

27.05.2024

StorageShelves

Architekturdoku V1.0

Enis Avdovic, Muhanad Khatib,
Dominik Rundau, Valentin Diehl

1. Einführung und Ziele	3
1.1 Aufgabenstellung	3
1.2 Qualitätsziele	3
1.3 Stakeholder	4
2. Randbedingungen	6
2.1. Inhalt	6
3. Kontextabgrenzung	6
3.1. Fachlicher Kontext	6
3.2. Technischer Kontext	6
3. Bausteinsicht	7
4. Laufzeitsicht	10



Template

Januar 2023

1. Einführung und Ziele

Siehe Anforderungsspezifikation (1.0 Zielstellung)

1.1 Aufgabenstellung

1. Logistiker können die Größe des Lagerraums definieren, in dem die Regale aufgebaut werden.
2. Die Breite und Länge der Regale können von den Logistikern selbst eingestellt werden. Darüber hinaus besteht die Möglichkeit, Regeln zu definieren, die sich nach dem jeweiligen Inhalt und den Eigenschaften der Kisten richten.
3. Die Software ist ebenfalls in der Lage, die Belastung der jeweiligen Regale und Kisten zu berechnen.
4. Die Software verfügt zudem über eine Funktion, mit der alle Kisten automatisch in den Schrank einsortiert werden können. Dazu muss lediglich der Auto-Sortier-Knopf betätigt werden.

1.2 Qualitätsziele

Qualitätskategorie	Qualität	Beschreibung	Szenario (Anforderungsspezifikation)
Usability	Fehler ausbessern	Undo und Redo ist mithilfe von strg-Z/Y möglich.	N/A
	Automatisch einsortieren	Bietet dem Nutzer die Möglichkeit, das einsortieren zu übernehmen	3.5

	Regal auf zulässige Platzierung überprüfen	System verweist auf Konflikte, im bezug zu platzierten Kisten, im bestehenden Regal	3.7
Leistung	Schnelles Laden und speichern von Profilen	Selbst mit den Schlechtesten Mindestanforderungen soll das Speichern und Laden von Profilen max 3 Sekunden Andauern	
Sicherheit	Profile Speicherung nur Lokal	Alle Profile werden lokal gespeichert und können von keiner ausstehenden Entität Remote zugegriffen werden	

1.3 Stakeholder

Rollenname	Kontakt	Erwartungen
Gastronomen	Restaurantbesitzer und -manager	Effiziente Verwaltung der Küchen- und Vorratslager, einfache Bedienung der Software
Logistikunternehmen	Logistikunternehmen	Optimierung der Lagerverwaltung, Kosteneffizienz, einfache Integration in bestehende Systeme
Lagerverwalter	Personen vor Ort im Lager	Übersichtliche und sichere Organisation des Inventars, Vermeidung von Überlastung der Regale
Mitarbeiter (Endnutzer)	Vorgesetzter	Benutzerfreundliche Oberfläche, schnelle und intuitive Bedienung, zuverlässige Fehlerwarnungen

Softwareentwickler und IT-Support	Abteilungsleiter und Einzelne Mitarbeiter	Wartung und Weiterentwicklung der Software, Unterstützung der Nutzer bei technischen Problemen
Investoren	Investoren und Finanziers	Rentabilität, Marktwachstum, positive Rendite auf Investitionen

2. Randbedingungen

2.1. Inhalt

Unsere Software muss flüssig und ohne Probleme auf einem System mit den Mindestanforderungen laufen.

