

# **ONLINE PET SHOPPING APP**

A Mini Project Report Submitted by

NEHA U  
(4NM18CS102)

NIDHI RAI  
(4NM18CS103)

**UNDER THE GUIDANCE OF**

Mrs.Shabari Shedthi B  
Assistant Professor  
Department of Computer Science and Engineering

in partial fulfillment of the requirements for the award of the Degree of

Bachelor of Engineering in  
Computer Science & Engineering  
from

Visveshvaraya Technological University, Belgaum



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
N.M.A.M. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution under VTU, Belgaum) (AICTE approved, NBA Accredited, ISO 9001:2008 Certified) NITTE -574 110, Udupi District, KARNATAKA.

**DEC 2021**



**NITTE**  
EDUCATION TRUST

**N.M.A.M. INSTITUTE OF TECHNOLOGY**

(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)

Nitte – 574 110, Karnataka, India

(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC

☎: 08258 - 281039 - 281263, Fax: 08258 - 281265

**Department of Computer Science and Engineering**

B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

**DEPARTMENT OF COMPUTER SCIENCE  
AND ENGINEERING  
CERTIFICATE**

“Online Pet Shopping App”

is a bonafide work carried out  
by

NEHA U(4NM18CS102)

NIDHI RAI(4NM18CS103)

in partial fulfillment of the requirements for the award of Bachelor of  
Engineering Degree in Computer Science and Engineering  
prescribed by Visvesvaraya Technological University,  
Belgaum during the year 2021-  
2022.

It is certified that all corrections/suggestions indicated for Internal  
Assessment have been incorporated in the report.

The Mini project report has been approved as it satisfies the academic  
requirements in respect of the project work prescribed for the Bachelor  
of Engineering Degree.

Signature of Guide

Signature of HOD

## **ACKNOWLEDGEMENT**

We believe that our project will be complete only after we thank the people who have contributed to make this project successful.

First and foremost, our sincere thanks to our beloved principal, **Dr. Niranjan N. Chiplunkar** for giving us an opportunity to carry out our project work at our college and providing us with all the needed facilities.

We express our deep sense of gratitude and indebtedness to our guide **Mrs. Shabari Shedthi B**, Assistant Professor, Department of Computer Science and Engineering, for her inspiring guidance, constant encouragement, support and suggestions for improvement during the course of our project.

We sincerely thank **Dr. Jyothi Shetty**, Head of Department of Computer Science and Engineering, Nitte Mahalinga Adyanthaya Memorial Institute of Technology, Nitte.

We also thank all those who have supported us throughout the entire duration of our project.

Finally, we thank the staff members of the Department of Computer Science and Engineering and all our friends for their honest opinions and suggestions throughout the course of our project.

**NEHA U(4NM18CS102)**  
**NIDHI RAI(4NM18CS103)**

# ***ABSTRACT***

Online shopping is the process where customers directly buy goods or services from a seller in real-time, without an intermediary service, over the Internet. It is a form of electronic commerce. This project is an attempt to provide the advantages of online shopping to customers. This app reduces the manual work of managing the details of the pet. Nowadays the lifestyle of the people is different. People feel uncomfortable and time consuming when going to crowded markets. So, E-Shopping is a boon as it saves a lot of time.

Online Pet Shop Project is an android app to provide customer an android platform to order pets. In this project, we develop 2 types of users. One for customers who order pets, and another is the Admin who enters new available pets into the database. This project manages all information related to pet's availability, its breed, price, and the time required to deliver it to the user's house based on the details entered by the user. The project is built at administrative end and only admin is allowed to add or make changes to the pet data.

# *Table of Contents*

1	Introduction	1
1.1	Problem Statement.....	1
1.2	Motivation .....	1
1.3	Scope of the Project .....	2
1.4	Report Structure.....	2
2	Software Requirements Specification	4
2.1	Hardware Requirements .....	4
2.2	Software Requirements.....	4
2.3	Functional Requirements .....	4
2.4	Non-Functional Requirements.....	5
3	Design Implementation	6
3.1	Design of Application.....	6
3.2	Source .....	7
4	Screen Shots	27
4.1	Running Application.....	27
4.2	Registration Page .....	27
4.2.1	Admin Registration.....	27
4.2.1	User Registration .....	28
4.3	Login Page .....	29
4.3.1	Admin Login.....	29
4.3.1	User Login.....	30
4.4	Add Pet Details Page .....	30
4.5	Categories/Buttons Page.....	30
4.6	Dogs Page .....	31
4.7	Cats Page .....	31
4.8	Birds Page.....	32
4.9	Rabbits Page .....	32
4.10	Details Page .....	33

4.11 Add to Cart Page.....	34
4.12 View Cart Page.....	35
4.13 Order Now Page .....	36
4.14 View Orders Page.....	37
4.15 Notification Screen.....	38
 5 References	 39

# ***INTRODUCTION***

This chapter gives an overview of the dissertation work entitled “Online Pet Shop”. The problem statement, motivation for the project and the scope of the project are also explained in this chapter.

Online pet shopping is a form of electronic commerce which allows consumers to directly buy pets from a seller over the Internet using a web browser or a mobile app. Consumers find a pet of interest by visiting the website of the retailer directly or by searching among alternative vendors using a shopping search engine, which displays the same product's availability and pricing at different e-retailers. As of 2020, customers can shop online using a range of different computers and devices, including desktop computers, laptops, tablet computers and smartphones.

E-Shopping is a boon as it saves a lot of time. Shoppers can visit web stores from the comfort of their house and shop as by sitting in front of the computer. Online stores are usually available 24 hours a day and many consumers have internet access both at work and at home. So it is very convenient for them to shop Online. One of the most enticing factors about online shopping, it alleviates the need to wait in long lines or search from a store for a particular item. Variety of pets are available online.

## **Problem Statement:**

The main task in the proposed work is to provide users a platform to buy pets of their choice in online mode without having to spend time visiting the shop manually. This app reduces the manual work of managing the details about the pet.

## **Motivation:**

Nowadays, shopping on the Internet has become an alternative for consumers since it is more comfortable than conventional shopping which is usually attributed with anxiety, crowdedness, traffic jams, limited time, parking space, etc. Apart from this, most of the network commodities' price is cheaper than the same product in store. Hence, online shopping is becoming one of the most popular shopping styles.

## **Scope of the Project:**

Online shopping is an emerging concept in the study area. E-Commerce has grown in importance because of all the advantages it offers to sellers and buyers. It is especially relevant in the current scenario of the Covid-19 pandemic when there are nationwide lockdowns. Purchasing and selling products and services over the internet without the need of going physically to the market is what online

shopping is all about. Online shopping is just like retail store shopping that we do by going to the market, but it is done through the internet. Online shopping has made shopping painless and added more fun.

### **Report Structure:**

This report is structured as follows:

- Chapter 1 presents an overview of the basic features available in the projects.  
Section 1.1 provides a brief idea of the project that was proposed.  
Section 1.2 provides the basic idea about the need of online shopping.  
Section 1.3 depicts the advantages of using online shopping and the scope of the project.
- Chapter 2 explains the methodology that was used in this work to arrive at the final results.  
Section 2.1 specifies the hardware requirements to run the application.  
Section 2.2 specifies the software requirements to build the application.  
Section 2.3 specifies the functional requirements that are implemented in the application.  
Section 2.4 specifies the non-functional requirements of the application.
- Chapter 3 presents the experimental results that were achieved. Section 3.1 depicts the steps involved in designing the application. Section 3.2 depicts the implementation details of the application.
- Chapter 4 presents the flow of the work carried out with the help of screen shots.  
Section 4.1 depicts running the emulator and application installed in it.  
Section 4.2 shows the user and admin registration.  
Section 4.3 shows the login page for user and the admin.  
Section 4.4 shows the page where the details of pets can be added by the admin.  
Section 4.5 shows the buttons to view orders, cart, and different categories of pets.  
Section 4.6 shows the different types of dogs available.  
Section 4.7 shows the different types of cats available.  
Section 4.8 shows the different types of birds available.  
Section 4.9 shows the different types of rabbits available.  
Section 4.10 is the details page which shows the information of the pets.  
Section 4.11 is the page where the pets can be added to cart.  
Section 4.12 shows the items in the cart.  
Section 4.13 shows the page where the pets can be ordered.



Section 4.14 is the view orders page which shows the orders made by the users.

Section 4.15 shows the notification that sent to the user on the day of delivery.

- Chapter 5 shows the reference links that are used for building the application.

# ***Software Requirements Specification***

In this chapter the hardware and software requirements for developing the application are discussed. This chapter also provides an overview of the functional and nonfunctional requirements of the application.

## **Hardware Requirements**

- Processor - Intel(R) Core(TM) i3-6006U CPU @ 2.00GHz 2.00 GHz (or above)
- RAM - 8.00 GB (or above)
- Operating System - Windows 10 (or above)

## **Software Requirements**

- Software used - Android studio
- Front End – XML
- Back End - Java

## **Functional Requirements**

- Registration  
If a customer wants to buy the product then he/she must be registered with the app. Un-registered users cannot surf the products.
- Login  
Customer logs in to the system by entering valid user id and password.
- View Product  
Customers can surf products after logging into the app.
- Adding to cart  
Customers can add the product of their choice to the cart
- Order Product  
Customers can proceed to buy the product of their choice.
- View Order  
Customers can view their orders.
- Cancel Order  
Customers have the option to cancel the order before delivery.
- View Cart  
Customers can view the products in their cart and they can also remove the products from cart.

- Admin Login  
Admin can login just by providing the password used at the time of registration.
- Add Pets  
Admin can add new pets to the application.

### **Non-Functional Requirements**

- 24X7 availability
- Better component design to get better performance at peak time.
- Flexible service based architecture will be highly desirable for future extension.

# *Design Implementation*

This chapter illustrates the application design with the help of relevant diagrams. The activity and layout files are shown here and their working is explained.

## **Design of Application**

Step 1: Users must register themselves to use the application.

Step 2: User logs in to the application using the registered username and password.

- Admin login: Access to add or modify pet data.
- Customer login: Access to place orders and buy pets.

Step 3: Customers can view list of product catalogs(pets) available for purchase.

Step 4: Customers can add selected pets into cart, which they would like to buy later.

Step 5: Customers can place their order from the cart or directly from the product catalog page.

Step 6: Customers can view history of all their orders made.

Step 7: Customers can cancel their order before delivery.

