

Speedcube-Metrics

Mobile Applikationen

Aaron Schweig, Michael Angermeier, Patrick
Mischka, Troy Keßler, Jan Grübener

Studiengang
Wirtschaftsinformatik
WWI 18 SEAC

Dozent
Michael Spengler



DHBW Mannheim
9. November 2020

Inhaltsverzeichnis

1	Worum geht es?	2
1.1	Die Idee	2
2	How to Use?	3
3	Technology & Architecture	9
3.1	Was ist ein BaaS	9
3.2	Wieso Firebase?	9
4	Testing	10
5	Backlog	11

1 Worum geht es?

1.1 Die Idee

Als Ernő Rubik 1974 den sogenannten Zauberwürfel entwarf wurde das Puzzle zu einem weltweiten Phänomen [4]. Millionen von Kindern und Erwachsenen versuchten sich an diesem Rätsel, wobei die Lösung oft als unmöglich galt. Schließlich kamen erste Lösungsanleitungen wie die Fridrich- oder die Roux-Methode. Damit konnte jeder Würfel mithilfe weniger Algorithmen gelöst werden. Schnell folgten erste Weltrekorde, der erste offizielle Weltrekord lag dabei noch bei 22.95 Sekunden, wobei die Zeiten schnell fielen. Heute liegt der Weltrekord für den 3x3 bei unglaublichen 3.47 Sekunden, eine Leistung die Glück und vorallem dediziertes Training erfordert [1].

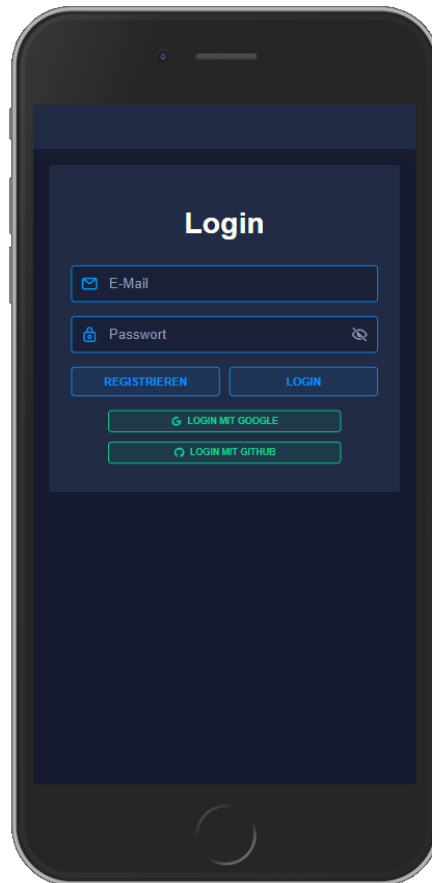
Spätestens mit dem bekannter werden von Speedcubing (dem Lösen eines Zauberwürfels in möglichst kurzer Zeit) entstanden erste Softwareunterstützungen für Training und Wettkämpfe. Es wurden Algorithmen entwickelt, mit denen man Zauberwürfel für Wettkämpfe verdrehen kann, sodass diese schwer genug und dennoch fair für alle Teilnehmer sind. Weiterhin wurden erste Apps und Webseiten entwickelt, die diese Wettkämpfe simulieren, sodass man auch von zu Hause aus seine Fähigkeiten unter Beweis stellen kann. Nach und nach wurden auch weitere Features hinzugefügt, mit denen man sich eher auf's reine Training konzentrieren konnte.

Wir haben jedoch beobachtet, dass besonders Webbasierte Seiten keine gute User Experience bieten. Das liegt unter anderem an dem recht altmodischen Design, als auch an fehlenden oder schlecht umgesetzten Features. Da sich in unserer Gruppe Speedcuber befinden, welche solche Seiten regelmäßig nutzen hatten wir die Idee im Rahmen dieser Vorlesung eine Webapp zu entwickeln, welche (unserer Meinung nach) eine bessere Alternative zu bestehenden Angeboten darstellt. Dabei wollen wir auf bestehenden und funktionierenden Methoden aufbauen und mit entsprechenden modernen Features und Designentscheidungen die User Experience und den Spaß am Speedcubing erhöhen.

2 How to Use?

Anmeldung und Registrierung

Beim Starten der App wird der Nutzer zuerst gebeten, sich mit einzuloggen oder zu registrieren.



