

Anforderungsspezifikation

SPAsS

(Semester-Planungs-Anwendung für studierende Studierende)

Version 1.2

Datum: 26.05.2022

Inhaltsverzeichnis

1. Projektgrundlagen	1
1.1 Einführung	1
1.2 Zielstellung	1
1.3 Rahmenbedingungen	1
2. Abläufe und Funktionen	2
2.1 Anwendungsszenarien	2
2.2 Funktionale Anforderungen	3
2.2.1 Anwendungsfall-Diagramm	4
2.2.2 Anwendungsfallbeschreibungen	5
2.3 Nicht-Funktionale Anforderungen	12
2.4 Überblick Anforderungen	13
3. Daten-/Domänenmodell.....	14
3.1 Gegenstandswelt des Systems/Datenmodell.....	14
4. Benutzungsschnittstelle.....	15
4.1 Struktur der Anwendung.....	15
4.2 Visuelles Erscheinungsbild	21
5. Ergänzende Bausteine	21
Glossar	21

1. Projektgrundlagen

1.1 Einführung

Die Anwendung „SPAsS“ (Semester-Planungs-Anwendung für studierende Studierende) ist eine interaktive Softwarelösung zur individuellen Studienplanung.

1.2 Zielstellung

Das Ziel der Applikation ist es, Studierenden das eigene Zusammenstellen eines Studienplans zu ermöglichen. Durch eine übersichtliche Aufbereitung des Curriculums soll den Studierenden die Planung des Studiums vereinfacht werden. Die Einschränkungen des jeweiligen Curriculums sowie persönliche Hintergründe und Präferenzen sollen dabei zu jeder Zeit berücksichtigt werden.

1.3 Rahmenbedingungen

Die Anwendung wird innerhalb von 9 Wochen von 4 Entwicklern und Enticklerinnen entwickelt.

Das Programm wird in der Sprache Java (Version 17) umgesetzt. Für die Benutzungsoberfläche wird die grafische Bibliothek JavaFx verwendet. Die Anwendung muss unter Linux voll funktionsfähig und mit Gradle build-, test- und ausführbar sein.

2. Abläufe und Funktionen

2.1 Anwendungsszenarien

Im nachfolgenden werden Szenarien beschrieben, die deutlich machen, von wem und mit welchem Ziel die Anwendung verwendet wird und welche Funktionen zentral sind.

Anwendungsszenario 1:

Die Studentin Activia ist 18 Jahre alt und beginnt im kommenden Wintersemester ihr Medieninformatik-Studium. Sie ist zielstrebig und möchte ihr Studium daher so schnell wie möglich meistern. Sie öffnet nun die SPAsS-Anwendung auf ihrem Laptop und lädt das Curriculum der Medieninformatik an der Hochschule RheinMain in die Software. Ihr werden alle Module in empfohlener Anordnung innerhalb weniger Sekunden angezeigt. Sie liest sich einige Modulbeschreibungen durch und ist mit dem vordefinierten Plan (Regelstudienzeit) zufrieden und speichert sich den Plan ab.

Anwendungsszenario 2:

Julius ist im vierten Semester und ist mit seinem Studium überlastet. Er hat von seinem Professor ein Tool zur Planung seines Studiums empfohlen bekommen, um Modulschiebmöglichkeiten zu ermitteln. Er öffnet die Anwendung, lädt das Curriculum hoch und hakt bereits abgeschlossenen Module ab. Somit erhält er einen Überblick darüber, welche Module ihm noch fehlen. Er schiebt per Drag&Drop Module aus dem kommenden Semester in ein höheres Semester, bekommt allerdings einen Hinweis, dass Kompetenzabhängigkeiten missachtet werden. Er macht die letzte Änderung rückgängig und schiebt stattdessen ein andere Fach. Er speichert seinen aktuellen Stand ab und verlässt die Anwendung.

Anwendungsszenario 3:

Fridolin studiert seit 6 Semestern Medieninformatik und hat den Überblick über die bestanden und ausstehenden Module verloren. Bei einer vergangenen Belegung hatte er bereits das Problem, dass er mit seiner Planung gegen die Fortschrittsregelung verstoßen würde. Daher möchte er sich nun rechtzeitig einen Überblick verschaffen. Er öffnet die Software und trägt bestandene Module und Noten ein. Für eine besonders übersichtliche

