Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Магнитогорский государственный технический университет им. Г. И. Носова»

Многопрофильный колледж

**Отчет**

**по учебной практике**

**по специальности 09.02.07 Информационные системы и программирование**

**(базовой подготовки)**

**ПМ. 01. Разработка модулей программного обеспечения для компьютерных систем**

|  |  |
| --- | --- |
|  | Студента гр. ИСпПК-22-7  Гусева Дениса Евгеньевича |
|  | Руководитель практики от МпК:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.Д. Рассадникова  (дата, подпись) |

Магнитогорск, 2023

**ВНУТРЕННЯЯ ОПИСЬ**

**документов, находящихся в отчете**

Обучающегося гр. ИСпПК-22-7 Д.Е. Гусева

|  |  |  |
| --- | --- | --- |
| **№п/п** | **Наименование документа** | **Стр.** |
|  | Задание на практику | 3 |
|  | Аттестационный лист | 5 |
|  | Отчет о выполнении заданий по практике | 6 |

**ЗАДАНИЕ**

**на учебную практику**

**Обучающегося** гр. ИСпПК-22-7 Гусева Дениса Евгеньевича

**специальности** 09.02.07 Информационные системы и программирование

**по профессиональному модулю** ПМ. 01. Разработка модулей программного обеспечения для компьютерных систем, Учебная практика

**Цели практики:**

1. Получение практического опыта:

1.1 Разрабатывать алгоритм решения поставленной задачи и реализовывать его средствами автоматизированного проектирования.

1.2 Разрабатывать код программного продукта на основе готовой спецификации на уровне модуля.

1.3 Использовать инструментальные средства на этапе отладки программного продукта.

1.4.1 Проводить тестирование программного модуля по определенному сценарию.

1.4.3 Распознавать сложные проблемы в знакомых ситуациях.

1.5.1 Анализировать алгоритмы, в том числе с применением инструментальных средств.

1.5.2 Осуществлять рефакторинг и оптимизацию программного кода.

2. Формирование профессиональных компетенций (ПК)

|  |  |
| --- | --- |
| Код и наименование формируемых компетенций | Виды и объем производственных работ, выполняемых в период практики в рамках формируемых компетенций |
| ПК1.2. Разрабатывать программные модули в соответствии с техническим заданием  ПК1.3. Выполнять отладку программных модулей с использованием специализированных программных средств  ПК1.4. Выполнять тестирование программных модулей  ПК1.5. Осуществлять рефакторинг и оптимизацию программного кода | 1. 1. Выбор метода разработки модуля.   Программирование модуля   1. 2. Логическая проверка модуля. Компиляция модуля 2. 3. Отладка и тестирование модулей   Отладка модуля с целью выявления логических ошибок  4. Верификация и аттестация модуля  5. Выбор критерия завершенности тестирования  6. Апробация работы модуля  7. Разработка простейшего приложения  8. Создание пользовательского интерфейса  9. Разработка спецификаций отдельных компонент  10. Разработка системы помощи для работы с программой  11. Компилирование исходных текстов программ. Компоновка программы их компонентов  12. Отладка и модификация программ |

3. Формирование общих компетенций (ОК)

ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам

ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности

ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие

ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 6. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей.

ОК 7. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.

ОК 8. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.

ОК 9. Использовать информационные технологии в профессиональной деятельности.

ОК10. Пользоваться профессиональной документацией на государственном и иностранном языках.

ОК11. Планировать предпринимательскую деятельность в профессиональной сфере.

Место практики: ФГБОУ ВО «МГТУ им. Г.И. Носова», Многопрофильный колледж, полигон учебных практик*.*

**Задание на практику**

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Содержание работ на практике** | **Примерные сроки выполнения (час)** |
|  | Создание архитектуры программы | 6 |
|  | Вёрстка экранов мобильного приложения | 8 |
|  | Реализация клиент-серверного взаимодействия приложения | 8 |
|  | Обеспечение безопасности хранения данных | 6 |
|  | Взаимодействие с аппаратными расширениями устройства | 8 |

Примерный перечень документов, прилагаемых в качестве приложения к отчету по практике

1. Аттестационный лист по практике;
2. Характеристика
3. Дневник по практике

Руководитель практики от МпК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.Д. Рассадникова (*подпись)*

13 мая 2024 г.

**АТТЕСТАЦИОННЫЙ ЛИСТ ПО учебной ПРАКТИКЕ**

Гусева Дениса Евгеньевича

Обучающийся на 3 курсе специальности 09.02.07 Информационные системы и программирование успешно прошел практику по профилю специальности по профессиональному модулю: ПМ. 01. Разработка модулей программного обеспечения для компьютерных систем в организации: ФГБОУ ВО «МГТУ им. Г.И. Носова», Многопрофильный колледж, полигон учебных практик*.*

**Виды и качество выполнения работ**

|  |  |  |
| --- | --- | --- |
| **Код формируемых компетенций** | **Виды и объем работ, выполненных обучающимися во время практики в рамках формируемых компетенций** | **Оценка**  **зачтено/**  **не зачтено** |
| ПК1.2. Разрабатывать программные модули в соответствии с техническим заданием | Создание архитектуры программы |  |
| ПК1.3. Выполнять отладку программных модулей с использованием | Вёрстка экранов мобильного приложения |  |
| ПК1.4. Выполнять тестирование программных модулей | Реализация клиент-серверного взаимодействия приложения |  |
| ПК1.5. Осуществлять рефакторинг и оптимизацию программного кода | Обеспечение безопасности хранения данных |  |
|  | Взаимодействие с аппаратными расширениями устройства |  |

Руководитель практики от МпК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.Д. Рассадникова

(*подпись)*

18 мая 2024 г.

М.П.

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ ПО УЧЕБНОЙ ПРАКТИКЕ**

Сначала была создана верстка для экрана «Sign Up» по макету приложения в Figma и переключение на экран «Log In» по нажатию на кнопку с проверкой заполнения всех полей и согласия пользователя при регистрации или по тексту, если аккаунт уже создан. Код представлен в листинге 1, а результат работы на рисунке 1.

Листинг 1. Верстка экрана”Sign UP”

@Composable

fun SignUp(navController: NavHostController) {

val mask = MaskVisualTransformation("+7(###)###-##-##") //маска для номера телефона

var visibility1 by remember { mutableStateOf(false) }

var visibility2 by remember { mutableStateOf(false) }

var license by remember {mutableStateOf(false)}

var name by remember { mutableStateOf("") }

var phone by remember { mutableStateOf("") }

var email by remember { mutableStateOf("") }

var password by remember { mutableStateOf("") }

var checkpass by remember { mutableStateOf("") }

Column(

modifier = Modifier

.fillMaxSize(),

verticalArrangement = Arrangement.Center,

) {

Spacer(Modifier

.size(30.dp))

Column (

Modifier

.weight(0.7f)

.padding(5.dp)

.fillMaxWidth()) {

Text("Create an account",

fontSize = 24.sp)

Text("Complete the sign up process to get started",

color = Color(207,207,207)

)

}

OutlinedTextField(name, {name=it}, Modifier

.weight(0.8f)

.padding(5.dp)

.fillMaxWidth(),

label={

Row() {

Text("Full Name",

color = Color(207,207,207))

}

},

placeholder = {

Text(text = "Ivanov Ivan",

color = Color(207,207,207))

}

)

OutlinedTextField(phone, {if (it.length<11) {phone=it}}, Modifier

.weight(0.8f)

.padding(5.dp)

.fillMaxWidth(),

visualTransformation = mask,

label={

Row() {

Text("Phone Number",

color = Color(207,207,207))

}

},

placeholder = {

Text(text = "+7(999)999-99-99",

color = Color(207,207,207))

}

)

OutlinedTextField(email, {email=it},

Modifier

.weight(0.8f)

.padding(5.dp)

.fillMaxWidth(),

label={

Row() {

Text("Email Adress",

color = Color(207,207,207))

}

},

placeholder = {

Text(text = "\*\*\*\*\*\*\*\*\*@mail.com",

color = Color(207,207,207))

},

)

OutlinedTextField(password, {password=it},

Modifier

.weight(0.8f)

.padding(5.dp)

.fillMaxWidth(),

visualTransformation = if (visibility1) VisualTransformation.None else PasswordVisualTransformation(),

label={

Text("Password",

color = Color(207,207,207))

},

placeholder = {

Text(text = "",

color = Color(207,207,207))

},

trailingIcon = { Image(painterResource(if (visibility1) R.drawable.eye else R.drawable.close\_eye),"",

Modifier

.size(20.dp)

.clickable { visibility1 = !visibility1 }

.align(Alignment.End)) }

)

OutlinedTextField(checkpass, {checkpass=it},

Modifier

.weight(0.8f)

.padding(5.dp)

.fillMaxWidth(),

visualTransformation = if (visibility2) VisualTransformation.None else PasswordVisualTransformation(),

label={

Text("Password",

color = Color(207,207,207))

},

placeholder = {

Text(text = "",

color = Color(207,207,207))

},

trailingIcon = { Image(painterResource(if (visibility2) R.drawable.eye else R.drawable.close\_eye),"",

Modifier

.size(20.dp)

.clickable { visibility2 = !visibility2 }

.align(Alignment.End)) }

)

Row(

Modifier

.weight(0.5f)

.fillMaxWidth(),

horizontalArrangement = Arrangement.Center,

verticalAlignment = Alignment.CenterVertically

) {

Checkbox(checked = license==true, onCheckedChange = {license=!license})

Text(text = "By ticking this box, you agree to our Terms and conditions and private policy")

}

Button(onClick = { if (RegistrationInputCheck(name, phone, email, password, checkpass, license)) navController.navigate("LogIn") },

