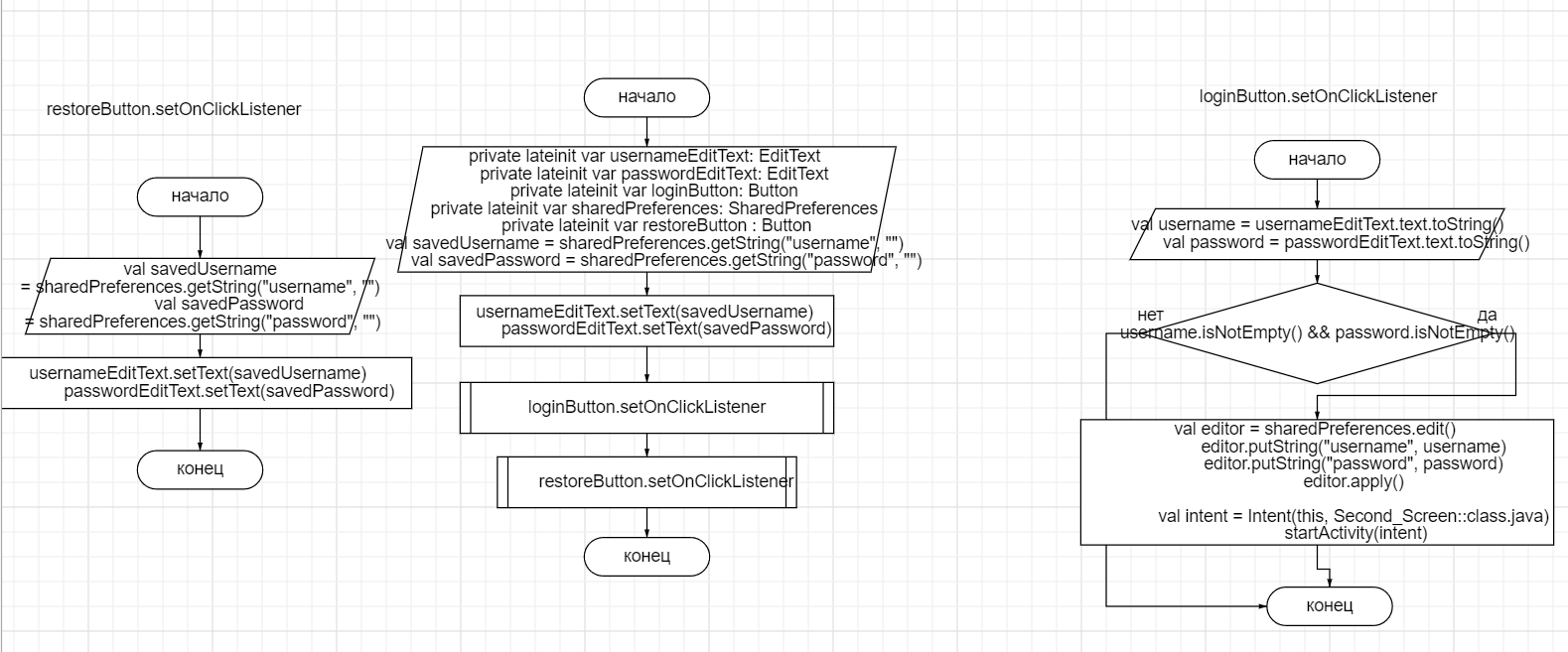
Отчет по практической работе 22\_2.

Подготовил Порядин Андрей ПР-31

Экран ActivityMain.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="match\_parent"  
 tools:context=".MainActivity"  
 android:background="@drawable/img">  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/numbersinfo"  
 android:textSize="30dp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.043" />  
  
 <EditText  
 android:id="@+id/usernameEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/username"  
 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="0.345"  
 />  
  
 <EditText  
 android:id="@+id/passwordEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/password"  
 android:inputType="textPassword"  
 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="0.411" />  
  
 <Button  
 android:id="@+id/loginButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="75dp"  
 android:text="@string/log"  
 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="0.536" />  
  
 <Button  
 android:id="@+id/restoreButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/retake"  
 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="0.625" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

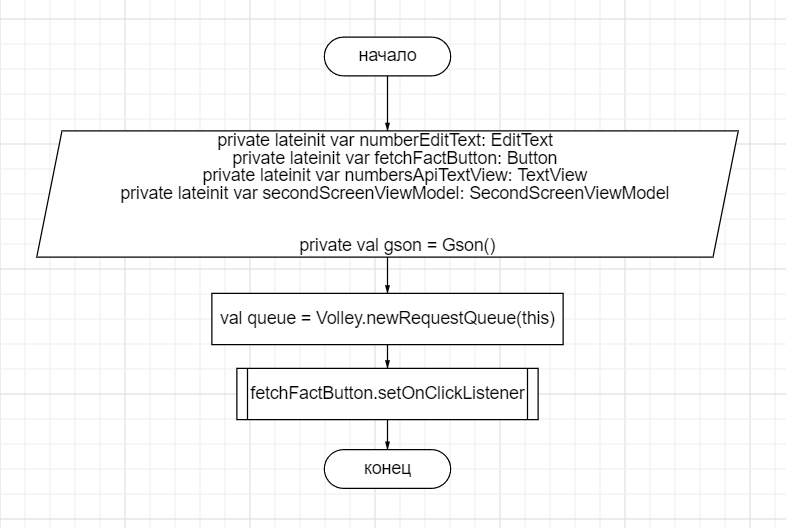
package com.example.myprak  
  
import android.content.Context  
import android.content.Intent  
import android.content.SharedPreferences  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.widget.Button  
import android.widget.EditText  
  
