

Studienplanung als Generierung von Workflows mit Compliance-Anforderungen: Planerstellung und Visualisierung

Testbericht

Nada Chatti
Daniel Jungkind
Hannes Kuchelmeister
Ulrike Rheinheimer
Paul Samuel M. Teuber
Tim Niklas Uhl

15. März 2017

Inhaltsverzeichnis

1	Einleitung	3
2	Testfälle	3
2.1	Server	3
2.2	Client	7
2.3	Server-Coverage-Daten	9
2.4	Client-Coverage-Daten	10
3	Änderungen an Testszenarien aus dem Pflichtenheft	11
3.1	Geänderte Anwendungsfälle	11
3.2	Geänderte Testszenarien	13
4	Geänderte Produktfunktionen	15
5	Behobene Fehler	15
6	Nutzer-Feedback	21

1 Einleitung

In diesem Testbericht wird die Qualitätssicherungsphase des Projekts „Studienplanung als Generierung von Workflows mit Compliance-Anforderungen: Planerstellung und Visualisierung“ vorgestellt. Während dieser Phase wurde mithilfe von Testfällen und -szenarien aus dem Pflichtenheft die Qualität der Software überprüft und verbessert.

Zuerst wurden anhand der Anwendungsfälle des Pflichtenheftes Fehler im Produkt und Abweichungen von der Produktdefinition gesammelt und festgestellt. Anschließend wurden diese behoben, wobei aufgrund der geänderten Anforderungen auch Anpassungen an den Anwendungsfällen vorgenommen wurden. Zudem wurde mit weiteren Unit- und Integrationstests die Qualität der Software verbessert. Außerdem haben wir Nutzer-Feedback zur Bedienbarkeit gesammelt, welches wir der Nachwelt nicht vorenthalten möchten.

2 Testfälle

Es wurden weitere Unit- und Integrationstests hinzugefügt, mit welchen zusätzliche Produkt-Stabilität einhergeht. Hier erfolgt nun eine Auflistung aller Unit- und Integrationstests.

2.1 Server

Unit-Tests

Testklasse	Beschreibung	Status
CategoryTest	Getter und Setter für <code>Category</code> getestet	ERFOLGREICH
DisciplineTest	Getter und Setter für <code>Discipline</code> getestet	ERFOLGREICH

SemesterTest	<p>Test der Semester-Abstandsberechnung zwischen:</p> <ul style="list-style-type: none"> • zwei Wintersemestern • zwei Sommersemestern • zwischen Winter- und Sommersemester • zwischen Sommer- und Wintersemester <p>Test der compareTo-Methoden Test der plus-Methode</p>	ERFOLGREICH
StandardVerifierTest	<p>Testet, ob der StandardVerifier folgendes erkennt:</p> <ul style="list-style-type: none"> • fehlende Pflichtmodule • Verletzung von Rule-Group-Bedingungen • Verletzung von Bereichsbedingungen (Fields) 	ERFOLGREICH
StandardVerifierConstraintTest	<p>Testet, ob der die Verletzung folgender ConstraintTypes in unterschiedlichen Ausartungen erkennt:</p> <ul style="list-style-type: none"> • Verletzung von Überlappung • Verletzung von Plan-Zusammengehörigkeit • Verletzung von Semester-Zusammengehörigkeit • Verletzung von Voraussetzungen <p>Dadurch wurden auch die isValid-Methoden der entsprechenden ConstraintTypes getestet</p>	ERFOLGREICH