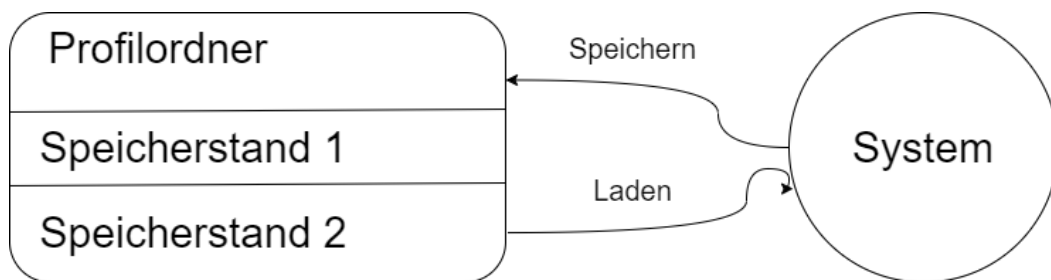
(Siehe Anforderungsspezifikationen “4.1 technischen Voraussetzungen”)

3. Kontextabgrenzung

3.1. Fachlicher Kontext

Unser System bedarf keine Interaktion mit anderen Systemen und wird ohne weitere Schnittstellen in der JRE laufen.

Die einzige kommunikation nach außen findet über das lesen und schreiben der Speicherdaten statt.

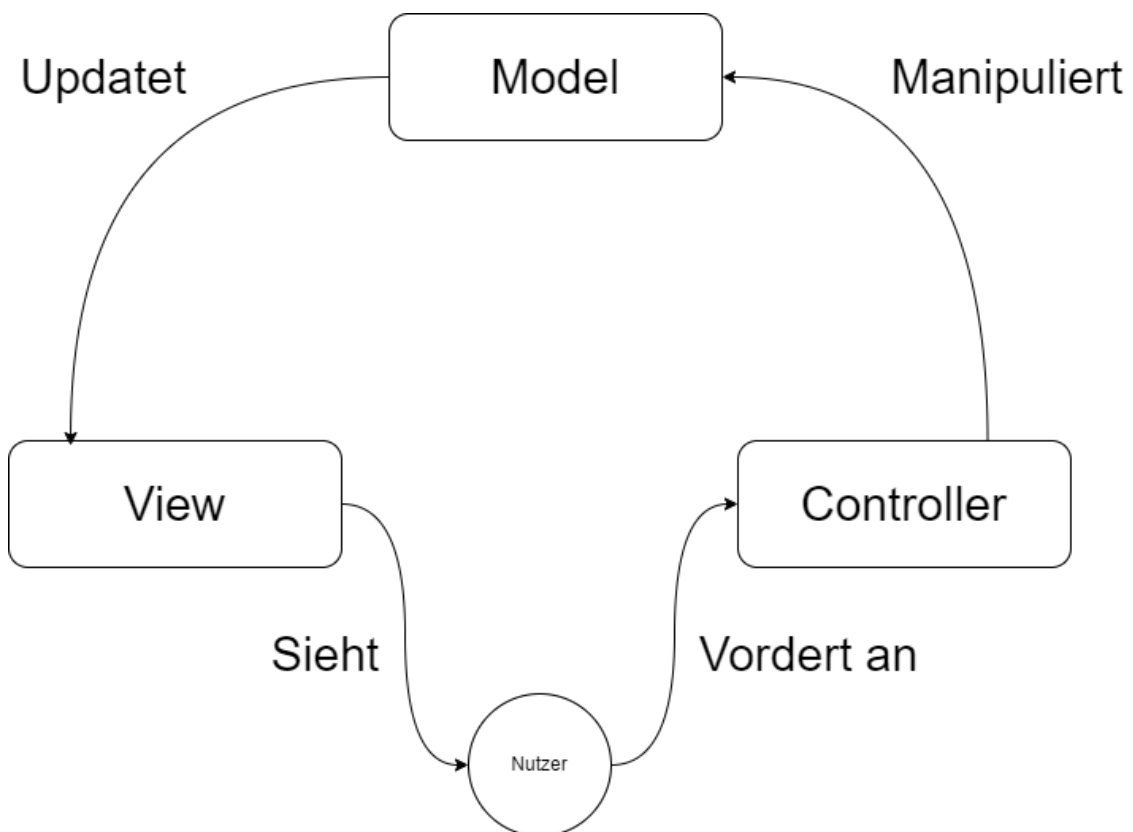
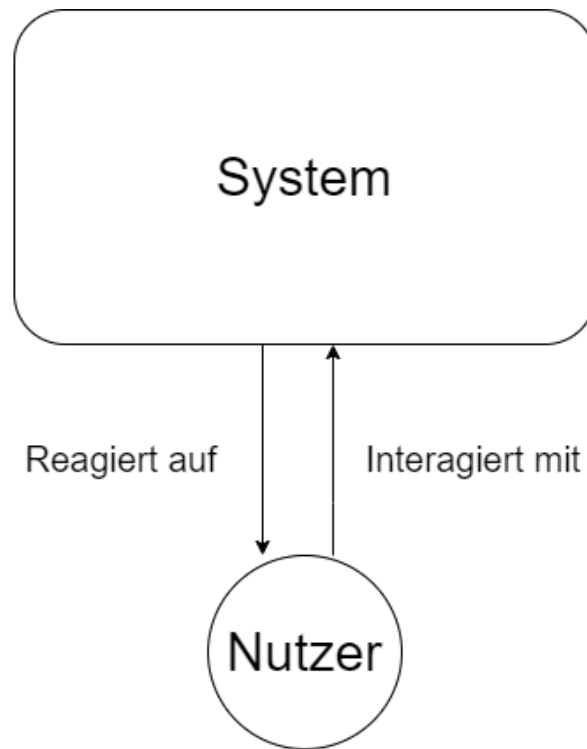


