Projektarbeit

Implementierung eines Sudoku-Spiels zur Evaluierung der technischen Konzepte von Flutter

Autor: Christian Martin Slupikowski

Matrikel-Nr.: 10049494

E-Mail: slupikowski.christianmartin@fh-swf.de

Prüfer: Prof. Dr. Christian Gawron

Abgabe: 17.02.2021, Iserlohn

Inhaltsverzeichnis

[Einführung 3](#_Toc62493380)

[Motivation und Zielsetzung 3](#_Toc62493381)

[Anwendungsbereiche und Zielgruppen 3](#_Toc62493382)

[Anforderungsanalyse 3](#_Toc62493383)

[Entwicklungsprozess 3](#_Toc62493384)

[Funktionale Anforderungen 3](#_Toc62493385)

[Nicht Funktionale Anforderungen 4](#_Toc62493386)

[Use Cases 4](#_Toc62493387)

[Entwurf 5](#_Toc62493388)

[Konzeption 5](#_Toc62493389)

[Sudoku Generierung Algorithmus 5](#_Toc62493390)

[Exkurs: Flutter 5](#_Toc62493391)

[Implementierungsphase 5](#_Toc62493392)

[Veröffentlichung im Google Play Store 6](#_Toc62493393)

[Fazit 6](#_Toc62493394)

[Mögliche Erweiterungen 6](#_Toc62493395)

[Abbildungsverzeichnis 7](#_Toc62493396)

[Glossar 8](#_Toc62493397)

[Literaturverzeichnis 9](#_Toc62493398)

# Einführung

## Motivation und Zielsetzung

Smartphones sind schon seit einiger Zeit der tägliche Begleiter von vielen Menschen. Dabei werden sie in sehr vielen Bereichen eingesetzt, ob es berufliche Zwecke oder private Zwecke sind. Daher werden die Ansprüche an entwickelter Software immer höher und neue Technologien werden entwickelt, um das Erstellen von neuen Applikationen zu vereinfachen. Dabei ist ein häufiges Problem, dass Applikationen für jedes Betriebssystem, wie Beispielweise Android und IOS, entwickelt werden müssen. Dadurch sind dies unabhängig voneinander arbeitende Applikationen, welche ihren eigenen Quellcode zugrunde liegen. Dabei sollen die Applikationen genau dasselbe tun und greifen meistens sogar auf das selbe Backend zu. Um die Entwicklung, sowie den Wartungsprozess von Applikationen zu beschleunigen, ist eine Applikation für beide Betriebssysteme deutlich einfacher. Für diesen Fall wurde Flutter entwickelt. Flutter kompiliert mit Dart geschrieben Quellcode in jeweilige Applikationen für Android und IOS. Eventuelle Anpassungen für Betriebssystem spezifische Funktionen, die nicht von Flutter für beide Betriebssysteme verallgemeinert werden können, können dennoch Betriebssystemspezifisch implementiert werden.

Um als Softwareentwickler auf den neusten Stand zu bleiben ist es wichtig in allen Bereichen einen Überblick zu behalten, welche Techniken aktuell sind und welche verwendet werden. Damit ich als Hauptberuflicher Web/Backend Entwickler mich auch in Smartphone Applikationen auskenne und einen Überblick von Flutter verschaffen kann, wird in dieser Projektarbeit eine Sudoku-App erstellt.

Sudoku ist ein aus den 80er Jahren stammendes, mittlerweile sehr populäres Logikrätsel, für das es eine ganze Reihe von Lösungsalgorithmen gibt. Dabei geht es bei Sudoku darum, ein 9x9 gefülltes Raster mit Zahlen von 1-9 zu füllen. Dabei dürfen die Zahlen 1-9 jeweils nur einmal Waagerecht, Senkrecht und in einem 3x3 Kästchen vorkommen.

## Anwendungsbereiche und Zielgruppen

Die Applikation ist für alle Rätselbegeisterte gedacht, die gerne auf Ihrem Smartphone, Tablett oder anderen Android Geräten Sudoku spielen möchten. Da verschiedene Schwierigkeitsgrade zur Auswahl stehen sollen, ist die Applikation sowohl für Anfänger als auch für fortgeschrittene Spieler geeignet.