class MainActivity : AppCompatActivity() {  
  
 private lateinit var usernameEditText: EditText  
 private lateinit var passwordEditText: EditText  
 private lateinit var loginButton: Button  
 private lateinit var sharedPreferences: SharedPreferences  
 private lateinit var restoreButton : Button  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 usernameEditText = findViewById(R.id.*usernameEditText*)  
 passwordEditText = findViewById(R.id.*passwordEditText*)  
 loginButton = findViewById(R.id.*loginButton*)  
 restoreButton = findViewById(R.id.*restoreButton*)  
 sharedPreferences = getSharedPreferences("user\_info", Context.*MODE\_PRIVATE*)  
  
  
 val savedUsername = sharedPreferences.getString("username", "")  
 val savedPassword = sharedPreferences.getString("password", "")  
  
 usernameEditText.setText(savedUsername)  
 passwordEditText.setText(savedPassword)  
  
 loginButton.setOnClickListener **{** val username = usernameEditText.*text*.toString()  
 val password = passwordEditText.*text*.toString()  
  
 if (username.*isNotEmpty*() && password.*isNotEmpty*()) {  
 val editor = sharedPreferences.edit()  
 editor.putString("username", username)  
 editor.putString("password", password)  
 editor.apply()  
  
 val intent = Intent(this, Second\_Screen::class.*java*)  
 startActivity(intent)  
 } else {  
  
 }  
 **}** restoreButton.setOnClickListener **{** val savedUsername = sharedPreferences.getString("username", "")  
 val savedPassword = sharedPreferences.getString("password", "")  
  
 usernameEditText.setText(savedUsername)  
 passwordEditText.setText(savedPassword)  
 **}** }  
}

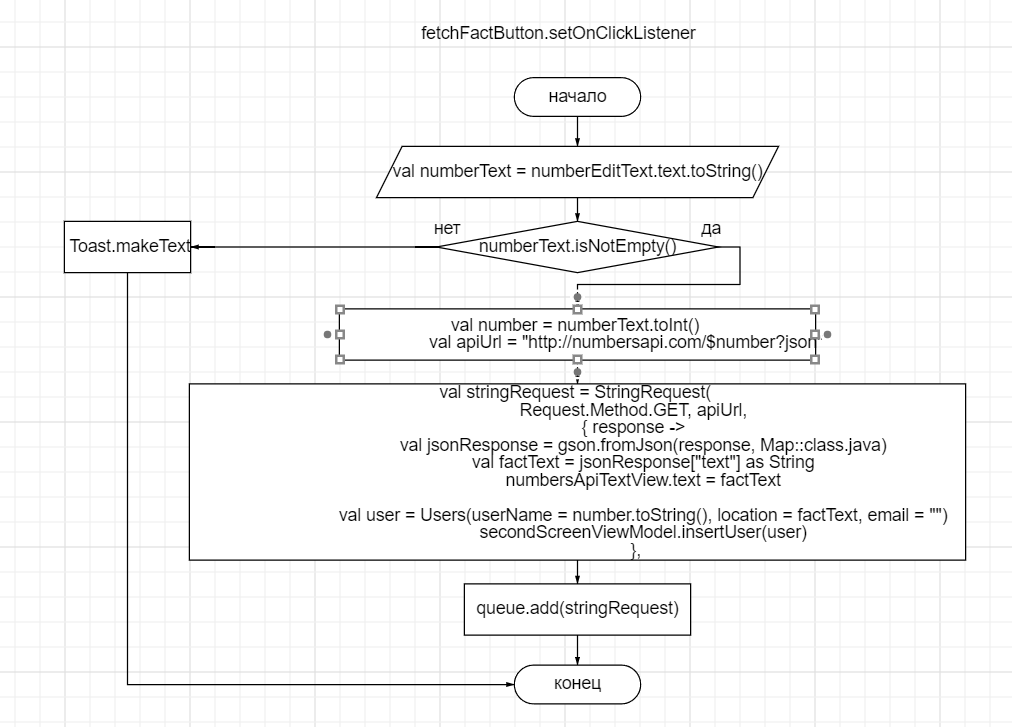


Экран Second\_Screen

<?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="match\_parent"  
 tools:context=".Second\_Screen"  
android:background="@drawable/img"  
 >  
  
  
 <TextView  
 android:textStyle="bold"  
 android:layout\_width="match\_parent"  
 android:layout\_height="80dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/numbersApiTextView"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:text="@string/yourfact"  
 android:textSize="24dp"  
 android:textAlignment="center"  
 />  
 <EditText  
 android:id="@+id/numberEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/writenumber"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.39" />  
  
