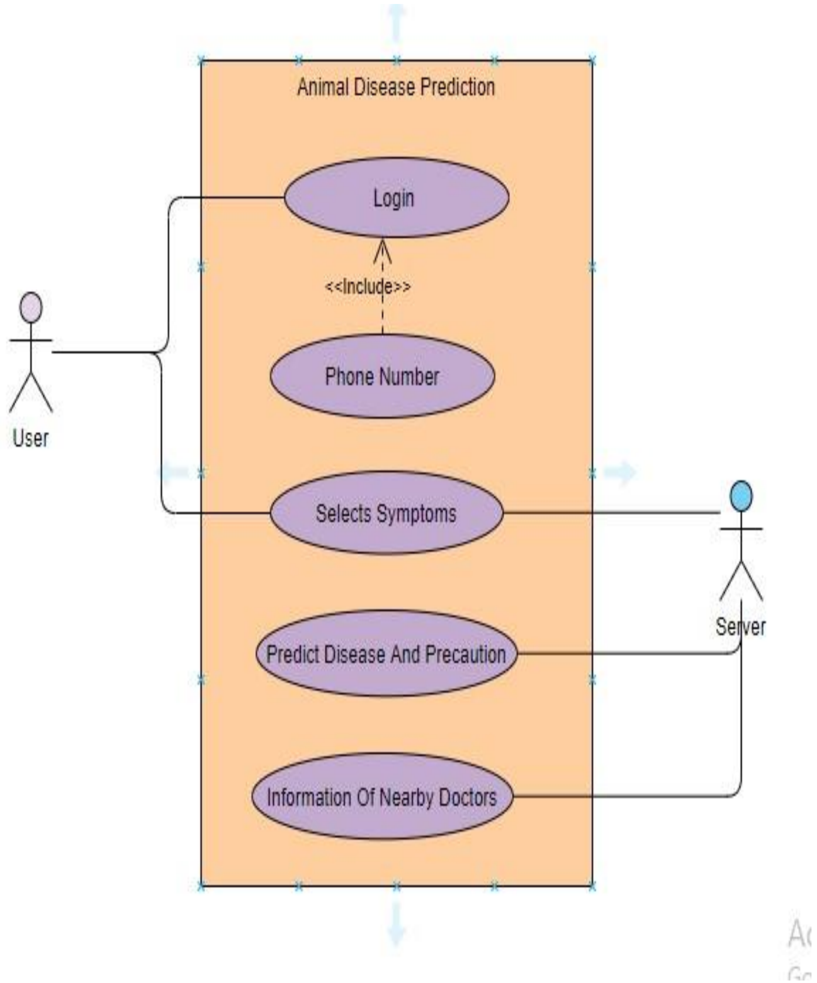

Animal Disease Prediction
Login
<<Include>>
Phone Number
User
Selects Symptoms
Server
Predict Disease And Precaution
Information Of Nearby Doctors

Fig 3.4 UML Diagram.

3.5.2 Activity Diagram

Activity Diagram is basically a flowchart to represent the flow from one activity to another activity. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.