# Anforderungsanalyse

## Entwicklungsprozess

Die Sudoku-App soll nach dem Wasserfall Modell entwickelt werden. Es wurde sich aus folgenden Gründen gegen einen agilen Entwicklungsprozess entschieden. Da die Projektarbeit in Einzelarbeit entwickelt wird, stehen keine konkreten Tester zur Verfügung, sodass die Software, neben eigenen Tests bei der Entwicklung, zusätzlich nur durch eine ausgewählte Gruppe zum Abnahmetest getestet wird. Ebenfalls ist nicht klar wie viele Kapazitäten pro Zeiteinheit zur Verfügung stehen, daher ist es einfacher in Paketen zu planen, als in Zeitlichen Sprints. Außerdem wird am Ende des Projektes ein fertiges Produkt entstanden sein, welches in der Regel nicht mehr weiterentwickelt wird. Wenn allerdings trotzdem weitere Features für das Projekt in Frage kommen sollten, dann könnten diese mit einem agilen Entwicklungsprozess entwickelt werden.

## Funktionale Anforderungen

1. Dem Anwender soll es möglich sein, ein Sudoku Spiel komplett zu spielen.
2. Der Anwender kann genau immer ein Feld gleichzeitig selektieren. Dabei sind die Felder ausgeschlossen, welche zum Start des Spiels vorbelegt sind.
3. Felder die beim Start des Spiels vorbelegt sind, werden durch eine dicke schwarze Schrift gekennzeichnet.
4. Das selektierte Feld wird farblich hinterlegt.
5. Im unteren Teil der Applikation sollen Zahlen zwischen 1 und 9 hinterlegt werden.
6. Wenn ein Feld selektiert ist, kann durch drücken einer dieser Zahlen, das Feld gefüllt werden.
7. Wurde das Feld bereits gefüllt, wird der Wert mit dem neuen Wert überschrieben.
8. Eine Zahl kann aus einem Feld entfernt werden, indem länger auf das Feld gedrückt wird.
9. Ein Menü soll es geben in dem der Anwender folgende Aktionen ausführen kann
   1. Generieren eines neuen Spiels, in verschiedenen Schwierigkeitsstufen (Leicht, Mittel, Schwer).
   2. Speichern des aktuellen Spielstands. Es soll immer nur genau ein Spielstand gespeichert werden können. Dabei muss sichergestellt werden, dass der Anwender nochmal gefragt wird, ob er wirklich den aktuell gespeicherten Spielstand überschreiben möchte.
   3. Laden des vorher gespeicherten Spielstands.
10. Wenn der Anwender ein Spiel gewinnt, wird diesem ein Pop-Up Fenster angezeigt.
11. Es soll eine Hilfe für den Anwender geben, welche aktivier-/deaktivierbar ist.
12. Wenn die Hilfe aktiviert ist, werden richtig eingefüllte Zahlen mit einem grünen Hintergrund und falsche mit einem roten Hintergrund befüllt.
13. Das Sudoku-Spiel soll nur im Hochformat angezeigt werden können.

## Nicht Funktionale Anforderungen

1. Das generieren eines neuen Sudoku Spiels soll nicht länger als 5 Sekunden dauern, unabhängig von dem Schwierigkeitsgrad.
2. Die Applikation soll auf allen Android Geräten mit einer Mindestversion von 1.1.1 laufen.
3. Die Applikation soll nicht für IOS Geräte veröffentlicht werden.

## Use Cases

Um das Verständnis des Use-Case Diagramms zu verbessern, wird zunächst auf jeden Use-Case anhand des Use-Case-Diagramms eingegangen.

**Spieler:** Der Spieler ist der Anwender, welcher die App bedient und ein Sudoku spielen möchte.

**Neues Spiel:** Startet ein Neues Spiel und löst somit das Generieren eines Sudokus aus.

**Schwierigkeitsgrad wählen:** Beim Starten eines neuen Spiels, kann der Schwierigkeitsgrad ausgewählt werden.

**Sudoku Generieren:** Generiert ein Neues Spiel, mit neuen Zahlwerten. Hinterlegt die initialen Werte in der Sudoku-Tabelle. Intern wird die Lösung gespeichert.

