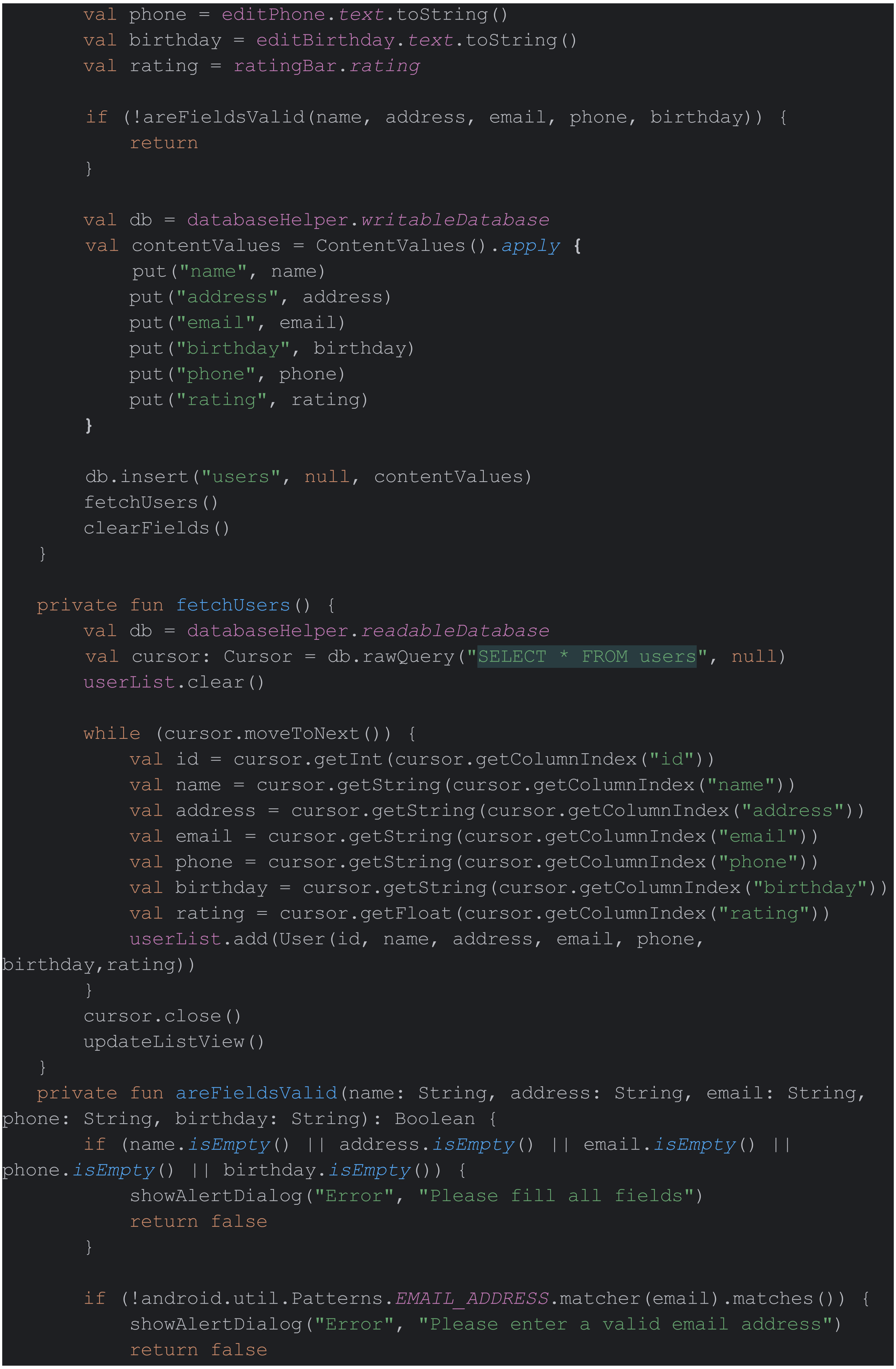
DatabaseHelper.kt

```
package com.example.myapplication

import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper

class DatabaseHelper(context: Context?) : SQLiteOpenHelper(context,
DATABASE_NAME, null, DATABASE_VERSION) {
   override fun onCreate(db: SQLiteDatabase) {
       val createTable = """
           CREATE TABLE users (
               id INTEGER PRIMARY KEY AUTOINCREMENT,
               name TEXT,
               address TEXT,
               email TEXT,
               birthday TEXT,
               phone TEXT,
               rating REAL
           )
       """
       db.execSQL(createTable)
   }

   override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int, newVersion:
Int) {
       if (oldVersion < 2) {
           // Add the new column 'rating' to the existing 'users' table
           db.execSQL("ALTER TABLE users ADD COLUMN rating REAL DEFAULT 0")
// Default value can be set to 0 or NULL
       }
       db.execSQL("DROP TABLE IF EXISTS users")
       onCreate(db)
   }

   companion object {
       private const val DATABASE_NAME = "users.db"
       private const val DATABASE_VERSION = 2
   }
}
```

