Kategorie: foo 1

Credit: foo 1

Spezialisierung: foo, auch Vertiefung genannt 1

Modul: foo 1

SPA – **Single Page Application**. 5

Studienordnungen: foo 1

1. Ist-Zustand

Das ursprüngliche Projekt, auf dem unsere Arbeit aufbaut, kann als öffentliches GitHub Repository hier gefunden werden: https://github.com/lost-university/web.

Wir haben uns entschieden, einen Fork des Originalprojekts zu erstellen, um unsere Anpassungen unabhängig davon vornehmen zu können. Dadurch stellen wir sicher, dass spätere Änderungen am Originalprojekt, die nach Beginn unserer Semesterarbeit erfolgen, keinen Einfluss auf unseren Entwicklungsprozess haben.

Ein weiterer Vorteil eines Forks gegenüber einem Branch ist die einfachere Handhabung des Hostings (siehe Section 1.1.3).

1.1. Architektur

1.1.1. Daten

Alle Daten zu Modulen, Kategorien, Spezialisierung und Studienordnungen, welche der Planer nutzt, sind öffentlich https://studien.ost.ch zugänglich.

Die Daten werden von einem Python-Crawler gesammelt, verarbeitet und anschliessend als JSON-Dateien im Data Repository abgelegt.

Die Detailseiten der Studienordnungen dienen dem Crawler als Einstiegspunkt. Informationen zu den geltenden Kategorien und benötigten Credits, den möglichen Spezialisierungen und den zugehörigen Modulen können so entnommen werden.

Die Aktualisierung der Daten erfolgt manuell. Vor Beginn eines jeden Semesters führt ein Maintainer den Crawler lokal aus, überprüft die Änderungen der Daten auf Spezialfälle und erstellt anschliessend ein neues Tag für die Daten. Zuletzt kann die verwendete Version der Daten im Client über den Tag in der URL angepasst werden.

Nachfolgend sind die relevanten Felder für eine Studienordnung gelistet. Die Beispieldaten dazu stammen von https://studien.ost.ch/allStudies/10191_I.json.

```
{
  "kredits": [
    {
      "kategorien": [
        {
          "bezeichnung": "Aufbau",
          "kuerzel": "I_Auf"
        },
        ...
      ],
      "minKredits": 48
    }
  ],
```

```
  "zuordnungen": [
    {
      "bezeichnung": "Application Architecture",
      "kuerzel": "M_AppArch",
      "url": "allModules/28236_M_AppArch.json",
      "istAbschlussArbeit": false,
      "istPflichtmodul": false,
      "semEmpfehlung": 7,
      "kategorien": [
        {
          "bezeichnung": "Aufbau",
          "kuerzel": "I_Auf",
          "kreditpunkte": 4
        },
        ...
      ]
    }
  ],
  "spezialisierungen": [
    {
      "bezeichnung": "Frontend Engineering",
      "kuerzel": "FrEng",
      "url": "allStudies/10193_FrEng.json"
    },
    ...
  ]
}
```

Für jedes Modul wird anhand der `"zuordnungen[].url"` eine Anfrage gestellt, um die Informationen des folgenden Feldes zu erhalten.

```
{
  "kreditpunkte": 4
}
```

Für jede Spezialisierung wird anhand der `"spezialisierungen[].url"` eine Anfrage gestellt, um die Informationen der folgenden Felder zu erhalten.

```
{
  "zuordnungen": [
    {
      "kuerzel": "M_AppArch"
    },
    ...
  ]
}
```

1.1.1.1. Speicherformat der Daten

Das Vorgehen des Crawlers bei der Verarbeitung der zuvor genannten Daten ist im Diagramm Figure 1 ersichtlich.