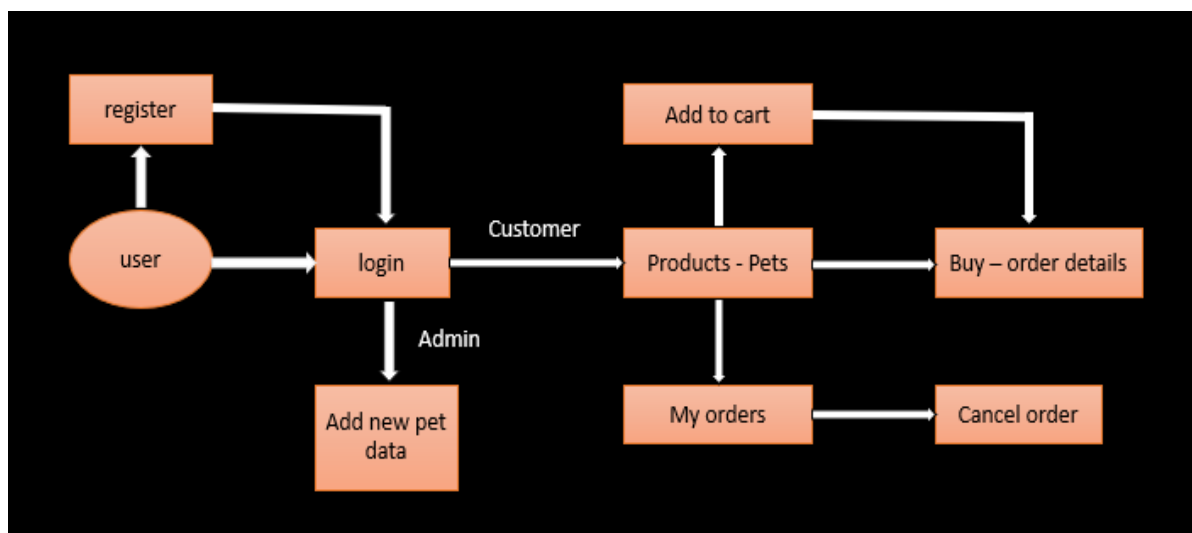


Figure 3.1: Process Flow Diagram

## **Source:**

### **MainActivity.java**

```
package com.example.petshop;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;

import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class MainActivity extends AppCompatActivity {
    Button regsignup,regsignin,admin;
    EditText reguser,regpwd;
    DatabaseHelper mydb;
    boolean status;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mydb=new DatabaseHelper(this);
        regsignup=(Button)findViewById(R.id.regsignup);
        regsignin=(Button)findViewById(R.id.regsignin);
        reguser=(EditText)findViewById(R.id.reguser);
```

```

regpwd=(EditText)findViewById(R.id.regpwd);
admin=(Button)findViewById(R.id.admin);
regsignup.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String username, password;
        username=reguser.getText().toString();
        password=regpwd.getText().toString();
        if(username.equals("") || password.equals(""))
        {
            Toast.makeText(getApplicationContext(), "Please enter all the
details!!", Toast.LENGTH_SHORT).show();
        }
        else if(mydb.searchUser(username.trim()))
        {
            Toast.makeText(MainActivity.this, "Username Already Taken",
Toast.LENGTH_SHORT).show();
        }
        else if (!isValidPassword(password.trim()))
        {
            Toast.makeText(MainActivity.this, "Enter Valid Password with
combination of capital letters, small letters, digits and special characters",
Toast.LENGTH_LONG).show();
        }
        else{
            status = mydb.insertUser(username.trim(), password.trim());
            if (status) {
                Toast.makeText(MainActivity.this, "Registration Successfull",
Toast.LENGTH_SHORT).show();
                reguser.setText("");
                regpwd.setText("");
                Intent intent = new Intent(MainActivity.this, Login.class);
                startActivity(intent);
            } else {
                Toast.makeText(MainActivity.this, "Registration Failed",
Toast.LENGTH_SHORT).show();
            }
        }
    }
});
regsignin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

```

```

        Intent intent = new Intent(MainActivity.this, Login.class);
        startActivity(intent);
    }
});
admin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this, Admin.class);
        startActivity(intent);
    }
});
}
public boolean isValidPassword(final String Password){
    Pattern pattern;
    Matcher matcher;
    String PASSWORD_PATTERN="^(?=.*[0-9])(?=.*[A-Z])(?=.*[a-z])(?=.*[@#$%^&*+=])(?=\\S+$).{8,}$";
    pattern=Pattern.compile(PASSWORD_PATTERN);
    matcher=pattern.matcher(Password);
    return matcher.matches();
}
}

```

This is the Registration activity where certain validations are made for the customer to register himself with the app.

Validations:

- Username and Password cannot be NULL.
- Username of all the customers should be UNIQUE.
- Password should be a combination of Capital letters, small letters, digits and special characters with a minimum of 8 digits making it a **STRONG PASSWORD**.

If all the validations are satisfied, the user details are successfully stored into the database.

Users can also navigate directly to the Login page if he is already a registered user and also to the Admin Login page if he is a ADMIN.

## **Login.java**

```
package com.example.petshop;
```

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
```

```
public class Login extends AppCompatActivity {
    Button login, signup;
    EditText user, pwd;
    DatabaseHelper mydb;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        login=(Button)findViewById(R.id.login);
        signup=(Button)findViewById(R.id.signup);
        user=(EditText)findViewById(R.id.user);
        pwd=(EditText)findViewById(R.id.pwd);
        mydb=new DatabaseHelper(this);
        login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String Currentuser=user.getText().toString().trim();
                String Currentpassword=pwd.getText().toString().trim();
```



```

        if(mydb.CheckLoginStatus(Currentuser,Currentpassword))
        {
            user.setText("");
            pwd.setText("");
            Bundle bundle = new Bundle();
            bundle.putString("username",Currentuser);
            Toast.makeText(Login.this, "Login Successfull",
Toast.LENGTH_SHORT).show();
            Intent intent = new Intent(Login.this, Buttons.class);
            intent.putExtras(bundle);
            startActivity(intent);
        }
        else
        {
            Toast.makeText(Login.this, "Invalid username or password",
Toast.LENGTH_SHORT).show();
        }
    }
});
signup.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(Login.this, MainActivity.class);
        startActivity(intent);
    }
});
}
}
}

```

This is the User Login activity where a user is allowed to login only in case if he is a registered user i.e, he/she is directly navigated to the Buttons page, if his details are found in the database. He/she can also navigate directly to the registration page if they are not already a registered user.

### **Buttons.java**

```

package com.example.petshop;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

```

```

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class Buttons extends AppCompatActivity {
    Button dogs,cats,birds,rabbits,orders,car;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_buttons);
        String username;
        dogs=(Button)findViewById(R.id.dogs);
        cats=(Button)findViewById(R.id.cats);
        birds=(Button)findViewById(R.id.birds);
        rabbits=(Button)findViewById(R.id.rabbits);
    }
}

```

```

orders=(Button)findViewById(R.id.orders);
cart=(Button)findViewById(R.id.cart);
Bundle bundle1=getIntent().getExtras();
username=bundle1.getString("username");
dogs.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Bundle bundle = new Bundle();
        bundle.putString("category","Dogs");
        bundle.putString("username",username);
        Intent intent = new Intent(Buttons.this, Home.class);
        intent.putExtras(bundle);
        startActivity(intent);
    }
});
cats.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Bundle bundle = new Bundle();
        bundle.putString("category","Cats");
        bundle.putString("username",username);
        Intent intent = new Intent(Buttons.this, Home.class);
        intent.putExtras(bundle);
        startActivity(intent);
    }
});
birds.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Bundle bundle = new Bundle();
        bundle.putString("category","Birds");
        bundle.putString("username",username);
        Intent intent = new Intent(Buttons.this, Home.class);
        intent.putExtras(bundle);
        startActivity(intent);
    }
});
rabbits.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Bundle bundle = new Bundle();
        bundle.putString("category","Rabbits");
        bundle.putString("username",username);

```

```

        Intent intent = new Intent(Buttons.this, Home.class);
        intent.putExtras(bundle);
        startActivity(intent);
    }
});
orders.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Bundle bundle = new Bundle();
        bundle.putString("username",username);
        Intent intent = new Intent(Buttons.this, Orders.class);
        intent.putExtras(bundle);
        startActivity(intent);
    }
});
cart.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Bundle bundle = new Bundle();
        bundle.putString("username",username);
        Intent intent = new Intent(Buttons.this, MyCart.class);
        intent.putExtras(bundle);
        startActivity(intent);
    }
});
}
}
}

```

This is the landing page, once the user successfully logins. Users can select one of the four categories i.e, Dogs, Cats, Birds and Rabbits which takes them to the Home page. The category name and username is passed through Intent Bundle to the Home page. There are two more buttons which take them to their orders page and my carts page where they can view the list of orders made by them and list of pets in their cart respectively.

### **Home.java**

```

package com.example.petshop;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import androidx.recyclerview.widget.LinearLayoutManager;
import android.database.Cursor;

```

```
import com.example.petshop.Adapters.MainAdapter;
import com.example.petshop.Models.MainModel;
import com.example.petshop.databinding.ActivityHomeBinding;
import java.util.ArrayList;
```

```
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
```

```
import android.database.Cursor;
import android.os.Bundle;
```

```
import com.example.petshop.Adapters.MainAdapter;
import com.example.petshop.Models.MainModel;
import com.example.petshop.databinding.ActivityHomeBinding;
```

```
import java.util.ArrayList;
```

```
public class Home extends AppCompatActivity {
    ActivityHomeBinding binding;
    DatabaseHelper mydb;
    Cursor cursor;
    String user;
    ArrayList<MainModel> list= new ArrayList<>();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mydb=new DatabaseHelper(this);
        String category,username;

        binding=ActivityHomeBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        Bundle bundle1=getIntent().getExtras();
        if(bundle1!=null){
            category=bundle1.getString("category");
            username=bundle1.getString("username");
            Cursor cursor=mydb.fetchData(category);
            fetch(cursor,username);
        }

        MainAdapter adapter=new MainAdapter(list,this);
        binding.recyclerview.setAdapter(adapter);
    }
}
```

```

        LinearLayoutManager layoutManager=new LinearLayoutManager(this);
        binding.recycleview.setLayoutManager(layoutManager);
    }
    public void fetch(Cursor cursor,String username)
    {
        cursor.moveToFirst();
        while (cursor.isAfterLast()==false)
        {
            String image = cursor.getString(0);
            String name = cursor.getString(1);
            String price = cursor.getString(2);
            String desc = cursor.getString(3);
            int
i=getResources().getIdentifier(image,"drawable",getPackageName());
            list.add(new MainModel(i,name,price,desc,username));
            cursor.moveToNext();
        }
    }
}

```

This is the Home page which displays a list of pets which is fetched from the database based on the category being passed in the Intent Bundle. This activity is binded with the activity\_home xml file (ActivityHomeBinding).

### **MainModel.java**

