

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Belagavi - 590 018



Mobile Application Development Laboratory Mini Project Report on

ONLINE PET SHOPPING APP

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

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

| Name | USN |
|---------------------|------------|
| Ms. Ankitha | 4SO18CS011 |
| Ms. Anusha J Shetty | 4SO18CS016 |

Under the guidance of

Ms. Sona Mundody
Assistant Professor
CSE Department, SJEC



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ST JOSEPH ENGINEERING COLLEGE

VAMANJOOR, MANGALURU-575 028, D.K, KARNATAKA

2020-2021

ST JOSEPH ENGINEERING COLLEGE
MANGALURU-575 028
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the Mini project entitled “Online Pet Shopping Application” is a bonafide work carried out by

ANKITHA

4SO18CS011

ANUSHA J SHETTY

4SO18CS016

Students of sixth semester B.E. Computer Science and Engineering, and submitted as a part of the course Mobile Application Development Laboratory (18CSMP68) during the academic year 2020-2021. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in this report.

Ms Sona Mundody
Course Coordinator

Dr Sridevi Saralaya
Head of the Department

External Viva Voce Examination

Name of the Examiners:

Signature with date :

1. -----

2. -----

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.

Acknowledgement

The satisfaction that accompanies the successful completion of any work would be incomplete without thanking the persons who made it perfect with their constant guidance and encouragement.

We take this opportunity to express my sincere thanks to our course coordinator, Ms Sona Mundody, Assistant Professor, Department of CSE, for her support and guidance. Her vision and suggestions throughout the project period has been fundamental in the completion of the project.

We extend our warm gratitude to Dr Sridevi Saralaya, Head of the Department, Department of CSE, for her constant support and advice that helped me to complete this project successfully.

I am extremely grateful to our beloved Principal, Dr Rio D'Souza for his encouragement to come up with new ideas and advice to express them in a systematic manner.

We would like to extend our deep sense of gratitude to our Director, Rev.Fr. Wilfred P D'Souza and Assistant Director Rev.Fr.Alwyn Richard D'Souza for their support and encouragement.

We also like to thank all Teaching & Non-teaching staff of the Computer Science and Engineering Department of St Joseph Engineering College, Mangaluru for their kind co-operation during the course of my work.

Finally we are extremely thankful to our family and friends who helped us in our work and made the project a successful one.

ANKITHA & ANUSHA J SHETTY

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 | 6 |
| 4 | Screen Shots | 68 |
| 4.1 | Running Application | 68 |
| 4.2 | Registration Page | 68 |
| 4.2.1 | Admin Registration | 68 |
| 4.2.1 | User Registration | 69 |
| 4.3 | Login Page | 70 |
| 4.3.1 | Admin Login | 70 |
| 4.3.1 | User Login | 71 |
| 4.4 | Add Pet Details Page | 71 |
| 4.5 | Categories/Buttons Page | 72 |
| 4.6 | Dogs Page | 73 |
| 4.7 | Cats Page | 73 |
| 4.8 | Birds Page | 74 |
| 4.9 | Rabbits Page | 74 |
| 4.10 | Details Page | 75 |
| 4.11 | Add to Cart Page | 75 |

| | |
|------------------------------------|----|
| 4.12 View Cart Page | 76 |
| 4.13 Order Now Page | 77 |
| 4.14 View Orders Page | 78 |
| 4.15 Notification Screen | 79 |
| 5 References | 80 |

Chapter 1

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.

1.1 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.

1.2 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.

1.3 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.

1.4 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

Chapter 2

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.

2.1 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)

2.2 Software Requirements

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

2.3 Functional Requirements

- **Registration**
If a customer wants to buy the product then he/she must be registered with the app. Unregistered 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.

2.4 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.

Chapter 3

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.

3.1 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.

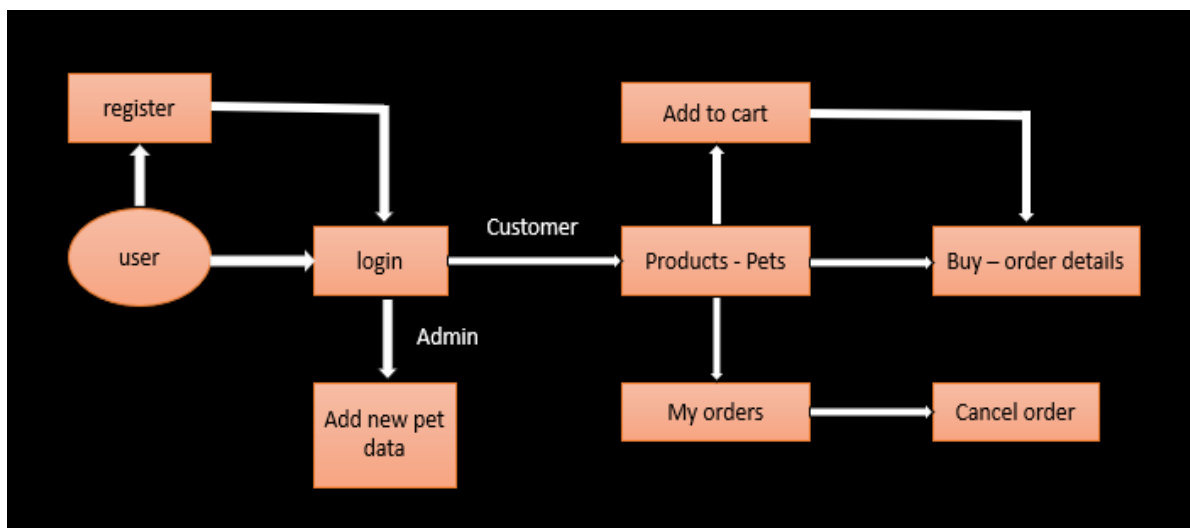


Figure 3.1: Process Flow Diagram

3.2 Source

3.2.1 MainActivity.java

```
package com.example.petshop;  
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();
    }
}
});
    resignin.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.

3.2.2 Login.java

```
package com.example.petshop;

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
Successful", 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.

3.2.3 Buttons.java

```
package com.example.petshop;

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);
        car=(Button)findViewById(R.id.car);
        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(MyCart.class,
MyCart.class);
        intent.putExtras(bundle);
        startActivity(intent);
    }
});
}
```

This is the landing page, once the user successfully logs in. 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 mycards page where they can view the list of orders made by them and list of pets in their cart respectively.

3.2.4 Home.java

```
package com.example.petshop;

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;
    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.recyclerview.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",getPackageNa
me());
        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).

3.2.5 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.

3.2.6 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);
    }
}
}
```

This is the mainadapter file with viewholder to support listview in the Home page. When an item in the list is clicked, it takes the user to the details page with image, name, price, description, and username passed through Intent Extra.

3.2.7 Detail.java

```
package com.example.petshop;

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;
    @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
imagenam=getResources().getResourceName(dimage);
        String[] images=imagenam.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.

3.2.8 PlaceOrder.java

```
package com.example.petshop;

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",getPackage
Name());
        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);
            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,alarmI
ntent,PendingIntent.FLAG_UPDATE_CURRENT);

alarmManager.set(AlarmManager.RTC_WAKEUP,calendar.getTimeInM
illis(),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.

3.2.9 MyCart.java

```
package com.example.petshop;

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

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

import com.example.petshop.Adapters.CartAdapter;
import com.example.petshop.Adapters.MainAdapter;
import com.example.petshop.Models.CartModel;
import com.example.petshop.Models.MainModel;
import com.example.petshop.databinding.ActivityHomeBinding;
import
com.example.petshop.databinding.ActivityMyCartBinding;

import java.util.ArrayList;

public class MyCart extends AppCompatActivity {
```

```
ActivityMyCartBinding binding;
DatabaseHelper mydb;
ArrayList<CartModel> list= new ArrayList<>();
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    mydb=new DatabaseHelper(this);
    String username;

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

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

    CartAdapter adapter=new CartAdapter(list,this);
    binding.recyclerview.setAdapter(adapter);

    LinearLayoutManager layoutManager=new
LinearLayoutManager(this);
    binding.recyclerview.setLayoutManager(layoutManager);
}
public void fetch(Cursor cursor,String username)
{
    cursor.moveToFirst();
    while (cursor.isAfterLast()==false)
    {
        String image = cursor.getString(1);
        Cursor cursor1=mydb.fetchDetail(image);
        if(cursor1.moveToFirst())
        {
            String imagec = cursor1.getString(0);
            String name = cursor1.getString(1);
            String price = cursor1.getString(2);
            String desc = cursor1.getString(3);
            int
i=getResources().getIdentifier(imagec,"drawable",getPackageName());
            list.add(new
CartModel(i,name,price,desc,username));
        }
        cursor.moveToNext();
    }
}
```

This activity is responsible for displaying all the pets in the cart in the list view. User's cart is fetched from the database and displayed. This activity is binded with the activity_my_cart xml file (ActivityMyCartBinding).

3.2.10 CartModel.java

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

public class CartModel {
    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 CartModel(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 MyCart page. Constructor and getter setter methods are implemented here.

3.2.11 CartAdapter.java

```
package com.example.petshop.Adapters;  
  
import android.content.Context;  
import android.content.Intent;  
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.Detail;  
import com.example.petshop.Models.CartModel;  
import com.example.petshop.R;  
import com.example.petshop.ViewCart;  
import java.util.ArrayList;  
  
public class CartAdapter extends  
RecyclerView.Adapter<CartAdapter.viewholder> {  
    ArrayList<CartModel> list;  
    Context context;  
  
    public CartAdapter(ArrayList<CartModel> list, Context  
context) {  
        this.list = list;  
        this.context = context;  
    }  
  
    @NonNull  
    @Override  
    public CartAdapter.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 viewholder holder,  
int position) {  
        final CartModel 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,
ViewCart.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);
            }
        }
    }
}
```

This is the cartadapter file with viewholder to support listview in the MyCart page. When an item in the list is clicked, it takes the user to the ViewCart page with image,name,price,description and username passed through Intent Extra

3.2.12 ViewCart.java

```
package com.example.petshop;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
```

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

public class ViewCart extends AppCompatActivity {
    ImageView image;
    TextView name,price,desc;
    Button cart,order;
    DatabaseHelper mydb;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_cart);

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

        int cimage=getIntent().getIntExtra("image",0);
        String cname=getIntent().getStringExtra("name");
        String cprice=getIntent().getStringExtra("price");
        String cdesc=getIntent().getStringExtra("desc");
        String cuser=getIntent().getStringExtra("username");
        String
imagenam=getResources().getResourceName(cimage);
        String[] images=imagenam.split("/");

        image.setImageResource(cimage);
        name.setText(cname);
        price.setText("Price: "+cprice);
        desc.setText(cdesc);

        order.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Bundle bundle = new Bundle();
                bundle.putString("username",cuser);
                bundle.putString("image",images[1]);
                Intent intent = new Intent(ViewCart.this,
PlaceOrder.class);
                intent.putExtras(bundle);
                startActivity(intent);
            }
        });
        cart.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(mydb.deleteCart(cuser,images[1]))
                {
```

```
        Toast.makeText(ViewCart.this,
"Removed from Cart", Toast.LENGTH_SHORT).show();
    }
    else {
        Toast.makeText(ViewCart.this, "Failed
to remove", Toast.LENGTH_SHORT).show();
    }
}
    });
}
}
```

This activity displays details of the pet selected from the cart. User can order that particular pet by clicking on the Order button, which lands them to PlaceOrder page. User can also remove the item from the cart by clicking on Remove from cart button, which would delete that particular pet from the Cart database table for that User.

3.2.13 Orders.java

```
package com.example.petshop;

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

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

import com.example.petshop.Adapters.MainAdapter;
import com.example.petshop.Adapters.OrderAdapter;
import com.example.petshop.Models.MainModel;
import com.example.petshop.Models.OrderModel;
import com.example.petshop.databinding.ActivityHomeBinding;
import com.example.petshop.databinding.ActivityOrdersBinding;

import java.util.ArrayList;

public class Orders extends AppCompatActivity {
    ActivityOrdersBinding binding;
    DatabaseHelper mydb;
    ArrayList<OrderModel> list= new ArrayList<>();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mydb=new DatabaseHelper(this);

        binding=ActivityOrdersBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
```

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

OrderAdapter adapter=new OrderAdapter(list,this);
binding.recyclerview.setAdapter(adapter);

LinearLayoutManager layoutManager=new
LinearLayoutManager(this);
binding.recyclerview.setLayoutManager(layoutManager);
}
public void fetch(Cursor cursor,String username)
{
    cursor.moveToFirst();
    while (cursor.isAfterLast()==false)
    {
        int ordeno = cursor.getInt(0);
        String image = cursor.getString(2);
        String address = cursor.getString(3);
        String contact = cursor.getString(4);
        int
i=getResources().getIdentifier(image,"drawable",getPackageName());
        list.add(new
OrderModel(i,Integer.toString(ordeno),address,contact));
        cursor.moveToNext();
    }
}
}
```

This activity is responsible for displaying all the pets, which are been ordered by the user, in the list view. User's orders are fetched from the database and displayed. This activity is binded with the activity_orders xml file (ActivityOrdersBinding).