Modifier

.weight(1f)

.padding(20.dp)

.fillMaxWidth()

.align(alignment = Alignment.CenterHorizontally),

shape = RoundedCornerShape(5.dp),

colors = ButtonDefaults.buttonColors(if (RegistrationInputCheck(name, phone, email, password, checkpass, license)) Color(5,96,250)

else Color(207, 207, 207))

){ Text(text = "Sign Up") }

Column (

Modifier

.weight(1.2f)

.fillMaxWidth(),

horizontalAlignment = Alignment.CenterHorizontally) {

Row () {

Text(

text = "Already have an account?",

color = Color(207, 207, 207))

Spacer(Modifier

.size(10.dp))

Text(

text = "Sign in",Modifier

.clickable{navController.navigate("LogIn")},

color = Color(5,96,250))

}

Text(

text = "or sign in using",

modifier = Modifier

.padding(10.dp),

color = Color(207, 207, 207))

Image(painterResource(R.drawable.google),"",Modifier

.size(30.dp))

}

Spacer(Modifier

.weight(0.1f))

}

}

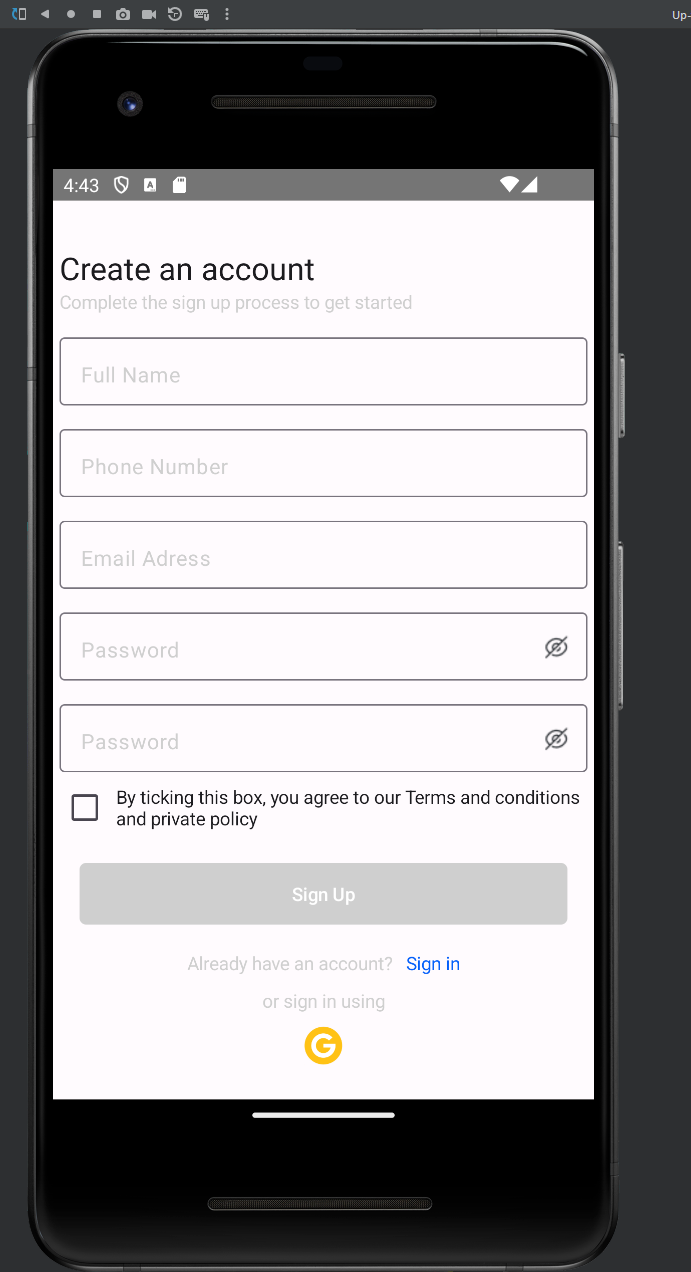


Рисунок 1 – Экран «Sign Up»

Далее была создана верстка для экрана «Log In» (листинг 2). Работа кода изображена на рисунке 2.

Листинг 2. Код для Log In.

@Composable

fun LogIn(navController: NavHostController) {

var visibility1 by remember { mutableStateOf(false) }

var remember by remember {mutableStateOf(false)}

var email by remember { mutableStateOf("") }

var password by remember { mutableStateOf("") }

Column(

modifier = Modifier

.fillMaxSize(),

verticalArrangement = Arrangement.Center,

) {

Spacer(Modifier.weight(0.8f))

Column (

Modifier

.weight(0.9f)

.padding(5.dp)

.fillMaxWidth()) {

Text("Welcome Back",

fontSize = 24.sp)

Text("Fill in your email and password to continue",

color = Color(207,207,207))

}

OutlinedTextField(email, {email=it},

Modifier

.weight(0.8f)

.padding(5.dp)

.fillMaxWidth(),

label={

Row() {

Text("Email Adress",

color = Color(207,207,207)) }

},

placeholder = {

Text(text = "\*\*\*\*\*\*\*\*\*@mail.com",

color = Color(207,207,207))

},

)

OutlinedTextField(password, {password=it},

Modifier

.weight(0.8f)

.padding(5.dp)

.fillMaxWidth(),

visualTransformation = if (visibility1) VisualTransformation.None else PasswordVisualTransformation(),

label={

Text("Password",

color = Color(207,207,207)) },

placeholder = {

Text(text = "",

color = Color(207,207,207)) },

trailingIcon = { Image(painterResource(if (visibility1) R.drawable.eye else R.drawable.close\_eye),"",

Modifier

.size(20.dp)

.clickable { visibility1 = !visibility1 }

.align(Alignment.End)) })

Row(

Modifier

.padding(5.dp)

.weight(0.5f)

.fillMaxWidth(),

horizontalArrangement = Arrangement.SpaceBetween,

verticalAlignment = Alignment.CenterVertically

) {

Row (verticalAlignment = Alignment.CenterVertically) {

Checkbox(checked = remember==true, onCheckedChange = {remember=!remember})

Text(text = "Remember password")

}

Row () {

Text(text = "Forgot Password?",Modifier

.clickable{navController.navigate("ForgotPassword")},

color = Color(5,96,250))

}

}

Spacer(Modifier.weight(1f))

Button(onClick = { navController.navigate("Home") }, Modifier

.weight(1f)

.padding(20.dp)

.fillMaxWidth()

.align(alignment = Alignment.CenterHorizontally),

shape = RoundedCornerShape(5.dp),

colors = ButtonDefaults.buttonColors(if (LoginInputCheck(email, password)) Color(5,96,250)

else Color(207, 207, 207))

){ Text(text = "Log In")}

Column (

Modifier

.weight(1.2f)

.fillMaxWidth(),

horizontalAlignment = Alignment.CenterHorizontally) {

Row () {

Text(

text = "Don't have an account?",

color = Color(207, 207, 207))

Spacer(Modifier

.size(10.dp))

Text(

text = "Sign Up",Modifier

.clickable{navController.navigate("SignUp")},

color = Color(5,96,250))

}

Text(

text = "or log in using",

modifier = Modifier

.padding(10.dp),

color = Color(207, 207, 207))

Image(painterResource(R.drawable.google),"",Modifier

.size(30.dp))

}

Spacer(Modifier.weight(0.8f))

}

}

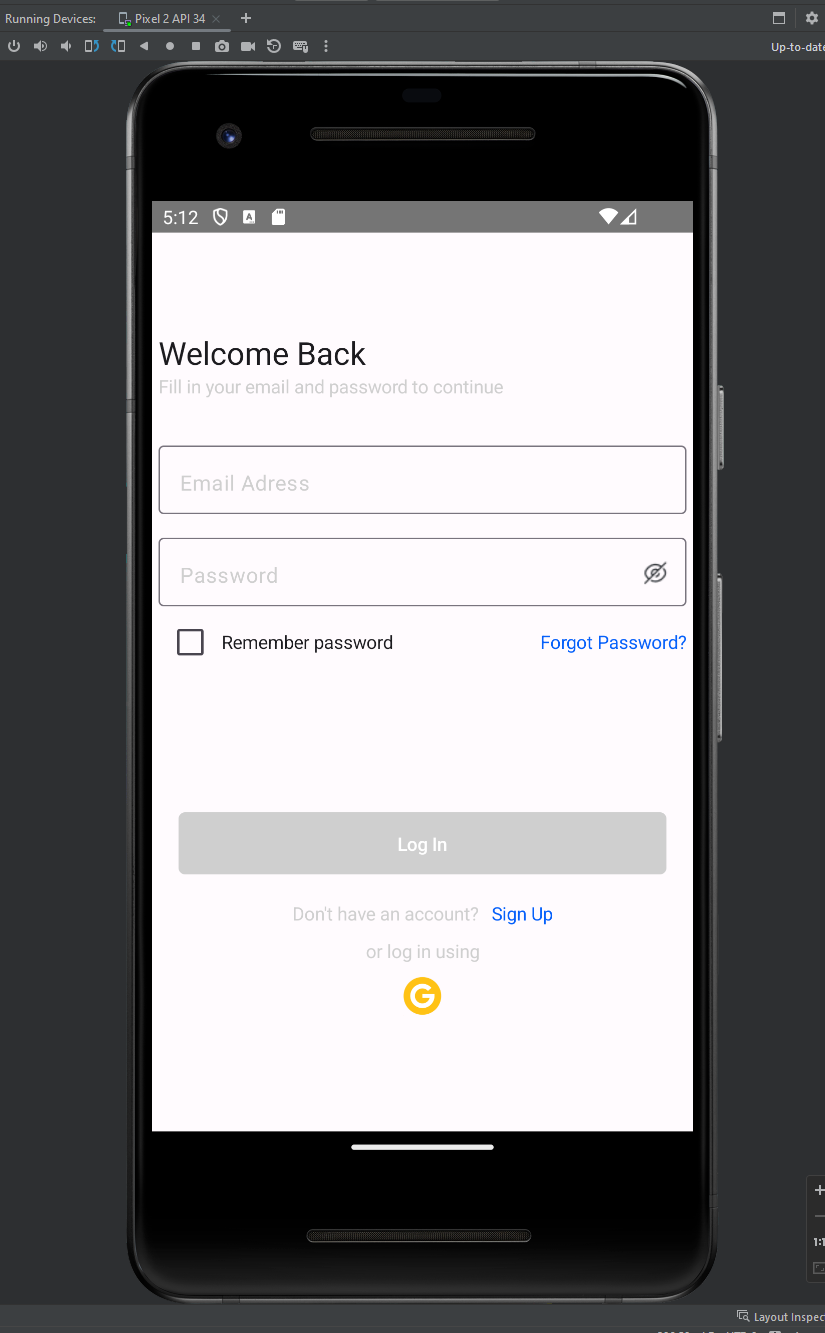


Рисунок 2 – Экран «Log In»

