

Aktualisierte
Anforderungsspezifikation
Experimenteverwaltung - Erweiterung
Buchungssystem (I2)

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

2021-07-01 11:52:13 +0200

Inhaltsverzeichnis

1. Vision: Experimenteverwaltung - Erweiterung Buchungssystem (I2)	1
1.1. Einführung	1
1.2. Positioning	1
1.3. Stakeholder Beschreibungen	4
1.4. Produkt-/Lösungsüberblick	5
1.5. Zusätzliche Produktanforderungen	6
1.6. Aussicht für zukünftige Projekte	6
2. Domain Model: Experimenteverwaltung - Erweiterung Buchungssystem (I2)	8
2.1. Domänenmodell für die Erweiterung des EMS	8
3. Use-Case Model: Experimenteverwaltung - Erweiterung Buchungssystem (I2)	9
3.1. Allgemeine Informationen	9
3.2. Anordnung der Use Cases	11
3.3. Use-Case: Experiment buchen	12
3.4. Use-Case: Dozentenwoche ansehen	16
3.5. Use-Case: Eigene Experimente im Buchungsjournal ansehen	18
3.6. Use-Case: Experimente im Buchungsjournal bearbeiten	20
3.7. Use-Case: Eigene Buchungen im Buchungsjournal löschen	22
3.8. Use-Case: Statistik anzeigen	23
3.9. Beziehungen der User	24
4. System-Wide Requirements: Experimenteverwaltung - Erweiterung Buchungssystem (I2)	25
4.1. Einführung	25
4.2. Systemweite funktionale Anforderungen	25
4.3. Qualitätsanforderungen für das Gesamtsystem	26
4.4. Zusätzliche Anforderungen	27
5. Glossar: Experimenteverwaltung - Erweiterung Buchungssystem (I2)	29
5.1. Einführung	29

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.

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 Kunden 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: [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. den 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 Folgewoche 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-Funktion: 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-Funktion: Buchungsübersicht des Dozenten, wo er alle vergangenen Buchungen einsehen und aktuelle Buchungen anpassen kann • Dozentenwochen-Funktion: 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	<ul style="list-style-type: none">• 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. Stakeholderanalyse

Kunde

Das Hauptinteresse des Kunden ist es, die Dozenten durch das System zufrieden zustellen. Seine Angst, dass das System auf Ablehnung durch die Dozenten trifft, ist sehr groß. Er spricht sich aber leider nur sehr wenig oder teils gar nicht mit den Dozenten ab. Er versucht die Wünsche und die Probleme der Dozenten selbst zu formulieren. Dadurch ist er oft verunsichert in seinen Forderungen oder versucht das System durch ständige Änderungen der Forderungen für die Dozenten zu optimieren. Auch die Tatsache, dass es sich um fünf Dozenten handelt und er versucht es jedem einzelnen recht zu machen, stellt ihn immer wieder vor Herausforderungen, was für uns bedeutet, dass anfängliche beschlossene Umsetzungen wieder geändert werden müssen.

Dozenten

Mit den Dozenten hatten wir außer Frau Dr. Krwietz keinen Kontakt. Ihr Hauptziel ist es, dass das System Sie bei Ihrer Arbeit unterstützt und auch weiterhin all Ihre Arbeitspraktiken durch das System gewährleistet sind. Sie ist unsrem System sehr wohlgesonnen eingestellt und lässt sich auf das System ein. Auch ein Umdenken zu Ihrer alten Arbeitsweise ist bei ihr erkennbar. Es gibt aber anscheinend Dozenten, die dem System nicht positiv eingestellt sind. Ihre Ziele sind es, das System verwenden zu können, ohne sich mit ihm beschäftigen zu müssen. Es muss sehr intuitiv sein. Diese Annahme besteht aber nur auf Hörensagen. === Benutzerumgebung

Kunde	Dozent
Ist-Zustand: Kunde erhält Mails mit Bestellwünschen der einzelnen Experimente pro Woche. Daraufhin drückt sich der Kunde die Wünsche aus. Unübersichtliches, uneinheitliches und zeitintensives Auswerten der einzelnen Dozentenwünsche.	Ist-Zustand: Die Dozenten suchen sich für ihre Vorlesungen Experimente aus dem EMS aus und senden ihre Experimentwünsche dem Kunden per Mail zu. Hierbei fallen zusätzliche Arbeiten an, da sie ihre Experimente zeitlich organisieren müssen.
Soll-Zustand: Der Kunde kann sich im EMS direkt einen Wochenplan für jeden Dozenten ausdrucken lassen. Es soll einfacheres und komfortableres Arbeiten ermöglicht werden.	Soll-Zustand: Die Dozenten können ihre Wunschexperimente direkt im EMS pro Vorlesung buchen.