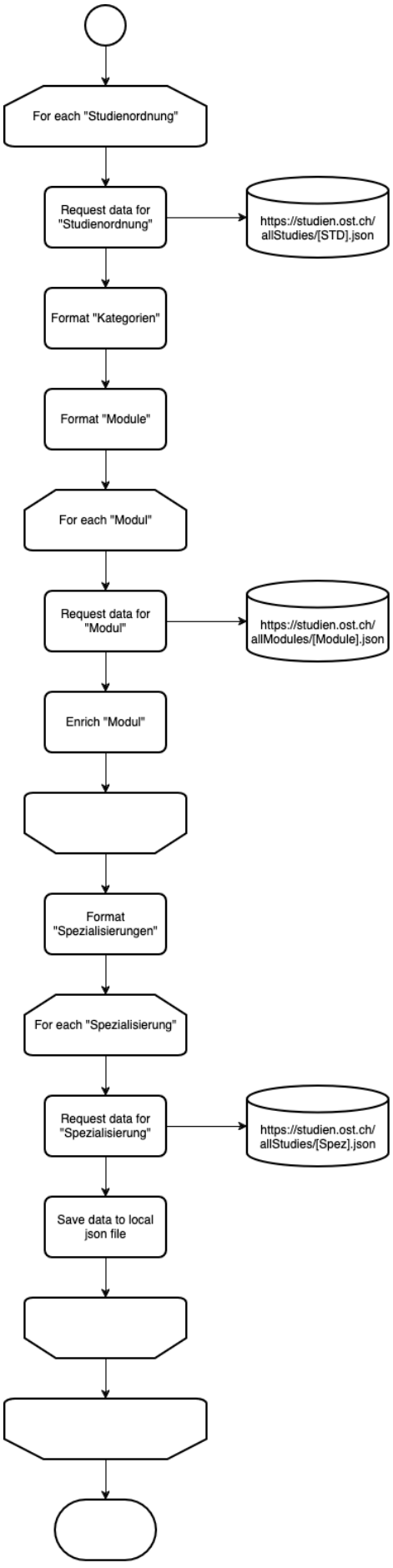

Figure 1: Ablauf des Crawlers

Wie bereits erwähnt, wird das Format der Daten für die weitere Nutzung angepasst.

Module werden in folgendem Format gespeichert.

```
[
  {
    "id": "AppArch",
    "name": "Application Architecture",
    "url": "allModules/28236_M_AppArch.json",
    "isThesis": false,
    "isRequired": false,
    "recommendedSemester": 7,
    "ects": 4,
    "categories_for_coloring": [
      "Auf",
      "Inf"
    ]
  },
  ...
]
```

Kategorien in folgendem Format.

```
[
  {
    "id": "Auf",
    "required_ects": 48,
    "name": "Aufbau",
    "modules": [
      {
        "id": "AppArch",
        "name": "Application Architecture",
        "url": "allModules/28236_M_AppArch.json"
      },
      ...
    ]
  },
  ...
]
```

Spezialisierungen in folgendem Format.

```
[
  {
    "id": "FrEng",
    "url": "allStudies/10193_FrEng.json",
    "name": "Frontend Engineering",
    "modules": [
      {
        "id": "AppArch",
        "name": "Application Architecture",
        "url": "allModules/28236_M_AppArch.json"
      },
      ...
    ]
  },
  ...
]
```

1.1.2. Code

Die Applikation selbst ist eine Single Page Application (SPA), entwickelt mit Vue. Anstelle von JavaScript wird dabei TypeScript verwendet. Die Icons kommen von Fontawesome, das Styling wird grösstenteils über Tailwind gemacht. Als Build-Tool kommt Vite zum Einsatz.

Vor Beginn unserer Arbeit wurde mit dem Stakeholder, welcher gleichzeitig Haupt-Maintainer ist, vereinbart, dass dieser Tech-Stack im Verlauf unserer Arbeit unverändert bleibt.

1.1.3. Hosting

Die Applikation wird über GitHub Pages gehostet. Da sie lediglich eine SPA ohne Backend ist, entfallen somit jegliche Kosten für das Hosting.

Die gewünschte URL, lost.university, wird auf GitHub unter Settings -> Pages -> Custom Domain hinterlegt. Die Domain wird so konfiguriert, dass sie über einen CNAME-Eintrag auf das Repository der Applikation verweist.

Wird ein Branch in den main-Branch gemergt, wird über einen GitHub Workflow eine GitHub Action ausgelöst, welche die SPA baut und deployt.

Anstelle einer Datenbank wird das Data Repository verwendet: https://github.com/lost-university/data. Die darin enthaltenen JSON-Dateien werden mithilfe von Tags versioniert.

1.2. Funktionalität

Folgende Funktionalitäten bestanden bereits vor Beginn dieser Semesterarbeit.

Als User kann ich über ein Dropdown einen Musterstudienplan auswählen, der vorausgefüllt angezeigt wird, sodass ich diesen als Grundlage für meinen eigenen Plan verwenden und bei Bedarf anpassen kann. Es stehen die Musterstudienpläne für alle Spezialisierungen, jeweils für das Teilzeit- und Vollzeit-Modell, zur Verfügung.

