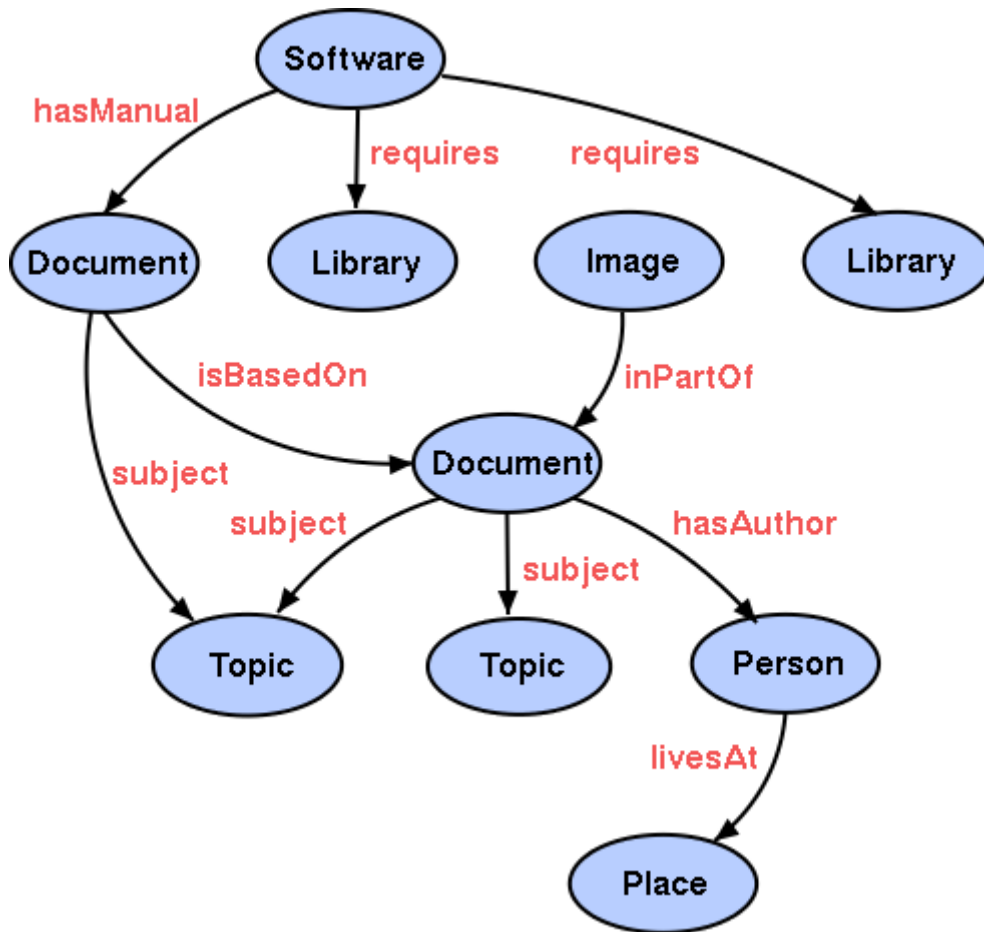


# Semantic Web



# 1 Revision History

| Version | Datum    | Autor | Bemerkung  |
|---------|----------|-------|--|
| 0.1     | 17.09.10 | Moos  | Initiale Struktur und erster Teil Requirements Engineering, technisches Design |
| 0.2     | 05.10.10 | Moos  | Erweiterung RE, Unterteilung in Use cases und NFR                              |
| 0.3     | 15.10.10 | Moos  | Titelblatt, GUI Entwurf  |
| 0.4     | 15.10.10 | Lahn  | ERM Entwurf  |

