

Jetpack Compose ile Android Uygulama Geliştirme Kursu

Kullanıcı Etkileşimi

Kasım ADALAN

Elektronik ve Haberleşme Mühendisi

Android - IOS Developer and Trainer

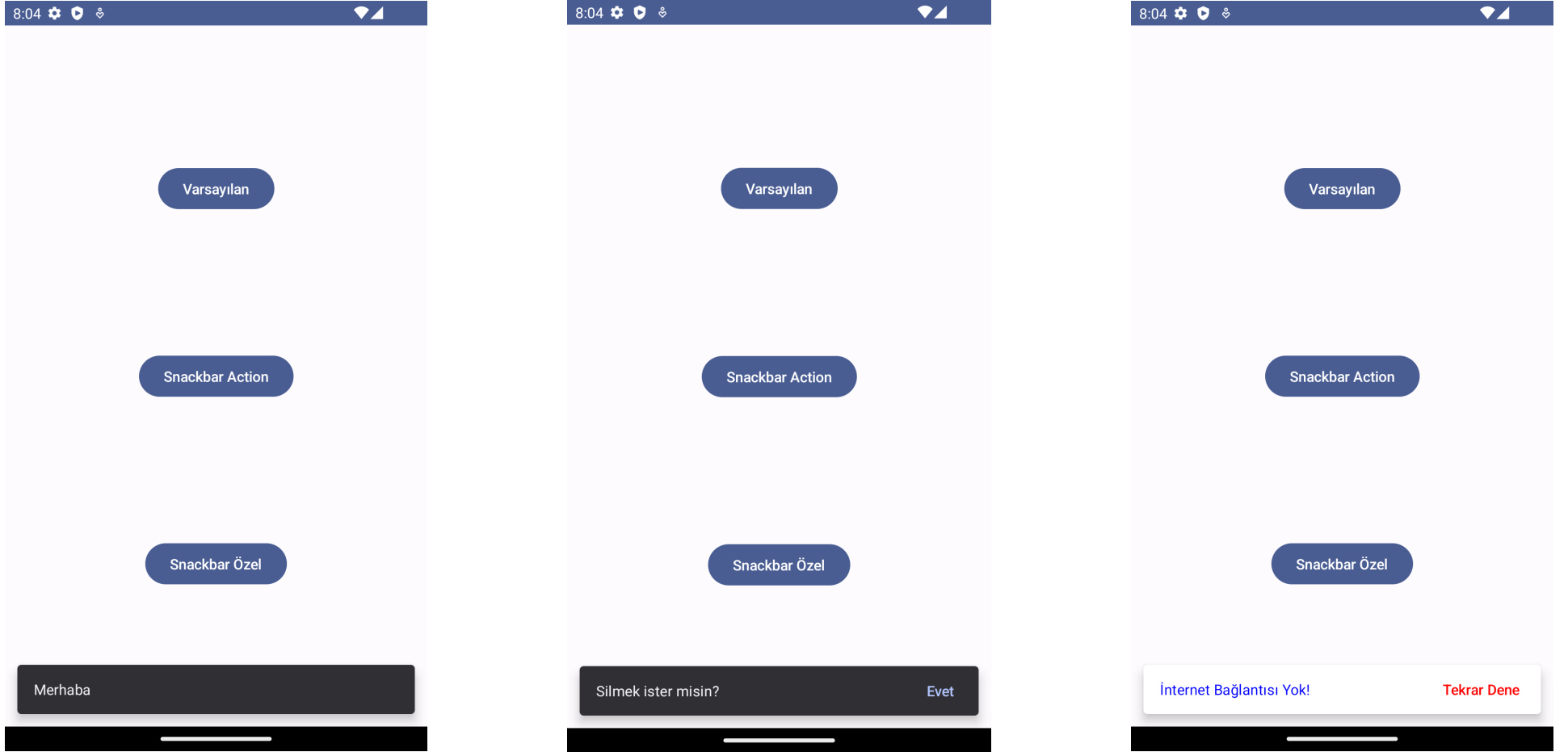
Eğitim İçeriği

- SnackBar
- UIAlertView

SnackBar

Snackbar

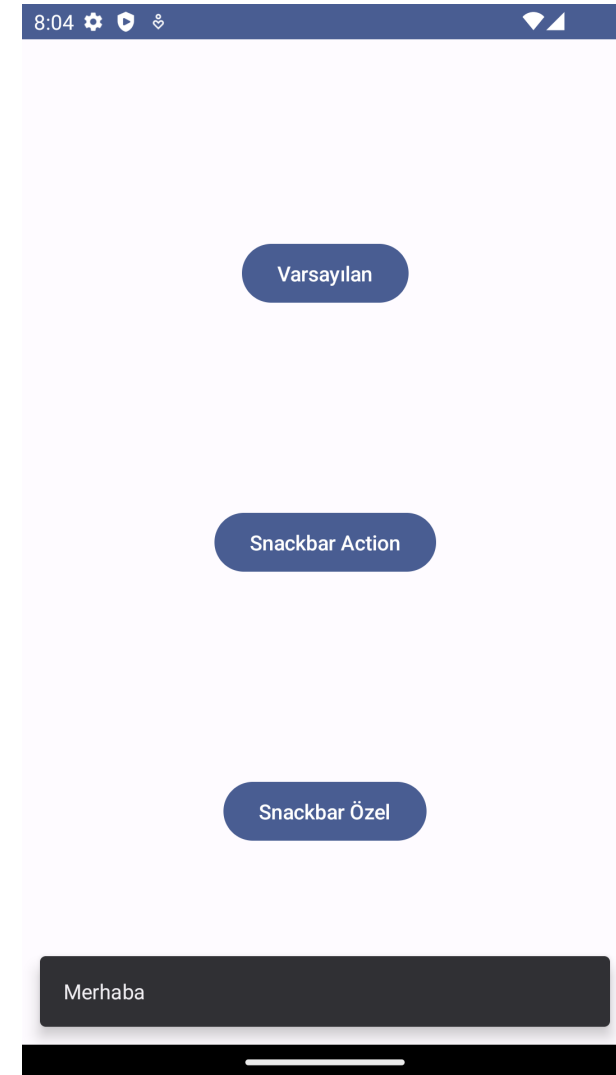
- Snackbar, kullanıcı etkileşimi için oluşturulmuş bir görsel nesnedir.
- Kullanıcıya yaptığı işlemler hakkında geri dönüş vermektedir.



Snackbar Varsayılan

```
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Sayfa() {
    val snackbarHostState = remember { SnackbarHostState() }
    val scope = rememberCoroutineScope()

    Scaffold(
        snackbarHost = {
            SnackbarHost(snackbarHostState)
        },
        content = { it: PaddingValues
            Column(
                modifier = Modifier.fillMaxSize(),
                verticalArrangement = Arrangement.SpaceEvenly,
                horizontalAlignment = Alignment.CenterHorizontally
            ) { this: ColumnScope
                Button(onClick = {
                    scope.launch { this: CoroutineScope
                        snackbarHostState.showSnackbar(message = "Merhaba")
                    }
                }) { this: RowScope
                    Text(text = "Varsayılan")
                }
            }
        }
    )
}
```