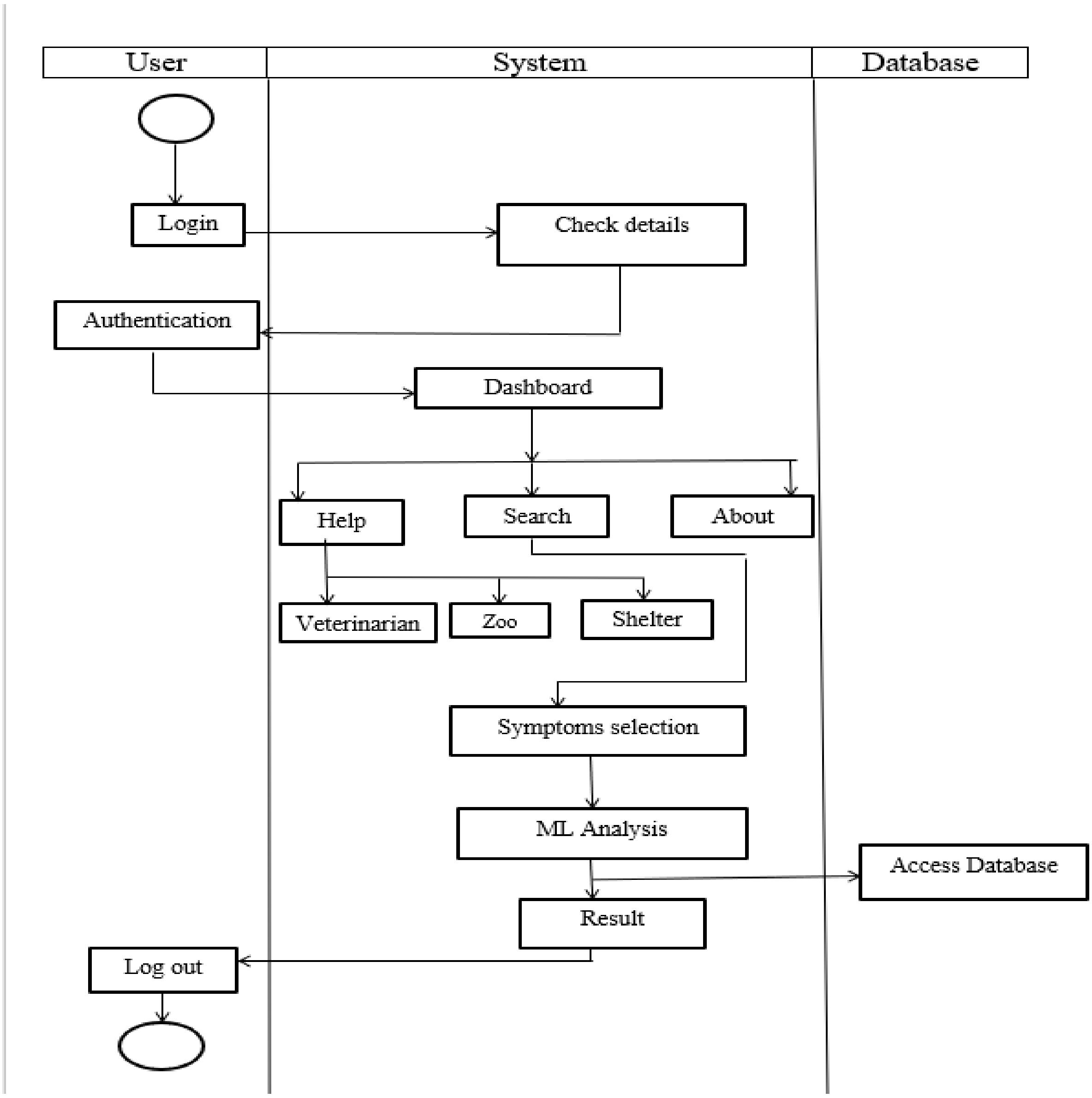


Fig 3.5 Activity Diagram

3.5.3 Class Diagram

Class Diagram is a type of static diagram that describes the structure of system by showing the system's classes, their attributes, operations and relationship among objects.

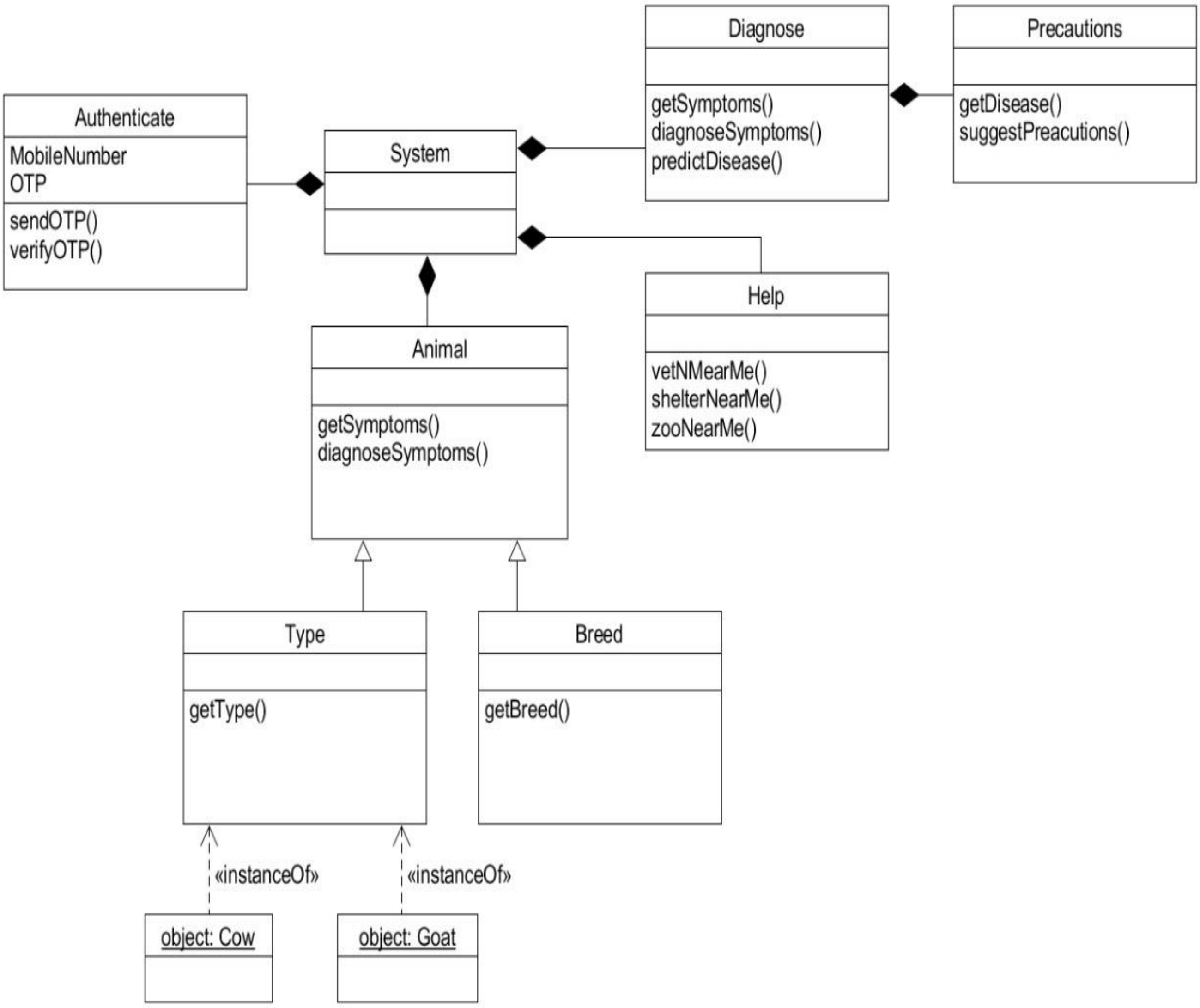


Fig. Class Diagram

3.5.4 Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place.

