

Bestellsystem KUK

DIPLOMARBEIT

verfasst im Rahmen der

Reife- und Diplomprüfung

an der

Höheren Abteilung für Informatik

Eingereicht von:

Abd Alrahman El-Hamaoui
Fabian Satybaldy-Ulu
Berdan Kadir

Betreuer:

Prof. Dipl.-Ing. Michael Bucek

Projektpartner:

Sina Taghavi, Guenter Mayrhofer, Oberösterreichische Gesundheitsholding GmbH
Medizininformatik und Informationstechnologie

Leonding, August 2023

Wir erklären an Eides statt, dass wir die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die Arbeit wurde bisher in gleicher oder ähnlicher Weise keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Die vorliegende Diplomarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Leonding, August 2023

Abd Alrahman El-Hamaoui, Fabian Satybaldy-Ulu
& Berdan Kadir

Abstract

In the context of this thesis, the KUK ordering system was developed to significantly improve the ordering process for employees of the Upper Austrian Health Holding GmbH. The previous method, based on the use of Excel forms, has been replaced by an advanced web-based application. This new development aims to digitally transform the entire ordering process while ensuring efficient data management on a Microsoft IIS server with SQL database integration.

The application is characterized by a user-friendly interface that allows orders to be recorded directly in the system and tagged with a unique identification number, facilitating clear referencing and tracking. A key aspect of this innovation is the flexibility in managing supplier information, made possible by the easy integration of new suppliers directly into the system. An integrated workflow system optimizes the order approval process by eliminating the need for separate documents and sending automatic notifications to support the process.

Furthermore, the storage of all relevant data in a relational database enables the creation of comprehensive evaluation reports. These reports provide a detailed overview of all order transactions, simplifying both the management and evaluation of order data. A special feature of the system is the ability to export order data, including all attachments, as a PDF file, which significantly facilitates access to and sharing of order information.

With the introduction of the KUK ordering system, the ordering process of the Upper Austrian Health Holding GmbH is not only made more efficient and transparent but also significantly more user-friendly. This project thus makes a substantial contribution to the digital transformation of internal processes and lays the foundation for future expansions and optimizations.

Zusammenfassung

Im Rahmen dieser Diplomarbeit wurde das Bestellsystem KUK entwickelt, um den Bestellprozess für die Mitarbeiter der Oberösterreichischen Gesundheitsholding GmbH signifikant zu verbessern. Die bisherige Methode, basierend auf dem Einsatz von Excel-Formularen, wurde durch eine fortschrittliche, webbasierte Anwendung ersetzt. Diese Neuentwicklung zielte darauf ab, den gesamten Bestellvorgang digital zu transformieren und dabei eine effiziente Datenverwaltung auf einem Microsoft IIS-Server mit SQL-Datenbankintegration zu gewährleisten.

Die Anwendung zeichnet sich durch eine benutzerfreundliche Oberfläche aus, die es ermöglichte, Bestellungen direkt im System zu erfassen und mit einer einzigartigen Kennnummer zu versehen, was eine eindeutige Referenzierung und Nachverfolgung erleichterte. Ein zentraler Aspekt der Innovation ist die Flexibilität bei der Verwaltung von Lieferanteninformationen, ermöglicht durch die einfache Einbindung neuer Lieferanten direkt in das System. Ein integriertes Workflow-System optimierte den Genehmigungsprozess für Bestellungen, indem es den Einsatz separater Dokumente überflüssig machte und automatische Benachrichtigungen zur Unterstützung des Prozesses verschickte.

Darüber hinaus erlaubte die Speicherung aller relevanten Daten in einer relationalen Datenbank die Erstellung umfassender Auswertereports. Diese boten einen detaillierten Überblick über sämtliche Bestellvorgänge und erleichterten sowohl die Verwaltung als auch die Auswertung der Bestelldaten. Eine besondere Funktion des Systems war die Möglichkeit, Bestelldaten inklusive aller Anhänge als PDF zu exportieren, was den Zugang zu und die Weitergabe von Bestellinformationen erheblich vereinfachte.

Mit der Einführung des Bestellsystems KUK wurde der Bestellprozess der Oberösterreichischen Gesundheitsholding GmbH nicht nur effizienter und transparenter, sondern auch deutlich nutzerfreundlicher gestaltet. Dieses Projekt leistete damit einen wesentlichen Beitrag zur digitalen Transformation interner Abläufe und legte den Grundstein für zukünftige Erweiterungen und Optimierungen.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Einblick	1
1.2	Auftraggeber	2
1.3	Ausgangssituation	2
1.4	Ziele	6
1.5	Systemarchitektur	8
2	Projektmanagement	11
2.1	Scrum	11
2.1.1	Warum ist man mit Scrum agil?	11
2.1.2	Rollen	12
2.1.3	Product Owner	12
2.1.4	Scrum-Master	12
2.1.5	Projektteam	12
2.1.6	Methoden und Regeln beim Projekt-Sprint	13
2.1.7	Product Backlog	13
2.1.8	Sprint Planning	14
2.1.9	Burndown Chart	14
2.1.10	Sprint Review	15
2.1.11	Daily Scrum	15
2.2	Versionierungsverwaltung mit GitHub	15
2.2.1	Verwendung	15
2.2.2	Github	16
2.2.3	Versionskontrolle	16
2.2.4	Git	16
2.2.5	Github Desktop	17