```
package com.example.petshop.Models;
```

```

public class MainModel {
    String name,price,desc,username;
    int image;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getPrice() {
        return price;
    }
}

```

```

    public void setPrice(String price) {
        this.price = price;
    }

    public String getDesc() {
        return desc;
    }

    public void setDesc(String desc) {
        this.desc = desc;
    }

    public int getImage() {
        return image;
    }

    public void setImage(int image) {
        this.image = image;
    }

    public String getUsername() {
        return username;
    }

    public MainModel(int image, String name, String price, String desc, String
username) {
        this.image = image;
        this.name = name;
        this.price = price;
        this.desc = desc;
        this.username= username;
    }
}

```

This is the model file to support listview in the Home page. Constructor and getter setter methods are implemented here.

### **MainAdapter.java**

```

package com.example.petshop.Adapters;

import android.content.Context;
import android.content.Intent;

```

```

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import com.example.petshop.Buttons;
import com.example.petshop.Detail;
import com.example.petshop.Home;
import com.example.petshop.Models.MainModel;
import com.example.petshop.R;
import java.util.ArrayList;

public class MainAdapter extends
RecyclerView.Adapter<MainAdapter.viewholder> {
    ArrayList<MainModel> list;
    Context context;

    public MainAdapter(ArrayList<MainModel> list, Context context) {
        this.list = list;
        this.context = context;
    }

    @NonNull
    @Override
    public MainAdapter.viewholder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
        View view= LayoutInflater.from(context).inflate(R.layout.sample, parent,
false);
        return new viewholder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull MainAdapter.viewholder holder,
int position) {
        final MainModel model=list.get(position);
        holder.image.setImageResource(model.getImage());
        holder.name.setText(model.getName());
        holder.price.setText(model.getPrice());
        holder.desc.setText(model.getDesc());
        holder.itemView.setOnClickListener(new View.OnClickListener() {

```



```

@Override
public void onClick(View v) {
    Intent intent = new Intent(context, Detail.class);
    intent.putExtra("image",model.getImage());
    intent.putExtra("name",model.getName());
    intent.putExtra("price",model.getPrice());
    intent.putExtra("desc",model.getDesc());
    intent.putExtra("username",model.getUsername());
    context.startActivity(intent);
}
});
}

@Override
public int getItemCount() {
    return list.size();
}

public class viewholder extends RecyclerView.ViewHolder {
    ImageView image;
    TextView name, price, desc;

    public viewholder(@NonNull View itemView) {
        super(itemView);
        image = itemView.findViewById(R.id.image);
        name = itemView.findViewById(R.id.name);
        price = itemView.findViewById(R.id.price);
        desc = itemView.findViewById(R.id.desc);
    }
}
}

```

## **Detail.java**

```

package com.example.petshop;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

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

```

```

import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import java.util.Locale;

public class Detail extends AppCompatActivity {
    ImageView image;
    TextView name,price,desc;
    Button cart,speak,order;
    TextToSpeech textToSpeech;
    DatabaseHelper mydb;
    Login login;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail);

        image=(ImageView)findViewById(R.id.image);
        name=(TextView)findViewById(R.id.name);
        price=(TextView)findViewById(R.id.price);
        desc=(TextView)findViewById(R.id.desc);
        speak=(Button)findViewById(R.id.speak);
        cart=(Button)findViewById(R.id.cart);
        order=(Button)findViewById(R.id.order);
        mydb=new DatabaseHelper(this);

        int dimage=getIntent().getIntExtra("image",0);
        String dname=getIntent().getStringExtra("name");
        String dprice=getIntent().getStringExtra("price");
        String ddesc=getIntent().getStringExtra("desc");
        String username=getIntent().getStringExtra("username");
        String imagename=getResources().getResourceName(dimage);
        String[] images=imagename.split("/");

        image.setImageResource(dimage);
        name.setText(dname);

```

```

price.setText(dprice);
desc.setText(ddesc);

textToSpeech=new TextToSpeech(Detail.this, new
TextToSpeech.OnInitListener() {
    @Override
    public void onInit(int status) {
        if(status != TextToSpeech.ERROR)
        {
            textToSpeech.setLanguage(Locale.UK);
        }
    }
});
speak.setOnClickListener(new View.OnClickListener() {
    @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
    @Override
    public void onClick(View v) {
        String text=name.getText().toString()+" Price
"+price.getText().toString()+" "+desc.getText().toString();
        textToSpeech.speak(text,TextToSpeech.QUEUE_FLUSH,null);
    }
});
cart.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(mydb.searchCart(username,images[1]))
            Toast.makeText(Detail.this, "Added to cart",
Toast.LENGTH_SHORT).show();
        else
            Toast.makeText(Detail.this, "Already in cart",
Toast.LENGTH_SHORT).show();
    }
});
order.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Bundle bundle = new Bundle();
        bundle.putString("username",username);
        bundle.putString("image",images[1]);
        Intent intent = new Intent(Detail.this, PlaceOrder.class);
        intent.putExtras(bundle);
        startActivity(intent);
    }
}

```

```

        });

    }
}

```

This activity shows details of the pet selected from the list in the home page. User can add this product to his cart which in turn stores the item details in the database table Cart. User can place his order by clicking the Order Now button, which takes him to the PlaceOrder page to which username and image is passed through Intent Bundle. A text to speech feature is added here to read out the pet details to the user.

### **PlaceOrder.java**

```

package com.example.petshop;

import androidx.appcompat.app.AppCompatActivity;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import android.content.Intent;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;

import java.util.regex.Matcher;
import java.util.regex.Pattern;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;

```

```

import android.database.Cursor;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;

import java.util.regex.Matcher;
import java.util.regex.Pattern;

import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.app.NotificationCompat;

import android.Manifest;
import android.app.AlarmManager;
import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.icu.util.Calendar;
import android.os.Build;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;

import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class PlaceOrder extends AppCompatActivity {

```

```

ImageView image;
EditText address,phone;
Button order;
Boolean status;
DatabaseHelper mydb;
boolean connected=true;
AlarmManager alarmManager;
public static final String NOTIFICATION_CHANNEL_ID = "10001" ;
private final static String default_notification_channel_id = "default" ;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_place_order);

    image=(ImageView)findViewById(R.id.image);
    address=(EditText)findViewById(R.id.address);
    phone=(EditText)findViewById(R.id.phone);
    order=(Button)findViewById(R.id.order);
    mydb=new DatabaseHelper(this);

    Bundle bundle1=getIntent().getExtras();
    String username=bundle1.getString("username");
    String images=bundle1.getString("image");
    int id=getResources().getIdentifier(images,"drawable",getPackageName());
    image.setImageResource(id);

    order.setOnClickListener(new View.OnClickListener() {
        @RequiresApi(api = Build.VERSION_CODES.N)
        @Override
        public void onClick(View v) {
            String orderaddress=address.getText().toString().trim();
            String orderphone=phone.getText().toString().trim();
            if(orderaddress.equals("") || orderphone.equals(""))
            {
                Toast.makeText(getApplicationContext(), "Please enter all the
details!!", Toast.LENGTH_SHORT).show();
            }
            else if (!isValidPhone(orderphone))
            {
                Toast.makeText(PlaceOrder.this, "Enter Valid Phone Number",
Toast.LENGTH_LONG).show();
            }
            else {

```

```

        status = mydb.insertOrder(username,
images,orderaddress,orderphone);
        if (status) {
            Toast.makeText(PlaceOrder.this, "Order Placed",
Toast.LENGTH_SHORT).show();
            address.setText("");
            phone.setText("");

            Cursor cursor=mydb.fetchNum();
            cursor.moveToFirst();
            String Orderno= String.valueOf(cursor.getInt(0));
            Cursor cursor1=mydb.fetchDetail(images);
            cursor1.moveToFirst();
            String name=cursor1.getString(1);
            String price=cursor1.getString(2);
            ActivityCompat.requestPermissions(PlaceOrder.this, new
String[]{Manifest.permission.SEND_SMS},
PackageManager.PERMISSION_GRANTED);
            SmsManager mySmsManager = SmsManager.getDefault();
            String text="YOUR ORDER IS
CONFIRMED\nPet:"+name+"\nPrice:"+price+"\nOrder
Number:"+Orderno+"\nDelivery Address:"+orderaddress+"\nTHANK YOU
FOR SHOPPING";

            mySmsManager.sendTextMessage(orderphone,null,text,null,null);

            alarmManager=(AlarmManager)getSystemService(ALARM_SERVICE) ;
            Calendar calendar=Calendar.getInstance();
            calendar.add(Calendar.DATE,2);
            calendar.set(Calendar.HOUR_OF_DAY,9);
            calendar.set(Calendar.MINUTE,0);
            calendar.set(Calendar.SECOND,0);
            //SimpleDateFormat f=new SimpleDateFormat("dd/MM/yyyy
HH:mm:ss");
            //Toast.makeText(PlaceOrder.this, f.format(calendar.getTime()),
Toast.LENGTH_SHORT).show();
            Intent alarmIntent=new Intent(PlaceOrder.this,
AlarmReceiver.class);
            alarmIntent.putExtra("noti",name+" IS OUT FOR DELIVERY
TODAY");
            int id=(int)System.currentTimeMillis();

```

