MainActivity

package com.example.walletcontrol;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuItem;  
  
public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 }  
  
 public boolean onCreateOptionsMenu(Menu opt){  
 getMenuInflater().inflate(R.menu.*menu*,opt);  
 return true;  
 }  
  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
 if (id == R.id.*mnuCuentas*) {  
 Intent cta = new Intent(this, Cuentas.class);  
 startActivity(cta);  
 }else if(id == R.id.*mnuIngresos*) {  
 Intent ing = new Intent(this, Ingresos.class);  
 startActivity(ing);  
 }else if(id == R.id.*mnuGastos*) {  
 Intent gast = new Intent(this, Gastos.class);  
 startActivity(gast);  
 }else if(id == R.id.*mnuReport*) {  
 Intent rep1 = new Intent(this, Reportes.class);  
 startActivity(rep1);  
 }  
 return super.onOptionsItemSelected(item);  
 }  
}

AdminBD

package com.example.walletcontrol;  
  
import android.content.Context;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
import androidx.annotation.Nullable;  
  
  
public class AdminBD extends SQLiteOpenHelper {  
 public AdminBD(@Nullable Context context, @Nullable String name, @Nullable SQLiteDatabase.CursorFactory factory, int version){  
 super(context,name,factory,version);  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 db.execSQL("create table Cuentas (codigo INTEGER PRIMARY KEY AUTOINCREMENT, descripC varchar,descripL varchar,estado boolean,diasdepago int)");  
 db.execSQL("create table Gastos (idgasto INTEGER PRIMARY KEY AUTOINCREMENT , idcuenta int,descripC varchar,monto int, fechapago date,fechareg date,obs varchar)");  
 db.execSQL("create table Ingresos (idingresos INTEGER PRIMARY KEY AUTOINCREMENT, descripC varchar, monto int, fechapago date,fechareg date,obs varchar)");  
  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
  
 }  
}

CuentaGasto

package com.example.walletcontrol;  
  
public class CuentaGasto {  
  
 int codigo;  
 String descripC;  
 String descripL;  
 boolean estado;  
 int diasdepago;  
  
 public CuentaGasto(){}  
  
 public CuentaGasto(int codigo, String descripC, String descripL,boolean estado,int diasdepago) {  
 this.codigo = codigo;  
 this.descripC = descripC;  
 this.descripL=descripL;  
 this.estado=estado;  
 this.diasdepago=diasdepago;  
 }  
 public int getcodigo() {  
 return codigo;  
 }  
  
 public void setcodigo(int codigo) {  
 this.codigo = codigo;  
 }  
  
 public String getDescripC() {  
 return descripC;  
 }  
  
 public void setDescripC(String descripC) {  
 this.descripC = descripC;  
 }  
 public void setDescripL(String descripL) {  
 this.descripL = descripL;  
 }  
 public boolean geestado() {  
 return estado;  
 }  
  
 public void setestado(boolean estado) {  
 this.estado = estado;  
 }  
 public int getdiasdepago() {  
 return diasdepago;  
 }  
  
 public void setdiasdepago(int diasdepago) {  
 this.diasdepago = diasdepago;  
 }  
  
 @Override  
 public String toString() {  
 return descripC;  
 }  
  
  
}

Cuentas

package com.example.walletcontrol;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.ContentValues;  
import android.content.Intent;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.Switch;  
import android.widget.TextView;  
import android.widget.Toast;  
  
  
public class Cuentas extends AppCompatActivity {  
 private EditText ECorta,Ediapago,ELarga,Ecod;  
 private TextView Editerror;  
 private Switch sw;  
 private ImageButton Imgbtn;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_cuentas*);  
 ECorta=findViewById(R.id.*editCorta*);  
 ELarga=findViewById(R.id.*editLarga*);  
 Ediapago=findViewById(R.id.*editmonto*);  
 Editerror=findViewById(R.id.*error*);  
 Ecod=findViewById(R.id.*editcod*);  
 sw=findViewById(R.id.*swEstado*);  
 sw.setChecked(true);  
 Imgbtn=findViewById(R.id.*BtnModif*);  
 Imgbtn.setEnabled(false);  
 }  
 public void Cerrar(View v){  
 finish();  
 }  
  
 public void Grabar(View v){  
 String DesC=ECorta.getText().toString();  
 String DesL=ELarga.getText().toString();  
 String DPago=Ediapago.getText().toString();  
 boolean estado = sw.isChecked();  
 if (!DesC.isEmpty() && !DesL.isEmpty() && !DPago.isEmpty()){  
 String mensaje=Valida1();  
 if (mensaje.isEmpty()) {  
  
