

Testdokumentation

Experimenteverwaltung - Erweiterung Buchungssystem (I2)

Piet Gutsche

2021-07-01 14:40:57 +0200

Inhaltsverzeichnis

1. Einleitung	1
2. Organisation	2
3. Verhalten bei entdeckten Fehlern	3
4. Beschreibung der Tests	4
5. Ausführung	5
6. Ergebnisse	6
7. Schwierigkeiten und Fazit	7
8. Testfallbeschreibung	8
8.1. Allgemeine Informationen und Bedingungen	8
8.2. Test Cases für Use Cases	8
8.2.1. Test Cases für Use Case "Experiment buchen"	8
8.2.2. Test Cases für Use Case "Buchungsjournal verwalten"	11
8.2.3. Test Cases für Use Case "Dozentenwoche"	13
8.2.4. Test Cases für Use Case "Statistik anzeigen"	16
8.3. Test Cases für systemweite Anforderungen	16
8.3.1. Test Cases für systemweite funktionale Anforderungen	16
8.3.2. Test Cases für nicht funktionale Anforderungen	18
8.4. Benutzertests	24
8.4.1. TC36 - Test als Dozent	24
8.4.2. TC37 - Test als Admin	25

1. Einleitung

Das vorliegende Dokument erläutert die eingesetzten Teststrategien, deren Organisation sowie dabei aufgetretene Probleme

2. Organisation

Zur Aufteilung der Verantwortlichkeit für die Tests orientierten wir uns an der Testpyramide. Das Fundament dieser Teststrategie waren Unit-Tests, welche vom Entwickler geschrieben und durchgeführt wurden. Für diese nutzen wir das bereits von unseren Vorgängern benutzte Ruby Testframework [RSpec](#).

Wir legten dafür fest, dass nur getesteter Code in den Main-Branch übernommen werden darf. Des weiteren wurde festgelegt, dass vor jedem Merge ein Codereview durchgeführt wird. Dabei sollte der Entwickler die wesentlichen erstellten Funktionen erläutern. Als Kriterium zum Bestehen des Reviews wurde festgelegt, dass der Code vom Reviewer verstanden werden muss. Zusätzlich wurden vor jedem Merge eines Moduls sämtliche Tests der Regressionstestsuite ausgeführt.

Für alle anderen Tests also UI-Tests, Funktionstest und Systemtests war der Tester verantwortlich. Wir entschieden uns dafür, wann immer mit sinnvollen Aufwand möglich, automatisierte Tests zu nutzen. Dafür nutzten wir das oben benannte Framework RSpec und [Capybara](#) zur Browserautomatisierung.

Somit wurde im Laufe unserer Arbeit die Regressionstestsuite, welche rudimentär schon von unseren Vorgängern aufgebaut wurde, stark erweitert. Diese erlaubt zukünftigen Entwicklern fehlerhafte Auswirkungen von Änderungen zügig zu erkennen.

Neben diesen automatisierten Tests führten wir eine Reihe manueller Tests durch, welche die automatisierten Test in Bereichen ergänzen in denen eine Automatisierung nur mit sehr hohem Aufwand möglich wäre.

Zusätzlich zu den von uns durchgeführten Tests erstellten wir ein Usertestprotokoll was von den zukünftigen Nutzern des Systems abgearbeitet werden sollte.

3. Verhalten bei entdeckten Fehlern

Wir legten fest, dass entdeckte Fehler dem Entwickler gemeldet werden. Gemeinsam mit diesem sollte eine Kategorisierung in optische Unschönheit (z.B. Button etwas zu lang oder nicht richtig angeordnet), leichter Fehler und schwerer Fehler vorgenommen werden. Schwere Fehler sollten dazu führen, dass die aktuellen Arbeiten pausiert und der Fehler sofort behoben werden. Leichte Fehler sollten nach Abschluss der aktuellen Arbeit an einer Funktion oder einem Feature behoben werden. Bei optischen Unschönheiten sollte im Rahmen der Weeklys diskutiert werden, ob diese behoben werden oder ob sie vertretbar wären (z.B. wenn Aufwand zum Beseitigen unverhältnismäßig groß wäre).

4. Beschreibung der Tests

Die Beschreibungen der erstellten Testcase finden sich unter dem Punkt [Testfallbeschreibung](#).

