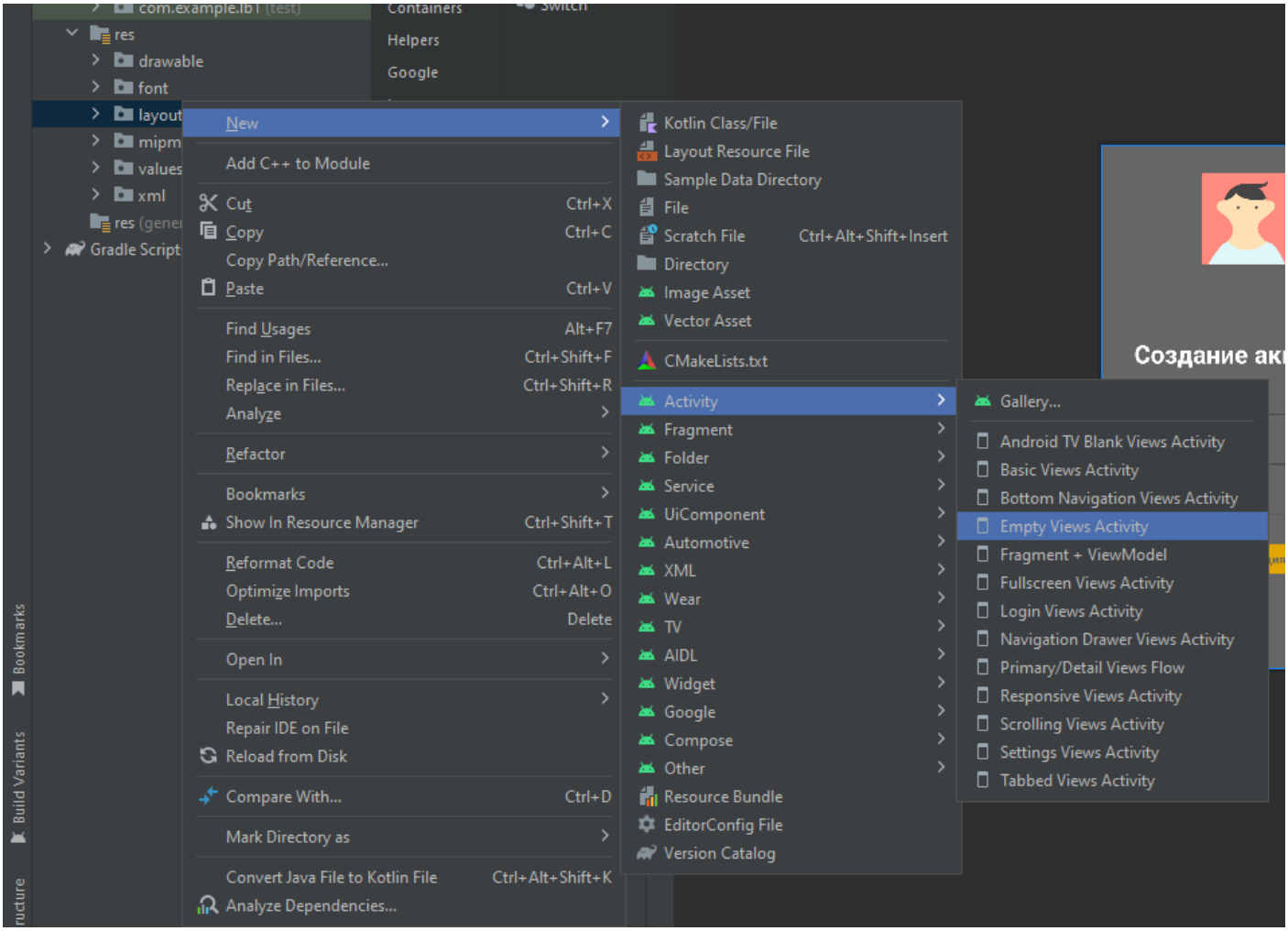
АВТОРИЗАЦИЯ

1. Для авторизации создаем новую страницу – новую активити:

Правой кнопкой мыши на папке layout – New – Activity – Empty Views Activity



Имя – AuthActivity

2. Копируем весь код из activity_main.xml в activity_ auth.xml

3. Заменяем теперь в activity_ auth.xml в коде tools:context на AuthActivity:

```
tools:context=".AuthActivity">
```

В TextView меняем android:text на «Войдите в аккаунт»:

```
android:text="Войдите в аккаунт"
```

4. Удаляем код EditText, где вводим email. Этот блок:

```
<EditText
    android:id="@+id/user_email"
    android:layout_width="240dp"
    android:layout_height="50dp"
    android:ems="10"
    android:layout_marginTop="20dp"
    android:layout_gravity="center"
    android:textColor="#fdfdfd"
    android:inputType="textEmailAddress"
    android:textColorHint="#ada"
    android:hint="Введите email" />
```

Т.е. авторизация будет происходить за счет логина и пароля.