3.2.14 OrderModel.java

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

public class OrderModel {
    int image;
    String address,contact,ordeno;

    public int getImage() {
        return image;
    }

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

```
    public String getOrderno() {
        return orderno;
    }

    public void setOrderno(String orderno) {
        this.orderno = orderno;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public String getContact() {
        return contact;
    }

    public void setContact(String contact) {
        this.contact = contact;
    }

    public OrderModel(int image, String orderno, String
address, String contact) {
        this.image = image;
        this.address = address;
        this.contact = contact;
        this.orderno = orderno;
    }
}
```

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

3.2.15 OrderAdapter.java

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

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
```

```
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.example.petshop.Buttons;
import com.example.petshop.CancelOrder;
import com.example.petshop.DatabaseHelper;
import com.example.petshop.Detail;
import com.example.petshop.Models.OrderModel;
import com.example.petshop.Orders;
import com.example.petshop.R;

import java.util.ArrayList;

public class OrderAdapter extends
RecyclerView.Adapter<OrderAdapter.viewholder> {
    ArrayList<OrderModel> list;
    Context context;

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

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

    @Override
    public void onBindViewHolder(@NonNull viewholder holder,
int position) {
        final OrderModel model=list.get(position);
        holder.image.setImageResource(model.getImage());
        holder.address.setText(model.getAddress());
        holder.phone.setText(model.getContact());
        holder.orderno.setText(model.getOrderno());
        holder.itemView.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(context,
CancelOrder.class);
```

```
        intent.putExtra("image",model.getImage());
        intent.putExtra("address",model.getAddress());
        intent.putExtra("contact",model.getContact());
        intent.putExtra("orderno",model.getOrderno());
        context.startActivity(intent);
    }
    });
}

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

public class viewholder extends RecyclerView.ViewHolder {
    ImageView image;
    TextView address,phone, orderno;
    public viewholder(@NonNull View itemView) {
        super(itemView);
        image = itemView.findViewById(R.id.oimage);
        address = itemView.findViewById(R.id.address);
        phone = itemView.findViewById(R.id.phone);
        orderno = itemView.findViewById(R.id.orderno);
    }
}
}
```

This is the ordersadapter file with viewholder to support listview in the Orders page. When an item in the list is clicked, it takes the user to the CancelOrder page with image,address, contact and order number passed through Intent Extra.

3.2.16 CancelOrder.java

```
package com.example.petshop;

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.speech.tts.TextToSpeech;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

public class CancelOrder extends AppCompatActivity {
```

```
ImageView image;
TextView orderno,address,contact;
Button cancel;
DatabaseHelper mydb;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_cancel_order);

    image=(ImageView) findViewById(R.id.image);
    orderno=(TextView) findViewById(R.id.orderno);
    address=(TextView) findViewById(R.id.address);
    contact=(TextView) findViewById(R.id.contact);
    cancel=(Button) findViewById(R.id.cancel);
    mydb=new DatabaseHelper(this);

    int cimage=getIntent().getIntExtra("image",0);
    String
    corderno=getIntent().getStringExtra("orderno");
    String
    caddress=getIntent().getStringExtra("address");
    String
    ccontact=getIntent().getStringExtra("contact");
    String
    imagename=getResources().getResourceName(cimage);
    String[] images=imagename.split("/");

    image.setImageResource(cimage);
    orderno.setText("Order Number: "+corderno);
    address.setText("Delivery Address: "+caddress);
    contact.setText("Contact Number: "+ccontact);

    cancel.setOnClickListener(new View.OnClickListener()
    {
        @Override
        public void onClick(View v) {

if(mydb.deleteOrder(Integer.parseInt(corderno)))
        {
            Toast.makeText(CancelOrder.this,
"Order Cancelled", Toast.LENGTH_SHORT).show();
            Cursor
            cursor1=mydb.fetchDetail(images[1]);
            cursor1.moveToFirst();
            String name=cursor1.getString(1);
            String price=cursor1.getString(2);

ActivityCompat.requestPermissions(CancelOrder.this, new
String[]{Manifest.permission.SEND_SMS},
PackageManager.PERMISSION_GRANTED);
            SmsManager mySmsManager =
SmsManager.getDefault();
            String text="YOUR ORDER IS
CANCELLED\nPet:"+name+"\nPrice:"+price+"\nOrder
Number:"+corderno+"\nDelivery Address:"+caddress+"\nTHANK
YOU FOR SHOPPING";
```

```
mySmsManager.sendMessage(ccontact, null, text, null, null);
    }
    else {
        Toast.makeText(CancelOrder.this,
"Failed to cancel", Toast.LENGTH_SHORT).show();
    }
}
});
}
}
```

This activity displays order details of the pet selected from the list of orders. User can cancel that particular order by clicking on Cancel Order button, which would delete that order from the Orders table in the database. After successfully cancellation of order, a SMS is sent to the phone number which was given during the order, with the order details of the cancelled order.

3.2.17 Admin.java

```
package com.example.petshop;

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 Admin extends AppCompatActivity {
    Button login;
    EditText pwd;
    DatabaseHelper mydb;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_admin);
        login=(Button)findViewById(R.id.login);
        pwd=(EditText)findViewById(R.id.pwd);
        mydb=new DatabaseHelper(this);

        login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String Currentuser="ADMIN";
                String
Currentpassword=pwd.getText().toString().trim();

                if(mydb.CheckLoginStatus(Currentuser,Currentpassword))
                {
```

```
        pwd.setText("");
        Toast.makeText(Admin.this, "Login
Successfull", Toast.LENGTH_SHORT).show();
        Intent intent = new Intent(Admin.this,
Addpets.class);
        startActivity(intent);
    }
    else
    {
        Toast.makeText(Admin.this, "Invalid
password", Toast.LENGTH_SHORT).show();
    }
}
});
}
```

This is the Login page for the ADMIN, where on entering the correct password with which he registered, ADMIN Login will be successful. On successful login, he is redirected to the AddPets page.

3.2.18 AddPets.java

```
package com.example.petshop;

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 Addpets extends AppCompatActivity {
    EditText imagefile,petname,price,desc,category;
    Button add;
    DatabaseHelper mydb;
    boolean status;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_addpets);

        imagefile=(EditText) findViewById(R.id.imagefile);
        petname=(EditText) findViewById(R.id.petname);
        price=(EditText) findViewById(R.id.price);
        desc=(EditText) findViewById(R.id.desc);
        category=(EditText) findViewById(R.id.category);
        add=(Button) findViewById(R.id.add) ;
        mydb=new DatabaseHelper(this);
```

```
add.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String i,pn,p,d,c;
        i=imagefile.getText().toString();
        pn=petname.getText().toString();
        p=price.getText().toString();
        d=desc.getText().toString();
        c=category.getText().toString();
        if(i.equals("") || pn.equals("") ||
p.equals("") || d.equals("") || c.equals(""))
        {
            Toast.makeText(getApplicationContext(),
"Please enter all the details!!", Toast.LENGTH_SHORT).show();
        }
        else if(mydb.searchImage(i.trim()))
        {
            Toast.makeText(Addpets.this, "Image File
Already in use", Toast.LENGTH_SHORT).show();
        }
        else
        {
            try {
                int
i1=getResources().getIdentifier(i.trim(), "drawable",
getPackageName());
                String
imagenam=getResources().getResourceName(i1);
                status = mydb.insertPet(i.trim(),
pn.trim(), p.trim(), d.trim(), c.trim());
                if (status) {
                    Toast.makeText(Addpets.this,
"Record Added!", Toast.LENGTH_SHORT).show();
                    imagefile.setText("");
                    petname.setText("");
                    price.setText("");
                    desc.setText("");
                    category.setText("");
                } else {
                    Toast.makeText(Addpets.this,
"Failed to add record", Toast.LENGTH_SHORT).show();
                }
            } catch (Exception e) {
                Toast.makeText(Addpets.this, "Image
file not found", Toast.LENGTH_SHORT).show();
            }
        }
    }
});
```

```
}  
}
```

Only ADMIN can access this page, where he can add new pets to the database. Image file name, name or breed of the pet, price, description and category (Dogs, Cats, Birds, Rabbits) are the fields to be added by the ADMIN, where neither of them can be NULL. Existing pet image cannot be added to the newly added pet.

3.2.19 DatabaseHelper.java

```
package com.example.petshop;  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
import android.widget.Toast;  
import androidx.annotation.Nullable;  
public class DatabaseHelper extends SQLiteOpenHelper {  
    static final String DATABASE = "PetShopDB";  
    static final int DATABASE_VERSION=1;  
    static final String TABLE_NAME="User";  
    static final String TABLE_NAME1="Pets";  
    static final String TABLE_NAME2="Cart";  
    static final String TABLE_NAME3="Orders";  
    static final String USERNAME = "username";  
    static final String PASSWORD = "password";  
    static final String IMAGE = "image";  
    static final String NAME = "name";  
    static final String PRICE = "price";  
    static final String DESC = "descrip";  
    static final String CATEGORY = "category";  
    public DatabaseHelper(Context context) {  
        super(context,DATABASE, null, DATABASE_VERSION);  
    }  
  
    @Override  
    public void onCreate(SQLiteDatabase db) {  
        db.execSQL("CREATE TABLE "+TABLE_NAME+"(username TEXT  
NOT NULL PRIMARY KEY,password TEXT NOT NULL)");  
        db.execSQL("CREATE TABLE "+TABLE_NAME1+"(image TEXT  
NOT NULL PRIMARY KEY,name TEXT NOT NULL,price TEXT NOT  
NULL,descrip TEXT NOT NULL,category TEXT NOT NULL)");  
        db.execSQL("CREATE TABLE "+TABLE_NAME2+"(username TEXT  
NOT NULL,image TEXT NOT NULL," +  
        "PRIMARY KEY(username,image),FOREIGN KEY  
(username) REFERENCES User(username) ON DELETE  
CASCADE,FOREIGN KEY (image) REFERENCES Pets(image) ON DELETE  
CASCADE)");  
    }  
}
```

```
        db.execSQL("CREATE TABLE "+TABLE_NAME3+" (orderno
INTEGER PRIMARY KEY AUTOINCREMENT,username TEXT NOT
NULL,image TEXT NOT NULL,address TEXT NOT NULL,contact TEXT
NOT NULL," +
        "FOREIGN KEY (username) REFERENCES
User(username) ON DELETE CASCADE,FOREIGN KEY (image)
REFERENCES Pets(image) ON DELETE CASCADE)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion,
int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS "+TABLE_NAME);
        db.execSQL("DROP TABLE IF EXISTS "+TABLE_NAME1);
        db.execSQL("DROP TABLE IF EXISTS "+TABLE_NAME2);
        db.execSQL("DROP TABLE IF EXISTS "+TABLE_NAME3);
        onCreate(db);
    }

    public boolean insertUser(String username,String password)
    //Donot use the name insert for method here. It will try
    to override default method.
    {
        SQLiteDatabase db=this.getWritableDatabase();
        long result;
        ContentValues values=new ContentValues();
        values.put(USERNAME,username);
        values.put(PASSWORD,password);
        result=db.insert(TABLE_NAME,null,values);
        if(result==-1)
            return false;
        else
            return true;
    }

    public boolean insertPet(String image,String name,String
price,String desc,String cat)
    //Donot use the name insert for method here. It will try
    to override default method.
    {
        SQLiteDatabase db=this.getWritableDatabase();
        long result;
        ContentValues values=new ContentValues();
        values.put(IMAGE,image);
        values.put(NAME,name);
        values.put(PRICE,price);
        values.put(DESC,desc);
        values.put(CATEGORY,cat);
        result=db.insert(TABLE_NAME1,null,values);
        if(result==-1)
            return false;
        else
```

```
        return true;
    }
    public boolean insertOrder(String username,String
image,String address,String phone)
    //Donot use the name insert for method here. It will try
to override default method.
    {
        SQLiteDatabase db=this.getWritableDatabase();
        long result;
        ContentValues values=new ContentValues();
        values.put(USERNAME,username);
        values.put(IMAGE,image);
        values.put("address",address);
        values.put("contact",phone);
        result=db.insert(TABLE_NAME3,null,values);
        if(result!=-1)
            return false;
        else
            return true;
    }
    public Cursor fetchData(String cat){
        SQLiteDatabase db=this.getWritableDatabase();
        Cursor cursor=db.rawQuery("SELECT * FROM
"+TABLE_NAME1+" WHERE category = '"+cat+"'",null);
        return cursor;
    }
    public Cursor fetchDetail(String image)
    {
        SQLiteDatabase db=this.getWritableDatabase();
        Cursor cursor=db.rawQuery("SELECT * FROM
"+TABLE_NAME1+" WHERE image = '"+image+"'",null);
        return cursor;
    }
    public Cursor fetchOrder(String username)
    {
        SQLiteDatabase db=this.getWritableDatabase();
        Cursor cursor=db.rawQuery("SELECT * FROM
"+TABLE_NAME3+" WHERE username = '"+username+"'",null);
        return cursor;
    }
    public Cursor fetchNum()
    {
        SQLiteDatabase db=this.getWritableDatabase();
        Cursor cursor=db.rawQuery("SELECT MAX (orderno) FROM
"+TABLE_NAME3,null);
        return cursor;
    }
    public Cursor fetchCart(String username)
    {
        SQLiteDatabase db=this.getWritableDatabase();
```

