

# Kotlin Multiplattform Mobile: Evaluation für HSLU-Apps

<b>Themenbereiche:</b>	Software-Erstellung, mobile Applikationen (Android/iOS), Kotlin Multiplattform Mobile, SwiftUI, Jetpack Compose
<b>Studierender:</b>	Valentin Bürgler
<b>Betreuungsperson:</b>	Prof. Dr. Ruedi Arnold
<b>Experte:</b>	Timo Furrer
<b>Auftraggeber:</b>	Hochschule Luzern - Informatik, Jürg Nietlispach
<b>Keywords:</b>	Mobile Applikationen, plattformübergreifende App-Entwicklung, Kotlin Multiplattform Mobile

## 1. Aufgabenstellung

Das Departement Informatik der Hochschule Luzern (HSLU) entwickelt und betreibt mobile Apps für die beiden Departemente Informatik (HSLU I) und Technik & Architektur (HSLU T&A), jeweils für die zwei Plattformen iOS und Android. Die Apps wurden im Frühlingsemester (FS) 2022 mit den jeweiligen aktuellen deklarativen GUI-Technologien SwiftUI und Jetpack Compose auf den neusten Stand der Technik gebracht. Dabei wurde auf beiden Plattformen eine modularisierte, konfigurierbare und mandantenfähige Architektur entwickelt, um aus jeweils einer Code-Basis die Apps für beide Departemente HSLU I und HSLU T&A erstellen zu können.

Vor diesem Hintergrund soll die Technologie Kotlin Multiplattform Mobile (KMM) evaluiert werden. Im Rahmen dieser Bachelorarbeit soll das Potential von KMM für die Weiterentwicklung der mobilen Apps der HSLU abgeschätzt und mit Prototypen illustriert werden. Die Vor- und Nachteile gegenüber der aktuellen Lösung sind zu evaluieren, und es soll gezeigt werden, ob es sich lohnen würde, die HSLU-Apps mit KMM zu implementieren.

## 2. Lösungskonzept

Das Projekt wurde in einer agilen, iterativen und inkrementellen Vorgehensweise durchgeführt. In regelmässigen, meist zweiwöchigen Iterationen wurden Betreuer und Auftraggeber über den aktuellen Stand des Projektes informiert, und das weitere Vorgehen wurde besprochen.

Zunächst wurden die zu evaluierenden, relevanten Aspekte anhand der HSLU-Apps analysiert. Hierfür dienten die Code-Basen beider Apps und die Berichte der Bachelorarbeiten aus dem FS 2022 als Ausgangslage. Es wurde identifiziert, welche Aspekte in einem Prototypen illustriert werden sollten. Daraufhin fand eine Einarbeitungsphase in die Technologien KMM, Jetpack Compose und SwiftUI statt, dies mithilfe der offiziellen Tutorials, sowie Anwendungsbeispielen aus der KMM-Community. Mit den so gewonnenen Erkenntnissen wurde schliesslich eine neue, auf KMM basierende, prototypische Anwendung implementiert, welche im Verlauf des Projektes inkrementell um die in den HSLU-Apps identifizierten Aspekte erweitert wurde.

### 3. Spezielle Herausforderungen

KMM befindet sich seit dem 10.10.2022 in der Beta-Release-Phase und ist eine entsprechend junge Technologie. Noch sind nicht alle Aspekte produktionsreif. Der erste Alpha-Release von KMM war am 31.08.2020 und es hat sich seitdem bereits viel wieder geändert. Lösungsvarianten und Anwendungsbeispiele der Community sind mit Vorsicht zu geniessen, da sie zum Teil auf bereits überholten Sachverhalten beruhen und alte Workarounds enthalten, die nicht mehr nötig sind.

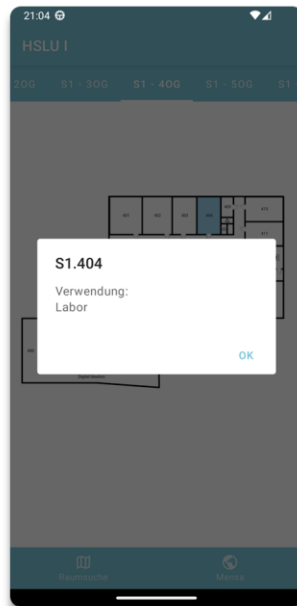
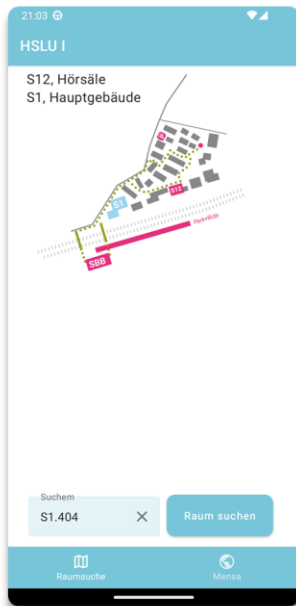
KMM liefert für viele Probleme von Haus aus keine Antworten, sondern stützt sich auf eine Vielzahl von Plugins und Bibliotheken. Je weiter sich ein Problem vom Standard-CRUD-Benutzerfall entfernt, umso unreifer werden die dafür von der Community empfohlenen Plugins und Bibliotheken. Dies führte während des Projektes mehrmals zu Kompatibilitätsproblemen unter den Abhängigkeiten.