Была создана верстка для экрана «Forgot Password», переход на который осуществляется по нажатию на текст «Forgot Password» (листинг 3 и рисунок 3). Так же присутствует возможность вернуться на предыдущий экран по выделенному тексту.

Листинг 3. Верстка экрана «Forgot Password»

@Composable

fun ForgotPassword(navController: NavHostController) {

var email by remember { mutableStateOf("") }

val regex = Regex(".\*@mail.com")

Column(

modifier = Modifier

.fillMaxSize(),

verticalArrangement = Arrangement.Center,) {

Spacer(Modifier.weight(1f))

Column (

Modifier

.weight(0.9f)

.padding(5.dp)

.fillMaxWidth()) {

Text("Forgot Password",

fontSize = 24.sp)

Text("Enter your email address",

color = Color(207,207,207)) }

OutlinedTextField(email, {email=it},

Modifier

.weight(0.8f)

.padding(5.dp)

.fillMaxWidth(),

label={

Row() {

Text("Email Adress",

color = Color(207,207,207))

}},

placeholder = {

Text(text = "\*\*\*\*\*\*\*\*\*@mail.com",

color = Color(207,207,207))

},)

Spacer(Modifier.weight(1f))

// Кнопка "Авторизации"

Button(onClick = { navController.navigate("OTPVerification") }, Modifier

.weight(1f)

.padding(20.dp)

.fillMaxWidth()

.align(alignment = Alignment.CenterHorizontally),

shape = RoundedCornerShape(5.dp),

colors = ButtonDefaults.buttonColors(if (regex.containsMatchIn(email)) Color(5,96,250)

else Color(207, 207, 207))

){ Text(text = "Send OTP") }

Row (Modifier

.align(Alignment.CenterHorizontally)) {

Text(

text = "Remember password? Back to",

color = Color(207, 207, 207))

Spacer(Modifier

.size(10.dp))

Text(

text = "Sign In",Modifier

.clickable{navController.navigate("LogIn")},

color = Color(5,96,250)) }

Spacer(Modifier.weight(2.5f))

}

}

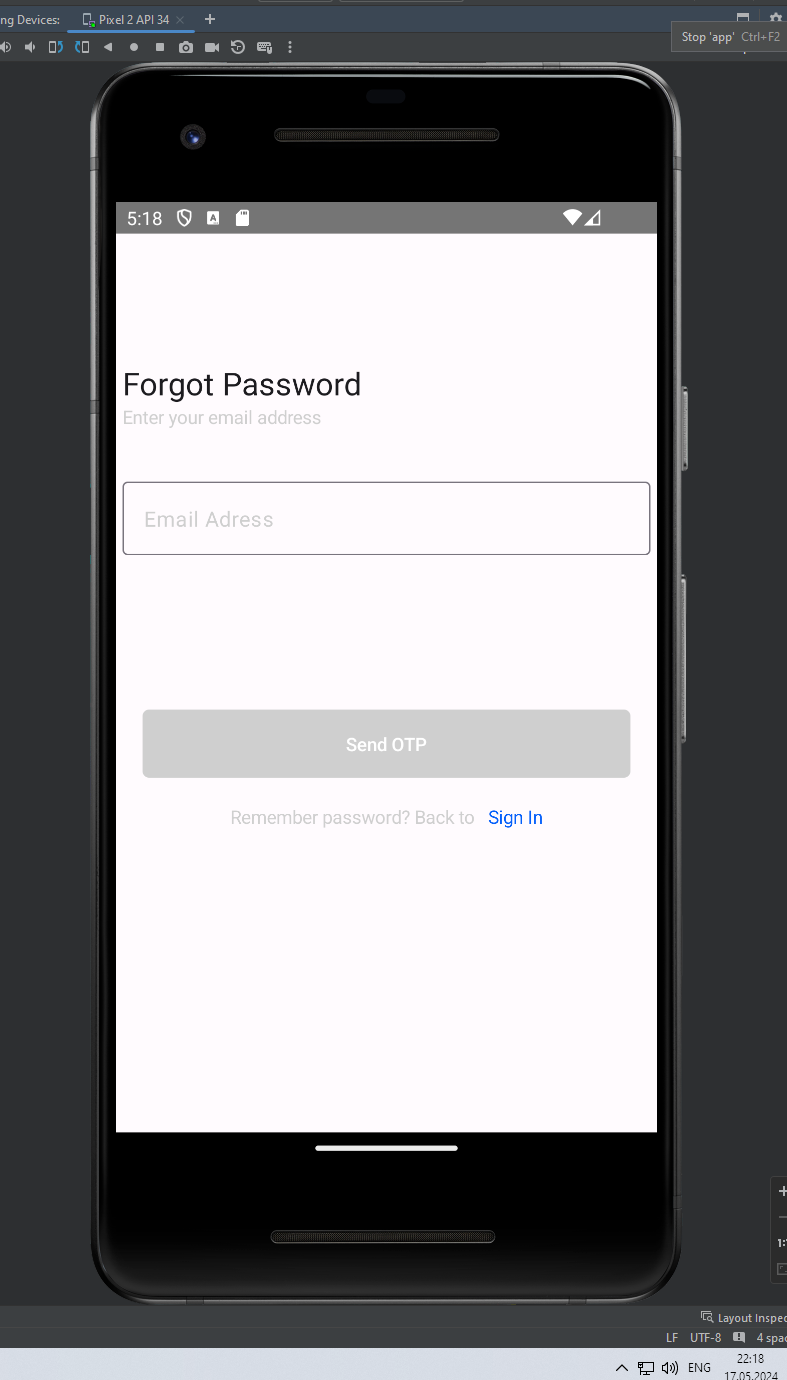


Рисунок 3 – Экран « Forgot Password »

После ввода почты и нажатию на кнопку пользователя перенаправляет на страницу «OTP Verification». Код представлен в листинге 4 и листинг 4.1, а форма страницы на рисунке 4.

Листинг 4. Создание «OTP Verification»

@Composable

fun OTPVerification(navController: NavHostController) {

var numb1 by remember { mutableStateOf("") }

var numb2 by remember { mutableStateOf("") }

var numb3 by remember { mutableStateOf("") }

var numb4 by remember { mutableStateOf("") }

var numb5 by remember { mutableStateOf("") }

var numb6 by remember { mutableStateOf("") }

Column(

modifier = Modifier

.fillMaxSize(),

verticalArrangement = Arrangement.Center,

) {

Spacer(Modifier.weight(0.8f))

Column (

Modifier

.padding(5.dp)

.fillMaxWidth()) {

Text("OTP Verification",

fontSize = 24.sp)

Text("Enter the 6 digit numbers sent to your email",

color = Color(207,207,207))

}

Spacer(Modifier.weight(0.2f))

Row (

Modifier

.padding(5.dp)

.fillMaxWidth(),

horizontalArrangement = Arrangement.SpaceBetween) {

OutlinedTextField(numb1, {if (it.length<2) numb1=it},

Modifier

.focusRequester(focusRequester = FocusRequester())

.size(50.dp)

)

OutlinedTextField(numb2, {if (it.length<2) numb2=it},

Modifier

.size(50.dp)

)

OutlinedTextField(numb3, {if (it.length<2) numb3=it},

Modifier

.size(50.dp)

)

OutlinedTextField(numb4, {if (it.length<2) numb4=it},

Modifier

.size(50.dp)

)

OutlinedTextField(numb5, {if (it.length<2) numb5=it},

Modifier

.size(50.dp)

)

OutlinedTextField(numb6, {if (it.length<2) numb6=it},

Modifier

.size(50.dp)

)

}

Row (

Modifier

.align(Alignment.CenterHorizontally)

.padding(5.dp)) {

var sec = Timer(60) //вывод оставшегося времени до повторной отправки

Text(

text = "If you didn't receive code, ",

color = Color(207, 207, 207))

if (sec!=0) Text(

text = "resend after $sec",

color = Color(207, 207, 207))

else Text(

text = "resend",

Modifier

.clickable{ },

color = Color(5,96,250))

}

Spacer(Modifier.weight(0.2f))

Button(onClick = { navController.navigate("NewPassword") }, Modifier

.weight(0.4f)

.padding(20.dp)

.fillMaxWidth()

.align(alignment = Alignment.CenterHorizontally),

shape = RoundedCornerShape(5.dp),

colors = ButtonDefaults.buttonColors(Color(207, 207, 207))