 SQLiteDatabase base = abreBase();  
 ContentValues crear = new ContentValues();  
 crear.put("descripC", DesC);  
 crear.put("descripL", DesL);  
 crear.put("diasdepago", DPago);  
 crear.put("estado", estado);  
 base.insert("Cuentas", null, crear);  
 base.close();  
 limpiar(v);  
 Toast.*makeText*(this, "Registro creado!!", Toast.*LENGTH\_LONG*).show();  
 }else{  
 Toast.*makeText*(this, mensaje, Toast.*LENGTH\_LONG*).show();  
 Editerror.setText(mensaje);  
 Editerror.setVisibility(View.*VISIBLE*);  
 }  
 }else{  
 Toast.*makeText*(this, "Debe ingresar todos los campos", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 }  
  
  
  
 public void buscarCuenta(View v){  
 SQLiteDatabase base=abreBase();  
 String desC=ECorta.getText().toString();  
 if (!desC.isEmpty() ){  
 Cursor rsfila=base.rawQuery("Select codigo,descripC ,descripL ,estado ,diasdepago from Cuentas where descripC ='"+desC+"';",null);  
 if(rsfila.moveToFirst()){  
 Ecod.setText(rsfila.getString(0));  
 ECorta.setText(rsfila.getString(1));  
 ELarga.setText(rsfila.getString(2));  
 String aux=rsfila.getString(3);  
 if (aux.equals("1")){  
 sw.setChecked(true);  
 }else{  
 sw.setChecked(false);  
 }  
 Ediapago.setText(rsfila.getString(4));  
 ECorta.setEnabled(false);  
 Imgbtn.setEnabled(true);  
  
 base.close();  
 }else{  
 Toast.*makeText*(this,"registro no existe", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 }  
  
 public void ModificarCuenta(View v){  
 SQLiteDatabase base=abreBase();  
 String DesC=ECorta.getText().toString();  
 String DesL=ELarga.getText().toString();  
 String DPago=Ediapago.getText().toString();  
 String Cod=Ecod.getText().toString();  
 boolean estado = sw.isChecked();  
 if (!DesC.isEmpty() && !DesL.isEmpty() && !DPago.isEmpty()) {  
 String mensaje=Valida2();  
 if (mensaje.isEmpty()) {  
 ContentValues modif = new ContentValues();  
 modif.put("descripC", DesC);  
 modif.put("descripL", DesL);  
 modif.put("diasdepago", DPago);  
 modif.put("estado", estado);  
 modif.put("codigo", Cod);  
 base.update("Cuentas", modif, "codigo=" + Cod, null);  
 base.close();  
 limpiar(v);  
 Toast.*makeText*(this, "Registro fue modificado!!", Toast.*LENGTH\_LONG*).show();  
 }else{  
 Editerror.setText(mensaje);  
 Editerror.setVisibility(View.*VISIBLE*);  
 }  
 } else {  
 Toast.*makeText*(this, "Debe ingresar todos los campos", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 }  
  
  
  
 public String Valida1(){  
 FuncGral f=new FuncGral();  
 String mensaje="";  
 boolean opt=true;  
 if (f.TieneEspacios(ECorta.getText().toString())){  
 mensaje="Descripciópn Corta no debe tener espacios";  
 opt=false;  
 }  
 if(!f.DiaMes(Integer.*parseInt*(Ediapago.getText().toString()))){  
 mensaje=mensaje+"\n Dia de Pago debe ser mayor a 0 y menor a 31";  
 opt=false;  
 }  
 if(ExisteCuenta(ECorta.getText().toString())){  
 Toast.*makeText*(this, "error", Toast.*LENGTH\_SHORT*).show();  
 mensaje=mensaje+"\n Cuenta con esa Descripción, ya existe";  
 opt=false;  
 }  
  
  
 return mensaje;  
 }  
  
 public String Valida2(){  
 FuncGral f=new FuncGral();  
 String mensaje="";  
 boolean opt=true;  
 if (f.TieneEspacios(ECorta.getText().toString())){  
 mensaje="Descripciópn Corta no debe tener espacios";  
 opt=false;  
 }  
 if(!f.DiaMes(Integer.*parseInt*(Ediapago.getText().toString()))){  
 mensaje=mensaje+"\n Dia de Pago debe ser mayor a 0 y menor a 31";  
 opt=false;  
 }  
 return mensaje;  
 }  
  