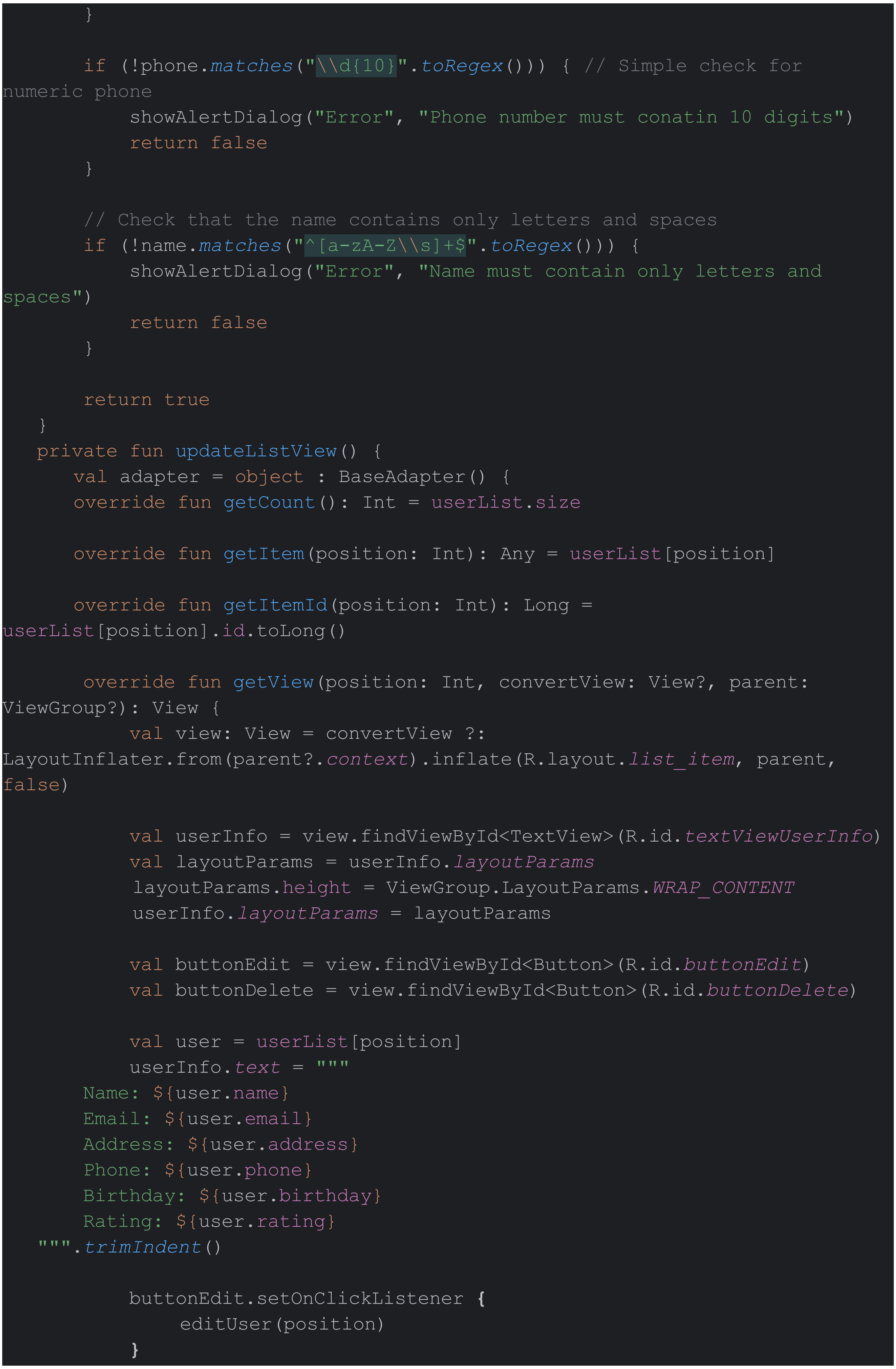
MainActivity.kt :

```
package com.example.myapplication

import android.os.Bundle
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import android.content.Intent
import android.database.Cursor
import android.view.View
import android.widget.*
import androidx.appcompat.app.AlertDialog
import java.text.SimpleDateFormat
import java.util.*
import android.content.ContentValues
import android.app.DatePickerDialog
import android.view.LayoutInflater
import android.view.ViewGroup
import android.app.Activity
import android.os.Handler
import android.os.Looper

class MainActivity : AppCompatActivity() {
   private lateinit var editName: EditText
   private lateinit var editAddress: EditText
   private lateinit var editEmail: EditText
   private lateinit var editPhone: EditText
   private lateinit var editBirthday: EditText
   private lateinit var buttonAdd: Button
   private lateinit var listView: ListView
   private lateinit var textViewDate: TextView
   private lateinit var databaseHelper: DatabaseHelper
   private lateinit var progressBar: ProgressBar
   private lateinit var ratingBar: RatingBar
   private lateinit var userList: MutableList<User>
   companion object {
       private const val EDIT_USER_REQUEST_CODE = 1 // Define the request
code
   }
   data class User(val id: Int, val name: String, val address: String, val
email: String, val phone: String, val birthday: String,val rating: Float)
   override fun onCreate(savedInstanceState: Bundle?) {
       super.onCreate(savedInstanceState)
       enableEdgeToEdge()
       setContentView(R.layout.activity_main)
       databaseHelper = DatabaseHelper(this)
       editName = findViewById(R.id.editName)
       editAddress = findViewById(R.id.editAddress)
       editEmail = findViewById(R.id.editEmail)
       editPhone = findViewById(R.id.editPhone)
       editBirthday = findViewById(R.id.editBirthday)
       buttonAdd = findViewById(R.id.buttonAdd)
       listView = findViewById(R.id.listView)
       progressBar = findViewById(R.id.progressBar)
       textViewDate = findViewById(R.id.textViewDate)
       ratingBar = findViewById(R.id.ratingBar)

       databaseHelper = DatabaseHelper(this)
       userList = mutableListOf()
```

