

# Studienplaner: SPAsS

Versionsnummer: 2

Datum: 27.05.2022

# Inhaltsverzeichnis

<b>Anwendungsszenarien</b>	<b>4</b>
Szenario 1	4
Szenario 2	6
Szenario 3	8
Szenario 4	11
<b>Erfüllungskriterien</b>	<b>13</b>
<b>Funktionale Anforderungen</b>	<b>14</b>
<b>Nichtfunktionale Anforderungen</b>	<b>15</b>
<b>Benutzeroberfläche</b>	<b>16</b>
<b>Domänen-Modell SPAsS</b>	<b>17</b>
<b>Technische Voraussetzungen</b>	<b>18</b>

## Zielstellung

Ziel ist es einen interaktiven Stundenplan zu programmieren, der den Studierenden das Planen vom Studium erleichtert. Es werden alle Informationen über bestandene und nicht bestandene Module eingetragen. Anstatt selbst den Überblick behalten zu müssen, welche Module belegt werden können und verfügbar sind bzw. welche Module schon bestanden wurden oder nicht, zeigt der Planer dies in einer Übersicht an.

Dabei soll ermöglicht werden, dass der Planer für allerlei Studiengänge einsetzbar ist, sowie die verschiedenen Regelungen angepasst werden können. So kann es bei manchen Studiengängen eine Fortschrittsregelung von drei Semestern geben, bei anderen wiederum nicht. Um das Arbeitspensum selbst bestimmen zu können, soll zudem eine Begrenzung der Credit Points möglich sein, durch die man eine passende Auswahl an Modulen erhält.

## Anwendungsszenarien

### Szenario 1: CP-Sperre

Kevin Emunds

Der Student Sievert des Studiengangs 'Medieninformatik' an der Hochschule RheinMain muss zwangsläufig, um sein Lebensunterhalt zu finanzieren, neben dem Studium arbeiten. Aufgrund der lächerlich hohen Mietpreise zur Zeit reicht da jedoch auch nicht mehr ein kleiner Nebenjob, sodass er sich dazu beschloss das Studium etwas langsamer anzugehen und seine CP im Semester geringfügig zu reduzieren.

Beim Zusammenstellen seines Stundenplans für das Semester in PLATZHALTER gibt er also zunächst an, dass er nur 17 CP pro Semester erreichen möchte. Er wählt seine Module an und bekommt beim vierten Modul angezeigt, dass die Wahl dieses Moduls seine eigenen Vorgaben der maximalen CP-Grenze überschreiten würde. Aufgrund dieser Warnung entscheidet er sich dazu, es bei den drei bereits ausgewählten Modulen zu belassen.

## Szenario 2: Modulvoraussetzungen und Details

Victoria Thee

Rufus studiert an der Goethe-Universität in Frankfurt Grundschullehramt im 3. Semester als Vollzeitstudium und möchte planen, welche Module er nächstes Semester belegt. Im zweiten Semester hat er nicht alle Module bestanden, weshalb er eine Warnung bekommt, als er ein Modul auswählt, dessen Voraussetzungen er nicht erfüllt. Daher beschließt Rufus zuerst das nicht bestandene Modul nachzuholen und das gewünschte Modul erst im nächsten Semester zu belegen, da jedes Semester alle Module angeboten werden. Danach erkundigt sich Rufus darüber, welche anderen Module Abhängigkeiten zu weiterführenden Modulen besitzen, damit er einen Fokus auf diese legen kann, um in folgenden Semestern keine Module auslassen zu müssen.

## Szenario 3: Wahlmöglichkeiten und bestandene Module

Alexandra Müller

Der/Die Student:in möchte für die Planung zukünftiger Semester gerne wissen, welche Module noch zu belegen, beziehungsweise belegbar, sind. Er/Sie öffnet die Studienplaner Applikation, welche eine Übersicht des gesamten Curriculums bietet. Die Applikation bietet nun die Möglichkeit, bereits bestandene Module zu kennzeichnen. Dadurch werden diese grün markiert und der/die Student:in kann schnell sehen, welche Module aus vergangenen Semestern vielleicht noch nachgeholt werden müssen. Durch eine Einschränkung der belegbaren Module durch die Applikation auf Grund der Fortschrittsregelung, sowie der Kennzeichnung bestandener Module, bekommt der/die Student:in nun seine/ihre gewünschte, übersichtliche Darstellung von Wahlmöglichkeiten für das kommende Semester.

