

SNACK SQUAD: A Customizable Snack Ordering and Delivery App

Application Name: MOFOOD Delivery App

Project Manual

SMARTINTERN App Development Project

Team Members

AKSHAT OMAR – 21BAI1792

SAJAL BRAHMA – 21BPS1045

RUSHIL BHUPTA – 21BCT0381

PREM SAGAR – 21BCE5563

Repository Links

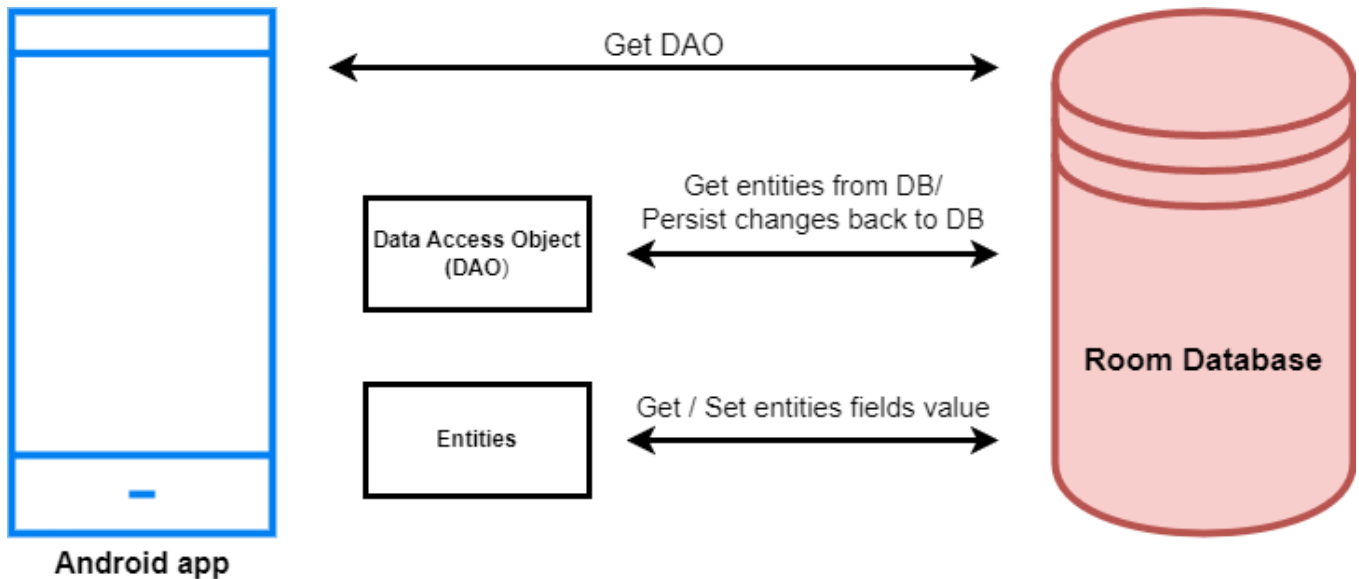
Project Repository Link: <https://github.com/AOkun1205/mofoodz>

Group Repository Link: <https://github.com/smartinternz02/Sl-GuidedProject-587493-1696871724>

Snack Squad: A Customizable Snack Ordering and Delivery App

A project that demonstrates the use of Android Jetpack Compose to build a UI for a snack squad app. Snack Squad is a sample project built using the Android ComposeUI toolkit. It demonstrates how to create a simple e-commerce app for snacks using the Compose libraries. The user can see a list of snacks, and by tapping on a snack, and by tapping on the "Add to Cart" button, the snack will be added to the cart. The user can also see the list of items in the cart and can proceed to checkout to make the purchase.

Architecture



Learning Outcomes:

By the end of this project:

- We will be able to work on Android Studios and build an app.
- We will be able to integrate the database accordingly.

Project Workflow:

- Users register into the application.
- After registration, user logs into the application.
- User enters into the main page
- User can view the items, select and order the items
- From admin login we can view the orders placed.

Tasks:

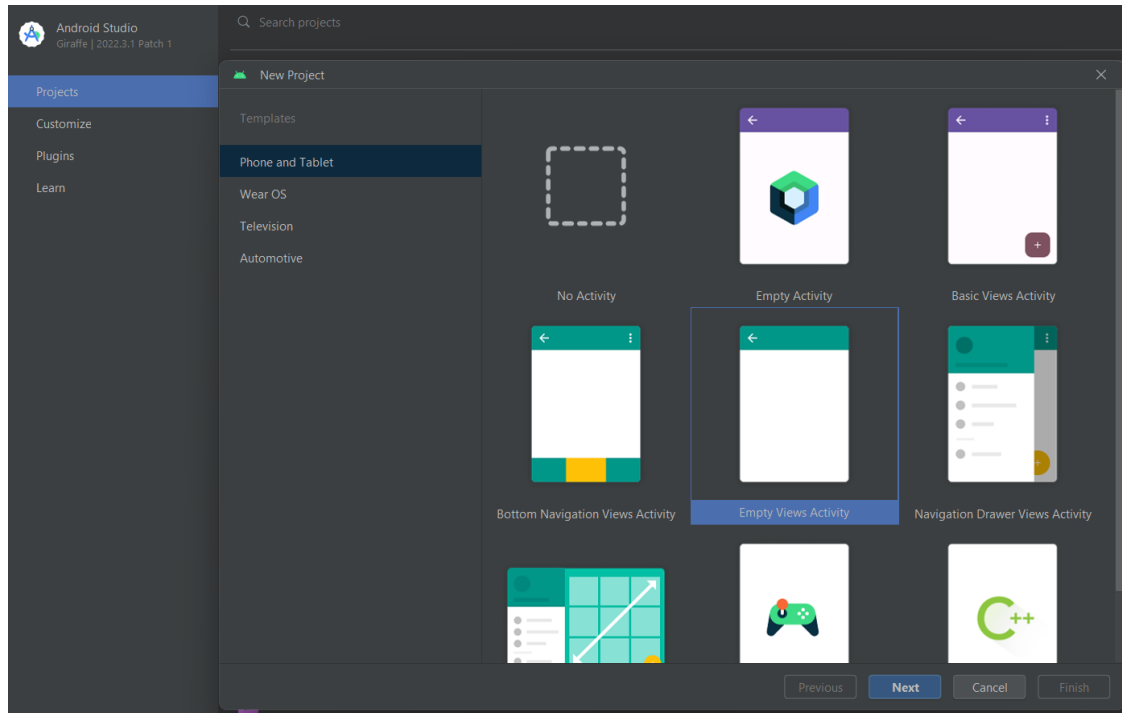
1. Install Android Studios using the provided link:

<https://developer.android.com/studio/install>

2. Create new project:

Step 1: Android studio > File > New > New Project > Empty Compose Activity.