2.2.1 Anwendungsfall-Diagramm

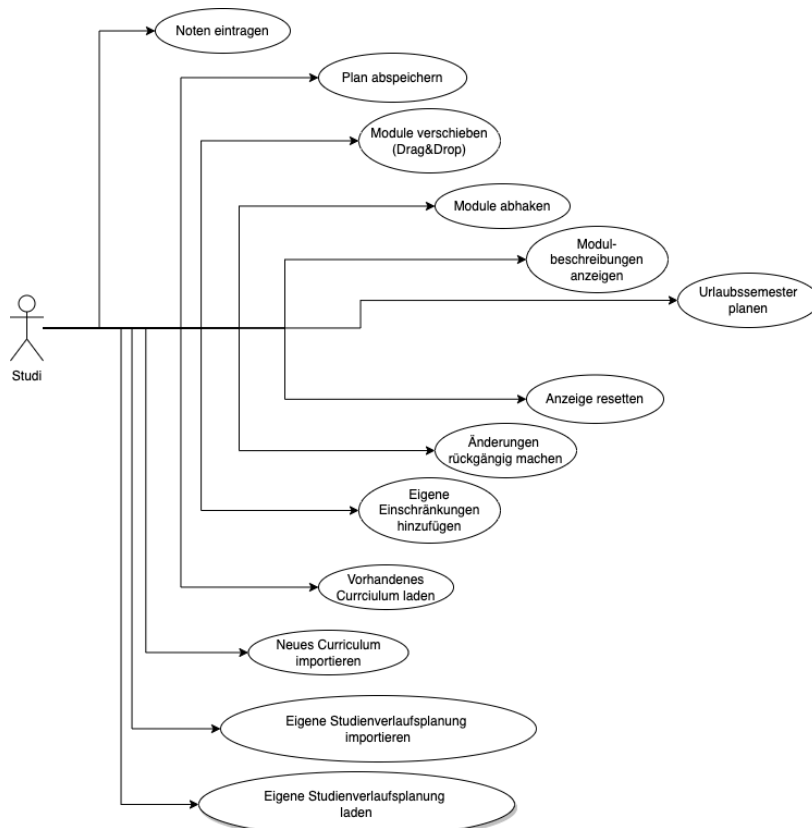


Abbildung 2-1 - Anwendungsfalldiagramm

Abbildung 2-1 veranschaulicht funktionale Anforderungen an die Anwendung. Die hauptsächlichen Akteure sind Studierende. Die jeweiligen Anforderungen, werden in den nachfolgenden Anwendungsfallbeschreibungen spezifiziert.

2.2.2 Anwendungsfallbeschreibungen

Titel: Neues Curriculum importieren

Akteure: Studi

Fachlicher Auslöser: Studienverlauf soll geplant werden

Vorbedingungen: Anwendung gestartet, Curriculum liegt als gültige Datei vor

Standardablauf:

1. Studi: Curriculum-Datei in Anwendung öffnen
2. System: Einlesen der Curriculum-Datei

Alternative Abläufe / Fehlersituationen / Sonderfälle:

- **2a:** Einlesen der Curriculum-Datei schlägt fehl
 - **2a1** System zeigt Fehlermeldung an
 - **2a2** weiter bei 1

Nachbedingung/Ergebnis:

- Default-Anzeige der Module

Nicht-funktionale Anforderungen:

- Ladezeit <2 Sekunden

Parametrisierbarkeit / Flexibilität:

Nutzungshäufigkeit / Mengengerüst:

- Ein Mal pro Studium pro Universität/Hochschule

Titel: Eigene Studienverlaufsplanung laden

Akteure: Studi

Fachlicher Auslöser: Studienverlauf soll geplant werden

Vorbedingungen: Anwendung gestartet, Studienverlaufsdatei liegt als gültige Datei vor (Z.B. vorher gespeichert)

Standardablauf:

1. System: Anzeige der alten Stände
2. Studi: Auswahl des alten Standes
3. System: Laden des alten Standes

Alternative Abläufe / Fehlersituationen / Sonderfälle:

- **1a:** Keine alten Stände vorhanden
 - **1a** weiter bei → Eigenen Studienverlaufsplanung importieren
 - **1b** weiter bei → Neues Curriculum importieren
 - **1c** weiter bei → Vorhandene Curriculum laden
- **3a:** Einlesen der Studienverlaufsdatei schlägt fehl
 - **3a1** System zeigt Fehlermeldung an
 - **3a2** weiter bei 1

Nachbedingung/Ergebnis:

- Individuelle Anzeige der Module nach Studienverlaufsdatei

Nicht-funktionale Anforderungen:

- Ladezeit <2 Sekunden

Parametrisierbarkeit / Flexibilität:

Nutzungshäufigkeit / Mengengerüst:

- Meistens bei Programmöffnen

Titel: Plan abspeichern

Akteure: Studi

Fachlicher Auslöser: Studienverlauf soll geplant werden

Vorbedingungen: Anwendung gestartet, Studienverlaufsdatei/Curriculum geladen, Module werden angezeigt

Standardablauf:

1. Studi: Betätigung des Speichern-Button
2. System: Ablage des aktuellen Studienverlaufsplans und zugehörigen Curriculums im Datei-Verzeichnis der Anwendung
3. System: Anzeige des erfolgreichen Speicherns

Alternative Abläufe / Fehlersituationen / Sonderfälle:

Nachbedingung/Ergebnis:

- Studienverlaufsdatei liegt im gültigen und wiederladbaren Format ab

Nicht-funktionale Anforderungen:

- Speicherzeit <2 Sekunden

Parametrisierbarkeit / Flexibilität:

Nutzungshäufigkeit / Mengengerüst:

- Meistens vor dem Schließen des Programms

Titel: Eigene Einschränkungen hinzufügen (max. CP/Semester)

Akteure: Studi

Fachlicher Auslöser: Studienverlauf soll individuell geplant werden

Vorbedingungen: Anwendung gestartet, Studienverlaufsdatei/Curriculum geladen, Module werden angezeigt

Standardablauf:

1. Studi: Auswahl/Eingabe der Einschränkung im Optionen-Menü
2. System: Aktualisierung der (Fehler-)Hinweise in der Modulanzeige

Alternative Abläufe / Fehlersituationen / Sonderfälle:

- **1a:** ungültige Eingabe
- **1b:** Fehlerhinweis, weiter bei 1

Nachbedingung/Ergebnis:

- Aktualisierte Modulanzeige

Nicht-funktionale Anforderungen:

- Ladezeit <2 Sekunden

Parametrisierbarkeit / Flexibilität:

Nutzungshäufigkeit / Mengengerüst:

- Gelegentlich bei Nutzung, wenn individuellere Einschränkungen vorhanden

Titel: Modulbeschreibung anzeigen

Akteure: Studi

Fachlicher Auslöser: Einholen von Informationen zu einzelnen Modulen

Vorbedingungen: Anwendung gestartet, Curriculum geöffnet und Module werden dargestellt

Standardablauf:

1. Studi: Auswahl eines Moduls durch Klicken
2. System: Darstellung der Informationen des jeweiligen Moduls

Alternative Abläufe / Fehlersituationen / Sonderfälle:

- **2a:** Keine Informationen zu Modul vorhanden
 - **2a1** System zeigt Fehlermeldung an
 - **2a2** weiter bei 1

Nachbedingung/Ergebnis:

- Anzeige der Informationen des ausgewählten Moduls

Nicht-funktionale Anforderungen:

- Ladezeit < 2 Sekunden

Parametrisierbarkeit / Flexibilität:

Nutzungshäufigkeit / Mengengerüst:

- Gelegentlich bei Nutzung, bei Planung eines Studienverlaufs

Titel: Module abhaken

Akteure: Studi

Fachlicher Auslöser: Übersicht über seine Erledigten Module bekommen

Vorbedingungen: Anwendung gestartet, Curriculum geöffnet und dargestellt

Standardablauf:

1. Studi: Check setzen bei bestandenen Modulen (bzw. einzelnen Lehrveranstaltungen der Module)
2. System: Aktualisierung der Modulanzeige mit angepasstem Plan

Alternative Abläufe / Fehlersituationen / Sonderfälle:

Nachbedingung/Ergebnis:

- Aktualisierte Modulanzeige

Nicht-funktionale Anforderungen:

- Direkte Aktualisierung der Modulanzeige

Parametrisierbarkeit / Flexibilität:

Nutzungshäufigkeit / Mengengerüst:

- Bei initialer Nutzung sowie jeweils am Semesterende

Titel: Module verschieben

Akteure: Studi

Fachlicher Auslöser: Studienverlauf soll individuell geplant werden

Vorbedingungen: Anwendung gestartet, Curriculum geöffnet und Module dargestellt

Standardablauf:

1. Studi: Drag&Drop eines Moduls ins gewünschte Semester
2. System: Aktualisierung der Modulanzeige mit angepasstem Plan

Alternative Abläufe / Fehlersituationen / Sonderfälle:

- **1a:** Modul löst nach Verschieben Warnhinweis wegen Einschränkung aus (Markierung)
- **1b:** weiter bei 2

Nachbedingung/Ergebnis:

- Aktualisierte Modulanzeige

Nicht-funktionale Anforderungen:

- Direkte Aktualisierung der Modulanzeige

Parametrisierbarkeit / Flexibilität:

Nutzungshäufigkeit / Mengengerüst:

- Häufig, bei jeder nahezu jeder Nutzung

Titel: Noten eintragen

Akteure: Studi

Fachlicher Auslöser: Überblick über aktuellen Studienstand erhalten und speichern

Vorbedingungen: Anwendung gestartet, Curriculum geöffnet und Module dargestellt

Standardablauf:

1. Studi: Noten in jeweilige Felder in Modulen eintragen
2. System: Aktualisierung der Modulanzeige mit angepasstem Plan

Alternative Abläufe / Fehlersituationen / Sonderfälle:

- **1a:** Ungültigen Wert eingeben
- **1b:** Fehlerhinweis, weiter bei 1

Nachbedingung/Ergebnis:

- Aktualisierte Modulanzeige mit aktuellem „bestanden“-Status, Notendurchschnitt

Nicht-funktionale Anforderungen:

- Direkte Aktualisierung der Modulanzeige

Parametrisierbarkeit / Flexibilität:

Nutzungshäufigkeit / Mengengerüst:

- Bei initialer Nutzung sowie jeweils am Semesterende

2.3 Nicht-Funktionale Anforderungen

Zeitverhalten

- Lade- und Speicherfunktionen des Stundenplans müssen innerhalb weniger Sekunden passieren
- Das Verschieben von Fächern und das damit verbundene aktualisieren des Plans muss ohne sichtbare Verzögerung geschehen

Usability

- Ein Student soll das Programm ohne weitere Schulungen/Einweisungen nutzen können

Erweiterbarkeit/Konfigurierbarkeit

- Ein Programmierer soll im Sourcecode weitere Hochschulspezifischen Einschränkungen ergänzen können

Testbarkeit

- Die Kernlogik der Anwendung wird weitgehend mit JUnit5-Tests abgedeckt

Verlässlichkeit

- Das Programm soll dem Nutzer nachvollziehbar zeigen, ob und warum eine bestimmte Fächerkonstellation nicht möglich ist

Kommentare

- Klassen und die wichtigsten Methoden werden mit JavaDoc-Kommentaren kommentiert