1.4. Produkt-/Lösungsüberblick

1.4.1. Bedarfe und Hauptfunktionen

Bedarf	Priorität	Features	Geplantes Release
Einheitliches Buchungssystem für EMS mit Buchungsverwaltungsfunktionalitäten	Hoch	<ul style="list-style-type: none">• Voreinstellung: Festlegung der Buchungsdetails vor Buchungsbeginn• Warenkorb: Experiment in Warenkorb legen und evtl. Reihenfolge anpassen• Journal: Einsehen vergangener und Bearbeiten aktueller Buchungen durch den Dozenten• Dozentenwoche: Einsehen der Buchungen je Kalenderwoche und pro Dozent	xx
Benutzerverwaltung der einzelnen Anwender	Hoch	Login: Erweitern auf dozentenspezifische Accounts	xx

Bedarf	Priorität	Features	Geplantes Release
Einfachere Auswahl eines Geräts bei der Erstellung eines Experiments	Mittel	Experimentfeld "Geräte" Anpassung: Ansicht der Geräte für Experimente übersichtlicher gestalten	xx
Buchungshäufigkeit auswerten	Mittel	Statistik: bei der Experimente-Auswahl, um die bestellten Experimente einsehen zu können	xx
Fehlerbehebung	Niedrig	<ul style="list-style-type: none"> • Verwaltung der Bilder: Bilder in Experimenten können ersetzt werden, Reihenfolge wird einbehalten • Bugfix bei Experimentfeld "Gefahren": Bei der Anzeige von Textfeldern werden diese ineinander geschoben, dies soll behoben werden 	xx

1.5. Zusätzliche Produktanforderungen

Anforderung	Priorität	Geplantes Release
Einfache Bedienbarkeit	Hoch	xx
Im Handbuch soll erklärt werden wie ein Update von Ruby on Rails durchzuführen ist	Hoch	xx
Der Kunde erhält ein Admin Konto und ein Dozenten Konto	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)
- 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

den. Wir halten den bisherigen Entwurf als Aufgliederung in zwei Semester als verbesserungswürdig. (Stand: 31.01.2021)

- Das Projekt soll der Öffentlichkeit zugänglich gemacht werden, was über den Gast Zugang realisiert werden soll. Hierbei sind große Interessen auf die Schützung der Personenbezogenen Daten so wie der Experimente Daten zu legen. (Stand: 27.04.2021)
- **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

1.6.1. Finale Zukunftsanforderungen



Die unten genannten Anforderungen resultieren aus dem Themenstellermeeting am 10.06.2021 und sind somit endgültig.

- das System soll den User informieren, wenn sich an einer bereits gedruckten Dozentenwoche etwas geändert hat
- bei der Bearbeitung einer bereits getätigten Buchung soll es möglich werden, auch die Voreinstellungen anzupassen
- im Journal soll es möglich werden, explizite Buchungen anhand einer Filterfunktion zu finden
- die Auswertung der Buchungshäufigkeit der Experimente soll anhand der Statistik umgesetzt werden
- eine Verwaltung der Studiegänge soll (ähnlich wie die der Experimente) ermöglicht werden

