

# Softwaretechnik Praktikum

Medieninformatik  
Prof. Dr. Wolfgang Weitz

Von: Enis Avdovic, Muhanad Khatib, Dominik Rundau, Valentin Diehl

<b>Zielstellung</b>	<b>3</b>
<b>Anwendungs-Szenarien</b>	<b>4</b>
Szenario 1:	4
Szenario 2:	4
Szenario 3:	4
Szenario 4:	4
<b>Funktionale Anforderungen</b>	<b>5</b>
1. Muhanad	5
2. Muhanad	6
1. Valentin	7
2. Valentin	8
1. Dominik	9
2. Dominik	10
1. Enis	11
2. Enis	12
<b>Gegenstandswelt</b>	<b>13</b>
<b>nichtfunktionale Anforderungen</b>	<b>14</b>
technischen Voraussetzungen	14
Qualitätsanforderungen	14
<b>Benutzungsoberfläche</b>	<b>15</b>

# Zielstellung

StorageShelves ist eine Software, die Gastronomen und Logistiker dabei unterstützt, ihr Inventar zu organisieren.

Die Software wird sie dabei unterstützen, ihre Ressourcen effizient und sicher in ihre Regale einzusortieren. Dies erfolgt mit Hilfe einer GUI, die Regale eines Lagerraums darstellt, welche mit verschiedenen großen Kisten mit unterschiedlichen Eigenschaften befüllt werden können.

Dabei ist es möglich, die Größe des Schrankes sowie der Kisten und den Raum individuell anzupassen. Zudem lassen sich Regeln definieren, je nach Inhalt und Eigenschaften der Kisten, sodass der Benutzer selbst definieren kann, welches Produkt wie und neben welchen Waren die Kiste gelagert werden kann.

Die Software ist ebenfalls in der Lage, die Belastung der jeweiligen Regale und Kisten zu berechnen. Falls es schnell gehen muss, ist es auch möglich, mithilfe eines Auto-Sortier-Knopfes alle Kisten automatisch in den Schrank einlagern zu lassen.

Die Kisten lassen sich über ein Inventarsystem per Drag und Drop in die jeweiligen Regale ziehen. Falls hierbei der Nutzer die Kisten falsch angeordnet wird, zum Beispiel die Belastungsgrenze überschreitet oder nicht kompatible Kisten nebeneinander lagert, wird dieser durch eine farbliche Hervorhebung und eine Fehlermeldung darauf hingewiesen.

Nach Abschluss der Konfiguration des Raumes kann der Nutzer diesen abspeichern und in Zukunft abgespeicherte Konfigurationen laden.

Es ist möglich, die Anzahl der eingelagerten Ressourcen in einer simplen Übersicht zu sehen.

# Anwendungs-Szenarien

## Szenario 1:

Gustavo, Gastronom einer Pizzeria, hat eine große Warenlieferung erhalten und steht nun planlos vor der Lieferung und dem Lager.

Bevor er mit dem Einräumen der Kisten beginnt, kommt er auf die gute Idee, zuerst mit dem Programm StorageShelves die Einlagerung zu planen und zu sehen, ob sein Plan funktioniert. Dabei stellt er fest, dass er die Kisten anders einräumen muss als geplant, was ihm viel Zeit und Arbeit erspart, die falsch eingeräumten Kisten wieder umzuräumen. So ist Gustavo in 15 Minuten fertig und kann entspannt weiterarbeiten.

## Szenario 2:

Logi, von Logilstick Logistick, arbeitet schon seit Jahren mit StorageShelves und speichert jeden Tag die aktuelle Tageskonfiguration auf einem Server in der Firma ab.

Als die uralten Rechner der Firma endlich durch moderne ersetzt wurden, musste Logi nicht seine ausgeklügelte und große Raumkonfiguration neu anlegen sondern konnte einfach die letzte Konfiguration in StorageShelves laden und ohne Unterbrechung weiterarbeiten

## Szenario 3:

Giovanni, Besitzer einer Lagerhalle für Pizzazubehör, steht nun vor einem großen Umzug in eine neue und größere Lagerhalle. Damit er sich nicht während der Planung und den ganzen Umzugssorgen um die Verräumung der Kisten kümmern muss, misst er lediglich die Lagerhalle aus und trägt die Anzahl sowie die Größe der möglichen Regale, die in sein Lager passen, ein. Im Anschluss fügt er nur noch die zugelieferten Kisten und deren Anzahl ein und lässt die Lagerplanung, über die "Automatisch sortieren" Funktion, die Software übernehmen. Somit spart Giovanni sehr viel Arbeitszeit und Aufwand.

## Szenario 4:

Joendhardt Bezos von Amazonas Logsitics hat aufgrund riesiger Lagerbestände und Warenankünfte mangelnde Übersicht über sein Lager. Nach unzähligen eingehenden Käsebestellungen muss sich Joendhardt einen Überblick über seinen Warenbestand verschaffen. Dies erfolgt bequem über das Inventar in StorageShelves, in welchem er seine kompletten Lagerbestände auf einen Blick einsehen und anpassen kann.

