



СибГУПИ

СИБИРСКИЙ
ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ
И ИНФОРМАТИКИ



Минцифры
России

Федеральное государственное бюджетное образовательное учреждение
высшего образования

Сибирский государственный университет телекоммуникаций и информатики



Кафедра прикладной математики и кибернетики института информатики и вычислительной техники

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Разработка Android приложения “Угадай картинку”, сгенерированную нейросетью

Выполнил

студент группы ИП-015

Шевельков Павел Сергеевич

Руководитель

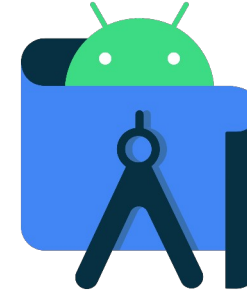
Старший преподаватель Дементьева Кристина
Игоревна

Новосибирск – 2024

ЦЕЛИ И ЗАДАЧИ

- Разработать макет интерфейса пользователя, обеспечивающий интуитивно понятное взаимодействие
- Реализовать основной функционал: угадывание картинок, ведение статистики пользователя
- Провести тестирование приложения и его отладку для обеспечения стабильной работы

- Среда разработки Android Studio



- Язык программирования Kotlin



- База данных Firebase



- Нейросеть Fusion brain



СХЕМА ПРИЛОЖЕНИЯ

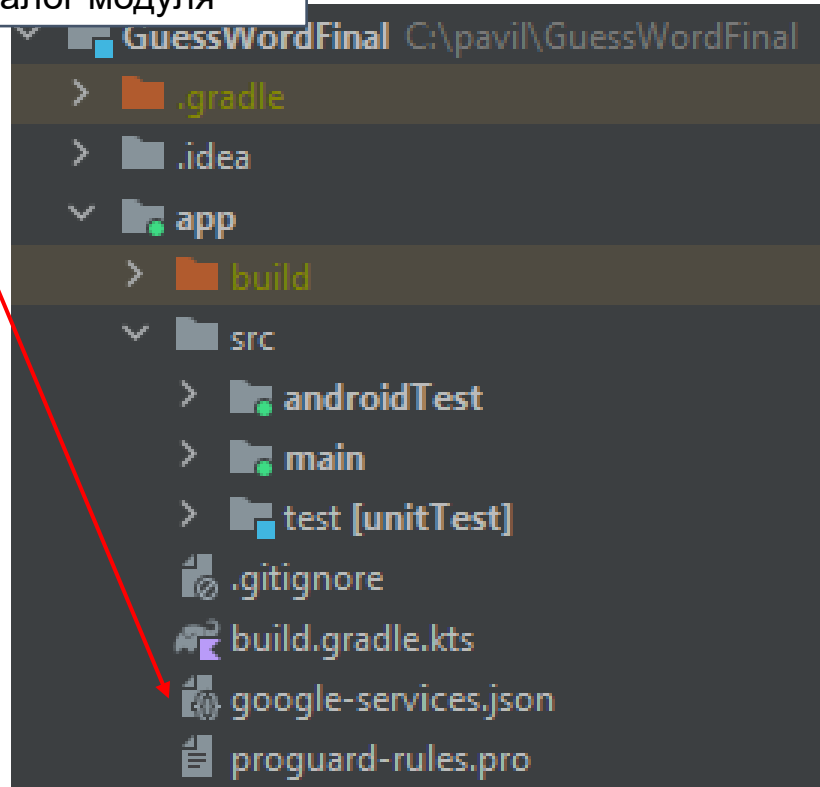
Навигация в приложении



ОПИСАНИЕ РАЗРАБОТКИ

Подключение базы данных к проекту

Добавление файла
google-services.json в
корневой каталог модуля



```
plugins { this: PluginDependenciesSpecScope
    id("com.android.application") version "8.0.2" apply false
    id("com.android.library") version "8.0.2" apply false
    id("org.jetbrains.kotlin.android") version "1.8.20" apply false
    id("com.google.gms.google-services") version "4.3.15" apply false
}

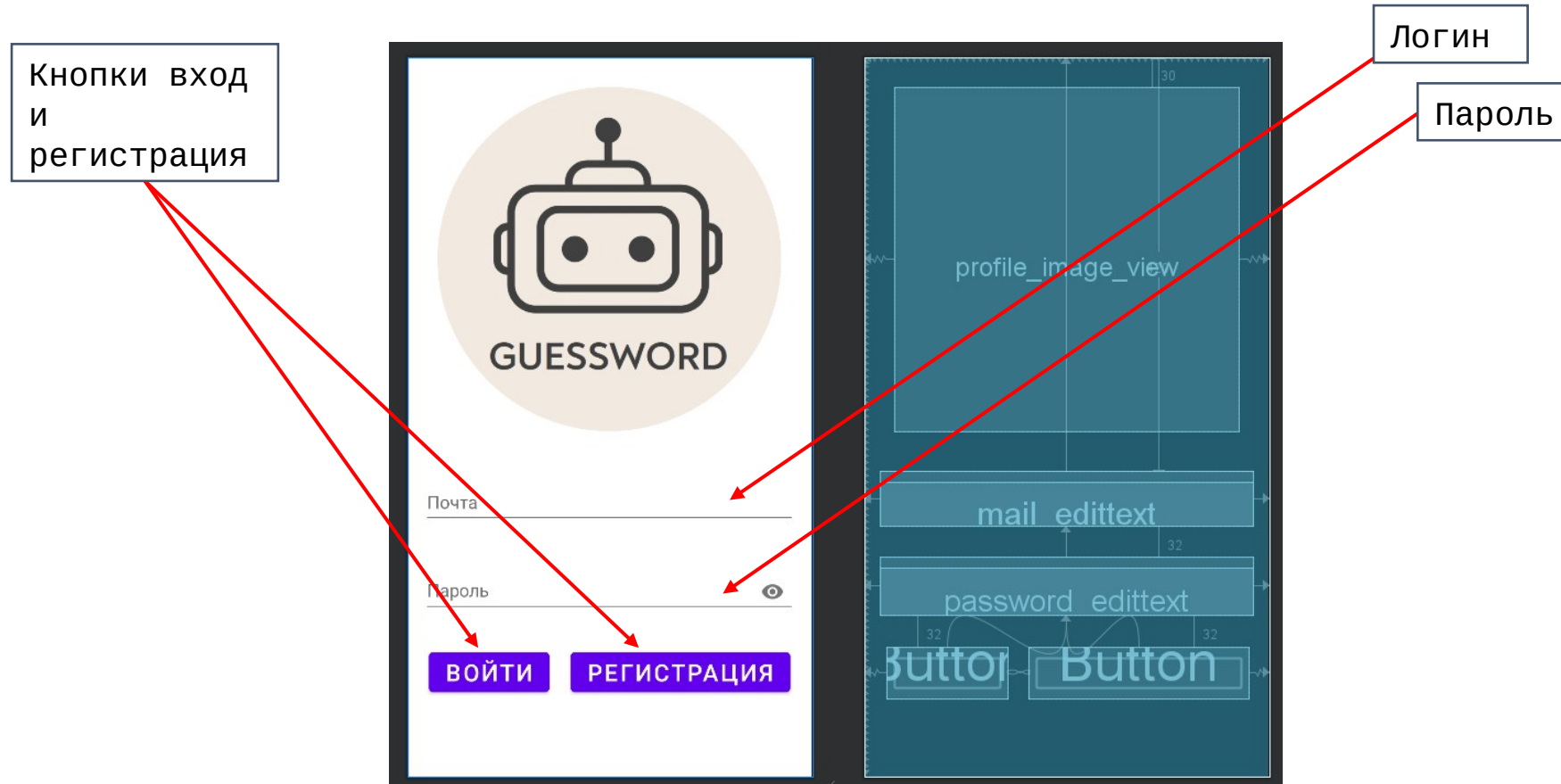
plugins { this: PluginDependenciesSpecScope
    id("com.android.application")
    id("org.jetbrains.kotlin.android")
    id("com.google.gms.google-services")
    id("kotlin-kapt")
    id("androidx.navigation.safeargs.kotlin")
    id("dagger.hilt.android.plugin")
}

// Firebase
implementation(platform("com.google.firebase:firebase-bom:32.2.0"))
implementation("com.google.firebase:firebase-analytics-ktx")
implementation("com.google.firebase:firebase-auth-ktx")
```

