

Speedcube-Metrics

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

25. Oktober 2020

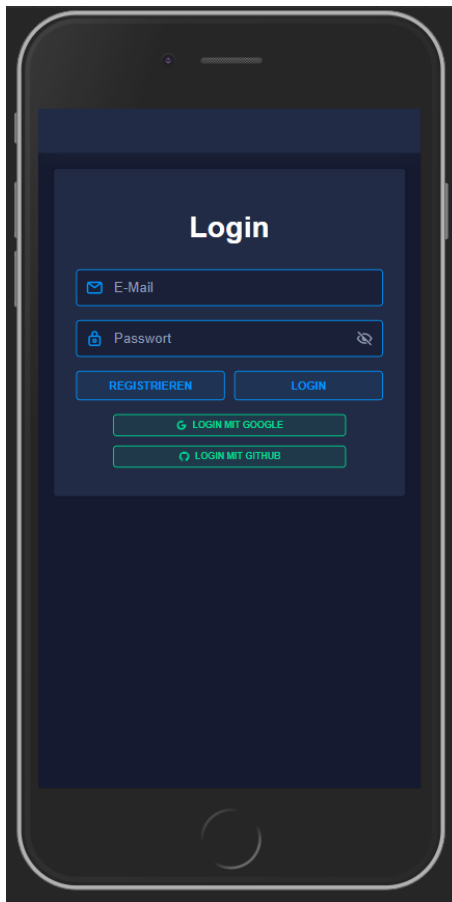
1 Worum geht es?

1.1 Die Idee

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			
17.10.20 17:19	00:00.67			
17.10.20 17:18	00:02.39			
17.10.20 17:26	00:00.26			

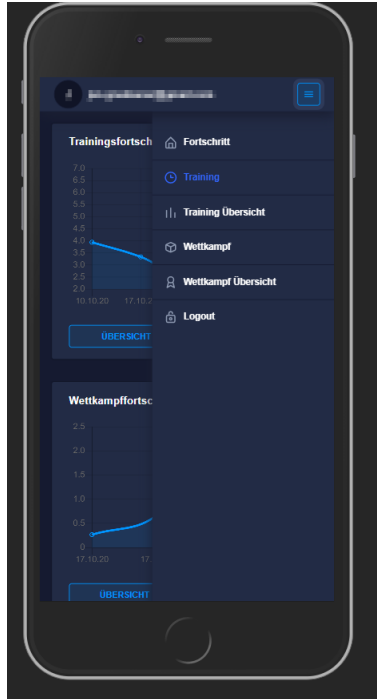
Dafür wird ihnen extra eine Tabelle mit den vergangenen Resultaten angezeigt. Für abgeschlossene Trainings wird die jeweilige Lösungszeit des Scrambels angezeigt. Für absolvierte Wettkämpfe wird hier der Average 3 angezeigt. Über die Spalte Analyse können die Ergebnisse der einzelnen Runden eines Wettkampfes 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

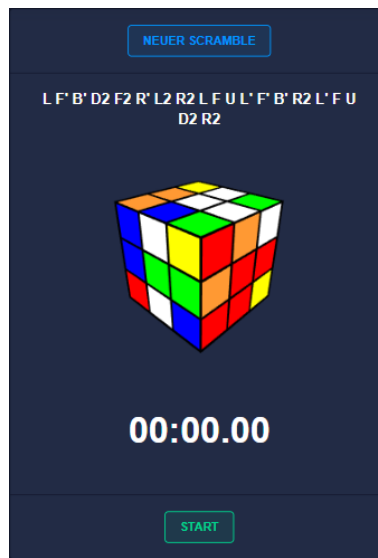
zu den logischen nächsten Schritten führen oder über die im Menü eingebundene Navigationsleiste.



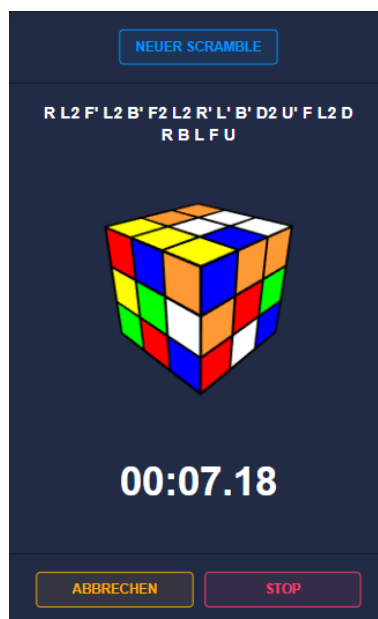
Die Navigationsleiste ermöglicht jederzeit das schnelle wechseln zu den Übersichten der letzten Trainings und Wettkämpfe sowie das Starten dieser. Hier findet der Nutzer auch die Möglichkeit vor sich auszuloggen, bevor er die App beendet.

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 Scrubels können mit dem "Neuer Scramble" Button einfach übersprungen werden.



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



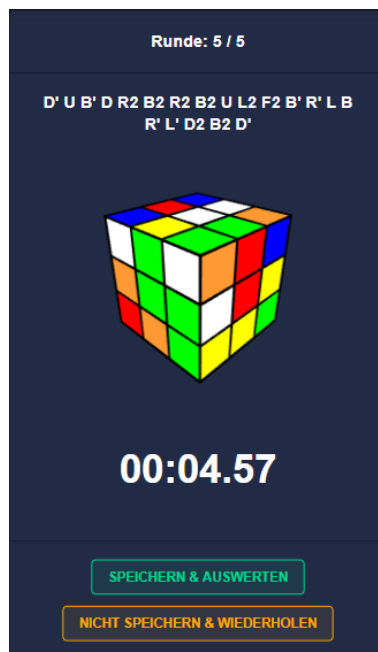
Wettkampf

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 seperat 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 UI Design

4 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.

4.1 Was ist ein BaaS

Definition 4.1 (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 ¹.

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 vorhandenen 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 populärer. 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.

4.2 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<https://www.cloudflare.com/de-de/learning/serverless/glossary/backend-as-a-service-baas/>

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“ ausgeführt.

5 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.

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.

6 Backlog