

Projektarbeit

Webapp für Studierende des Studienganges AIMMT

Gedruckt am: 20. Mai 2020

von: Nicole Goldmann
geboren am 20. August 1997
in Herbolzheim

von: Stefanie Weidemann
geboren am 25. August 1995
in Anklam

Betreuer: Prof. Dr.-Ing. Herbert Litschke

Inhaltsverzeichnis

1	Einleitung	3
2	Untersuchung der Hochschul Website	4
2.1	Analyse des Ist-Zustandes	4
2.2	Festlegung Systemanforderung	5
3	Grundlagen	6
3.1	Webcomponents	6
3.1.1	LitElement	6
3.1.2	Shadow-Dom	7
3.1.3	Templates	7
3.1.4	Properties	7
3.2	Router	8
4	Konzept	9
4.1	Use-Cases der Anwendung	9
4.2	Systementwurf	9
5	Implementierung	11
5.1	Components	11
5.2	Router	11
6	Zusammenfassung und Ausblick	12
	Literaturverzeichnis	13
	Abbildungsverzeichnis	14
	Selbstständigkeitserklärung	15

1 Einleitung

2 Untersuchung der Hochschul Website

Studenten die Informationen zu ihrem Studiengang suchen brauchen teilweise sechs Klicks um von der Startseite der HS Wismar zur Semesterübersicht (AIMMT) zu gelangen. Innerhalb dieser Seiten gibt es einige Unstimmigkeiten und fehlerhaftes Fehlerhaftes Verhalten, welches im folgendem näher erläutert wird.

2.1 Analyse des Ist-Zustandes

Die Hochschulwebsite hat einen großen allgemeinen Teil der Informationen über die Hochschule enthält und hat Verlinkungen zu den drei Fakultätsseiten. Da die entwickelte Webanwendung für Studierende des Studiengangs Angewandte Informatik und Multimendiatechnik gedacht ist wird an dieser Stelle nur die Fakultätsseite der Ingenieurstechnik bzw. des Bereiches Elektrotechnik und Information.

In der oberen Navigationsleiste gibt es nicht eindeutig erkennbare Icons für den Schnelleinstig und Informationen. Werden diese angeklickt, klappt sich ein Panel nach oben aus. Die obere Navigationsleiste ist nicht fixiert. Der Footer ist sehr groß und enthält teilweise die selben Verlinkungen wie in der Seitennavigation oben. Auf der Seite der Semesterübersicht gib es eine Akkordeonmenü welches nicht an allen Punkten anklickbar ist. Dieses Menü enthält ein Pfeil-Icon nach unten zeigend, welches dem User suggeriert das sich hinter dem Icon mehr Informationen enthalten. Dieses Icon ist aber nicht anklickbar. Die Pfeil-Icons gibt es auf der gesamten Website als wiederkehrendes Symbol für Verlinkungen. Auf den Informationsseiten werden diese Icons allerdings auch als Auflistungszeichen verwendet. Weiterhin gibt es viel zu viele Querverweise. Beispielsweise auf einer Modulübersichtsseite gibt es Verlinkungen zu weiteren Studiengängen die dieses Modul besuchen, weitere Module die der/die Professor |in unterrichtet, die Forschungsthemen, Thesenthemen, und Jobs die der/die Professor|in anbietet. Diese Informationen gibt als sowohl auf der Modulübersicht als auch auf der persönlichen Seite der/des Professor|in. Der User wird überfordert, da zu viele Informationen gegeben werden. Desweiteren fehlt die

Auflistung der Wahlpflichtmodule. Diese müssen umständlich im Modulhandbuch gesucht werden.

2.2 Festlegung Systemanforderung

Die Hochschulwebsite ist sowohl für Studieninteressierte, als auch für eingeschriebene Studierende konzipiert. Um schneller an die wichtigsten Informationen zu gelangen, soll die Anwendung nur für eingeschriebene Studierende des Studiengangs Angewandte Informatik und Multimediatechnik dienen. Weiterhin sollen nur die wichtigsten Information kurz und knapp dargestellt werden. Daher hat jede Seite ein klares Ziel. So ergeben sich weniger Verlinkungen, sodass der User immer genau weiß wo er sich befindet. Die heutigen Studierenden gehören zur Generation Smartphone. Daher soll der Ansatz "mobile first" verfolgt werden.

