

# Studienplanung als Generierung von Workflows mit Compliance-Anforderungen: Planerstellung und Visualisierung. IPD Böhm

5. Dezember 2016

Phase	Verantwortlicher
Pflichtenheft	Ali Bejhad
Entwurf	Janek Westfechtel
Implementierung	Clemens Naseband
Qualitätssicherung	Robin Berger
Präsentation	Jacques Huss

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>4</b>
<b>2</b>	<b>Zielbestimmung</b>	<b>5</b>
2.1	Musskriterien . . . . .	5
2.2	Wunschkriterien . . . . .	5
2.3	Abgrenzungskriterien . . . . .	6
<b>3</b>	<b>Produkteinsatz</b>	<b>7</b>
3.1	Anwendungsbereich . . . . .	7
3.2	Zielgruppe . . . . .	7
3.3	Betriebsbedingungen . . . . .	7
<b>4</b>	<b>Produktumgebung</b>	<b>8</b>
4.1	Software . . . . .	8
4.2	Hardware . . . . .	8
<b>5</b>	<b>Produktfunktionen</b>	<b>9</b>
5.1	Grundfunktionen . . . . .	9
5.2	Erweiterte Funktionen . . . . .	9
<b>6</b>	<b>Produktdaten</b>	<b>10</b>
6.1	Systemdaten . . . . .	10
6.2	Benutzerdaten . . . . .	10
<b>7</b>	<b>Systemmodell</b>	<b>11</b>
<b>8</b>	<b>Produktleistungen</b>	<b>14</b>
<b>9</b>	<b>Bedienoberfläche</b>	<b>15</b>
9.1	Einführung . . . . .	15
9.2	Aufbau . . . . .	16
9.3	Bedienung . . . . .	16
<b>10</b>	<b>Qualitätszielbestimmungen</b>	<b>17</b>
<b>11</b>	<b>Testfälle und Testszenarien</b>	<b>18</b>
11.1	Testfälle . . . . .	18
11.1.1	Basis-Testfälle . . . . .	18
11.1.2	Erweiterte Testfälle . . . . .	19
11.2	Testszenarien . . . . .	19
11.2.1	Testszenario: Studienplan Generierung . . . . .	19
11.2.2	Testszenario: Möglichkeiten prüfen . . . . .	20
11.2.3	Testszenario: Systemstart . . . . .	20

11.2.4	Testszenario: Fehlerhafter Studienplan . . . . .	21
11.2.5	Manuelles Erstellen . . . . .	21
11.2.6	Testszenario: Studienplan wird gespeichert . . . . .	22
11.2.7	Testszenario: Studienplan wird geladen . . . . .	22
11.3	Testszenario: Alternativer Studienplan . . . . .	23
11.3.1	Testszenario: Studienplan extern speichern (#) . . . . .	24
11.3.2	Testszenario: Studienfach-/ Sprachauswahl(#) . . . . .	24
<b>12</b>	<b>Entwicklungsumgebung</b>	<b>25</b>
12.1	Software . . . . .	25
12.2	Hardware . . . . .	26

# 1 Einleitung

Mit wechselnden Prüfungsordnungen und großen Modulhandbüchern steigt die Unübersichtlichkeit des Studienplanes. Mithilfe dieser Software soll es Studenten ermöglicht werden, ihre Studienpläne mit allen Pflichtmodulen, Orientierungsprüfungen, Wahlmodulen und Sonstigen, als ein Prozessmodell/Workflow darzustellen. Hierbei sollen sowohl Constraints als auch benötigte ECTS vom System berücksichtigt werden. Zusätzlich soll der Student in der Lage sein, zu bestimmen wie viele Semester er studieren will und in welchem er welches Modul abschließen will. Hierbei soll dem Studenten eine Rückmeldung zurückgegeben werden, die ihn auf Fehler aufmerksam macht.

Das System soll intuitiv nutzbar sein und den Nutzer bei der Bedienung unterstützen.

Die Constraints sind hierbei von einer Datenbank vorgegeben. Es sollen Tools zur Generierung und Verifizierung des Studienplanes erstellt werden, welche später durch andere ersetzt werden können.

Mit dieser Software sollen Fehler und Missverständnisse beim Studium minimiert werden. Einsätze von Workflows haben sich bereits in anderen Gebieten als hilfreich erwiesen.

