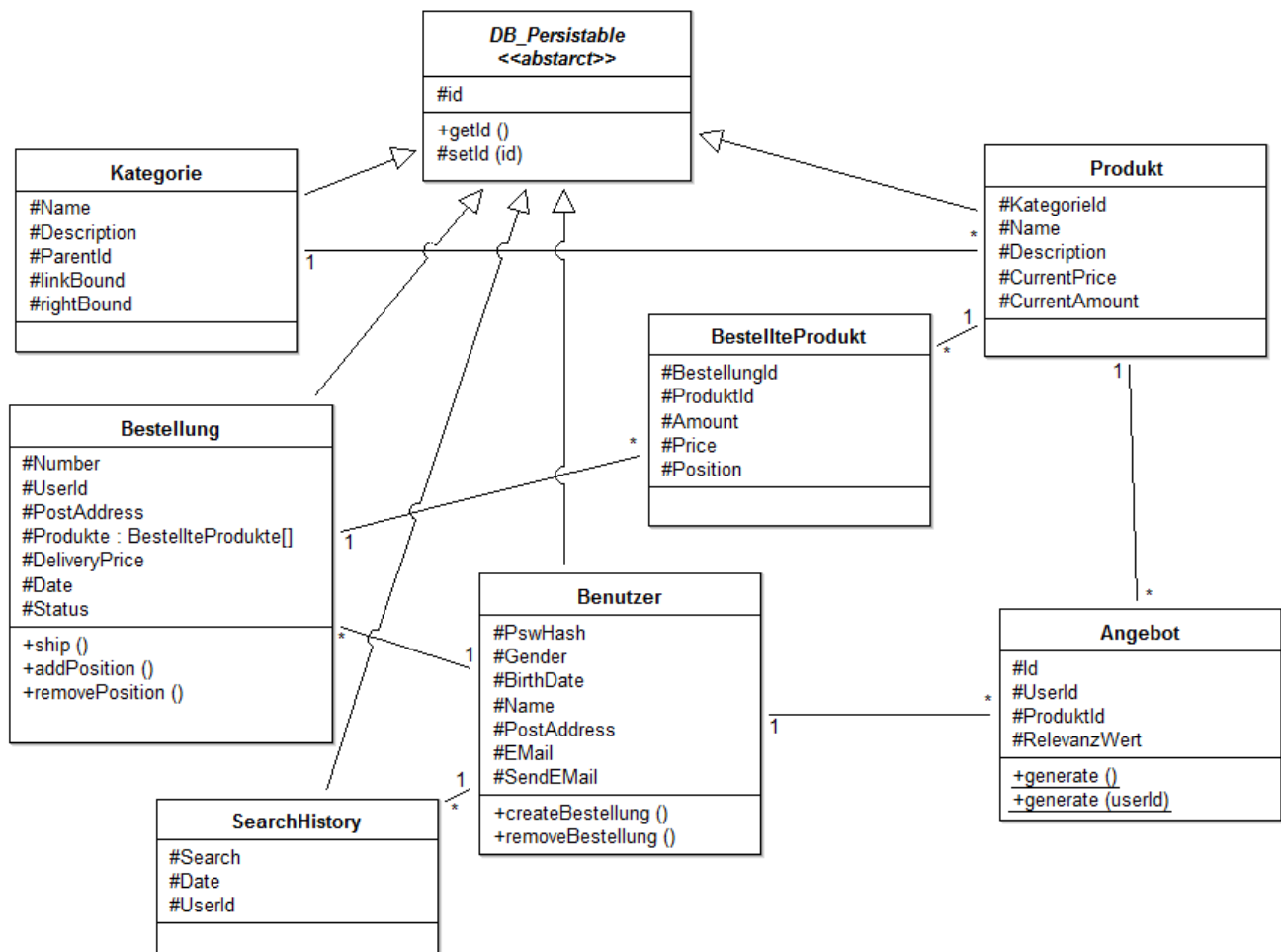


Textuelle Beschreibung - Online-Shop «La Cucaracha»

Ich habe mich für das Projekt «Online-Shop» entschieden. Im Shop können die Kunden nach den Produkten suchen und sie bestellen. Außerdem analysiert der Shop die Aktivitäten von Kunden und stellt passende Angebote auf der Basis von Ihren Interessen.