 <Button  
 android:id="@+id/fetchFactButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/fetchfact"  
 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="0.909" />  
  
 <Button  
 android:id="@+id/DataBaseButton"  
 android:onClick="db\_click"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/database"  
 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="0.989" />  
  
  
  
 <TextView  
 android:id="@+id/numbersApiTextView"  
 android:layout\_width="424dp"  
 android:layout\_height="137dp"  
 android:text=""  
 android:textSize="24dp"  
 android:textStyle="bold"  
 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="0.139" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

package com.example.myprak  
import android.annotation.SuppressLint  
import android.content.Intent  
import android.os.Bundle  
import android.view.View  
import android.widget.Button  
import android.widget.EditText  
import android.widget.TextView  
import android.widget.Toast  
import androidx.appcompat.app.AppCompatActivity  
import androidx.lifecycle.ViewModelProvider  
import androidx.room.Room  
import com.android.volley.Request  
import com.android.volley.Response  
import com.android.volley.VolleyError  
import com.android.volley.toolbox.StringRequest  
import com.android.volley.toolbox.Volley  
import com.google.gson.Gson  
import kotlinx.coroutines.CoroutineScope  
import kotlinx.coroutines.Dispatchers  
import kotlinx.coroutines.launch  
import kotlinx.coroutines.runBlocking  
import kotlinx.coroutines.withContext  
  
  
class Second\_Screen : AppCompatActivity() {  
 private lateinit var numberEditText: EditText  
 private lateinit var fetchFactButton: Button  
 private lateinit var numbersApiTextView: TextView  
 private lateinit var secondScreenViewModel: SecondScreenViewModel  
  
  
 private val gson = Gson()  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_second\_screen*)  
  
 secondScreenViewModel = ViewModelProvider(this).get(SecondScreenViewModel::class.*java*)  
 numberEditText = findViewById(R.id.*numberEditText*)  
 fetchFactButton = findViewById(R.id.*fetchFactButton*)  
 numbersApiTextView = findViewById(R.id.*numbersApiTextView*)  
  
 val queue = Volley.newRequestQueue(this)  
 fetchFactButton.setOnClickListener **{** try{  
 val numberText = numberEditText.*text*.toString()  
  
 if (numberText.*isNotEmpty*()) {  
  
 val number = numberText.*toInt*()  
 val apiUrl = "http://numbersapi.com/$number?json"  
  
 val stringRequest = StringRequest(  
 Request.Method.*GET*, apiUrl,  
 **{** response **->** val jsonResponse = gson.fromJson(response, Map::class.*java*)  
 val factText = jsonResponse["text"] as String  
 numbersApiTextView.*text* = factText  
  
 val user = Users(userName = number.toString(), location = factText, email = "")  
 secondScreenViewModel.insertUser(user)  
 **}**,  
 **{** error **->** if (error is VolleyError) {  
 if (error.networkResponse != null && error.networkResponse.data != null) {  
 val errorMessage = *String*(error.networkResponse.data)  
 numbersApiTextView.*text* = "Ошибка: $errorMessage"  
 } else {  
 numbersApiTextView.*text* = "Ошибка: ${error.message}"  
 }  
 }  
 **}** )  
 queue.add(stringRequest)  
 } else {  
 Toast.makeText(this, "Номер не указан", Toast.*LENGTH\_SHORT*).show()  
 }  
 }catch(e: Exception){  
 Toast.makeText(this, "Неверный формат", android.widget.Toast.*LENGTH\_SHORT*).show()  
 }  
  
 **}** }  
 fun db\_click(view: View) {  
 val intent = Intent(this,DataBase::class.*java*)  
 startActivity(intent)  
 }  
}