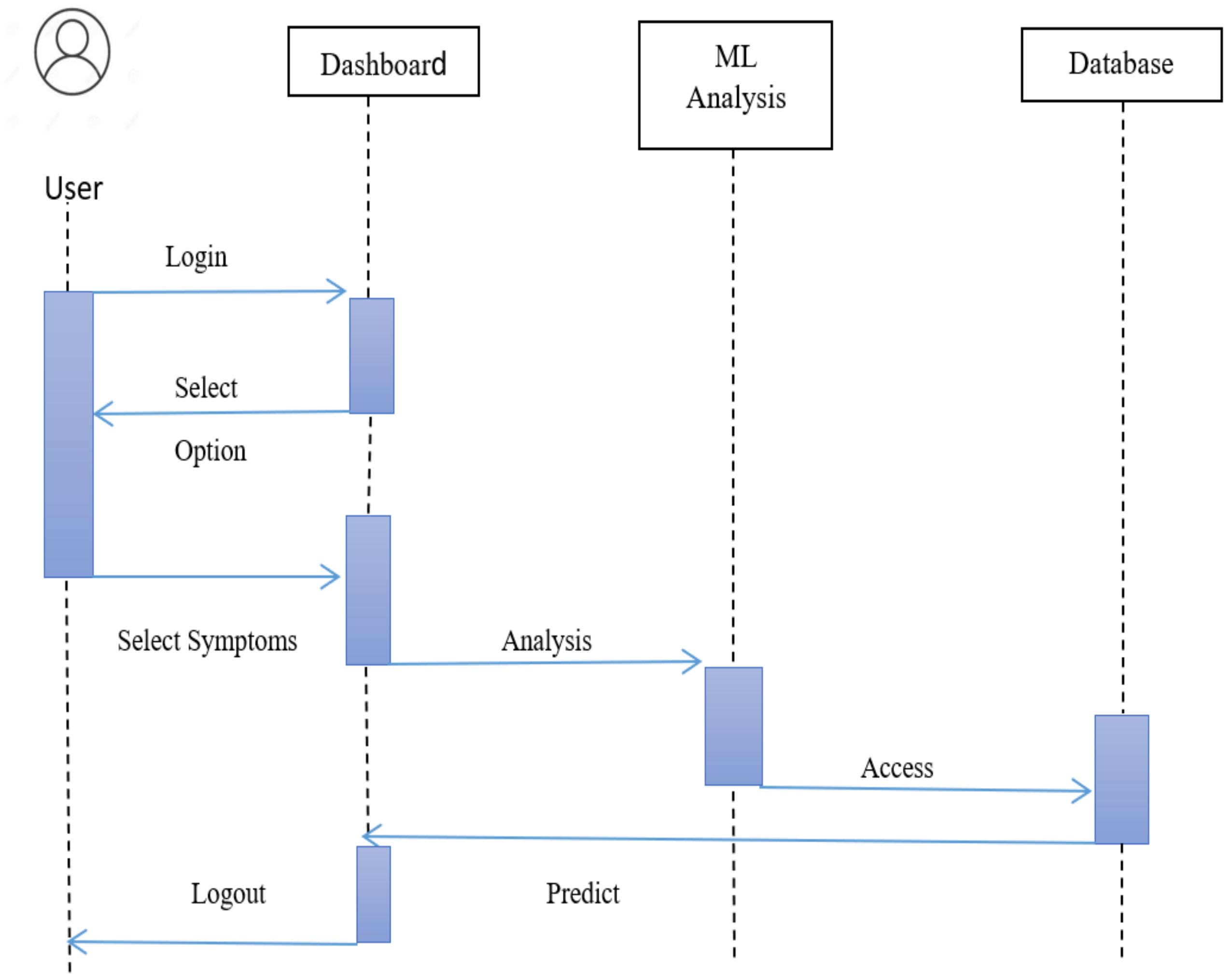


Fig 3.6 sequence diagram

Deployment Diagram:-

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middle ware connecting them.

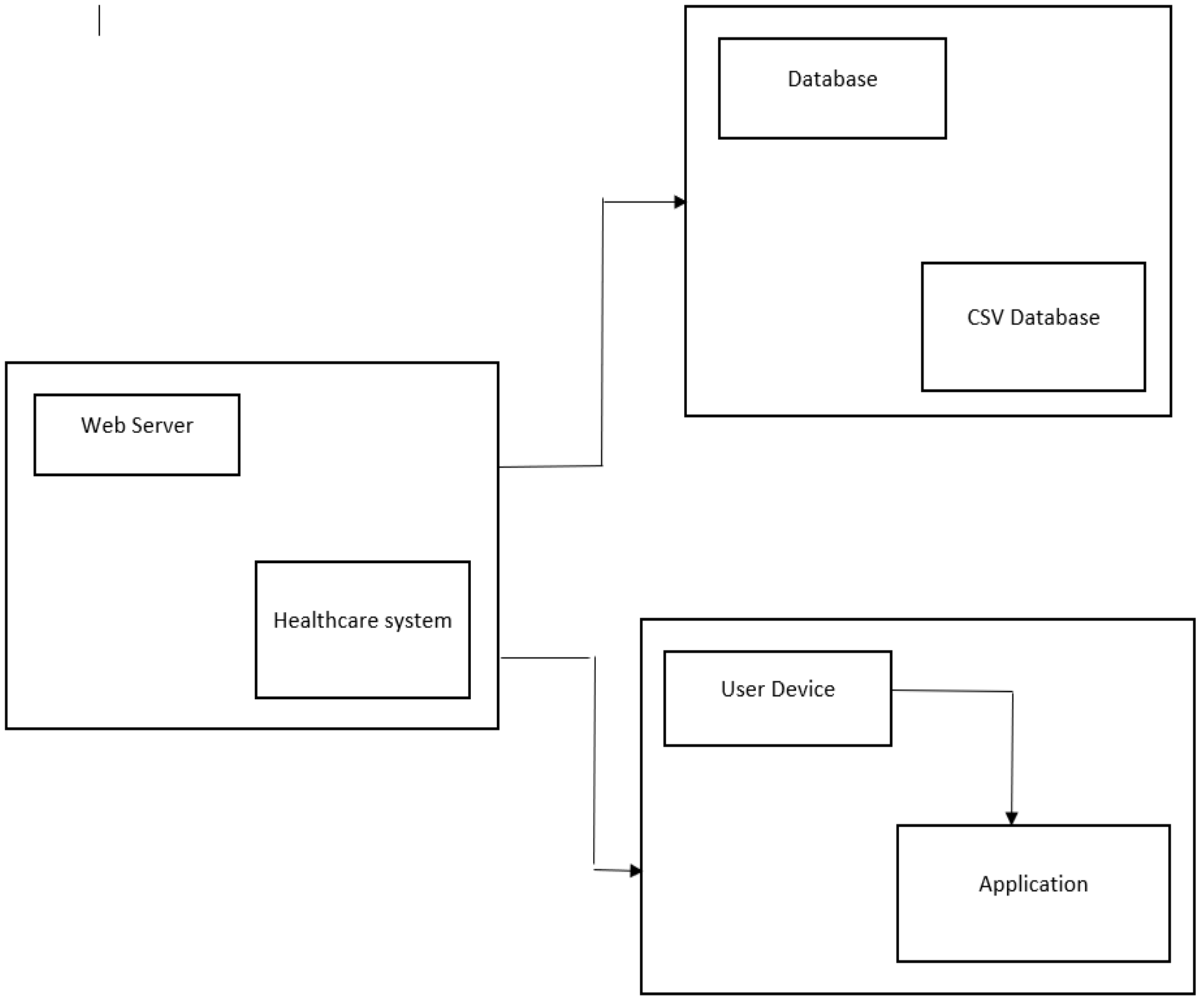


Fig. Deployment Diagram

Chapter 4 Implementation and coding

4.1Introduction

This chapter covers the role of various subsystems/modules/classes along with implementation details listing of the code for the major functionalities