Der Shop speichert die Daten über ihre Produkte (starke Entität), jedes Produkt hat eine Kategorie. Kategorien sind hierarchisch aufgebaut (starke Entität mit unärer Beziehung). Die Kunden und ihre Bestellungen werden auch gespeichert (2 starke Entität). Der Shop hat noch eine Entität, in der die potenziellen Produkte für die Kunden gespeichert werden (schwache Entität). ER-Diagramm befindet sich am Ende de Dokuments.

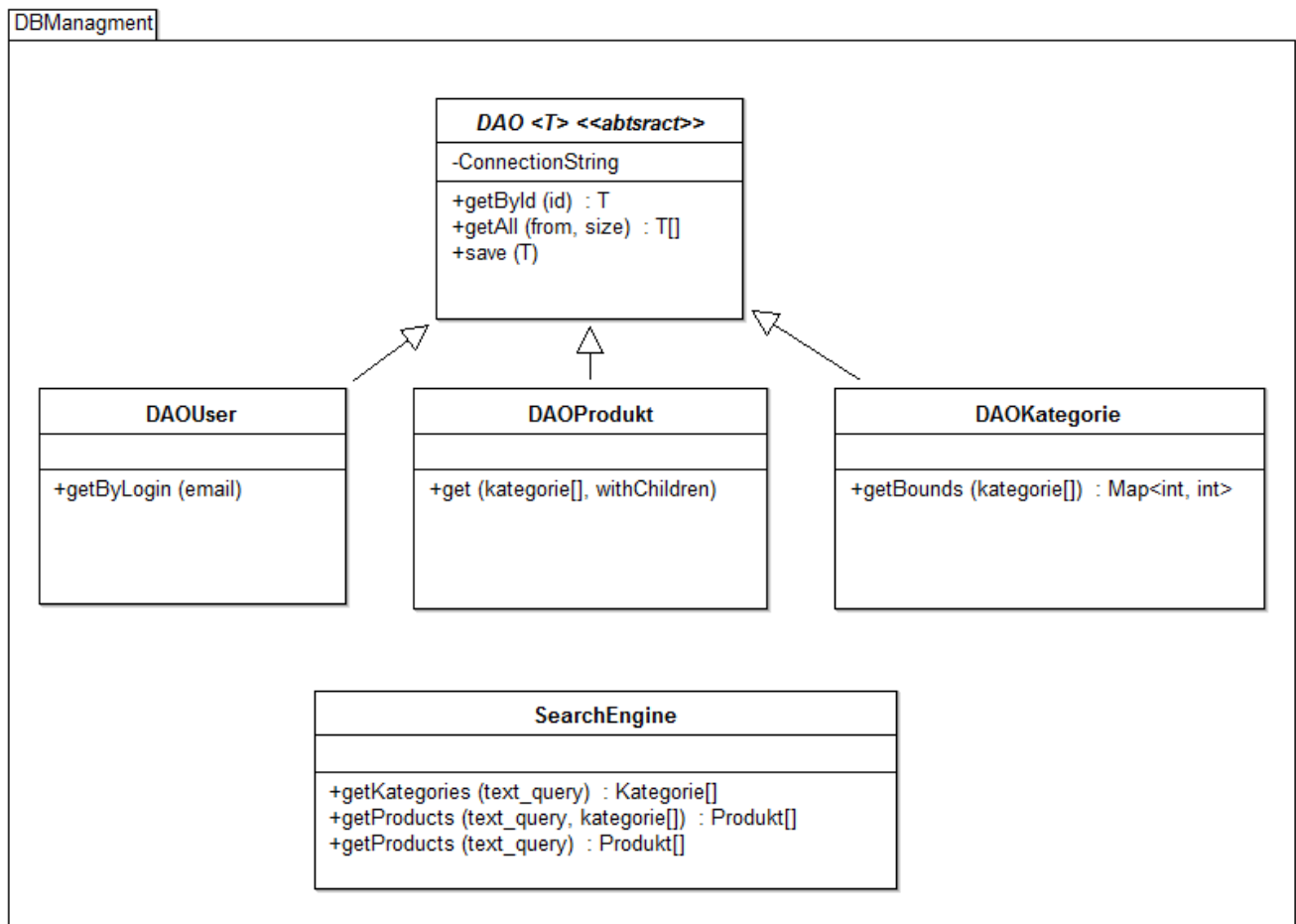
Klassendiagramm



Auf der Klassendiagramme sind Hauptelemente des Systems dargestellt. Vielleicht wird die Klasse «Angebot» nicht im Java implementiert, weil sie nur auf der DBS-Ebene benutzt wird. Andere Klassen haben notwendige Attribute. Als UserLogin wird «Email» des Benutzers verwendet.

