BELEGARBEIT

Software Engineering 1 – I6 Inventur Verleih

vorgelegt von:

Fabian Krähmer
Silko Grellmann
Jamal Alkharrat
Anja Claus
Felix Helmert
Jonas Koloska
Moritz Lehmann
Dominique Linzmajer

Dresden, den 31.01.2022

SE I - Belegabgabe: I6 Inventur Verleih

Fabian Krähmer <s82628@htw-dresden.de>
Silko Grellmann <s82695@htw-dresden.de>
Jamal Alkharrat <s82035@htw-dresden.de>
Anja Claus <s81009@htw-dresden.de>
Felix Helmert <s82049@htw-dresden.de>
Jonas Koloska <s82066@htw-dresden.de>
Moritz Lehmann <s82015@htw-dresden.de>
Dominique Linzmajer <s82063@htw-dresden.de>

Abgabedatum: 31. Januar 2022

Technische Spezifikation

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

1. Vision: I6 Inventur Verleih

1.1. Einführung

Der Zweck dieses Dokuments ist es, die wesentlichen Bedarfe und Funktionalitäten des digitalen Inventur- und Verleihsystems zu sammeln, zu analysieren und zu definieren. Der Fokus liegt auf den Fähigkeiten, die von Stakeholdern und adressierten Nutzern benötigt werden, und der Begründung dieser Bedarfe. Die Details, wie das digitale Inventur- und Verleihsystem diese Bedarfe erfüllt, werden in der Use-Case und Supplementary Specification beschrieben.

1.1.1. Zweck

Der Zweck dieses Dokuments ist es, die wesentlichen Anforderungen an das System aus Sicht und mit den Begriffen der künftigen Anwender zu beschreiben.

1.1.2. Gültigkeitsbereich (Scope)

Dieses Visions-Dokument bezieht sich auf das Digitale Inventur und Verleih System, das von Team

I6 entwickelt wird. Das System wird es der HTW Arbeitsgruppe für Smart Production Systems erlauben, Ausleihen und zurückgeben von Geggenständen zu dokumentieren, Benutzer im System anzulegen und desen Rechte zu bearbeiten, sich per Login zu authentifizieren und Backups anzulegen, um damit eine strukturierte Inventarverwaltung zu erreichen und Wartungen besser zu planen.

1.1.3. Definitionen, Akronyme und Abkürzungen

Siehe Glossar.

1.1.4. Referenzen

(hier externe Verweise zu anderen Dokumenten, Quellen, Standards etc. einfügen, sofern notwendig)

1.2. Positionierung

1.2.1. Fachliche Motivation

1.2.2. Problem Statement

Das Problem	[beschreiben Sie das Problem]
betrifft	[die Stakeholder, die vom Problem betroffen sind]
die Auswirkung davon ist	[welche Auswirkungen hat das Problem?]
eine erfolgreiche Lösung wäre	Führen Sie einige wesentliche Vorteile einer erfolgreichen Lösung auf]

1.2.3. Positionierung des Produkts

Für	HTW Smart Production Systems
der / die	verbesserte Inventarverwaltung mit Ausleihsystem
Das Produkt / die Lösung ist eine	Datenbankanwendung
Die / Das	Finden von verfügbaren und dem Gegenstand entsprechenden Lagerplätzen.
Im Gegensatz zu	einer Excel-Tabelle
Unser Produkt	ist struktureiert, modular aufgebaut, besitzt eine User- und Rechteverwaltung und es erlaubt parallelen Zugriff.

1.3. Stakeholder Beschreibungen

1.3.1. Zusammenfassung der Stakeholder

Name	Beschreibung	Verantwortlichkeiten
HTW Smart Production Systems	Auftraggeber	- erteilen einer präzise Aufgabenstellung - überprüfen, ob Lösungsvorschläge mit Erwartungen übereinstimmen
HTW Rechenzentrum / Sever Informatik/ Mathematik	stellt Betriebsumgebung	- Bereitstellung der VM - Regelung zum Betrieb der VM - Bereitstellung/Berieb Intranet
Gesetzgeber	gibt rechtliche Rahmenbedingungen vor, z.B. durch Gesetze für Datenschutz und Lagerung von Gefahrstoffen	überwacht Gesetze und Regelungen hinsichtlich des Umgang mit personenbezogenen Daten und Gefahrstoffen
Anwender (User)	Jeder Mitarbeiter der als User im System Regestriert ist	Einfache Ausleihe / Rückgabe von Gegenständen

1.3.2. Benutzerumgebung

Beschreiben Sie die Arbeitsumgebung des Nutzers. Hier sind einige Anregungen:

1. Der Benutzer steht im Lager, eine Verbindung zum System ist per Drahtloser Datenverbindung möglich
2. Der Benutzer hat ein mobiles Endgerät (bspw. Android, iOS) zur Verfügung

1.4. Produkt-/Lösungsüberblick

1.4.1. Bedarfe und Hauptfunktionen

Bedarf	Priorität	Features	Geplantes Release
Gegenstände anschauen	hoch	Anzeige Lagerplatz/Status	xx
Gegenstände hinzufügen	hoch	Suche nach leeren Lagerplatz, Eingabe von Gegeständs Daten	xx
Authentifikation der Clients	mittel	Sichere Datenübermittlung, Login	xx
Gegenstände ausleihen	hoch	Suchen und Status ändern	xx
Gegenstände zurückgeben	mittel	eingabe ID → Anzeige Lagerort, Status ändern	xx
Erstellung von Backups	niedrig	Kopieren der Datenbank	xx

1.5. Zusätzliche Produktanforderungen

Anforderung	Priorität	Geplantes Release
einfache Bedienbarkeit mittels GUI	hoch	xx
System kann nur online im HTW-Netz genutzt werden (nicht offline/ auserhalb)	mittel	xx
System muss auf allen gängigen Browsern sowie auf mobilen Endgeräten lauffähig sein	mittel	xx
Software muss modular aufgebaut sein	mittel	xx
Einsatz einer relationales Datenbankmodell	hoch	xx

2. Use-Case Model: I6 Inventur Verleih

imagesdir: images

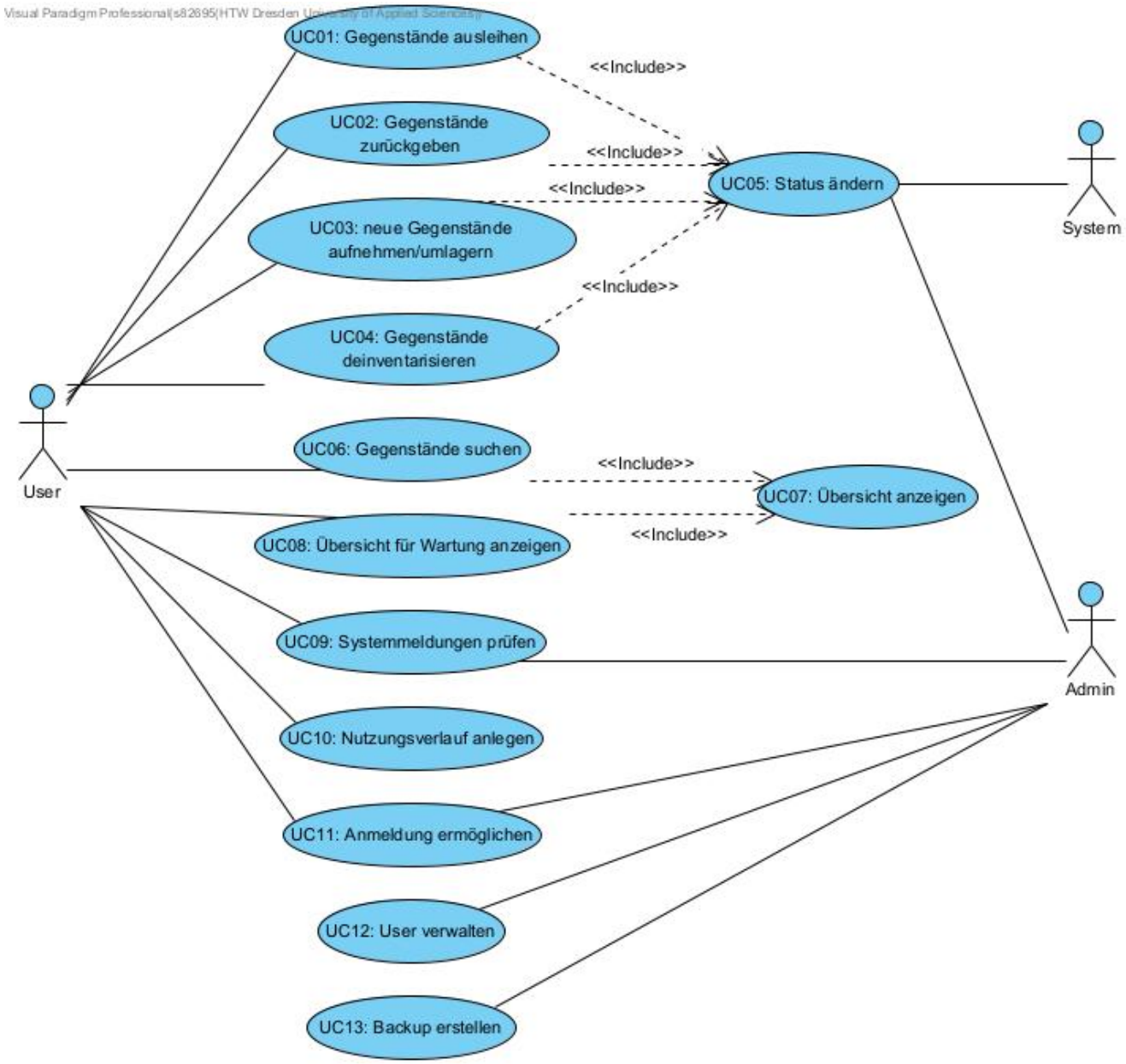
2.1. Allgemeine Informationen

2.2. Identifizierte Use Cases

für Akteur User:

- **UC01**: Gegenstände ausleihen
- **UC02**: Gegenstände zurückgeben
- **UC03**: neue Gegenstände aufnehmen
- **UC04**: Gegenstände deinventarisieren
- **UC05**: Status manuell ändern
- **UC06**: Gegenstände suchen
- **UC07**: Übersicht anzeigen
- **UC08**: Übersicht für Wartung anzeigen
- **UC09**: Systemmeldungen prüfen
- **UC10**: Nutzungsverlauf anlegen
- **UC11**: Anmeldung ermöglichen
- **UC12**: User verwalten
- **UC13**: Backup erstellen

2.3. Use Case Diagramm


Visual Paradigm Professional(s82695(HTW Dresden University of Applied Sciences))
UC01: Gegenstände ausleihen
<<Include>>
UC02: Gegenstände zurückgeben
<<Include>>
<<Include>>
UC05: Status ändern
System
UC03: neue Gegenstände aufnehmen/umlagern
<<Include>>
UC04: Gegenstände deinventarisieren
User
UC06: Gegenstände suchen
<<Include>>
UC07: Übersicht anzeigen
<<Include>>
UC08: Übersicht für Wartung anzeigen
UC09: Systemmeldungen prüfen
Admin
UC10: Nutzungsverlauf anlegen
UC11: Anmeldung ermöglichen
UC12: User verwalten
UC13: Backup erstellen

2.4. Use-Case: Gegenstände ausleihen

2.4.1. Kurzbeschreibung

Der Use Case beschreibt den Vorgang sich einen Gegenstand auszuleihen.

2.4.2. Kurzbeschreibung der Akteure

Anwender

Möchte sich einen Gegenstand ausleihen.

HTW Smart Production Systems

Möchte Anwender die Möglichkeiten geben, sich einen Gegenstand auszuleihen.

2.4.3. Vorbedingungen

- Gegenstand ist im System aufgeführt und als verfügbar angezeigt
- Anwender hat den Lagerplatz des Gegenstandes gefunden
- Anwender ist angemeldet
- System ist aufgerufen

2.4.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Anwender einen Barcode eines Gegenstandes scannt.
2. Das System findet den Gegenstand in der Datenbank und zeigt die Informationen (u.a. den Status) an.
3. Der Anwender ändert den Status von "auf Lager" zu "ausgeliehen".
4. Das System ändert automatisch den Status des Fachs auf "Gegenstand, der in dieses Fach gehört wurde ausgeliehen".
5. Der Anwender bestätigt die Statusänderung.
6. Das System speichert die Änderungen.
7. Der Use Case ist abgeschlossen.

2.4.5. Alternative Abläufe

Ungültiger Barcode

Wenn Anwender im Schritt 1 des Standardablauf einen unbekannten Barcode scannt, dann geht der Use Case nicht weiter zum nächsten Schritt. Der Use Case bleibt im Schritt 1 stehen und wartet auf einen gültigen Barcode.

manuelles Statusändern durch Admin

1. Der Use Case beginnen, wenn der Admin im Barcodescanner die manuelle Eingabe auswählt.
2. Der Admin sucht durch Nutzung der Filter oder Eingabe der Seriennummer den Gegenstand
3. Das System findet den Gegenstand in der Datenbank und zeigt die Informationen an.
4. Der Admin ändert den Status.
5. Das System ändert automatisch den Stuatus des Faches.
6. Der Admin bestätigt die Statusänderung.
7. Das System speichert die Änderung.
8. Der Use Case ist abgeschlossen.

2.4.6. Unterabläufe (subflows)

2.4.7. Wesentliche Szenarios

erfolgreiche Durchführung

1. Herr X möchte einen bestimmten Gegenstand mit der Bezeichnung YZ ausleiehen.
2. Herr X scannt den Barcode und ändert den Status des Gegenstandes. Das Fach mit der Bezeichnung XZ in dem der Gegenstand vorher lag wird als leer angezeigt.

falscher Barcode

1. Herr X scannt einen falschen Barcode.
2. Das System bleibt im Schritt 1 stehen.

2.4.8. Nachbedingungen

erfolgreiche Durchführung

Der Anwender hat den richtigen Gegenstand ausgewält und den Status erfolgreich geändert. Der Status beleibt solange erhalten, bis zur nächsten Änderung.

2.4.9. Besondere Anforderungen

Usability

Statusänderung muss schnell, einfach und standardisiert möglich sein.

Performance

Nach dem Scannen des Barcodes muss der Gegenstand in unter 5 s in der Datenbank gefunden werden.

Regeln

Anwender muss die nötigen Rechte haben, sich diesen Gegenstand ausleihen zu können. <<<

2.5. Use-Case: Gegenstände zurückgeben

2.5.1. Kurzbeschreibung

Der Use Case beschreibt die Rückgabe eines Gegenstänsds durch einen Benutzer.

2.5.2. Kurzbeschreibung der Akteure

Anwender

Möchte einen ausgeliehenen Gegenstand wieder zurück geben.

2.5.3. Vorbedingungen

Der Anwender hat die Webseite ausgewählt und ist eingeloggt.

2.5.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Anwender einen Barcode eines Gegenstandes scannt.
2. Das System findet den Gegenstand in der Datenbank und zeigt die Informationen (u.a. den Status) an.
3. Der Anwender wählt "Rückgabe" aus
4. Das System zeigt den geplanten Lagerort an.
5. Der Anwender legt den Gegeenstand in das geplannte Fach
6. Der Anwender bestätigt die Statusänderung von "ausgeliehen" auf "auf Lager" sowie den neuen Lagerort
7. Der Use Case ist abgeschlossen.

2.5.5. Alternative Abläufe

Alternativer Ablauf Manuelle Eingabe

Wenn Anwender im Schritt 1 des Standardablauf nicht den Barcode scannt.

1. Der Anwender wählt im Barcode Scanner die Manuelle Eingabe aus.
2. Der Anwender sucht in der Suche Maske durch nutzung von Filtern den passenden Gegenstand.
3. Der Use Case wird im Schritt 2 fortgesetzt.

manuelles Statusändern durch Admin

1. Der Use Case beginnen, wenn der Admin im Barcodescanner die manuelle Eingabe auswählt.

2. Der Admin sucht durch Nutzung der Filter oder Eingabe der Seriennummer den Gegenstand
3. Das System findet den Gegenstand in der Datenbank und zeigt die Informationen an.
4. Der Admin ändert den Status.
5. Das System ändert automatisch den Stuatus des Faches.
6. Der Admin bestätigt die Statusänderung.
7. Das System speichert die Änderung.
8. Der Use Case ist abgeschlossen.

2.5.6. Wesentliche Szenarios

erfolgreiche Rückgabe

1. Anwender scannt Barcode und ändert den Status des Gegenstandes
2. Status des Fachs wird automatisch geändert
3. alle Statusänderungen werden gespeichert

2.5.7. Nachbedingungen

bei erfolgreicher Rückgabe

- Statusänderungen von Gegenstand und Fach wurden gespeichert

2.5.8. Besondere Anforderungen

Erweiterungen

- Anzeige aller Aktuell vom Anwender ausgeliehenen Gegenstände. Zur schnelleren Manueller Auswahl → gegebenfalls in Suchmaske als Filter Funktion. <<<

2.6. Use-Case: neue Gegenstände aufnehmen

2.6.1. Kurzbeschreibung

Der Use Case beschreibt den Vorgang einen neuen Gegenstand ins System aufzunehmen und einen Platz zu finden.

2.6.2. Kurzbeschreibung der Akteure

Anwender / HTW Smart Production Systems

Hat einen neuen Gegenstand und will für diesen einen Platz finden, sowie ihn in das System aufnehmen.