# Funktionale Anforderungen

## 1. Muhanad

**Titel:** neue Warenlieferung einlagern

**Akteure:** Lagerarbeiter

**Fachlicher Auslöser:** der neue Ware ist eingetroffen und muss zusammen mit dem Bestandsware ins Lager einsortiert werden

**Vorbedingungen:** aktuelle Lagerraum und Ware sind im StorageShelves bereits gespeichert

**Standardablauf:**

1. Lagerarbeiter: Laden das aktuelle Profil
2. Lagerarbeiter: Menge der neuen Packungen eingeben
3. Lagerarbeiter: Eigenschaften an die neue Ware vergeben
4. Lagerarbeiter: Lagerarbeiter passt die neuen Ware direkt mit den alten an (Nach Fifo einsortieren)
5. System: Profil speichern

**Alternative Abläufe / Fehlersituationen / Sonderfälle:**

- 4a Lagerarbeiter merkt, er muss die Ware komplett neu einsortieren
- 4a1 Lagerarbeiter entleert alles aus den Regalen und lagert wieder alles ein
- 4a2 Lagerarbeiter hat nun einen Plan
- System: Profil speichern

**Nachbedingung/Ergebnis:**

- Die neuen Waren werden in die Regale eingeräumt

**Parametrisierbarkeit / Flexibilität:**

- Die Ware löschen und erneut einlagern

**Nutzungshäufigkeit / Mengengerüst:**

- In jeder Zeit möglich

## 2. Muhanad

**Titel:** Ware bestellen

**Akteure:** Gastronom

**Fachlicher Auslöser:** Der Verbrauch von Produkten war zu hoch und der Gastronom möchte wissen, was zu bestellen ist

**Vorbedingungen:** die gespeicherte Anzahl von Produkten

**Standardablauf:**

1. Gastronom: Laden das aktuelle Profil
2. Gastronom: die aktuelle Anzahl der Produkte kontrollieren und die fehlende Anzahl von Produkten feststellen oder (die verbrauchte Anzahl von Produkten eingeben)
3. System berechnet die Anzahl der Produkte neu und liefert die aktuelle und fehlende Anzahl von Produkten
4. Gastronom nimmt die Anzahl zur Kenntnis und speichert den Zustand ab

**Alternative Abläufe / Fehlersituationen / Sonderfälle:**

- 4a Gastronom nutzt die Gelegenheit und löscht die verbrauchten Produkte aus den Regalen
- 4a1 Gastronom spart sich Zeit, damit er das später nicht machen sollte
- 4a2 Gastronom speichert den Plan mit dem neuen Zustand ab

**Nachbedingung/Ergebnis:**

- Profil anzeigen

**Nicht-funktionale Anforderungen**

- Die Berechnung stimmt nicht ganz.

**Parametrisierbarkeit / Flexibilität:**

- Die aktuelle Anzahl von den vorhandenen Produkten sind prüfbar

**Nutzungshäufigkeit / Mengengerüst:**

- Jede Zeit möglich

# 1. Valentin

**Titel:** Regalgrößen anpassen

**Akteure:** Lagerist

**Fachlicher Auslöser:** Vergrößerung der Lagerfläche

**Vorbedingungen:** Vorheriges Lager ist bereits als Profil gespeichert

**Standardablauf:**

1. Lagerist: Bereits vorhandenes Profil laden
2. System: Lädt das Lagerprofil des aktuellen Lagers
3. Lagerist: Hover-Interaktion über eine Regalsäule
4. System: Anzeige verschiedener Interaktionen
5. Lagerist: Wählt "Größe anpassen"
6. Lagerist: Schiebt Regal größer/kleiner
7. Lagerist: Änderungen speichern
8. System: Speichert das Profil des Lagers

**Alternative Abläufe / Fehlersituationen / Sonderfälle:**

- 7a. Lagerist speichert Aktion nicht
  - 7a1 System speichert Änderung nicht beim Schließen.

**Nachbedingung/Ergebnis:**

- Profil mit den Änderungen überschrieben.

**Nicht-funktionale Anforderungen**

**Parametrisierbarkeit / Flexibilität:**

- Für jede Grundfläche individuell anpassbar

**Nutzungshäufigkeit / Mengengerüst:**

- Zu jeder Lageanpassung

## 2. Valentin

**Titel:** Inventur zusammenzählen

**Akteure:** Lagerist

**Fachlicher Auslöser:** Neuer Wareneingang erfordert umräumen im Lager

**Vorbedingungen:** Vorherige Produkte im Lager korrekt im System erfasst

**Standardablauf:**

1. Lagerist: Bereits vorhandenes Profil laden
2. System: Lädt das Lagerprofil des aktuellen Lagers
3. Lagerist: Klick-Interaktion auf Inventarleiste am unteren Bildschirmrand
4. System: Klappt Inventarleiste aus
5. Lagerist: Fügt neue Ware direkt über das "Plus"-Icon hinzu.
6. System: Updated Warenbestand
7. Lagerist: Klickt auf "Neue-Ware"-Icon
8. System: Öffnet Dialogfenster zur Produkterstellung
9. Lagerist: Füllt Name, Eigenschaften, Standardgröße und Anzahl aus
10. System: Fügt neues Produkt in Inventarliste ein

**Alternative Abläufe / Fehlersituationen / Sonderfälle:**

- 5a. Lagerist fügt versehentlich zu viele Produkte hinzu
  - 5a1 Lagerist klickt auf "Minus"-Icon des Systems
  - 5a2 weiter bei 6

**Nachbedingung/Ergebnis:**

- System hat den neuen Warenbestand aufgenommen und wartet auf "Speichern" des Lagerist

**Nicht-funktionale Anforderungen**

- Korrekte Informationen des Lageristen über Warenbestand

**Parametrisierbarkeit / Flexibilität:**

- Alle Produkte können in das System eingepflegt werden

**Nutzungshäufigkeit / Mengengerüst:**

- Zu jedem Wareneingang / Warenausgang



# 1. Dominik

**Titel:** Automatisch einsortieren

**Akteure:** Lagerist

**Fachlicher Auslöser:** Sehr viel neue Ware wurde geliefert. Ein Lagerarbeiter möchte die Funktion unseres Programms nutzen, um Zeit effizient direkt die Ware einzuräumen.