```

        PendingIntent
        pIntent=PendingIntent.getBroadcast(PlaceOrder.this,id,alarmIntent,PendingInte
nt.FLAG_UPDATE_CURRENT);

        alarmManager.set(AlarmManager.RTC_WAKEUP,calendar.getTimeInMillis(),
        pIntent);
    } else {
        Toast.makeText(PlaceOrder.this, "Order Failed",
        Toast.LENGTH_SHORT).show();
    }
}
});
}
public boolean isValidPhone(final String phone)
{
    Pattern pattern;
    Matcher matcher;
    String PASSWORD_PATTERN="^\\d{10}$";
    pattern=Pattern.compile(PASSWORD_PATTERN);
    matcher=pattern.matcher(phone);
    return matcher.matches();
}
}

```

This activity takes address and phone number from the user before placing the order. Both the fields cannot be NULL and phone number must be a 10 digit number. If the order is successful, the details are stored in database table Orders and an SMS is sent to the mentioned phone number with the details of the order. Also notification is set after 2 days, to remind the user of the order.



# SCREENSHOTS

The screenshots in this chapter depict the working of the application.

## 4.1 Running Application

The application to be tested is executed as shown in Figure 4.1. The emulator starts and the application is installed in it. The main activity is brought to the forefront. The user can then start the activity on which the experiments are carried out.

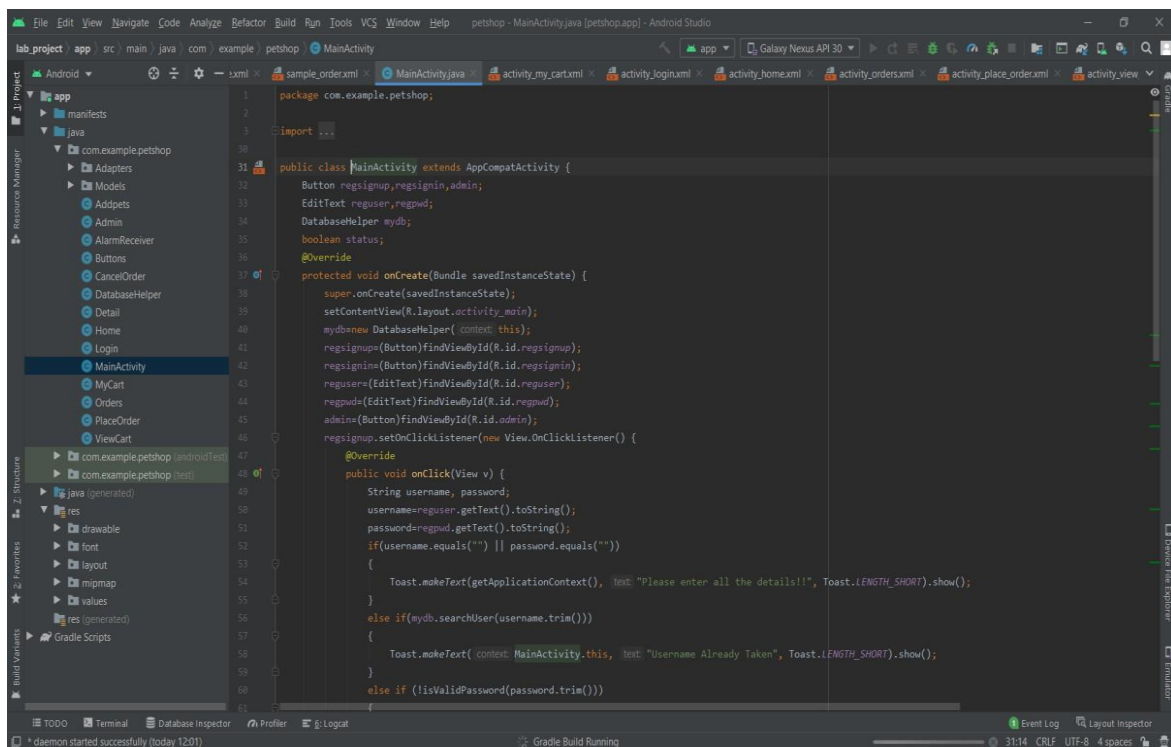
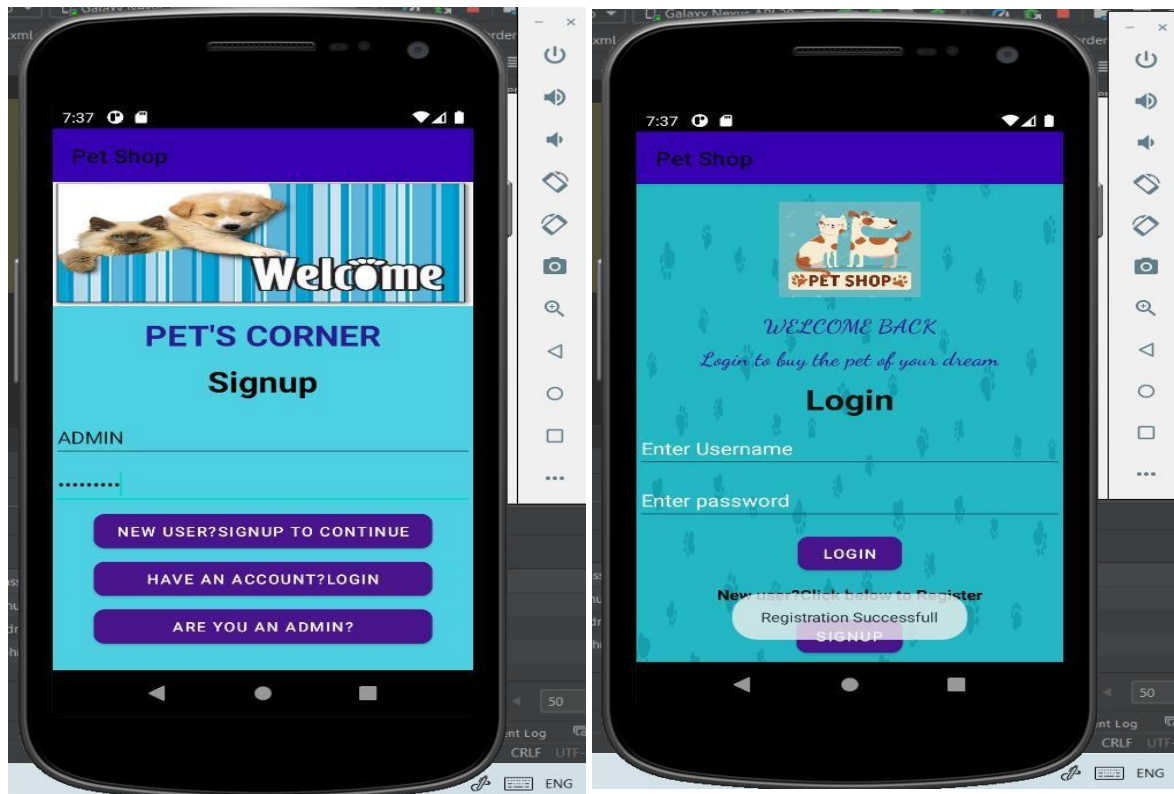


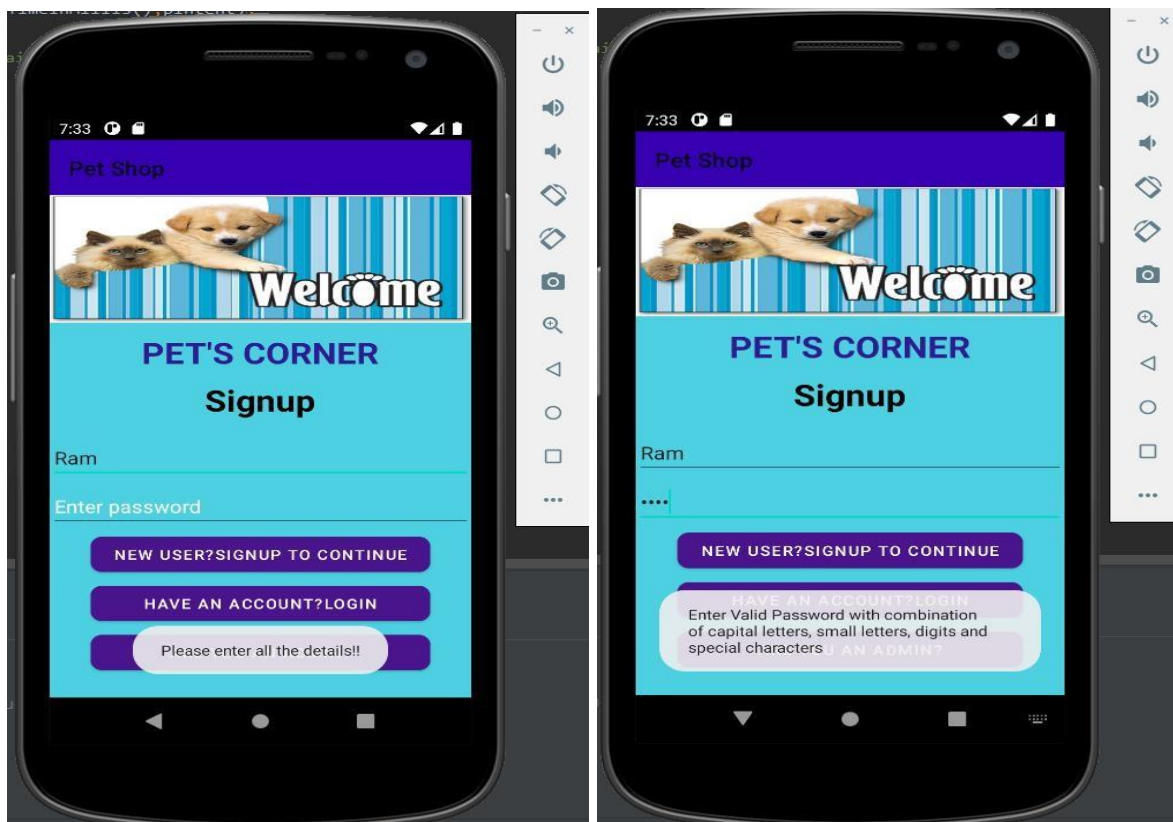
Figure 4.1: Running application code

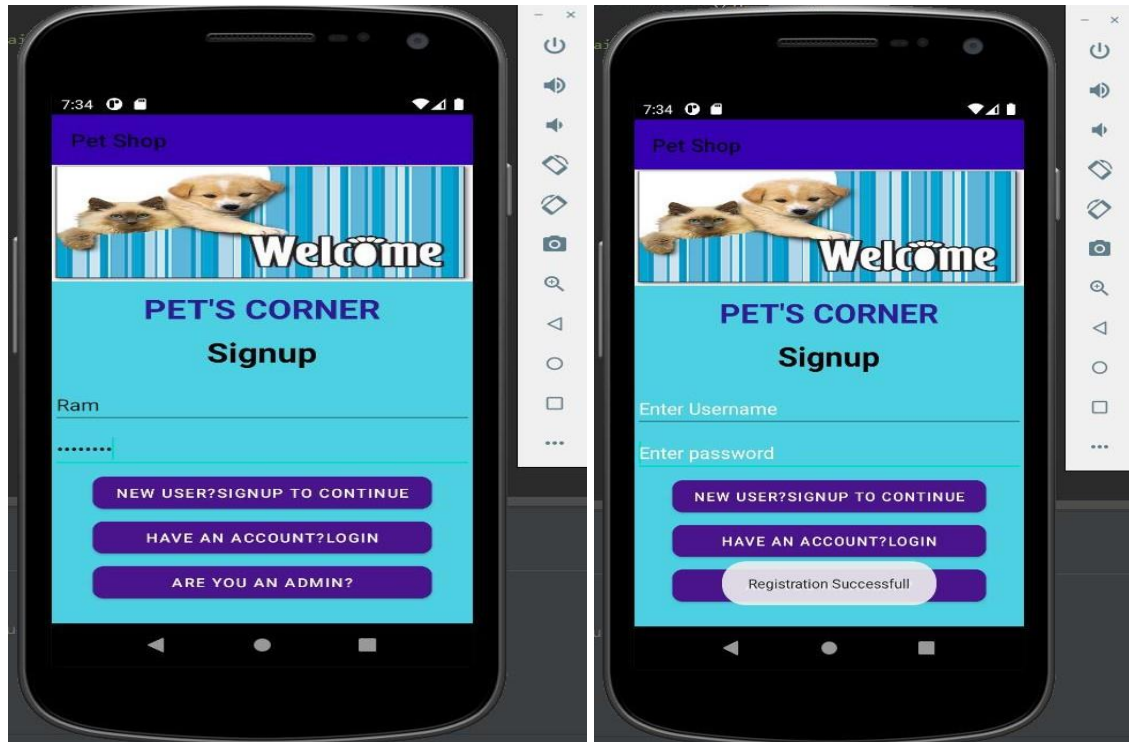
## 4.2 Registration Page

**4.2.1 Admin Registration:** ADMIN registers with the application before launching the application to customers.



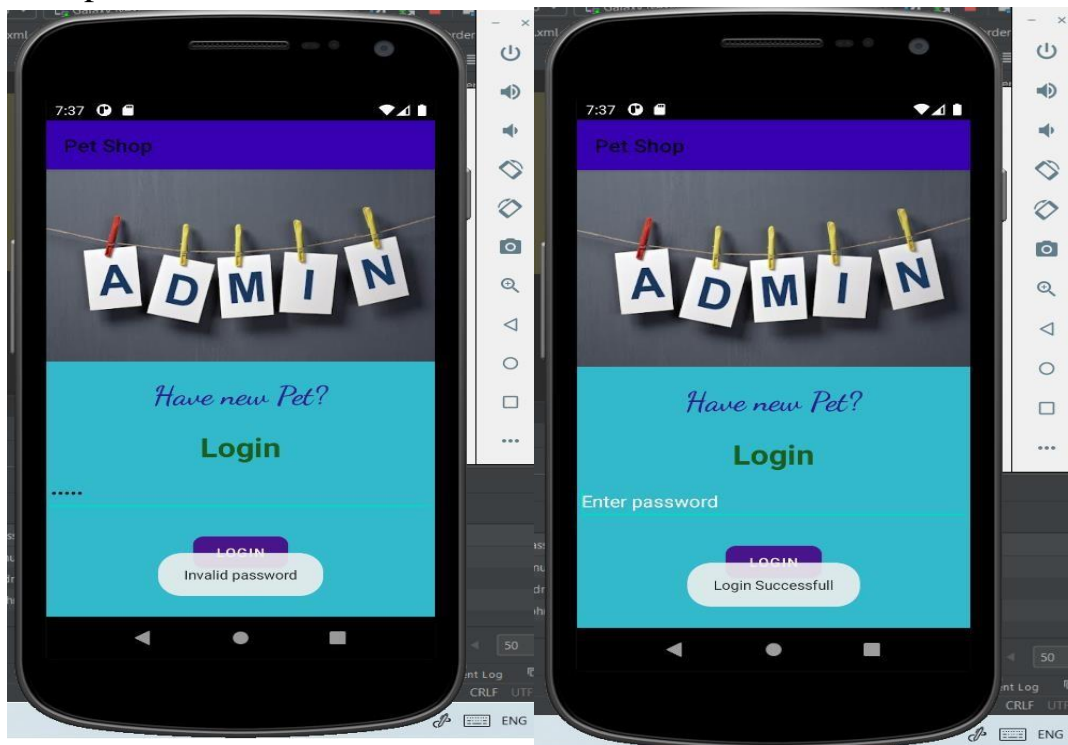
**4.2.2 User Registration:** User registers himself with the application.



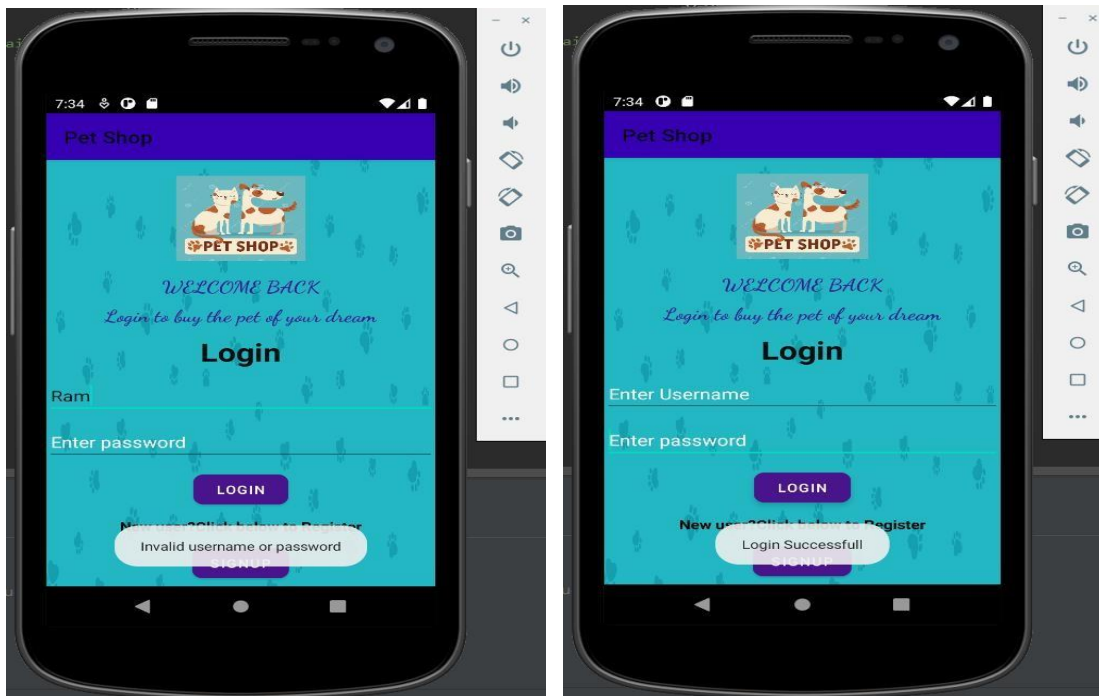


