

Bericht über die Qualitätsicherungsphase Crayons 2.0

Marc Jepsen, Natalia Krylova,
Levin Schickle, Ian Fitzgerald, Ali Akil, Julius Busch

20. März 2017

Inhaltsverzeichnis

1	Einleitung	4
2	Gefundene Bugs	5
2.1	Page Reload	5
2.1.1	Fehlersymptom	5
2.1.2	Behebung	5
2.2	Mehrere Kurse mit gleichem Titel	5
2.2.1	Fehlersymptom	5
2.2.2	Fehlergrund	5
2.2.3	Behebung	5
2.3	Benutzer hat mehrere Kurse mit den gleichen Namen	6
2.3.1	Fehlersymptom	6
2.3.2	Fehlergrund	6
2.3.3	Behebung	6
2.4	Bearbeiten des Kurses in CourseModificationWindow	6
2.4.1	Fehlersymptom	6
2.4.2	Fehlergrund	6
2.4.3	Behebung	6
2.5	Modifizieren des Kurstitels	6
2.5.1	Fehlersymptom	6
2.5.2	Fehlergrund	7
2.5.3	Behebung	7
2.6	Löschen des Kurses direkt nach dem Modifizieren	7
2.6.1	Fehlersymptom	7
2.6.2	Fehlergrund	7
2.6.3	Behebung	7
2.7	Automatisches Ändern der Tabelle in Benutzerreche-Ansicht	7
2.7.1	Fehlersymptom	7
2.7.2	Behebung	7
2.8	Sprache der Benutzeroberfläche	8
2.8.1	Fehlersymptom	8
2.8.2	Behebung	8
2.9	Graphrendering	8
2.9.1	Fehlersymptom	8
2.9.2	Behebung	8

2.10	Hinzufügen eines Knotens im Graphen	8
2.10.1	Fehlersymptom	8
2.10.2	Behebung	9
2.11	Speichern von Lerneinheiten in die Datenbank	9
2.11.1	Fehlersymptom	9
2.11.2	Behebung	9
2.12	Laden von Lerneinheiten in die Benutzeransicht	9
2.12.1	Fehlersymptom	9
2.12.2	Behebung	9
2.13	Ändern des Benutzernames	10
2.13.1	Fehlersymptom	10
2.13.2	Behebung	10
2.14	Sonderzeichen bei Vorname und Nachname	10
2.14.1	Fehlersymptom	10
2.14.2	Behebung	10
2.15	Connector	10
2.15.1	Fehlersymptom	10
2.15.2	Behebung	10
2.16	Naviagation nach Login	11
2.16.1	Fehlersymptom	11
2.16.2	Behebung	11
2.17	Ändern des Benutzerpassworts	11
2.17.1	Fehlersymptom	11
2.17.2	Behebung	11
3	UI Tests	12
3.1	RegistrationTest	12
3.2	LoginTest	12
3.3	AdminTest	12
3.4	CourseCreationTest	12
3.5	CourseAvailabilityTest	12
3.6	GraphInteractionTest	12
3.7	PreferenceTest	13
4	Login	14
5	Unittests	15

1 Einleitung

In der Qualitätsicherungsphase konnten die in der Implementierungsphase entstandenen Bugs erfolgreich behandelt werden. Zunächst wurde der schwerwiegendste und für die Entwicklung gravierendste Bug des Unit Ladens gefixt. Dies konnte mit Hilfe von einigen Umstellungen im Uniteditor und CK-Editor erreicht werden. Im Bereich Lessons learned sind vor Allem nützliche Erkenntnisse zur Konfiguration und Benutzung von Testbench zu finden, welches ein Vaadin Protool ist und zum Testen der UI verwendet wurde.

2 Gefundene Bugs

2.1 Page Reload

2.1.1 Fehlersymptom

Nach dem Reload einer Seite kommt es zu einem Logout und die Applikation wird neu gestartet. Nach einem Reload ist jedoch nur das neue Laden der Seite, nicht das Beenden der Applikation die zu erwartende Benutzerinteraktion.

2.1.2 Behebung

Mittels @PreserveOnRefresh, welches eine Vaadin Annotation ist konnte Abhilfe verschafft werden. Diese wurde über der Vaadin Hauptmethode platziert und hatte die gewünschte Funktionalität zur Folge.

2.2 Mehrere Kurse mit gleichem Titel

2.2.1 Fehlersymptom

Jedem Benutzer war es erlaubt, die Kurse mit den beliebigen (auch gleichen) Titeln zu erstellen. Es hat mehrere Problemen mit dem Erstellen, Modifizieren und Löschen von Kursen ausgelöst. Zum Beispiel, wenn einen Kurs gelöscht wurde, wurden alle Kurse mit dem Titel aus der Datenbank entfernt.

