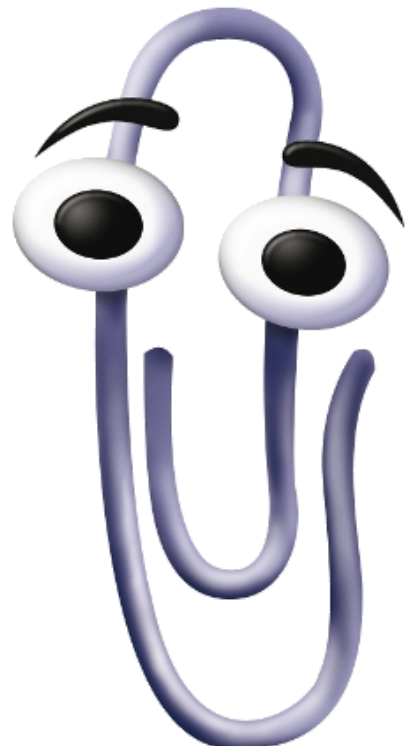


SWT Projekt

Individueller interaktiver Studienplaner

Anforderungsspezifikationen von
Mike Daudrich, Christian Kunkel, Lara Reitz, Pablo Schneider



Gliederung

1 Zielstellung	1
2 Anwendungsszenarien	3
2.1 Erstsemester	
2.2 Fortschrittsregelung	
2.3 Fach ohne Voraussetzungen	4
2.4 Studienplaner einrichten	4
3 Funktionale Anforderungen	5
3.1 Das System MUSS	5
3.2 Das System SOLL	5
3.3 Das System WIRD	5
4 Anwendungsspezifikationen	6
4.1 Anwendungsfalldiagramm	6
4.2 Anwendungsfälle	7
4.2.1 Hochladen von Semesterplänen	7
4.2.2 Auswahl von bestandenen Modulen	8
4.2.3 Infos zu Modulen anzeigen lassen	9
4.2.4 Abhängigkeiten anzeigen lassen	10
4.2.5 Auswahl von Modulen	11
4.2.6 Änderung von Einträgen	12
4.2.7 Studiengangwechsel	13
4.2.8 Abspeichern von Semesterplänen	14
5 Nichtfunktionale / technische Anforderungen	15
5.1 Nichtfunktionale Anforderungen	15
5.2 Technische Anforderungen	15
6 Dämonenmodell / Daten	16
6.1 Dämonenmodell	16
6.2 Datentypenverzeichnis	17
7 Benutzungsschnittstelle	18
8 Glossar	23

1 Zielstellung

Das Ziel der Anwendung ist es, Studierende bei der Planung ihres Studiums zu unterstützen. Dabei ist es irrelevant, ob sie sich im ersten oder beliebig hohen Semester befinden.

Es gibt die Möglichkeit per Datei verschiedene Studiengänge zu nutzen. Diese müssen ein bestimmtes Format haben, damit das System sie fehlerfrei einlesen und verarbeiten kann. Damit ist man nicht an einen Studiengang gebunden und kann variabel Fortschrittsregelungen, Abhängigkeiten, CP-Anzahl pro Modul u.v.m. einstellen.

Dazu ermöglicht das System eine modularer Anordnung der einzelnen Fächer, auch Module genannt. Man kann sich diese frei und interaktiv zusammenstellen, wie man möchte. Jedoch gibt es bestimmte Fortschrittsregelungen und Abhängigkeiten zwischen den Modulen. Sobald man solche Regelungen bricht, wird man von dem System gewarnt.

Um der Fortschrittsregelung gerecht zu werden muss man seinen Plan entsprechend anpassen. Dies sieht dann so aus, dass man bestimmte Module für ein Semester gar nicht belegen kann, wenn man andere Module noch nicht abgeschlossen hat. Das System wird einen dann aktiv davon abhalten, dieses Modul in dem Semester zu belegen.

Bei den Modulabhängigkeiten kann man entscheiden, ob man das Risiko eingeht, dass einem Kompetenzen für das Fach fehlen, oder ob man seinen Plan auch dahingehend verändert. Kompetenzen werden nämlich von den meisten Modulen erwartet, das heißt man sollte die entsprechenden Fächer schon erledigt haben, allerdings ist es keine Pflicht. Da Module viele Kompetenzen haben können und diese als Paket geprüft werden heißt es nicht unbedingt, dass wenn man ein Modul nicht bestanden hat, dass man dann auch alle damit verbundenen Kompetenzen nicht hat. So kann man nach eigenem Ermessen entscheiden ob man auch ohne bestandenes Modul andere Module belegt, die darauf aufbauen.

Wenn man ein Modul auswählt, werden einem ergänzende Informationen dazu angezeigt. Darunter gehört, welche Veranstaltungen zu diesem Modul gehören. Ein Modul besteht in der Regel aus der Vorlesung und entweder Praktikum oder Übung. Wenn das Modul ein Praktikum beinhaltet wird nochmal spezifisch angezeigt, dass da dann auch Anwesenheitspflicht gilt. Zudem werden einem dabei auch die Abhängigkeiten zu anderen Modulen hervorgehoben, sodass man direkt weiß, welche Module man vorher erledigt haben sollte.

Man kann bei jedem Modul auch markieren, ob man da die Prüfung und gegebenenfalls auch das Praktikum schon bestanden hat. Das gibt der studierenden Person einen besseren Überblick darüber, welche Module oder Prüfungen noch erledigt werden müssen.