5. На кнопке пишем слово «Войти».

6. Поменяйте задний фон, фон кнопки и цвет текста на кнопке

7. Поменяем id для объектов добавив в конце «_auth»:

```
- android:id="@+id/user_login_auth"

- android:id="@+id/user_pass_auth"

- android:id="@+id/button_auth"
```

8. Создаем текст с возможностью перехода на страницу регистрации.

Копируем код «TextView» и вставляем внизу. Начинаем редактировать:

Размер текста 17sp

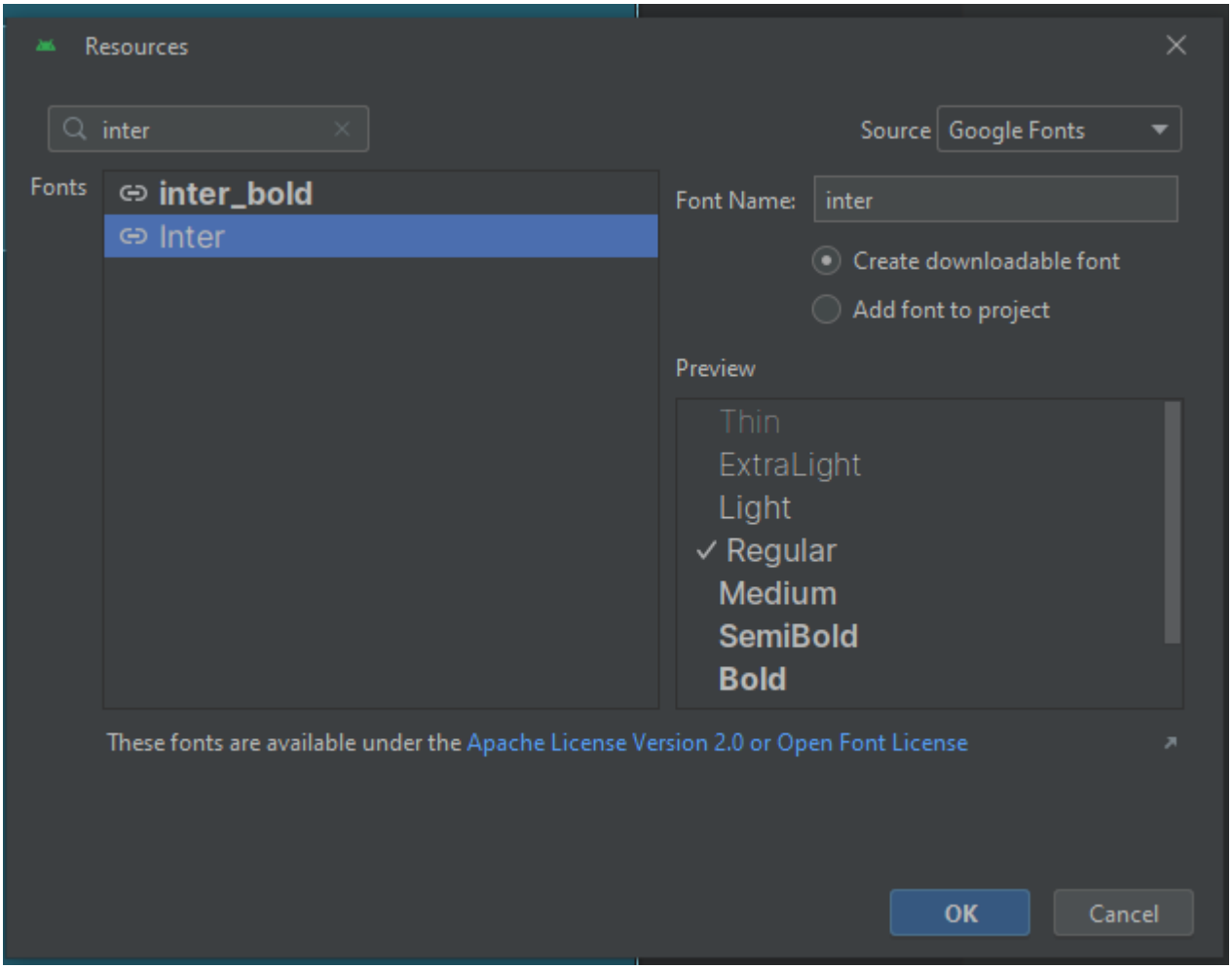
Отступ сверху 120dp

Текст – «Зарегистрироваться»