SimpleGeneratorTest	<p>Testet die generate-Methode des SimpleGenerator. Ein Plan mit einem einzelnen ModuleEntry wird:</p> <ul style="list-style-type: none"> • anhand der Benutzer-Präferenzen und Constraints verschiedener Art vervollständigt und modifiziert • anhand einer Zielfunktion optimiert <p>Dafür werden sog. Mock-Objekte benutzt, die nur zum Testen erstellt wurden, d.h. bei der echten Verwendung des Systems werden diese nicht benutzt, sondern stattdessen die Daten aus der Datenbank. Das Verhalten von Methoden der Klassen ModuleDao und Plan wurde hierfür mittels Mockito angepasst.</p>	ERFOLGREICH
NodesListTest	Testet die topologische Sortierung der Klasse NodesList.	ERFOLGREICH
VerificationManager	Getter für Verifier getestet.	ERFOLGREICH
ConditionTest	Factory-Methoden getestet	ERFOLGREICH
ContainsFilterDescriptorTest, ListFilterDescriptorTest, RangeFilterDescriptorTest, FilterDescriptorTest	Korrekte JSON-Serialisierung, Parsen der Filterdaten aus einer Anfrage getestet; dabei auch ungültige Eingaben überprüft	ERFOLGREICH
FilterDescriptorProviderTest	Invarianten des Providers getestet (untersch. URI-Identifizier u. IDs, jeder veröffentlichte Filter kann mit dem Identifizier gefunden werden)	ERFOLGREICH
MultiFilterTest	Korrekte Listenglättung der Sub-Condition-Listen getestet	ERFOLGREICH
TrueFilterTest	Rückgabe einer leeren Condition-Liste getestet	ERFOLGREICH

Integrations-Tests

Testklasse	Beschreibung	Status
DisciplinesResourceIntegrationTest	Überprüft die korrekte JSON-Serialisierung von GET /disciplines.	ERFOLGREICH
FieldsResourceIntegrationTest	Überprüft die korrekte JSON-Serialisierung von GET /fields.	ERFOLGREICH
ObjectiveFunctionResourceIntegrationTest	Überprüft die korrekte JSON-Serialisierung von GET /objective-functions.	ERFOLGREICH
FilterResourceIntegrationTest	Überprüft die korrekte JSON-Serialisierung von GET /filters (die drei Typen range, list und contains samt zugehöriger Spezifikation).	ERFOLGREICH
StudentResourceIntegrationTest	Überprüft, ob der Nutzer beim Neuanlegen korrekt initialisiert ist, ob das Setzen von Fachrichtung und Studienstart funktioniert, ob fehlerhafte Fachrichtungen/Studienstarts (zukünftig/prähistorisch) abgefangen werden, ob bestandene Module korrekt gesetzt werden, ob fehlerhafte bestandene Module (falsches Modul, falsches Semester) abgefangen werden, ob die JSON-Rückgaben korrekt serialisiert werden und ob eine Autorisierung vor dem Zugriff auf die Ressourcen verlangt wird.	ERFOLGREICH
ModuleResourceIntegrationTest	Überprüft, ob GET /modules/id korrekt als JSON serialisiert wird und auch die korrekten Daten gesendet werden; testet mehrere Filteraufrufe auf korrekte Rückgabe sowie fehlerhafte Eingabe.	ERFOLGREICH
PlansResourcePlanCreationIntegrationTest	Folgendes wurde überprüft: <ul style="list-style-type: none"> • Anlegen eines Plans möglich • korrekte Serialisierung aller Pläne eines Nutzers • doppelte Plannamen werden abgefangen • ungültige Plannamen werden abgefangen (leerer Name, zu langer Name) • Zugriff nur autorisiert möglich 	ERFOLGREICH

PlansResourcePlanEditIntegrationTest	<p>Folgendes wurde überprüft:</p> <ul style="list-style-type: none"> • Planumbenennung möglich; doppelte Namen, leere Namen, zu lange Namen werden abgefangen • korrekte Serialisierung eines Plans • Platzierung von Modulen im Studienplan • Bewegung von Modulen im Studienplan • Entfernen von Modulen aus dem Studienplan • Setzen und Entfernen von Modulpräferenzen • Zugriff auf die Exportfunktion • Zugriff auf Ressourcen nur autorisiert möglich <p>Dies beinhalten auch das Überprüfen ungültiger Eingabeparameter.</p>	ERFOLGREICH
--------------------------------------	--	-------------

2.2 Client

Unit-Tests

Testfall	Beschreibung	Status
ModuleCollection-Initialisierung	Wird eine ModuleCollection für gegebene Daten erfolgreich initialisiert?	ERFOLGREICH
ModuleConstraint-Initialisierung	Wird ein ModuleConstraint für gegebene Daten erfolgreich initialisiert?	ERFOLGREICH
Module-Initialisierung	Wird ein Modul für gegebene Daten erfolgreich initialisiert?	ERFOLGREICH
Preference-Initialisierung	Wird eine Preference für gegebene Daten erfolgreich initialisiert?	ERFOLGREICH
Plan-Initialisierung	Wird ein Plan für gegebene Daten erfolgreich initialisiert?	ERFOLGREICH
Discipline-Initialisierung	Wird eine Discipline für gegebene Daten erfolgreich initialisiert?	ERFOLGREICH

