



## **Abstract**

# Contents

Abstract .....	2
Ist-Zustand .....	4
Architektur .....	4
Daten .....	4
Code .....	8
Hosting .....	8
Funktionalität .....	8
Probleme .....	12
Bibliography .....	13

## Ist-Zustand

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

Wir haben uns dazu entschieden, einen Fork von diesem Projekt zu erstellen, auf welchem wir unsere Anpassungen durchführen. Dies damit allfällige Änderungen am Original, welche nach dem Start unserer Arbeit anfallen, keinen direkten Einfluss auf unsere Arbeit haben.

Ein eigener Fork hat auch den Vorteil gegenüber einem Branch, dass das Hosting einfacher zu bewerkstelligen ist.

## Architektur

### Daten

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

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

Die Detailseiten der Studienordnungen dienen hier als Einstiegspunkt. Informationen zu den geltenden Kategorien und benötigten Credits, die möglichen Spezialisierungen und die zugehörigen Module können so entommen werden.

Die Aktualisierung der Daten erfolgt manuell. Vor Beginn eines jeden Semester 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.

Folgend sind die relevanten Felder für eine Studienordnung gelistet, mit Beispielesdaten anhand [https://studien.ost.ch/allStudies/10191\\_I.json](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 gemacht, woraus das folgende Feld relevant ist.

```

{
    "kreditpunkte": 4
}

```

Für jede Spezialisierung wird anhand der "spezialisierungen[].url" eine Anfrage gemacht, woraus die folgenden Felder relevant sind.

```

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

```

Das Diagramm Figure 1 stellt den Ablauf, bei dem der Crawler die oben erwähnten Daten verarbeitet, grob dar.

