**MainActivity.java**

package com.example.yashmistrymealdelivery;

import androidx.appcompat.app.AppCompatActivity;

import android.app.DatePickerDialog;

import android.os.Bundle;

import android.text.InputType;

import android.util.Log;

import android.view.View;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.DatePicker;

import android.widget.RadioGroup;

import android.widget.Toast;

import com.example.yashmistrymealdelivery.databinding.ActivityMainBinding;

import com.example.yashmistrymealdelivery.models.MealCost;

import com.google.android.material.snackbar.Snackbar;

import java.util.Calendar;

public class MainActivity extends AppCompatActivity implements AdapterView.OnItemSelectedListener, AdapterView.OnItemClickListener, View.OnClickListener, RadioGroup.OnCheckedChangeListener {

// Variable initialization

String subcriptionPlanType;

double cookedMealCost;

double unCookedMealCost;

double veggiesWithRecipeCost;

double veggiesOnlyCost;

double lemonade3L;

double milk4L;

double planTypeCost;

double typeOfDeliveryCost;

String typeOfDelivery;

String delivery ;

Object plan;

String PlanType;

double totalCost = 0;

// object initialization

ActivityMainBinding binding;

DatePickerDialog datePicker;

MealCost mCost;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

binding=ActivityMainBinding.inflate(getLayoutInflater());

View view = binding.getRoot();

setContentView(view);

// types of subscription radio-button default value

binding.rdMonthly.setChecked(false);

binding.rdYearly.setChecked(false);

// check-box value

binding.chkUpdates.setChecked(false);

binding.chkDesigns.setChecked(false);

// type of delivery spinner adapter

ArrayAdapter<CharSequence> typeOfDeliveryAdapter = ArrayAdapter.createFromResource(this,R.array.type\_of\_delivery, androidx.appcompat.R.layout.support\_simple\_spinner\_dropdown\_item);

binding.spnDesignTypes.setAdapter(typeOfDeliveryAdapter);

// disabling keyboard

binding.edtDate.setInputType(InputType.TYPE\_NULL);

setListeners();

}

// displaying selected item and setting flavour cost

@Override

public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {

delivery = binding.spnDesignTypes.getItemAtPosition(i).toString();

if (binding.rgSubsPlan.getCheckedRadioButtonId() == R.id.rdMonthly) {

if (delivery.equals("Store pick-up")){

typeOfDelivery = "Store pick-up";

typeOfDeliveryCost = 0;

} else if (delivery.equals("Community box")){

typeOfDelivery = "Community box";

typeOfDeliveryCost = 5.0;

} else if (delivery.equals("Door Step")){

typeOfDelivery = "Door Step";

typeOfDeliveryCost = 4.0;

} else {

typeOfDeliveryCost = 0;

}

Log.d("MonthlyDeliveryType", "Type of Delivery: Monthly :" + typeOfDeliveryCost);

} else if (binding.rgSubsPlan.getCheckedRadioButtonId() == R.id.rdYearly) {

if (delivery.equals("Store pick-up")){

typeOfDelivery = "Store pick-up";

typeOfDeliveryCost = 0;

} else if (delivery.equals("Community box")){

typeOfDelivery = "Community box";

typeOfDeliveryCost = 22.0;

} else if (delivery.equals("Door Step")){

typeOfDelivery = "Door Step";

typeOfDeliveryCost = 42.0;

} else {

typeOfDeliveryCost = 0;

}

Log.d("YearDeliveryType", "Type of Delivery: Yearly :" + typeOfDeliveryCost);

} else{

typeOfDeliveryCost = 0;

}

plan = binding.spnPlanType.getItemAtPosition(i);

if (plan != null){

plan = plan.toString();

if (plan.equals("Cooked Meal")){

PlanType = "Cooked Meal";

planTypeCost = 22.0;

} else if (plan.equals("Uncooked Meal")){

PlanType = "Uncooked Meal";

planTypeCost = 17.0;

} else if (plan.equals("Veggies with Recipe")){

PlanType = "Veggies with Recipe";

planTypeCost = 110.0;

} else if (plan.equals("Veggies Only")){

PlanType = "Veggies Only";

planTypeCost = 164.0;

} else {

PlanType = "error";

planTypeCost = 0;

}

Log.d("PlanType", "Plan Type Cost:" + plan + " " + planTypeCost);

} else {

plan = "error";

}

}

@Override

public void onNothingSelected(AdapterView<?> adapterView) {

}

// Events

private void setListeners() {

// type of subscription plans radio

binding.rgSubsPlan.setOnCheckedChangeListener(this);

// Additionals check-box

binding.chkUpdates.setOnClickListener(this);

binding.chkDesigns.setOnClickListener(this);

// plan type spinner

binding.spnPlanType.setOnItemSelectedListener(this);

// type of delivery spinner

binding.spnDesignTypes.setOnItemSelectedListener(this);

binding.edtDate.setOnClickListener(this);

binding.btnSubmit.setOnClickListener(this);

}

@Override

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

}

@Override

public void onClick(View v) {

// Additional milk/lemonade (check-box)

if (binding.rgSubsPlan.getCheckedRadioButtonId() == R.id.rdMonthly) {

if(v.getId() == binding.chkDesigns.getId()) {

if (binding.chkDesigns.isChecked()) {

lemonade3L = 7.0;

} else {

lemonade3L = 0.0;

}

} else if(v.getId() == binding.chkUpdates.getId()) {

if (binding.chkUpdates.isChecked()) {

milk4L = 6.0;

} else {

milk4L = 0.0;

}

}

} else if(binding.rgSubsPlan.getCheckedRadioButtonId() == R.id.rdYearly) {

if(v.getId() == binding.chkDesigns.getId()) {

if (binding.chkDesigns.isChecked()) {

lemonade3L = 70.0;

