Министерство образования Республики Беларусь

Учреждение образования «Полоцкий государственный университет имени Евфросинии Полоцкой»

Факультет информационных технологий

Кафедра технологий программирования

**Отчёт по лабораторной работе № 3 по курсу «Программирование мобильных систем»**

«Использование управляющих элементов в пользовательском интерфейсе. Работа с базами данных в Android»

ВЫПОЛНИЛ студент группы 21-ИТ-1

Чиникайло А.П.

ПРОВЕРИЛ преподаватель

Васильева Д.М.

Полоцк 2024 г.

Цель работы: Научиться использовать в интерфейсе пользователя различные управляющие элементы: кнопки с изображениями, радиокнопки, чекбоксы и другие. Научиться взаимодействовать с базой данных в Android приложении.

**ХОД РАБОТЫ**

Задание:

1. Добавить в приложение графическую кнопку и проследить, как изменяется изображение кнопки в разных состояниях, а также как функционирует обработчик нажатия на кнопку.
2. Добавить в приложение чекбокс и проследить за поведением чекбокса в разных ситуациях.
3. Добавить в приложение виджет ToggleButton.
4. Добавить в приложение виджет RadioButton:
5. Добавить в приложение виджет EditText и кнопку «Очистить» в разметку и написать обработчик, очищающий текстовое поле (используя метод setText виджета EditText).
6. Создать новый проект SQLTest, главная Активность которого будет расширять класс ListActivity.
7. В методе onCreate главной Активности сделать открытие или создание БД, используя в качестве основы информацию из пункта «Работа с СУБД без адаптера» и метода onUpgrade вспомогательного класса из предшествующего ему пункта. После создания БД создать таблицу с любым полем (плюс индекс), в которую добавьте 3..5 уникальных записей со строковым полем. Индекс должен инкрементироваться автоматически.
8. В этом же методе сделать запрос к таблице, получающий все строки и с помощью имеющегося курсора запишите данные в массив строк.
9. С помощью дополнительной разметки и адаптера ArrayAdapter отобразить полученные из СУБД данные на экране Активности.
10. Добавить обработчик клика на элемент списка, чтобы при клике запрашивался из БД индекс элемента с этой строкой и отображался с помощью Toast.
11. Добавить контекстное меню к элементам списка, содержащее пункт «удалить» и реализовать удаление. После удаления должны производится действия из п.п. 3 и 4 для изменения состава отображаемых данных.
12. Реализовать редактирование записей по нажатию на кнопку, а удаление через контекстное меню. Работа с СУБД должна осуществляться с использованием адаптера. Пример обработчика выбора пункта контекстного меню приведен в соответствующем разделе данного методического руководства.

Листинг 1 – MainActivity.kt

package com.example.lab\_3  
  
import android.annotation.SuppressLint  
import android.os.Bundle  
import android.view.KeyEvent  
import android.view.View  
import android.widget.Button  
import android.widget.CheckBox  
import android.widget.EditText  
import android.widget.RadioButton  
import android.widget.Toast  
import android.widget.ToggleButton  
import androidx.activity.ComponentActivity  
import androidx.compose.material3.Text  
import androidx.compose.runtime.Composable  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.tooling.preview.Preview  
import com.example.lab\_3.ui.theme.Lab\_3Theme  
  
class MainActivity : ComponentActivity() {  
 @SuppressLint("MissingInflatedId")  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.main);  
 val userName = findViewById<EditText>(R.id.user\_name)  
 userName.setOnKeyListener { v, keyCode, event ->  
 if (event.action == KeyEvent.ACTION\_DOWN && keyCode == KeyEvent.KEYCODE\_ENTER) {  
 Toast.makeText(applicationContext, userName.text, Toast.LENGTH\_SHORT).show()  
 true  
 } else {  
 false  
 }  
 }  
 }  
 fun onButtonClicked(v: View) {  
 Toast.makeText(this, "Кнопка нажата", Toast.LENGTH\_SHORT).show()  
 }  
 fun onCheckboxClicked(v: View) {  
 if ((v as CheckBox).isChecked) {  
 Toast.makeText(this, "Отмечено", Toast.LENGTH\_SHORT).show()  
 } else {  
 Toast.makeText(this, "Не отмечено", Toast.LENGTH\_SHORT).show()  
 }  
 }  
 fun onToggleClicked(v: View) {  
 if ((v as ToggleButton).isChecked) {  
 Toast.makeText(this, "Включено", Toast.LENGTH\_SHORT).show()  
 } else {  
 Toast.makeText(this, "Выключено", Toast.LENGTH\_SHORT).show()  
 }  
 }  
 fun onRadioButtonClicked(v: View) {  
 val rb = v as RadioButton  
 Toast.makeText(this, "Выбрано животное: ${rb.text}", Toast.LENGTH\_SHORT).show()  
 }  
  