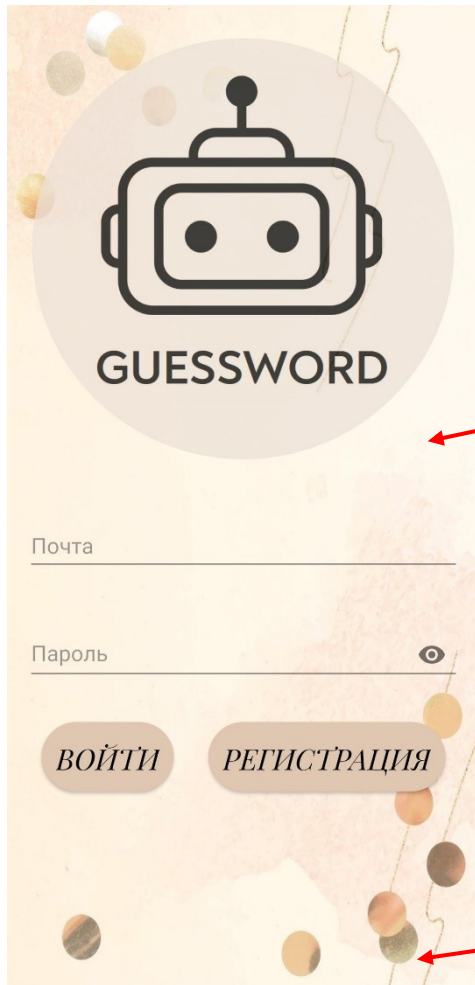
Добавление плагинов и
зависимостей для
Firebase

ОПИСАНИЕ РАЗРАБОТКИ

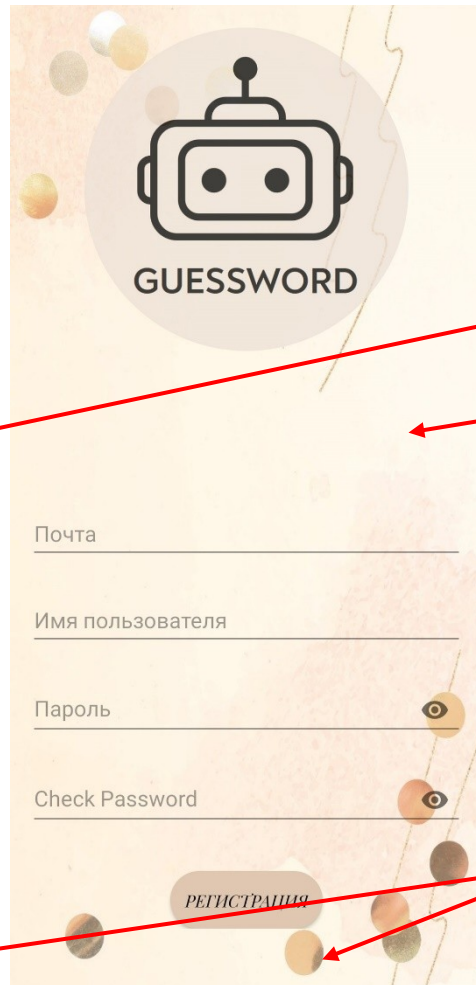
Создание макета в Android Studio



Реализация фрагментов авторизации и регистрации



UI for the login screen. It features a robot icon in a circle with the text "GUESSWORD" below it. Below the icon are two input fields: "Почта" (Email) and "Пароль" (Password) with a toggle eye icon. At the bottom are two buttons: "ВОЙТИ" (Login) and "РЕГИСТРАЦИЯ" (Registration).