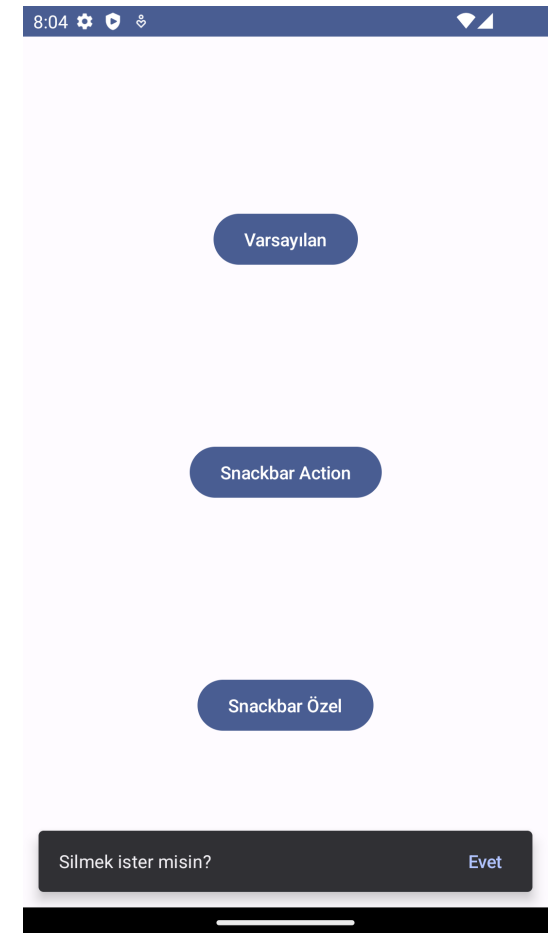


Snack bar Action

SnackBar üzerinde tıklanılma işlemi için button yer alır , bu butona action denir.

```
Button(onClick = {  
    scope.launch { this: CoroutineScope  
        val sb = snackbarHostState  
        .showSnackBar(message = "Silmek ister misin?", actionLabel = "Evet")  
  
        if(sb == SnackbarResult.ActionPerformed){  
            snackbarHostState.showSnackBar(message = "Silindi")  
        }  
    }  
}) { this: RowScope  
    Text(text = "SnackBar Action")  
}
```

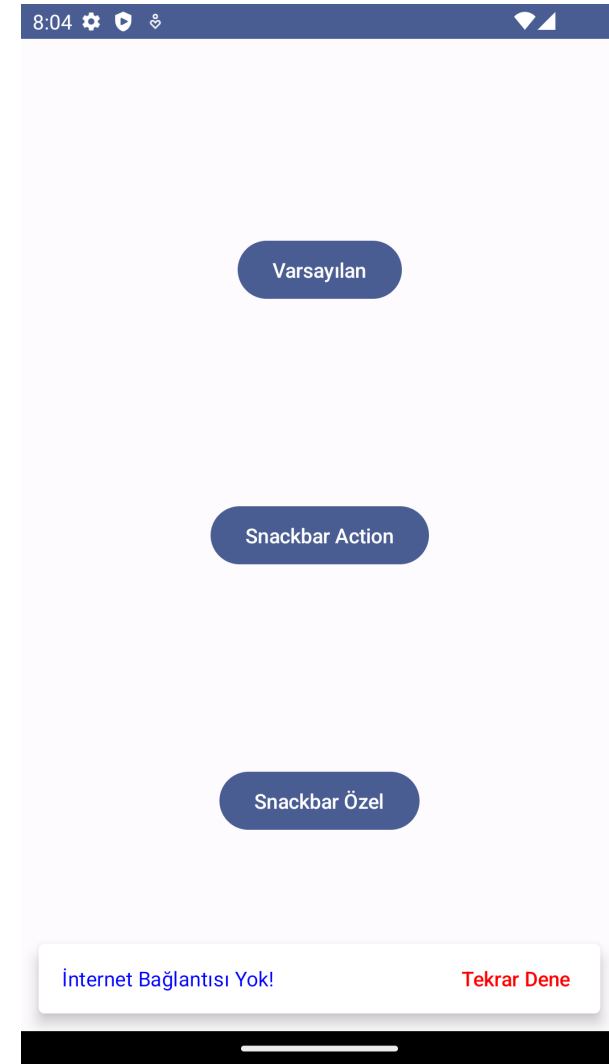
Kasım ADALAN



Özelleştirilmiş Snackbar

```
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Sayfa() {
    val snackbarHostState = remember { SnackbarHostState() }
    val scope = rememberCoroutineScope()
    Scaffold(
        snackbarHost = {
            SnackbarHost(snackbarHostState){ it: SnackbarData
                Snackbar(
                    containerColor = Color.White,
                    contentColor = Color.Blue,
                    actionColor = Color.Red,
                    snackbarData = it
                )
            }
        },
        Button(onClick = {
            scope.launch { this: CoroutineScope
                val sb = snackbarHostState
                .showSnackbar(message = "İnternet Bağlantısı Yok!", actionLabel = "Tekrar Dene")

                if(sb == SnackbarResult.ActionPerformed){
                    snackbarHostState.showSnackbar(message = "Tekrar denendi")
                }
            }
        }) { this: RowScope
            Text(text = "Snackbar Özel")
        }
    }
```