3	Technologien	18
3.1	Windows Server 2022	18
3.1.1	Server Manager	18
3.1.2	AD DS	19
3.1.3	DNS	20
3.1.4	IIS	21
3.1.5	Inbound Rules	22
3.1.6	Outbound Rules	22
3.2	Frontend	24
3.2.1	JavaScript	24
3.2.2	Bootstrap	25
3.2.3	Node.js	25
3.2.4	Bootstrap Icons	26
3.2.5	TTF	26
3.2.6	Angular	27
3.2.7	TypeScript Hero	28
3.2.8	IntelliSense for CSS class names in HTML	29
3.2.9	HTML CSS Support	30
3.2.10	Angular Snippets	31
3.2.11	Angular Language Service	32
3.2.12	Move TS	33
3.3	Backend	34
3.3.1	C#	34
3.3.2	ASP.NET	35
3.3.3	Entity Framework Core	36
3.3.4	MYSQL	37
3.3.5	Nuget-Pakete	38
3.3.6	BouncyCastle.Cryptography	38
3.3.7	DotNetZip	39
3.3.8	itext	40
3.3.9	itext.bouncy-castle-adapter	40
3.3.10	Microsoft.AspNetCore.Authentication.Negotiate	41
3.3.11	Microsoft.EntityFrameworkCore.Tools	41
3.3.12	Microsoft.Extensions.Configuration.Json	42
3.3.13	MySql.Data	42

3.3.14	MySqlConnection	43
3.3.15	MySql.EntityFrameworkCore	43
3.3.16	System.DirectoryServices.AccountManagement	44
3.3.17	Swagger	45
4	Implementierung	46
4.1	Prozessablauf	46
4.1.1	Anmeldung	46
4.1.2	Authentifizierung über Active Directory	49
4.1.3	Admin-Dashboard	50
4.1.4	Rollenbasierte Zugriffssteuerung im Active Directory	53
4.1.5	Bestellformular	55
4.1.6	Lieferant anlegen	58
4.1.7	Lieferant bearbeiten	59
4.1.8	Auswertungen	60
4.1.9	Bestell-Übersicht	63
4.1.10	Benachrichtigungs-Seite	67
4.1.11	PDF-Exportseite für Bestellungen mit Datei-Anhängen	76
4.1.12	PDF: Bestellansicht	77
4.1.13	PDF: Bestellung Tabelle	79
4.1.14	PDF-Generierung für Bestellformulare	80
4.1.15	Einstellungen	81
4.2	Frontend	89
4.2.1	REST Implementierung	89
4.3	Backend	92
4.3.1	Technologientscheidungen für das Backend	92
4.3.2	Aufbau des Backends	92
4.3.3	UnitOfWork	98
4.3.4	Funktionsweise des Workflows	105
4.3.5	Email Client	108
5	Deployment	110
5.1	Deployment: Verbindung mit einer Domäne	110
5.1.1	Windows Server 2022 Setup	110
5.1.2	Windows 11 Workstation Setup	110

5.2	Netzwerkeinstellungen	111
5.2.1	Windows Server 2022	111
5.2.2	Windows 11 Pro Workstation	111
5.3	Konfiguration von Windows Server 2022	112
5.4	Serverrollen im Windows Server 2022	113
5.4.1	Gründe für die manuelle Installation	113
5.4.2	Zweck innerhalb des Projekts	113
5.4.3	Active Directory	114
5.5	Entwicklungsumgebung mit Visual Studio 2022	115
5.5.1	Einleitung	115
5.5.2	Öffentlicher Zugriff auf lokalen Entwicklungsserver	115
5.5.3	Funktionsweise eines lokalen Entwicklungsservers	115
5.5.4	Integration mit Visual Studio 2022	115
5.5.5	Einsatz von Dev Tunnels für die Teamarbeit	116
5.5.6	Dev Tunnels - Erstellen eines Dev Tunnels	117
5.5.7	Notwendigkeit eines Microsoft-Accounts	117
5.5.8	Name und Tunnel-Typ	117
5.5.9	Zugangsberechtigung	118
5.6	Angular-Konfiguration für Externen Zugriff	118
5.7	Netzwerkkonfiguration	119
5.7.1	Integration in die Domäne <code>bestellsystem.local</code>	120

Abbildungsverzeichnis

VII

Literaturverzeichnis

X

1 Einleitung

1.1 Einblick

Im Sommer 2023, von Juli bis Anfang August, absolvierten wir ein Praktikum bei der Oberösterreichischen Gesundheitsholding GmbH (OÖG). Dieses Praktikum war Teil unserer fünfjährigen Ausbildung an der HTL Leonding, in der wir für acht Wochen ein Praktikum absolvieren müssen. Wir verbrachten vier Wochen in der IT-Abteilung der OÖG und erhielten dort ein Projekt, das wir als Diplomarbeit umsetzen.

