

Lager- und Bestandsverwaltungssystem

Pflichtenheft

Version 1.0

Inhaltsverzeichnis

1	Beschreibung der Projektidee	3
1.1	Ausgangssituation	3
1.2	Zielsetzung	3
2	Muss- und Kann-Kriterien	4
2.1	Muss-Kriterien	4
2.2	Kann-Kriterien	4
3	Use-Case Diagramme	5
4	Projektplan	6
5	Produktumgebung	6
5.1	Definitiv enthalten:	6
5.1.1	Visual Studio	6
5.1.2	C#	6
5.1.3	MSSQL	6
5.1.4	ASP.NET Core	6
5.1.5	Entity-Framework	6
5.1.6	Caliburn.Micro	6
5.2	Möglicherweise enthalten:	7
5.2.1	SWAGGER	7
5.2.2	Dapper	7
5.2.3	Automapper	7
5.2.4	xUnit	7
5.2.5	MahApps.Metro	7
6	GUI	8
6.1	Login	8
6.2	Bestand	8
7	DB-Entwurf	9
8	GIT-Repository	9
9	Testplan	10

1 Beschreibung der Projektidee

1.1 Ausgangssituation

Das Programm ist für die Ausführung auf Windowsrechnern konzipiert und soll in einer Datenbank zentral verwaltet werden, um den Zugriff von mehreren Nutzern simultan zu gewähren.

Mittels API-Controller wird die Authentifizierung einzelner Nutzer gewährleistet und dies als Erweiterungspunkt konzipiert, um eventuell zusätzliche Endpunkte für den Gebrauch der Software vorzubereiten, damit das Hinzufügen weiterer Zugangspunkte in der Zukunft einfacher umzusetzen ist.

Die Anwendung soll in einer Firmenumgebung als Lager und Bestandsverwaltungstool eingesetzt werden. Dies soll es ermöglichen Abgänge, sowie Zugänge zu Verbuchen und sämtliche Änderungen zu verwalten.

1.2 Zielsetzung

Zielsetzung ist es den Arbeitsaufwand der Mitarbeiter zu verringern um Kosten zu sparen.

Dies wird durch zentralisierte Speicherung der Daten im digitalen Format ermöglicht.

Zudem soll durch eine Benutzerfreundliche Oberfläche ein einfaches Verwalten und Einsehen aller bestandbezogenen Informationen möglich sein.

Durch die Zentralisierung wird der Zugriff von mehreren Endgeräten von verschiedenen Nutzern zur gleichen Zeit realisiert.

Mittels Authentifizierung wird ebenfalls der Zugriff und die Berechtigungen der Arbeiter reglementiert und gegebenenfalls eingeschränkt, sofern dies in einzelnen Bereichen erforderlich ist.

Sämtliche Buchungen werden zeitlich festgehalten um vergangene Änderungen nachvollziehen zu können und belegen zu können.

Ebenfalls erlaubt dies fehlerhafte Buchungen und mangelhafte Ware zurückzuverfolgen um die Integrität des Lagers/Lagerinhaltes zu gewährleisten.

2 Muss- und Kann-Kriterien

2.1 Muss-Kriterien

Kriterium	Zeitaufwand
Anmeldung mit Nutzernamen und Passwort	5 Stunden
Abrufen des aktuellen Bestands und Ergebnisse filtern	30 Stunden
Zu- und Abbuchung einzelner Positionen	20 Stunden
Anlegen neuer Materialien	20 Stunden
Korrektur von bereits abgeschlossenen Buchungen	40 Stunden

2.2 Kann-Kriterien

Kriterium	Zeitaufwand
Benachrichtigung bei Erreichen einer Mindestbestandsgrenze	15 Stunden
Automatisches Erzeugen von Bestellungen bei Erreichen einer Mindestbestandsgrenze	20 Stunden
Verschiedene Rollen mit unterschiedlichen Berechtigungen	15 Stunden
Graphische Darstellung der Bestandsänderungen zu einer ausgewählten Position innerhalb eines ausgewählten Zeitraums	40 Stunden
Individuelle Designkonfiguration (z.B. Dark-Mode)	10 Stunden