## 2 Zielbestimmung

### 2.1 Musskriterien

- Das Programm liest die Module und deren Constraints, zu Beginn des Einsatzes, aus einer Datenbank aus.
- Der Benutzer kann mehrere Studienpläne als Alternativen erstellen und jederzeit wechseln.
- Das Programm generiert Studienpläne, unter Berücksichtigung der Constraints.
- Der Benutzer kann angeben, unter welchen Punkten seine Studienplangenerierung berücksichtigt werden soll. Darunter gehören, die Wunschlänge des Studiums, Anzahl der ECTS pro Semester, Wunschmodule oder Vertiefungsgebiete.
- Der Benutzer kann seine bisher erbrachten Leistungen angeben, welche vom System berücksichtigt werden.
- Manuelle Änderungen am Studienplan werden verifiziert.
- Studienpläne werden dem Benutzer, in Form einer Tabelle, auf einer aufrufbaren Website, visuell dargestellt.
- Das System ordnet die ausgewählten und ergänzten Module in deren zugehörigen Semester ein und zeigt diese an.
- Das System unterscheidet zwischen Pflichtmodulen, Wahlmodulen und Vertiefungsfächer.

### 2.2 Wunschkriterien

- Kurze Wartezeit des Nutzers bei kleinen Änderungen.
- Der Studienplan wird in Echtzeit aktualisiert.
- Das System informiert den Nutzer über die Art des Übungsscheins eines Moduls.
- Exportieren des generierten Studienplan als Bild oder PDF.

- Speichern der bereits ausgewählten Einstellungen und Eingaben des Benutzers und laden dieser bei der nächsten Verwendung des Systems.
- Die GUI ist auf verschiedenen Sprachen verfügbar.
- Dem System werden weitere Optimierungsoptionen zur Verfügung gestellt.

### **2.3 Abgrenzungskriterien**

- Es werden keine anderen Studiengänge als Informatik Bachelor/Master betrachtet, jedoch im Entwurf berücksichtigt.
- Die Seite wird nicht für Mobilgeräte wie Mobiltelefone oder Tablets angeboten.
- Eine Ansicht für Dozenten wird im Entwurf beachtet, jedoch nicht implementiert.

## **3 Produkteinsatz**

### **3.1 Anwendungsbereich**

Das Programm dient der Planung des Studiums. Der Nutzer kann sich einen vorgeschlagenen Ablauf seines Studiums als Tabelle anzeigen lassen und diesen modifizieren und anpassen. Dazu verwendet er einen Browser auf einem Computer.

### **3.2 Zielgruppe**

Das Programm richtet sich an Informatikstudenten des KIT, die ihr Studium planen wollen. Die graphische Benutzeroberfläche des Systems soll benutzerfreundlich sein. Die Bedienung des Systems setzt von Studentenseite keine Vorkenntnisse voraus.

Für den Einsatz durch die Systemadministratoren wird eine Bedienungsanleitung in Verbindung mit ausreichender Dokumentation verfasst.

### **3.3 Betriebsbedingungen**

Es müssen folgende Voraussetzungen erfüllt sein:

- Ein Computer mit aktuellem Browser und Internetanbindung.
- Ein Server mit JVM Unterstützung.
- Während der Nutzung muss die Strom- und Internetversorgung, sowohl auf Seite des Client als auch des Servers, gewährleistet sein.

## 4 Produktumgebung

### 4.1 Software

- Die JVM 1.8 oder höher auf der Serverseite.
- Apache Tomcat als Java Servlet Umgebung auf der Serverseite.
- Ein moderner Webbrowser wie Google Chrome 54.0, Opera 40.0 oder Firefox 50.0 auf der Clientseite.
- JavaScript muss auf dem Browser aktiviert sein.

### 4.2 Hardware

- Ein Computer mit Internetanschluss und mindestens 2GB Arbeitsspeicher als Server.
- Ein Computer mit Internetanschluss und mindestens 1GB Arbeitsspeicher als Client.
- Ein Farbbildschirm mit einer Auflösung von 1280x720 Pixel (HD) am Client.
- Mindestens eine Maus als Eingabegerät, sowie eine Tastatur zur Unterstützung.



## 5 Produktfunktionen

### 5.1 Grundfunktionen

Nummer	Beschreibung	Serverseite/Clientseite
/F010/	Das Programm lädt die Datenbank.	S
/F020/	Anzeigen des Startbildschirmes.	C
/F030/	Auswahl eines Moduls aus der Liste.	C
/F040/	Modul hinzufügen.	C
/F050/	Modul als schon Bestanden markieren.	C
/F060/	Erstellung eines Studienplanes.	S
/F070/	Eingaben des Benutzers beim Client speichern.	C
/F080/	Eingaben des Benutzers beim Client laden.	C
/F090/	Wechseln von Studienplänen (Neuer Studienplan).	C
/F100/	Anzeigen eines Studienplanes.	C
/F110/	Studienplan verifizieren.	S

### 5.2 Erweiterte Funktionen

Notwendige Funktionen um die Wunschkriterien zu erfüllen.

