

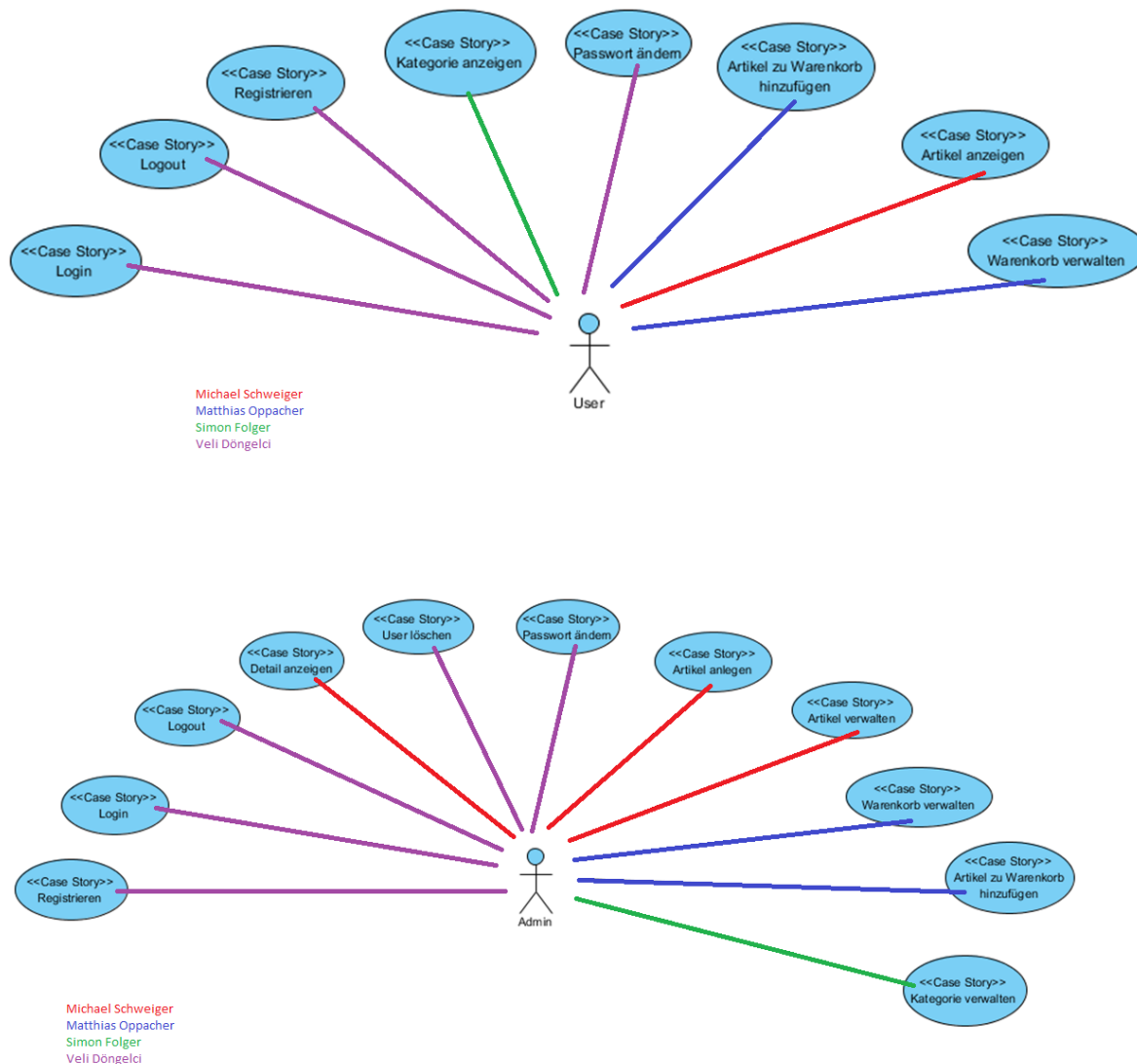
Web-Services 2016

Einzelbericht

Team: Artikelverwaltung

Name: Simon Folger

1 Use Cases



Die Use-Cases werden je nach Berechtigung in einen Admin- und Userbereich aufgeteilt. Bei der Bearbeitung war ich für sämtliche Aktionen, die im Zusammenhang mit den Kategorien stehen, sowie für das Layout verantwortlich.