## Szenario 4: Fehlversuche und Fortschrittsregelung

Astrid Klemmer

Die Studentin Anna hat das dritte Semester abgeschlossen und verschafft sich auf dem Studienplaner eine Übersicht darüber, welche Module sie im nächsten Semester belegen kann. Sie sieht, dass alle Module vom vierten Semester und darüber ausgegraut sind. Obwohl es ausgegraut ist, klickt sie auf ein Modul im vierten Semester. Sie bekommt die Meldung, dass sie aufgrund der Fortschrittsregelung ab dem vierten Semester nichts belegen kann. Kurz ist sie verwirrt und schaut in den Semestern davor nach. Dabei erkennt sie, dass im ersten Semester Analysis rot hinterlegt ist. Bei einem Klick darauf bekommt sie die Information, dass sie dieses Modul einmal nicht bestanden hatte. Sie beschließt, Analysis dieses Semester nachholen zu wollen und muss erneut feststellen, dass es nicht möglich ist, da diese Vorlesung nur im Wintersemester verfügbar ist. Stattdessen wird ihr vorgeschlagen Module aus dem zweiten Semester nachzuholen. Hier ist IT-Recht und Datenschutz nicht grün, sondern noch weiß, da sie dieses Modul geschoben hatte. Auszeichnungssprachen ist rot markiert. Sie klickt darauf, um mehr Informationen zu erhalten und erkennt, dass sie das Praktikum nicht bestanden hatte. Sie wählt die beiden Module aus und erfährt, wie viele Credit Points diese insgesamt geben.

## Erfüllungskriterien:

1. Module
  - 1.1. Das Programm muss fähig sein dem Nutzer anzuzeigen, welche Module er bestanden hat und welche belegbar sind.
  - 1.2. Das Programm muss die Kerndaten der Module anzeigen.
  - 1.3. Das Programm sollte beim Anzeigen eines Moduls auf die Abhängigkeiten zu weiterführenden Modulen verweisen.
  - 1.4. Das Programm muss Abhängigkeiten zu vorherigen Modulen anzeigen.
  - 1.5. Das Programm sollte bei Abhängigkeiten zu einem Modul im nächsten Semester bei nicht ausgewählten Modul aus dem aktuellen Semester warnen.
  - 1.6. Das Programm muss die Fortschritts Regel bei der Auswahl der Module berücksichtigen
  - 1.7. Das Programm sollte es dem Nutzer ermöglichen, seine Noten für jeweilige Module eintragen zu können.
    - 1.7.1. Das Programm sollte sich alternativ die Noten durch die Einlogg-Daten des Nutzers laden können.
    - 1.7.2. Das Programm sollte aus den eingetragenen Noten den Notenspiegel berechnen können.
  - 1.8. Das Eintragen von neuen Modulen sollte Möglich sein.
2. CP
  - 2.1. Das Programm muss dem Nutzer ermöglichen seine eigene CP-Grenze festzulegen.
  - 2.2. Das Programm muss den Nutzer warnen, wenn die CP-Grenze überschritten wird.
3. GUI
  - 3.1. Das Programm muss eine benutzerfreundliche, interaktive 2D-Oberfläche besitzen.
  - 3.2. Das Programm muss intuitiv sein
  - 3.3. Drag&Drop sollte möglich sein.
4. Systemanforderungen
  - 4.1. Das Programm muss ohne Internet funktionieren

## Funktionale Anforderungen

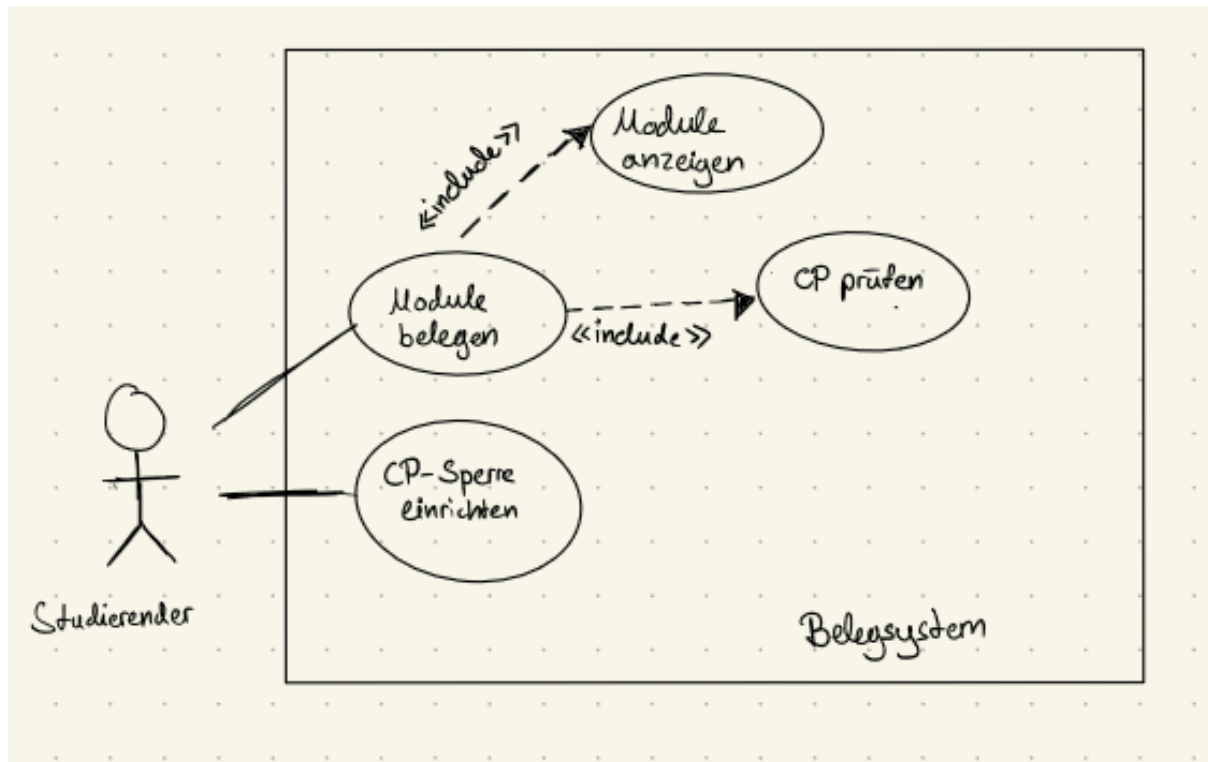
Das System muss anzeigen, welche Module belegbar und bestanden sind. Zudem müssen Abhängigkeiten von Modulen bei Auswahl eines Moduls angezeigt werden. Wenn die Abhängigkeiten nicht erfüllt sind, tritt eine Warnung auf und wenn Voraussetzungen eines Moduls nicht erfüllt sind, wie eine Fortschrittsregelung, kann das gewünschte Modul nicht gewählt werden und auch hier erscheint eine Warnung.