## Inhaltsverzeichnis

|  |   |
|--|---|
| 1 Revision History.....                        | 2 |
| 2 Requirements Engineering .....               | 3 |
| 2.1 – Use cases.....                           | 3 |
| 2.1.1 Verwaltung von Begriffen.....            | 3 |
| 2.1.2 Verwaltung von Beziehungstypen.....      | 3 |
| 2.1.3 Erfassen und Ändern von Beziehungen..... | 3 |
| 2.1.4 Locking der einzelnen Einträge.....      | 3 |
| 2.1.5 History.....                             | 3 |
| 2.1.6 Accounts und Rollen.....                 | 3 |
| 2.2 Non-functional requirements.....           | 4 |
| 2.2.1 Darstellung.....                         | 4 |
| 2.2.2 Mehrbenutzerfähigkeit.....               | 4 |
| 3 Technisches Design.....                      | 4 |
| 3.1 Entity-Relationship-Modell.....            | 4 |
| 4 GUI Entwurf.....                             | 5 |
| 4.1 Ablauf.....                                | 5 |
| 4.1.1 Login.....                               | 5 |
| 4.1.2 Begriff hinzufügen.....                  | 6 |
| 4.1.3 Begriff bearbeiten.....                  | 6 |
| 4.1.4 Begriffsübersicht.....                   | 7 |
| 4.1.5 History.....                             | 8 |

## 2 Requirements Engineering

### 2.1 – Use cases

Die Grundlage für das Requirements Engineering stellt das abgegebene Dokument „[Aufgabenstellung Semesterarbeit "Cooperative Modeling"](#)“ dar.

Die Implementierungsdetails darunter sind per Definition frei. Alle gemachten Entscheide sollten der Übersicht halber hier kurz erwähnt werden.

#### 2.1.1 Verwaltung von Begriffen

Es müssen Begriffe verwaltet werden können. Diese Funktionalität ist eine der grundlegenden Bausteine der Applikation.

#### 2.1.2 Verwaltung von Beziehungstypen

Es müssen verschiedene Beziehungstypen erfasst werden können. Dies könnte den Admin-Accounts vorbehalten sein.

#### 2.1.3 Erfassen und Ändern von Beziehungen

Die definierten Begriffe werden untereinander verknüpft. Dies geschieht über die gerichteten Begriffstypen aus 2.1.2.

#### 2.1.4 Locking der einzelnen Einträge

Wenn ein Benutzer einen entsprechenden Eintrag ändern will und die Maske aufruft, werden der Eintrag und alle Beziehungen damit gesperrt. Diese Sperre wird aufgehoben, wenn dieselbe Session eine andere Seite öffnet resp. die Änderungen abschliesst.

Für den Fall, dass der Benutzer seinen Browser ohne Logout schliesst, wird die Sperre nach 2 Minuten wieder aufgehoben.

Die anderen Benutzer sehen ein gesperrtes Item mit einer anderen Farbe und bekommen eine Warnung wenn sie dieses ändern wollen.

#### 2.1.5 History

Es soll eine History geführt werden, in welcher die Änderungen von Begriffen und Relationen kurz kommentiert werden muss. Dies dient der Absprache mit den anderen Teammitgliedern und soll ein separates Tool (Forum / Chat) unnötig machen.

#### 2.1.6 Accounts und Rollen

Accounts müssen verwaltet werden können. Zudem werden Benutzer in eigentliche User und Admins unterteilt, wobei Admins die gesamte Userverwaltung erledigen.

## 2.2 Non-functional requirements

### 2.2.1 Darstellung

Die Darstellung der Ontologien und deren Beziehungen soll möglichst anwenderfreundlich sein. Eine Möglichkeit stellt die Benutzung von dynamisch generierten Vektorgrafiken dar. Der Aufwand dafür muss noch abgeklärt werden.

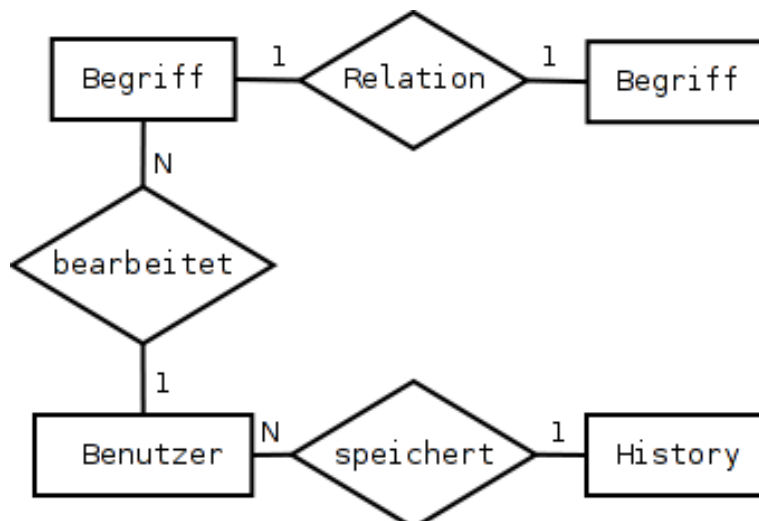
Es soll nicht zu viel Zeit in dieses Problem investiert werden.

