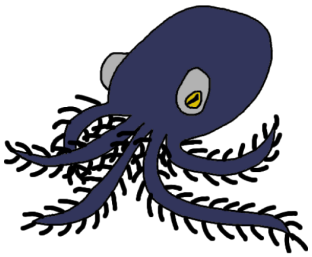
intension

DHBW
Duale Hochschule
Baden-Württemberg

Featherkraken: Bestpreissuche für Flugangebote mit variablen Abflughäfen



STUDIENARBEIT

des Studienganges Informatik
an der Dualen Hochschule Baden-Württemberg Stuttgart
von
Ingo Kuba

Matrikelnummer, Kurs	place, holder
Ausbildungsfirma	intension GmbH
Betreuer	Place Holder

Erklärung zur Eigenleistung

Hiermit erkläre ich, dass ich die vorliegende Studienarbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe.
Die Stellen der Studienarbeit, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind durch Angaben der Herkunft kenntlich gemacht. Dies gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen sowie für Quellen aus dem Internet.

Ostfildern, den 26. Mai 2020 _____

Zusammenfassung

Inhaltsverzeichnis

Abbildungsverzeichnis

Akronyme

ERM Entity Relationship Model. 1–4

UML Unified Markup Language. 2

1 Einleitung

1.1 Motivation

In der Regel möchte ein Fluggast den günstigsten Preis für eine bestimmte Route A nach B. Jede Flugsuchmaschine im Internet bietet diese Feature. Manchmal sucht ein Fluggast auch einfach nach Inspiration und möchte Angebote von A nach X, wobei X variabel ist. Einige Suchmaschinen bieten diese Suche bereits an. Worum es in dieser Studienarbeit geht, ist der umgekehrte Fall: X nach B. Also von welchem beliebigen Flughafen kommt man möglich günstig an ein festes Ziel. Gerade auf hochpreisigen Strecken kann es sich lohnen einen Umweg zu fliegen.

1.2 Aufgabenstellung

Bei der Suche sollen die klassischen Filterkriterien implementiert werden. Das heißt die Unterscheidung ob man nur einen Hinflug oder Hin- und Rückflug buchen möchte. Des weiteren soll man jeweils ein Datum für An- und Abreise festlegen können, welches um drei Tage flexibel sein soll. Neben der Buchungsklasse (Economy, Business, First Class) soll auch die Wahl der Airline oder Allianz eingeschränkt werden können. Außerdem soll man Passagier- und Umsteigeanzahl wählen können.
Zusätzlich soll ein Entfernungsfilter um einen möglichen Abflughafen bereitgestellt werden. Zum Beispiel wird nur nach Angeboten gesucht, bei dem sich der Startflughafen maximal 800km (Entfernungsfilter) vom Flughafen Stuttgart (möglicher Abflughafen) entfernt befindet.
Diese Flugsuchmaschine soll über ein Web-Frontend vom Nutzer bedient werden können. (Mustermann, 2020)

2 Grundlagen

2.1 Entity Relationship Model

Um das Datenmodell der Anwendung darzustellen wurde eine vereinfachte Variante des Entity Relationship Model (ERM) verwendet.

Entities

Ein Objekt oder Entity wird in einem Rechteck dargestellt und kann Attribute besitzen, wobei komplexe Attribute als Beziehungen zu anderen Objekten dargestellt werden. Der Name der Beziehung entspricht hierbei dem Attributnamen im Code. Die Anzahl der möglichen Relationen wird in UML-Notation angegeben. Zum Beispiel ist in Abbildung 1 zu sehen, dass eine Person null bis n Autos besitzen kann.

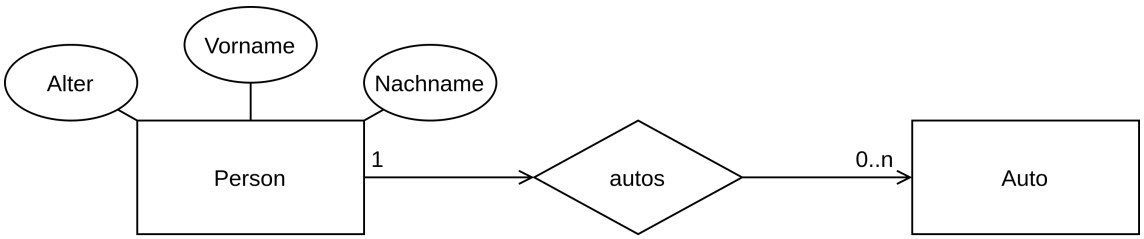


Abbildung 1: Beispiel für eine Entity mit Attributen und einer Beziehung

Attribute

Attribute können dabei eindeutig, optional oder mehrwertig sein. Die Unterscheidung zwischen Datum, Zahl oder Zeichenkette wird in einem Entity Relationship Model nicht dargestellt.

Abbildung 2: Attribute (v.l.): eindeutig, optional und mehrwertig

3 Entwurf

3.1 Auswahl der externen API

Um Flugdaten zu erhalten muss eine externe Schnittstelle benutzt werden, welche die Pflichtanforderungen erfüllt und dabei leicht zu verwenden ist. Die Wahl der Schnittstelle fiel dabei auf eine Rest-API, da diese sehr einfach zu benutzen sind. Für die Auswahl des Anbieters wurde eine Tabelle[3] erstellt, welche die Pflichtanforderungen mit den Funktionalitäten der jeweiligen Schnittstelle abgleicht. Dabei erfüllte die API von Kiwi nicht nur alle Anforderungen, sondern war auch sehr gut dokumentiert und mit Beispielen beschrieben. Des Weiteren bot Kiwi auch eine Rest-API um Flughäfen im Umkreis gegebener Koordinaten zu finden, was der Aufgabe entgegen kam.

