

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
Кафедра Програмної Інженерії

ЗВІТ
з дисципліни «Архітектура програмного забезпечення»
з лабораторної роботи №5

Виконав
ст. гр. ПЗП-20-7
Крупчак Євгеній

Перевірив:
Старш. викл. кафедри ПІ
Сокорчук І. П.

Харків 2022

Рисунок 1 – Діаграма прецедентів мобільного застосунку

На рисунку 2 зображено діаграму компонентів.

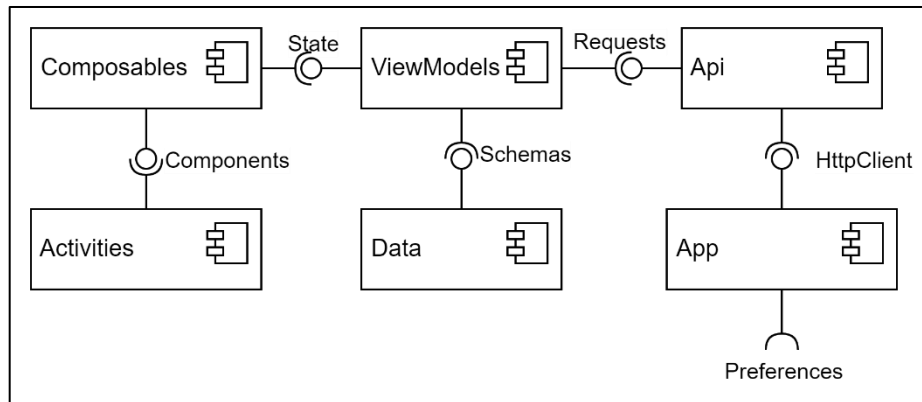


Рисунок 2 – Діаграма компонентів мобільного застосунку

На рисунку 3 зображено діаграму діяльності процесу створення замовлення.

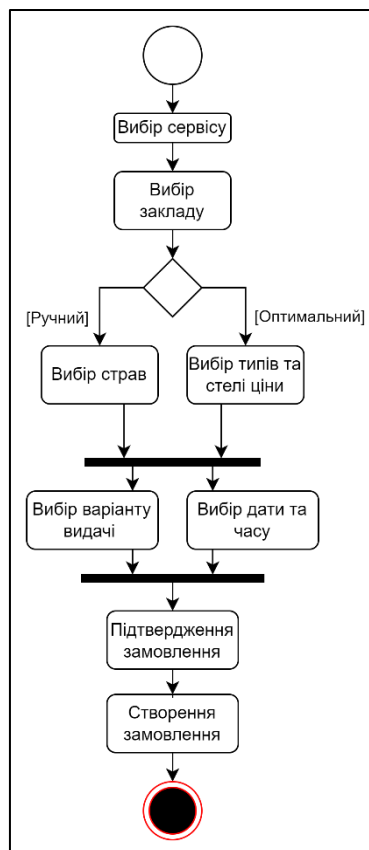


Рисунок 3 – Діаграма діяльності процесу створення замовлення

На рисунку 4 зображено діаграму пакетів мобільної частини.

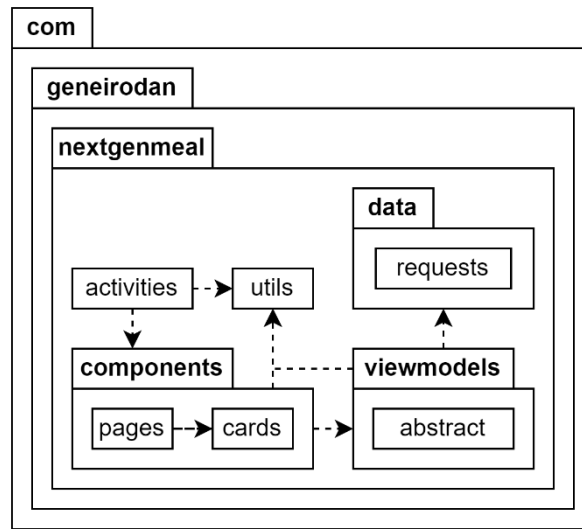


Рисунок 4 – Діаграма пакетів мобільної частини

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 дані 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 проекту 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/1sBPljA5AF166BZejiDRfry6l0LPELsV7?usp=share_link