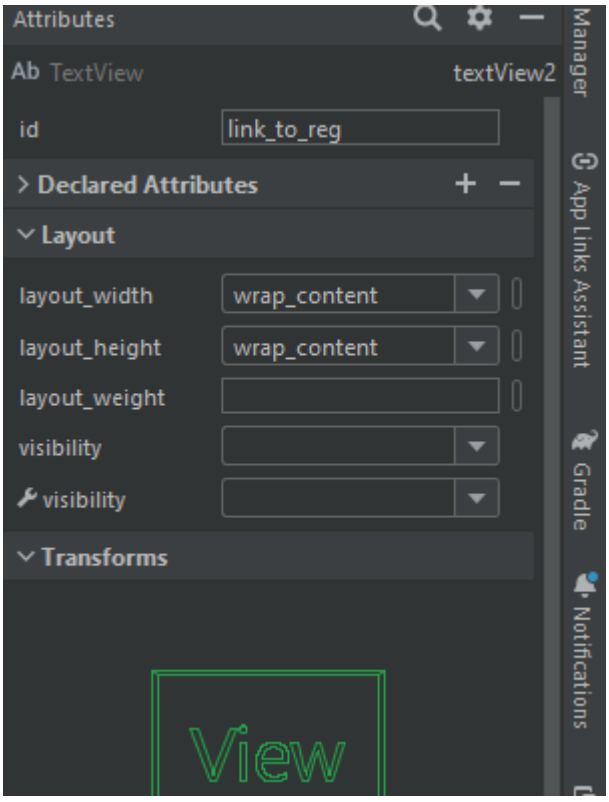
Удаляем стиль «bold»

```
android:textStyle="bold"
```

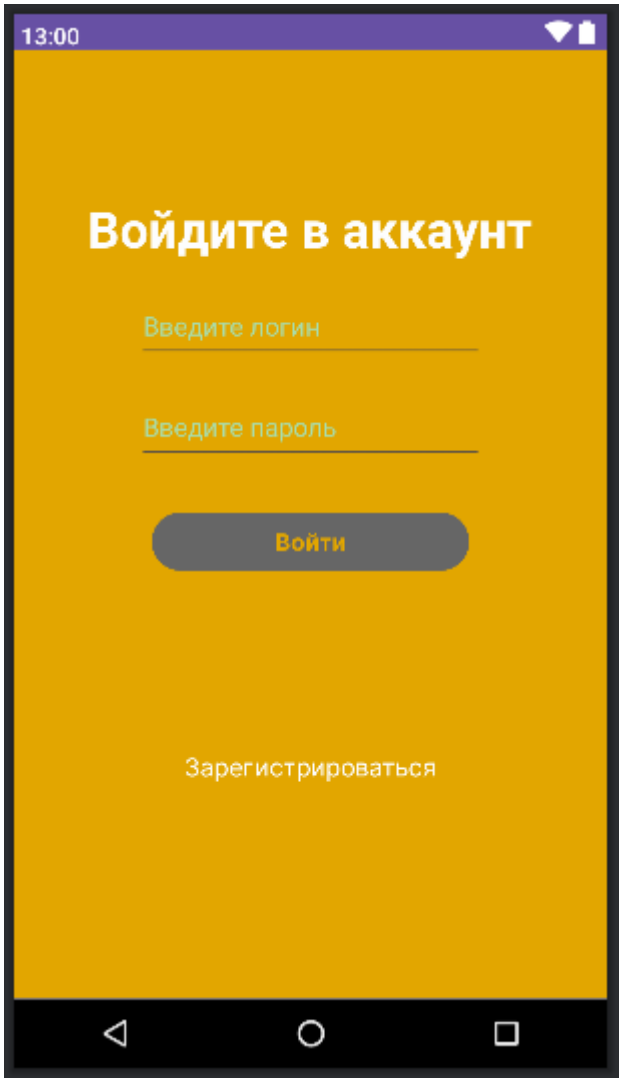
Сделаем стиль текста inter – Regular:



Добавим надписи id в режиме Design «link_to_reg»:



ИТОГ:




```
<?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=".AuthActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#e2a600"
        android:orientation="vertical">



        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginTop="100dp"
            android:gravity="center_horizontal"
            android:text="Войдите в аккаунт"
```

```
            android:textColor="#fdfdfd"
            android:textSize="35sp"
            android:textStyle="bold" />

        <EditText
            android:id="@+id/user_login_auth"
            android:layout_width="240dp"
            android:layout_height="50dp"
            android:ems="10"
            android:layout_marginTop="20dp"
            android:layout_gravity="center"
            android:textColor="#fdfdfd"
            android:inputType="text"
            android:textColorHint="#ada"
            android:hint="Введите логин" />

        <EditText
            android:id="@+id/user_pass_auth"
            android:layout_width="240dp"
            android:layout_height="50dp"
            android:ems="10"
            android:layout_marginTop="20dp"
            android:layout_gravity="center"
            android:textColor="#fdfdfd"
            android:inputType="textPassword"
            android:textColorHint="#ada"
            android:hint="Введите пароль" />

        <Button
            android:id="@+id/button_auth"
            android:layout_width="220dp"
            android:layout_height="wrap_content"
            android:fontFamily="@font/inter_bold"
            android:text="Войти"
            android:textStyle="bold"
            android:textSize="16sp"
            android:textColor="#e2a600"
            android:layout_gravity="center"
            android:backgroundTint="#666"
            android:layout_marginTop="30dp"
            />

        <TextView
            android:id="@+id/link_to_reg"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginTop="120dp"
            android:fontFamily="@font/inter"
            android:gravity="center_horizontal"
            android:text="Зарегистрироваться"
            android:textColor="#fdfdfd"
            android:textSize="17sp" />

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

В файле AuthActivity в коде создаем переменную linkToReg основанную на классе TextView и указываем объект с ID «link_to_reg»:

```
val linkToReg: TextView = findViewById(R.id.link_to_reg)
```

Обратимся к LinkToReg к функции setOnClickListener, которая вещает обработчик при нажатии.

```
linkToReg.setOnClickListener {}
```

При нажатии на текст «Зарегистрироваться» будем выполнять создание новой переменной, в которую будем записывать новую страницу на которую будем переходить. Для выполнения перехода, для указания с какими старицами мы работаем используем класс Intent.

В этот класс передаем контекст в котором мы сейчас (this) находимся и указываем на какую страницу мы переходим – MainActivity.

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

Пишем метод startActivity и передаем объект intent:

```
startActivity(intent)
```

В итоге код в файле AuthActivity:

```
package com.example.lb1

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.TextView

class AuthActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_auth)

        val linkToReg: TextView = findViewById(R.id.link_to_reg)

        linkToReg.setOnClickListener {
            val intent = Intent(this, MainActivity::class.java)
            startActivity(intent)
        }
    }
}
```

Сделаем аналогичный переход с первой страницы «Создание аккаунта»

1. Копируем код TextView с файла «activity_auth.xml» и вставляем на странице «activity_main.xml»

Редактируем вставленный код на странице «activity_main.xml»:

– текст меняем на «Авторизоваться»;

– меняем ID на «link_to_auth»»

2. Переходим в код файла «MainActivity.kt»

Создаем новую переменную linkToAuth, основанную на TextView, и прописываем уже ID «link_to_auth»

```
val linkToAuth: TextView = findViewById(R.id.link_to_auth)
```

Из файла AuthActivity.kt копируем кусок кода

```
linkToReg.setOnClickListener {
    val intent = Intent(this, MainActivity::class.java)
    startActivity(intent)
}
```

и вставляем его в файл «MainActivity.kt».

Заменяем linkToReg на linkToAuth и указываем, что при нажатии будем переходить теперь на активити «AuthActivity»