Nummer	Beschreibung	Serverseite/Clientseite
/F120/	Studienfach auswählen.	C
/F130/	Studienplan als PDF/Bild exportieren.	C/S
/F140/	Sprache auswählen.	C
/F150/	Drag and Drop Unterstützung.	C

## **6 Produktdaten**

### **6.1 Systemdaten**

/D010/ Module mit zugehörigen Vorlesungen.

/D020/ Prüfungsvoraussetzungen und Abhängigkeiten zu anderen Modulen.

/D030/ Anzahl der ECTS der Module.

/D040/ Vorlesung im Sommer- oder Wintersemester.

/D050/ Sprachdateien (properties).

/D060/ Styledateien (CSS).

### **6.2 Benutzerdaten**

/D070/ Bereits bestandene Prüfungen.

/D080/ Fachsemester.

/D090/ Einstellungen und Präferenzen.

## 7 Systemmodell

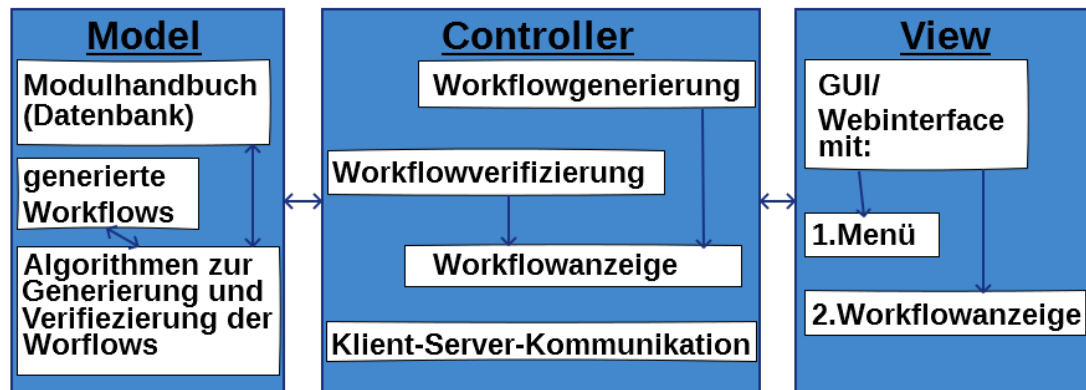


Abbildung 1: Systemmodell

Die Grundstruktur dieses Programmes basiert auf einer MVC-Struktur. Dabei soll die Dreiteilung in Model, View und Controller sowohl eine erleichterte Modifikation am Programmcode als auch ein einfaches Einbinden von weiterführenden Tools gewährleisten. Die drei Komponenten sollen dabei folgenden Aufgaben erfüllen:

### Model:

Das Model enthält die Datenstruktur des Modulhandbuchs, die generierten Workflows und die Workflowalgorithmen. Die Datenstruktur (eine MySQL-Datenbank) speichert die einzelnen Module und Veranstaltungen des Modulhandbuchs, mit ihren Einschränkungen und Voraussetzungen. Die Algorithmen dienen zur Generierung und Verifizierung der Workflows. Die Algorithmen werden modularisiert, so dass verschiedene Strategien die Probleme lösen können.

### View:

Die View hat zwei Aufgaben. Zum einen gibt sie dem Benutzer eine Weboberfläche, in dem er seine Präferenzen für den Studienplan angeben kann, damit diese in den Workflow einfließen können. Zum anderen stellt sie den generierten Workflow dar.

### Controller:

Der Controller ist die Zentraleinheit des Systems. Er koordiniert die Erstellung und die Verifizierung der Workflows, indem er die Angaben des Benutzers empfängt, auswertet

und den generierten Workflow auf dem Webinterface anzeigen lässt. Zudem stellt er für den Benutzer die Option bereit, den eigenen generierten Workflow herunterzuladen und abzuspeichern, um diesen auch ohne die Benutzung des Internetdienstes anzuschauen. Ebenso wird es möglich sein, einen abgespeicherten Workflow zu laden und diesen weiter zu modifizieren. Die Workflow-Generierung und Workflow-Verifizierung sind einzelne Tools, die beliebig ausgetauscht werden können.