){ Text(text = "Set New Password") }

Spacer(Modifier.weight(0.8f))

}

}

Листинг 4.1. Функция для обратного отсчёта времени

@Composable

fun Timer(time: Int):Int {

var seconds by remember { mutableStateOf(time) }

LaunchedEffect(seconds){

while (seconds>0){

delay(1000)

seconds--

}

}

return seconds

}

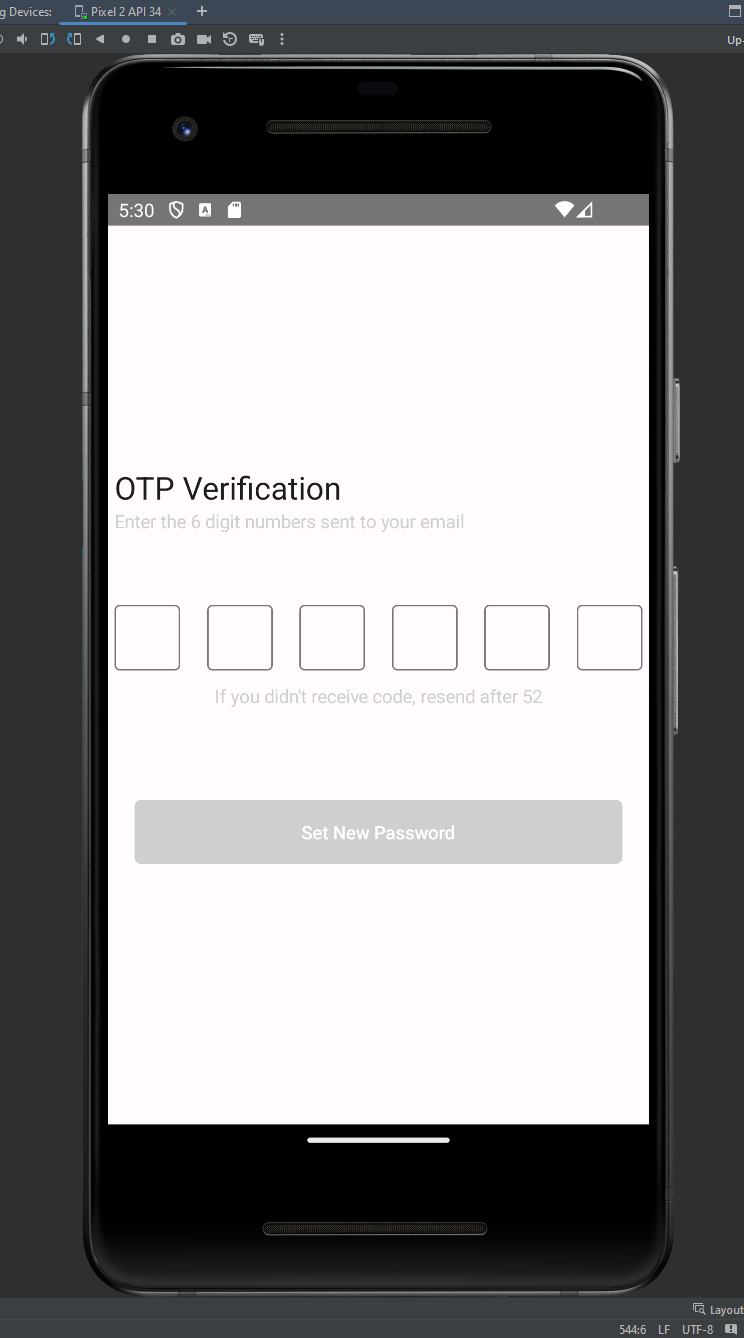


Рисунок 4 – Экран «OTP Verification»

При успешном подтверждении кода можно изменить старый пароль (листинг 5 и рисунок 5).

Листинг 5. Верстка «New Password»

@Composable

fun NewPassword(navController: NavHostController) {

var password by remember { mutableStateOf("") }

var checkpass by remember { mutableStateOf("") }

var visibility1 by remember { mutableStateOf(false) }

var visibility2 by remember { mutableStateOf(false) }

Column(

modifier = Modifier

.fillMaxSize(),

verticalArrangement = Arrangement.Center,

) {

Spacer(Modifier.weight(1f))

Column (

Modifier

.weight(0.7f)

.padding(5.dp)

.fillMaxWidth()) {

Text("New Password",

fontSize = 24.sp)

Text("Enter new password",

color = Color(207,207,207))

}

// Поле ввода пароля

OutlinedTextField(password, {password=it},

Modifier

.weight(0.6f)

.padding(5.dp)

.fillMaxWidth(),

visualTransformation = if (visibility1) VisualTransformation.None else PasswordVisualTransformation(),

label={

Text("Password",

color = Color(207,207,207))

},

placeholder = {

Text(text = "",

color = Color(207,207,207))

},

trailingIcon = { Image(painterResource(if (visibility1) R.drawable.eye else R.drawable.close\_eye),"",

Modifier

.size(20.dp)

.clickable { visibility1 = !visibility1 }

.align(Alignment.End)) }

)

// Поле ввода пароля повторно

OutlinedTextField(checkpass, {checkpass=it},

Modifier

.weight(0.6f)

.padding(5.dp)

.fillMaxWidth(),

visualTransformation = if (visibility2) VisualTransformation.None else PasswordVisualTransformation(),

label={

Text("Confirm Password",

color = Color(207,207,207))

},

placeholder = {

Text(text = "",

color = Color(207,207,207))

},

trailingIcon = { Image(painterResource(if (visibility2) R.drawable.eye else R.drawable.close\_eye),"",

Modifier

.size(20.dp)

.clickable { visibility2 = !visibility2 }

.align(Alignment.End)) }

)

Spacer(Modifier.weight(0.2f))

// Кнопка "Авторизации"

Button(onClick = { }, Modifier

.weight(0.7f)

.padding(20.dp)

.fillMaxWidth()

.align(alignment = Alignment.CenterHorizontally),

colors = ButtonDefaults.buttonColors(if (password==checkpass && password.isNotEmpty() && checkpass.isNotEmpty()) Color(5,96,250)

else Color(207, 207, 207))

) {

Text(text = "Log In")

}

Spacer(Modifier.weight(1f))

}

}

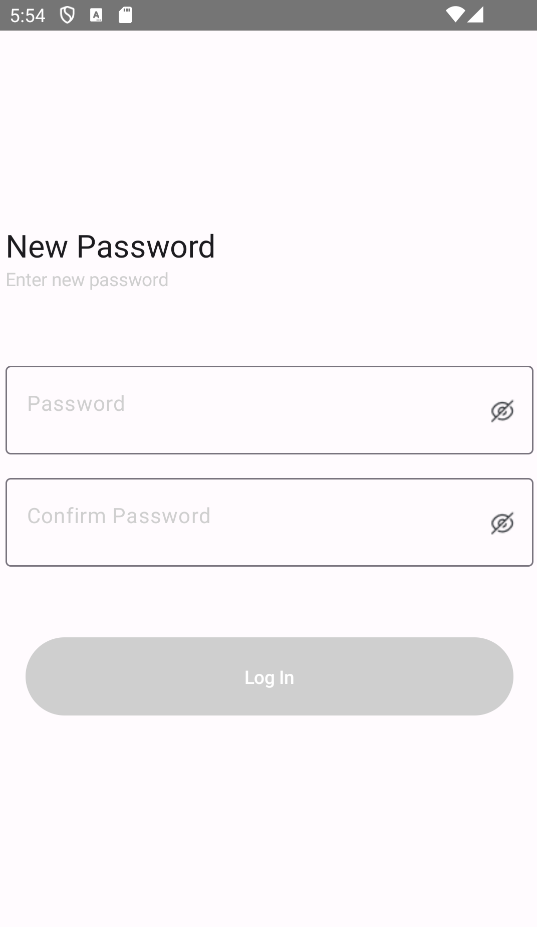


Рисунок 5 – экран «New Password»

Следом был написан код для создания «Profile» по макету (листинг 5 и рисунок 5). Для данной страницы так же были разработаны экраны «AddPayment» (листинг 5.1 и рисунок 5.1) и «Notification» (листинг 5.2 и рисунок 5.2) с переходом по нажатию на поля «Card and Bank account settings» и «Notification Settings» соответственно.