Er besitzt dabei die Möglichkeit einen Account mittels E-Mail Adresse und eigens gewähltem Passwort zu erstellen oder auf die Autorisierungsdienste von Google oder GitHub zuzugreifen und sich mit seinen bereits vorhandenen Accounts anzumelden. Das bei der Registrierung gewählte verfahren, muss zukünftig auch für jede Anmeldung verwendet werden.

Übersicht

Nach dem erfolgreichen einloggen kann der sieht der Nutzer eine Auswertung seiner letzten Ergebnisse. Hier wird nach Trainings und Wettkämpfen unterschieden und die Ergebnisse des vergangenen Monats graphisch aufgezeigt. So lassen sich Erfolge und Tendenzen schnellstmöglich erkennen.

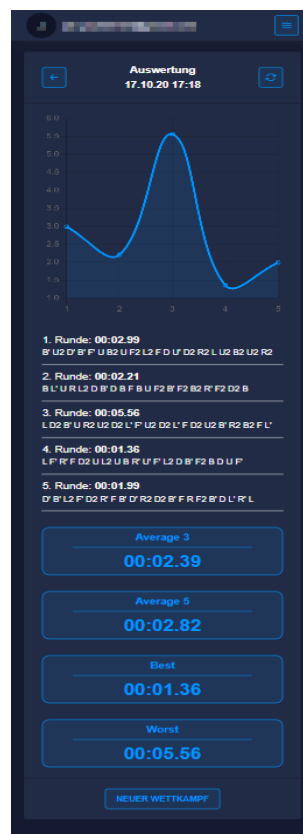


Um schnellstmöglich mit dem Spiel beginnen zu können, kann gibt es direkt unter der graphischen Auswertung Buttons um ein Training oder Wettkampf zu starten. Daten interessierte können die Möglichkeit nutzen, sich mit der Hilfe des Übersicht Buttons genauer in ihre erreichten Resultate einzulesen und so sich weiter zu verbessern.

Wettkampfübersicht + ↺

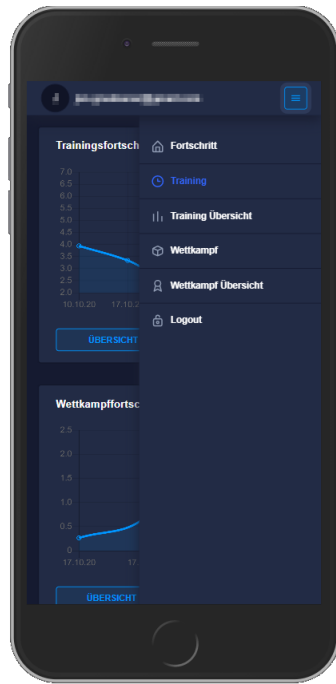
Datum	Average 3	Analyse
17.10.20 17:26	00:00.26	ib
17.10.20 17:19	00:00.67	ib
17.10.20 17:18	00:02.39	ib
17.10.20 17:26	00:00.26	ib

Dafür wird ihnen extra eine Tabelle mit den vergangenen Resultaten angezeigt. Für abgeschlossene Trainings wird die jeweilige Lösungszeit des Scrambels angezeigt. Für absolvierte Wettkämpfe wird hier der Average 3 angezeigt. Über die Spalte Analyse können die Ergebnisse der einzelnen Runden eines Wettkampfes sowieso weitere Daten wie Average 5, Best, Worst, etc. angezeigt werden.



Navigation

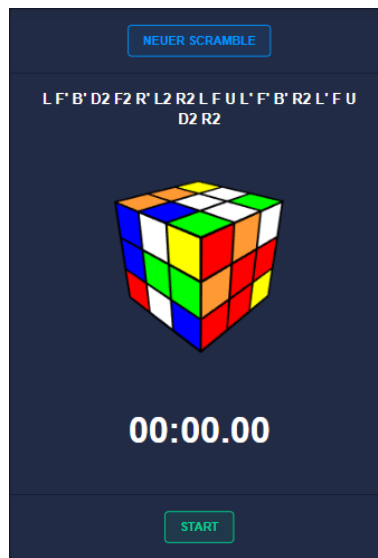
Die Navigation innerhalb der App kann entweder durch die in den einzelnen Abschnitten eingepflegten Buttons erfolgen, die mit einer einfachen Beschreibung zu den logischen nächsten Schritten führen oder über die im Menü eingebundene Navigationsleiste.



