Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическое занятие 22**»

Выполнил: Малюгин Кирилл Алексеевич

Группа: ПР-32

Преподаватель: Мирошниченко Г.В

2023

**Задание:**

1. Научиться работать c API

2. Повторить работу с форматом данных JSON и базой ROOM на языке Kotlin.

3. Индивидуальное задание под вариантом 2. Приложение «Курсы валют» Currate API ( https://currate.ru/ ).

**Входные данные:**

**LoginActivity:**

login – EditText в который пользователь вписывает логин для входа.

pass – EditText в который пользователь вписывает свой пароль для входа.

**MainActivity:**

edit – EditText для добавления новой валюты в ListView (например: USDRUB).

**EditBaseActivity:**

editName – EditText для изменения названия валюты.

editNumb – EditText для изменения стоимости валюты.

**Листинг программы:**

**LoginActivity:**

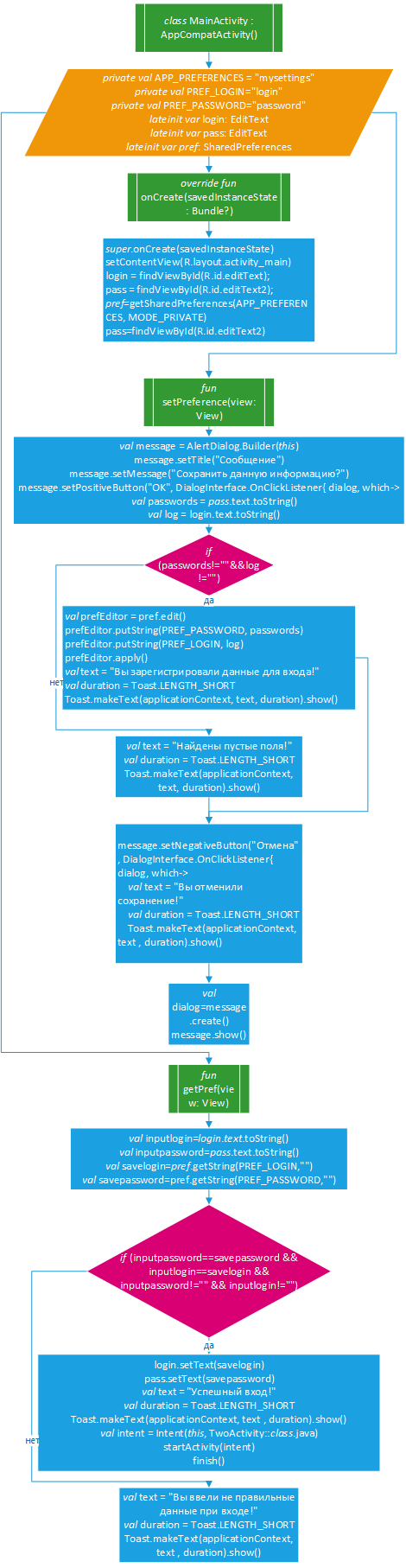
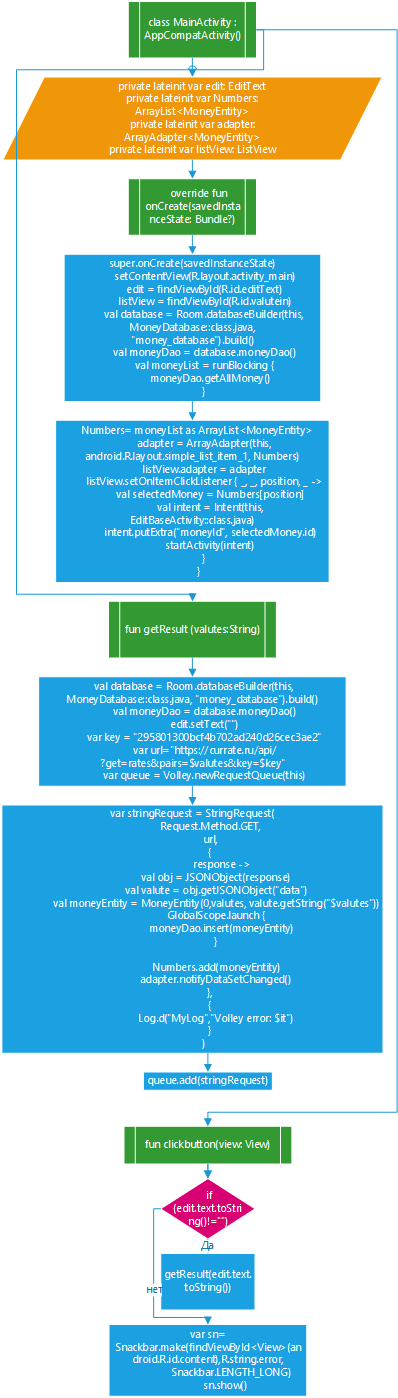
class LoginActivity : AppCompatActivity() {  
 private val APP\_PREFERENCES = "mysettings"  
 private val PREF\_LOGIN="login"  
 private val PREF\_PASSWORD="password"  
 lateinit var login: EditText  
 lateinit var pass: EditText  
 lateinit var pref: SharedPreferences  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_login*)  
 login = findViewById(R.id.*editText*);  
 pass = findViewById(R.id.*editText2*);  
 pref=getSharedPreferences(APP\_PREFERENCES, *MODE\_PRIVATE*)  
 pass=findViewById(R.id.*editText2*)  
 }  
  