ДОДАТОК А

Програмний код файлу MainActivity.kt

```
1  package com.geneirodan.nextgenmeal.activities
2
3  import android.os.Bundle
4  import androidx.activity.compose.setContent
5  import androidx.compose.foundation.layout.Arrangement
6  import androidx.compose.foundation.layout.Box
7  import androidx.compose.foundation.layout.Row
8  import androidx.compose.foundation.layout.fillMaxWidth
9  import androidx.compose.foundation.layout.padding
10 import androidx.compose.material.BottomNavigation
11 import androidx.compose.material.BottomNavigationBarItem
12 import androidx.compose.material.Icon
13 import androidx.compose.material.MaterialTheme
14 import androidx.compose.material.Scaffold
15 import androidx.compose.material.Text
16 import androidx.compose.material.rememberScaffoldState
17 import androidx.compose.runtime.getValue
18 import androidx.compose.ui.Modifier
19 import androidx.compose.ui.res.stringResource
20 import androidx.navigation.compose.NavHost
21 import androidx.navigation.compose.composable
22 import androidx.navigation.compose.currentBackStackEntryAsState
23 import androidx.navigation.compose.rememberNavController
24 import com.geneirodan.nextgenmeal.utils.BottomItem
25 import com.geneirodan.nextgenmeal.components.LangDropDown
26 import com.geneirodan.nextgenmeal.components.pages.NewOrderPage
27 import com.geneirodan.nextgenmeal.components.pages.OrdersPage
28 import com.geneirodan.nextgenmeal.components.pages.SettingsPage
29 import com.geneirodan.nextgenmeal.components.AppTheme
30 import com.geneirodan.nextgenmeal.utils.Role
31
32 class MainActivity : BaseActivity() {
33     override fun onCreate(savedInstanceState: Bundle?) {
34         super.onCreate(savedInstanceState)
35         setContent {
36             AppTheme {
37                 val navController = rememberNavController()
38                 scaffoldState = rememberScaffoldState()
39                 val map = mapOf(
40                     Pair(
41                         Role.CUSTOMER,
42                         listOf(BottomItem.Orders, BottomItem.NewOrders,
43 BottomItem.Settings)
44                     ),
45                     Pair(Role.EMPLOYEE, listOf(BottomItem.Orders,
46 BottomItem.Settings))

```

```

47         )
48         AppTheme {
49             Scaffold(
50                 scaffoldState = scaffoldState,
51                 bottomBar = {
52                     BottomNavigation(backgroundColor =
53 MaterialTheme.colors.primary) {
54                         val backStackEntry by
55 navController.currentBackStackEntryAsState()
56                         val currentRoute =
57 backStackEntry?.destination?.route
58                         map[role]!!.forEach {
59                             val title = stringResource(it.title)
60                             BottomNavigationItem(
61                                 selected = currentRoute ==
62 it.route,
63                                 onClick = {
64 navController.navigate(it.route) },
65                                 icon = { Icon(it.icon, title) },
66                                 label = { Text(title) })
67                         }
68                     },
69                 topBar = {
70                     Row(
71                         modifier = Modifier.fillMaxWidth(),
72                         horizontalArrangement = Arrangement.End
73                     ) {
74                         LangDropDown()
75                     }
76                 }
77             ) {
78                 Box(modifier = Modifier.padding(it)) {
79                     NavHost(navController = navController,
80 startDestination = "orders") {
81                         composable("orders") { OrdersPage() }
82                         composable("new_order") {
83 navController.NewOrderPage() }
84                         composable("settings") { SettingsPage() }
85                     }
86                 }
87             }
88         }
89     }
90 }
91 }
92 }
93 }