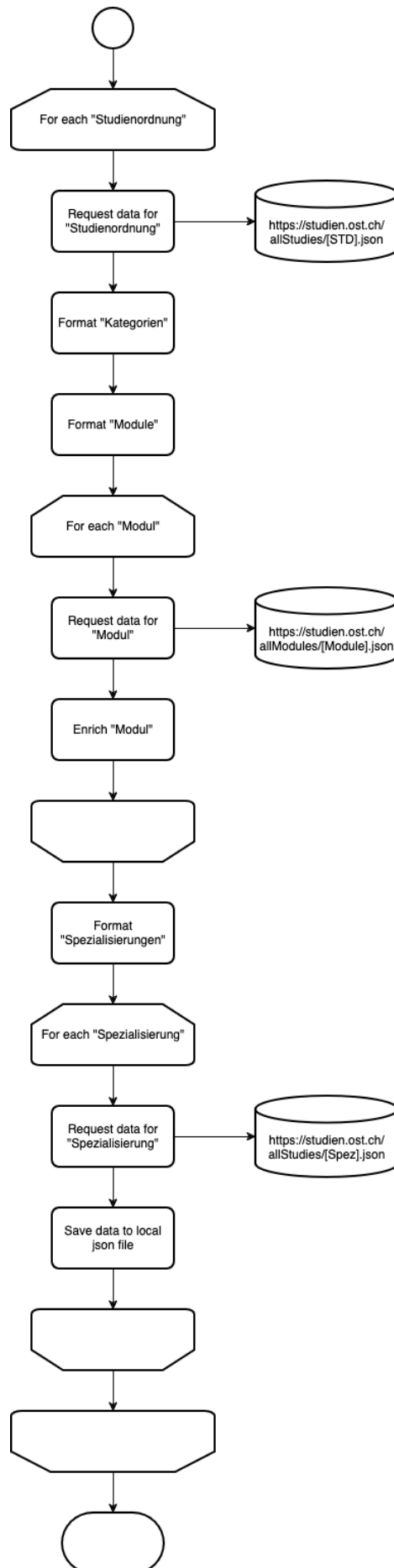


Figure 1: Ablauf des Crawlers

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"
      },
      ...
    ]
  },
  ...
]
```

## Code

Die Applikation selbst ist eine Single Page Application (SPA), geschrieben in Vue. Es wird Typescript anstatt JavaScript verwendet. Die Icons kommen von Fontawesome, das Styling wird grösstenteils über Tailwind gemacht. Als Build Tool wird Vite eingesetzt.

Vor dem Start unserer Arbeit wurde mit dem Stakeholder, welcher auch Haupt-Maintainer ist, besprochen, dass sich dieser TechStack nicht durch unsere Arbeit verändern soll.

## 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](https://lost.university), wird auf GitHub unter Settings -> Pages -> Custom Domain hinterlegt. Bei der Domain selbst wird die Repository als ein CNAME hinterlegt.

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

Als Alternative zur einer Datenbank wird das Data Repository verwendet: <https://github.com/lost-university/data>. Die verwendeten JSON Dateien werden über Tags versioniert.

## Funktionalität

Folgende Funktionalitäten bestanden bereits vor Beginn unserer Arbeit.

- Als User kann ich über ein Dropdown einen Musterplan wählen und diesen anzeigen lassen, um meinen Plan darauf basierend aufzubauen. Es stehen die Musterpläne für alle Spezialisierungen, jeweils für das Teilzeit- und Vollzeit-Modell, zur Verfügung.
  - Figure 2
- Als User sehe ich eine visuelle Representation meines Planes.
  - Figure 3
- Als User kann ich ein Semester im Plan hinzufügen und entfernen, um meine Studiendauer abzubilden.
- Als User kann ich in einem Semester ein Modul hinzufügen und entfernen, um meinen Modulplan zu gestalten.
- Als User erhalte ich eine Fehlermeldung, wenn ich versuche ein Modul in einem zweiten Semester hinzuzufügen, um ungültige Pläne zu vermeiden.
  - Figure 4
- Als User sehe ich wieviele Credits ein geplantes Modul wert ist und anhand der Farbe auch, zu welcher Kategorie es gehört.
- Als User kann ich über einen Klick auf den Namen eines geplanten Modules zu dessen Modulbeschreibung auf Adunis gelangen, um mir dort weitere Informationen zu holen.