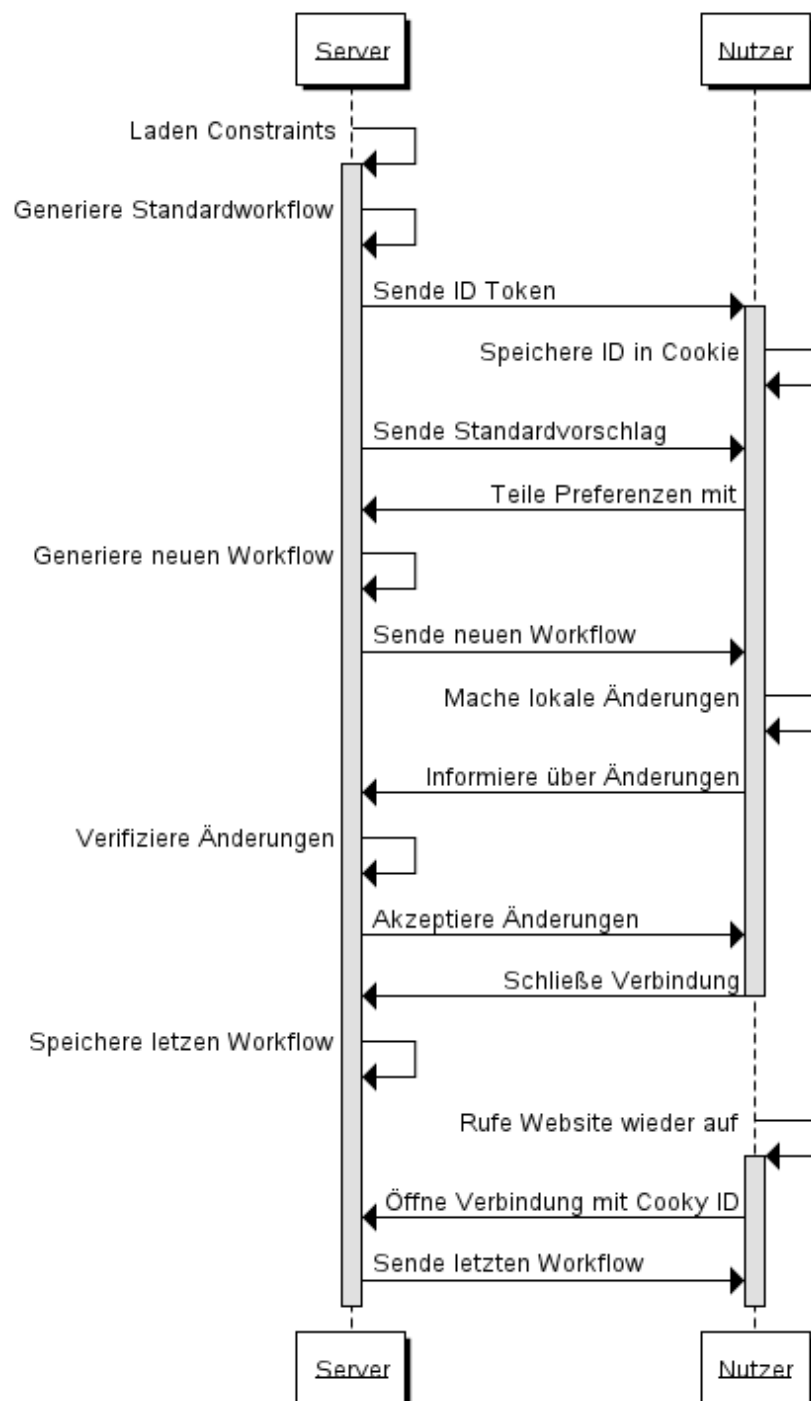


Abbildung 2: Beispielinteraktion zwischen Server und Client

## 8 Produktleistungen

- /L010/ Das Einlesen der Datenbank, darf nur einmalig beim Start des System oder bei Eingabe eines Administrators stattfinden. Dieser Vorgang darf lange dauern.
- /L020/ Als Serveranwendung muss das System dauerhaft laufen.
- /L030/ Das System darf, beim öffnen der Seite, nicht länger als 3 Sekunden brauchen, um das Auswahl- und Eingabefenster für den Benutzer anzuzeigen.
- /L040/ Das System muss vollständig mit einer Maus bedienbar sein.

## 9 Bedienoberfläche

### 9.1 Einführung

Da die Bedienoberfläche in einem Browser angezeigt wird muss diese mit flexiblen Dimensionen zurecht kommen und dennoch dem Nutzer möglichst simpel und eingängig sein. Deswegen wird eine Explorer-GUI mit einem festen Menü am linken Rand und einer großen variablen Leinwand zur Anzeige des Studienplans verwendet.