Auch hilfreich dabei ist der CP-Zähler. Dabei wird nicht nur für jedes Modul die zugehörige CP Anzahl angezeigt, sondern es gibt noch zusätzlich einen Summenzähler. Durch das Abhaken von Prüfungs- und Praktikumsleistung wird dieser hochgezählt und man kann so leicht ablesen, wie viele CP man insgesamt bereits gesammelt hat.

2 Anwendungsszenarien

2.1 Erstsemester Lara Reitz

Die Studentin Siegfriede hat gerade erst angefangen zu studieren und wurde auf den Studienplaner aufmerksam gemacht. Zu Beginn erstellt sie sich einen neuen leeren Semesterplan für ihren Studiengang indem sie eine entsprechende Datei mit allen Informationen zu den integrierten Modulen usw. hochlädt. Nachdem dies getan ist schaut sie sich die für ihr Semester angebotenen Fächer und deren CP-Anzahl an und plant sich fröhlich das ganze Semester mit allen möglichen Fächern voll, bis sie von dem System davor gewarnt wird, dass sie die nötigen Kompetenzen für ein spezifisches Modul noch nicht besitzt. Enttäuscht darüber, dass sie nicht alles auf einmal lernen kann, entfernt Sie die zusätzlichen Fächer aus dem zweiten Semester und legt diese schon einmal präventiv für das nächste Semester an. Damit sie auch nicht vergisst was sie alles ausgewählt hat, möchte sie das erste Semester gerne ausgedruckt an Ihrem Schreibtisch haben und konvertiert das Ganze in eine PDF-Datei um es dann auszudrucken. Des Weiteren speichert sie die Datei noch präventiv zur späteren Anpassung als .json Datei ab.

2.2 Fortschrittsregelung Christian Kunkel

Joris ist ein Student der Hochschule RheinMain im Fachbereich Medieninformatik. Er befindet sich im fünften Semester und hat alle Fächer aus dem Ersten bestanden, doch im zweiten Semester fehlt ihm noch Lineare Algebra, um auch dies zu komplettieren. Als er für das aktuelle Semester seinen Plan anlegen will, fällt ihm auf, dass er keine Kurse aus dem fünften Semester auswählen kann. Ein Hinweis bestätigt ihm seine Vermutung. Durch den fehlenden Abschluss des Moduls Lineare Algebra, darf er dieses, durch die Fortschrittsregelung nicht belegen. Da die Fortschrittsregelung in dem Studiengang bei 3 Semestern liegt, wird alles darüber für ihn geblockt, bis er das zweite Semester vollständig bestanden hat. Joris Priorität liegt ab jetzt darauf.

2.3 Fach ohne Voraussetzungen Mike Daudrich

Der Student Friedfert möchte im kommenden Semester das Modul Computergrafik aus dem vierten Semester belegen. Allerdings fehlt ihm Programmieren 3 aus dem dritten Semester. Da Computergrafik die Kompetenz Python aus diesem Modul voraussetzt wird ihm warnend angezeigt, dass er noch diese fehlende Kompetenz hat und daher besser erst Programmieren 3 abschließen sollte. Aufgrund dessen entscheidet sich Friedfert dazu, Computergrafik in diesem Semester erstmal nicht zu machen und wählt es für ein späteres Semester aus. Damit verschwindet auch die Warnung.

2.4 Studienplaner einrichten Pablo Schneider

Die Studentin Joghurta, hat gerade die Semesterplaner-App geladen, und will sie nun einrichten. Dafür downloadet sie eine Textfile von der Hochschuleseite, welche alle Informationen für den Studiengang und dessen Module enthält. Nach dem sie die App geöffnet hat, muss sie die Textfile in die App laden. Danach kann sie fröhlich ihren Studienverlauf planen.

3 Funktionale Anforderungen

3.1 Das System **MUSS**:

- Speicherung von einem personalisierten Plan ermöglichen.
- für verschiedene Studiengänge nutzbar sein.
- Modulare Konfiguration ermöglichen.
- Einträge korrigierbar machen.
- Module nach einer Fortschrittsregelung (wenn gegeben) kontrollieren.
- fehlerhaft eingelesene Dateien erkennen können (Plan Einspeisung).
- zusätzliche Informationen zu einzelnen Modulen anzeigen.

3.2 Das System **SOLL**:

- darauf hinweisen, wenn benötigte Voraussetzungen (Module in Abhängigkeiten) für ein gewisses Modul nicht gegeben sind.
- Abhängigkeiten anzeigen, wenn ein Modul ausgewählt wird. (Fach anklicken -> dependencies werden hervorgehoben)
- einen CP-Zähler für alle bisher bestandenen Module.
- Anwesenheitspflicht für Module Info.
- per spezifischen Design-Regeln ansprechend gestaltet sein.
- einem Module mit geschafft oder durchgefallen markieren lassen können.
- die Möglichkeit des Resets anbieten.
- das Öffnen und Weiterbearbeiten von einem bereits gespeicherten Plan ermöglichen.
- eine TO-PDF oder TO-PRINT-Funktion für den Studienplan anbieten.

3.3 Das System **WIRD**:

- dir deinen Stundenplan vorschlagen oder anpassen, falls du ein geplantes Modul nicht schaffen solltest.
- Werbung für die Partner-Unternehmen der Universitäten / Hochschulen schalten. (Inspiration für Praktika etc.)
- Sportplanung und Sprachkurse integrieren.
- ein gruseliges Maskottchen erhalten (Karl die Klammer).