- Figure 2
- Als User sehe ich eine visuelle Repräsentation meines Planes.
  - Figure 3
- Als User kann ich Semester im Plan hinzufügen und entfernen, um meine Studiendauer abzubilden.
- Als User kann ich einem Semester ein Modul hinzufügen, das Modul beliebig verschieben und auch entfernen, um meinen Modulplan zu gestalten.
- Als User erhalte ich eine Fehlermeldung, wenn ich versuche, ein Modul einem zweiten Semester hinzuzufügen, um ungültige Pläne zu verhindern.
  - Figure 4
- Als User sehe ich, wie viele Credits ein geplantes Modul wert ist, und erkenne anhand der Farbgebung auch, zu welcher Kategorie es gehört.
- Als User kann ich durch einen Klick auf den Namen eines geplanten Modules zur entsprechenden Modulbeschreibung auf Adunis gelangen, um dort weitere Informationen abzurufen.
- Als User sehe ich die Summe der Credits aller Module eines Semesters in meinem Plan.
- Als User kann ich optional mein Startsemester eingeben, damit die Semester mit einem passenden Namen beschriftet, die bereits erreichten, geplanten und noch benötigten Credits pro Kategorie dargestellt und nur mögliche Spezialisierungen angezeit werden.
  - Figure 5

- Als User sehe ich, welche Module noch benötigt werden, um eine Spezialisierung zu erreichen.
  - Figure 6
- Als User erhalte ich eine Fehlermeldung, wenn mein Plan Module enthält, die nicht korrekt aufgelöst werden können. Diese Meldung gibt mir die Möglichkeit, das betroffene Modul aus meinem Plan zu entfernen.
  - Figure 7
- Als User sehe ich Memes im Planer, um die Stimmung beim Planen aufzulockern.
- Als Maintainer oder potentieller Maintainer sehe ich die Namen anderer Maintainer, die mit ihren GitHub-Profilen verlinkt sind, sowie einen Link zur GitHub-Seite des Planers, um mich zur Mitarbeit zu motivieren.
- Die Module, gruppiert nach Semester, und das Startsemester werden in der URL als Queryparam gespeichert.
- Die URL zum Plan wird im LocalStorage gespeichert.
- Beim Öffnen eines leeren Planes, wird der Plan aus dem LocalStorage verwendet, wenn ein solcher existiert.
- Beim Öffnen eines Plans, der den alten Namen eines Modules enthält, wird der Modulname automatisch auf den neuen Namen migriert.