3.2. Technischer Kontext

System muss mit den Mindestanforderungen flüssig laufen

Siehe Anforderungsspezifikationen (4.1 technischen Voraussetzungen)

3. Bausteinsicht



Begründung

Unser Programm wird anhand eines MVC Modells aufgebaut sein, sodass jeder Baustein wartbar und Modular ist.

Enthaltene Bausteine

System: Das System stellt die vollumfängliche StorageShelves Applikation dar.

Nutzer: Beschreibt einen beliebigen Nutzer der mit dem System interagiert

Model: Das Model repräsentiert die logischen Datenstrukturen und Geschäftslogik der Anwendung. Es verwaltet die Daten, Regeln und die Zustände der Anwendung und stellt sie der View zur Verfügung..

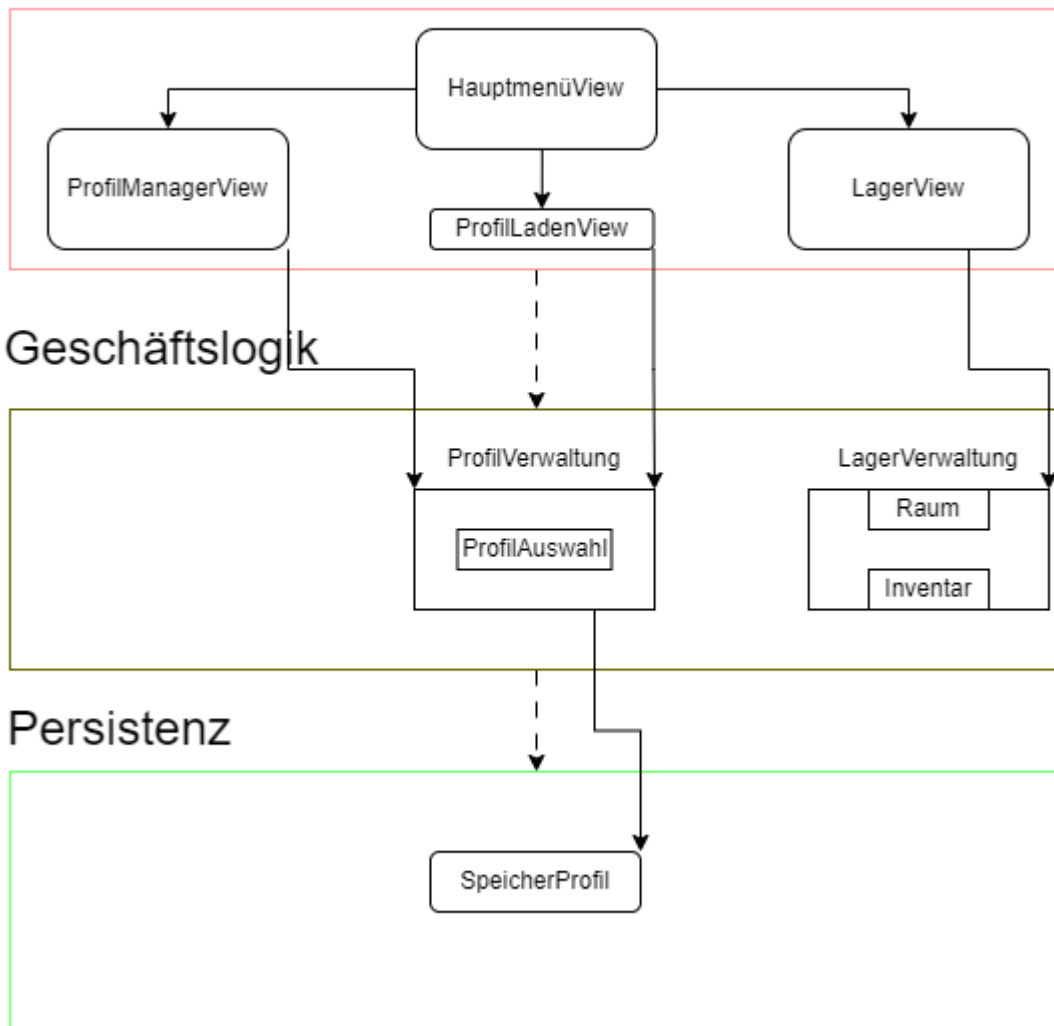
View: Die View ist die Darstellungsschicht der Anwendung. Sie zeigt die Daten, die vom Model bereitgestellt werden, an und stellt die Benutzeroberfläche dar, mit der der Nutzer interagiert.

Controller: Der Controller ist die Steuereinheit der Anwendung. Er verarbeitet Benutzereingaben, interpretiert sie und aktualisiert das Model und/oder die View entsprechend. Der Controller enthält die Kernlogik, die die Interaktion zwischen Model und View steuert.

Wichtige Schnittstellen

Keine Schnittstelle vorhanden

WebUI



1. WebUI:

- HauptmenüView: Der Zugangspunkt für Nutzerinteraktionen.
- ProfilManagerView und LagerView: Separate Ansichten für die Verwaltung von Profilen und Lagerbeständen.
- ProfilLadenView: Eine Ansicht bzw. Overlay, die spezifisch für das Laden von Profilen verantwortlich ist.

2. Geschäftslogik:

- ProfilVerwaltung:
 - ProfilAuswahl: Geschäftslogik, die die Auswahl und die Anpassung von Profilen ermöglicht.
- LagerVerwaltung:
 - Raum und Inventar: Steuern die Organisation und Verwaltung von

