

# Privileges.App Ein Privilegienspiel

Abschlussarbeit im Rahmen des Moduls:  
Android App Entwicklung mit Kotlin (M3)

Betreut durch den Dozenten: Till Hemmerich.

Und den Tutor\*innen: Ahmad Alkhalaf, Jana Jansen, Justus Finke,  
Lukas Altmann, Noah Diemel, Pavel Keßler, Talha Taskaya und Enie dem Hund.

In dem Kurs "**Android App Entwicklung mit Kotlin**" haben wir die Programmiersprache Kotlin anhand Themen und Konzepten wie bspw. RecyclerView, Navigation und Architektur, API und Datenbank erlernt.

Neben täglichen Vorlesungen am Vormittag, waren die Nachmittage mit Praxisaufgaben gefüllt. Am Ende des knapp 15-wöchigen Moduls stand die Entwicklung einer eigenen App.

#### WOCHE 1:

Tag 01: Einstieg in die Android App Entwicklung.  
Tag 02: Einführung in das Integrated Development Environment für die Android App Entwicklung.  
Tag 03: User Input  
Tag 04: Praxisaufgabe Layouts.

#### WOCHE 2:

Tag 01: Themes  
Tag 02: ViewBinding: Einführung in das Konzept ViewBinding.  
Tag 03: Lifecycle: Verständnis des Lebenszyklus einer Android App.  
Tag 04: Praxis-Freitag

#### WOCHE 3:

Tag 01: RecyclerView-Einstieg.  
Tag 02: Einführung in LayoutManager für effiziente Layouts in RecyclerView.  
Tag 03: Erstellung einer eigenen RecyclerView zur Darstellung von Daten. Tag 04: RecyclerView mit verschiedenen Item-Layouts  
Tag 05: Praxisfreitag.

#### WOCHE 4:

Tag 01: Verständnis der Verwendung von Activities und Intents in Android Apps.  
Tag 02: Einführung in die Verwendung von Fragmenten und Navigation in Android Apps. Tag 03: Wiederholung der Konzepte zur Navigation.  
Tag 04: Einführung in die Bottom Navigation als Teil der Benutzeroberfläche.  
Tag 05: Praxisfreitag.

#### WOCHE 5:

Tag 01: RecyclerView.  
Tag 02: Layouts

#### WOCHE 6:

Tag 01: MVVM: Einführung in das Model-View-ViewModel (MVVM) Architekturmuster für Android Apps.  
Tag 02: LiveData: Verständnis der Verwendung von LiveData für reaktive App-Entwicklung.  
Tag 03: SharedViewModel: Anwendung von SharedViewModel zur gemeinsamen Nutzung von Daten zwischen Fragmenten.  
Tag 04: MVVM Wiederholung.  
Tag 05: Praxisfreitag.

#### WOCHE 7:

Tag 01: Coroutines: Einführung in Coroutines für asynchrone Programmierung in Kotlin.  
Tag 02: Verständnis und Umsetzung von API-Aufrufen.  
Tag 03: API Calls und Bilder.  
Tag 04: Praxisfreitag.

#### WOCHE 9:

Tag 01: Einführung in die Grundlagen von SQL.  
Tag 02: Verständnis und Anwendung der Room Persistence Library.  
Tag 03: Einführung in das Repository Design Pattern für eine saubere Datenarchitektur.  
Tag 04: Erstellung eines Pflichtenhefts und Einrichtung eines GitHub Repositories für unsere Modul-Abschlussarbeiten.

# Privileges.App

## Über meine Abschlussarbeit

Privileges.App ist eine Weiterentwicklung meines terminal-basierten Privilegienspiels, das ich in Modul 2 (Grundlagen der Programmierung) entwickelt hatte ([Github-Link](#)).

Privilegienspiele sind eine interaktive Methode, um den Teilnehmenden das Konzept von Privilegien bewusst zu machen und sie dazu anzuregen, über soziale Ungleichheiten und ihre eigenen Vorteile nachzudenken.

## STYLEGUIDE & WIREFRAMES

Die Farben wurden größtenteils aus der ursprünglichen terminalbasierten Version übernommen. Hinzugekommen sind Farben für die jeweiligen Fortschrittsbalken der Spieler\*innen.



# Game Installation

## Game Flow

Die aktuellste Version des Privilegienspiels ist hier verlinkt: [www.privileges.app](http://www.privileges.app). Zur Zeit gibt es die App im Google-Playstore sowie als .APK zum Direktdownload.