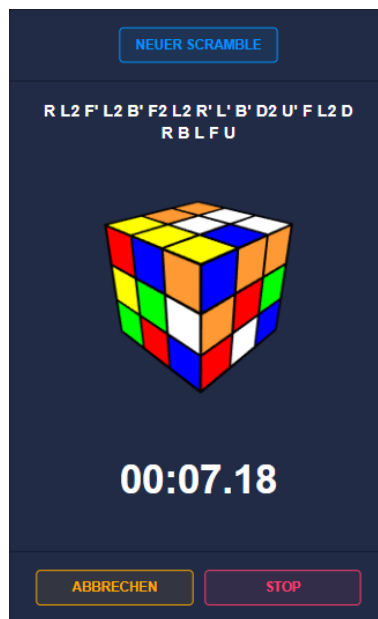
Die Navigationsleiste ermöglicht jederzeit das schnelle wechseln zu den Übersichten der letzten Trainings und Wettkämpfe sowie das Starten dieser. Hier findet der Nutzer auch die Möglichkeit vor sich auszuloggen, bevor er die App beendet.

Training

Ein Training beginnt immer damit, dass die App einen automatisch generierten Scrupel vorschlägt, welcher im Anschluss der Nutzer auf seinem Rubik's Cube übernimmt. Dies kann er entweder über den als Text angezeigten Scrubel oder mithilfe des 3 Dimensional dargestellten Cubes. Zur einfacheren Handhabung und Kontrolle ist dieser durch halten und Bewegungen im Raum dreh und bewegbar. Unpassende Scrumbels können mit dem "Neuer Scrumble" Button einfach übersprungen werden.



Der Start Button startet automatisch den eingeblendeten Timer. Wird man bei der Lösung des Scrambels gestört oder möchte aus anderen Gründen einen neuen Versuch starten, so kann der aktuelle Durchgang durch den Abbrechen Button unterbrochen und der Scrambel erneut gelöst werden. Stop speichert das Training und bietet sofort einen neuen Scrambel für einen neuen Versuch.

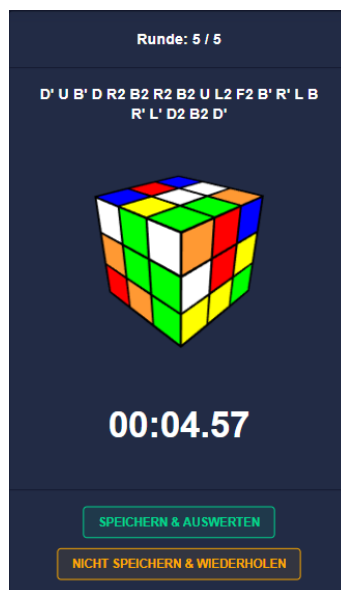


Wettkampf

Ein Wettkampf besteht immer aus 5 unterschiedlichen Scrumbels, die schnellstmöglich hintereinander gelöst werden sollen. Das drücken des Stopp Buttons zeigt immer automatisch den nächsten Scrumbel an, der mit dem eigenen Rubiks Cube gelöst werden soll. Hierfür muss der Timer wieder separat gestartet und gestoppt werden.



Nach dem alle 5 Scrumbels gelöst wurden, bekommt der Nutzer die Möglichkeit den Wettkampf zu verwerfen oder ihn zu speichern. Das Verwerfen führt automatisch zu einem neuen Wettkampf mit anderen Scrumbels, das speichern zeigt die Ergebnisse des Wettkampfes an und berechnet wichtige Daten wie Average 4 und Average 5 um sich mit alten Wettkämpfen oder Freunden vergleichen zu können.



3 Technology & Architecture

Im folgenden Abschnitt wird kurz die Wahl eines **BaaS** (Backend as a Service) Anbieters erläutert. Dabei liegt der Fokus vor allem darauf, wieso ein BaaS unter dem gegebenen Anwendungsfall sinnvoll ist.

3.1 Was ist ein BaaS

Definition 3.1 (Backend as a Service). Backend-as-a-Service (BaaS) ist ein Cloud-Service-Modell, bei dem Entwickler alle Hintergrundaspekte einer Web- oder Mobilanwendung auslagern, so dass sie nur das Frontend schreiben und warten müssen. BaaS-Anbieter bieten vorab geschriebene Software für Aktivitäten auf Servern an, z. B. Benutzerauthentifizierung, Datenbankverwaltung, Remote Updates und Push-Benachrichtigungen (für mobile Apps) sowie Cloud-Storage und -Hosting [2].