Für alle automatisierten Tests finden sich in diesem Dokument die grobe Beschreibung des Tests und dessen Ziel. Die genauen Werte und die genaue Durchführung wurde in Form von Kommentaren den jeweiligen Specfiles angefügt. Die Zuordnung von Testcase zu automatisiertem Test geschieht über die TC-Nummer (z.B. TC01) Dies verhindert zum einen die doppelte Beschreibung der Tests. Zum anderen erlaubt die genaue Beschreibung im Code zukünftigen Entwicklern ein zügiges Verständnis dieses Codes sowie eine einfache Erweiterbarkeit. Zu finden sind diese unter <https://github.com/Schmiddl99/experimenteverwaltung-i2/tree/main/src/spec/features>.

In diesem Ordner befinden sich auch die vom alten Team erstellte Tests, welche leider kaum Beschreibungen oder Kommentare enthalten. Von diesen alten Tests waren 10 ausgesetzt. Diese behielten wir bei, da sie Funktionen betreffen die noch nicht umgesetzt wurden, aber von Kunden für die Zukunft gewünscht sind. Damit wären sie mit etwas Überarbeitung für ein zukünftiges Team nutzbar.

5. Ausführung

Die automatisierten Tests lassen sich vom Projektordner starten mit:

```
cd src  
rspec
```

6. Ergebnisse

Von den 133 automatisierten Tests waren 10 ausgesetzt die restlichen 123 verliefen ohne Fehler. Die von Rspec errechnete Codecoverage wurde mit 2360 / 2434 LOC also 96.96% angegeben. Die manuellen Tests wurden alle bestanden und auch die Anwendertests förderten keine Fehler zu Tage.

7. Schwierigkeiten und Fazit

Die Hauptschwierigkeit war die zu Beginn recht steile Lernkurve für den Tester. Da er vorher noch keinen Kontakt mit Ruby hatte, musste er sich erst in eine neue Programmiersprache einarbeiten, dann in das auf diese aufbauende Framework Ruby on Rails und zusätzlich noch in das zugehörige Testframework RSpec. Dabei verzögerte die fehlende Fähigkeit in Ruby flüssig programmieren zu können die Erstellung der Tests doch erheblich. Auch behinderte der fehlende Rubybackground ein zügiges einarbeiten und verstehen der alten Testcases. Nachdem diese Hürde überwunden war, konnte die Testsuite sinnvoll erweitert werden, so dass alle von uns erstellten Features einer hohen Testabdeckung unterliegen.

8. Testfallbeschreibung

8.1. Allgemeine Informationen und Bedingungen

- valides Datum: Datum muss in der Zukunft liegen
- valide Uhrzeit: Entsprechend den Möglichkeiten im Drop-Down-Menü
- valider Studiengang: Entsprechend den Möglichkeiten im Drop-Down-Menü

8.2. Test Cases für Use Cases

8.2.1. Test Cases für Use Case "Experiment buchen"

TC01 - Erfolgreich buchen - automatisiert

Beschreibung	Der Dozent legt drei Experimente in den Warenkorb und legt einen Kommentar an. Im Warenkorb löscht er das erste Experiment und schließt die Buchung ab. Es wird erwartet das die zwei verbliebenen Experimente im Warenkorb erfolgreich gebucht werden.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none">• ein Dozent ist mit validen Zugangsdaten angemeldet• es existieren mindestens drei Experimente• valides Datum, valide Uhrzeit und valider Studiengang
Nachbedingung	<ul style="list-style-type: none">• Datenbankeintrag für die Bestellung ist angelegt• Dozent befindet sich auf der Buchungsbestätigungseite

TC02 - Dozent kann keine invaliden Voreinstellungen auswählen

Beschreibung	Der Dozent versucht bewusst falsche Eingaben in den Voreinstellungen zu treffen. Falsche Eingaben sind ein Datum in der Vergangenheit oder fehlerhafte Werte für die Zeit.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none">• ein Dozent ist mit validen Zugangsdaten angemeldet• es wurde eine Buchung begonnen• invalides Datum, invalide Uhrzeit

Nachbedingung	<ul style="list-style-type: none"> • es können nur verfügbare Studiengänge über ein Dropdown Menü ausgewählt werden • es kann nur ein aktuelles oder zukünftiges Datum eingegeben werden • es kann nur eine korrekte Uhrzeit in Verbindung der Unterrichtszeiten eingegeben werden
----------------------	---