Außerdem muss eine gewünschte Anzahl CP für ein Semester angegeben werden können, um die Planung damit einzuschränken. Dadurch wird bei Überschreiten der gewünschten CP Anzahl eine Warnung angezeigt. Wenn keine CP Anzahl angegeben wird, wird ein Default Wert von 30 CP genutzt.

Auch soll das System die Möglichkeit bieten, den aktuellen Notenspiegel zu berechnen, der sich aus der Gewichtung der Module und den Noten in diesen zusammensetzt.

Um die Module anzeigen zu können wird eine externe Datei benötigt, welche den Modulnamen, die CP und alle Abhängigkeiten besitzt. Für die Berechnung des Notenspiegels müssen zusätzlich noch die einzelnen Noten bei den Modulen ergänzt werden, ansonsten wird kein Notenspiegel angezeigt. Zu guter Letzt muss noch eine Fortschrittsregelung und eine reguläre Studienzeit in der Datei angegeben werden, da unterschiedliche Studiengänge verschiedene Fortschrittsregelungen und Semesterzahlen besitzen.





**Titel:** CP-Sperre einrichten

**Akteure:** Student, Studentin

**Fachliche Auslöser:** Maximale Auslastung der Studierenden begrenzen

**Vorbedingungen:** /

**Standardablauf:**

- 1. Studierender: Klickt auf Einstellungen.
- 2. System: zeigt eine detaillierte Ansicht der Einstellungen an.
- 3. Studierender: wählt das Feld "CP-Grenze" aus
- 4. Studierender: Löscht seinen alten Wert und gibt neuen ein.
- 5. System: Übernimmt neuen Wert zur Belegung.

**Alternative Abläufe:**

- 5b Momentane Auswahl an Modulen überschreitet die neue Grenze
  - 5b1 System: Warnung erscheint, neue Grenze wird noch nicht gesetzt
    - 5b2a Studierender: Bestätigt Warnung positiv
    - 5b2a1 System: System übernimmt neue Grenze und resettet Modulwahl
    - 5b2b Studierender: Bestätigt Warnung negativ
    - 5b2b1 System: System übernimmt neue Grenze nicht.

**Nachbedingung:** CP-Sperre ist eingerichtet

**Parametrisierbarkeit:** Keine negativen Werte

**Nutzungshäufigkeit:** 2x jährlich vor den Modulbelegungen

**Titel:** Module belegen mit CP-Grenze

**Akteure:** Student, Studentin

**Fachliche Auslöser:** Modulwahl am Semesterstart

**Vorbedingungen:** Module werden sachgemäß angezeigt.  
CP-Sperre ist eingerichtet

**Standardablauf:**

- 1. Studierender: wählt ein Modul zum Belegen aus
- 2. System: zeigt eine detaillierte Ansicht des Moduls an
- 3. System: prüfen der Voraussetzung
- 4. Studierender: belegen des Moduls
- 5. System: gewähltes Modul als belegt anzeigen

