

# Dokumentationsmappe

## BikeShop

By

Yannick Schmid

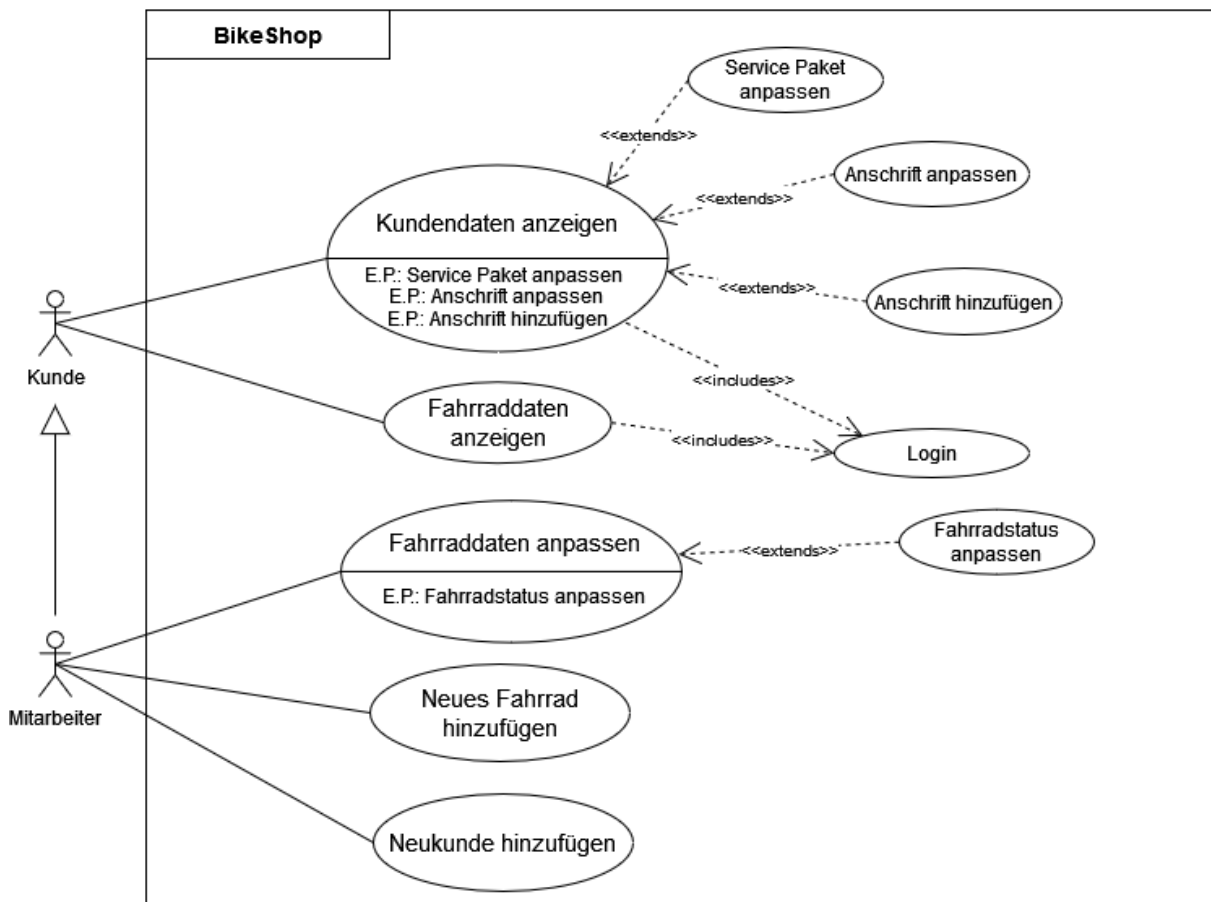
Denis Piekenbrock

## Inhaltsverzeichnis

1.	Definition und Zielsetzung der Datenbankprojekts.....	3
1.1	Projektanstoß .....	3
1.2	Planung und Durchführung .....	4
1.3	Reflexion der Projektarbeit .....	4
2.	Beispiele Anwendungsfälle / SQL Query .....	5
2.1	Kundendaten anzeigen.....	5
2.2	Fahrraddaten anzeigen.....	5
2.3	Fahrraddaten anpassen.....	5
2.4	Neues Fahrrad hinzufügen.....	5
2.5	Neukunde hinzufügen .....	6
2.6	Service Paket anpassen.....	6
2.7	Anschrift anpassen .....	6
2.8	Anschrift hinzufügen.....	6
2.9	Fahrradstatus anpassen.....	6
3.	Datenbank.....	7
3.1	Semantisches Modell .....	7
3.2	Datenbankschema.....	7

# 1. Definition und Zielsetzung der Datenbankprojekts

## 1.1 Projektanstoß



Zu Beginn war der Plan eine Datenbank zu erstellen, die es Nutzern ermöglicht auf verschiedenen Ebenen Kundendaten und deren zugehörigen Fahrraddaten zu verwalten. Die beiden relevanten Akteure sind dabei der Kunde und der Mitarbeiter

Bevor der Kunde die Datenbank nutzen kann, muss er sich mit seiner E-Mail und seinem, bei der erstmaligen Anmeldung gewählten, Passwort anmelden. E-Mail sowie Passwort werden in der Datenbank gespeichert. Da in diesem Fall kein User-Interface vorliegt, muss die Kundeneingabe manuell mit den gespeicherten Einträgen in den Kundendaten verglichen werden.