}

Листинг 2 – main.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <Button  
 android:id="@+id/button1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/smile\_button"  
 android:onClick="onButtonClicked"  
 android:padding="10dp"  
 android:maxWidth="10dp"  
 android:maxHeight="10dp"  
 android:text="Ok"/>  
 <CheckBox  
 android:id="@+id/checkbox1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="onCheckboxClicked"  
 android:text="Выбери меня" />  
 <ToggleButton android:id="@+id/togglebutton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textOn="Звонок включен"  
 android:textOff="Звонок выключен"  
 android:onClick="onToggleClicked"/>  
  
 <RadioGroup  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical" >  
 <RadioButton  
 android:id="@+id/radio\_dog"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="onRadioButtonClicked"  
 android:text="Собака" />  
 <RadioButton  
 android:id="@+id/radio\_cat"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="onRadioButtonClicked"  
 android:text="Кошка" />  
 <RadioButton  
 android:id="@+id/radio\_rabbit"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="onRadioButtonClicked"  
 android:text="Кролик" />  
 </RadioGroup>  
  
 <EditText  
 android:id="@+id/user\_name"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Введите имя"/>  
 <ListView  
 android:id="@android:id/list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:longClickable="true"/>  
</LinearLayout>

Листинг 3 – DBActivity.kt

package com.example.lab\_3  
  
import android.content.Context  
import android.content.Intent  
import android.graphics.Rect  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.util.TypedValue  
import android.view.ContextMenu  
import android.view.MenuItem  
import android.view.View  
import android.widget.Button  
import android.widget.EditText  
import android.widget.Toast  
import androidx.recyclerview.widget.LinearLayoutManager  
import androidx.recyclerview.widget.RecyclerView  
import android.view.inputmethod.InputMethodManager  
  
class DBActivity : AppCompatActivity() {  
 private lateinit var recyclerView: RecyclerView  
 private lateinit var userAdapter: UserAdapter  
 private lateinit var dbHandler: DBHandler  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.activity\_dbactivity)  
  
 var btnInsert = findViewById<Button>(R.id.btnInsert)  
 var etvName = findViewById<EditText>(R.id.etvName)  
 var etvAge = findViewById<EditText>(R.id.etvAge)  
  
 btnInsert.setOnClickListener {  
 if (etvName.text.toString().isNotEmpty() &&  
 etvAge.text.toString().isNotEmpty()  
 ) {  
 var user = User(etvName.text.toString(), etvAge.text.toString().toInt())  
 etvName.text.clear()  
 etvAge.text.clear()  
  
 dbHandler.insertData(user)  
 updateUsers()  
  
 val imm = getSystemService(Context.INPUT\_METHOD\_SERVICE) as InputMethodManager  
 imm.hideSoftInputFromWindow(it.windowToken, 0)  
 } else {  
 Toast.makeText(this, "Please fill all fields", Toast.LENGTH\_SHORT).show()  
 }  
 }  
  
 recyclerView = findViewById(R.id.recycler\_view)  
 recyclerView.setHasFixedSize(true)  
 recyclerView.layoutManager = LinearLayoutManager(this)  
  
 dbHandler = DBHandler(this)  
 var users = dbHandler.readData()  
  
 userAdapter = UserAdapter(users)  
 recyclerView.adapter = userAdapter  
  
 userAdapter.onItemClick = {  
 Toast.makeText(this, "User ID: ${it.id}", Toast.LENGTH\_LONG).show()  
  
 val intent = Intent(this, EditUserActivity::class.java)  
 intent.putExtra("USER\_ID", it.id)  
 startActivity(intent)  
 }  
  