4 Anwendungsspezifikationen

4.1 Anwendungsfalldiagramm

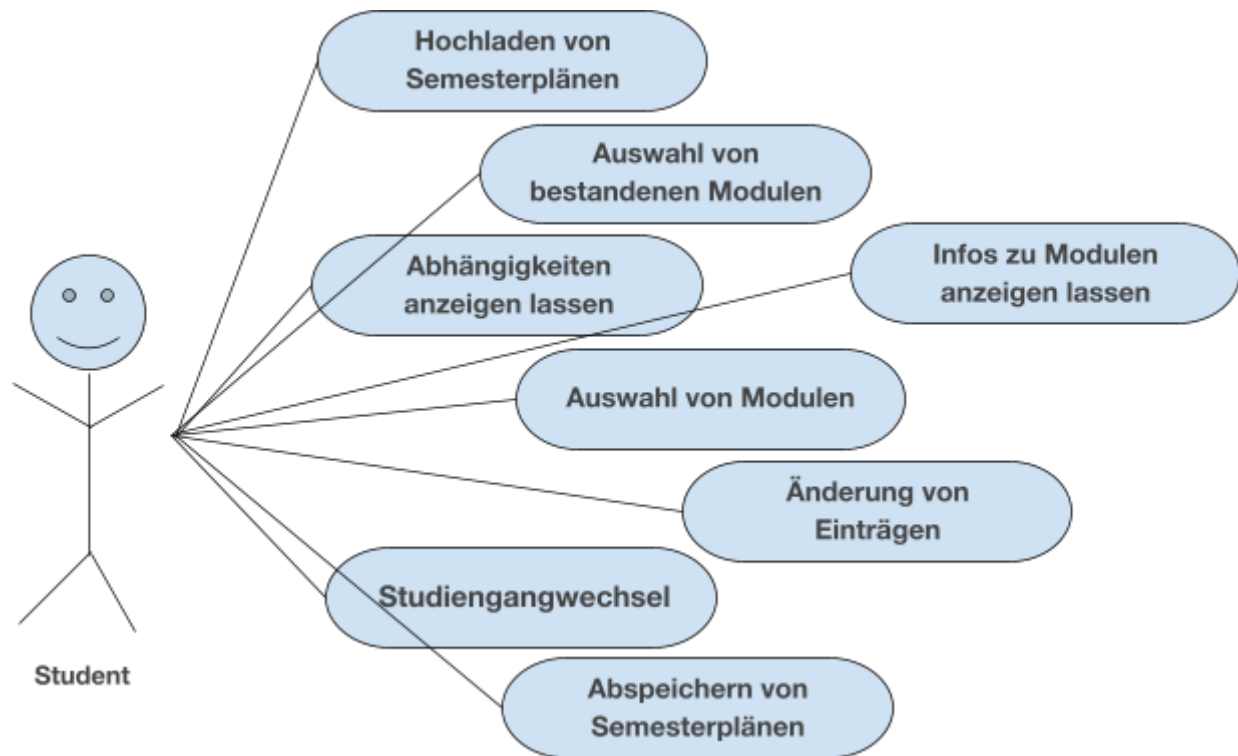


Abbildung 1: UML Diagramm

Das Anwendungsfall-Diagramm (siehe Abbildung 1) stellt alle Anwendungsfälle und den entsprechenden Akteur dazu anschaulich dar. Die einzelnen Fälle werden in den Anwendungsfällen ausführlicher beschrieben.

4.2 Anwendungsfälle

4.2.1 Hochladen von Semesterplänen Lara Reitz

Titel: Hochladen von Semesterplänen

Akteure: Student

Fachlicher Auslöser:

Bearbeiten von bereits erstellten Plänen

Vorbedingungen: Zuvor abgespeicherte Pläne liegen vor. Semesterpläne haben das korrekte Format

Standardablauf:

1. Student/in: Semesterplan, der bearbeitet werden soll, wird ausgewählt und im Programm hochgeladen
2. System: Vollständigkeit und Plausibilität (Format eingehalten?) der hochgeladenen Datei sicherstellen
3. System: Erzeugen eines leeren Studienplans Befüllen des Selbigen mit Modulen, Semestern und Informationen, wie durch die Datei vorgegeben
4. System: Anlage des Semesterplans bestätigen
5. System: Bereitstellung des Studienganges für die Nutzer

Alternative Abläufe / Fehlersituationen / Sonderfälle:

3a. System erkennt fehlerhafte oder unvollständige Dateiinhalte und erstellt keinen entsprechenden Semesterplan und oder Komponenten

3a1. Student bricht den Vorgang ab und versetzt das Programm somit in seinen Ursprungszustand

3a2. Student lädt eine andere Datei hoch oder passt die Vorherige an.

Nachbedingung / Ergebnis:

System geht in Bereitstellungsmodus über. Semesterplan wird angezeigt und kann bearbeitet werden

Nicht-funktionale Anforderungen:

geringer Speicherverbrauch, Performance

Parametrisierbarkeit / Flexibilität:

- Studiengang unabhängig solange Format eingehalten wird
- Anlegen von Vorlagen möglich (Abspeichern und wieder hochladen wenn benötigt)

Nutzungshäufigkeit / Mengengerüst:

- Bei Bedarf, wenn anderer Studiengang oder neuer Plan erstellt werden soll
- Einmal pro Semester im Idealfall
- Bei Änderungen in der Prüfungsordnung oder Wegfallen des Studiengangs

Titel: Auswahl von bestandenen Modulen

Akteure: Student

Fachlicher Auslöser: Planen des aktuellen Semesters

Vorbedingungen: Starten des Programms

Standardablauf:

1. Der Student wählt die bereits bestandenen Module aus
2. Die Module werden nach dem Auswählen farblich markiert, um sie von den noch nicht bestandenen zu differenzieren
3. Die CP des Moduls werden auf die bereits bestandenen drauf gerechnet und angezeigt

Alternative Abläufe / Fehlersituationen / Sonderfälle:

- Es werden keine Module ausgewählt
- Es werden alle Module ausgewählt

Nachbedingung / Ergebnis:

- Man ist nun im Stande sein Semester zu planen anhand der ausgewählten Module

Nicht-funktionale Anforderungen:

- Visuelle Veränderung in unter einer Sekunde

Parametrisierbarkeit / Flexibilität:

Nutzungshäufigkeit / Mengengerüst:

- Immer möglich

Titel: Infos zu Modulen anzeigen lassen

Akteure: Student

Fachlicher Auslöser: mehr infos zu einem Modul, für Planung wissen wollen

Vorbedingungen: Studiengang muss hochgeladen sein.