In der App startest du als eine\*r von mehreren Spieler\*innen, und dein Ziel ist es, durch eine Reihe von Fragen und Fakten zu navigieren. Der Ablauf des Spiels ist folgendermaßen:

1. Intro und Onboarding: Alle Spieler\*innen geben ihre Namen ein, und die App speichert diese in der lokalen Datenbank. Der Plus-Button fügt eine\*n neue\*n Spieler\*in hinzu. Der Weiter-Button startet das Spiel.
2. Fragen: Die App stellt abwechselnd allen Spieler\*innen eine Frage bzw. konfrontiert sie mit einer Aussage. Du hast die Wahl, mit "Ja" oder "Nein" zu antworten. Deine Antworten beeinflussen deinen Fortschritt im Spiel, der mit der..
3. Fortschrittsanzeige angezeigt wird.
4. Nach jeder Fragerunde erscheint ein Fakt zu der jeweiligen Aussage.
5. Spielende: Wenn alle Fragen beantwortet sind, endet das Spiel. Du hast dann die Möglichkeit, ein Zertifikat mit deinem Fortschritt herunterzuladen oder göeich aus der App zu teilen.

# Privileges.App Funktionen & Code-Snippets

Spielerdaten-Verwaltung mit Room: Die App nutzt eine lokale **Room-Datenbank** für das Management von Spielerdaten wie ID, Namen und Fortschritt. Der Datensatz der verwendeten Fragen wird hingegen extern verwaltet und mittels API-Call geladen. Dies geschieht beim **Abruf eines PHP-Scripts**, das die Felder einer **MySQL-Tabelle als JSON übergibt**.

Die App verwendet **Coroutines**, um **asynchrone Aufgaben zu handeln**. Wenn z.B. weniger als 8 menschliche Spieler\*innen vorhanden sind, fügt die App automatisch NPCs hinzu, um die Runde zu vervollständigen. Dies geschieht im Hintergrund mit Coroutines. Zusätzlich aktualisiere ich die Benutzeroberfläche mit Fortschrittsbalken, die den Fortschritt jedes Spielers anzeigen. Auch hier kommen Coroutines ins Spiel, diesmal auf dem Dispatchers.Main Thread, um die UI reibungslos zu aktualisieren.

```
Just161 *
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = OnboardingNameBinding.inflate(layoutInflater)
    setContentView(binding.root)

    db = DatabaseProvider.getInstance(applicationContext)

    lifecycleScope.launch(Dispatchers.IO) { this: CoroutineScope
        db.playerDao().nukeTable()
    }

    binding.weiterButton.setOnClickListener { it: View?
        val name = binding.username.text.toString()
        if (name.isNotBlank()) {
            lifecycleScope.launch(Dispatchers.IO) { this: CoroutineScope
                val newPlayer = Player(name = name, isHuman = true)
                val newId = db.playerDao().insertPlayer(newPlayer)
                Log.d("tag: #####", msg: "Neue Spieler-ID: $newId, Name: $name")
            }
        }
    }
}
```

```
private suspend fun loadFragenFakten() {
    withContext(Dispatchers.IO) { this: CoroutineScope
        val url = URL("https://privileges.app/fragenFakten_privileges.php")
        with(url.openConnection() as HttpURLConnection) { this: HttpURLConnection
            try {
                val text = inputStream.bufferedReader().use(BufferedReader::readText)
                val json = JSONArray(text)

                withContext(Dispatchers.Main) { this: CoroutineScope
                    for (i in 0 until json.length()) {
                        val item = json.getJSONObject(i)
                        if (item.has("frage") && item.has("fakt")) {
                            val frage = item.getString("frage")
                            val fakt = item.getString("fakt")
                            fragenFaktenDEU.add(Pair(frage, fakt))
                        }
                    }

                    fragenFaktenDEU.shuffle()

                    if (fragenFaktenDEU.isNotEmpty()) {
                        val currentQuestion =
                            "Frage ${currentIndex + 1} / ${fragenFaktenDEU.size}: ${fragenFaktenDEU[currentIndex].first}"
                        binding.centralText.text = currentQuestion
                    } else {
                        catch (e: Exception) {
                            Log.e("tag: MyApp", msg: "Ups!, no data..", e)
                        } finally {
                            disconnect()
                        }
                    }
                }
            }
        }
        Log.d("tag: MyApp", msg: "Anzahl geladener Fragen: ${fragenFaktenDEU.size}")

        withContext(Dispatchers.Main) { this: CoroutineScope
            val sharedPreferences = getSharedPreferences("MyApp", Context.MODE_PRIVATE)
            val editor = sharedPreferences.edit()
            editor.putInt("totalQuestions", fragenFaktenDEU.size)
        }
    }
}
```

```
1 package com.just161.privileges_4.data.room
2
3 import android.content.Context
4 import androidx.room.Database
5 import androidx.room.Room
6 import androidx.room.RoomDatabase
7 import com.just161.privileges_4.data.dao.PlayerDao
8 import com.just161.privileges_4.data.model.Player
9
10 Just161
11 @Database(entities = [Player::class], version = 6)
12 abstract class AppDatabase : RoomDatabase() {
13     Just161
14     abstract fun playerDao(): PlayerDao
15 }
16
17 Just161 *
18 object DatabaseProvider {
19     private var instance: AppDatabase? = null
20
21     Just161
22     fun getInstance(context: Context): AppDatabase {
23         return instance ?: synchronized(lock: this) {
24             instance ?: Room.databaseBuilder(
25                 context.applicationContext,
26                 AppDatabase::class.java,
27                 "player-database"
28             )
                .fallbackToDestructiveMigration()
                .build().also { instance = it }
29         }
30     }
31 }
```