Листинг 5. Создание страницы «Profile»

@Composable

fun Profile(paddingValues: PaddingValues, navController: NavHostController) {

val name by remember { mutableStateOf("Ken") }

val money by remember { mutableStateOf(0f) }

var visibility by remember { mutableStateOf(false) }

var mode by remember { mutableStateOf(false) }

LazyColumn (

Modifier

.padding(paddingValues)

.fillMaxWidth()) {

items(1){

Spacer(Modifier

.size(8.dp))

Row (

Modifier

.padding(5.dp)

.fillMaxWidth(),

horizontalArrangement = Arrangement.Center,

verticalAlignment = Alignment.CenterVertically) {

Column(Modifier

.weight(1f)) {

Image(painterResource(R.drawable.profile\_photo ),"",

Modifier

.size(70.dp))

}

Column(Modifier

.weight(4f)) {

Text(text = "Hello $name",

fontSize = 18.sp)

Row () {

Text(text = "Current balance: ")

Text(text = "${if (visibility) money else "\*\*\*\*"}",

color = Color(5, 96, 250))

}

}

Column(Modifier

.weight(0.5f)) {

Image(painterResource(if (visibility) R.drawable.eye else R.drawable.close\_eye),"",

Modifier

.size(20.dp)

.clickable { visibility = !visibility }

.align(Alignment.End))

}

}

Row(

Modifier

.padding(5.dp)

.fillMaxWidth(),

horizontalArrangement = Arrangement.SpaceBetween,

verticalAlignment = Alignment.CenterVertically) {

Text(text = "Enable dark Mode",

fontSize = 18.sp,

fontWeight = FontWeight.Bold)

Switch(mode, {mode=!mode},

colors = SwitchDefaults.colors(

checkedThumbColor = Color(255,255,255),

checkedTrackColor = Color(5, 96, 250),

checkedBorderColor = Color(5, 96, 250),

uncheckedThumbColor = Color(255,255,255),

uncheckedTrackColor = Color(205,205,205,205),

uncheckedBorderColor = Color(205,205,205,205)

)

)

}

Row (Modifier

.height(70.dp)

.padding(5.dp)

.clickable { }

.border(2.dp, Color(205, 205, 205, 205))

.padding(5.dp)

.fillMaxWidth(),

horizontalArrangement = Arrangement.Center,

verticalAlignment = Alignment.CenterVertically) {

Column(Modifier

.weight(0.7f)) {

Image(painterResource(R.drawable.icon\_edit\_profile),"",

Modifier

.size(20.dp)

.align(Alignment.CenterHorizontally)

)

}

Column(Modifier

.weight(5f)) {

Text(text = "Edit Profile",

fontSize = 18.sp,

fontWeight = FontWeight.Medium )

Text(text = "Name, phone no, address, email ...",

color = Color(205,205,205,205))

}

Column(Modifier

.weight(0.4f)) {

Icon(

Icons.Default.KeyboardArrowRight, "", Modifier

.align(Alignment.Start)

.size(20.dp)

)

}

}

Row (Modifier

.height(70.dp)

.padding(5.dp)

.clickable { }

.border(2.dp, Color(205, 205, 205, 205))

.padding(5.dp)

.fillMaxWidth(),

horizontalArrangement = Arrangement.Center,

verticalAlignment = Alignment.CenterVertically) {

Column(Modifier

.weight(0.7f)) {

Image(painterResource(R.drawable.icon\_state\_reports),"",

Modifier

.size(20.dp)

.align(Alignment.CenterHorizontally)

)

}

Column(Modifier

.weight(5f)) {

Text(text = "Statements & Reports",

fontSize = 18.sp,

fontWeight = FontWeight.Medium )

Text(text = "Download transaction details, orders, deliveries",

color = Color(205,205,205,205))

}

Column(Modifier

.weight(0.4f)) {

Icon(

Icons.Default.KeyboardArrowRight, "", Modifier

.align(Alignment.Start)

.size(20.dp)

)

}

}

Row (Modifier

.height(70.dp)

.padding(5.dp)

.clickable { navController.navigate("Notification") }

.border(2.dp, Color(205, 205, 205, 205))

.padding(5.dp)

.fillMaxWidth(),

horizontalArrangement = Arrangement.Center,

verticalAlignment = Alignment.CenterVertically) {

Column(Modifier

.weight(0.7f)) {

Image(painterResource(R.drawable.icon\_notification),"",

Modifier

.size(20.dp)

.align(Alignment.CenterHorizontally)

)

}

Column(Modifier

.weight(5f)) {

Text(text = "Notification Settings",

fontSize = 18.sp,

fontWeight = FontWeight.Medium )

Text(text = "Mute, unmute, set location & tracking setting",

color = Color(205,205,205,205))

}

Column(Modifier

.weight(0.4f)) {

Icon(

Icons.Default.KeyboardArrowRight, "", Modifier

.align(Alignment.Start)

.size(20.dp)

)

}

}

Row (Modifier

.height(70.dp)

.padding(5.dp)

.clickable { navController.navigate("AddPayment") }

.border(2.dp, Color(205, 205, 205, 205))

.padding(5.dp)

.fillMaxWidth(),

horizontalArrangement = Arrangement.Center,

verticalAlignment = Alignment.CenterVertically) {

Column(Modifier

.weight(0.7f)) {

Image(painterResource(R.drawable.icon\_card\_bank),"",

Modifier

.size(20.dp)

.align(Alignment.CenterHorizontally)

)

}

Column(Modifier

.weight(5f)) {

Text(text = "Card and Bank account settings",

fontSize = 18.sp,

fontWeight = FontWeight.Medium )

Text(text = "Change cards, delete card details",

color = Color(205,205,205,205))

}

Column(Modifier

.weight(0.4f)) {

Icon(

Icons.Default.KeyboardArrowRight, "", Modifier

.align(Alignment.Start)

.size(20.dp)

)

}

}

Row (Modifier

.height(70.dp)

.padding(5.dp)

.clickable { }

.border(2.dp, Color(205, 205, 205, 205))

.padding(5.dp)

.fillMaxWidth(),

horizontalArrangement = Arrangement.Center,

verticalAlignment = Alignment.CenterVertically) {

Column(Modifier

.weight(0.7f)) {

Image(painterResource(R.drawable.icon\_referrals),"",

Modifier

.size(20.dp)

.align(Alignment.CenterHorizontally)

)

}

Column(Modifier

.weight(5f)) {

Text(text = "Referrals",

fontSize = 18.sp,

fontWeight = FontWeight.Medium )

Text(text = "Know more about us, terms and conditions",

color = Color(205,205,205,205))

}

Column(Modifier

.weight(0.4f)) {

Icon(

Icons.Default.KeyboardArrowRight, "", Modifier

.align(Alignment.Start)

.size(20.dp)

)

}

}

Row (Modifier

.height(70.dp)

.padding(5.dp)

.clickable { }

.border(2.dp, Color(205, 205, 205, 205))

.padding(5.dp)

.fillMaxWidth(),

horizontalArrangement = Arrangement.Center,

verticalAlignment = Alignment.CenterVertically) {

Column(Modifier

.weight(0.7f)) {

Image(painterResource(R.drawable.icon\_about),"",

Modifier

.size(20.dp)

.align(Alignment.CenterHorizontally)

)

}

Column(Modifier

.weight(5f)) {

Text(text = "About Us",

fontSize = 18.sp,

fontWeight = FontWeight.Medium )

Text(text = "Name, phone no, address, email ...",

color = Color(205,205,205,205))

}

Column(Modifier

.weight(0.4f)) {

Icon(

Icons.Default.KeyboardArrowRight, "", Modifier

.align(Alignment.Start)

.size(20.dp)

)

}

}

Row (Modifier

.height(70.dp)

.padding(5.dp)

.clickable { }

.border(2.dp, Color(205, 205, 205, 205))

.padding(5.dp)

.fillMaxWidth(),

horizontalArrangement = Arrangement.Center,

verticalAlignment = Alignment.CenterVertically) {

Column(Modifier

.weight(0.7f)) {

Image(painterResource(R.drawable.icon\_log\_out),"",

Modifier

.size(20.dp)

.align(Alignment.CenterHorizontally)

)

}

Column(Modifier

.weight(5f)) {

Text(text = "Log Out",

fontSize = 18.sp,

fontWeight = FontWeight.Medium )

}

Column(Modifier

.weight(0.4f)) {

Icon(

Icons.Default.KeyboardArrowRight, "", Modifier

.align(Alignment.Start)

.size(20.dp))

}

}

}

}

}

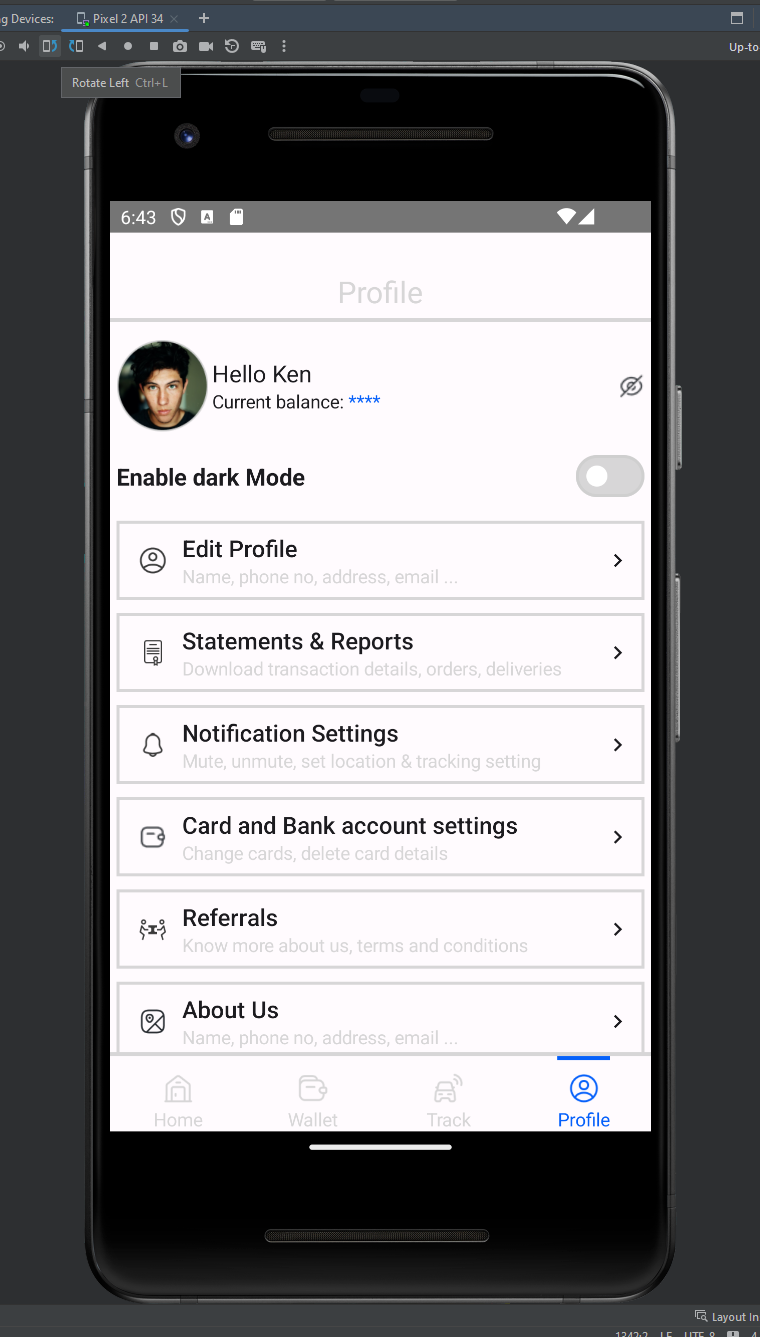


Рисунок 5 – Экран «Profile»