Standardablauf:

1. Man klickt auf ein Modul
2. Ein Info-View für das Modul erscheint.
3. Man klickt das Infofenster weg.

Alternative Abläufe / Fehlersituationen / Sonderfälle:

- 2.a1 Benutzer liest infos durch.
- 2.a2 Benutzer fügt eine Anmerkung hinzu.
- 2.a3 weiter bei 3.

Nachbedingung / Ergebnis:

-

Nicht-funktionale Anforderungen:

Angezeigte Infos müssen Übersichtlich sein.

Info-View sollte flüssig in der Szene erscheinen und ebenso wieder verschwinden.

Parametrisierbarkeit / Flexibilität:

- eigene Anmerkung hinzufügbar

Nutzungshäufigkeit / Mengengerüst:

- immer nutzbar und so oft man will.

Titel: Abhängigkeiten anzeigen lassen

Akteure: Studi

Fachlicher Auslöser: Mehr Informationen zur sinnvollen Einteilung der Module benötigt

Vorbedingungen: Studiengang wurde ausgewählt

Standardablauf:

1. Studi: wählt Modul aus, über das er mehr Infos möchte
2. System: hebt entsprechendes Modul hervor, sowie alle Module, von denen dieses fachlich abhängt
3. Studi: deselektiert Modul

Alternative Abläufe / Fehlersituationen / Sonderfälle:

- 2a. Modul hat keine Abhängigkeiten
 - 2a1. System zeigt entsprechende Meldung
 - 2a2. Weiter bei 3

Nachbedingung / Ergebnis:

- System geht in den Standardzustand über

Nicht-funktionale Anforderungen:

- Reaktionszeit max. 3 Sekunden

Parametrisierbarkeit / Flexibilität:

- Für jedes Modul Abhängigkeiten einzeln einstellbar

Nutzungshäufigkeit / Mengengerüst:

- Beliebig oft

Titel: Auswahl von Modulen

Akteure: Student

Fachlicher Auslöser: Planen des Semesters

Vorbedingungen: Starten des Programms und bei bedarf Auswahl von bestandenen Module

Standardablauf:

1. Eines der angebotenen Module nehmen
2. Modul in den Planer ziehen
3. Modul an der Gewünschten Position loslassen

Alternative Abläufe / Fehlersituationen / Sonderfälle:

3a1. Modul verstößt gegen eine Regel und bleibt nicht am Drop Ort liegen sondern fällt zurück an seine vorherige Position.

3a2. Warnung lässt die Benutzten wissen, wieso es nicht funktioniert hat.

3b1. Modul besitzt vorausgesetzte Kompetenzen, die nicht erfüllt werden. Modul bleibt an der gewünschten Position liegen.

3b2. Warnung wird angezeigt, um auf die Fehlenden Kompetenzen aufmerksam zu machen.

Nachbedingung / Ergebnis:

Nicht-funktionale Anforderungen:

- Flüssiges Drag & Drop

Parametrisierbarkeit / Flexibilität:

Nutzungshäufigkeit / Mengengerüst:

- Immer möglich

Titel: Änderung von Einträgen

Akteure: Student

Fachlicher Auslöser: Zu viel vorgenommen -> Semesterplan ausdünnen

Vorbedingung: Modul(e) sind schon im eigenen Semesterplan vorhanden

Standardablauf:

1. Student wählt Modul aus, das er Verschieben will.
2. Student zieht das Modul auf ein anderes Semester.
3. Student bestätigt Änderung.

Alternative Abläufe/ Fehlersituationen / Sonderfälle:

2a. System lehnt das Verschieben ab (für das Semester gibt es keine Veranstaltungen von diesem Modul)

2a1. Student verschiebt das Modul auf ein Semester wo es angeboten wird.

2a2. weiter bei 3

2b. System wirft eine Meldung, dass es nun Abhängigkeiten oder sogar die Fortschrittsregelung nicht mehr eingehalten werden.

2b1. Student verschiebt weitere Module, so dass auf jeden Fall die Fortschrittsregelung eingehalten wird.

2b2. weiter bei 3

Nachbedingung / Ergebnis:

Veränderung wird grafisch so wie im Speicher gespeichert.

Nicht-funktionale Anforderungen:

drag and Drop: sollte nicht zu ruckelig sein.

Parametrisierbarkeit / Flexibilität:

Nutzungshäufigkeit / Mengengerüst: Jederzeit möglich

4.2.7 Studiengangwechsel Pablo Schneider

Titel: Studiengangwechsel

Akteure: Student

Vorbedingung: Hatte schon einen Studiengang im Planer geladen.

Standardablauf:

1. neue Datei auswählen
2. Datei hochladen
3. Bestätigen.

Alternative Abläufe/ Fehlersituationen / Sonderfälle:

Nachbedingung / Ergebnis:

- Daten vom alten Studiengang werden gelöscht, und der neue Module Werden in den Planer geladen.
- Gui wird resetet.