```

ДОДАТОК Б

Програмний код компоненту CustomInputField.kt

```
1 package com.geneirodan.nextgenmeal.components
2
3 import androidx.compose.foundation.interaction.MutableInteractionSource
4 import androidx.compose.foundation.layout.Column
5 import androidx.compose.foundation.layout.padding
6 import androidx.compose.foundation.text.KeyboardActions
7 import androidx.compose.foundation.text.KeyboardOptions
8 import androidx.compose.material.LocalTextStyle
9 import androidx.compose.material.MaterialTheme
10 import androidx.compose.material.OutlinedTextField
11 import androidx.compose.material.Text
12 import androidx.compose.material.TextFieldColors
13 import androidx.compose.material.TextFieldDefaults
14 import androidx.compose.runtime.Composable
15 import androidx.compose.runtime.MutableState
16 import androidx.compose.runtime.getValue
17 import androidx.compose.runtime.mutableStateOf
18 import androidx.compose.runtime.remember
19 import androidx.compose.runtime.setValue
20 import androidx.compose.ui.Modifier
21 import androidx.compose.ui.focus.onFocusChanged
22 import androidx.compose.ui.graphics.Color
23 import androidx.compose.ui.graphics.Shape
24 import androidx.compose.ui.platform.LocalContext
25 import androidx.compose.ui.text.TextStyle
26 import androidx.compose.ui.text.input.VisualTransformation
27 import androidx.compose.ui.unit.dp
28 import com.geneirodan.nextgenmeal.utils.InputState
29 import
30 com.geneirodan.nextgenmeal.viewmodels.abstractions.FormViewModel.Companion
31 n.onChange
32 import
33 com.geneirodan.nextgenmeal.viewmodels.abstractions.FormViewModel.Companion
34 n.onFocusChange
35
36 @Composable
37 fun CustomInputField(
38     value: MutableState<InputState>,
39     modifier: Modifier = Modifier,
40     enabled: Boolean = true,
41     readOnly: Boolean = false,
42     textStyle: TextStyle = LocalTextStyle.current,
43     label: @Composable (() -> Unit)? = null,
44     placeholder: @Composable (() -> Unit)? = null,
45     leadingIcon: @Composable (() -> Unit)? = null,
46     trailingIcon: @Composable (() -> Unit)? = null,
```



```

47     visualTransformation: VisualTransformation =
48     VisualTransformation.None,
49     keyboardOptions: KeyboardOptions = KeyboardOptions.Default,
50     keyboardActions: KeyboardActions = KeyboardActions(),
51     singleLine: Boolean = false,
52     maxLines: Int = Int.MAX_VALUE,
53     interactionSource: MutableInteractionSource = remember {
54     MutableInteractionSource() },
55     shape: Shape = MaterialTheme.shapes.small,
56     colors: TextFieldColors =
57     TextFieldDefaults.outlinedTextFieldColors(),
58 ) {
59     var touched by remember { mutableStateOf(false) }
60     val context = LocalContext.current
61     Column {
62         OutlinedTextField(
63             value = value.value.text,
64             placeholder = placeholder,
65             enabled = enabled,
66             readOnly = readOnly,
67             textStyle = textStyle,
68             label = label,
69             leadingIcon = leadingIcon,
70             trailingIcon = trailingIcon,
71             isError = !value.value.isValid,
72             visualTransformation = visualTransformation,
73             keyboardOptions = keyboardOptions,
74             keyboardActions = keyboardActions,
75             singleLine = singleLine,
76             maxLines = maxLines,
77             interactionSource = interactionSource,
78             shape = shape,
79             colors = colors,
80             onValueChange = {
81                 touched = true
82                 value.onChange(it)
83             },
84             modifier = modifier.onFocusChanged { if (touched)
85 value.onFocusChange(context) },
86         )
87         if (!value.value.isValid && touched) {
88             Text(
89                 text = value.value.errorMessage,
90                 color = Color.Red,
91                 modifier = Modifier.padding(start = 15.dp)
92             )
93         }
94     }
95 }

```

ДОДАТОК В

Програмний код класу RegisterViewModel

```
1 package com.geneirodan.nextgenmeal.viewmodels
2
3 import android.content.Context
4 import androidx.compose.material.ScaffoldState
5 import androidx.compose.runtime.MutableState
6 import androidx.compose.runtime.mutableStateOf
7 import androidx.lifecycle.ViewModelScope
8 import com.geneirodan.nextgenmeal.Api
9 import com.geneirodan.nextgenmeal.data.requests.RegisterRequest
10 import com.geneirodan.nextgenmeal.utils.InputState
11 import com.geneirodan.nextgenmeal.utils.InputType
12 import com.geneirodan.nextgenmeal.viewmodels.abstractions.FormViewModel
13 import io.ktor.http.HttpStatusCode
14 import kotlinx.coroutines.launch
15
16 class RegisterViewModel : FormViewModel() {
17     val name = mutableStateOf(InputState(type = InputType.TEXT))
18     val email = mutableStateOf(InputState(type = InputType.EMAIL))
19     val password = mutableStateOf(InputState(type = InputType.PASSWORD))
20     val confirmPassword = mutableStateOf(InputState(type =
21 InputType.PASSWORD))
22     override val fields = mutableListOf(name, email, password,
23 confirmPassword)
24     fun register(
25         openDialog: MutableState<Boolean>,
26         scaffoldState: ScaffoldState,
27         context: Context
28     ) {
29         onFocusChanged(context)
30         if (validate()) viewModelScope.launch {
31             val response = Api.post<RegisterRequest>(
32                 url = "Account/Register",
33                 params = mapOf(Pair("callbackUrl", "/")),
34                 body = RegisterRequest(
35                     name.value.text,
36                     email.value.text,
37                     password.value.text,
38                     confirmPassword.value.text
39                 )
40             )
41             if (response.status == HttpStatusCode.Created) {
42                 openDialog.value = true
43             } else Api.handleError(response, scaffoldState)
44         }
45     }
46 }
```