cht, ihre Religion zu praktizieren und neue Energie zu tanken, w'u00e4hrend andere dies nicht k'u00f6nnen. Wenn Menschen nicht an religi'u00f6sen Feiertagen, die ihnen wichtig sind, frei haben, kann dies zu Stress und Frustration fu00f6hren. Es kann Arbeit und der Religion fu00f6hren.",{"id":4,"frage":"Du bist privat versichert.", "fakt": "Fakt: In Deutschland sind rund 10 % der Bev'u00f6lkerung privat versichert. Die meisten Privatversicherten sind Beamte, Selbst'u00e4ndige oder Arbeitnehmer Quelle: GKV-Spitzenverband, Zahlen und Fakten zur GKV"}nDie beste medizinische Versorgung kann dazu beitragen, Krankheiten fr'u00f6hzeitig zu erkennen und zu behandeln. Dies kann dazu fu00f6hren, dass Krankheiten weniger schwerwiegend n'k'u00f6nnen. Die beste medizinische Versorgung kann auch dazu beitragen, die Lebensqualit'u00e4t von Menschen mit chronischen Krankheiten zu verbessern. Durch die Bereitstellung von qualitativ hochwertigen Behandlungen und Unterst'u00fctzung it chronischen Krankheiten ein aktives und erf'u00f6lltes Leben fu00f6hren. nDie Tatsache, dass die Privatversicherung ein Privileg ist, das nicht jeder hat, ist ein Problem. Denn die beste medizinische Versorgung sollte fu00f6r alle Menschen zug'u00eangig von ihrem Einkommen.", {"id":5,"frage":"Du hast einen deutschen Pass.", "fakt": "Nicht alle Menschen haben einen Pass. Sch'u00e4tzungen zufolge haben rund 1,3 Milliarden Menschen weltweit keinen Pass. Diese Menschen haben oft nur eingesckkeiten, sich zu bewegen und ihr Leben zu gestalten.", {"id":6,"frage":"Du wurdest noch nie gemobbt.", "fakt": "Mobbing ist ein ernstes Problem, das fu00f6r die Betroffenen erhebliche negative Folgen haben kann. Quelle: Deutsches Jugendinstitut, Mobbing ist eine Form von Gewalt, die von einer oder mehreren Personen gegen'u00f6ber einer anderen Person ausge'u00f6hrt wird. Das Ziel ist, der betroffenen Person Angst einzujagen, sie zu verletzen oder sie zu kontrollieren. Die meisten Opfer von Nungen, Personen mit Migrationshintergrund oder Personen mit psychischen Erkrankungen. nDie Folgen von Mobbing k'u00f6nnen fu00f6r die Betroffenen sehr schwerwiegend sein. Mobbing kann zu psychischen Problemen wie Angst, Depression und Es kann auch zu k'u00f6rperlichen Problemen wie Schlaflosigkeit und Kopfschmerzen fu00f6hren. Mobbing kann auch die Schulleistungen und die soziale Entwicklung der Betroffenen beeintr'u00e4chtigen.", {"id":7,"frage":"Wenn du knapp bei Kassa unterst'u00fctzen.", "fakt": "Laut einer Studie erhalten etwa 40% der jungen Erwachsenen in Deutschland finanzielle Unterst'u00fctzung von ihren Eltern. nDie finanzielle Unterst'u00fctzung durch die Eltern kann fu00f6r junge Erwachsene eine gro'u000 in Phasen der Unsicherheit, wie z.B. w'u00e4hrend des Studiums oder beim Berufseinstieg. Allerdings hat nicht jeder das Privileg, solche Unterst'u00fctzung zu erhalten. Dies kann zu Ungleichheiten fu00f6hren, da diejenigen, die Unterst'u00fctzung erler Bildung und Karriere voranzukommen. Auf der anderen Seite kann eine st'u00e4ndige finanzielle Abh'u00e4ngigkeit von den Eltern auch das Selbstwertgef'u00fchl und die Unabh'u00e4ngigkeit junger Erwachsener beeintr'u00e4chtigen. Es ist wichtig terst'u00fctzung und Selbstst'u00e4ndigkeit zu finden.", {"id":8,"frage":"Bist du in dem Land geboren, dessen Amtssprache deine Muttersprache ist?", "fakt": "Laut dem Statistischen Bundesamt waren im Jahr 2020 rund 21% der in Deutschland lebende geboren. Quelle: Statistisches Bundesamt. nIn einer globalisierten Welt ist Migration ein h'u00e4ufiges Ph'u00e4nomen. Viele Menschen leben in einem Land, dessen Amtssprache nicht ihre Muttersprache ist. Dies kann verschiedene Gri'u00f6nde haben Das Leben in einem solchen Umfeld kann sowohl Herausforderungen als auch Chancen bieten. Die Herausforderungen k'u00f6nnen sprachliche Barrieren, kulturelle Unterschiede und Anpassungsschwierigkeiten umfassen. Fu00f6r diese Menschen k'u00