Während unseres Aufenthalts bei der OÖG arbeiteten wir eng mit dem Team der Medizininformatik und Informationstechnologie (MIT) zusammen. Unser Ziel war es, ein einfaches und effizientes System zu erstellen, das es den Mitarbeitern der OÖG ermöglicht, verschiedene Dinge für ihre Arbeit zu bestellen. Das umfasste alles, von Büromaterialien bis hin zu medizinischem Equipment. Wir haben viel Zeit damit verbracht, die Bedürfnisse der Mitarbeiter zu verstehen und dann das System entsprechend zu entwerfen und umzusetzen.

Unsere Arbeit bestand aus vielen Schritten, von der Planung bis zur Umsetzung. Wir haben darauf geachtet, dass das System sicher und benutzerfreundlich ist. Nachdem wir alles fertiggestellt hatten, haben wir unsere Ergebnisse präsentiert und positives Feedback erhalten. Das positive Feedback hat uns ermutigt und bestätigt, dass unsere Bemühungen Früchte getragen haben. Es hat uns gezeigt, dass wir auf dem richtigen Weg sind und uns ermutigt, unsere Arbeit mit derselben Sorgfalt fortzusetzen.

Durch das Praktikum und die Diplomarbeit haben wir nicht nur unsere Ausbildung an der HTL erfolgreich fortgeführt, sondern auch wertvolle Erfahrungen gesammelt. Diese werden uns auf unserem beruflichen Weg weiterhelfen. Unsere Zeit bei der OÖG war herausfordernd, aber äußerst lohnend. Sie hat uns gezeigt, wie wichtig es ist, IT-Lösungen effektiv einzusetzen, um Arbeitsabläufe in Unternehmen zu verbessern.

1.2 Auftraggeber

Die Oberösterreichische Gesundheitsholding GmbH (OÖG) betreibt neben dem Kepler Universitätsklinikum in Linz ,sechs Regionalkliniken an acht Standorten. Die Direktion Medizininformatik und Informationstechnologie (MIT) wickelt den IT-Betrieb aller Einrichtungen der OÖ Gesundheitsholding ab. Sie betreiben ein eigenes Rechenzentrum und setzen innovativ und zukunftsorientiert die IT-Projekte des Unternehmens um[3].



1.3 Ausgangssituation

Die Oberösterreichische Gesundheitsholding GmbH steht vor vielfältigen Herausforderungen im Zusammenhang mit der Erstellung und dem Versand von Bestellformularen als Excel-Dateien per E-Mail. Diese Herausforderungen haben direkte Auswirkungen auf die Effizienz der Arbeitsweise innerhalb des Unternehmens. Aktuell erfordert der Prozess zur Verfolgung des Dokumentenstatus und zur Einholung von Genehmigungen viel Zeit und Ressourcen. Dies führt zu einer ineffizienten Arbeitsweise, da die ständige Erstellung neuer Excel-Dateien die Komplexität erhöht und eine einheitliche Verwaltung der Bestellvorgänge erschwert.

Die bestehende Vorgehensweise beinhaltet das Hin- und Herschicken von Excel-Dateien für Bestellanfragen, die jeweils an den nächsten Vorgesetzten weitergeleitet werden müssen. Wenn eine Anfrage abgelehnt wird, muss sie per E-Mail zurückgeschickt werden, wobei die Gründe für die Ablehnung angegeben werden müssen. Dieser Prozess ist manuell und zeitaufwändig, da es keine klare Nachverfolgung gibt, wer sich in welchem Genehmigungsstadium befindet, und es führt dazu, dass ständig neue Excel-Dateien erstellt werden müssen.

Das aktuelle Bestellformular ist als Excel-Dokument umgesetzt und beinhaltet Pflichtfelder sowie optionale Felder für die Eingabe von Bestelldaten und Informationen. Es muss vom Besteller ausgefüllt werden, wobei das Fehlen von Pflichtangaben die weitere Bearbeitung verhindert. Darüber hinaus müssen Bestellungen nach der Erstellung von verschiedenen Hierarchieebenen genehmigt werden, was den Prozess zusätzlich verkompliziert.

Es besteht daher ein klarer Bedarf nach einem optimierten und zentralisierten System, das den gesamten Bestellprozess vereinfacht, eine bessere Kontrolle ermöglicht und die Arbeitsabläufe effizienter gestaltet. Durch die Implementierung eines solchen Systems können die bestehenden Herausforderungen bewältigt und die Produktivität des Unternehmens gesteigert werden.