2.2.2 Fehlergrund

Der Kurs wurde nur nach dem Titel gesucht. Wenn es mehrere Kurse mit dem gleichen Namen gab, wurde der erste gefundene Kurs zurückgegeben.

2.2.3 Behebung

Zuerst wurde es einem Benutzer verboten, die Kurse mit den gleichen Titeln zu erstellen. CourseService und CourseDAO wurden erweitert, sodass die Kurse nach dem Titel und der E-Mail-Adresse des Autors gesucht werden können. Das hat verschiedenen Autoren ermöglicht, die Kurse mit den gleichen Titeln zu erstellen, zu modifizieren und zu löschen, ohne dass andere Kurse zu ändern.

2.3 Benutzer hat mehrere Kurse mit den gleichen Namen

2.3.1 Fehlersymptom

Wenn es einem Benutzer mehrere Kursen mit den gleichen Titeln zugeteilt wurde, und er/sie einen von diesen Kursen verlassen wollte, verließ er/sie alle Kursen mit diesem Titel.

2.3.2 Fehlergrund

Der Kurs wurde nur nach dem Titel gesucht. Wenn es mehrere Kurse mit dem gleichen Namen gab, wurde der erste gefundene Kurs zurückgegeben.

2.3.3 Behebung

Beim Erstellen der Tabs im Benutzerbibliothek wurde auch der zugehörige Kurs berücksichtigt. Im ClickListener für “Verlassen”-Button ins der Kurs nach dem Titel und der E-Mail-Adresse des Autors gesucht. Damit ist immer der richtige Kurs gefunden.

2.4 Bearbeiten des Kurses in CourseModificationWindow

2.4.1 Fehlersymptom

Die Änderungen, die in CourseModificationWindow gemacht wurden, wurden nicht gespeichert.

2.4.2 Fehlergrund

@Autowired-Annotation funktioniert nicht im components-Paket, deswegen hat CourseService nicht funktioniert.

2.4.3 Behebung

CourseService wurde beim Aufrufen von CourseModificationWindow aus der Autorenbibliothek übergeben.

2.5 Modifizieren des Kurstitels

2.5.1 Fehlersymptom

Es war nicht erlaubt, in CourseModificationWindow nur den Kurstitel zu modifizieren, ohne die Kursbeschreibung zu ändern.

2.5.2 Fehlergrund

Vorher wurde das Modifizieren eines Kurses nur dann erlaubt, wenn der Autor einen Kurs mit dem gleichen Titel nicht besitzt. Solche Vorgehensweise war nötig, um die Existenz der Kursen mit den gleichen Titeln zu vermeiden.

2.5.3 Behebung

Jetzt ist es zuerst überprüft, ob den Titel in CourseModificationWindow geändert wurde. Und nur wenn es der Fall ist, wird die Existenz der Kursen mit dem neuen Titel überprüft. Ansonsten wird nur die Beschreibung des Kurses geändert.

2.6 Löschen des Kurses direkt nach dem Modifizieren

2.6.1 Fehlersymptom

Wenn ein Kurs modifiziert wurde und danach gelöscht werden soll, gibt es eine Null Pointer Exception.

2.6.2 Fehlergrund

Es wurde in der Datenbank nach dem Kurs gesucht, der einen Titel hat, der beim Erstellen des Tabs benutzt wurde (=der alte Titel).

2.6.3 Behebung

Der neue Titel kann aus dem Tab abgelesen werden.

2.7 Automatisches Ändern der Tabelle in Benutzerrecherche-Ansicht

2.7.1 Fehlersymptom

Nach dem Klicken auf die Tabelle mit den Benutzerdaten wurde die Reihenfolge der Zeilen und Spalten verändert.

2.7.2 Behebung

Bei jedem Ändern der Benutzerdaten wird zuerst die Methode `setVisibleColumns()` angewendet, um die Reihenfolge der Spalten zu speichern. Danach wird die Information in der Tabelle aktualisiert und die Methode `setVisibleColumns()`

benutzt, um die Spalten in die richtige Reihenfolge zu bringen. Schließlich werden die Zeilen nach der E-Mail-Adresse (standardmäßig) oder nach dem vom Benutzer gewählten Parameter sortiert.

2.8 Sprache der Benutzeroberfläche

2.8.1 Fehlersymptom

Nachdem ein Benutzer sich einloggt, wird die Benutzeroberfläche immer auf Deutsch gezeigt, sogar wenn der Benutzer in den Einstellungen “English” ausgewählt hat.