Filter-Initialisierung	Wird ein Filter für gegebene Daten erfolgreich initialisiert?	ERFOLGREICH
LanguageManager-Funktion	Gibt der Language Manager die richtigen Textbausteine zurück?	ERFOLGREICH
Notification-Collection-Initialisierung	Gibt getInstance() das richtige Objekt zurück?	ERFOLGREICH
ObjectiveFunctionCollection-Initialisierung	Wird eine ObjectiveFunctionCollection für gegebene Daten erfolgreich initialisiert?	ERFOLGREICH
TemplateManager-Funktion	Gibt der TemplateManager die richtige HTML-Ausgabe zurück?	ERFOLGREICH
CookieSync-Funktion	Speichert/Lädt CookieSync ein Modell erfolgreich?	ERFOLGREICH
OAuthSync-Funktion	Übergibt OAuthSync beim Speichern/Laden die richtigen Header?	ERFOLGREICH
MainView-Funktion	Lädt die MainView die richtigen Views und zeigt diese an?	ERFOLGREICH

Integrations-Tests

Testfall	Beschreibung	Status
DisciplineCollection	Test der Kommunikation zwischen DisciplineCollection und Server-Mock /disciplines	ERFOLGREICH
FieldCollection	Test der Kommunikation zwischen FieldCollection und Server-Mock /fields	ERFOLGREICH
FilterCollection	Test der Kommunikation zwischen FilterCollection und Server-Mock /filters	ERFOLGREICH
Module	Test der Kommunikation zwischen Module und Server-Mock /plans/{id}/modules/{id}	ERFOLGREICH
ModuleCollection	Test der Kommunikation zwischen ModuleCollection und Server-Mock /plans/{id}/modules	ERFOLGREICH

ObjectiveFunctionCollection	Test der Kommunikation zwischen ObjectiveFunctionCollection und Server-Mock /objective-functions	ERFOLGREICH
Plan	Test der Kommunikation zwischen Plan und Server-Mock /plans/{id}	ERFOLGREICH
PlanCollection	Test der Kommunikation zwischen PlanCollection und Server-Mock /plans	ERFOLGREICH
ProposedPlan	Test der Kommunikation zwischen ProposedPlan und Server-Mock /plans/{id}/proposal/{objective-function}	ERFOLGREICH
Student	Test der Kommunikation zwischen Student und Server-Mock /student	ERFOLGREICH

2.3 Server-Coverage-Daten

Im Folgenden sind Coverage-Statistiken zur Server-Überdeckung aufgeführt.

Unit-Tests Für das rest-Package wurden keine Unit-Tests verfasst, weshalb es hier auch nicht aufgeführt ist. Die REST-Schnittstellen werden sinnvollerweise in den Integrationstests abgedeckt. Da der Plugin-Manager seine Funktion weitgehend verloren hat, wurden nur die für die Nutzung relevanten Bestandteile getestet. Ebenso wurde auf das Testen einiger primitiver Getter und Setter im Modell verzichtet.

Package	Zeilenüberdeckung
filter	81 %
generation	67 %
model	36 %
pluginmanager	14 %
verification	85 %
Gesamt	57 %

Tabelle 5: Unit-Test-Überdeckung Server

Integrations-Tests Aufgrund von Schwierigkeiten beim Zusammenspiel von JaCoCo und unserer Tomcat-Serverumgebung sind für Integrationstests keine Coverage-Daten verfügbar. Als grober Überblick diene stattdessen die manuell bestimmte Methodenüberdeckung für die einzelnen REST-Ressourcen. Auf Integrationstests für Verifizierung und Generierung wurde verzichtet, da diese schon ausreichend mit Unit-Tests abgedeckt sind.