 fun setPreference(view: View) {  
 val message = AlertDialog.Builder(this)  
 message.setTitle("Сообщение")  
 message.setMessage("Сохранить данную информацию?")  
 message.setPositiveButton("OK", DialogInterface.OnClickListener**{** dialog, which**->** val passwords = pass.*text*.toString()  
 val log = login.*text*.toString()  
 if (passwords!=""&&log!="") {  
 val prefEditor = pref.edit()  
 prefEditor.putString(PREF\_PASSWORD, passwords)  
 prefEditor.putString(PREF\_LOGIN, log)  
 prefEditor.apply()  
 val text = "Вы зарегистрировали данные для входа!"  
 val duration = Toast.*LENGTH\_SHORT* Toast.makeText(*applicationContext*, text, duration).show()  
 }  
 else {  
 val text = "Найдены пустые поля!"  
 val duration = Toast.*LENGTH\_SHORT* Toast.makeText(*applicationContext*, text, duration).show()  
 }  
 **}**)  
 message.setNegativeButton("Oтмена", DialogInterface.OnClickListener**{** dialog, which**->** val text = "Вы отменили сохранение!"  
 val duration = Toast.*LENGTH\_SHORT* Toast.makeText(*applicationContext*, text, duration).show()  
 **}**)  
 val dialog=message.create()  
 message.show()  
 }  
 fun getPref(view: View) {  
 val inputlogin=login.*text*.toString()  
 val inputpassword=pass.*text*.toString()  
 val savelogin=pref.getString(PREF\_LOGIN,"")  
 val savepassword=pref.getString(PREF\_PASSWORD,"")  
 if (inputpassword==savepassword && inputlogin==savelogin && inputpassword!="" && inputlogin!="") {  
 login.setText(savelogin)  
 pass.setText(savepassword)  
 val text = "Успешный вход!"  
 val duration = Toast.*LENGTH\_SHORT* Toast.makeText(*applicationContext*, text, duration).show()  
 val intent = Intent(this, MainActivity::class.*java*)  
 startActivity(intent)  
 finish()  
 }  
 else  
 {  
 val text = "Вы ввели не правильные данные при входе!"  
 val duration = Toast.*LENGTH\_SHORT* Toast.makeText(*applicationContext*, text, duration).show()  
 }  
 }  
}