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

2.1. Domänenmodell für die Erweiterung des EMS

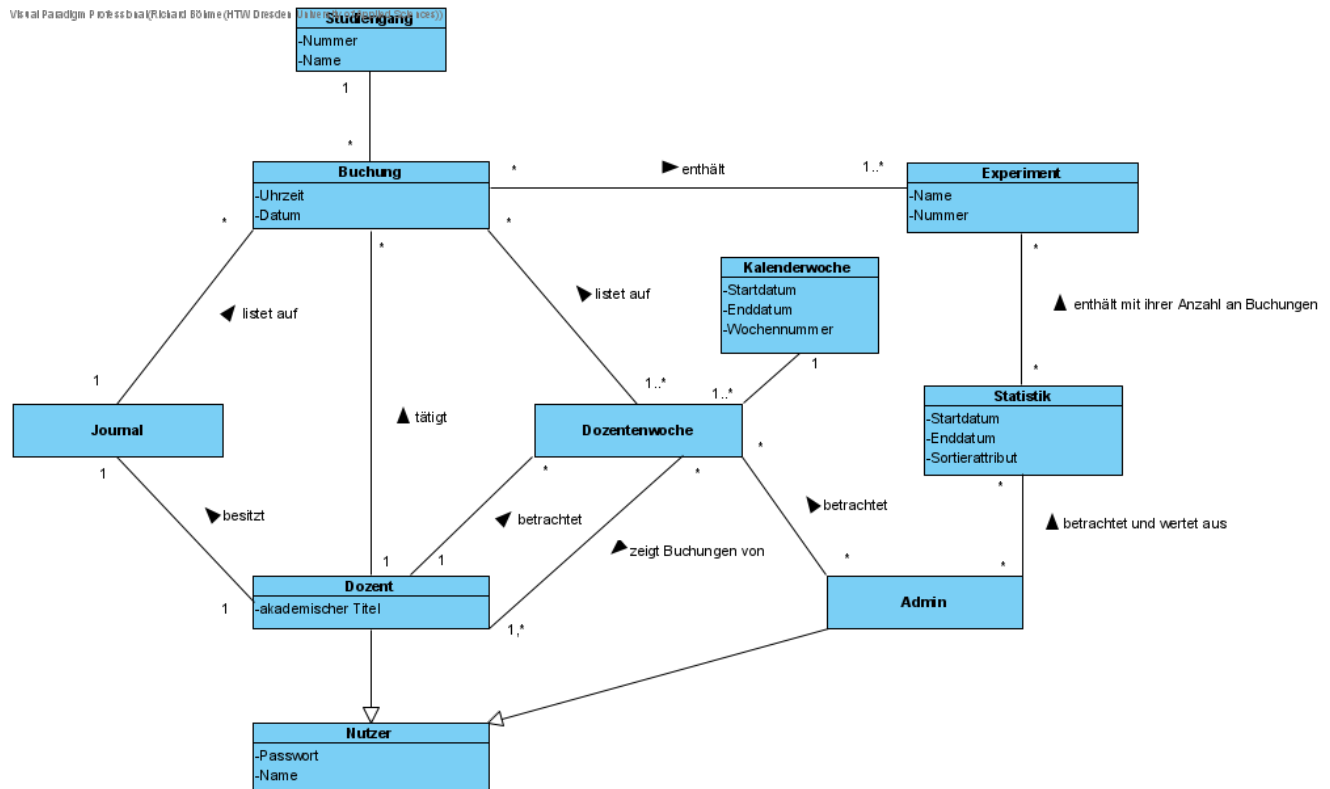
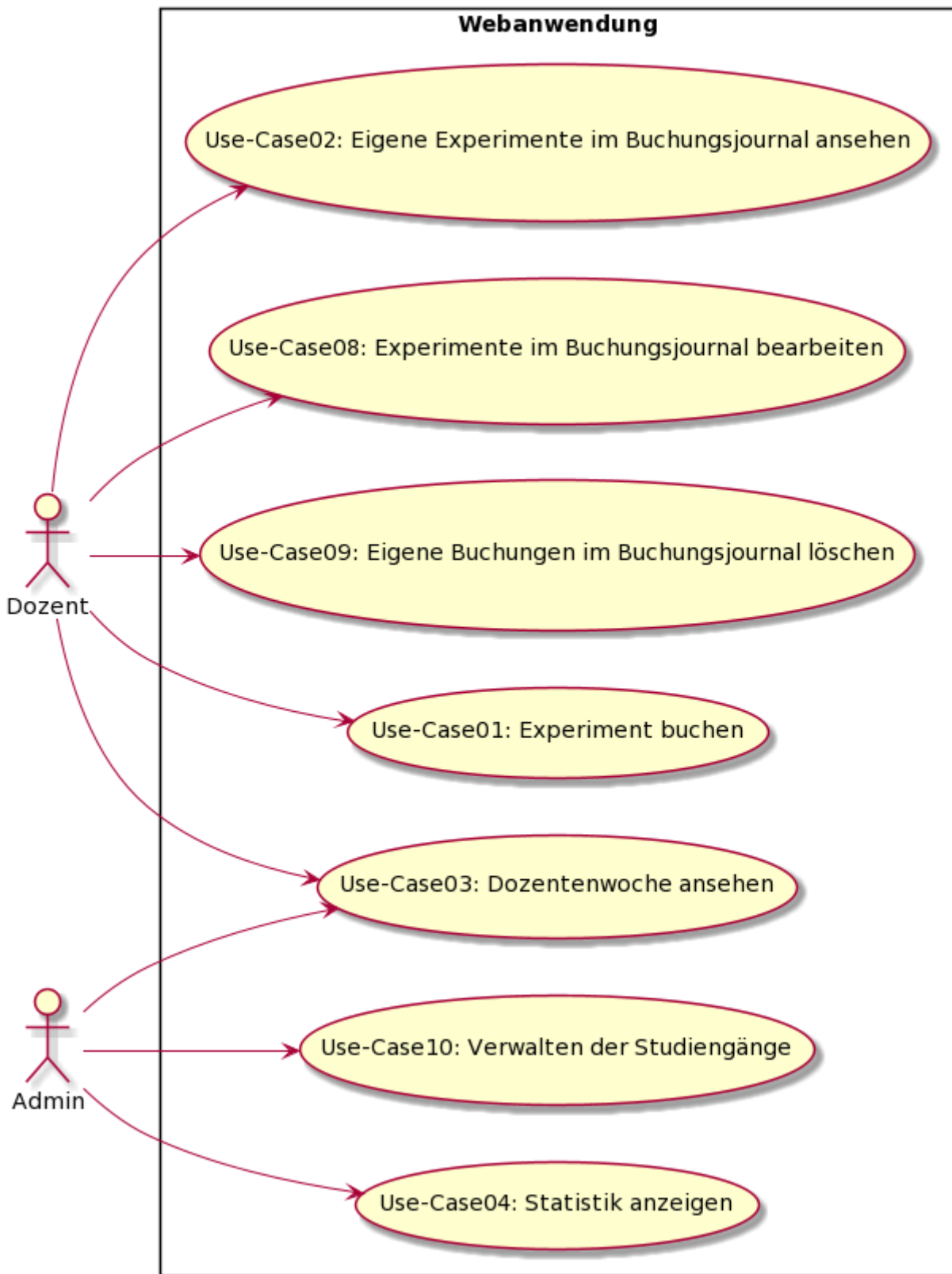


Abbildung 1. zuletzt aktualisiert am 17.06.2021

3. Use-Case Model: Experimenteverwaltung - Erweiterung Buchungssystem (I2)

3.1. Allgemeine Informationen

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



Für die einfache Darstellung 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.

Aus diesem Grund wählten wir sehr detaillierte Darstellungen, welche stellenweise über einen klassischen Wireframe hinaus gehen mögen. Die bildliche Vorstellung begünstigte eine erfolgreiche

Verständigung mit den Stakeholdern über die Lösungsansätze.

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.