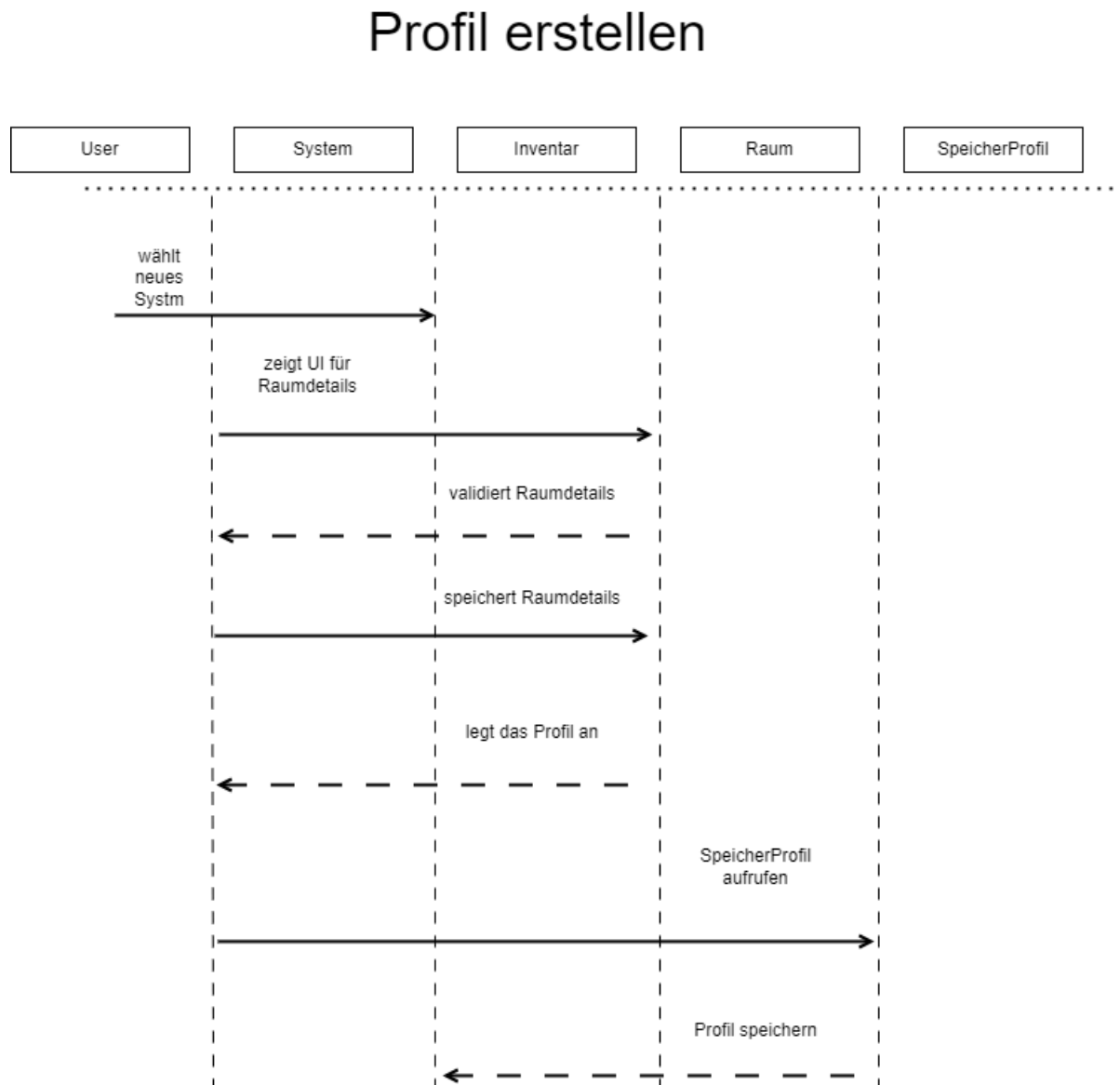
Räumen und Inventar im Lager.

3. Persistenz:

- SpeicherProfil: Eine Komponente zur Speicherung von Profildaten.