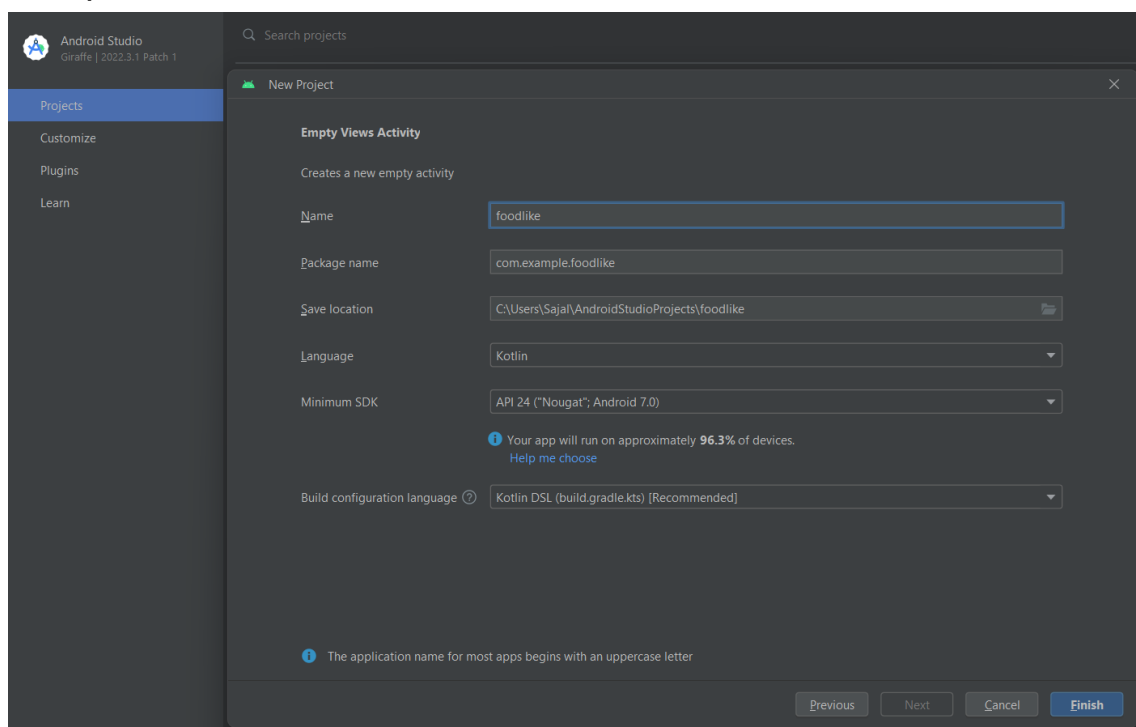
Step 2: Click on **Next** button.



Step 3: Give name to the new project.

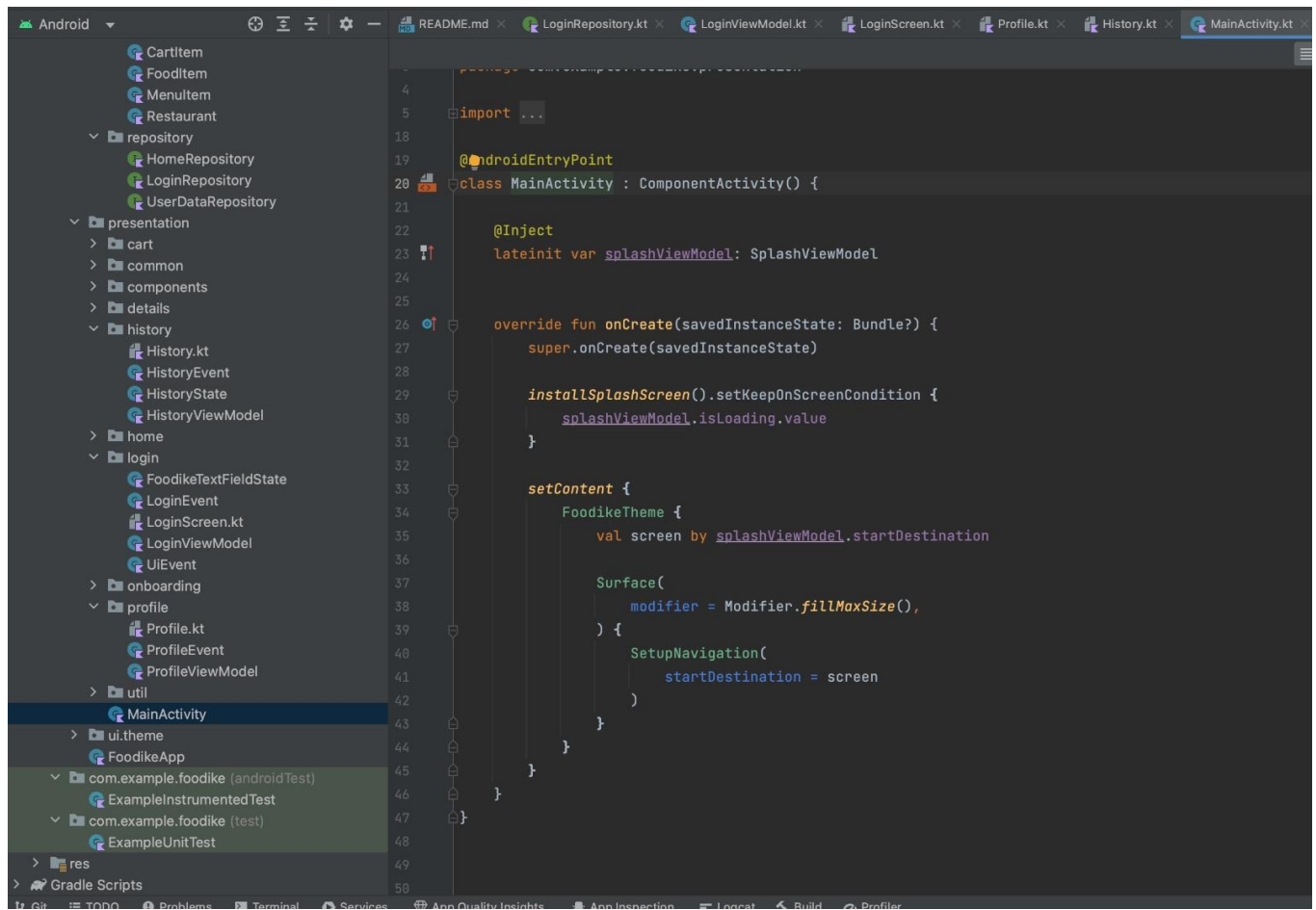
Step 4: Give the Minimum SDK value.

Step 5: Click Finish.



MainActivity.kt file:

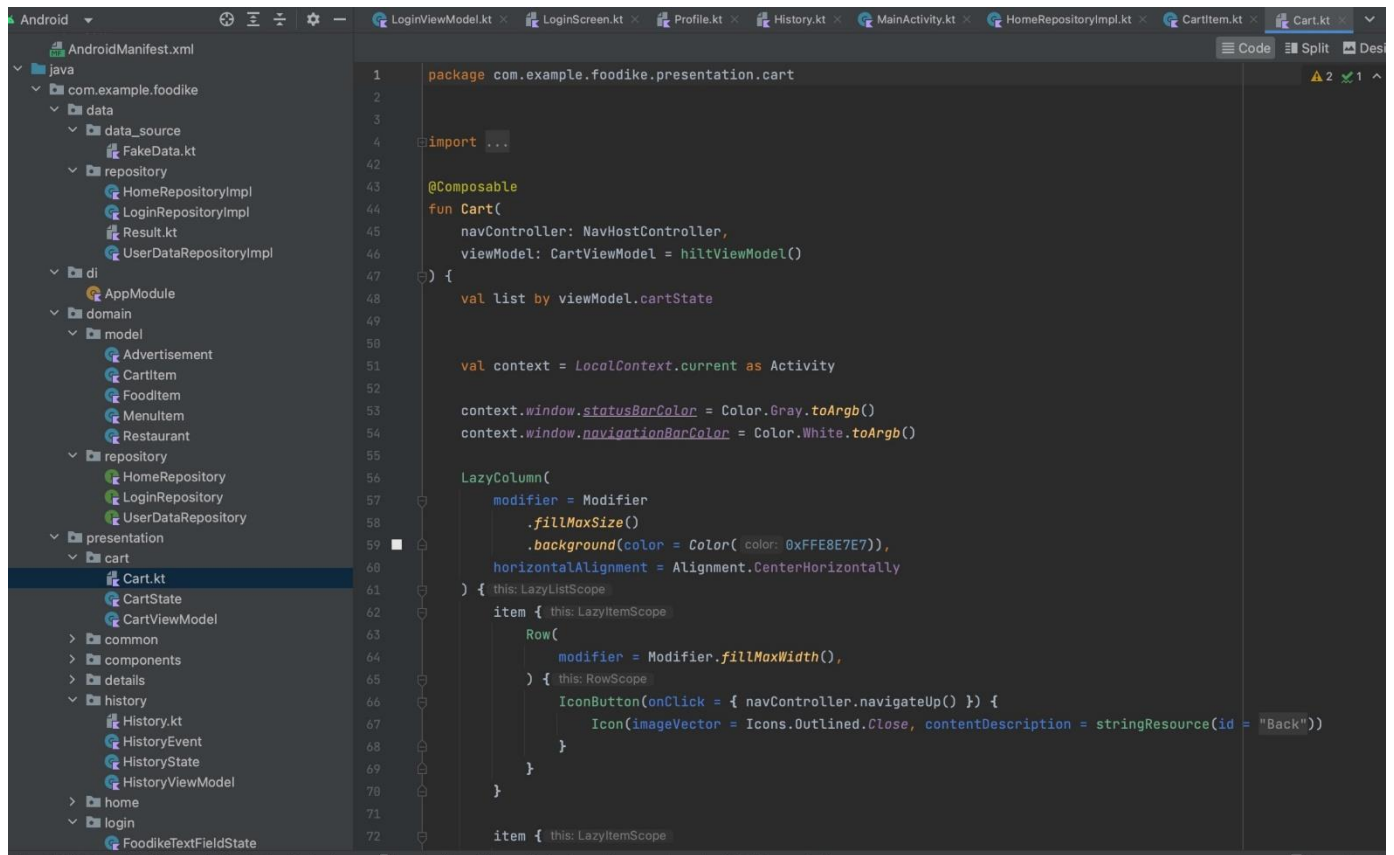
<https://github.com/AOkun1205/mofoodz/blob/main/app/src/main/java/com/example/foodike/presentation/MainActivity.kt>



```
1 package com.example.foodike.presentation
2
3
4
5 import ..
6
7
8
9
10
11
12
13
14
15
16
17
18
19 @AndroidEntryPoint
20 class MainActivity : AppCompatActivity() {
21
22     @Inject
23     lateinit var splashViewModel: SplashViewModel
24
25
26     override fun onCreate(savedInstanceState: Bundle?) {
27         super.onCreate(savedInstanceState)
28
29         installSplashScreen().setKeepOnScreenCondition {
30             splashViewModel.isLoading.value
31         }
32
33         setContent {
34             FoodikeTheme {
35                 val screen by splashViewModel.startDestination
36
37                 Surface(
38                     modifier = Modifier.fillMaxSize(),
39                 ) {
40                     SetupNavigation(
41                         startDestination = screen
42                     )
43                 }
44             }
45         }
46     }
47 }
48
49
50
```

