

Teaching Android Jetpack Compose UI Framework

Studiengang: BSc in Informatik | Vertiefung: Distributed Systems and IoT
Betreuer: Prof. Dr. Ulrich Fiedler

Jetpack Compose ist das neue empfohlene Toolkit von Android zur Implementierung von Benutzeroberflächen. An der BFH wird noch immer die Vorgänger Technologie „XML Views“ unterrichtet. Dies möchte ich mit dieser Thesis ändern.

Jetpack Compose

Jetpack Compose ist das von Android empfohlene moderne Toolkit zum Erstellen nativer Benutzeroberflächen. Es vereinfacht und beschleunigt die UI-Entwicklung auf Android. Jetpack Compose ist ein deklaratives UI Framework welches ähnlich funktioniert wie z.B. Flutter, Swift UI oder React

Vorgehen

Zur Veranschaulichung haben wir eine Beispiel-App entwickelt, die Jetpack Compose verwendet. Im Projekt 2 „Smart management of tea/coffee capsules“ wurde bereits ein Proof of Concept entwickelt, den wir weiter verbessert und für Lehrzwecke aufbereitet haben. Wir haben Unterrichtsmaterialien wie z.B. Folien und Dokumentation erstellt, die die Kernkonzepte von Jetpack Compose veranschaulichen. Um zu testen, ob sich unser Unterrichtskonzept bewährt, haben wir einen Feldtest mit einer Testgruppe durchgeführt. Nach dem Testlauf haben wir Feedback

gesammelt und anhand diesem das Unterrichtsmaterial überarbeitet.

Feedback aus dem Feldtest

Im Feldtest habe ich wertvolles Feedback erhalten, mit welchem ich das Unterrichtsmaterial verbessern konnte. Dies konnte grössten Teils noch umgesetzt werden. So wurde eine Folie eingebaut die den direkten Vergleich zwischen Jetpack Compose und dem Ansatz via XML Views aufzeigt.



Matthias Ossola

Ergebnisse

Wir haben festgestellt, dass unser Unterrichtskonzept bei der Testgruppe sehr gut angekommen ist. Besonders das Code-Together war für die Teilnehmer sehr interessant, da so der Unterrichtsstoff interaktiv gelernt werden konnte. Ulrich Fiedler wird diese Unterrichtssequenz im kommenden Semester im Unterricht im Modul Mobile Systems 1 BTI4101 verwenden.

Comparison to XML Views

Jetpack Compose

```
class MainActivity : AppCompatActivity() {
    private lateinit var listItems: List<String>

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        listItems = resources.getStringArray(R.array.cities).toList()
        setContentView {
            JetpackComposeExampleTheme {
                Surface {
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                } {
                    ListView(listItems, applicationContext)
                }
            }
        }
    }
}
```

XML View

```
class MainActivity : AppCompatActivity() {
    private lateinit var listView: ListView
    private lateinit var listItems: Array<String>

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        listView = findViewById<View>(R.id.listView) as ListView
        listItems = resources.getStringArray(R.array.cities)
        val adapter = ArrayAdapter(
            this,
            R.layout.list_item, R.id.textView, listItems
        )
        listView.adapter = adapter
        listView.setOnItemClickListener { adapterView, view,
            position, l ->
                val value = adapter.getItem(position)
                Toast.makeText(applicationContext, value,
                    Toast.LENGTH_SHORT).show()
            }
        }
    }
}
```