3.2. Anordnung der Use Cases

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

Use Case Nummer	Use Case Name
01	Experiment buchen
03	Dozentenwoche ansehen
02	Eigene Experimente im Buchungsjournal ansehen
08	Experimente im Buchungsjournal bearbeiten
09	Eigene Buchungen im Buchungsjournal löschen
10	Verwalten der Studiengänge
04	Statistik anzeigen

3.3. Use-Case: Experiment buchen

3.3.1. Kurzbeschreibung

Buchungsablauf eines Experiments vom Buchungsstart bis zum fertigen Eintrag in der DB

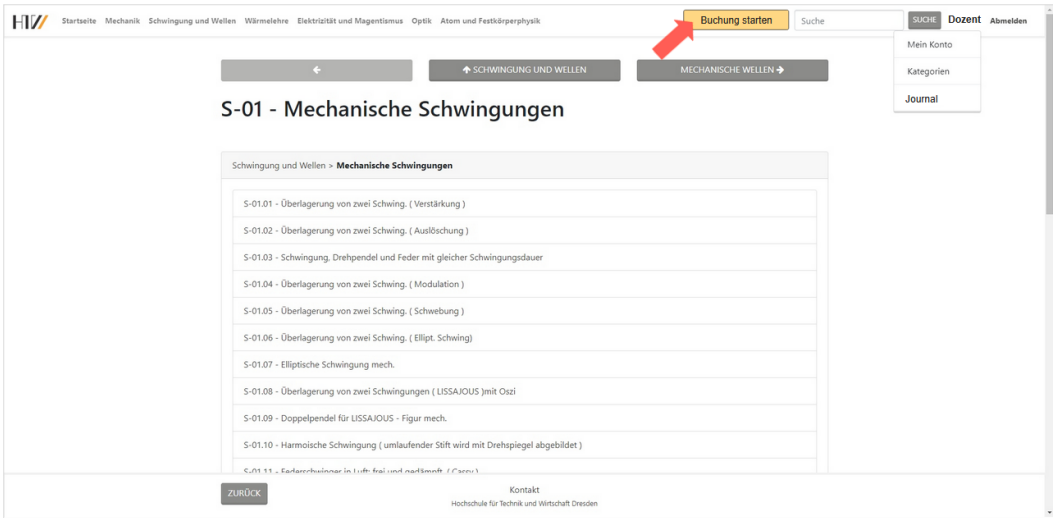
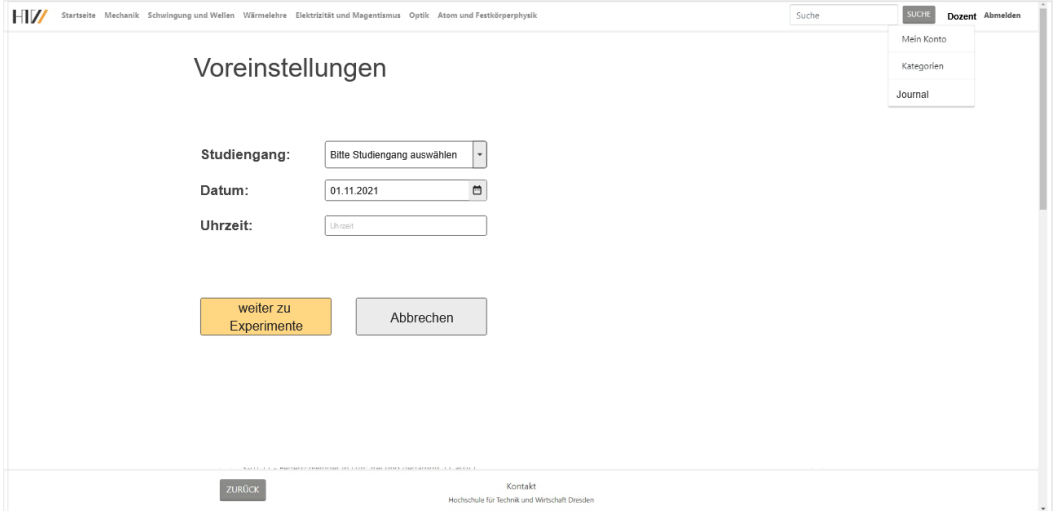
3.3.2. Kurzbeschreibung der Akteure