2.8.2 Behebung

Zuerst werden die Benutzerdaten überprüft. Wenn der Benutzername und das Passwort richtig sind, wird die Sprache der Benutzeroberfläche gewählt. Erst danach wird der Benutzer weitergeleitet.

2.9 Graphrendering

2.9.1 Fehlersymptom

Das Graphrendering ist bei der Hälfte der Höhe abgeschnitten, nachdem mehr als zwei Noten hinzugefügt worden sind

2.9.2 Behebung

Da das Verhalten am von Vaadin verwendeten Javascriptconnector liegt und hier leider keine Exceptions geworfen wurden oder Errors auf der JavaScript Konsole erschienen, konnte der Bug an sich nicht behoben werden. Auch ausführliche Recherche zu dem Thema brachte keinen Aufschluss. Indem bei kritischen Bearbeitungsschritten, welche das ungewünschte Verhalten zur Folge haben konnte die Seite neu geladen wird, ist das Problem umgangen worden.

2.10 Hinzufügen eines Knotens im Graphen

2.10.1 Fehlersymptom

Sofern neue Knoten im Grapheneditor hinzugefügt worden sind hatte dies zunächst keine Probleme zur Folge. Jedoch kam es zu einem kleinen Durcheinander sofern Kanten im UnitConnection Editor bearbeitet worden sind. Dies lag

daran, dass die addUnit Methode im Graphenmodell nur den Vorgänger aktualisiert hat und somit der beim Nachfolger der neu hinzugefügte Knoten nicht als Elternknoten abgespeichert wurde.

2.10.2 Behebung

Der methode addUnit wurde der Nachfolgerknoten als weiterer Parameter hinzugefügt.

2.11 Speichern von Lerneinheiten in die Datenbank

2.11.1 Fehlersymptom

Das Speichern von Lerneinheiten war auf Grund von Zahlreichen Hürden erschwert und hat leeren Datenbankeinträgen, NullPointerExceptions und Serialisierungsfehlern aufgewiesen.

2.11.2 Behebung

Die Einheiten bestehen aus einzelnen Elementen, welche aus der Einheit gezogen, identifiziert und in ihre Bestandteile zerlegt werden müssen. Diese Teile müssen ihren Inhalt verfügbar machen (-> Komplikationen mit z.B. Vaadin-Image und Vaadin-OptionGroup) um zu jeder Zeit ihren Zustand exportieren zu können.

2.12 Laden von Lerneinheiten in die Benutzeransicht

2.12.1 Fehlersymptom

Es traten Fehler auf beim Blockieren der Components in der Benutzeransicht, verursacht durch den Textbaustein.

2.12.2 Behebung

Bereits beim Erstellen der Seitenkomponenten muss der Editor aussenvor gelassen werden, anstatt später die sonst nützliche Funktion 'readOnly' von Vaadin-components zu verwenden.

2.13 Ändern des Benutzernames

2.13.1 Fehlersymptom

Der Benutzer war in der Lage sein Benutzername, bzw seine Email, zu ändern.

2.13.2 Behebung

Die Funktion ist weiterhin vorhanden. Jedoch wurde im UI das Feld, in dem die Email bzw der Benutzername angezeigt werden, zu readOnly gesetzt damit der Benutzer keine Eingabe tätigen kann für dieses Feld.

2.14 Sonderzeichen bei Vorname und Nachname

2.14.1 Fehlersymptom

Bei der Registrierung war es möglich, dass der Vor- bzw Nachname Sonderzeichen enthalten, was nicht wünschenswert ist

2.14.2 Behebung

Es wurde einen regulären Ausdruck erstellt, der den fall abdeckt.

2.15 Connector

2.15.1 Fehlersymptom

nachdem ein User sich eingeloggt hat einen Graph bearbeitet und sich danach wieder einloggt und diesen wiederum bearbeiten will oder sich danach ein beliebiger Nutzer in der gleichen JVM einloggt kommt es zur einer Exception `connector with id xy already registered`"

2.15.2 Behebung

Zunächst wurde der Haupt UI Klasse ein Connector Tracker hinzugefügt um logging beim consolenoutput zu ermöglichen und den Ort des Problems zu lokalisieren. Leider brachte dies keinen großen Aufschluss. Jedoch ist durch ausprobieren rausgefunden worden, dass das Problem am Javascript Connector liegt und, dass nach dem erneuten Einloggen die alten Komponenten von der alten Ui gestohlen werden. Das Problem lag schlussendlich am Connector Tracker von Vaadin, welcher anders implementiert ist als die restlichen Komponenten

und mit @Preserveon Refresh nicht funktioniert. Allerdings wurde das in Vaadin 8 anscheinend gefixt, so wie es vaadin.com zu entnehmen ist. Allerdings ist nachdem Versuch die Applikation auf vaadin 8 umzustellen ein Error beim Import des AbstractJavaScriptComponent zu Stande kommen. Die Ursache dieses Bugs und die Behebung konnte schlussendlich aus Zeitgründen und zu großer Komplexität nicht behoben werden.