```
        Cursor cursor=db.rawQuery("SELECT * FROM
"+TABLE_NAME2+" WHERE username = '"+username+"'",null);
        return cursor;
    }
    public boolean searchUser(String uname){
        SQLiteDatabase db=this.getWritableDatabase();
        Cursor cursor=db.rawQuery("SELECT * FROM
"+TABLE_NAME2+" WHERE username = '"+uname+"'",null);
        if(cursor.moveToFirst())
            return true;
        else
            return false;
    }
    public boolean deleteOrder(int ordeno)
    {
        SQLiteDatabase db=this.getWritableDatabase();
        long result=db.delete(TABLE_NAME3, "orderno =
"+orderno,null) ;
        if(result==-1)
            return false;
        else
            return true;
    }
    public boolean deleteCart(String user,String image)
    {
        SQLiteDatabase db=this.getWritableDatabase();
        long result=db.delete(TABLE_NAME2, "username = ? AND
image = ?",new String[]{user,image}) ;
        if(result==-1)
            return false;
        else
            return true;
    }
    public boolean searchCart(String uname,String image)
    {
        SQLiteDatabase db=this.getWritableDatabase();
        Cursor cursor=db.rawQuery("SELECT * FROM
"+TABLE_NAME2+" WHERE username = '"+uname+"'"+" AND image =
 '"+image+"'",null);
        if(cursor.moveToFirst()) {
            return false;
        }
        else {
            long result;
            ContentValues values=new ContentValues();
            values.put(USERNAME,uname);
            values.put(IMAGE,image);
            result=db.insert(TABLE_NAME2,null,values);
            if(result==-1)
                return false;
            else
```

```
        return true;
    }
}

public boolean CheckLoginStatus(String uname,String pwd)
{
    SQLiteDatabase db=this.getWritableDatabase();
    Cursor cursor=db.rawQuery("SELECT * FROM
"+TABLE_NAME+" WHERE username = '"+uname+"'"+" AND password =
 '"+pwd+"'",null);
    if(cursor.moveToFirst())
        return true;
    else
        return false;
}

public boolean searchImage(String iname)
{
    SQLiteDatabase db=this.getWritableDatabase();
    Cursor cursor=db.rawQuery("SELECT * FROM
"+TABLE_NAME1+" WHERE image = '"+iname+"'",null);
    if(cursor.moveToFirst())
        return true;
    else
        return false;
}
}
```

This is the DatabaseHelper file, where 4 tables are created i.e, User, Pets, Cart and Orders, with required methods implemented.

InsertUser, InsertPet, InsertOrder methods, inserts new user, new pets and new orders to the database respectively.

FetchData method, fetches the pet data from Pets table based on Category name.

FetchDetail method, fetches the pet data from Pets table based on Image file name.

FetchOrder method, fetches the orders detail from Orders table based on username.

FetchNum method, fetches the most recent order number(MAX) from the database.

FetchCart method, fetches the all the pets in the cart from cart table based on username.

SearchUser method, fetches the users detail from User table based on username.

DeleteOrder method, deletes a order from the Orders table, based on order number.

DeleteCart method, deletes a cart item based on username and image file name.

SearchCart method, searches the username and image file name in the carts table. If it is not present, it inserts the new pet to the cart.

CheckLoginstatus method, checks whether the username and password is present in the database to validate the Login of the user.

SearchImage method, fetches the pet data from Pets table based on Image file name and returns true if it is present, else false.

3.2.20 AlarmReceiver.java

```
package com.example.petshop;

import android.app.AlarmManager;
import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.widget.Button;

import androidx.core.app.NotificationCompat;

public class AlarmReceiver extends BroadcastReceiver {
    boolean connected=true;
    public static final String NOTIFICATION_CHANNEL_ID =
"10001" ;
    private final static String
default_notification_channel_id = "default" ;
    @Override
    public void onReceive(Context context, Intent intent) {
        String noti=intent.getStringExtra("noti");
        NotificationCompat.Builder builder = new
NotificationCompat.Builder(context,default_notification_chann
el_id)
            .setSmallIcon(R.drawable.
ic_launcher_foreground )
            .setContentTitle("PET'S CORNER")
            .setContentText(noti)
            .setAutoCancel(true)
            .setChannelId(NOTIFICATION_CHANNEL_ID);
        NotificationManager
manager=(NotificationManager)context.getSystemService(Context
.NOTIFICATION_SERVICE);
        Notification notification = builder.build() ;
        if (android.os.Build.VERSION.SDK_INT >=
android.os.Build.VERSION_CODES.O ) {
            int importance = NotificationManager.
IMPORTANCE_HIGH ;
            NotificationChannel notificationChannel = new
NotificationChannel( NOTIFICATION_CHANNEL_ID ,
"NOTIFICATION_CHANNEL_NAME" , importance) ;

manager.createNotificationChannel(notificationChannel) ;
        }
        assert manager != null;
        if ( connected ) {
            manager.notify( 1 , notification) ;
            connected = false;
        }
    }
}
```

```
        } else {
            manager.cancel( 1 ) ;
            connected = true;
        }
    }
}
```

This is the broadcast receiver file, to set notification after 2 days of order (Delivery day).

3.2.21 activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#4DD0E1"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".MainActivity"

    xmlns:android="http://schemas.android.com/apk/res/android">
    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="130dp"
        android:id="@+id/signupimg"
        android:layout_marginTop="0dp"
        android:background="@drawable/sigpuppage">
    </RelativeLayout>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="PET'S CORNER"
        android:textSize="30sp"
        android:textStyle="bold"
        android:layout_centerHorizontal="true"
        android:layout_below="@+id/signupimg"
        android:layout_marginTop="10dp"
        android:textColor="#311B92"
        android:id="@+id/textv1">
    </TextView>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Signup"
        android:layout_below="@id/textv1"
```



```
        android:textSize="30sp"
        android:textStyle="bold"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="8dp"
        android:textColor="#000"
        android:id="@+id/textv2">
</TextView>
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_below="@id/textv2"
    android:layout_marginTop="16dp"
    android:id="@+id/reguser"
    android:hint="Enter Username"

    android:inputType="textVisiblePassword|textNoSuggestions"
    android:textColorHint="#FFFFFF">
</EditText>
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_below="@id/reguser"
    android:layout_marginTop="4dp"
    android:id="@+id/regpwd"
    android:hint="Enter password"
    android:inputType="textPassword"
    android:textColorHint="#FFFFFF">
</EditText>
<Button
    android:id="@+id/regsignup"
    android:layout_width="290dp"
    android:layout_height="wrap_content"
    android:layout_below="@id/regpwd"
    android:layout_centerHorizontal="true"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="2dp"
```

```
        android:backgroundTint="#4A148C"
        android:text="New user?Signup to continue"
        app:cornerRadius="10dp">
</Button>
<Button
    android:id="@+id/regsigin"
    android:layout_width="290dp"
    android:layout_height="wrap_content"
    android:layout_below="@+id/regsingup"
    android:layout_centerHorizontal="true"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="2dp"
    android:backgroundTint="#4A148C"
    android:text="Have an account?LOGIN"
    app:cornerRadius="10dp">
</Button>
<Button
    android:id="@+id/admin"
    android:layout_width="290dp"
    android:layout_height="wrap_content"
    android:layout_below="@+id/regsigin"
    android:layout_centerHorizontal="true"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="2dp"
    android:backgroundTint="#4A148C"
    android:text="ARE YOU AN ADMIN?"
    app:cornerRadius="10dp">
</Button>
</RelativeLayout>
```

This is a xml file for the registration page that consists of the name of the application, two edit text fields to enter the username and password. It also consists of three buttons for sign-up, customer login and admin login.

3.2.22 activity_login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background22"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".Login"
```

```
xmlns:android="http://schemas.android.com/apk/res/android">
    <ImageView
        android:layout_width="120dp"
        android:layout_height="120dp"
        android:id="@+id/title1"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp"
        android:src="@drawable/loginlogo">
    </ImageView>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="WELCOME BACK"
        android:textSize="20sp"
        android:fontFamily="cursive"
        android:textStyle="bold"
        android:layout_below="@+id/title1"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp"
        android:textColor="#311B92"
        android:id="@+id/view3">
    </TextView>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login to buy the pet of your dream"
        android:fontFamily="cursive"
        android:textSize="20sp"
        android:textStyle="bold"
        android:layout_below="@+id/view3"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp"
        android:textColor="#311B92"
        android:id="@+id/textv3">
    </TextView>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login"
        android:layout_below="@id/textv3"
        android:textSize="30sp"
        android:textStyle="bold"
        android:layout_centerHorizontal="true"
```

```
        android:layout_marginTop="10dp"
        android:textColor="#0E0E0E"
        android:id="@+id/textv4">
</TextView>

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_below="@id/textv4"
    android:layout_marginTop="10dp"
    android:id="@+id/user"
    android:hint="Enter Username"

    android:inputType="textVisiblePassword|textNoSuggestions"
    android:textColorHint="#FFFFFF">
</EditText>
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_below="@id/user"
    android:layout_marginTop="10dp"
    android:id="@+id/pwd"
    android:hint="Enter password"
    android:inputType="textPassword"
    android:textColorHint="#FFFFFF">
</EditText>
<Button
    android:id="@+id/login"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/pwd"
    android:layout_centerHorizontal="true"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="10dp"
    android:backgroundTint="#4A148C"
    android:text="Login"
```

```
        app:cornerRadius="10dp"></Button>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New user?Click below to Register"
        android:layout_below="@id/login"
        android:textSize="15sp"
        android:textStyle="bold"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp"
        android:textColor="#000"
        android:id="@+id/view5">
    </TextView>

    <Button
        android:id="@+id/signup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/view5"
        android:layout_centerHorizontal="true"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="10dp"
        android:backgroundTint="#4A148C"
        android:text="SIGNUP"
        app:cornerRadius="10dp"></Button>
</RelativeLayout>
```

This is the xml file for the login page which consists of some textview to provide some information to the user, and two edit text fields to enter the username and password of the user. It also consists of two buttons. Users can press the login button to login to the application or signup button to register themselves with the application.

3.2.23 activity_addpets.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Addpets">
    <RelativeLayout
        android:layout_width="match_parent"
```

```
        android:layout_height="120dp"
        android:id="@+id/addpetrell1"
        android:background="@drawable/addpetbackground">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="ADD PET DETAILS"
            android:textSize="30sp"
            android:textStyle="bold"
            android:fontFamily="cursive"
            android:layout_marginTop="80dp"
            android:layout_centerHorizontal="true"
            android:id="@+id/addpetview1">
        </TextView>
    </RelativeLayout>
    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/addpetrell1"
        android:layout_marginTop="4dp"
        android:id="@+id/imagefile"
        android:hint="Enter Image file name without
extension"
        android:textColorHint="#B388FF">
    </EditText>
    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_below="@id/imagefile"
        android:layout_marginTop="4dp"
        android:id="@+id/petname"
        android:hint="Enter Pet Name"
        android:textColorHint="#B388FF">
    </EditText>

    <EditText
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_below="@id/petname"
        android:layout_marginTop="4dp"
        android:id="@+id/category"
        android:hint="Enter Pet
Category (Dogs/Cats/Birds, Rabbits) "
        android:textColorHint="#B388FF">
</EditText>

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_marginTop="4dp"
    android:id="@+id/price"
    android:layout_below="@+id/category"
    android:hint="Enter Price"
    android:textColorHint="#B388FF">
</EditText>
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_below="@id/price"
    android:layout_marginTop="4dp"
    android:id="@+id/desc"
    android:hint="Enter Description"
    android:textColorHint="#B388FF">
</EditText>
<Button
    android:id="@+id/add"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
        android:layout_below="@+id/desc"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="20dp"
        android:backgroundTint="#4A148C"
        android:text="ADD"
        app:cornerRadius="10dp">
</Button>

</RelativeLayout>
```

This is the xml file for the add pets page which allows the admin to add new pets to the application. It consists of 5 edit text fields where the admin can enter the name of the image file, name or breed of the pet, its category, price, and description. It Also consists of a button, on pressing which the admin can add this product to his application.