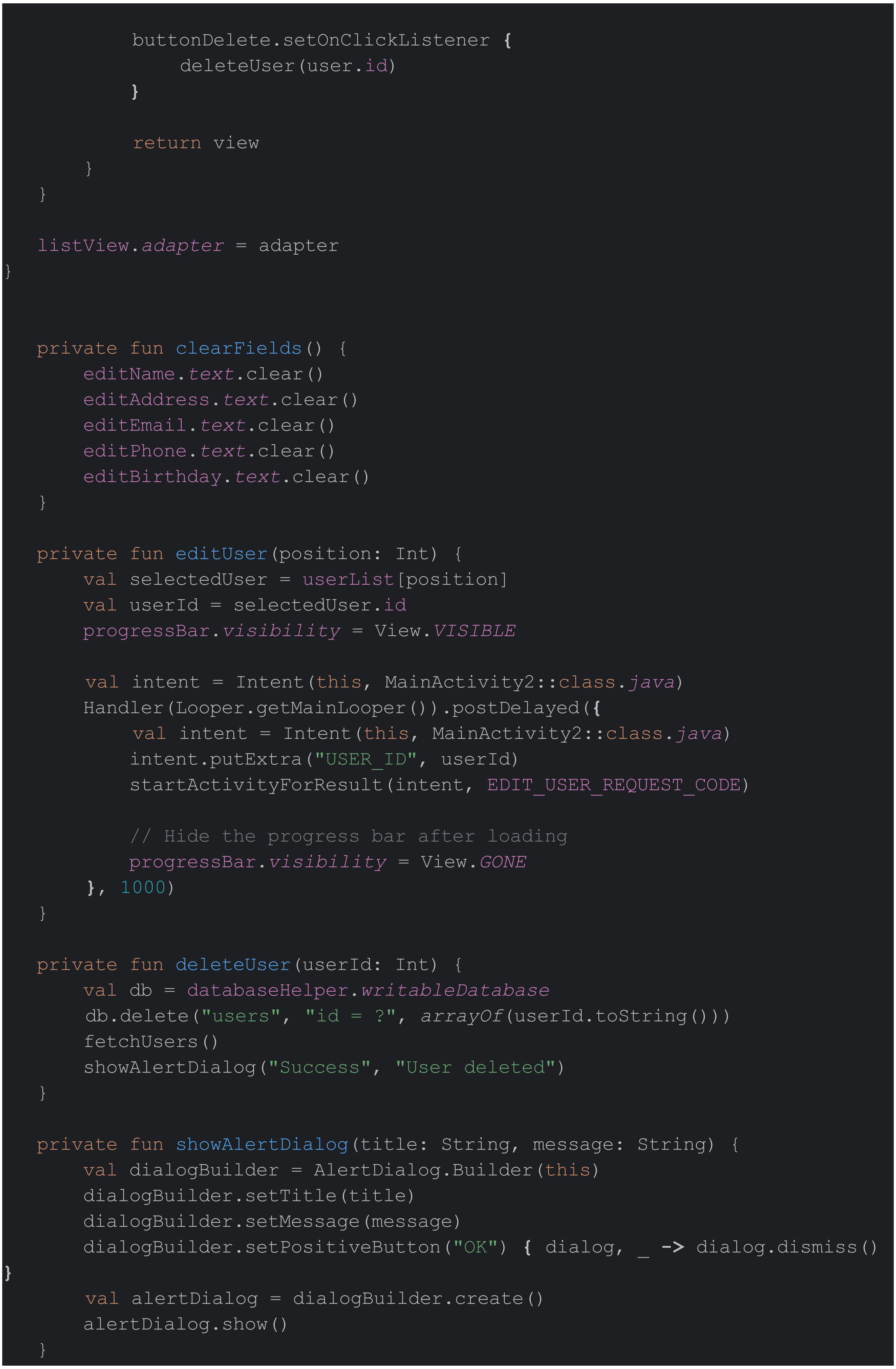
```
        buttonAdd.setOnClickListener { addUser() }
        listView.setOnItemClickListener { _, _, position, _ ->
editUser(position) }

        // Set DatePickerDialog for birthday
        editBirthday.setOnClickListener { showDatePickerDialog() }

        displayCurrentDateTime()
        fetchUsers()
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) {
v, insets ->
            val systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom)
            insets
        }
    }
    override fun onActivityResult(requestCode: Int, resultCode: Int, data:
Intent?) {
        super.onActivityResult(requestCode, resultCode, data)
        if (requestCode == EDIT_USER_REQUEST_CODE && resultCode ==
Activity.RESULT_OK) {
            fetchUsers() // Refresh user data from the database
            updateListView() // Update the ListView with the new data
        }
    }
    private fun showDatePickerDialog() {
        val calendar = Calendar.getInstance()
        val year = calendar.get(Calendar.YEAR)
        val month = calendar.get(Calendar.MONTH)
        val day = calendar.get(Calendar.DAY_OF_MONTH)

        val datePickerDialog = DatePickerDialog(this, { _, selectedYear,
selectedMonth, selectedDay ->
            val selectedDate = "$selectedDay/${selectedMonth +
1}/$selectedYear"
            editBirthday.setText(selectedDate)
        }, year, month, day)

        datePickerDialog.show()
    }

    private fun displayCurrentDateTime() {
        val currentDateTime = SimpleDateFormat("yyyy-MM-dd HH:mm:ss",
Locale.getDefault()).format(Date())
        textViewDate.text = currentDateTime
    }

    private fun addUser() {
        val name = editName.text.toString()
        val address = editAddress.text.toString()
        val email = editEmail.text.toString()
```



```
        val phone = editPhone.text.toString()
        val birthday = editBirthday.text.toString()
        val rating = ratingBar.rating

        if (!areFieldsValid(name, address, email, phone, birthday)) {
            return
        }

        val db = databaseHelper.writableDatabase
        val contentValues = ContentValues().apply {
            put("name", name)
            put("address", address)
            put("email", email)
            put("birthday", birthday)
            put("phone", phone)
            put("rating", rating)
        }

        db.insert("users", null, contentValues)
        fetchUsers()
        clearFields()
    }

    private fun fetchUsers() {
        val db = databaseHelper.readableDatabase
        val cursor: Cursor = db.rawQuery("SELECT * FROM users", null)
        userList.clear()

        while (cursor.moveToNext()) {
            val id = cursor.getInt(cursor.getColumnIndex("id"))
            val name = cursor.getString(cursor.getColumnIndex("name"))
            val address = cursor.getString(cursor.getColumnIndex("address"))
            val email = cursor.getString(cursor.getColumnIndex("email"))
            val phone = cursor.getString(cursor.getColumnIndex("phone"))
            val birthday = cursor.getString(cursor.getColumnIndex("birthday"))
            val rating = cursor.getFloat(cursor.getColumnIndex("rating"))
            userList.add(User(id, name, address, email, phone,
birthday,rating))
        }
        cursor.close()
        updateListView()
    }
    private fun areFieldsValid(name: String, address: String, email: String,
phone: String, birthday: String): Boolean {
        if (name.isEmpty() || address.isEmpty() || email.isEmpty() ||
phone.isEmpty() || birthday.isEmpty()) {
            showAlertDialog("Error", "Please fill all fields")
            return false
        }

        if (!android.util.Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
            showAlertDialog("Error", "Please enter a valid email address")
            return false
```



```
       }

       if (!phone.matches("\\d{10}".toRegex())) { // Simple check for
numeric phone
           showAlertDialog("Error", "Phone number must conatin 10 digits")
           return false
       }

       // Check that the name contains only letters and spaces
       if (!name.matches("^[a-zA-Z\\s]+$".toRegex())) {
           showAlertDialog("Error", "Name must contain only letters and
spaces")
           return false
       }

       return true
   }
   private fun updateListView() {
      val adapter = object : BaseAdapter() {
      override fun getCount(): Int = userList.size

      override fun getItem(position: Int): Any = userList[position]

      override fun getItemId(position: Int): Long =
userList[position].id.toLong()

       override fun getView(position: Int, convertView: View?, parent:
ViewGroup?): View {
           val view: View = convertView ?:
LayoutInflater.from(parent?.context).inflate(R.layout.list_item, parent,
false)

           val userInfo = view.findViewById<TextView>(R.id.textViewUserInfo)
           val layoutParams = userInfo.layoutParams
           layoutParams.height = ViewGroup.LayoutParams.WRAP_CONTENT
           userInfo.layoutParams = layoutParams

           val buttonEdit = view.findViewById<Button>(R.id.buttonEdit)
           val buttonDelete = view.findViewById<Button>(R.id.buttonDelete)

           val user = userList[position]
           userInfo.text = """
       Name: ${user.name}
       Email: ${user.email}
       Address: ${user.address}
       Phone: ${user.phone}
       Birthday: ${user.birthday}
       Rating: ${user.rating}
   """.trimIndent()

           buttonEdit.setOnClickListener {
               editUser(position)
           }
```