REST-Ressource	Methodenüberdeckung
/disciplines	100 %
/fields	100 %
/filters	100 %
/modules	100 %
/objective-functions	100 %
/plans	72 %
/student	100 %
Gesamt	96 %

Tabelle 6: Integrations-Test-Überdeckung Server

2.4 Client-Coverage-Daten

Die beim Client gemessene Testüberdeckung beträgt 44,98%. Leider ist dieser Wert aus mehreren Gründen ungenau.

Testen der Nutzeroberfläche Neben den vorhandenen Unit- und Integrations-Tests, welche von dieser Metrik erfasst werden, wurden auch eine Vielzahl von manuellen und halb-automatisierten Nutzeroberflächen-Tests durchgeführt (teilweise manuell, teilweise mit Hilfe der Selenium-IDE). Diese Testfälle orientierten sich insbesondere an den im Pflichtenheft festgelegten globalen Testfällen. Aus technischen Gründen können diese Tests jedoch nicht in die Testüberdeckung miteingerechnet werden.

Dennoch waren insbesondere diese Tests sehr effektiv beim Aufdecken von Fehlern, da durch das Bedienen der Nutzeroberfläche die Kommunikation und das Zusammenspiel der verschiedenen Subsysteme exzellent überprüft werden konnte. Die hier genannten Testmethoden haben uns dabei geholfen, viele Fehler in der Software zu finden und zu beseitigen.

Daten im Code Aus Performance-Gründen hatten wir uns bereits in der Entwurfsphase dafür entschieden, verschiedene Daten wie Templates und Sprachdaten zur Compilezeit in den Code miteinzubauen. Hierdurch wurden die notwendigen Serveranfragen beim Ausführen des Clients im Webbrowser deutlich reduziert. Dies wirkt sich sehr positiv auf die Geschwindigkeit der Anwendung aus, da zur Laufzeit lediglich die Daten der REST-Schnittstelle geladen werden müssen. Gleichzeitig führt diese Entscheidung allerdings zu Ungenauigkeiten bei der Messung der Testüberdeckung, da diese Daten ebenfalls als „Codezeilen“ aufgefasst werden.

3 Änderungen an Testszenarien aus dem Pflichtenheft

3.1 Geänderte Anwendungsfälle

Durch die während der Entwicklung geänderten Anforderungen erfordern manche Anwendungsfälle zusätzliche Anpassungen, welche hier aufgelistet sind. Gestrichen wurden Anwendungsfälle, deren zugrundeliegende Produktfunktionalität entfernt wurde.

Anwendungsfall	Änderung
A10: Erstanmeldung / A20: Login	Auf Einbindung des Shibboleth Identity Providers wurde (wie angekündigt) verzichtet.
A40: Profil bearbeiten	Eingabe von Studienbeginn/Studiengang nur bei Erstanmeldung möglich; „Konto speichern“ kehrt nicht zur letzten offenen Seite zurück. Der Nutzer kann außerdem sein Konto löschen.
A50: Neuen Studienplan anlegen	Das Namens-Popup hat keine Voreinstellung.
A70: Studienplan duplizieren	Es wird nach einem neuen Namen für den Plan gefragt. (Benutzerfreundlicher)
A85: Studienplan exportieren	Es wird eine HTML-Zusammenfassung des Studienplans im Browser angezeigt.
A90: Mehrere Studienpläne löschen	Duplizieren und Teilen wurden entfernt und das Aktions-Wahlfeld durch eine einzige Schaltfläche ersetzt.
A100: Studienpläne vergleichen	Entfernt.
A110: Module filtern	Filtern nach Fachrichtung und Vorkommen im Plan entfernt; Filtern nach Bereich hinzugefügt.

A130: Modulinfo-Leiste	Möglichkeit zur Bewertung entfernt; Modul-Constraints werden nun angezeigt.
A140: Modul einfügen	Module können auch in abgeschlossene Semester eingefügt werden.
A170: Änderung rückgängig machen	Entfernt.
A210: Semesterzeile löschen	Da Zeilen nicht explizit gespeichert werden und leere Zeilen gar nicht: Entfernt.
A215: Abgeschlossene Semester ein-/ausblenden	Entfernt.
A220: Studienplan überprüfen	Es gibt keine Tooltips beim Drüberfahren mit der Maus, stattdessen werden die Verifizierungsergebnisse gesammelt als Popup angezeigt.