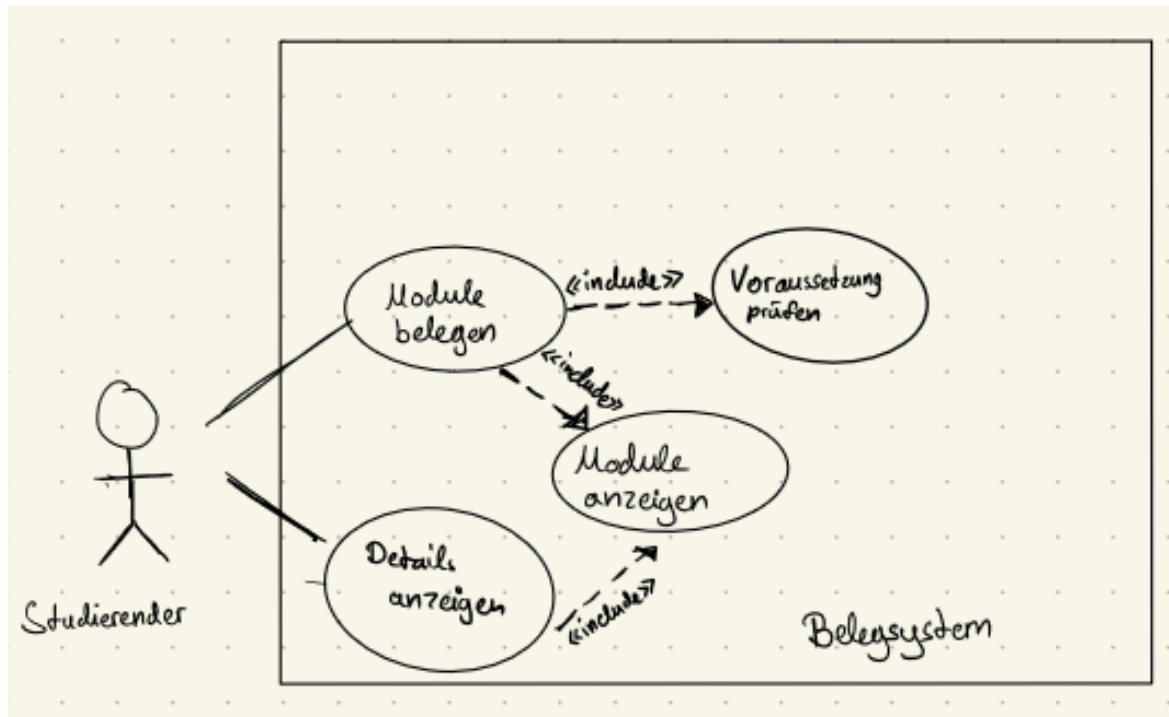
**Alternative Abläufe:**

- 3b Voraussetzungen für die CP-Belegung sind nicht erfüllt
  - 3b1 System: Warnung erscheint
  - 3b2 Studierender: entscheidet, ob das Modul trotzdem belegt werden soll

**Nachbedingung:** Modul ist belegt

**Parametrisierbarkeit:** /

**Nutzungshäufigkeit:** 2x jährlich bei der Modulbelegungen



**Titel:** Voraussetzungen von Modulen prüfen

**Akteure:** Studierender

**Fachliche Auslöser:** Module für das nächste Semester planen

**Vorbedingungen:** Veranstaltungen, deren Voraussetzungen, bereits bestandene Module und die Fortschrittsregelung wurden mitgegeben

**Standardablauf:**

- 1. Studierender: wählt ein Modul zum Belegen aus
- 2. System: zeigt eine detaillierte Ansicht des Moduls an
- 3. System: prüfen der Voraussetzung
- 4. Studierender: belegen des Moduls
- 5. System: gewähltes Modul als belegt anzeigen

**Alternative Abläufe:**

- 3b Voraussetzungen für das Modul sind nicht erfüllt
  - 3b1 System: Warnung erscheint
  - 3b2 Studierender: entscheidet, ob das Modul trotzdem belegt werden soll

**Nachbedingung:** Modul belegt

**Parametrisierbarkeit:** /

**Nutzungshäufigkeit:** 2x jährlich vor den Modulbelegungen

**Titel:** Details einzelner Module anzeigen

**Akteure:** Studierender

**Fachliche Auslöser:** Module für das nächste Semester planen

**Vorbedingungen:** Veranstaltungen, deren Voraussetzungen, bereits bestandene Module und die Fortschrittsregelung wurden mitgegeben