Anhand dieser Definition lässt sich erkennen, dass ein BaaS darauf abzielt, Entwicklungszeiten zu verkürzen und klassische Aufgaben bei der Erstellung einer Webapp zu standardisieren. Zu dem typischen Elementen, die durch BaaS-Anbieter übernommen werden gehören:

- Datenbankverwaltung
- Cloud-Speicher
- Authentifizierung
- Push-Notifications
- Hosting

Es ist zu erkennen, dass so die Entwicklung und die initiale Hürde verloren geht, da elementare Bestandteile einer Webapplikation bereits vorhanden sind und nicht für jeden Anwendungsfall erneut entwickelt werden müssen. Mithilfe einer vorhanden Datenbank und einer sicheren, verwalteten Nutzerauthentifizierung sind bereits Grundsteine gelegt, damit eine Webapp erfolgreich aufgebaut werden kann. Das Modell eines BaaS ist noch nicht sehr alt und wurde innerhalb der letzten Jahre immer populäre. Ereignisse, wie die strikte Regulierung der DSGVO stellt die Nutzung dieser Modelle in einem Enterpriseumfeld vor Herausforderungen. Diese müssen vor einer Entscheidung für einen solchen Dienst abgewogen werden.

3.2 Wieso Firebase?

Innerhalb dieses Anwendungsfalles lassen sich verschiedene Aspekte finden, welche für die Nutzung eines BaaS und auch Firebase als populären Anbieter eines solchen Service sprechen.

1. Speicherung weniger bis garkeiner nutzerspezifischer Daten:

In diesem Anwendungsfall werden Nutzerdaten nur gespeichert, damit der eigene Fortschritt visualisiert werden kann. Das bedeutet im Kern, dass außer einer eindeutigen Mailadresse keine zusätzlichen Informationen eines

Nutzers benötigt werden. Mithilfe von verschiedenen Social-Auth Providern kann auch eine sichere Authentifizierung gewährleistet werden, da diese immer einem aktuellen Web-Standard - OAuth 2.0 - folgen.

2. kaum speziell im Backend auszuführende Logik:

Dadurch, dass diese Applikation vorerst rein zum visualisieren des persönlichen Fortschritts dient, gibt es sehr wenig Logik, die auf einem Server ausgeführt werden müssten. Durch die direkte Verbindung zwischen Frontend und Datenbank mittels Firebase können Lese- und Schreib Anforderungen direkt im Frontend gesteuert werden. Die Logik zum generieren eines zufälligen Scrumbles und zum erstellen des 3D-Cubes wird mithilfe einer serverless Function ¹ ausgeführt.

3. einfache Kommunikation zwischen Frontend und Datenpersistierung ohne Schnittstellenprobleme:

Mithilfe von Firebase ist es möglich, wie bereits oben beschrieben, direkt aus dem Frontend mit der Datenbank zu interagieren. Dabei werden natürlich alle Sicherheitsstandards seitens Firebase eingehalten, damit es nicht zu einem Missbrauch der Daten kommen kann. Während der Entwicklung einer Applikation steht ein Team oft vor dem Problem der Schnittstellendefinition. Dabei gilt es vor allem folgende Fragen zu beantworten:

- (a) Welches Format wird verwendet (REST, GraphQL, SOAP, etc.)?
- (b) Welche Struktur haben die Daten?

Sind diese Fragen beantwortet, so muss zuerst ein gesamtes Backend entwickelt werden, in welchem die Funktionalität unter Berücksichtigung vieler Sicherheitsaspekte oder Best-Practices umgesetzt werden müssen. Das kostet Zeit und bietet ein hohes Potenzial für menschenbedingte Fehler. Zum Beispiel können Endpunkte nicht funktionieren, es bestanden unterschiedliche Ansichten bzgl. der Nutzung und des Funktionsumfangs des gewählten Schnittstellenformats oder gewünschte Operationen konnten nicht rechtzeitig implementiert werden.

All diese Probleme sind durch die direkte Kommunikation zwischen den UI-Elementen und der Datenpersistierung gelöst. Es kann sich voll und ganz auf das sinnvolle Erstellen einer Datenstruktur konzentriert werden, sodass am Ende auch hier keine Missverständnisse zwischen verschiedenen Teams entstehen. Im Falle der hier vorgestellten Applikation erleichtert uns das die Entwicklung enorm, weswegen auch unter Berücksichtigung eines Zeitfaktors viele Vorteile entstehen.