```
linkToAuth.setOnClickListener {
    val intent = Intent(this, AuthActivity::class.java)
    startActivity(intent)
}
```

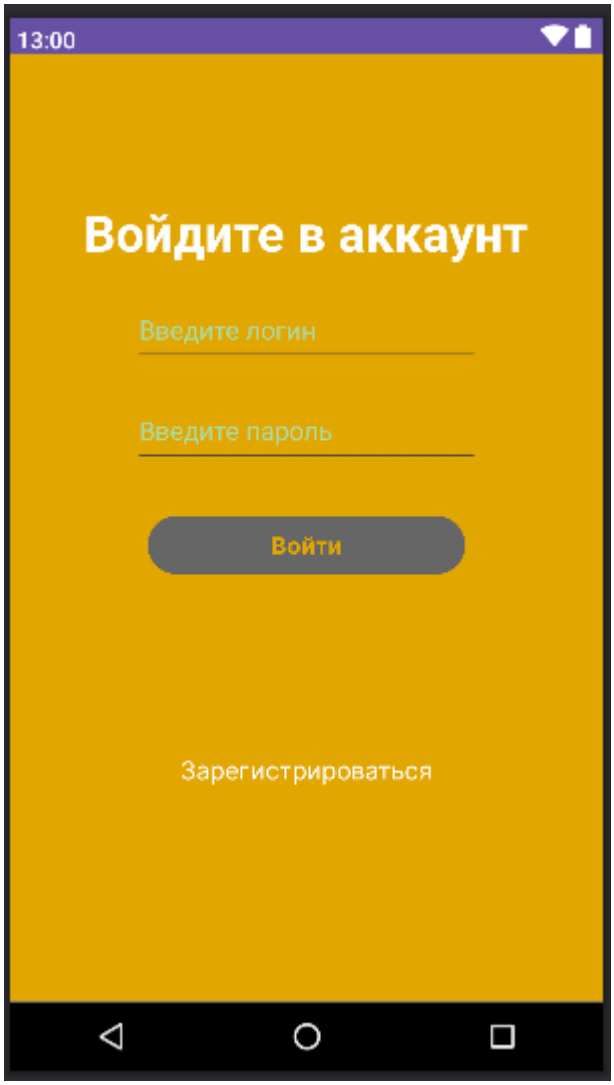
Т.е. будет переход с этой страницы (`this`) на страницу `AuthActivity`

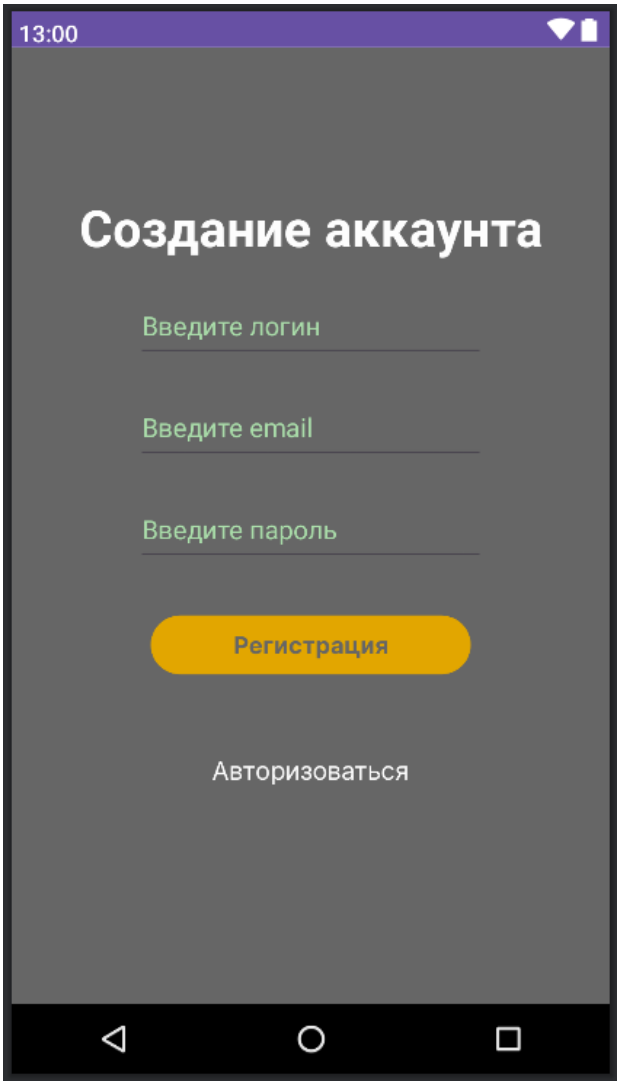
В ИТОГЕ:

```
package com.example.lb1

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast

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

        val userLogin: EditText = findViewById(R.id.user_login)
        val userEmail: EditText = findViewById(R.id.user_email)
        val userPass: EditText = findViewById(R.id.user_pass)
        val button: Button = findViewById(R.id.button_reg)
        val linkToAuth: TextView = findViewById(R.id.link_to_auth)

        linkToAuth.setOnClickListener {
            val intent = Intent(this, AuthActivity::class.java)
            startActivity(intent)
        }
```

```
        button.setOnClickListener {
            val login = userLogin.text.toString().trim()
            val email = userEmail.text.toString().trim()
            val pass = userPass.text.toString().trim()

            if(login == "" || email == "" || pass == "")
                Toast.makeText(this, "Не все поля заполнены",
Toast.LENGTH_LONG).show()
            else {
                val user = User(login, email, pass)

                val db = DbHelper(this, null)
                db.addUser(user)
                Toast.makeText(this, "Пользователь $login добавлен",
Toast.LENGTH_LONG).show()

                userLogin.text.clear()
                userEmail.text.clear()
                userPass.text.clear()
            }
        }
    }
}
```

Запускаем эмулятор и проверяем переходы между активити при нажатии «Авторизоваться», «Зарегистрироваться».


13:00
Войдите в аккаунт
Введите логин
Введите пароль
Войти
Зарегистрироваться


13:00
Создание аккаунта
Введите логин
Введите email
Введите пароль
Регистрация
Авторизоваться

АВТОРИЗАЦИЯ

Работаем в файле DbHelper.kt с классом DbHelper

1. В этом классе создаем новую функцию getUser.

Функция будет принимать несколько параметров:

– логин login, тип данных String;

– пароль pass, тип данных String.

В этой функции необходимо создавать объект db, указываем readableDatabase т.к. необходимо будет получать какие-то значения из базы данных, прочитать.

2. Создаем новую переменную result , в которую прописываем то что мы обращаемся к db, метод rawQuery().

В метод rawQuery() передаем SQL-команду за счет которой мы получим записи из базы данных.

Суть команды – Выбираем все поля из таблицы users и указываем условие: находим только те записи, у которых логин будет равен тому значению, которое передается сюда внутрь функции getUser (login), и пароль тоже равен тому что переадно сюда внутрь функции getUser.

Второй параметр null, т.к. никакие динамические данные мы не передаем, а всё предаем сразу в SQL-команду.

```
val result = db.rawQuery("SELECT * FROM user WHERE login = '$login' AND pass = '$pass'", null)
```

Возвращаем результат, но не все записи, для этого указываем moveToFirst чтобы взять первую совпавшую запись.

В итогу получается функция:

```
fun getUser(login: String, pass: String): Boolean{
    val db = this.readableDatabase

    val result = db.rawQuery("SELECT * FROM user WHERE login = '$login' AND pass = '$pass'", null)
    return result.moveToFirst()
}
```

Расшифруем: в функцию передаем логин login и пароль pass. По SQL-команде в таблице user ищем запись у которой логин и пароль будут совпадать с теми данными, которые сюда передаются – login и pass.

И дальше функция будет возвращать либо True либо False в зависимости от того найдена или нет эта запись. moveToFirst позволит взять первую совпавшую запись и если такая запись есть, то будет возвращено значение True, если такой записи не существует, то False. Т.е. сможем понять есть ли пользователь в базе данных, если есть, то будем авторизовывать.

В итоге код в DbHelper.kt:

```
package com.example.lb1

import android.content.ContentValues
import android.content.Context
import android.database.sqlite.SQLiteDatabase
```

```
import android.database.sqlite.SQLiteOpenHelper

class DbHelper(val context: Context, val factory:
SQLiteDatabase.CursorFactory?) :
    SQLiteOpenHelper(context, "app", factory, 1){
    override fun onCreate(db: SQLiteDatabase?) {
        val query = "CREATE TABLE user (id INTEGER PRIMARY KEY, login
TEXT, email TEXT, pass TEXT)"
        db!!.execSQL(query)
    }

    override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int,
newVersion: Int) {
        db!!.execSQL("DROP TABLE IF EXISTS users")
        onCreate(db)
    }

    fun addUser(user: User){
        val values = ContentValues()
        values.put("login", user.login)
        values.put("email", user.email)
        values.put("pass", user.pass)

        val db = this.writableDatabase
        db.insert("user", null, values)

        db.close()
    }

    fun getUser(login: String, pass: String): Boolean{
        val db = this.readableDatabase

        val result = db.rawQuery("SELECT * FROM user WHERE login =
'$login' AND pass = '$pass'", null)
        return result.moveToFirst()
    }