4. Laufzeitsicht

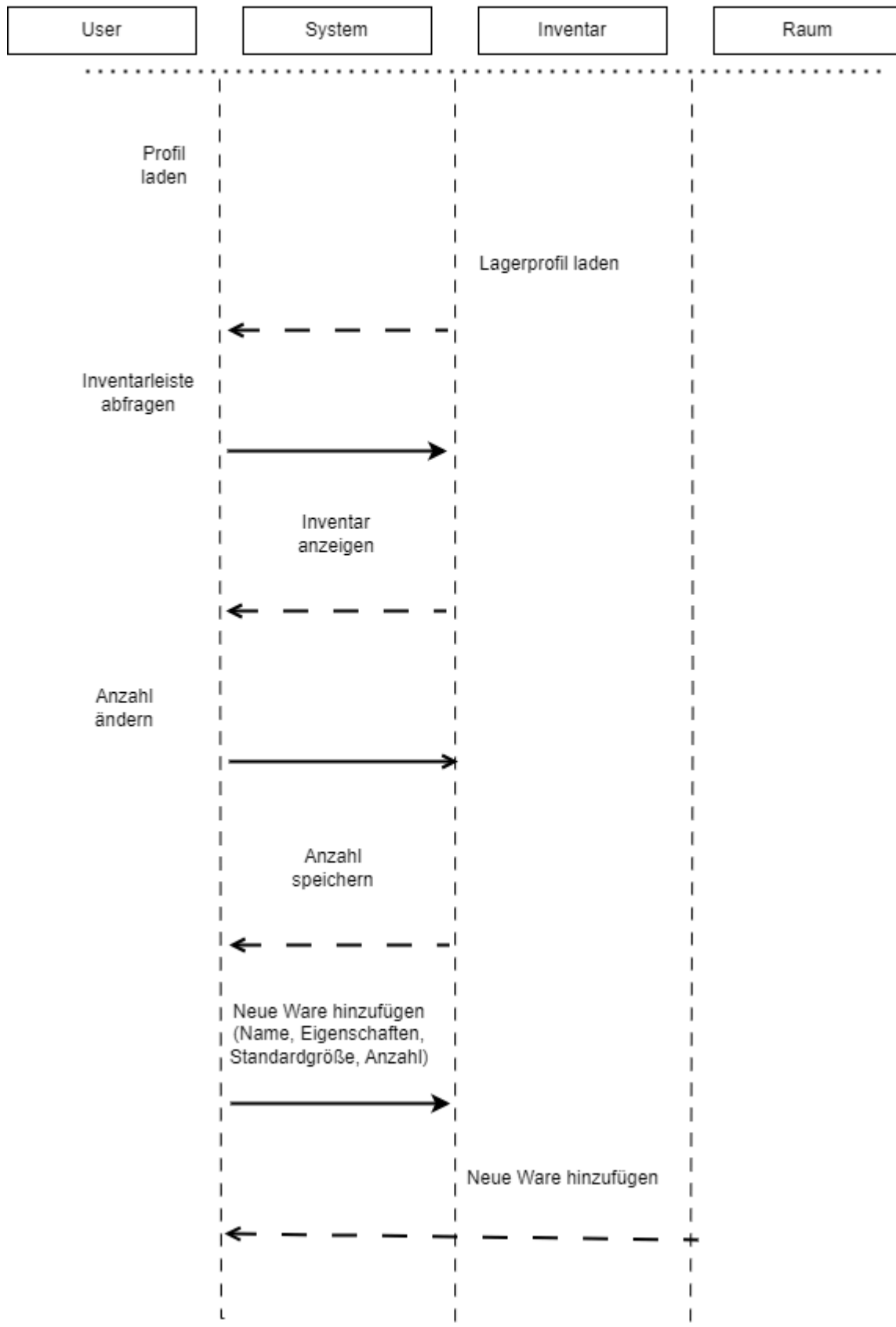
Profil erstellen



Erklärung: siehe Anforderungsspezifikation 3.6. Profil erstellen

Ware hinzufügen

Ware hinzufügen



Erklärung: siehe Anforderungsspezifikation 3.1. Ware hinzufügen

Platzierung eines Kartons

Erklärung: siehe Anforderungsspezifikation 3.7. Platzierung eines Kartons

Regal anlegen

Erklärung: siehe Anforderungsspezifikation 3.4. Regal anlegen

Glossar

Begriff	Definition
MVC	Model View Controller
JRE	Java Runtime Environment