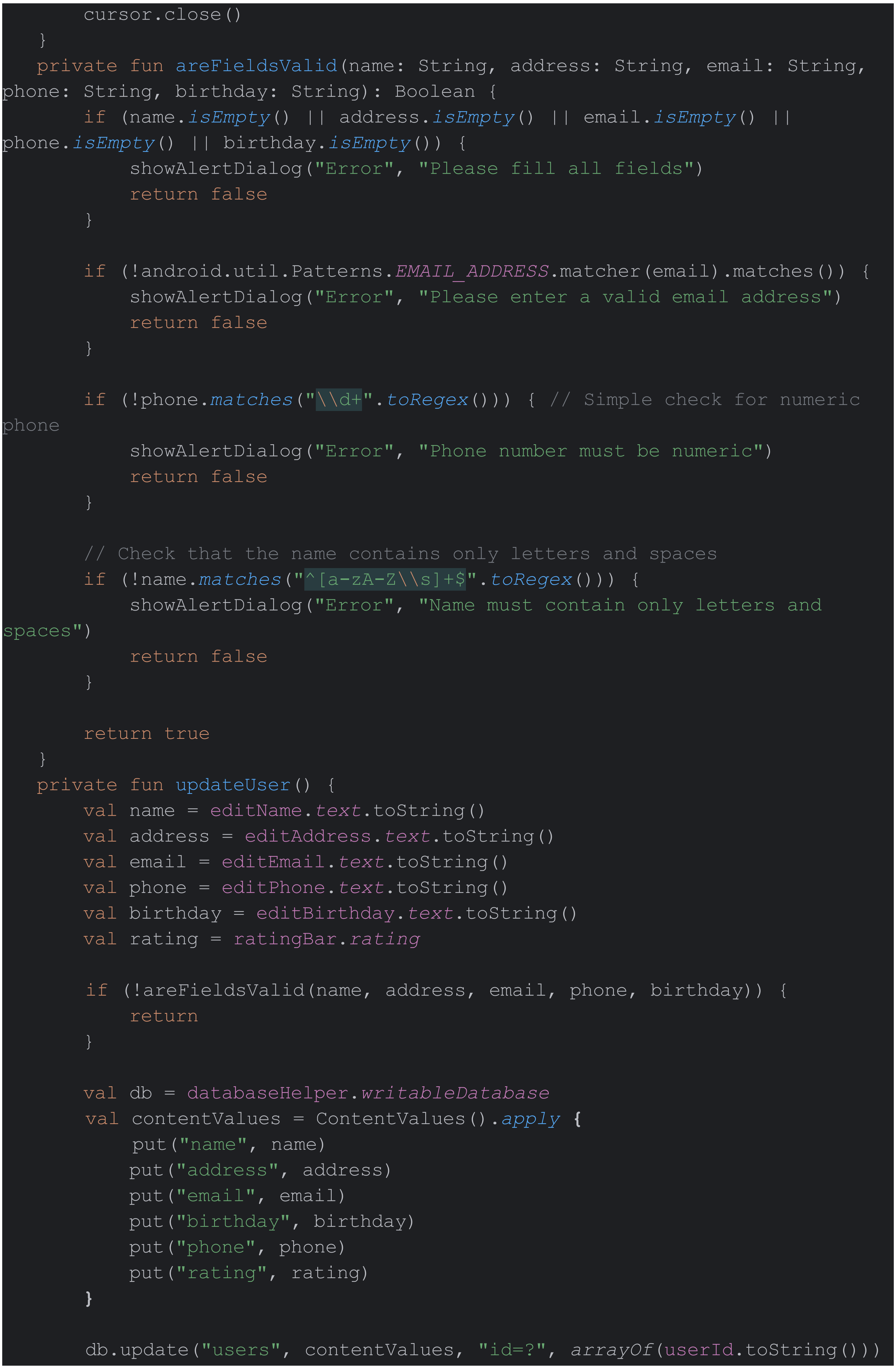
```
            buttonDelete.setOnClickListener {
                deleteUser(user.id)
            }

            return view
        }
    }

    listView.adapter = adapter
}


    private fun clearFields() {
        editName.text.clear()
        editAddress.text.clear()
        editEmail.text.clear()
        editPhone.text.clear()
        editBirthday.text.clear()
    }

    private fun editUser(position: Int) {
        val selectedUser = userList[position]
        val userId = selectedUser.id
        progressBar.visibility = View.VISIBLE

        val intent = Intent(this, MainActivity2::class.java)
        Handler(Looper.getMainLooper()).postDelayed({
            val intent = Intent(this, MainActivity2::class.java)
            intent.putExtra("USER_ID", userId)
            startActivityForResult(intent, EDIT_USER_REQUEST_CODE)

            // Hide the progress bar after loading
            progressBar.visibility = View.GONE
        }, 1000)
    }

    private fun deleteUser(userId: Int) {
        val db = databaseHelper.writableDatabase
        db.delete("users", "id = ?", arrayOf(userId.toString()))
        fetchUsers()
        showAlertDialog("Success", "User deleted")
    }

    private fun showAlertDialog(title: String, message: String) {
        val dialogBuilder = AlertDialog.Builder(this)
        dialogBuilder.setTitle(title)
        dialogBuilder.setMessage(message)
        dialogBuilder.setPositiveButton("OK") { dialog, _ -> dialog.dismiss()
}
        val alertDialog = dialogBuilder.create()
        alertDialog.show()
    }
```

```
}
```

MainActivity2.kt :

```
package com.example.myapplication

import android.os.Bundle
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import android.content.Intent
import android.database.Cursor
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import android.content.ContentValues
import android.app.Activity
import androidx.appcompat.app.AlertDialog
import android.app.DatePickerDialog
import android.widget.RatingBar
import java.util.*


class MainActivity2 : AppCompatActivity() {
   private lateinit var editName: EditText
   private lateinit var editAddress: EditText
   private lateinit var editEmail: EditText
   private lateinit var editPhone: EditText
   private lateinit var editBirthday: EditText
   private lateinit var buttonUpdate: Button
   private lateinit var databaseHelper: DatabaseHelper
   private lateinit var ratingBar: RatingBar
   private var userId: Int = -1

   override fun onCreate(savedInstanceState: Bundle?) {
       super.onCreate(savedInstanceState)
       enableEdgeToEdge()
       setContentView(R.layout.activity_main2)
       editName = findViewById(R.id.editName)
       editAddress = findViewById(R.id.editAddress)
       editEmail = findViewById(R.id.editEmail)
       editPhone = findViewById(R.id.editPhone)
       editBirthday = findViewById(R.id.editBirthday)
       buttonUpdate = findViewById(R.id.buttonUpdate)
       ratingBar = findViewById(R.id.ratingBar)

       databaseHelper = DatabaseHelper(this)
       userId = intent.getIntExtra("USER_ID", -1)

       fetchUserDetails()
       editBirthday.setOnClickListener { showDatePickerDialog() }
```

```
        buttonUpdate.setOnClickListener { updateUser() }
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) {
v, insets ->
            val systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom)
            insets
        }
    }
    private fun showAlertDialog(title: String, message: String) {
        val dialogBuilder = AlertDialog.Builder(this)
        dialogBuilder.setTitle(title)
        dialogBuilder.setMessage(message)
        dialogBuilder.setPositiveButton("OK") { dialog, _ -> dialog.dismiss()
}
        val alertDialog = dialogBuilder.create()
        alertDialog.show()
    }
    private fun showDatePickerDialog() {
        val calendar = Calendar.getInstance()
        val year = calendar.get(Calendar.YEAR)
        val month = calendar.get(Calendar.MONTH)
        val day = calendar.get(Calendar.DAY_OF_MONTH)

        val datePickerDialog = DatePickerDialog(this, { _, selectedYear,
selectedMonth, selectedDay ->
            val selectedDate = "$selectedDay/${selectedMonth +
1}/$selectedYear"
            editBirthday.setText(selectedDate) // Set selected date to the
EditText
        }, year, month, day)

        datePickerDialog.show()
    }
    private fun fetchUserDetails() {
        val db = databaseHelper.readableDatabase
        val cursor: Cursor = db.rawQuery("SELECT * FROM users WHERE id=?",
arrayOf(userId.toString()))

        if (cursor.moveToFirst()) {
            editName.setText(cursor.getString(cursor.getColumnIndex("name")))

editAddress.setText(cursor.getString(cursor.getColumnIndex("address")))

editEmail.setText(cursor.getString(cursor.getColumnIndex("email")))

editPhone.setText(cursor.getString(cursor.getColumnIndex("phone")))

editBirthday.setText(cursor.getString(cursor.getColumnIndex("birthday")))
            ratingBar.rating =
cursor.getFloat(cursor.getColumnIndex("rating"))
        }
```