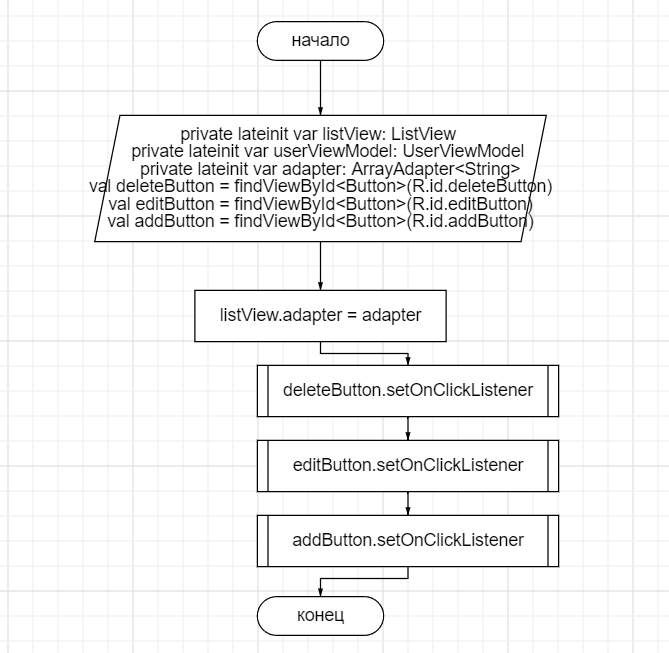


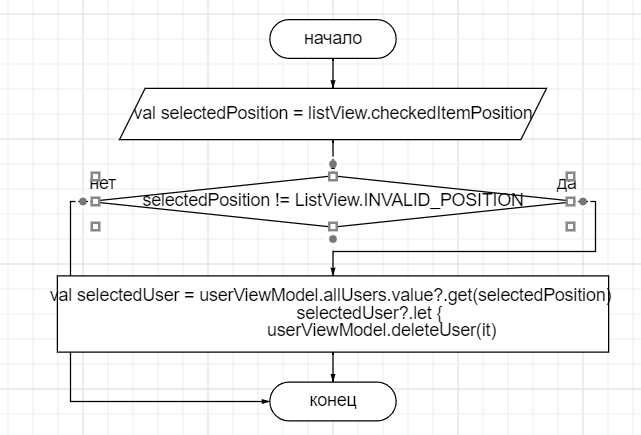


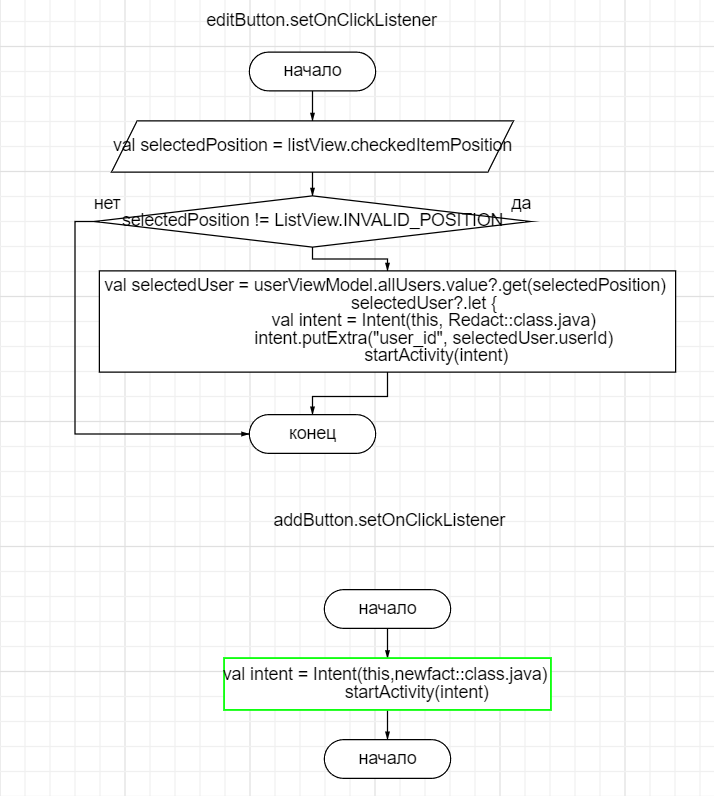
Экран Database

<?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"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".DataBase"  
 android:orientation="vertical">  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/factlist"  
 android:textSize="18sp"  
 android:layout\_marginBottom="16dp"/>  
  
 <ListView  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1"  
 android:id="@+id/listView"  
 android:choiceMode="singleChoice"  
 />  
  
<LinearLayout  
 android:layout\_width="match\_parent"  
  
 android:layout\_height="100dp">  
  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/delete"  
 android:id="@+id/deleteButton"  
 android:layout\_marginTop="16dp"/>  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/red"  
 android:id="@+id/editButton"  
 android:layout\_marginTop="16dp"/>  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/add"  
 android:id="@+id/addButton"  
 android:layout\_marginTop="16dp"/>  
</LinearLayout>  
  
  
</LinearLayout>

package com.example.myprak  
  
  
import android.app.Application  
import android.content.Intent  
import android.os.Bundle  
import android.widget.ArrayAdapter  
import android.widget.Button  
import android.widget.ListView  
import androidx.appcompat.app.AppCompatActivity  
import androidx.lifecycle.AndroidViewModel  
import androidx.lifecycle.LiveData  
import androidx.lifecycle.ViewModel  
import androidx.lifecycle.ViewModelProvider  
import androidx.lifecycle.*lifecycleScope*import androidx.room.Room  
import kotlinx.coroutines.Dispatchers  
import kotlinx.coroutines.GlobalScope  
import kotlinx.coroutines.launch  
import kotlinx.coroutines.withContext  
  
class DataBase : AppCompatActivity() {  
 private lateinit var listView: ListView  
 private lateinit var userViewModel: UserViewModel  
 private lateinit var adapter: ArrayAdapter<String>  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_data\_base*)  
  
 listView = findViewById(R.id.*listView*)  
 userViewModel = ViewModelProvider(this).get(UserViewModel::class.*java*)  
 adapter = ArrayAdapter(this, android.R.layout.*simple\_list\_item\_1*)  
  
 listView.*adapter* = adapter  
  
 userViewModel.allUsers.observe(this, **{** users **->** val facts = users.*map* **{ it**.location **}** adapter.clear()  
 adapter.addAll(facts)  
 **}**)  
  