Der Nutzer benötigt keine Vorkenntnisse, die Bedienung soll nur durch die Maus erfolgen und bei Eingaben durch eine Tastatur unterstützt werden.

Er kann den vorgeschlagenen Studienplan durch Änderungen im Menü beliebig ändern und gegebenen Falls bei Missfallen rückgängig machen. Die intuitive Verwendung wird durch vertraute Symbole sichergestellt (eine Lupe ermöglicht das Vergrößern, ein Disketensymbol speichert, usw.).

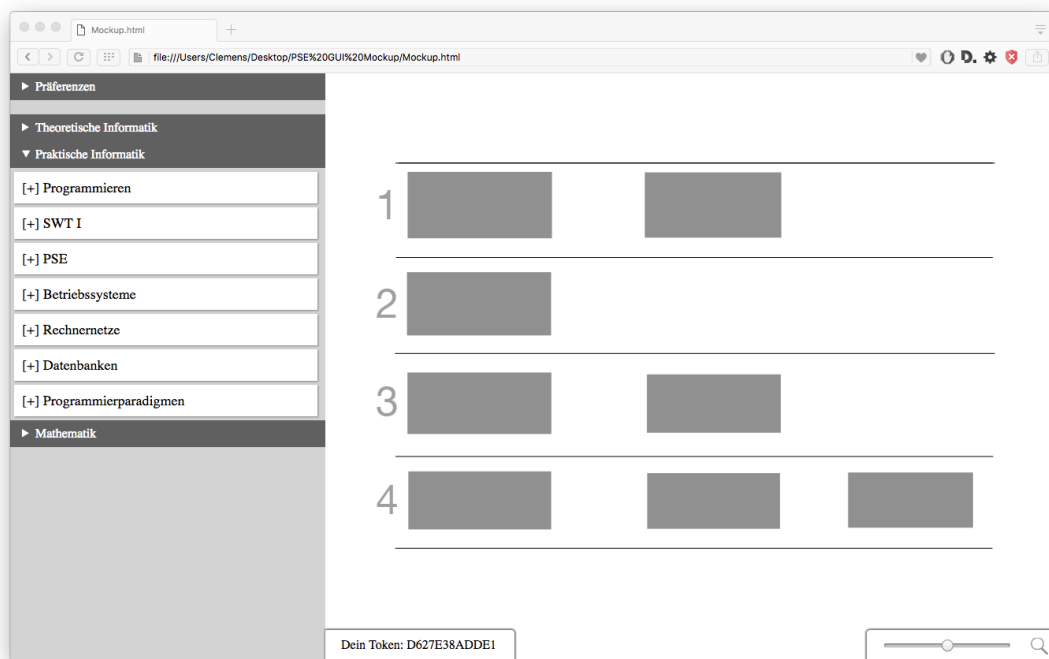


Abbildung 3: Ansicht im Browser

Der hier angegebene Entwurf ist nur Vorläufig und dient als Leitfaden. Die Bedienoberfläche kann sich auf Grund von Wunschkriterien oder Vereinfachungen ändern.

## **9.2 Aufbau**

Das grau hinterlegte Menü auf der linken Seite ermöglicht dem Nutzer weitere Module hinzuzufügen. Die Menü-Hierarchie soll möglichst flach sein. Zunächst sieht der Nutzer nur die eingeklappten Überpunkte aus denen er auswählen kann. Die Breite des Menüs ist festgelegt.

Auf der rechten Seite befindet sich die Leinwand, welche dem Nutzer seinen Studienplan präsentiert. Sie nimmt die restliche Breite der Seite ein, da auf ihr das Hauptaugenmerk liegen soll.

Im unteren Bereich der Leinwand, werden dem Nutzer Informationen, wie seine ID, mit der er von einem anderen Rechner auf seinen Studienplan zugreifen kann, angezeigt oder ihm die Möglichkeit geboten, die Größe des Studienplans manuell zu verändern.

## **9.3 Bedienung**

Im gesonderten ersten Punkt kann er seine Präferenzen eingeben wie z.B. Anzahl der ECTS Punkte, angestrebte Studiendauer oder Ähnliches. Darunter werden ihm alle Überkategorien der belegbaren Module angezeigt.

Durch einen Klick auf den Überpunkt kann der Nutzer sich diesen anzeigen oder ausblenden lassen, falls er ihn nicht benötigt.