Testergebnisse

Anwendungsfall	Status
A10: Erstanmeldung	ERFOLGREICH
A20: Login	ERFOLGREICH
A30: Logout	ERFOLGREICH
A40: Profil bearbeiten	ERFOLGREICH
A50: Neuen Studienplan anlegen	ERFOLGREICH
A55: Studienplan anzeigen / A57: schließen	ERFOLGREICH
A60: Plan umbenennen	ERFOLGREICH
A70: Plan duplizieren	ERFOLGREICH
A80: Plan löschen	ERFOLGREICH
A85: Plan exportieren	ERFOLGREICH
A90: Mehrere Studienpläne löschen	ERFOLGREICH
A110: Module filtern	ERFOLGREICH
A130: Modulinfo anzeigen	ERFOLGREICH
A140: Modul einfügen	ERFOLGREICH
A150: Modul löschen	ERFOLGREICH
A160: Modul verschieben	ERFOLGREICH
A180/190: Modul positiv/negativ bewerten	ERFOLGREICH

A200: Semesterzeile einfügen	ERFOLGREICH
A220: Plan überprüfen	ERFOLGREICH
A230: Plan vervollständigen	ERFOLGREICH

3.2 Geänderte Testszenarien

Im Folgenden eine Auflistung der konsequenterweise ebenfalls angepassten Testszenarien.

T1: Erststart mit „halbherziger Bedienung“

1. A10: Erstanmeldung (ohne Angabe bereits bestandener Module)
2. A50: Neuen Studienplan anlegen
3. A220: Studienplan auf Korrektheit überprüfen – Ergebnis „fehlerhaft“ (da unvollständig)
4. A230: Studienplan vervollständigen lassen mit anschließendem Verwerfen
5. A57: Schließen der Studienplan-Ansicht
6. A80: Studienplan löschen
7. A30: Logout

T2: Einfache Vervollständigung

1. A20: Login
2. A40: Profil bearbeiten – erste zwei Semester anschließend belegt
3. A50: Neuen Studienplan anlegen
4. A230: Studienplan vervollständigen lassen mit anschließendem Übernehmen des Studienplans
5. A60: Studienplan umbenennen
6. A57: Schließen der Studienplan-Ansicht
7. A85: Studienplan exportieren
8. A30: Logout

T3: Bearbeitung eines Studienplans

Nutzer ist bereits eingeloggt und hat mind. einen Studienplan angelegt.

1. A55: Studienplan anzeigen
2. A110: Module filtern
3. A130: Modulinfo anzeigen und wieder schließen
4. A140: Modul einfügen
5. A150: Modul löschen
6. A160: Modul verschieben

7. A160: Modul negativ bewerten
8. A180: Modul positiv bewerten (selbes Modul – es ist dann positiv bewertet)
9. A140: Modul einfügen
10. A57: Schließen der Studienplan-Ansicht

T4: Profil bearbeiten

Nutzer ist bereits eingeloggt und hat mind. einen Studienplan angelegt.

1. A55: Studienplan anzeigen
2. A40: Profil bearbeiten – dabei Änderung der Semester-Belegung
3. Anschließend sollte die Änderung im Studienplan gezeigt werden.

T5: Vervollständigung mit mehreren Alternativen

1. A20: Login
2. A50: Neuen Studienplan anlegen
3. A110: Module filtern
4. A140: Modul einfügen
5. Schritte 3–4 mehrmals wiederholen, sodass Abhängigkeitsfehler vorhanden sind und der Studienplan noch unvollständig ist
6. A220: Studienplan auf Korrektheit überprüfen: Es werden Abhängigkeitsfehler gemeldet
7. Mittels „A150: Modul löschen“ und „A160: Modul verschieben“ Abhängigkeitsfehler beheben
8. A220: Studienplan auf Korrektheit überprüfen: Der Studienplan ist unvollständig
9. A230: Studienplan vervollständigen lassen mit anschließendem Speichern unter neuem Namen
10. A57: Schließen der Studienplan-Ansicht
11. A55: Studienplan anzeigen (den in Schritt 2 erstellten Studienplan)
12. A230: Studienplan vervollständigen lassen mit anderen Zielkriterien als in Schritt 9 und anschließendem Speichern unter neuem Namen
13. A57: Schließen der Studienplan-Ansicht

T6: Studienpläne duplizieren und löschen

1. A20: Login
2. A50: Neuen Studienplan anlegen
3. A57: Schließen der Studienplan-Ansicht
4. A70: Studienplan duplizieren
5. Mehrmaliges Wiederholen von Schritt 4.
6. A90: Mehrere Studienpläne löschen: Alle Studienpläne löschen.