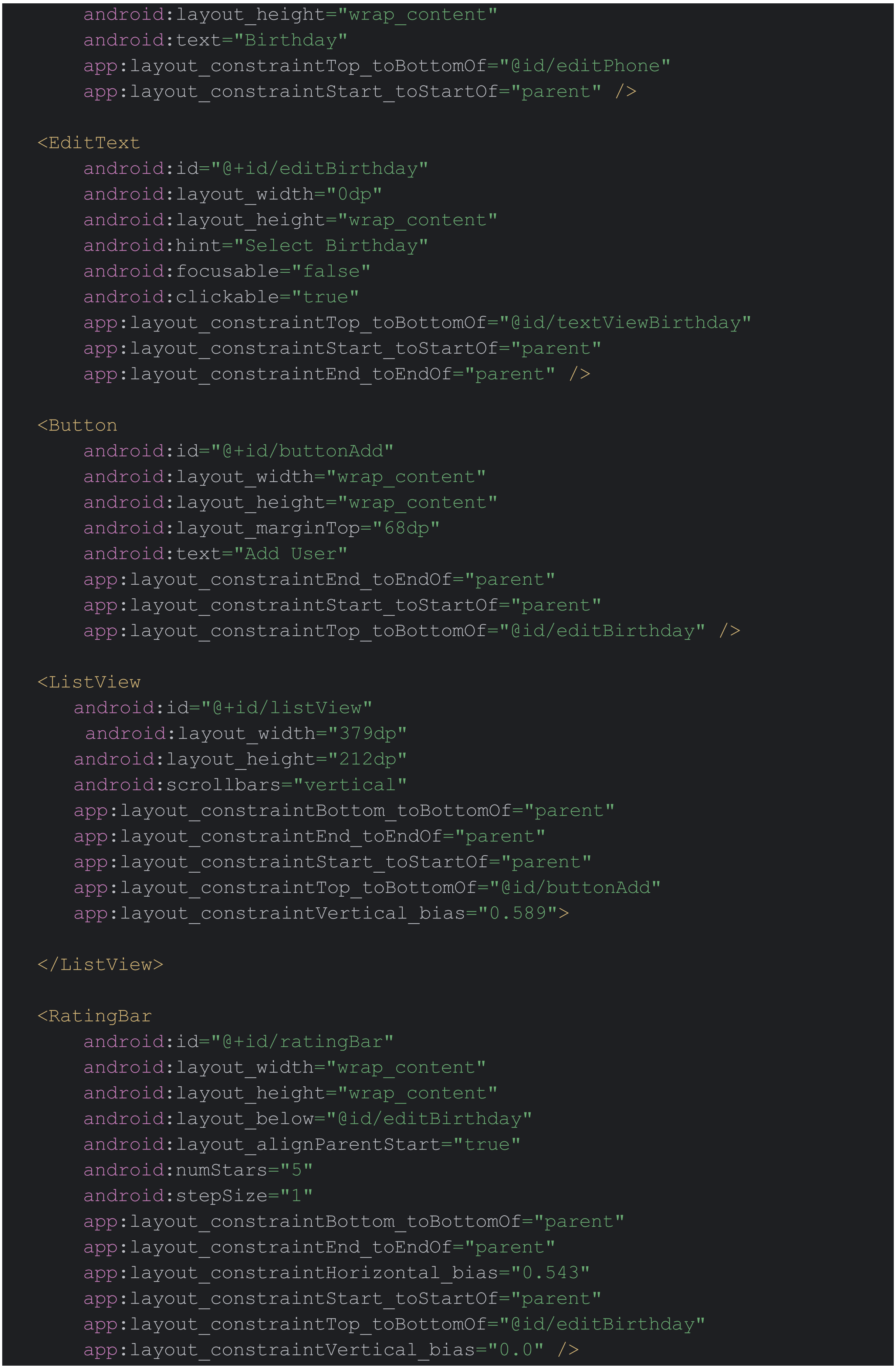


```
        cursor.close()
    }
    private fun areFieldsValid(name: String, address: String, email: String,
phone: String, birthday: String): Boolean {
        if (name.isEmpty() || address.isEmpty() || email.isEmpty() ||
phone.isEmpty() || birthday.isEmpty()) {
            showAlertDialog("Error", "Please fill all fields")
            return false
        }

        if (!android.util.Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
            showAlertDialog("Error", "Please enter a valid email address")
            return false
        }

        if (!phone.matches("\\d+".toRegex())) { // Simple check for numeric
phone
            showAlertDialog("Error", "Phone number must be numeric")
            return false
        }

        // Check that the name contains only letters and spaces
        if (!name.matches("^[a-zA-Z\\s]+$".toRegex())) {
            showAlertDialog("Error", "Name must contain only letters and
spaces")
            return false
        }

        return true
    }
    private fun updateUser() {
        val name = editName.text.toString()
        val address = editAddress.text.toString()
        val email = editEmail.text.toString()
        val phone = editPhone.text.toString()
        val birthday = editBirthday.text.toString()
        val rating = ratingBar.rating

        if (!areFieldsValid(name, address, email, phone, birthday)) {
            return
        }

        val db = databaseHelper.writableDatabase
        val contentValues = ContentValues().apply {
            put("name", name)
            put("address", address)
            put("email", email)
            put("birthday", birthday)
            put("phone", phone)
            put("rating", rating)
        }

        db.update("users", contentValues, "id=?", arrayOf(userId.toString()))
        Toast.makeText(this, "User updated", Toast.LENGTH_SHORT).show()
        setResult(Activity.RESULT_OK)
        finish()
    }

}
```