### 4.3 Login Page

**4.3.1 Admin Login:** ADMIN logs to the application by entering the ADMIN password.

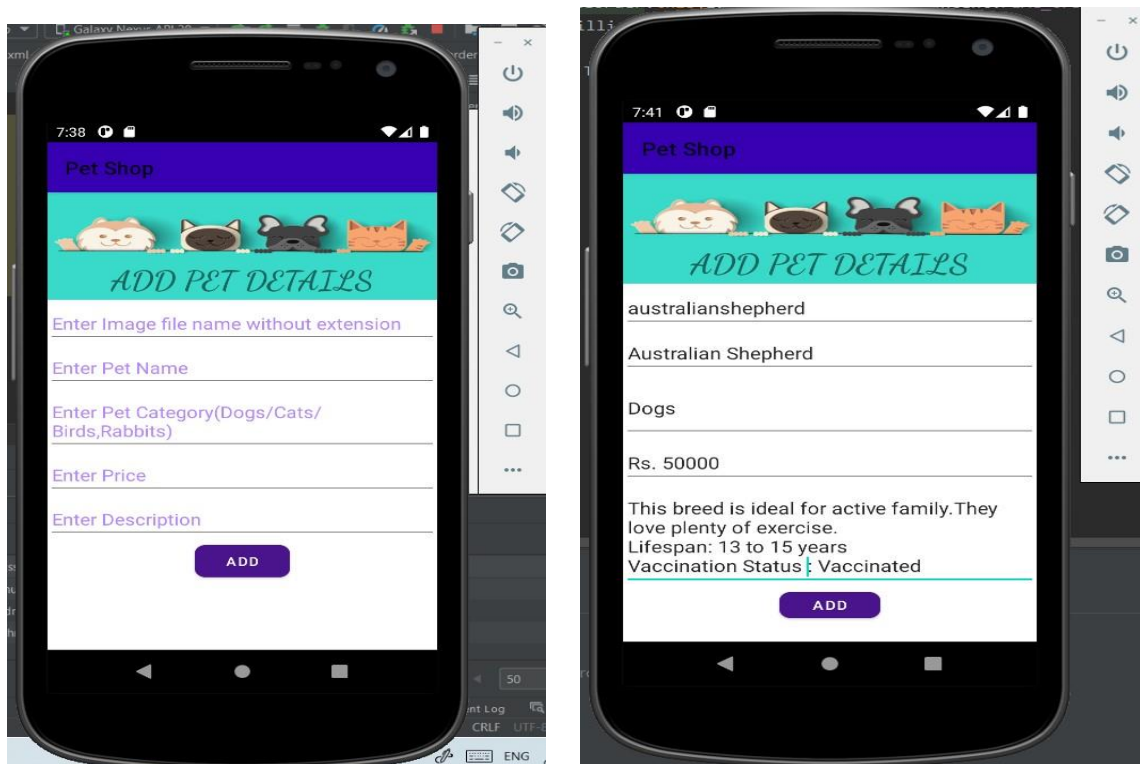


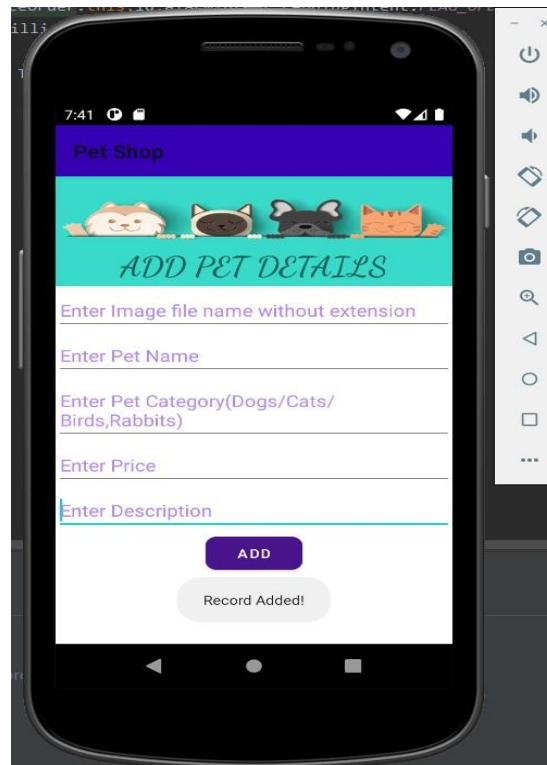
#### 4.3.2 User login: User logs in to the application with the registered username and password.



#### 4.4 Add Pet Details Page:

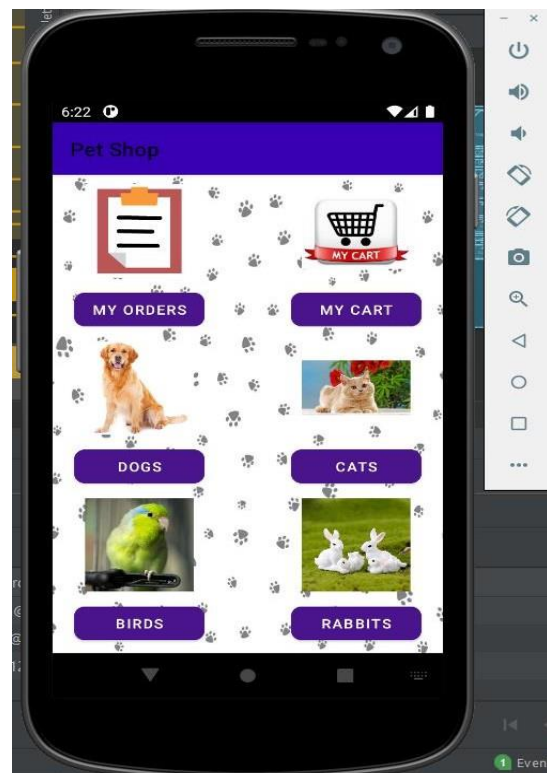
After ADMIN login, he is allowed to enter new available pets to the application.





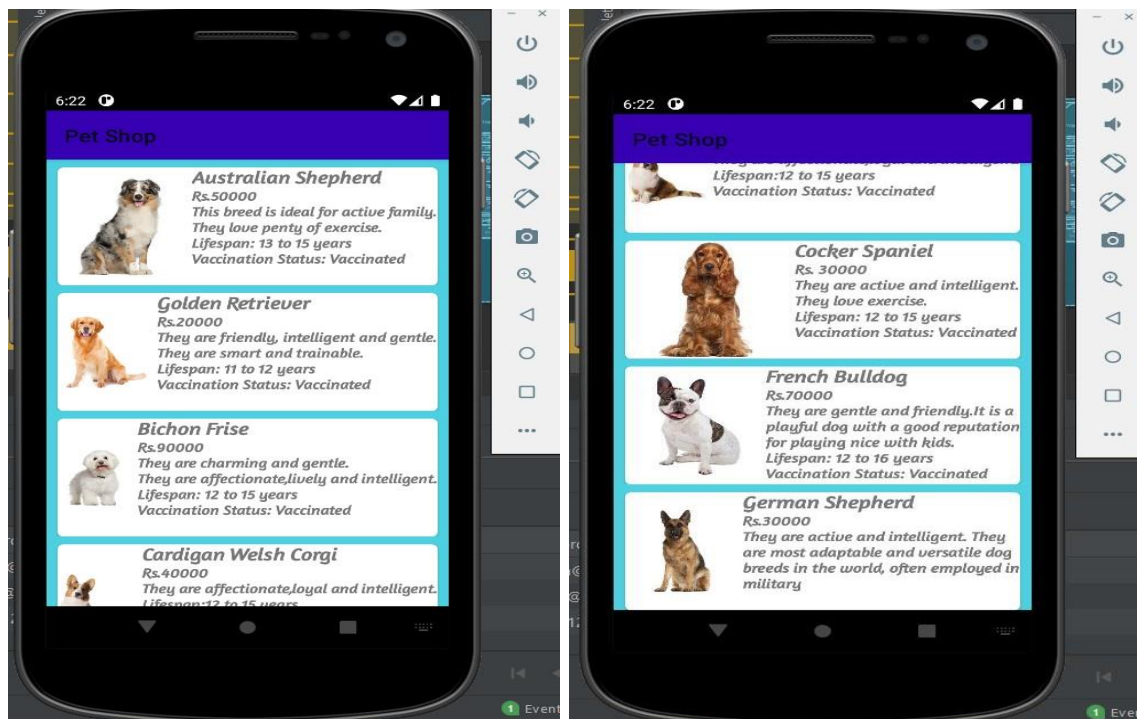
#### **4.5 Categories/Buttons Page:**

Four categories of pets are available, with 2 extrabuttons to navigate to My Orders section and My Carts section.

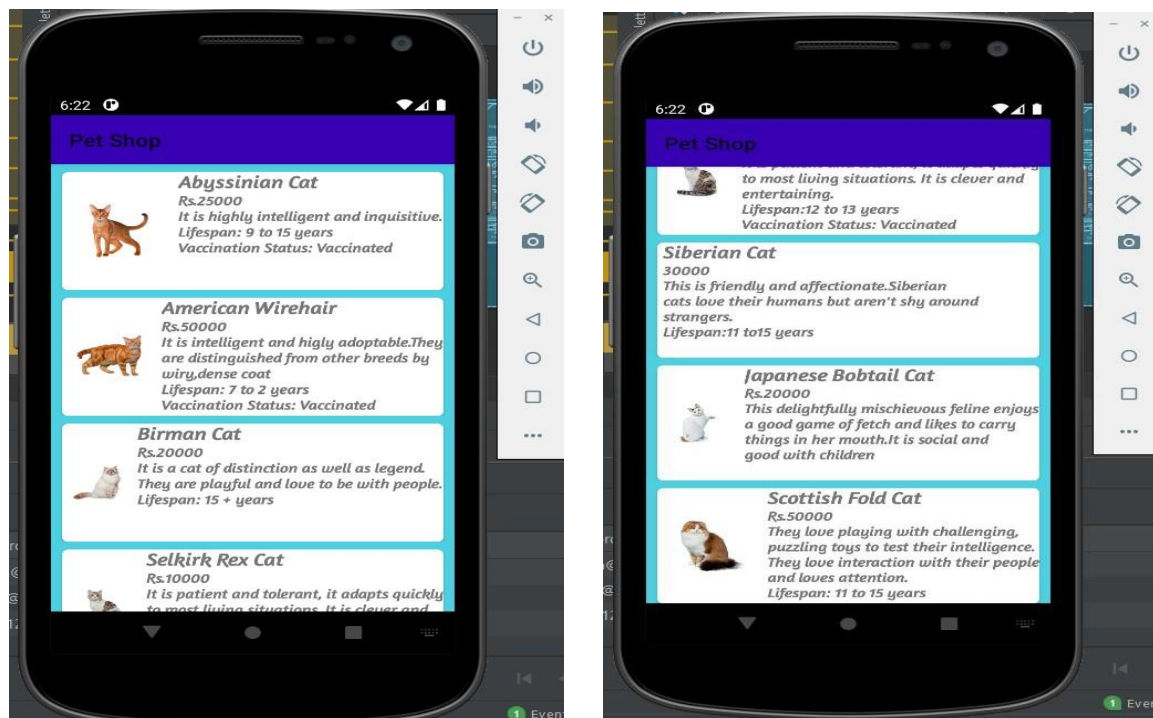




**4.6 Dogs:** When clicked on Dogs category, list of available dogs is displayed in a listview.

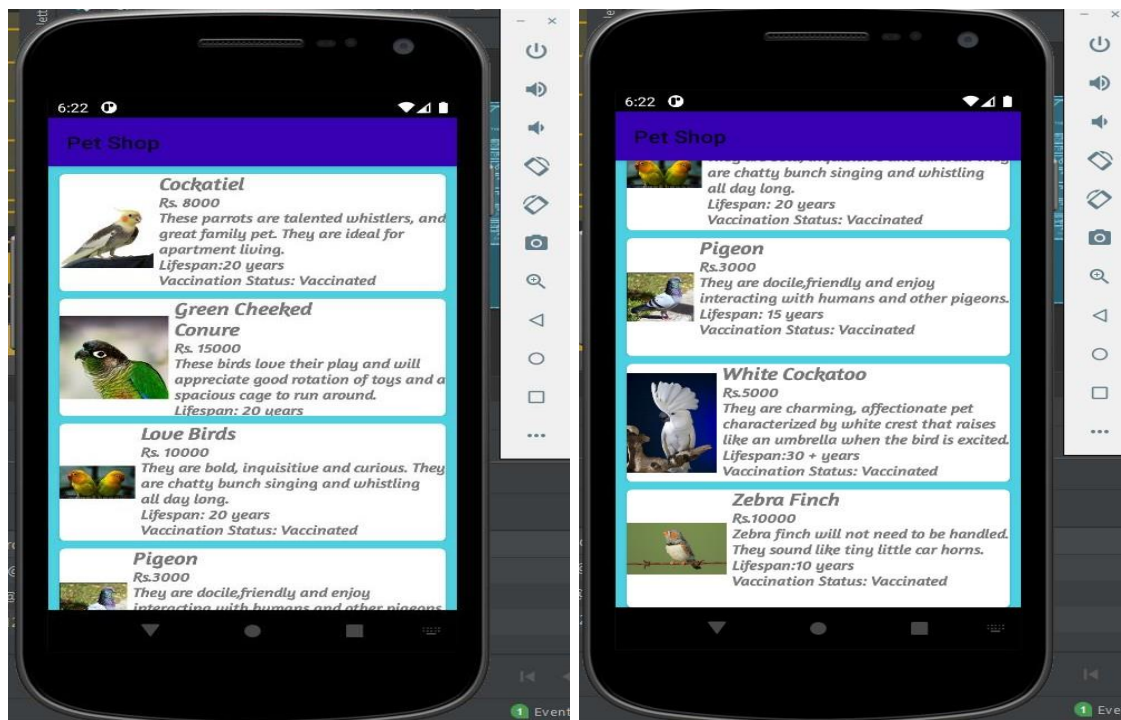