### 2.2.2 Mehrbenutzerfähigkeit

Das Tool muss von mehreren Benutzern gleichzeitig bearbeitet werden können. Dabei darf es nicht zu inkonsistenten Daten kommen.

## 3 Technisches Design

### 3.1 Entity-Relationship-Modell



## 4 GUI Entwurf

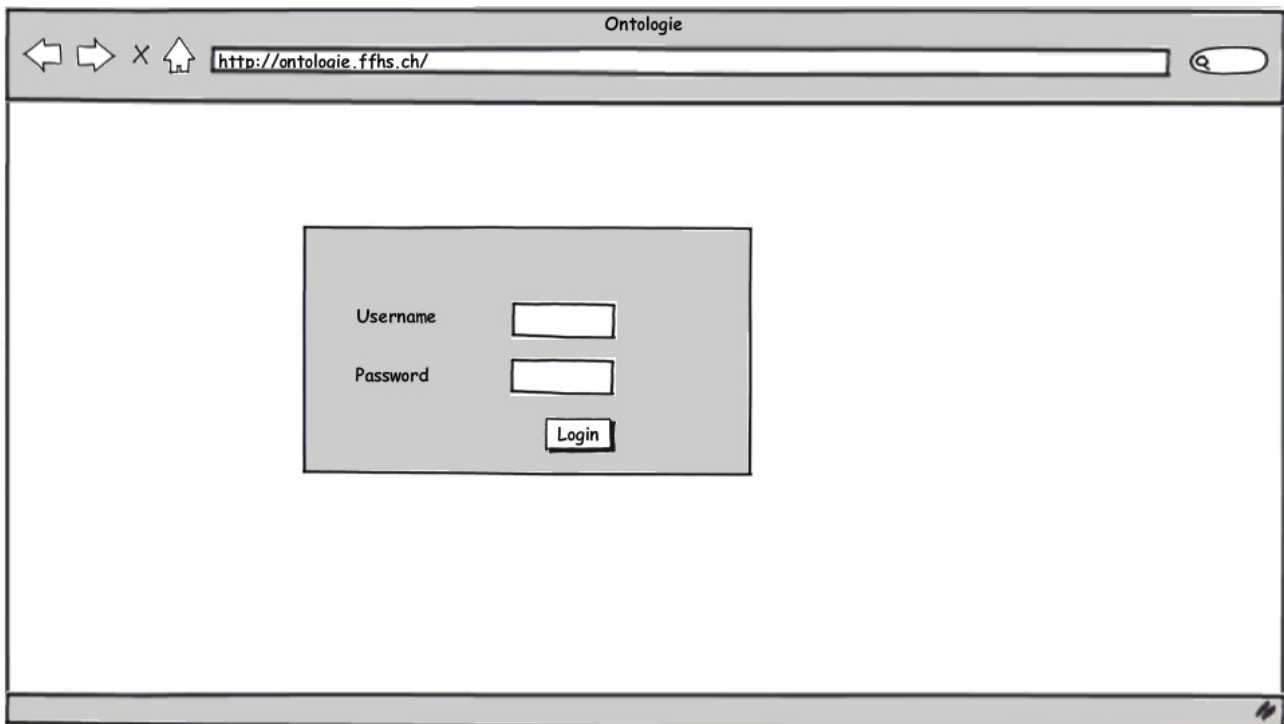
Der GUI Entwurf soll kurz darstellen, wie die Applikation aussehen könnte. Sie dient vor allem als Grundlage, dem Kunden einen ersten Eindruck liefern zu können.