3.2.24 activity_admin.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#32BACC"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".Login"

xmlns:android="http://schemas.android.com/apk/res/android">
    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="220dp"
        android:id="@+id/re11"
        android:layout_marginTop="0dp"
        android:background="@drawable/admin1">
    </RelativeLayout>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Have new Pet?"
        android:textSize="30sp"
        android:textStyle="bold"
        android:fontFamily="cursive"
        android:layout_centerHorizontal="true"
        android:layout_below="@+id/re11"
        android:layout_marginTop="20dp"
```

```
        android:textColor="#311B92"
        android:id="@+id/textv3">
</TextView>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Login"
    android:layout_below="@id/textv3"
    android:textSize="30sp"
    android:textStyle="bold"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"
    android:textColor="#1B5E20"
    android:id="@+id/textv4">
</TextView>
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_below="@id/textv4"
    android:layout_marginTop="10dp"
    android:id="@+id/pwd"
    android:hint="Enter password"
    android:inputType="textPassword"
    android:textColorHint="#FFFFFF">
</EditText>
<Button
    android:id="@+id/login"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="50dp"
    android:backgroundTint="#4A148C"
    android:text="Login"
    app:cornerRadius="10dp">
</Button>

</RelativeLayout>
```

This is the xml file for the admin login page which consists of one edit text field where the admin can enter his password. It also consists of a button, on pressing this admin can login to the application.

3.2.25 activity_buttons.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/buttonsbackground"
    tools:context=".Buttons">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="10dp"
        android:id="@+id/linear0">
        <ImageView
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:id="@+id/categimg5"
            android:layout_marginLeft="30dp"
            android:src="@drawable/myorders">
        </ImageView>
        <ImageView
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:id="@+id/categimg6"
            android:layout_marginLeft="100dp"
            android:src="@drawable/mycart">
        </ImageView>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/linear1"
        android:layout_below="@+id/linear0"
        android:orientation="horizontal"
        android:layout_marginTop="10dp">
        <Button
```

```
        android:id="@+id/orders"
        android:layout_width="120dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="20dp"
        android:backgroundTint="#4A148C"
        android:text="My Orders"
        app:cornerRadius="10dp">
    </Button>
    <Button
        android:id="@+id/cart"
        android:layout_width="120dp"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginLeft="80dp"
        android:backgroundTint="#4A148C"
        android:text="My Cart"
        app:cornerRadius="10dp">
    </Button>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="10dp"
    android:id="@+id/linear2"
    android:layout_below="@+id/linear1">
    <ImageView
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:src="@drawable/goldenretriever"
        android:id="@+id/categimg1"
        android:layout_marginLeft="30dp">
    </ImageView>
    <ImageView
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:src="@drawable/categorycat"
        android:id="@+id/categimg2"
        android:layout_marginLeft="100dp">
    </ImageView>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
        android:orientation="horizontal"
        android:layout_below="@+id/linear2"
        android:layout_marginTop="10dp"
        android:id="@+id/linear3">
        <Button
            android:id="@+id/dogs"
            android:layout_width="120dp"
            android:layout_height="wrap_content"
            android:layout_marginLeft="20dp"
            android:backgroundTint="#4A148C"
            android:text="dogs"
            app:cornerRadius="10dp">
        </Button>
        <Button
            android:id="@+id/cats"
            android:layout_width="120dp"
            android:layout_height="wrap_content"
            android:layout_marginLeft="80dp"
            android:backgroundTint="#4A148C"
            android:text="cats"
            app:cornerRadius="10dp">
        </Button>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:id="@+id/linear4"
        android:layout_below="@+id/linear3"
        android:layout_marginTop="10dp">
        <ImageView
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:src="@drawable/parrotwo"
            android:id="@+id/categimg3"
            android:layout_marginLeft="30dp">
        </ImageView>
        <ImageView
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:src="@drawable/categoryrabbit"
            android:id="@+id/categimg4"
            android:layout_marginLeft="100dp">
        </ImageView>
```

```
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="10dp"
    android:layout_below="@+id/linear4"
    android:id="@+id/linear5">
    <Button
        android:id="@+id/birds"
        android:layout_width="120dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="20dp"
        android:backgroundTint="#4A148C"
        android:text="birds"
        app:cornerRadius="10dp">
    </Button>
    <Button
        android:id="@+id/rabbits"
        android:layout_width="120dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="80dp"
        android:backgroundTint="#4A148C"
        android:text="rabbits"
        app:cornerRadius="10dp">
    </Button>
</LinearLayout>
</RelativeLayout>
```

This is the xml file for the category page which consists of six buttons where four buttons consist of different categories of pets, one button to view the orders and one button to view the cart.

3.2.26 activity_cancel_order.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#4DD0E1"
    tools:context=".Detail">
```

```
<ImageView
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:id="@+id/image">
</ImageView>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:id="@+id/orderno"
    android:textStyle="bold"
    android:layout_below="@+id/image"
    android:layout_marginTop="20dp">
</TextView>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:id="@+id/address"
    android:textStyle="bold"
    android:layout_below="@+id/orderno"
    android:layout_marginTop="20dp">
</TextView>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:id="@+id/contact"
    android:textStyle="bold"
    android:layout_below="@+id/address"
    android:layout_marginTop="20dp">
</TextView>

<Button
    android:id="@+id/cancel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/contact"
    android:layout_centerHorizontal="true"
    android:layout_marginLeft="5dp"
    android:layout_marginTop="20dp"
    android:text="Cancel Order"
    app:cornerRadius="10dp"></Button>
</RelativeLayout>
```

This is the xml file for the cancel order page that consists of a button to cancel the order placed by the user with textviews for the order number, address and contact number. Also imageview for the pet image.

3.2.27 activity_detail.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#00B0FF"
    tools:context=".Detail">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:id="@+id/image">
    </ImageView>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:id="@+id/name"
        android:layout_below="@+id/image"
        android:textStyle="bold"
        android:textSize="30sp"
        android:fontFamily="@font/amaranth_italic"
        android:layout_marginTop="10dp">
    </TextView>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:id="@+id/price"
        android:textStyle="bold"
        android:layout_below="@+id/name"
        android:layout_marginTop="10dp">
    </TextView>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:layout_centerHorizontal="true"
        android:id="@+id/desc"
        android:textStyle="bold"
        android:layout_below="@+id/price"
        android:layout_marginTop="10dp">
</TextView>
<Button
    android:id="@+id/speak"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_below="@+id/desc"
    android:layout_marginTop="10dp"
    android:text="SPEAK"
    app:cornerRadius="10dp">
</Button>
<Button
    android:id="@+id/cart"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/speak"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="10dp"
    android:text="ADD TO CART"
    app:cornerRadius="10dp">
</Button>

<Button
    android:id="@+id/order"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/cart"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="10dp"
    android:text="ORDER NOW"
    app:cornerRadius="10dp">
</Button>
</RelativeLayout>
```

This is a xml file for the details page which consists of details of the pet. It also consists of three buttons namely 'speak', 'add to cart' and 'order now'. On pressing the speak button text-to-feature will be implemented. Users can add the product to cart by pressing the add to cart button and they can also order the product by pressing the order now button.

3.2.28 activity_home.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".Login"
    android:background="#4DD0E1"

    xmlns:android="http://schemas.android.com/apk/res/android">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recycleview"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:listitem="@layout/sample" />
</RelativeLayout>
```

This is the XML file for the home page that displays the list of pets based on the category selected. It makes use of the RecyclerView, which makes it easy to efficiently display large sets of data. RecyclerView library dynamically creates the elements when they're needed. RecyclerView recycles those individual elements. When an item scrolls off the screen, RecyclerView doesn't destroy its view. Instead, RecyclerView reuses the view for new items that have scrolled on screen.

3.2.29 activity_my_cart.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".Login"
    android:background="#4DD0E1"

    xmlns:android="http://schemas.android.com/apk/res/android">

    <androidx.recyclerview.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/recycleview"
        tools:listitem="@layout/sample"/>
</RelativeLayout>
```

This is the XML file for my cart page that displays the list of items in the cart. It is created using the RecyclerView.

3.2.30 activity_orders.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".Login"
    android:background="#4DD0E1"

    xmlns:android="http://schemas.android.com/apk/res/android">

    <androidx.recyclerview.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/recycleview"
        tools:listitem="@layout/sample"/>
</RelativeLayout>
```

This is the XML file for orders page that displays the list of items ordered by the user. It is created using the RecyclerView.

3.2.31 activity_place_order.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/teal_200"
    tools:context=".PlaceOrder">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:id="@+id/image">
    </ImageView>
    <EditText
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_below="@id/image"
        android:layout_marginTop="10dp"
        android:id="@+id/address"
        android:hint="Enter valid Address"
        android:textColorHint="#FFFFFF">
</EditText>
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_below="@id/address"
    android:layout_marginTop="10dp"
    android:id="@+id/phone"
    android:hint="Enter valid contact number"
    android:inputType="phone"
    android:textColorHint="#FBFBFB">
</EditText>
<Button
    android:id="@+id/order"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/phone"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"
    android:text="ORDER NOW"
    app:cornerRadius="10dp">
</Button>
</RelativeLayout>
```

This is a xml file for the place order page where the users can enter the order details. It consists of two edit text fields to enter the address and phone number. It also consists of a button i.e., order now, pressing which the order will be placed.

3.2.32 activity_view_cart.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Detail">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:id="@+id/image">
    </ImageView>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:id="@+id/name"
        android:textStyle="bold"
        android:textSize="30sp"
        android:layout_below="@+id/image"
        android:layout_marginTop="20dp">
    </TextView>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:id="@+id/price"
        android:textStyle="bold"
        android:layout_below="@+id/name"
        android:layout_marginTop="20dp">
    </TextView>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:id="@+id/desc"
        android:textStyle="bold"
        android:layout_below="@+id/price"
        android:layout_marginTop="20dp">
    </TextView>

    <Button
```

```
        android:id="@+id/cart"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/desc"
        android:layout_marginLeft="5dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp"
        android:text="Remove From Cart"
        app:cornerRadius="10dp">
    </Button>

    <Button
        android:id="@+id/order"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/cart"
        android:layout_centerHorizontal="true"
        android:layout_marginLeft="5dp"
        android:layout_marginTop="20dp"
        android:text="Order now"
        app:cornerRadius="10dp">
    </Button>
</RelativeLayout>
```

This is a XML file with which the users can view a particular item in their cart. It consists of two buttons namely ‘remove from cart’ and ‘order now’. Using these buttons the user can remove that item from the cart or he can proceed to buy it respectively.

3.2.33 sample.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="144dp"
    android:layout_below="@+id/radiogroup">
    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="9dp"
        android:layout_marginRight="10dp"
```

```
        app:cardCornerRadius="5dp"
        app:cardMaxElevation="5dp">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="match_parent">
            <ImageView
                android:id="@+id/image"
                android:layout_width="190dp"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:src="@drawable/categorycat">
            </ImageView>
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginLeft="5dp"
                android:orientation="vertical">
                <TextView
                    android:id="@+id/name"
                    android:layout_width="176dp"
                    android:layout_height="match_parent"

                    android:fontFamily="@font/amaranth_italic"
                    android:text="Name: cat"
                    android:textSize="20dp"
                    android:textStyle="bold">
                </TextView>
                <TextView
                    android:id="@+id/price"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"

                    android:fontFamily="@font/amaranth_italic"
                    android:text="Price: 1000Rs"
                    android:textSize="15dp"
                    android:textStyle="bold">
                </TextView>
                <TextView
                    android:id="@+id/desc"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"

                    android:fontFamily="@font/amaranth_italic"
                    android:text="Description: "
```

```
        android:textSize="15dp"
        android:textStyle="bold">
    </TextView>
</LinearLayout>
</LinearLayout>
</androidx.cardview.widget.CardView>
</LinearLayout>
```

This is an XML file that contains `CardView` for each item of the list in the home page. These cards have a default elevation above their containing view group, so the system draws shadows below them. Cards provide an easy way to contain a group of views while providing a consistent style for the container. Imageview, and three textview for name, price and description is created here.

3.2.34 sample_order.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="144dp"
    android:layout_below="@+id/radiogroup">
    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="9dp"
        android:layout_marginRight="10dp"
        app:cardCornerRadius="5dp"
        app:cardMaxElevation="5dp">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="match_parent">
            <ImageView
                android:id="@+id/oimage"
                android:layout_width="190dp"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:src="@drawable/goldenretriever">
            </ImageView>
            <LinearLayout
                android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp"
        android:orientation="vertical">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
            <TextView
                android:id="@+id/order"
                android:layout_width="wrap_content"
                android:layout_height="match_parent"

        android:fontFamily="@font/amaranth_italic"
            android:text="Order number: "
            android:textSize="20dp"
            android:textStyle="bold">
        </TextView>
        <TextView
            android:id="@+id/orderno"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"

        android:fontFamily="@font/amaranth_italic"
            android:text="1"
            android:textSize="20dp"
            android:textStyle="bold">
        </TextView>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <TextView
            android:id="@+id/addressname"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