T7: Semester-Zeilen anpassen

Entfällt.

Testergebnisse

Testszenario	Status
T1: Erststart mit „halbherziger Bedienung“	ERFOLGREICH
T2: Einfache Vervollständigung	ERFOLGREICH
T3: Bearbeitung eines Studienplans	ERFOLGREICH
T4: Profil bearbeiten	ERFOLGREICH
T5: Vervollständigung mit mehreren Alternativen	ERFOLGREICH
T6: Studienpläne duplizieren und löschen	ERFOLGREICH

4 Geänderte Produktfunktionen

Es folgt eine Auflistung der hinzugefügten, geänderten und entfernten Produktfunktionen.

Hinzugefügt:

- Bereichs-Filter in der Suchleiste
- Hilfe-Funktion: Bietet dem Nutzer Hilfestellungen zur Bedienung und erläutert die Funktionen

Geändert:

- Zielfunktionen geändert: Es gibt jetzt nur noch „minimale ECTS“, „minimale Semesterzahl“ und „Präferenzen berücksichtigen“ als Zielfunktionen. Gleichmäßige ECTS-Verteilung wird unabhängig von der Zielfunktion immer berücksichtigt.

5 Behobene Fehler

Es folgt eine Auflistung aller behobenen Produktmängel. (Bei mit „—“ markierten Spalten erschien die Nennung von Ursache/Behebung überflüssig.)

Fehler	Ursache	Behebung
expires_in-Daten werden nicht in SessionInformation gespeichert/verarbeitet	—	—
Modulinfo-Leiste zeigt nichts an und lässt sich nicht schließen.	Schließen: Ein Event war nicht richtig registriert; Keine Anzeige: Falsche Übergabe der Daten an Template	—
Module können in jedem beliebigen Turnus abgelegt werden	Client/Server validieren die Eingabe nicht	Validierung, ob das gewählte Semester mit dem Modul-Turnus kompatibel ist
Nutzer wird nach Löschen des Kontos nicht ausgeloggt	Fehlende/Fehlerhafte URI-Weiterleitung	—
Beim Erstlogin kann ein Semester aus der Zukunft erstellt werden	Fehler beim dynamischen Erstellen der Auswahlmöglichkeiten auf Basis des aktuellen Datums	Erstlogin-Wizard zeigt keine Semester aus der Zukunft mehr an
Modulsuchleisten-Scrollbar wird nach Filter-Anwendung zurückgesetzt	Der Lade-Bildschirm führte automatisch zu einem neu-rendern der Seite	—
Server akzeptiert Modulbelegungen in nicht-positive Semester	Fehlende Validierung der Eingabe	Eingabevalidierung ergänzt
Der Kategorie-Filter kann nicht verändert werden	Bei Neu-Rendern des Felds wurde das aktuell ausgewählte Feld nicht berücksichtigt	Aktuell ausgewähltes Feld bei Rendering als ausgewählt setzen
Bestandene Module erscheinen nach Verifizierung fehlerhaft, sollten aber ignoriert werden	SemesterLinkConstraint#isValid() berücksichtigte abgeschlossene Module nicht	Wenn beide Module im Plan vorhanden und mind. eines von beiden bestanden ist, wird true zurückgegeben
Keine Zielfunktionen im Generierungs-Wizard	FunctionDto-Klasse war nicht statisch und konnte daher nicht serialisiert werden	+ static
Client rendert Ansicht zu häufig	—	Render-Aufrufe überprüft
Server akzeptiert Erstellung eines namenlosen Plans	Fehlende Eingabevalidierung	Validierung hinzugefügt