UI for the registration screen. It features a robot icon in a circle with the text "GUESSWORD" below it. Below the icon are four input fields: "Почта" (Email), "Имя пользователя" (Username), "Пароль" (Password) with a toggle eye icon, and "Check Password" with a toggle eye icon. At the bottom is a button labeled "РЕГИСТРАЦИЯ" (Registration).

```
binding.signIn.setOnClickListener { it: View!
    val allValidation = inputList.map { it.isValid() }
    if (allValidation.all { it }) {
        viewModel.sendCredentials(
            email = binding.authMail.text(),
            password = binding.authPassword.text(),
            username = ""
        )
    }
}
```

```
binding.startSignUp.setOnClickListener { it: View!
    val allValidation = inputList.map { it.isValid() }
    if (allValidation.all { it }) {
        viewModel.sendCredentials(
            email = binding.signUpEmail.text(),
            username = binding.signUpUserName.text(),
            password = binding.signUpPasswordLayout.text()
        )
    }
}
```

```
viewModel.authState.observe(viewLifecycleOwner) { it: AuthResult!
    when (it) {
        AuthResult.Loading -> binding.progressBarRegistration.visibility = View.VISIBLE
        is AuthResult.Error -> {
            binding.progressBarRegistration.visibility = View.GONE
            Toast.makeText(requireContext(), it.e.message.toString(), Toast.LENGTH_LONG)
                .show()
        }
        is AuthResult.SuccessUser -> {
            findNavController().navigate(R.id.action_registrationFragment_to_homeFragment)
        }
        else -> {
            Log.d(tag: "SignUpFragment", msg: "что-то случилось")
        }
    }
}
```


ОПИСАНИЕ РАЗРАБОТКИ

Реализация фрагмента домашней страницы



```
val goToGame = view?.findViewById<Button>(R.id.startAppBtn)
goToGame?.setOnClickListener { it: View!
    navController.navigate(R.id.action_homeFragment_to_startGameFragment, args: null, navOptions)
}

val goToReference = view?.findViewById<Button>(R.id.referenceBtn)
goToReference?.setOnClickListener { it: View!
    navController.navigate(R.id.action_homeFragment_to_referenceFragment, args: null, navOptions)
}

val goToProfile = view?.findViewById<Button>(R.id.profileBtn)
goToProfile?.setOnClickListener { it: View!
    navController.navigate(R.id.action_homeFragment_to_profileFragment, args: null, navOptions)
}
```

ОПИСАНИЕ РАЗРАБОТКИ

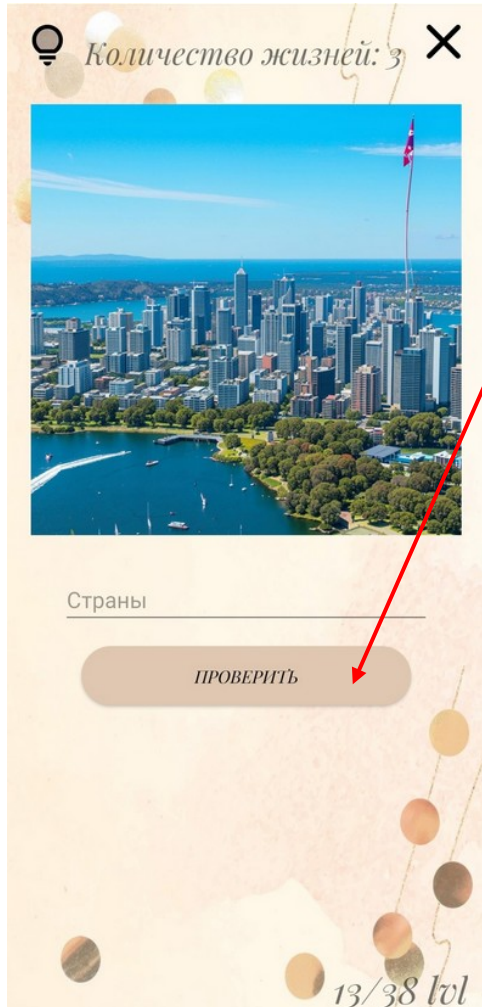
Реализация фрагмента выбора режима игры



```
val countriesButton = view?.findViewById<Button>(R.id.countries_btn)
countriesButton?.setOnClickListener { it: View!
    homeViewModel.countUserCountries.observe(viewLifecycleOwner){count->
        homeViewModel.allCountries.observe(viewLifecycleOwner){all->
            if(startGameViewModel.checkStartLvlInStartFragment(count,all)){
                val navController = findNavController()
                val navOptions = NavOptions.Builder()
                    .setPopUpTo(R.id.countriesFragment, inclusive: true)
                    .build()
                navController.navigate(
                    R.id.action_startGameFragment_to_countriesFragment,
                    args: null,
                    navOptions
                )
            }
            else {
                Toast.makeText(
                    requireContext(),
                    text: "Вы уже прошли все уровни данного раздела!",
                    Toast.LENGTH_SHORT
                ).show()
            }
        }
    }
}
```