**Vorbedingungen:** Vorheriges Lager ist bereits als Profil gespeichert

**Standardablauf:**

1. Lagerist: Bereits vorhandenes Profil laden
2. System: Lädt das Lagerprofil des aktuellen Lagers
3. Lagerist: Klick-Interaktion auf Inventarleiste am unteren Bildschirmrand
4. System: Klappt Inventarleiste aus
5. Lagerist: Fügt neue Ware direkt über das "Plus"-Icon hinzu.
6. System: Updated Warenbestand
7. Lagerist: Klickt auf "Neue-Ware"-Icon
8. System: Öffnet Dialogfenster zur Produkterstellung
9. Lagerist: Füllt Name, Eigenschaften, Standardgröße und Anzahl aus
10. System: Fügt neues Produkt in Inventarliste ein
11. Lagerist: Wählt die Funktion "Automatisch einsortieren"
12. System: Platziert Ware in den Regalen effektiv
13. Lagerist: Platziert Ware wie im Programm vorgegeben

**Alternative Abläufe / Fehlersituationen / Sonderfälle:**

- 1a. Lagerist: Muss erst ein Profil erstellen (Raumgröße, Regale)

Nachbedingung/Ergebnis:

- Das System hat die Ware selbst einsortiert. Nächster Schritt speichern des Profils

Nicht-funktionale Anforderungen:

- Korrekte Informationen des Lageristen über Warenbestand
- Genug Regale bzw. Platz um die Ware unterzubringen

Parametrisierbarkeit / Flexibilität:

- Leicht zu bedienen, Zeit effizienter

Nutzungshäufigkeiten / Mengengerüst:

- Zu jedem Wareneingang

## 2. Dominik

**Titel:** Erstes Profil erstellen

**Akteure:** Lagerist

**Fachlicher Auslöser:** Noch kein Bestehendes Profil erstellt

**Vorbedingungen:** Öffnen Des Programms

**Standardablauf:**

1. Lagerarbeiter: Öffnet Programm
2. Lagerarbeiter: Wählt neues Lagersystem
3. Lagerarbeiter: Erstellt den Raum (Größe)
4. Lagerarbeiter: Speichert Profil

**Alternative Abläufe / Fehlersituationen / Sonderfälle:**

- 4a. Lagerarbeiter: Speichert nicht das Profil

**Nachbedingung/Ergebnis:**

- Gespeicherter Raum kann nun einfach genutzt werden

**Nicht-funktionale Anforderungen:**

- Zugang zum Programm ist gewährleistet

**Parametrisierbarkeit / Flexibilität:**

- Leicht verständlich

**Nutzungshäufigkeiten / Mengengerüst:**

- Jeder Neukunde Muss ein Profil erstellen
- Bei Erstellen eines neuen Lagerraums

# 1. Enis

**Titel:** Regal auf zulässige Plazierung überprüfen

**Akteure:** Postarbeiter

**Fachlicher Auslöser:** Neue Lieferung braucht einen gültigen Platz auf dem Regal

**Vorbedingungen:** Aktuelles Lager ist bereits als Profil gespeichert

## **Standardablauf:**

1. Postarbeiter: Laden vom aktuellen Profil
2. Postarbeiter: Eingabe der Masse von neuem Paket
3. System: Neues Paket im Inventar abgelegt
4. Postarbeiter: Plaziert das Paket auf Regalbrett
5. System: Erlaubt Plazierung keine Konflikte
6. Postarbeiter: Speichern die neue Konfig
7. System: Neuer Stand ist im Profil gespeichert
8. Postarbeiter: Sortiert die Ware am passenden Ort ein

## **Alternative Abläufe / Fehlersituationen / Sonderfälle:**

- 5a. System: Ungültige Position Karton kollidiert mit anderen Objekten
- 5b. System: Ungültige Position Boden ist überlastet
- 5c. System: Ungültige Position Karton darf nicht neben Lebensmitteln eingelagert werden
  - 5abc1. Postarbeiter: Paket wird wo anders abgestellt weiter mit 5
- 6a. Postarbeiter: Schließt ohne zu Speichern das Programm
  - 6a1: System: Neue Configuration wird nicht übernommen alter Stand bleibt gespeichert (Ende des Ablaufs)

## **Nachbedingung/Ergebnis:**

Postarbeiter weiß wie er das Paket einräumen kann und je nachdem ob Fall 6 oder 6a Erfolgt wird der Stand für den nächsten aufruf gespeichert.

