

# Resource-Genius

Die Zielsetzung des Softwareprojekts **Resource-Genius** ist die Entwicklung einer funktionsfähigen Anwendung zur Verwaltung und Verrechnung von Schulungen und Räumen, die für einen Tagessatz gebucht werden können. Die Applikation soll in Front- und Backend aufgeteilt werden und auf Container-Technologie deployed werden.

## Funktionale Anforderungen

---

Für die nachfolgenden Domains sollen einfache, intuitive Eingabemöglichkeiten erstellt werden. Wenn die Eingabe von Daten abhängig ist, die aus anderen Domains stammen, muss eine automatische Eingabevervollständigung oder eine Auswahlliste zur Verfügung gestellt werden. Eingaben von Zeitwerten (Datum/Uhrzeit) müssen durch Date-Picker erfolgen, hierbei ist auf die richtige lokale Darstellung der Werte zu achten.

### Kundenverwaltung

Die Applikation muss Kunden verwalten können, die Attribute des Kunden sind, "ID", "Name", "Abteilung", "Firma", und "Solution ID". Die Kundenverwaltung enthält keine personenbezogene Daten.

### Ressourcen (Schulungen und Räume)

Die Ressourcen (Schulungen und Räume) haben die Attribute, "ID", "Name", "Beschreibung", "Adresse", "E-Mail" und "Kosten pro Tag".

### Verrechnung

Für die Verrechnung müssen folgende Daten erfasst werden, ID, Datum, Ressource(Schulung bzw. Raum), Kunde, Anzahl der Teilnehmer und ein Flag ob die Buchung verrechnungsrelevant ist.

Die Verrechnung erfolgt nach Kunde getrennt. Es können somit Teilnehmer mehrere Kunden eine Ressource (Schulung bzw. Raum) nutzen.

Beispiel: Schulung: 'SQL Basis Schulung', Kosten: 700,00 € pro Tag

Kunde 1: Turbo-SI, 6 Teilnehmer Kunde 2: Turbo-AE, 6 Teilnehmer Kunde 3: BC-Kunde, 4

## Teilnehmer

ID	Datum	Ressource	Kunde	Teilnehmer	rechnungsrelevant
1000	14.04.2023	SQL Basis Schulung	Turbo-AE	6	Nein
1001	14.04.2023	SQL Basis Schulung	Turbo-SI	6	Nein
1002	14.04.2023	SQL Basis Schulung	BC-Kunde	4	Ja

## Generierung von Reports

Aus den erfassten Verrechnungsdaten sollen Reports erstellt werden. Es wird ein Report benötigt um alle verrechnungsrelevanten Daten eines auswählbaren Zeitraums zu generieren, der als Grundlage für eine Rechnungsstellung verwendet werden kann.

Um die Reports zu generieren, wird die folgende Kalkulation der Kosten verwendet.

Kosten pro Ressource / Anzahl der Teilnehmer-Gesamt x Anzahl der Teilnehmer-Kunden.

Am Beispiel von oben:

Kosten pro Teilnehmer: 700,00 € / 16 = 43,75 €

Kosten Turbo-AE: 43,75 € x 6 = 262,50 €

Kosten Turbo-SI: 43,75 € x 6 = 262,50 €

Kosten BC-Kunde: 43,75 € x 4 = 175,00 €

Das Flag 'Rechnungsrelevant' wird zur Erstellung der verrechnungsrelevanten Reports verwendet. Nur Positionen, bei denen dieses Flag gesetzt ist, werden ausgewiesen. Im Beispiel von oben wird nur eine Rechnung für BC-Kunde von 175,00 € erstellt.

Weitere Reports dienen der Dokumentation und sollen nach den folgenden Gesichtspunkten erstellt werden:

- Anzahl Buchungen eines Zeitraums
- Anzahl Teilnehmer eines Zeitraums
- Gesamtkosten eines Zeitraums

## Technische Anforderungen

---

## Generierung von IDs

Bei allen Domainobjekten wie "Kunden", "Schulungen/Räume" und "Verrechnung" wird die "ID" von der Anwendung generiert.

## Nichtfunktionale Anforderungen

---

### Benutzerfreundlichkeit

Die Anwendung soll einfach und intuitiv zu bedienen sein, um den Benutzern ein positives Benutzererlebnis zu bieten.

Es ist wichtig, ein durchgängiges, responsives Design zu verwenden. Die Anwendung muss sowohl auf Desktop- als auch auf mobilen Endgeräten, wie Handys und Tablets, bedienbar sein.

### Skalierbarkeit

Die Anwendung sollte skalierbar sein und mehrere Benutzer und große Datenmengen verwalten können. Die Anwendung soll auf der Basis von Microservices aufgebaut sein.

### Sicherheit

Die Anwendung muss sicher sein und Schutzmechanismen haben, um unautorisierten Zugriff und Datenverlust zu verhindern. Die Übertragung von Daten zwischen Client und Server muss abgesichert sein.

### Leistung

Die Anwendung sollte schnell und reaktionsschnell sein, um ein effizientes Arbeiten zu ermöglichen.

### Wartbarkeit

Die Anwendung sollte wartbar sein, um zukünftige Änderungen oder Erweiterungen einfach zu implementieren.

Ein Buchungsmodul für die Ressourcen (Schulungen und Räume) soll später als Erweiterung einfach in das System integriert werden können.

### Verfügbarkeit

Die Anwendung sollte jederzeit verfügbar sein, um die Geschäftskontinuität sicherzustellen.

## **Technologien**

Die Auswahl der Technologien bleibt den Entwicklern vorbehalten. Das Backend könnte mit Spring Boot und das Frontend mit einem Javascript Framework wie Svelte, Vue.js oder React entwickelt werden.

## **Deployment**

Frontend und Backend müssen mit Container-Technologie deployed werden.

## **Dokumentation und Präsentation**

Neben der Erstellung der funktionsfähigen Applikation, ist eine Dokumentation und eine Präsentation des Projekts erforderlich. Das Team stellt die entwickelte Anwendung am letzten Tag der Projektwochen, in einer Präsentation, vor. Die Präsentation wird durch alle Team Mitglieder gestaltet. Jedes Teammitglied stellt seinen Teil vor, den er oder sie übernommen hat.

## **Versionsverwaltung**

Alle Sourcen, Dokumentationen und Präsentation sind in einem GIT Repository zu verwalten. Die einzelnen Tasks sind als Feature-Branch von dem Entwickler in das Repository einzupflegen, dass nachvollziehbar ist wer welchen Task umgesetzt hat.