3 Grundlagen

3.1 Webcomponents

Web Components stellen eine Reihe von Webplattform-APIs dar. Mit Ihnen kann wiederverwendbares gekapseltes HTML erstellt und erweitert werden. Das HTML wird in Komponenten gegliedert die in jedem modernen Browser und jeder JavaScript-Bibliothek und jedem Framework verwendet werden kann. Web Components basieren auf vier Anforderungen:

- *Custom Elements*: Grundlage für das Erstellen neuer DOM-Elemente
- *Shadow Dom*: Definiert wie gekapseltes HTML in den Components verwendet wird
- *ES Modules*: Definiert die Einbindung und Wiederverwendung der JavaScript-Dateien
- *HTML Templates*: Definiert HTML-Fragmente die erst zur Laufzeit instanziiert werden

Es existieren viel Bibliotheken mit denen die Erstellung von Web Components erleichtert wird, wie zum Beispiel Hybrids, LitElement, Polymer usw. Im folgenden wird die Bibliothek LitElement näher erläutert. [1]

3.1.1 LitElement

LitElement ist eine Basisklasse zum Erstellen von Web Components. Sie verwendet lit-html um Templates zu definieren und zu rendern und fügt eine API zum Verwalten von Eigenschaften und Attributen hinzu. Die Eigenschaften werden beobachtet und die Elemente werden asynchron aktualisiert, wenn sich ihre Eigenschaft ändert.

3.1.2 Shadow-Dom

Der SHadow-Dom wird verwendet um den Template-Dom zu Kapseln. Er bietet drei wesentliche Vorteile:

- *Dom-Scoping*: DOM-APIs finden keine Element im Schatten-Dom sodass globale Scripte keinen Zugriff haben
- *Style-Scoping*: gekapselte Styles haben keine Auswirkung auf den Rest des DOM-Baumes
- *Composition*: der Schatten-Dom der Komponente ist von untergeordneten Elementen getrennt, so kann gesteuert werden wie untergeordnete Elemente in das Template gerendert werden sollen [1]

3.1.3 Templates

In einer Render-Funktion der Elementklasse wird das Template für die Component definiert. In dieser Funktion wird das rohe HTML in einem JavaScript template literal innerhalb von back-ticks geschrieben (siehe Abb.). Die Render Methode kann alles zurückgeben was lit-html rendern kann. [2]

```
render() {  
  return html`  
    <p id="message">Loading</p>  
  `;  
}
```

Abbildung 3.1: Render-Funktion
[2]

3.1.4 Properties

LitElement verwaltet deklarierte Eigenschaften in einem statischen properties Getter, die entsprechenden Attribute werden in einem Elementkonstruktor initialisiert. So kann sicher gestellt werden das das bei einer geplanten Elementaktualisierung sich die deklarierte Eigenschaft ändert. [2]

3.2 Router

Der Router übernimmt das Navigieren auf der gesamten Seite. Er legt die angezeigte URL fest und verwaltet welche SPA (mit dazugehörigen Webcomponents) dafür angezeigt wird. Die verschiedenen Routen werden mit Namen und einem URL Pattern angelegt und können auf diesem Wege auseinander gehalten werden. Über Parameter und Querys können dann auch Informationen übertragen werden, die für die neu aufgerufene Seite von belangen sind.

Wird nun ein Button mit einem Redirect auf eine neue SPA belegt, registriert auch der Router den Klick und wechselt die Path Variable auf den entsprechenden Namen, welcher zu der neuen URL zugehörig ist. Anhand dieser Variable können alle nicht benötigten SPA's ausgeblendet und die neue SPA angezeigt werden.

4 Konzept

4.1 Use-Cases der Anwendung

Das in Abbildung 4.1 gezeigte Use-Case-Diagramm beinhalten zu Beginn zwei mögliche Interaktionen. Der User kann zwischen der Übersicht der Semester oder der Übersicht der Professoren und Mitarbeiter wählen. Wählt der User die Semester Übersicht kann er nun ein Semester wählen und bekommt anschließend die Module des jeweiligen Semesters angezeigt. Entscheidet er sich hier für ein Modul werden ihm die Details des Moduls gezeigt. In der Professoren und Mitarbeiter Übersicht kann der User eine Person auswählen und bekommt dann ebenfalls die Details zu dieser Person gezeigt.

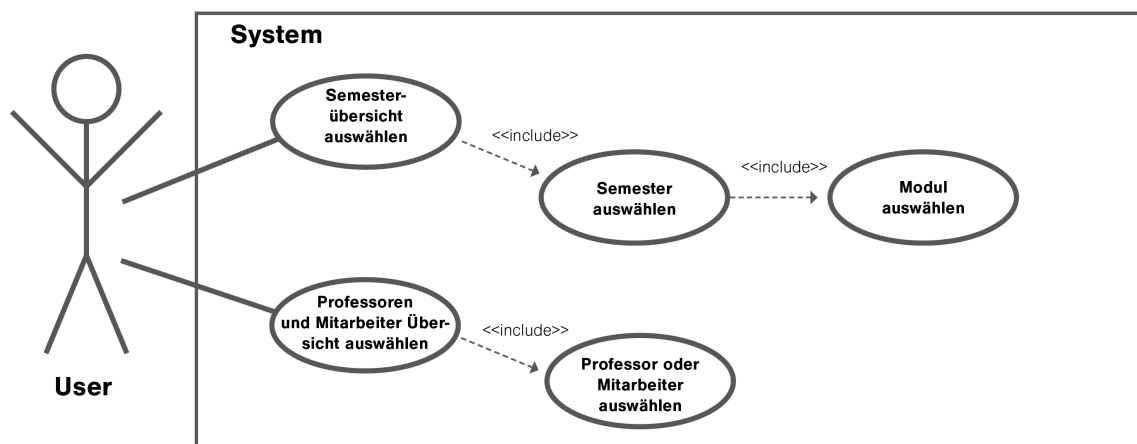


Abbildung 4.1: Use-Case-Diagramm (eigene Darstellung)