Momentan ist es nicht abgeschlossen, wie die Daten im DBS gespeichert werden. Es kann «Hibernate» sein oder selbst-entwickelte Klassen. Unten ist eine selbst-entwickelte vorläufige Variante dargestellt (die obenstehende abstrakte Klasse «DB_Persistable» gehört auch zu dieser Variante).

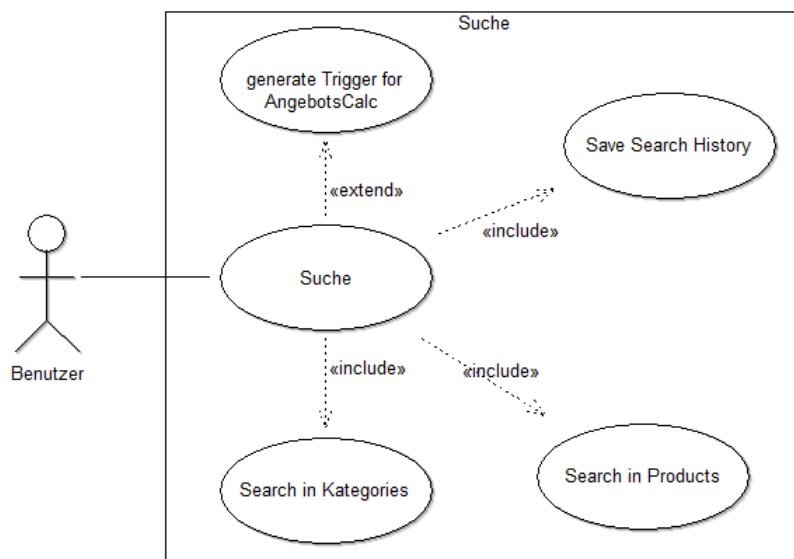
Die Suche nach dem Text wird höchstwahrscheinlich mit dem Full-Text-Search auf der DBS-Ebene implementiert. Aber die Variante mit dem Text-Index (bei den Felder «Name», «Description» der Tabellen «Produkt», «Kategorie») ist auch nicht ausgeschlossen.