2.Coding

```
 package com.techdecode.sadfghjk;


 import
android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import
android.widget.Button;
import
android.widget.EditText;
import
android.widget.TextView;
import android.widget.Toast;

  import androidx.annotation.NonNull; import
  androidx.appcompat.app.AppCompatActivity;

  import
  com.google.android.gms.tasks.OnCompleteListener;
  import com.google.android.gms.tasks.Task; import
  com.google.firebase.FirebaseException; import
  com.google.firebase.auth.AuthResult; import
com.google.firebase.auth.FirebaseAuth; import
com.google.firebase.auth.FirebaseUser; import
com.google.firebase.auth.PhoneAuthCredential;
import
com.google.firebase.auth.PhoneAuthProvider;

import java.util.concurrent.TimeUnit;
```

```
public class MainActivity extends AppCompatActivity {



   EditText e1, e2;
   Button b1, b2;
   TextView t1;
   FirebaseAuth mauth;
   FirebaseUser user;
   String mVerificationId;
   ProgressDialog progressDialog;
   PhoneAuthProvider.ForceResendingToken mResendToken;
   PhoneAuthProvider.OnVerificationStateChangedCallbacks callbacks;
   String phonenumber;
   @Override    protected void onCreate(Bundle
savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);        mauth =
FirebaseAuth.getInstance();        e1 = (EditText)
findViewById(R.id.pnumber);        e2 = (EditText)
findViewById(R.id.code);        b1 = (Button)
findViewById(R.id.send_code);        b2 = (Button)
findViewById(R.id.verify);
t1=(TextView)findViewById(R.id.t1);
 progressDialog = new ProgressDialog(this);
      b1.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            String phonenumber= t1.getText().toString()+e1.getText().toString();
if (TextUtils.isEmpty(phonenumber)) {
               Toast.makeText(com.techdecode.sadfghjk.MainActivity.this, "Enter the phone number",
Toast.LENGTH_SHORT).show();
            } else {
progressDialog.setTitle("phone verfication");
progressDialog.setMessage("wait plz");
progressDialog.show();
```

```
   progressDialog.setCanceledOnTouchOutside(false);
                  PhoneAuthCredential credential = PhoneAuthProvider.getCredential(mVerificationId,
verfycode);

                 signInWithPhoneAuthCredential(credential);

              }

            }


       }) ;
       callbacks = new PhoneAuthProvi der.OnVerificationStateChangedCallbacks() {
           @Override              public void onVerificationCompleted(PhoneAuthCredential

phoneAuthCredential) {                signInWithPhoneAuthCredential(phoneAuthCredential);

}



           @Override              public void
onVerificationFailed(FirebaseException e) {
          progressDialog.dismiss();
                    Toast.makeText(com.techdecode.sadfghjk.MainActivity.this, "Invalid number with
country code", Toast.LENGTH_SHORT).show();

b1.setVisibility(View.VISIBLE);

e1.setVisibility(View.VISIBLE);
          t1.setVisibility(View.VISIBLE);
          b2.setVisibility(View.INVISIBLE);
          e2.setVisibility(View.INVISIBLE);


           }
```

```
public void onCodeSent(String verificationId,


                                    PhoneAuthProvider.ForceResendingToken token) {



                mVerificationId = verificationId;
mResendToken = token;
progressDialog.dismiss();
                Toast.makeText(com.techdecode.sadfghjk.MainActivity.this, "Code is sent",
Toast.LENGTH_SHORT).show();
                b1.setVisibility(View.INVISIBLE);
e1.setVisibility(View.INVISIBLE);
t1.setVisibility(View.INVISIBLE);
b2.setVisibility(View.VISIBLE);
e2.setVisibility(View.VISIBLE);


            }
        };


    }


    private void signInWithPhoneAuthCredential(PhoneAuthCredential credential) {
mauth.signInWithCredential(credential)
                .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
if (task.isSuccessful()) {                              phonenumber=
t1.getText().toString()+e1.getText().toString();


                            progressDialog.dismiss();
                            Toast.makeText(com.techdecode.sadfghjk.MainActivity.this, "sucess",
Toast.LENGTH_SHORT).show();
                            Intent intent = new Intent(com.techdecode.sadfghjk.MainActivity.this,
first.class);                            startActivity(intent);                            finish();
} else {

                    String msg = task.getException().toString();
                    Toast.makeText(com.techdecode.sadfghjk.MainActivity.this, "Error " + msg,
Toast.LENGTH_SHORT).show();

                }
                }

            });
  }



  @Override

  protected void onStart() {

      super.onStart();


      user= FirebaseAuth.getInstance().getCurrentUser();


      if(user!=null)
      {
           Intent intent = new Intent(com.techdecode.sadfghjk.MainActivity.this,first.class);


startActivity(intent);          }

}

}
```