  
 recyclerView.addItemDecoration(object : RecyclerView.ItemDecoration() {  
 override fun getItemOffsets(outRect: Rect, view: View, parent: RecyclerView, state: RecyclerView.State) {  
 super.getItemOffsets(outRect, view, parent, state)  
  
 val position = parent.getChildAdapterPosition(view)  
 val isLastItem = position == state.itemCount - 1  
  
 if (!isLastItem) {  
 outRect.bottom = 10.dpToPx(parent.context)  
 }  
 }  
 })  
  
 registerForContextMenu(recyclerView)  
 }  
  
 override fun onResume() {  
 super.onResume()  
 updateUsers()  
 }  
  
 private fun updateUsers() {  
 val users = dbHandler.readData()  
 userAdapter.updateUsers(users)  
 }  
  
 override fun onCreateContextMenu(menu: ContextMenu, v: View, menuInfo: ContextMenu.ContextMenuInfo?) {  
 super.onCreateContextMenu(menu, v, menuInfo)  
 menu.add(0, v.id, 0, "Удалить")  
 menu.add(0, v.id, 0, "Изменить")  
 }  
  
 override fun onContextItemSelected(item: MenuItem): Boolean {  
 if (item.title == "Удалить") {  
 val user = userAdapter.getSelectedUser()  
 dbHandler.deleteById(user.id)  
  
 val users = dbHandler.readData()  
 userAdapter.updateUsers(users)  
 }  
 if(item.title == "Изменить"){  
 val user = userAdapter.getSelectedUser()  
 Toast.makeText(this, "User ID: ${user.id}", Toast.LENGTH\_LONG).show()  
  
 val intent = Intent(this, EditUserActivity::class.java)  
 intent.putExtra("USER\_ID", user.id)  
 startActivity(intent)  
 }  
 return super.onContextItemSelected(item)  
 }  
  
 private fun Int.dpToPx(context: Context): Int {  
 val metrics = context.resources.displayMetrics  
 return TypedValue.applyDimension(TypedValue.COMPLEX\_UNIT\_DIP, this.toFloat(), metrics)  
 .toInt()  
 }  
}

Листинг 4 – activity\_dbactivity.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="match\_parent"  
 android:orientation="vertical"  
 android:padding="20dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
 <EditText  
 android:id="@+id/etvName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"  
 android:hint="Name" />  
 <EditText  
 android:id="@+id/etvAge"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"  
 android:inputType="number"  
 android:hint="Age" />  
 <Button  
 android:id="@+id/btnInsert"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"  
 android:text="Insert" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:weightSum="3">  
 </LinearLayout>  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="10dp">  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recycler\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
 </androidx.constraintlayout.widget.ConstraintLayout>  
</LinearLayout>

Листинг 5 – 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:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<androidx.cardview.widget.CardView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

android:padding="20dp">

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:id="@+id/text\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:textSize="16sp"

android:padding="10dp"

android:text="TextView"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.cardview.widget.CardView>

</androidx.constraintlayout.widget.ConstraintLayout>

Листинг 6 – EditUserActivity.kt

package com.example.lab\_3  
  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.widget.Button  
import android.widget.EditText  
import android.widget.Toast  
  
class EditUserActivity : AppCompatActivity() {  
 private lateinit var dbHandler: DBHandler  
 private lateinit var user: User  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.activity\_edit\_user)  
  
 var nameEditText: EditText = findViewById(R.id.name\_edit\_text)  
 var ageEditText: EditText = findViewById(R.id.age\_edit\_text)  
 var saveButton: Button = findViewById(R.id.save\_button)  
  
 dbHandler = DBHandler(this)  
  
 val userId = intent.getIntExtra("USER\_ID", -1)  
 if (userId == -1) {  
 Toast.makeText(this, "Invalid user ID", Toast.LENGTH\_LONG).show()  
 finish()  
 return  
 }  
  
 user = dbHandler.getUser(userId)  
 nameEditText.setText(user.name)  
 ageEditText.setText(user.age.toString())  
  
 saveButton.setOnClickListener {  
 val name = nameEditText.text.toString()  
 val age = ageEditText.text.toString().toIntOrNull()  
 if (name.isBlank() || age == null) {  
 Toast.makeText(this, "Please enter valid data", Toast.LENGTH\_LONG).show()  
 return@setOnClickListener  
 }  
  
 user.id = userId  
 user.name = name  
 user.age = age  
 dbHandler.updateUser(user)  
  