TC03 - Leerer Warenkorb darf nicht zur Buchung führen - automatisiert

Beschreibung	Der Dozent hat die Voreinstellungen getroffen und versucht ohne ein Experiment ausgewählt zu haben die Buchung abzuschließen.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • ein Dozent ist mit validen Zugangsdaten angemeldet • Buchung wurde mit validen Daten gestartet • valides Datum, valide Uhrzeit und valider Studiengang
Nachbedingung	<ul style="list-style-type: none"> • der Button "Buchung abschließen" ist bis zur Auswahl mindestens eines Experiments ausgegraut und kann nicht geklickt werden • Dozent bleibt auf der Seite Experimentenübersicht

TC04 - Keine Mehrfachauswahl eines Experiments pro Buchung möglich - automatisiert

Beschreibung	Der Dozent versucht das selbe Experiment mehrmals in den Warenkorb mitaufzunehmen.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • ein Dozent ist mit validen Zugangsdaten angemeldet • Buchung wurde mit validen Daten gestartet • es wurde mindestens ein Experiment ausgewählt • es muss mindestens ein Experiment geben
Nachbedingung	<ul style="list-style-type: none"> • es ist nicht möglich das selbe Experiment mehrmals auszuwählen

TC05 - Abbruch muss möglich sein - automatisiert

Beschreibung	Der Dozent beginnt einen Buchungsprozess, entscheidet sich aber zu einem beliebigen Zeitpunkt vor Buchungsabschluss dazu, diesen abzuberechnen. Erwartet wird, dass ein Buchungsabbruch jederzeit fehlerfrei möglich ist.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none">• ein Dozent ist mit validen Zugangsdaten angemeldet• Buchung wurde mit validen Daten gestartet• es existiert mind. ein Experiment
Nachbedingung	<ul style="list-style-type: none">• Buchung wurde abgebrochen und es kam zu keinem Datenbankeintrag• Dozent befindet sich auf der Startseite der Webanwendung

TC06 - Buchung eines Dummyexperiments muss möglich sein - automatisiert

Beschreibung	Der Dozent beginnt einen Buchungsprozess. Er wählt den Menüpunkt manuell Experiment hinzufügen aus, vergibt einen Namen und schließt die Buchung ab.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none">• ein Dozent ist mit validen Zugangsdaten angemeldet• Buchung wurde mit validen Daten gestartet
Nachbedingung	<ul style="list-style-type: none">• Datenbankeintrag für die Bestellung und das Dummyexperiment ist angelegt• Dozent befindet sich auf der Buchungsbestätigungseite

TC07 - Buchung eines regulären und eines Dummyexperiments muss möglich sein - automatisiert

Beschreibung	Der Dozent beginnt einen Buchungsprozess. Er legt ein Experiment in den Warenkorb und wählt den Menüpunkt manuell Experiment hinzufügen aus, vergibt einen Namen und schließt die Buchung ab. Es ist sowohl möglich erst das manuelle Experiment als auch das reguläre Experiment zuerst zur Bestellung hinzuzufügen.
---------------------	---

Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • ein Dozent ist mit validen Zugangsdaten angemeldet • Buchung wurde mit validen Daten gestartet • es existiert mindestens ein Experiment
Nachbedingung	<ul style="list-style-type: none"> • Datenbankeintrag für die Bestellung und das Dummyexperiment ist angelegt • Dozent befindet sich auf der Buchungsbestätigungseite

TC08 - Die Reihenfolge der Experimente kann getauscht werden - automatisiert

Beschreibung	Der Dozent legt 2 Experimente in den Warenkorb und geht zum Buchungsabschluss. Erwartet wird das eine Vertauschung der Experimentenreihenfolge möglich ist.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • ein Dozent ist mit validen Zugangsdaten angemeldet • Buchung wurde mit validen Daten gestartet • es existieren mindestens zwei Experimente
Nachbedingung	<ul style="list-style-type: none"> • das zuerst gebuchte Experiment befindet sich nach der Vertauschung an zweiter Stelle.