Nicht-funktionale Anforderungen:

Sollte nicht länger als 15 sec dauern

Parametrisierbarkeit / Flexibilität:

Nutzungshäufigkeit / Mengengerüst: 1x zum Studienbeginn beim neuen Studium.

Titel: Abspeichern von Semesterplänen

Akteure: Studi

Fachlicher Auslöser: Student ist fertig mit Studienplanerstellung

Vorbedingungen: Mind. 1 Modul wurde ausgewählt

Standardablauf:

1. Studi: betätigt Herunterladen
2. System: speichert Datei im Zielort ab
3. Studi: ruft Datei aus Zielort aus auf

Alternative Abläufe / Fehlersituationen / Sonderfälle:

- 2a. System findet Zielort nicht
 - 2a1. System wirft Error
 - 2a2. Studi verändert Zielort
 - 2a3. Weiter bei 2

Nachbedingung / Ergebnis:

- System geht in Standardzustand über

Nicht-funktionale Anforderungen:

- Speicherzeit max. 10 Sekunden

Parametrisierbarkeit / Flexibilität:

- Konfigurierbarer Zielort der Abspeicherung

Nutzungshäufigkeit / Mengengerüst:

- Beliebig oft

5 Nichtfunktionale / technische Anforderungen

5.1 Nichtfunktionale Anforderungen

- Speicherverbrauch: max. 2GB
- Performance: 60 FPS
- Ein effizienter Suchalgorithmus für Login.
- Das System soll eine sinnvolle Speichermethode für Registrierung um schnelles Suchen beim Login zu ermöglichen.
- Drag and Drop sollte nicht merklich ruckelig sein.
- Das Laden von Files sollte maximal 5 Sekunden dauern.
- Ein Student soll das Programm ohne weitere Anleitung problemlos bedienen können.
- Alle Klassen haben JavaDoc-Kommentare.

5.2 Technische Anforderungen

- Läuft auf: Windows 9 und 10, Linux Mint, Ubuntu, MacOS
- Java Version: 17 oder neuer
- Prozessor: mind. Intel Pentium oder Ryzen 3
- Arbeitsspeicher: mind. 4GB RAM
- Speicherkapazität: mind. 2GB
- Auflösung: Full HD (1920 x 1080)
- Hex Standard

6 Dämonenmodell / Daten

6.1 Dämonenmodell

Der Planer hat genau einen Studiengang, da es ausgeschlossen ist, dass Studenten mehreren Studiengängen gleichzeitig nachgehen (siehe Abbildung 2). Diese Studiengänge haben natürlich Module, welche Wissen vermitteln und selbst noch einmal unterteilt sind in Vorlesung, Prüfung, Praktikum, Tutorium etc. Wir nehmen allerdings nur bewertete Veranstaltungen in den Planer, da es wichtig ist nachzuverfolgen, ob etwas bestanden ist oder nicht.

Diese benennen wir als Modulbestandteile. Nur, wenn alle Modulbestandteile erfüllt sind, gilt ein Modul dann auch als bestanden. Jedes Modulbestandteil ist entweder eine Prüfung oder ein Praktikum.

Jedes Modul wird von einem oder mehreren Dozenten gelehrt.

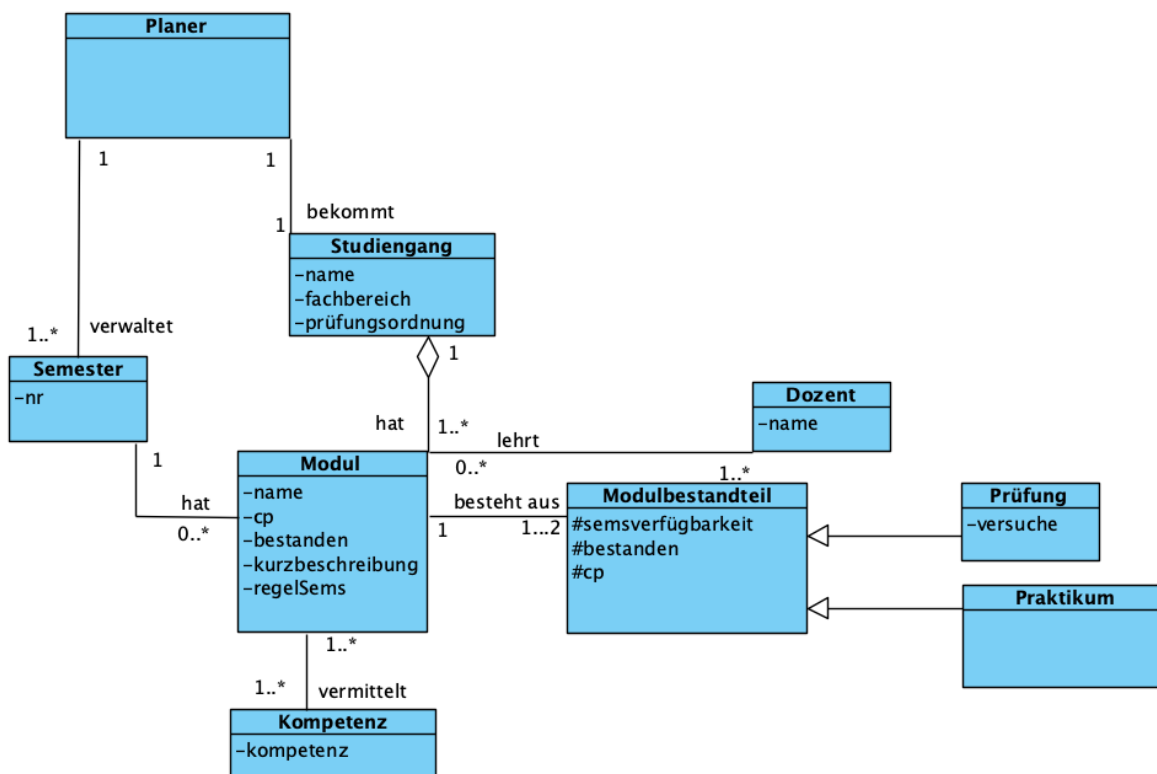


Abbildung 2: Dämonenmodell

Es soll zudem geplant werden können, welche Module in welchen Semestern gemacht werden. Dazu bekommt der Planer eine Liste von den ganzen Modulen des Studiengangs, um diese dann einzelnen Semestern zuzuordnen.

