

SE I - Belegabgabe
Experimenteverwaltung - Erweiterung
Buchungssystem (I2)

Alena Bischoff, Gregor Müller, Julius Schmidt, Richard Böhme, Taejun Kim, Tim
Würzburg, Tobias Karsch, William Elsner

12. Februar 2021

Inhaltsverzeichnis

Technische Spezifikation	1
1. Vision: Experimenteverwaltung - Erweiterung Buchungssystem (I2)	2
1.1. Einführung	2
1.2. Positioning	2
1.3. Stakeholder Beschreibungen	4
1.4. Produkt-/Lösungsüberblick	6
1.5. Zusätzliche Produkthanforderungen	6
1.6. Aussicht für zukünftige Projekte	6
2. Use-Case Model: Experimenteverwaltung I2	8
2.1. Allgemeine Informationen	8
2.2. Use-Case: Experiment buchen	9
2.3. Use-Case: Dozentenwoche ansehen	11
2.4. Use-Case: Eigene Dozentenwoche ansehen	12
2.5. Use-Case: Eigene Dozentenwoche drucken	13
2.6. Use-Case: Dozentenwoche drucken	14
2.7. Use-Case: Eigene Experimente im Buchungsjournal ansehen	15
2.8. Use-Case: Experimente im Buchungsjournal bearbeiten	17
2.9. Use-Case: Eigene Experimente im Buchungsjournal löschen	19
2.10. Use-Case: Statistik anzeigen	21
2.11. Grafische Veranschaulichung	23
3. Wireframes Übersicht und Erklärungen: Experimenteverwaltung - Erweiterung Buchungssystem (I2)	25
3.1. Wireframes	25
4. System-Wide Requirements: Experimenteverwaltung - Erweiterung Buchungssystem (I2) ...	31
4.1. Einführung	31
4.2. Systemweite funktionale Anforderungen	31
4.3. Qualitätsanforderungen für das Gesamtsystem	32
4.4. Zusätzliche Anforderungen	33
5. Glossar: Experimenteverwaltung I2	35
5.1. Einführung	35
6. Domain Model: Experimenteverwaltung - Erweiterung Buchungssystem (I2)	39
6.1. Domänenmodell für die Erweiterung des EMS	39
Projektdokumentation	40
7. Projektplan: Experimenteverwaltung - Erweiterung Buchungssystem (I2)	41
7.1. Einführung	41
7.2. Projektorganisation	41
7.3. Praktiken und Bewertung	41
7.4. Meilensteine und Ziele	43

7.5. Deployment	44
7.6. Erkenntnisse (Lessons learned)	44
8. Risikoliste: Experimenteverwaltung - Erweiterung Buchungssystem (I2)	45
8.1. Attribute	45
8.2. Risiken	45
9. Iteration Plan 02: Experimenteverwaltung - Erweiterung Buchungssystem (I2)	49
9.1. Meilensteine	49
9.2. Wesentliche Ziele	49
9.3. Aufgabenzuordnung	49
9.4. Probleme (optional)	52
9.5. Bewertungskriterien	53
9.6. Assessment	53
Entwurfsdokumentation	58
10. Architecture Notebook: Experimenteverwaltung - Erweiterung Buchungssystem (I2)	59
10.1. Zweck	59
10.2. Architekturziele und Philosophie	59
10.3. Beschränkungen, Annahmen und Abhängigkeiten	59
10.4. Architektur-relevante Anforderungen	60
10.5. Entscheidungen, Nebenbedingungen und Begründungen	61
10.6. Architekturmechanismen	62
10.7. Wesentliche fachliche Abstraktionen	64
10.8. Schichten oder Architektur-Framework	64
10.9. Architektursichten (Views)	65
11. Test Cases: Experimenteverwaltung - Erweiterung Buchungssystem (I2)	68
11.1. Allgemeine Informationen und Bedingungen	68
11.2. Test Cases für Use Cases	68
11.3. Test Cases für systemweite Anforderungen:	75

Technische Spezifikation

- Vision
- Use Case Model (inkl. Wireframes, sofern vorhanden)
- System-wide Requirements
- Glossar
- Domänenmodel

1. Vision: Experimenteverwaltung - Erweiterung Buchungssystem (I2)

1.1. Einführung

Ziel dieses Dokumentes ist es, einen genaueren Überblick über die wesentlichen Bestandteile der Änderung an der Web-App “Experimente-Management-Software” (EMS) aufzuzeigen und Informationen zu sammeln, zu analysieren und genauer zu definieren. Wir werfen einen Blick auf die einzelnen Stakeholder der Anwendung, deren Wünsche und Bedingungen. Die genaueren Details, wie die angegebenen Bedürfnisse realisiert werden, finden sich in dem [Use Case Model](#) und der [System-Wide-Requirements](#) wieder.

Stakeholder:

- Kunde: Herr Matthias Heisig (fortan mit “Kunde” bezeichnet)
- Dozenten (=Nutzer des Buchungssystems)
- Technischer Ansprechpartner für Virtuelle Maschine (TAVM): Herr Hornoff

1.1.1. Zweck

Der Zweck dieses Dokuments ist es, die wesentlichen an das System gestellten Anforderungen aus Sicht der künftigen Anwender zu beschreiben. Formulierungen werden so gestaltet, dass sie für alle Stakeholder verständlich sind.

1.1.2. Gültigkeitsbereich (Scope)

Dieses Visions-Dokument bezieht sich auf das Experimente-Management-System, das von Team I2 entwickelt wird. Das System wird es dem Admin erlauben, ein Buchungssystem zu betreiben, um damit das umständliche Bestellen per E-Mail abzulösen.

1.1.3. Definitionen, Akronyme und Abkürzungen

siehe [Glossar](#)

1.1.4. Referenzen

based on: https://www2.htw-dresden.de/~anke/openup/core.tech.common.extend_supp/guidances/templates/resources/vision_tpl.dot [EPL V1.0]

1.2. Positioning

1.2.1. Fachliche Motivation

Unser Kunde besitzt bereits eine Web-Anwendung, welche es ihm und seinen Kollegen (Dozenten) möglich macht, physikalische Experimente in passenden Kategorien zu betrachten. Details und

Bestandteile können angesehen und eine Karteikarte generiert werden, welche genau diese Informationen druckbar macht, um diese physisch vorliegen zu haben. Für die Dozenten ist es wichtig, bei unserem Kunden ihre Wahl an durchzuführenden Experimenten für die kommende Woche in Auftrag zu geben. Dies geschieht bisher formlos per Mail an den Kunden.

Diese Methode ist sehr unübersichtlich, da die einzelnen Mails der Dozenten ausgewertet werden müssen und keine einheitliche Bestellvorlage verwendet wird. Die Mails müssen händisch nach Wochentag und Dozent sortiert werden, was einen deutlichen Mehraufwand darstellt.

Um diesen Mehraufwand zu umgehen, möchte der Kunde die Möglichkeit, Experimente in einen “Warenkorb” hinzuzufügen. In diesem sollen je Vorlesung verschiedene Experimente hinterlegt und als Auftrag bestellt werden können. Schlussendlich sollen diese Daten dann automatisiert abrufbar gemacht werden und zur einfachen Planung der Bestellungen als in Tage gegliederte Wochen pro Dozent angezeigt werden.

Ebenfalls soll die Möglichkeit einer Auswertung der vergangenen Bestellungen ermöglicht werden. Dies ist wichtig, um herauszufinden, welche Experimente besonders oft gebucht werden, um bspw. Verschleiß der Geräte vorauszusagen und in Folge dessen mehr Geräte oder Instrumente einzukaufen.

Als Zusatzziel gilt die Verwaltung der in den Experimenten enthaltenen Bilder. Bei der Anschaffung von neuen Instrumenten für die Experimente müssen auch die Bilder entsprechend erneuert werden. Dabei soll, um Konsistenz zu bewahren, die Bezeichnung der Bilder gleich bleiben, das Bild jedoch in Datei und Reihenfolge anpassbar gemacht werden.

1.2.2. Problem Statement

Das Problem	<ul style="list-style-type: none">• Dozent muss für die Folgeweche eine Mail verfassen, welche seine gewünschte Experimente-Auswahl enthält• Jede Änderung erfordert das erneute Senden einer Mail• Kunde muss händisch E-Mails der Dozenten auswerten und sortieren; sich selbst einen Plan der benötigten Experimente je Wochentag erarbeiten
Betrifft	<ul style="list-style-type: none">• Kunde• Dozenten
Die Auswirkung davon ist	<ul style="list-style-type: none">• Dozenten haben verschiedene “Templates” für das Versenden ihrer Wünsche per Mail → kein normales “Kopieren” möglich• Keine automatische Sortierung pro Tag; muss händisch durch Kunden in richtige Reihenfolge gebracht werden• Unnötiges Aufkommen von vielen Mails• Es ist nicht ersichtlich, wessen Wochenplan noch nicht aufgegeben wurde

Eine erfolgreiche Lösung wäre	<ul style="list-style-type: none"> • Warenkorb-System: Die Experimente können nach Festlegung der Voreinstellungen für eine Vorlesung in einen Warenkorb abgelegt werden. Anschließend kann die Reihenfolge noch angepasst werden. Nach erfolgreicher Buchung werden sie dem persönlichen Journal hinzugefügt. • Journal-System: Buchungsübersicht des Dozenten, wo er alle vergangenen Buchungen einsehen und aktuelle Buchungen anpassen kann • Dozentenwochen-System: Buchungsübersicht des Admins, wo er alle Buchungen pro Dozent und sortiert je Wochentag einsehen kann
-------------------------------	---

1.2.3. Product Position Statement

Für	Dozenten, Admin [Kunde]
Der / Die	<ul style="list-style-type: none"> • Dozenten: <ul style="list-style-type: none"> ◦ ... die Experimente für die kommenden Vorlesungen buchen wollen • Admin: <ul style="list-style-type: none"> ◦ ... der einsehen will, welche Experimente gebucht wurden ◦ ... der einsehen will, in welchem Umfang die Experimente in der Vergangenheit gebucht wurden
Das Produkt / Die Lösung ist eine	... Erweiterung der Web-Anwendung "Experimente-Management-System"
Welche	... die Buchung von Experimenten und Einsicht der Buchungen sowie deren Historie ermöglicht
Im Gegensatz zu	... der Buchung über E-Mail und der manuellen Historie sowie Aufbereitung
Unser Produkt	... ermöglicht die einfache Bestellung von Experimenten und Generierung eines für den Anwender optimierten Buchungsplans, sowie eine Übersicht der bisherigen Buchungen

1.3. Stakeholder Beschreibungen

1.3.1. Zusammenfassung der Stakeholder

Name	Beschreibung	Verantwortlichkeiten
Admin/Kunde	<ul style="list-style-type: none"> • Mitarbeiter der HTW • Vorlesungsassistent im Bereich Energietechnik/Physik • Versorgt Dozenten mit Experimenten (“Interner Dienstleister”) • Möchte seine wöchentliche Arbeit optimieren • Möchte, dass die Dozenten möglichst minimalen Aufwand haben 	<ul style="list-style-type: none"> • Stellt Experimente nach Bestellung bereit • “Verwalter” des Webtools – stellt Informationen zu den Experimenten bereit • Führt diese durch • Kümmt sich um Instandhaltung der Experimentiergeräte • Verantwortlicher für Datenschutzaufgaben (Datenschutzerklärung verfassen & herausgeben)
Dozent	<ul style="list-style-type: none"> • Mitarbeiter im Bereich Energietechnik/Physik der HTW • Hat Interesse an Experimenten für seine Vorlesung • Interesse an vereinfachter Buchung für bestimmten Zeitslot • Ist Frontend-Benutzer des Systems 	<ul style="list-style-type: none"> • Pünktlich Experimente buchen • Ort und Zeit muss gewährleistet werden
TAVM (Herr Hornoff)	<ul style="list-style-type: none"> • Betreuer der VM 	*Gewährleistung eines Arbeitsumfeldes für uns
Rechenzentrum der HTW	<ul style="list-style-type: none"> • Ist "Standort" der Nutzeranwendung und Datenbank 	<ul style="list-style-type: none"> • Muss die Bereitstellung der Anwendung sicherstellen

1.3.2. Benutzerumgebung

Für Admin:

- Bisherige Webanwendung soll verwendet werden (bekannt und etabliert)
- Soll Bestellungen komfortabel einsehen können
- Einfache Handhabung (Kunde ist kein Power-User)
- Muss Bestellhistorie (“komprimiert”) einsehen können

Für Dozent:

- Bisherige Webanwendung soll verwendet werden (bekannt und etabliert)
- Einfache Bestellung; intuitiv, schnell und anpassbar
- Bestellung erfolgt nur über Webanwendung (keine Mails mehr)

- Anzahl der Dozenten kann sich ändern

1.4. Produkt-/Lösungsüberblick

1.4.1. Bedarfe und Hauptfunktionen

Bedarf	Priorität	Features	Geplantes Release
Voreinstellungen	Hoch	Festlegung der Buchungsdetails vor Buchungsbeginn	xx
Warenkorb	Hoch	Experiment in Warenkorb legen und evtl. Reihenfolge anpassen	xx
Journal ansehen	Hoch	Einsehen vergangener und Bearbeiten aktueller Buchungen durch den Dozenten	xx
Dozentenwoche	Hoch	Einsehen der Buchungen je Wochentag und pro Dozent durch den Admin; Einleitung des Druckprozesses	xx
Statistik ansehen	Mittel	Statistik (z.B. als Top 10) der (bestellten) Experimente einsehen	xx
Verwaltung der Bilder	Niedrig	Bilder in Experimenten können ersetzt werden, Reihenfolge wird einbehalten.	xx

1.5. Zusätzliche Produktanforderungen

Anforderung	Priorität	Geplantes Release
Einfache Bedienbarkeit	Hoch	xx
Hilfestellung / Kundenfreundliche Dokumentation	Niedrig	xx

1.6. Aussicht für zukünftige Projekte

Einige Ideen des Teams wurden vom Themensteller abgelehnt. Da das Team I2 dennoch von der Notwendigkeit und dem Nutzen dieser skizzierten Funktionen für die Vorlesungsassistenz überzeugt ist, wollen wir sie nicht in in Vergessenheit geraten lassen. Zukünftige (studentische) Projektteams sollen davon profitieren können. Hier ist der Platz, sie festzuhalten.