2.6.3. Vorbedingungen

- Es gibt noch Lagerplätze mit dem Status "leer".
- Anwender ist angemeldet
- System ist aufgerufen

2.6.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Anwender oder Mitglieder der HTW Smart Production Systems (Arbeitsgruppe) sich die Fächer anzeigen lässt, die den Status "leer" haben
2. Anwender oder Mitglieder der Arbeitsgruppe wählt ein geeignetes Fach für den Gegenstand aus
3. System erfragt Informationen zu dem neuen Gegenstand
4. Anwender oder Mitglied der Arbeitsgruppe gibt die Informationen ein
5. System zeigt Übersicht über die Daten an
6. Anwender oder Mitglied der Arbeitsgruppe bestätigt die Daten
7. System erfragt ob ein Barcode für den Gegenstand generiert werden soll
8. Anwender oder Mitglied der Arbeitsgruppe bestätigt und druckt den Barcode aus
9. System erfragt den Barcode.
10. Kunde scannt den Barcode
11. System speichert die Daten und ändert den Status des Faches von "leer" auf "Gegenstand Y befindet sich im Fach"
12. Der Use Case ist abgeschlossen.

2.6.5. Alternative Abläufe

unzureichende Informationseingabe

Wenn Anwender oder Mitglied der Arbeitsgruppe im Schritt 4 des Standardablauf keine oder unzureichende Informationen eingbibt, dann wird der Use Case im Schritt 3 fortgesetzt.

manuelles Statusändern durch Admin

1. Der Use Case beginnen, wenn der Admin im Barcodescanner die manuelle Eingabe auswählt.
2. Der Admin sucht durch Nutzung der Filter oder Eingabe der Seriennummer den Gegenstand
3. Das System findet den Gegenstand in der Datenbank und zeigt die Informationen an.
4. Der Admin ändert den Status.
5. Das System ändert automatisch den Stuatus des Faches.
6. Der Admin bestätigt die Statusänderung.
7. Das System speichert die Änderung.
8. Der Use Case ist abgeschlossen.

2.6.6. Wesentliche Szenarios

erfolgreiche Neufaufnahme

1. System findet geeignetes Leeres Fach
2. Anwender gibt Informationen ein, fordert den Barcode an und scannt diesen
3. System speichert Informationen und neuen Status des Faches.

fehlgeschlagene Neuaufnahme (kein freier Platz)

1. Anwender oder Mitglied der Arbeitsgruppe lässt sich freie Lagerplätze anzeigen
2. System findet keine und zeigt diese Information an
3. Neuaufnahme wird abgebrochen

2.6.7. Nachbedingungen

nach erfolgreicher Durchführung des Use Case:

- Anwender oder Mitglied der Arbeitsgruppe hat alle Informationen eingegeben.
- alle Daten wurden gespeichert.
- das Lagerfach hat einen neuen Status.
- Gegenstand und Lagerfach wurden einander zugewießen.

2.6.8. Besondere Anforderungen

Usability

Der Use Case ist eine der wichtigsten Funktionen des Systems und muss deshalb einfach zu bedienen sein.

2.7. Use-Case: Gegenstände deinventarisieren

2.7.1. Kurzbeschreibung

Der Use Case beschreibt den Vorgang einen Gegenstand aus dem System zu entfernen.

2.7.2. Kurzbeschreibung der Akteure

Anwender oder Mitglieder der Arbeitsgruppe

Möchte Gegenstände aus dem System entfernen können.

Mitglieder von HTW Smart Production Systems

Möchte die Möglichkeit haben Gegenstände, die physisch entsorgt wurden oder nicht mehr gelagert werden, aus dem System entfernen.

2.7.3. Vorbedingungen

- es muss mindestens ein Gegenstand im System vorhanden sein
- Anwender ist angemeldet
- System ist aufgerufen

2.7.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Anwender den Barcode eines Gegenstands gescannt hat
2. System findet den Gegenstand in der Datenbank und zeigt Informationen zu diesem Gegenstand an
3. Anwender oder Mitglied der Arbeitsgruppe ändert den Status von "auf Lager" zu "entsorgt"
4. System erfragt eine Bestätigung der Stausänderung
5. Kunde bestätigt diese
6. System ändert und speichert den Status des Fachs von "Gegenstand Z befindet sich in dem Fach" zu "leer" und des Gegenstandes von "auf Lager" zu "entsorgt"
7. Der Use Case ist abgeschlossen.

2.7.5. Alternative Abläufe

keine Bestätigung der Sicherheitsabfrage

Wenn im Schritt 5 des Standardablauf keine Bestätigung erfolgt, dann wird die Statusänderung zurückgesetzt und der Use Case wird im Schritt 2 fortgesetzt.

manuelles Statusändern durch Admin

1. Der Use Case beginnen, wenn der Admin im Barcodescanner die manuelle Eingabe auswählt.

2. Der Admin sucht durch Nutzung der Filter oder Eingabe der Seriennummer den Gegenstand
3. Das System findet den Gegenstand in der Datenbank und zeigt die Informationen an.
4. Der Admin ändert den Status.
5. Das System ändert automatisch den Stuatus des Faches.
6. Der Admin bestätigt die Statusänderung.
7. Das System speichert die Änderung.
8. Der Use Case ist abgeschlossen.

2.7.6. Wesentliche Szenarios

erfolgreiche Entsorgung

1. Anwender oder Mitglied der Arbeitsgruppe scannt den Barcode und sieht alle Informationen des Gegenstandes im System
2. Anwender oder Mitglied der Arbeitsgruppe ändert den Status und bestätigt die Sicherheitsabfrage
3. Statusänderung von Fach und Gegenstand wird gespeichert

2.7.7. Nachbedingungen

- Der Anwender oder das Mitglied der Arbeitsgruppe hat den Status des Gegenstandes geändert
- Das Fach und der Gegenstand hat einen neuen Status

2.7.8. Besondere Anforderungen

Usability

- einfache Bedinbarkeit
- Sicherheitsabfrage und gute Beschreibungen <<<

2.8. Use-Case: Status manuell ändern

2.8.1. Kurzbeschreibung

Der abhängige Use Case beschreibt den Vorgang den Status eiens Gegenstands manuell zu ändern.

2.8.2. Kurzbeschreibung der Akteure

Administrator

Möchte die Möglichekeit den Status eines Gegenstandes zu ändern.

2.8.3. Vorbedingungen

- System ist aufgerufen
- Administrator ist angemeldet

2.8.4. Standardablauf (Basic Flow)

1. Der Use Case beginnen, wenn der Admin im Barcodescanner die manuelle Eingabe auswählt.
2. Der Admin sucht durch Nutzung der Filter oder Eingabe der Seriennummer den Gegenstand
3. Das System findet den Gegenstand in der Datenbank und zeigt die Informationen an.
4. Der Admin ändert den Status.
5. Das System ändert automatisch den Stuatus des Faches.
6. Der Admin bestätigt die Statusänderung.
7. Das System speichert die Änderung.
8. Der Use Case ist abgeschlossen.

2.8.5. Nachbedingungen

- Status von Gegenstand und Fach wurden erfolgreich geändert
- Statusänderung wurde gespeichert

2.9. Use-Case: Gegenstände suchen

2.9.1. Kurzbeschreibung

Der Use Case beschreibt den Vorgang einen bestimmten Gegenstand zu suchen, sodass der Anwender herrauffindet ob der Gegenstand ausgliehen ist und von wem oder in welchem Lagerfach er sich befindet.

2.9.2. Kurzbeschreibung der Akteure

Anwender | HTW Smart Production System

Möchte herrausfinden wo sich ein Gegenstand befindet.

2.9.3. Vorbedingungen

- Anwender ist angemeldet
- System ist aufgerufen

2.9.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Anwender die Suchfunktion aufruft.
2. Der Anwender gibt seine Suchkriterien ein.
3. System zeigt Suchergebnisse und deren Status und Lagerort an
4. Anwender wählt einen Gegenstand aus
5. System zeigt weitere Informationen zum Gegenstand
6. Der Use Case ist abgeschlossen.

2.9.5. Alternative Abläufe

1. Werden keine Suchkriterien eingegeben, werden alle Gegenstände angezeigt. (Use Case 7)

Gegenstand nicht vorhanden

Wenn Anwender im Schritt 1 des Standardablauf eine Objektbezeichnung eingibt, die es nicht gibt, dann erscheint eine Meldung, dass dieses Objekt sich nicht im System befindet. Der Use Case ist damit abschlossen.

2.9.6. Wesentliche Szenarios

erfolgreiches Herausfinden der Lagerposition

1. Anwender gibt die Objektbezeichnung ein
2. System zeigt die Lagerposition an
    - Bsp. Kunde kann sich beispielsweise den Gegenstand ausleihen / ihn entsorgen etc.

erfolgreiches Herausfinden des Status

1. Anwender gibt die Objektbezeichnung ein
2. System zeigt an das der Gegenstand nicht verfügbar ist, weil der Status auf "ausgeliehen" steht
3. Anwender wählt den Gegenstand aus
4. System zeigt an wann und von wem der Gegenstand ausgleihen wurde

Objekt nicht vorhanden

1. Anwender gibt die Objektbezeichnung ein
2. System gibt eine Meldung, dass sich ein solches Objekt nicht im System befindet
    - Bsp. Anwender weiß nun, dass er sich den Gegenstand anderweitig beschaffen muss

2.9.7. Nachbedingungen

Bei erfolgreicher Suche

- Anwender bekommt Überblick über alle Gegenstände, die auf sein Filterkriterum passen
- Anwender weiß an welchem Lagerort oder bei welcher Person sich bestimmte Gegenstände befinden
- zusätzliche Informationen werden über Gegenstände eingeholt

2.9.8. Besondere Anforderungen

Usability

- schnelle und einfache Suche, da der Use Case eine grundlegende Hauptfunktion des Systems ist <<<

2.10. Use-Case: Übersicht anzeigen

2.10.1. Kurzbeschreibung

Der Use Case bescheibt das Anzeigen von Übersichten nach bestimmten Filterkriterien oder das Anzeigen einer Gesamtübersicht über alle Gegenstände die im System erfasst sind.