Cart.kt file:

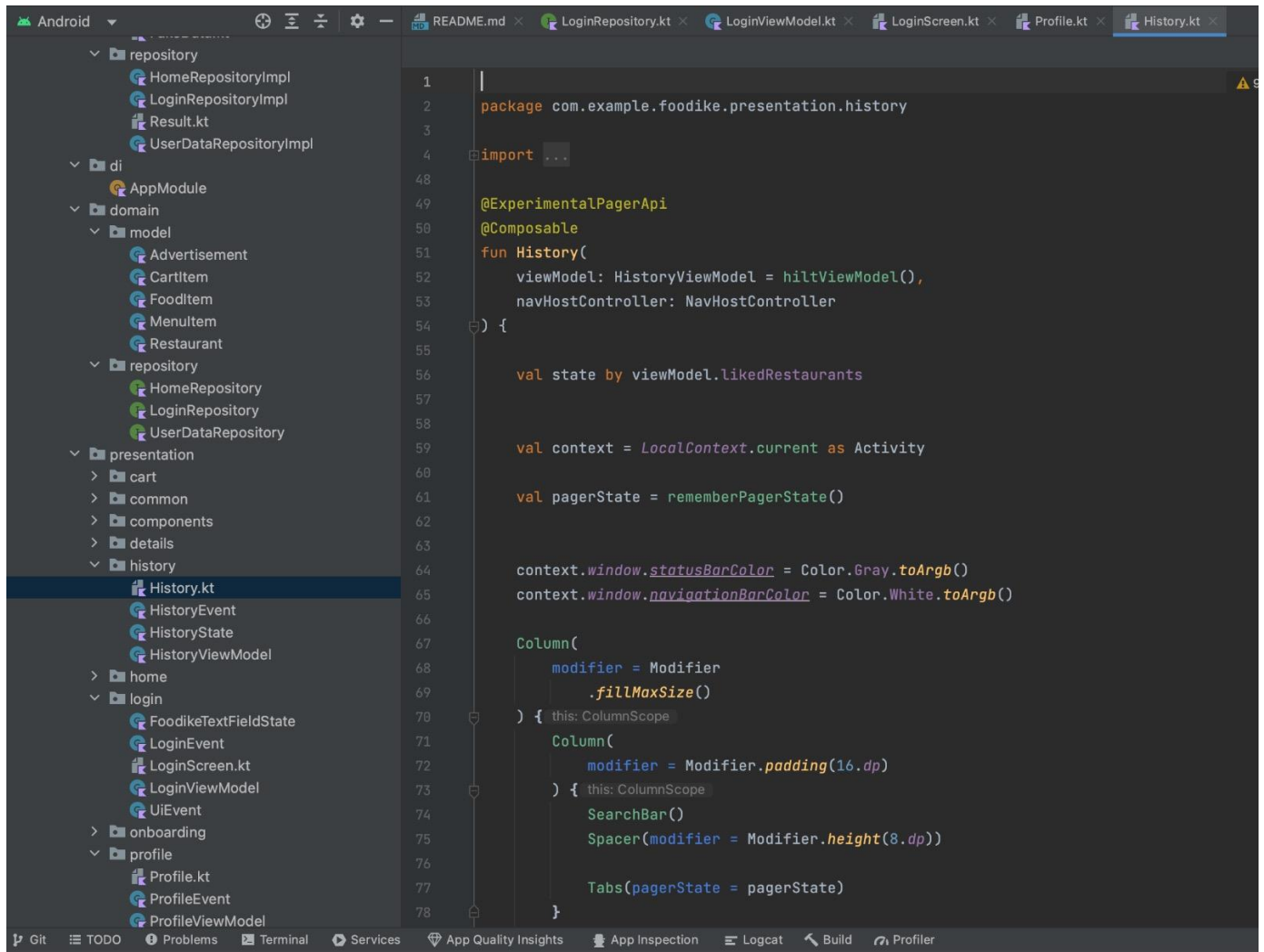
<https://github.com/AOkun1205/mofoodz/blob/main/app/src/main/java/com/example/foodike/presentation/cart/Cart.kt>



```
1 package com.example.foodike.presentation.cart
2
3
4 import ...
5
6 @Composable
7 fun Cart(
8     navController: NavHostController,
9     viewModel: CartViewModel = hiltViewModel()
10 ) {
11     val list by viewModel.cartState
12
13     val context = LocalContext.current as Activity
14
15     context.window.statusBarColor = Color.Gray.toArgb()
16     context.window.navigationBarColor = Color.White.toArgb()
17
18     LazyColumn(
19         modifier = Modifier
20             .fillMaxSize()
21             .background(color = Color( color = 0xFFE8E7E7)),
22         horizontalAlignment = Alignment.CenterHorizontally
23     ) {
24         item { this: LazyListScope
25             Row(
26                 modifier = Modifier.fillMaxWidth(),
27             ) {
28                 this: RowScope
29                 IconButton(onClick = { navController.navigateUp() }) {
30                     Icon(imageVector = Icons.Outlined.Close, contentDescription = stringResource(id = "Back"))
31                 }
32             }
33         }
34     }
35
36     item { this: LazyItemScope
```

History.kt file:

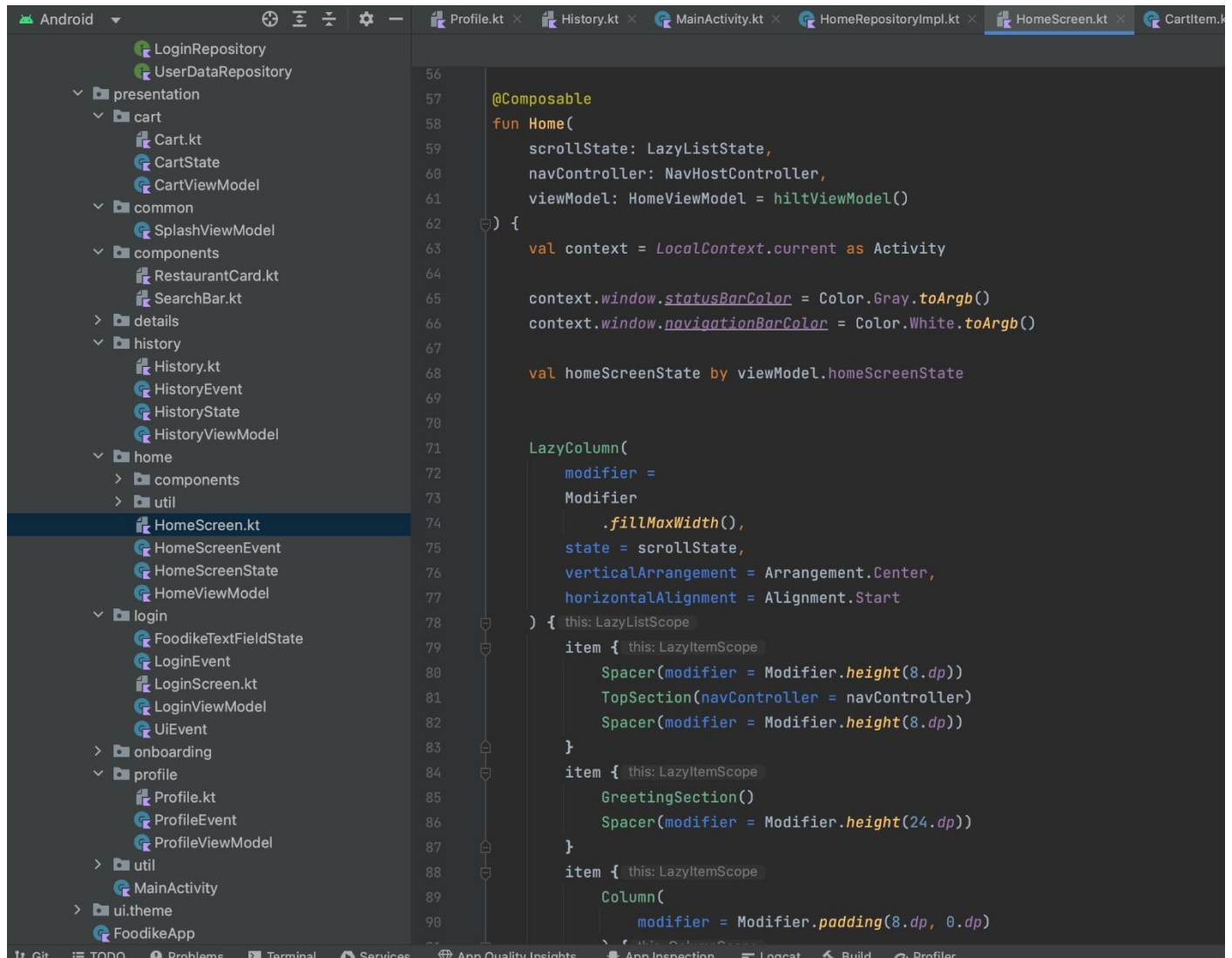
[mofoodz/app/src/main/java/com/example/foodike/presentation/history/History.kt](https://github.com/AOkun1205/mofoodz/blob/main/app/src/main/java/com/example/foodike/presentation/history/History.kt) at main · AOkun1205/mofoodz · GitHub



```
1 |
2 package com.example.foodike.presentation.history
3
4 import ...
5
48
49 @ExperimentalPagerApi
50 @Composable
51 fun History(
52     viewModel: HistoryViewModel = hiltViewModel(),
53     navHostController: NavHostController
54 ) {
55
56     val state by viewModel.likedRestaurants
57
58     val context = LocalContext.current as Activity
59
60     val pagerState = rememberPagerState()
61
62
63
64     context.window.statusBarColor = Color.Gray.toArgb()
65     context.window.navigationBarColor = Color.White.toArgb()
66
67     Column(
68         modifier = Modifier
69             .fillMaxSize()
70     ) { this: ColumnScope
71         Column(
72             modifier = Modifier.padding(16.dp)
73         ) { this: ColumnScope
74             SearchBar()
75             Spacer(modifier = Modifier.height(8.dp))
76
77             Tabs(pagerState = pagerState)
78         }
79     }
```

HomeScreen.kt file:

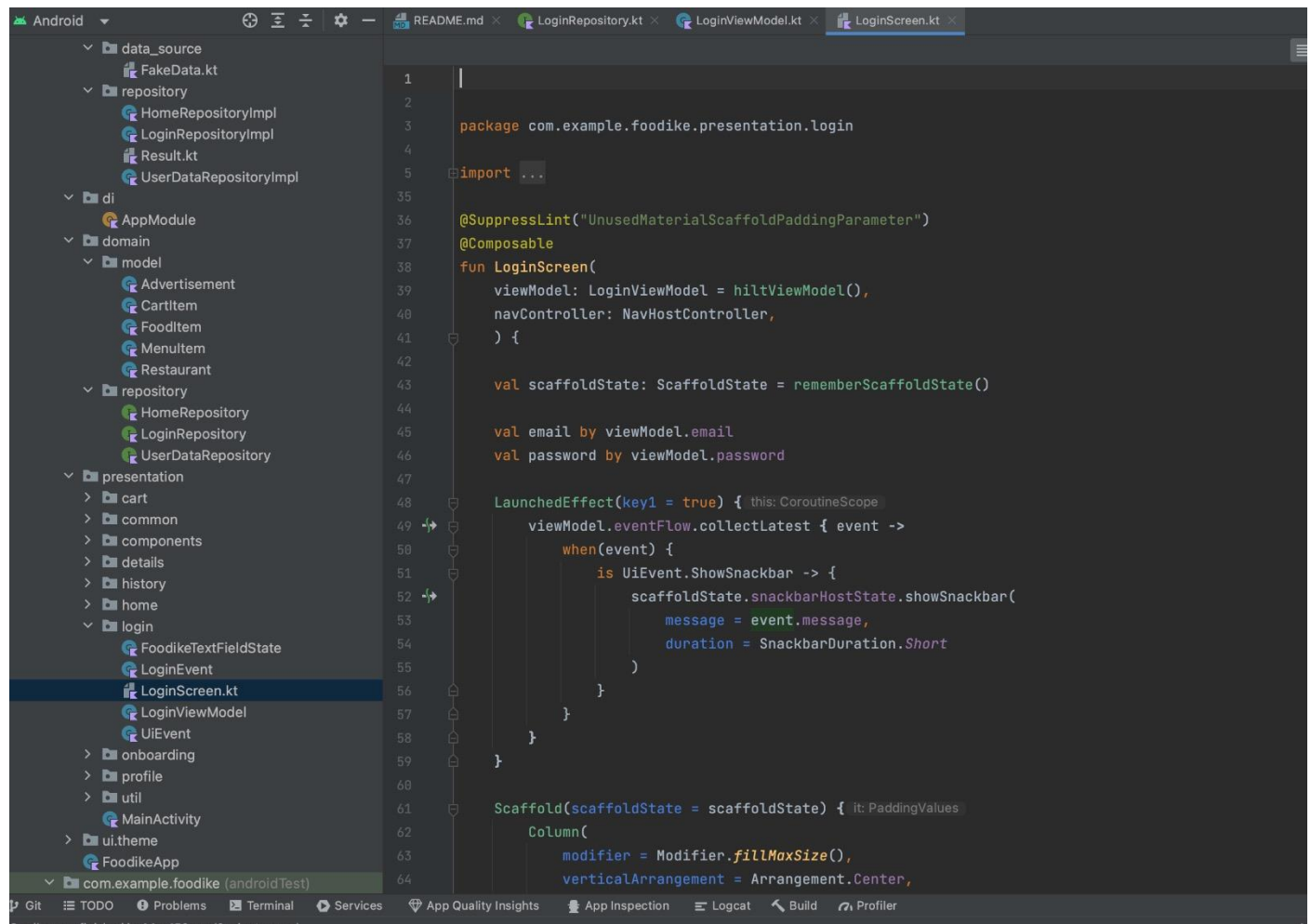
[mofoodz/app/src/main/java/com/example/foodike/presentation/home/HomeScreen.kt](https://github.com/AOkun1205/mofoodz/blob/main/app/src/main/java/com/example/foodike/presentation/home/HomeScreen.kt) at main · AOkun1205/mofoodz · GitHub



```
56
57 @Composable
58 fun Home(
59     scrollState: LazyListState,
60     navController: NavHostController,
61     viewModel: HomeViewModel = hiltViewModel()
62 ) {
63     val context = LocalContext.current as Activity
64
65     context.window.statusBarColor = Color.Gray.toArgb()
66     context.window.navigationBarColor = Color.White.toArgb()
67
68     val homeScreenState by viewModel.homeScreenState
69
70
71     LazyColumn(
72         modifier =
73             Modifier
74                 .fillMaxWidth(),
75         state = scrollState,
76         verticalArrangement = Arrangement.Center,
77         horizontalAlignment = Alignment.Start
78     ) {
79         this: LazyListScope
80         item { this: LazyItemScope
81             Spacer(modifier = Modifier.height(8.dp))
82             TopSection(navController = navController)
83             Spacer(modifier = Modifier.height(8.dp))
84         }
85         item { this: LazyItemScope
86             GreetingSection()
87             Spacer(modifier = Modifier.height(24.dp))
88         }
89         item { this: LazyItemScope
90             Column(
91                 modifier = Modifier.padding(8.dp, 0.dp)
92             ) {
93                 // ...
94             }
95         }
96     }
97 }
```

