Echoes

Damien Flury, Tim Hess

30. November 2018

Inhaltsverzeichnis

1 Auftrag

Unser Auftrag besteht darin, eine Applikation mit Datenbankanbindung zu entwickeln. Technologien können frei gewählt werden.

2 Idee

Unsere Idee ist die Entwicklung einer Webapplikation für die Planung von Hausaufgaben und Prüfungen.

3 Projektbeschreibung

Die Webapplikation soll die Verwaltung verschiedener Klassen mit Schülern ermöglichen. Man soll einen Account erstellen, Hausaufgaben und Prüfungen zu einer Klasse hinzufügen und diese auch als erledigt markieren können. Auf die jeweiligen Hausaufgaben einer Klasse haben alle zugehörigen Schüler Zugriff und können diese lesen. Ob sie Aufträge auch bearbeiten können, ist noch nicht endgültig entschieden. Momentan planen wir, die Bearbeitung für den Ersteller des Auftrags zu ermöglichen, aber nicht für die anderen Schüler. Man soll Aufträge kommentieren können.

4 Technologien

Für die Webapplikation verwenden wir verschiedene Frameworks. Serverseitig bieten wir eine REST API mit ASP.NET Core an, Clientseitig konsumieren wir diese API mit dem Single Page Application (SPA) Framework Angular.

4.1 Backend

- ASP.NET Core 2.1
- C# 7.3
- MariaDB
- Entity Framework Core

4.2 Frontend

- Angular 7
- Typescript
- Bootstrap 4

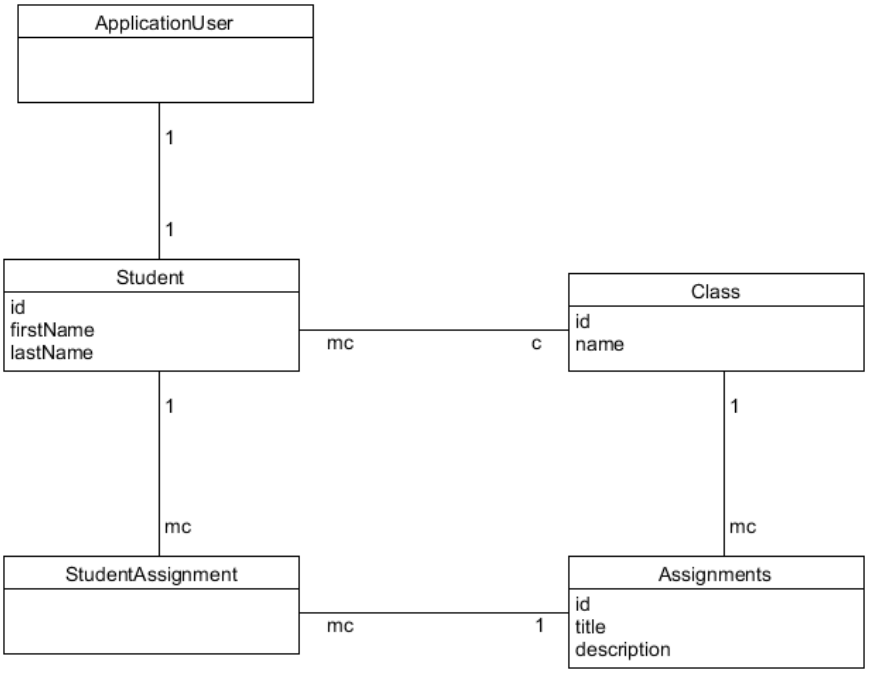

Abbildung 1: Entity Relationship Diagram (ERD)

5 Datenbank und konzeptionelles Datenmodell

Wie in Abbildung 1 ersichtlich, verwenden wir eine relationale Datenbank. Für die Anbindung an die Applikation verwenden wir Entity Framework Core. Die Tabelle *ApplicationUser* ist für das Login zuständig und besitzt eine 1-1 Beziehung mit *Student*. Jeder Student gehört einer oder keiner *Class* (Schulklasse) an. Eine Schulklasse besitzt keine oder mehrere *Assignments*, also Aufträge. Die Tabelle *StudentAssignment* bietet die Möglichkeit, die Hausaufgaben, welche bereits erledigt wurden, zu markieren.

6 REST API

Wir verwenden das platformunabhängige Framework ASP.NET Core für die Erstellung einer REST-API. Der Datenbankzugriff läuft über Entity Framework Core. Die Tabellen werden automatisch aus dem Model generiert. Es wird zunächst eine Datenbankmigration erstellt, welche danach auf die Datenbank mit dem *update*-Command auf die Datenbank angewandt werden kann.

```
$ dotnet ef migrations add Initial
$ dotnet ef database update
```

ASP.NET Core verwendet standardmässig das MVC-Pattern. Somit haben wir Controller, welche die API-Schnittstellen darstellen (Siehe Listing 1).