## **Nicht-funktionale Anforderungen**

Lade-, Speicher- + Verarbeitungszeiten < 30sek

## **Parametrisierbarkeit / Flexibilität:**

Ja Paketeigenschaften und Regal Configuration sind variierbar

## **Nutzungshäufigkeit / Mengengerüst:**

Bei der Lagerung (Täglich sehr häufig)

## 2. Enis

**Titel:** Mindesthaltbarketsdatum überprüfen

**Akteure:** Lebensmittelhändler

**Fachlicher Auslöser:** Abgelaufene Ware muss aussortiert werden

**Vorbedingungen:** Aktuelles Lager ist bereits als Profil gespeichert

**Standardablauf:**

1. Lebensmittelhändler: Laden vom aktuellen Profil
2. System: Abgelaufene Ware wird farblich markiert angezeigt
3. Lebensmittelhändler: Entsorgt die Ware die abgelaufen ist
4. Lebensmittelhändler: Löscht die Boxen aus dem System
5. Lebensmittelhändler: Speichert neuen Konfig
6. System: Neuer Stand ist im Profil gespeichert

**Alternative Abläufe / Fehlersituationen / Sonderfälle:**

- 5a. System: Ungültige Position Karton kollidiert mit anderen Objekten
- 5b. System: Ungültige Position Boden ist überlastet
- 5c. System: Ungültige Position Karton darf nicht neben Lebensmitteln eingelagert werden
  - 5abc1. Postarbeiter: Paket wird wo anders abgestellt weiter mit 5
- 6a. Postarbeiter: Schließt ohne zu Speichern das Programm
  - 6a1: System: Neue Configuration wird nicht übernommen alter Stand bleibt gespeichert

**Nachbedingung/Ergebnis:**

Ware ist erfolgreich aussortiert und nicht mehr im aktuellen Stand des Systemes

**Nicht-funktionale Anforderungen**

Lade-, Speicher- + Verarbeitungszeiten < 30sek

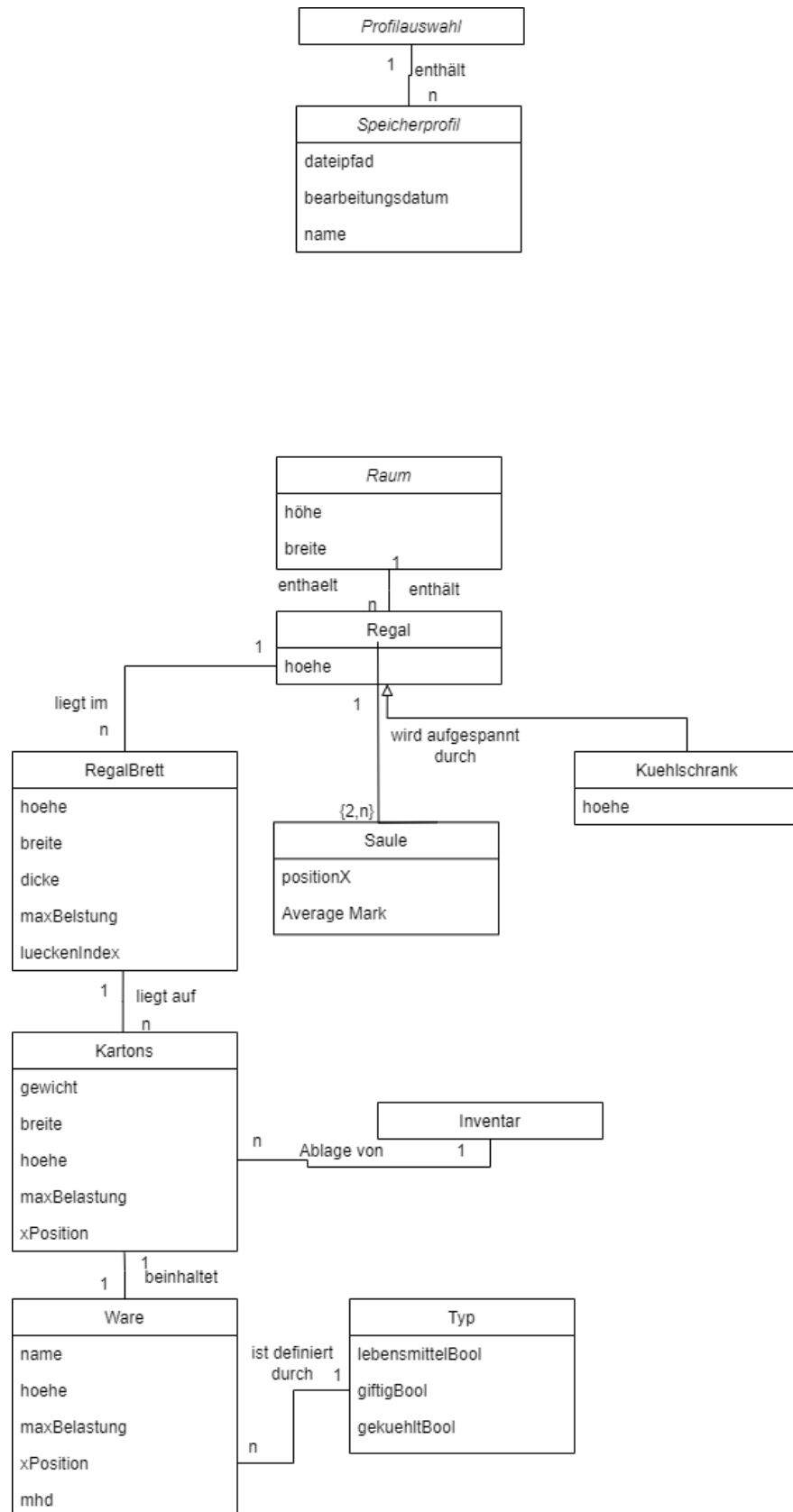
**Parametrisierbarkeit / Flexibilität:**