	Skyscanner	Hipmunk	Kajak	Flight Data	Flight Bookings	Kiwi Flights
Single flight	yes	yes	yes	yes	yes	yes
Two directions	no	no	no	yes	no	yes
Specific date	yes	yes	yes	yes	yes	yes
Flexible date	no	limited	limited	yes	no	yes
Class	yes	yes	yes	limited	yes	yes
Passengers	yes	yes	yes	no	yes	yes
Airline	yes	no	no	no	no	yes
Stops	no	no	no	no	no	yes

Abbildung 3: Tabelle zum Vergleich der APIs

3.2 Datenmodell

Das Entity Relationship Model für das Datenmodell wurde hier aufgeteilt in Suchanfrage (SearchRequest) und Antwort der Anwendung (SearchResult).

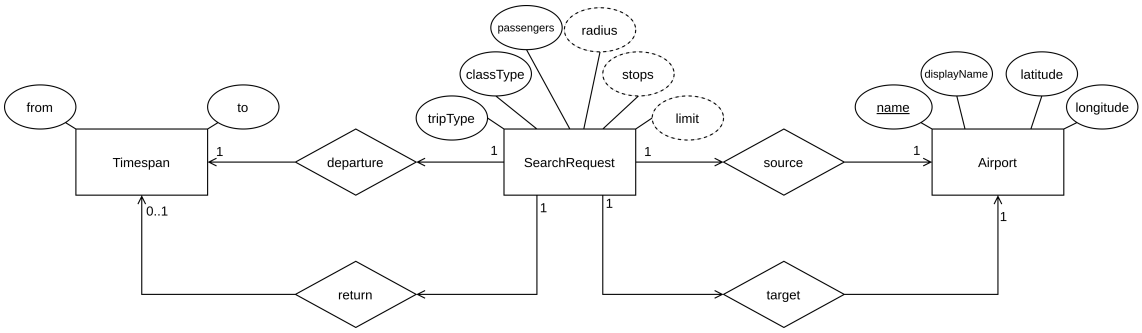


Abbildung 4: Entity Relationship Model des SearchRequest Objekts

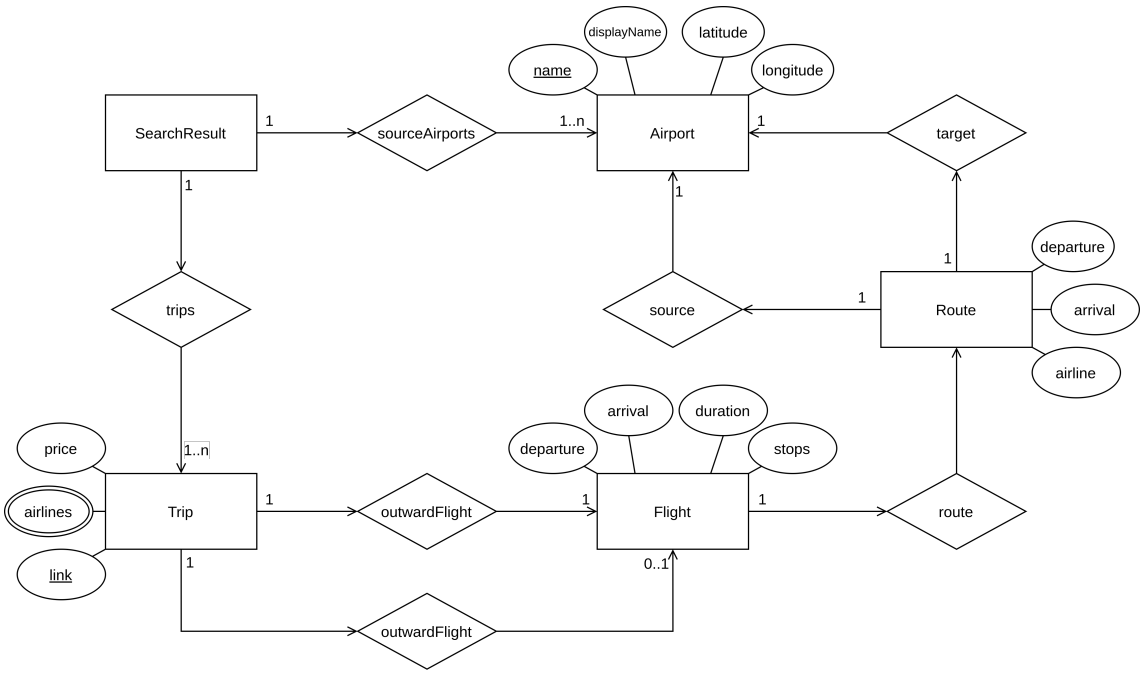


Abbildung 5: Entity Relationship Model des SearchResult Objekts

3.3 Framework

Auswahl des Frameworks für Backend und Frontend

4 Implementierung

Wie portabel ist das System? -> Docker, React -> static html+js

5 Testing

Unit tests? How to test external api? Aspects of an API: speed? maybe Beständigkeit in der Speed?

6 Zusammenfassung

6.1 Ausblick

-> Monitor speed of my requests!!

Literatur

Mustermann, M. (2020). *Beispielhafte Quelle.* Zugriff auf http://www.example.com/