Plan-Duplizierung schlägt fehl	Bei Clientanfrage fehlt Modulliste, dem Server fehlt Validierung	Modulliste wird mitgesendet; Aufbereitung der Eingabe zur Speicherung hinzugefügt
Planansicht verzählt sich bei Semesternummer	Fehlende „< 0“-Abfrage, unerwartetes Verhalten bei der Längenmessung von JS-Arrays	—
Beim Neuladen der Seitenleiste wird nicht nur dort ein Ladesymbol angezeigt	—	Neues Ladesymbol erstellt und verdrahtet
Client zeigt zu viele Benachrichtigungen an	Benachrichtigungen werden pro Anfrage ausgegeben. Deshalb werden bei mehreren (ähnlichen) aufeinanderfolgenden Anfragen die Benachrichtigungen mehrfach angezeigt.	Sperre: Jede Benachrichtigung darf nicht häufiger als alle zwei Sekunden gezeigt werden.
Filter sind nach einmaligem Setzen nicht mehr änderbar	Fehlende Ausnutzung der JS-Typumwandlung bei Vergleichen, fehlerhafter Bezeichner	—
Ändern von Modulpräferenzen bleibt wirkungslos	Server ändert vorhandene Präferenzen nicht	—
Verifizierungs-Ergebnis-Popup kann nur einmal verschoben werden	Das Drop-Event einer Semesterzeile wurde getriggert, daraufhin saß das Popup fest	—
Verifizierungs-Ergebnis-Popup hat einen Schließen-Button ohne Symbol	Fehlende Einbindung eines jQuery-UI-Icon-Sets	—
Fehlermeldungen haben wenig Aussagekraft	—	verfassen von genaueren, situationsspezifischen Fehlermeldungen
Nach Klick auf „Konto speichern“ friert der Client ein	Fehlendes Aufheben des Lade-Bildschirms	—
Client sendet beim Profil-Speichern nur neu hinzugefügte Module an den Server	Spezifikation der REST-Schnittstelle wurde nicht berücksichtigt	—

„Plan löschen“ hatte keine Sicherheitsabfrage	—	Hinzugefügt.
Modulpräferenzen können nicht im Client zurückgesetzt werden	Fehlerhafter Eventhandler	—
Modul-Constraints werden in der Seitenleiste mehrfach angezeigt	Fehlendes Zurücksetzen der Liste der Modul-Constraints im View (hierdurch wird das Modul beim Rendern wieder und wieder hinzugefügt)	—
Verifizierungs-Ergebnis-Popup nicht immer im Vordergrund (überdeckbar)	Fehlerhafte z-Indizes	—
Wahlfilter-Dropdown hatte zu geringe Breite, Zahlenfeld zu breit	—	—
Suche nach spezifischem Turnus liefert keine Module, die in beiden angeboten werden	Server filtert nur nach gegebenem Turnus (BOTH ≠ WINTER_TERM)	Sonderfall im Condition-QueryConverter eingefügt
Moduleinfügen löst keinen sichtbaren Verlust des Verifikationsstatus aus	Fehlendes UI-Refreshing	—
Modulinfo: Constraints werden mit falscher Richtung angezeigt	Fehlende Weiterverarbeitung der vom Server gelieferten Constraints	—
Modulboxen nicht vollständig anklickbar	—	—
Fehlende Semesterbegrenzung nach oben	—	Festgelegt auf max. 200
Durch Verifizierung ehemals fehlerhaft markierte Module verlieren Status erst nach Neuladen und erneutem Verifizieren	Fehlerhaftes Event- und Rendering-Triggern	—
Modulinfo in der Profilan-sicht/Generierungsergebnisansicht nicht aufrufbar	Aufrufen von plans/Plan-ID/modules mit Plan-ID = undefined	Fallback eingebaut