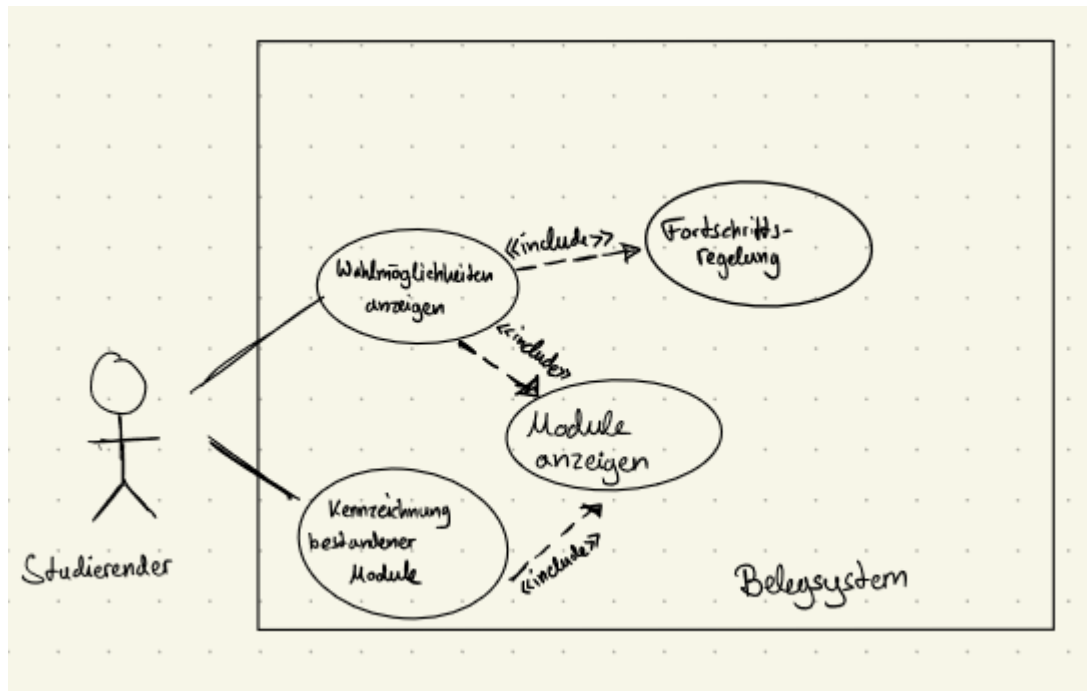
**Standardablauf:**

- 1. Studierender: wählt ein Modul aus
- 2. System: öffnet detaillierte Ansicht des Moduls

**Nachbedingung:** /

**Parametrisierbarkeit:** /

**Nutzungshäufigkeit:** 2x jährlich vor den Modulbelegungen



**Titel:** Wahlmöglichkeiten anzeigen

**Akteure:** Nutzer:in

**Fachlicher Auslöser:** Modulbelegungen zum Semesterbeginn stehen bevor

**Vorbedingungen:** Curriculum wurde in die Datenbank hinzugefügt

**Standardablauf:**

- 1. System: Anzeigen der Module
- 2. Nutzer:in: Markieren von bereits bestandenen Modulen
- 3. Aktualisierung des Modulplans und der in diesem Semester wählbaren Module durch die Anwendung der Fortschrittsregel und Einbezug von Modulvoraussetzungen

**Alternative Abläufe / Fehlersituationen / Sonderfälle:**

- 2a Nutzer:in versucht Module als bestanden zu markieren, die von nicht als bestanden markierten Modulen abhängen
  - 2a1 Das System macht auf den Fehler aufmerksam und bittet darum, zuerst die Module als bestanden zu markieren, von denen das Modul abhängt
  - 2a2 Weiter bei 4

**Nachbedingung / Ergebnis:**

Anzeigen von Wahlmöglichkeiten

**Parametrisierbarkeit / Flexibilität:****Nutzungshäufigkeit / Mengengerüst:**

2x jährlich (zum Semesterbeginn)

**Titel:** Kennzeichnung bestandener Module

**Akteure:** Nutzer:in

**Fachlicher Auslöser:** Modulbelegungen zum Semesterbeginn oder Ende der Prüfungsphase

**Vorbedingungen:** Curriculum wurde in die Datenbank hinzugefügt  
Standardablauf:

- 1. System: Anzeigen der Module
- 2. Nutzer:in: Markieren von bereits bestandenen Modulen und optionale Eintragung der Modulnote
- 3. System: Einfärbung bestandener Module in grün
- 4. System: Einspeichern der Note in der Datenbank
- 5. System: Aufnahme des Moduls in einsehbare Liste von bestandenen Modulen mit Notenspiegel

**Alternative Abläufe / Fehlersituationen / Sonderfälle:**

- 2a Nutzer:in versucht Module als bestanden zu markieren, die von nicht als bestanden markierten Modulen abhängen
  - 2a1 Das System macht auf den Fehler aufmerksam und bittet darum, zuerst die Module als bestanden zu markieren, von denen das Modul abhängt
  - 2a2 Weiter bei 2
- 2b Nutzer:in trägt Note ein die schlechter als 4.0 ist
  - 2b1 Das System macht auf den Fehler aufmerksam und bittet, entweder eine gültige Note einzutragen oder die Markierung abzuberechnen
  - 2b2 Weiter bei 2