**Spiel Laden:** Lädt ein Sudoku Spielstand. Setzt voraus, dass bereits ein Spielstand gespeichert wurde.

**Spiel Speichern:** Speichert den aktuellen Spielstand. Setzt voraus, dass ein Sudoku Spiel generiert worden ist.

**Feld selektieren:** Selektiert ein Feld, welches kein initiales Feld sein darf. Setzt voraus, dass ein Sudoku Spiel generiert worden ist.

**Zahl auswählen:** Eine Zahl zwischen 1 und 9 kann ausgewählt werden. Setzt voraus, dass ein Feld selektiert wurde.

**Zahl löschen:** Löscht die eine Zahl aus einem ausgewählten Feld. Setzt voraus, dass dort eine Zahl ausgewählt wurde.

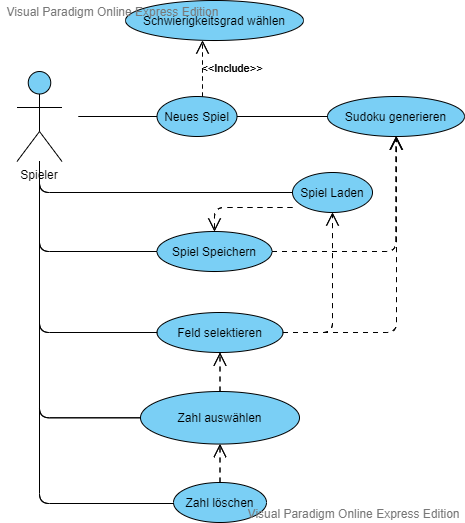


Abbildung 1 Use-Case Diagramm der Sudoku-App

## Entwurf

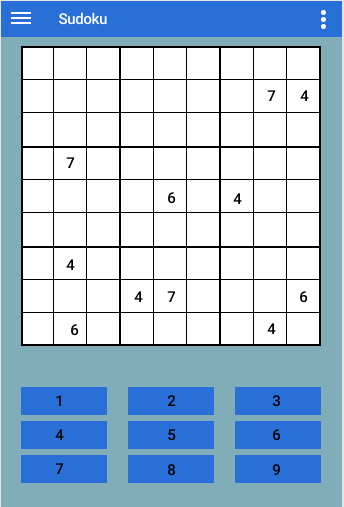
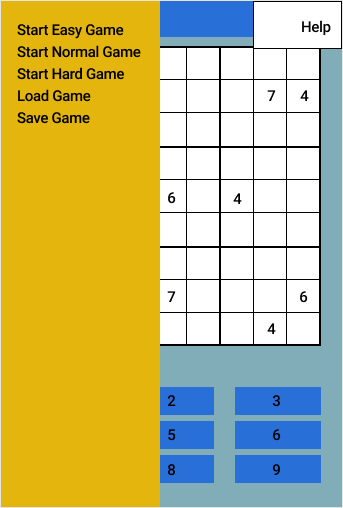
 

Abbildung 2 Mockups der Sudoku-App

# Konzeption

## Sudoku Generierung Algorithmus

## Exkurs: Flutter

* Aus der Sicht eines Webentwicklers

## Implementierungsphase

* Besonderheiten bei der mobilen Entwicklung
* Speichern

## Veröffentlichung im Google Play Store

# Fazit

# Mögliche Erweiterungen

* Kompilierung mit der neuen Flutter Beta in HTML/Javascript Quellcode

# Abbildungsverzeichnis

[Abbildung 1 Use-Case Diagramm der Sudoku-App 5](#_Toc62574275)

[Abbildung 2 Mockups der Sudoku-App 6](#_Toc62574276)

# Glossar

# Literaturverzeichnis

Computing, 1., 2017. *Backtracking Algorithm – Sudoku Solver.* [Online]   
Available at: https://www.101computing.net/backtracking-algorithm-sudoku-solver/

Computing, 1., 2019. *Sudoku Generator Algorithm.* [Online]   
Available at: https://www.101computing.net/sudoku-generator-algorithm/