4 Testing

Aufgrund unserer Wahl für Firebase als Backend sind Backendtests hinfällig und lediglich UI-Tests im Frontend werden benötigt. Hierfür wird cypress als Testing-Framework genutzt.

¹Der Begriff „Serverless“ (serverlos) bezieht sich auf ein cloudnatives Entwicklungsmodell, bei dem Entwickler Anwendungen erstellen und ausführen können, ohne Server verwalten zu müssen. [3]

Im Rahmen der Frontend-Tests wird zuerst getestet, dass dem Nutzer die korrekte Anmeldemaske angezeigt wird, welche die beiden Anmeldefelder mit Email und Passwort, ein Registrierungsbutton, ein Anmeldebutton und je ein Button um sich mit Google oder Github anmelden zu können, enthalten soll. Desweiteren wird hier getestet ob ein erfolgreiches An- und Abmelden möglich ist.

Nach dem erfolgreichen Anmelden wird die Fortschrittsseite angezeigt, welche sowohl den Trainingsfortschritt, als auch den Wettkampffortschritt enthält. Unter jeder Fortschrittsansicht sind zwei Buttons. Mit einem kann die jeweilige Übersicht detaillierter angezeigt werden, mit dem anderen kann ein Wettkampf / Training gestartet werden.

Das korrekte Öffnen und Schließen des seitlichen Menüs wird ebenfalls getestet. Über dieses Menü können alle weiteren Funktionalitäten der Anwendung erreicht werden. Das Öffnen dieser Funktionalitäten ist ebenfalls getestet.

Die spezifischen Inhalte von den unterschiedlichen Funktionalitäten sind ebenfalls getestet.

5 Backlog

Das Backlog beinhaltet alle Features, welche es nicht in die Implementierung geschafft haben. Grund dafür ist unsere Einschätzung bezüglich des Nutzen-Aufwands.

- Eine Berechnung des Lösungsalgorithmus anhand von Image Recognition des Zauberwürfels. Dabei soll der User mithilfe seiner Kamera den Würfel einscannen und die Software den entsprechenden Lösungsalgorithmus berechnen. Die Challenge hier ist das Einscannen des Würfels, da das Training eines Models sehr aufwendig ist (alle Testdaten müssten selbst generiert werden) und die Fehlerquote womöglich aufgrund Belichtung und anderer Einflüsse sehr hoch ist.
- Die Berechnung der Umdrehungen pro Sekunde anhand der Geräusche die ein Würfel beim drehen von sich gibt. Dabei ist ein wichtiger, aber auch sehr schwer zu messender Indikator für einen Speedcuber die TPS (Turns per Second). Wir hatten die Idee mithilfe von Audio Recognition Software diese Umdrehungen zu messen und für den User entsprechend darzustellen. Damit kann dann auch berechnet werden, wie flüssig der Speedcuber den Zauberwürfel gelöst hat, indem man sich die Spannweite der TPS betrachtet. Mögliche Challenges wäre hier wieder die Audio Recognition, obwohl diese womöglich wesentlich einfacher ist als die Image Recognition. Mögliche Fehlerquellen wären hier störende Hintergrundgeräusche oder der individuelle Klang jedes Würfels.
- Das Vergleichen der eigenen Leistung mit Freunden und anderen Nutzern der Webapp. Da Konkurrenz wohl die größte Motivation darstellt sich zu verbessern soll der User die Möglichkeit besitzen seine Leistungen mit allen zu teilen. Dies soll mithilfe einer globalen Rangliste möglich sein. Challenges wären hier die Verifikation, da man ohne weiteres die Zeit stoppen kann und diese veröffentlichen kann.

Literatur

- [1] World Cube Association. Records. <https://www.worldcubeassociation.org/results/records>. Aufgerufen: 27.10.2020.
- [2] Cloudflare. Was ist baas? — backend-as-a-service im vergleich zu serverless. <https://www.cloudflare.com/de-de/learning/serverless/glossary/backend-as-a-service-baas/>. Aufgerufen: 25.10.2020.
- [3] Redhat. Was ist serverless? <https://www.redhat.com/de/topics/cloud-native-apps/what-is-serverless>. Aufgerufen: 25.10.2020.
- [4] Wikipedia. Ernő rubik. https://de.wikipedia.org/wiki/Ern%C5%91_Rubik. Aufgerufen: 27.10.2020.