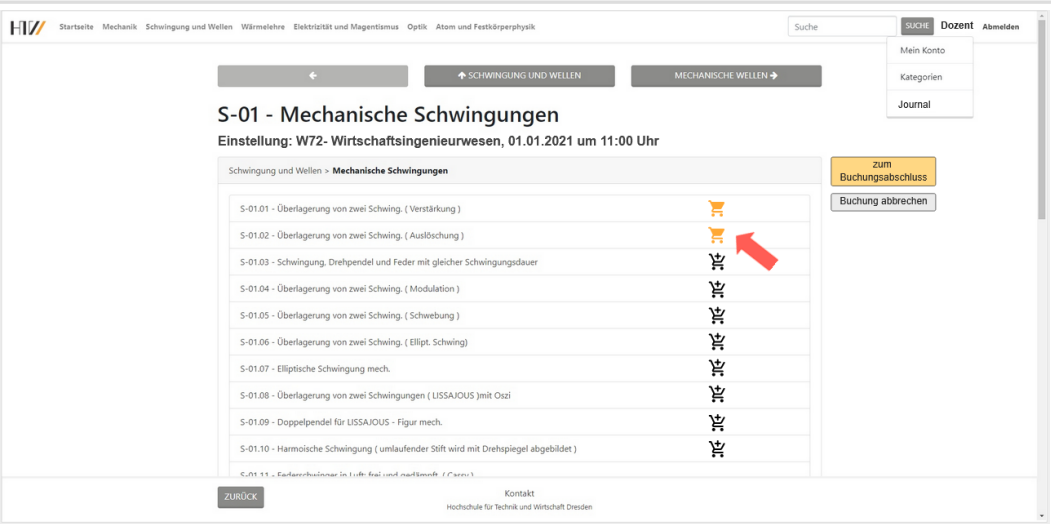
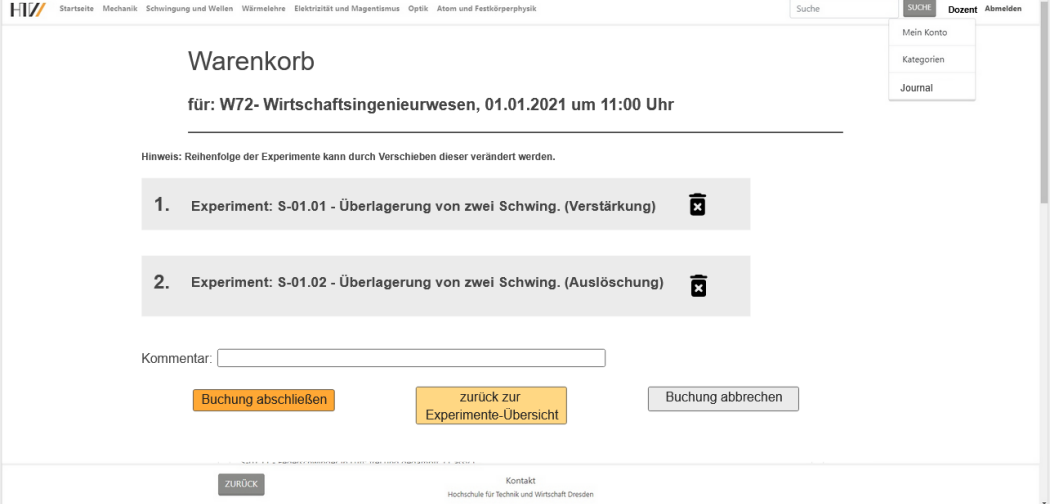
- Dozent

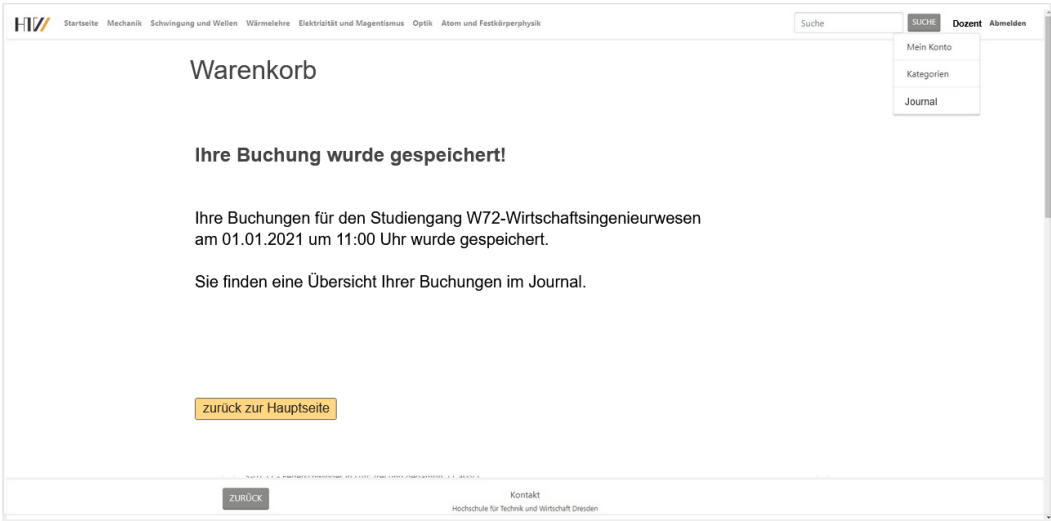
3.3.3. Vorbedingungen

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

3.3.4. Standardablauf (Basic Flow)

Beschreibung	Wireframe
1. Der Use Case beginnt, wenn User auf "Buchung starten" klickt	
2. Neue Seite mit Formular (optionaler Titel, Uhrzeit, Datum, Studiengang) → Vorauswahl für aktuellen Buchungsvorgang	
3. Exp.-Übersicht wird geöffnet	Kein Wireframe verfügbar

<p>Beschreibung</p> <p>4. Exp. können über Warenkorb-symbol zum aktuellen Buchungsvorgang hinzugefügt werden</p>	<p>Wireframe</p> 
<p>5. Wird auf "Buchung abschließen" geklickt, wird man zum Checkout (neue Seite) weitergeleitet</p>	<p>Kein Wireframe verfügbar</p>
<p>6. Im Checkout werden die aktuellen Buchungen aufgezählt, mit Möglichkeit zur Löschung von Experimenten und Kommentierung der Vorlesung. Auch ist es möglich die Reihenfolge der Experimente mit Drag and Drop zu ändern</p>	

Beschreibung	Wireframe
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	Kein Wireframe verfügbar

3.3.5. Alternative Abläufe

Benutzerdefiniertes Experiment buchen

Es ist möglich ein datenbankfremdes Experiment hinzuzufügen. Dazu gibt es die Möglichkeit nach dem Klick auf einen Button einen selbstgewählten Titel einzugeben.