Als Tool haben wir uns für Balsamiq Mockups entschieden, welches schnelle und simple Entwürfe ermöglicht.

### 4.1 Ablauf

#### 4.1.1 Login

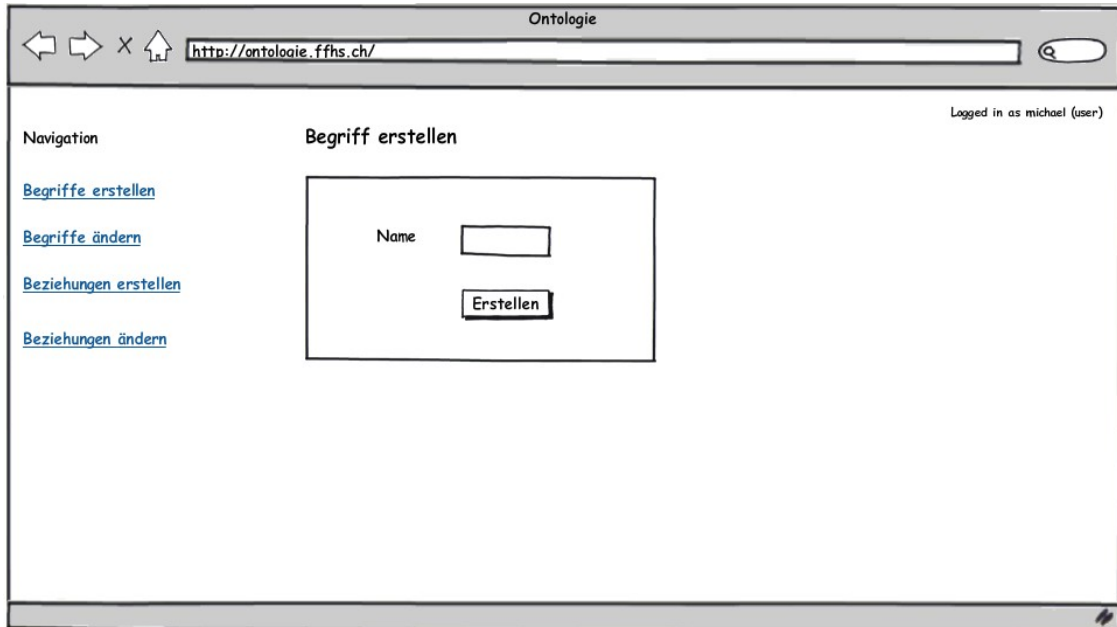
In einem ersten Schritt kann sich der Benutzer einloggen.



Danach erscheint die eigentliche Applikation, links die Navigation.

### 4.1.2 Begriff hinzufügen

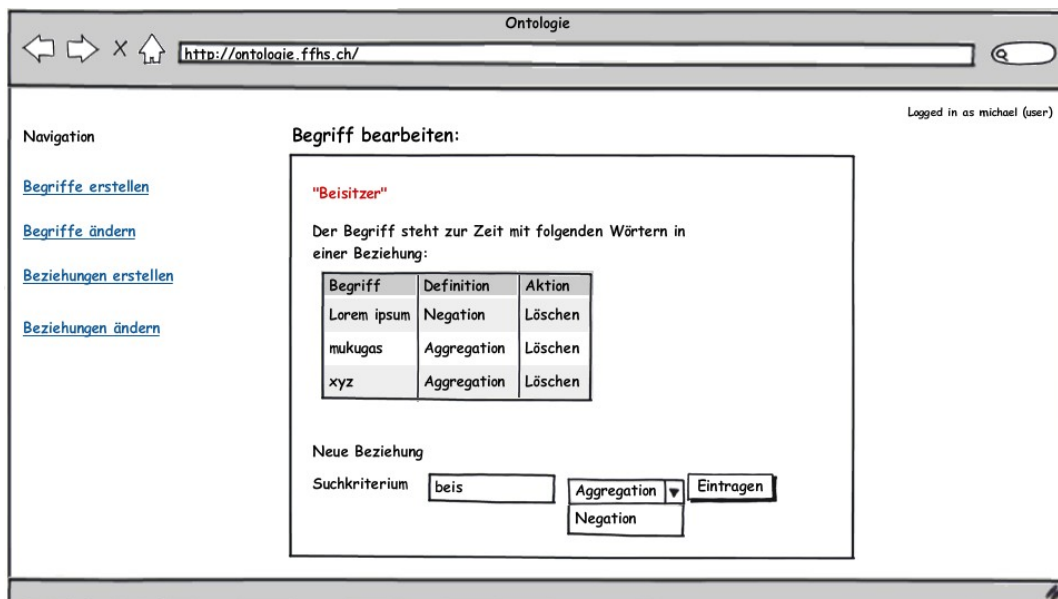
Die erste Funktion nach dem Login (im Benutzerbereich) dient dazu, Begriffe hinzuzufügen.