Die Hierarchieebene in dieser Unternehmen sieht gefogt aus:



Abbildung 1: Hierarchieebene

In Oberösterreichische Gesundheitsholding GmbH (OÖG) gibt es eine klare Struktur, die in drei Ebenen von Vorgesetzten unterteilt ist. An der Spitze steht die Finanzabteilung, die die oberste Autorität in finanziellen Angelegenheiten innehat. Diese Abteilung überwacht und steuert alle Aspekte der Finanzen des Unternehmens.

Wenn es um den Bestellprozess geht, folgen wir einem genau festgelegten Ablauf. Das bedeutet, dass jeder, unabhängig von seiner Position, die gleichen Schritte befolgen muss, um eine Bestellung aufzugeben. Selbst wenn ein Vorgesetzter persönlich etwas bestellen möchte, muss er sich an diese Richtlinien halten.

Durch diese klare Struktur und den einheitlichen Bestellprozess stellen wir sicher, dass alle Bestellungen ordnungsgemäß abgewickelt werden. Dies hilft, Missverständnisse zu vermeiden und sicherzustellen, dass die finanziellen Interessen des Unternehmens geschützt sind. Zudem fördert es die Effizienz und Transparenz im Unternehmen.

Die Excel Datei sieht folgendermaßen aus:

Datum	Besteller	Abteilung	BL

Lieferant	KNr.	#NV
BBG-Shop-Bestellung: nein		
Bekanntmachung nach BVergG 2018 nötig nein		
GZ-Nummer wenn BBG-Shop-Bestellung: GZ		

Betreff

Anmerkungen

Lieferbedingungen	Lieferadresse

ID	Anzahl	Komponentenbezeichnung	Einzelpreis netto €	Gesamt netto €	Betr.KH	Budgetposition	Zuordnung	Kostenstelle	Auftrag/KST oder Invest
1				€ 0,00					0
2				€ 0,00					0
3				€ 0,00					0
4				€ 0,00					0
5				€ 0,00					0
6				€ 0,00					0
7				€ 0,00					0
8				€ 0,00					0
9				€ 0,00					0
				€ 0,00					

Bestellanforderung KUK




Abbildung 2: Excel Datei

In der Vorlage für die Bestellanforderung gibt es Pflichtfelder (orange hinterlegt) und optionale Felder (weiß hinterlegt). Wenn eines der Pflichtfelder vom Besteller nicht ausgefüllt wird, kann die Bestellanforderung nicht weiterbearbeitet werden.

1.4 Ziele

Das Hauptziel dieses Projekts ist es, ein effizientes und benutzerfreundliches System zu entwickeln, das den gesamten Bestellprozess innerhalb der Oberösterreichischen Gesundheitsholding GmbH optimiert. Hierbei sollen speziell folgende Ziele erreicht werden:

Automatisierung des Bestellprozesses: Die Entwicklung einer Website oder Anwendung, die es den Mitarbeitern ermöglicht, Bestellungen schnell und einfach aufzugeben, ist von zentraler Bedeutung. Durch die Automatisierung dieses Prozesses wird die manuelle Bearbeitung von Bestellanforderungen eliminiert, was zu einer Zeitersparnis führt.

Transparenter Genehmigungsprozess: Ein weiteres Ziel ist es, den Genehmigungsprozess transparent und nachvollziehbar zu gestalten. Sobald ein Formular vollständig ausgefüllt ist, soll es automatisch an die nächste Instanz zur Genehmigung weitergeleitet werden. Dabei wird der Genehmigungsprozess so gestaltet, dass er sich an die Struktur der Firma anpasst. Dies bedeutet, dass das System automatisch die richtigen Personen für die Genehmigung auswählt, basierend auf der Organisationsstruktur des Unternehmens.

Effiziente Formularerstellung: Die Anwendung soll maßgeschneiderte Formulare generieren, die von den Mitarbeitern einfach ausgefüllt werden können. Diese Formulare sollen alle erforderlichen Informationen enthalten und den Mitarbeitern eine klare Struktur für die Auftragserteilung bieten.

Steigerung der Effizienz und Produktivität: Durch die Implementierung dieses Systems wird die Effizienz und Produktivität im Bestellprozess erheblich gesteigert. Mitarbeiter können Bestellungen schneller aufgeben, Genehmigungen werden zügig erteilt, und die gesamte Abwicklung verläuft reibungsloser und effizienter.

Datenexport: Ein weiteres wichtiges Ziel des Projekts ist es, einen effektiven Datenexport zu implementieren, um die Bestellungen auch an Personen weiterleiten zu können, die keinen direkten Zugriff auf das Programm haben. Hierbei sollen die Daten aus dem Bestellformular in Form einer PDF-Datei exportiert werden, wobei auch die Anhänge der Dateianhänge in die PDF integriert werden.