### 4. Ergebnisse

Es wurde ein Prototyp in Form einer KMM-Anwendung implementiert, welcher aufzeigt, wie die mobilen HSLU-Apps in Zukunft aus einer einzigen Code-Basis aufgebaut sein könnten. Die daraus generierten iOS- und Android-Apps sind, wie bereits die Apps aus dem FS 2022, anhand des Entwurfsmusters Model-View-ViewModel (MVVM) und mit deklarativer GUI-Technologie implementiert. Anders als bisher teilen sie sich sämtliche gemeinsam genutzte Funktionalität aus dem Shared-Modul: Von der Mandantenkonfiguration (Funktionalitäten, Farbschema und Ressourcen), über Netzwerk, Serialisierung, Persistenz, Dependency-Injection, den Models, bis hin zu den ViewModels wird dieselbe Code-Basis verwendet.

Umgesetzte Anforderungen:

- Neue Implementation auf Basis KMM mit gemeinsamer Funktionalität im Shared-Modul für
  - Kern-Pakete (Netzwerk, Serialisierung, Persistenz, Dependency-Injection)
  - Feature-Pakete (View-Models, Models)
  - Gemeinsame Ressourcendateien (Logo, Farben, Texte [Deutsch und Englisch])
  - Mandantenfähigkeit
    - Mit dem Build-Tool Gradle konfigurierbare Mandanten zur Build-Zeit
    - Konfiguration der Komponenten anhand der Mandantenkonfiguration im Dependency-Injection-Container
    - Komponente zur mandantenbasierten Lokalisierung der benötigten Ressourcen
    - Konfigurierbare Backend-Links
  - Modularisierung
    - Module gemäss der von KMM verwendeten Projektstruktur
    - Software-Pakete, die sich an der Code-Basis von FS 2022 orientieren
    - Konfiguration des Feature-Umfangs anhand der Mandantenkonfiguration im Dependency-Injection-Container
- Integration der Raumsuche für HSLU I in iOS und Android
- Prototypische Integration der Raumsuche für HSLU T&A in iOS und Android, mit hart-codierten Quelldateien für die Übersicht, den Trakt „Innenarchitektur“ und dessen einziges Geschoss
- Integration des Parkplatzzählers für den Mandanten HSLU T&A in Android
- Navigation basierend auf den für den Mandanten konfigurierten Funktionalitäten in Android
- Darkmodes in iOS und Android
- Für die Studierenden des Moduls MOBLAB wurde eine Präsentation über KMM gehalten



Abbildungen 1 & 2: Android-App - HSLU Informatik

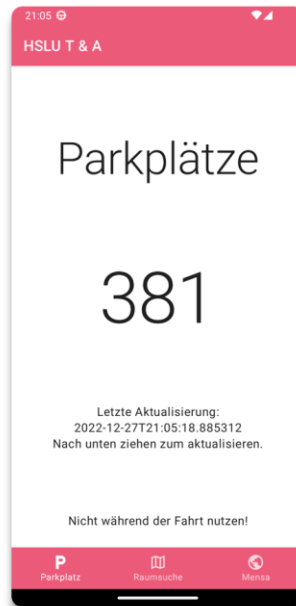


Abbildung 3: Android-App - HSLU Technik & Architektur

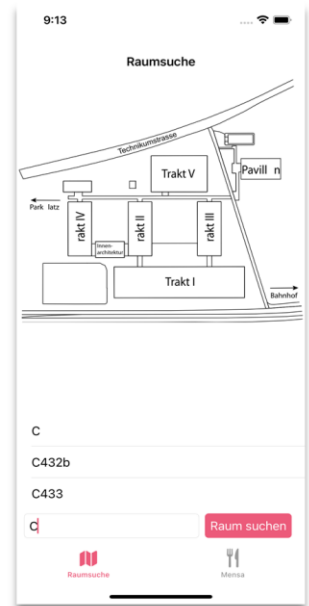


Abbildung 4: iOS-App - HSLU Technik & Architektur

Die Fertigstellung der Raumpläne für den Campus der HSLU T&A wurde geprüft, konnte aber nicht mehr innerhalb dieses Projektes umgesetzt werden. Eine Integration in die bestehende CI / CD Infrastruktur mit dem Build-Tool Fastlane konnte ebenfalls nur angedacht, aber nicht mehr umgesetzt werden.

Im Bericht wurde eine Zusammenstellung der für die HSLU-Apps wesentlichen Aspekte von KMM geliefert. Es wurde aufgezeigt, wo noch weitere Recherchen nötig sein werden und welche Probleme noch nicht abschliessend gelöst werden konnten. Die Vor- und Nachteile der Technologie wurden dem aktuellen Entwicklungsansatz anhand der erfolgten Recherche und der gemachten Erfahrungen bei der Entwicklung des Prototypen gegenübergestellt. Zuletzt konnte eine begründete Empfehlung für eine Umstellung auf KMM abgegeben werden.

## 5. Ausblick

KMM steht laut offizieller Aussage von JetBrains seit Oktober 2022 kurz vor der Produktionsreife. In vieler Hinsicht konnte das im Verlauf dieser Arbeit grundsätzlich bestätigt werden.

Die Mandantenfähigkeit stellt einen Kernaspekt der HSLU-Apps dar. Dieser Aspekt liess sich innerhalb der für diese Arbeit zur Verfügung gestandenen Zeit nur unbefriedigend mit KMM im Prototypen illustrieren. Die umgesetzten Lösungen sind nicht wirklich nachhaltig und sie decken nicht alles ab, was der bisherige, plattformspezifische Ansatz zu bieten hat. Es würde sich lohnen, mehr Zeit und Aufmerksamkeit in die Lösung dieses Aspekts zu investieren.

Im FS 2022 wurden die Android-App und die iOS-App mandantenfähig. Die Apps für die jeweiligen Mandanten stammen seither aus der gleichen Code-Basis. Dass nun auch die Code-Basen beider Plattformen zusammengeführt wird, wäre der nächste logische Schritt für die HSLU-Apps. Mit einer nachhaltigen Lösung für die Mandantenfähigkeit in KMM wäre ein Wechsel langfristig nur zu empfehlen.