- 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 angezeigt werden.
  - Figure 5
- Als User sehe ich bei einer Spezialisierung, welche Module noch zu deren Erreichung benötigt werden.



- Figure 6
- Als User erhalte ich eine Fehlermeldung, sollte mein Plan Module enthalten, die nicht korrekt aufgelöst werden können. Ich habe durch diese Fehlermeldung die Möglichkeit, dieses Modul aus meinem Plan zu entfernen.
- Figure 7
- Als User sehe ich ein paar Memes im Planer, um die Stimmung auch beim Planen noch locker zu halten.
- Als Maintainer oder potentieller Maintainer sehe ich die Namen anderer Maintainer, mit Verlinkung zu ihrem GitHub-Profil, und einem 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 Planes, der den alten Namen eines Modules enthält, wird dieser automatisch zum neuen Namen migriert.

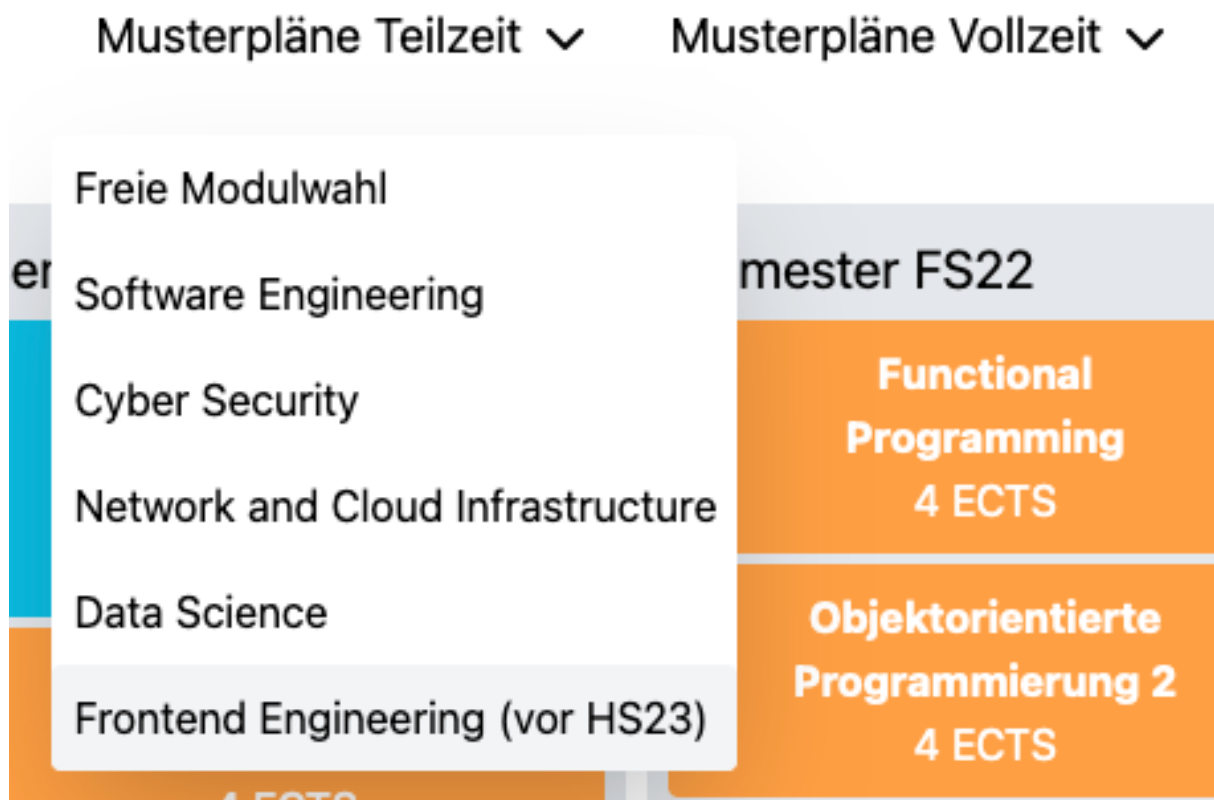


Figure 2: Dropdown zur Auswahl eines Musterplanes.

Semester HS21	Semester FS22	Semester HS22	Semester FS23	Semester HS23	Semester FS24	Semester HS24	Semester FS25	Semester HS25
Rhetorische Kommunikation für Ingenieurinnen 4 ECTS	Functional Programming 4 ECTS	Web Engineering 1 4 ECTS	Web Engineering 2 4 ECTS	Projekt- und Qualitätsmanagement 4 ECTS	User Experience 4 ECTS	Computer Grafik 4 ECTS	Cloud Solutions 4 ECTS	+
Objektorientierte Programmierung 1 4 ECTS	Objektorientierte Programmierung 2 4 ECTS	Algorithmen und Datenstrukturen 4 ECTS	Secure Software 4 ECTS	Web Engineering 3 4 ECTS	Distributed Systems 4 ECTS	Application Architecture 4 ECTS	Bachelor-Arbeit Informatik 12 ECTS	
Automatisierung mit Python 2 ECTS	Digitale Codierungen 4 ECTS	Betriebssysteme 1 4 ECTS	Betriebssysteme 2 4 ECTS	.Net Technologien 4 ECTS	SE Project 4 ECTS	Studienarbeit Informatik 8 ECTS	Physik Anwendungen für Informatik 4 ECTS	
Computernetze 1 6 ECTS	Cyber Security Foundations 4 ECTS	AI Foundations 4 ECTS	AI Applications 4 ECTS	UI Patterns and Frameworks 4 ECTS	SE Practices 2 4 ECTS	Business Processes für Informatik 4 ECTS	Modulsuche	
Diskrete Mathematik für Informatik 4 ECTS	Automaten und Sprachen 4 ECTS	Datenbanksysteme 1 4 ECTS	Data Engineering 4 ECTS	SE Practices 1 4 ECTS	Digital Business für Informatik 4 ECTS	+	Phy	
