



# «Fründinne» «DIGITALER KUNDENDIENST DER NÄCHSTEN GENERATION»

---

Technische Informationen für die Jury



## Technische Informationen für die Jury

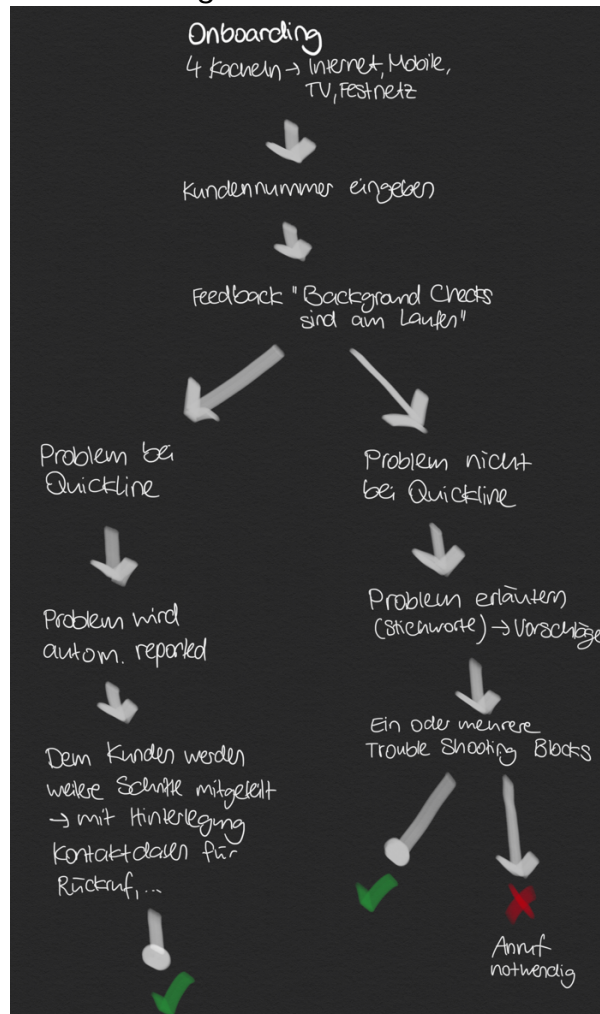
### Aktueller Stand des Sourcecodes

Repository: <https://github.com/fruendinne/bernhackt-support-app>

### Ausgangslage

- Worauf habt ihr euch fokussiert?  
Hilfreiche Anleitungen für das Lösen von Problemen sind grundsätzlich auf der Support Webseite vorhanden, allerdings werden diese oft nicht gefunden. Der Fokus unserer Lösung ist ein benutzerfreundliches Design welches möglichst relevante Suchresultate liefert.
- Welche technischen Grundsatzentscheide habt ihr gefällt?  
Unser System soll aus dem Unwissen der Menschen lernen und damit ein sich stetig verbesserndes Sucherlebnis bieten. Insbesondere wenn das Problem direkt via Google und nicht über die Firmenwebseite gesucht wird.

Wir haben folgenden User-Flow definiert:



## Technischer Aufbau

- Welche Komponenten und Frameworks habt ihr verwendet?  
Vue.js Framework, Django CMS, Vuetify Library für Komponenten, FuzzyWuzzy String Search Library für Python
- Wozu und wie werden diese eingesetzt?  
Applikation basiert auf Vue.js, Django CMS (mit Python) und Vuetify. Die FuzzyWuzzy String Search Library, ermittelt Scores für Blocks mittels Levenshtein Distanz. Die Scores werden dann verwendet um herauszufinden welcher Block als nächstes angezeigt wird.

## Implementation

- Gibt es etwas Spezielles, was ihr zur Implementation erwähnen wollt?  
Für die Umsetzung der Logik haben wir uns folgendes überlegt:

### Troubleshooting Logic Block (TLB)

Die Knowledge Base besteht aus Hilfeartikeln, FAQ, Community, usw. Insbesondere bei der Community gibt es viele Wiederholungen. Eine schnelle und effiziente Suche wird schwer für den Kunden.