        android:fontFamily="@font/amaranth_italic"
            android:text="Address: "
            android:textSize="15dp"
            android:textStyle="bold">
        </TextView>
        <TextView
            android:id="@+id/address"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:fontFamily="@font/amaranth_italic"
        android:text="Address"
        android:textSize="15dp"
        android:textStyle="bold">
    </TextView>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/phonename"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:fontFamily="@font/amaranth_italic"
        android:text="contact: "
        android:textSize="15dp"
        android:textStyle="bold">
    </TextView>
    <TextView
        android:id="@+id/phone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:fontFamily="@font/amaranth_italic"
        android:text="contact"
        android:textSize="15dp"
        android:textStyle="bold">
    </TextView>
</LinearLayout>
</LinearLayout>
</LinearLayout>
</androidx.cardview.widget.CardView>
</LinearLayout>
```

This is a XML file that contains CardView for each item of the list in the orders page. It consists of details such as image of the pet, order number, delivery address and contact details.

Chapter 4

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.

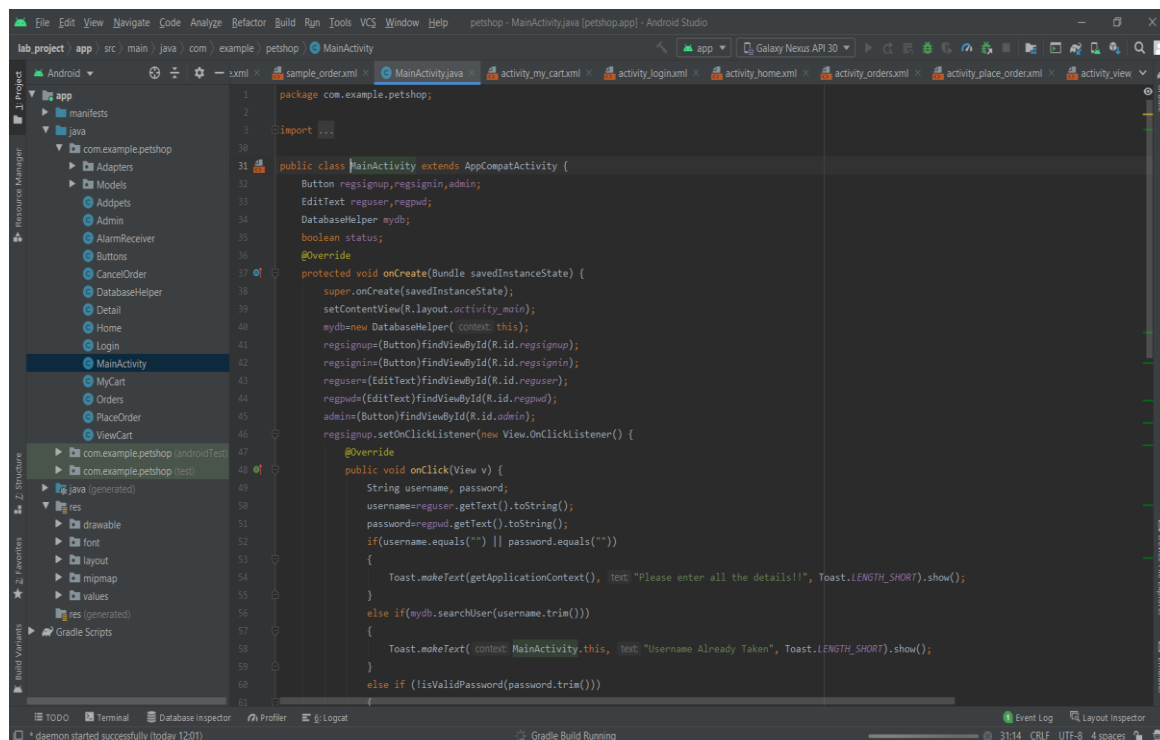
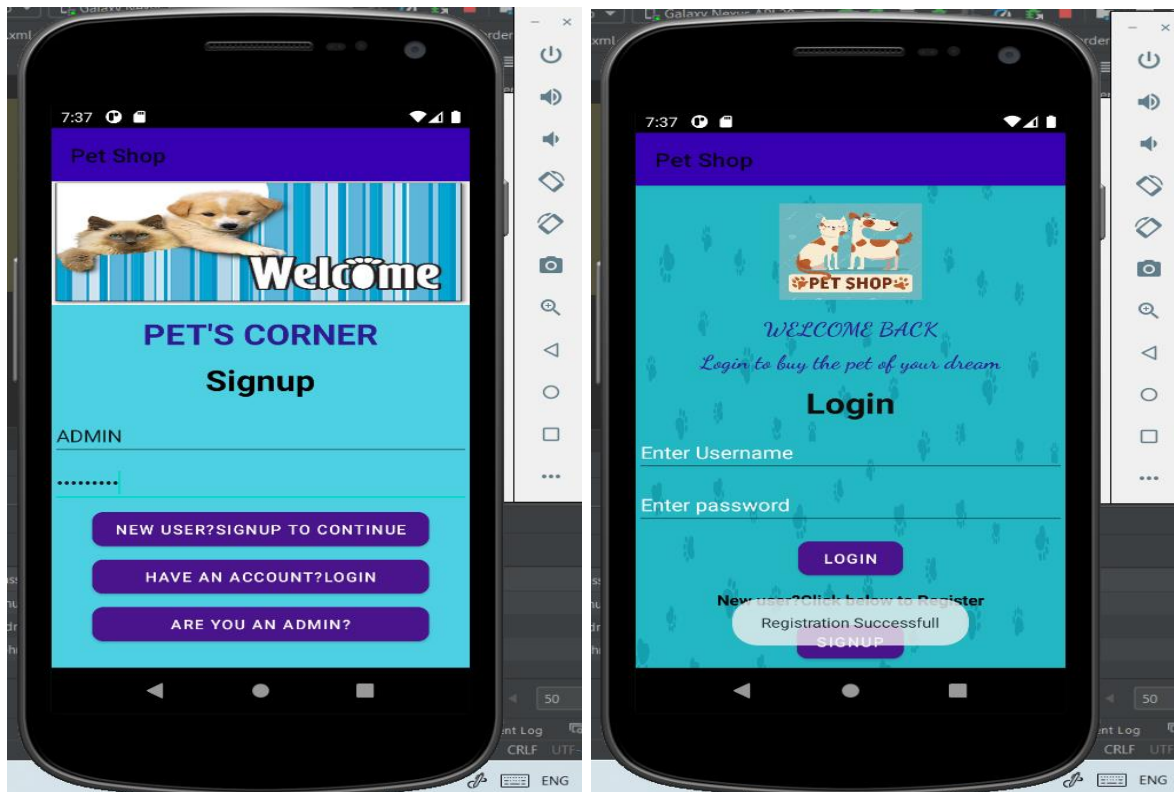


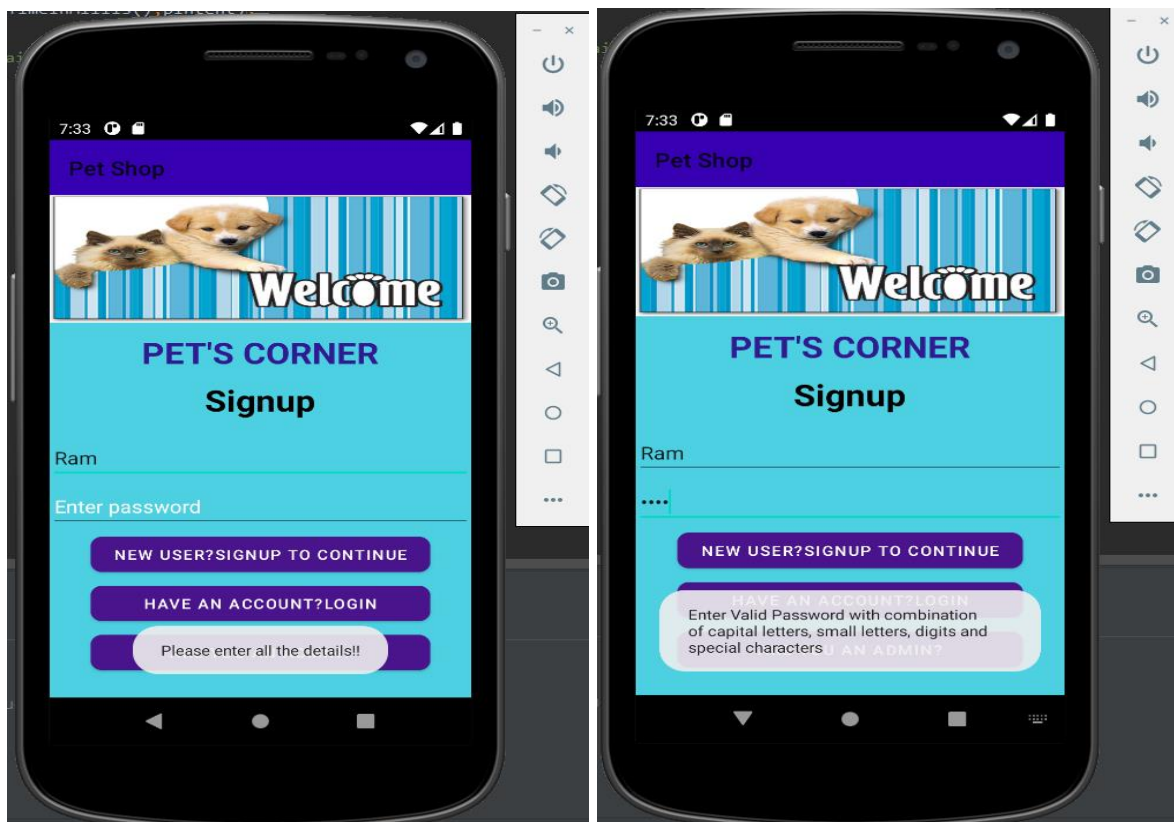
Figure 4.1: Running application code

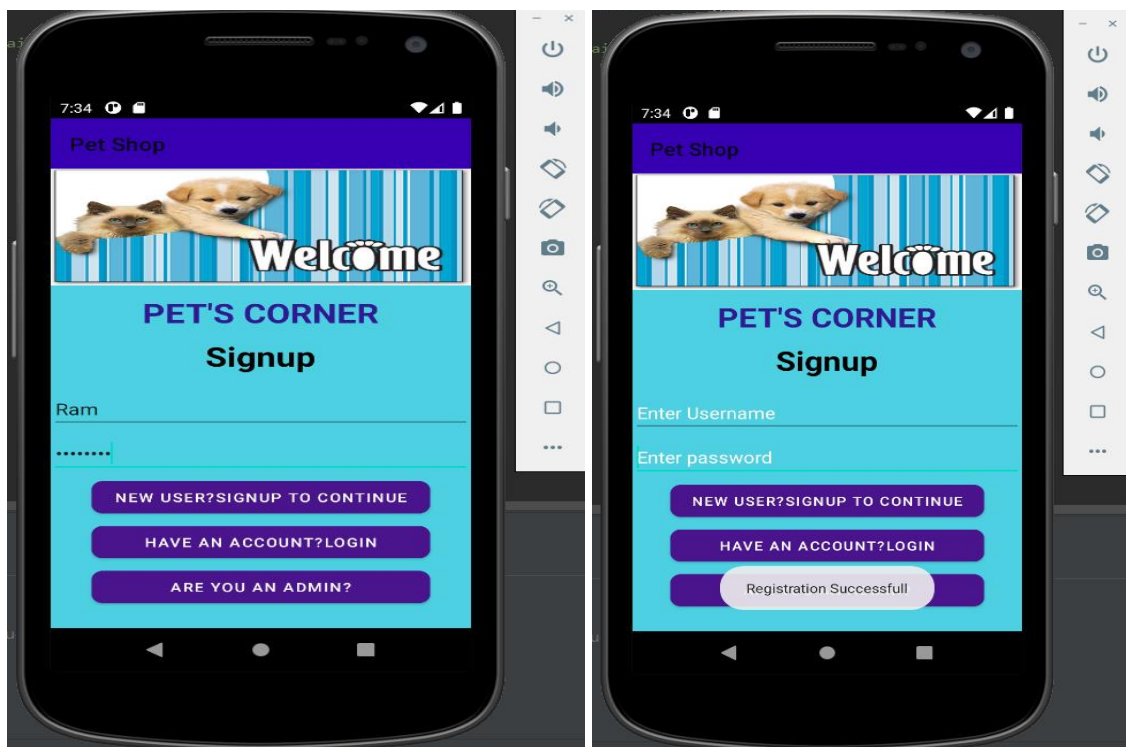
4.2 Registration Page

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



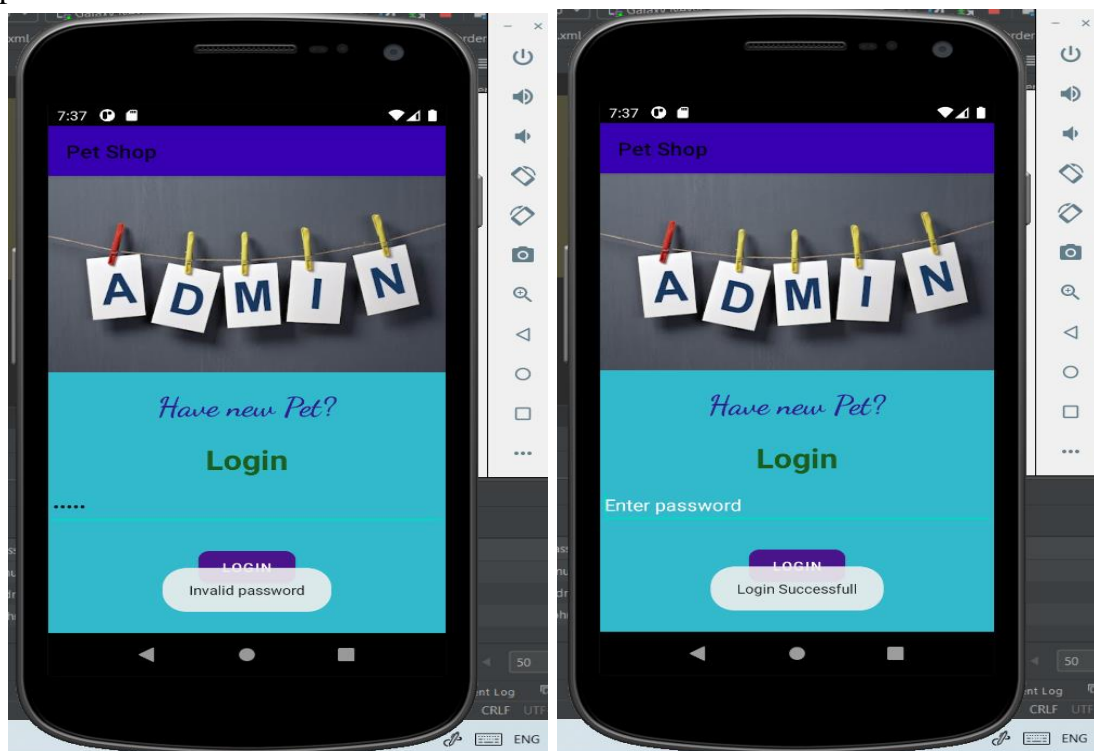
4.2.2 User Registration: User registers himself with the application.



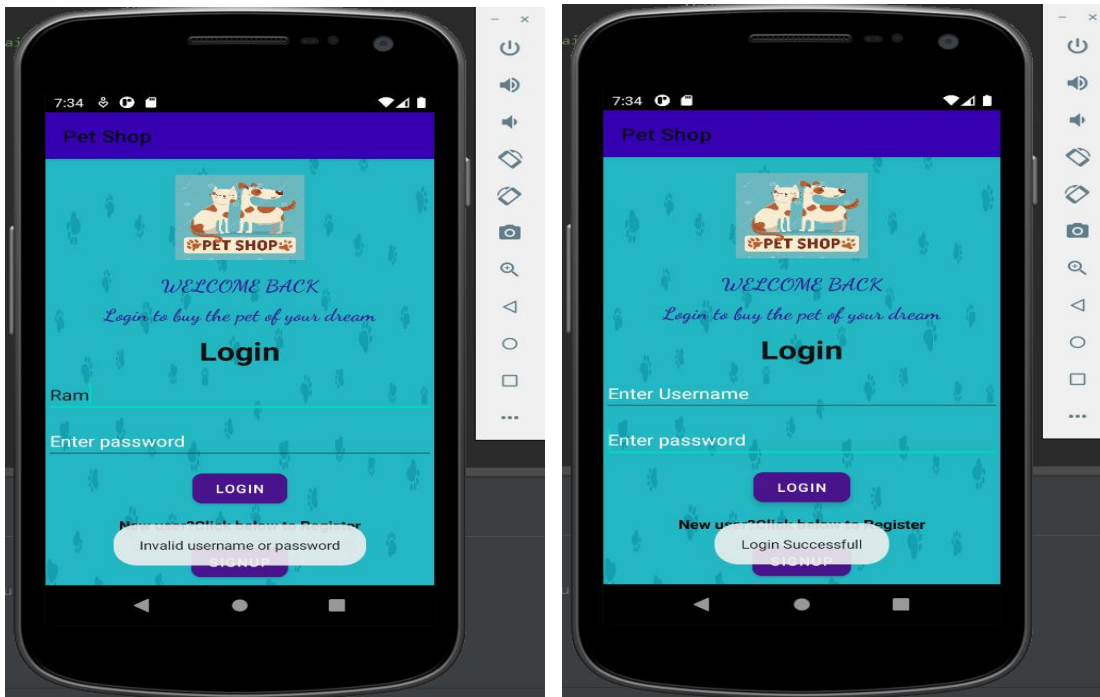


4.3 Login Page

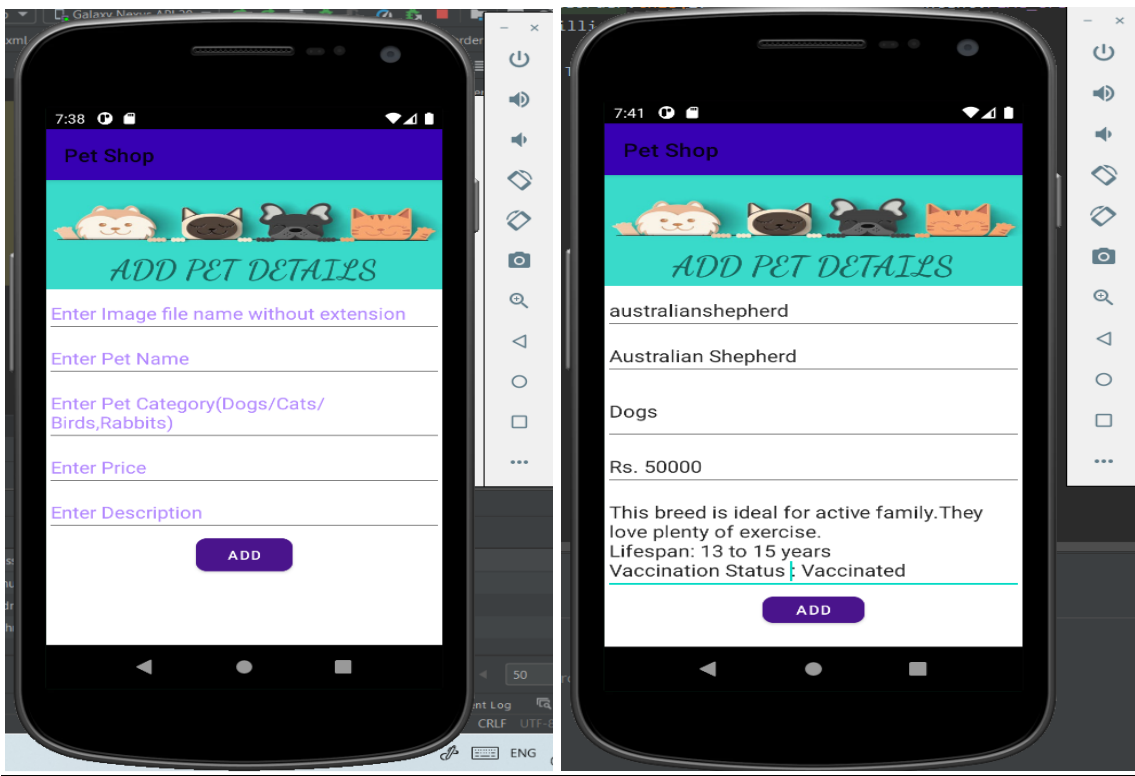
4.3.1 Admin Login: ADMIN logs in to the application by entering the ADMIN password.

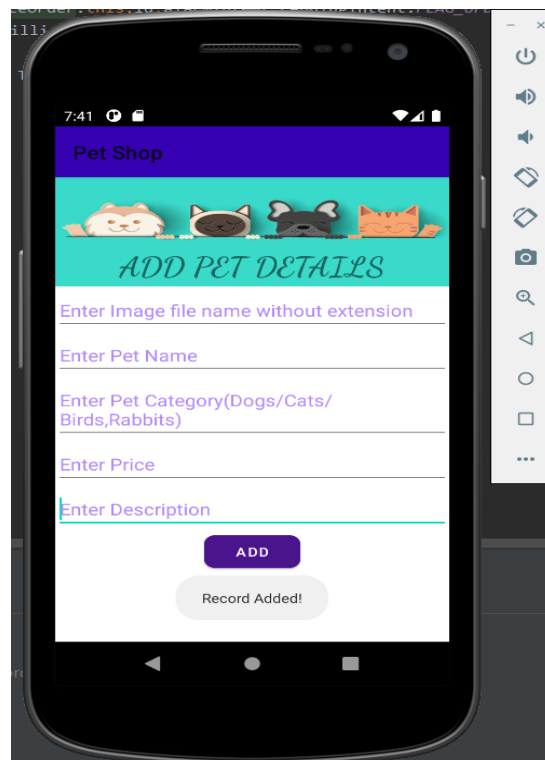


4.3.2 User login: User logs in to the application with the registered username and password.

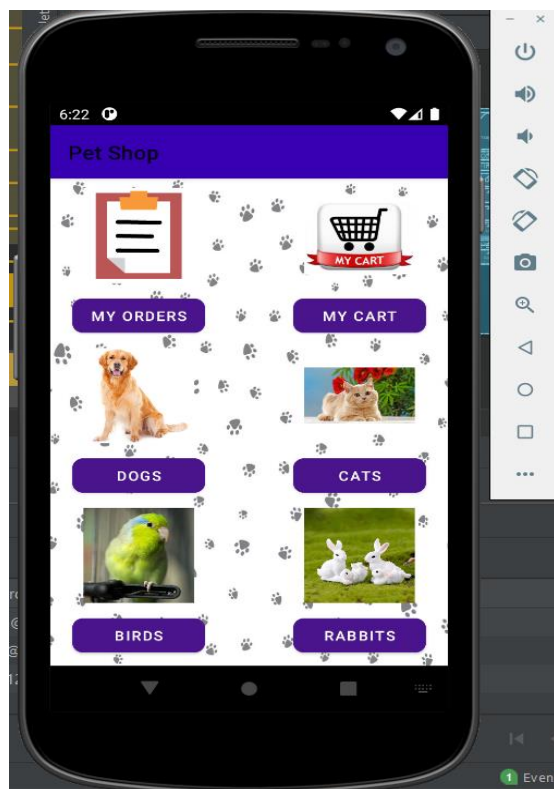