Dienlich zum Verständnis ist bspw. das Lesen des Protokolles zu TS-M3 vom 07.01.2021 (Word-Dokument; Ansprechpartner: gregor.mueller@htw-dresden.de, tobias.karsch@htw-dresden.de)

- Auswahl des Studienganges bei jeder Buchung ist umständlich – das Problem wurde auf Später

vertagt, wir entwickeln die Lösung nach den besprochenen Wireframes (Stand: 31.01.2021)

- Ein "Dummy-Experiment" für **Freitextbuchungen** war eine vom Team favorisierte Lösung, welche aber nicht umgesetzt werden sollte (Stand: 31.01.2021)
- Statistik/Historie: Ein Date-Picker-Mechanismus sollte von uns im Deployment hinterlegt worden sein. Darauf kann evtl. bei einer Neugestaltung der Historien-Ansicht zurückgegriffen werden. Wir halten den bisherigen Entwurf als Aufgliederung in zwei Semester als verbesserungswürdig. (Stand: 31.01.2021)
- **Kopierfunktion:** Eine Kopierfunktion zur Reproduktion aller Buchungen eines vergangenen Semesters wurde gewünscht. Wir sahen hier einen Zielkonflikt und einigten uns mit allen Stakeholdern darauf, dass wir das bisher entworfene Buchungssystem entwickeln (vgl. **I03**). Sollte diese Funktion in der Zukunft erneut gefordert werden, so sind hier unsere Überlegungen dazu aufgelistet:

Agreement (Stand: 07.01.2021)	Probleme, die zu adressieren sind
<ul style="list-style-type: none">• Problem wurde von allen Parteien erkannt• Die Kopierfunktion entspringt der grundsätzlichen Motivation des Projektes: Den Dozenten soll die Arbeit im höchstmöglichen Umfang vereinfacht werden. Deshalb soll der einmal bestrittene Ablauf von Experimenten in einem Semester leicht wieder auf ein Neues anzuwenden sein.• Erkenntnis aus dem derzeitigen Projekt: Um Statistik zu erhalten, müssen Dozenten immer wieder neu buchen → muss den Dozenten klar gemacht werden• Dozent muss sich seine Woche drucken können → braucht auch Zugang zu SEINER Dozentenwoche (Journal als Historie, Dozentenwochen aber auch)	<ul style="list-style-type: none">• Datumsanpassung jeder kopierten Vorlesung ist erforderlich• Was ist beim Wechsel Winter- zu Sommersemester?• Je mehr kopiert werden soll (einzelne VL-Tage-Wochen-Semester), desto mehr Angaben muss der User wieder anpassen• Je einfacher die Kopierfunktion wird, desto mehr Daten fallen weg: MO der 06. um 11:10 wird zu MO• Wochentage ändern sich von Semester zu Semester! (Feiertage etc.) + Unterschiedlicher Fortschritt in den Vorlesungen

2. Use-Case Model: Experimenteverwaltung I2

2.1. Allgemeine Informationen

Die Anordnung der Use Cases entspricht der Priorisierung in der Entwicklung.

Der Begriff "User" in den Use Cases greift die vorher genannten Akteure auf und impliziert, dass der User sich mit einem der Rolle entsprechenden Account eingeloggt hat.

Jeder User hat einen eigenen persönlichen Account. Nur der Gast-User kann mehrfach verwendet werden.

Weiteres: tbd

2.2. Use-Case: Experiment buchen

2.2.1. Kurzbeschreibung

Buchungsablauf von Exp. vom Buchungsstart bis zum fertigen Eintrag in der DB

2.2.2. Kurzbeschreibung der Akteure

- Dozent

2.2.3. Vorbedingungen

- User muss eingeloggt sein
- User muss sich an einer beliebigen Stelle im EMS befinden

2.2.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn User auf "Buchung starten" klickt
2. Neue Seite mit Formular (Uhrzeit, Datum, Studiengang) → Vorauswahl für aktuellen Buchungsvorgang
3. Exp.-Übersicht wird geöffnet
4. Exp. können über Warenkorbsymbol zum aktuellen Buchungsvorgang hinzugefügt werden
5. Wird auf "Buchung abschließen" geklickt, wird man zum Checkout (neue Seite) weitergeleitet
6. Im Checkout werden die aktuellen Buchungen aufgezählt, mit Möglichkeit zur Löschung und Kommentierung jeder Buchung
7. Nach Klicken des Users auf "Buchung speichern", wird der User auf eine Buchungsbestätigungsseite geleitet
8. Der UC endet damit, dass der User die Buchungsbestätigung bestätigt

2.2.5. Alternative Abläufe

Abbruch des Buchungsvorgangs

Wenn der User ab Schritt 2 des Standardablaufes die Buchung abbricht, dann wird der Use Case beendet. Es wird keinen Eintrag in der DB geben. Selbiges ist im Checkout unter Schritt 6 möglich.

2.2.6. Unterabläufe (subflows)

--- keine Vorhanden ---

2.2.7. Wesentliche Szenarios

--- keine abweichenden Szenarios vom Standardablauf ---

2.2.8. Nachbedingungen

- DB-Eintrag muss initial angelegt werden

2.2.9. Besondere Anforderungen

- Buchungen müssen jederzeit möglich sein
- Daten müssen valide sein (Bspw.: Datumseingabe für Exp. darf nicht in der Vergangenheit liegen; Studiengang muss existieren)
- Gäste und der Admin dürfen keine Buchungen tätigen
- Buchungen müssen persistent gespeichert werden; Datenverlust muss ausgeschlossen werden
- Buchungseingaben müssen temporär gespeichert werden (um ungewollten Datenverlust zu vermeiden, bspw. durch Verbindungsabbruch)

2.3. Use-Case: Dozentenwoche ansehen

2.3.1. Kurzbeschreibung

Admin möchte sich Wochenplan (Übersicht von allen Exp. inkl. Details) für bestimmte KW anzeigen lassen

2.3.2. Kurzbeschreibung der Akteure

- Admin

2.3.3. Vorbedingungen

- User muss eingeloggt sein
- User muss Wochenplanansicht aufgerufen haben
- Es müssen gebuchte Exp. vorhanden sein - ansonsten Informationsmeldung

2.3.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn User die Wochenplanansicht aufruft
2. Wochenplanansicht wird dargestellt
3. KW muss ausgewählt werden
4. Filter (Dozent) kann angewendet werden
5. Use Case endet damit, dass die Ansicht generiert wird

2.3.5. Alternative Abläufe

--- keine alternativen Abläufe möglich ---

2.3.6. Unterabläufe (subflows)

--- keine Unterabläufe ---

2.3.7. Wesentliche Szenarios

--- keine Szenarios ---

2.3.8. Nachbedingungen

--- keine Nachbedingungen ---

2.3.9. Besondere Anforderungen

--- keine Nachbedingungen ---

2.4. Use-Case: Eigene Dozentenwoche ansehen

2.4.1. Kurzbeschreibung

Dozent möchte sich seinen Wochenplan (Übersicht von allen eigenen Exp. inkl. Details) für bestimmte KW anzeigen lassen

2.4.2. Kurzbeschreibung der Akteure

- Dozent

2.4.3. Vorbedingungen

- User muss eingeloggt sein
- User muss Wochenplanansicht aufgerufen haben

2.4.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn User die Wochenplanansicht aufruft
2. Wochenplanansicht wird dargestellt
3. KW muss ausgewählt werden
4. Use Case endet damit, dass die Ansicht generiert wird

2.4.5. Alternative Abläufe

--- keine alternativen Abläufe möglich ---

2.4.6. Unterabläufe (subflows)

--- keine Unterabläufe ---

2.4.7. Wesentliche Szenarios

--- keine Szenarios ---

2.4.8. Nachbedingungen

--- keine Nachbedingungen ---

2.4.9. Besondere Anforderungen

--- keine Nachbedingungen ---

2.5. Use-Case: Eigene Dozentenwoche drucken

2.5.1. Kurzbeschreibung

Dozent möchte seinen (vorgefilterten) Wochenplan (Übersicht von allen eigenen Exp. inkl. Details) für eine bestimmte KW ausdrucken

2.5.2. Kurzbeschreibung der Akteure

- Dozent

2.5.3. Vorbedingungen

- User muss eingeloggt sein
- User muss Wochenplanansicht aufgerufen haben

2.5.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn User die Wochenplanansicht drucken möchte
2. (Im Browser integrierte) Druckvorschau öffnet sich (weitere Funktionen sind vom Browser abhängig und deshalb nicht Teil unseres Systems)
3. Nach der Bestätigung wird der Druckauftrag erteilt
4. Use Case endet

2.5.5. Alternative Abläufe

Wochenplanansicht drucken

--- keine alternativen Abläufe möglich ---

2.5.6. Unterabläufe (subflows)

--- keine Unterabläufe ---

2.5.7. Wesentliche Szenarios

--- keine Szenarios ---

2.5.8. Nachbedingungen

--- keine Nachbedingungen ---

2.5.9. Besondere Anforderungen

--- keine besonderen Anforderungen ---

2.6. Use-Case: Dozentenwoche drucken

2.6.1. Kurzbeschreibung

Admin möchte den (vorgefilterten) Wochenplan (Übersicht von allen Exp. inkl. Details) für eine bestimmte KW ausdrucken

2.6.2. Kurzbeschreibung der Akteure

- Admin

2.6.3. Vorbedingungen

- User muss eingeloggt sein
- User muss Wochenplanansicht aufgerufen haben

2.6.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn User die Wochenplanansicht drucken möchte
2. (Im Browser integrierte) Druckvorschau öffnet sich (weitere Funktionen sind vom Browser abhängig und deshalb nicht Teil unseres Systems)
3. Nach der Bestätigung wird der Druckauftrag erteilt
4. Use Case endet

2.6.5. Alternative Abläufe

Wochenplanansicht drucken

--- keine alternativen Abläufe möglich ---

2.6.6. Unterabläufe (subflows)

--- keine Unterabläufe ---

2.6.7. Wesentliche Szenarios

--- keine Szenarios ---

2.6.8. Nachbedingungen

--- keine Nachbedingungen ---

2.6.9. Besondere Anforderungen

--- keine besonderen Anforderungen ---

2.7. Use-Case: Eigene Experimente im Buchungsjournal ansehen

2.7.1. Kurzbeschreibung

Dozent will seine gebuchten Exp. im Buchungsjournal ansehen.

2.7.2. Kurzbeschreibung der Akteure

- Dozent

2.7.3. Vorbedingungen

- User muss eingeloggt sein
- User muss auf Buchungsjournalseite sein

2.7.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn User Buchungsjournalseite aufruft
2. Möglichkeit die Ansicht zu filtern (Vorführungsdatum + Studiengang, Studiengang, Kalenerwoche) → siehe Unterablauf 1
3. Use Case endet mit Auflistung aller bisher getätigten Buchungen nach Studiengang und Vorführungsdatum

2.7.5. Alternative Abläufe

--- keine alternativen Abläufe ---

2.7.6. Unterabläufe (subflows)

Filtern

1. Filtern
 1. Filter auswählen
 2. Filter anwenden und die Seite wird automatisch neu geladen

2.7.7. Wesentliche Szenarios

--- keine wesentliche Szenarios ---

2.7.8. Nachbedingungen

--- keine Nachbedingungen ---

2.7.9. Besondere Anforderungen

- Der User darf nur **selbst** gebuchte Exp. einsehen
- Gäste und der Admin dürfen keine Einsicht in die jeweiligen Journale haben

2.8. Use-Case: Experimente im Buchungsjournal bearbeiten

2.8.1. Kurzbeschreibung

Dozent will seine gebuchten Exp. im Buchungsjournal bearbeiten.

2.8.2. Kurzbeschreibung der Akteure

- Dozent

2.8.3. Vorbedingungen

- User muss eingeloggt sein
- User muss auf Buchungsjournalseite sein

2.8.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn User Bearbeitungsformular aufruft
2. User kann Exp. ändern
3. User bestätigt Änderung
4. Use Case endet damit, dass die Seite neu geladen und die Änderung aktiv wird

2.8.5. Alternative Abläufe

--- keine alternative Abläufe ---

2.8.6. Unterabläufe (subflows)

--- keine Unterabläufe ---

2.8.7. Wesentliche Szenarios

Falsche Änderung

Änderung bei Wiederholung des Standardablaufes ab Schritt 1

2.8.8. Nachbedingungen

DB-Eintrag

DB-Eintrag wird aktualisiert und nicht neu angelegt oder gelöscht

2.8.9. Besondere Anforderungen

- Datumsbegrenzung: Nur zukünftige Exp. dürfen bearbeitet werden

- Der User darf nur selbst gebuchte Exp. editieren
- Gäste und der Admin darf keine Einsicht in die jeweiligen Journale haben
- Buchungsänderungen müssen temporär gespeichert werden (um ungewollten Datenverlust zu vermeiden, bspw. durch Verbindungsabbruch)

2.9. Use-Case: Eigene Experimente im Buchungsjournal löschen

2.9.1. Kurzbeschreibung

Dozent will seine gebuchten Exp. im Buchungsjournal löschen

2.9.2. Kurzbeschreibung der Akteure

- Dozent

2.9.3. Vorbedingungen

- User muss eingeloggt sein
- User muss auf Buchungsjournalseite sein

2.9.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn User eine Buchung löschen will
2. User wird nach Bestätigung gefragt
3. Use Case endet damit, dass die Seite neu geladen und die Löschung aktiv wird

2.9.5. Alternative Abläufe

--- keine alternativen Abläufe ---

2.9.6. Unterabläufe (subflows)

--- keine Unterabläufe ---

2.9.7. Wesentliche Szenarios

Falsche Buchung gelöscht

Neue Buchung gemäß Use Case "Experiment buchen"

2.9.8. Nachbedingungen

DB-Eintrag

DB-Eintrag wird aktualisiert und nicht neu angelegt oder gelöscht

2.9.9. Besondere Anforderungen

- Datumsbegrenzung: Nur zukünftige Exp. dürfen gelöscht werden
- Der User darf nur selbst gebuchte Exp. einsehen, editieren und löschen

- Gäste und der Admin darf keine Einsicht in die jeweiligen Journale haben
- Buchungsänderungen müssen temporär gespeichert werden (um ungewollten Datenverlust zu vermeiden, bspw. durch Verbindungsabbruch)

2.10. Use-Case: Statistik anzeigen

2.10.1. Kurzbeschreibung

Admin möchte sich eine Statistik über gebuchte Experimente anzeigen lassen

2.10.2. Kurzbeschreibung der Akteure

- Admin

2.10.3. Vorbedingungen

- User muss eingeloggt sein
- User muss sich auf Statistik-Seite befinden
- Es müssen gebuchte Exp. vorhanden sein - ansonsten Informationsmeldung

2.10.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, dass der User auf "Statistik" klickt
2. Zeitraum begrenzen (Optionen: Sommersemester, Wintersemester, Gesamt)
3. Auflistung der Exp., welche in dem angegebenen Zeitraum genutzt wurden (Standardansicht: Anzahl absteigend) → siehe Unterablauf 1
4. Use Case endet

2.10.5. Alternative Abläufe

Filtern

Nach Schritt 3 des Standardablaufes gibt es die Möglichkeit, die aktuelle Ansicht zu filtern (Anzahl oder Alphabet, auf- wie auch absteigend)