Fehlerhafte Daten im Checkout angeben

Versucht der Nutzer eine Buchung mit null Experimenten oder einem Kommentar, der länger als x Zeichen ist zu speichern, erscheint eine Fehlermeldung, die den Nutzer auf seinen Fehler hinweist.

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.

3.3.6. Nachbedingungen

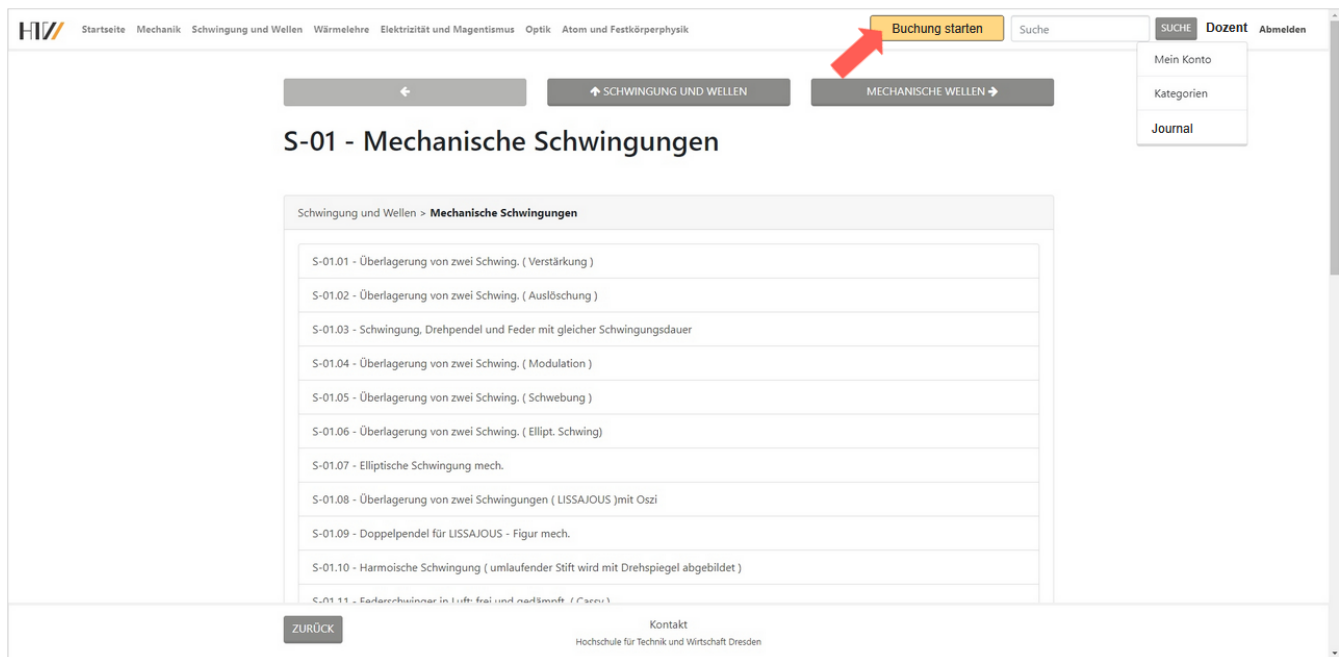
- DB-Eintrag muss initial angelegt werden

3.3.7. 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)

3.3.8. Wireframes



Erklärung:

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

3.4. Use-Case: Dozentenwoche ansehen

3.4.1. Kurzbeschreibung

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

3.4.2. Kurzbeschreibung der Akteure

- Admin

3.4.3. Vorbedingungen

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

3.4.4. Standardablauf (Basic Flow)

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

3.4.5. Wireframe Dozentenwoche ansehen

HTW Startseite Mechanik Schwingung und Wellen Wärmelehre Elektrizität und Magnetismus Optik Atom und Festkörperphysik

Suche **SUCHE** Admin Abmelden

Mein Konto
Kategorien
Dozentenwoche

Dozentenwoche

Dozenten-Auswahl
Bitte Dozentin auswählen

Zeitraum
KW 01 - 04.01.2021-10.01.2021

Anzeigen

ZURÜCK Kontakt
Hochschule für Technik und Wirtschaft Dresden

3.4.6. Alternative Abläufe

Dozent will eigenen Wochenplan einsehen

Wenn der Dozent seinen eigenen Wochenplan einsehen will, entfällt Schritt 4 des Standardablaufs. Es ist ihm nicht möglich Wochenpläne von anderen Dozenten einzusehen.

Keine Buchungen in ausgewählter Woche vorhanden

Wenn in der ausgewählten Woche des Dozenten keine Buchungen existieren, wird eine Meldung ausgegeben, die beinhaltet, dass keine Buchungen vorhanden sind.

3.4.7. Nachbedingungen

- Wochenplanansicht wird angezeigt

3.4.8. Besondere Anforderungen

Ansicht der Druckansicht

Die Druckansicht muss dem folgenden Wireframe entsprechen. Es wird die integrierte Druckfunktion des Browsers verwendet.