#Code for Search option

```
package com.techdecode.sadfghjk;

            import android.o

           s.Bundle; import

android.view.LayoutInflater;

             import
android.view.View;
             import
android.view.ViewGroup;
             import
android.widget.AdapterView; import
android.widget.ArrayAdapter;
import android.widget.Spinner;


import androidx.annotation.NonNull;

            import android

           x.annotation.Nullable; import

androidx.fragment.app.Fragment;



             public class search extends Fragment implements
AdapterView.OnItemSelectedListener {

    @Nullable     @Override        public View onCreateView(@NonNull LayoutInflater inflater,
@Nullable ViewGro   up container, @Nullable Bundle savedInstanceState) {



        String [] data = {"Animal info","Symtoms","Result"};

        int current=0;

        final View view = inflater.inflate(R.layout.search, container, false);

        Spinner spinner = view.findView ById(R.id.spiner);
       ArrayAdapter<CharSequence> adapter =
ArrayAdapter.createFromResource(getContext(),R. array.animal,android.R.layout.simple_spinner_item);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);


        spinner.setAdapter(adapter);

        spinner.setOnItemSelectedListener(this);

 return view;
```

#Code For Navigation Buttons

```
package com.techdecode.sadfghjk;


import android.content.Intent;
import android.os.Bundle; import
android.view.MenuItem;

import androidx.annotation.NonNull; import
androidx.appcompat.app.AppCompatActivity; import
androidx.fragment.app.Fragment; import
com.google.android.material.bottomnavigation.BottomNavigationView;

public class first extends AppCompatActivity {
    @Override     protected void onCreate(Bundle
savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_first);
        BottomNavigationView bottomNavigationView = findViewById(R.id.bottomview);
bottomNavigationView.setOnNavigationItemSelectedListener(nav);
getSupportFragmentManager().beginTransaction().replace(R.id.fragcont, new help()).commit();
    }

    private BottomNavigationView.OnNavigationItemSelectedListener nav = new
BottomNavigationView.OnNavigationItemSelectedListener() {
        @Override
        public boolean onNavigationItemSelected(@NonNull MenuItem menuItem)
{             Fragment select = null;             switch (menuItem.getItemId()) {
case R.id.help:
select = new help();
getSupportFragmentManager().beginTransaction().replace(R.id.fragcont,select).commit();
break;} return true;}};
}
```

5.1 Introduction

This chapter covers the testing approach used and the test cases.

5.2 Unit Testing

Unit testing is a level of software testing where individual units/components of a software tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

TEST CASE ID	TEST CASE	INPUT	EXPECTED OUTPUT
UT1	Mobile no Verification	Mobile Number	OTP should be generated.
UT2	Animal Specification	Animal Type Animal breed	Data should be store
UT3	Symptoms Specification	Select Symptoms	Symptoms Should get store
UT4	Help option	Select necessary option	Option should work properly.

Table 4.1 Unit Testing

5.3 Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing. It is a level of software testing where individual units are combined and tested as a group.

The purpose of this level of testing is to expose faults in the interaction between integrated units.

TEST CASE ID	TEST CASE	INPUT	EXPECTED OUTPUT
IT1	Mobile Verification	Mobile number	OTP should Generate Successfully.
IT2	Disease diagnosis	Symptoms	Predicted disease