Daneben sollen sich die Semester merken, welche Kompetenzen erarbeitet wurden, sodass man in höheren Semestern davor warnen kann, wenn diese Kompetenzen für Module noch nicht erreicht worden sind. Diese Kompetenzen können sich die Semester über die Module ziehen.

Die Planerklasse wäre dann noch für weitere Logik da, z.B. für die Einhaltung der Fortschrittsregelung.

6.2 Datentypenverzeichnis

Klasse	Attribut	Datentyp
Semester	nr	String
Modul	name	String
	cp	int
	bestanden	Boolean
	kurzbeschreibung	String
	regelSems	int
Dozent	name	String
Modulbestandteil	Semsverfügbarkeit	{SO, WI}
	bestanden	Boolean
	cp	int
Studiengang	name	String
	fachbereich	String
	prüfungsordnung	int

7 Benutzungsschnittstelle

Unsere Benutzungsschnittstelle ist so aufgebaut, dass man zentral eine Übersicht für die einzelnen Semester angezeigt bekommt (siehe Abbildung 3).

Links sieht man, welche Module zu welchem Semester gehören, und kann hier auch per “+” weitere Semester hinzufügen, falls man über die Regelstudienzeit hinweg studieren möchte. Rechts ist die Auswahl der einzelnen Module, auch sortiert nach Semester. Von dort aus kann man sich Module nach links in die zentrale Übersicht in das Semester, wo man das Modul belegen möchte, per Drag & Drop ziehen.

Am oberen Rand dieser Leiste wählt man zwischen Prüfung und Praktikum aus. Jedes Fach hat entweder nur eine Prüfungsleistung oder wird in Prüfungs- und Praktikumsleistung unterteilt. Die Unterteilung ist wichtig, da man Prüfungen und Praktika in unterschiedlichen Semestern machen kann.

Oben links neben der Modulauswahlleiste steht noch klein die insgesamt CP-Anzahl, welche sich aus allen als “bestanden” markierten Modulen zusammensetzt.

Unten rechts sind zwei Buttons platziert. Mit dem linken kann man einen neuen Plan erstellen. Damit wird der Vorherige zurückgesetzt. Beim Speichern-Button kommt dann erst noch eine kurze Auswahl, ob man den Planer so abspeichern möchte, dass man danach wieder darauf zugreifen kann, oder ob man ihn als PDF ausdrückbar abspeichert.

The screenshot displays a user interface for managing a study plan. On the left, a vertical list of semesters (1 to 6) is shown, with a '+' button at the top for adding more. To the right of this list, a grid of gray rectangular blocks represents modules assigned to each semester. For example, semester 1 has five modules, semester 2 has three, and semester 3 has three. On the far right, a vertical sidebar contains a list of modules, each with a corresponding semester number (1 to 7) to its right. At the top of this sidebar, there are two tabs: 'Prüfung' (Exam) and 'Praktikum' (Practical). At the bottom of the sidebar, there are two buttons: 'Neuer Plan +' (New Plan) and 'Speichern' (Save) with a dropdown arrow. The top right corner of the main area shows 'CP: 120'.

Abbildung 3: Übersicht der Anwendung

Per einfachem Klick auf ein Modul werden die Abhängigkeiten zu anderen Module angezeigt. Dabei werden alle Module, die keine Abhängigkeit zu dem ausgewählten Modul haben, ausgegraut. Dadurch stechen die Modulabhängigkeiten hervor.

Per Rechtsklick auf ein Modul werden zusätzliche Informationen in einem Pop-Up Fenster angezeigt (siehe Abbildung 4).