2.4 Überblick Anforderungen

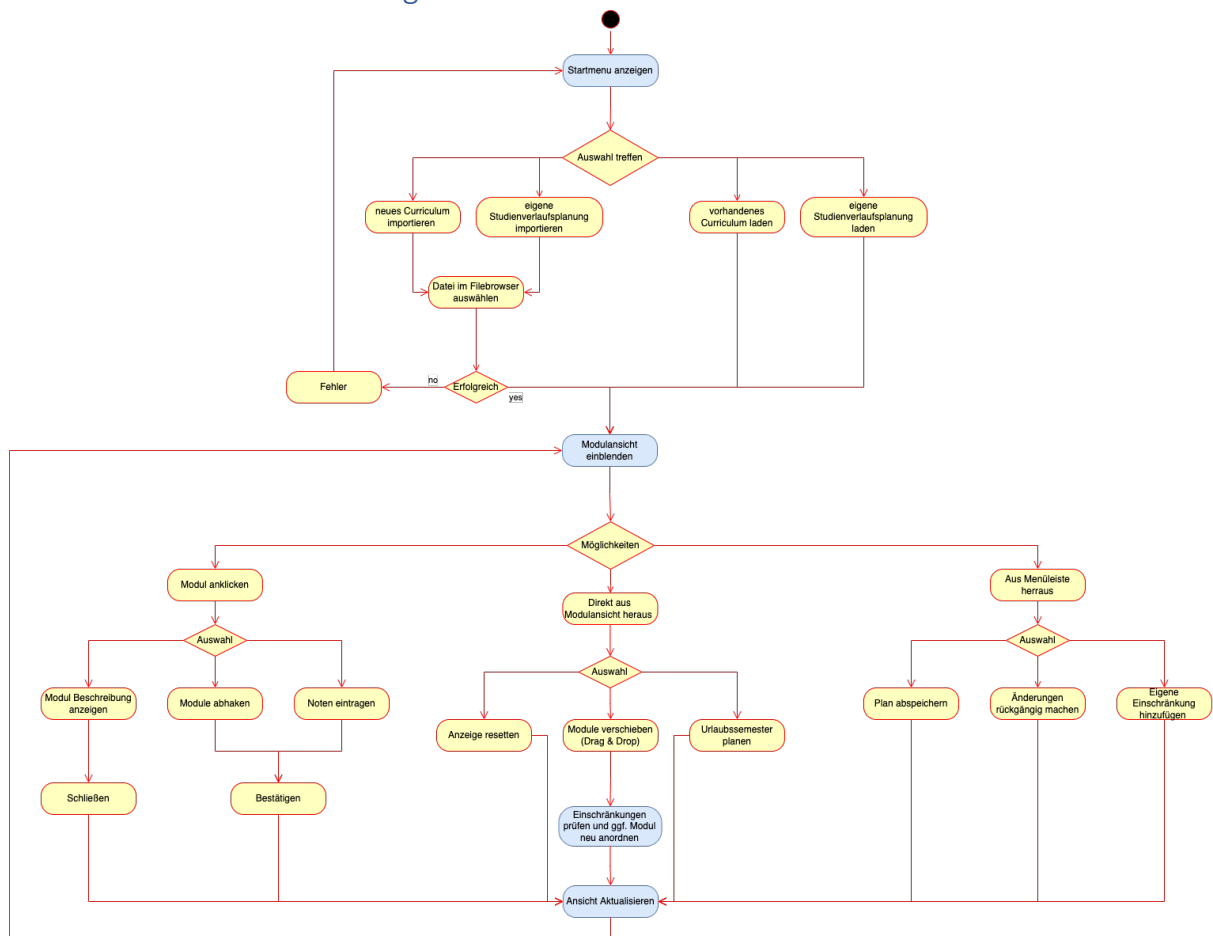


Abbildung 2-2 Aktivitätsdiagramm

Abbildung 2-2 beschreibt in Form eines Aktivitätsdiagramm, wie sich die Nutzenden der Anwendung durch das Programm navigieren.

3. Daten-/Domänenmodell

3.1 Gegenstandswelt des Systems/Datenmodell

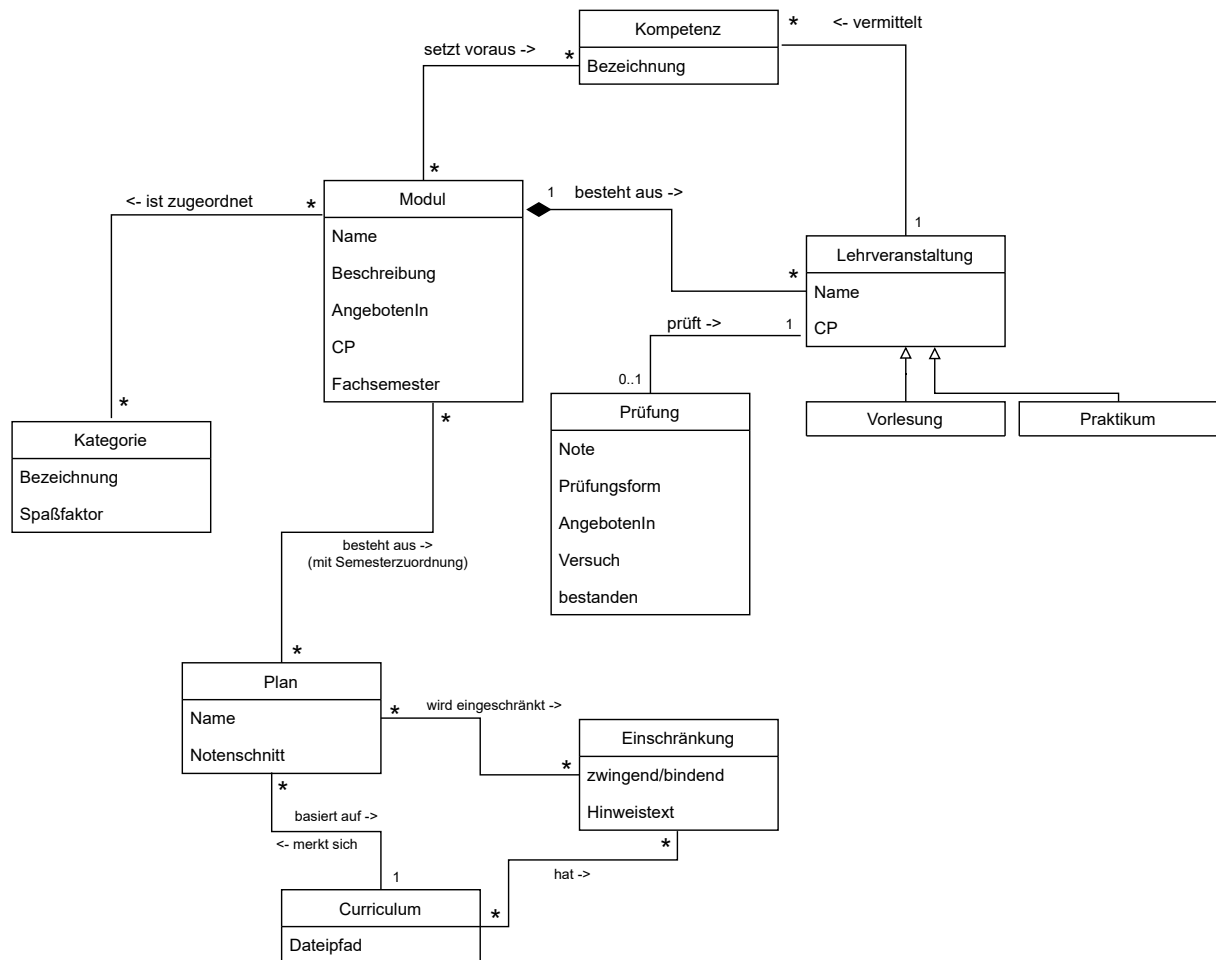


Abbildung 3-1 Domänendiagramm

Aus den Anforderungen an die Anwendungen, wie sie in Kapitel 2 beschrieben wurden, wurden im Domänendiagramm Entitäten und Relationen abgeleitet.

