

**Schnappsidee**

**Spendido**

**Technische Informationen für die Jury**

**Technische Informationen für die Jury**

Aktueller Stand des Sourcecodes

* Link zu Github Repository
  + <https://github.com/JanKoch99/schnappsidee>

Ausgangslage

* Worauf habt ihr euch fokussiert?
  + Wir haben uns darauf fokussiert, einen funktionsfähigen Prototypen zu entwickeln und weiter auszubauen, den wir nun gerne mit den anderen Teilnehmer:innen teilen und testen möchten. Unser Ziel war es, den Prototypen bereits am ersten Tag einsatzbereit zu machen, sodass wir damit auch unsere Videos erstellen konnten. Ausserdem wollten wir eine robuste Webapplikation schaffen, die ohne Registrierung vollumfänglich nutzbar ist. Dabei haben wir besonderen Wert auf eine einfache und intuitive Bedienbarkeit gelegt, da unsere primäre Zielgruppe junge Menschen sind, welche den Umgang mit dem Mobiltelefon gewohnt sind.
* Welche technischen Grundsatzentscheide habt ihr gefällt?
  + Während wir unsere App kontinuierlich weiterentwickelt haben, benötigten wir eine flexible Datenbankstruktur. Zusätzlich war es uns wichtig, die beiden Nutzergruppen - Barkeeper und Benutzer - klar voneinander zu trennen. Daher haben wir zwei separate Webapplikationen erstellt, die beide auf dasselbe Backend zugreifen. Um den Entwicklungsprozess zu vereinfachen, entschieden wir uns ausserdem, möglichst alles in einer einzigen Programmiersprache umzusetzen, sodass wir nahtlos zwischen Frontend und Backend wechseln konnten.

Technischer Aufbau

* Welche Komponenten und Frameworks habt ihr verwendet?
  + Wir haben den MEARN-Stack verwendet, der aus MongoDB, Express, React und Node besteht. Wie bereits erwähnt, haben wir dabei zwei separate React-Applikationen entwickelt.
* Wozu und wie werden diese eingesetzt?
  + Der MEARN-Stack ermöglicht eine vollständige Full-Stack-Entwicklung mit JavaScript, von der Datenbank bis zum Frontend. Hier ist, wie wir die einzelnen Komponenten einsetzen:
    - **MongoDB**: Diese NoSQL-Datenbank speichert unsere Daten flexibel und skalierbar. Sie wird für die Speicherung und Verwaltung aller Nutzerdaten, App-Inhalte und Interaktionen verwendet.
    - **Express**: Dieses Framework für Node.js fungiert als Middleware und erleichtert die Handhabung der HTTP-Anfragen und -Antworten auf unserem Server. Es sorgt dafür, dass die Kommunikation zwischen Frontend und Backend effizient und zuverlässig erfolgt.
    - **React**: Wir haben zwei separate React-Applikationen entwickelt – eine für die Barkeeper und eine für die Benutzer. React ermöglicht es uns, eine dynamische, reaktive Benutzeroberfläche zu erstellen, die sowohl auf Desktop- als auch auf mobilen Geräten gut funktioniert.
    - **Node.js**: Auf der Server-Seite verwenden wir Node.js, um unsere serverseitige Logik zu implementieren und die Express-Anwendung auszuführen. Es ermöglicht uns, die Backend-Operationen effizient zu verarbeiten und mit der Datenbank zu kommunizieren.
  + Durch die Verwendung des MEARN-Stacks können wir eine nahtlose Integration zwischen Frontend und Backend gewährleisten und eine skalierbare, leistungsstarke Webanwendung entwickeln.

Implementation

* Gibt es etwas Spezielles, was ihr zur Implementation erwähnen wollt?
  + In der App gibt es mehrere Easter Eggs, die beim Entdecken besonders viel Spass machen – daher sollte der Ton unbedingt aktiviert sein. Die App ist am unterhaltsamsten, wenn sie gemeinsam mit Freunden erkundet wird. Zusätzlich bietet die App für Barkeeper ein einzigartiges Tool, mit dem sie den Kundenkontakt intensivieren und sowohl lustige als auch peinliche Momente schaffen können.
* Was ist aus technischer Sicht besonders cool an eurer Lösung?
  + Wir sind besonders stolz auf den Stil und die Fröhlichkeit, die unsere Lösung ausstrahlt. Viel Zeit und Mühe wurden in die Entwicklung eines durchdachten UI/UX-Konzepts investiert, und das spiegelt sich deutlich in der Benutzererfahrung wider. Technisch beeindruckt uns insbesondere die Echtzeitverarbeitung der Daten und deren ansprechende Darstellung auf den jeweiligen Oberflächen. Ein sehr besonderes Feature ist, dass die App mithilfe von Slack die Teilnehmer:innen des Hackathons erreichen und benachrichtigen kann.

Abgrenzung / Offene Punkte

* Welche Abgrenzungen habt ihr bewusst vorgenommen und damit nicht implementiert? Weshalb?
  + Wir haben bewusst darauf verzichtet, die Zahlungsabwicklung für Rechnungen zu integrieren, um die Nutzererfahrung beim Pitchen nicht zu beeinträchtigen. Als Beispiel haben wir an die Lösung von Twint gedacht, bei der das Geld bei Nichterfüllung der Challenges zurückerstattet werden könnte. Ausserdem haben wir keine Unterstützung für mehrere Bars implementiert, da wir zunächst testen wollen, wie die App bei „vera calma“ funktioniert, bevor wir diese Funktion hinzufügen. Auch auf die Erstellung eines Kontos haben wir verzichtet, um mögliche Hürden für die Nutzung der App zu vermeiden und so den Einstieg und das Benutzererlebnis zu vereinfachen.