Verbesserung der Nachverfolgbarkeit: Durch die Digitalisierung des Bestellprozesses wird die Nachverfolgbarkeit von Bestellungen verbessert. Mitarbeiter können den Status ihrer Bestellungen jederzeit einsehen und wissen genau, in welchem Genehmigungsstadium sich ihre Anfragen befinden.

Insgesamt trägt die Implementierung eines effektiven Datenexports dazu bei, die Reichweite und Zugänglichkeit der Bestellinformationen zu erhöhen und den Informationsaustausch sowohl innerhalb als auch außerhalb des Unternehmens zu erleichtern.

Dieses Projekt zielt darauf ab, den gesamten Bestellprozess zu optimieren und die Arbeitsabläufe innerhalb der Oberösterreichischen Gesundheitsholding GmbH effizienter und transparenter zu gestalten. Durch die Implementierung eines automatisierten Systems werden Zeit- und Ressourceneinsparungen erzielt und die Produktivität des Unternehmens gesteigert.

1.5 Systemarchitektur

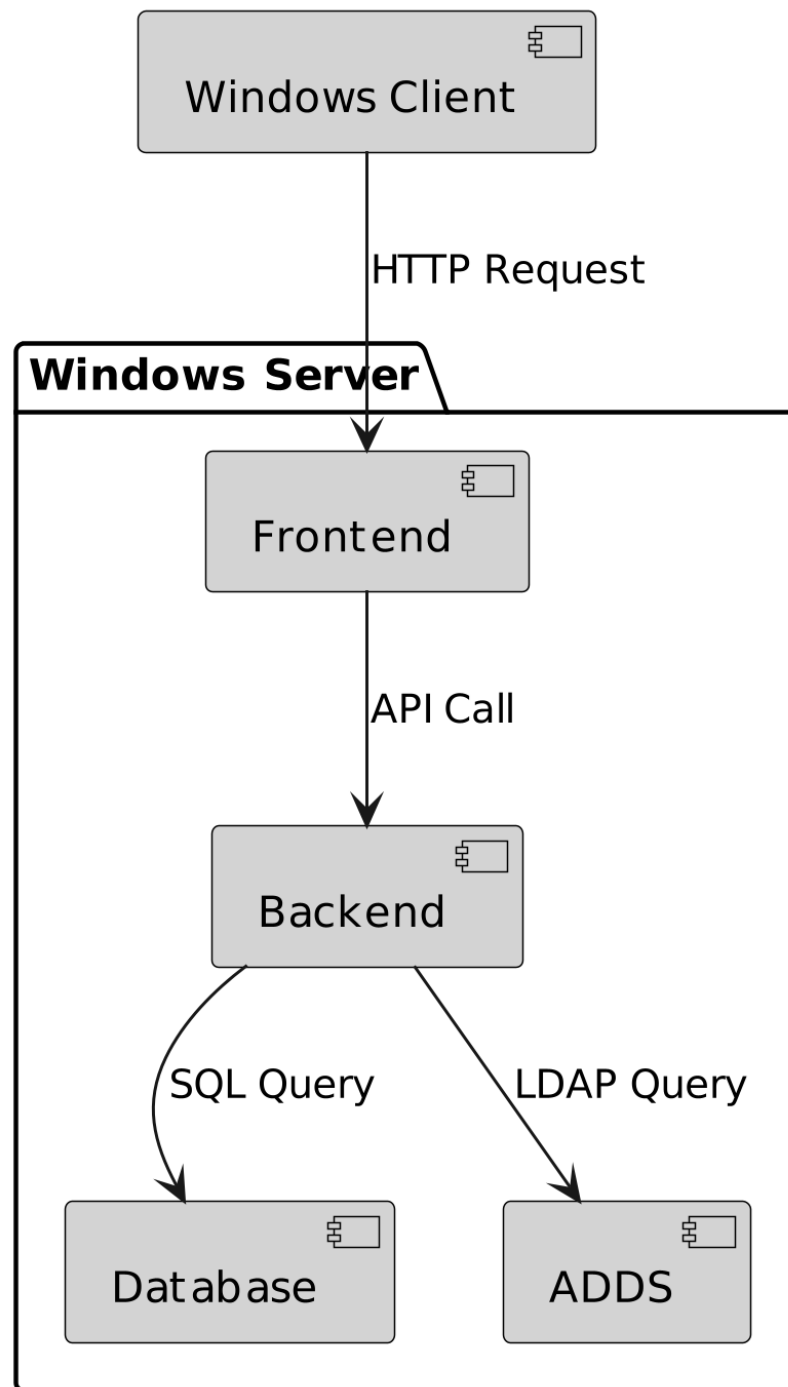


Abbildung 3: Systemarchitektur

Unsere Systemarchitektur ist darauf ausgelegt, die Funktionalität unserer Anwendung effizient zu unterstützen und gleichzeitig eine reibungslose Kommunikation zwischen den verschiedenen Komponenten zu gewährleisten. Der Windows Server bildet das Herzstück unseres Systems und beherbergt eine Vielzahl von Komponenten, die nahtlos zusammenarbeiten, um die Anforderungen unserer Nutzer zu erfüllen.