}
```

Переходим в файл «AuthActivity.kt»

1. Копируем часть кода из файла по созданию переменных «MainActivity.kt» (нам не требуется только переменная user_email:

```
val userLogin: EditText = findViewById(R.id.user_login)
val userPass: EditText = findViewById(R.id.user_pass)
val button: Button = findViewById(R.id.button_reg)
```

и вставляем в «AuthActivity.kt», добавляя в конце «_auth» (у кнопки «button_auth»):

```
val userLogin: EditText = findViewById(R.id.user_login_auth)
val userPass: EditText = findViewById(R.id.user_pass_auth)
val button: Button = findViewById(R.id.button_auth)
```

2. Копируем из «MainActivity.kt» код который срабатывает при нажатии кнопки:

```
button.setOnClickListener {
    val login = userLogin.text.toString().trim()
    val email = userEmail.text.toString().trim()
    val pass = userPass.text.toString().trim()

    if(login == "" || email == "" || pass == "")
        Toast.makeText(this, "Не все поля заполнены",
Toast.LENGTH_LONG).show()
    else {
        val user = User(login, email, pass)

        val db = DbHelper(this, null)
        db.addUser(user)
        Toast.makeText(this, "Пользователь $login добавлен",
Toast.LENGTH_LONG).show()

        userLogin.text.clear()
        userEmail.text.clear()
        userPass.text.clear()
    }
}
```

и вставляем в файл «AuthActivity.kt».

Удаляем данные с email.

В коде мы получаема данные от пользователя – логин и пароль, записываем их в переменные login и pass.

Далее эти данные проверяем – если не заполнен логин или пароль, то будет выводиться фраза «Не все поля заполнены». Далее создаем объект db на основе класса DbHelper и будем вызывать функцию getUser(). В эту функцию будем передавать логин login и пароль pass, которые получаем от пользователя.

db.getUser(login, pass) запишем в переменную isAuth

```
val isAuth = db.getUser(login, pass)
```

Далее эту переменную isAuth будем проверять:

Если isAuth будет True, то будет создаваться всплывающая подсказка «Пользователь с таким-то логином авторизован»:

```
if(isAuth){
    Toast.makeText(this, "Пользователь $login авторизован",
Toast.LENGTH_LONG).show()

}
```

Если isAuth будет равен False , то будет создаваться всплывающая подсказка «Пользователь с таким-то логином НЕ авторизован»:

```
if(isAuth){
    Toast.makeText(this, "Пользователь $login авторизован",
Toast.LENGTH_LONG).show()

} else
    Toast.makeText(this, "Пользователь $login НЕ авторизован",
Toast.LENGTH_LONG).show()
```

Перемещаем код по очистки полей в случае успешной авторизации:

```
if(isAuth){
    Toast.makeText(this, "Пользователь $login авторизован",
Toast.LENGTH_LONG).show()
    userLogin.text.clear()
    userPass.text.clear()
} else
    Toast.makeText(this, "Пользователь $login НЕ авторизован",
Toast.LENGTH_LONG).show()
```

В ИТОГЕ:

```
package com.example.lb1

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast

class AuthActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_auth)

        val userLogin: EditText = findViewById(R.id.user_login_auth)
        val userPass: EditText = findViewById(R.id.user_pass_auth)
        val button: Button = findViewById(R.id.button_auth)
        val linkToReg: TextView = findViewById(R.id.link_to_reg)
```

```
        linkToReg.setOnClickListener {
            val intent = Intent(this, MainActivity::class.java)
            startActivity(intent)
        }

        button.setOnClickListener {
            val login = userLogin.text.toString().trim()
            val pass = userPass.text.toString().trim()

            if(login == "" || pass == "")
                Toast.makeText(this, "Не все поля заполнены",
Toast.LENGTH_LONG).show()
            else {
                val db = DbHelper(this, null)
                val isAuth = db.getUser(login, pass)

                if(isAuth){
                    Toast.makeText(this, "Пользователь $login
авторизован", Toast.LENGTH_LONG).show()
                    userLogin.text.clear()
                    userPass.text.clear()
                } else
                    Toast.makeText(this, "Пользователь $login НЕ
авторизован", Toast.LENGTH_LONG).show()

            }
        }
    }
}
```

Проверяем