Activity_main.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:id="@+id/main"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity">

   <TextView
       android:id="@+id/textViewDate"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:layout_alignParentTop="true"
       android:layout_alignParentRight="true"
       android:padding="16dp"
       android:textColor="@android:color/black"
       android:textSize="16sp"
       app:layout_constraintBottom_toBottomOf="parent"
       app:layout_constraintEnd_toEndOf="parent"
       app:layout_constraintHorizontal_bias="0.788"
       app:layout_constraintStart_toStartOf="@+id/editName"
       app:layout_constraintTop_toTopOf="parent"
       app:layout_constraintVertical_bias="0.0" />

   <TextView
       android:id="@+id/textViewName"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Name"
       app:layout_constraintTop_toTopOf="parent"
       app:layout_constraintStart_toStartOf="parent" />

   <EditText
       android:id="@+id/editName"
       android:layout_width="0dp"
       android:layout_height="wrap_content"
       android:hint="Enter Name"
       app:layout_constraintTop_toBottomOf="@id/textViewName"
       app:layout_constraintStart_toStartOf="parent"
       app:layout_constraintEnd_toEndOf="parent" />

   <TextView
```

```
        android:id="@+id/textViewAddress"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Address"
        app:layout_constraintTop_toBottomOf="@id/editName"
        app:layout_constraintStart_toStartOf="parent" />

    <EditText
        android:id="@+id/editAddress"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Enter Address"
        app:layout_constraintTop_toBottomOf="@id/textViewAddress"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

    <TextView
        android:id="@+id/textViewEmail"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Email"
        app:layout_constraintTop_toBottomOf="@id/editAddress"
        app:layout_constraintStart_toStartOf="parent" />

    <EditText
        android:id="@+id/editEmail"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Enter Email"
        app:layout_constraintTop_toBottomOf="@id/textViewEmail"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

    <TextView
        android:id="@+id/textViewPhone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Phone Number"
        app:layout_constraintTop_toBottomOf="@id/editEmail"
        app:layout_constraintStart_toStartOf="parent" />

    <EditText
        android:id="@+id/editPhone"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Enter Phone Number"
        app:layout_constraintTop_toBottomOf="@id/textViewPhone"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

    <TextView
        android:id="@+id/textViewBirthday"
        android:layout_width="wrap_content"
```



```
        android:layout_height="wrap_content"
        android:text="Birthday"
        app:layout_constraintTop_toBottomOf="@id/editPhone"
        app:layout_constraintStart_toStartOf="parent" />

    <EditText
        android:id="@+id/editBirthday"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Select Birthday"
        android:focusable="false"
        android:clickable="true"
        app:layout_constraintTop_toBottomOf="@id/textViewBirthday"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

    <Button
        android:id="@+id/buttonAdd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="68dp"
        android:text="Add User"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/editBirthday" />

    <ListView
       android:id="@+id/listView"
        android:layout_width="379dp"
       android:layout_height="212dp"
       android:scrollbars="vertical"
       app:layout_constraintBottom_toBottomOf="parent"
       app:layout_constraintEnd_toEndOf="parent"
       app:layout_constraintStart_toStartOf="parent"
       app:layout_constraintTop_toBottomOf="@id/buttonAdd"
       app:layout_constraintVertical_bias="0.589">

    </ListView>

    <RatingBar
        android:id="@+id/ratingBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/editBirthday"
        android:layout_alignParentStart="true"
        android:numStars="5"
        android:stepSize="1"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.543"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/editBirthday"
        app:layout_constraintVertical_bias="0.0" />
```

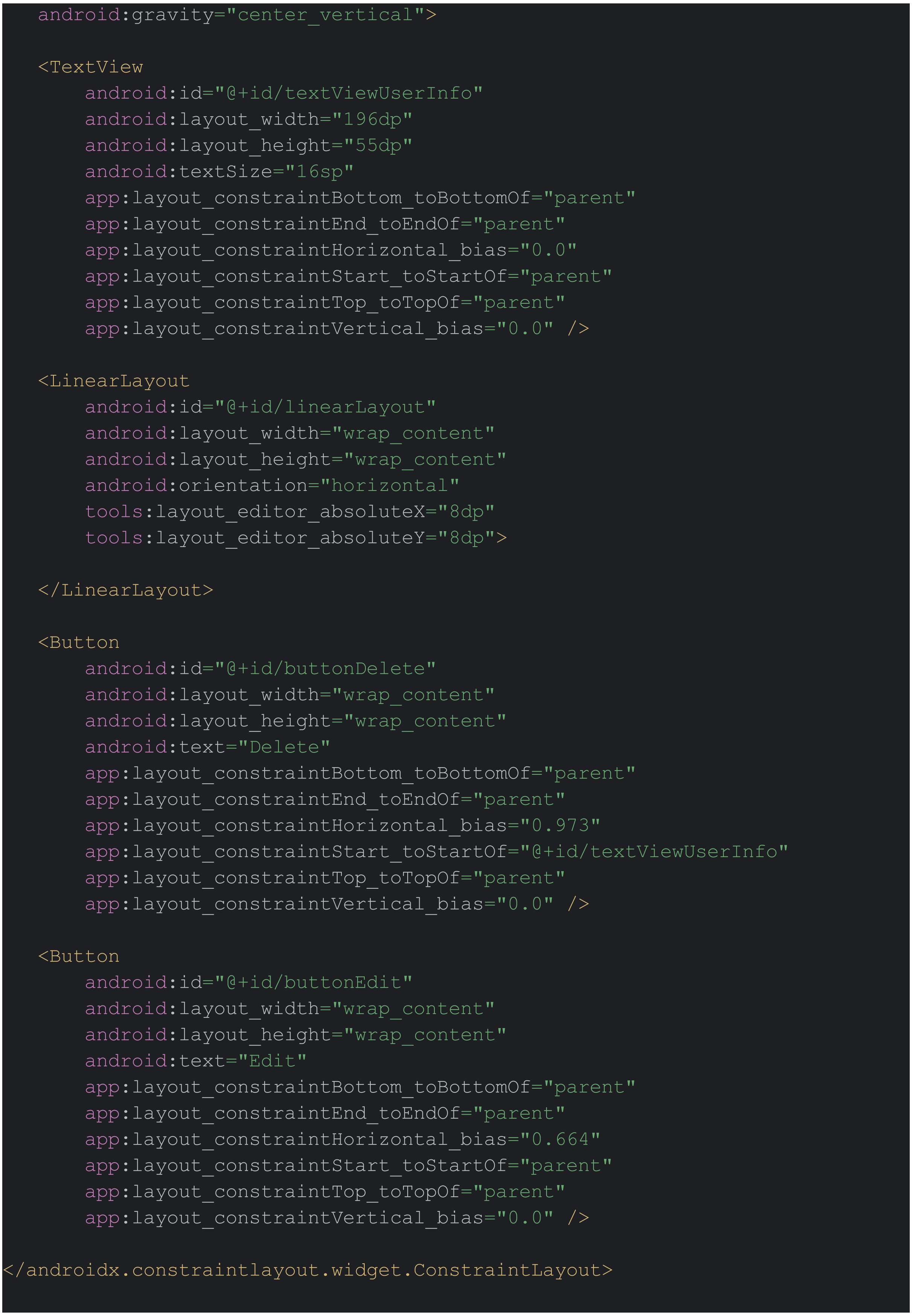
```
   <ProgressBar
       android:id="@+id/progressBar"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       app:layout_constraintTop_toBottomOf="@id/listView"
       app:layout_constraintBottom_toBottomOf="parent"
       app:layout_constraintStart_toStartOf="parent"
       app:layout_constraintEnd_toEndOf="parent"
       android:visibility="gone" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_main2.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:id="@+id/main"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity2">
   <EditText
       android:id="@+id/editName"
       android:layout_width="0dp"
       android:layout_height="wrap_content"
       android:hint="Enter Name"
       app:layout_constraintTop_toTopOf="parent"
       app:layout_constraintStart_toStartOf="parent"
       app:layout_constraintEnd_toEndOf="parent" />