# Privileges.App

## Funktionen & Code-Snippets

**Zertifikatserstellung** mit Bitmaps: Nach Abschluss des Spiels wird ein Zertifikat generiert, welches den Namen und den Fortschritt der Spieler\*innen anzeigt. Hierbei wurde das Konzept der **Bitmap-Generierung** angewendet, um das Zertifikat als Bilddatei zu kreieren. Dieses wird anschließend im Gerätespeicher abgelegt.

```
canvas.drawBitmap(certificateBitmap, left: 0f, top: 0f, paint: null)

val paint = Paint()
paint.color = Color.parseColor(colorString: "#1d151e")
paint.textSize = 168f
paint.textAlign = Paint.Align.CENTER

val typeface = ResourcesCompat.getFont(context, R.font.unnabold)
paint.typeface = typeface

print ("${outputBitmap.height} ${outputBitmap.width}")
canvas.drawText(text: "$playerName", centerX, y: centerY + 80, paint)
canvas.drawText(text: "$playerScore", centerX, y: centerY + 410, paint)
```



# Weiterentwicklung

## Lizenz und Kontaktinformationen

Das Projekt Privileges.App freut sich über alle die Lust haben sich in irgendeiner Weise zu beteiligen. Sei es durch alternative Fragenkataloge die verschiedene Kontexte beleuchten, Übersetzungen in andere Sprachen oder eine Verbesserung der Spiellogik. Anregungen und Kritik ist mehr als willkommen: [hello@privilege.app](mailto:hello@privilege.app).

Die App ist unter der MIT-Lizenz auf Github zu finden.

**SOURCES:** Privileges' has been developed by drawing insights from various sources, such as 'EINEN SCHRITT NACH VORN' by [handicap-international.de](http://handicap-international.de), "Der Lauf des Leben" by Flutter and 'abgehängt - Ein Privilegienspiel' by [bne-sachsen.de](http://bne-sachsen.de). These resources have greatly influenced the game's design and mechanics, providing valuable perspectives on privilege and societal dynamics.

Used Libraries: [shimmer-android](https://github.com/meta-platforms/shimmer-android) by META.  
Copyright (c) Meta Platforms, Inc. and affiliates. All rights reserved.

Privileges is an idea conceived by Boris Niehaus. It was developed as part of the Mobile App Development course at the Syntax Institute - Copyright 2023.

Contact: [bo.niehaus@gmail.com](mailto:bo.niehaus@gmail.com)  
Web: <https://borisniehaus.de>  
Github: <https://github.com/just1984>