Frontend: Das Frontend fungiert als Benutzerschnittstelle unserer Anwendung, über die Nutzer mit dem System interagieren. Es ist verantwortlich für die Darstellung von Informationen und die Entgegennahme von Eingaben der Benutzer. Durch den Empfang von HTTP-Anfragen vom Client ermöglicht das Frontend eine einfache und intuitive Interaktion mit der Anwendung. Es kommuniziert mit dem Backend über API-Aufrufe, um Daten zu verarbeiten und dem Benutzer entsprechende Rückmeldungen zu geben.

Backend: Im Backend findet die eigentliche Verarbeitung von Anfragen statt. Hier werden komplexe Operationen ausgeführt, wie das Ausführen von Datenbankabfragen in der Datenbankkomponente und das Interagieren mit den Active Directory Domain Services (ADDS) für Authentifizierung und Autorisierung. Das Backend ist das Rückgrat unserer Anwendung und gewährleistet eine robuste und zuverlässige Verarbeitung von Nutzeranfragen.

Database: Die Datenbankkomponente dient als Speicher für alle persistenten Daten, die von unserer Anwendung verwendet werden. Hier werden Informationen gespeichert, abgerufen und aktualisiert, um die Funktionalität der Anwendung sicherzustellen. Das Backend greift auf die Datenbank zu, um Informationen zu verarbeiten und dem Frontend entsprechende Daten bereitzustellen.

Active Directory Domain Services (ADDS): ADDS bietet wichtige Dienste für die Verwaltung und Organisation von Benutzern, Computern und Ressourcen im Netzwerk. Durch die Integration mit unserem System ermöglicht es eine sichere und effiziente Authentifizierung und Autorisierung von Benutzern, um den Zugriff auf Ressourcen zu steuern und die Sicherheit der Anwendung zu gewährleisten.

Client: Der Client ist die Schnittstelle zwischen dem Benutzer und unserem System. Er kann eine Benutzeranwendung oder ein Gerät sein, das HTTP-Anfragen an das Frontend des Servers sendet, um verschiedene Aktionen auszuführen und Informationen abzurufen. Der Client spielt eine entscheidende Rolle bei der Bereitstellung einer nahtlosen Benutzererfahrung und ermöglicht es den Nutzern, effizient mit der Anwendung zu interagieren.

Diese Architektur ist darauf ausgerichtet, eine effiziente Kommunikation zwischen den verschiedenen Komponenten zu ermöglichen und gleichzeitig eine stabile und skalierbare Grundlage für die Entwicklung und Bereitstellung unserer Anwendung zu bieten.

2 Projektmanagement

2.1 Scrum

Für die reibungslose Abwicklung des Projekts wurde nach Scrum gearbeitet.

Scrum ist eine Methode des agilen Projektmanagements, die ihren Ursprung im Rugby hat. Der Begriff Scrum bezieht sich auf eine Spielsituation im Rugby, bei der Spieler sich um den Ball drängen.

Im Kontext des agilen Projektmanagements bedeutet Scrum, dass alle Teammitglieder regelmäßig zusammenkommen, um über laufende Aufgaben und den Fortschritt im Projekt zu sprechen.

Diese Methode beinhaltet spezifische Regeln und Praktiken, die darauf abzielen, die starren und unflexiblen Aspekte traditioneller Projektplanung zu ersetzen. Das Ziel ist es, Projekte agiler zu gestalten und somit effektiver auf Veränderungen reagieren zu können.[18].

2.1.1 Warum ist man mit Scrum agil?

Scrum ermöglicht Agilität, indem es klare und einfache Regeln hat. Diese Regeln sollen das Projektteam dabei unterstützen, das gemeinsame Projektziel zu erreichen. Die Hauptregel ermöglicht es dem Team, sich im agilen Projektmanagement selbst zu organisieren. Nachdem die Projektziele mit dem Auftraggeber vereinbart wurden, hat das Team die Freiheit, seine Arbeitsweise anzupassen, ohne einen festen Projektplan zu haben. Zusätzlich betont Scrum auch die Bedeutung von schnellem Feedback, kontinuierlicher Verbesserung und der Nutzung von Sprints, um iterativ und dynamisch auf das Ziel zuzusteuern.[18].