Ja MHB und Regal Configuration sind variierbar

**Nutzungshäufigkeit / Mengengerüst:**

Bei der Lagerung (Täglich sehr häufig)

# Gegenstandswelt



# nichtfunktionale Anforderungen

## technischen Voraussetzungen

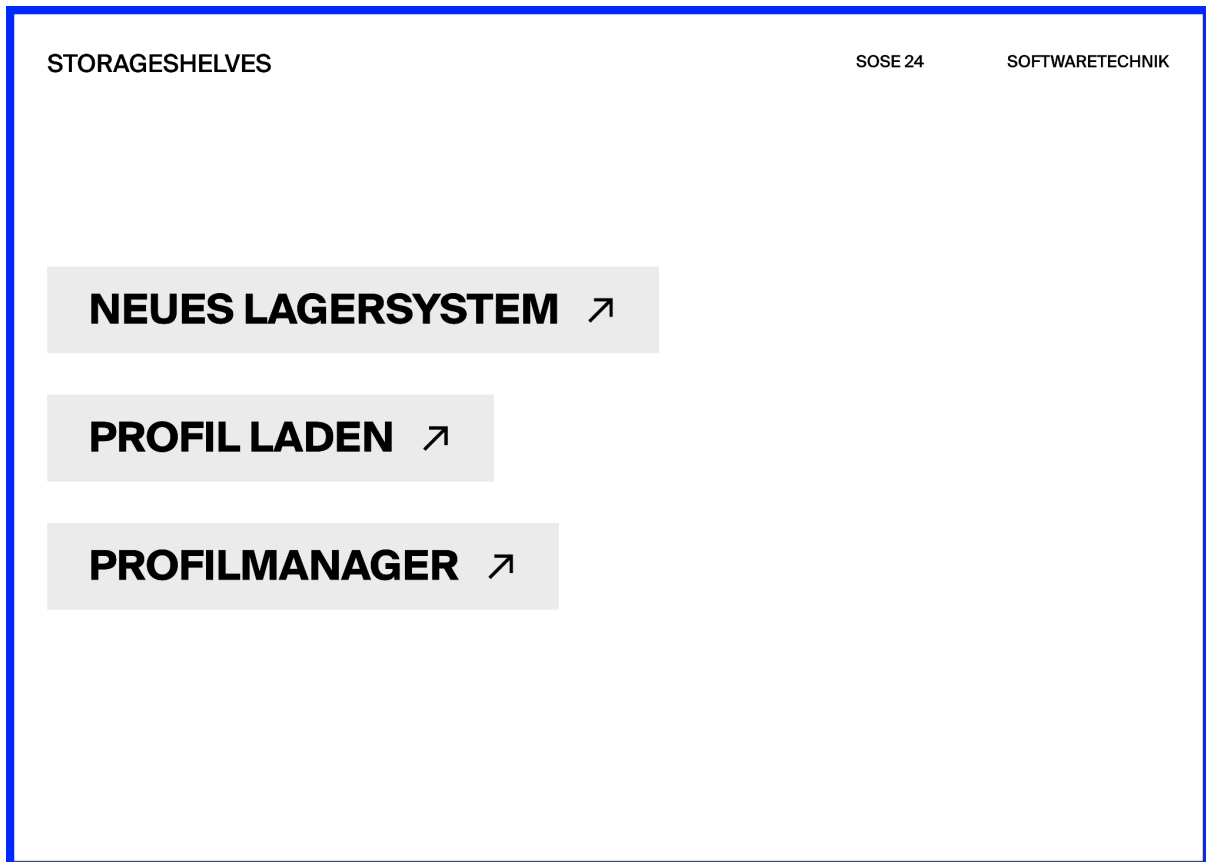
	Erforderlich	Empfohlen
Environment	min. JDK/JRE 21	
Betriebssystem	min. Windows 7 / Dabian based Linux	
CPU	x86 2,5GHz 4x Kerne 4x Threads	x86 3GHz 6x Kerne 12x Threads
Beispiel	Intel Core i5 7400	Intel Core i7 11700
GPU	2GB VRAM 1000Mhz Takt	6GB VRAM 1500Mhz Takt
Beispiel	NVIDIA GeForce GT 1030	NVIDIA GeForce 1660 Ti
RAM	2GB DDR3 1500Mhz Takt	4GB DDR4 DDR4 3200Mhz Takt
Monitor	640 x 480 4:3 (VGA) 30Hz 20"	1920 x 1080 16:9 (FHD) 60Hz 30"
Farbtiefe	min 8bit (16,7 Mio Farben)	
24h Betrieb	Ausreichende Kühlung OLED Minitor nicht empfohlen (Burn in Gefahr)	