4.4 Add Pet Details Page: After ADMIN login, he is allowed to enter new available pets to the application.

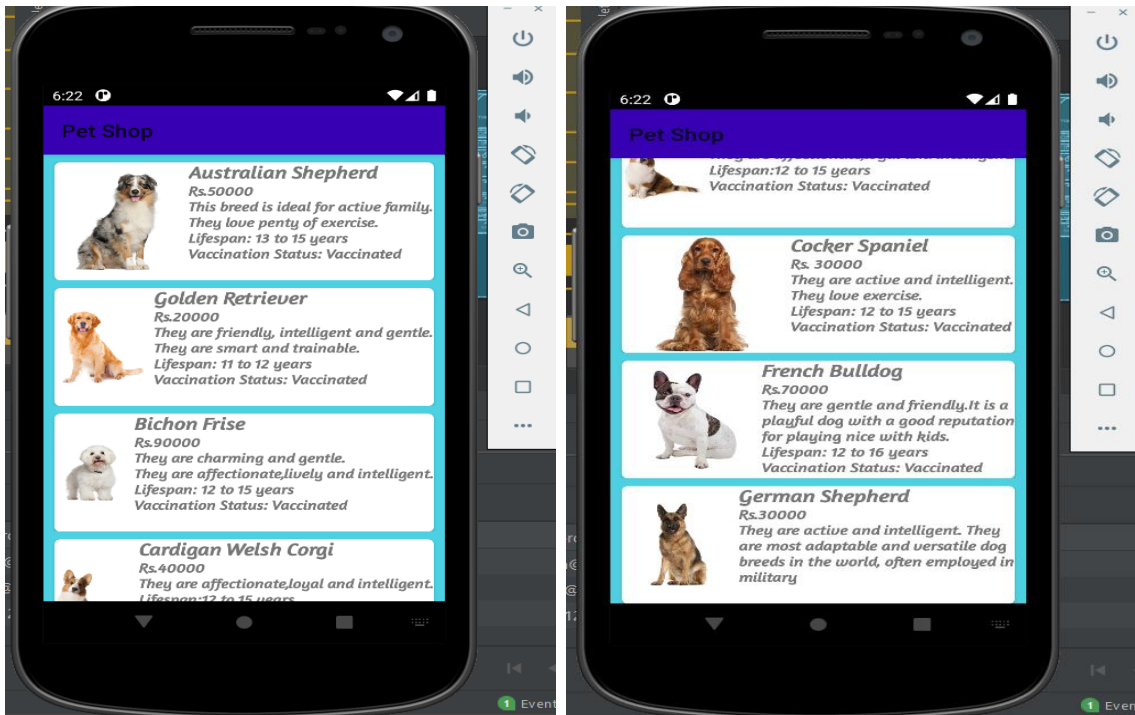




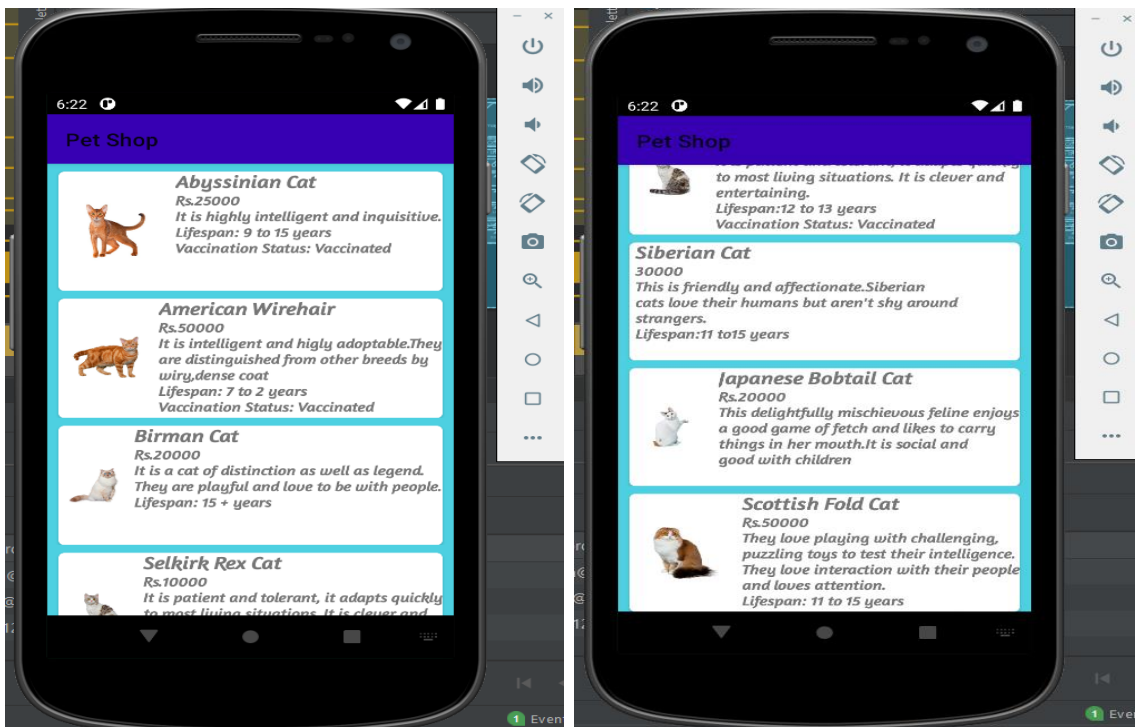
4.5 Categories/Buttons Page: Four categories of pets are available, with 2 extra buttons to navigate to My Orders section and My Carts section.



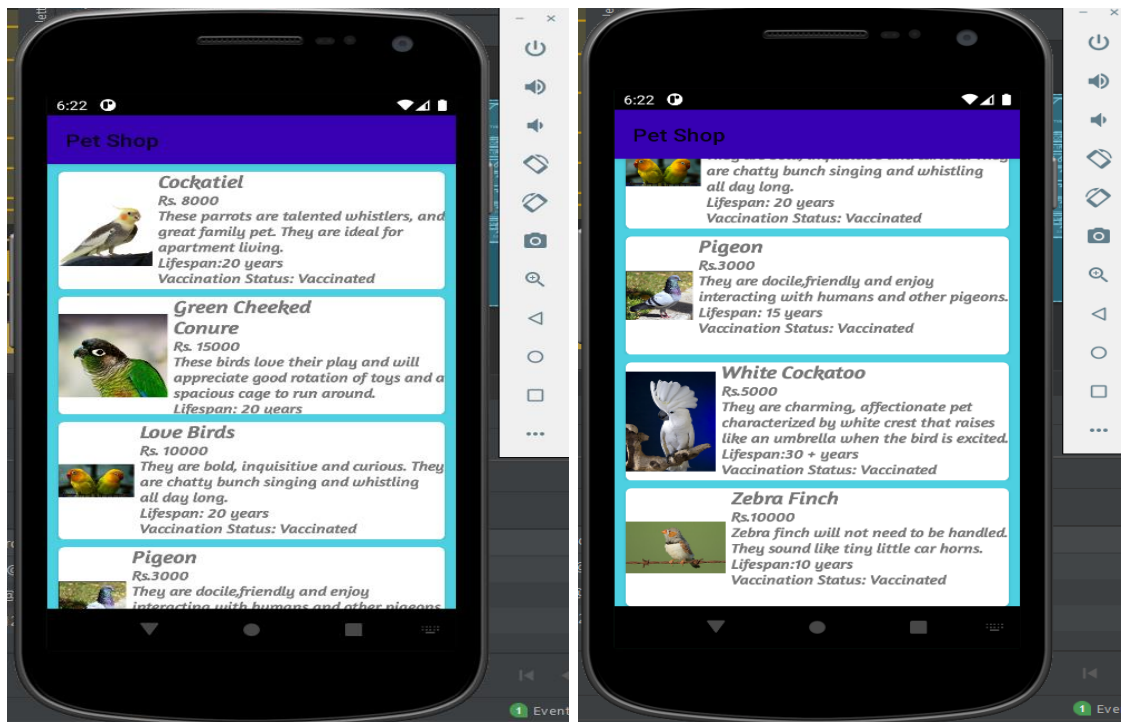
4.6 Dogs: When clicked on Dogs category, list of available dogs is displayed in a list view.



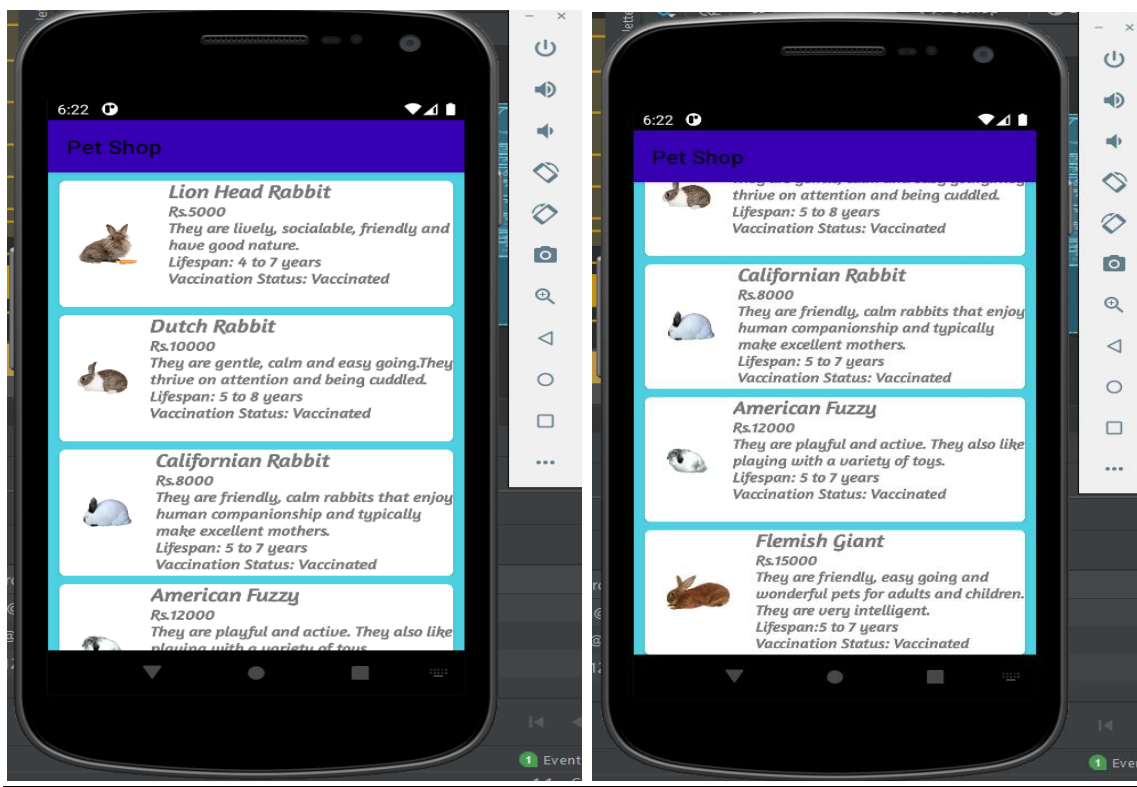
4.7 Cats: When clicked on Cats category, list of available cats is displayed in a list view.



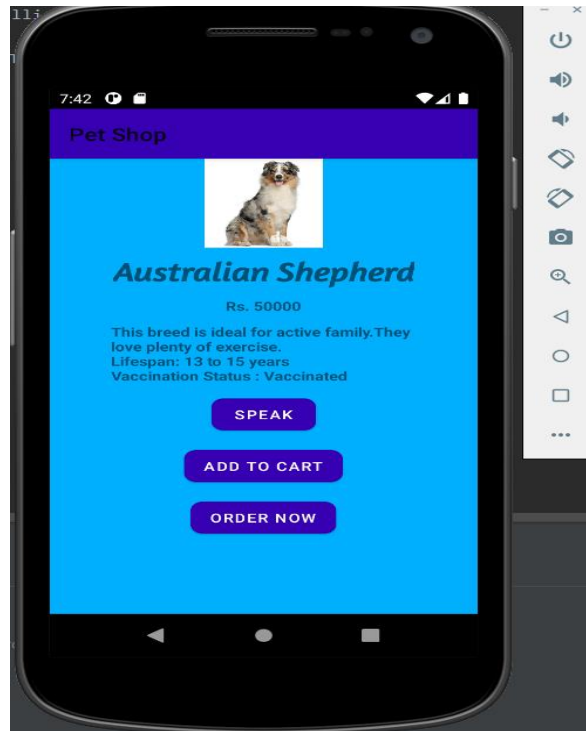
4.8 Birds : When clicked on Birds category, list of available birds is displayed in a list view.



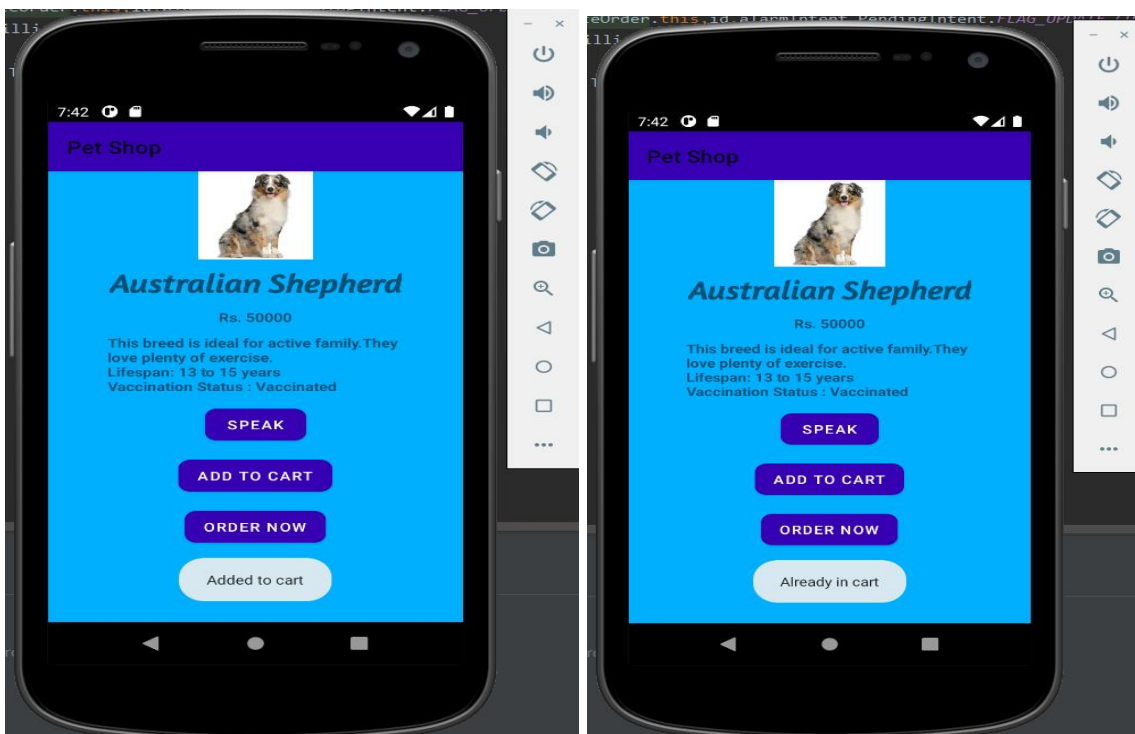
4.9 Rabbits: When clicked on Rabbits category, list of available rabbits is displayed in a list view.



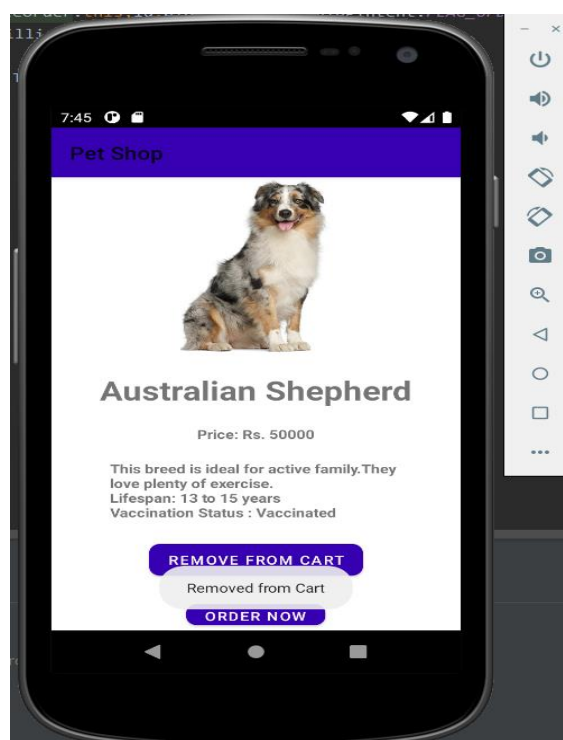
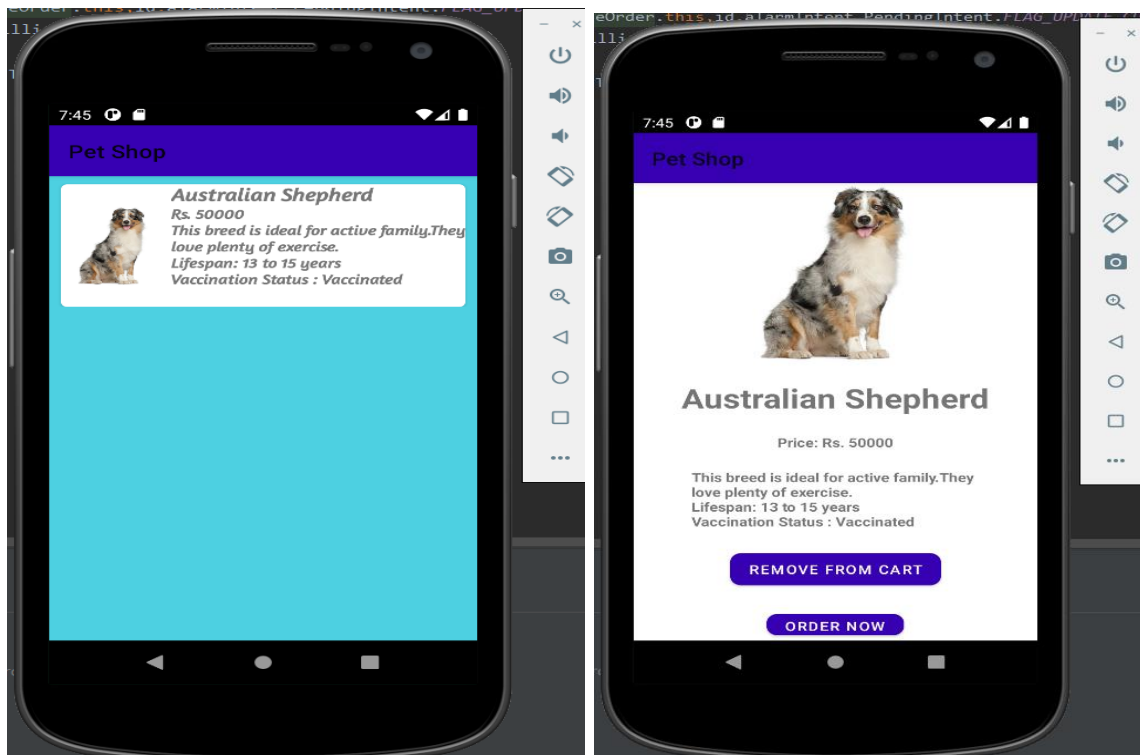