**Nachbedingung / Ergebnis:**

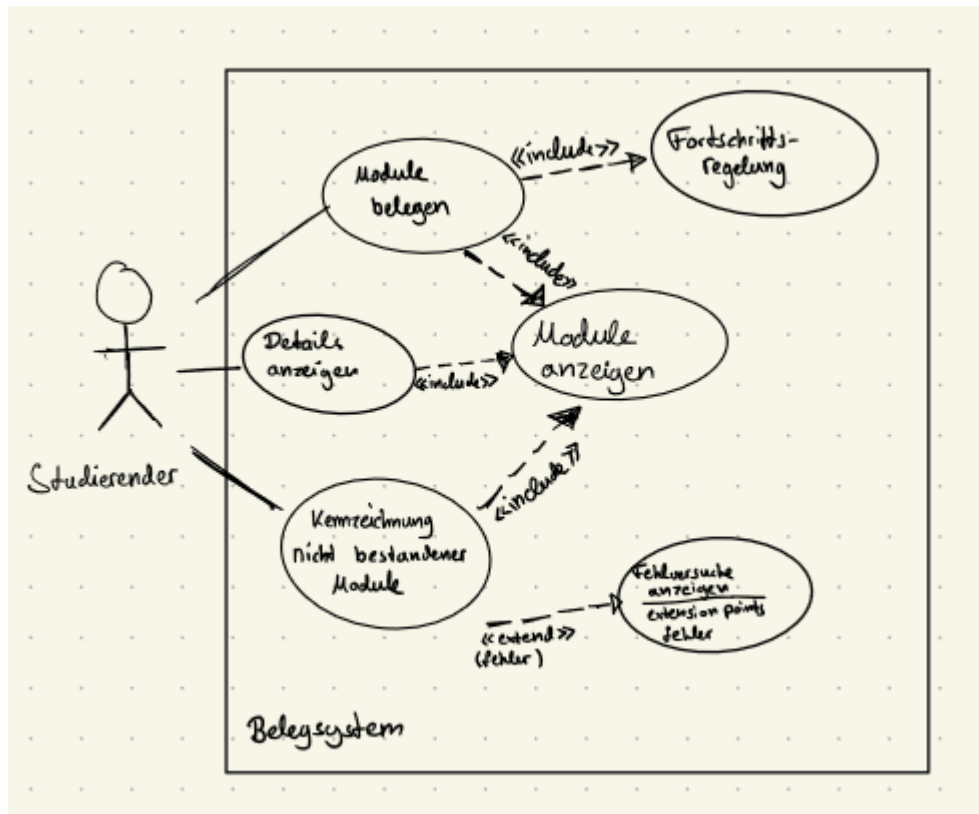
Anzeige von Liste mit bestandenen Modulen

**Parametrisierbarkeit / Flexibilität:**

Noten, keine negativen Zahlen

**Nutzungshäufigkeit / Mengengerüst:**

2 - 4x jährlich (zum Semesterbeginn oder zum Ende einer Prüfungsphase)



**Titel:** Fehlversuche anzeigen

**Akteure:** Studierender

**Fachlicher Auslöser:** Aktuellen Stand bestandener und nicht bestandener Module ermitteln

**Vorbedingungen:** Modul ist im Planer vorhanden, Studierender hat nicht bestandene Module eingetragen

**Standardablauf:**

- 1. System: Plan erzeugen
- 2. Studierender: bestandene und nicht bestandene Module eintragen
- 3. System: Änderungen überprüfen
- 4. System: Änderungen durchführen und Plan mit aktualisierten Einträgen anzeigen
- 5. Studierender: mit aktuellem Stand weiterplanen

**Alternative Abläufe / Fehlersituationen / Sonderfälle:**

- 3a Studierender hat maximale Fehlversuche erreicht
  - 3a1 System gibt Warnung aus über durchgefallenen Modul
  - 3a2 bei 4 fortfahren
- 4a Studierender macht Änderung rückgängig
  - 4a1 wieder bei 2 beginnen

**Nachbedingung/Ergebnis:**

Plan wird angezeigt

**Parametrisierbarkeit / Flexibilität:**

Je Modul Anzahl nicht bestandener Module eintragen/austragen können

**Nutzungshäufigkeit / Mengengerüst:**

Bei Bedarf, vor allem vor/während/nach Klausurenphasen

**Titel:** Belegung bei Fortschrittsregelung

**Akteure:** Studierender

**Fachlicher Auslöser:** Fortschrittsregelung erreicht und alternativen ermitteln

**Vorbedingungen:** Studierender hat bestandene und nicht bestandene Module eingetragen

**Standardablauf:**

- 1. System: Plan erzeugen
- 2. Studierender: verfügbare Module belegen
- 3. System: Belegung überprüfen
- 4. System: Belegung durchführen und anzeigen
- 5. Studierender: mit aktuellem Stand weiterplanen

**Alternative Abläufe / Fehlersituationen / Sonderfälle:**

- 3a Studierender hat Fortschrittsregelung erreicht
  - 3a1 System gibt Warnung aus, dass im neuen Semester nichts belegt werden kann
  - 3a2 bei 2 erneut starten
- 3b Studierender kann die Module nicht in diesem Semester nachholen
  - 3b1 System gibt Warnung aus, dass die Module dieses Semester nicht verfügbar sind
  - 3b2 bei 2 erneut starten
- 4a Studierender nimmt Änderungen vor
  - 4a1 wieder bei 2 beginnen

**Nachbedingung/Ergebnis:**

Belegung wird angezeigt und CP ausgegeben

**Parametrisierbarkeit / Flexibilität:** /

**Nutzungshäufigkeit / Mengengerüst:**

Bei Bedarf, vor allem nach Klausurenphasen/zu Semesterbeginn

## Nichtfunktionale Anforderungen

Studierende sollen das System ohne weitere Anleitung nutzen können.

Die Übersicht muss sich innerhalb von 20ms auf Änderungen wie eine Credit Point Begrenzung oder hinzugefügte bestandene Module anpassen können.

Der Planer sollte verschiedene Auswahlmöglichkeiten laden können ohne, dass die CPU eines Intel i3 Prozessors nicht mehr als 30% ausgelastet ist.