## 10 Qualitätszielbestimmungen

- /Q010/ Die Robustheit, für bis zu 100 Anfragen, zur gleichen Zeit, wird sichergestellt.
- /Q020/ Die Unterstützung der gängigsten Browser (Mozilla Firefox, Google Chrome, Opera) wird gewährleistet.
- /Q030/ Die Benutzerfreundlichkeit wird, zum einen durch eine einfache übersichtliche GUI verwirklicht, und zum anderen, durch eine einfache Handhabung gewährleistet.
- /Q040/ Eine kurze Wartezeit von wenigen Sekunden zur Generierung der Studienplans wird sichergestellt.
- /Q050/ Die Korrektheit der Studienpläne, im Sinne des Modulhandbuchs und der Anforderungen des Benutzers, wird sichergestellt.
- /Q060/ Wart – und Wiederverwendbarkeit bzw. eine leichte Weiterentwicklung und Einbinden von Tools, wird durch die Implementierung der MVC-Architektur gewährleistet.
- /Q070/ Eine geringe Auslastung der Ressourcen des Benutzers, wird durch eine Client-Serverstruktur gewährleistet.
- /Q080/ Eine dauerhafte Verfügbarkeit des Dienstes für die Benutzer wird durch den Einsatz eines Servers gewährleistet.

## 11 Testfälle und Testszenarien

### 11.1 Testfälle

Jede Funktion wird durch einen Testfall geprüft. Testfälle sind atomare Vorgänge. Es wird zwischen Basis, und erweiterten Testfällen unterschieden.

#### 11.1.1 Basis-Testfälle

- /T010/ Webseite lässt sich aufrufen.
- /T020/ System kann sich mit Datenbank verbinden.
- /T030/ System liest Datenbank korrekt aus.
- /T040/ System verarbeitet Moduldateien korrekt.
- /T050/ GUI stellt Module korrekt dar.
- /T060/ Modul lässt sich auswählen.
- /T070/ Modul lässt sich als bestanden markieren.
- /T080/ Modul lässt sich hinzufügen.
- /T090/ Präferenzen lassen sich auswählen.
- /T100/ Studienplan wird generiert.
- /T110/ GUI zeigt Studienplan korrekt an.
- /T120/ Studienplan wird verifiziert.
- /T130/ Studienplan wird gespeichert.
- /T140/ Studienplan wird geladen.
- /T150/ Neuer Studienplan wird erstellt.

### 11.1.2 Erweiterte Testfälle

/T160/ Ein Studienfach lässt sich auswählen.

/T170/ Studienplan wird als PDF/Bild ausgegeben.

/T180/ GUI zeigt bei Sprachänderung korrekte Zeichen und die Zielsprache an.

## 11.2 Testszenarien

Die Anwendungsszenarien sollen sicherstellen dass das System größtenteils fehlerfrei benutzbar ist. Sie stellen beispielhaft Szenarien dar, die das System bei Benutzung durchläuft. Sie setzen sich aus den Testfällen zusammen.

Mit (#) markierte Testszenarien setzen erweiterte Testfälle ein.

### 11.2.1 Testszenario: Studienplan Generierung

Der Benutzer ruft die Seite auf, wählt verschiedene Präferenzen (Studienlänge, Wahlmodule, Vertiefungsgebiet, etc.) aus, und lässt sich vom System einen Studienplan generieren.

1. /T010/ Webseite lässt sich aufrufen
2. /T050/ GUI stellt Module korrekt dar.
3. /T060/ Modul lässt sich auswählen.
4. /T070/ Modul lässt sich als bestanden markieren.
5. /T080/ Modul lässt sich hinzufügen.
6. /T090/ Präferenzen lassen sich auswählen.
7. /T100/ Studienplan wird generiert.
8. /T110/ GUI zeigt Studienplan korrekt an.

### 11.2.2 Testszenario: Möglichkeiten prüfen

Der Benutzer gibt seine bereits bestandenen Studienfächer ein, und prüft verschiedene Konfigurationen an Modulen und Bedingungen.

1. /T010/ Webseite lässt sich aufrufen
2. /T050/ GUI stellt Module korrekt dar.
3. /T060/ Modul lässt sich auswählen.
4. /T070/ Modul lässt sich als bestanden markieren.
5. /T080/ Modul lässt sich hinzufügen.
6. /T090/ Präferenzen lassen sich auswählen.
7. /T100/ Studienplan wird generiert.
8. /T110/ GUI zeigt Studienplan korrekt an.
9. /T060/ Modul lässt sich auswählen.
10. /T070/ Modul lässt sich als bestanden markieren.
11. /T080/ Modul lässt sich hinzufügen.
12. /T090/ Präferenzen lassen sich auswählen.
13. Wiederholen von 9.-12.
14. /T120/ Studienplan wird verifiziert.
15. Beliebiges Wiederholen von 9.-14.