2.10.2. Kurzbeschreibung der Akteure

Anwender

Möchte sich Gesamtübersichten oder Übersichten beispielsweise sortiert nach Zimmer, nach Art des Gegenstands, nach Verfügbarkeit etc. anzeigen lassen

2.10.3. Vorbedingungen

- Benutzer ist angemeldet

- System ist aufgerufen

2.10.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Anwender die Suchfunktion aufruft.
2. Der Anwender gibt keine Suchkriterien ein.
3. System zeigt alle Gegegenstände und deren Status und Lagerort an
4. Der Anwender kann die Ergebnisliste durch Klicken auf die Spaltenüberschrift neusortier.
5. Anwender wählt einen Gegenstand aus
6. System zeigt weitere Informationen zum Gegenstand
7. Der Use Case ist abgeschlossen.

2.10.5. Alternative Abläufe

Alle Objekte sollen angezeigt werden

Wenn Anwender im Schritt 3 des Standardablauf "alles anzeigen" auswählt, dann werden im Schritt 4 alle Gegenstände alphabetisch aufgelistet. Der Use Case ist damit auch abgeschlossen.

2.10.6. Wesentliche Szenarios

erfolgreiches Anzeigen einer Übersicht

1. Anwender wählt Filterkriterum aus
2. System zeigt alle Gegenstände gemäß des Kriteriums an

keine Übersicht für das Filterkriterum

1. Anwender wählt Filterkriterum aus
2. System erkennt, dass kein Gegenstand zu dem Kriterium passt und gibt diese Information an den Anwender

2.10.7. Nachbedingungen

bei erfolgreichem Anzeigen

- Anwender bekommt Überblick über alle Gegenstände, die auf sein Filterkriterum passen

2.10.8. Besondere Anforderungen

Usability

- schnelles und einfaches Bedienen, da es sich um eine Hauptfunktion des Systems handelt <<<

2.11. Use-Case: Übersicht für Wartung anzeiegn

2.11.1. Kurzbeschreibung

Der Use Case beschreibt das Aufrufen einer Übersicht, welche alle Gegenstände auflistet, die in die Wartung müssen.

2.11.2. Kurzbeschreibung der Akteure

<User>

Will wissen welche Gegenstände in die Wartung müssen.

2.11.3. Vorbedingungen

1. Der User muss eingeloggt sein.

2.11.4. Standardablauf (Basic Flow)

1. Der Anwender ruft die Wartungsplanung auf
2. Das System nutzt die Filtersuche (Use Case 6) um alle Gegenstände mit demnächst (bsp. nächste Woche) geplanter Wartung anzuzeigen
3. Der Use Case ist abgeschlossen.

2.11.5. Alternative Abläufe

<Es muss kein Gegenstand zur Wartung>

Wenn in der nächsten Woche kein Gegenstand zur Wartung muss, wird eine entsprechende Meldung ausgegeben.

2.11.6. Wesentliche Szenarios

<Es muss mindestens ein Gegenstand in der nächsten Woche zur Wartung>

1. Es wird dem User eine Tabelle mit den Gegenständen angezeigt, die zur Wartung müssen.
2. In der Tabelle ist der Gegenstand, die Gegenstandskennung (z.B. Seriennummer), der Lagerort, der Status und das genaue Datum aufgelistet.

<Es muss kein Gegenstand in der nächsten Woche zur Wartung>

1. Es wird dem User eine Meldung angezeigt, dass in der nächsten Woche kein Gegenstand zur Wartung muss. <<<

2.12. Use-Case: Systemmeldungen prüfen

2.12.1. Kurzbeschreibung

Der Use Case beschreibt den Erhalt von Systembenachrichtigungen bei jeder Änderung.

2.12.2. Kurzbeschreibung der Akteure

<User>

Will eine Rückmeldung über erfolgreiche Änderungen an der Datenbank

<Admin>

Will eine Rückmeldung über erfolgreiche Änderungen an der Datenbank

2.12.3. Vorbedingungen

1. Der User/Admin muss erfolgreich eingeloggt sein.
2. Der User/Admin muss die Berechtigungen zum Ändern besitzen

2.12.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der User/Admin eine Änderung an der Datenbak vornimmt.
2. Der User/Admin Bestätigt das Ausführen einer Aktion (Gegenstand ausleihen, neuen Gegenstand hinzufügen, Gegenstand deinventarisieren, neues Userkonto anlegen, bestehendes Userkonto berbeiten).
3. Der User/Admin erhält eine Mitteilung das die Aktion erfolgreich war.
4. Der Use Case ist abgeschlossen.

2.12.5. Alternative Abläufe

<Aktion ist fehlgeschlagen>

Wenn das Schreiben der Änderung in der Datenbank fehlgeschlagen ist: . erhält der User/Admin eine Mitteilung über das Fehlschlagen der Aktion und der Bitte es erneut zu versuchen . der User/Admin muss die Änderung neu veranlassen.

2.12.6. Wesentliche Szenarios

<User leiht einen Gegenstand erfolgreich aus>

Der User wählt einen verfügbaren Gegenstand aus, gibt das Datum der Rückgabe an und wählt "ausleihen" aus. Danach bestätigt er die Aktion. Anschließend erhält er die Meldung, dass die Aktion erfolgreich war.

<User leiht einen Gegenstand nicht erfolgreich aus>

Der User wählt einen verfügbaren Gegenstand aus, gibt das Datum der Rückgabe an und wählt "ausleihen" aus. Danach bestätigt er die Aktion. Anschließend erhält er die Meldung, dass

2.12.7. Nachbedingungen

Die Änderungen sind erfolgreich in der Datenbank gespeichert.

2.12.8. Besondere Anforderungen

Es müssen Systemmeldungen aus dem Backend ausgelesen werden können. <<<

2.13. Use-Case: Nutzungsverlauf anlegen

2.13.1. Kurzbeschreibung

Der Use Case beschreibt das automatische Anlegen einer Gegenstandshistorie.

2.13.2. Kurzbeschreibung der Akteure

<User>

Will die Ausleih- und Wartungshistorie eines Gegenstandes sowie dessen erste Aufnahme in die Datenbank einsehen

2.13.3. Vorbedingungen

1. Der User muss erfolgreich eingeloggt sein
2. Der User muss einen Gegenstand ausgewählt haben

2.13.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Uer einen neuen Gegenstand in die Datenbank aufnimmt.
2. Der User nimmt eine Zustandsänderung an einem Gegenstand vor.
3. Die Zustandsänderungen werden in eine Historie geschrieben.
4. Der User wählt einen Gegenstand aus und ruft die Historie auf.
5. Es werden alle Zustandsänderungen und Wartungen angezeigt.
6. Der Use Case ist abgeschlossen.

2.13.5. Alternative Abläufe

2.13.6. Unterabläufe (subflows)

2.13.7. Wesentliche Szenarios

<Ausleihen eines Gegenstandes>

1. Der User x will Gegenstand y bis zum Datum z ausleihen, gibt alle dafür notwendigen Daten ein und bestätigt die Aktion.

2. Anschließend kann der User x seine Änderung in der Historie von Gegenstand y einsehen.

2.13.8. Nachbedingungen

1. Eine Logtabelle wurde angelegt oder verändert.

2.13.9. Besondere Anforderungen

2.14. Use-Case: Anmeldung ermöglichen

2.14.1. Kurzbeschreibung

Der Use Case beschreibt den prozess das Logins durch einen Benutzer oder Admin.

2.14.2. Kurzbeschreibung der Akteure

Anwender

2.14.3. Vorbedingungen

2.14.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Anwender/Admin die Webseite aufgerufen hat.
2. Der Anwender gibt seine Zugangsdaten aus Benutzername und Passwort ein und wählt "Login".
3. Das System findet Benutzer in der Datenbank und prüft seine Berechtigungen.
4. Das System leitet den Benutzer auf die Hauptseite weiter.
5. Der Use Case ist abgeschlossen.

2.14.5. Alternative Abläufe

Alternativer Ablauf Manuelle Eingabe

Abbruch wenn: . Passwort oder Benutzername nicht gefunden. . Login für Nutzer gesperrt ist (gelöscht)

2.14.6. Unterabläufe (subflows)

<Unterablauf 1>

1. <Unterablauf 1, Schritt 1>
2. …
3. <Unterablauf 1, Schritt n>

2.14.7. Wesentliche Szenarios

<Szenario 1>

1. <Szenario 1, Schritt 1>
2. …
3. <Szenario 1, Schritt n>

2.14.8. Nachbedingungen

<Nachbedingung 1>

2.14.9. Besondere Anforderungen

<Besondere Anforderung 1>

Erweiterungen

2.15. Use-Case: User verwalten

2.15.1. Kurzbeschreibung

Der Use Case beschreibt das Verwalten (anlegen, bearbeiten, löschen) von Benutzern durch einen Admin.