Sowohl der normale User als auch der Administrator bekommen beim Klick auf den Menüpunkt „Artikel“ zuerst alle Kategorien als Liste ausgegeben. Beim Klick auf eine Kategorie werden die entsprechenden Artikel dieser Kategorie angezeigt.

Der Administrator hat einen zusätzlichen Menüpunkt „Artikelverwaltung“, der sich vom Aufbau her genau wie der Punkt „Artikel“ verhält, nur, dass man unter diesem Punkt viel mehr Konfigurationsmöglichkeiten hat. Zusätzlich zur Liste der Kategorien gibt es einen Button, der beim Klick und nach Angabe eines Namens eine Kategorie hinzufügt. Zudem ist es möglich durch Rechtsklick auf eine Kategorie zwei weitere Optionen zur Verfügung hat. Zum einen das

Umbenennen der ausgewählten Kategorie, wobei man in einem Popup, wie beim Hinzufügen, den gewünschten Namen in einem Eingabefeld eingibt und bestätigt, und zum anderen das Löschen einer Kategorie, das in einem zusätzlichen Popup noch bestätigt werden muss.

Das Layout, beziehungsweise das Design, der Webseite sollte einfach und modern aufgebaut sein. Zudem sollte die Seite auch responsiv sein. Hierzu wurde das Angular Material Framework verwendet, das sehr schöne Layout Beispiele mitbringt.

2 Schnittstelle

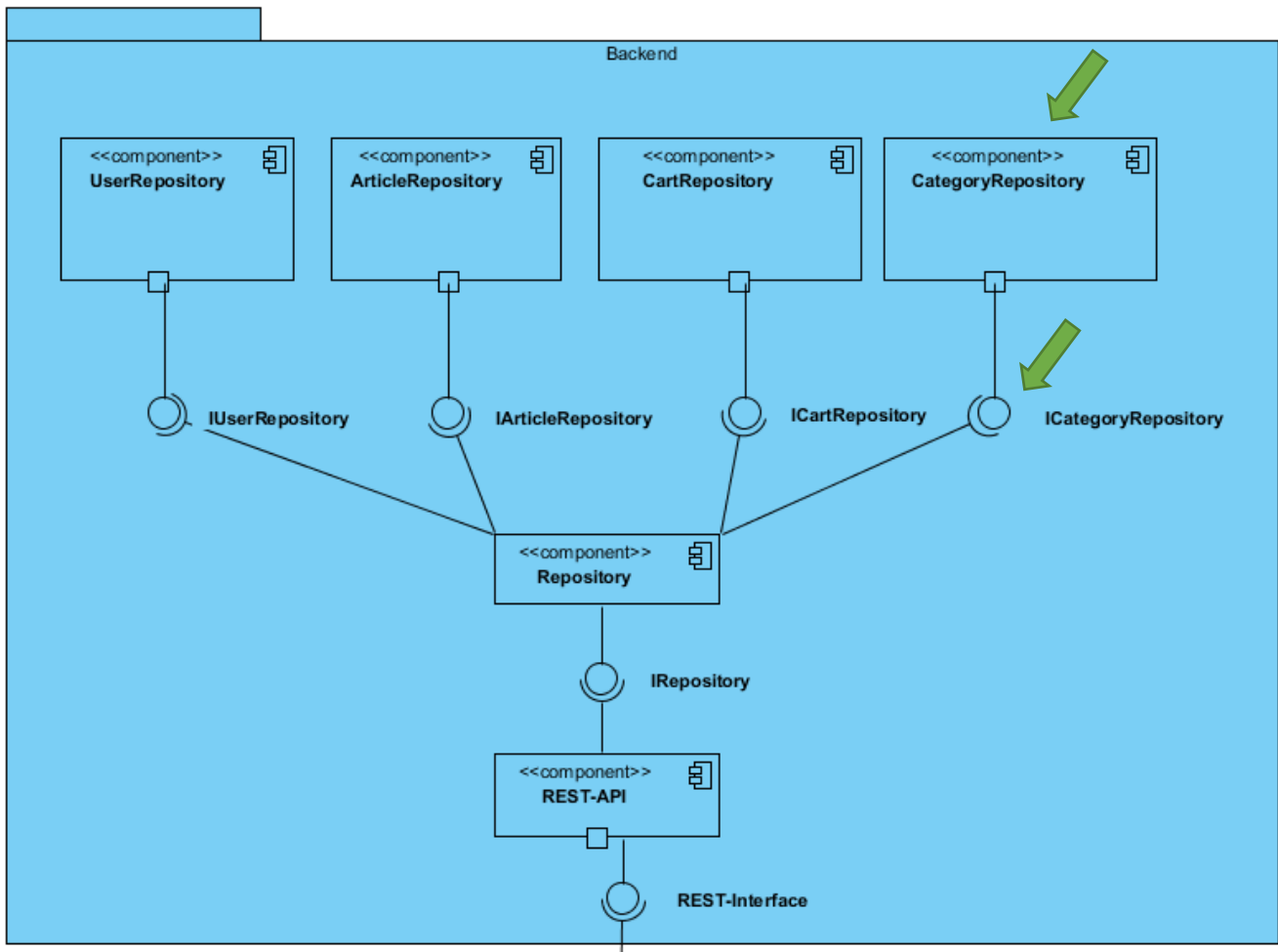
Der BaseApiController ist die Vaterklasse von allen Controllern des Service. Meine Aufgabe bestand darin die Kategorie-Schnittstelle zu implementieren. Dazu sollte es möglich sein, Kategorien anzuzeigen und hinzuzufügen sowie bestehende Kategorien umzubenennen und zu löschen. Dazu habe ich die folgenden http-Methoden ausprogrammiert.