8.2.2. Test Cases für Use Case "Buchungsjournal verwalten"

TC09 - Journal enthält chronologisch geordnete Buchungen -automatisiert

Beschreibung	Der Dozent will seine Buchungen im Journal einsehen. Erwartet wird, dass die Buchungsauflistung chronologisch geordnet und inhaltlich richtig ist.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • ein Dozent ist mit validen Zugangsdaten angemeldet • Buchungen wurden erfolgreich getätigt • Buchungseinträge (mind. 3) in der Datenbank

Nachbedingung	<ul style="list-style-type: none"> • Dozent befindet sich auf der Journalseite • Buchungsauflistung wird chronologisch geordnet (bezüglich der Datums- und Zeitangaben (und der Reihenfolge bei mehreren Buchungen pro Vorlesung)) und inhaltlich richtig angezeigt
----------------------	---

TC10 - Leeres Journal anzeigen -automatisiert

Beschreibung	Der Dozent will das Journal einsehen. Erwartet wird, dass dieses leer ist und keine Einträge angezeigt werden, wenn dieser Dozent noch nichts gebucht hat.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • ein Dozent ist mit validen Zugangsdaten angemeldet • keine Buchungen für diesen Dozenten
Nachbedingung	<ul style="list-style-type: none"> • Dozent befindet sich auf der (leeren) Journalseite

TC11 - Buchung im Journal bearbeiten -automatisiert

Beschreibung	Ein Dozent bearbeitet eine Buchung im Journal indem er ein Experiment hinzufügt, eines entfernt und den Kommentar zur Buchung ändert. Es wird erwartet, dass alle geänderten Daten übernommen werden.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • ein Dozent ist mit validen Zugangsdaten angemeldet • Dozent befindet sich im Journal • mindestens drei Experimente, sodass eins zur Buchung hinzugefügt werden kann • Buchung von dem Dozenten mit mindestens zwei Experimenten und einem Kommentar
Nachbedingung	<ul style="list-style-type: none"> • Datenbankeintrag für die Buchung wurde aktualisiert • Dozent befindet sich wieder im Journal, wobei die aktualisierte Buchung angezeigt wird

TC12 - Buchung kann im Journal gelöscht werden - automatisiert

Beschreibung	Ein Dozent entfernt eine Buchung im Journal. Es wird erwartet, dass diese Buchung nicht mehr im Journal angezeigt wird.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none">• ein Dozent ist mit validen Zugangsdaten angemeldet• Dozent befindet sich im Journal• mindestens ein Experiment• Buchung von dem Dozenten
Nachbedingung	<ul style="list-style-type: none">• Datenbankeintrag für die Buchung wurde entfernt• Dozent befindet sich wieder im Journal, wobei das Experiment nicht mehr angezeigt wird

TC13 - Buchungen in der Vergangenheit können nicht bearbeitet werden - automatisiert

Beschreibung	Ein Dozent wählt eine Buchung in der Vergangenheit aus und versucht über die entsprechenden Icons seine Buchung zu bearbeiten oder zu löschen. Es wird erwartet dass die Icons nicht angezeigt werden, falls die Buchung in der Vergangenheit liegt.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none">• ein Dozent ist mit validen Zugangsdaten angemeldet• Dozent befindet sich im Journal• mindestens ein Experiment• Buchung von dem Dozenten mit einem Datum in der Vergangenheit
Nachbedingung	<ul style="list-style-type: none">• Dozent befindet sich weiterhin im Journal

8.2.3. Test Cases für Use Case "Dozentenwoche"

TC14 - Dozentenwoche von einem Dozenten wird korrekt angezeigt - automatisiert

Beschreibung	Der Admin möchte sich für einen bestimmten Dozenten die Dozentenwoche anzeigen lassen, inklusive aller gebuchten Experimente.
---------------------	---

Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • Admin ist angemeldet • Admin befindet sich auf der Website für die Dozentenwoche • mind. ein Dozent hat eine Buchung getätigt
Nachbedingung	<ul style="list-style-type: none"> • Die gewählte Dozentenwoche wird angezeigt

TC15 - Dozentenwoche von mehreren Dozenten wird korrekt angezeigt - automatisiert

Beschreibung	Der Admin möchte sich für verschiedene Dozenten die Dozentenwoche anzeigen lassen, inklusive aller gebuchten Experimente.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • Admin ist angemeldet • Admin befindet sich auf der Website für die Dozentenwoche • mind. 2 Dozenten haben eine Buchung getätigt
Nachbedingung	<ul style="list-style-type: none"> • Die gewählten Dozentenwochen werden angezeigt.