3 Use-Case Diagramme



4 Projektplan

	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	105	110	115	120	125	130	135	140	145	150	155	160	165	170	175	180	185	190	195	200	205	210	215		
Anmeldung mit Nutzernamen und Passwort																																													
Abrufen des aktuellen Bestands und Ergebnisse filtern																																													
Zu- und Abbuchung einzelner Positionen																																													
Anlegen neuer Materialien																																													
Korrektur von bereits abgeschlossenen Buchungen																																													
Benachrichtigung bei Erreichen einer Mindestbestandsgrenze																																													
Automatisches erzeugen von Bestellungen bei erreichen einer Mindestbestandsgrenze																																													
Verschiedene Rollen mit unterschiedlichen Berechtigungen																																													
Graphische Darstellung der Bestandsänderungen zu einer ausgewählten Position innerhalb eines ausgewählten																																													
Individuelle Designkonfiguration (z.B. Dark-Mode)																																													

5 Produktumgebung

5.1 Definitiv enthalten:

5.1.1 Visual Studio

Die Entwicklungsumgebung von Microsoft in der der Quellcode geschrieben wird

5.1.2 C#

Eine moderne, objektorientierte und typsichere Programmiersprache, die Entwicklern das Erstellen zahlreicher sicherer und robuster Anwendungen, die in .NET ausgeführt werden ermöglicht.

5.1.3 MSSQL

Microsoft-Datenbank-Server

5.1.4 ASP.NET Core

Ein Webframework zum Erstellen von RESTful-HTTP-Diensten

5.1.5 Entity-Framework

Entity Framework Core ist ein moderner Objekt-Datenbank-Mapper für .NET.

5.1.6 Caliburn.Micro

Ein Framework design für alle XAML-Plattformen mit starker Unterstützung von MVVM Entwurfsmustern

5.2 Möglicherweise enthalten:

5.2.1 SWAGGER

Sammlung von Open-Source-Werkzeugen, um HTTP-Webservices zu entwerfen, zu erstellen, zu dokumentieren und zu nutzen.

5.2.2 Dapper

Dapper ist ein simpler Micro-ORM mit sehr hoher Performance

5.2.3 Automapper

Eine simple Bibliothek, die das Mappen von einem Objekt zu einem anderen ermöglicht

5.2.4 xUnit

Ein kraftvolles weit verbreitetes Unit-Test-Framework

5.2.5 MahApps.Metro

Ermöglicht Entwicklern eine Anwendung im Metro bzw. Modern-UI Stil zu designen.

6 GUI

6.1 Login

Lagerverwaltung

Username/Email:

example@company.com

Password:

Login

6.2 Bestand

Lagerverwaltung

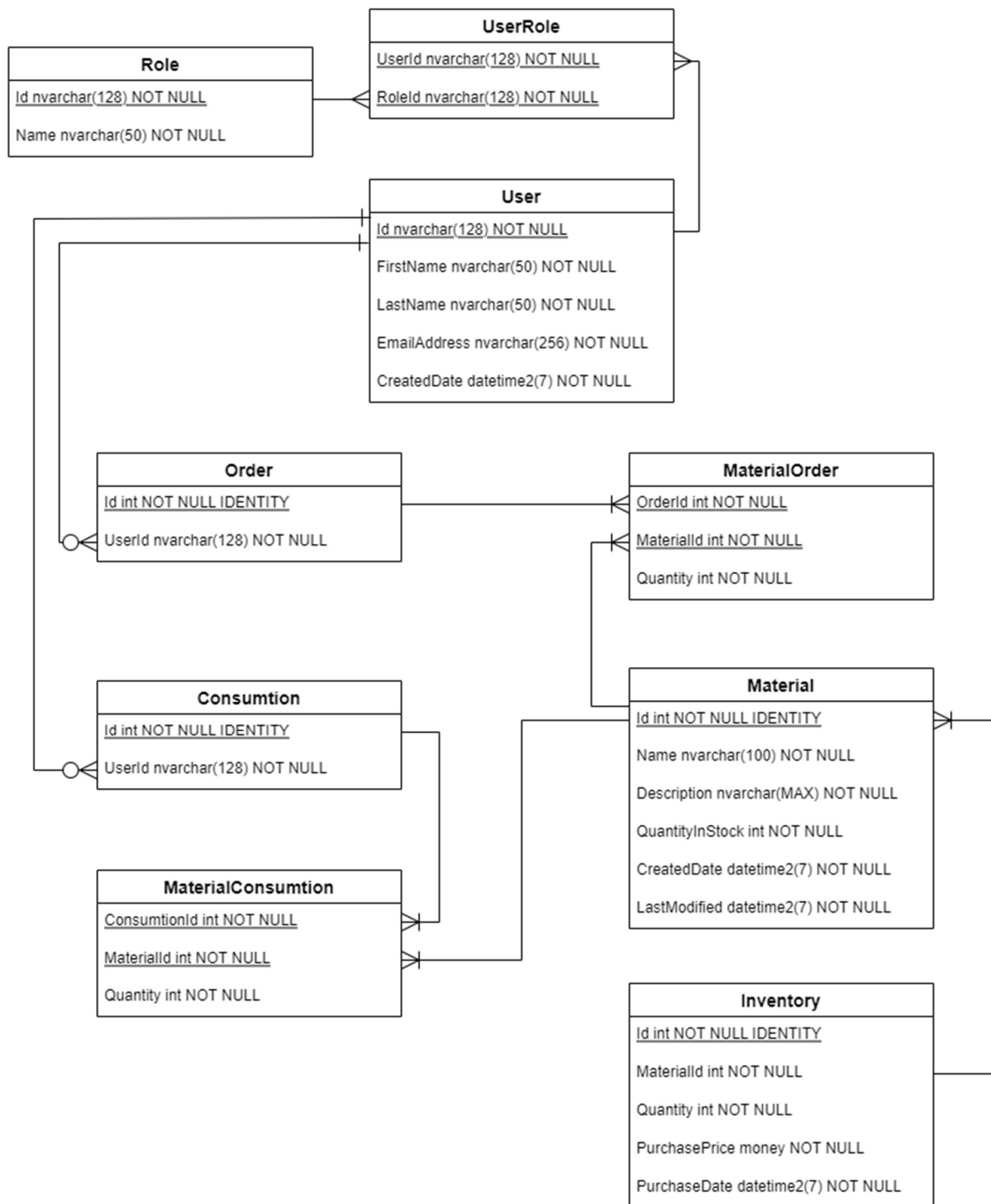
Bestand

Anlegen

Suche:

Name	Beschreibung	Bestand	Erstellungsdatum	Letzte Änderung
Schrauben 60x10	Spax schrauben	100	01.01.2022	05.01.2022
Beilagscheibe 50x5	Beilagscheiben ...	150	04.01.2022	04.01.2022
...

7 DB-Entwurf



8 GIT-Repository

<https://github.com/AdrianLion/Lager-und-Bestandsverwaltung>

9 Testplan

ID:	T01
Beschreibung:	Einloggen
Vorbedingung:	Das Programm ist im Anmeldedialog
Test-Schritte:	<ol style="list-style-type: none"> 1. Im Feld "Username/Email" wird die Email eines Users eingegeben. 2. Im Feld "Password" wird ein passendes Passwort eingegeben. 3. Der Login wird durchgeführt.
Erwartetes Resultat:	Der User ist eingeloggt und die Hauptansicht wird geladen
ID:	T02
Beschreibung:	Filtern des angezeigten Bestands
Vorbedingung:	Der User ist eingeloggt
Test-Schritte:	Im Suchfeld wird ein Artikel eingegeben
Erwartetes Resultat:	Die Produktansicht wird nach dem Suchkriterium eingeschränkt
ID:	T03
Beschreibung:	Entnahme von Material aus dem Lager
Vorbedingung:	<ol style="list-style-type: none"> 1. Der User ist eingeloggt 2. Der User besitzt die nötigen Rechte 3. Der Bestand ist ausreichend
Test-Schritte:	<ol style="list-style-type: none"> 1. Das zu entnehmende Material wird ausgewählt 2. Die Menge wird angegeben 3. Die Entnahme wird durchgeführt
Erwartetes Resultat:	Der Bestand verringert sich um die angegebene Menge

ID:	T04
Beschreibung:	Neues Material anlegen
Vorbedingung:	<ol style="list-style-type: none"> 1. Der User ist eingeloggt 2. Der User besitzt die nötigen Rechte 3. Das Material ist noch nicht vorhanden
Test-Schritte:	<ol style="list-style-type: none"> 1. Die benötigten Informationen über das Material werden eingetragen 2. Das Material wird angelegt
Erwartetes Resultat:	Das neue Material wird dem Lager hinzugefügt
ID:	T05
Beschreibung:	Material zubuchen
Vorbedingung:	<ol style="list-style-type: none"> 1. Der User ist eingeloggt 2. Der User besitzt die nötigen Rechte 3. Das Material ist vorhanden
Test-Schritte:	<ol style="list-style-type: none"> 1. Das zu verbuchende Material wird ausgewählt 2. Die Menge wird angegeben 3. Die Buchung wird durchgeführt
Erwartetes Resultat:	Der Bestand erhöht sich um die angegebene Menge