2.10.6. Unterabläufe (subflows)

Filtern

1. Filtern
 1. Filter auswählen
 2. Filter anwenden

2.10.7. Wesentliche Szenarios

--- keine Szenarios ---

2.10.8. Nachbedingungen

--- keine Nachbedingungen ---

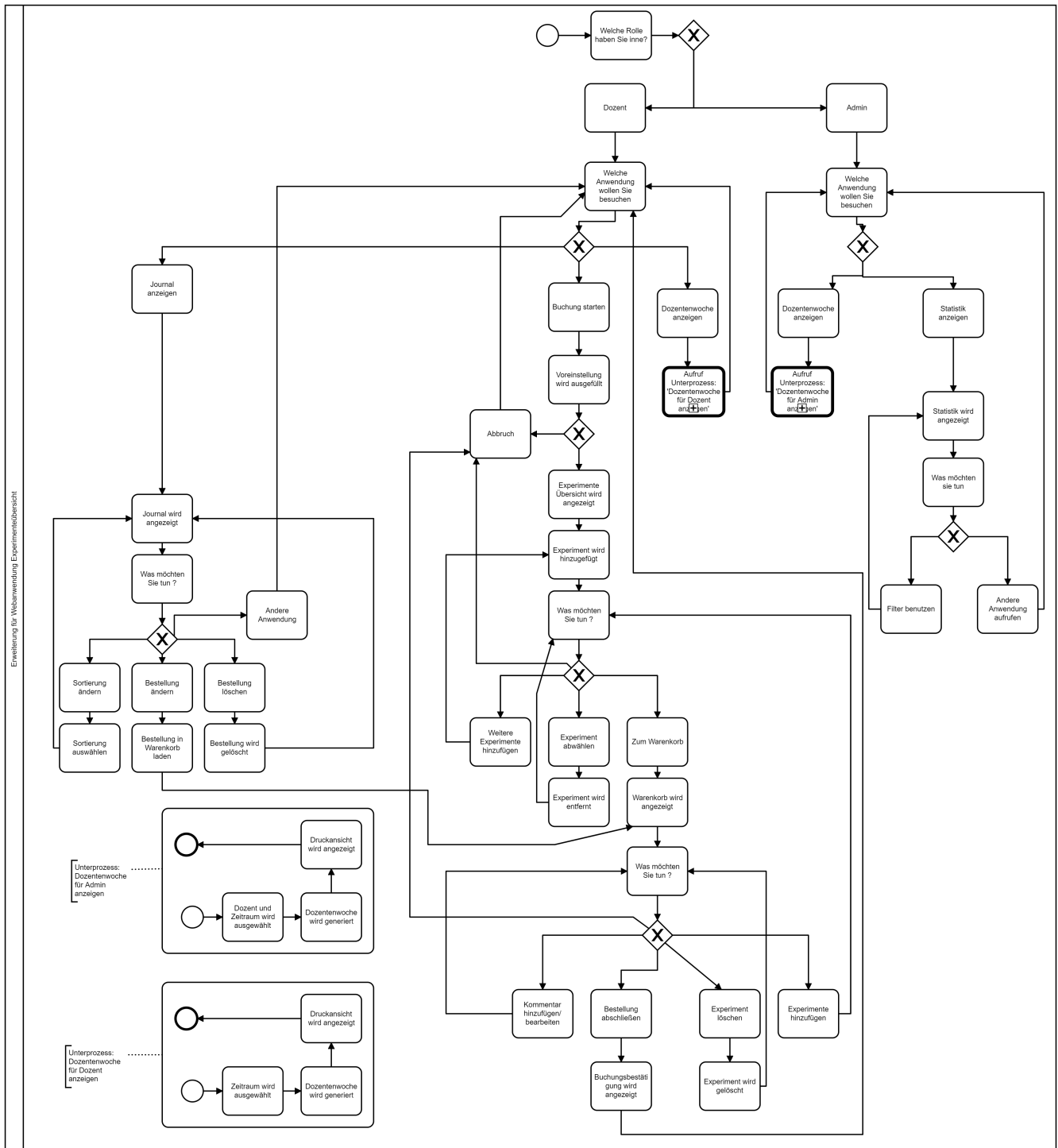
2.10.9. Besondere Anforderungen

- Nur der Admin darf die Statistik einsehen

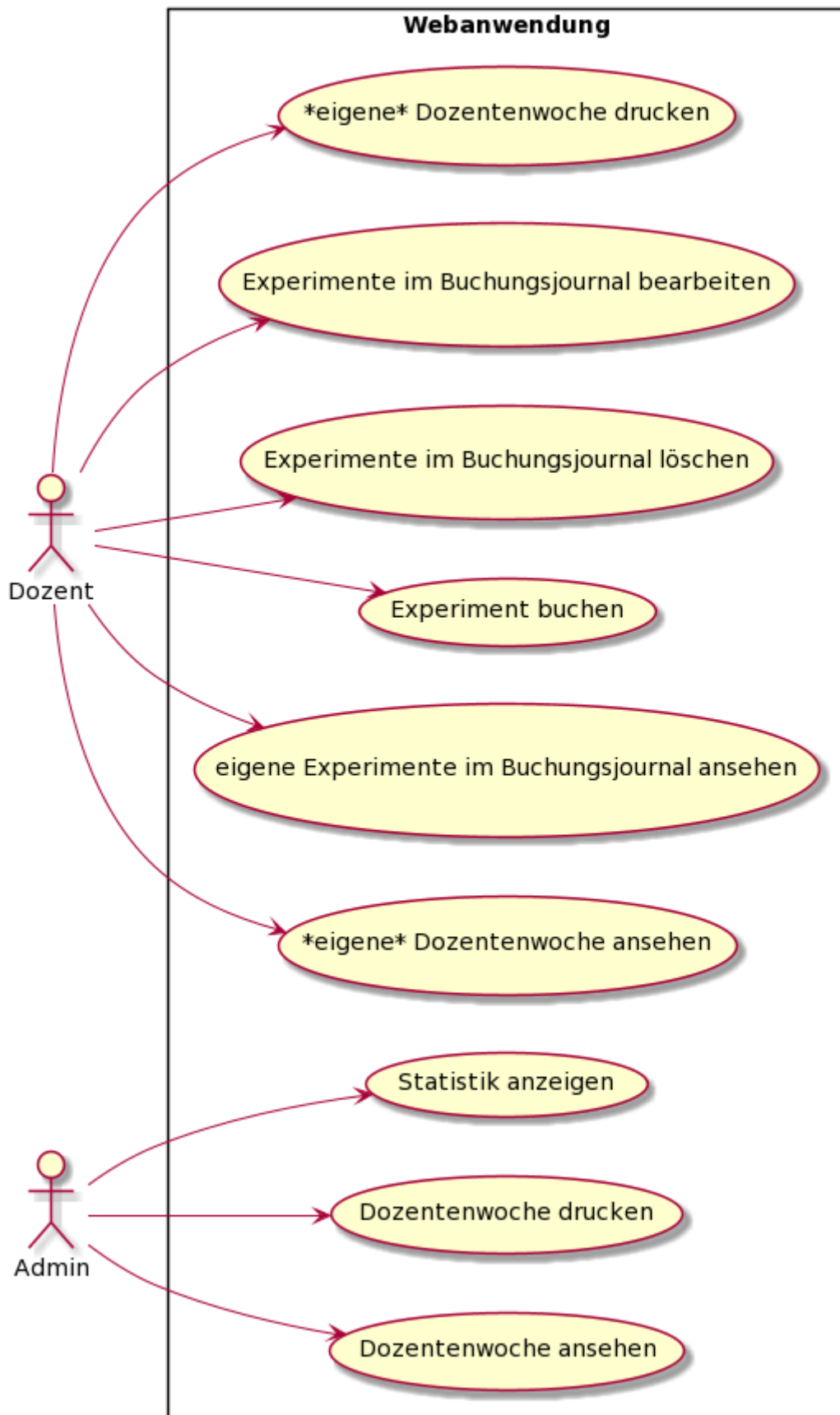
2.11. Grafische Veranschaulichung

Um eine einfache Übersicht über die Use Cases zu erlangen, wurde ein BPMN Diagramm erstellt das alle Use Cases darstellt und ein UML Diagramm um die einzelnen Beziehungen der User auf die Use Cases aufzuzeigen.

2.11.1. BPMN Diagramm für alles UCs



2.11.2. UML Diagramm für die Beziehungen der User



3. Wireframes Übersicht und Erklärungen: Experimenteverwaltung - Erweiterung Buchungssystem (I2)

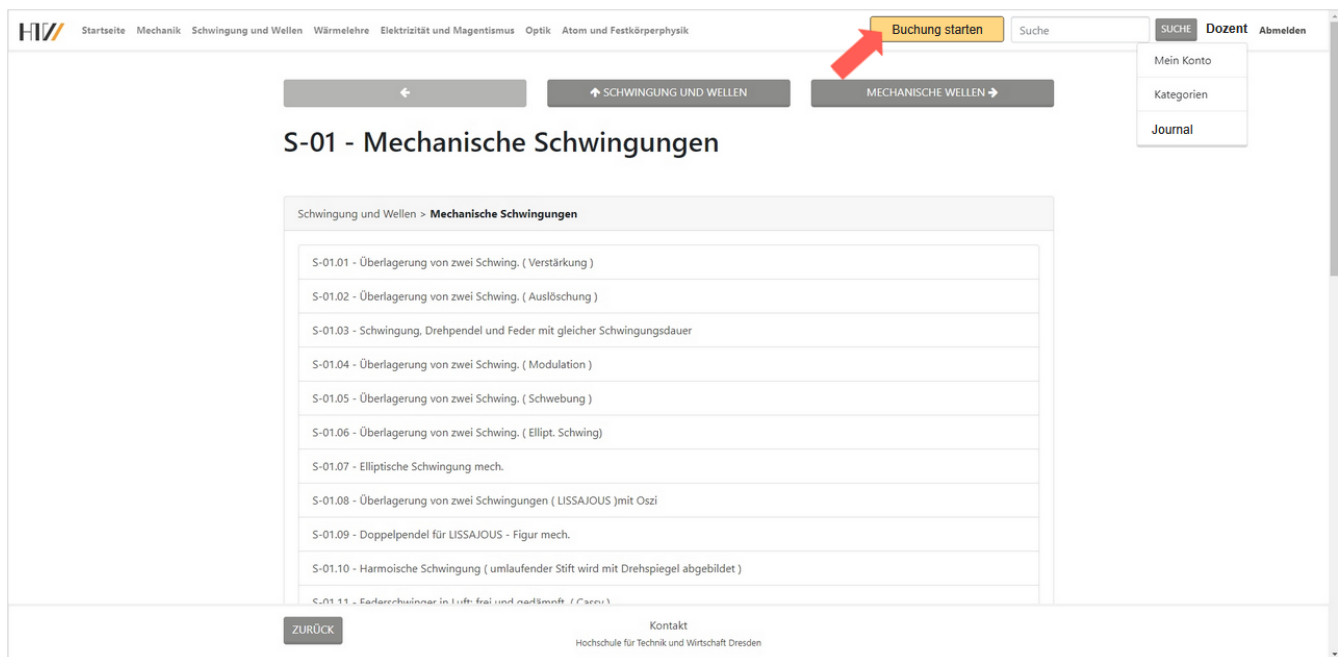
3.1. Wireframes

3.1.1. Erklärung

Für die einfache Darstellungen von Sachverhalten und der Nutzung unserer Erweiterung der Webanwendung haben wir uns für die Verwendung von Wireframes entschieden. Besonders bei der Darstellung des Buchungsvorganges und unserer Vorstellung zu den Benutzeroberflächen war uns die Nutzung besonders wichtig, da unser Kunde mit visuellen Darstellungen unsere Ideen besser nachvollziehen konnte.

Anbei nun eine Zusammenstellung der einzelnen Wireframes.

3.1.2. WF0: Buchung starten



Erklärung:

Hier wird die eigentliche Buchung gestartet. Dies geschieht mit einem einfachen Klick auf den "Buchung starten" Knopf.

3.1.3. WF1: Voreinstellungen

Erklärung:

Nun müssen die Voreinstellungen für die Buchung festgelegt werden. Dabei handelt es sich um Studiengang, Datum und Uhrzeit, welche für die gesamte Buchung benutzt werden sollen. Es kann weiter zu der Experimenteübersicht navigiert werden oder die Buchung abgebrochen werden.

3.1.4. WF2: Experimente wählen

Erklärung:

Mit einem Klick auf die Warenkorb Symbole werden diese eingefärbt (d.h. sie befinden sich im Warenkorb; Rückgängig machen mit erneutem Klick). Sind alle gewünschten Experimente ausgewählt, kann zum Buchungsabschluss navigiert werden oder abgebrochen werden

3.1.5. WF3: Checkout

[Startseite](#)
[Mechanik](#)
[Schwingung und Wellen](#)
[Wärmelehre](#)
[Elektrizität und Magnetismus](#)
[Optik](#)
[Atom und Festkörperphysik](#)

SUCHE

Dozent

Abmelden

Mein Konto

Kategorien

Journal

Warenkorb

für: W72- Wirtschaftsingenieurwesen, 01.01.2021 um 11:00 Uhr

Hinweis: Reihenfolge der Experimente kann durch Verschieben dieser verändert werden.

1. Experiment: S-01.01 - Überlagerung von zwei Schwing. (Verstärkung)

2. Experiment: S-01.02 - Überlagerung von zwei Schwing. (Auslöschung)

Kommentar:

Buchung abschließen

zurück zur Experimente-Übersicht

Buchung abbrechen

ZURÜCK

Kontakt
Hochschule für Technik und Wirtschaft Dresden

Erklärung:

Die Voreinstellungen und gewählten Experimente werden erneut aufgelistet. Hier kann die Reihenfolge der Durchführung dieser geändert werden (per Drag&Drop), das Experiment entfernt werden und ein Kommentar für die Buchung hinterlassen werden. Die Buchung kann dann abgeschlossen werden, Abgebrochen werden oder zurück zur Experimenteübersicht navigiert werden (um weitere Experimente zu buchen).

3.1.6. WF4: Buchung bestätigen

[Startseite](#)
[Mechanik](#)
[Schwingung und Wellen](#)
[Wärmelehre](#)
[Elektrizität und Magnetismus](#)
[Optik](#)
[Atom und Festkörperphysik](#)

SUCHE

Dozent

Abmelden

Mein Konto

Kategorien

Journal

Warenkorb

Ihre Buchung wurde gespeichert!

Ihre Buchungen für den Studiengang W72-Wirtschaftsingenieurwesen am 01.01.2021 um 11:00 Uhr wurde gespeichert.

Sie finden eine Übersicht Ihrer Buchungen im Journal.

zurück zur Hauptseite

ZURÜCK

Kontakt
Hochschule für Technik und Wirtschaft Dresden

Erklärung:

Bestätigung, das die Buchung erfolgreich war.