4. Benutzungsschnittstelle

4.1 Struktur der Anwendung

Wireframes veranschaulichen das visuelle Grundgerüst der Anwendung und zeigen, wo die Nutzenden mit der Anwendung interagieren.

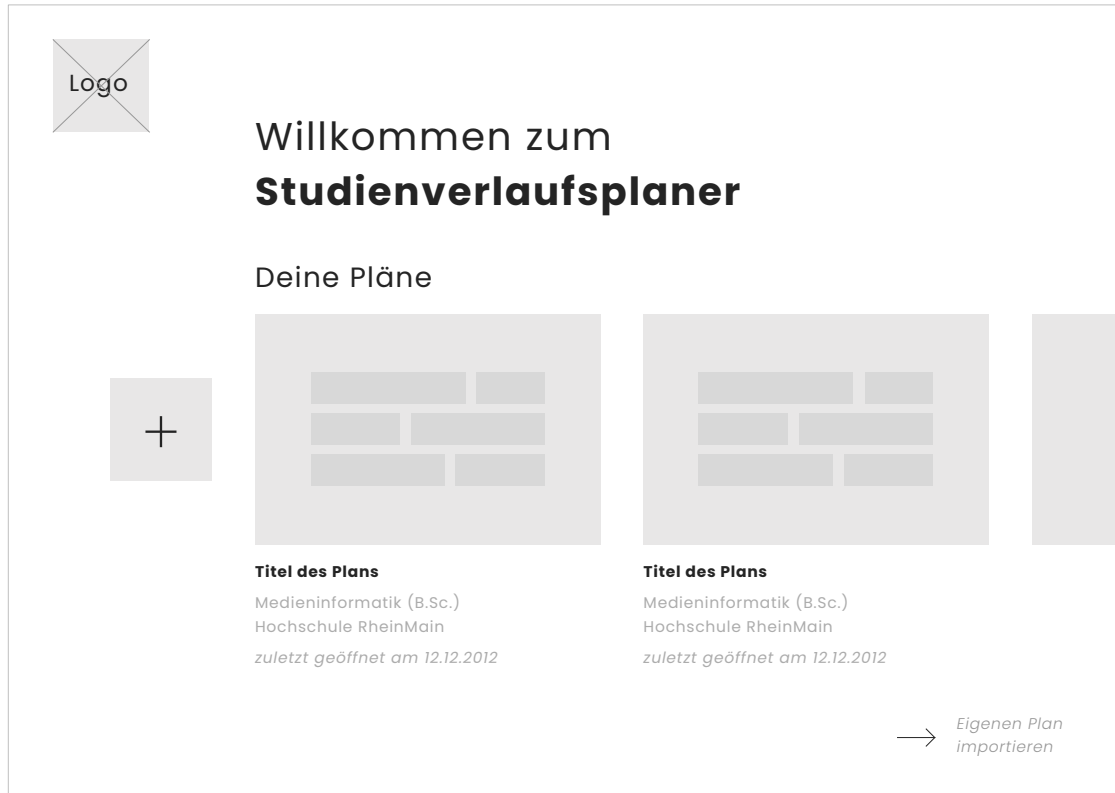


Abbildung 4-1 Einstieg (wenn gespeicherte Pläne vorhanden)



Abbildung 4-2 Einstieg (wenn keine gespeicherten Pläne vorhanden)