## Qualitätsanforderungen

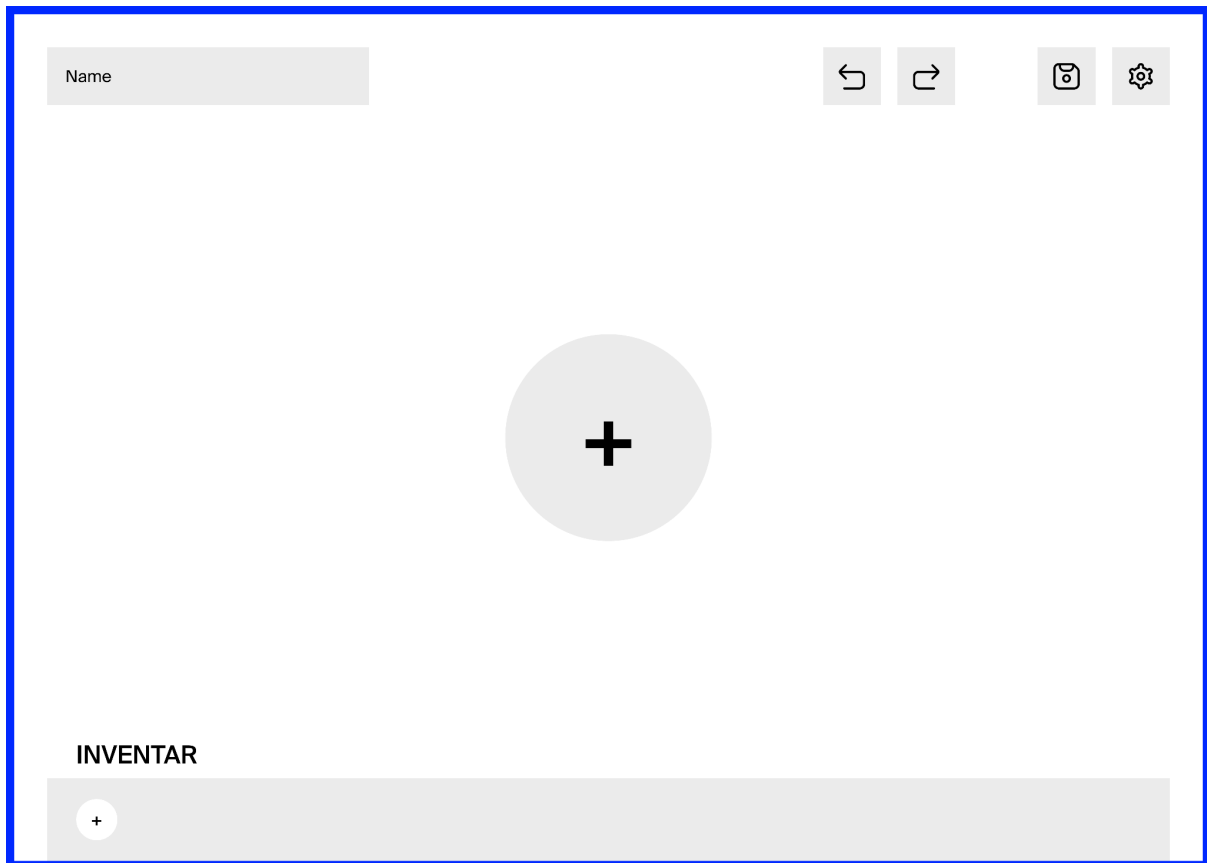
Unser Programm soll möglichst vielseitig ein Regal in jeder möglichen Größe und mit einer nahezu unbegrenzten Anzahl von Waren unterschiedlichster Boxgrößen konfigurieren können.

Dabei sollen die einzelnen Funktionen simpel gehalten werden und das Exportieren und Importieren von Speicherdaten mit wenigen Klicks erfolgen.

# Benutzungsoberfläche



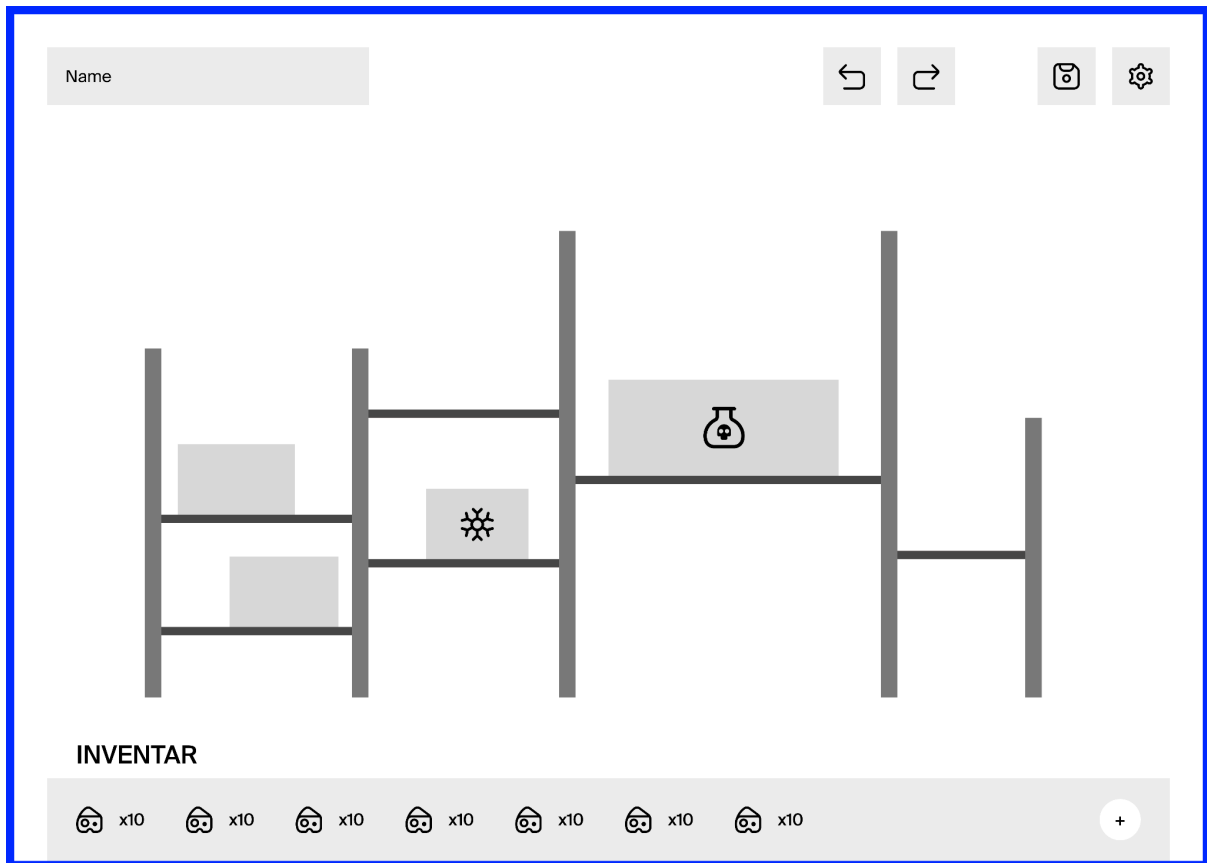
Die Hauptmenü-Ansicht von Storageshelves beinhaltet die 3 Buttons "Neues Lagersystem", bei welchem man in einer leeren Lagerview landet. Beim Klick auf "Profil Laden" öffnet sich ein Dialogfenster zum Laden eines bestehenden Profils. Im Profilmanager kann man Profile verwalten.