LoginScreen.kt file:

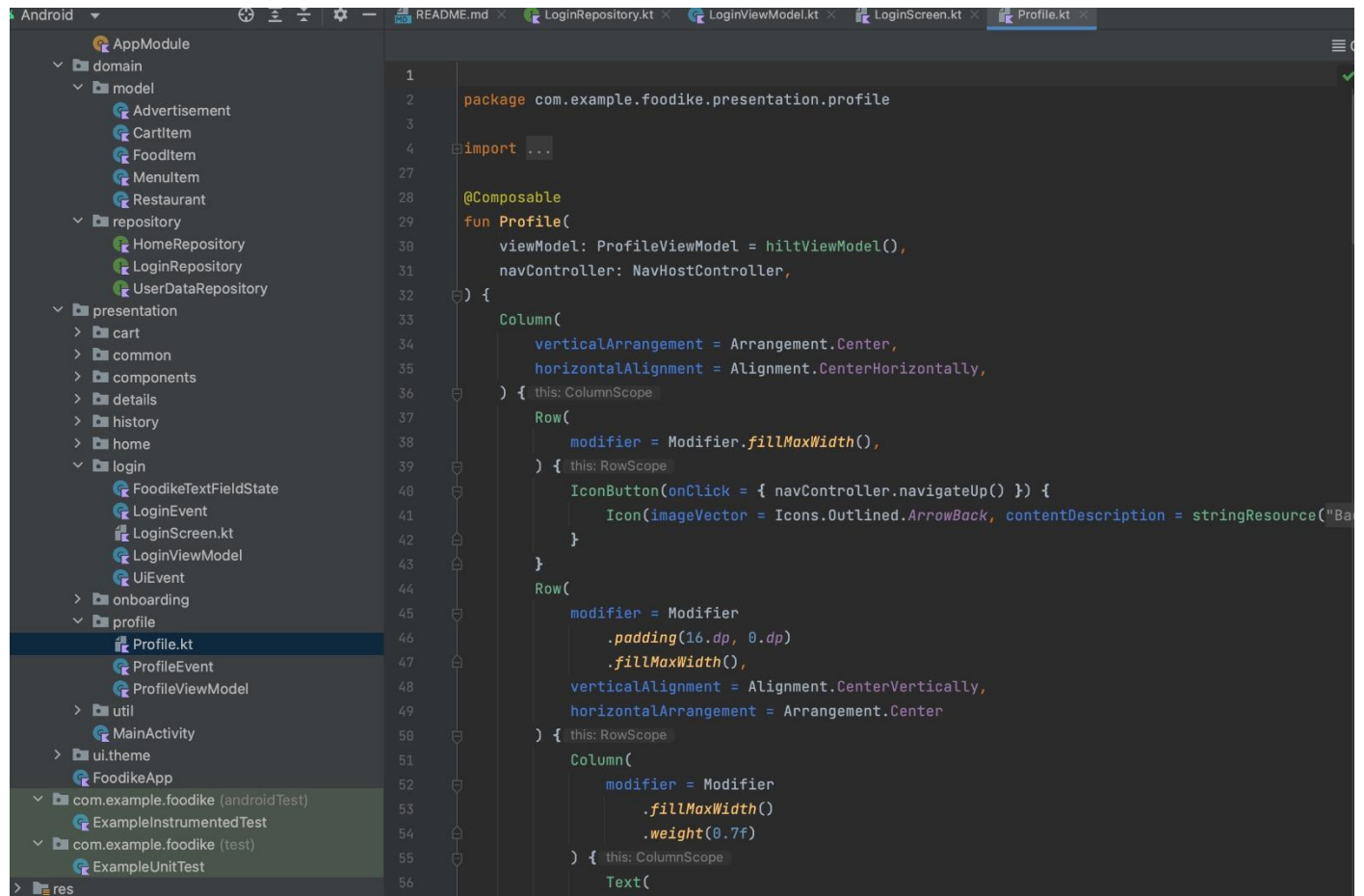
[mofoodz/app/src/main/java/com/example/foodike/presentation/login/LoginScreen.kt](https://github.com/AOkun1205/mofoodz/blob/main/app/src/main/java/com/example/foodike/presentation/login/LoginScreen.kt) at main · AOkun1205/mofoodz · GitHub



```
1  
2  
3 package com.example.foodike.presentation.login  
4  
5 import ...  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37 @SuppressLint("UnusedMaterialScaffoldPaddingParameter")  
38 @Composable  
39 fun LoginScreen(  
40     viewModel: LoginViewModel = hiltViewModel(),  
41     navController: NavHostController,  
42 ) {  
43  
44     val scaffoldState: ScaffoldState = rememberScaffoldState()  
45  
46     val email by viewModel.email  
47     val password by viewModel.password  
48  
49     LaunchedEffect(key1 = true) { this: CoroutineScope  
50         viewModel.eventFlow.collectLatest { event ->  
51             when(event) {  
52                 is UiEvent.ShowSnackbar -> {  
53                     scaffoldState.snackbarHostState.showSnackbar(  
54                         message = event.message,  
55                         duration = SnackbarDuration.Short  
56                     )  
57                 }  
58             }  
59         }  
60     }  
61  
62     Scaffold(scaffoldState = scaffoldState) { it: PaddingValues  
63         Column(  
64             modifier = Modifier.fillMaxSize(),  
65             verticalArrangement = Arrangement.Center,  
66         )  
67     }  
68 }  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100
```


Profile.kt file:

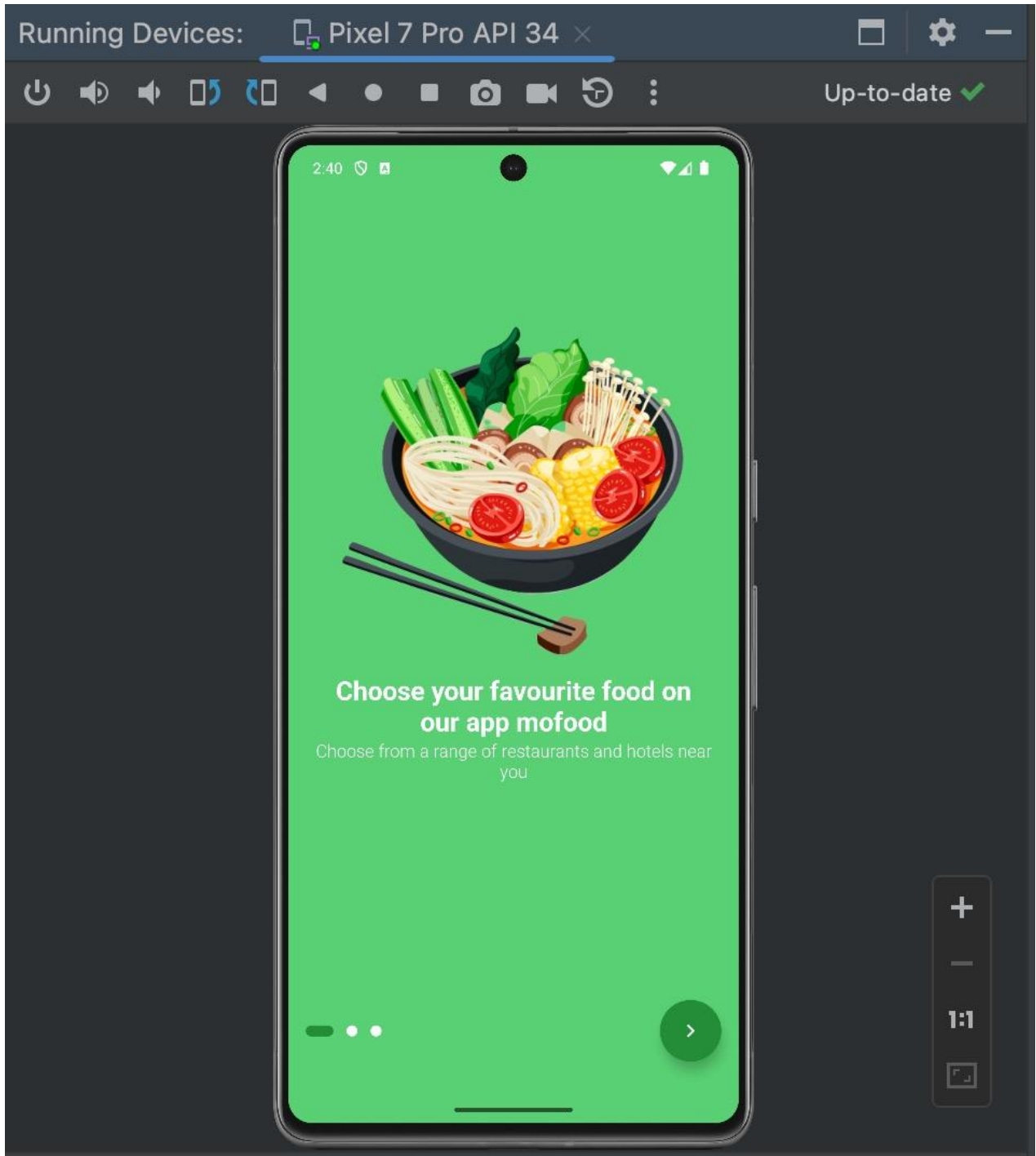
[mofoodz/app/src/main/java/com/example/foodike/presentation/profile/Profile.kt](https://github.com/AOkun1205/mofoodz/blob/main/app/src/main/java/com/example/foodike/presentation/profile/Profile.kt) at main · AOkun1205/mofoodz · GitHub



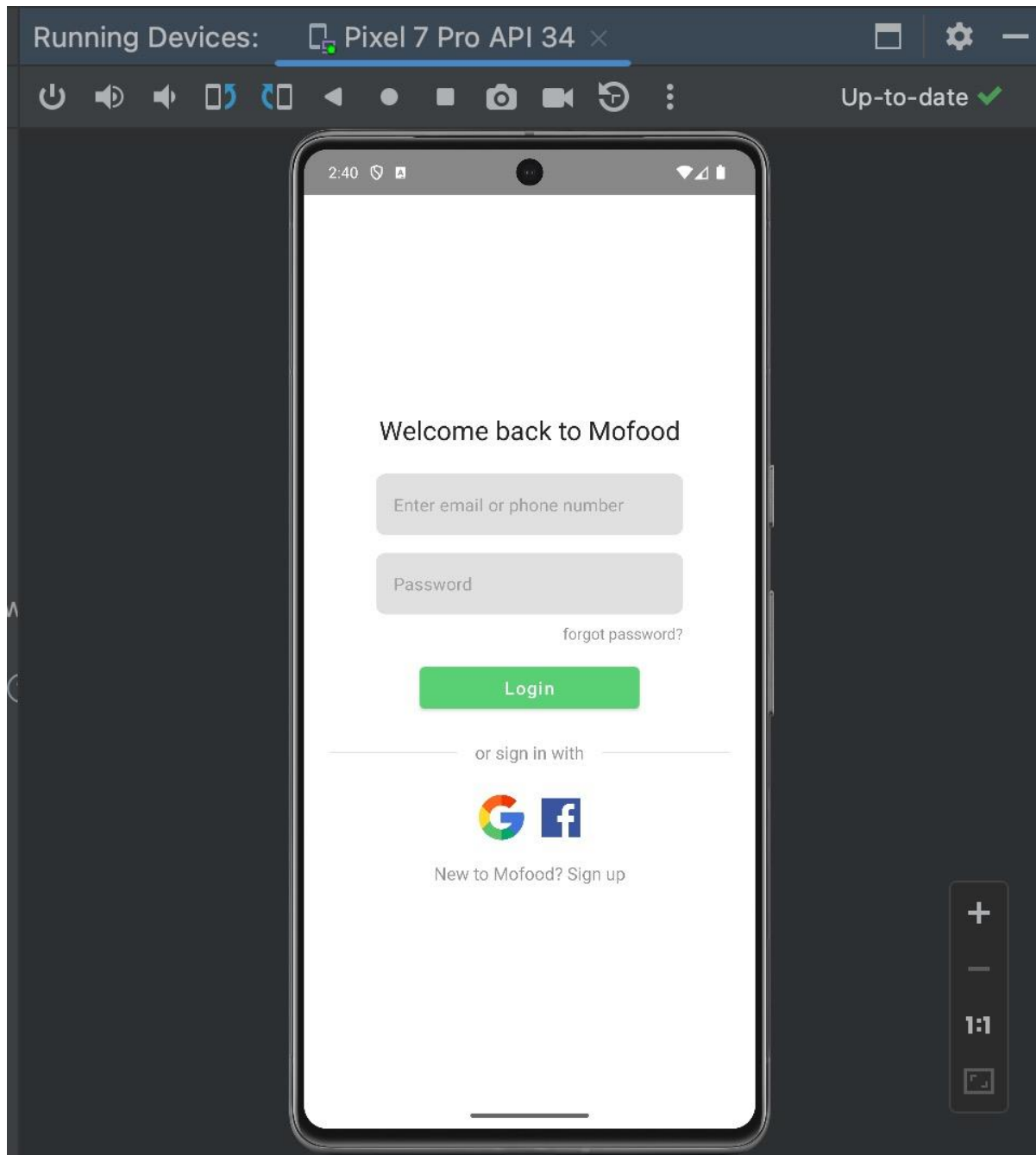
```
1
2 package com.example.foodike.presentation.profile
3
4 import ...
5
27
28 @Composable
29 fun Profile(
30     viewModel: ProfileViewModel = hiltViewModel(),
31     navController: NavHostController,
32 ) {
33     Column(
34         verticalArrangement = Arrangement.Center,
35         horizontalAlignment = Alignment.CenterHorizontally,
36     ) { this: ColumnScope
37         Row(
38             modifier = Modifier.fillMaxWidth(),
39         ) { this: RowScope
40             IconButton(onClick = { navController.navigateUp() }) {
41                 Icon(imageVector = Icons.Outlined.ArrowBack, contentDescription = stringResource("Ba
42             })
43         }
44         Row(
45             modifier = Modifier
46                 .padding(16.dp, 0.dp)
47                 .fillMaxWidth(),
48             verticalAlignment = Alignment.CenterVertically,
49             horizontalArrangement = Arrangement.Center
50         ) { this: RowScope
51             Column(
52                 modifier = Modifier
53                     .fillMaxWidth()
54                     .weight(0.7f)
55             ) { this: ColumnScope
56                 Text(
```

