Custom Relationship Managementsystem (CRM)

Wir wollen uns ein einfaches CRM bauen. Ziel der Applikation ist es, Kundendaten, und zu jedem Kunden eine Liste von Gesprächsnotizen zu verwalten. Diese Daten sollen in einem Webbrowser angezeigt und verwaltet werden können.

Analyse

UseCase

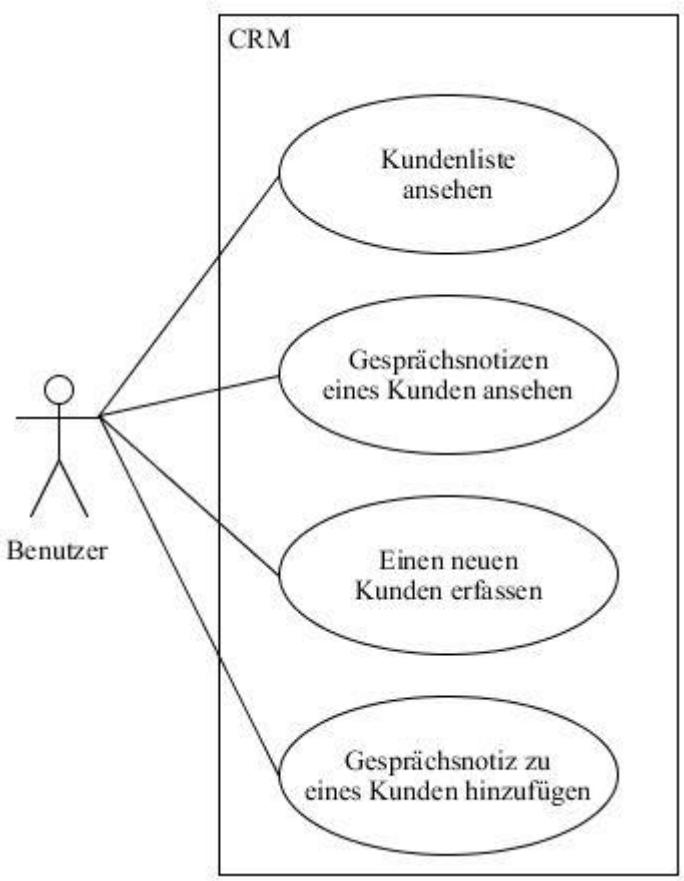

Abbildung 1 UseCase-Diagramm CRM

Benutzer kann:

- Kundenliste ansehen
- Die Gesprächsnotizen für einen Kunden ansehen
- Einen neuen Kunden erfassen
- Eine Gesprächsnotiz zu einem Kunden hinzufügen

Daraus folgt die benötigte Funktionalität:

- getCustomerList()
- getCustomer(«customer id»)
- addCustomer(«customer information»)
- addMemoToCustomer(«customer id», «memo text»)

Business Objects

Kunde (Customer)

- ein Kunde hat eine Kundennummer (customerId)
- Einen Namen und Vornamen (name)
- Eine Strasse und Hausnummer (street)
- Einen Wohnort mit Postleitzahl (city)
- eine Liste von Gesprächsnotizen (memo)

Alternativ könnte ein Kunde aus einer Kundennummer und einer Adresse bestehen. Dann brauchen wir zusätzlich ein Adressobjekt.

Adresse (Address)

- Einen Namen und Vornamen (name)
- Eine Strasse und Hausnummer (street)
- Einen Wohnort mit Postleitzahl (city)

Gesprächsnotiz (Memo)

- ein Erfassungsdatum (coverageDate)
- die erfasste Notiz (note)

Assoziationen

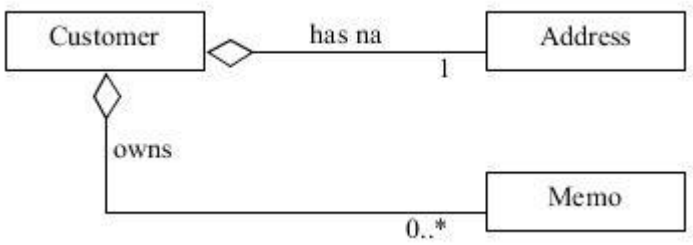

Abbildung 2 Assoziationsdiagramm CRM

oder einfacher:

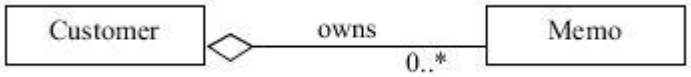

Abbildung 3 Assoziationsdiagramm CRM einfach

Design

Aus dem Dokument Serverapplikationen wissen wir, dass ein Business Object zwei Aspekte hat. Einerseits die Abstraktion (Was macht einen Customer zu einem Customer) und Andererseits die Implementierung (Wie funktioniert ein Customer, wie wird sein Zustand verwaltet). Dies führt zu folgendem Klassendiagramm:

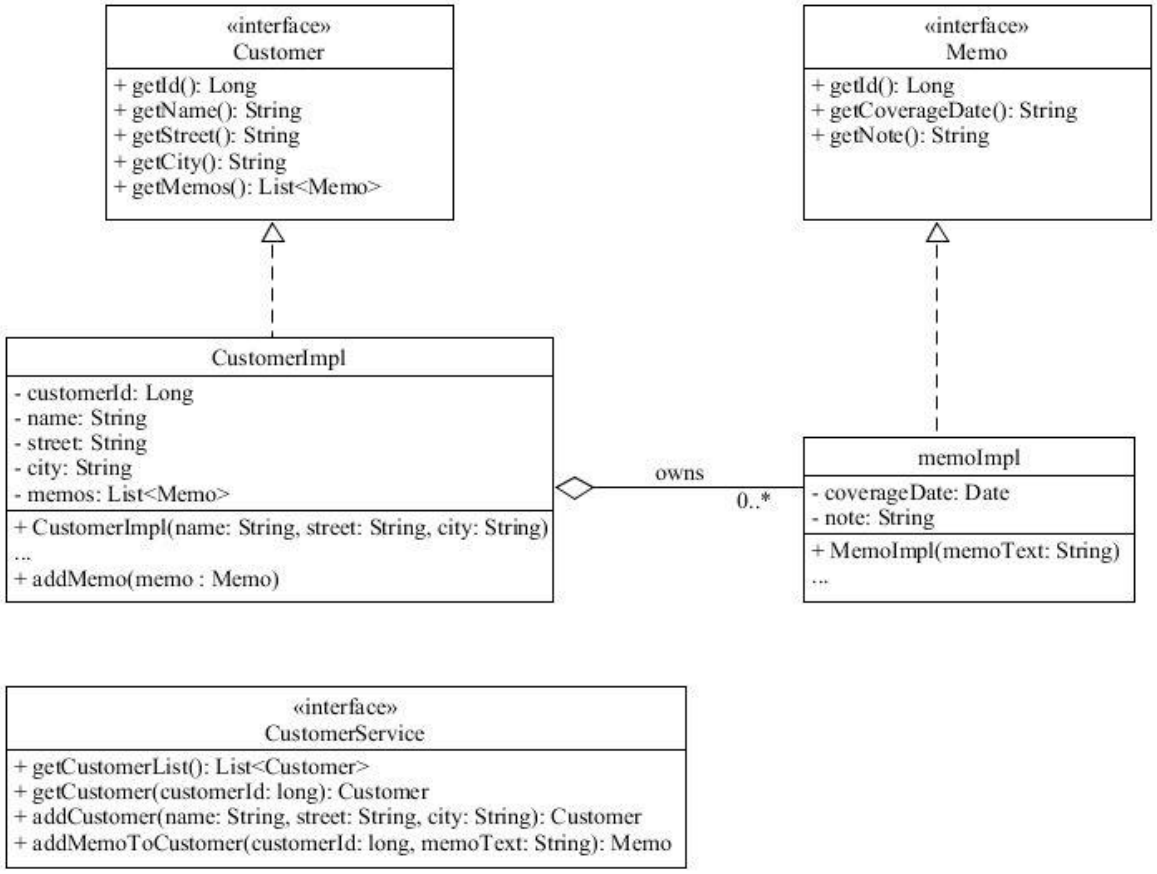

Abbildung 4 Klassendiagramm CRM

Weiter müssen wir noch festlegen, welche Funktionalität unsere Serverapplikation anbietet:

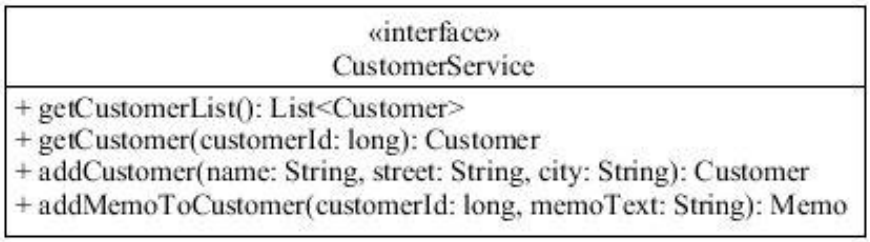

Abbildung 5 Interface CustomerService