Alert Dialog

Alert Dialog

- AlertDialog, kullanıcıya uyarı mesajı verebildiğimiz bir yapıdır.
- Kullanıcıdan alert yardımıyla girdide alabiliriz.
- Kullanıcı alertı kapatmadığı sürece ekranda durur.



Kasım ADALAN



Basit Alert

```
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Sayfa() {
    val acilisKontrolVarsayilan = remember { mutableStateOf( value: false) }
    Column(
        modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.SpaceEvenly,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        this: ColumnScope
        Button(onClick = { acilisKontrolVarsayilan.value = true }) { Text(text = "Varsayilan") }

        if (acilisKontrolVarsayilan.value) {
            AlertDialog(
                onDismissRequest = { acilisKontrolVarsayilan.value = false },//Boşluğu seçip kapatma
                title = { Text(text = "Başlık") },//Başlık
                text = { Text(text = "Mesaj") },//İçerik
                confirmButton = {
                    Text(text = "Tamam",color = Color.Blue,fontWeight = FontWeight.Bold,
                        modifier = Modifier.padding(10.dp).clickable {
                            acilisKontrolVarsayilan.value = false
                            Log.e( tag: "Tamam", msg: "Seçildi")
                        })
                },
                dismissButton = {
                    Text(text = "İptal",color = Color.Blue,fontWeight = FontWeight.Bold,
                        modifier = Modifier.padding(10.dp).clickable {
                            acilisKontrolVarsayilan.value = false
                            Log.e( tag: "İptal", msg: "Seçildi")
                        })
                })
        }
    }
}
```



Özelleştirilmiş Alert

```
@Composable
fun Sayfa() {
    val acilisKontrolVarsayilan = remember { mutableStateOf( value: false) }
    val acilisKontrolOzel = remember { mutableStateOf( value: false) }
    val tf = remember { mutableStateOf( value: "") }
    Column(
        modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.SpaceEvenly,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        this: ColumnScope
        Button(onClick = { acilisKontrolVarsayilan.value = true }) { Text(text = "Varsayılan") }
        Button(onClick = { acilisKontrolOzel.value = true }) { Text(text = "Özel") }
    }
}
```



```
if (acilisKontrolOzel.value) {
    AlertDialog(
        onDismissRequest = { acilisKontrolOzel.value = false }, //Boşluğu seçip kapatma
        title = { Text(text = "Başlık", color = Color.White) }, //Başlık
        text = { TextField(
            value = tf.value,
            onValueChange = {tf.value = it},
            label = { Text(text = "Veri")} )
        }, //İçerik
        confirmButton = {
            Text(text = "Tamam", color = Color.Red, fontWeight = FontWeight.Bold,
                modifier = Modifier.padding(10.dp).clickable {
                    acilisKontrolOzel.value = false
                    Log.e( tag: "Tamam", msg: "Seçildi : ${tf.value}")
                    tf.value = "" //Textfield içini sıfırlama
                })
        },
        dismissButton = {
            Text(text = "İptal", color = Color.Red, fontWeight = FontWeight.Bold,
                modifier = Modifier.padding(10.dp).clickable {
                    acilisKontrolOzel.value = false
                    Log.e( tag: "İptal", msg: "Seçildi")
                    tf.value = ""
                })
        },
        containerColor = Color.LightGray
    )
}
```

Teşekkürler...



kasım-adalan



kasimadalan@gmail.com



kasimadalan