3.1.7. WF5: Statistik

[Startseite](#)
[Mechanik](#)
[Schwingung und Wellen](#)
[Wärmelehre](#)
[Elektrizität und Magnetismus](#)
[Optik](#)
[Atom und Festkörperphysik](#)

SUCHE

Admin

Abmelden

Statistik

Sortierung:

nach Nutzung

Sortierung:

Nach Nutzung

Alphabetisch

Zeitraum von:

01/01/2020

bis

01/01/2021

Name Experiment	Wintersemester 2020	Sommersemester 2021	Gesamtnutzung
S-01.01: Überlagerung von zwei Schwing. (Verstärkung)	25	15	40
S-01.02: Überlagerung von zwei Schwing. (Auslöschung)	22	12	34
S-01.03: Schwingung Drehpendel und Feder mit gleicher Schwingungsdauer	19	9	28
S-01.04: Überlagerung von zwei Schwing. (Modulation)	15	4	9

ZURÜCK

Kontakt

Hochschule für Technik und Wirtschaft Dresden

Erklärung:

Anzeigen einer passenden Statistik für die Nutzung der einzelnen Experimente, aufgegliedert nach Sommer- und Wintersemester und der Gesamtnutzung. Eine Sortierung nach Alphabet, Nutzung und Zeitraum kann vorgenommen werden.

3.1.8. WF6: Journal

[Startseite](#)
[Mechanik](#)
[Schwingung und Wellen](#)
[Wärmelehre](#)
[Elektrizität und Magnetismus](#)
[Optik](#)
[Atom und Festkörperphysik](#)

SUCHE

Dozent

Abmelden

Journal

Filtern nach:

Auswählen

Buchungen für: B11 - Bauingenieurwesen am 12.01.2021 um 11:00 Uhr

1

S-01.01 - Überlagerung von zwei Schwing (Verstärkung)

2

S-01.02 - Überlagerung von zwei Schwing (Auslöschung)

3

S-01.03 - Schwingung Drehpendel und Feder m. Gleicher Schwingungsdauer

Kommentar zur Buchung:

bitte selber durchführen lassen

Buchungen für: W72 - Wirtschaftsingenieurwesen am 13.01.2021 um 12:30 Uhr

1

S-01.01 - Überlagerung von zwei Schwing (Verstärkung)

2

S-01.02 - Überlagerung von zwei Schwing (Auslöschung)

3

S-01.03 - Schwingung Drehpendel und Feder m. Gleicher Schwingungsdauer

Kommentar zur Buchung:

ZURÜCK

Kontakt

Hochschule für Technik und Wirtschaft Dresden

Erklärung:

Im Journal können die getätigten Buchungen eingesehen werden. Auch hier sind verschiedene Sortierungen möglich (z.B. nach Studiengang, Tag, ...). Die Möglichkeit einer Löschung und Änderung besteht hier ebenfalls.

28

3.1.9. WF7: Dozentenwoche aufrufen

Erklärung:

Über die Dozentenwoche kann der gesamte Wocheplan vom Admin (oder Dozent für sich selber) abgerufen werden. Mit der Einstellung des Gewünschten Dozents und dem Zeitraum lassen sich Einschränkungen vornehmen.

3.1.10. WF8: Dozentenwoche

KW 48 | 23.11.2020 - 29.11.2020

Frau Professor Dr. Krawietz			
Bauingenieure 12/27 Mo 7:30 Uhr 23.11.2020	Wirtschaftsingenieure 10/21 Mo 13:20 Uhr 23.11.2020	Bauingenieure 13/27 Mi 11:10 Uhr 25.11.2020	Wirtschaftsingenieure 11/21 Fr 11:10 Uhr 27.11.2020
1. E-02.03 Stromabhängigkeit von Leiterlänge und	1. M-07.04 Drehung um eine feste Achse, Drehimpuls	1. S-02.36 Geometrische Pegel-abnahme	1. M-07.22 Kreisel in kardanischer Aufhängung
2. E-02.04 spezifischer Widerstand verschiedene	2. M-07.10 Drehschemelversuch – Änderung des	2. S-02.37 Dissipation im Nebel	2. M-07.18 Modell Nutations-bewegung
3. E-02.02 Reihenschaltung von Widerständen, M	3. M-07.11 Drehschemelversuch – Drehimpuls vor		3. M-07.19 Erzwungene Präzession, Kreisel unter
4. E-02.01 Parallelschaltung von Widerständen, M	4. M-07.12 Drehschemelversuch – Drehimpuls vor		4. M-07.25 Präzessionsbewegung eines Fahrradk
	5. M-07.13 Stabilität freier Achsen, Hauptträgheits		5. M-07.20 Kreisel auf Spitz
	6. M-07.14 Hauptträgheitsachsen – Quader am Sti		6. M-07.27 Fahrradmodell
	7. M-07.15 Hauptträgheitsachsen – Scheibe am St		7. M-07.21 Hauptträgheitsachsen – Quader am Sti
	8. M-07.16 Drehung eines außerhalb des Schwer		8. M-07.21 Reihenschaltung von Widerständen, M
	9. M-07.17 Modell Nutations-bewegung		
	10. M-07.18 Parallelschaltung von Widerständen, M		
	11. M-07.19 Hauptträgheitsachsen – Quader am Sta		
	12. M-07.20 Reihenschaltung von Widerständen, M		
	13. M-07.21 Erzwungene Präzession, Kreisel unter Z		
	14. M-07.23 Fahrradmodell		
	15. M-07.24 Dissipation im Nebel		

Kommentar: für Montag Wirtschaftsingenieure, Frisby wird bei Experiment 5. mitgebracht, für Fr Wirtschaftsingenieure, vor 1. Kleine Spielzeugkreisel zeigen, inklusive „Stehaufkreisel“, zu 1. kleiner Kreisel, Model, zwischen 2. und 3. Fidget Spinner, bringe ich mit, zu 5. Fahrradfelge im Rahmen

Erklärung:

Ansicht des beispielhaften Wochenplans.

4. System-Wide Requirements: Experimenteverwaltung - Erweiterung Buchungssystem (I2)

4.1. Einführung

In diesem Dokument werden die systemweiten Anforderungen für das Projekt Experimenteverwaltung - Erweiterung Buchungssystem (I2) spezifiziert. Die Gliederung erfolgt nach der FURPS+ Anforderungsklassifikation:

- Systemweite funktionale Anforderungen (F),
- Qualitätsanforderungen für Benutzbarkeit, Zuverlässigkeit, Effizienz und Wartbarkeit (URPS) sowie
- Zusätzliche Anforderungen (+) für technische, rechtliche, organisatorische Randbedingungen



Die funktionalen Anforderungen, die sich aus der Interaktion von Nutzern mit dem System ergeben, sind als Use Cases in einem separaten Dokument festgehalten. [[Use Case Model](#)]



"To write a good requirement, you must write it as a complete sentence, with a subject and a predicate (usually a verb). The subject is an Actor, a stakeholder, the system under development, or a design entity that is related to the requirement. The predicate specifies a condition, action, or intended result that is done for, by, with, or to the subject." - (OpenUP; siehe <https://www2.htw-dresden.de/~anke/openup>)



Unsere Anforderungen sollen den Projekterfolg garantieren. Je gewissenhafter wir sie ausarbeiten und pflegen, desto besser wird das Endprodukt.



Aussagen des ESSENCE Kernel: Opportunity focuses on Requirements. Requirements set scopes & constraints for work.



Anforderungsanalyse ist Teil der Solution, damit essentieller Teil der Lösung. Stakeholder haben Anforderungen. Ziele der Stakeholder sind Teil der Anforderungsspezifikation!

4.2. Systemweite funktionale Anforderungen

- **SWFA-1:** Das System muss Zugriffsbeschränkungen gewährleisten, da nicht jeder Anwender jeden Use Case ausführen darf. Diese Zugriffsbeschränkungen erlauben es dem Anwender, nur für ihn vorgesehene Funktionen auszuführen und nur auf für ihn zugängliche Daten zuzugreifen. Dafür ist eine Zuordnung aller Anwender zu Rollen und einem Login notwendig.

- Zu Überprüfungszwecken wird anhand von Test-Logins versucht auf fremde Daten und Rollenfunktionalitäten zuzugreifen.
- **SWFA-2:** Das System muss Buchungs- und Anwenderdaten persistent speichern.
 - Überprüfbar wird dies durch Einfügen von Daten. Nach einem Neustart des Systems ist es notwendig, dass die Daten weiterhin vollständig sind.
- **SWFA-3:** Das System muss Zugangsdaten der Anwender vor Diebstahl schützen.
 - Überprüfbar wird dies durch einen Versuch, Testzugangsdaten auszulesen.

4.3. Qualitätsanforderungen für das Gesamtsystem

4.3.1. Benutzbarkeit (Usability)

- **NFAU-1:** Alle Prozesse und Datenausgaben müssen übersichtlich, eindeutig und verständlich sein.
 - **NFAU-1.1:** Die Anzahl der Interaktionselemente soll komprimiert sein.
 - Es soll jeweils nur eine Schaltfläche pro Aktion geben (Bsp.: Warenkorb-Icon zum Buchen). Zu Überprüfungszwecken werden alle Interaktionselemente in ihrer Funktion verglichen.
 - **NFAU-1.2:** Die Auswahlmenüs zur Detailfestlegung müssen verständlich und inhaltlich komprimiert sein.
 - Es soll vorgefertigte Drop-Down-Menüs und Widgets zur Auswahl aus von festgelegten Wertebereichen geben. Zur Überprüfung wird getestet, ob invalide Eingaben möglich sind.
 - **NFAU-1.3:** Die Anwendung (inkl. Änderung) muss einfach erfassbar und nutzbar sein.
 - Es soll ausschließlich deutsche Sprache verwendet werden. Auf Anglizismen und Fachterminologie wird verzichtet. Große informatische und technische Kenntnisse dürfen nicht zur Bedienung erforderlich sein. Zur Überprüfung wird eine themenfremde Testperson die Anwendung allgemein nutzen, wobei auf externe Hilfe verzichtet werden muss.
 - **NFAU-1.4:** Die Möglichkeiten für Fehlentscheidungen oder Orientierungslosigkeit sollen minimiert werden.
 - Die Dozenten sollen für die Buchung eines Experiments maximal 15 Klicks (von Anmeldung bis „Buchungsabschluss“) benötigen. Temporär nicht verfügbare Interaktionsmöglichkeiten sollen ausgegraut werden. Zu Überprüfungszwecken wird ein Testdurchlauf stattfinden, bei dem die Klicks gezählt werden und darauf geachtet wird, ob es möglich ist, nicht erreichbare Funktionalitäten aufzurufen.
 - **NFAU-1.5:** Texte und Schaltflächen sollen auf ca. 50cm Entfernung leicht lesbar und erkennbar sein.
 - Text muss eine passende Größe und augenschonende, sowie leserliche Farbe aufweisen.
 - Schaltflächen und Symbole müssen von guter grafischer Qualität und scharf sein.
 - Zur Überprüfung wird eine Testperson die Anwendung mit einem Abstand von 50cm

Entfernung zum Bildschirm nutzen.

- **NFAU-2:** Die Eingabemöglichkeiten sollen nicht als Pop-ups erscheinen.
 - Zur Überprüfung wird die Anwendung auf das Vorhandensein von Pop-up-Fenstern getestet.

4.3.2. Zuverlässigkeit (Reliability)

- **NFAR-1:** Das System darf aufgrund von Netzwerkausfällen keinen Schaden nehmen.
 - Testweise wird beim Buchen ein Verbindungsabbruch erzwungen.

4.3.3. Effizienz (Performance)

- **NFAP-1:** Die Nutzbarkeit muss gewährleistet sein, wenn alle derzeitigen Anwender gleichzeitig arbeiten wollen.
 - Überprüft wird dies durch einen Stresstest mit 10 Nutzern.

4.3.4. Wartbarkeit (Supportability)

- **NFAS-1:** Das Entwicklerteam muss eine (fortgeführte) Entwicklerdokumentation bereitstellen, welche Wartung und Weiterentwicklung des Systems ermöglicht.
- **NFAS-2:** Die Software muss auf folgenden Browsern funktionieren:
 - Edge
 - Internet Explorer (Version 11)
 - Chrome
 - Firefox
 - Safari
 - Zur Überprüfung muss die Anwendung in allen Browsern mit der aktuellen Version (wenn nicht anders spezifiziert), getestet werden.

4.4. Zusätzliche Anforderungen

4.4.1. Einschränkungen

- Die Verwaltung des Sourcecodes und der Dokumentation muss mittels **GitHub** erfolgen.
- **Ruby on Rails** stellt das hauptsächlich zu nutzende Framework dar.
- Folgende Programmiersprachen sind weiterhin vorgegeben:
 - **Ruby**
 - HTML/ Slim
 - Sass
 - JavaScript

4.4.2. Interface Requirements

- Die Benutzeroberfläche soll gängige Symbole, wie beispielsweise einen Einkaufswagen für den Warenkorb, verwenden.
- Deutsche Bezeichnungen sollen reichen (siehe Usability).
- Das Design soll an der bereits bestehenden Anwendung orientiert sein.
- Eine offene Menüführung muss gewährleistet sein. Interaktionselemente und Links sollen nicht unnötig in Menüs versteckt sein.

4.4.3. Rechtliche Anforderungen

- Der Kunde muss über eine mögliche Speicherung von Nutzungsdaten informiert werden.
- Eine etwaige Datenschutzerklärung muss beim Login vorliegen und vor der Nutzung durch Dozenten anerkannt werden. Der Verantwortliche hierbei sollte der Admin sein.
- Keine fremden Werke dürfen fälschlicherweise als Eigene ausgewiesen werden. Urheberrecht ist zu beachten, gegebenenfalls müssen eigene Grafiken erstellt werden.

5. Glossar: Experimenteverwaltung I2

5.1. Einführung

In diesem Dokument werden die wesentlichen Begriffe aus dem Anwendungsgebiet (Fachdomäne) des EMS definiert. Zur besseren Übersichtlichkeit sind Begriffe, Abkürzungen und Datendefinitionen gesondert aufgeführt.