Außerdem sind alle include / extend Arme durch das Fehlen eines User-Interfaces zeitlich unabhängig. Die Idee ist, dass in einem User-Interface die gewünschten Daten angezeigt werden lassen können und dann direkt über die Oberfläche zu bearbeiten sind. Durch eine manuelle Eingabe der SQL Befehle liegt diese Zeitliche Abfolge nicht vor, denn man kann z. B. direkt auf die Tabelle ServicePaket zugreifen.

Nach dem Login hat der Kunde primär die Möglichkeit seine privaten Daten, sowie seine Fahrraddaten einzusehen. Darüber hinaus kann er bei Bedarf eine Anschrift bearbeiten oder neu hinzufügen.

Jedem Kunden ist außerdem ein Service Paket zugeordnet, welches das Level des Kundendienstes bestimmt. Zu Beginn muss der Kunde entscheiden, welches Service Paket für ihn in Frage kommt, da davon auch die monatliche Beitragshöhe abhängt.

Der Mitarbeiter hat alle Fähigkeiten des Kunden, sowie einige weitere administrative Rechte inne. So kann der Mitarbeiter die Fahrraddaten nicht nur einsehen, sondern diese auch anpassen. Jedes Fahrrad besitzt einen Status, der Auskunft darüber gibt, wo das Fahrrad sich aktuell befindet und, z.B. im Falle einer Reparatur, in welcher Phase es sich befindet.

Außerdem kann der Mitarbeiter ein Fahrrad zu einem bestehenden Kunden hinzufügen oder auch einen kompletten Neukunden in die Datenbank aufnehmen.

Es ist möglich einem einzelnen Kunden mehrere Adressen zuzuweisen. (z. B. eine Liefer- und Rechnungsadresse) Außerdem ist es möglich mehrere Kunden einer Adresse zuzuweisen. (z. B. Geschwister oder Mitbewohner) Ein Kunde kann auch mehrere Fahrräder besitzen und umgekehrt ein Fahrrad kann mehreren Kunden zugewiesen sein. (z. B. Bike-Sharing unter Freunden)

## **1.2 Planung und Durchführung**

Begonnen haben wir mit der Erstellung eines Use-Case-Diagramms. Dadurch konnte frühzeitig geklärt werden, welche konkreten Anwendungsfälle wir mit der Datenbank abdecken wollten.

Nachdem dieser Schritt erledigt war, war eine gewisse Grundstruktur für das Projekt gegeben und wir konnten dazu über gehen uns damit zu beschäftigen, wie die Anwendungsfälle am besten umgesetzt werden können. Zu diesem Zwecke beschäftigten wir uns mit der Erstellung eines semantischen Datenbankmodells und daraufhin mit der Erstellung eines Datenbankschemas.

Im Laufe des Projekts wurden alle drei Modelle stetig verändert, neue Ideen wurden eingebaut und die Modelle dementsprechend angepasst.

Auf Grund der überschaubaren Größe des Projekts sahen wir es als am sinnvollsten an alle Aufgaben gemeinsam zu erledigen sowie alle Lösungen zusammen zu erarbeiten.

## **1.3 Reflexion der Projektarbeit**

Eines der größten Schwierigkeiten lag darin zu beurteilen, welche Entscheidungen wichtig für eine in der realen Welt funktionierende Datenbank sinnvoll waren. Oft erschienen Lösungen von einem rein technischen Standpunkt aus sehr sinnvoll, dachte man allerdings an einen konkreten Anwendungsfall fiel oft auf, dass eine rein technische Betrachtung der Probleme im Gesamtbild nicht immer die sinnvollste war. (z. B. bestimme Datenbanken Beziehungen)

Probleme mussten immer aus mehreren Blickwinkeln betrachtet werden und ein für alle Seiten befriedigender Mittelweg gewählt werden.