**4.7 Cats:** When clicked on Cats category, list of available cats is displayed

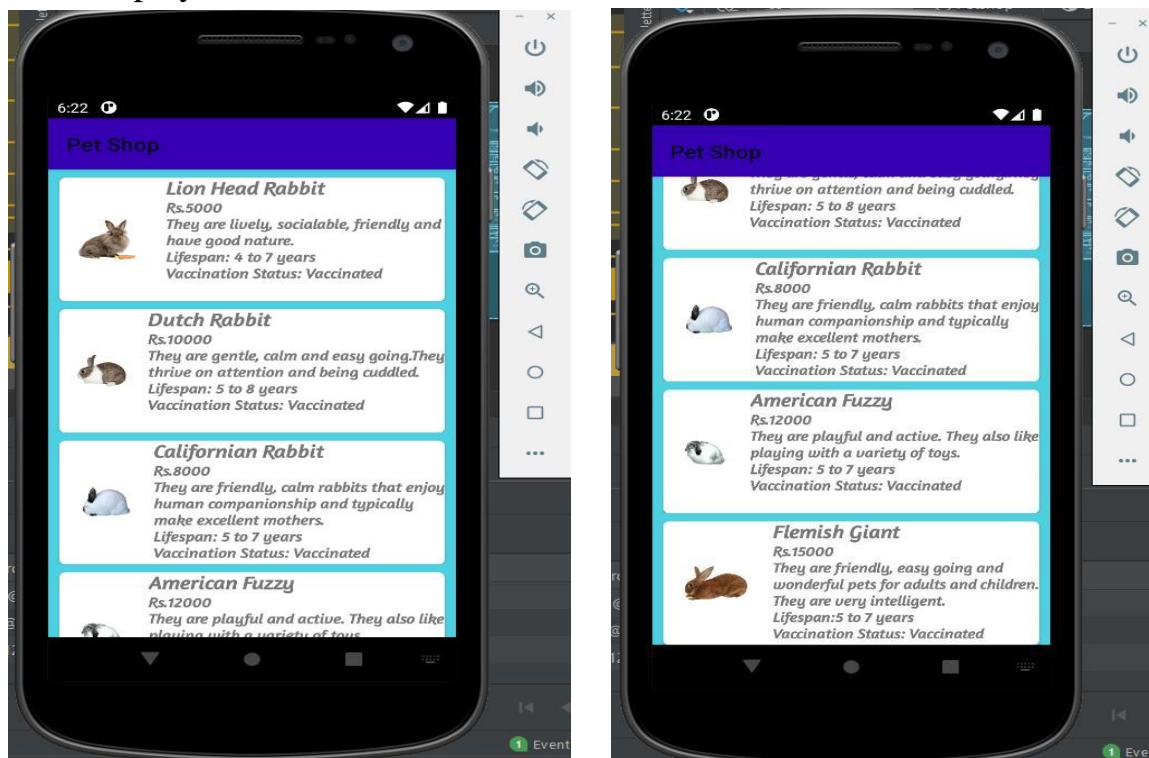


in a list view.

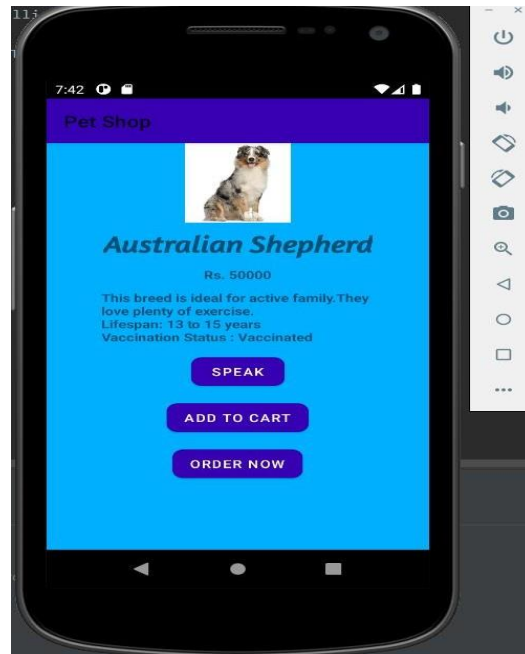
**4.8 Birds** : When clicked on Birds category, list of available birds is displayed in a listview.



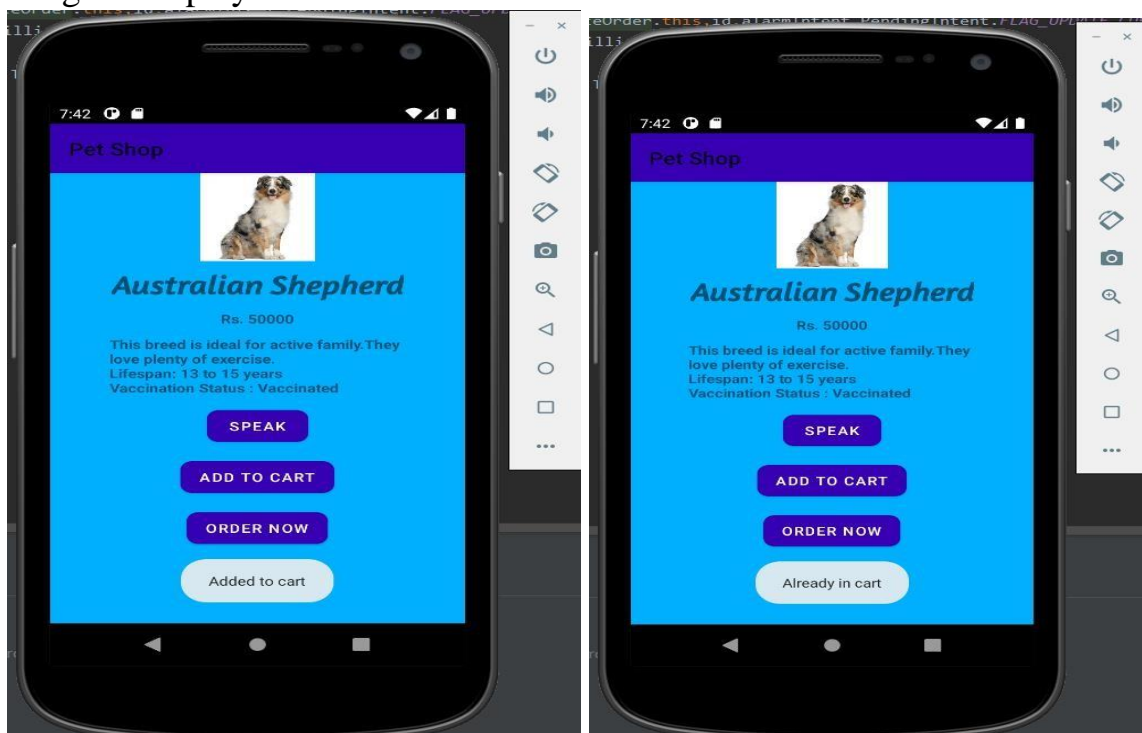
**4.9 Rabbits**: When clicked on Rabbits category, list of available rabbits is displayed in a list view.



**4.10 Details Page:** When clicked on any of the pet, under any section, details of the selected pet is displayed with 'text to speech' button to read out the pet details, 'Add to cart' button to add the pet to cart and 'Order Now' button to place the order.

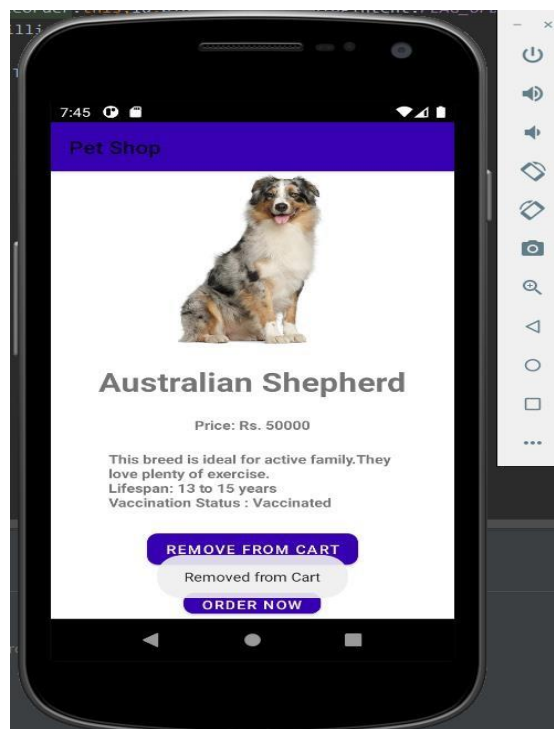
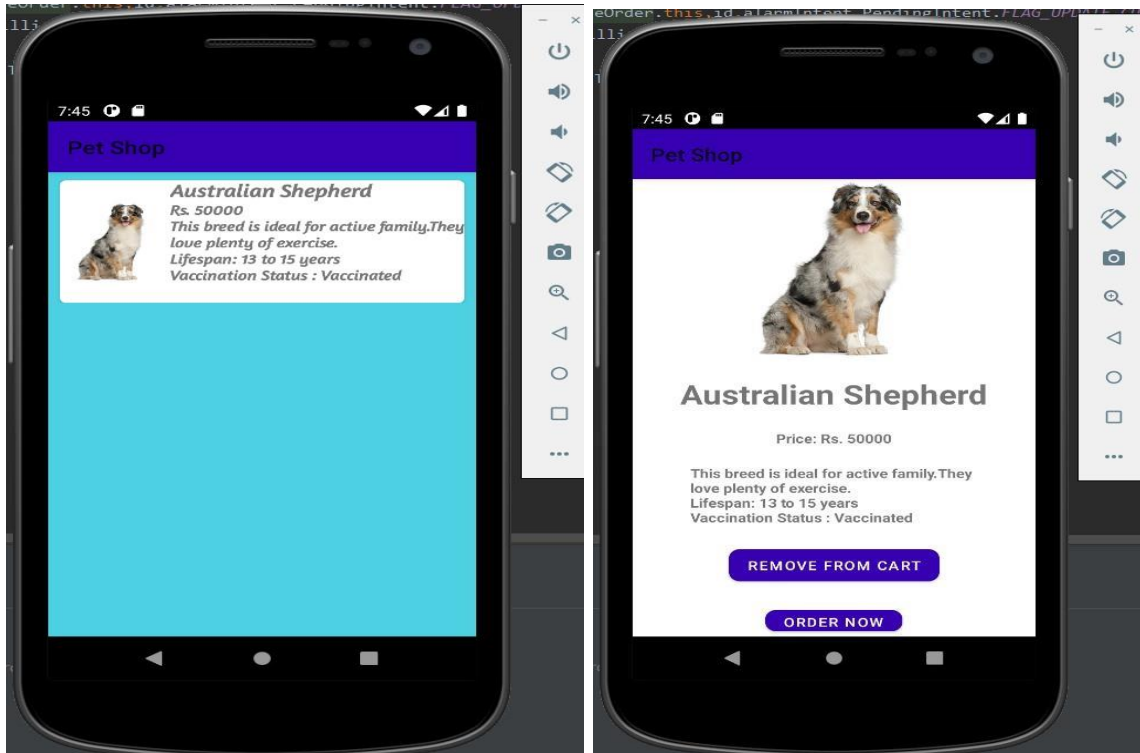


**4.11 Add to Cart:** To add the pet to the Cart, if already present, respective message is displayed .

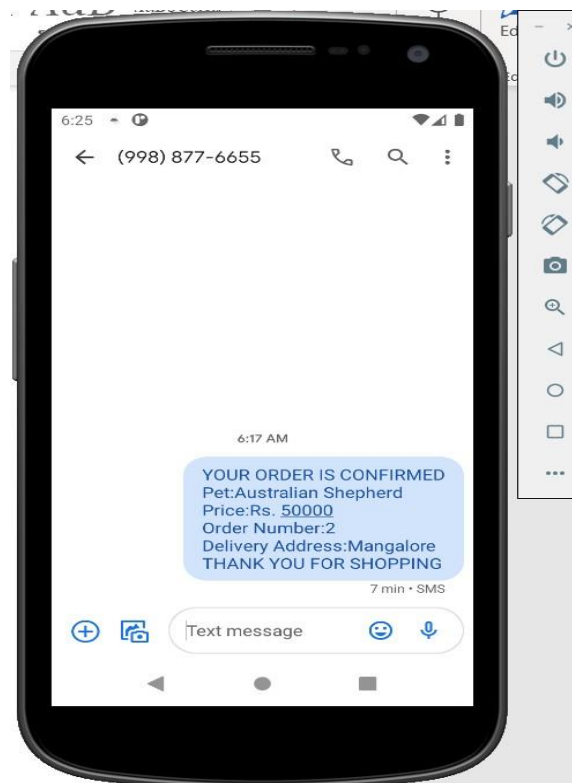
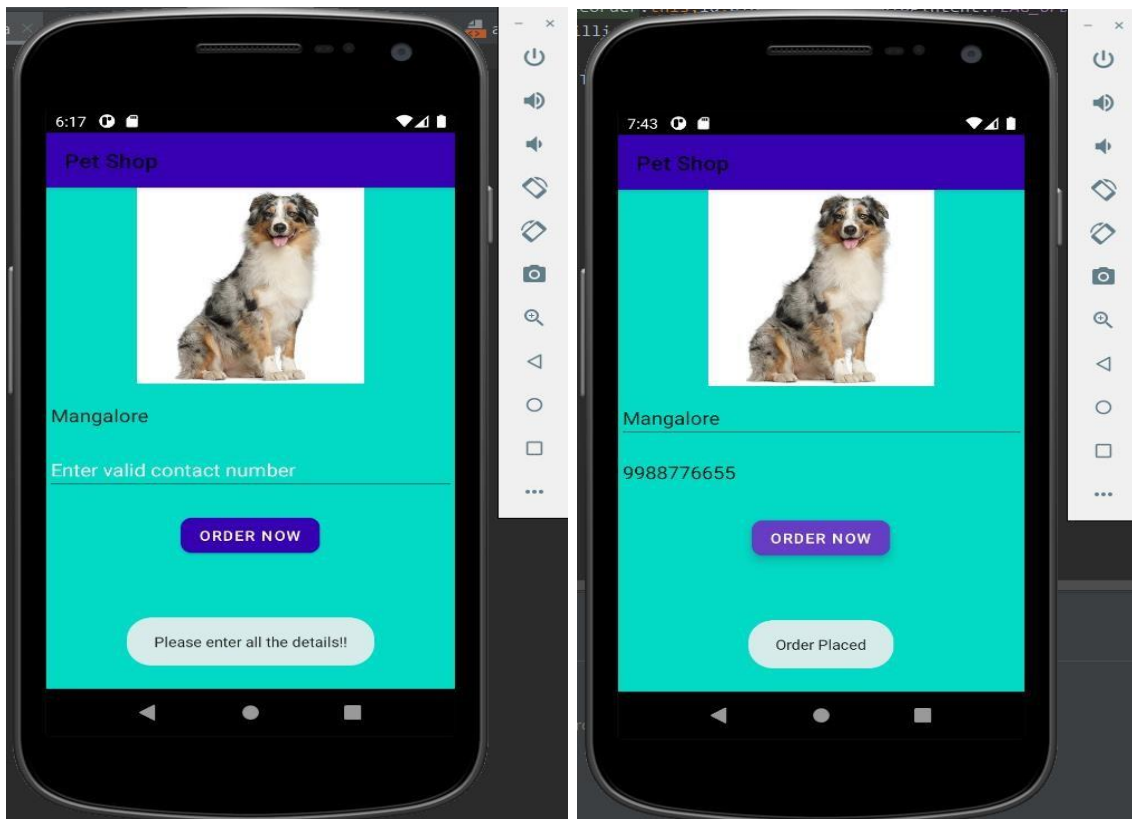