  
 public boolean ExisteCuenta(String cadena){  
 SQLiteDatabase base=abreBase();  
 boolean aux=false;  
 if (!cadena.isEmpty() ){  
 Cursor rsfila=base.rawQuery("Select codigo from Cuentas where descripC='"+cadena+"';",null);  
 if(rsfila.moveToFirst()){  
 Ecod.setText(rsfila.getString(0));  
 base.close();  
 aux= true;  
 }else{  
 aux= false;  
 }  
 }  
 return aux;  
 }  
  
  
 public SQLiteDatabase abreBase(){  
 AdminBD admin= new AdminBD(this,"bd\_test",null,1);  
 SQLiteDatabase base=admin.getWritableDatabase();  
 return base;  
 }  
  
 public void limpiar(View v) {  
 ECorta.setText("");  
 ELarga.setText("");  
 Ediapago.setText("");  
 sw.setChecked(true);  
 Imgbtn.setEnabled(false);  
 ECorta.setEnabled(true);  
 Editerror.setText("");  
 Editerror.setVisibility(View.*INVISIBLE*);  
 }  
  
  
 public void ListaCuentas(View v){  
 Intent lstcta = new Intent(this, ListCuentas.class);  
 startActivity(lstcta);  
 }  
}

FuncGral

package com.example.walletcontrol;  
  
import java.text.SimpleDateFormat;  
import java.util.Calendar;  
import java.util.Date;  
  
public class FuncGral {  
  
 public boolean TieneEspacios(String cadena){  
 int cantidadDeEspacios = 0;  
 for (int i = 0; i < cadena.length(); i++) {  
 if (cadena.charAt(i) == ' ') cantidadDeEspacios++;  
 }  
 if (cantidadDeEspacios>0){  
 return true;  
 }else{  
 return false;  
 }  
 }  
  
 public boolean DiaMes(int dia){  
 if (dia<30 && dia>0){  
 return true;  
 }else{  
 return false;  
 }  
 }  
  
 public String FechaActual(){  
 Date c = Calendar.*getInstance*().getTime();  
 SimpleDateFormat df = new SimpleDateFormat("yyyy/MM/dd");  
 String formattedDate = df.format(c);  
 return formattedDate;  
 }  
}

Gastos

package com.example.walletcontrol;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.DatePickerDialog;  
import android.content.ContentValues;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.ArrayAdapter;  
import android.widget.DatePicker;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.Spinner;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.Locale;  
  
public class Gastos extends AppCompatActivity {  
 private Spinner gast;  
 private EditText ECorta, EMonto, EFecha, EObs, Ecod;  
 private TextView Editerror;  
 private ImageButton ImgbtnM,ImgbtnD;  
 ArrayList<String> listaGastos;  
 ArrayList<CuentaGasto> GastosList;  
  
 Calendar calendario = Calendar.*getInstance*();  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_gastos*);  
 gast = findViewById(R.id.*spCuentas*);  
 consultarListaGastos();  
 ArrayAdapter<String> adaptador = new ArrayAdapter<>(this, android.R.layout.*simple\_spinner\_item*, listaGastos);  
  
 gast.setAdapter(adaptador);  
  
 ECorta=findViewById(R.id.*editCorta*);  
 EMonto=findViewById(R.id.*editmonto*);  
 EFecha=findViewById(R.id.*editFecha*);  
 EObs=findViewById(R.id.*editObs*);  
 Editerror=findViewById(R.id.*error2*);  
 Ecod=findViewById(R.id.*editcod*);  
  
 ImgbtnM=findViewById(R.id.*BtnModif*);  
 ImgbtnM.setEnabled(false);  
 ImgbtnD=findViewById(R.id.*BtnBorrar*);  
 ImgbtnD.setEnabled(false);  
  
  
  
  
 final Calendar calendario = Calendar.*getInstance*();  
 int anio = calendario.get(Calendar.*YEAR*);  
 int mes = calendario.get(Calendar.*MONTH*);  
 int diaDelMes = calendario.get(Calendar.*DAY\_OF\_MONTH*);  
  
 EFecha.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 new DatePickerDialog(Gastos.this, date, calendario  
 .get(Calendar.*YEAR*), calendario.get(Calendar.*MONTH*),  
 calendario.get(Calendar.*DAY\_OF\_MONTH*)).show();  
 }  
 });  
 }  
  
 DatePickerDialog.OnDateSetListener date = new DatePickerDialog.OnDateSetListener() {  
  