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 St		6. M-07.27 Fahrradmodell
	7. M-07.15 Hauptträgheitsachsen – Scheibe am St		7. M-07.21 Hauptträgheitsachsen – Quader am St
	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 St		
	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, Modell, zwischen 2. und 3. Fidget Spinner, bringe ich mit, zu 5. Fahrradfelge im Rahmen

3.5. Use-Case: Eigene Experimente im Buchungsjournal ansehen

3.5.1. Kurzbeschreibung

Dozent will seine gebuchten Exp. im Buchungsjournal ansehen.

3.5.2. Kurzbeschreibung der Akteure

- Dozent

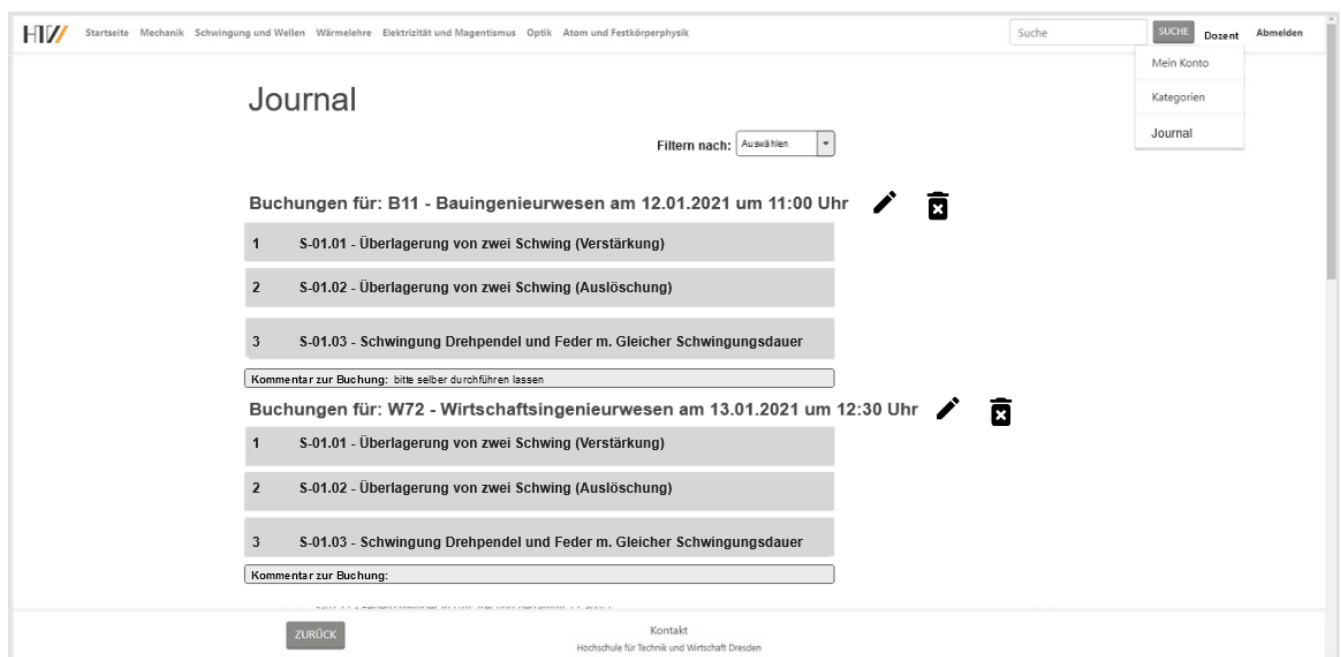
3.5.3. Vorbedingungen

- User muss eingeloggt sein
- User muss auf Buchungsjournalseite sein

3.5.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn User Buchungsjournalseite aufruft
2. Von den gebuchten Vorlesungen ist nur der Titel zu lesen, durch Klicken werden die ausgewählten Experimente der Vorlesung sichtbar. Durch erneutes Klicken auf den Titel verschwinden die Experimente wieder.
3. Möglichkeit die Ansicht zu filtern (Vorführungsdatum + Studiengang, Studiengang, Kalenerwoche) → siehe Unterablauf 1
4. Use Case endet mit Auflistung aller bisher getätigten Buchungen nach Studiengang und Vorführungsdatum

3.5.5. Wireframe Journal Ansicht



Filtern

1. Filtern

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

3.5.6. Besondere Anforderungen

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

3.6. Use-Case: Experimente im Buchungsjournal bearbeiten

3.6.1. Kurzbeschreibung

Dozent will seine gebuchten Exp. im Buchungsjournal bearbeiten.

3.6.2. Kurzbeschreibung der Akteure

- Dozent

3.6.3. Vorbedingungen

- User muss eingeloggt sein
- User muss auf Buchungsjournalseite sein

3.6.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn User Bearbeitungsformular aufruft
2. User kann die Buchung bearbeiten
 - a. Er kann Experimente aus der Buchung entfernen oder weitere aus der Experimenten-Übersicht hinzufügen
 - b. Er kann die Experimente-Reihenfolge ändern
 - c. Er kann den Kommentar bearbeiten
3. User bestätigt Änderung
4. Use Case endet damit, dass die Seite neu geladen und die Änderung aktiv wird

3.6.5. Alternative Abläufe

Fehlerhafte Daten angegeben

Versucht der Nutzer eine Buchung mit null Experimenten oder einem Kommentar, der länger als x Zeichen ist zu speichern, erscheint eine Fehlermeldung, die den Nutzer auf seinen Fehler hinweist.