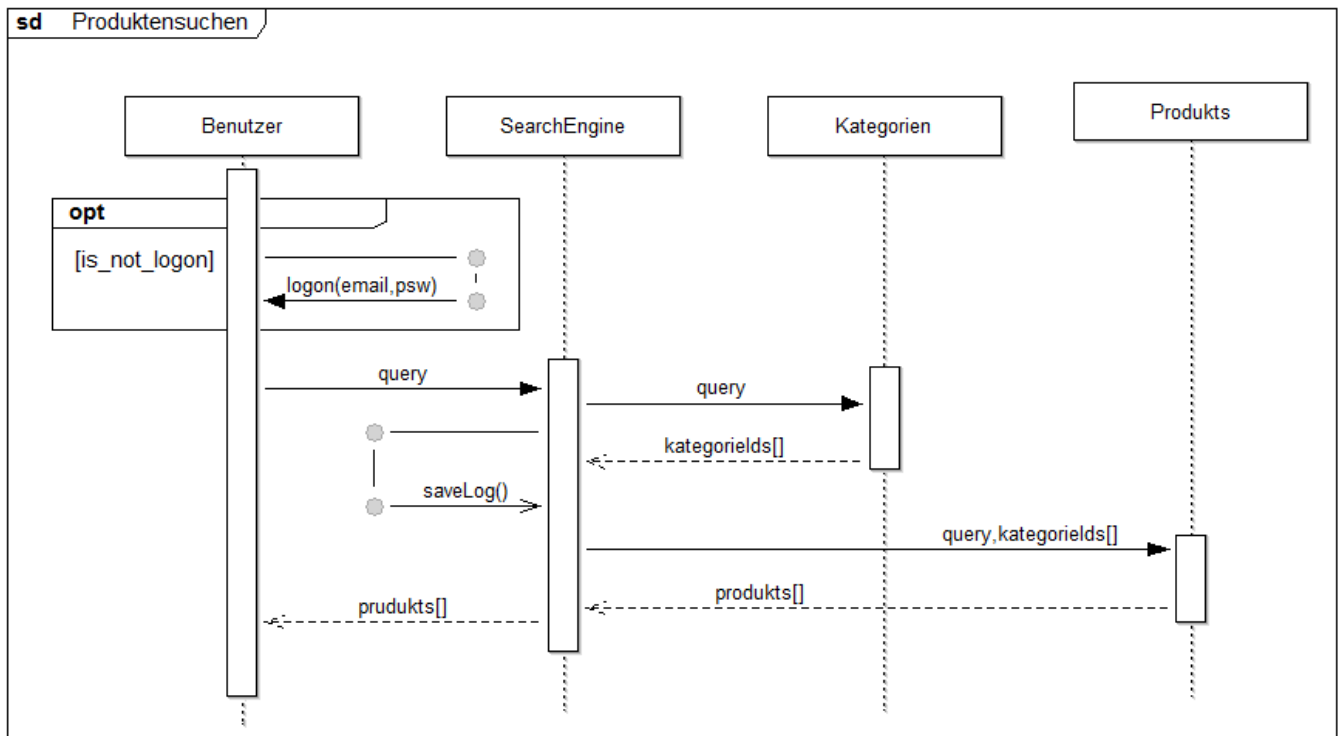


Use-Case und Verhaltensdiagramme

1. Suche nach einem Produkt

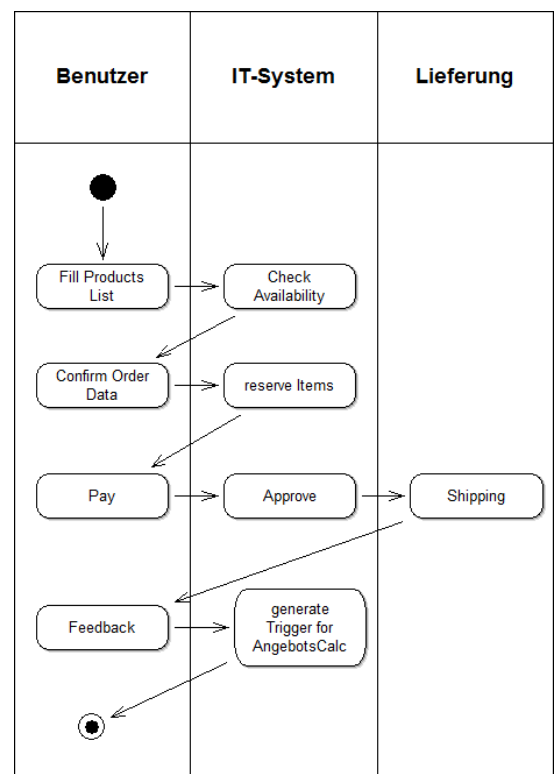
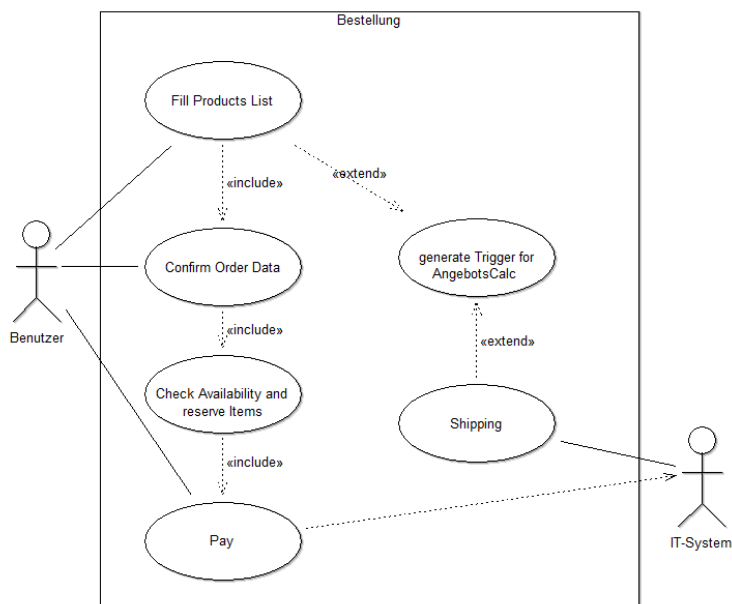
Es gibt zwei Entitys, die nach dem Text-Suchen gebraucht werden: «Produkt» und «Kategorie». Der Benutzer gibt einen Text ein. Wenn die Kategorien gefunden werden, dann sind die Produkte von diesen und unterstehenden Kategorien während der Textsuche höher bewertet (Text-Suche Relevanz-Wert).