Листинг 5.1. Создание «AddPayment»

@Composable

fun AddPayment(paddingValues: PaddingValues, navController: NavHostController) {

Column (Modifier

.fillMaxWidth(),

horizontalAlignment = Alignment.CenterHorizontally,

verticalArrangement = Arrangement.Center) {

}

}



Рисунок 5.1 – Экран «AddPayment»

Листинг 5.2. Создание «Notification»

@Composable

fun Notification(paddingValues: PaddingValues, navController: NavHostController) {

Column (

Modifier

.padding(paddingValues)

.fillMaxHeight()

.fillMaxWidth(),

horizontalAlignment = Alignment.CenterHorizontally,

verticalArrangement = Arrangement.Center) {

Image(painterResource(R.drawable.icon\_notification\_big),"",

Modifier

.size(100.dp)

.align(Alignment.CenterHorizontally)

)

Text(text = "You have no notifications",

fontSize = 18.sp)

}

}

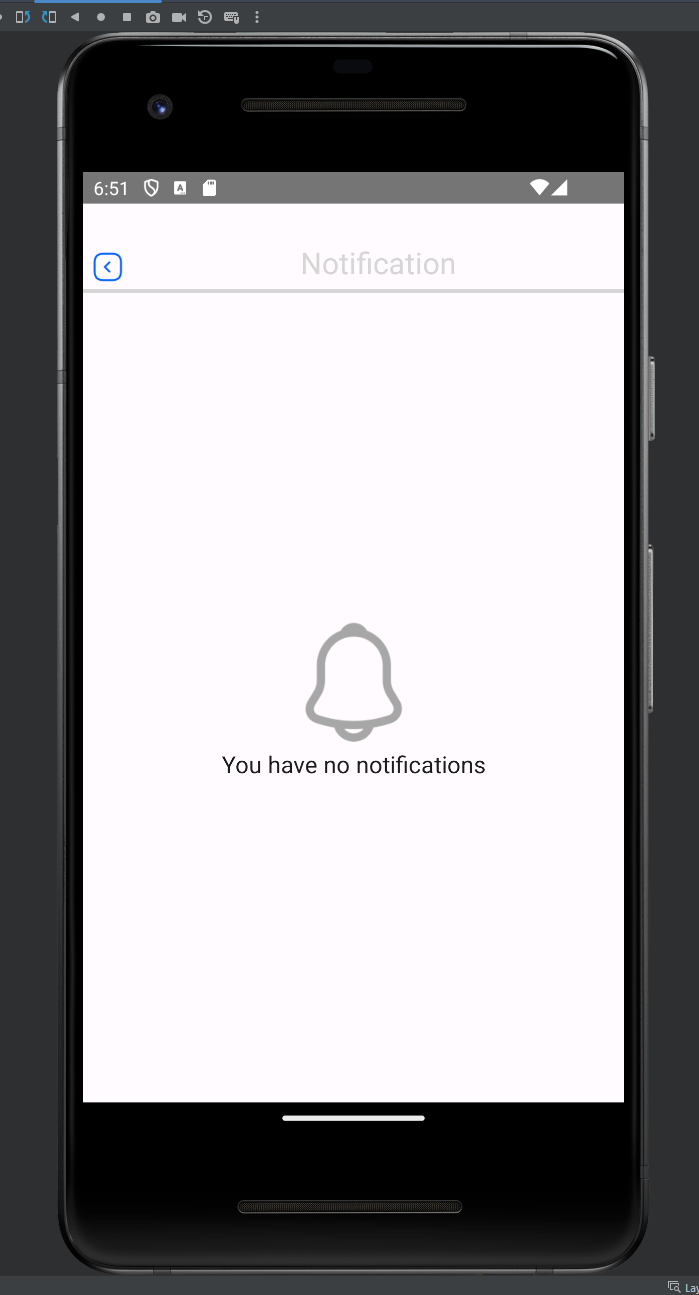


Рисунок 5.2 – Экран «Notification»

Далее была сделана верстка для окна «Home». Код отображен в листинге 6 и результат работы кода на рисунке 6.

Листинг 6. Код для страницы «Home»

@Composable

fun Home(paddingValues: PaddingValues, navController: NavHostController) {

var search by remember { mutableStateOf("") }

LazyColumn (

Modifier

.padding(paddingValues)

.fillMaxWidth()) {

items(1){

Spacer(Modifier

.size(10.dp))

OutlinedTextField(search, {search=it},

Modifier

.padding(5.dp)

.height(50.dp)

.background(Color(207, 207, 207))

.fillMaxWidth(),

placeholder = {

Text(text = "Search services", color = Color(167,167,167))

},

trailingIcon = { Image(painterResource(R.drawable.icon\_search),"",

Modifier

.size(20.dp)

.clickable { }) }

)

Spacer(Modifier.size(10.dp))

Box (

Modifier

.fillMaxWidth()

.padding(5.dp)) {

Image(painterResource(R.drawable.test\_photo),"",

Modifier

.clip(RoundedCornerShape(10.dp))

.height(110.dp)

.background(Color(0, 108, 236))

.fillMaxWidth())

Row (

Modifier

.align(Alignment.Center)

.fillMaxWidth(),

verticalAlignment = Alignment.CenterVertically) {

Image(painterResource(R.drawable.profile\_photo ),"",

Modifier

.weight(1f)

.size(70.dp))

Column (Modifier.weight(3f)) {

Text(text = "Hellow Ken!", fontSize = 24.sp, color = Color(207, 207, 207))

Text(text = "We trust you are having a great time", color = Color(207, 207, 207))

}

Image(painterResource(R.drawable.icon\_notification\_big ),"",

Modifier

.weight(0.8f)

.size(30.dp),

colorFilter = ColorFilter.tint(Color.White)

)

}

}

Spacer(Modifier.size(20.dp))

Row (Modifier

.padding(5.dp)

.fillMaxWidth(),

horizontalArrangement = Arrangement.SpaceBetween) {

Text(text = "Special for you", fontSize = 16.sp, color = Color(236,128,0))

Image(painterResource(R.drawable.icon\_orange\_errow\_left),"",Modifier

.clickable{ }

.size(20.dp))

}

LazyRow(Modifier.padding(5.dp),horizontalArrangement = Arrangement.spacedBy(10.dp)) {

items(1) {

Image(painterResource(R.drawable.first\_special),"")

Image(painterResource(R.drawable.second\_special),"")

}

}

Spacer(Modifier.size(20.dp))

Text(text = "What would you like to do",Modifier.padding(5.dp), fontSize = 16.sp, color = Color(5, 96, 250))

Row (Modifier.padding(5.dp)) {

Box (

Modifier

.clip(RoundedCornerShape(20.dp))

.weight(1f),) {

Text(text = " ", Modifier

.size(200.dp)

.clickable { }

.background(Color(207, 207, 207)))

Column (

Modifier

.padding(10.dp)

.align(Alignment.Center)) {

Image(painterResource(R.drawable.icon\_customer),"",Modifier

.size(40.dp))

Text(text = "Customer Care",

fontSize = 18.sp,

color = Color(5, 96, 250))

Text(text = "Our customer care service line is available from 8-9pm week days and 9-5 weekends - tap to call us today",

fontSize = 12.sp)

}

}

Spacer(Modifier.weight(0.1f))

Box (

Modifier

.clip(RoundedCornerShape(20.dp))

.weight(1f),) {

Text(text = " ", Modifier

.size(200.dp)

.clickable { navController.navigate("SendPackage") }

.background(Color(207, 207, 207)))

Column (

Modifier

.padding(10.dp)

.align(Alignment.Center)) {

Image(painterResource(R.drawable.icon\_box),"",Modifier

.size(40.dp))

Text(text = "Send a package",

fontSize = 18.sp,

color = Color(5, 96, 250))

Text(text = "Request for a driver to pick up or deliver your package for you",

fontSize = 12.sp)

}

}

}

Row (Modifier.padding(5.dp)) {

Box (

Modifier

.clip(RoundedCornerShape(20.dp))

.weight(1f),) {

Text(text = " ", Modifier

.size(200.dp)

.clickable { }

.background(Color(207, 207, 207)))

Column (

Modifier

.padding(10.dp)

.align(Alignment.Center)) {

Image(painterResource(R.drawable.icon\_fund),"",Modifier

.size(40.dp))

Text(text = "Fund your wallet",

fontSize = 18.sp,

color = Color(5, 96, 250))

Text(text = "To fund your wallet is as easy as ABC, make use of our fast technology and top-up your wallet today",

fontSize = 12.sp)

}

}

Spacer(Modifier.weight(0.1f))

Box (

Modifier

.clip(RoundedCornerShape(20.dp))