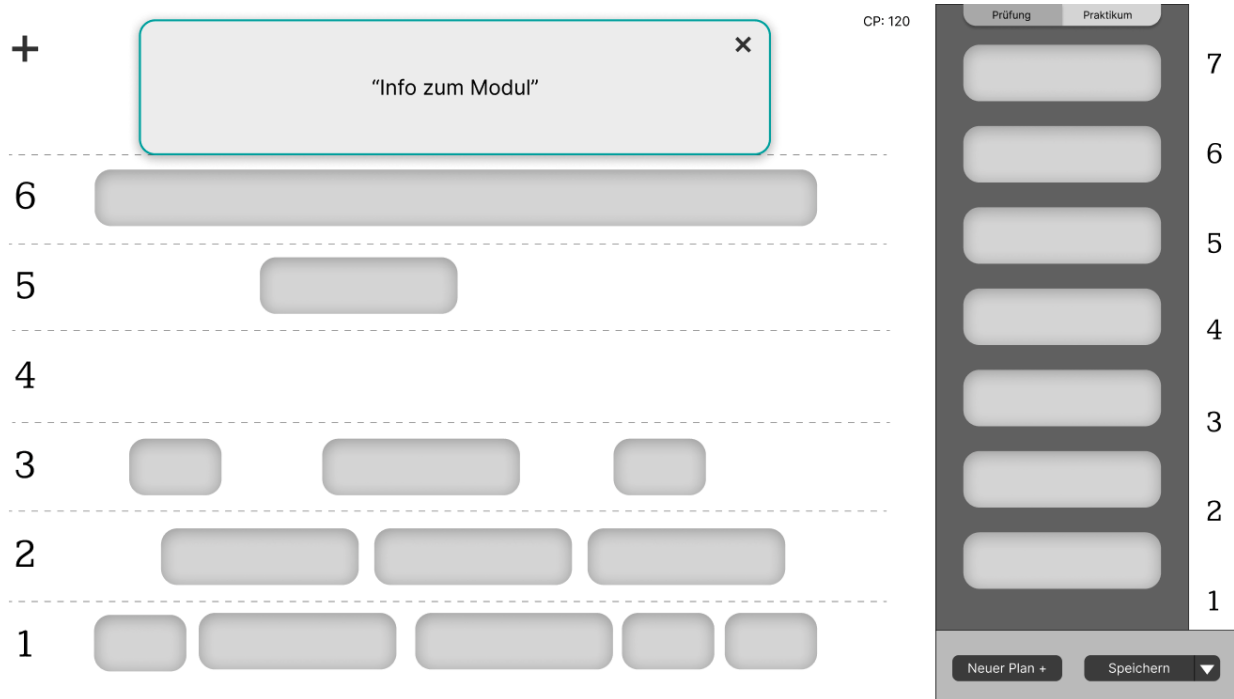




Abbildung 4: Infofenster Anzeige

Das Infofenster gibt einen alle nötigen Informationen für das jeweilige Modul bzw. Praktikum (siehe Abbildung 5) oder Prüfung (siehe Abbildung 6) des Modules an. Man bekommt bei einem Praktikum die empfohlenen Voraussetzungen, den Dozent und weitere Wichtige Informationen für das Modul.

Softwaretechnik (Praktikum) 4142, 3CP
Anwesenheitspflicht!

Bestanden?  

Dozent: Prof. Dr. Wolfgang Weitz

Informationen:
Die Fähigkeit zur Auswahl, Bewertung und praktischen Anwendung von Konzepten und Methoden zur systematischen Entwicklung (großer) Softwaresystemen stellt eine zentrale Qualifikation für Medieninformatikerinnen und Medieninformatiker dar. Dabei sind die Phasen Analyse / Design von grundlegender Bedeutung für das Gelingen eines Softwareprojekts:

- Software im Team entwickeln unter Einsatz entsprechender Vorgehensmodelle
- Modellieren von Anwendungsproblemen und Lösungskonzepten unter Einsatz der Unified Modeling Language (UML)
- Systematische Erhebung, Modellierung und Dokumentation von Anforderungen
- Ableitung einer und Beschreibung einer adäquaten Software-Architektur unter Berücksichtigung grundlegender Qualitäts-Eigenschaften
- Überblick über und zielgerichteter Einsatz von aktuellen Architektur- und Entwurfsmustern
- Einsatz von Softwarewerkzeugen zur Unterstützung des kompletten Softwarelebenszyklus

empfohlene Voraussetzungen:

- Programmieren 3
- Datenbanksysteme
- Entwicklung interaktiver Benutzungsoberflächen
- Programmieren 2

Abbildung 5: Praktikums-Infofenster

Bei der Prüfung wäre es wichtig, den Prüfer, die Anzahl der Versuche und natürlich auch die Prüfungsform zu wissen. Es sollte hier auch eine Möglichkeit gegeben werden, das Modul als geschafft zu markieren (siehe Abbildung 6).

Softwaretechnik (Prüfung) 4141, 3CP

Bestanden?  

Dozent: Prof. Dr. Wolfgang Weitz

Versuche 1 / 3

Prüfungsform: Schriftliche Prüfung

Weitere Infos:

- Hilfsmittel:
- Dauer:
- ...

Abbildung 6: Prüfungs-Infofenster

Wenn ein Modul bestanden ist, wird es grün verfärbt. Orange wird es, wenn man ein Modul machen will, für welches einem Kompetenzen fehlen. Und rot würde einem markiert werden, wenn man die Fortschrittsregelung missachtet (siehe Abbildung 7). Beim Letzten sollte der Drag and Drop verhindert werden. Die Kachel wird auf der rechten Seite gefärbt und daneben sollte eine Infobox mit rotem und bösem Text erscheinen (siehe Abbildung 8). Nach dem Wegklicken der Warnung wird die Farbe der Kachel zurückgesetzt.