5.1.1. Begriffe

Term	Definition/ Bedeutung	Synonym
Admin	<ul style="list-style-type: none">• Gleichzusetzten mit "Kunde"	
Bestellung	<ul style="list-style-type: none">• Anforderung eines Versuchsaufbaus mit Durchführung zu einer festgelegten Zeit	
Bestelldetails	<ul style="list-style-type: none">• Bei Buchung eines konkreten Experimentes anzugebende Informationen• enthält: Studiengang, Datum, Zeit, Kommentar, Anmeldedaten (siehe 1.3 Datenstrukturen)	
Buchungsschaltfläche	<ul style="list-style-type: none">• Schaltfläche, die dem Nutzer per Klick die Buchung eines Experiments ermöglicht• Dargestellt durch Warenkorb-Symbol	<ul style="list-style-type: none">• Warenkorbicon• Warenkorbsymbol
Deadline	<ul style="list-style-type: none">• Englisch für Grenze/Frist• Datum, bis zu dem Buchungen geändert/stoniert werden können	
Dozent	<ul style="list-style-type: none">• Nutzer des Buchungssystems• "Kunde des Kunden"	

Term	Definition/ Bedeutung	Synonym
Dozentenwoche	<ul style="list-style-type: none"> • Darstellung aller für eine Woche durch einen bestimmten Dozenten gebuchten Experimente • "Wochenplan pro Dozent" • Übersicht/Druckansicht für Admin 	<ul style="list-style-type: none"> • Wochenplan • Wochenplanansicht
Frontend-Benutzer	<ul style="list-style-type: none"> • Nutzer der öffentlich erreichbaren Weboberfläche 	<ul style="list-style-type: none"> • User • Nutzer • Anwender
Historie	<ul style="list-style-type: none"> • Häufigkeit der vergangenen Buchungen pro Experiment 	<ul style="list-style-type: none"> • Anzahl der Buchungen pro Experiment • Statistik • Auswertung
Journal	<ul style="list-style-type: none"> • Übersicht der eigenen Buchungen mit Anpassungsmöglichkeiten für Dozenten 	
Karteikarte	<ul style="list-style-type: none"> • analoges Abbild der Versuchs-/ Experimentdetails als Alternative zum Tablet/ Webbrowser 	<ul style="list-style-type: none"> • Versuchsübersicht
Kunde	<ul style="list-style-type: none"> • Auftraggeber des Projekts, vorrangiger Stakeholder • Betreiber des Verwaltungssystem • Hr. Heisig 	<ul style="list-style-type: none"> • Themensteller
Open Unified Process	<ul style="list-style-type: none"> • Vorgehensmodell • siehe https://www2.htw-dresden.de/~anke/openup/index.htm • <i>OpenUP is a lean Unified Process that applies iterative and incremental approaches within a structured lifecycle</i> 	<ul style="list-style-type: none"> • Vorgehendmodell • OpenUP
Rolle	<ul style="list-style-type: none"> • Bezeichnung für Zuordnung der Zugriffsrechte (siehe "Dozent", "Admin") 	

Term	Definition/ Bedeutung	Synonym
Template	<ul style="list-style-type: none"> • Englisch für Vorlage/Schablone 	
Virtuelle Maschine	<ul style="list-style-type: none"> • Softwaretechnische Kapselung eines Rechnersystems innerhalb eines lauffähigen Rechnersystems 	<ul style="list-style-type: none"> • Testbereich des Systems • Abkürzung: VM • Betreuung durch TAVM
Voreinstellungen	<ul style="list-style-type: none"> • Festlegung der Bestelldetails (ohne Anmeldedaten) vor Auswahl der Buchungen 	
Warenkorb	<ul style="list-style-type: none"> • Temporärer Sammelpeicher für bestellte Experimente pro Dozent • Bestellübersicht mit Datums- und Zeitauswahl 	<ul style="list-style-type: none"> • Bestellzusammenfassung • Checkout

5.1.2. Abkürzungen und Akronyme

Abkürzung	Bedeutung	Erläuterung
AQM	Ansprechpartner QM	alena.bischoff@htw-dresden.de
DB	Datenbank	Gemeint ist die Datenbank, in welche Buchungseinträge geschrieben werden.
EMS	Das weiterzuentwickelnde Projekt	https://mphyda.mw.htw-dresden.de/
Exp.	Experiment	
KW	Kalenderwoche	-
<ul style="list-style-type: none"> • LCO • LCA • IOC • PRM 	<ul style="list-style-type: none"> • Life Cycle Objective Milestone • Life Cycle Architecture Milestone • Initial Operative Capability Milestone • Product Release Milestone 	Meilensteine des OpenUP. Siehe Meilensteine und Ziele im Projektplan
OpenUP	Open Unified Process	siehe Term <i>Open Unified Process</i>
PM	Projektmanager	Als Singular oder Plural zu verwenden
QM	Qualitätsmanagement	Interne Sicherung der Qualität in Zusammenarbeit und im Ergebnis
TAVM	Technischer Ansprechpartner für VM	hornoff@htw-dresden.de Kontaktverantwortlicher: Richard Böhme

Abkürzung	Bedeutung	Erläuterung
TS	Themensteller	Siehe Term <i>Kunde</i>
TS-Mx	Themensteller-Meeting Nr. x	Bezeichnung für ein Treffen mit dem Themensteller, fortlaufende Nummerierung
VM	Virtuelle Maschine	siehe Term <i>Virtuelle Maschine</i> in <i>Begriffe</i>

5.1.3. Verzeichnis der Datenstrukturen

Bezeichnung	Defnition	Format	Gültigkeitsregeln	Aliase
Anmeldedaten	Zusammensetzung von Benutzernamen und Passwort	String	E-Mail-Adresse muss @-Zeichen, Punkt und "htw-dresden"-Domaine enthalten	Login

6. Domain Model: Experimenteverwaltung - Erweiterung Buchungssystem (I2)

6.1. Domänenmodell für die Erweiterung des EMS

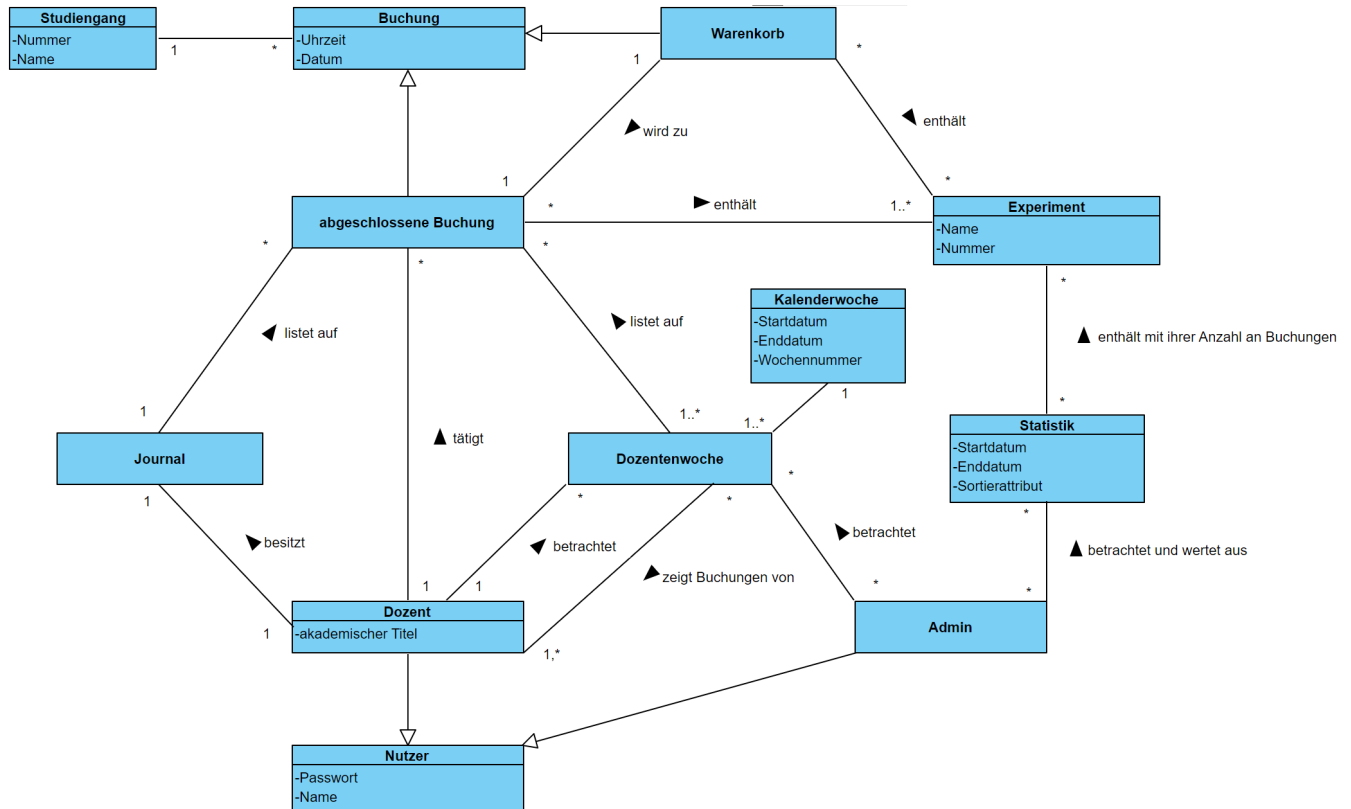


Abbildung 1. zuletzt aktualisiert am 28.01.2021

Projektdokumentation

- Projektplan
- Risikoliste
- Iteration Plan (für zwei ausgewählte Iterationen)

7. Projektplan: Experimenteverwaltung - Erweiterung Buchungssystem (I2)

7.1. Einführung

Wir (Team I2) orientieren uns bei unserem Projekt am Vorgehensmodell Open Unified Process. Das Projekt umfasst die vier Phasen:

- Inception
- Elaboration
- Construction
- Transition

Der Projektplan wird fortlaufend ergänzt und verfeinert. Verantwortlich zeichnen die Project Manager.

7.2. Projektorganisation

[Rollenübersicht](#)

7.3. Praktiken und Bewertung

7.3.1. Management practices

[Management Practices OpenUP](#)

- **Iterative Development:** Plan Iteration - Manage Iteration - Assess Results

Das Team führt 14-tägige Iterationen durch. Wir nutzen ein modifiziertes [Scrum-Board](#) auf GitHub für Planung und Steuerung jeder Iteration. Iterationen werden in ALLEN Phasen des Projektes generell mit der Klassifizierung I1, I2, ... I[n] benannt. Dafür haben wir uns entschieden, da die OpenUP-Meilensteine nicht zwangsweise am Ende einer Iteration, sondern während der Arbeit erreicht wurden, was eine Aufteilung in I-, E-, C- und T-Phasen erschwerte. Regeln zum Umgang mit den Aufgaben (Issues) und mit Pull Requests sind im [Wiki](#) festgehalten. Der Fortschritt einer jeden Iteration bildet sich durch das "Wandern" der Aufgaben von der Spalte "Work Item List" in die Spalte "DONE" ab. Zu Beginn einer jeden Iteration wird die Work Item List wieder gefüllt. Der globale Projektfortschritt wird außerdem mithilfe des ESSENCE Navigator Tools sowie durch die Definition und Messung zu erreichender OpenUP-Meilensteine veranschaulicht.

Das Ende einer jeden Iteration wird mit einer **Review** und einer **Retrospective** abgeschlossen. Dabei ist das gesamte Team mit einzubeziehen!

- Wöchentliche Besprechungen (**Weekly**) sind obligatorisch.
 - Dabei hat sich die Moderation durch einen zuvor bestimmten Redeführer etabliert

- Die Agenda wird im Voraus von den PM entworfen und der Gruppe zugesandt
- Ein "Weekly Scrum" wird zu Beginn eines jeden Weekly durchgeführt
 - Dauer insgesamt max. 10 min
 - Jedes Teammitglied erläutert knapp woran er/sie gerade arbeitet bzw. in der vergangenen Woche gearbeitet hat
 - Jedes Teammitglied berichtet, was sie/ihn aktuell an der Arbeit behindert
 - Äußerungen werden nicht kommentiert
- **Team Change Management:** Diese OpenUP-Praktik wird durch unser [Scrum-Board](#) auf GitHub realisiert. Änderungsanfragen am Scope für diese/folgende Iteration(en) sowie Change Requests an Work Items werden durch das Anlegen eines neuen Issues mit entsprechender Beschreibung nach den Regeln im [Wiki](#) vorgenommen. Die PM müssen zeitnah darauf reagieren.
- **Risk-Value Life Cycle:** Im OpenUP ermöglicht diese Praxis das systematische Erkennen und Reagieren auf Risiken. Uns dient sie in erster Linie zur internen Bewertung des Projektfortschritts. Die definierten Meilensteine - LCO, LCA, IOC, PRM - werden in jeder Iteration Review überprüft.

Die Position der Stakeholder wird dabei zu gleichen Teilen, aber auf indirekte Art und Weise berücksichtigt. Die Bewertung der Risiken und dem bisherigen Wertzuwachs - kurzum des Projektzustands - wird in den Stakeholdermeetings erfragt. Die Erläuterung der OpenUP-Praktiken und -Begrifflichkeiten gegenüber dem Themensteller erfolgt jedoch aufgrund der begrenzten Zeit nicht. Die Meetings sollen stattdessen noch stärker dem Wertzuwachs am eigentlichen Produkt dienen.

- **Whole Team:**

- Workshops für teaminternen Wissensaustausch werden ad-hoc durchgeführt (z.B. für die Arbeit in GitHub und Ruby). Alle Teammitglieder sollen vom Wissen der Anderen profitieren können
 - Der erste Workshop zum Thema "GitHub" wurde am 21.12.2020 von richard.boehme@htw-dresden.de, taejun.kim@htw-dresden.de und gregor.mueller@htw-dresden.de durchgeführt (Siehe I2)
- Konventionen und Anleitungen werden im GitHub-Wiki geteilt: [Wiki](#)
- Das **Qualitätsmanagement** wird durch die Verantwortliche alena.bischoff@htw-dresden.de gestaltet und durchgeführt

7.3.2. Technical Practices

Technical Practices OpenUP

- **Shared Vision:** Diese OpenUP-Praktik verbessert die Akzeptanz des späteren Produktes durch alle Stakeholder. In unserem Fall strebten wir die Übereinstimmung über die Vision und die damit beschriebene Opportunity gemeinsam mit dem Themensteller an. Das Team konnte diese Übereinstimmung in Iteration I03 erreichen.
- **Use Case Driven Development:** OpenUP empfiehlt die Zugrundelegung eines **Use Case Models** sowie von **System-Wide Requirements** zur Beschreibung, Gliederung und Auswertung der

erhobenen Anforderungen. Durch die Nutzung dieser Praktik versprechen wir uns eine effizientere Umsetzung und ein höheres Erfolgspotential des gesamten Projektes.

- **Evolutionary Architecture:** Das Architecture Notebook wird in Abhängigkeit des Projektzustandes bearbeitet. Bei dieser schrittweisen Ausarbeitung und Verfeinerung der Architektur wird darauf geachtet, die identifizierten Risiken zu berücksichtigen und, wenn möglich, mit den Architekturmechanismen adäquat auf sie zu reagieren.

7.4. Meilensteine und Ziele

Iteration	Primary objectives (risks and use case scenarios)	Scheduled start or milestone
I01	<ul style="list-style-type: none"> • Gruppenfindung, Team kennenlernen • Rollenverteilung, Namen der Projektmanager übermitteln • Aufgabe kennenlernen • Kontakt zu Themensteller aufnehmen • TS-M1 • Kontakt zu Coach 	16.11.2020-16.12.2020
I02	Iterationsplan 02	17.12.2020-31.12.2020
I03	Iterationsplan 03	07.01.2021-21.01.2021
I04	<ul style="list-style-type: none"> • Domänenmodell • Überarbeitung der Use Cases • Überarbeitung der Test Cases • Anfertigung des Beleges für das Modul: Überprüfung der Dokumente (siehe QM) • Ausscheiden der Teilnehmer juliusheiko.schmidt@htw-dresden.de, tim.wuerzburg@htw-dresden.de, tobias.karsch@htw-dresden.de aufgrund Beendigung des Moduls I925 Requirements Engineering 	22.01.2021-04.02.2021

7.4.1. Bemerkungen zum weiteren Projektverlauf

- Im Team herrscht Einigkeit darüber, dass Serveroptimierung nicht in unseren Aufgabenbereich fällt. Vom Vorhandensein ausreichender technischer Ressourcen und der Wartung des Fakultätsservers durch die Verantwortlichen außerhalb des Projektes wird ausgegangen.
- In das Benutzerhandbuch ist eine Anleitung zum Systemneustart zu integrieren

Bei der Planung zukünftiger Iterationen und ihrer Ziele sind diese Punkte zu verwirklichen.