4.2 Systementwurf

Die Abbildung 4.2 zeigt die Zustände, in denen sich der User während der Benutzung der Anwendung befinden kann. Im Zustand der Startseite kann durch ein klick auf die gewünschte Übersicht zum einen in den Zustand der Semester Übersicht und zum andern in die Übersicht der Professoren und Mitarbeiter gewechselt werden. Diesen

Zustandswechsel der beiden Übersichten kann jederzeit durch die untere Navigation getätigt werden. Im Zustand der Semester Übersicht kann durch ein klick auf ein Semester in den Zustand der Modul Übersicht gewechselt werden. Wählt der User hier durch ein klick ein Modul aus, wird die entsprechende Detailseite aufgerufen. Im Zustand der Professoren und Mitarbeiter Übersicht kann durch einen klick auf eine Person die entsprechende Detailseite der Person aufgerufen werden.

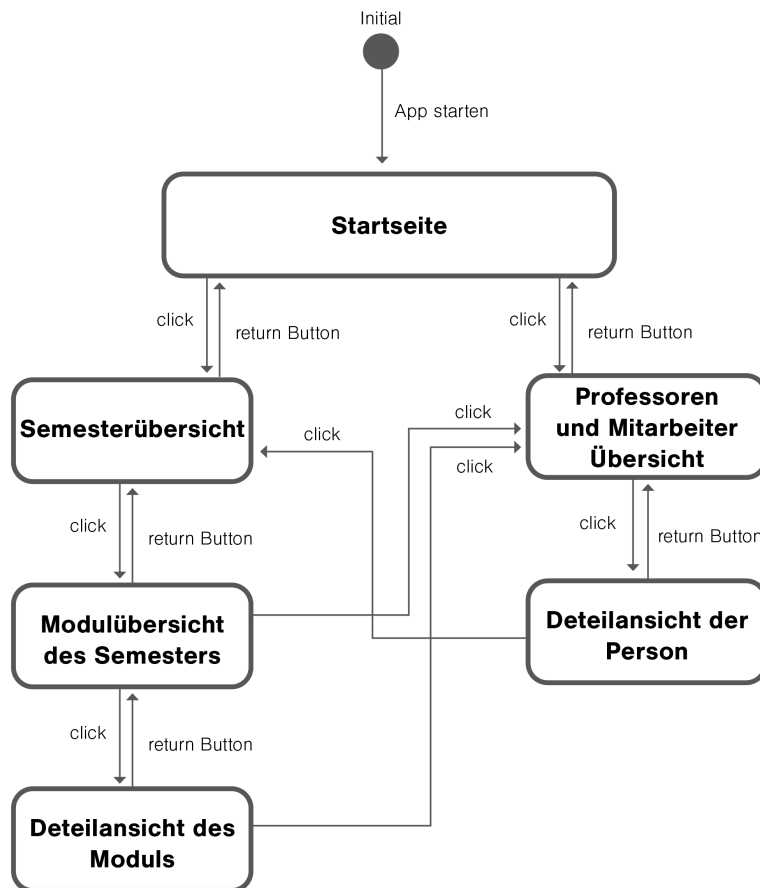


Abbildung 4.2: Zustandsdiagramm (eigene Darstellung)

5 Implementierung

5.1 Components

5.2 Router

6 Zusammenfassung und Ausblick

Literaturverzeichnis

- [1] *Introduction - What are web componts?*
<https://www.webcomponents.org/introduction> [17.05.2020]
- [2] *LitElement* 2018 Polymer Project
<https://lit-element.polymer-project.org/> [17.05.2020]
- [3] *Steam*, Valve Corporation
<https://store.steampowered.com/about/> [19.05.2019]
- [4] *Steam Web API*, Valve Developer Community, 2019
https://developer.valvesoftware.com/wiki/Steam_Web_API [17.05.2019]
- [5] *Eine Graphendatenbank für alle*, Michael Hunger, 2014, Entwickler.Press, ISBN: 978-3-86802-128-0
- [6] *Diese Vorteile bieten Graphdatenbanken*, Kommentar von Stefan Kolmar, 2017
<https://www.bigdata-insider.de/diese-vorteile-bieten-graphdatenbanken-a-615118/> [17.05.2019]

Abbildungsverzeichnis

3.1	Render-Funktion	7
4.1	Use-Case-Diagramm (eigene Darstellung)	9
4.2	Zustandsdiagramm (eigene Darstellung)	10

Selbstständigkeitserklärung

Hiermit erklären wir, dass wir die hier vorliegende Arbeit selbstständig, ohne unerlaubte fremde Hilfe und nur unter Verwendung der aufgeführten Hilfsmittel angefertigt haben.

Ort, Datum

Unterschriften