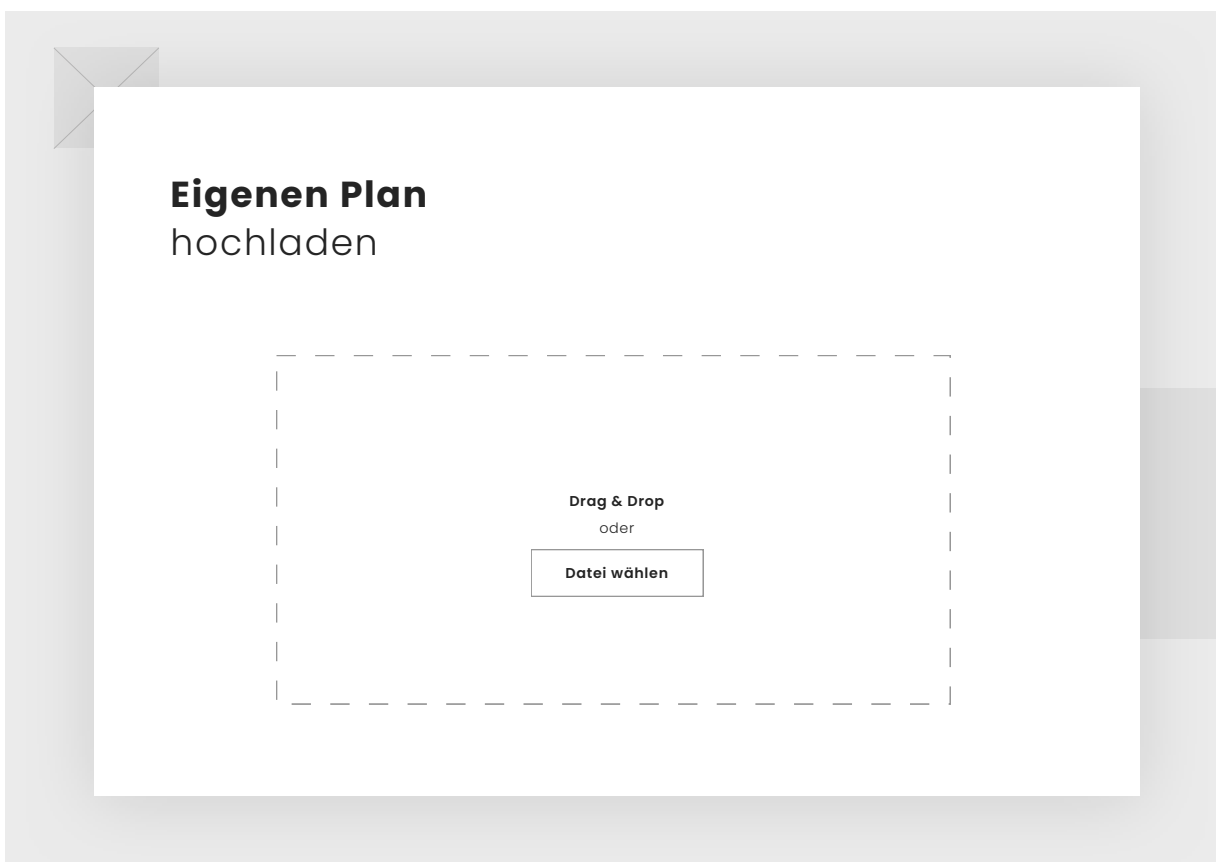


Abbildung 4-3 Upload eigener Pläne

Studiengang wählen

Medieninformatik (B.Sc.)
Hochschule RheinMain



Medieninformatik (B.Sc.)
Hochschule RheinMain



Medienin
Hochschu

Neue Studiengangsdatei
hinzufügen

Drag & Drop
oder

Datei wählen

Abbildung 4-4 Upload/Auswahl Curriculum

Studiengangsdatei hochladen

Drag & Drop
oder

Datei wählen

Abbildung 4-5 Upload Curriculum

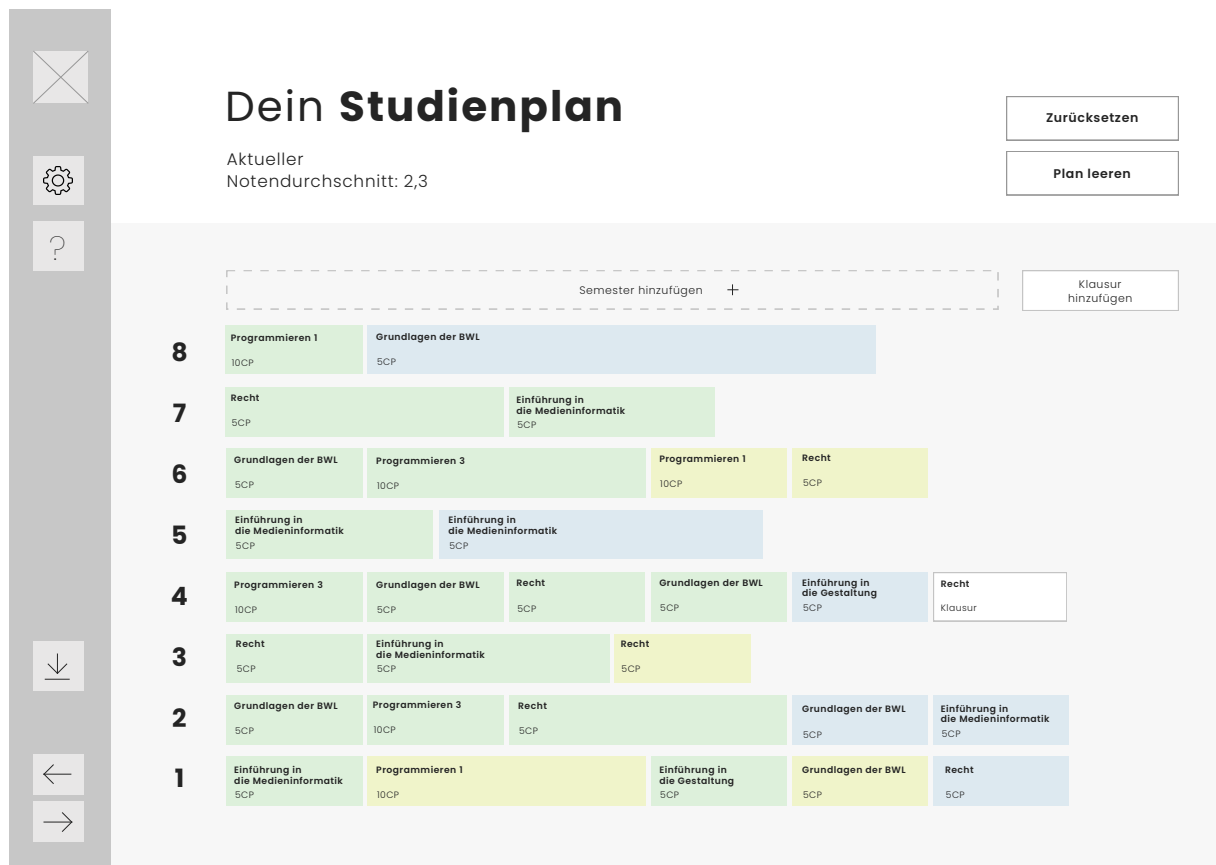


Abbildung 4-6 Modulansicht

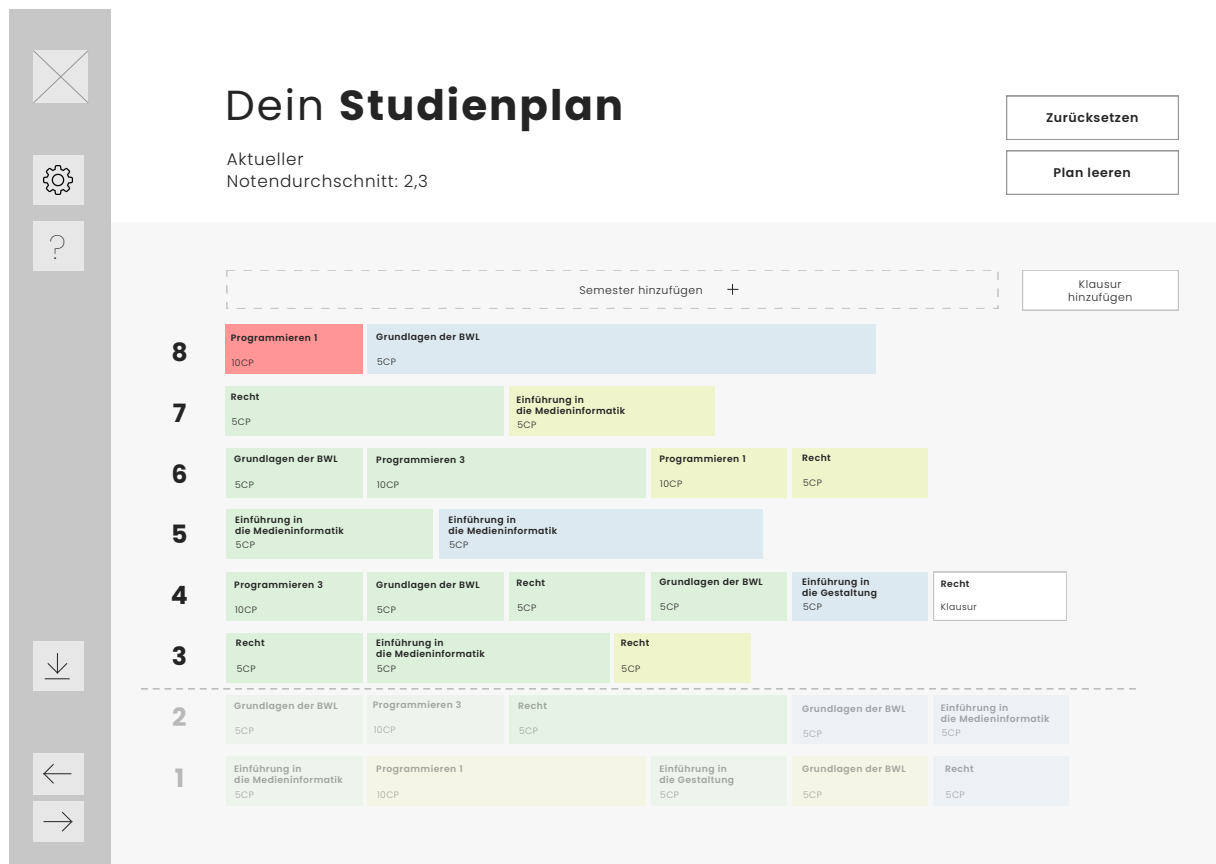


Abbildung 4-7 Modulansicht mit Personalisierungen

✕

⚙️

?

↓

←

→

Dein Studienplan

	Semester h		
8	Programmieren 1 10CP	Grundlagen der BWL 5CP	
7	Recht 5CP		Einführung in die Medieninformatik 5CP
6	Grundlagen der BWL 5CP	Programmieren 3 10CP	
5	Einführung in die Medieninformatik 5CP		Einführung in die Medieninformatik 5CP
4	Programmieren 3 10CP	Grundlagen der BWL 5CP	Recht 5CP
3	Recht 5CP	Einführung in die Medieninformatik 5CP	Recht 5CP
2	Grundlagen der BWL 5CP	Programmieren 3 10CP	Recht 5CP
1	Einführung in die Medieninformatik 5CP	Programmieren 1 10CP	

→

Einführung in die Gestaltung

Kategorie Gestaltung
6 CP

Modul besteht aus