   <EditText
       android:id="@+id/editAddress"
       android:layout_width="0dp"
       android:layout_height="wrap_content"
       android:hint="Enter Address"
       app:layout_constraintTop_toBottomOf="@id/editName"
       app:layout_constraintStart_toStartOf="parent"
       app:layout_constraintEnd_toEndOf="parent" />

   <EditText
       android:id="@+id/editEmail"
       android:layout_width="0dp"
       android:layout_height="wrap_content"
       android:hint="Enter Email"
       app:layout_constraintTop_toBottomOf="@id/editAddress"
       app:layout_constraintStart_toStartOf="parent"
       app:layout_constraintEnd_toEndOf="parent" />

   <EditText
       android:id="@+id/editPhone"
       android:layout_width="0dp"
       android:layout_height="wrap_content"
       android:hint="Enter Phone Number"
       app:layout_constraintTop_toBottomOf="@id/editEmail"
       app:layout_constraintStart_toStartOf="parent"
       app:layout_constraintEnd_toEndOf="parent" />

   <EditText
       android:id="@+id/editBirthday"
       android:layout_width="0dp"
       android:layout_height="wrap_content"
       android:hint="Enter Birthday"
       app:layout_constraintTop_toBottomOf="@id/editPhone"
       app:layout_constraintStart_toStartOf="parent"
       app:layout_constraintEnd_toEndOf="parent" />

   <Button
       android:id="@+id/buttonUpdate"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:layout_marginTop="116dp"
       android:layout_marginEnd="140dp"
       android:text="Update User"
       app:layout_constraintEnd_toEndOf="parent"
       app:layout_constraintTop_toBottomOf="@id/editBirthday" />

   <RatingBar
       android:id="@+id/ratingBar"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:layout_below="@id/editBirthday"
       android:layout_marginEnd="132dp"
       android:numStars="5"
       android:stepSize="1"
       app:layout_constraintBottom_toBottomOf="parent"
       app:layout_constraintEnd_toEndOf="parent"
       app:layout_constraintTop_toTopOf="parent"
       app:layout_constraintVertical_bias="0.348" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

list_item.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:orientation="horizontal"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:padding="8dp"
```



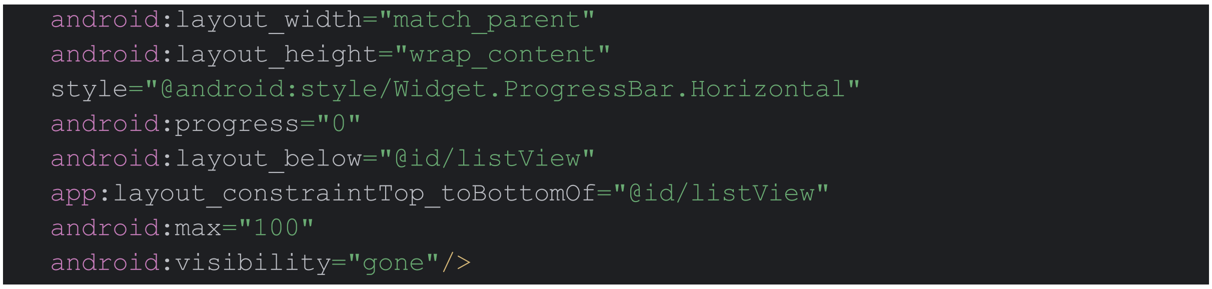
```
    android:gravity="center_vertical">

    <TextView
        android:id="@+id/textViewUserInfo"
        android:layout_width="196dp"
        android:layout_height="55dp"
        android:textSize="16sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0" />

    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        tools:layout_editor_absoluteX="8dp"
        tools:layout_editor_absoluteY="8dp">

    </LinearLayout>

    <Button
        android:id="@+id/buttonDelete"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Delete"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.973"
        app:layout_constraintStart_toStartOf="@+id/textViewUserInfo"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0" />

    <Button
        android:id="@+id/buttonEdit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Edit"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.664"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

If changes to “Horizontal Progress bar “ change edituser in mainactivity.kt :

```
private fun editUser(position: Int) {
   val selectedUser = userList[position]
   val userId = selectedUser.id
   progressBar.visibility = View.VISIBLE
   progressBar.progress = 0  // Reset progress

   val intent = Intent(this, MainActivity2::class.java)

   // Simulating progress incrementally
   val handler = Handler(Looper.getMainLooper())
   var progress = 0
   val delay = 50  // Time interval for progress update

   val runnable = object : Runnable {
       override fun run() {
           if (progress < 100) {
               progress += 10  // Increase progress
               progressBar.progress = progress
               handler.postDelayed(this, delay.toLong())  // Re-run the task
           } else {
               // Once done, start the activity and hide the progress bar
               intent.putExtra("USER_ID", userId)
               startActivityForResult(intent, EDIT_USER_REQUEST_CODE)
               progressBar.visibility = View.GONE
           }
       }
   }

   handler.post(runnable)  // Start the progress animation
}
```

Also change the progressbar in activity_main.xml :

```
<ProgressBar
   android:id="@+id/progressBar"
```



```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    style="@android:style/Widget.ProgressBar.Horizontal"
    android:progress="0"
    android:layout_below="@id/listView"
    app:layout_constraintTop_toBottomOf="@id/listView"
    android:max="100"
    android:visibility="gone"/>
```