TC16 - Dozentenwoche wird korrekt gedruckt - automatisiert

Beschreibung	Der Admin möchte sich seine entsprechend gefilterte, inkl. aller gebuchten Experimente, Dozentenwoche drucken lassen. Um eine automatisierte Überprüfung möglich zu machen, wird überprüft ob der korrekte Befehl an das System übermittelt wurde.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • Admin ist angemeldet • Admin befindet sich auf der Website für die Dozentenwoche • mind. ein Dozent hat eine Buchung getätigt
Nachbedingung	<ul style="list-style-type: none"> • Das Druckermenü für die gewünschte Darstellung wurde aufgerufen

TC17 - Dozentenwoche wird korrekt gedruckt - manuell

Beschreibung	Der Admin möchte sich seine entsprechend gefilterte, inkl. aller gebuchten Experimente, Dozentenwoche drucken lassen.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • Admin ist angemeldet • Admin befindet sich auf der Website für die Dozentenwoche • mind. ein Dozent hat eine Buchung getätigt
Nachbedingung	<ul style="list-style-type: none"> • Die Dozentenwoche wird in der gewünschten Darstellung gedruckt.

TC18 - Dozentenwoche für einen Dozenten wird korrekt angezeigt - automatisiert

Beschreibung	Der Dozent möchte sich seine Dozentenwoche für eine bestimmte KW ausgeben lassen.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • ein Dozent ist mit validen Zugangsdaten angemeldet • Dozent befindet sich auf der Website für die Dozentenwoche • valide KW, valider Dozentennamen • mind. eine Buchung vom Dozenten existiert
Nachbedingung	<ul style="list-style-type: none"> • die entsprechend gefilterte, inkl. aller gebuchten Experimente, Dozentenwoche wird angezeigt

TC19 - Dozentenwoche wird korrekt gedruckt - automatisiert

Beschreibung	Der Dozent möchte sich seine Dozentenwoche drucken lassen. Um eine automatisierte Überprüfung möglich zu machen, wird überprüft ob der korrekte Befehl an das System übermittelt wurde.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • Dozent ist angemeldet mit validen Zugangsdaten • Dozent befindet sich auf der Website für die Dozentenwoche • der Dozent hat mind. eine Buchung getätigt
Nachbedingung	<ul style="list-style-type: none"> • Das Druckermenü für die gewünschte Darstellung wurde aufgerufen

TC20 - Dozentenwoche wird korrekt gedruckt - manuell

Beschreibung	Der Dozent möchte sich seine entsprechend gefilterte, inkl. aller gebuchten Experimente, Dozentenwoche drucken lassen.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none">• Dozent ist angemeldet• Dozent befindet sich auf der Website für die Dozentenwoche• der Dozent hat mind. eine Buchung getätigt
Nachbedingung	<ul style="list-style-type: none">• Die Dozentenwoche wird in der gewünschten Darstellung gedruckt.

8.2.4. Test Cases für Use Case "Statistik anzeigen"

TC X1 - Statistik wird richtig angezeigt - nicht umgesetzt da Usecase nicht umgesetzt wurde

Beschreibung	Ein Admin betrachtet die Statistik. Es wird erwartet, dass alle Buchungen aus den letzten zwei Semestern in die Statistik einbezogen werden.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none">• ein Admin ist mit validen Zugangsdaten angemeldet• Admin befindet sich auf der Statistik-Seite• mindestens drei Experimente, um die Sortierung zu testen• mehrere Buchungen der vorhandenen Experimente innerhalb sowie außerhalb der beiden Semester
Nachbedingung	<ul style="list-style-type: none">• Anzahl der Buchungen der letzten beiden Semester sowie die Gesamtzahl ist richtig berechnet• die aufgelisteten Experimente sind nach der Gesamtnutzung sortiert

8.3. Test Cases für systemweite Anforderungen

8.3.1. Test Cases für systemweite funktionale Anforderungen

TC21 - SWFA-1 - Auf fremde Daten/Funktionalitäten kann nicht zugegriffen werden - manuell

Beschreibung	Es wird erwartet, dass der Admin keinen Zugriff auf das Dozenten-Journal hat, die Dozenten nur ihr persönliches Journal einsehen dürfen und nur der Admin die Statistik abrufen kann. Auch dürfen die Dozenten nur ihre eigene Dozentenwoche abrufen.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none">• ein Admin ist mit Zugangsdaten angemeldet• ein Dozent ist mit validen Zugangsdaten angemeldet
Nachbedingung	<ul style="list-style-type: none">• Admin hatte keinen Zugriff auf beliebiges Journal• Dozent konnte kein fremdes Journal einsehen• Dozent konnte die Statistik nicht aufrufen• Dozent konnte keine fremde Dozentenwoche einsehen