### 11.2.3 Testszenario: Systemstart

Das System liest Einträge aus der Datenbank ein und verarbeitet diese korrekt.

1. /T020/ System kann sich mit Datenbank verbinden.

2. /T030/ System liest Datenbank korrekt aus.
3. /T040/ System verarbeitet Moduldateien korrekt.

#### **11.2.4 Testszenario: Fehlerhafter Studienplan**

Ein Benutzer gibt Vorgaben deren Einhaltung unmöglich ist. Das System gibt eine Fehlermeldung.

1. /T010/ Webseite lässt sich aufrufen
2. /T050/ GUI stellt Module korrekt dar.
3. /T060/ Modul lässt sich auswählen.
4. /T070/ Modul lässt sich als bestanden markieren.
5. /T080/ Modul lässt sich hinzufügen.
6. /T090/ Präferenzen lassen sich auswählen.
7. /T100/ Studienplan wird generiert.
8. /T110/ GUI zeigt Studienplan korrekt an.
9. /T060/ Modul lässt sich auswählen.
10. /T070/ Modul lässt sich als bestanden markieren.
11. /T080/ Modul lässt sich hinzufügen.
12. /T090/ Präferenzen lassen sich auswählen.
13. /T120/ Studienplan wird verifiziert (Schlägt fehl).

#### **11.2.5 Manuelles Erstellen**

Ein Benutzer erstellt sich einen Studienplan komplett manuell, ohne Hilfe der Generierung.

1. /T010/ Webseite lässt sich aufrufen.
2. /T050/ GUI stellt Module korrekt dar.
3. /T060/ Modul lässt sich auswählen.
4. /T070/ Modul lässt sich als bestanden markieren.
5. /T080/ Modul lässt sich hinzufügen.
6. Wiederholen von 3.-5. bis der Studienplan vollständig gefüllt ist.
7. /T120/ Studienplan wird verifiziert.

#### **11.2.6 Testszenario: Studienplan wird gespeichert**

Ein Benutzer bearbeitet einen Studienplan, dieser wird gespeichert.

1. /T010/ Webseite lässt sich aufrufen.
2. /T050/ GUI stellt Module korrekt dar.
3. /T060/ Modul lässt sich auswählen.
4. /T070/ Modul lässt sich als bestanden markieren.
5. /T080/ Modul lässt sich hinzufügen.
6. /T090/ Präferenzen lassen sich auswählen.
7. /T100/ Studienplan wird generiert.
8. /T110/ GUI zeigt Studienplan korrekt an.
9. /T130/ Studienplan wird gespeichert.

#### **11.2.7 Testszenario: Studienplan wird geladen**

Das System lädt einen gespeicherten Studienplan, und der Benutzer bearbeitet diesen. Wenn die Bearbeitung fertig ist, wird der neue Studienplan abgespeichert

1. /T010/ Webseite lässt sich aufrufen.
2. /T140/ Studienplan wird geladen.
3. /T060/ Modul lässt sich auswählen.
4. /T070/ Modul lässt sich als bestanden markieren.
5. /T080/ Modul lässt sich hinzufügen.
6. /T090/ Präferenzen lassen sich auswählen.
7. /T120/ Studienplan wird verifiziert.
8. /T130/ Studienplan wird gespeichert.

### **11.3 Testszenario: Alternativer Studienplan**

Der Benutzer möchte einen alternativen Studienplan erstellen, wobei der alte nicht gelöscht wird

1. /T010/ Webseite lässt sich aufrufen.
2. /T140/ Studienplan wird geladen.
3. /T150/ Neuer Studienplan wird erstellt.
4. /T060/ Modul lässt sich auswählen.
5. /T070/ Modul lässt sich als bestanden markieren.
6. /T080/ Modul lässt sich hinzufügen.
7. /T090/ Präferenzen lassen sich auswählen.
8. /T100/ Studienplan wird generiert.
9. /T110/ GUI zeigt Studienplan korrekt an.

### 11.3.1 Testszenario: Studienplan extern speichern (#)

Ein Benutzer speichert seinen Studienplan als PDF/Bild ab.

1. /T010/ Webseite lässt sich aufrufen.
2. /T050/ GUI stellt Module korrekt dar.
3. /T060/ Modul lässt sich auswählen.
4. /T070/ Modul lässt sich als bestanden markieren.
5. /T080/ Modul lässt sich hinzufügen.
6. /T090/ Präferenzen lassen sich auswählen.
7. /T100/ Studienplan wird generiert.
8. /T110/ GUI zeigt Studienplan korrekt an.
9. /T160/ Studienplan wird als PDF ausgegeben.