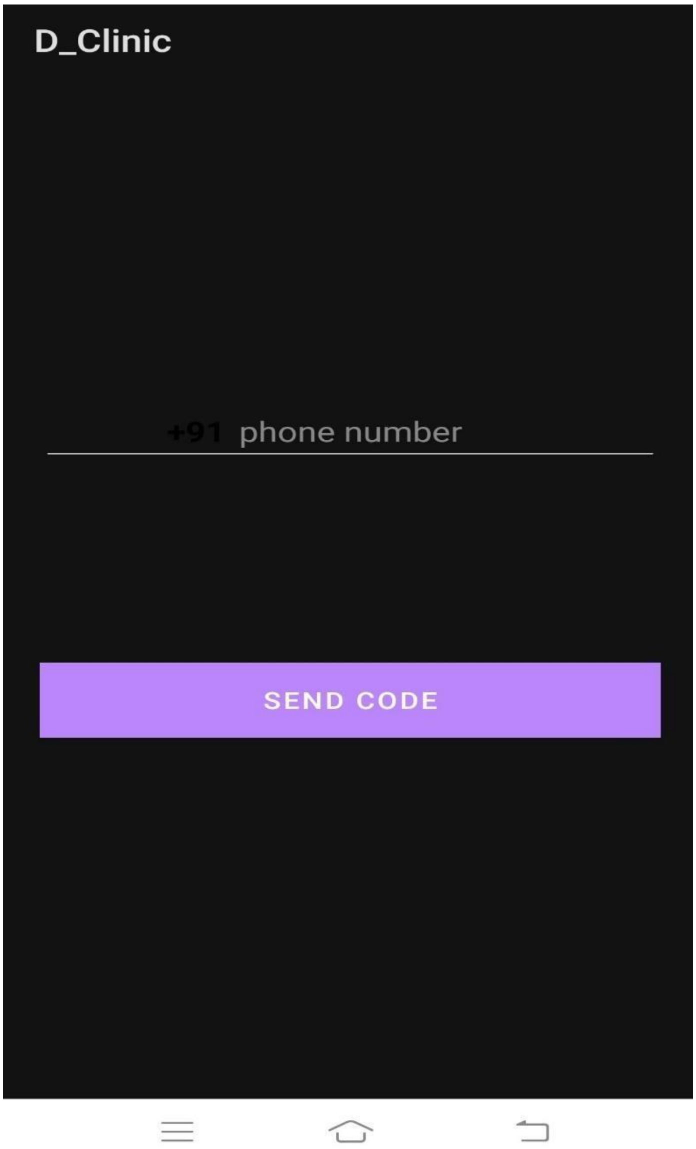
Table 4.2 Integration Testing

Chapter 6

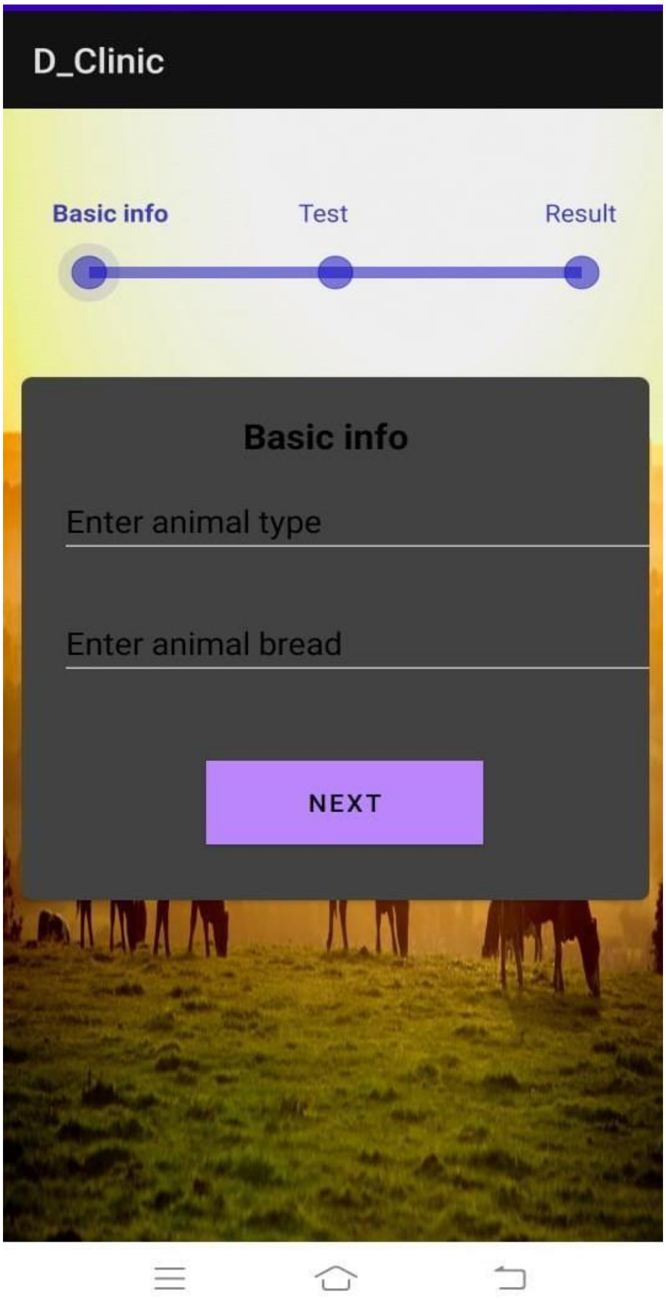
Results and Discussions

6.1 Main GUI Snapshots

This is the home page of our mobile app.



This is second Page of our mobile app.