7.5. Deployment

- Übertragung des Systems von der VM auf den regulären Fakultätsserver durch TAVM

7.6. Erkenntnisse (Lessons learned)

- Kommunikation mit den Stakeholdern ist der Knackpunkt für die Anforderungsanalyse, das tiefgründige gegenseitige Verständnis initial der wichtigste Faktor.
- Bestimmung eines Redeführers und Vorbereitung der Tagesordnung machen Meetings deutlich zeiteffizienter.
- Wissensweitergabe im Team lässt sich über selbstorganisierte Workshops sehr gut realisieren - von der Erfahrung einzelner Teammitglieder können so alle profitieren.

8. Risikoliste: Experimenteverwaltung - Erweiterung Buchungssystem (I2)

In diesem Dokument sind die wesentlichen Risiken des Projekts aufgeführt.

8.1. Attribute

Dabei werden folgende Attribute verwendet:

- **Typ:** Ressourcen, Geschäftlich, Technisch, Zeitlich
- **Auswirkung (IMP):** Wert zwischen 1 (niedrig) und 5 (hoch), der die Auswirkungen auf das Projekt angibt, wenn das Risiko eintritt
- **Wahrscheinlichkeit (PRB):** Prozentangabe für die Eintrittswahrscheinlichkeit des Risikos
- **Stärke (MAG):** Produkt aus Auswirkung und Wahrscheinlichkeit (damit kann die Liste sortiert werden)

8.2. Risiken

Die Risiken sind in folgender Tabelle: [Tabelle 1, "Risiken"](#) dargestellt. Das Datum des Dokuments (oben) gibt an, wann die Risikoliste zuletzt aktualisiert wurde.

Tabelle 1. Risiken

Risk ID	Date Identified	Headline	Description	Type	IMP	PRB	MAG	Owner	Mitigation Strategy
1	01.12.2020	Datenschutz-Login Professoren	Die Daten der Dozenten (Benutzername, Passwort) müssen geschützt werden. Die gebuchten Experimente sowie zusätzliche Angaben sind nur für den Admin + dem jeweiligen Dozenten einsehbar.	Technical	5	5%	0,3	TW; RB	Passwort-Verordnung (Zahl, Buchstabe, Groß- und Kleinschreibung); Passwortschutz durch System (Hashen) → Architecture Notebook
2	01.12.2020	Datenschutz-Lizenzen, Urheberrecht	Es besteht die Gefahr, dass Grafiken missbräuchlich genutzt werden. Dies kann zu einer Urheberrechtsverletzung führen.	Business	5	5%	0,3	TW; AB	Lizenzvereinbarungen der neugenenutzten Grafiken klären oder eigene Grafiken erstellen.

Risk ID	Date Identified	Headline	Description	Type	IMP	PRB	MAG	Owner	Mitigation Strategy
3	21.12.2020	Missverständnisse mit Auftraggeber	In den Gesprächen mit dem Auftraggeber kommt/ kann es zu Missverständnissen in der Kommunikation und im Verständnis kommen. Diese werden vor allem durch einen unterschiedlichen Sprachgebrauch als auch durch unterschiedliche Vorstellungen der technischen Umsetzung und der technischen Sinnhaftigkeit getragen.	Business; Schedule; direct	5	80%	4	TW, TKa, GM, AB	Häufige Meetings und reger Austausch mit dem Auftraggeber. Einfacher Sprachgebrauch und Vermeidung von Fachbegriffen. Optische Darstellungen nutzen um ein besseres Verständnis zu erreichen. Miteinbeziehen von Dozenten (Anwendern).
4	05.01.2021	Zeit für Bearbeitung reicht nicht aus	Es besteht das Risiko, dass der Umfang des Arbeitsaufwandes unterschätzt wird und dadurch die rechtzeitige Fertigstellung in Gefahr ist.	Schedule	3	20%	0,6	GM	Deadlines für Meilensteine setzen. Im Team den Aufwand gemeinsam kritisch beurteilen. → Architecture Notebook

Risk ID	Date Identified	Headline	Description	Type	IMP	PRB	MAG	Owner	Mitigation Strategy
5	05.01.2021	technische Umsetzung ist nicht möglich	Es besteht das Risiko, dass die in den Anforderungen erarbeiteten Funktionen, aufgrund von mangelnden Kenntnissen, nicht technisch umsetzbar sind.	Technical	5	50%	2,5	RB, GM	<p>Alle notwendigen Schritte durchgehen und die mögliche Umsetzung prüfen. Aufgaben nach Kompetenzen zuteilen. Im Zweifel eigenständige Weiterbildung notwendig bzw. gegenseitige Unterstützung.</p> <p>→ Architecture Notebook</p>
6	07.01.2021	Datenverlust bei Rückübertragung der virtuellen Maschine	Die Sorge des Auftraggebers ist, dass die Daten die von Ihm im aktuell laufenden System eingegeben wurden, in der neuen Version wegfallen könnten, da die Auftragnehmer das System auf Basis eines veralteten Datenbestandes entwickeln.	Technical	1	20%	0,2	RB	<p>Es ist geplant, die Datenbank neu zu inkludieren. Somit würde man nur mit aktuellen Datenbeständen arbeiten.</p> <p>→ Architecture Notebook</p>

9. Iteration Plan 02: Experimenteverwaltung - Erweiterung Buchungssystem (I2)

9.1. Meilensteine

Meilenstein	Datum
Beginn der Iteration	17.12.2020
TS-M2	17.12.2020?
Git-Workshop für Teammitglieder	21.12.2020
Ende der Iteration	31.12.2020
LCO	-

9.2. Wesentliche Ziele

- Erreichen Bestätigung der Vision durch Kunden, Über die Initialen Anforderungen herrscht Einigkeit.
- Erhebung der Systemweiten Anforderungen
- Use Case Model
- Risikoliste
- Wireframes für jeden essentiellen Funktionsschritt nach aktuellem Anforderungskatalog
- **Wissensaustausch:** Interessierte Mitglieder sollen ein GitHub-Tutorial erhalten, was einerseits die Arbeit im Team fördert als auch einen weiteren Mehrwert innerhalb dieses Moduls schafft.

9.3. Aufgabenzuordnung

Aufgaben werden in der Work Item List (<https://github.com/Schmiddl99/experimenteverwaltung-i2/projects/3#column-11862065>) geordnet und priorisiert.

Die in dieser Iteration geplanten Aufgaben wurden in der To-Do-List dargestellt: <https://github.com/Schmiddl99/experimenteverwaltung-i2/projects/3#column-11861913>. Die To-Do-List verhält sich dynamisch. Deswegen sind nachfolgend die Work Items der Iteration II aufgeführt.

Die folgenden Aufgaben werden in dieser Iteration bearbeitet:

Aufgabe bzw. Beschreibung	#Github-Issue	Schätzung der Größe (Skala A-C; A=höchster Aufwand)	Priorität (Skala 1-3; 1=höchste Prio)	Status	Zugewiesen (Name)
Erarbeitung "System wide requirements" nach TS-M2	#28	A	1	DUE: 18.12.20	siehe Issue
Meeting Coach Anforderungsproblematik	#43	A	1	DUE: nach Ferien??	tobias.karsch@htw-dresden.de
GitHub-Workshop vorbereiten	#35	A	1	DUE:19.12.20	richard.boehme@htw-dresden.de gregor.mueller@htw-dresden.de
Existierende Use Cases vervollständigen	#39	A	1	DUE: 23.12	siehe Issue
TS-M3	#56	A	2	DUE:29.12.	tim.wuerzburg@htw-dresden.de
Git-Workshop	#31	B	1	DUE: 21.12.	richard.boehme@htw-dresden.de gregor.mueller@htw-dresden.de
System Wide Requirements: adoc anlegen mit aktuellem Stand	#60	B	1	DUE:27.12.	william.elsner@htw-dresden.de
Projektplan erstellen	#4	B	3	DUE: 31.12	PM
ESSENCE: Projekt updaten	#51	C	3	DUE:31.12.	tobias.karsch@htw-dresden.de

Aufgabe bzw. Beschreibung	#Github-Issue	Schätzung der Größe (Skala A-C; A=höchster Aufwand)	Priorität (Skala 1-3; 1=höchste Prio)	Status	Zugewiesen (Name)
Vision überarbeiten: Neue Stakeholderziele + Zukunftsaussichten für kommendes Projekt	#47	C	2	DUE: 23.12.	tobias.karsch@htw-dresden.de
Vision 2.0 zur Bestätigung MH vorlegen	#42	C	3	DUE:31.12	tim.wuerzburg@htw-dresden.de
Risk List: Datenschutz	#46	C	2	DUE: 22.12.20	tim.wuerzburg@htw-dresden.de
Glossarupdate	#50	C	3	DUE: 31.12.	juliusheiko.schmidt@htw-dresden.de william.elsner@htw-dresden.de
Wireframe: Druckansicht Dozentenwoche	#41	C	2	DUE: 28.12.?	gregor.mueller@htw-dresden.de alena.bischoff@htw-dresden.de
Wireframe: Header-Menü (Drop-Down) für Studiengang (siehe Anforderungen) und Uhrzeitauswahl (VL-Zeit)	#44	C	2	DUE: 28.12.?	gregor.mueller@htw-dresden.de alena.bischoff@htw-dresden.de

Aufgabe bzw. Beschreibung	#Github-Issue	Schätzung der Größe (Skala A-C; A=höchster Aufwand)	Priorität (Skala 1-3; 1=höchste Prio)	Status	Zugewiesen (Name)
Wireframe: [Dozentenwoche] für Admin	#58	C	2	DUE: 28.12.	alena.bischoff@htw-dresden.de gregor.mueller@htw-dresden.de
Wireframe: [Journalansicht] für Dozenten	#59	C	2	DUE: 28.12.	alena.bischoff@htw-dresden.de gregor.mueller@htw-dresden.de
Dok: Zugänge (Anwendung, VM...) in Teams anlegen	#57	C	3	DUE:31.12.	richard.boehme@htw-dresden.de
Iteration Plan Pflegen	#54	x	x	Fortlaufende Aufgabe	tobias.karsch@htw-dresden.de
END: Iteration: Review	#48	B	3	DUE: 31.12.	ALL
END: Iteration: Retrospective	#49	B	3	DUE: 31.12.	ALL

9.4. Probleme (optional)

Problem	Status	Notizen
Feature Creep	entgegengewirkt	<ul style="list-style-type: none"> Nach dem Entwurf der ersten Wireframes wurden diese teilweise wieder verworfen, da von TS-Seite veränderte Anforderungen gestellt wurden. Verworfen werden mussten u.a. die Deadline, die Benachrichtigung von Dozenten, die Pop-Up-Buchungseingabe. Größte Sorgen bereitet uns die "Kopierfunktion für ein ganzes Semester". Diese ist in unseren Augen unpraktisch, stark fehleranfällig - und steht im Gegensatz zu einem "Buchungssystem", da hier im Grunde eine stetig zu replizierende Vorlage gefordert wird (Zielkonflikt). Eine Konsultation des Coaches sowie eine Bestätigung der überarbeiteten Vision durch TS sind angedacht. (Stand:22.12.20) Update 07.01.21: Im TS-M3 legen wir die Problematik offen. Alle Beteiligten erkannten die Schwierigkeit. Wir konnten uns darauf einigen, die Kopierfunktion in einer <i>Zukunftsvision</i> vorerst nur zu beschreiben. Die Arbeit des Teams fokussiert sich primär auf das funktionierende Buchungssystem.

9.5. Bewertungskriterien

- Der Kunde hat die überarbeitete Version per Mail (schriftlich) bestätigt.
- Jedes Element der Vision findet sich in den *system wide requirements* abgedeckt.
- 6 Use Cases wurden dokumentiert und in einem Modell zusammengefasst (Doku liegt als adoc vor)
- 5 Risiken und entsprechende Gegenmaßnahmen wurden identifiziert. Die Dokumentation liegt als .csv vor.
- Für jeden dokumentierten Use Case existiert ein Wireframe. Jede dokumentierte Anforderung lässt sich anhand mindestens eines Wireframes erläutern.
- Der GitHub-Workshop wurde durchgeführt und aufgezeichnet.

9.6. Assessment

Review	Gesamte Iteration 2
Assessment Datum	05.01.2021
Teilnehmer	gesamtes Team

Review	Gesamte Iteration 2
Projektstatus: gelbgrün	<ul style="list-style-type: none"> • Ziel: Erreichung LCO-Meilenstein - Inception Phase. Nicht erreicht. <ul style="list-style-type: none"> ◦ Understand what to build: Grundsätzlich verstanden. Erfolgskriterien sind noch nicht zu 100% klar - z.B. Ablaufplan des Buchens. Journal wurde angenommen, Konfliktpunkt ist die Kopierfunktion. Grenzen wurden gesetzt: Aussprache gegen mobile Version. Oberstes Erfolgskriterium: Leichte Verständlichkeit (In Design und Sprache anwendergerecht) ◦ Identify key system functionality: Die kritischsten Anforderungen wurden identifiziert: Buchungen durchführen, Buchungen anzeigen (→ Dozentenwoche), Historie (als Sahnehäubchen?) ◦ Determine at least one possible solution: Ja, die Vision ist technisch umsetzbar. ◦ Understand: <ul style="list-style-type: none"> ▪ Risiken sind verstanden ▪ Wir bekommen kein Geld (→ nur "Zeitkosten" fallen an) ▪ Zeitplan: wurde nicht abgestimmt. (Orientierung an Empfehlung aus Vorlesung) • Überprüfung selbst gesteckter Iterationsziele: Y/N (4/2) <ul style="list-style-type: none"> ◦ Der Kunde hat die überarbeitete Version per Mail (schriftlich) bestätigt. Y ◦ Jedes Element der Vision findet sich in den system wide requirements abgedeckt. Y ◦ 6 Use Cases wurden dokumentiert und in einem Modell zusammengefasst (Doku liegt als adoc vor) N <p>Nein –4 Use Cases ausgearbeitet, Rest nur erfasst, adocs liegen vor</p> <ul style="list-style-type: none"> • 5 Risiken und entsprechende Gegenmaßnahmen wurden identifiziert. Die Dokumentation liegt als .csv vor. N <p>5 Risiken liegen vor, aber nicht als .csv</p> <ul style="list-style-type: none"> • Für jeden dokumentierten Use Case existiert ein Wireframe. Jede dokumentierte Anforderung lässt sich anhand mindestens eines Wireframes erläutern. Y <p>→ Drucken-Funktion fehlt noch (Brauchen wir auch nicht?)</p> <p>→ Kopierfunktion nicht</p> <ul style="list-style-type: none"> • Der GitHub-Workshop wurde durchgeführt und aufgezeichnet. Y

Retrospective	Gesamte Iteration 2
Assessment Datum	05.01.2021
Teilnehmer	gesamtes Team
Projektstatus: grün	<p>Wie gut waren unsere Schätzungen? Welche Hindernisse gab es? Wie können wir besser werden?</p> <ul style="list-style-type: none"> • Zu viele Leute – ineffektiv? Drei Leute pro TS-Meeting sind am effizientesten! • Schätzungen im Iteration Board (Aufwand/Prio im GitHub) trafen zu, funktioniert auch mit weniger “Schätzern” • Ausgesprochene Zufriedenheit mit eigenem Teamwork • Teamwork und Gemeinschafts-Gedanke spiegelte sich z.B. im GitHub-Workshop wider

9.6.1. ALPHA States

Alphas the things to work with

📍 Software System	(0/6)
📍 Requirements	CONCEIVED (1/6)
📍 Team	SEEDED (1/5)
📍 Stakeholders	RECOGNIZED (1/6)
📍 Opportunity	SOLUTION NEEDED (2/6)
📍 Way of Working	(0/6)
📍 Work	(0/6)



Unresolved directive in se1_belegabgabe_tI2.adoc -
include::./project_management/iteration_plan_03.adoc[lines=1..1;4..-1,leveloffset=+1] <<<

Entwurfsdokumentation

- Architektur-Notizbuch
- Test Cases

10. Architecture Notebook: Experimenteverwaltung - Erweiterung Buchungssystem (I2)

2021-02-07 16:22:12 +0100

10.1. Zweck

Dieses Dokument beschreibt die Philosophie, Nebenbedingungen, Begründungen, wesentliche Elemente und andere übergreifende Aspekte, die Einfluss auf Entwurf und Implementierung bezüglich der Realisierung eines in der [Vision](#) beschriebenen Buchungssystems für Physikexperimente haben. Die für die Architektur getroffenen Entscheidungen und Ihre Hintergründe werden hier festgehalten.