 val deleteButton = findViewById<Button>(R.id.*deleteButton*)  
 deleteButton.setOnClickListener **{** val selectedPosition = listView.*checkedItemPosition* if (selectedPosition != ListView.*INVALID\_POSITION*) {  
 val selectedUser = userViewModel.allUsers.*value*?.get(selectedPosition)  
 selectedUser?.*let* **{** userViewModel.deleteUser(**it**)  
 **}** }  
 **}** val editButton = findViewById<Button>(R.id.*editButton*)  
 editButton.setOnClickListener **{** val selectedPosition = listView.*checkedItemPosition* if (selectedPosition != ListView.*INVALID\_POSITION*) {  
 val selectedUser = userViewModel.allUsers.*value*?.get(selectedPosition)  
 selectedUser?.*let* **{** val intent = Intent(this, Redact::class.*java*)  
 intent.putExtra("user\_id", selectedUser.userId)  
 startActivity(intent)  
 **}** }  
 **}** val addButton = findViewById<Button>(R.id.*addButton*)  
 addButton.setOnClickListener **{** val intent = Intent(this,newfact::class.*java*)  
 startActivity(intent)  
 **}** }  
}



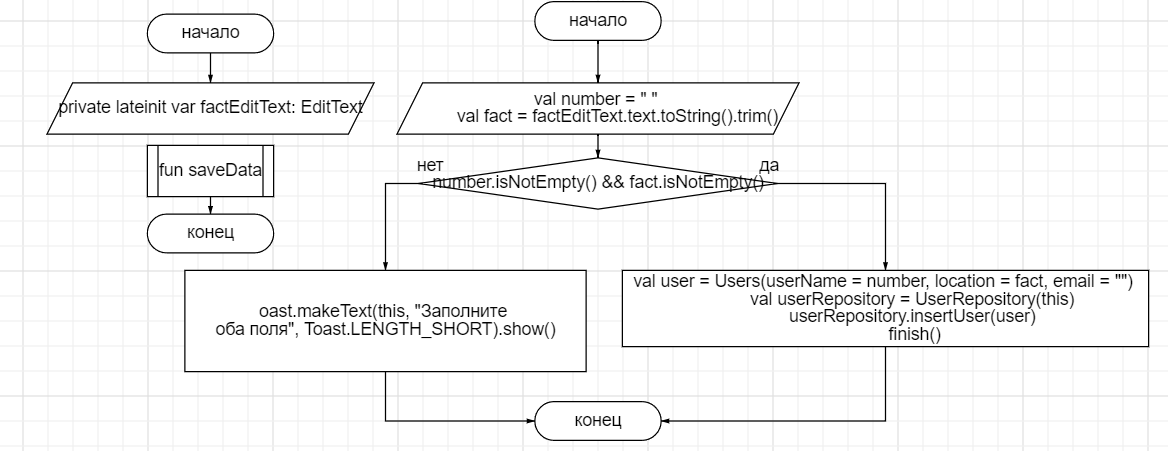




Экран newfact

<?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"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".newfact">  
  
  
  
 <EditText  
 android:id="@+id/factEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/fact" />  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/save"  
 android:onClick="saveData" />  
  
</LinearLayout>

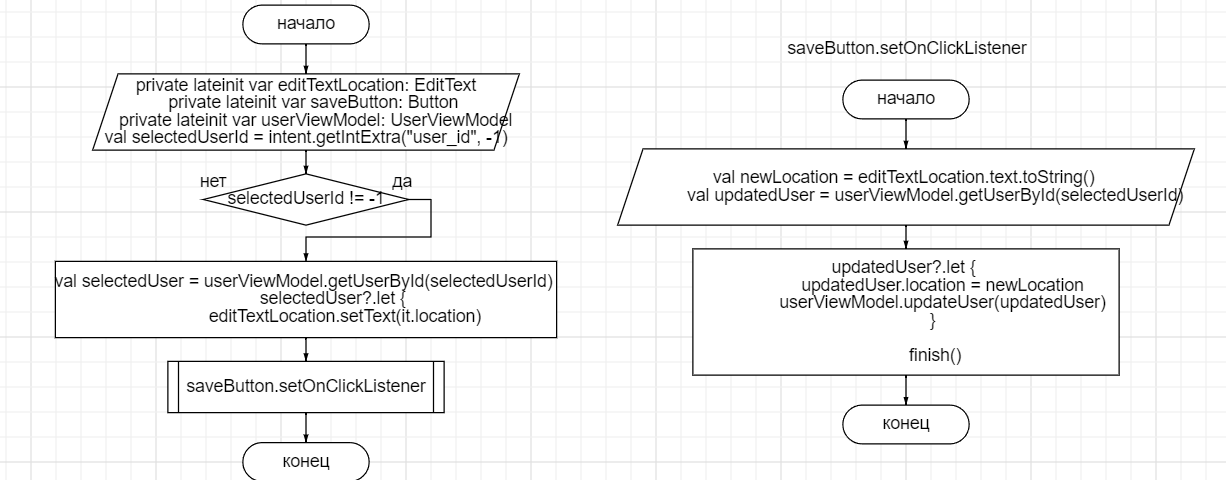
package com.example.myprak  
  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.View  
import android.widget.EditText  
import android.widget.Toast  
  
class newfact : AppCompatActivity() {  
 private lateinit var numberEditText: EditText  
 private lateinit var factEditText: EditText  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_newfact*)  
  
  
 factEditText = findViewById(R.id.*factEditText*)  
  