Hierbei muss lediglich der Name des neuen Begriffs eingegeben werden. Im nächsten Fenster können Beziehungen zu anderen Wörtern aufgebaut werden.

### 4.1.3 Begriff bearbeiten

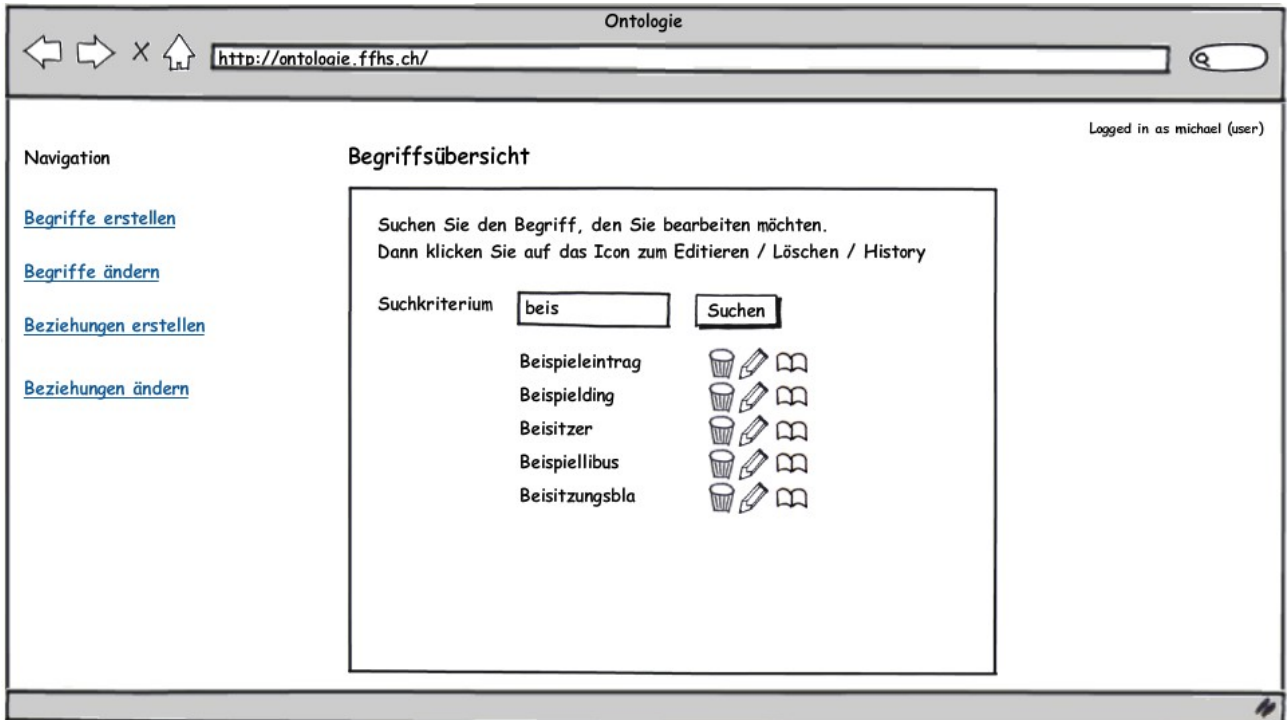
In diesem Formular erscheinen bereits bestehende Beziehungen zu anderen Begriffen – unten können neue hinzugefügt werden.