 Toast.makeText(this, "User updated", Toast.LENGTH\_LONG).show()  
 finish()  
 }  
 }  
}

Листинг 7 – activtity\_edit\_user.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <EditText  
 android:id="@+id/name\_edit\_text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Name" />  
  
 <EditText  
 android:id="@+id/age\_edit\_text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Age"  
 android:inputType="number" />  
  
 <Button  
 android:id="@+id/save\_button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Save" />  
  
</LinearLayout>

Листинг 8 – DBHandler.kt

package com.example.lab\_3  
  
import android.content.Context  
import android.content.ContentValues  
import android.database.sqlite.SQLiteDatabase  
import android.database.sqlite.SQLiteOpenHelper  
import android.widget.Toast  
  
const val DATABASE\_NAME = "MyDB"  
const val TABLE\_NAME = "Users"  
const val COL\_NAME = "name"  
const val COL\_AGE = "age"  
const val COL\_ID = "id"  
  
class DBHandler(var context: Context) : SQLiteOpenHelper(context, DATABASE\_NAME, null, 1) {  
 override fun onCreate(db: SQLiteDatabase?) {  
 val createTable = "CREATE TABLE " + TABLE\_NAME +" (" +  
 COL\_ID +" INTEGER PRIMARY KEY AUTOINCREMENT," +  
 COL\_NAME + " VARCHAR(256)," +  
 COL\_AGE +" INTEGER)"  
  
 db?.execSQL(createTable)  
  
 if (db != null) {  
 insertStaticData(db)  
 }  
 }  
  
 private fun insertStaticData(db: SQLiteDatabase) {  
 val users = listOf(  
 User("Misha", 24),  
 User("Dima", 22),  
 User("Lesha", 20),  
 User("Sergey", 17),  
 User("Alesya", 14)  
 )  
 for (user in users) {  
 val cv = ContentValues()  
 cv.put(COL\_NAME, user.name)  
 cv.put("age", user.age)  
 db.insert(TABLE\_NAME, null, cv)  
 }  
 }  
  
 override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {  
 db?.execSQL("DROP TABLE IF EXISTS $TABLE\_NAME")  
 onCreate(db)  
 }  
  
 fun insertData(user: User) {  
 val db = this.writableDatabase  
 var cv = ContentValues()  
 cv.put(COL\_NAME, user.name)  
 cv.put(COL\_AGE, user.age)  
 var result = db.insert(TABLE\_NAME, null, cv)  
  
 if (result == (-1).toLong()) {  
 Toast.makeText(context, "Failed", Toast.LENGTH\_SHORT).show()  
 } else {  
 Toast.makeText(context, "Success", Toast.LENGTH\_SHORT).show()  
 }  
 }  
  
 fun readData() : MutableList<User>{  
 var list : MutableList<User> = ArrayList()  
  
 val db = this.readableDatabase  
 val query = "Select \* from $TABLE\_NAME"  
 val result = db.rawQuery(query,null)  
 if(result.moveToFirst()){  
 do {  
 var user = User()  
 user.id = result.getString(result.getColumnIndexOrThrow(COL\_ID)).toInt()  
 user.name = result.getString(result.getColumnIndexOrThrow(COL\_NAME))  
 user.age = result.getString(result.getColumnIndexOrThrow(COL\_AGE)).toInt()  
 list.add(user)  
 } while (result.moveToNext())  
 }  
  
 result.close()  
 db.close()  
 return list  
 }  
  
 fun getUser(id: Int): User {  
 val db = this.readableDatabase  
 val cursor = db.query("users", null, "id = ?",  
 arrayOf(id.toString()), null, null, null)  
  
 if (cursor.moveToFirst()) {  
 val name = cursor.getString(cursor.getColumnIndexOrThrow("name"))  
 val age = cursor.getInt(cursor.getColumnIndexOrThrow("age"))  
 return User(name, age)  
 } else {  
 throw IllegalArgumentException("No user with ID $id")  
 }  
 }  
  
 fun updateUser(user: User) {  
 val db = this.writableDatabase  
 val values = ContentValues().apply {  
 put("name", user.name)  
 put("age", user.age)  
 }  
 db.update("users", values, "id = ?", arrayOf(user.id.toString()))  
 }  
  
 fun deleteById(id: Int): Boolean {  
 val db = this.writableDatabase  
 val success = db.delete(TABLE\_NAME, "$COL\_ID=?", arrayOf(id.toString()))  
 db.close()  
 return Integer.parseInt("$success") != -1  
 }  
}