10.2. Architekturziele und Philosophie

Die Architektur muss eine Webanwendung zum Erstellen und Verwalten von Buchungen von Experimenten für Vorlesungen verwirklichen. Diese basiert auf einer bereits bestehenden Anwendung und ist somit als Weiterentwicklung zu betrachten. Aufgrund dessen muss die Architektur auf eben diese Weiterentwicklung bezogen sein und weiterhin durch Übersichtlichkeit, Eindeutigkeit und Verständlichkeit für eine hohe Nutzerfreundlichkeit sorgen.

Weitere Ziele, die die Architektur in Struktur und Verhalten erfüllen muss, sind:

- Gewährleistung einer hohen Nutzungsdauer - das System soll über einen langen Zeithorizont genutzt werden und muss für eine variierende Nutzergruppe ausgelegt sein
- es muss Kompatibel für zu erwartende Änderungen sein, bspw.:
 - die Einführung einer Kopierfunktion für vergangene Buchungen
 - die freie Auswahl von betrachteten Zeiträumen bei der Statistik
 - die Änderung der Entwicklerbasis - das System soll zu einem unbestimmten Zeitpunkt von einem anderen, dem jetzigen nicht bekannten Entwicklerteam weiterentwickelt werden

10.3. Beschränkungen, Annahmen und Abhängigkeiten

- **Grundannahme:**
 - das bestehende System wird weiter-, nicht neuentwickelt
 - ein System zur Verwaltung von Physikexperimenten und eine zugehörige Datenbank bestehen bereits
- Ruby ist als Programmiersprache zu verwenden
- Vorgabe: Rails-Framework
- es wird angenommen, dass die VM ausreichend Ressourcen bietet, um die Webanwendung in

ihrem zukünftigen Ausmaße zu betreiben

- der Server nutzt das Betriebssystem Linux
- SQLite ist als Datenbankmanagementsystem zu verwenden
- die Nutzung von NGINX und Passenger als Webserver werden vom bestehenden System verlangt

10.4. Architektur-relevante Anforderungen

Über die systemweiten funktionalen und nichtfunktionalen Anforderungen kann im [Dokument zu den systemweiten Anforderungen](#) nachgelesen werden.

Anforderung	Systemkomponente	Architekturmechanismen
UC01	<ul style="list-style-type: none">• Webanwendung• Weboberfläche• Datenbank• Experiment	<ul style="list-style-type: none">• Sicherheit• Persistenz• Kommunikation• Eingabelogik• Archivierung• Session-Management• Error Management
UC02	<ul style="list-style-type: none">• Webanwendung• Weboberfläche• Buchung• Datenbank• Experiment	<ul style="list-style-type: none">• Zugriffsschutz• Kommunikation• Eingabelogik• Informationsaustausch• Error Management
UC03	<ul style="list-style-type: none">• Webanwendung• Weboberfläche• Buchung• Datenbank• Experiment	<ul style="list-style-type: none">• Zugriffsschutz• Kommunikation• Eingabelogik• Error Management
SWFA-1	<ul style="list-style-type: none">• Webanwendung	<ul style="list-style-type: none">• Zugriffsschutz
SWFA-2	<ul style="list-style-type: none">• Datenbank	<ul style="list-style-type: none">• Persistenz
SWFA-3	<ul style="list-style-type: none">• Webanwendung	<ul style="list-style-type: none">• Verschlüsselung
NFAU-1, NFAU-2	<ul style="list-style-type: none">• Weboberfläche	<ul style="list-style-type: none">• Eingabelogik

Anforderung	Systemkomponente	Architekturmechanismen
NFAR-1	• Webanwendung	• Session-Management
NFAS-1	• Webanwendung	• Informationsaustausch
NFAS-2	• Weboberfläche	• Kompatibilität

10.5. Entscheidungen, Nebenbedingungen und Begründungen

1. Weiterverwendung folgender Aspekte aufgrund zeitlicher Aufwandsbegrenzung:
 - a. Datenbank
 - b. Framework
 - c. Sprache
 - d. Frontend
 - e. visuelles Design
 - f. Bibliotheken
 - g. Backup-Routine
 - h. Klassen
 - i. Objekte
 - j. Konfigurationen
2. Verwendung von kostenlosen Bibliotheken (Open Source) und Lizenzen
 - a. Grund: dringende Vermeidung von Kosten
3. Verwendung von Ruby/Rails
 - a. Wiederverwendbarkeit = schnelle Entwicklung (adressiert Risk No. 4)
 - b. Weniger komplex; für alle Entwickler in gegebener Zeit erlernbar sofern Kompetenz noch nicht ausreichend (adressiert Risk No. 5)
4. Festlegung einer funktionalen Grenze: Kopierfunktion als Zukunftsvision
 - a. Grund: Architekturbezogene Komplexität mit Zeitbudget nicht realisierbar
5. Einführung von Quellcodekommentierung
 - a. Grund: Arbeitserleichterung des Nachfolge-Entwicklerteams
6. Verwendung von Drops-Downs, Datepicker etc. um entsprechend der Anwenderanforderungen für intuitiv bedienbare Elemente zu sorgen
 - a. Vermeidung von PopUps um der Anwenderanforderung gerecht zu werden

10.6. Architekturmechanismen

1. Archivierung

- Zustand: Implementation (bestehendes System)
- Zweck: Daten dürfen im Normalbetrieb bei möglichen Systemausfällen oder Anwendungsfehlern nicht verloren gehen
- Eigenschaften/Attribute: Backups werden im Moment täglich von Datenbank und Bilddateien angelegt, wobei stets nur die letzten 5 Backups behalten werden
- Funktion: Automatisches Backup von Datenbank und Bilddateien wird auf dem Fakultätsserver angelegt

2. Kommunikation

- Zustand: Implementation (Framework wurde festgelegt)
- Zweck: Kommunikation zwischen Datenbank und Webanwendung
- Eigenschaften/Attribute: lückenlos, synchron
- Funktion: z.B. zwischen DB/Buchungssystem

3. Error Management

- Zustand: Problem
- Zweck: Information des Anwenders bei einem unerwarteten Fehler
- Eigenschaften/Attribute: Standardmäßig im Browser
- Funktion: Rails und der Webservice geben bei einem unerwarteten Fehler eine Standard-Fehlerseite aus

4. Informationsaustausch

- Zustand: Analysis
- Zweck: Übergabe von Informationen zwischen zwei Akteuren (Menschen oder Systemen, auch untereinander)
- Eigenschaften/Attribute: fehlerfrei, verständlich, vollständig
- Funktion:
 - “Übergabe” der gebuchten Dozentenwoche: Dozent → Admin durch System
 - Kann auch Kommentare im Quellcode beschreiben (=zwischen Entwicklerteams)

5. Kompatibilität

- Zustand: Problem
- Zweck: System muss auf allen (in den SWRs (siehe SWR)) erwähnten Browsern funktionieren
- Eigenschaften/Attribute: gegeben oder nicht gegeben
- Funktion: Sicherstellen der Benutzbarkeit

6. Eingabelogik

- Zustand: Design

- Zweck:
 - Designvorschriften für schnelles Nutzerverständnis
 - Folge von Eingaben muss in eingegrenztem Schema ablaufen
- Eigenschaften/Attribute: einfach, nicht fehleranfällig, klar strukturiert
- Funktion:
 - über Drop-Down-Menüs, Date-Picker, Freitextfelder, Ausgrauungen werden die Auswahlmöglichkeiten begrenzt und kontrolliert
 - keine Formulare in Popups

7. Persistenz

- Zustand: Implementation (bestehendes System)
- Zweck:
 - Daten müssen für Admin verfügbar sein
 - Daten müssen für Journal und Statistik existieren
 - Daten dürfen nicht verschwinden (Chronologie der Statistik)
- Eigenschaften/Attribute: Datenbank ist eine SQLite-DB
- Funktion:
 - Bestellungen werden in der Datenbank gespeichert
 - Datenbank muss ausreichend Felder für alle Daten bereitstellen

8. Session-Management

- Zustand: Analysis
- Zweck: Buchungsdaten bleiben während einer validen Anmeldung bestehen
- Eigenschaften/Attribute: semipersistent
- Funktion: Session-Token wird bei Anmeldevorgang vergeben und dient als Zuordnungsschlüssel zur Sitzung und ihrer Daten

9. Datenmigration

- Zustand: Analysis
- Zweck: Daten dürfen bei der “Rückübertragung” von der VM nicht verloren gehen → Schutz vor Datenverlust (adressiert Risk No. 6)
- Eigenschaften/Attribute: vollständig, fehlerfrei
- Funktion: Ausführung von Migrationsskript(en)

10. Zugriffsschutz

- Zustand: Problem
- Zweck:
 - Authentifizierung
 - Login

- aktive Sessions
- → Datenschutz (adressiert Risk No. 1)
- Eigenschaften/Attribute: jeder Nutzer hat nur spezifischen Zugriff
- Funktion: bei Anmeldung Passwörterhalt für Dozenten, bei Login Eintreten der Zugriffsbeschränkungen

11. Verschlüsselung

- Zustand: Implementation (bestehendes System)
- Zweck: Nutzerdaten werden sicher gespeichert
- Eigenschaften/Attribute: sicher, nicht auslesbar
- Funktion:

10.7. Wesentliche fachliche Abstraktionen

- **Klasse Experiment**: Teil der Buchung - Das Objekt wird beim Buchungsvorgang aus der Datenbank bezogen. Das Objekt wird in Dozentenwoche angezeigt.
- **Klasse Nutzer (Vererbung)**:
 - **Admin**: Anwender-Rolle - Erstellt und bearbeitet Experimente. Verwaltet Rechte der Dozenten. Betrachtet alle Dozentenwochen. Betrachtet Statistik.
 - **Dozent**: Anwender-Rolle - Führt Buchung durch. Betrachtet seine Dozentenwoche. Betrachtet Journal.
- **Klasse Buchung**: Wird vom Dozenten erstellt. Beinhaltet ein oder mehrere Experimente. Besitzt zusätzliche Daten.
- **Dozentenwoche**: Ausgabe - Wird von Admin und/oder Dozent betrachtet. Enthält alle Buchungen eines Dozenten innerhalb des bestimmten Zeitabschnitts=einer Woche.
- **Journal**: Ausgabe - Wird vom Dozenten angefordert. Auflistung aller getätigten Buchungen eines Dozenten. Wird vom Dozenten betrachtet.
- **Statistik**: Ausgabe - Wird vom Admin angefordert. Wird vom Admin betrachtet. Enthält Anzahl von allen in einer bestimmten Periode gebuchten Experimenten, sortiert nach spezifischen Kategorien.

10.8. Schichten oder Architektur-Framework

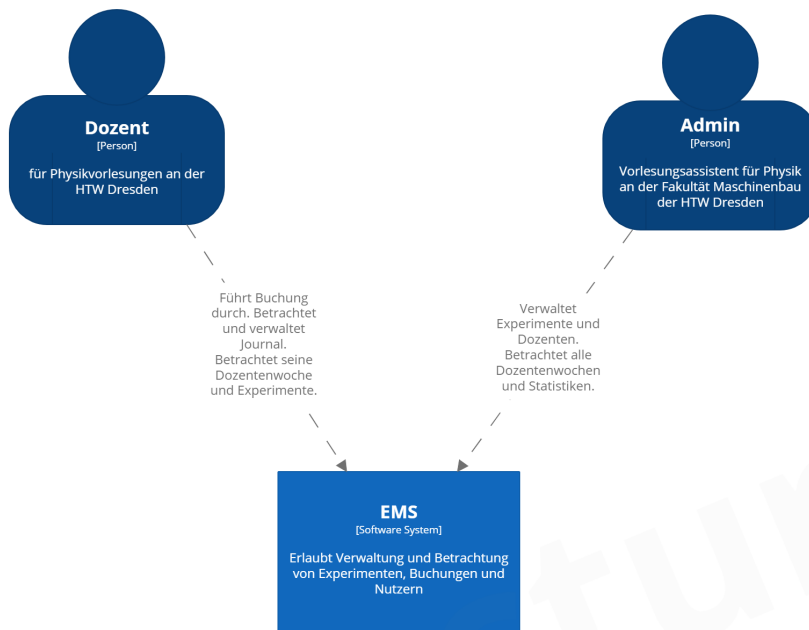
10.8.1. Model-View-Controller (MVC)

- ergibt sich durch Rails
- **Model** = Schicht der Geschäftslogik und ihrer Anwendung (siehe [Domain-Modell](#))
- **View** = Schicht der HTML-Verarbeitung zur Darstellung der Ressourcen
- **Controller** = Schicht der Requestverarbeitung durch http-Anfragen

10.9. Architektursichten (Views)

10.9.1. Logische Sicht (C4-Modell)

Kontext



System Context diagram for EMS

Try Structurizr for free at structurizr.com | Montag, 25. Januar 2021, 17:55 Mittteleuropäische Normalzeit