	bestanden	Note
Vorlesung/Klausur	<input type="checkbox"/>	<input type="text"/>
Praktikum	<input type="checkbox"/>	
Übung		

Modulbeschreibung

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Vermittelte Kompetenzen

Objektorientierte Programmierung, Datenstrukturen

Benötigte Kompetenzen

Objektorientierte Programmierung, Datenstrukturen

Abbildung 4-8 Vertiefung Modul

✕

⚙️

?

↓

←

→

Dein Studienplan

	Semester h		
8	Programmieren 1 10CP	Grundlagen der BWL 5CP	
7	Recht 5CP		Einführung in die Medieninformatik 5CP
6	Grundlagen der BWL 5CP	Programmieren 3 10CP	
5	Einführung in die Medieninformatik 5CP		Einführung in die Medieninformatik 5CP
4	Programmieren 3 10CP	Grundlagen der BWL 5CP	Recht 5CP
3	Recht 5CP	Einführung in die Medieninformatik 5CP	Recht 5CP
2	Grundlagen der BWL 5CP	Programmieren 3 10CP	Recht 5CP
1	Einführung in die Medieninformatik 5CP	Programmieren 1 10CP	

→

Einführung in die Gestaltung

Kategorie Gestaltung
6 CP

Warnhinweis

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Modul besteht aus

	bestanden	Note
Vorlesung/Klausur	<input type="checkbox"/>	<input type="text"/>
Praktikum	<input type="checkbox"/>	
Übung		