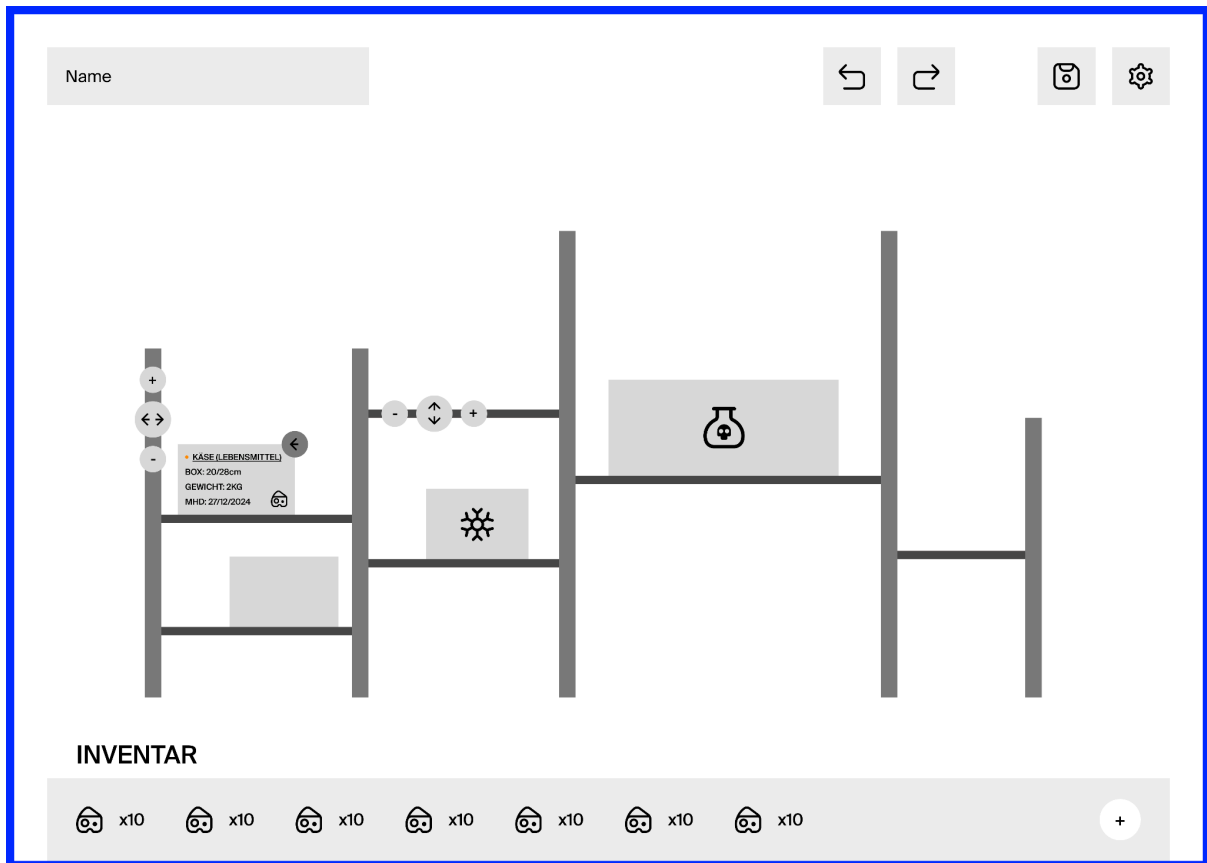
Beim initialen Laden eines neuen Profils, sieht man eine leere View, mit der Menüleiste, die überall in den "Bearbeiten"-Views zu finden ist. Hier kann man den Namen seines Profils anpassen, Aktionen wie Undo und Redo über Buttons nutzen, sowie das Speichern und die Einstellungen schnell und einfach über diese Leiste betätigen.