2.1.2 Rollen

Es gibt bei Scrum drei Rollen:

- Product Owner
- der Scrum-Master
- das Mitglied im Projektteam

2.1.3 Product Owner

Der Product Owner repräsentiert die Anwender oder Stakeholder eines Produkts. Das können Nutzer, Produktmanager, Marketing, Vertrieb und Kundendienst sein. Ihr Ziel ist es, die Bedürfnisse der Kunden zu verstehen und dem Projektteam klar zu kommunizieren.[18].

2.1.4 Scrum-Master

Der Scrum-Master ist für die Leitung des Scrum-Prozesses verantwortlich. Er fungiert als Moderator und unterstützt das Projektteam, indem er Hindernisse beseitigt und eine effektive Zusammenarbeit fördert. Zudem sorgt er für die Bereitstellung der benötigten Ressourcen und ist die Kontaktperson für externe Parteien. Der Scrum-Master hilft dem Team bei methodischen Fragen und gewährleistet die Einhaltung der Regeln des agilen Projektmanagements.[18].

2.1.5 Projektteam

Ein Scrum-Team besteht aus fünf bis zehn Mitgliedern ohne Hierarchie. Jedes Teammitglied hat gleiche Rechte und Pflichten, aber unterschiedliche Kompetenzen. Alle relevanten Fachbereiche sollten im Team vertreten sein und die Mitglieder sollten eigenverantwortlich und motiviert an Projekten arbeiten können.[18].

2.1.6 Methoden und Regeln beim Projekt-Sprint

Das Projekt läuft in mehreren Phasen und verwendet verschiedene Methoden und Regeln. Das tägliche Meeting, das als Daily Scrum bekannt ist, ist ein wesentlicher Bestandteil jedes Sprints.

Ein Sprint ist eine fest definierte Zeitspanne innerhalb eines Scrum-Projekts, in der ein bestimmter Arbeitsumfang erledigt wird. In der Regel dauert ein Sprint zwischen einer und vier Wochen.

Um die Aufgaben und Aktivitäten im Sprint effizient zu überprüfen und durchzuführen, werden verschiedene Methoden und Tools angewendet. Darüber hinaus werden sie verwendet, um die Fortschritte nach einem Sprint zu bewerten. Zu den bekanntesten gehören das Product Backlog, Daily Scrum, Sprint Review, Sprint Planning und das Burndown-Chart.[18].

2.1.7 Product Backlog

Im Product Backlog werden Prioritäten von dem Product Owner vergeben. User Stories sind ein gängiges Format, um die Anforderungen im Product Backlog zu spezifizieren. User Stories beschreiben aus der Perspektive des Nutzers, welche Funktionalität benötigt wird und welchen Wert sie für den Nutzer bringt. Diese Stories werden in der Regel vom Product Owner priorisiert, wobei Elemente mit höherer Priorität als besonders wichtig erachtet werden, um die Zufriedenheit der Anwender zu maximieren und den größten Nutzen zu liefern. Andere Anforderungen können als weniger wichtig betrachtet und entsprechend behandelt werden, sei es durch das Aussortieren, Zusammenlegen, Verschieben oder Überarbeiten zu einem späteren Zeitpunkt.

Es ist wichtig zu beachten, dass der Product Backlog ein dynamisches Dokument ist, das sich im Laufe des Projekts weiterentwickelt. Der Product Owner überprüft und aktualisiert den Backlog regelmäßig, um sicherzustellen, dass die höchstpriorisierten Elemente für das Team bereit sind, um sie im nächsten Sprint zu bearbeiten. Dabei wird fortlaufend darauf geachtet, dass die Kundenbedürfnisse und Geschäftsziele be-

rücksichtigt werden.[18]

2.1.8 Sprint Planning

Die Planung eines Sprints ist ein entscheidender Schritt im Scrum-Prozess. Dabei werden die ausgewählten User Stories aus dem Product Backlog genommen und für das kommende Sprint Backlog in konkrete Aufgaben, auch bekannt als Tasks, verfeinert. Hierbei wird jedes Task so definiert, dass es einen klaren Beitrag zur Erfüllung der entsprechenden User Story leistet.

Zusätzlich zur Aufgabenplanung wird während der Sprint-Planung auch ein Sprintziel festgelegt. Dieses Ziel definiert den übergeordneten Fokus oder die Absicht des Sprints. Es dient als Leitstern für das Entwicklungsteam, um sicherzustellen, dass die Arbeit während des Sprints darauf ausgerichtet ist, dieses Ziel zu erreichen.[18]

In der Sprint-Planung werden also im Wesentlichen zwei Fragen geklärt:

- Was genau wird im nächsten Sprint entwickelt, erstellt oder durchgeführt?
- Wie sollen die relevanten Aufgaben und Arbeiten im Sprint konkret ausgeführt werden, um das Sprintziel zu erreichen?

