Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра Програмної Інженерії

ЗВІТ

з дисципліни «Архітектура програмного забезпечення»

з лабораторної роботи №5

|  |  |
| --- | --- |
| Виконав | Перевірив: |
| ст. гр. ПЗПІ-20-7 | Старш. викл. кафедри ПІ |
| Крупчак Євгеній | Сокорчук І. П. |

Харків 2022

**ЛАБОРАТОРНА РОБОТА №5. РОЗРОБКА МОБІЛЬНОЇ ЧАСТИНИ**

**1.1 Мета роботи**

Мета п’ятої лабораторної роботи – розробити мобільний програмний  застосунок для будь-якої поширеної на сьогодні мобільної платформи для проєкту за темою «Програмна система для автоматизації видачі боксів із їжею».

**1.2 Хід роботи**

Мобільний застосунок написано на мові программування Kotlin за допомогою Jetpack Compose. У якость HTTP клієнту використовується Ktor.

На рисунку 1 зображено діаграму прецедентів, на якій зображено відношення між акторами та прецедентами в системі.

A picture containing text, screenshot, circle, font

Description automatically generated

Рисунок 1 – Діаграма прецендентів мобільного застосунку

На рисунку 2 зображено діаграму компонентів.

A screenshot of a computer

Description automatically generated with medium confidence

Рисунок 2 – Діаграма компонентів мобільного застосунку

На рисунку 3 зображено діаграму діяльності процесу створення замовлення.

A screenshot of a cell phone

Description automatically generated with medium confidence

Рисунок 3 – Діаграма діяльності процесу створення замовлення

На рисунку 4 зображено діаграму пакетів мобільної частини.

A black background with white rectangles

Description automatically generated with low confidence

Рисунок 4 – Діаграма пакетів мобільної частини

Пакет «activities» має у собі так звані активності, що представляють собою окремі екрани застосунку. Приклад наведено у додатку А.

Пакет «components» має у собі Composable компоненти, що представляють собою компоненти UI. Приклад такого компоненту наведено у додатку Б.

Пакет «viewmodels» має у собі моделі представлення, які містять у собі стан та логіку застосунку. Приклад такої моделі наведено у додатку В.

Пакет «data» має у собі дата класи відповідей та запитів до сервера. Приклад такого для замовлення:

@Serializable

data class Order(

val id: Int,

val price: Double,

val status: String,

val time: LocalDateTime,

val isBox: Boolean,

val customerId: String?,

val cateringId: Int,

val orderDishes: List<OrderDish>,

val catering: Catering)

**1.3 Висновки**

У данній лабораторній роботі було розроблено мобільну частину програмної системи для проєкту за темою «Програмна система для автоматизації видачі боксів із їжею».

Посилання на папку "DEMO" із відео та кодом на Google Drive: https://drive.google.com/drive/folders/1sBPljA5AFl66BZejiDRfry6l0LPELsV7?usp=share\_link

**ДОДАТОК А**

Програмний код файлу MainActivity.kt

package com.geneirodan.nextgenmeal.activities

import android.os.Bundle

import androidx.activity.compose.setContent

import androidx.compose.foundation.layout.Arrangement

import androidx.compose.foundation.layout.Box

import androidx.compose.foundation.layout.Row

import androidx.compose.foundation.layout.fillMaxWidth

import androidx.compose.foundation.layout.padding

import androidx.compose.material.BottomNavigation

import androidx.compose.material.BottomNavigationItem

import androidx.compose.material.Icon

import androidx.compose.material.MaterialTheme

import androidx.compose.material.Scaffold

import androidx.compose.material.Text

import androidx.compose.material.rememberScaffoldState

import androidx.compose.runtime.getValue

import androidx.compose.ui.Modifier

import androidx.compose.ui.res.stringResource

import androidx.navigation.compose.NavHost

import androidx.navigation.compose.composable

import androidx.navigation.compose.currentBackStackEntryAsState

import androidx.navigation.compose.rememberNavController

import com.geneirodan.nextgenmeal.utils.BottomItem

import com.geneirodan.nextgenmeal.components.LangDropDown

import com.geneirodan.nextgenmeal.components.pages.NewOrderPage

import com.geneirodan.nextgenmeal.components.pages.OrdersPage

import com.geneirodan.nextgenmeal.components.pages.SettingsPage

import com.geneirodan.nextgenmeal.components.AppTheme

import com.geneirodan.nextgenmeal.utils.Role

class MainActivity : BaseActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContent {

AppTheme {

val navController = rememberNavController()

scaffoldState = rememberScaffoldState()

val map = mapOf(

Pair(

Role.CUSTOMER,

listOf(BottomItem.Orders, BottomItem.NewOrders, BottomItem.Settings)

),

Pair(Role.EMPLOYEE, listOf(BottomItem.Orders, BottomItem.Settings))

)

AppTheme {

Scaffold(

scaffoldState = scaffoldState,

bottomBar = {

BottomNavigation(backgroundColor = MaterialTheme.colors.primary) {

val backStackEntry by navController.currentBackStackEntryAsState()

val currentRoute = backStackEntry?.destination?.route

map[role]!!.forEach {

val title = stringResource(it.title)

BottomNavigationItem(

selected = currentRoute == it.route,

onClick = { navController.navigate(it.route) },

icon = { Icon(it.icon, title) },

label = { Text(title) })

}

}

},

topBar = {

Row(

modifier = Modifier.fillMaxWidth(),

horizontalArrangement = Arrangement.End

) {

LangDropDown()