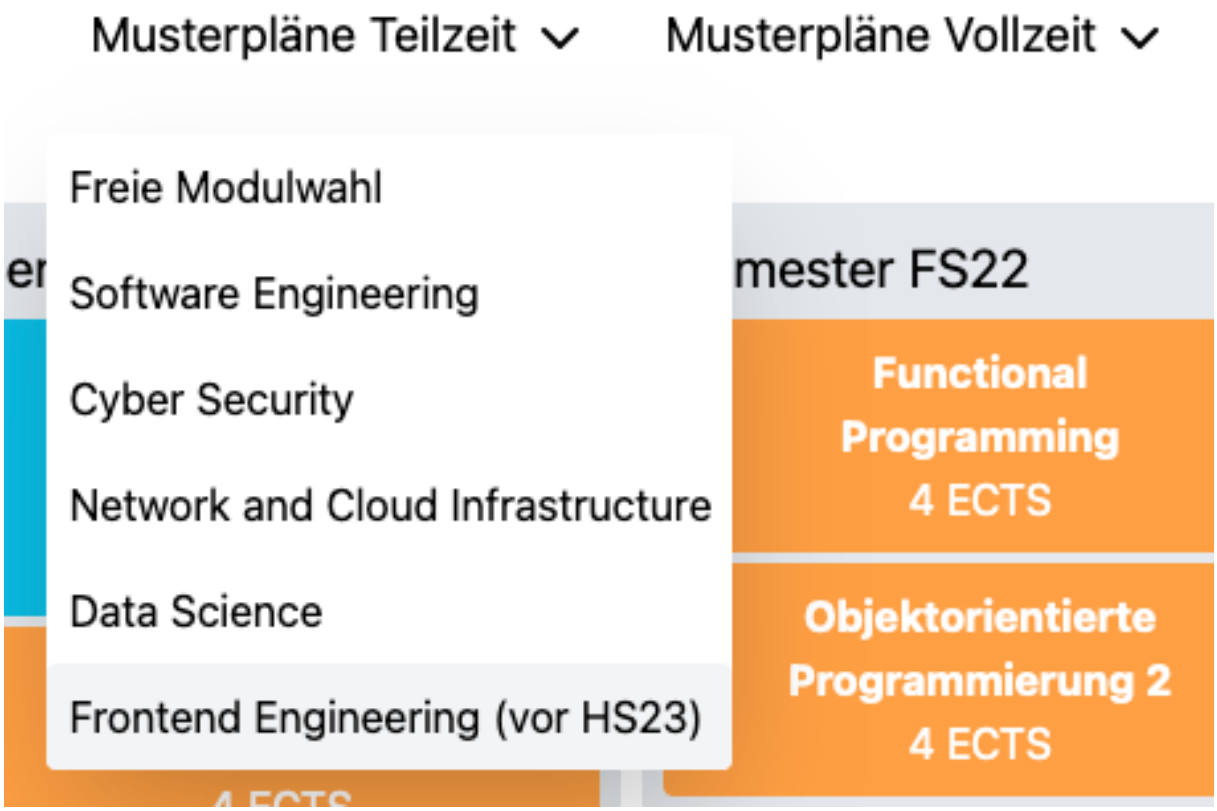

Figure 2: Dropdown zur Auswahl eines Musterstudienplans.

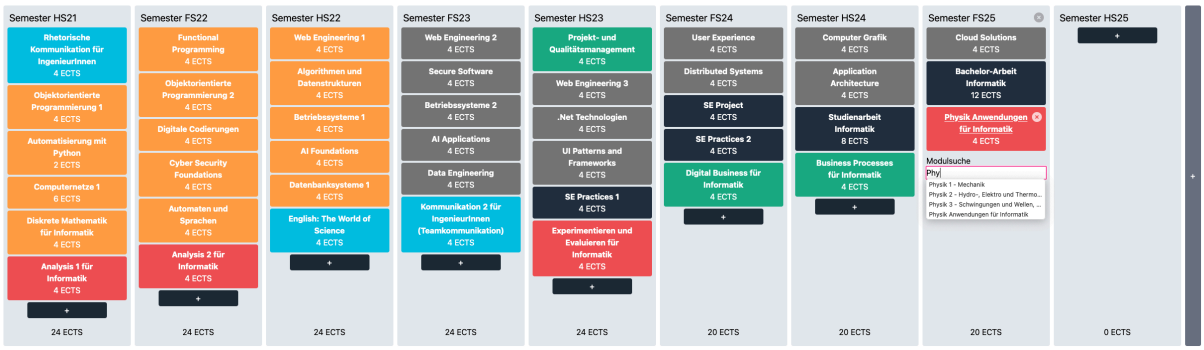

Figure 3: Ein Beispiel für einen Plan.

Figure 4: Die Fehlermeldung beim erneuten Hinzufügen eines Modules.