Листинг 9 – User.kt

package com.example.lab\_3  
  
class User {  
 var id = 0  
 var name = ""  
 var age = 0  
  
 constructor() {}  
  
 constructor(name: String, age: Int) {  
 this.name = name  
 this.age = age  
 }  
  
 constructor(id: Int, name: String, age: Int) {  
 this.id = id  
 this.name = name  
 this.age = age  
 }  
}

Листинг 10 – UserAdapter.kt

package com.example.lab\_3  
  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.TextView  
import androidx.recyclerview.widget.RecyclerView  
  
class UserAdapter(private var users: List<User>) : RecyclerView.Adapter<UserAdapter.UserViewHolder>() {  
 var onItemClick : ((User) -> Unit)? = null  
 var onItemLongClick : ((User) -> Unit)? = null  
 var selectedPosition = -1  
  
 class UserViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  
 val textView: TextView = itemView.findViewById(R.id.text\_view)  
 }  
  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): UserViewHolder {  
 val view = LayoutInflater.from(parent.context).inflate(R.layout.list\_item, parent, false)  
 return UserViewHolder(view)  
 }  
  
 override fun getItemCount(): Int {  
 return users.size  
 }  
  
 override fun onBindViewHolder(holder: UserViewHolder, position: Int) {  
 val user = users[position]  
 holder.textView.text = "${user.name}, ${user.age}"  
  
 holder.itemView.setOnClickListener {  
 onItemClick?.invoke(user)  
 }  
  
 holder.itemView.setOnLongClickListener {  
 selectedPosition = position  
 onItemLongClick?.invoke(user)  
 false  
 }  
 }  
  
 fun getSelectedUser(): User {  
 return users[selectedPosition]  
 }  
  
 fun updateUsers(newUsers: List<User>) {  
 this.users = newUsers  
 notifyDataSetChanged()  
 }  
}

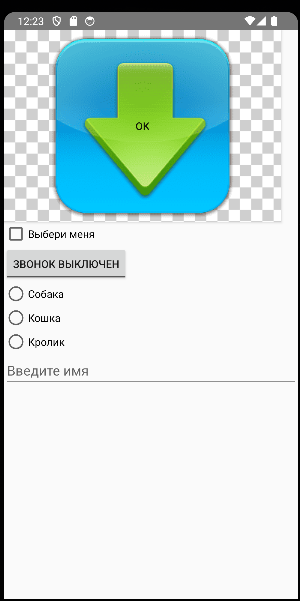


Рисунок 1 – Демонстрация главной активности

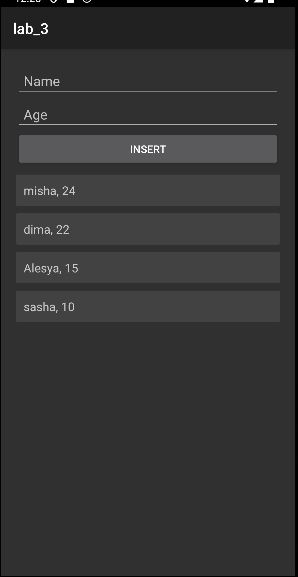


Рисунок 2 – Демонстрация данных базы данных

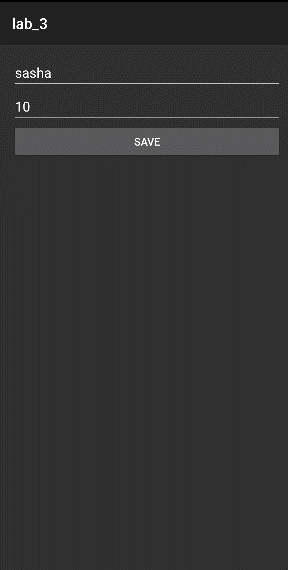


Рисунок 3 – Демонстрация активности изменения данных

Вывод: В результате выполнения лабораторной работы я научился использовать в интерфейсе пользователя различные управляющие элементы: кнопки с изображениями, радиокнопки, чекбоксы и другие, а также взаимодействовать с базой данных в Android приложении.