2.16 Navigation nach Login

2.16.1 Fehlersymptom

Nachdem sich ein Benutzer eingeloggt hat navigiert die Applikation nicht zu einer View. Es ist nur die Navigationsleiste zu sehen und der Contentbereich ist leer.

2.16.2 Behebung

Nachdem rausgefunden wurde, dass nach einem Login zum Mainscreen navigiert wurde und nicht einer speziellen View im Contentbereich konnte damit abhille verschafft werden, dass der navigator des mainscreens auch gleich auf eine View gesetzt wurde. Als Einstiegspunkt wurde die Benutzerbibliothek gewählt, da diese bei allen Rechtegruppen vertreten ist.

2.17 Ändern des Benutzerpassworts

2.17.1 Fehlersymptom

Nach der Änderung des Benutzerpassworts in der Einstellungen, tauchte eine Benachrichtigung dass der Prozess erfolgreich war jedoch die Datenbank hatte noch das alte Passwort enthalten.

2.17.2 Behebung

In der Preferences-Klasse wurde in der Clicklistener des Save-knopfes die Inhalte der Felder mit “==” anstatt “.equals()” verglichen. Wobei die erste operation die Referenzen vergleicht und die zweite die Inhalte.

3 UI Tests

Alle UI Tests sind mittels Testbench geschrieben und laufen automatisiert. Die Ansichten wurden mit den unterschiedlichen Szenarien getestet, die nicht nur das UI sondern auch das Verhalten des gesamten Systems bis zum DAOs überprüfen.

3.1 RegistrationTest

Ein neuer Benutzeraccount wurde mit Hilfe des Anmeldeformulars erstellt. Die Fehlermeldungen bei den falschen Eingabedaten wurden getestet.

3.2 LoginTest

Ein- und Ausloggen wurde ausgeführt.

3.3 AdminTest

Ein Administrator hat die Rechte eines Benutzers auf “Schüler”, “Autor” und “Administrator” gesetzt. Jedes Mal wurde überprüft, ob es dem Benutzer richtige (erlaubte) Ansichten im Navigationsmenü gezeigt wurden. Darüber hinaus wurde das Löschen eines Benutzers im Benutzerrecherche-Ansicht getestet.

3.4 CourseCreationTest

Erstellen und Löschen eines Kurses wurden getestet. Es wurde überprüft, dass der Benutzer die richtigen Fehlermeldungen bekommt, wenn er einen Kurs mit dem ungültigen Namen erstellt.

3.5 CourseAvailabilityTest

Es wurde überprüft, dass, wenn ein Autor einen Kurs einem bestimmten Schüler zuteilt, dann sieht der Schüler diesen Kurs in seiner Benutzerbibliothek. Wenn der Autor den Schüler aus dem Kurs entfernt oder der Schüler selbst den Kurs verlässt, wird der Kurs in der Benutzerbibliothek nicht mehr gezeigt.

3.6 GraphInteractionTest

Die Graphinteraktion in der Kursansicht wurde getestet.

3.7 PreferenceTest

Es wird überprüft, dass ein Benutzer seine Daten ändern kann.

4 Login

Im Login Test loggt sich ein in der Datenbank schon existierender Nutzer ein. Nachdem er eingeloggt ist navigiert er zu Einstellungsseite. Hier wird schlussendlich überprüft, ob die angezeigt Email mit der übereinstimmt, die beim Login benutzt wurde.

5 Unittests

Tabelle 1: Überdeckung: Service und DAO

Test	CourseDAO	CourseService	UserDAO	UserService
<i>CourseServiceTest</i>	67,2%	58,2%	67,5%	35,5%
<i>UserServiceTest</i>	-	-	100%	100%

Tabelle 2: Überdeckung: Graph

Test	Graph	DbCourse	UnitNode	Course
<i>GraphTest</i>	95,8%	51,7%	45%	21,6%

Tabelle 3: Überdeckung: Course und Language

Test	Course	Language	LanguageService
<i>CourseTest</i>	31,3%	51,7%	45%

Tabelle 4: Überdeckung: CrayonsUser

Test	CrayonsUser	Language	LanguageService	CurrentCourseDummy
<i>CrayonsUserTest</i>	82%	58%	33,8%	76,5%