МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Факультет информационных систем и технологий  
Кафедра «Информационные системы»  
Дисциплина «Программирование мобильных устройств»

Лабораторная работа №2

Выполнил:  
студент гр. ПИбд-32  
Алейкин А. М.  
Проверил:  
доцент кафедры  
Филиппов А.А.

Ульяновск, 2023 г

**Задание**

Разработка экранных форм приложения и настройка навигации.

Необходимо:

1. На основе макетов, полученных в рамках ЛР № 1, разработать экранные формы приложения. При разработке экранных форм можно использовать фрагменты и/или композицию (compose).

2. Настроить навигацию между экранными формами приложения.

3. Разработать необходимые классы-сущности и создать на их основе тестовые наборы данных.

4. Использовать тестовые данные для демонстрации работы приложения.

**Выполнение**

1. Были разработаны экранные формы приложения на языке Kotlin с использованием композиции (compose). Пример простейшей экранной формы:

|  |
| --- |
| @Composable |
|  | fun MainScreen(navController: NavHostController) { |
|  | Column { |
|  | Box(modifier = Modifier |
|  | .padding(horizontal = 10.dp) |
|  | .fillMaxHeight(0.9f)) { |
|  | Column( |
|  | modifier = Modifier.verticalScroll(rememberScrollState()) |
|  | ) { |
|  | ListItem( |
|  | name = "Ламборджини Авентадор 2010", |
|  | mileage = 4765, |
|  | location = "Россия, г. Ульяновск", |
|  | price = 5000000, |
|  | imageResourceId = com.example.myapplication.R.drawable.lambo\_car1, |
|  | navController = navController |
|  | ) |
|  | ListItem( |
|  | name = "Оранжевый Москвич 1983", |
|  | mileage = 522011, |
|  | location = "Россия, г. Москва", |
|  | price = 125000, |
|  | imageResourceId = com.example.myapplication.R.drawable.moscowich\_car3, |
|  | navController = navController |
|  | ) |
|  | ListItem( |
|  | name = "Феррари Лаферрари 2013", |
|  | mileage = 17698, |
|  | location = "Италия, г. Неаполь", |
|  | price = 1249990, |
|  | imageResourceId = com.example.myapplication.R.drawable.ferrari\_laferrari\_car2, |
|  | navController = navController |
|  | ) |
|  | ListItem( |
|  | name = "Оранжевый Москвич 1983", |
|  | mileage = 522011, |
|  | location = "Россия, г. Москва", |
|  | price = 125000, |
|  | imageResourceId = com.example.myapplication.R.drawable.moscowich\_car3, |
|  | navController = navController |
|  | ) |
|  | ListItem( |
|  | name = "Феррари Лаферрари 2013", |
|  | mileage = 17698, |
|  | location = "Италия, г. Неаполь", |
|  | price = 1249990, |
|  | imageResourceId = com.example.myapplication.R.drawable.ferrari\_laferrari\_car2, |
|  | navController = navController |
|  | ) |
|  | } |
|  | } |
|  | Column(modifier = Modifier |
|  | .fillMaxSize() |
|  | .background(SkyBlue), |
|  | verticalArrangement = Arrangement.Center) { |
|  | navBar(navController = navController) |
|  | } |
|  | } |
|  | } |

2. Была настроена навигация между экранными формами приложения с помощью файла Navbar.

@Composable  
fun navBar(navController: NavHostController) {  
 Row(  
 horizontalArrangement = Arrangement.SpaceAround,  
 modifier = Modifier  
 .*fillMaxWidth*(),  
 verticalAlignment = Alignment.CenterVertically  
 ) **{** navItem(navController = navController, ImageId = R.drawable.*home\_image*, "mainScreen")  
 navItem(navController = navController, ImageId = R.drawable.*note\_image*, "editcard")  
 navItem(navController = navController, ImageId = R.drawable.*profile\_image*, "userSettings")  
 **}**}  
  
@Composable  
fun navItem(navController: NavHostController, ImageId: Int, destination: String) {  
 Image(  
 painter = painterResource(id = ImageId),  
 contentDescription = "image",  
 modifier = Modifier  
 .*height*(45.*dp*)  
 .*width*(45.*dp*)  
 .*clickable* **{** navController.navigate(destination) **}**)  
}  
  
@Composable  
fun navButton(navController: NavHostController, destination: String, label: String, backgroundColor: Color, textColor: Color)  
{  
 Button(onClick = **{** navController.navigate(destination) **}**,  
 modifier = Modifier  
 .*fillMaxWidth*()  
 .*padding*(top = 8.*dp*, bottom = 8.*dp*, start = 50.*dp*, end = 50.*dp*),  
 colors = ButtonDefaults.buttonColors( containerColor = backgroundColor),  
 shape = *RoundedCornerShape*(25.*dp*)  
 ) **{** Text(text = label,  
 fontSize = 20.*sp*,  
 color = textColor  
 )  
 **}**}

@Composable  
fun AppNavigation(navController: NavHostController){  
 NavHost(  
 navController = navController, startDestination = "authorization"  
 ) **{** *composable*("authorization") **{** Authorization(navController = navController)  
 **}** *composable*("mainScreen")**{** MainScreen(navController = navController)  
 **}** *composable*("registration") **{** Registration(navController = navController)  
 **}** *composable*("userSettings") **{** UserSettings(navController = navController)  
 **}** *composable*("editcard") **{** EditCardScreen(navController = navController)  
 **}** *composable*(  
 "editcard/{id}",  
 arguments = *listOf*(*navArgument*("id") **{** type = NavType.IntType **}**) //С аргументом  
 ) **{** backStackEntry **->** backStackEntry.arguments?.*let* **{** EditCardScreen(navController = navController, cardId = **it**.getInt("id"))  
 **}  
 }** *composable*("edituser")**{** EditUserScreen(navController = navController)  
 **}  
 }**}

3. Был разработан классы-сущности User.kt, Card.kt, содержащий в себе поля: название, расположение, пробег, цена и номер пользователя(продавец).

data class User(  
 val login: String,  
 val password: String  
)

data class Card(  
 val id: Int,

val name: String,  
 val location: String,  
 val mileage: Int,  
 val image: Int,

val price: Int

)

4. Демонстрация работы приложения. При запуске открывается форма авторизации, с которой можно перейти на форму регистрации. После успешной регистрации пользователя перебрасывает на главную страницу с объявлениями. После успешного входа в свой аккаунт пользователь попадает на главную страницу с автомобилями, внизу появляется навигационная панель, по которой можно также перейти в личный кабинет, либо в форму создания объявлений.

  





