Technical Documentation

BEKB Challenge - Team #bankingGuys

Link zu Git Repository mit Angabe des Branches:

* <https://github.com/mariusgiger/bernhackt-18-bekb-challenge>
* Branch master

**Ausgangslage**

Worauf habt ihr euch fokussiert?

* Die Finanzverwaltung von Bänz stellt einen sehr wichtigen Touchpoint für ihn mit der BEKB dar, weshalb wir uns besonders auf eine einfache und intuitive User Experience und den bewussten Einsatz von persönlichen Interaktionen fokussiert haben. Eine schlichte Datenvisualisierung bringt dabei auch für Bänz, der nicht täglich im Finanzumfeld tätig ist, wertvolle Insights zum Cashflow seines Unternehmens.

Welche technischen Grundsatzentscheide habt ihr gefällt?

* Aus zeitlichen Gründen haben wir die Implementation eines Backends von Anfang an ausgeklammert.
* Anstelle der Entwicklung von native mobile Applications haben wir uns einfachheitshalber auf einen webbasierten Approach mit einem responsiven Design festgelegt. Jedoch wäre aufgrund der Frameworkwahl (ReactJS) die Portierung auf eine mobile Application einfach (ReactNative oder ein ähnliches Framework).

**Technischer Aufbau**

* Welche Komponenten und Frameworks habt ihr verwendet? Wozu und wie werden diese eingesetzt?

|  |  |
| --- | --- |
| **Komponente/Framework** | **Zweck** |
| ReactJs (JSX, HTML, JavaScript, CSS) | Single Page Applikation |
| MaterialUI | Designkomponenten |
| FontAwesome | Icons |
| Chart.js | Charts/Dashboard |
| SweetAlert2 | Modal |
| Moment.js | Vereinfachtes Handling von Date/Time |
| Azure | Hosting der Webapplikation |
| Github | Sourcecodeverwaltung |
| Travis | Continuous Integration and Deployment |
| Sonarcloud | Code Quality |

**Implementation**

Gibt es etwas Spezielles, was ihr zur Implementation erwähnen wollt?

* Intelligente Rechnungserfassung mit User Guidance

Was ist aus technischer Sicht besonders cool an eurer Lösung?

* State of the Art Frontend Applikation mit ReactJS
* Modernes Look and Feel mit MaterialUI
* Data Visualization
* Sehr einfache Dateneingabe mit intelligenten Komponenten
* Continuous Integration Pipeline mit Quality Gate (Sonarcloud)

**Abgrenzung / Offene Punkte**

Welche Abgrenzungen habt ihr bewusst vorgenommen und damit nicht implementiert?

Weshalb?

* Keine Backend Implementation: Der Fokus dieses Hackathons lag auf der Umsetzung eines funktionalen UI-Prototyps für Bänz.
* Keine Integration mit den APIs von BEKB, da dies aus rechtlichen Gründen nicht möglich ist und diese nicht zur Verfügung gestellt wurden