3.6.6. Wesentliche Szenarios

Falsche Änderung

Änderung bei Wiederholung des Standardablaufes ab Schritt 1

3.6.7. Nachbedingungen

DB-Eintrag

DB-Eintrag wird aktualisiert und nicht neu angelegt oder gelöscht

3.6.8. 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)

3.7. Use-Case: Eigene Buchungen im Buchungsjournal löschen

3.7.1. Kurzbeschreibung

Dozent will seine Buchungen im Buchungsjournal löschen

3.7.2. Kurzbeschreibung der Akteure

- Dozent

3.7.3. Vorbedingungen

- User muss eingeloggt sein
- User muss auf Buchungsjournalseite sein

3.7.4. 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

3.7.5. Wesentliche Szenarios

Falsche Buchung gelöscht

Neue Buchung gemäß Use Case "Experiment buchen"

3.7.6. Nachbedingungen

DB-Eintrag

DB-Eintrag wird aktualisiert und nicht neu angelegt oder gelöscht

3.7.7. 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)

3.8. Use-Case: Statistik anzeigen

3.8.1. Kurzbeschreibung

Admin möchte sich eine Statistik über gebuchte Experimente anzeigen lassen

3.8.2. Kurzbeschreibung der Akteure

- Admin

3.8.3. Vorbedingungen

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

3.8.4. 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

3.8.5. Wireframe Statistik

Statistik

Sortierung: nach Nutzung

Zeitraum von: 01/01/2020 bis 01/01/2021

Sortierung: Nach Nutzung, Alphabetisch

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

3.8.6. Alternative Abläufe

Filtern

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

3.8.7. Unterabläufe (subflows)

Filtern

1. Filtern
 1. Filter auswählen
 2. Filter anwenden

3.8.8. Nachbedingungen

- spezifische Statistik wird angezeigt

3.8.9. Besondere Anforderungen

- Nur der Admin darf die Statistik einsehen

3.9. Beziehungen der User

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

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." - (siehe [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 Rol-

lenfunktionalitäten 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 minimal sein.
 - Es soll jeweils nur eine Schaltfläche pro Aktion geben (Bsp.: Warenkorb-Icon zum Buchen). Zu Überprüfungs Zwecken werden alle Interaktionselemente in ihrer Funktion verglichen.
 - **NFAU-1.2:** Die Auswahlmenüs zur Detailfestlegung müssen verständlich und inhaltlich minimal 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 Überprüfungs Zwecken 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-1.6:** Freitextfelder sind in ihrer Zeichenkapazität begrenzt.
 - Der Name des Dummy-Experiments fasst maximal 75 Zeichen.
 - Der Name des Studiengangs fasst maximal 75 Zeichen.
 - Der Buchungskommentar fasst 200 Zeichen
- **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 keine Daten verlieren oder Daten fehlerhaft speichern.
 - 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 vor der Registrierung 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 - Erweiterung Buchungssystem (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
Entwicklungsumgebung	<ul style="list-style-type: none"> • Linux-System, auf dem implementiert und getestet werden kann 	
Experiment	<ul style="list-style-type: none"> • Ein Experiment ist eine Veranschaulichung innerhalb einer Physikvorlesung • Es beinhaltet alle Informationen die zur Durchführung nötig sind 	
Experimentefeld	<ul style="list-style-type: none"> • Bei jedem Experiment gibt es die Möglichkeit Zusatzinformationen hinzuzufügen • Diese sind optional und dienen der besseren Übersicht • Es gibt die Felder: Bilder, Geräte, Gefahren, Videos, Links, Instruktionen, Anmerkung 	
Frontend-Benutzer	<ul style="list-style-type: none"> • Nutzer der öffentlich erreichbaren Weboberfläche 	<ul style="list-style-type: none"> • User • Nutzer • Anwender
Handbuch	<ul style="list-style-type: none"> • Das Handbuch dient als Nachschlagewerk für die Benutzer • In ihm sind alle Funktionen der Anwendung beschrieben • Es wird zusätzlich technische Informationen enthalten die den Update Prozess von Ruby on Rails beschreibt 	

Term	Definition/ Bedeutung	Synonym
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 Open Unified Process • <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
optionaler Titel	<ul style="list-style-type: none"> • Bei den Voreinstellungen ist es möglich einen Titel für jede Vorlesung zu vergeben • Die Vergabe eines Titels ist optional 	
Rolle	<ul style="list-style-type: none"> • Bezeichnung für Zuordnung der Zugriffsrechte (siehe "Dozent", "Admin") 	
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

Term	Definition/ Bedeutung	Synonym
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 Experimente-Management-System	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
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>

5.1.3. Verzeichnis der Datenstrukturen

Bezeichnung	Definition	Format	Gültigkeitsregeln	Aliase
Anmeldedaten	Zusammensetzung von Benutzername und Passwort	String	E-Mail-Adresse muss @-Zeichen, Punkt und "htw-dresden"-Domaine enthalten	Login
Buchungsdaten	Daten die eine Buchung beschreiben (Studiengang, Datum, optionaler Titel, Kommentar)	Date, String	Datum muss in der Zukunft liegen, Studiengang muss existieren	-
Experimentedaten	Es gibt die Felder: Kennung, Name, Bilder, Versuchsbeschreibung, Geräte, Gefahren, Videos, Links, Instruktionen, Anmerkung	String, Files	Kennung, Name und Versuchsbeschreibung müssen ausgefüllt sein	-