Output Images:

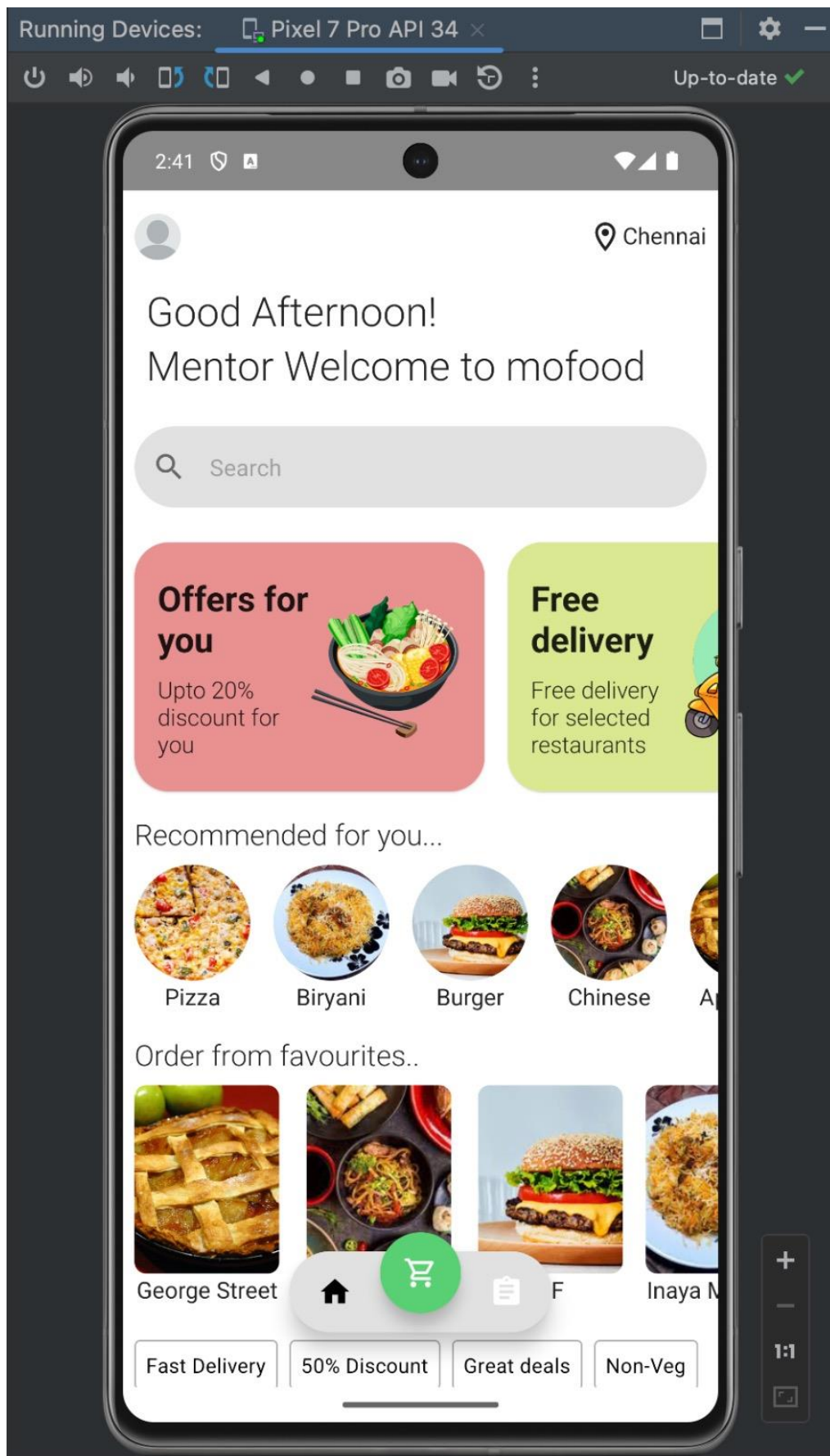
Front Page:



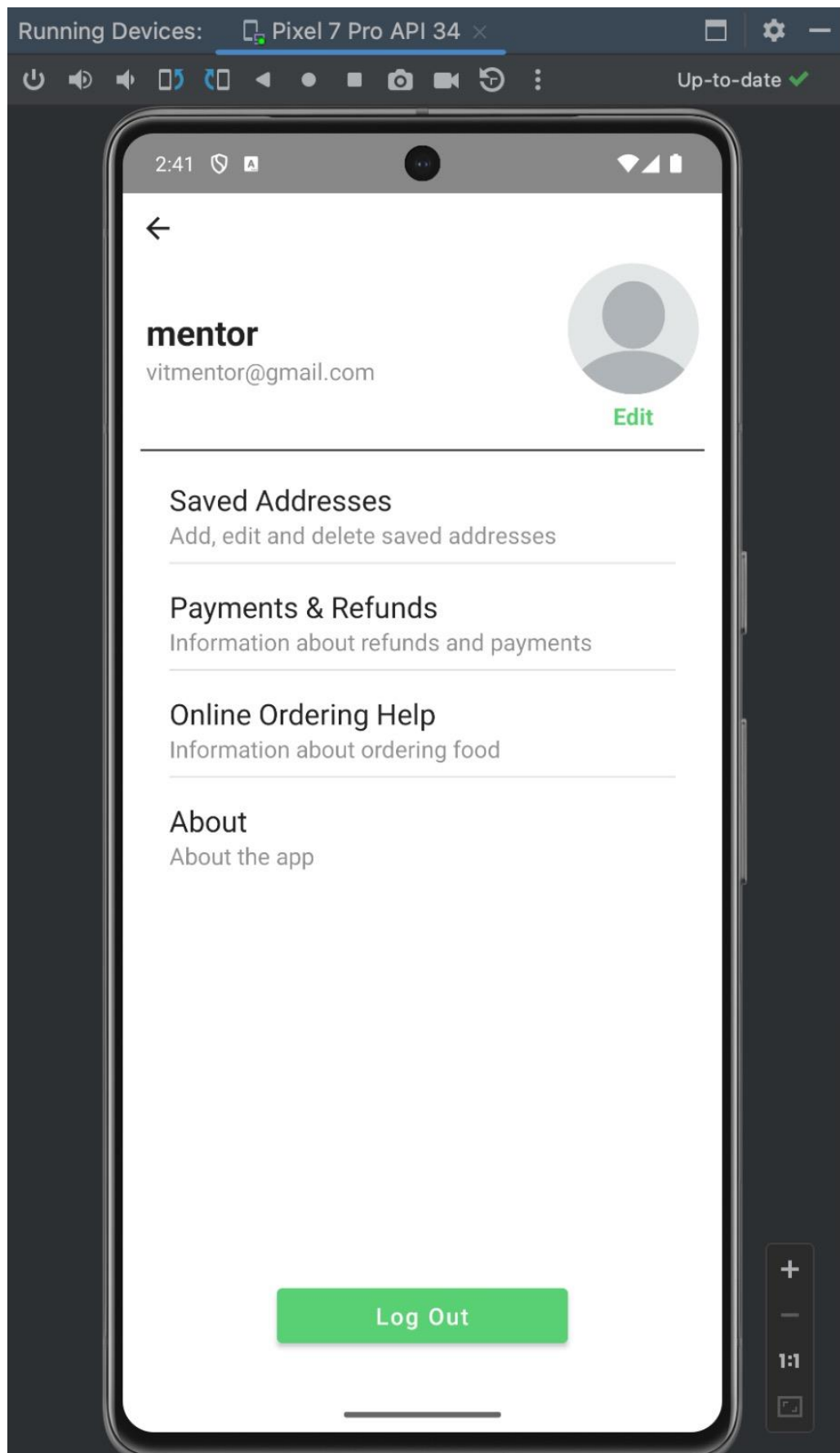
Login Page



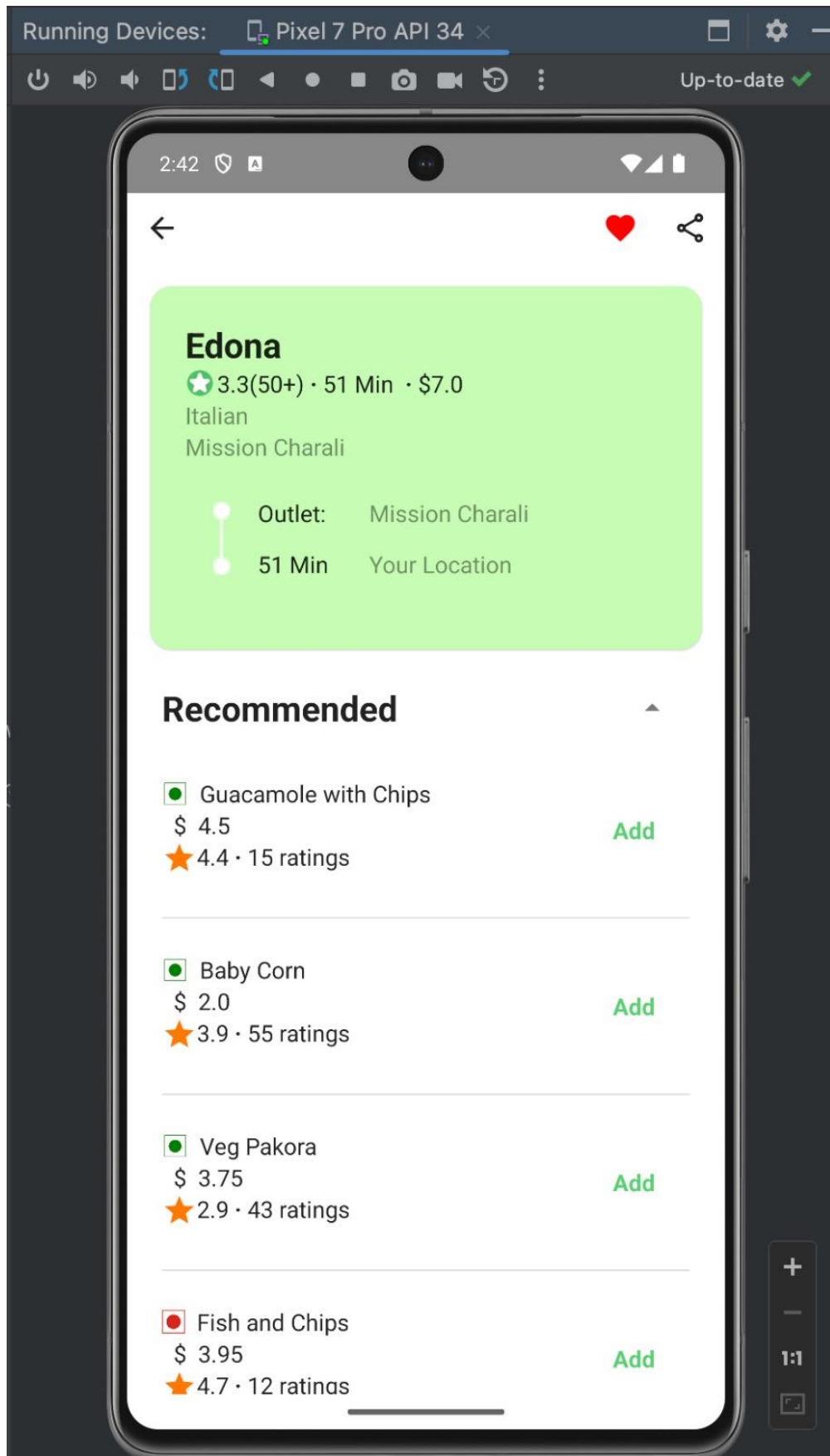
Home Page



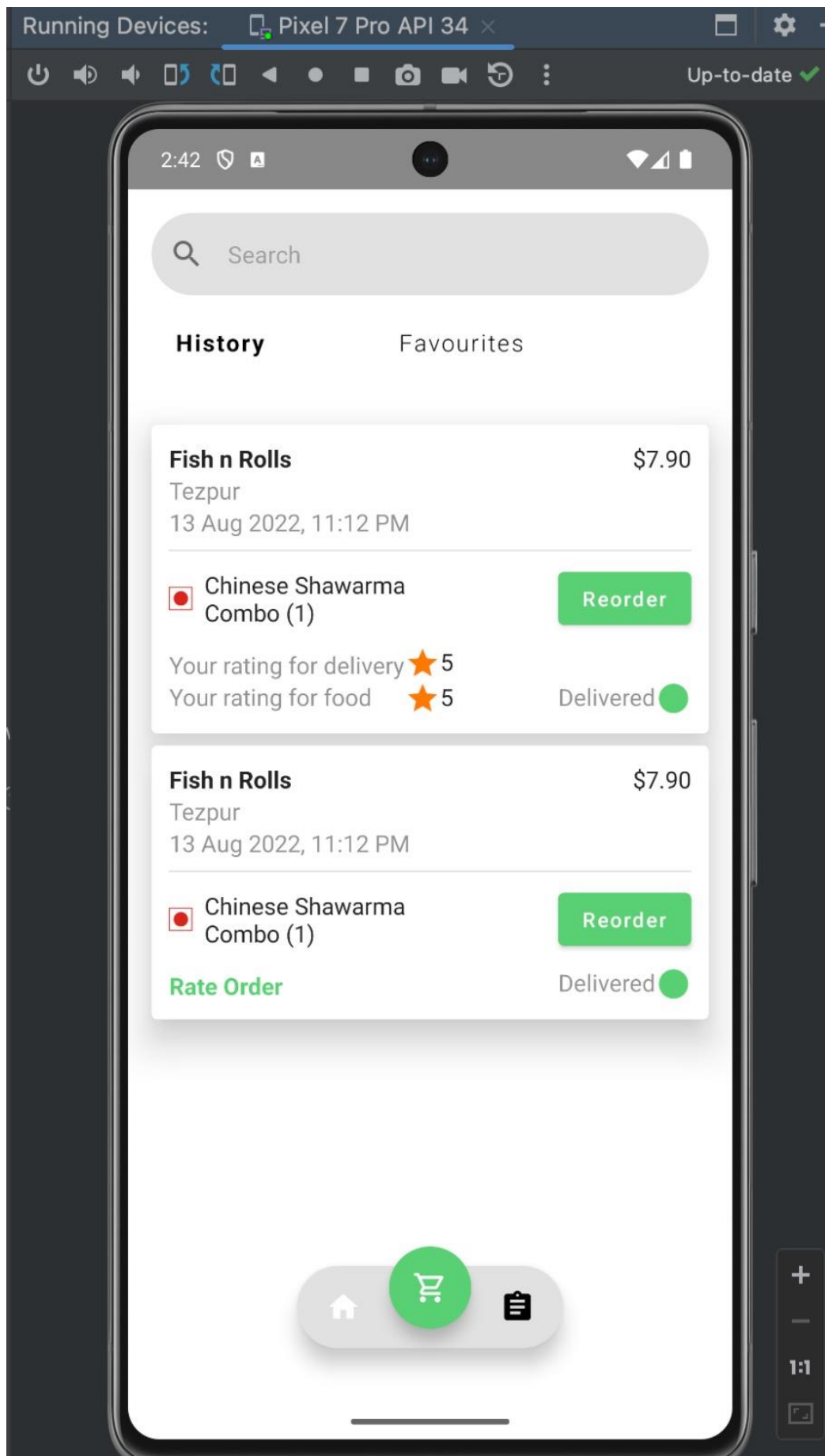
Profile Page



Cart Page



History Page



Payment Page