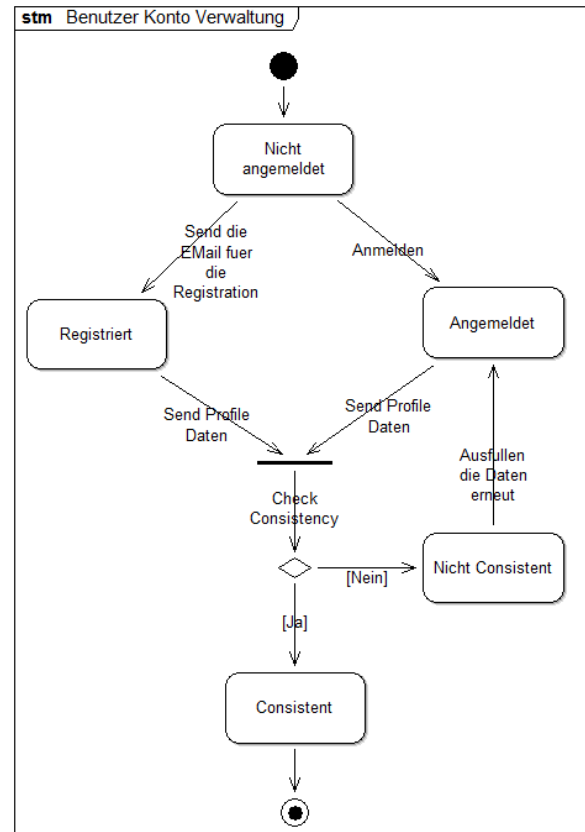
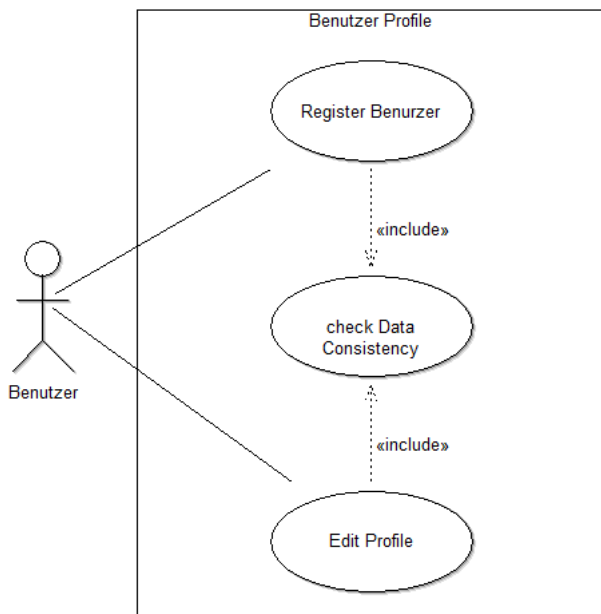


2. Bestellung vom Produkt

Die Pay-Funktion wird nicht implementiert werden. Shipping heißt nur die Message-Übergabe zur einer anderen Abteilung (auch nicht implementiert wird).