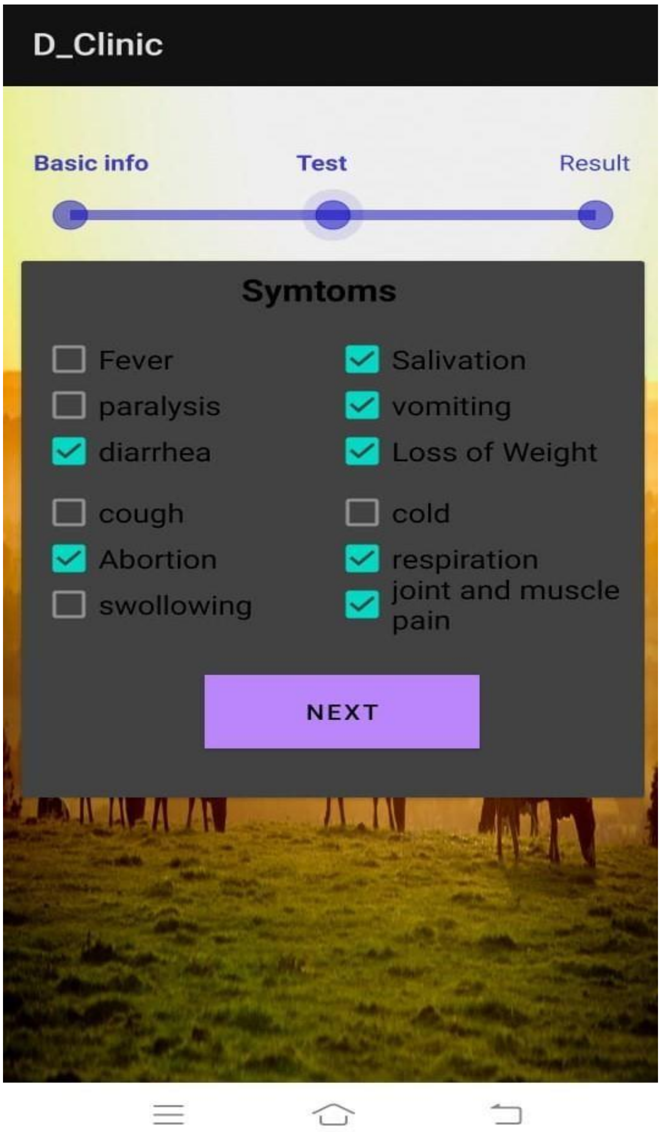
D_Clinic

Basic info

Enter animal type

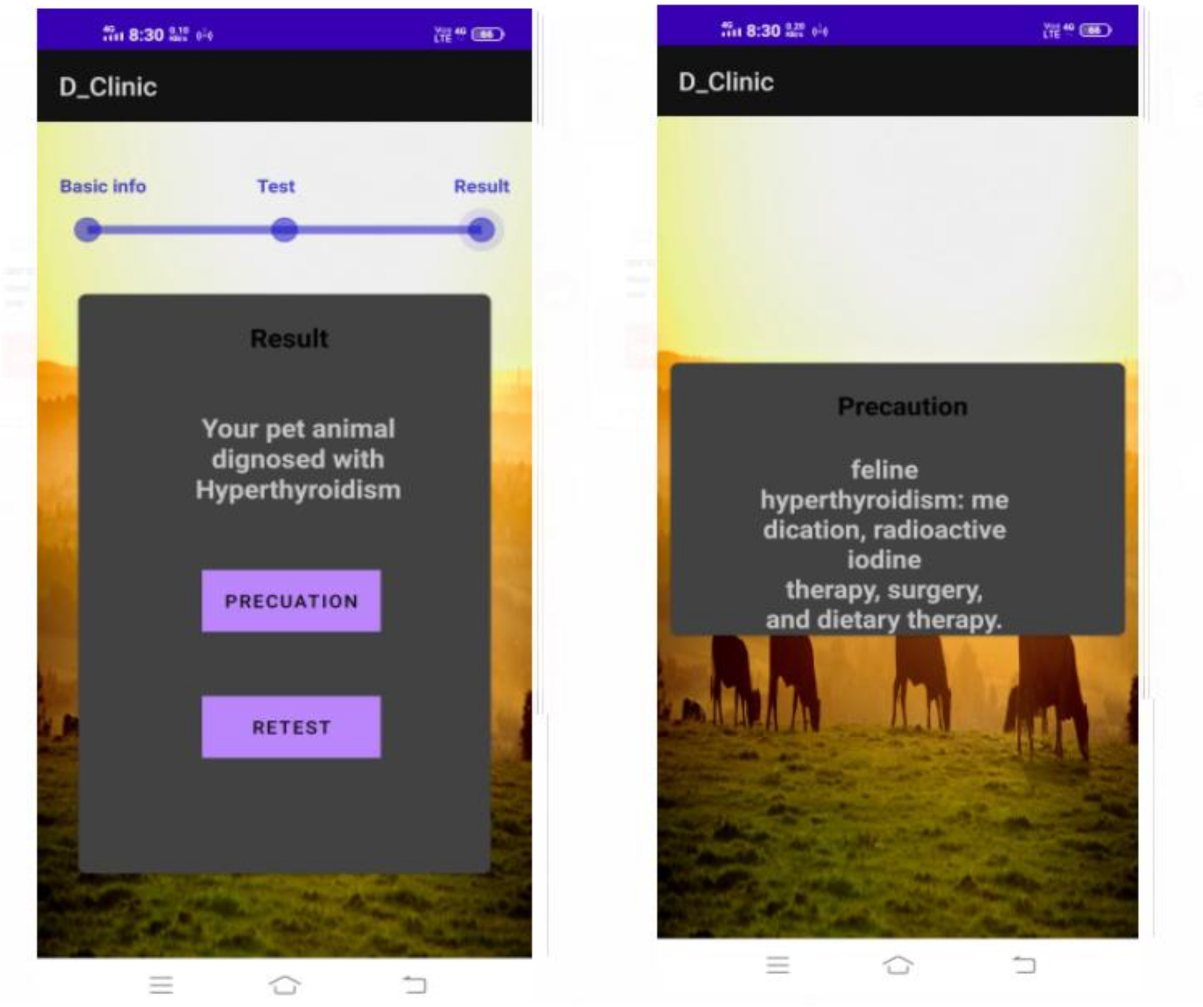
Enter animal bread

This is main page of our mobile app



Symtoms

☐ Fever
☐ paralysis
☑ diarrhea
☐ cough
☑ Abortion
☐ swollowing
☑ Salivation
☑ vomiting
☑ Loss of Weight
☐ cold
☑ respiration
☑ joint and muscle pain