2.15.2. Kurzbeschreibung der Akteure

Admin

Admin möchte Benutzer verwalten.

2.15.3. Vorbedingungen

Der Admin hat die Webseite ausgewählt und ist eingeloggt.

2.15.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn <akteur> <macht>...
2. <Standardablauf Schritt 1>
3. ...
4. <Standardablauf Schritt n>
5. Der Use Case ist abgeschlossen.

2.15.5. Alternative Abläufe

<Alternativer Ablauf 1>

Wenn <Akteur> im Schritt <x> des Standardablauf <etwas macht>, dann . <Ablauf beschreiben> . Der Use Case wird im Schritt <y> fortgesetzt.

2.15.6. Unterabläufe (subflows)

<Unterablauf 1>

1. <Unterablauf 1, Schritt 1>
2. ...
3. <Unterablauf 1, Schritt n>

2.15.7. Wesentliche Szenarios

<Szenario 1>

1. <Szenario 1, Schritt 1>
2. ...

3. <Szenario 1, Schritt n>

2.15.8. Nachbedingungen

<Nachbedingung 1>

2.15.9. Besondere Anforderungen

<Besondere Anforderung 1>

2.16. Use-Case: Backup erstellen

2.16.1. Kurzbeschreibung

Der Use Case beschreibt das Erstellen eines Backup durch einen Admin.

2.16.2. Kurzbeschreibung der Akteure

Admin

Admin möchte einen Backup erstellen.

2.16.3. Vorbedingungen

Der Admin hat die Webseite ausgewählt und ist eingeloggt.

2.16.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Admin die Schaltfläche Backup wählt.
2. Der Adminnwender bestätigt die Meldung zum erstellen das Backups.
3. Das System erstellt eine komprimierte Datei mit allen Daten der Datenbank und legt diese im Archiv-Ordner ab.
4. Das System zeigt dem Admin die Bestätigung der Backup erstellung sowie den Dateiname (Zeitstempel) an.
5. Das System leitet den Admin zurück zum Hauptseite.
6. Der Use Case ist abgeschlossen.

2.16.5. Alternative Abläufe

Alternativer Ablauf

Abbruch wenn: . Meldung nicht bestätigt wird . Rückkehr zum Hauptseite

2.16.6. Unterabläufe (subflows)

<Unterablauf 1>

1. <Unterablauf 1, Schritt 1>
2. …
3. <Unterablauf 1, Schritt n>

2.16.7. Wesentliche Szenarios

<Szenario 1>

1. <Szenario 1, Schritt 1>

2. …
3. <Szenario 1, Schritt n>

2.16.8. Nachbedingungen

<Nachbedingung 1>

2.16.9. Besondere Anforderungen

<Besondere Anforderung 1>

3. System-Wide Requirements: I6 Inventur Verleih

3.1. Einführung

In diesem Dokument werden die systemweiten Anforderungen für das Projekt <Thema> 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.

3.2. Systemweite funktionale Anforderungen

- **SWFR-1: Das System muss alle Inventar-, Ausleih- und Benutzerdaten persistent speichern.**
  Test: Die Datenbank(en) wird/werden mit Testdaten gefüllt. Anschließend wird das System ausgeschaltet. Die Daten müssen bei Wiederinbetriebnahme vollständig vorhanden sein.
- **SWFR-2: Die Kommunikation zwischen Clients und Server muss verschlüsselt sein.**
  Test: Vorhandensein von (TLS) Bei einem MitM-Angriff dürfen keine Daten in Klartext lesbar sein.
- **SWFR-3: Nach dem Aufrufen der Anwendung müssen sich die Clients/User am Server durch die Eingabe von Benutzernamen und Passwort authentifizieren.**
  Test: Werden Security-Token verwendet, ist dies sicher gegen Session-Hijacking?
- **SWFR-4: Die Daten müssen auf dem Server vor unberechtigten Zugriff geschützt sein.**
  Test: Das Ausführen von SQL-Injections darf nicht möglich sein.
- **SWFR-5: Die Passwörter zu den Useraccounts müssen serverseitig durch die Verwendung von Hashfunktionen, Salt, Pepper und Stretching sicher gespeichert werden.**
  Test: In der Datenbank nach Passwörtern suchen (sollten nicht in Klartext sein)

3.3. Qualitätsanforderungen für das Gesamtsystem

3.3.1. Benutzbarkeit (Usability)

- **NFRU-1: Die Sprache der Benutzeroberfläche ist Deutsch.**
- **NFRU-2: Die Startseite soll alle möglichen Aktionen anzeigen, für die der Benutzer die Rechte hat.**
  Test: Unterschiedliche Personen (Admin/ normaler Benutzer) werden unterschiedliche

Aktionen auf der Startseite angezeigt.

- **NFRU-3: Die Eingaben bei der Suchfunktion werden durch Vorschläge unterstützt.**
  Test: Eingabe von nur den ersten 3 Buchstaben eines Gegenstandes in die Suchfunktion. Es wird überprüft, ob der vollständige Name des Gegenstandes angezeigt wird.
- **NFRU-4: Es werden pro Ansicht nur 6 Gegenstände angezeigt.**
  Test: Es werden mindestens 7 Gegenstände im System eingeführt. Es wird überprüft, wie viele Gegenstände auf der Ansicht dargestellt werden.
- **NFRU-5: Ein Gegenstand, der sich im System befindet, soll mit 5 Klicks oder weniger ausgeliehen oder zurückgegeben werden können.**
  Test: Unterschiedliche Gegenstände werden ausgeliehen oder zurückgegeben und die Klicks werden gezählt.
- **NFRU-6: Alle im System befindlichen Daten (Rückgabedatum, Wartungsdatum etc.) sollen in folgendem Format angezeigt werden: DD.MM.JJJJ.**
  Test: Es werden einige Anwendungsfällen mit Datumseingabe, durchgeführt. Anschließend wird kontrolliert, ob das System das Datum im richtigen Format gespeichert hat.
- **NFRU-7: Die Gegenstände müssen über einen Barcode gefunden werden können**
  Test: EAN-13 Barcode aus Seriennummer generieren und mit einem Barcodeleser scannen.

3.3.2. Zuverlässigkeit (Reliability)

- **NFRR-1: Es darf kein falscher Benutzername oder falsches Passwort angegeben werden.**
  Test: Zur Überprüfung wird getestet, ob in diesen Anwendungsfällenin entsprechender Hinweistext angezeigt wird.
- **NFRR-2: Ein neuer Benutzername darf kein breits vergeben Namen erhalten.**
- **NFRR-3: Alle Daten müssen in der Datenbank konstistent nach dem ACID-Prinzip gespeichert werden.**
  Test: Überprüfung auf Inkonsistenz der Daten.
- **NFRR-4: Das System sollte eine Verfügbarkeit von 95% gewährleisten.**
  Test: System gezielt zum Absturz bringen, Zeit bis System wieder verfügbar messen

3.3.3. Effizienz (Performance)

- **NFRP-1: 10 User müssen gleichzeitig das System nutzen können.**
  Test: Mehr Nutzerzugriffe gleichzeitig ausführen.

3.3.4. Wartbarkeit (Supportability)

- **NFRS-1: Vorliegen einer Bedienungsanleitung**
  Test: Ausführen/Starten des Servers von einer projektfremden Person.

3.4. Zusätzliche Anforderungen

3.4.1. Einschränkungen

- SpringBoot
- ReactJS
- min Java 8; empfohlen Java 11
- Systeme:
  - Windows Vista bis 11, Windows Server 2008 R2 SP1 bis 2022
    Linux Oracle / Red Hat / Suse / Ubuntu
    macOS X 10.8.3+ (Mountain Lion) bis macOS 12

Systemanforderungen
ca. 4 GB RAM, 2 GHz Prozessor (min. 1,4 GHz x64), Speicherplatz Festplatte ca. 7 GB

3.4.2. Organisatorische Randbedingungen

- Das System muss von geschultem Personal gepflegt und gewartet werden.
- Neue User sollten mit der Software vor der ersten Nutzung vertraut gemacht werden.
- Die Software ist nur in englischer Sprache gehalten.

3.4.3. Rechtliche Anforderungen

- Die Lizensierung der Anwendungen wird vorausgesetzt.
- Die Einhaltung der DSGVO wird vorausgesetzt und ist vom Anwender zu prüfen.
- Der Urheberrechtsschutz muss vom Anwender gewährleistet werden.

4. Glossar: I6 Inventur Verleih

4.1. Einführung

In diesem Dokument werden die wesentlichen Begriffe aus dem Anwendungsgebiet (Lagerhaltung) der I6 Inventur Verleih definiert. Zur besseren Übersichtlichkeit sind Begriffe, Abkürzungen und Datendefinitionen gesondert aufgeführt.