 @Override  
 public void onDateSet(DatePicker view, int year, int monthOfYear,  
 int dayOfMonth) {  
 // *TODO Auto-generated method stub*  
calendario.set(Calendar.*YEAR*, year);  
 calendario.set(Calendar.*MONTH*, monthOfYear);  
 calendario.set(Calendar.*DAY\_OF\_MONTH*, dayOfMonth);  
 actualizarInput();  
 }  
 };  
  
  
 private void actualizarInput() {  
 String formatoDeFecha = "yyyy-MM-dd"; //In which you need put here  
 SimpleDateFormat sdf = new SimpleDateFormat(formatoDeFecha, Locale.*US*);  
  
 EFecha.setText(sdf.format(calendario.getTime()));  
 }  
  
  
 public void Cerrar(View v) {  
 finish();  
 }  
  
 public SQLiteDatabase abreBase() {  
 AdminBD admin = new AdminBD(this, "bd\_test", null, 1);  
 SQLiteDatabase base = admin.getWritableDatabase();  
 return base;  
 }  
  
  
 public void consultarListaGastos() {  
 abreBase();  
 CuentaGasto p1 = null;  
 GastosList = new ArrayList<CuentaGasto>();  
 Cursor rs = abreBase().rawQuery("select codigo,descripC,descripL from Cuentas where estado=true; ", null);  
 while (rs.moveToNext()) {  
 p1 = new CuentaGasto();  
 p1.setcodigo(rs.getInt(0));  
 p1.setDescripC(rs.getString(1));  
 p1.setDescripL(rs.getString(2));  
 GastosList.add(p1);  
 }  
 obtenerGastos();  
 }  
  
 public void obtenerGastos() {  
 listaGastos = new ArrayList<String>();  
 for (int i = 0; i < GastosList.size(); i++) {  
 listaGastos.add(GastosList.get(i).getcodigo() + ":" + GastosList.get(i).getDescripC());  
 }  
 }  
  
 public void limpiar(View v) {  
 ECorta.setText("");  
 EObs.setText("");  
 EMonto.setText("0");  
 EFecha.setText("");  
 Ecod.setText("");  
 ImgbtnM.setEnabled(false);  
 ImgbtnD.setEnabled(false);  
 ECorta.setEnabled(true);  
 Editerror.setText("");  
 Editerror.setVisibility(View.*INVISIBLE*);  
 }  
  
  
 public String Valida1(){  
 FuncGral f=new FuncGral();  
 String mensaje="";  
 boolean opt=true;  
 if (f.TieneEspacios(ECorta.getText().toString())){  
 mensaje="Descripciópn Corta no debe tener espacios";  
 opt=false;  
 }  
 return mensaje;  
 }  
  
  
 public void Grabar(View v){  
 FuncGral f=new FuncGral();  
 String DesC=ECorta.getText().toString();  
 String Obs=EObs.getText().toString();  
 String Monto=EMonto.getText().toString();  
 String Fecha=EFecha.getText().toString();  
 String Cta[]=gast.getSelectedItem().toString().split(":");  
 if (!DesC.isEmpty() && !Monto.isEmpty() && !Fecha.isEmpty()){  
 String mensaje=Valida1();  
 if (mensaje.isEmpty()) {  
  
 SQLiteDatabase base = abreBase();  
 ContentValues crear = new ContentValues();  
 crear.put("descripC", DesC);  
 crear.put("idcuenta",Cta[0] );  
 crear.put("monto", Monto);  
 crear.put("fechapago", Fecha);  
 crear.put("obs", Obs);  
 crear.put("fechareg", f.FechaActual());  
  
 base.insert("Gastos", null, crear);  
 base.close();  
 limpiar(v);  
 Toast.*makeText*(this, "Registro creado!!", Toast.*LENGTH\_LONG*).show();  
 }else{  
 Toast.*makeText*(this, mensaje, Toast.*LENGTH\_LONG*).show();  
 Editerror.setText(mensaje);  
 Editerror.setVisibility(View.*VISIBLE*);  
 }  
 }else{  
 Toast.*makeText*(this, "Debe ingresar todos los campos", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 }  
  
  
 public void BuscarGastos(View v){  
 SQLiteDatabase base=abreBase();  
 String desC=ECorta.getText().toString();  
 if (!desC.isEmpty() ){  
  