Generierung funktioniert nicht	—	—
Generierungsergebnis wird nicht angezeigt	—	—
Generierungsergebnis-Buttons sind ohne Effekt	Kleiner Fehler	—
Generierung verwendet nur max-semester-ects-Parameter	—	Beachtung weiterer Parameter eingebaut
Verifizierung überprüft Turnus nicht	—	—
Nach Generierung Wechsel zum Hauptmenü nicht möglich	Fehlerhafter Event-Handler	—
Bei Duplizierung eines Plans wird erst nach Neuladen korrekte ECTS-Summe angezeigt	Serialisierungsfehler, falscher Inhalt wurde gesendet	—
Nach dem Umbenennen eines Plans verschwinden alle platzierten Module	Server serialisiert ein leeres modules-Attribut	Serialisierung des modules-Attributs entfernt.
Präferieren von Modulen erst nach zweimaligem Klick möglich	Fehlender null-Check	—
Übernommenes Generierungsergebnis hat keinen Verifikationsstatus	PUT /plans lässt kein clientseitiges Setzen des Status zu	Aufruf von GET /plans/id/verify nach Speichern des Ergebnisses
Wintersemester werden mit nur einer Jahreszahl dargestellt	—	Wintersemester werden – wie gewohnt – mit „WS XXXX/YYYY“ dargestellt
Präferenzen/Bestandene Module löschen zeigt nur temporäre Wirkung	Server vergisst gelöschte Modulpräferenzen/bestandene Module (Hibernate-Problem)	In den Hibernate-Annotationen zu den Planpräferenzen/bestandenenen Modulen orphanRemoval = true gesetzt

In Generierungs-Ergebnis-Ansicht können Module verschoben werden, Semesterzeilen hinzugefügt werden etc.	Fehlende Sperre	—
Plan-Duplizierung lässt Client einfrieren	Parsing-Probleme, welche verhindern, dass nach der GET-Anfrage keine POST- und PUT-Anfrage gesendet werden	—
Bei mehrfachem Klick auf „Überprüfen“ erscheint das Ergebnisfenster mehrfach	—	—
In Modulinfo-Leiste werden Constraint-Module viel zu groß angezeigt	fehlendes CSS (u.a. fehlender max-height-Parameter)	—
Buttons der Generierungs-Sidebar nach Anzeigen von Modulinformationen nicht mehr verwendbar	Fehlerhaftes Setzen der Events	—
Cycle-Type-Sperre darf in Profilsicht nicht gesetzt sein	Eingestellt in Semesterklasse, die sowohl bei Planbearbeitung als auch in Profil verwendet wird	Behandlung der einzelnen Fälle
„Phantom-Plan“ bei Duplikationsversuch unter gleichem Namen	Fehlendes Rollback nach Plan-Erstellung vor Plan-Ersetzung	—
Sicherheitsabfrage beim Plan-Verwerfen fehlt	—	—
Es lassen sich keine Informationen über Constraint-Modul in Info-Sidebar laden	Fehlende Ressourcen-URL im Constraint-Modul	—
Bei Modul-Bewertung/Modul-Detail-Anzeige werden Verifikations-Informationen zurückgesetzt	Verifikations-Informationen werden bei jeder Planänderung zurückgesetzt	Verifikations-Informationen werden nur noch beim Hinzufügen und Entfernen von Modulen sowie Ändern des Modul-Semesters zurückgesetzt.

Der Button „Filter zurücksetzen“ funktioniert nur bei einmaliger Benutzung	Referenzen wurden übergeben, jedoch keine Kopie erzeugt	Erzeugen einer Kopie
Diverse Rechtschreibfehler in den UI-Texten	—	—

6 Nutzer-Feedback

Während der Qualitätssicherungsphase wurden einige externe Nutzer von uns zur Bedienbarkeit der Software befragt. Die hierbei gesammelten Verbesserungsvorschläge sind im Folgenden aufgelistet.

Vorschlag	Status
Bei gedrückter Maus visuelle Rückmeldung für den Nutzer, wohin er Module ziehen kann	ERLEDIGT
GUI-Elemente mit Tab-Taste navigierbar machen	ZURÜCKGESTELLT (zu aufwändig)
Beschreibungstext für Hauptansicht	ERLEDIGT
Verbesserte Fehlermeldung im Falle doppelter Modulplatzierung	ERLEDIGT
Aktive Filter sollten durch eine leichte KIT-farbene Blässe hervorgehoben werden	ZURÜCKGESTELLT (zu aufwändig)
Orientierungsprüfungen bei Überprüfung/Generierung berücksichtigen; Bachelor-Thesis erst zum Schluss erlauben	ZURÜCKGESTELLT (nicht in Datenbasis vorhanden, kann durch geeignete Constraints erreicht werden)
Hilfe-Funktion, die die Bestandteile der Anwendung erklärt	ERLEDIGT
Plannamen in Planliste anklickbar machen (daraufhin Anzeigen des Plans)	ERLEDIGT