Analysis 1 für Informatik 4 ECTS	Analysis 2 für Informatik 4 ECTS	English: The World of Science 4 ECTS	Kommunikation 2 für Ingenieurinnen (Teamkommunikation) 4 ECTS	Experimentieren und Evaluieren für Informatik 4 ECTS	+		Physik 1 – Mechanik Physik 2 – Hydro, Elektro und Thermo... Physik 3 – Schwingungen und Wellen, ... Physik Anwendungen für Informatik	
+	+	+	+	+				
24 ECTS	24 ECTS	24 ECTS	24 ECTS	24 ECTS	20 ECTS	20 ECTS	20 ECTS	0 ECTS

Figure 3: Ein Beispiel für einen Plan.

Modul Physik Anwendungen für Informatik ist bereits im Semester 8

Figure 4: Die Fehlermeldung beim erneuten Hinzufügen eines Modules.

## Übersicht der ECTS Punkte

Erstes Semester: **HS21** ▾

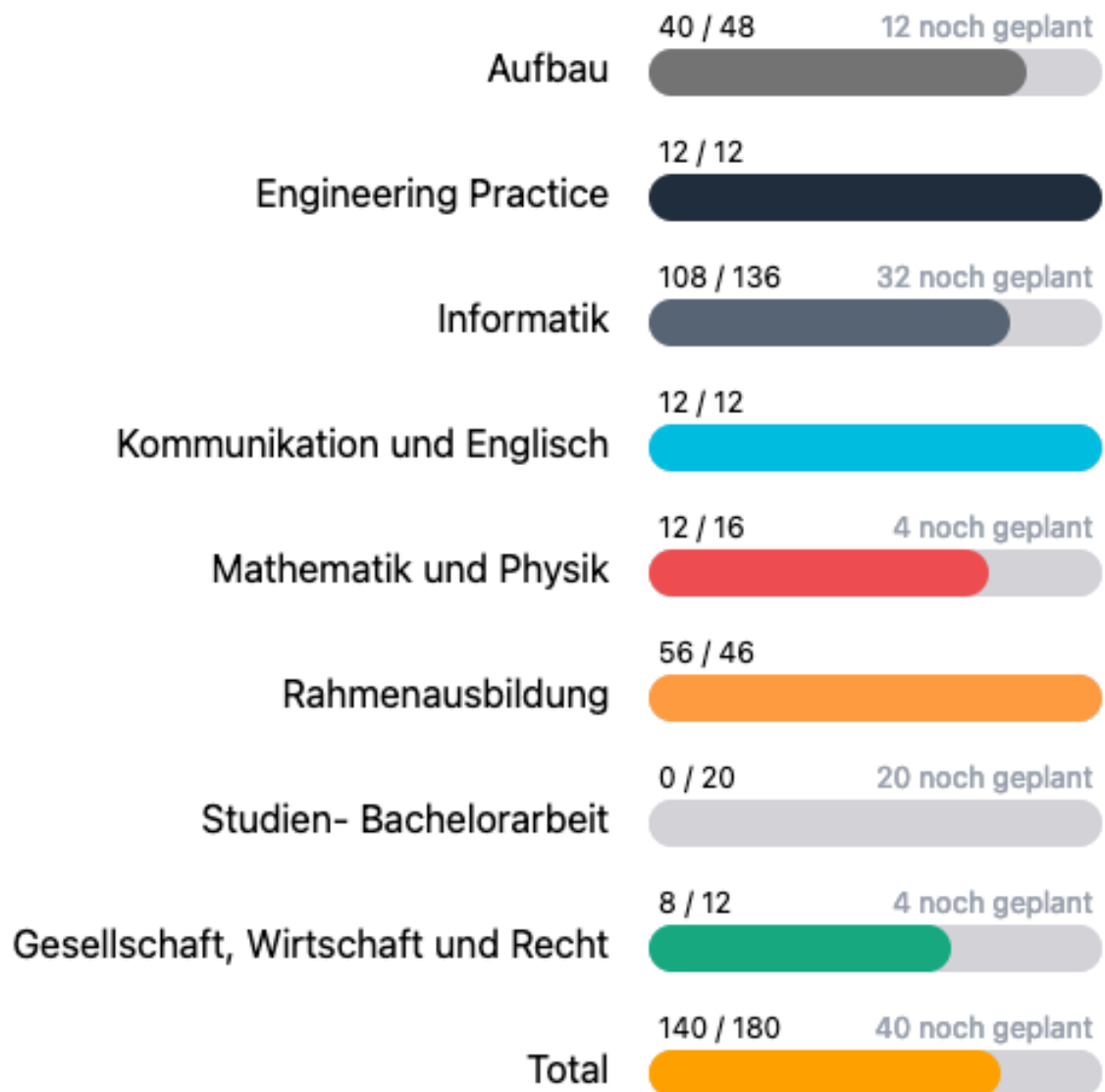


Figure 5: Die Auswahl des Startsemesters und die Übersicht der erreichten, geplanten und nötigen Credits pro Kategorie.

## Vertiefungen

Cybersecurity	6 Module werden noch benötigt	▼
Data Engineering and Machine Intelligence	5 Module werden noch benötigt	▼
Frontend Engineering	Vertiefung geplant	▼
Network and Cloud Infrastructure	6 Module werden noch benötigt	▼
Software Engineering	2 Module werden noch benötigt	▲

Für die Vertiefung können noch folgende Module geplant werden:

- C++
- Patterns und Frameworks
- Parallele Programmierung

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

Folgende Module konnten nicht wiederhergestellt werden  
- OOP in semester 1

Alle aus URL entfernen

US24 Semester ES25

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

### Probleme

Im folgenden werden alle Probleme und Bugs dokumentiert, die bereits zu Beginn unserer Arbeit bestanden, aber nicht im Rahmen unserer geplanten Anpassungen behoben werden sollen. Sollte genügend Zeit nach Erreichen unseres eigentlichen Zieles übrig sein, könnten wir diese ebenfalls noch angehen.

- Suche
- Mobile
- Adunis?
- Migration von Daten

# **Bibliography**

*Kategorie:* foo 4

*Credit:* foo 4

*Spezialisierung:* foo, auch Vertiefung genannt 4

*Modul:* foo 4

**SPA – Single Page Application.** 8

*Studienordnungen:* foo 4