 Cursor rsfila=base.rawQuery("Select idgasto,idcuenta,descripC ,monto ,fechapago ,fechareg,obs from Gastos where descripC ='"+desC+"';",null);  
 if(rsfila.moveToFirst()){  
 Ecod.setText(rsfila.getString(0));  
 Seleccionaritem(rsfila.getString(1));  
 ECorta.setText(rsfila.getString(2));  
 EMonto.setText(rsfila.getString(3));  
 EFecha.setText(rsfila.getString(4));  
 EObs.setText(rsfila.getString(6));  
 ECorta.setEnabled(false);  
 ImgbtnM.setEnabled(true);  
 ImgbtnD.setEnabled(true);  
  
 base.close();  
 }else{  
 Toast.*makeText*(this,"registro no existe", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 }  
  
  
 public int BuscarItemGastos(String indice) {  
 int aux=0;  
 for (int i = 0; i < GastosList.size(); i++) {  
 if (GastosList.get(i).getcodigo()==Integer.*parseInt*(indice)){  
 aux=i;  
 }  
 }  
 return aux;  
 }  
 public void Seleccionaritem(String indice){  
 gast.setSelection(BuscarItemGastos(indice));  
 }  
  
  
  
 public void ModificarGastos(View v){  
 SQLiteDatabase base=abreBase();  
 String DesC=ECorta.getText().toString();  
 String Monto=EMonto.getText().toString();  
 String Fecha=EFecha.getText().toString();  
 String Cod=Ecod.getText().toString();  
 String Obs=EObs.getText().toString();  
 String Cta[]=gast.getSelectedItem().toString().split(":");  
  
  
 if (!DesC.isEmpty() && !Monto.isEmpty() && !Fecha.isEmpty()) {  
 String mensaje=Valida1();  
 if (mensaje.isEmpty()) {  
 ContentValues modif = new ContentValues();  
 modif.put("descripC", DesC);  
 modif.put("monto", Monto);  
 modif.put("fechapago", Fecha);  
 modif.put("obs", Obs);  
 modif.put("idcuenta",Cta[0] );  
 // modif.put("codigo", Cod);  
 base.update("Gastos", modif, "idgasto=" + Cod, null);  
 base.close();  
 limpiar(v);  
 Toast.*makeText*(this, "Registro fue modificado!!", Toast.*LENGTH\_LONG*).show();  
 }else{  
 Editerror.setText(mensaje);  
 Editerror.setVisibility(View.*VISIBLE*);  
 }  
 } else {  
 Toast.*makeText*(this, "Debe ingresar todos los campos", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 }  
  
  
 public void EliminarGasto(View v){  
 SQLiteDatabase base;  
 base=abreBase();  
 String codigo=Ecod.getText().toString();  
 if (!codigo.isEmpty() ){  
 base.delete("Gastos","idgasto="+codigo,null);  
 base.close();  
 limpiar(v);  
 Toast.*makeText*(this,"registro eliminado", Toast.*LENGTH\_LONG*).show();  
 }else{  
 Toast.*makeText*(this,"registro no existe", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
  
  
}

Ingresos

package com.example.walletcontrol;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.DatePickerDialog;  
import android.content.ContentValues;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.DatePicker;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.Switch;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.text.SimpleDateFormat;  
import java.util.Calendar;  
import java.util.Locale;  
  
public class Ingresos extends AppCompatActivity {  
 private EditText ECorta,EMonto,EFecha,Ecod,EObs;  
 private TextView Editerror;  
 private ImageButton ImgbtnM,ImgbtnD;  
 Calendar calendario = Calendar.*getInstance*();  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_ingresos*);  
 ECorta=findViewById(R.id.*editCorta*);  
 EMonto=findViewById(R.id.*editmonto*);  
 EFecha=findViewById(R.id.*editFecha*);  
 EObs=findViewById(R.id.*editObs*);  
 Editerror=findViewById(R.id.*error3*);  
 Ecod=findViewById(R.id.*editcod*);  
  
 ImgbtnM=findViewById(R.id.*BtnModif*);  
 ImgbtnM.setEnabled(false);  
 ImgbtnD=findViewById(R.id.*BtnBorrar*);  
 ImgbtnD.setEnabled(false);  
  
  
 final Calendar calendario = Calendar.*getInstance*();  
 int anio = calendario.get(Calendar.*YEAR*);  
 int mes = calendario.get(Calendar.*MONTH*);  
 int diaDelMes = calendario.get(Calendar.*DAY\_OF\_MONTH*);  
  
  
 EFecha.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 new DatePickerDialog(Ingresos.this, date, calendario  
 .get(Calendar.*YEAR*), calendario.get(Calendar.*MONTH*),  
 calendario.get(Calendar.*DAY\_OF\_MONTH*)).show();  
 }  
 });  
 }  
  
 DatePickerDialog.OnDateSetListener date = new DatePickerDialog.OnDateSetListener() {  
  