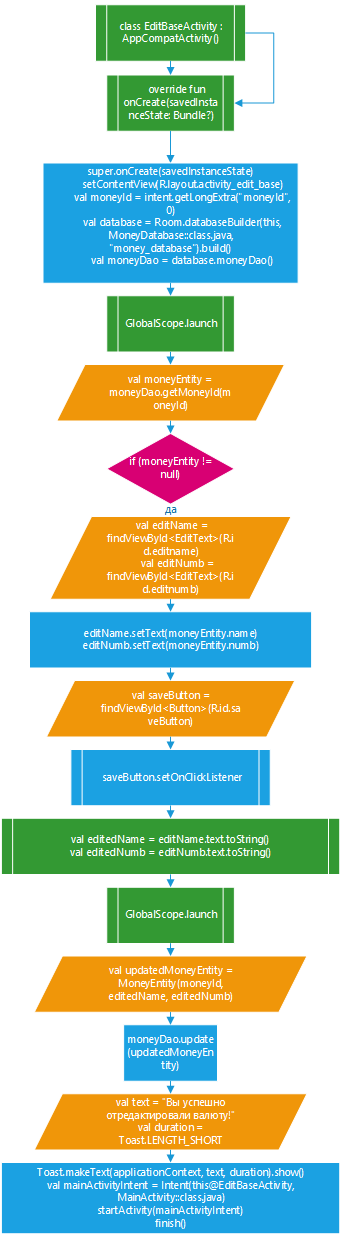
**MainActivity:**

private lateinit var *edit*: EditText  
private lateinit var *Numbers*: ArrayList<MoneyEntity>  
private lateinit var *adapter*: ArrayAdapter<MoneyEntity>  
private lateinit var *listView*: ListView  
class MainActivity : AppCompatActivity() {  
  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 *edit* = findViewById(R.id.*editText*)  
 *listView* = findViewById(R.id.*valutein*)  
 val database = Room.databaseBuilder(this, MoneyDatabase::class.*java*, "money\_database").build()  
 val moneyDao = database.moneyDao()  
 val moneyList = *runBlocking* **{** moneyDao.getAllMoney()  
 **}** *Numbers*= moneyList as ArrayList<MoneyEntity>  
 *adapter* = ArrayAdapter(this, android.R.layout.*simple\_list\_item\_1*, *Numbers*)  
 *listView*.*adapter* = *adapter  
 listView*.setOnItemClickListener **{** \_, \_, position, \_ **->** val selectedMoney = *Numbers*[position]  
 val intent = Intent(this, EditBaseActivity::class.*java*)  
 intent.putExtra("moneyId", selectedMoney.id)  
 startActivity(intent)  
 **}** }  
  
  
 fun getResult (valutes:String)  
 {  
 val database = Room.databaseBuilder(this, MoneyDatabase::class.*java*, "money\_database").build()  
 val moneyDao = database.moneyDao()  
 *edit*.setText("")  
 var key = "295801300bcf4b702ad240d26cec3ae2"  
 var url="https://currate.ru/api/?get=rates&pairs=$valutes&key=$key"  
 var queue = Volley.newRequestQueue(this)  
 var stringRequest = StringRequest(  
 Request.Method.*GET*,  
 url,  
 **{** response **->** val obj = JSONObject(response)  
 val valute = obj.getJSONObject("data")  
 val moneyEntity = MoneyEntity(0,valutes, valute.getString("$valutes"))  
 GlobalScope.*launch* **{** moneyDao.insert(moneyEntity)  
 **}** *Numbers*.add(moneyEntity)  
 *adapter*.notifyDataSetChanged()  
 **}**,  
 **{** Log.d("MyLog","Volley error: $**it**")  
 **}** )  
 queue.add(stringRequest)  
 }  
  
 fun clickbutton(view: View) {  
 if (*edit*.*text*.toString()!="")  
 {  
 getResult(*edit*.*text*.toString())  
 }  
 else  
 {  
 var sn= Snackbar.make(findViewById<View>(android.R.id.*content*),R.string.*error*,  
 Snackbar.*LENGTH\_LONG*)  
 sn.show()  
 }  
 }  
}