4.10 Details Page: When clicked on any of the pet, under any section, details of the selected pet is displayed with 'text to speech' button to read out the pet details, 'Add to cart' button to add the pet to cart and 'Order Now' button to place the order.



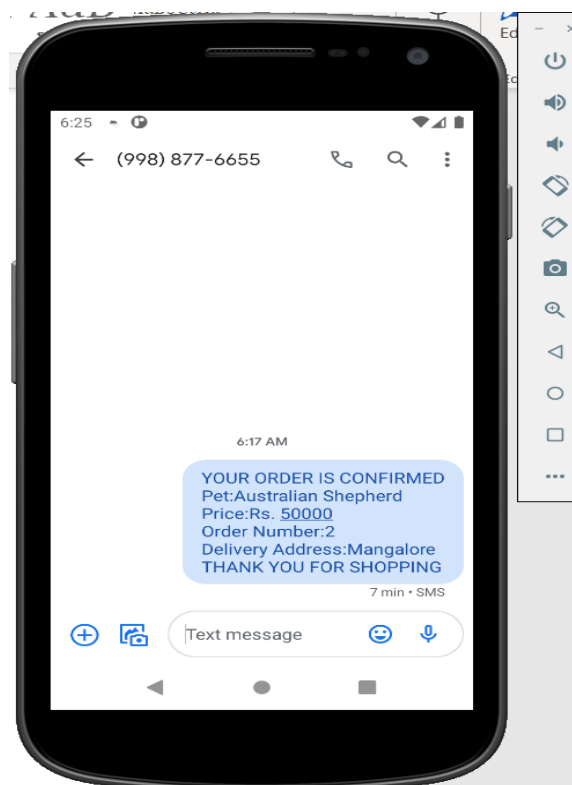
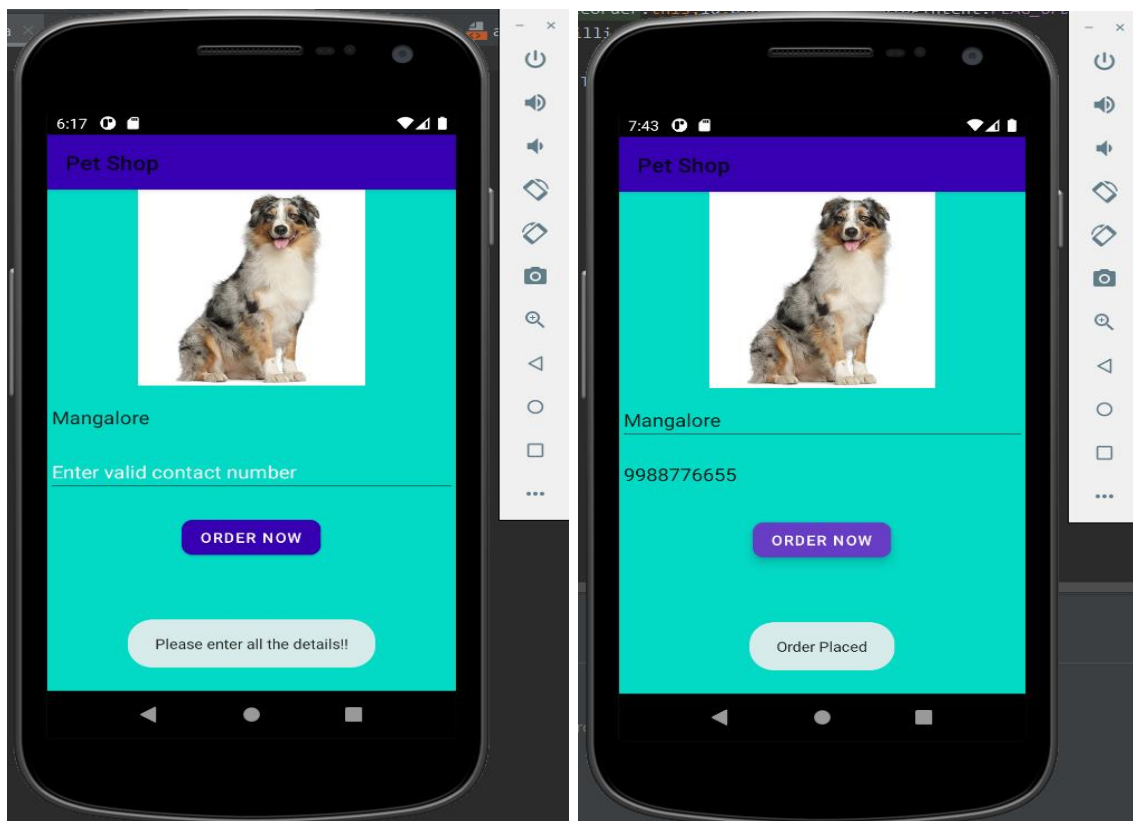
4.11 Add to Cart: To add the pet to the Cart, if already present, respective message is displayed.



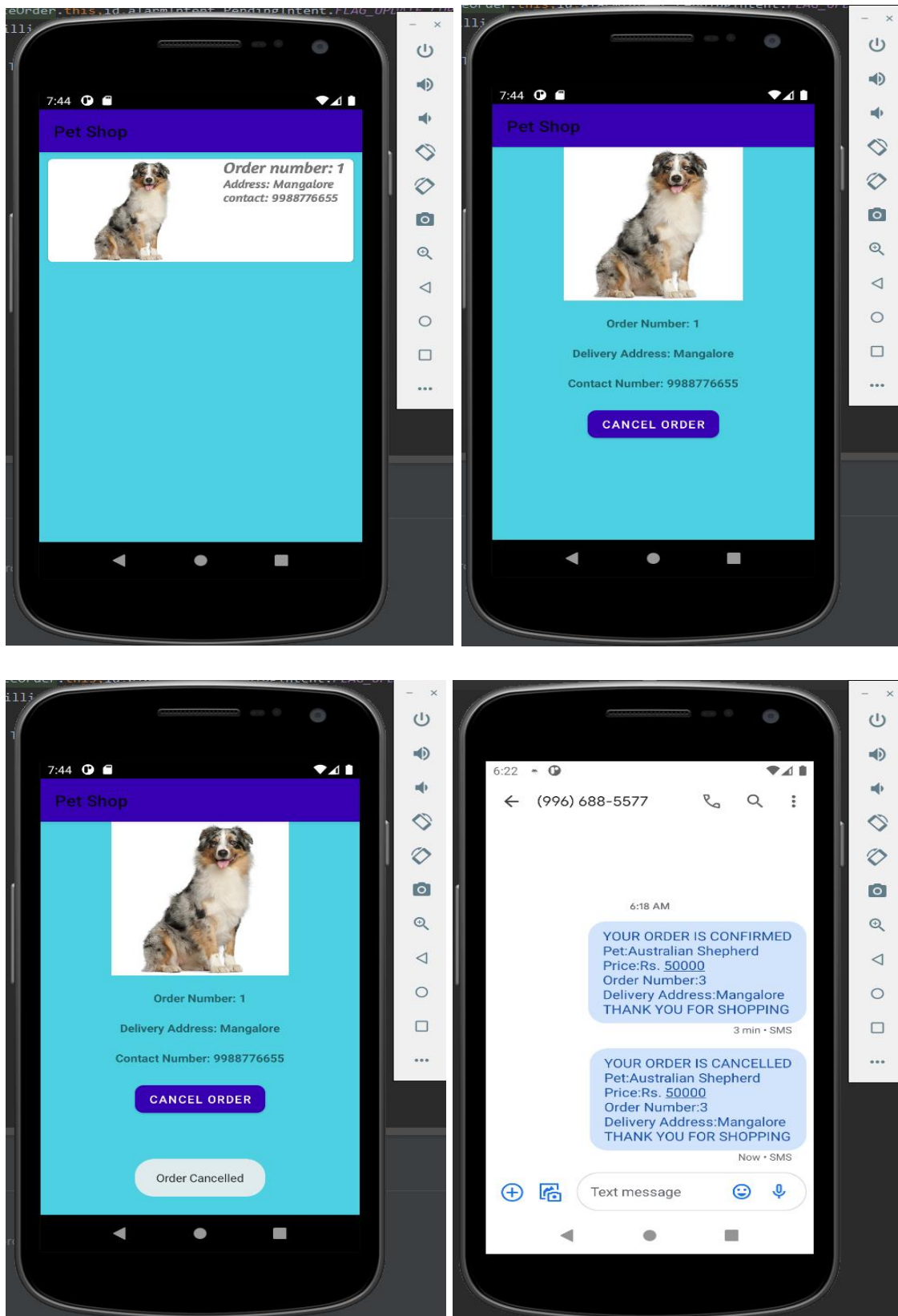
4.12 View Cart: View all the pets in the cart. When clicked on the item in the cart, pet details with 2 buttons, 'Remove from cart' to remove the item from the cart and 'order now' are displayed on the next page.



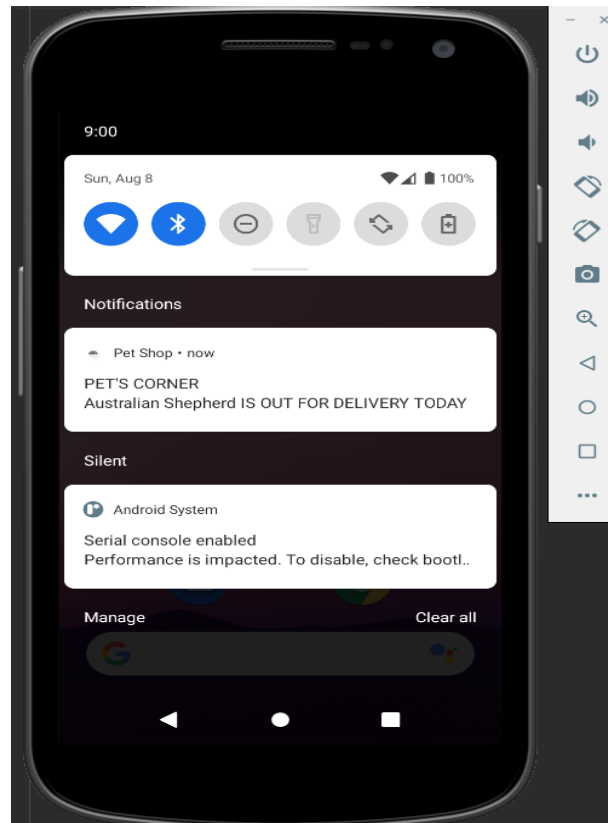
4.13 Order Now Page: To order the pet, address and phone number is being asked. SMS with order details is sent to the given phone number.



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.



Chapter 5

References

- [1] <https://www.geeksforgeeks.org/>
- [2] <http://www.stackoverflow.com>.
- [3] <https://www.youtube.com/watch?v=Le47R9H3qow>
- [4] <https://www.youtube.com/watch?v=JkblErXt1r4&t=2772s>