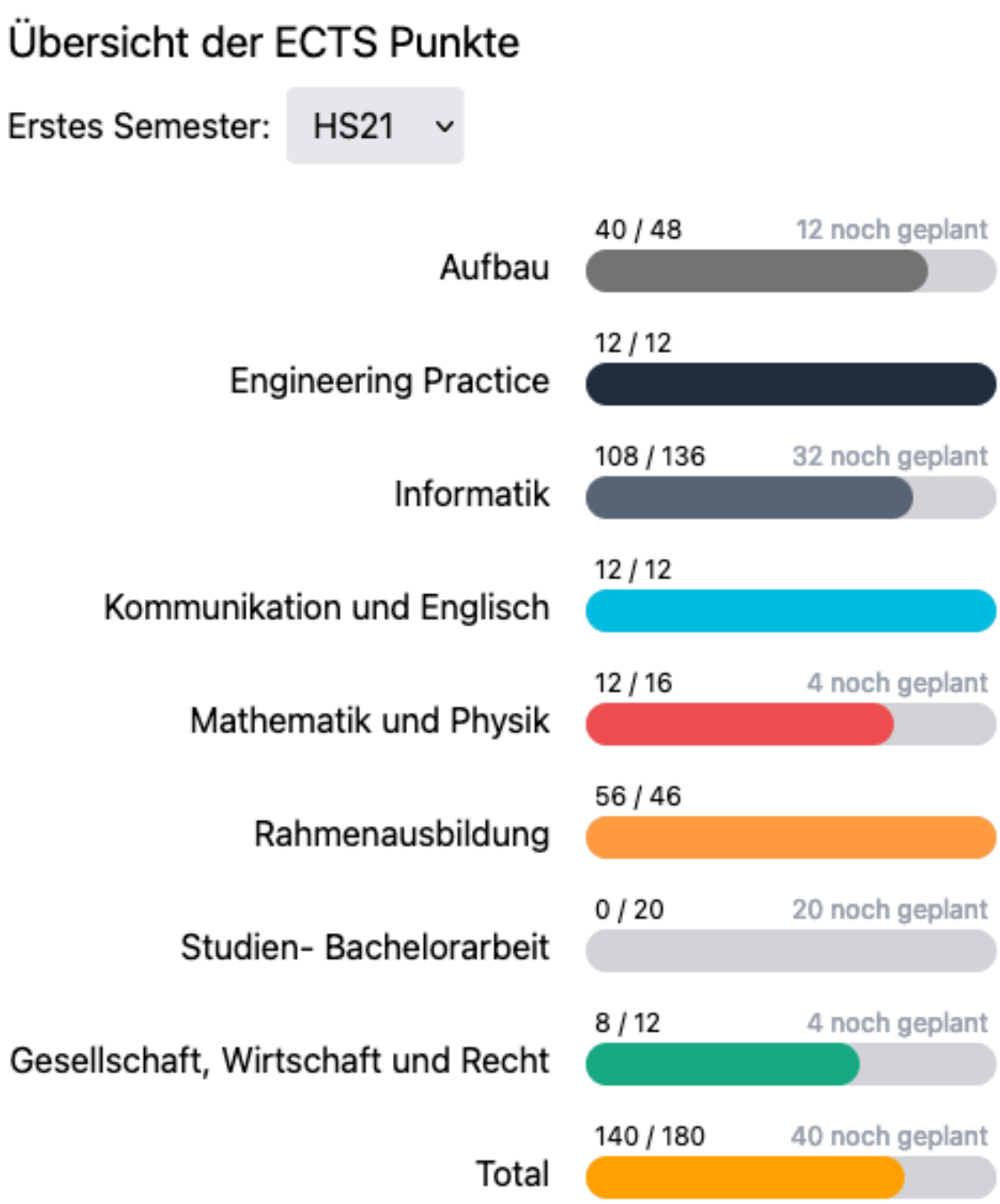

Figure 5: Die Auswahl des Startsemesters und die Übersicht der erreichten, geplanten und nötigen Credits pro Kategorie.