Ein Troubleshooting Logic Block repräsentiert eine einzelne Diagnose- oder Lösungsmassnahme wie beispielsweise «Router Neustarten».

Durch eine verständliche Anleitung wird man in diesem Schritt begleitet. Man wird dann gefragt, ob damit das Problem gelöst wurde. Wenn nein, erhält man einen weiteren TLB, ein sogenannter «**Successor**», als Beispiel «Glasfaserkabel überprüfen». Successors sind eine Liste von möglichen Nachfolgeschritten (TLBs).

Weiter arbeiten wir mit Tags: Wie auf einem Blog, gehören diese nicht zu Lösung, sondern zu Problemen wie „Rote LED blinkt“ / „Router brennt“ / „Katze Kabel gebissen“. Tags helfen also, schneller ähnliche Probleme und damit Lösungen zu finden.

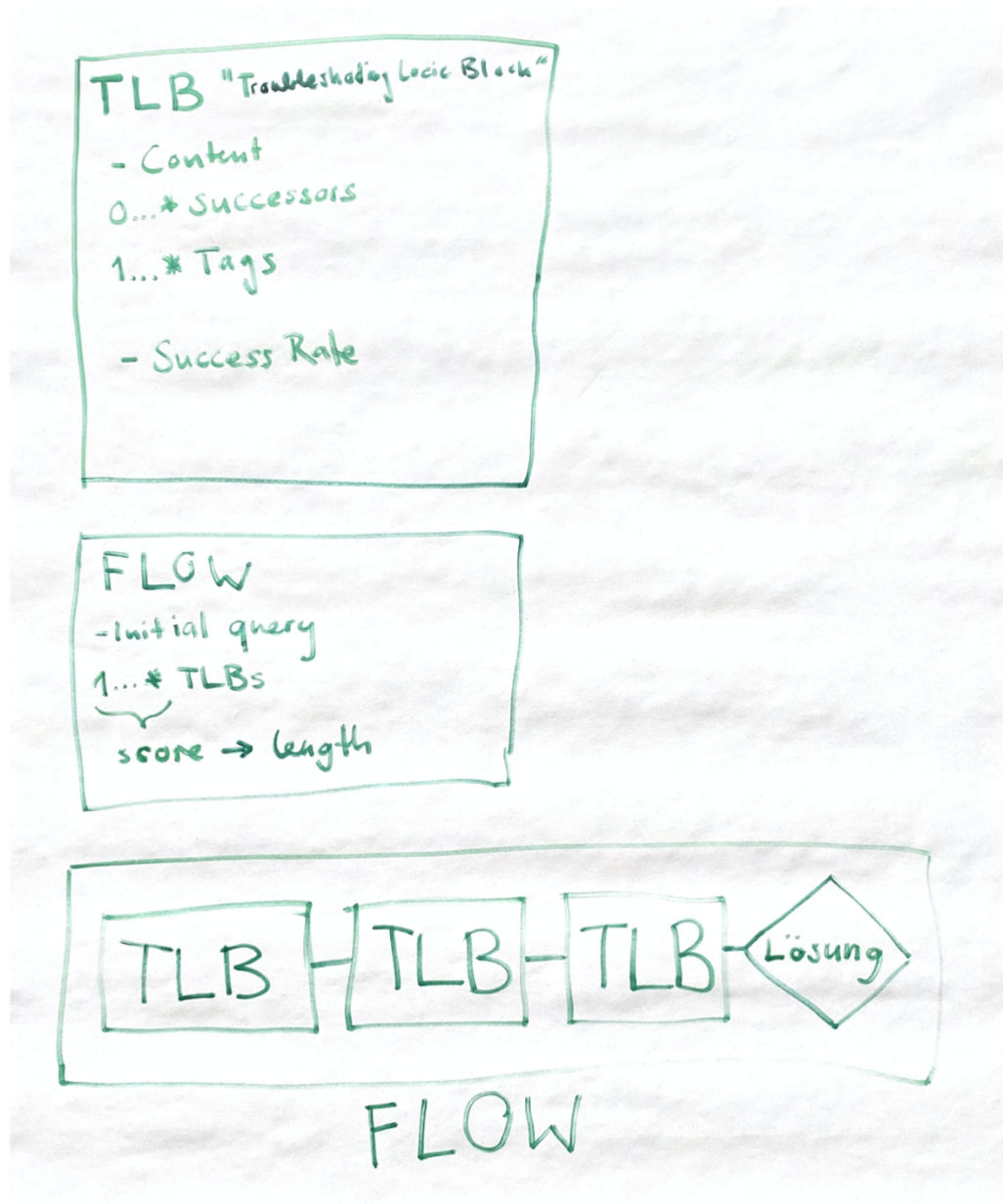
Mittels einer **Success Rate** (bezieht sich auf Content) wird die Erfolgsquote des Blocks berechnet. Das wird dann gebraucht, um für ein bestimmtes Problem jeweils genau den Inhalt zu wählen, mit dem die meisten Kunden Erfolg hatten.

