Random-Generator

Dein zuverlässiger Helfer bei spontan notwendigen Entscheidungsfindungen

Dennis Kiefer, Tobias Mächtel

Inhaltsverzeichnis

[1 Allgemeine Beschreibung 2](#__RefHeading___Toc824_1663482902)

[1.1 Sinn und Zweck der Anwendung 2](#__RefHeading___Toc827_1663482902)

[1.2 In Frage kommende Benutzergruppen 2](#__RefHeading___Toc829_1663482902)

[1.3 Problemstellung aus Sicht der Benutzer 2](#__RefHeading___Toc831_1663482902)

[2 Benutzeroberfläche und UI-Skizzen 3](#__RefHeading___Toc709_1233727210)

[2.1 Allgemeine Gestaltung der Benutzeroberfläche 3](#__RefHeading___Toc857_1663482902)

[2.2 Funktionsbereich 1 (Kapitel bitte sinnvoll umbenennen) 3](#__RefHeading___Toc859_1663482902)

[2.3 Funktionsbereich 2 (Kapitel bitte sinnvoll umbenennen) 3](#__RefHeading___Toc837_16634829021)

[2.4 Funktionsbereich 3 (Kapitel bitte sinnvoll umbenennen) 3](#__RefHeading___Toc839_16634829021)

[3 Funktionale Anforderungen 4](#__RefHeading___Toc833_1663482902)

[3.1 Funktionsbereich 1 (Kapitel bitte sinnvoll umbenennen) 4](#__RefHeading___Toc835_1663482902)

[3.2 Funktionsbereich 2 (Kapitel bitte sinnvoll umbenennen) 4](#__RefHeading___Toc837_1663482902)

[3.3 Funktionsbereich 3 (Kapitel bitte sinnvoll umbenennen) 4](#__RefHeading___Toc839_1663482902)

[4 Technologieauswahl 5](#__RefHeading___Toc292_1663482902)

[4.1 Art der Webanwendung 5](#__RefHeading___Toc861_1663482902)

[4.2 Serverseitige Programmiersprachen und Frameworks 5](#__RefHeading___Toc863_1663482902)

[4.3 Clientseitige Programmiersprachen und Frameworks 5](#__RefHeading___Toc865_1663482902)

# Allgemeine Beschreibung

## Sinn und Zweck der Anwendung

Jeder kennt es: Man sitzt mit seinen Freunden zuhause in einer gemütlichen Runde zusammen und bekommt nach dem ein oder anderen „Erfrischungsgetränk“ Heißhunger auf eine Pizza. Doch wer meldet sich freiwillig zur Bestellung und zur Abholung? Richtig, niemand! In diesen Fällen ist es häufig üblich, durch irgendwelche Spiele, wie zum Beispiel „Schere-Stein-Papier“, dem Verlierer den „Pizza-Dienst“ aufzudrücken. Das Ganze ist allerdings nicht nur an pizzaholen gebunden. Solche spontanen Entscheidungen

## In Frage kommende Benutzergruppen

Für welche Art von Benutzern ist die Anwendung gedacht? Bitte alle Arten von Benutzern aufzählen, die es geben wird. Zum Beispiel bei einer Anwendung zur Erfassung von Urlaubsanträgen: Mitarbeiter, Vorgesetzte, Administratoren.

Zu jeder Benutzergruppe beschreibt bitte, welches relevante Fachwissen sie besitzen, wie sicher sie im Umgang mit der IT sind und alles, was man sonst noch über sie wissen muss, um eine für sie passende Anwendung zu erstellen.

## Problemstellung aus Sicht der Benutzer

Hier zählt bitte für jede Benutzergruppe auf, welche Aufgaben bzw. Probleme sie mit der Anwendung lösen will. Zum Beispiel bei unseren Urlaubsanträgen:

### Mitarbeiter

* Einen neuen Urlaubsantrag stellen
* Schnell erkennen, ob ein Antrag angenommen oder abgelehnt wurde
* Nachvollziehen, warum ein Antrag abgelehnt wurde
* Eine Übersicht mit allen Urlaubsanträgen aufrufen

### Vorgesetzte

* Eine Übersicht mit allen zu genehmigten Anträgen aufrufen
* Einen Antrag annehmen oder ablehnen
* Eine Begründung zur getroffenen Entscheidung dokumentieren

### Administratoren

* Urlaubsanträge anhand frei definierbarer Kriterien suchen
* Fehlerhafte Urlaubsanträge nachbearbeiten und abändern
* Fehlerhafte Urlaubsanträge löschen
* Die Urlaubsanträge vergangener Jahre löschen

Gerne könnt ihr hier auch beschreiben, welche Anforderungen eure Anwendung nicht erfüllen muss. :-)

# Benutzeroberfläche und UI-Skizzen

Fügt hier Skizzen der geplanten Anwendung ein und beschreibt, wie die Anwendung aussehen und gegliedert sein soll. Ihr könnt die Skizzen entweder von Hand erstellen und abfotografieren oder Anwendungen wie [https://draw.io](https://draw.io/) oder Inkscape hierfür verwenden.