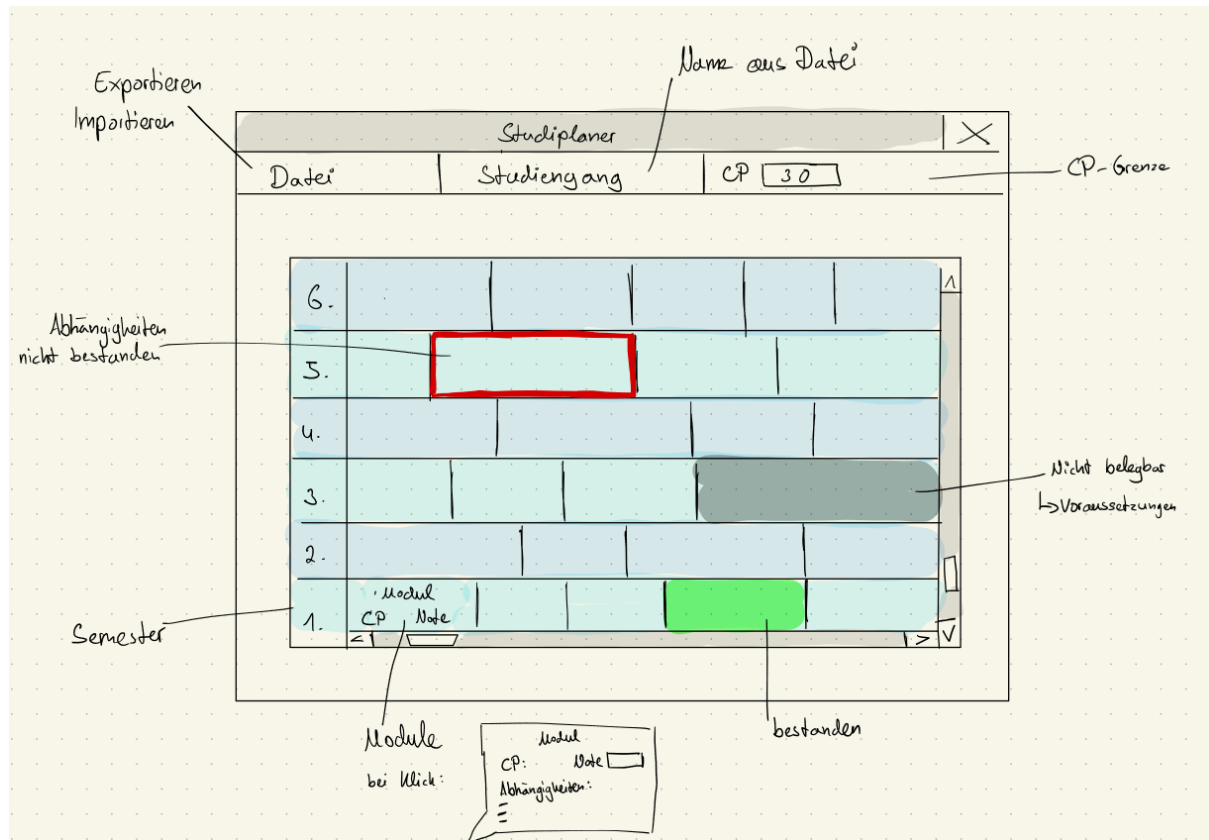
Die Ansicht muss mit verschiedensten Auflösungen dargestellt werden können und skalierbar sein ohne, dass es verzerrt wird.

Das Programm darf nur lokal Informationen abspeichern.

Das Programm muss auch ohne Internet funktionieren können.



## Benutzungsoberfläche:



Die CP Grenze kann mit einem Klick auf das Zahlenfeld angepasst werden, der Ablauf ist dann wie im Use-Case CP-Grenze einrichten beschrieben. Um dann die Module passend zur CP Grenze zu erhalten, muss eine Bestätigung erfolgen, hierzu dient der Use-Case Module belegen mit CP-Grenze als Unterstützung.

Es werden stetig die Voraussetzungen geprüft, um ein Modul belegen zu können, dies geschieht bei jeder Veränderung von CP bzw. hinzugefügten bestandenen oder nicht bestandenen Modulen. Im Use-Case Voraussetzungen von Modulen prüfen, wird dies genauer erläutert.

- Details einzelner Module anzeigen
- Wahlmöglichkeiten anzeigen
- Wahlmöglichkeiten anzeigen
- Kennzeichnung bestandener Module
- Fehlversuche anzeigen
- Belegung bei Fortschrittsregelung