## 2. Beispiele Anwendungsfälle / SQL Query

### 2.1 Kundendaten anzeigen

Kunde / Mitarbeiter will Kundendaten für Lucy Müller anhand ihrer E-Mail abrufen:

```
select * from Kunde k
join KundeAnschrift ka on k.KundenID = ka.KundenID
join Anschrift a on ka.AnschriftID = a.AnschriftID
join ServicePaket sp on k.ServiceID = sp.ServiceID
where k.Email = 'Müller.Lucy@web.de'
```

### 2.2 Fahrraddaten anzeigen

Kunde / Mitarbeiter will Fahrraddaten für Lucy Müller anhand ihrer E-Mail abrufen:

```
select * from Kunde k
join KundeFahrrad kf on k.KundenID = kf.FahrradID
join Fahrrad f on kf.FahrradID = f.FahrradID
join FahrradStatus fs on f.StatusID = fs.StatusID
where k.Email = 'Müller.Lucy@web.de'
```

### 2.3 Fahrraddaten anpassen

Auf Grund neuer Lackierung soll die Farbe von einem Fahrrad anhand der Rahmennummer angepasst werden:

```
update Fahrrad
set Farbe = 'Rot-Weiß'
where Rahmennummer = '613XSTQFR94494'
```

### 2.4 Neues Fahrrad hinzufügen

Lucy Müller hat ein neues Fahrrad gekauft und der Mitarbeiter pflegt es in der Datenbank ein und verknüpft es mit ihrem Konto:

Erster insert:

```
insert into Fahrrad (Marke, Model, Rahmennummer, Farbe, Art,
StatusID)
values ('Cube', 'CF3-B', '4545BKJHB234CF', 'Grün', 'Rennrad', 1)
```

Zweiter insert:

```
insert into KundeFahrrad (KundenID, FahrradID)
values (1,8)
```

## 2.5 Neukunde hinzufügen

Ein Neukunde wird von einem Mitarbeiter aufgenommen:

```
insert into Kunde (Name, Vorname, Email, Passwort, ServiceID)
values ('Toretto', 'Dominik', 'd.toretto@web.de', 'Sonne_12', 4)
```

## 2.6 Service Paket anpassen

Lucy Müller möchte ein ServicePaket vom Level Gold abonnieren:

```
update Kunde
set ServiceID = 4
where KundenID = 1
```

## 2.7 Anschrift anpassen

Lucy Müller ist umgezogen und möchte ihre Anschrift anpassen:

```
update Anschrift
set StrasseHausnummer = 'Paul-Diethei Straße 5'
where AnschriftID = 1
```

## 2.8 Anschrift hinzufügen

Dominik Toretto möchte zu seinem Konto eine neue Anschrift hinzufügen:

Erster insert:

```
insert into Anschrift (StrasseHausnummer, PLZ, Ort, AnschriftArt)
values ('Lechweg 2', 87435, 'Kempten', 'Liefer- & Rechnungsadresse')
```

Zweiter insert:

```
insert into KundeAnschrift (KundenID, AnschriftID)
values (8, 8)
```

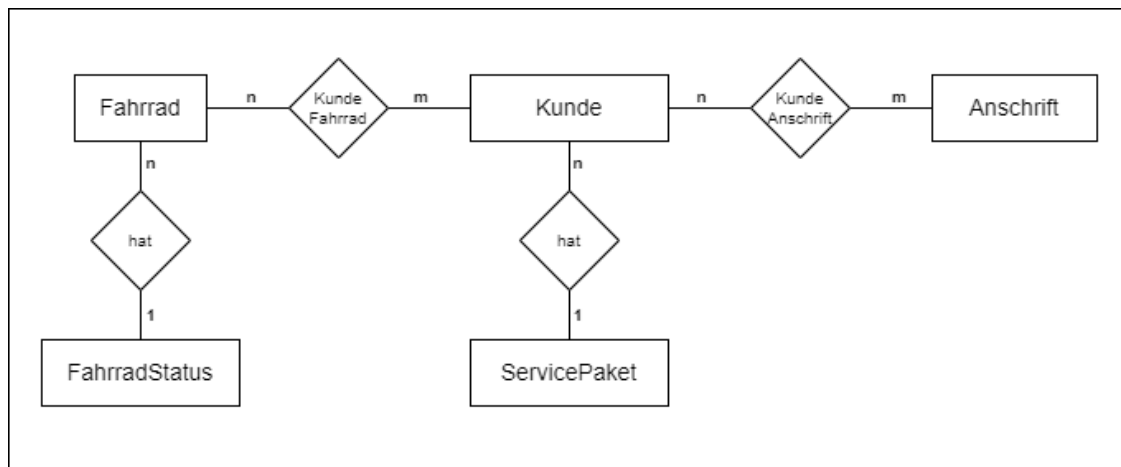
## 2.9 Fahrradstatus anpassen

Bei dem Fahrrad von Lucy Müller ist eine Speiche gebrochen. Sie bringt das Fahrrad zur Reparatur in die Werkstatt des BikeShop. Der Mitarbeiter pflegt den Fahrradstatus entsprechend ein:

```
update Fahrrad
set StatusID = 2
where FahrradID = 1
```

### 3. Datenbank

#### 3.1 Semantisches Modell



#### 3.2 Datenbankschema