## Allgemeine Gestaltung der Benutzeroberfläche

## Funktionsbereich 1 (Kapitel bitte sinnvoll umbenennen)

## Funktionsbereich 2 (Kapitel bitte sinnvoll umbenennen)

## Funktionsbereich 3 (Kapitel bitte sinnvoll umbenennen)

# Funktionale Anforderungen

## Funktionsbereich 1 (Kapitel bitte sinnvoll umbenennen)

Hier müsst ihr alle Anforderungen beschreiben, die die Anwendung erfüllen muss. Am Besten einfach als Aufzählung, die ihr bei der Programmierung und beim Testen dann Schritt für Schritt abarbeiten könnt. Achtet deshalb darauf, wirklich nichts zu vergessen und auch scheinbar unwichtige Details aufzuzählen. Denn, wenn es hier nicht dokumentiert ist, wird es sehr wahrscheinlich auch nicht umgesetzt. :-) Beispiel:

* Wenn sich ein Mitarbeiter einloggt, soll er eine Übersicht mit all seinen Urlaubsanträgen sehen.
* Gibt es noch keine Urlaubsanträge, soll stattdessen die Meldung „Mensch, du hast ja noch gar keine Urlaubsanträge gestellt. Gönne Dir doch einmal Urlaub und stelle gleich einen Antrag!“ erscheinen.
* Auf der Übersichtsseite soll daher immer ein Button „Antrag stellen“ sichtbar sein. Klickt der Anwender auf den Button, erscheint eine neue Seite mit dem Erfassungsformular.
* Klickt der Anwender stattdessen auf einen vorhandenen Antrag, soll dasselbe Formular erscheinen, jedoch nur im Anzeigemodus. Die Eingabefelder soll also nicht eingabebereit sein.
* Mitarbeiter mit Personalverantwortung sollen zusätzlich einen Button mit der Beschriftung „Zu genehmigende Anträge (x)“ sehen, wobei „x“ die Anzahl der zu genehmigenden Anträge ist.
* Und so weiter und so fort. :-)

## Funktionsbereich 2 (Kapitel bitte sinnvoll umbenennen)

## Funktionsbereich 3 (Kapitel bitte sinnvoll umbenennen)

# Technologieauswahl

## Art der Webanwendung

Hier bitte auswählen und begründen, was für eine Art von Webanwendung erstellt werden soll:

* Komplett browserseitig / Single Page App
* Komplett serverseitig
* Browserseitige Anwendung mit API-Methoden auf dem Server

Bitte nicht vergessen, die Auswahl zu begründen. Zwar lassen sich alle Anforderungen mit jeder Art von Webanwendung irgendwie realisieren. Aber warum, habt ihr euch gerade für die eine und nicht für eine andere Art entschieden? Gibt es nicht-funktionale Anforderungen, die dafür sprechen, wie zum Beipsiel „Wir wollen später eine öffentliche API für unsere Kunden bereitstellen“ oder „Die Anwendung soll möglichst einfach zu deployen sein“ und so weiter? Bevorzugt ihr einen bestimmten Programmierstil, Programmiersprachen, Frameworks? Kennt ihr euch in eurer Gruppe mit bestimmten Technologien besser aus als mit anderen? Wollt ihr mal was neues probieren? Seit kreativ und nennt uns eure Gründe. :-)

## Serverseitige Technologien

Falls die Webanwendung serverseitige Funktionalitäten besitzt, welche Programmiersprachen und Frameworks plant ihr, hierfür zu einzusetzen. Warum wollt ihr sie nutzen? Hierunter fallen auch öffentliche APIs und Cloud-Dienste von Drittanbietern, die ihr zwar nicht selbst programmieren, in Ihrer Anwendung aber trotzdem nutzen wollt.

## Clientseitige Technologien

Eure Webanwendung wird auf jeden Fall ein Frontend haben, denn sonst würden wir ja gar nichts auf dem Bildschirm sehen. :-) Welche Technologien wollt ihr im Frontend einsetzen? Gutes altes HTML, CSS und JavaScript ohne weitere Hilfsmittel? Frameworks wie Bootstrap? CSS-Preprozessoren wie Less oder Sass? Die Paketverwaltung npm? Es ist alles erlaubt. Erzählt davon und erklärt, welche Vorteile ihr euch jeweils daraus versprecht.