4.2. Begriffe

Begriff	Definition und Erläuterung	Synonyme
Administrator	Kann Anwender anlegen, bearbeiten und löschen.	Admin
Anwender	Kann Lager einsehen, Zustand und Status des Inventars auslesen, Inventar ausleihen und zurückgeben.	User
Status	Beschreibt den Verfügbarkeitszustand eines Gegenstandes. Kann die Werte verfügbar, ausgeliehen, in Wartung annehmen. Oder beschreibt den Verfügbarkeitszustand eines Lagerfachs. Kann die Werte Gegenstand in diesem Fach wurde ausgliehen, Gegenstand befindet sich in diesem Fach, leer annehmen.	Zustand
Rechte	Regelt die verschiedenen Ebenen und Vererbungen diverser Rechte zum Anlegen, ändern und löschen des Inventars	Userrechte, Anwenderrechte
Gegenstand	Gekennzeichnet durch eine eindeutige Nummer.	Objekt
Anmeldung	… ist der Vorgang der Authentifizierung durch Benutzernamen und Passwort.	Login

4.3. Abkürzungen und Akronyme

Abkürzung	Bedeutung	Erläuterung
xx	xx	xx

4.4. Verzeichnis der Datenstrukturen

Bezeichnung	Definition	Format	Gültigkeitsregeln	Aliase
Anmeldedaten	Zusammensetzung von Benutzername und Passwort.	String	Emailadresse muss `@`-Zeichen und `.` Punkt enthalten.	Login
xx	xx	xx	xx	xx

5. Domain Model: I6 Inventur Verleih

5.1. Allgemeine Informationen

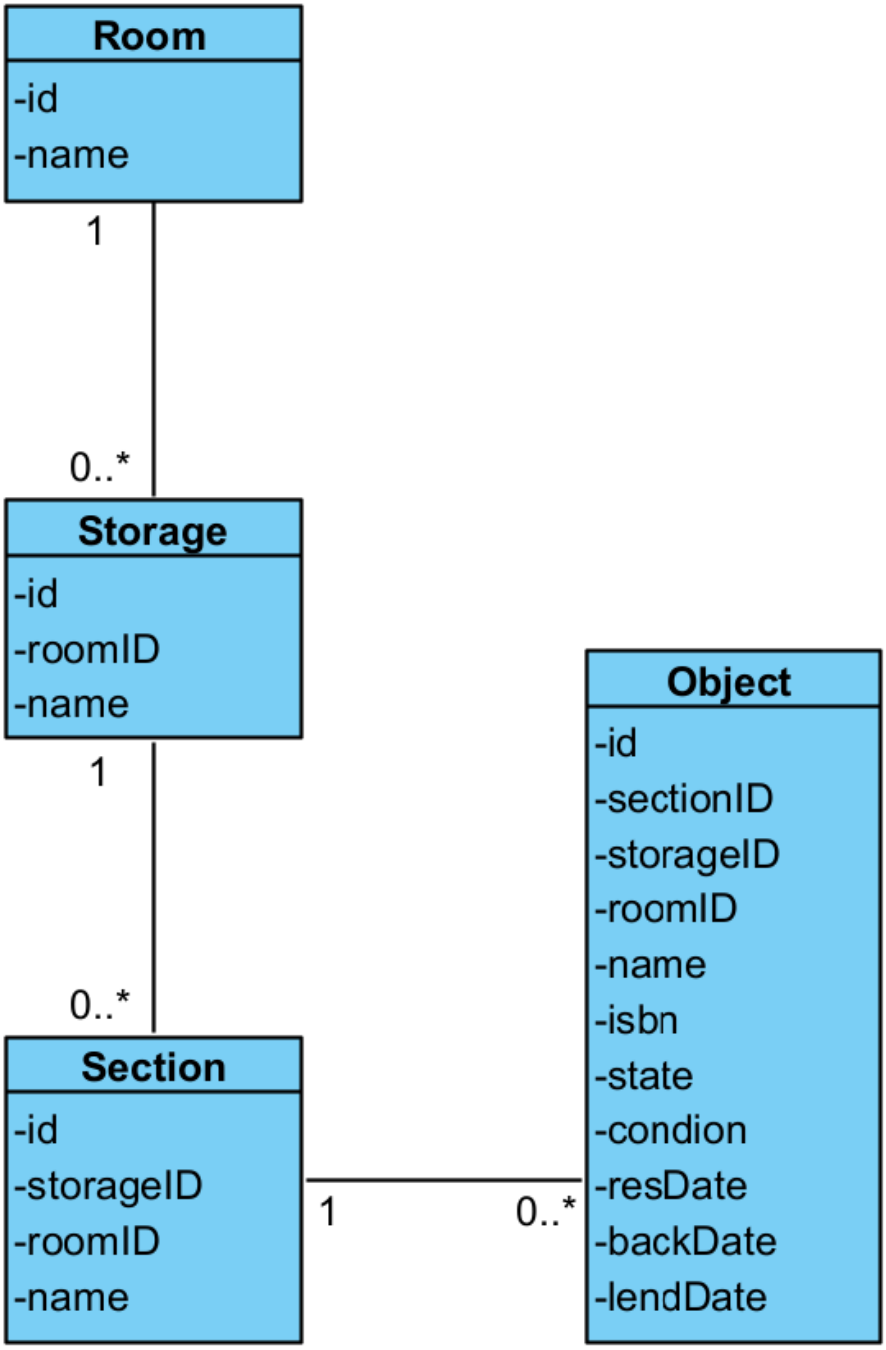


Abbildung 1. Domänenmodell für die Datenbank

Projektdokumentation

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

1. Projektplan: I6 Inventur Verleih

1.1. Einführung

Ziel des Projektplans ist es, einen guten Ablauf für die verschiedenen Stufen des Projektes gewährleisten zu können. Er soll unter anderem die Kommunikationsmittel und Werkzeuge definieren.

1.2. Projektorganisation

Die Aufgaben der Softwareentwicklung werden wie folgt aufgeteilt:

1. Anja Claus (Wing) – Analyse – Analyst
2. Felix Helmert – Analyse und Entwurf
3. Jamal Alkharrat – Entwurf – Architect
4. Dominique Linzmajer – Entwurf und Implementierung - Technical Writer
5. Moritz Lehmann – Implementierung - Developer
6. Silko Grellmann (Verwaltungsinfo) – Analyse/Test – Deployment Engineer
7. Jonas Koloska (Winfo) – Test - Tester
8. Fabian Krähmer – Projektmanagement

1.3. Praktiken und Bewertung

1.3.1. Kommunikationsmedien

Microsoft Teams - Plattform für Meetings

Discord/WhatsApp - Plattform für Chats und Ideenfindung

1.3.2. Transportmedien

Github - Plattform für Austausch des Programmcodes und Aufgabenverteilung

OneDrive - Plattform für Datenablage

NextCloud - Plattform für Backups

1.3.3. Entwicklungsumgebung

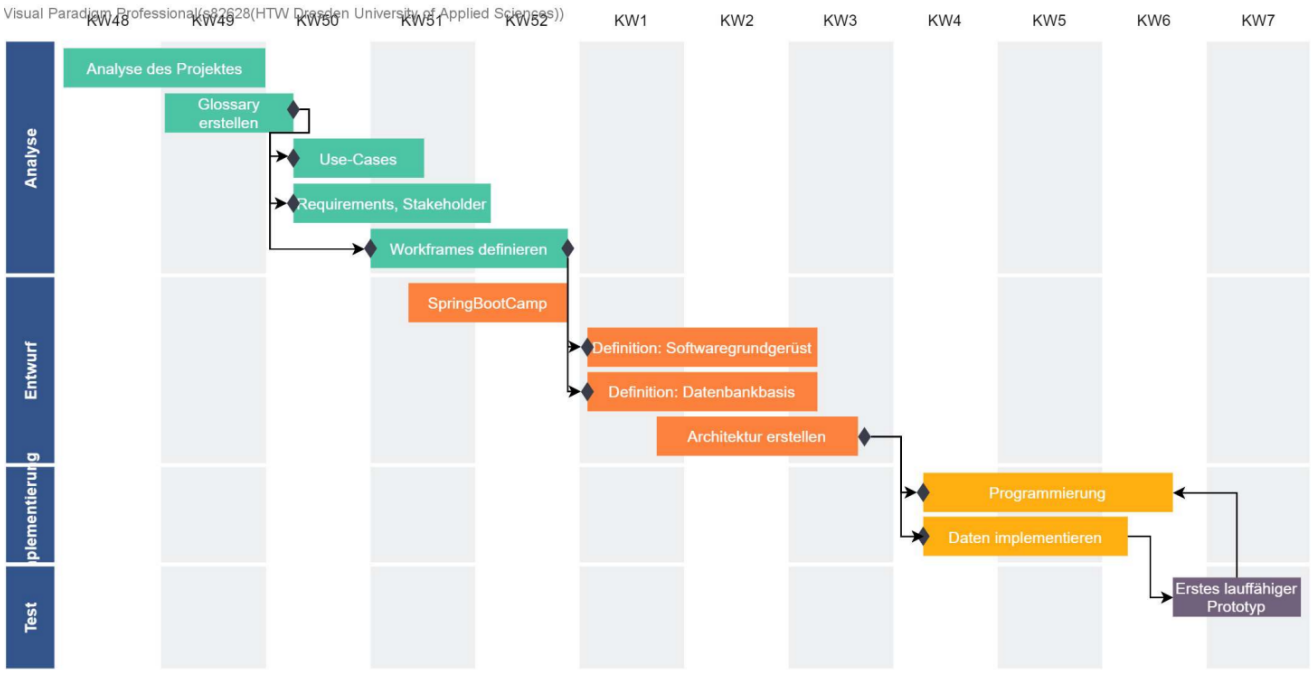
VS Code - Plattform für Entwicklung

1.3.4. Textverarbeitung

Microsoft Word - Anfertigung von Protokollen für Meetings

Microsoft Excel - Kurzfristige Ordnung von Daten/Aufgabenzuordnung

1.4. Meilensteine und Ziele



Iteration	Primary objectives (risks and use case scenarios)	Scheduled start or milestone	Target velocity
LCO	• Aufgabe verstehen • Vision erarbeiten	Date 15.11.21	KW50
LCA/1	• Analyse • Use-Cases, Stakeholder erarbeiten	Date 10.12.21	KW52
LCA/2	• Analyse • Design, Wireframes, Design, Domain Model, Vision, SWR	Date 01.01.22	KW04