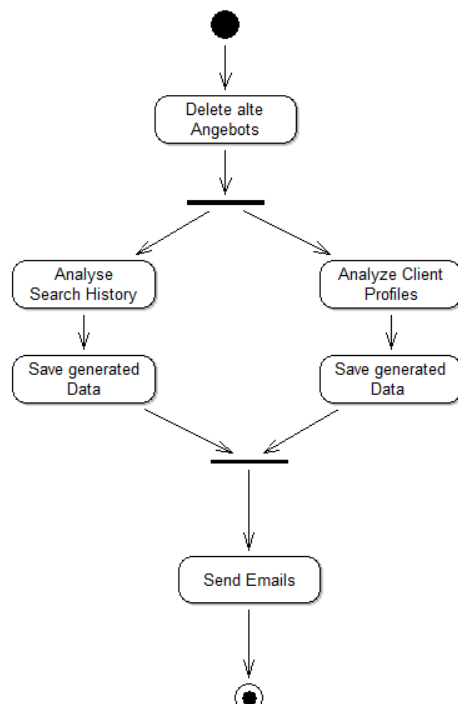
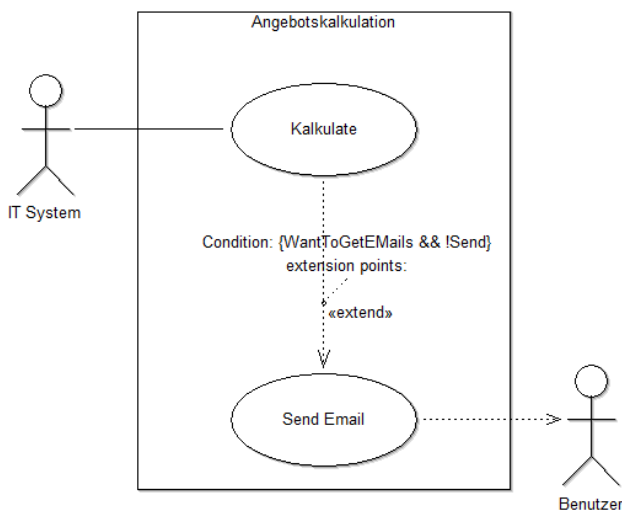


3. Benutzer Profil Verwaltung

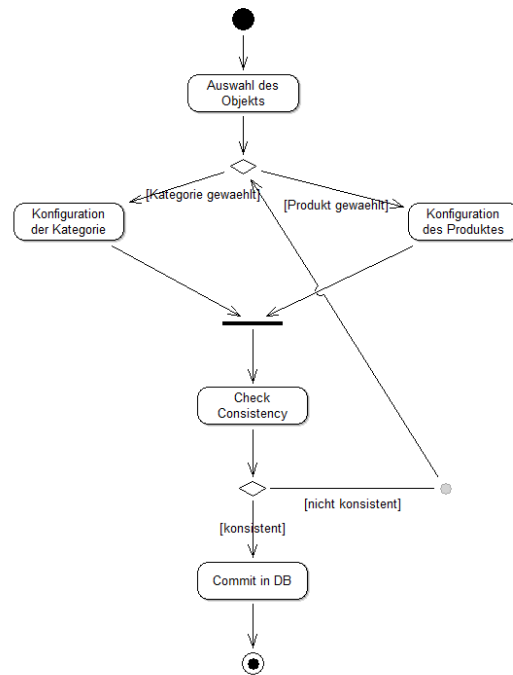
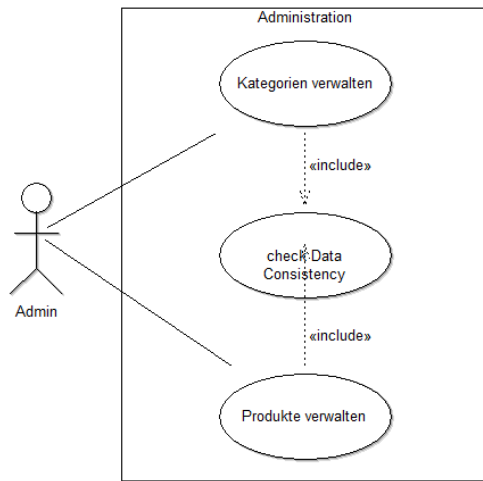