 @Override  
 public void onDateSet(DatePicker view, int year, int monthOfYear,  
 int dayOfMonth) {  
 // *TODO Auto-generated method stub*  
calendario.set(Calendar.*YEAR*, year);  
 calendario.set(Calendar.*MONTH*, monthOfYear);  
 calendario.set(Calendar.*DAY\_OF\_MONTH*, dayOfMonth);  
 actualizarInput();  
 }  
  
 };  
  
 private void actualizarInput() {  
 String formatoDeFecha = "yyyy-MM-dd"; //In which you need put here  
 SimpleDateFormat sdf = new SimpleDateFormat(formatoDeFecha, Locale.*US*);  
  
 EFecha.setText(sdf.format(calendario.getTime()));  
 }  
  
  
  
  
  
  
  
  
 public void Cerrar(View v){  
 finish();  
 }  
  
 public SQLiteDatabase abreBase(){  
 AdminBD admin= new AdminBD(this,"bd\_test",null,1);  
 SQLiteDatabase base=admin.getWritableDatabase();  
 return base;  
 }  
  
 public void limpiar(View v) {  
 ECorta.setText("");  
 EObs.setText("");  
 EMonto.setText("0");  
 EFecha.setText("");  
 ImgbtnM.setEnabled(false);  
 ImgbtnD.setEnabled(false);  
 ECorta.setEnabled(true);  
 Editerror.setText("");  
 Editerror.setVisibility(View.*INVISIBLE*);  
 }  
  
 public String Valida1(){  
 FuncGral f=new FuncGral();  
 String mensaje="";  
 boolean opt=true;  
 if (f.TieneEspacios(ECorta.getText().toString())){  
 mensaje="Descripciópn Corta no debe tener espacios";  
 opt=false;  
 }  
 if(ExisteIngresos(ECorta.getText().toString())){  
 Toast.*makeText*(this, "error", Toast.*LENGTH\_SHORT*).show();  
 mensaje=mensaje+"\n Cuenta con esa Descripción, ya existe";  
 opt=false;  
 }  
 return mensaje;  
 }  
  
 public void Grabar(View v){  
 FuncGral f=new FuncGral();  
 String DesC=ECorta.getText().toString();  
 String Obs=EObs.getText().toString();  
 String Monto=EMonto.getText().toString();  
 String Fecha=EFecha.getText().toString();  
  
 if (!DesC.isEmpty() && !Monto.isEmpty() && !Fecha.isEmpty()){  
 String mensaje=Valida1();  
 if (mensaje.isEmpty()) {  
  
 SQLiteDatabase base = abreBase();  
 ContentValues crear = new ContentValues();  
 crear.put("descripC", DesC);  
 crear.put("monto", Monto);  
 crear.put("fechapago", Fecha);  
 crear.put("obs", Obs);  
 crear.put("fechareg", f.FechaActual());  
  
 base.insert("Ingresos", null, crear);  
 base.close();  
 limpiar(v);  
 Toast.*makeText*(this, "Registro creado!!", Toast.*LENGTH\_LONG*).show();  
 }else{  
 Toast.*makeText*(this, mensaje, Toast.*LENGTH\_LONG*).show();  
 Editerror.setText(mensaje);  
 Editerror.setVisibility(View.*VISIBLE*);  
 }  
 }else{  
 Toast.*makeText*(this, "Debe ingresar todos los campos", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 }  
  
  
 public void BuscarIngresos(View v){  
 SQLiteDatabase base=abreBase();  
 String desC=ECorta.getText().toString();  
 if (!desC.isEmpty() ){  
 Cursor rsfila=base.rawQuery("Select idingresos,descripC ,monto ,fechapago ,fechareg,obs from Ingresos where descripC ='"+desC+"';",null);  
 if(rsfila.moveToFirst()){  
 Ecod.setText(rsfila.getString(0));  
 ECorta.setText(rsfila.getString(1));  
 EMonto.setText(rsfila.getString(2));  
 EFecha.setText(rsfila.getString(3));  
 EObs.setText(rsfila.getString(5));  
 ECorta.setEnabled(false);  
 ImgbtnM.setEnabled(true);  
 ImgbtnD.setEnabled(true);  
  
 base.close();  
 }else{  
 Toast.*makeText*(this,"registro no existe", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 }  
  
  
 public void ModificarIngresos(View v){  
 SQLiteDatabase base=abreBase();  
 String DesC=ECorta.getText().toString();  
 String Monto=EMonto.getText().toString();  
 String Fecha=EFecha.getText().toString();  
 String Cod=Ecod.getText().toString();  
 String Obs=EObs.getText().toString();  
  