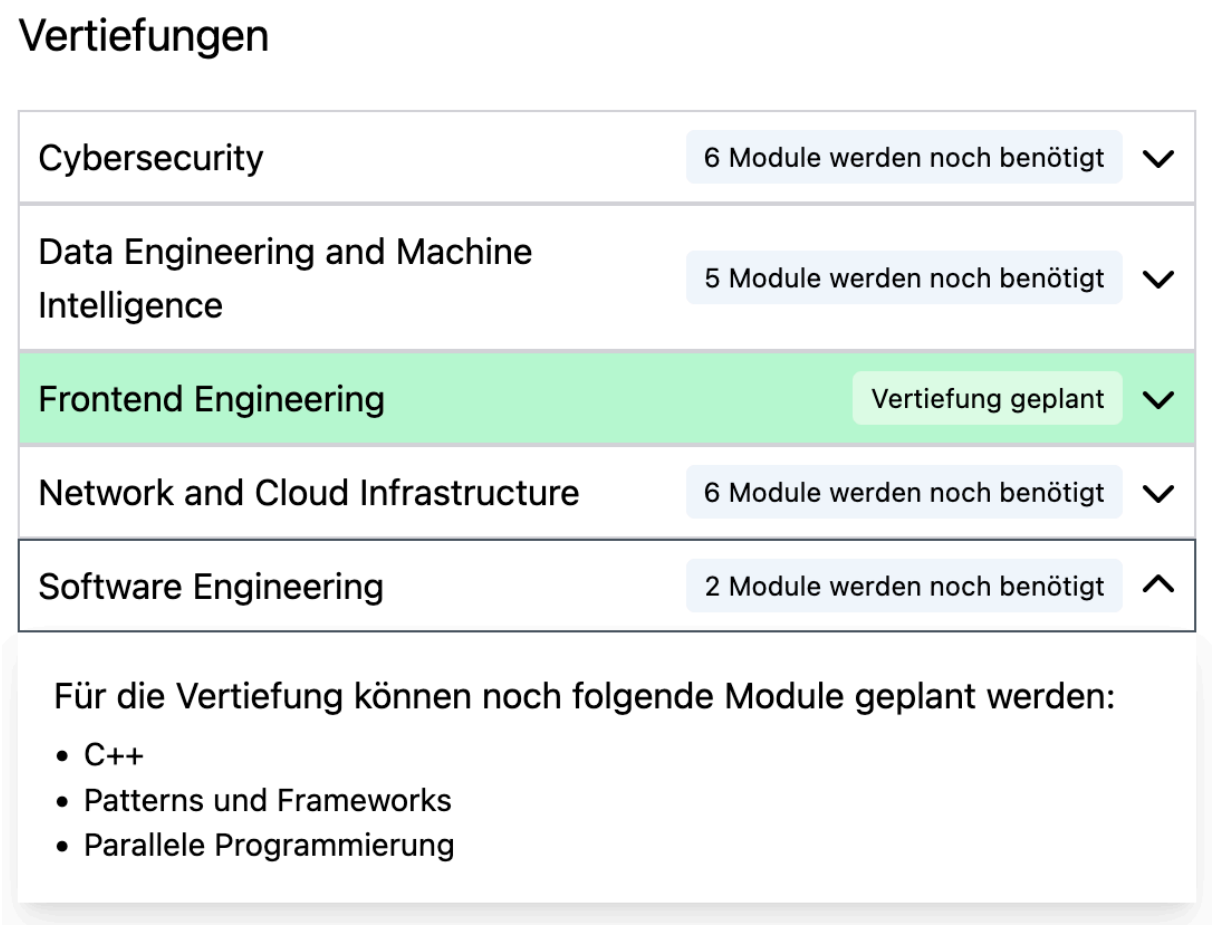

Figure 6: Die Spezialisierungen, mit Hinweis zu deren Erreichung.

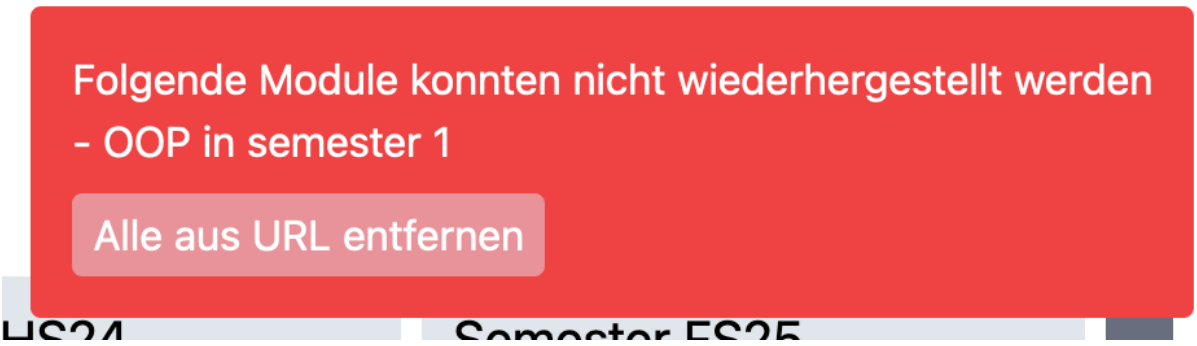

Figure 7: Die Fehlermeldung bei einem unbekannten Modul in der URL.

1.3. Probleme

Im folgenden werden alle Probleme und Bugs dokumentiert, die bereits vor Beginn dieser Semesterarbeit in der Applikation vorhanden waren. Im Verlauf dieser Arbeit werden nur jene Probleme und Bugs behandelt, die im Rahmen des User-Centered-Design-Ansatzes als relevant identifiziert wurden. Die übrigen fallen nicht in den Umfang dieser Semesterarbeit.

- Suche
- Mobile
- Adunis?
- Migration von Daten