  
 }  
  
 fun saveData(view: View) {  
 val number = " "  
 val fact = factEditText.*text*.toString().*trim*()  
  
 if (number.*isNotEmpty*() && fact.*isNotEmpty*()) {  
 val user = Users(userName = number, location = fact, email = "")  
 val userRepository = UserRepository(this)  
 userRepository.insertUser(user)  
 finish()  
 } else {  
 Toast.makeText(this, "Заполните оба поля", Toast.*LENGTH\_SHORT*).show()  
 }}  
}



Экран Redact

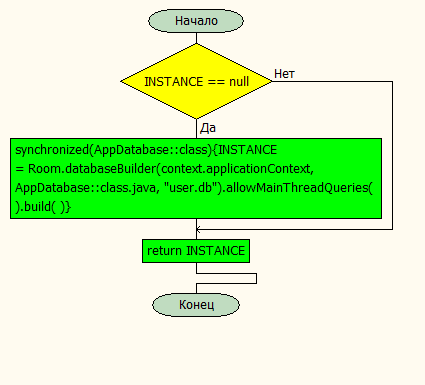
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".Redact">  
  
 <EditText  
 android:id="@+id/editTextLocation"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 />  
  
 <Button  
 android:id="@+id/saveButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintTop\_toBottomOf="@id/editTextLocation"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:text="@string/saveres"  
 />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

package com.example.myprak  
  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.widget.Button  
import android.widget.EditText  
import androidx.lifecycle.ViewModelProvider  
  
class Redact : AppCompatActivity() {  
 private lateinit var editTextLocation: EditText  
 private lateinit var saveButton: Button  
 private lateinit var userViewModel: UserViewModel  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_redact*)  
  
 editTextLocation = findViewById(R.id.*editTextLocation*)  
 saveButton = findViewById(R.id.*saveButton*)  
  
 userViewModel = ViewModelProvider(this).get(UserViewModel::class.*java*)  
  
 val selectedUserId = *intent*.getIntExtra("user\_id", -1)  
  
 if (selectedUserId != -1) {  
 val selectedUser = userViewModel.getUserById(selectedUserId)  
 selectedUser?.*let* **{** editTextLocation.setText(**it**.location)  
 **}** }  
  
 saveButton.setOnClickListener **{** val newLocation = editTextLocation.*text*.toString()  
  
  
 val updatedUser = userViewModel.getUserById(selectedUserId)  
 updatedUser?.*let* **{** updatedUser.location = newLocation  
 userViewModel.updateUser(updatedUser)  
 **}** finish()  
 **}** }  
}



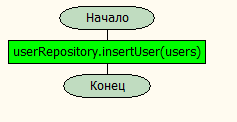
Класс AppDatabase.kt

package com.example.myprak  
  
import androidx.room.Database  
import androidx.room.Room  
import androidx.room.RoomDatabase  
import androidx.room.TypeConverters  
import android.content.Context  
  
@Database(entities = [Users::class], version = 1, exportSchema = false)  
@TypeConverters(Converters::class)  
abstract class AppDatabase : RoomDatabase() {  
  
 abstract fun userDao() : UserDao  
  
 companion object {  
 private var INSTANCE: AppDatabase? = null  
  
 fun getInstance(context: Context): AppDatabase? {  
 if (INSTANCE == null) {  
 *synchronized*(AppDatabase::class) **{** INSTANCE = Room.databaseBuilder(context.*applicationContext*,  
 AppDatabase::class.*java*, "user.db").allowMainThreadQueries()  
 .build()  
 **}** }  
 return INSTANCE  
 }  
 }  
}



Класс SecondScreenViewModel

package com.example.myprak  
  
import android.app.Application  
import androidx.lifecycle.AndroidViewModel  
  
class SecondScreenViewModel(application: Application) : AndroidViewModel(application) {  
 private val userRepository: UserRepository = UserRepository(application)  
  
 fun insertUser(users: Users) {  
 userRepository.insertUser(users)  
 }  
}

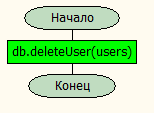
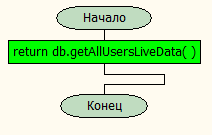
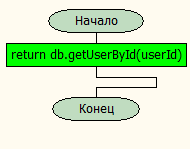
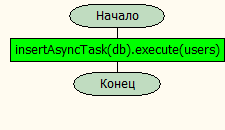
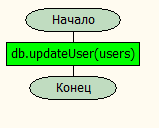


interface UserDao

package com.example.myprak  
  
import androidx.lifecycle.LiveData  
import androidx.room.Dao  
import androidx.room.Delete  
import androidx.room.Insert  
import androidx.room.Query  
import androidx.room.Update  
  
@Dao  
interface UserDao {  
  
 @Insert  
 fun insertUser(users: Users)  
  
 @Query("SELECT \* FROM user")  
 fun getAllUsersLiveData(): LiveData<List<Users>>  
  
 @Update  
 fun updateUser(users: Users)  
  
 @Delete  
 fun deleteUser(users: Users)  
  
 @Query("SELECT \* FROM user WHERE userId = :userId")  
 fun getUserById(userId: Int): Users?  
  