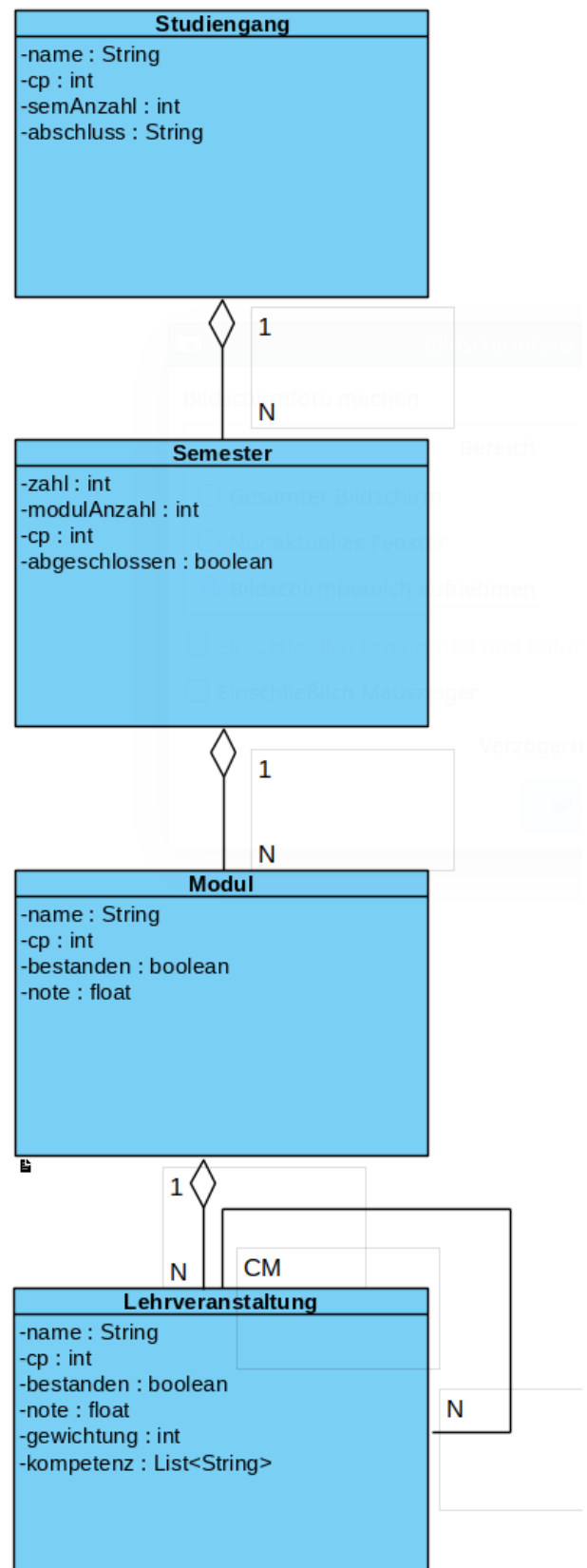
## Domänen-Modell SPAsS:

Beim Erstellen der Gegenstandswelt für den Studiengangplaner haben wir mit dem Studiengang begonnen, der sich aus mind. einem Semester zusammensetzt. Ein Studiengang besteht dabei aus einem Namen, CP, der Semesteranzahl und dem damit verbundenen Abschluss.

Ein Semester existiert nicht ohne den zugehörigen Studiengang. Jedes Semester umfasst mind. ein Modul. Das Semester bekommt eine Semesterzahl, eine Modulanzahl, CP und einen Wahrheitswert, ob es abgeschlossen wurde.

Die Module sind wiederum vom Semester abhängig. Zu einem Modul gehört mind. eine Lehrveranstaltung. Das Modul hat einen Namen, CP, eine Note und einen Wahrheitswert, ob es bestanden wurde.

Die Lehrveranstaltung kann eine Vorlesung, ein Praktikum, eine Übung oder sonstiges sein. Zudem kann mind. eine Lehrveranstaltung von keiner oder mehr Lehrveranstaltungen abhängen. Die Lehrveranstaltung erhält einen Namen, CP, eine Note, eine Gewichtung, Kompetenzen und einen Wahrheitswert, ob es bestanden wurde. Dabei werden die Lehrveranstaltungen durch die dazugehörigen Kompetenzen verknüpft.



## Technische Voraussetzungen

- Gradle als Build Tool
- Die GUI soll in Java FX angeboten, jedoch austauschbar sein
- Lauffähigkeit unter Linux zu gewährleisten
- Lauffähigkeit unter Java 17 zu gewährleisten
- Die Anwendung soll ohne große Hardware Leistung ausführbar sein

### Anmerkungen:

Datum und Versionsnummer helfen direkt zu unterscheiden wo/wann Änderungen geschehen sind, damit Personen, die nichts geändert haben wissen, was anders ist  
**check**

Sortierung  
alle Szenarien, Use Cases zsm **check**

Use Cases:

theoretisch wird ein großes Diagramm erstellt

Auflistung ab Vorbedingung

KEINE Sachen die vor dem Titel passiert sind, wie z.B. Fehlersuche und dann kommt Plan wird erstellt

Nachbedingung: nur direkte Nachbedingung, nicht was in der Zukunft noch passieren könnte **check**

Beschreibung Benutzungsoberfläche mit zugehörigen Use Cases

Domänen-Modell:

Kardinalitäten fehlen **check** und Beschreibung/Bedingung Modul schleife **check**

Praktikum und Vorlesung zu einer Klasse Lehrveranstaltung **check**

nicht Module sondern Lehrveranstaltungen voneinander abhängig -> Kompetenzen **check**  
mehr Beschreibung wäre gut **check**