Modulbeschreibung

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos

Abbildung 4-9 Vertiefung Modul mit Warnhinweis



Abbildung 4-10 Vertiefung Klausur hinzufügen

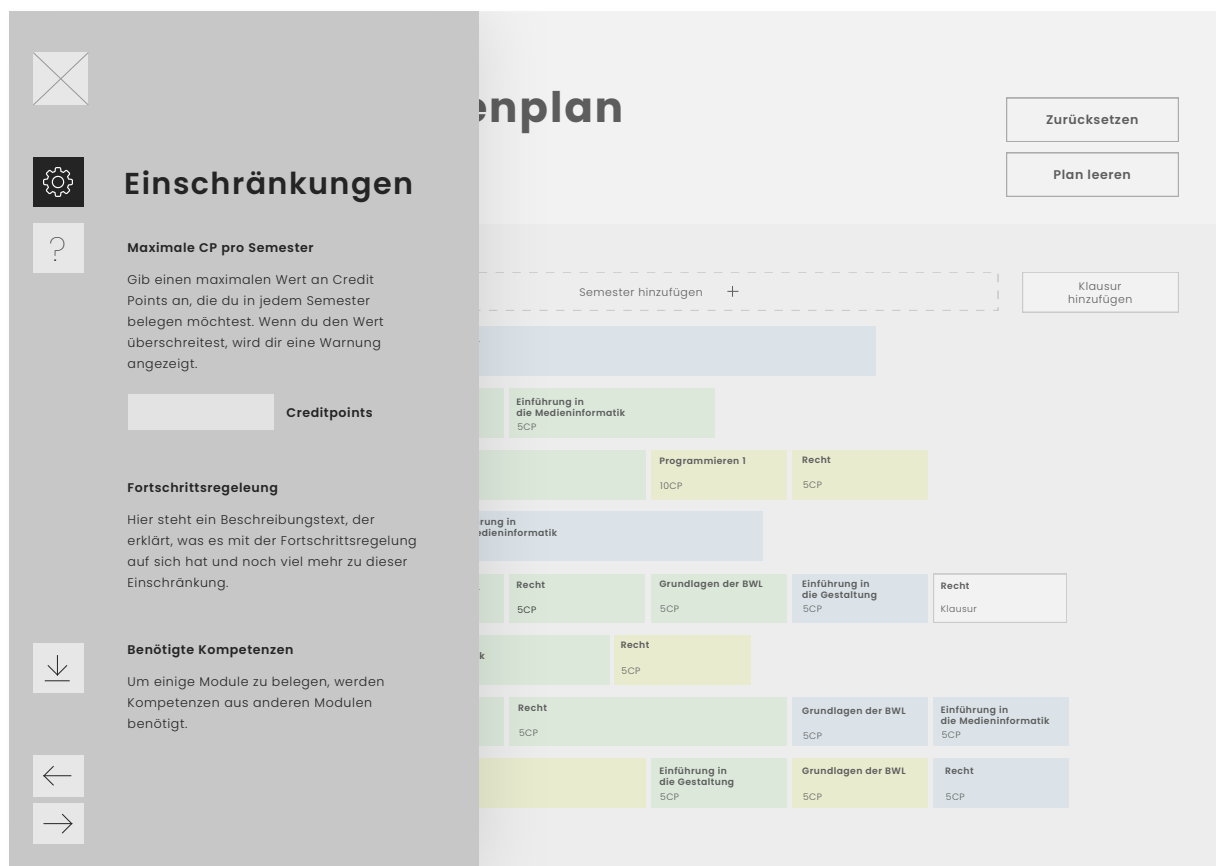


Abbildung 4-11 Vertiefung Einschränkungen

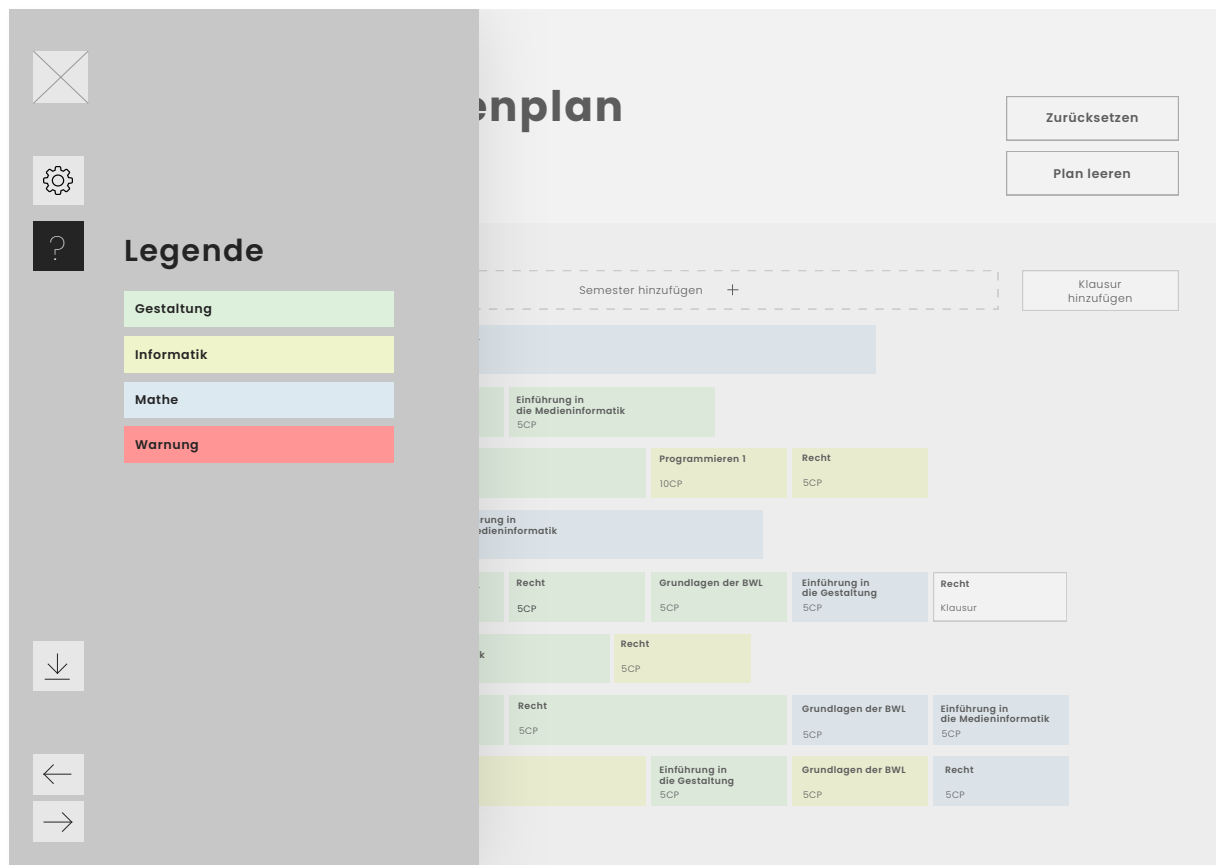


Abbildung 4-12 Vertiefung Legende

4.2 Visuelles Erscheinungsbild

5. Ergänzende Bausteine

Glossar