ОПИСАНИЕ РАЗРАБОТКИ

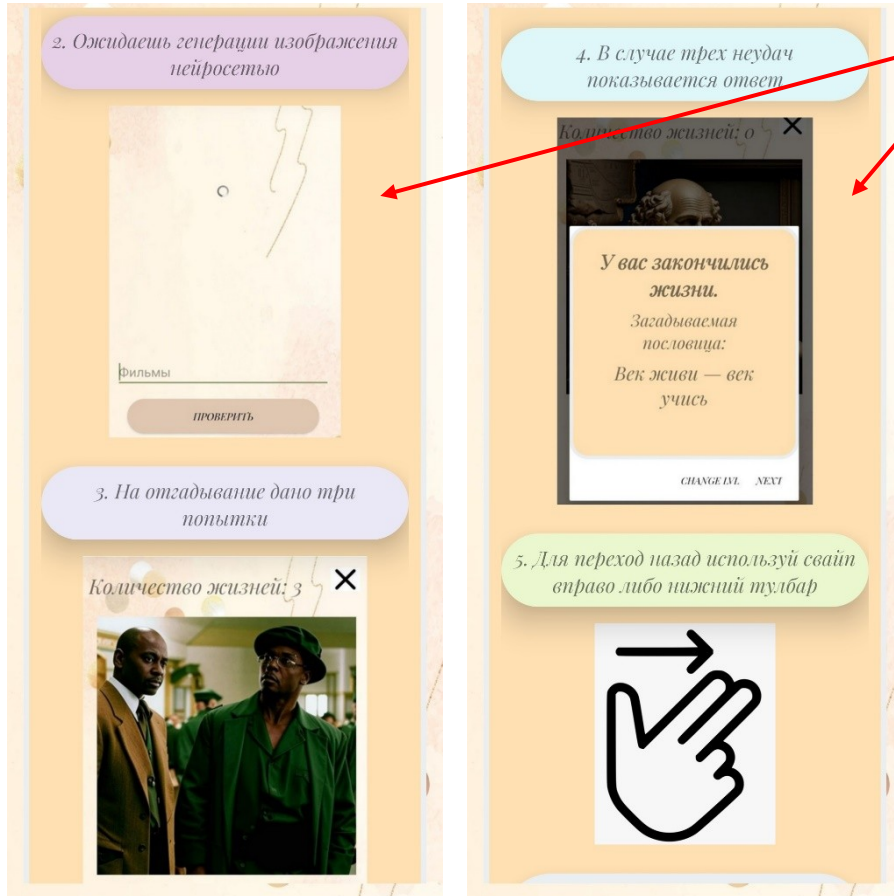
Реализация фрагмента режима игры



```
val checkAnswer = view?.findViewById<Button>(R.id.checkAnswer)
checkAnswer?.setOnClickListener { it: View!
    val inputAnswer = view?.findViewById<EditText>(R.id.inputAnswer)?.text.toString()
    homeViewModel.countUserCountries.observe(viewLifecycleOwner) { countCountries ->
        val correctAnswer = DataStore.countries[countCountries]
        if (inputAnswer.isEmpty()) {...}
        else if ((inputAnswer.trim().equals(correctAnswer.trim(), ignoreCase = true)) or (startGameViewModel.lives.value ?: 1 == 1)) {
            var dialog_title = ""
            homeViewModel.countUserCountries.observe(viewLifecycleOwner){countCountries->
                homeViewModel.allCountries.observe(viewLifecycleOwner){allCountries->
                    val dialogView = LayoutInflater.from(context).inflate(R.layout.custom_dialog, root: null)
                    if ((startGameViewModel.lives.value ?: 1 == 1) and !(inputAnswer.trim().equals(correctAnswer.trim(), ignoreCase = true))) {...} else {
                        homeViewModel.updateUserDataForWin( nameGame: "countries")
                        Toast.makeText(requireContext(), "Правильный ответ!", Toast.LENGTH_SHORT).show()
                        dialog_title = "Поздравляем! Вы угадали!"
                    }
                    if(startGameViewModel.checkNextLvl(countCountries,allCountries)){...}
                    else {...}
                    val textViewDialogWord = dialogView.findViewById<TextView>(R.id.dialog_word)
                    textViewDialogWord.text = correctAnswer
                    val textViewDialogMessege = dialogView.findViewById<TextView>(R.id.dialog_messege)
                    textViewDialogMessege.text = "Загадываемая страна:"
                    val textViewTitle = dialogView.findViewById<TextView>(R.id.dialog_title)
                    textViewTitle.text = dialog_title
                }
            }
        }
        else {
            Toast.makeText(requireContext(), "Неправильный ответ."+"Попробуйте еще раз!", Toast.LENGTH_SHORT).show()
            startGameViewModel.decrementLives()
        }
    }
}
```