If asked for live time :

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    startUpdatingTime()  // Start live clock (Replace showcurrentdatetime with this)
}
```

And also add :

```
private val handler = Handler(Looper.getMainLooper())

private val updateTimeRunnable = object : Runnable {
    override fun run() {
        val currentDateTime = SimpleDateFormat("yyyy-MM-dd HH:mm:ss",
Locale.getDefault()).format(Date())
        textViewDate.text = currentDateTime  // Update TextView
        handler.postDelayed(this, 1000) // Repeat every second
    }
}

private fun startUpdatingTime() {
    handler.post(updateTimeRunnable) // Start updating time
}

private fun stopUpdatingTime() {
    handler.removeCallbacks(updateTimeRunnable) // Stop updating time
}
```

If u want to refer for code:

```
val button1=findViewById<Button>(R.id.button) as Button
val tv1=findViewById<TextView>(R.id.textView) as TextView
button1.setOnClickListener(){
      Toast.makeText(this,"I am so Full",Toast.LENGTH_SHORT).show()
      tv1.text="I am so Full"
      val imageView=findViewById<ImageView>(R.id.imageView)
      imageView.setImageResource(R.drawable.smile)
      button1.setText( "Done")
}
```

```
val button1 = findViewById<Button>(R.id.button)
         val tv1 = findViewById<EditText>(R.id.editTextText)
         val tv2 = findViewById<EditText>(R.id.editTextText2)

         button1.setOnClickListener {
              val c = tv1.text.toString()
             val f = tv2.text.toString()
             if (c.isNotEmpty()) {
                  val celsius = c.toDouble()
                  val fahrenheit = (celsius * 9 / 5) + 32
                  Toast.makeText(this, "Fahrenheit: %.2f".format(fahrenheit),
Toast.LENGTH_SHORT).show()
                  tv2.setText(fahrenheit.toString())
             } else if (f.isNotEmpty()) {
                  val fahrenheit = f.toDouble()
                  val celsius = (fahrenheit - 32) * 5 / 9
                  Toast.makeText(this, "Celsius: %.2f".format(celsius),
Toast.LENGTH_SHORT).show()
                  tv1.setText(celsius.toString())
             } else {
                  Toast.makeText(this, "Please enter a value",
Toast.LENGTH_SHORT).show()
             }
         }
```

```
ratingBar.setOnRatingBarChangeListener { _, rating, _ ->
            val ratingInt = rating.toInt()
            when (ratingInt) {
                5 -> ratingMessage.text = getString(R.string.awesome_text)
                4 -> ratingMessage.text = getString(R.string.good_text)
                3 -> ratingMessage.text = getString(R.string.satisfied_text)
                2 -> ratingMessage.text = getString(R.string.not_good_text)
                1 -> ratingMessage.text =
getString(R.string.disappointed_text)
                else -> ratingMessage.text = ""
            }
        }
```

```
import android.content.Intent
```

```
val intent = Intent(this, MainActivity2::class.java)
               intent.putExtra("FEEDBACK_MESSAGE", feedback)
               startActivity(intent)
```

```
val feedback = intent.getStringExtra("FEEDBACK_MESSAGE")
```

```
<string-array name="my_spinner_items">
       <item>Select A Country</item>
       <item>USA</item>
       <item>UK</item>
       <item>India</item>
       <item>Pakistan</item>
   </string-array>
```

```
 val Country = visitTypeSpinner.selectedItem.toString()
           val selectedRadioButtonId = radioGroup.checkedRadioButtonId
           val radioButton =
findViewById<RadioButton>(selectedRadioButtonId)
           val visitType = radioButton?.text.toString()
           val intent = Intent(this, MainActivity2::class.java).apply {
               putExtra("FIRST_NAME", firstName)
               putExtra("LAST_NAME", lastName)
               putExtra("COUNTRY", Country)
               putExtra("VISIT_TYPE", visitType)
           }
```

```
private fun isValidName(name: String): Boolean {
       // Regular expression to match only alphabets (both uppercase and
lowercase)
       return name.matches(Regex("^[a-zA-Z]+$"))
   }
```

```
 displayTextView.text = "First Name: $firstName\nLast Name:
$lastName\nVisit Type: $visitType\nCountry: $Country"
```

```
android:entries="@array/my_spinner_items"
```

```
generatedCaptcha = (1..6).map { characters.random() }.joinToString("")
```

```
val userDetails = intent.getStringExtra("USER_DETAILS") ?: ""
        val details = userDetails.split("\n").map { it.split(": ")[1] }
```

```
private fun showDatePickerDialog() {
    val calendar = Calendar.getInstance()
    val year = calendar.get(Calendar.YEAR)
    val month = calendar.get(Calendar.MONTH)
    val day = calendar.get(Calendar.DAY_OF_MONTH)

    val datePickerDialog = DatePickerDialog(this, { _, selectedYear, selectedMonth, selectedDay ->
        val date = "$selectedDay/${selectedMonth + 1}/$selectedYear"
        birthdayEditText.setText(date)
    }, year, month, day)

    datePickerDialog.show()
```

```
 }
```

```
val result = """
        First Name: $firstName
        Last Name: $lastName
        Phone: $phone
        Birthday: $birthday
        Gender: $gender
        Address: $address
    """.trimIndent()
```

```
 editButton.setOnClickListener {
            val intent = Intent(this, MainActivity::class.java)
            intent.putExtra("EDIT_MODE", true)
            intent.putExtra("USER_DETAILS", details)
            startActivity(intent)
            finish()
        }

        confirmButton.setOnClickListener {
            val firstName = details?.substringAfter("First Name:
")?.substringBefore("\n") ?: "User"
            val intent = Intent(this, MainActivity3::class.java)
            intent.putExtra("FIRST_NAME", firstName)
            startActivity(intent)
            finish()
        }
```

```
<LinearLayout

        android:id="@+id/linearLayout"

        android:layout_width="334dp"

        android:layout_height="69dp"

        android:orientation="horizontal"

        app:layout_constraintBottom_toBottomOf="parent"

        app:layout_constraintEnd_toEndOf="parent"

        android:gravity="center|top"

        android:dividerPadding="13dp"

        app:layout_constraintHorizontal_bias="0.376"

        app:layout_constraintStart_toStartOf="parent"

        app:layout_constraintTop_toBottomOf="@+id/textView"
        app:layout_constraintVertical_bias="0.628">
```

```
        <TextView

        android:id="@+id/yearstextView"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_marginEnd="26dp"

        android:textAlignment="center"

        android:textColor="@color/text_color2"

        android:text="Years \n" />
```

```
private lateinit var differenceTextView: TextView
    private var fromDate: Calendar? = null
    private var toDate: Calendar? = null
```