1.5. Deployment

1.6. Erkenntnisse (Lessons learned)

2. Risikoliste: I6 Inventur Verleih

In diesem Dokument sind die wesentlichen Risiken des Projekts aufgeführt.

2.1. 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)

2.2. 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

ID	Datum	Name	Beschreibung	Typ	IMP	PRB	MAG	Owner	Gegenmaßnahme
1	25.11.2021	Teammeetings generell unorganisiert	Teammeetings finden zu lang statt und führen zu wenig Ergebnissen	keine Agenda festgelegt	3	25%	0,75	Fabian Krähmer	Agenda definieren und Meetings nur bei Ergebnissen ansetzen
2	08.12.2021	Absage eines Teammitglieds	Mitglied des Teams kann dauerhaft nicht mehr am Projekt teilnehmen	wegen Krankheit	4	20%	0,8	Fabian Krähmer	Backup-Rollen nutzen und Know-How zwischen diesen Personen kontinuierlich teilen
3	13.12.2021	Wissen eines Teammitglieds	Mitglied des Teams hat unzureichendes Wissen über die Vorgehensweise	keine Beschäftigung mit der Matrie	3	30%	0,9	Fabian Krähmer	Mitglied setzt sich erneut an die Vorlesungen/Übungen

3. Iteration Plan (Analyse) 01: I6 Inventur Verleih

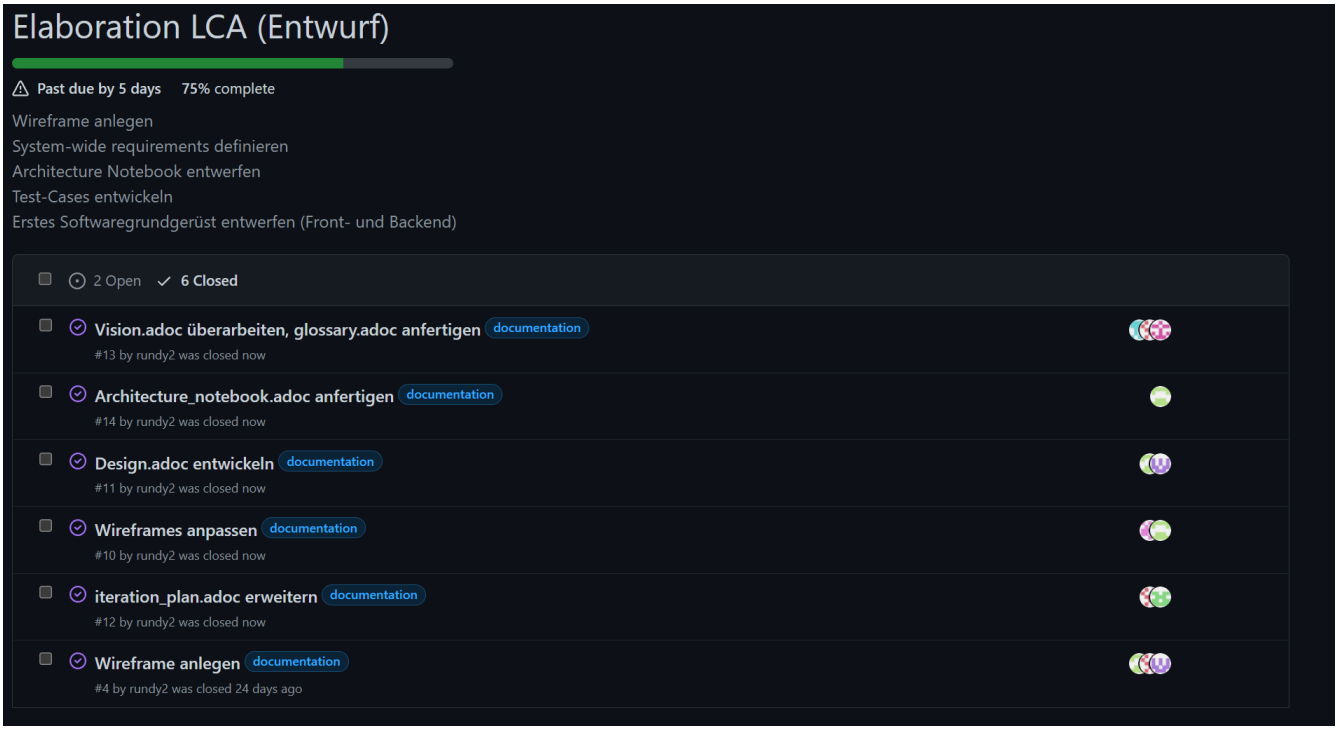
3.1. Meilensteine

Meilenstein	Datum
Beginn der Iteration	15.12.2021
Ausarbeitung Wireframes	28.12.2021
Ausarbeitung Design	28.12.2021
Ende der Iteration	31.12.2021

3.2. Wesentliche Ziele

3.3. Aufgabenzuordnung

Die in dieser Iteration geplanten Aufgaben sind in der Work Items List dargestellt.



3.4. Probleme (optional)

Problem	Status	Notizen
Jamal oft nicht anwesend	gelöst	Einführung von Standards für Kommunikation (Zweischienen-System)

3.5. Bewertungskriterien

3.6. Assessment

Alphas the things to work with

Software System		(0/6)
Requirements	COHERENT	(3/6)
Team	PERFORMING	(4/5)
Stakeholders	IN AGREEMENT	(4/6)
Opportunity	IDENTIFIED	(1/6)
Way of Working	IN PLACE	(4/6)
Work	UNDER CONTROL	(4/6)

Alphas Overview

Entwurfsdokumentation

- Architektur-Notizbuch
- Test Cases
- Design

1. Architecture Notebook: I6 Inventur Verleih

1.1. Zweck

Dieses Dokument beschreibt die Philosophie, Entscheidungen, Nebenbedingungen, Begründungen, wesentliche Elemente und andere übergreifende Aspekte des Systems, die Einfluss auf Entwurf und Implementierung haben.

1.2. Architekturziele und Philosophie

Das System ist für die Verwaltung der Inventur konzipiert. Die Anwendung ist mit HTW Login zugänglich. Das System ermöglicht den Mitarbeitenden eine bessere Orginisation der Gegenständen sowie Verbrauchsmaterialen.

Für die Akzeptanz des Systems bei den Kunden ist eine gute und vor allem einfache Bedienbarkeit erforderlich.

1.3. Annahmen und Abhängigkeiten

Annahmen:

- Jeder Benutzer besitzt eine stabile Internetverbindung und eine ausreichende Bandbreite.
- Jeder Benutzer verfügt über HTW Login.
- Jeder Benutzer stellt die ausgeliehenen Artikel am richtigen Ort zurück.

Abhängigkeiten:

- Wir sind von der HTW Server und von unserer Datenbank abhängig.

1.4. Architektur-relevante Anforderungen

Funktion:

- **SWFR-1:** Das System muss alle Inventar-, Ausleih- und Benutzerdaten persistent speichern.
- **SWFR-2:** Die Kommunikation zwischen Clients und Server muss verschlüsselt sein.

- **SWFR-3:** Nach dem Aufrufen der Anwendung müssen sich die Clients/User am Server durch die Eingabe von Benutzernamen und Passwort authentifizieren.+
- **SWFR-4:** Die Daten müssen auf dem Server vor unberechtigten Zugriff geschützt sein.
- **SWFR-5:** Die Passwörter zu den Useraccounts müssen serverseitig ~~durch die Verwendung von Hashfunktionen, Salt, Pepper und Stretching~~ sicher gespeichert werden.

Effizienz (Performance):

- **NFRP-1:** 10 User müssen gleichzeitig das System nutzen können.