} else {

lemonade3L = 0.0;

}

} else if(v.getId() == binding.chkUpdates.getId()) {

if (binding.chkUpdates.isChecked()) {

milk4L = 40.0;

} else {

milk4L = 0.0;

}

}

}

if (v.getId() == R.id.edtDate) {

// date picker

Calendar cal = Calendar.getInstance();

int dayofSales= cal.get(Calendar.DAY\_OF\_MONTH);

int monthofSales =cal.get(Calendar.MONTH);

int yearofSales = cal.get(Calendar. YEAR);

datePicker = new DatePickerDialog(this, new DatePickerDialog.OnDateSetListener() {

// formatting date

@Override

public void onDateSet(DatePicker datePicker, int year, int month, int day) {

binding.edtDate.setText(day+"/"+(month+1)+"/"+year);

}

}, yearofSales, monthofSales, dayofSales);

// disabling future dates

datePicker.getDatePicker().setMinDate(cal.getTimeInMillis());

datePicker.show();

}

else if (v.getId() == R.id.btnSubmit){

// validating required fields

if (formValidated()){

// displaying result

mCost = new MealCost(binding.edtName.getText().toString(), binding.edtPhone.getText().toString(), subcriptionPlanType, PlanType, planTypeCost, milk4L, lemonade3L, typeOfDelivery, typeOfDeliveryCost, binding.edtDate.getText().toString());

String result = mCost.getMealCost();

// Snackbar

Snackbar.make(binding.ltHost, result, Snackbar.LENGTH\_INDEFINITE)

.setAction("Okay", new View.OnClickListener() {

@Override

public void onClick(View view) {

Toast.makeText(getApplicationContext(), "Thanks you for using this application", Toast.LENGTH\_LONG).show();

}

}).show();

}

}

Log.d("Additionals", "Additionals: " + milk4L + " " + lemonade3L);

}

// type of plan (radio-button)

@Override

public void onCheckedChanged(RadioGroup radioGroup, int i) {

// for Monthly/Yearly radio-button

if (binding.rgSubsPlan.getCheckedRadioButtonId() == R.id.rdMonthly) {

subcriptionPlanType = "Monthly";

// binding monthly array to plan type spinner

ArrayAdapter<CharSequence> monthlySubscriptionAdapterSpace = ArrayAdapter.createFromResource(this, R.array.monthly\_subscription, androidx.appcompat.R.layout.support\_simple\_spinner\_dropdown\_item);

binding.spnPlanType.setAdapter(monthlySubscriptionAdapterSpace);

} else {

subcriptionPlanType = "Yearly";

// binding yearly array to plan type spinner

ArrayAdapter<CharSequence> yearlySubscriptionAdapterSpace = ArrayAdapter.createFromResource(this, R.array.yearly\_subscription, androidx.appcompat.R.layout.support\_simple\_spinner\_dropdown\_item);

binding.spnPlanType.setAdapter(yearlySubscriptionAdapterSpace);

}

// setting meal prices - monthly, yearly

if (binding.rgSubsPlan.getCheckedRadioButtonId() == R.id.rdMonthly) {

cookedMealCost = 22.0;

unCookedMealCost = 17.0;

} else {

veggiesWithRecipeCost = 110.0;

veggiesOnlyCost = 164.0;

}

}

// Form validation

private boolean formValidated(){

// name

if (binding.edtName.length() == 0){

binding.edtName.setError("This field is required");

return false;

}

// phone number

if (binding.edtPhone.length() != 10){

binding.edtPhone.setError("Enter valid phone number");

return false;

}

// radio button

if (!binding.rdYearly.isChecked() && !binding.rdMonthly.isChecked()){

Toast.makeText(getApplicationContext(), "Select subscription plan type", Toast.LENGTH\_LONG).show();

binding.rdYearly.setError("Select at least one");

binding.rdMonthly.setError("Select at least one");

return false;

} else {

// clearing error when at least one radio button is checked

binding.rdYearly.setError(null);

binding.rdMonthly.setError(null);

}

// date

if (binding.edtDate.length() == 0){

binding.edtDate.setError("This field is required");

return false;

} else {

binding.edtDate.setError(null);

}

return true;

}

}

**MealCost.java**

package com.example.yashmistrymealdelivery.models;

// class

public class MealCost {

private String customerName;

private String phoneNumber;

private String subcriptionPlanType;

private String PlanType;

private double planTypeCost;

private double lemonade3L;

private double milk4L;

private String typeOfDelivery;

private double typeOfDeliveryCost;

private String salesDate;

// constructor

public MealCost(String customerName, String phoneNumber, String subcriptionPlanType, String PlanType, double planTypeCost, double milk4L, double lemonade3L, String typeOfDelivery, Double typeOfDeliveryCost, String salesDate) {

this.customerName = customerName;

this.phoneNumber = phoneNumber;

this.subcriptionPlanType = subcriptionPlanType;

this.PlanType = PlanType;

this.planTypeCost = planTypeCost;

this.lemonade3L = lemonade3L;

this.milk4L = milk4L;

this.typeOfDelivery = typeOfDelivery;

this.typeOfDeliveryCost = typeOfDeliveryCost;

this.salesDate = salesDate;

}

// method

public String getMealCost() {

double additionalCost = lemonade3L + milk4L;

double subtotal = 0;

double totalCost = 0;

// Calculating total cost

subtotal = planTypeCost + additionalCost + + typeOfDeliveryCost;

totalCost = subtotal;

// Result

return customerName + ", " + PlanType + ", " + salesDate + ", Total: $" + String.format("%.2f", totalCost) + " CAD";

}

}