GET	/api/v1/categories
GET	/api/v1/categories/{id}
POST	/api/v1/categories
PUT	/api/v1/categories/{id}
DELETE	/api/v1/categories/{id}

3 Technische Architektur – Blockschaubild

Ich war aufgrund der Aufgabenverteilung für das CategoryRepository und alle zugehörigen Methoden verantwortlich. Wir haben das Entity Framework verwendet, um mit der Datenbank zu kommunizieren. Das Ninject-Package haben wir verwendet, um im Controller auf das Repository zuzugreifen. Trotz der negativen Kommentare über Ninject, die in der Vorlesung gefallen sind, sind wir relativ gut mit dem Package zurechtgekommen.

Das folgende Diagramm zeigt meine Verantwortlichkeiten in diesem Bereich.



4 Technische Architektur – Datenhaltung

Als Datenbankmodell haben wir uns für den Microsoft SQL Server entschieden.

Ich war vor allem für die den Datenhaltungs-Teil der Kategorien zuständig. Die Tabelle enthält die Spalten ID, ParentID und Name. Die ParentID ist aus dem Grund noch in der Tabelle, weil wir eigentlich eine verschachtelte Kategorie-Hierarchie implementieren und dazu im Frontend eventuell eine Baumstruktur anzeigen wollten. Dazu sind wir jedoch aus Zeitgründen nicht mehr gekommen.

Zudem musste ich mich mit dem Herrn Schweiger abstimmen, der für die Implementierungen der Artikel zuständig war, da unsere beiden Tabellen direkt voneinander abhängen. Dazu haben wir auch eine Kreuztabelle angelegt, damit einzelne Artikel bestimmten Kategorien zugeordnet werden können.

In den folgenden Tabellen ist der Aufbau meiner Kategorie-Tabelle und der Kreuztabelle für die Verbindung zwischen Kategorie und Artikel.

Category:

ID	int	ID der Kategorie
ParentID	int	(ID der Vaterkategorie)
Name	nvarchar(255)	Name der Kategorie

ArticleCategory:

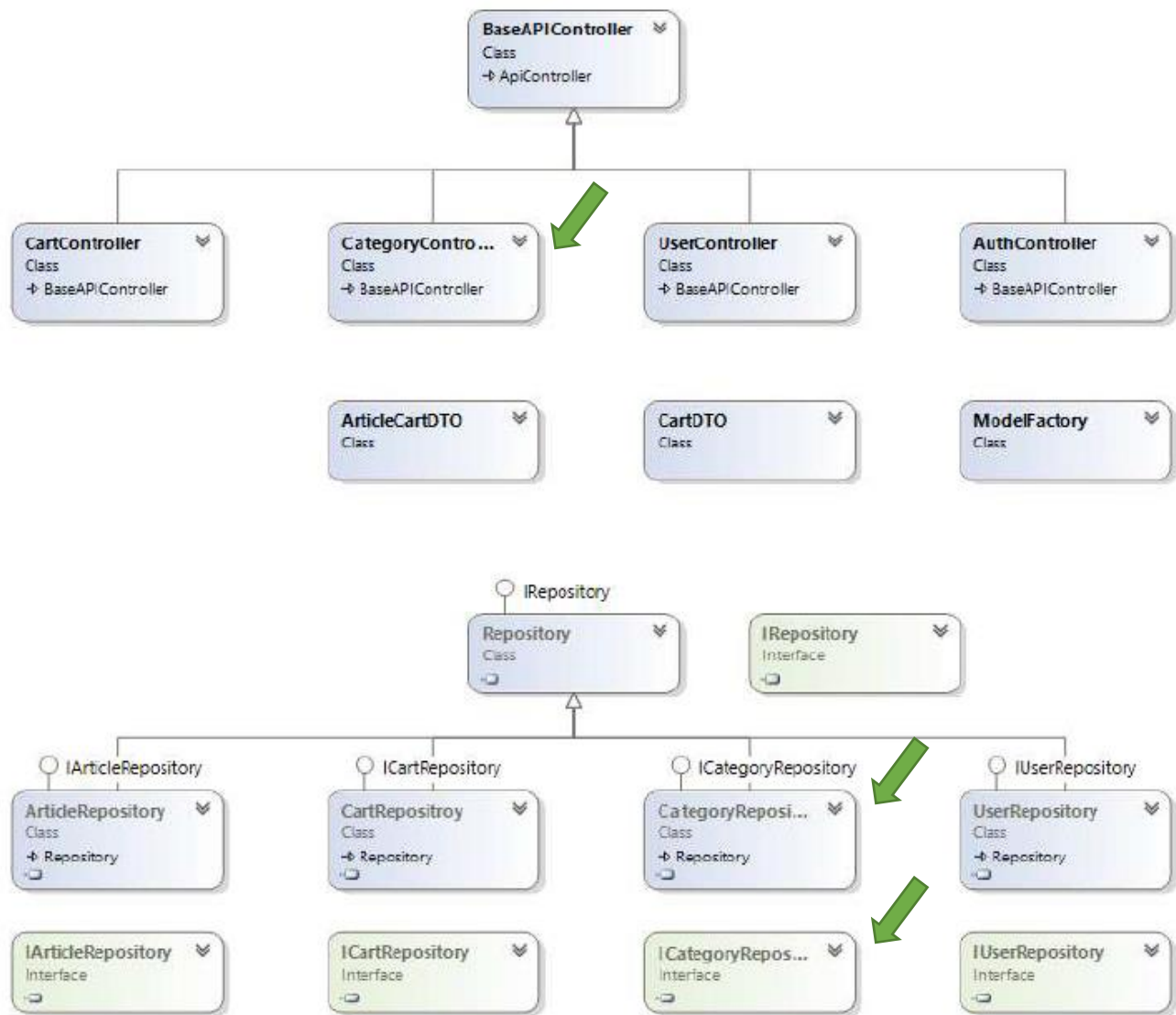
ID	int	ID der Verbindung
CategoryID	int	ID der Kategorie
ArticleID	int	ID des Artikels

5 Technische Architektur – Klassendiagramm

Hier haben wir das Repository Entwurfsmuster verwendet um den Datenzugriff zu kapseln. Falls mal ein anderes Framework für die Daten verwendet werden soll, ist dies sehr hilfreich.

Die Klassen die ich implementiert habe und für die ich verantwortlich war, werden in der nachfolgenden Tabelle und der Grafik veranschaulicht.

CategoryController	Schnittstelle der Kategorien
CategoryRepository	Entsprechende Methoden der Kategorien
CategoryDTO	DTO der Kategorien



6 Testkonzept

Tests haben wir aus Zeitgründen nicht geschrieben. Jedoch haben wir das komplette Projekt per Hand während und nach der Entwicklung durchgetestet.

7 Implementierung der App

In unserem Projekt war ich hauptsächlich damit beschäftigt, das Design der App zu entwickeln. Zudem war ich noch für viele kleinere Frontend-Aufgaben verantwortlich. Zudem habe ich für die Icons die Material Design Icons Bower Komponente verwendet. Für die Tabellen habe ich die Bower Komponente SmartTable verwendet, die einige nützliche Funktionen für Tabellen mitbringt.

Zu meinem zweiten Aufgabenteil, der Kategorien, habe ich hier die entsprechenden Controller, Services und Angular-Ressourcen implementiert. Dazu gehören die Rechtsklick-Funktionen

und das Hinzufügen von Kategorien sowie das Anzeigen.