Jedes Mal, wenn der User „Hat Problem nicht gelöst“ anklickt, versuchen wir, den sinnvollsten Content für den User zu finden.

Der durchlaufene Flow, also die Reihenfolge aller probierten Lösungsvorschläge, sowie die initiale Problembeschreibung werden gespeichert, egal ob erfolgreich oder nicht.

Wenn das Problem noch immer nicht gelöst werden kann, hat der Kunde dann trotzdem noch die Möglichkeit, den Support anzurufen.

Dann gibt es noch den **Flow-Score**. Das ist die Anzahl von TLB Blöcke in einem Flow. Dies ermöglicht es in Zukunft aufgrund der kürzesten Flows automatische Hilfspages zu generieren, die dann von Suchmaschinen indexiert werden.



- Was ist aus technischer Sicht besonders cool an eurer Lösung?
  - Die Suche verbessert sich mit der Zeit
  - direkter Zugang zum richtigen Hilfeartikel über Google
  - einfache Bedienung

- Matching der Hilfethemen basierend auf Natural Language Processing

### **Abgrenzung / Offene Punkte**

- Welche Abgrenzungen habt ihr bewusst vorgenommen und damit nicht implementiert? Weshalb?

Gewisse «einfachere» Dinge haben wir nicht implementiert. Dazu gehören:

- Visuelles Glossar als Hilfe in Konversation mit Hotline und im Allgemeinen für den Kunden. Dazu hätten wir ein Glossar mit Grafiken erstellt. Ein Beispiel wäre ein beschrifteter Router gewesen.
- Implementation SEO, da wichtig für Google Suche.
- Weg von PDF. Diese sind nicht benutzerfreundlich und wenn man beispielsweise kein Internet hat und nur begrenzt Handydaten, dann möchte man nicht noch ein PDF herunterladen müssen.

Ebenfalls weggelassen haben wir einen Check, wie gross das Fachwissen des Kunden ist, also wie komplex die dargestellte Dokumentation ist. Wir wollten den Kunden damit dann ursprünglich in unterschiedliche Prozesse schicken. Als ein «Noob» wäre der Kunde an der Hand genommen worden, es hätte viele Piktogramme gegeben und der Prozess wäre so einfach wie möglich gestaltet worden. Als «Techie» wäre man direkt an den Second Level Support gelangt.

Diese Features waren aus unserer Sicht zwar interessant, aber für die Challenge nicht relevant genug.

Ein weiterer wichtiger Teil des «Onboarding» Ablaufs ist die Selbstdiagnose des Systems. Diese konnten mir aufgrund fehlender Schnittstellen zu Quickline nur konzeptionell andeuten.