### 4.1.4 Begriffsübersicht

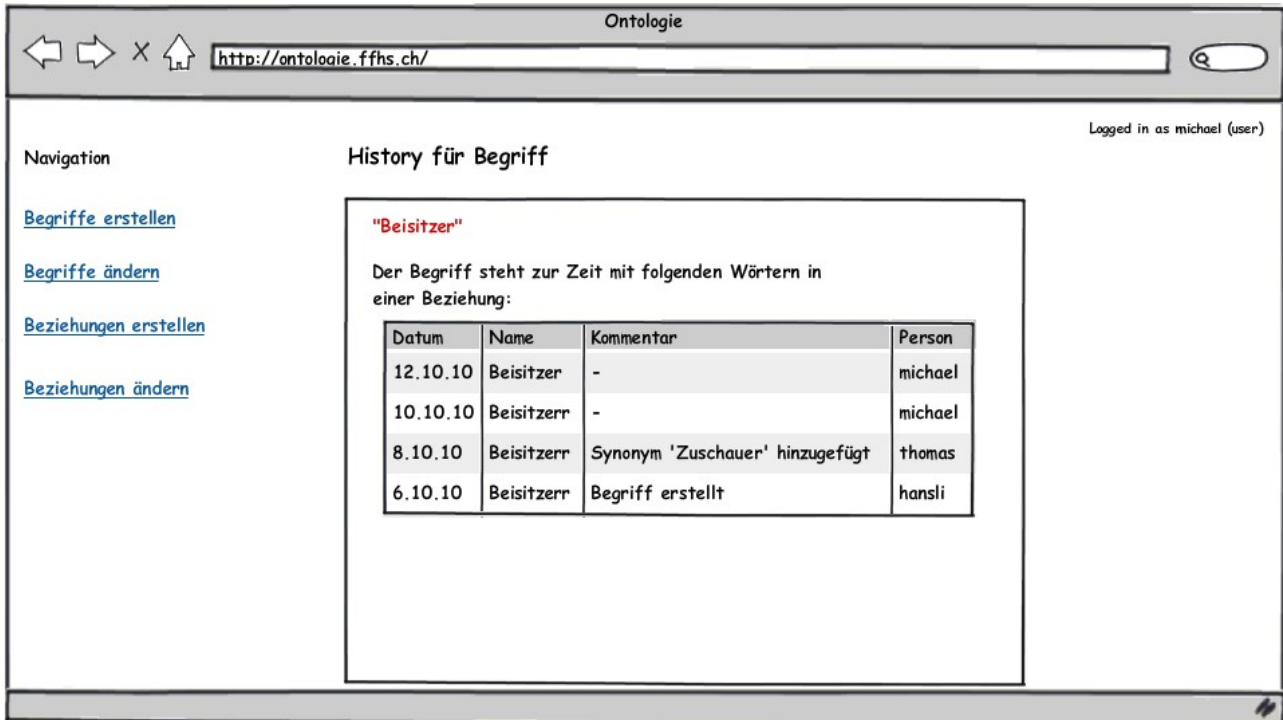
Wenn Begriffe geändert werden sollen, wird zuerst in einer Übersicht der entsprechende Eintrag ausgewählt. Dieser kann hier direkt gelöscht werden. Über die Editierfunktion kommt man auf die Bearbeitungs-Funktion zurück (siehe Hinzufügen).

Die letzte hier verfügbare Funktion ist die History. Diese wird im folgenden Abschnitt beschrieben.



### 4.1.5 History

Die History dient dazu, Änderungen zu kommentieren. So wird jede Änderung an einem Begriff systematisch geloggt. Es ist auf jeden Fall möglich, Kommentare zu erfassen, damit andere Mitarbeiter wissen, warum etwas geändert wurde (analog bsp. SVN Commits).



Die Beziehungen funktionieren analog zu den Begriffen. Es wird deshalb in diesem Rahmen darauf verzichtet, diese noch explizit aufzuführen.