Abbildung 2. Context-Diagramm

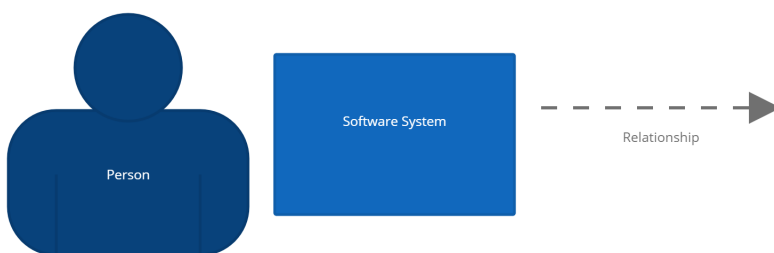
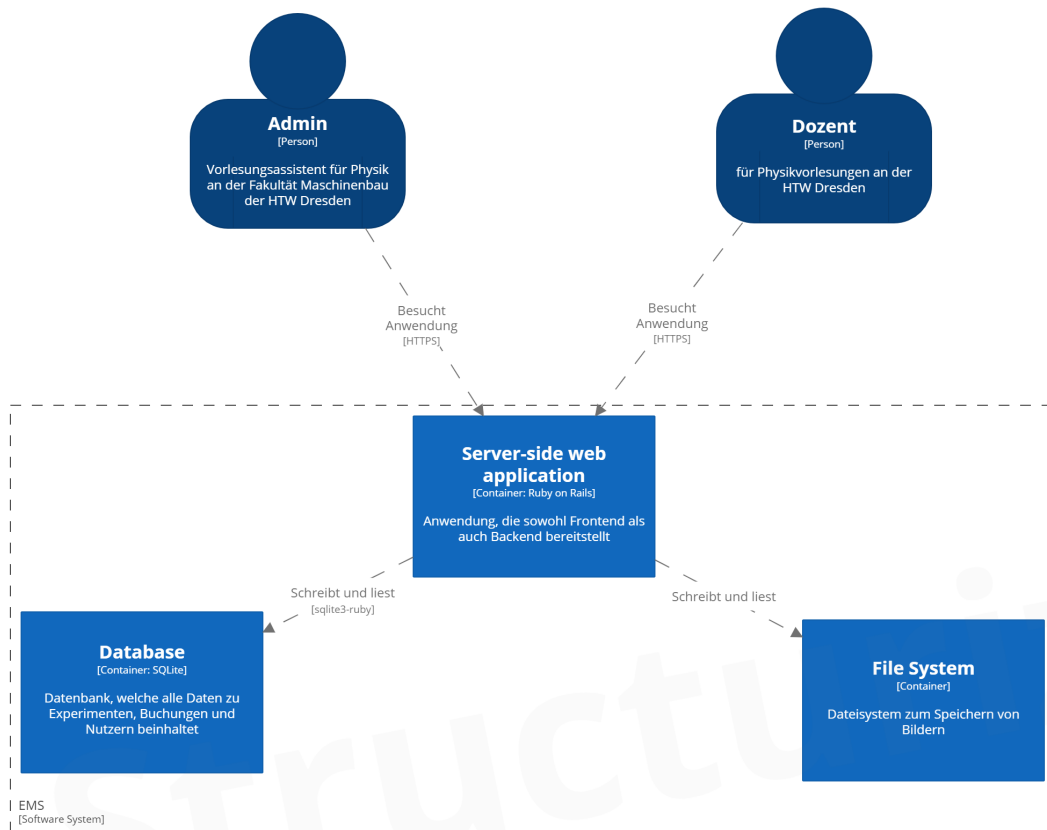


Abbildung 3. Context-Diagramm - Legende

Container



Container diagram for EMS

Montag, 25. Januar 2021, 17:55 Mitteleuropäische Normalzeit | Try Structurizr for free at structurizr.com

Abbildung 4. Container-Diagramm

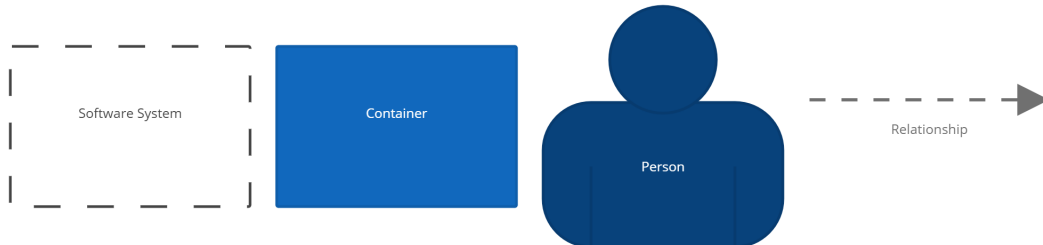


Abbildung 5. Container-Diagramm - Legende

10.9.2. Use cases

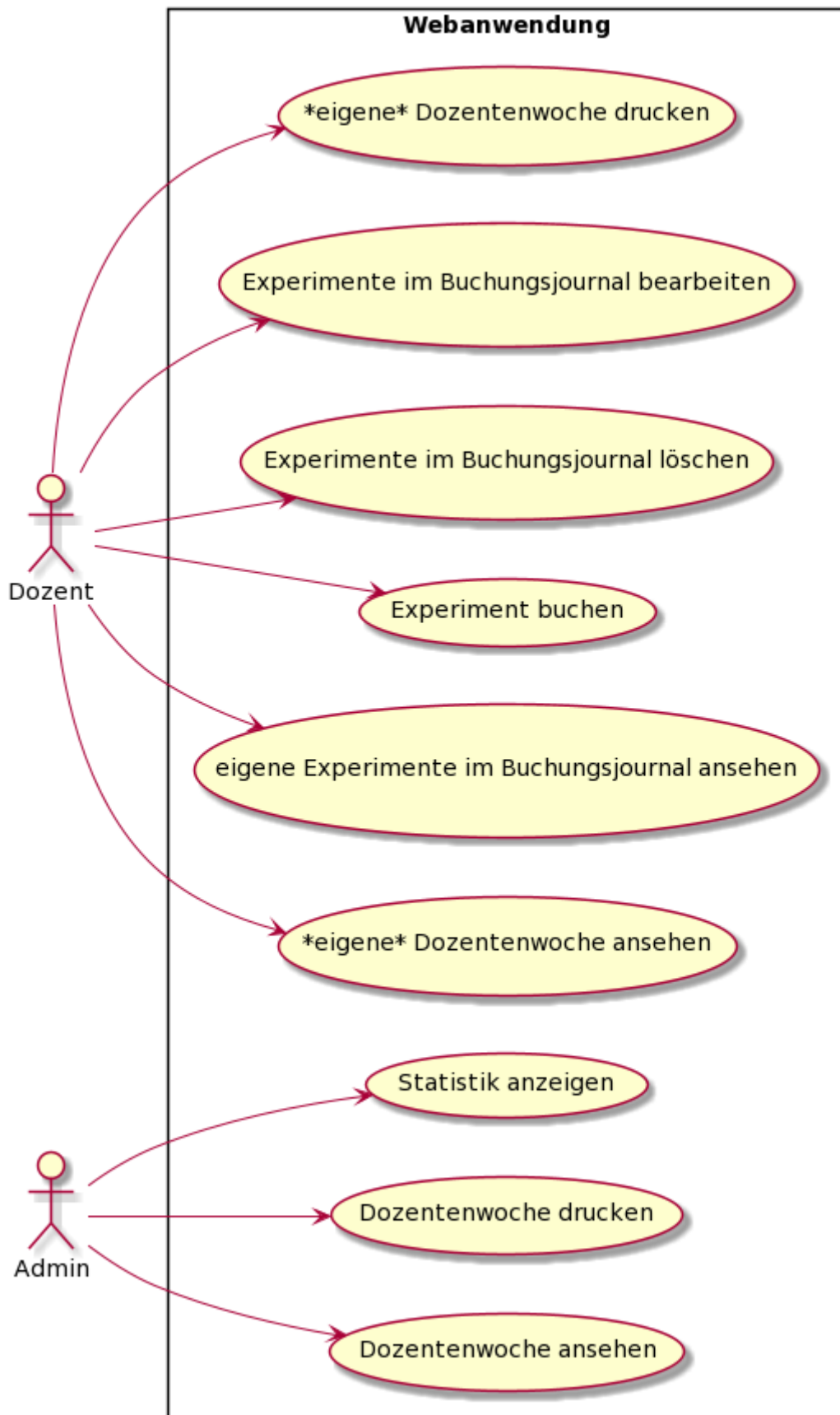


Abbildung 6. Use Case Diagramm

11. Test Cases: Experimenteverwaltung - Erweiterung Buchungssystem (I2)

11.1. 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

11.2. Test Cases für Use Cases

11.2.1. Test Cases für Use Case "Experiment buchen"

TC01 - Erfolgreich buchen

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• es muss ein Dozentenkonto geben
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. Damit sind keine menschlichen Fehler gemeint, wie z.B. ein falsches Datum ausgewählt zu haben, sondern ein Datum in der Vergangenheit zu wählen.
---------------------	---

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 und invalider Studiengang
Nachbedingung	<ul style="list-style-type: none"> • es können nur verfügbare Studiengänge über ein Dropdown Menü ausgewählt werden • es kann nur ein zeitlich korrektes Datum aus dem Pick Menü ausgewählt werden • es kann nur eine korrekte Uhrzeit in Verbindung der Unterrichtszeiten eingegeben werden

TC03 - Leerer Warenkorb darf nicht zur Buchung führen

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 • es wurde eine Buchung begonnen • es wurden valide Daten in die Voreinstellungen eingegeben • 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

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 • es wurde eine Buchung begonnen • es wurden valide Daten in die Voreinstellungen eingegeben • es wurde mindestens ein Experiment ausgewählt • valides Datum, valide Uhrzeit und valider Studiengang • 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

Beschreibung	Der Dozent beginnt einen Buchungsprozess, entscheidet sich aber zu einem beliebigen Zeitpunkt vor Buchungsabschluss dazu, diesen abubrechen. 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 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

11.2.2. Test Cases für Use Case "Buchungsjournal verwalten"

TC06 - Journal enthält chronologisch geordnete Buchungen

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. 4) 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

TC07 - Leeres Journal anzeigen

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

TC08 - Buchung im Journal bearbeiten

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
----------------------	---

TC09 - Buchung kann im Journal gelöscht werden

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

11.2.3. Test Cases für Use Case "Statistik anzeigen"

TC10 - Statistik wird richtig angezeigt

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 • es muss ein Dozentenkonto geben • 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
----------------------	---

11.2.4. Test Cases für Use Case "Dozentenwoche anzeigen"

TC11 - Dozentenwoche für einen Dozenten wird korrekt angezeigt

Beschreibung	Der Admin möchte sich die Dozentenwoche für eine bestimmte KW und einen bestimmten Dozenten ausgeben lassen.
Vorbedingungen und benötigte Daten	<ul style="list-style-type: none"> • Admin ist angemeldet • Admin befindet sich auf der Website für die Dozentenwoche • valide KW, valider Dozentennamen • die von dem Dozenten gebuchten Experimente mit dem jeweiligen Datum, Uhrzeit und Studiengang
Nachbedingung	<ul style="list-style-type: none"> • die entsprechend gefilterte Dozentenwoche, inkl. aller gebuchten Experimente, wird angezeigt

11.2.5. Test Cases für Use Case "Dozentenwoche drucken"

TC12 - Dozentenwoche wird korrekt gedruckt

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 • gefilterte Dozentenwoche, inkl. aller gebuchten Experimente, wird angezeigt • gefilterte, inkl. aller gebuchten Experimente, Dozentenwoche

Nachbedingung	<ul style="list-style-type: none"> • die entsprechend gefilterte, inkl. aller gebuchten Experimente, Dozentenwoche wird in der gewünschten Darstellung gedruckt
----------------------	--

11.2.6. Test Cases für Use Case "eigene Dozentenwoche anzeigen"

TC13 - Dozentenwoche für einen Dozenten wird korrekt angezeigt

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 • die von dem Dozenten gebuchten Experimente mit dem jeweiligen Datum, Uhrzeit und Studiengang
Nachbedingung	<ul style="list-style-type: none"> • die entsprechend gefilterte, inkl. aller gebuchten Experimente, Dozentenwoche wird angezeigt

11.2.7. Test Cases für Use Case "eigene Dozentenwoche drucken"

TC14 - eigene Dozentenwoche wird korrekt gedruckt

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"> • ein Dozent ist mit validen Zugangsdaten angemeldet • Dozent befindet sich auf der Website für die Dozentenwoche • gefilterte, inkl. aller gebuchten Experimente, Dozentenwoche wird angezeigt • gefilterte, inkl. aller gebuchten Experimente, Dozentenwoche

Nachbedingung	<ul style="list-style-type: none"> • die entsprechend gefilterte, inkl. aller gebuchten Experimente, Dozentenwoche wird in der gewünschten Darstellung gedruckt
----------------------	--

11.3. Test Cases für systemweite Anforderungen:

11.3.1. Test Cases für systemweite funktionale Anforderungen

TC15 - SWFA-1 - Auf fremde Daten/Funktionalitäten kann nicht zugegriffen werden

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

TC16 - SWFA-2 - Buchungs- und Anwenderdaten werden persistent gespeichert (manueller Test)

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

TC17 - SWFA-3 - Zugangsdaten können nicht ausgelesen werden (manueller Test)

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

11.3.2. Test Cases für nicht funktionale Anforderungen

TC18 - NFAU-1.1 - Es gibt keine redundanten Schaltflächen

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 Funktionsumfangs existieren keine Schaltflächen, die denselben Zweck erfüllen oder dieselbe Funktion aufrufen

TC19 - NFAU-1.2 - Alle Eingaben mit festem Wertebereich sind nicht durch Freitextfelder realisiert (manueller Test)

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/ Dozentwochenseite 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

TC20 - NFAU-1.2 - Alle vorgegebenen Eingabemöglichkeiten sind valide (manueller Test)

Beschreibung	Die auswählbaren Werte der Auswahlmenüs 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/ Dozentwochenseite 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

TC21 - NFAU-1.3 - Themenfremde Person kann die Anwendung ohne fremde Hilfe benutzen (manueller Test)

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

TC22 - NFAU-1.4 - Buchung eines Experiments kann innerhalb von 15 Klicks abgeschlossen werden (manueller Test)

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

TC23 - NFAU-1.4 - Nicht erreichbare Funktionalitäten können nicht aufgerufen werden

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
----------------------	--

TC24 - NFAU-1.5 - Texte und Schaltflächen sind gut lesbar (manueller Test)

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

TC25 - NFAU-2 - Eingaben geschehen nicht in Pop-Ups (manueller Test)

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
----------------------	---

TC26 - NFAR-1 - Buchung kann nach Verbindungsabbruch des Nutzers fortgesetzt werden (manueller Test)

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

TC27 - NFAP-1 - Das System funktioniert bei Auslastung durch 10 Nutzer (manueller Test)

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

TC28 - NFAS-2 - Die Anwendung muss in allen geforderten Browsern funktionieren

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