**EditBaseActivity:**

class EditBaseActivity : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_edit\_base*)  
 val moneyId = *intent*.getLongExtra("moneyId", 0)  
 val database = Room.databaseBuilder(this, MoneyDatabase::class.*java*, "money\_database").build()  
 val moneyDao = database.moneyDao()  
  
 GlobalScope.*launch* **{** val moneyEntity = moneyDao.getMoneyId(moneyId)  
 if (moneyEntity != null) {  
 val editName = findViewById<EditText>(R.id.*editname*)  
 val editNumb = findViewById<EditText>(R.id.*editnumb*)  
 editName.setText(moneyEntity.name)  
 editNumb.setText(moneyEntity.numb)  
 val saveButton = findViewById<Button>(R.id.*saveButton*)  
 saveButton.setOnClickListener **{** val editedName = editName.*text*.toString()  
 val editedNumb = editNumb.*text*.toString()  
  
 GlobalScope.*launch* **{** val updatedMoneyEntity = MoneyEntity(moneyId, editedName, editedNumb)  
 moneyDao.update(updatedMoneyEntity)  
 **}** val text = "Вы успешно отредактировали валюту!"  
 val duration = Toast.*LENGTH\_SHORT* Toast.makeText(*applicationContext*, text, duration).show()  
 val mainActivityIntent = Intent(this@EditBaseActivity, MainActivity::class.*java*)  
 startActivity(mainActivityIntent)  
 finish()  
 **}** }  
 **}** }  
}

**Блок схемы**

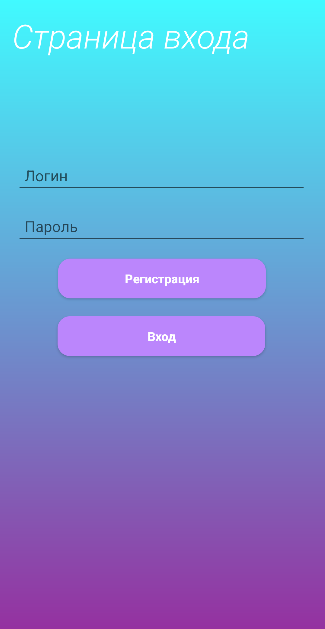
****

**Вывод:**

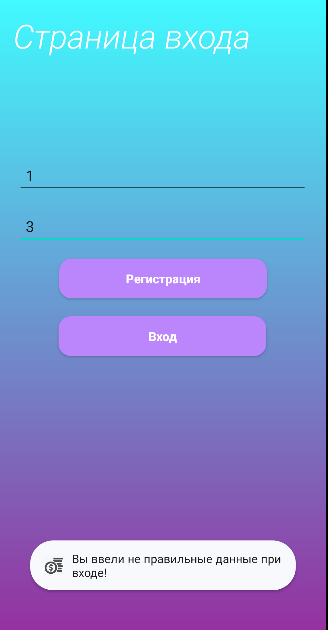
1. **При запуске приложения запускается стартовый экран с иконкой:**

****

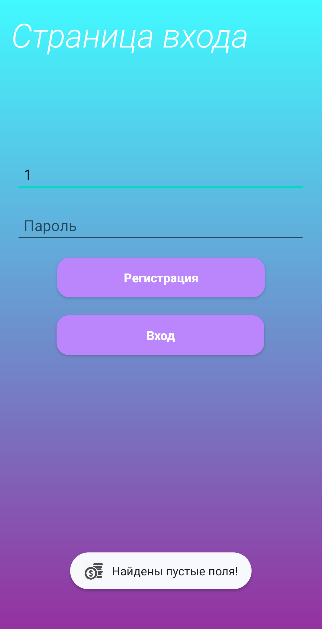
1. **После стартового экрана нас приветствует экран регистрации-входа:**