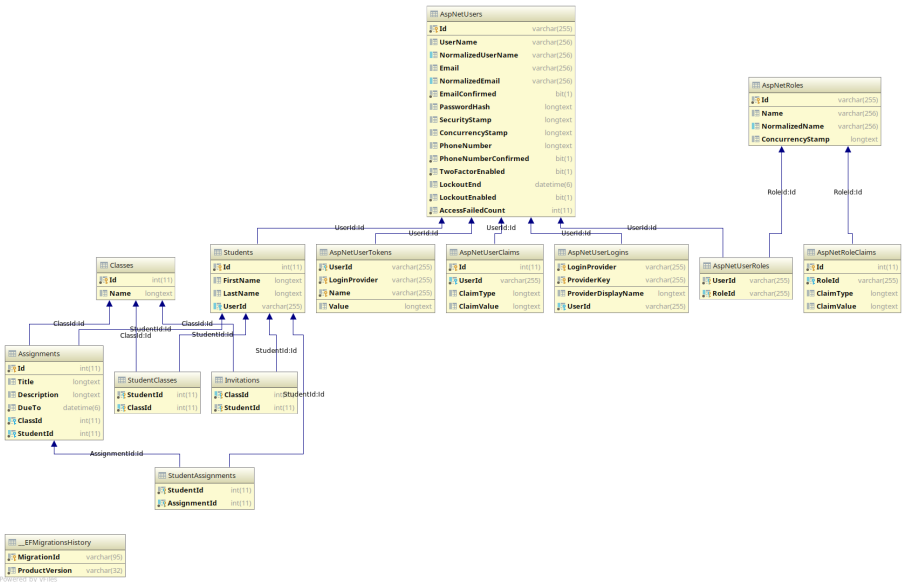

Abbildung 2: Entity Relationship Diagram (Generated)

```
[Route("api/[controller]")]
[ApiController]
[Authorize]
public class AssignmentsController : ControllerBase
{
    [HttpGet]
    public ActionResult<IEnumerable<Assignment>> Get()
        => Ok(GetAll());
}
```

Listing 1: Controller

```
var assignments = GetAll()
    .Where(assignment => assignment.DueTo < DateTime.Now)
    .OrderBy(assignment => assignment.DueTo);
```

Listing 2: Lambda-Syntax

```
var assignments = from assignment in GetAll()
    where assignment.DueTo < DateTime.Now
    orderby assignment.DueTo
    select assignment;
```

Listing 3: Query-Syntax

```
export class AssignmentsService {
    constructor(private http: HttpClientService) {}

    getAssignments(): Observable<Assignment[]> {
        return this.http
            .get<Assignment[]>('/api/Assignments');
    }
}
```

Listing 4: HttpClient

Die Datenbank-Aufrufe erfolgen über *LINQ* (Language Integrated Query), eine Abstraktion, um SQL-Statements in C# selbst zu schreiben. Dies erfolgt entweder über die Lambda-Syntax (Siehe Listing 2), oder über die Query-Syntax (Siehe Listing 3). Das Mapping auf die Objekte erfolgt automatisch.

7 App

Für das Frontend verwenden wir Angular, ein Single Page Application (SPA) Framework. Dies ist Component-Based und arbeitet viel mit Property-Binding. Für die API-Aufrufe verwenden wir die Klasse *HttpClient* (Siehe Listing 4).

8 Endprodukt

Bis jetzt hat Echoes einige nützliche Funktionen für Schüler und Lehrer. Sowohl Schüler als auch Lehrer können sogenannte Assignments in jeder Klasse erstellen und ein Datum dazusetzen. Diese Assignments werden danach auf dem Tab *My Assignments* angezeigt. Wenn das Datum des Assignments abgelaufen ist, wird es im Subtab *Inactive* angezeigt. Diese Assignments werden zusätzlich nur angezeigt, wenn der User sich in der Klasse befindet, in der das Assignment erstellt wurde. Klassen kann man sehr einfach erstellen und gewisse Schüler mit ihrer eingegebenen Email-Adresse zur Klasse einladen. Wird ein Schüler eingeladen, kann er die Einladung im *Invitations* Tab finden und akzeptieren.

9 Zukünftige Erweiterungen und Updates

Wir haben definitiv noch weitere Ideen, die wir bei Echoes einbauen können. Da jetzt der grundlegende Baustein des Projektes gelegt ist, kann man noch viele zukünftige Updates darauf aufbauen und erweitern. Wir planten, dass Schüler abhaken können, ob sie ein Assignment schon fertiggestellt haben und dass diese Liste von Schülern für den Ersteller des Assignments sichtbar ist.