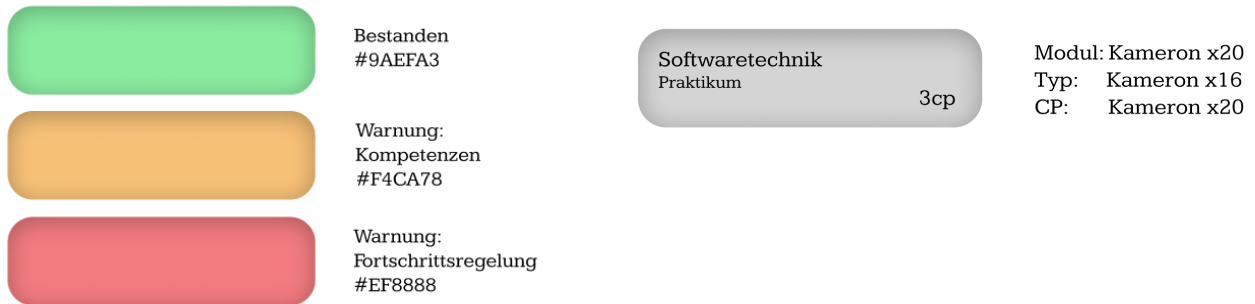


Abbildung 7: Farbschema und Modulbeispiel

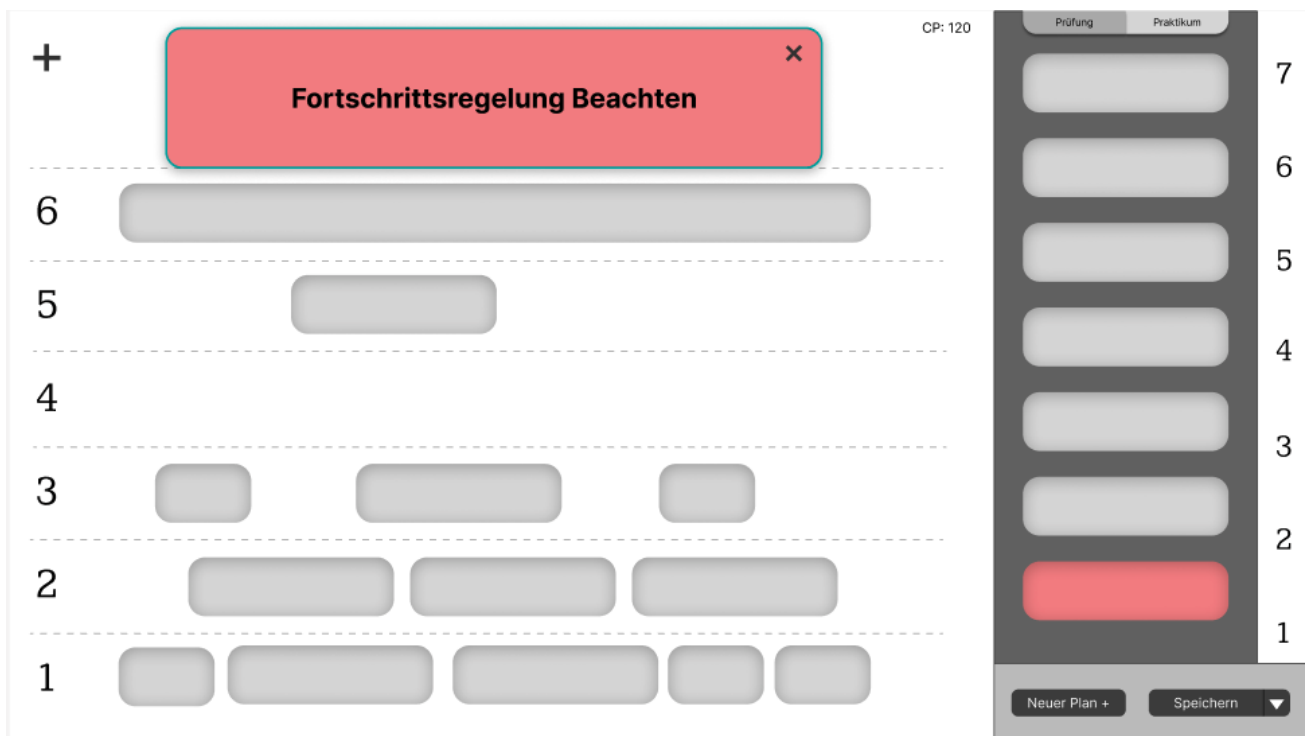


Abbildung 8: Fortschrittsregelung missachtet

Wir haben uns für die Zukunft noch vorgenommen Werbung, Sport- und Sprachangebote und ein zuckersüßes Maskottchen hinzuzufügen. Dafür haben wir uns auch mal überlegt, wie das aussehen könnte (siehe Abbildung 9). Die Werbung, so wie das Maskottchen, würden immer in zufälligen Intervallen unkontrolliert erscheinen.

Dabei versucht das Maskottchen seinen ungeheim großen Appetit zu stillen, in dem es die armen unschuldigen Module auf fürchterliche Art und Weise verschlingt. Der Benutzer muss das Maskottchen davon auf heldenhafter Weise davon abhalten, und es wieder dahin drängen, wo es hergekommen ist.



CP: 120

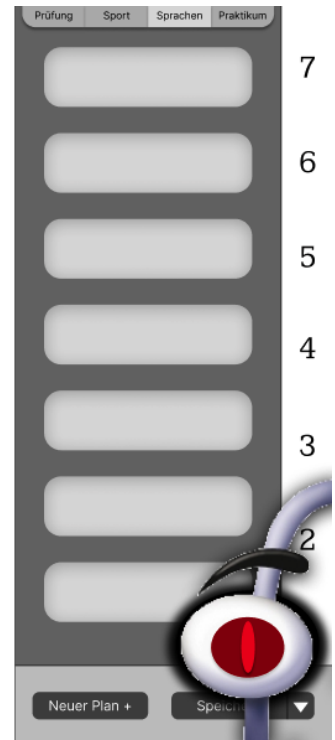


Abbildung 9: In Planung - Werbung, Maskottchen, Sport und Sprachen

8 Glossar

Begriff	Bedeutung
CP	Creditpoints
Dr.	Titel: Doktor
Kompetenzen	Durch ein Modul zu vermittelnde Fähigkeiten
Modul	Fach
PO	Prüfungsordnung
Praktikum	Eine durch CP gewichtete Pflichtveranstaltung
Prof.	Titel: Professor
Prüfungstypen	mündliche, schriftliche und praktische Prüfungen möglich
Sems	Semester
Vorlesung	Eine von einem Dozenten gehaltene Lesung mit optionalem Interaktiven Charakter