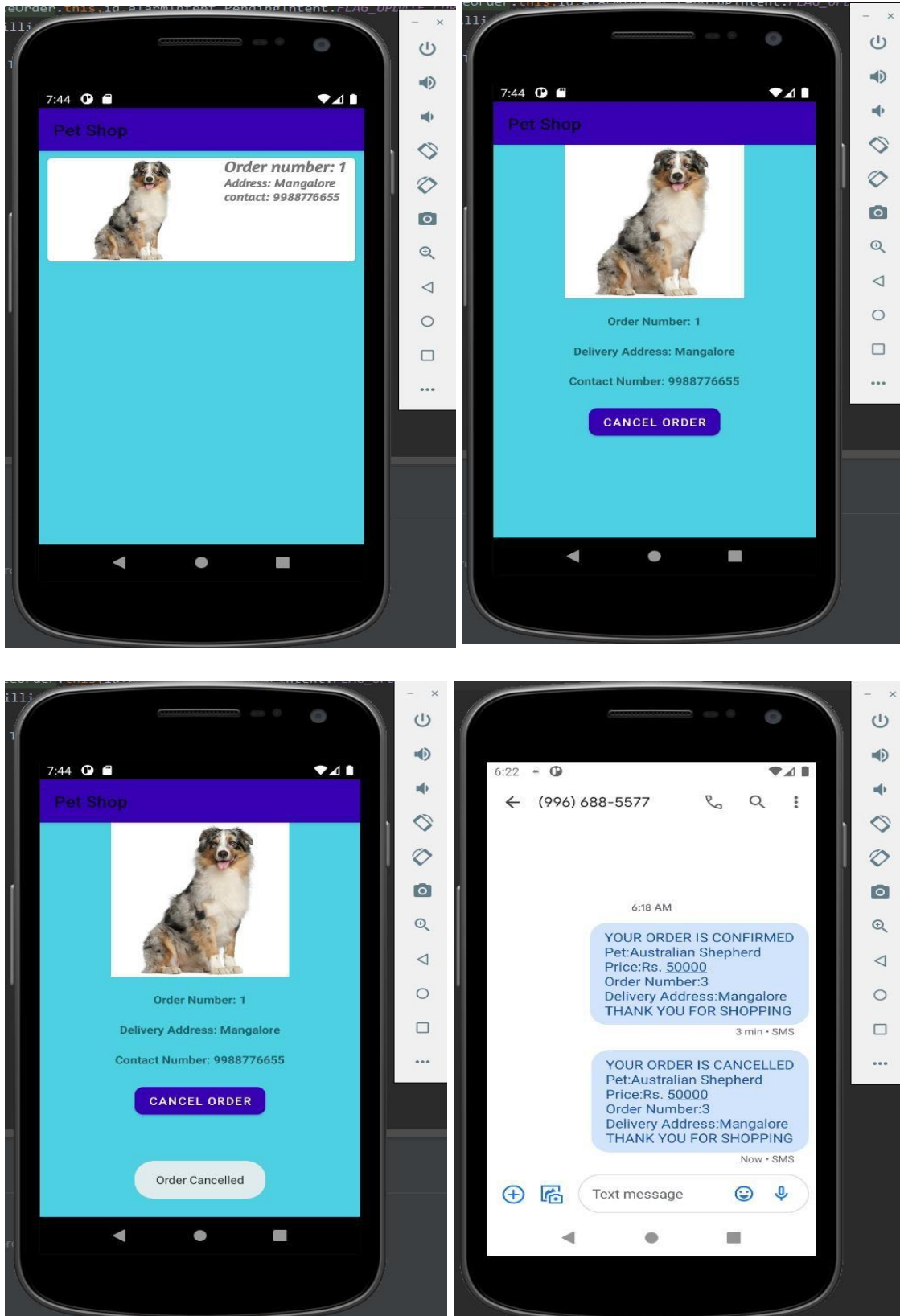
**4.12 View Cart:** View all the pets in the cart. When clicked on the item in the cart, pet details with 2 buttons, 'Remove from cart' to remove the item from the cart and 'order now' are displayed on the next page.



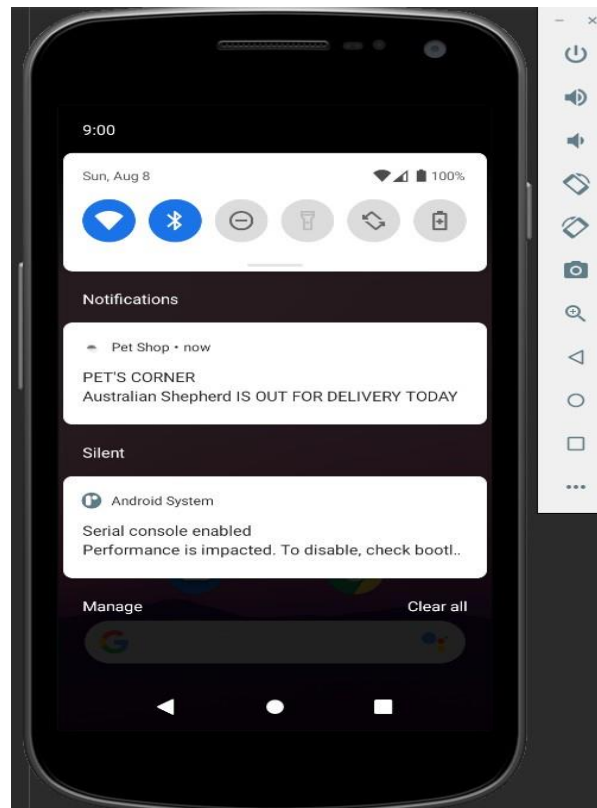
**4.13 Order Now Page:** To order the pet, address and phone number is being asked. SMS with order details is sent to the given phone number.



**4.14 View Orders:** In My Orders, when an ordered pet is clicked, the details of the pet is displayed on the next page with a button to cancel the order.



**4.15 Notification Screen:** Notification pops up after 2 days of order i.e on the day of delivery.



# *References*

- [1] <https://www.geeksforgeeks.org/>
- [2] <http://www.stackoverflow.com>.
- [3] <https://www.youtube.com/watch?v=Le47R9H3gow>
- [4] <https://www.youtube.com/watch?v=JkblErXt1r4&t=2772s>