}

}

) {

Box(modifier = Modifier.padding(it)) {

NavHost(navController = navController, startDestination = "orders") {

composable("orders") { OrdersPage() }

composable("new\_order") { navController.NewOrderPage() }

composable("settings") { SettingsPage() }

}

}

}

}

}

}

}

}

**ДОДАТОК Б**

Програмний код компоненту CustomInputField.kt

package com.geneirodan.nextgenmeal.components

import androidx.compose.foundation.interaction.MutableInteractionSource

import androidx.compose.foundation.layout.Column

import androidx.compose.foundation.layout.padding

import androidx.compose.foundation.text.KeyboardActions

import androidx.compose.foundation.text.KeyboardOptions

import androidx.compose.material.LocalTextStyle

import androidx.compose.material.MaterialTheme

import androidx.compose.material.OutlinedTextField

import androidx.compose.material.Text

import androidx.compose.material.TextFieldColors

import androidx.compose.material.TextFieldDefaults

import androidx.compose.runtime.Composable

import androidx.compose.runtime.MutableState

import androidx.compose.runtime.getValue

import androidx.compose.runtime.mutableStateOf

import androidx.compose.runtime.remember

import androidx.compose.runtime.setValue

import androidx.compose.ui.Modifier

import androidx.compose.ui.focus.onFocusChanged

import androidx.compose.ui.graphics.Color

import androidx.compose.ui.graphics.Shape

import androidx.compose.ui.platform.LocalContext

import androidx.compose.ui.text.TextStyle

import androidx.compose.ui.text.input.VisualTransformation

import androidx.compose.ui.unit.dp

import com.geneirodan.nextgenmeal.utils.InputState

import com.geneirodan.nextgenmeal.viewmodels.abstractions.FormViewModel.Companion.onChange

import com.geneirodan.nextgenmeal.viewmodels.abstractions.FormViewModel.Companion.onFocusChange

@Composable

fun CustomInputField(

value: MutableState<InputState>,

modifier: Modifier = Modifier,

enabled: Boolean = true,

readOnly: Boolean = false,

textStyle: TextStyle = LocalTextStyle.current,

label: @Composable (() -> Unit)? = null,

placeholder: @Composable (() -> Unit)? = null,

leadingIcon: @Composable (() -> Unit)? = null,

trailingIcon: @Composable (() -> Unit)? = null,

visualTransformation: VisualTransformation = VisualTransformation.None,

keyboardOptions: KeyboardOptions = KeyboardOptions.Default,

keyboardActions: KeyboardActions = KeyboardActions(),

singleLine: Boolean = false,

maxLines: Int = Int.MAX\_VALUE,

interactionSource: MutableInteractionSource = remember { MutableInteractionSource() },

shape: Shape = MaterialTheme.shapes.small,

colors: TextFieldColors = TextFieldDefaults.outlinedTextFieldColors(),

) {

var touched by remember { mutableStateOf(false) }

val context = LocalContext.current

Column {

OutlinedTextField(

value = value.value.text,

placeholder = placeholder,

enabled = enabled,

readOnly = readOnly,

textStyle = textStyle,

label = label,

leadingIcon = leadingIcon,

trailingIcon = trailingIcon,

isError = !value.value.isValid,

visualTransformation = visualTransformation,

keyboardOptions = keyboardOptions,

keyboardActions = keyboardActions,

singleLine = singleLine,

maxLines = maxLines,

interactionSource = interactionSource,

shape = shape,

colors = colors,

onValueChange = {

touched = true

value.onChange(it)

},

modifier = modifier.onFocusChanged { if (touched) value.onFocusChange(context) },

)

if (!value.value.isValid && touched) {

Text(

text = value.value.errorMessage,

color = Color.Red,

modifier = Modifier.padding(start = 15.dp)

)

}

}

}

**ДОДАТОК В**

Програмний код класу RegisterViewModel

package com.geneirodan.nextgenmeal.viewmodels

import android.content.Context

import androidx.compose.material.ScaffoldState

import androidx.compose.runtime.MutableState

import androidx.compose.runtime.mutableStateOf

import androidx.lifecycle.viewModelScope

import com.geneirodan.nextgenmeal.Api

import com.geneirodan.nextgenmeal.data.requests.RegisterRequest

import com.geneirodan.nextgenmeal.utils.InputState

import com.geneirodan.nextgenmeal.utils.InputType

import com.geneirodan.nextgenmeal.viewmodels.abstractions.FormViewModel

import io.ktor.http.HttpStatusCode

import kotlinx.coroutines.launch

class RegisterViewModel : FormViewModel() {

val name = mutableStateOf(InputState(type = InputType.TEXT))

val email = mutableStateOf(InputState(type = InputType.EMAIL))

val password = mutableStateOf(InputState(type = InputType.PASSWORD))

val confirmPassword = mutableStateOf(InputState(type = InputType.PASSWORD))

override val fields = mutableListOf(name, email, password, confirmPassword)

fun register(

openDialog: MutableState<Boolean>,

scaffoldState: ScaffoldState,

context: Context

) {

onFocusChanged(context)

if (validate()) viewModelScope.launch {

val response = Api.post<RegisterRequest>(

url = "Account/Register",

params = mapOf(Pair("callbackUrl", "/")),

body = RegisterRequest(

name.value.text,

email.value.text,

password.value.text,

confirmPassword.value.text

)

)

if (response.status == HttpStatusCode.Created) {

openDialog.value = true

} else Api.handleError(response, scaffoldState)

}

}

}