2

Building User Screen Flows

Activity 2.02 – creating a login form with registerForActivityResult

Solution

Perform the following steps to solve the problem:

1. Create a new Android Studio project with an empty Activity.
2. Add three `MutableState` properties for a username, a password, and an error message below the `MainActivity` class header:

```
private var username by mutableStateOf("")
private var password by mutableStateOf("")
private var message by mutableStateOf("")
```

3. Create a form with username and password `TextField` Composables and a login button:

```
Scaffold(
    modifier = Modifier
        .fillMaxSize()
) { innerPadding ->
    Box(
        Modifier
            .fillMaxSize()
            .padding(innerPadding),
        contentAlignment = Alignment.Center
    ) {
        Column(
            modifier = Modifier
                .fillMaxSize()
                .padding(16.dp),
            verticalArrangement = Arrangement.Top
        ) {
            TextField(
                value = username,
                onValueChange = { username = it },
                label = { Text("Username") },
                modifier = Modifier.fillMaxWidth()
            )
            Spacer(modifier = Modifier.height(8.dp))
            TextField(
                value = password,
                onValueChange = { password = it },
                label = { Text("Password") },
                modifier = Modifier.fillMaxWidth()
            )
            Spacer(modifier = Modifier.height(16.dp))
            Button(
                onClick = {},
                modifier = Modifier.fillMaxWidth()
            ) {
                Text("Login")
            }
            Spacer(modifier = Modifier.height(8.dp))
            Text(
                text = message,
                modifier = Modifier.padding(16.dp)
            )
        }
    }
}
```

4. Add two constants called `USERNAME_KEY` and `PASSWORD_KEY` above the class header to send the username and password as `Extra` keys in a `Bundle`:

```
const val USERNAME_KEY = "USERNAME_KEY"
const val PASSWORD_KEY = "PASSWORD_KEY"
```

5. Create another constant above `LOGIN_RESULT` to set the data returned from `LoginActivity`:

```
const val LOGIN_RESULT = "LOGIN_RESULT"
```

6. Create a new empty Activity called `LoginActivity` (you can do this by selecting **File** | **New** | **Activity** | **Gallery** | **Empty Activity**) and create two properties called `username` and `password` in `onCreate` to retrieve the `USERNAME_KEY` and `PASSWORD_KEY` values:

```
val username = intent.getStringExtra(USERNAME_KEY)
val password = intent.getStringExtra(PASSWORD_KEY)
```

7. Create a property in `MainActivity` to register for an `Activity` result and set a message after evaluating `LOGIN_RESULT` to welcome the user if validation is successful, and an error message if it is not:

```
private val startForResult:
ActivityResultLauncher<Intent> =
    registerForActivityResult(
      ActivityResultContracts.StartActivityForResult()
    ) { result ->
        val data = result.data
        val loginResult = data?.getBooleanExtra(
            LOGIN_RESULT,
            false
        ) ?: false
        if (loginResult) {
            message = "Welcome, $username!"
        } else {
            message =
                "Login failed. Please try again."
        }
    }
```

8. Add a ClickListener to the login button in MainActivity to validate that the form fields are filled in correctly and start LoginActivity, sending the username and password, and the register for an Activity result property:

```
onClick = {
    if (
        username.isNotEmpty() &&
        password.isNotEmpty()
    ) {
        val intent = Intent(
            this@MainActivity,
            LoginActivity::class.java
        ).apply {
            putExtra(USERNAME_KEY, username)
            putExtra(PASSWORD_KEY, password)
        }
        startForResult.launch(intent)
    } else {
        message = "Please fill in all fields."
    }
},
```

9. Finally, in onCreate in LoginActivity, check the submitted username and password fields against the hardcoded values in LoginActivity and send the result back to MainActivity with LOGIN_RESULT Extra and finish the Activity by calling finish():

```
val loginResult = (
    username == "username" &&
    password == "password"
)

val resultIntent = Intent().apply {
    putExtra(LOGIN_RESULT, loginResult)
}
setResult(RESULT_OK, resultIntent)
finish()
```