 if (!DesC.isEmpty() && !Monto.isEmpty() && !Fecha.isEmpty()) {  
 String mensaje=Valida2();  
 if (mensaje.isEmpty()) {  
 ContentValues modif = new ContentValues();  
 modif.put("descripC", DesC);  
 modif.put("monto", Monto);  
 modif.put("fechapago", Fecha);  
 modif.put("obs", Obs);  
 // modif.put("codigo", Cod);  
 base.update("Ingresos", modif, "idingresos=" + Cod, null);  
 base.close();  
 limpiar(v);  
 Toast.*makeText*(this, "Registro fue modificado!!", Toast.*LENGTH\_LONG*).show();  
 }else{  
 Editerror.setText(mensaje);  
 Editerror.setVisibility(View.*VISIBLE*);  
 }  
 } else {  
 Toast.*makeText*(this, "Debe ingresar todos los campos", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 }  
  
  
 public void EliminarIngreso(View v){  
 SQLiteDatabase base;  
 base=abreBase();  
 String codigo=Ecod.getText().toString();  
 if (!codigo.isEmpty() ){  
 base.delete("Ingresos","idingresos="+codigo,null);  
 base.close();  
 limpiar(v);  
 Toast.*makeText*(this,"registro eliminado", Toast.*LENGTH\_LONG*).show();  
 }else{  
 Toast.*makeText*(this,"registro no existe", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
  
  
 public boolean ExisteIngresos(String cadena){  
 SQLiteDatabase base=abreBase();  
 boolean aux=false;  
 if (!cadena.isEmpty() ){  
 Cursor rsfila=base.rawQuery("Select idingresos from Ingresos where descripC='"+cadena+"';",null);  
 if(rsfila.moveToFirst()){  
 Ecod.setText(rsfila.getString(0));  
 base.close();  
 aux= true;  
 }else{  
 aux= false;  
 }  
 }  
 return aux;  
 }  
  
  
 public String Valida2(){  
 FuncGral f=new FuncGral();  
 String mensaje="";  
 boolean opt=true;  
 if (f.TieneEspacios(ECorta.getText().toString())){  
 mensaje="Descripciópn Corta no debe tener espacios";  
 opt=false;  
 }  
  
 return mensaje;  
 }  
  
  
}

ListCuentas

package com.example.walletcontrol;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
  
import java.util.ArrayList;  
  
public class ListCuentas extends AppCompatActivity {  
 private ListView lv;  
 ArrayAdapter adaptador;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_list\_cuentas*);  
 lv=findViewById(R.id.*lista*);  
 CargalvCuentas();  
 }  
  
 public void CargalvCuentas(){  
 SQLiteDatabase base = abreBase();  
 ArrayList<String> lista=new ArrayList<>();  
  
 Cursor rg=base.rawQuery("select codigo,descripC,diasdepago from Cuentas ;",null);  
 if (rg.moveToFirst()){  
 do{  
 lista.add("id:"+rg.getString(0)+"//Cuenta:"+rg.getString(1)+"//Dia de pago:"+rg.getString(2));  
 }while(rg.moveToNext());  
 }  
 adaptador=new ArrayAdapter(this, android.R.layout.*simple\_list\_item\_1*,lista);  
 lv.setAdapter(adaptador);  
 }  
  
 public SQLiteDatabase abreBase(){  
 AdminBD admin= new AdminBD(this,"bd\_test",null,1);  
 SQLiteDatabase base=admin.getWritableDatabase();  
 return base;  
 }  
  
 public void Cerrar(View v){  
 finish();  
 }  
}

Reportes

package com.example.walletcontrol;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
import android.widget.Spinner;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.util.ArrayList;  
  
public class Reportes extends AppCompatActivity {  
 private Spinner spM,spA,spR;  
 private ListView lv;  
 private ArrayList<String> lista;  
 private TextView txtv;  
 ArrayAdapter adaptador;  
 private String[] Ames = {"Seleccione","01:Enero","02:Febrero","03:Marzo","04:Abril","05:Mayo","06:Junio","07:Julio","08:Agosto","09:Septiembre","10:Octubre","11:Noviembre","12:Diciembre"};  
 private String[] Aanios = {"Seleccione","2018","2019","2020","2021","2022","2023","2024","2025","2026","2027"};  
 private String[] Arep = {"Seleccione","Reporte Gastos","Reporte Ingresos"};  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_reportes*);  
  
 lv=findViewById(R.id.*lista*);  
  
 spM = findViewById(R.id.*spmeses*);  
 //spM.setAdapter(new ArrayAdapter<String>(getApplicationContext(), android.R.layout.simple\_spinner\_dropdown\_item,meses));  
  