ОПИСАНИЕ РАЗРАБОТКИ

Реализация фрагмента справка



```
override fun onCreateView(  
    inflater: LayoutInflater, container: ViewGroup?,  
    savedInstanceState: Bundle?  
): View? {  
    val view = inflater.inflate(R.layout.home_to_fragment_reference, container, attachToRoot: false)  
    view?.let { it: View  
        val scrollView = it.findViewById<ScrollView>(R.id.scrollView)  
        val swipeTouchListener = object : OnSwipeTouchListener(requireContext(), scrollView) {  
            override fun onSwipeRight() {  
                val navController = findNavController()  
                val navOptions = NavOptions.Builder()  
                    .setPopUpTo(R.id.homeFragment, inclusive: true)  
                    .build()  
                navController.navigate(R.id.action_referenceFragment_to_homeFragment, args: null, navOptions)  
            }  
        }  
        scrollView.setOnTouchListener(swipeTouchListener)  
    }  
    return view  
}
```



ОПИСАНИЕ РАЗРАБОТКИ

Реализация фрагмента профиля



```
homeViewModel.userName.observe(viewLifecycleOwner){userName->
    binding.usernameTv.text = userName
}
homeViewModel.allResult.observe(viewLifecycleOwner){allResult->
    binding.allResultUser.text = allResult
}
homeViewModel.maxSeries.observe(viewLifecycleOwner){maxSeries->
    binding.bestSeriesUser.text = maxSeries
}
homeViewModel.currentSeries.observe(viewLifecycleOwner){currentSeries->
    binding.currentSeriesUser.text = currentSeries
}
homeViewModel.countUserCountries.observe(viewLifecycleOwner){countUserCountries->
    binding.countriesUser.text = countUserCountries.toString()
}
homeViewModel.countUserMovies.observe(viewLifecycleOwner){countUserMovies->
    binding.moviesUser.text = countUserMovies.toString()
}
homeViewModel.countUserProverbs.observe(viewLifecycleOwner){countUserProverbs->
    binding.proverbsUser.text = countUserProverbs.toString()
}
```

Отображение
имени и
статистики
пользователя

```
val logoutButton = view?.findViewById<ImageButton>(R.id.logout_btn)
logoutButton?.setOnClickListener { it: View!
    homeViewModel.logout()
    val navController = findNavController()
    val navOptions = NavOptions.Builder()
        .setPopUpTo(R.id.homeFragment, inclusive: true)
        .build()
    navController.navigate(R.id.action_profileFragment_to_authorizationFragment, args: null, navOptions)
}
```

ЗАКЛЮЧЕНИЕ

- ✓ В ходе выполнения данного проекта было реализовано мобильное приложение.
- ✓ Проект был успешно разработан и протестирован, что подтвердило его работоспособность.



СибГУТИ

СИБИРСКИЙ
ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ
И ИНФОРМАТИКИ



Минцифры
России

Спасибо за внимание