An der Unterkante findet man das Inventar, das zunächst leer ist. Hier kann man dann Produkte hinzufügen (siehe "Popup"-Screen). Zentral ist ein Plus-Button platziert, über welchen man direkt die erste Säule seines Regals platzieren kann.





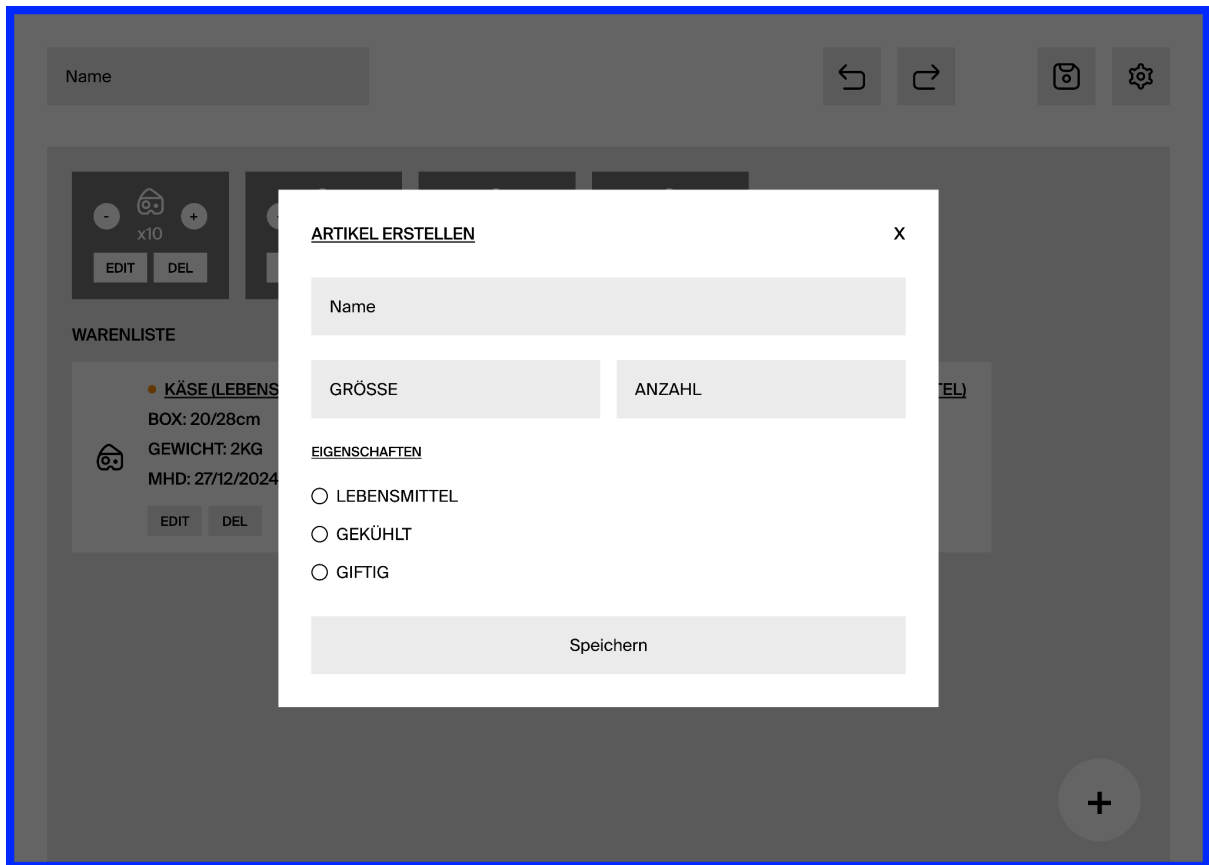
Ein klassisches Profil könnte wie folgt aussehen: Zentral findet man die einzelnen Regale und deren Boxen. Direkt sieht man die Sondereigenschaften der Box (Giftig/Gekühlt). Im Inventar sieht man eine Übersicht an Produkten zur Schnellauswahl. Beim Klicken auf einer dieser Produkte wird das zuletzt angelegte Produkt ausgewählt und kann platziert werden.



Beim Hovern über Regalsäulen stellt das System 3 Buttons zur Verfügung: Regalsäule hinzufügen/entfernen und verschieben der aktuell ausgewählten Säule. Die gleiche Interaktion bietet das System für Regalbretter. Diese kann man lediglich vertikal anstatt horizontal verschieben. Beim Hovern über eine Box sieht man Inhalte und Eigenschaften und hat die Möglichkeit, diese wieder zurück ins Inventar zu verschieben.



Klickt man auf das Inventar an der Unteren Leiste, fährt sich dieses aus und bietet eine Übersicht über alle Waren im Detail, sowie die Option, Produkte hinzuzufügen oder vorhandene zu bearbeiten/Löschen oder deren Anzahl zu verändern.



Möchte man einen neuen Artikel ins System einpflegen, erhält der Nutzer ein Popup, in welchem er Name, Größe der Box, sowie die Anzahl über Input-Felder eingeben kann. Zudem können Eigenschaften hinzugefügt und der Artikel gespeichert werden. Dieser landet anschließend in der Warenliste.