4. Die Generation von (potenziellen) Angeboten

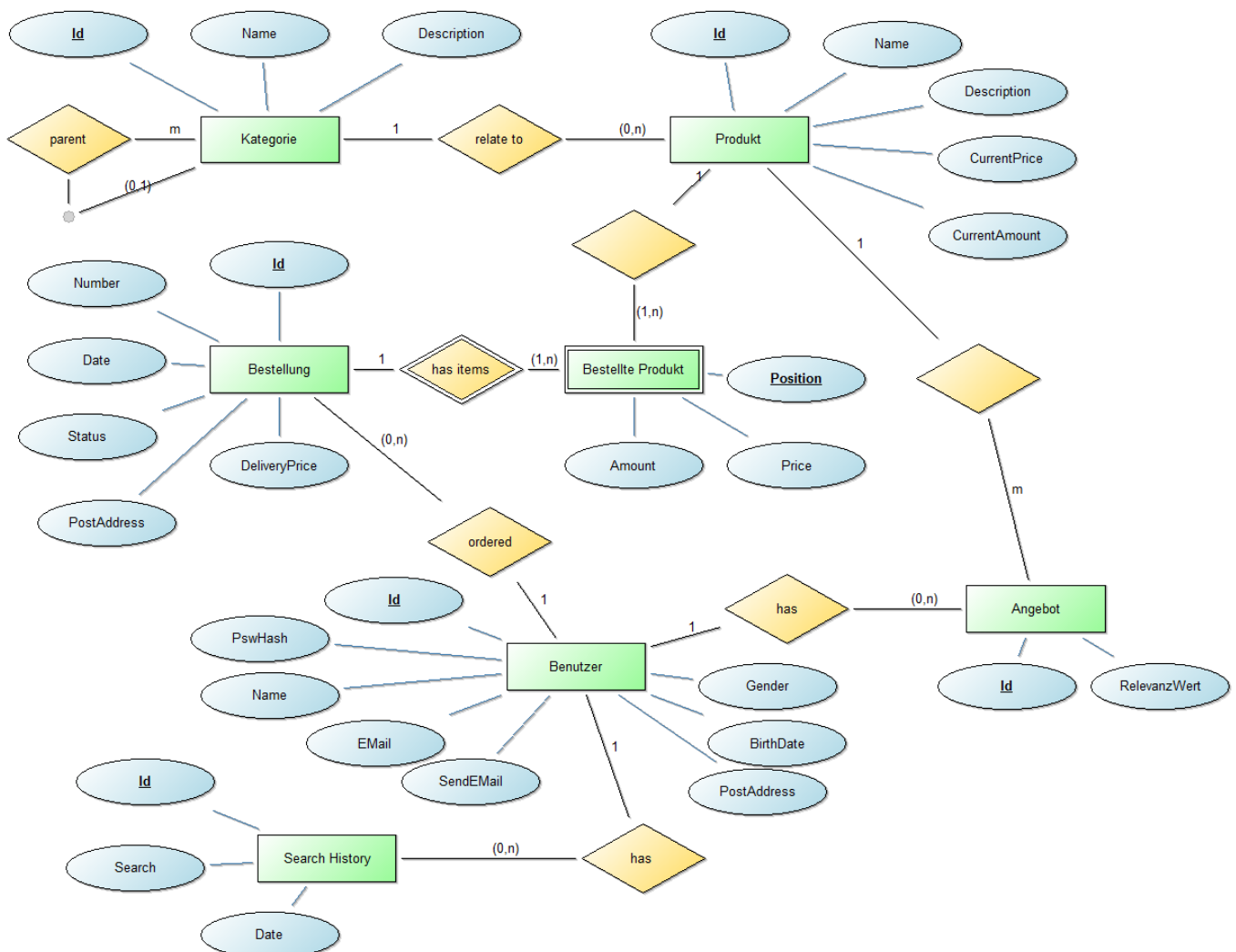
Der Akteur in diesem Use-Case ist IT-System. Es gibt einen Job, der täglich durchgeführt wird. Er kalkuliert potenziell interessante Angebote für die Kunden. Wenn die Kunden diese Angebote per Email bekommen wollen, dann schickt der Job Emails. Potenzielle Angebote werden auf der DBS-Ebene kalkuliert. Dafür wird die Rücksicht auf die Benutzerdaten, ihre Bestellungen und Suchinteresse genommen.



5. Die Shop-Daten-Verwaltung



Datenmodell - ER-Model



Relationales Schema

Im Schema wurden zwei Felder «left», «right» hinzugefügt. Sie dienen als Zusatzfelder beim Model «Nested Sets» (https://de.wikipedia.org/wiki/Nested_Sets). Das Model wird für die Hierarchiesuche im Baum (Kategorien) benutzt.

Kategorie (Id, *ParentId*, Name, Description, left, right)

Produkt (Id, Name, Description, CurrentPrice, CurrentAmount, *KategorieId*)

Benutzer (Id, Email, PswHash, SendEMail, Name, Gender, BirthDate, PostAddress)

SearchHistory (Id, Search, Date, *BenutzerId*)

Angebot(Id, RelevanzWert, *BenutzerId*, *ProduktId*)

Bestellung (Id, Number, Date, PostAddress, DeliveryPrice, Status, *BenutzerId*)

BestellteProdukt (*BestellungId*, Position, *ProduktId*, Price, Amount)