}

class UserRepository

package com.example.myprak  
  
import android.content.Context  
import android.os.AsyncTask  
import androidx.lifecycle.LiveData  
  
class UserRepository(context: Context) {  
  
 var db: UserDao = AppDatabase.getInstance(context)?.userDao()!!  
  
 fun getAllUsersLiveData(): LiveData<List<Users>> {  
 return db.getAllUsersLiveData()  
 }  
 fun getUserById(userId: Int): Users? {  
 return db.getUserById(userId)  
 }  
  
 fun insertUser(users: Users) {  
 insertAsyncTask(db).execute(users)  
 }  
  
 fun updateUser(users: Users) {  
 db.updateUser(users)  
 }  
  
 fun deleteUser(users: Users) {  
 db.deleteUser(users)  
 }  
  
 private class insertAsyncTask internal constructor(private val usersDao: UserDao) :  
 AsyncTask<Users, Void, Void>() {  
  
 override fun doInBackground(vararg params: Users): Void? {  
 usersDao.insertUser(params[0])  
 return null  
 }  
 }  
}

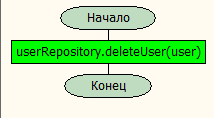
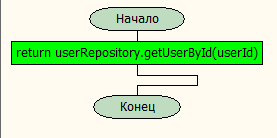
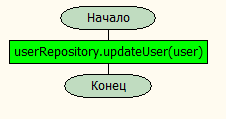
    

Класс Users

package com.example.myprak  
  
import androidx.room.Entity  
import androidx.room.PrimaryKey  
import java.io.Serializable  
  
@Entity(tableName = "user")  
data class Users(@PrimaryKey(autoGenerate = true)var userId: Int? = null,  
 val userName: String, var location: String, val email: String):Serializable

Класс UserViewModel

package com.example.myprak  
  
import android.app.Application  
import androidx.lifecycle.AndroidViewModel  
import androidx.lifecycle.LiveData  
  
class UserViewModel(application: Application) : AndroidViewModel(application) {  
 private val userRepository = UserRepository(application)  
 val allUsers: LiveData<List<Users>> = userRepository.getAllUsersLiveData()  
  
 fun deleteUser(user: Users) {  
 userRepository.deleteUser(user)  
 }  
 fun updateUser(user: Users) {  
 userRepository.updateUser(user)  
 }  
 fun getUserById(userId: Int): Users? {  
 return userRepository.getUserById(userId)  
 }  
}