TC22 - SWFA-2 - Buchungs- und Anwenderdaten werden persistent gespeichert - manuell

Beschreibung	Nach einem Neustart des Systems müssen die Anwenderdaten noch abrufbar sein.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none">• es muss ein Adminkonto geben• es muss ein Dozentenkonto geben• es muss eine Buchung geben
Nachbedingung	<ul style="list-style-type: none">• Daten müssen abrufbar sein

TC23 - SWFA-3 - Zugangsdaten können nicht ausgelesen werden - manuell

Beschreibung	Es darf nicht möglich sein, die Zugangsdaten der Konten auszulesen.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none">• es muss ein Adminkonto geben• es muss ein Dozentenkonto geben
Nachbedingung	<ul style="list-style-type: none">• Die Daten konnten nicht ausgelesen werden

8.3.2. Test Cases für nicht funktionale Anforderungen

TC24 - NFAU-1.1 - Es gibt keine redundanten Schaltflächen - manuell

Beschreibung	Es wird erwartet, dass es zu keinem Zeitpunkt Schaltflächen gibt, die denselben Zweck erfüllen oder dieselbe Funktion aufrufen.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none">• Admin ist angemeldet• Dozent ist angemeldet• es existiert mindestens ein Experiment• es existiert mindestens eine Buchung
Nachbedingung	<ul style="list-style-type: none">• In allen möglichen Abläufen des Funktionsumfanges existieren keine Schaltflächen, die denselben Zweck erfüllen oder dieselbe Funktion aufrufen

TC25 - NFAU-1.2 - Alle Eingaben mit festem Wertebereich sind nicht durch Freitextfelder realisiert - manuell

Beschreibung	Die Werte der Voreinstellungen (Studiengang, Uhrzeit, Datum) und die Auswahl der Details von Ausgaben (Dozentenwoche: Dozent, Datum; Statistik: KW, Semester) und alle Filtermöglichkeiten werden aus einem festgelegten Wertebereich gewählt. Realisiert wird dies durch Dropdown-Menüs und einen Datepicker. Es wird erwartet, dass die oben genannten Eingaben nicht durch Freitextfelder erfasst werden können.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none">• Dozent oder Admin ist angemeldet• Buchung wurde gestartet/ Dozentwochen- seite wurde aufgerufen/ Statistikseite wurde aufgerufen• auswählbare Daten bezüglich Datum, Uhrzeit, Studiengang, Dozent, Semester, KW, Filtermöglichkeiten

Nachbedingung	<ul style="list-style-type: none"> • Dozent befindet sich auf Voreinstellungsseite mit ausgewählten Einstellungen (bei Buchung) oder Admin betrachtet ausgefüllte Menüs (bei Dozentenwoche oder Statistik) • zum Auswählen mussten keine Tastatureingabe getätigt werden
----------------------	--

TC26 - NFAU-1.2 - Alle vorgegebenen Eingabemöglichkeiten sind valide - manuell

Beschreibung	Die auswählbaren Werte der Auswahlsmenüs beziehen sich auf feste Wertebereiche. Erwartet wird, dass alle Eingabemöglichkeiten valide sind.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • Dozent/Admin ist angemeldet • Buchung wurde gestartet/ Dozentwoche wurde aufgerufen/ Statistikseite wurde aufgerufen • auswählbare Daten bezüglich Datum, Uhrzeit, Studiengang
Nachbedingung	<ul style="list-style-type: none"> • Dozent befindet sich auf Voreinstellungsseite mit ausgewählten Einstellungen oder Admin betrachtet ausgefüllte Menüs (bei Dozentenwoche oder Statistik) • es wurde keine invalide Eingabemöglichkeit geboten

TC27 - NFAU-1.3 - Themenfremde Person kann die Anwendung ohne fremde Hilfe benutzen - manuell

Beschreibung	Die Anwendung sollte nutzerfreundlich und leicht erfassbar sein. Erwartet wird, dass eine themenfremde Person die Anwendung (bezogen auf einen Buchungsdurchlauf) ohne Hilfe durch Dritte nutzen kann.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • Testperson ist mit einem Dozentenkonto angemeldet • es gibt mindestens ein buchbares Experiment