.weight(1f),) {

Text(text = " ", Modifier

.size(200.dp)

.clickable { }

.background(Color(207, 207, 207)))

Column (

Modifier

.padding(10.dp)

.align(Alignment.Center)) {

Image(painterResource(R.drawable.icon\_chat),"",Modifier

.size(40.dp))

Text(text = "Chats",

fontSize = 18.sp,

color = Color(5, 96, 250))

Text(text = "Search for available rider within your area",

fontSize = 12.sp)

}

}

}

}

}

}

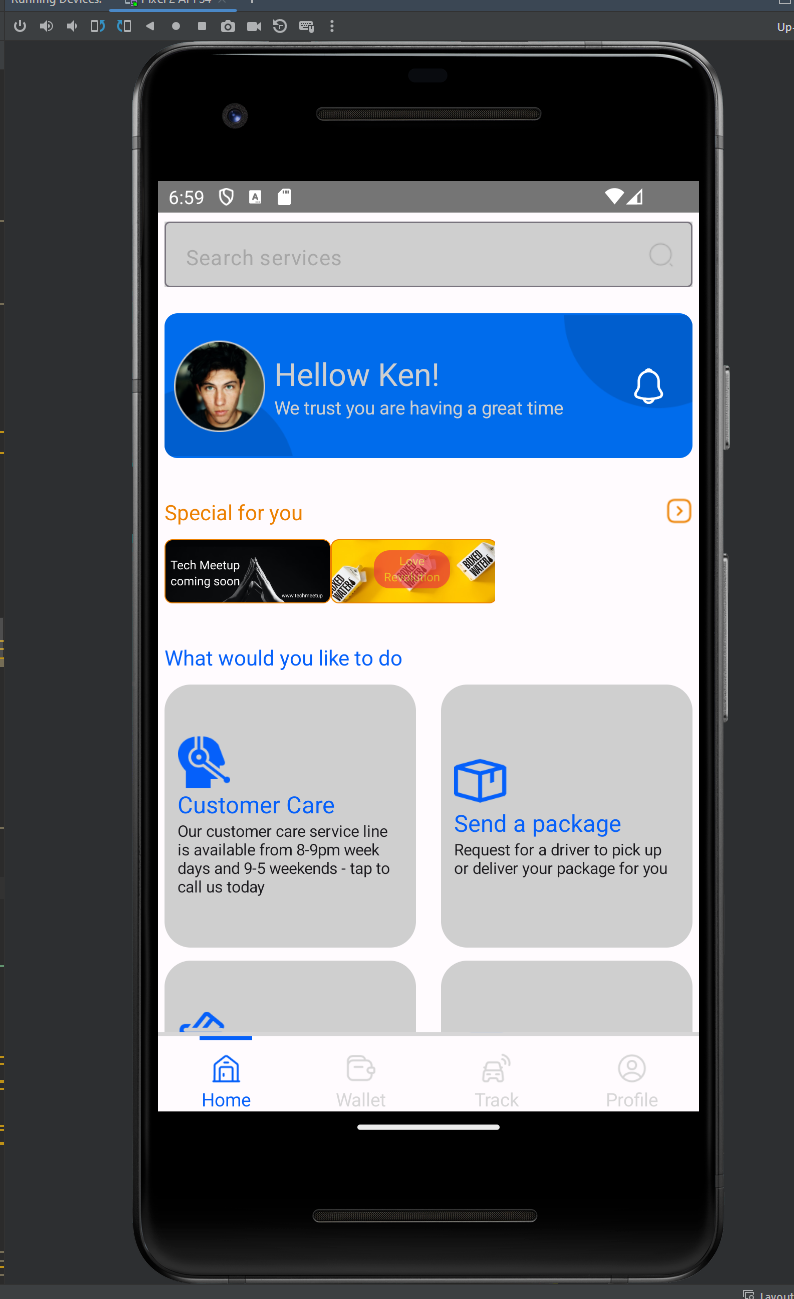


Рисунок 6 – Экран «Home»

Для перехода на «SendPackage» необходмио нажать на поле с посылкой, после чего откроется окно с заказом (листинг 7 и рисунок 7).

Листинг 7. Создание «SendPackage»

@Composable

fun SendPackage(paddingValues: PaddingValues, navController: NavHostController) {

var orig\_address by remember { mutableStateOf("") }

var orig\_country by remember { mutableStateOf("") }

var orig\_phone by remember { mutableStateOf("") }

var orig\_others by remember { mutableStateOf("") }

var dest\_address by remember { mutableStateOf("") }

var dest\_country by remember { mutableStateOf("") }

var dest\_phone by remember { mutableStateOf("") }

var dest\_others by remember { mutableStateOf("") }

var pack\_items by remember { mutableStateOf("") }

var pack\_weight by remember { mutableStateOf("") }

var pack\_worth by remember { mutableStateOf("") }

LazyColumn (

Modifier

.padding(paddingValues)

.fillMaxWidth()) {

items(1){

Spacer(Modifier.size(20.dp))

Row (

Modifier

.padding(5.dp)

.fillMaxWidth(),

verticalAlignment = Alignment.CenterVertically) {

Image(painterResource(R.drawable.icon\_origin),"",Modifier.size(20.dp))

Spacer(Modifier

.size(5.dp))

Text(text = "Origin Details")

}

OutlinedTextField(orig\_address, {orig\_address=it},

Modifier

.padding(5.dp)

.fillMaxWidth(),

label={

Row() {

Text("Adress",

color = Color(207,207,207))

}

}

)

OutlinedTextField(orig\_country, {orig\_country=it},

Modifier

.padding(5.dp)

.fillMaxWidth(),

label={

Row() {

Text("State, Country",

color = Color(207,207,207))

}

}

)

OutlinedTextField(orig\_phone, {orig\_phone=it},

Modifier

.padding(5.dp)

.fillMaxWidth(),

label={

Row() {

Text("Phone Number",

color = Color(207,207,207))

}

}

)

OutlinedTextField(orig\_others, {orig\_others=it},

Modifier

.padding(5.dp)

.fillMaxWidth(),

label={

Row() {

Text("Others",

color = Color(207,207,207))

}

}

)

Spacer(Modifier

.size(30.dp))

Row (Modifier

.padding(5.dp)

.fillMaxWidth(),

verticalAlignment = Alignment.CenterVertically) {

Image(painterResource(R.drawable.icon\_destination),"",Modifier.size(20.dp))

Spacer(Modifier

.size(5.dp))

Text(text = "Destination Details")

}

OutlinedTextField(dest\_address, {dest\_address=it},

Modifier

.padding(5.dp)

.fillMaxWidth(),

label={

Row() {

Text("Adress",

color = Color(207,207,207))

}

}

)

OutlinedTextField(dest\_country, {dest\_country=it},

Modifier

.padding(5.dp)

.fillMaxWidth(),

label={

Row() {

Text("State, Country",

color = Color(207,207,207))

}

}

)

OutlinedTextField(dest\_phone, {dest\_phone=it},

Modifier

.padding(5.dp)

.fillMaxWidth(),

label={

Row() {

Text("Phone Number",

color = Color(207,207,207))

}

}

)

OutlinedTextField(dest\_others, {dest\_others=it},

Modifier

.padding(5.dp)

.fillMaxWidth(),

label={

Row() {

Text("Others",

color = Color(207,207,207))

}

}

)

Spacer(Modifier

.size(10.dp))

Row (Modifier

.padding(5.dp)

.fillMaxWidth(),

verticalAlignment = Alignment.CenterVertically) {

Image(painterResource(R.drawable.icon\_add),"",Modifier.size(20.dp))

Spacer(Modifier

.size(5.dp))

Text(text = "Add destination")

}

Spacer(Modifier

.size(30.dp))

Row (

Modifier

.padding(5.dp)

.fillMaxWidth(),

verticalAlignment = Alignment.CenterVertically) {

Text(text = "Package Details")

}

OutlinedTextField(pack\_items, {pack\_items=it},

Modifier

.padding(5.dp)

.fillMaxWidth(),

label={

Row() {

Text("Package Items",

color = Color(207,207,207))

}

}

)

OutlinedTextField(pack\_weight, {pack\_weight=it},

Modifier

.padding(5.dp)

.fillMaxWidth(),

label={

Row() {

Text("Weight Of Item(kg)",

color = Color(207,207,207))

}

}

)

OutlinedTextField(pack\_worth, {pack\_worth=it},

Modifier

.padding(5.dp)

.fillMaxWidth(),

label={

Row() {

Text("Worth Of Item",

color = Color(207,207,207))

}

}

)

Spacer(Modifier

.size(30.dp))

Text(text = "Select delivery type", Modifier

.padding(5.dp,0.dp,0.dp,0.dp))

Spacer(Modifier

.size(10.dp))

Row (

Modifier

.padding(5.dp)

.fillMaxWidth()) {

Button(onClick = { },

Modifier

.height(70.dp)

.weight(1f),

shape = RoundedCornerShape(5.dp),

colors = ButtonDefaults.buttonColors(Color(207, 207, 207))

){ Column (horizontalAlignment = Alignment.CenterHorizontally) {

Image(painterResource(R.drawable.icon\_clock),"",

Modifier

.size(20.dp))

Text(text = "Instant delivery")}

}

Spacer(Modifier

.size(30.dp))

Button(onClick = { },

Modifier

.height(70.dp)

.weight(1f),

shape = RoundedCornerShape(5.dp),

colors = ButtonDefaults.buttonColors(Color(207, 207, 207))

){Column (horizontalAlignment = Alignment.CenterHorizontally) {

Image(painterResource(R.drawable.icon\_calendar),"",

Modifier

.size(20.dp))