****

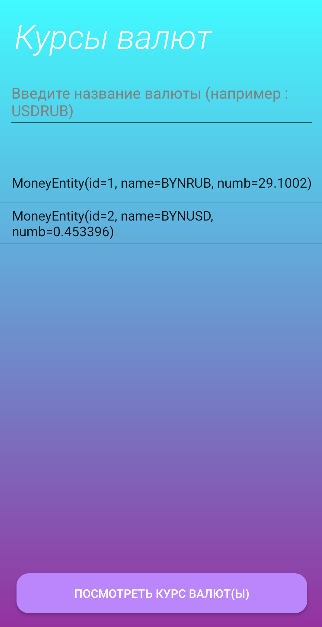
**Если данные будут некорректны или пусты, мы увидим сообщение:**

****

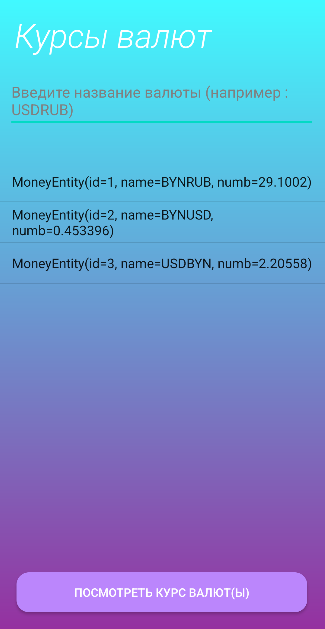
**При нажатии на кнопку регистрации в приложении реализована проверка на корректность логина и пароля (например, приложение не принимает пустые поля и выдает сообщение):**

****

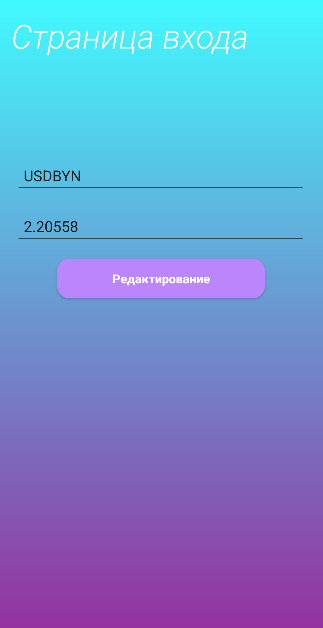
1. **При успешном входе, мы переходим на след. экран, где реализована возможность добавления валют:**

****

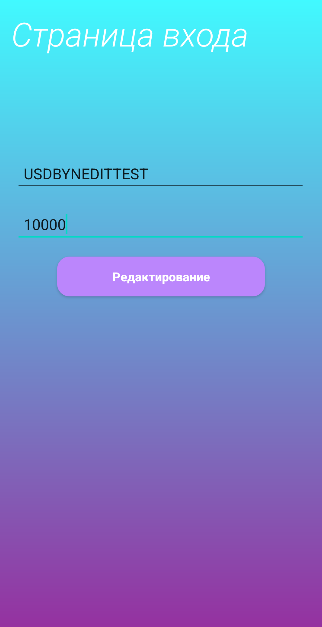
**При добавлении новой валюты, она добавляется в базу данных и выводится в ListView:**

****

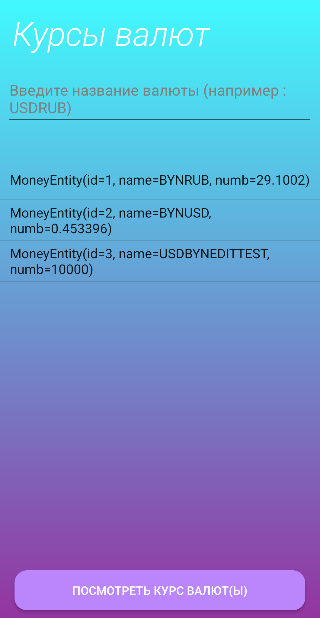
1. **При нажатии на любую из добавленных валют в ListView происходит переход на страницу редактирования этой самой валюты (вы можете отредактировать название валюты и ее стоимость):**

****

**Например, я изменю данную валюту и нажму на кнопку редактирования:**

****

**После нажатия на кнопку редактирования, мы возвращаемся на пред. экран и видим наши изменения в ListView:**

****