 ArrayAdapter<String> adaptadorM= new ArrayAdapter<>(this, android.R.layout.*simple\_spinner\_item*,Ames);  
 spM.setAdapter(adaptadorM);  
  
 spA = findViewById(R.id.*spanios*);  
// spA.setAdapter(new ArrayAdapter<String>(getApplicationContext(), android.R.layout.simple\_spinner\_dropdown\_item,anios));  
 ArrayAdapter<String> adaptadorA= new ArrayAdapter<>(this, android.R.layout.*simple\_spinner\_item*,Aanios);  
 spA.setAdapter(adaptadorA);  
  
 spR = findViewById(R.id.*spreportes*);  
 // spR.setAdapter(new ArrayAdapter<String>(getApplicationContext(), android.R.layout.simple\_spinner\_dropdown\_item,rep));  
 ArrayAdapter<String> adaptadorR= new ArrayAdapter<>(this, android.R.layout.*simple\_spinner\_item*,Arep);  
 spR.setAdapter(adaptadorR);  
  
 txtv=findViewById(R.id.*textTotal*);  
 txtv.setText("");  
 }  
 public void Cerrar(View v){  
 finish();  
 }  
  
  
  
 public void BuscarRep(View v) {  
 if (Valida()) {  
 switch (spR.getSelectedItem().toString()) {  
 case "Reporte Gastos":  
 CargalvGastos();  
 break;  
 case "Reporte Ingresos":  
 CargalvIngresos();  
 break;  
 }  
 }else{  
 Toast.*makeText*(this,"Debe ingresar todos los filtros",Toast.*LENGTH\_LONG*).show();  
 }  
  
 }  
  
 public void CargalvIngresos(){  
 int Suma=0;  
 SQLiteDatabase base = abreBase();  
 ArrayList<String> lista=new ArrayList<>();  
 String M[]=spM.getSelectedItem().toString().split(":");  
 String A=spA.getSelectedItem().toString();  
 Cursor rg=base.rawQuery("select idingresos,descripC,monto,fechapago from Ingresos where strftime('%m', fechapago)='"+M[0]+"' and strftime('%Y',fechapago)='"+A+"';",null);  
 if (rg.moveToFirst()){  
 do{  
 lista.add("id:"+rg.getString(0)+"//Glosa:"+rg.getString(1)+"//Monto:"+rg.getString(2)+"//Fecha:"+rg.getString(3));  
 Suma=Suma+rg.getInt(2);  
 }while(rg.moveToNext());  
 }  
 txtv.setText(" "+Suma);  
 adaptador=new ArrayAdapter(this, android.R.layout.*simple\_list\_item\_1*,lista);  
 lv.setAdapter(adaptador);  
 }  
  
 public void CargalvGastos(){  
 int Suma=0;  
 SQLiteDatabase base = abreBase();  
 ArrayList<String> lista=new ArrayList<>();  
 String M[]=spM.getSelectedItem().toString().split(":");  
 String A=spA.getSelectedItem().toString();  
 Cursor rg=base.rawQuery("select g.idgasto , g.idcuenta ,g.descripC ,g.monto , g.fechapago,c.descripC from Gastos g, Cuentas c where g.idcuenta=c.codigo and strftime('%m', fechapago)='"+M[0]+"' and strftime('%Y',fechapago)='"+A+"';",null);  
 if (rg.moveToFirst()){  
 do{  
 lista.add("id:"+rg.getString(0)+"//Cuenta:"+rg.getString(5)+"//Glosa:"+rg.getString(2)+"//Monto:"+rg.getString(3)+"//Fecha:"+rg.getString(4));  
 Suma=Suma+rg.getInt(3);  
 }while(rg.moveToNext());  
  
 }  
  
 txtv.setText(" "+Suma);  
 adaptador=new ArrayAdapter(this, android.R.layout.*simple\_list\_item\_1*,lista);  
 lv.setAdapter(adaptador);  
 }  
  
 public boolean Valida(){  
 boolean aux=true;  
 if (spR.getSelectedItem().toString().equals("Seleccione") ){  
 aux=false;  
 }  
 if (spA.getSelectedItem().toString().equals("Seleccione")){  
 aux=false;  
 }  
 if (spM.getSelectedItem().toString().equals("Seleccione") ){  
 aux=false;  
 }  
 return aux;  
 }  
  
 public SQLiteDatabase abreBase(){  
 AdminBD admin= new AdminBD(this,"bd\_test",null,1);  
 SQLiteDatabase base=admin.getWritableDatabase();  
 return base;  
 }  
}