Nachbedingung	<ul style="list-style-type: none"> • die Testperson befindet sich auf der Bestätigungsseite und hat diese ohne externe Hilfestellung bezüglich der Nutzung erreicht • dementsprechend wurde eine Buchung erfolgreich getätigt und ist im Journal nachvollziehbar
----------------------	--

TC28 - NFAU-1.4 - Buchung eines Experiments kann innerhalb von 15 Klicks abgeschlossen werden - manuell

Beschreibung	Der Buchungsprozess sollte unkompliziert und gut orientiert sein. Es wird deshalb erwartet, dass die erfolgreiche Buchung eines Experiments mit 15 Klicks oder weniger passiert.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • Dozent ist angemeldet • es gibt mindestens ein buchbares Experiment
Nachbedingung	<ul style="list-style-type: none"> • Dozent befindet sich auf der Bestätigungsseite • die Anzahl der gezählten Klicks ist gleich 15 oder weniger

TC29 - NFAU-1.4 - Nicht erreichbare Funktionalitäten können nicht aufgerufen werden - manuell

Beschreibung	Es wird erwartet, dass Funktionen, die aktuell nicht verfügbar sind, nicht aufgerufen werden können. Im Speziellen betrifft das die Schaltfläche zur Warenkorb-Ansicht, wenn noch kein Experiment in den Warenkorb hinzugefügt wurde.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • ein Dozent ist mit validen Zugangsdaten angemeldet • es wurde eine Buchung, mit validen Voreinstellungen, begonnen
Nachbedingung	<ul style="list-style-type: none"> • Warenkorb-Ansicht ist nicht aufrufbar • Schaltflächen dazu sind deaktiviert

TC30 - NFAU-1.5 - Texte und Schaltflächen sind gut lesbar - manuell

Beschreibung	Eine Testperson nutzt die Anwendung mit einem Abstand von 50cm zum Bildschirm. Dabei wird erwartet, dass alle Texte und Schaltflächen gut lesbar sind und Grafiken von guter grafischer Qualität sind.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none">• ein Experiment• valides Datum, valide Uhrzeit und valider Studiengang• ein Testzugang für einen Dozenten sowie für einen Admin mit gültigen Anmeldedaten
Nachbedingung	<ul style="list-style-type: none">• während eines Buchungsvorgangs sind die Erwartungen erfüllt• beim Betrachten des Journals und der eigenen Dozentenwoche sind die Erwartungen erfüllt• beim Betrachten der Dozentenwoche aller Dozenten sowie der Statistik als Admin sind die Erwartungen erfüllt

TC31 - NFAU-2 - Eingaben geschehen nicht in Pop-Ups - manuell

Beschreibung	Eine Testperson nutzt alle Funktionen der Anwendung. Dabei wird erwartet, dass keine Eingaben über Pop-Ups erfolgen.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none">• ein Experiment• valides Datum, valide Uhrzeit und valider Studiengang• ein Testzugang für einen Dozenten sowie für einen Admin mit gültigen Anmeldedaten

Nachbedingung	<ul style="list-style-type: none"> • während eines Buchungsvorgangs sind die Erwartungen erfüllt • beim Betrachten und Einstellen des Journals und der eigenen Dozentenwoche sind die Erwartungen erfüllt • beim Bearbeiten und Löschen von Buchungen sind die Erwartungen erfüllt • beim Betrachten und Einstellen der Dozentenwoche aller Dozenten sowie der Statistik als Admin sind die Erwartungen erfüllt
----------------------	---

TC32 - NFAR-1 - Buchung kann nach Verbindungsabbruch des Nutzers fortgesetzt werden - manuell

Beschreibung	Wird die Verbindung zum Server während einer Buchung unterbrochen (durch Internetausfall, Browser bzw. Tab schließen o.Ä.) gehen bereits gewählte Voreinstellungen oder Experimente nicht verloren. Dazu wird eine Testperson während eines Buchungsprozesses, bei welchem bereits Voreinstellungen getätigt wurden und ein Experiment in den Warenkorb gelegt wurde, einen Internetausfall simulieren.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • ein Dozent ist mit validen Zugangsdaten angemeldet • valides Datum, valide Uhrzeit und valider Studiengang • ein Experiment
Nachbedingung	<ul style="list-style-type: none"> • nach dem simulierten Internetausfall sind die Voreinstellungen noch ausgewählt und das Experiment befindet sich noch im Warenkorb • die Buchung ist erfolgreich abgeschlossen