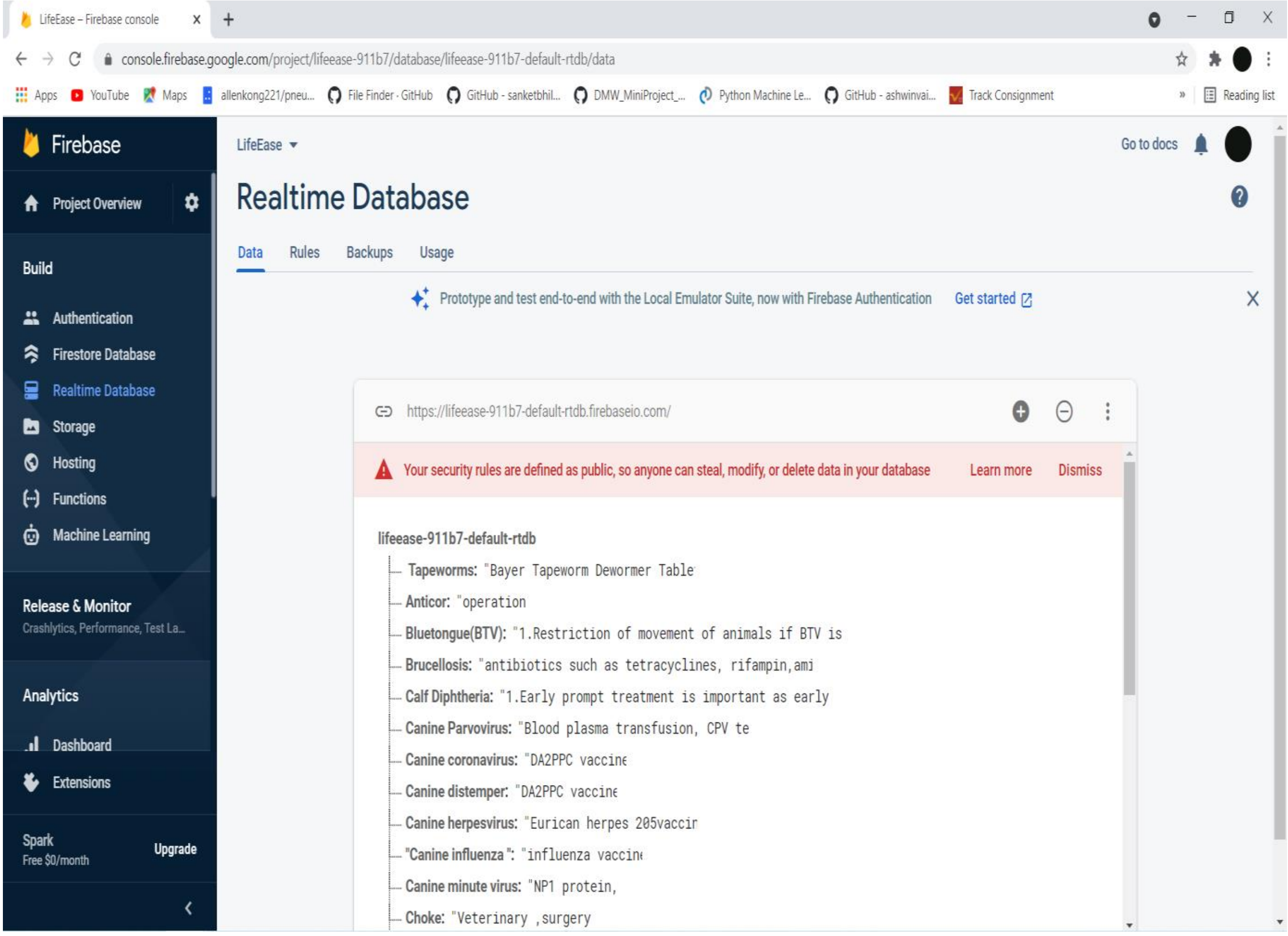
Result

Your pet animal dignosed with Hyperthyroidism

Precaution

feline hyperthyroidism: me dication, radioactive iodine therapy, surgery, and dietary therapy.

Predicted Disease and its precautions



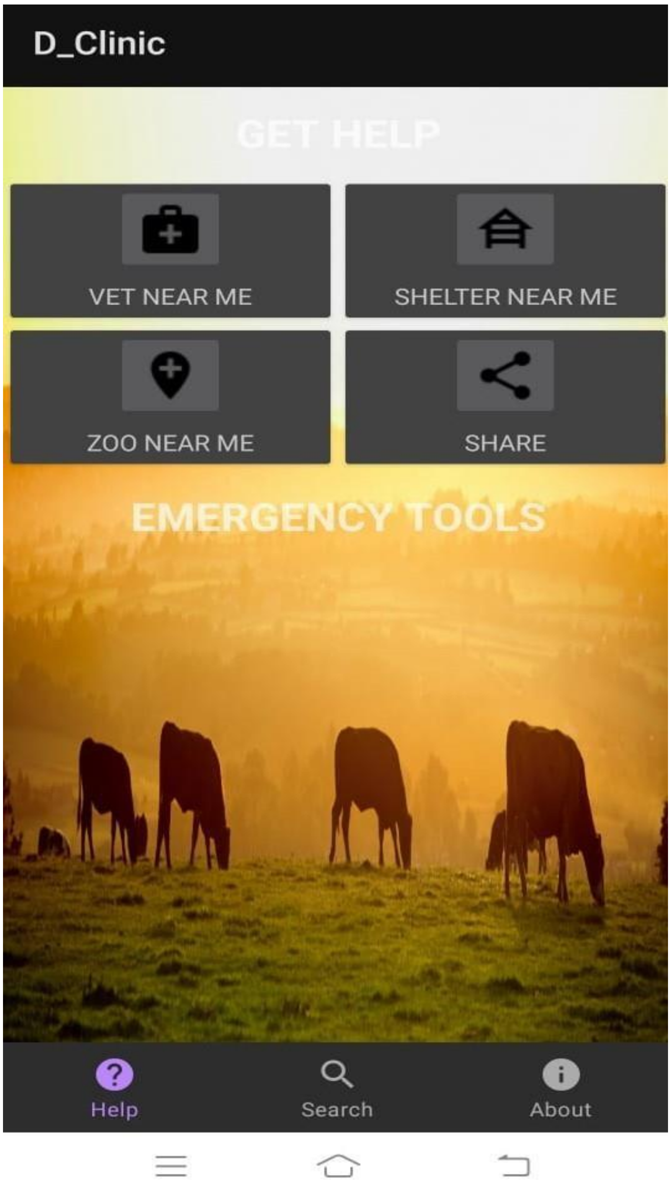
Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication

Your security rules are defined as public, so anyone can steal, modify, or delete data in your database

```
lifeease-911b7-default-rtdb
  Tapeworms: "Bayer Tapeworm Dewormer Table
  Anticor: "operation
  Bluetongue(BTV): "1.Restriction of movement of animals if BTV is
  Brucellosis: "antibiotics such as tetracyclines, rifampin,ami
  Calf Diphtheria: "1.Early prompt treatment is important as early
  Canine Parvovirus: "Blood plasma transfusion, CPV te
  Canine coronavirus: "DA2PPC vaccine
  Canine distemper: "DA2PPC vaccine
  Canine herpesvirus: "Eurican herpes 205vaccir
  "Canine influenza ": "influenza vaccine
  Canine minute virus: "NP1 protein,
  Choke: "Veterinary ,surgery
```

Dataset on server side(firebase Console)

Help options:-



Chapter 7
Conclusion

- Proposed, is a Random Forest algorithm for predicting the disease Animals using the attributes collected from the user.
- Proper adaptation of Algorithm for detecting the Diseases ,providing their Information , providing antibiotics Information and Providing information about nearby doctors Implemented efficiently.
- Created easy to use Application for effective use.

References

[1]. Learning to recommend descriptive tags for Health Seekers using Deep Learning Ms. Vidhi L. Chawda VishwanathS. Mahalle Systems and Control (ICISC-2017)

[2]. Jang-Jae Lee, Byuong-Ho Song, Tae-Yeun Kim, Dae- Woong Seo, Sang-Hyun Bae1 1375, Seosuk-Dong, Dong- Gu, Kwangju, South Korea { Jang-Jae Lee, Byuong-Ho Song, TaeYeun Kim, Sang-Hyun Bae] A Design and Implementation of U-health Diagnosis System using Expert System and Neural Network

[3]. Dhaval Raval1, Dvijesh Bhatt2, Malaram K Kumhar3, Vishal Parikh4, Daiwat Vyas5 12345Nirma University, Computer Science and Engineering Department, Ahmedabad, India] Medical Diagnosis Using Machine Learning 11 September 2015-March 2016 p pp p. .1 17 77 7-1 18 82 2

[4]. Long Wan, Wenxing Bao Animal Disease Diagnoses Expert System Based on SVM 12 Aug 2014

[5]. Tibor TRNOVSZKY, Patrik KAMENCAY, Richard ORJ ESEK, Miroslav BENCO, Peter SYKORA Animal Recognition System Based on Convolutional Neural Network 2017 | SEPTEMBER