### 11.3.2 Testszenario: Studienfach-/ Sprachauswahl(#)

Der Benutzer wählt ein anderes Studienfach oder eine Sprache aus.

- /T010/ Webseite lässt sich aufrufen
- /T150/ Ein Studienfach lässt sich auswählen.
- /T170/ GUI zeigt bei Sprachänderung korrekte Zeichen und die Zielsprache an.
- /T050/ GUI stellt Module korrekt dar.



## 12 Entwicklungsumgebung

### 12.1 Software

- Betriebssysteme
  - Windows 7/8.1/10
  - Antergos
  - MAC OS X 10.11
- Modellierung
  - BOUML
- Entwicklung
  - Eclipse Neon (4.6.1)
  - Die JVM-Entwicklerumgebung 1.8.
  - HTML
  - CSS
- Versionsverwaltung
  - Git
- Continuous Integration
  - Jenkins CI 2.4
- Sonstige
  - draw.io
  - Gimp 2.x
  - L<sup>A</sup>T<sub>E</sub>X

– Pencil

## 12.2 Hardware

- Standard-PCs mit echten und virtualisierten Intel und AMD CPUs.

## Glossar

Atomarer Vorgang	Ein Vorgang, bei dem sichergestellt ist, der von keinem anderen Vorgang beeinflusst werden kann.
Browser	Programme die zum Aufrufen von Webseiten benötigt werden. Je nach Browser unterscheiden sich Anzeige und Unterstützung für bestimmte Tools.
Constraint	Einschränkungen gegeben durch Auflagen und Studienfach.
CSS	<b>Cascading Style Sheets</b> ist eine Stylesheet-Sprache, die für HTML verwendet wird und von der W3C weiterentwickelt wird. Stylesheet-Sprachen legen das Erscheinungsbild von Dokumenten/Benutzeroberflächen fest.
Datenbank	Ein System zu elektronischer Datenverwaltung, in diesem Fall auf Basis von SQL.
draw.io	<a href="https://www.draw.io/">https://www.draw.io/</a>
ECTS	Credits die für den Abschluss von Modulen angerechnet werden.
Gimp	<a href="https://www.gimp.org/">https://www.gimp.org/</a>
GUI	Abkürzung: Graphical user interface. Bezeichnet eine Form der Benutzerschnittstelle zwischen Benutzer und Computer. Sie hat die Aufgabe, Anwendungssoftware auf einem Rechner mittels grafischer Symbole bedienbar zu machen.
HTML	<b>Hypertext Markup Language</b> ist eine Auszeichnungssprache zur Strukturierung von digitalen Dokumenten. HTML Dokumente werden von Webbrowsern dargestellt.
JavaScript	JavaScript ist eine Skriptsprache für dynamische Interaktionen mit HTML Dokumenten.

Jenkins CI	Eine Continuous Integration System, das automatisch Softwareprojekte baut, testet und in einer Testumgebung startet.
JVM	Die Java Virtual Machine, abgekürzt JVM ist die Virtuelle Maschine, die als Laufzeitumgebung für Java-Programme dient.
Menü-Hierarchie	Ein Menü, welches selber eigene Menüs verfügt. Solche Anordnungen sind hierarchisch.
Moduldatei	Eine Datei die einem Modul zugeordnet ist, wie z.B. Modul-Vorraussetzungen, Semester, erreichbare ECTS, etc.
Modulhandbuch	Eine Übersicht über die angebotenen Module, sowie deren Constraints. Diese Übersicht soll als Datenbank dargestellt werden.
Orientierungsprüfung	Module die bis zu einem bestimmten Semester versucht bzw. bestanden sein muss.
Pencil	<a href="http://pencil.evolus.vn/">http://pencil.evolus.vn/</a>
Pflichtmodul	Module die zum erfolgreichen Abschluss des Studiums benötigt sind.
properties	Sprachdateien werden in .properties Dateien gespeichert. In diesen werden alle Texte in den unterschiedlichen Sprachen angegeben.
Servlet	Eine Webserverumgebung, die es erlaubt, mit Java-Klassen dynamisch webseiten zu erzeugen.
Wahlmodul	Module die vom Studenten gewählt werden.
Webseite	Ein virtueller Platz im World Wide Web, an dem sich meist mehrere Webseiten (Dateien) und andere Ressourcen befinden.
Workflow	Eine definierte Abfolge von Aktivitäten in einem Arbeitssystem.
Übungsschein	Zusatzbedingung für ein Modul. Je nach Modul freiwillig, verpflichtend oder als Bonus, sowie in Form einer Klausur möglich.