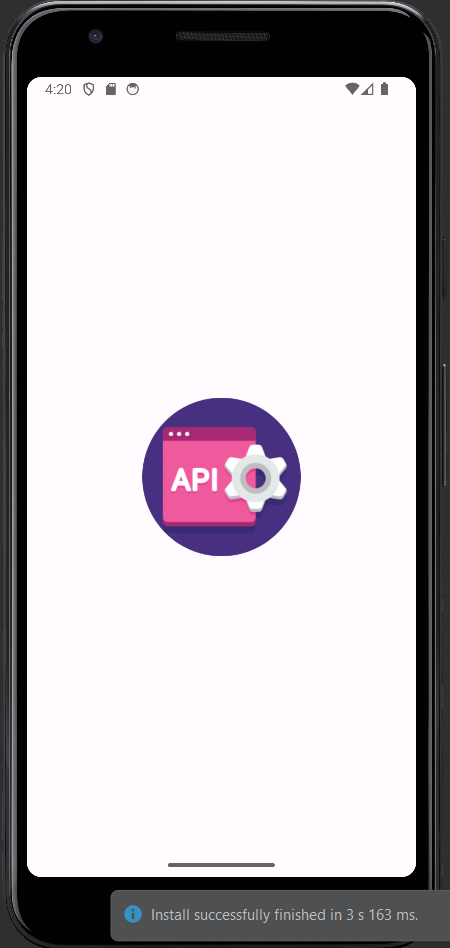
файл strings.xml

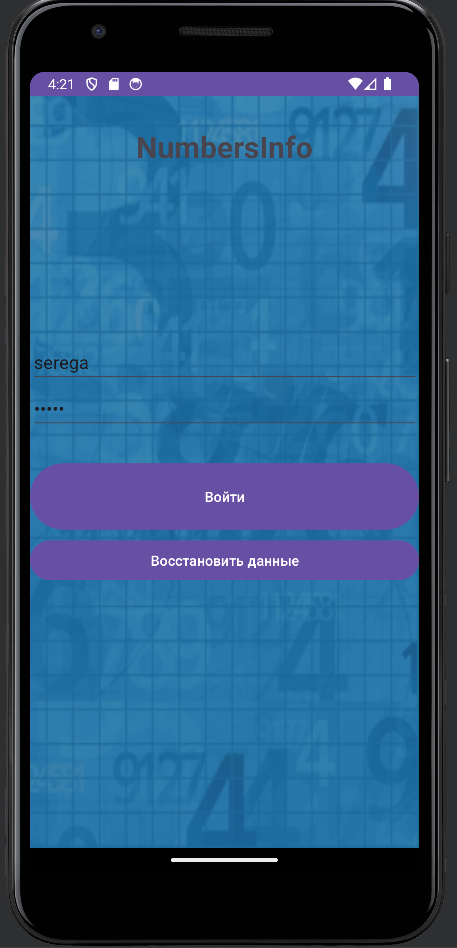
<resources>  
 <string name="app\_name">NumbersAPI</string>  
 <string name="factlist">Список фактов из базы данных:</string>  
 <string name="fact">Факт</string>  
 <string name="save">Сохранить</string>  
 <string name="yourfact">Ваш факт:</string>  
 <string name="writenumber">Введите число</string>  
 <string name="fetchfact">Получить факт</string>  
 <string name="database">База данных</string>  
 <string name="addfuct">Добавить факт вручную</string>  
 <string name="numbersinfo">NumbersInfo</string>  
 <string name="username">Имя пользователя</string>  
 <string name="password">Пароль</string>  
 <string name="log">Войти</string>  
 <string name="retake">Восстановить данные</string>  
 <string name="delete">Удалить</string>  
 <string name="red">Редактировать</string>  
 <string name="add">Добавить</string>  
 <string name="saveres">Сохранить</string>  
</resources>

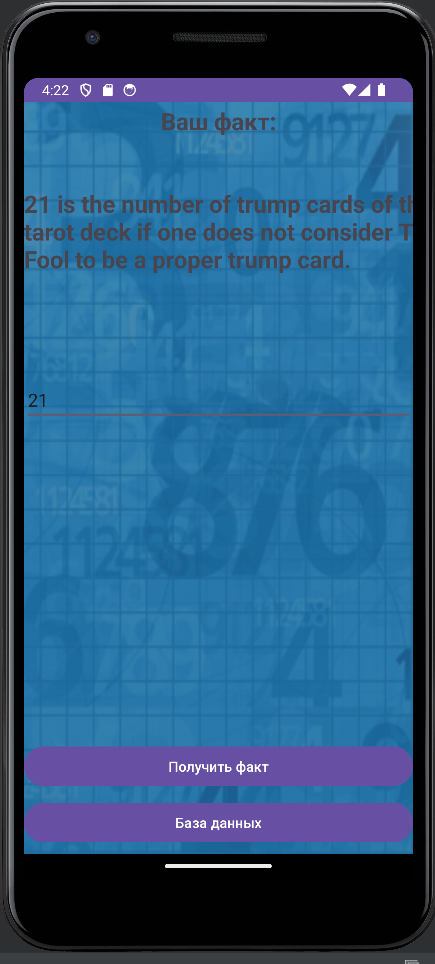
Подключенные библиотеки

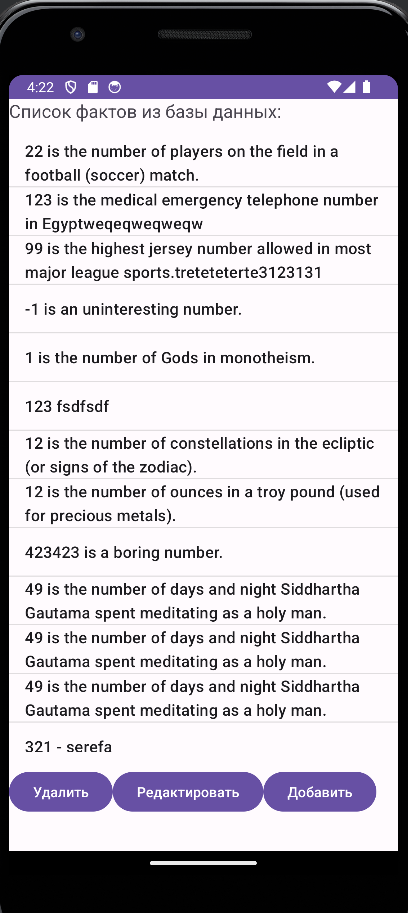
*dependencies* **{** *implementation*("androidx.room:room-ktx:2.5.0")  
 *implementation*("org.jetbrains.kotlinx:kotlinx-coroutines-android:1.5.0")  
 *kapt*("androidx.room:room-compiler:2.5.0")  
 *implementation*("androidx.room:room-runtime:2.5.0")  
 *annotationProcessor*("androidx.room:room-compiler:2.3.0")  
 *implementation*("com.android.volley:volley:1.2.0")  
 *implementation*("com.google.code.gson:gson:2.8.8")  
 *implementation*("androidx.core:core-ktx:1.9.0")  
 *implementation*("androidx.appcompat:appcompat:1.6.1")  
 *implementation*("com.google.android.material:material:1.10.0")  
 *implementation*("androidx.constraintlayout:constraintlayout:2.1.4")  
 *testImplementation*("junit:junit:4.13.2")  
 *androidTestImplementation*("androidx.test.ext:junit:1.1.5")  
 *androidTestImplementation*("androidx.test.espresso:espresso-core:3.5.1")  
  
**}**

Скрины программы :











Блок схемы :