Text(text = "Scheduled delivery")}}

}

}

}

}

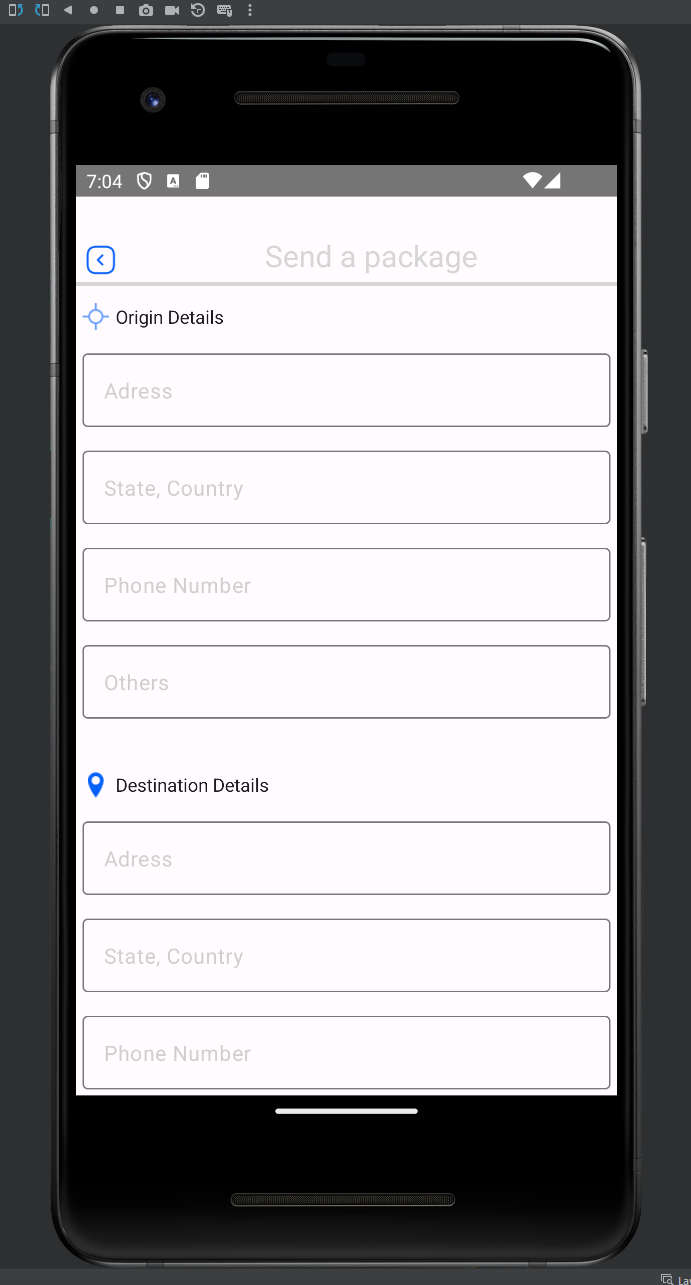


Рисунок 7 – Окно «SendPackage»

Также были созданы верхнее и нижнее меню, код которых представлен в листинге 8 и 8.1. Данные функции передавались в scaffold и отрисовывались на необходимых страницах.

Листинг 8. Верхнее меню

@Composable

fun TopBar(navController: NavHostController) {

val currentRoute = navController.currentBackStackEntry?.destination?.route

val navBackStackEntry by navController.currentBackStackEntryAsState()

when (val currentRoute = navBackStackEntry?.destination?.route) { }

if (PageCheck(currentRoute) || currentRoute=="Home"){ }

else {

Column (Modifier

.fillMaxWidth()) {

Row(

Modifier

.height(60.dp)

.fillMaxWidth(),

verticalAlignment = Alignment.Bottom,

horizontalArrangement = Arrangement.Center){

if (currentRoute=="Home") Text(text = "")

if (currentRoute=="Wallet") Text(text = "Wallet",

fontSize = 23.sp,

color = Color(205,205,205,205))

if (currentRoute=="Track") Text(text = "")

if (currentRoute=="Profile") Text(text = "Profile",

fontSize = 23.sp,

color = Color(205,205,205,205))

if (currentRoute=="AddPayment") {

Image(painterResource(R.drawable.icon\_blue\_errow\_left),"",

Modifier

.size(24.dp)

.clickable { navController.navigate("Profile") }

.weight(0.1f))

Text(text = "Add Payment method",Modifier

.weight(1f),

textAlign = TextAlign.Center,

fontSize = 23.sp,

color = Color(205, 205, 205, 205)

)

}

if (currentRoute=="Notification") {

Image(painterResource(R.drawable.icon\_blue\_errow\_left),"",

Modifier

.size(24.dp)

.clickable { navController.navigate("Profile") }

.weight(0.1f))

Text(

text = "Notification",Modifier

.weight(1f),

textAlign = TextAlign.Center,

fontSize = 23.sp,

color = Color(205, 205, 205, 205)

)

}

if (currentRoute=="SendPackage") {

Image(painterResource(R.drawable.icon\_blue\_errow\_left),"",

Modifier

.size(24.dp)

.clickable { navController.navigate("Home") }

.weight(0.1f))

Text(

text = "Send a package",Modifier

.weight(1f),

textAlign = TextAlign.Center,

fontSize = 23.sp,

color = Color(205, 205, 205, 205)

)

}

else {

}

}

Spacer (Modifier

.size(5.dp))

if (currentRoute=="Home" || currentRoute=="Track"){

}

else {

Divider(

Modifier

.fillMaxWidth()

.height(3.dp),

color = Color(205,205,205,205))

}

}

}

}

Листинг 8.1. Нижнее меню

@Composable

fun BottomBar(navController: NavHostController) {

val currentRoute = navController.currentBackStackEntry?.destination?.route

val navBackStackEntry by navController.currentBackStackEntryAsState()

when (val currentRoute = navBackStackEntry?.destination?.route) { }

if (PageCheck(currentRoute) || currentRoute=="Notification" || currentRoute=="AddPayment" || currentRoute=="SendPackage") { }

else {

Divider(

Modifier

.fillMaxWidth()

.height(3.dp),

color = Color(205,205,205,205))

Row(

Modifier

.height(60.dp)

.fillMaxWidth(),

verticalAlignment = Alignment.CenterVertically,

horizontalArrangement = Arrangement.SpaceAround){

Column (Modifier

.fillMaxHeight(),

horizontalAlignment = Alignment.CenterHorizontally) {

Text(text = "", Modifier

.padding(0.dp, 3.dp, 0.dp, 0.dp)

.height(3.dp)

.width(40.dp)

.background(

if (currentRoute == "Home") Color(5, 96, 250)

else Color(0, 0, 0, 0)

))

Spacer(Modifier

.size(7.dp))

Image(painterResource(R.drawable.icon\_home),"",

Modifier

.size(24.dp)

.clickable { navController.navigate("Home") }

.weight(0.1f),

colorFilter = ColorFilter.tint(if (currentRoute=="Home") Color(5,96,250)

else Color (205,205,205,205)))

Text(text = "Home",

color = if (currentRoute=="Home") Color(5,96,250)

else Color(205,205,205,205))

}

Column (Modifier

.fillMaxHeight(),horizontalAlignment = Alignment.CenterHorizontally) {

Text(text = "", Modifier

.padding(0.dp, 3.dp, 0.dp, 0.dp)

.height(3.dp)

.width(40.dp)

.background(

if (currentRoute == "Wallet") Color(5, 96, 250)

else Color(0, 0, 0, 0)

))

Spacer(Modifier

.size(7.dp))

Image(painterResource(R.drawable.icon\_wallet),"",

Modifier

.size(24.dp)

.clickable { navController.navigate("Wallet") }

.weight(0.1f),

colorFilter = ColorFilter.tint(if (currentRoute=="Wallet") Color(5,96,250)

else Color (205,205,205,205)))

Text(text = "Wallet",

color = if (currentRoute=="Wallet") Color(5,96,250)

else Color(205,205,205,205))

}

Column (Modifier

.fillMaxHeight(),horizontalAlignment = Alignment.CenterHorizontally) {

Text(text = "", Modifier

.padding(0.dp, 3.dp, 0.dp, 0.dp)

.height(3.dp)

.width(40.dp)

.background(

if (currentRoute == "Track") Color(5, 96, 250)

else Color(0, 0, 0, 0)

))

Spacer(Modifier

.size(7.dp))

Image(painterResource(R.drawable.icon\_track),"",

Modifier

.size(24.dp)

.clickable { navController.navigate("Track") }

.weight(0.1f),

colorFilter = ColorFilter.tint(if (currentRoute=="Track") Color(5,96,250)

else Color (205,205,205,205)))

Text(text = "Track",

color = if (currentRoute=="Track") Color(5,96,250)

else Color(205,205,205,205))

}

Column (Modifier

.fillMaxHeight(),horizontalAlignment = Alignment.CenterHorizontally) {

Text(text = "", Modifier

.padding(0.dp, 3.dp, 0.dp, 0.dp)

.height(3.dp)

.width(40.dp)

.background(

if (currentRoute == "Profile") Color(5, 96, 250)

else Color(0, 0, 0, 0)

))

Spacer(Modifier

.size(7.dp))

Image(painterResource(R.drawable.icon\_profile),"",

Modifier

.size(24.dp)

.clickable { navController.navigate("Profile") }

.weight(0.1f),

colorFilter = ColorFilter.tint(if (currentRoute=="Profile") Color(5,96,250)

else Color (205,205,205,205)))

Text(text = "Profile",

color = if (currentRoute=="Profile") Color(5,96,250)

else Color(205,205,205,205))

}

}

}

}