TC33 - NFAP-1 - Das System funktioniert bei Auslastung durch 10 Nutzer - manuell

Beschreibung	Das System wird einem Stresstest durch 10 gleichzeitige Nutzer unterzogen. Es wird erwartet, dass das System fehlerlos funktioniert.
---------------------	--

Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • 10 verschiedene Accounts für die Testnutzer (2 Admin, 7 Dozent und 1 Gast) • alle 10 Testnutzer sind angemeldet • alle 10 Testnutzer benutzen jede Funktionalität des Systems
Nachbedingung	<ul style="list-style-type: none"> • das System zeigt keine Fehler an und hatte keine Probleme während des Testlaufes

TC34 Das System funktioniert auch nachdem eine sehr große Anzahl Buchungen getätigt wurden - automatisiert

Beschreibung	Es wird überprüft ob das System auch nach 2000 getätigten Buchungen fehlerlos und ohne Verzögerungen funktioniert. Nachdem diese Buchungen automatisiert erstellt wurden, tätigt der Nutzer eine Buchung und besucht im Anschluss sein Journal und überprüft dort die Buchung.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • 2000 getätigte Buchungen • ein Dozent ist angemeldet *
Nachbedingung	<ul style="list-style-type: none"> • das System zeigt keine Fehler an und hatte keine Probleme während des Testlaufes *Dozent befindet sich im Journal und sieht dort seine getätigte Buchung

TC35 - NFAS-2 - Die Anwendung muss in allen geforderten Browsern funktionieren - manuell

Beschreibung	Das System wird in allen geforderten Browser auf volle Funktionalität getestet. Es wird erwartet, dass das System in allen getesteten Browsern vollumfänglich und mit gleichen Ergebnissen funktioniert.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • ein Admin-, ein Dozenten- und ein Gastkonto • alle geforderten Browser in der aktuellen Version
Nachbedingung	<ul style="list-style-type: none"> • die Funktionen des Systems haben in den unterschiedlichen Browsern wie erwartet und entsprechend den Berechtigungen der einzelnen Accounts funktioniert • die Webanwendung ist korrekt dargestellt

8.4. Benutzertests

8.4.1. TC36 - Test als Dozent

Aufgabe	Aktionen
Login in das System	<ul style="list-style-type: none">• Website aufrufen• Benutzername und Passwort eingeben• einloggen
Buchung starten und Voreinstellungen wählen	<ul style="list-style-type: none">• auf Buchung starten klicken• Voreinstellungen ausfüllen<ul style="list-style-type: none">◦ Chemie◦ Mechanikvorlesung◦ 21.06.2021◦ 15:10• Voreinstellungen speichern
Buchungsprozess	<ul style="list-style-type: none">• Mechanik auswählen• Kinematik auswählen• Experimente mit Einkaufswagen auswählen (Anzahl nicht vorgegeben, min. zwei)• Manuell Experiment Hinzufügen anklicken• „Mein Eigenes Experiment“ eingeben• Experiment hinzufügen• Buchung abschließen
Buchungsübersicht verwalten	<ul style="list-style-type: none">• Mechanik auswählen• Reihenfolge per Drag and Drop ändern• „Das ist ein Text“ beim Kommentar eingeben• Änderungen speichern
Journal verwenden	<ul style="list-style-type: none">• Zu Journal navigieren• Auf angelegte Vorlesung klicken• Auf Stift klicken• Ein Experiment löschen• Änderung speichern• Bearbeitung abschließen

Dozentenwoche verwenden	<ul style="list-style-type: none"> • Dozentenwoche auswählen • Voreinstellungen ausfüllen <ul style="list-style-type: none"> ◦ 2021 ◦ KW 25 • Auf Anzeigen klicken • Taste „STRG“ + „P“ drücken • Bei Ziel „als PDF speichern“ auswählen • Speichern anklicken und Speicherpfad wählen
-------------------------	---

8.4.2. TC37 - Test als Admin

Aufgabe	Aktionen
Login in das System	<ul style="list-style-type: none"> • Website aufrufen • Benutzername und Passwort eingeben • einloggen
Dozentenwoche verwenden	<ul style="list-style-type: none"> • Dozentenwoche auswählen • Voreinstellungen ausfüllen <ul style="list-style-type: none"> ◦ Frau Prof. Krawietz ◦ 2021 ◦ KW 25 • Auf Anzeigen klicken • Taste „STRG“ + „P“ drücken • Bei Ziel gewünschter Drucker auswählen • Drucken anklicken