Zuverlässigkeit (Reliability):

- **NFRR-1:** Es darf kein falscher Benutzername oder falsches Passwort angegeben werden.
- **NFRR-2:** Ein neuer Benutzername darf kein breits vergeben Namen erhalten.
- **NFRR-3:** Alle Daten müssen in der Datenbank konstistent nach dem ACID-Prinzip gespeichert werden.
- **NFRR-4:** Das System sollte eine Verfügbarkeit von 95% gewährleisten.

1.5. Entscheidungen, Nebenbedingungen und Begründungen

1. Wir nutzen MySQL Datenbanksystem, damit wir Datentypen gut abbilden können.
2. Wir nutzen React beim Frontend (User-Interface)
3. Wir nutzen Bearer Tokens beim Einloggen, um unberechtigte Zugriffe auf unsere Daten zu verhindern.
4. Wir definieren Typen, um eine einfache Erweiterbarkeit zu ermöglichen

1.6. Architekturmechanismen

Doku "Concept: Architectural Mechanism"

1. Wir haben in der Analyse die Entscheidung getroffen, dass unsere Daten persistent (SWFR-1) sein müssen. Auf der Grundlage dieser Ausgangspunkte hatten wir im Entwurf verschiedene Möglichkeiten, unsere Daten zu speichern. Von den drei Möglichkeiten SQL-Datenbanken, NoSQL-Datenbanken oder einfache Dateien entschieden wir uns für eine SQL-Datenbank, da dies unsere Datentypen gut abbilden kann und wir bereits Erfahrung mit diesen haben. In der Implementierung hatten wir die Wahl zwischen verschiedenen Datenbanken. Wir haben uns für MySQL entschieden, da diese Open Source ist.
2. Wir haben die Entscheidung getroffen, dass die Kommunikation zwischen Clients und Server verschlüsselt (SWFR-2) sein muss. Die Daten müssen auf dem Server vor unberechtigten Zugriff geschützt (SWFR-4) sein. Schließlich müssen die Passwörter zu den Useraccounts serverseitig durch die Verwendung von Hashfunktionen, Salt, Pepper und Stretching sicher (SWFR-5) gespeichert werden. Damit kein ungewollte Zugriff auf unsere Anwendung und deren Daten hat, müssen sich Nutzer zunächst mit ihrer Email und einem Passwort registrieren. Die Email

und das Passwort werden in der Datenbank gespeichert, wobei das Passwort verschlüsselt wird, also nicht im Klartext in der Datenbank steht. Wenn sich ein registrierter Nutzer dann einloggt generiert unsere Anwendung einen Sicherheitstoken, welcher einen Tag gültig sein wird. Wir haben uns für die Verwendung des Bearer Tokens entschieden. Die Dauer der Gültigkeit des Sicherheitstoken kann ggf. angepasst werden. Dieser Token wird bei Abfragen von Daten mit an das Backend geschickt welches seine Gültigkeit prüft. Ist der Token gültig werden die Daten an das Frontend bzw. an den Nutzer gesendet. Wenn der Token ungültig ist wird eine entsprechende Fehlermeldung ausgegeben.

3. Beim User-Interface haben wir uns für React entschieden, weil React eine Vielzahl von Funktionen und schöne User-Elemente bietet. Außerdem ist es kostenlos und einfach zu verwenden.

1.7. Wesentliche Abstraktionen

Typen: Typen werden gebildet, um die Daten, die zur Laufzeit im Programm benötigt werden, zur Verfügung zu haben und zwar an unterschiedlichen Stellen im System.

- Room: enthält der Name des entsprechenden Raums an der Hochschule.
- Storage: enthält die Bezeichnung des Lagerplatzes.
- Section: enthält die genauere Beschreibung, wo die Artikel gelagert sind.
- Object: enthält alle relevante Informationen zum Artikel.

Services: Services werden genutzt, um die Verarbeitung von Daten zu übernehmen. Dies geschieht optimalerweise für jeden Typen separat, um eine möglichst geringe Kopplung zu erhalten.

Workflows: Workflows beschreiben den Arbeitsablauf des Programms und bilden die Reihenfolge ab, mit welcher in einem Programm gearbeitet wird.

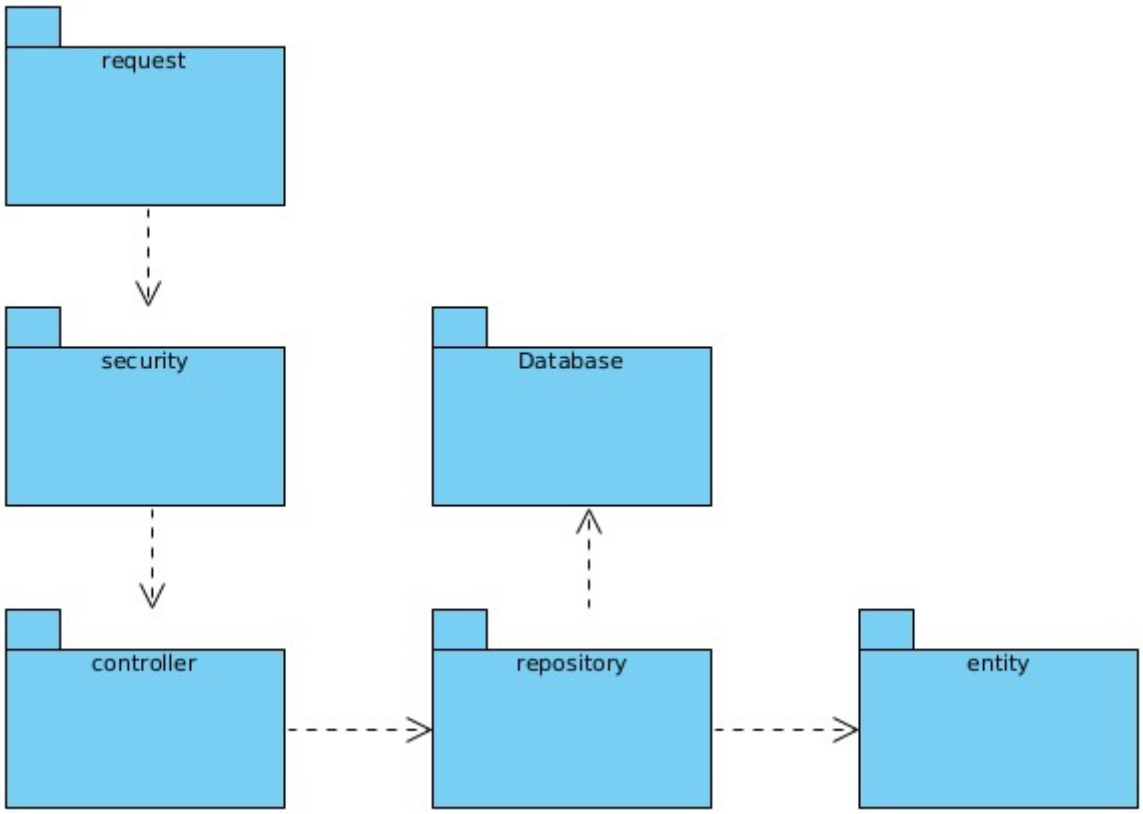
1.8. Schichten oder Architektur-Framework

- Frontend → Backend → Datenbanksystem

1.9. Architektursichten (Views)

Folgende Sichten werden empfohlen:

1.9.1. Logische Sicht



- request: prüft Anfrage des Nutzers und leitet diese passend weiter
- security: prüft ob der Nutzer befugt für entsprechende Anfrage ist
- conntroller: leitet bestimmte Funktion anhand der Anfrage ein (Daten abfragen/ändern/löschen)
- repository: Kommunikation mit Datenbank
- entity: Abbildung der realen Objekte mit denen die Anwendung arbeiten soll

1.9.2. Physische Sicht (Betriebssicht)

- Server-Client

2. Test Cases: I6 Inventur Verleih

2.1. Allgemeine Informationen

Die Test Cases sollen festellen ob verschiedene Funktionen innerhalb des Programms wie erwartet funktionieren.

2.2. Test Cases für "Accountmanagment"

2.2.1. TC01 - Anmelden eines Users/Administrator

Vorbedingungen

- Es besteht eine Internetverbindung

Beschreibung

Der Anwender oder Administrator gibt seine Anmeldedaten an und loggt sich ein.

Eingabedaten

- gültige Anmeldedaten

Erwartetes Ergebnis

1. Das Login ist erfolgreich.
2. Das Login ist nicht erfolgreich.

2.3. Test Cases für "Inventar-/ Datenbankverwaltung"

2.3.1. TC02 - Ausleihen von einem Gegenstand

Vorbedingungen

- Anwender ist mit gültigen Anmeldedaten angemeldet
- Es besteht eine Internetverbindung
- Der Gegenstand ist Verfügbar

Beschreibung

Der Anwender leiht ein Gegenstände aus und dieser wird dem Inventar hinzugefügt.

Eingabedaten

- keine

Erwartetes Ergebnis

1. Der Gegenstand wird erfolgreich ausgeliehen.
2. Das Ausleihen des Gegenstandes ist fehlgeschlagen.

3. Design: I6 Inventur Verleih

3.1. Allgemeine Informationen

3.2. ...