2.1.9 Burndown Chart

Ein Burn-Down-Diagramm zeigt, wie viel Arbeit noch übrig ist im Vergleich zur Zeit. Es wird oft in der agilen Softwareentwicklung wie Scrum benutzt. Aber man kann es auch für andere Projekte verwenden, die über die Zeit fortschreiten.[19].

2.1.10 Sprint Review

Ein Sprint-Review ist eine gemeinsame Besprechung, die typischerweise am Ende jedes Sprints stattfindet. Dabei überprüft das Team die während des Sprints oder der Iteration erledigten Aufgaben. Ziel des Sprint-Reviews ist es, sicherzustellen, dass die Stakeholder informiert sind und ihnen die Möglichkeit zu geben, Feedback zu geben. Nach jedem Sprint-Review verwendet der Product Owner oder Teamleiter dieses Feedback, um die erforderlichen Arbeitsaufgaben im Product Backlog zu aktualisieren.[20].

2.1.11 Daily Scrum

In den täglichen 15-minütigen treffen, teilt jeder Teilnehmer nacheinander mit, was seit dem letzten Meeting erreicht wurde und welche Aufgaben bis zum nächsten Treffen erledigt werden sollen. Es wird auch mitgeteilt ,ob es irgendwelche Hindernisse bei der Arbeit gibt. Innerhalb dieses Prozesses bleibt das Sprint-Ziel im Fokus und es wird überprüft, ob es weiterhin erreichbar ist. Der Scrum-Master greift ein, um Hindernisse anzugehen und bei ihrer Beseitigung zu unterstützen. Darüber hinaus werden die Fortschritte des Projekts im Sprint-Burndown-Chart visualisiert, um den Verlauf des Projekts zu verfolgen.[18].

2.2 Versionierungsverwaltung mit GitHub

2.2.1 Verwendung

In Bezug auf unsere Diplomarbeit haben wir GitHub als Versionierungsverwaltung verwendet, um den Quellcode unserer Softwareprojekte zu speichern, Änderungen nachzuverfolgen und die Zusammenarbeit im Team zu erleichtern. Durch GitHub konnten wir effizienter arbeiten und sicherstellen, dass alle Teammitglieder stets Zugriff auf den aktuellsten Stand des Codes hatten.

2.2.2 Github

GitHub ist eine webbasierte Plattform, die Programmierern eine zentrale Umgebung bietet, um ihre Softwareprojekte zu verfolgen und Versionen zu verwalten. Sie ist auf Git basierend und ermöglicht das Hosting, Teilen, Verwalten und die kollaborative Bearbeitung von Code. Um die Frage „Was ist GitHub?“ zu beantworten, müssen zwei sehr verwandte Konzepte erklärt werden: Git und Versionsverwaltung. [21].



2.2.3 Versionskontrolle

Die Versionsverwaltung ist entscheidend für die Überwachung und Verwaltung von Codeänderungen in Softwareprojekten. Es ist nicht möglich, den offiziellen Quellcode in großen kollaborativen Projekten direkt zu bearbeiten. Die Versionsverwaltung ermöglicht es Programmierern, sicher zu arbeiten, indem sie Verzweigungen und Zusammenführungen verwendet. Ein Entwickler kann beim Verzweigen einen Teil des Quellcodes (Repository) kopieren und Änderungen vornehmen, ohne den Rest des Projekts zu beeinträchtigen. Anschließend können diese Änderungen in die Hauptquellcode integriert werden. Es ist möglich, alle Änderungen zu überprüfen und bei Bedarf zu korrigieren. [21].

2.2.4 Git

Git basiert auf einer Reihe von Grundprinzipien. Das Repository ist das Hauptelement, das den gesamten Codeverlauf und alle damit verbundenen Metadaten speichert. Im Rahmen dieses Repositories können Entwickler Änderungen in Form von Commits vornehmen, um genau zu dokumentieren, welche Änderungen vorgenommen wurden und von wem. Darüber hinaus ermöglicht Git das Erstellen und Wechseln von Branche zu Branche, um verschiedene Codeversionen zu verwalten und gleichzeitig an verschiedenen Features zu arbeiten.

Git ermöglicht parallele Entwicklung durch die Verwendung von Branches. Jeder Branch repräsentiert eine unabhängige Linie der Entwicklung, was es Entwicklern erlaubt, an neuen Features, Bugfixes oder Experimenten zu arbeiten, ohne den Hauptcode (üblicherweise im "master"- oder "main"-Branch) zu beeinflussen. Durch die Nutzung von Branches können Entwickler isoliert arbeiten und Änderungen unabhängig voneinander testen, bevor sie in den Hauptcode integriert werden.

Die Fähigkeit, Änderungen aus verschiedenen Zweigen zusammenzuführen, ist ein wichtiger Teil von Git. Dieser Prozess, der auch als Merge bekannt ist, ermöglicht es Entwicklern, die Änderungen in ihrem Code mit den Änderungen anderer Entwickler zu vereinen. Dies fördert die Zusammenarbeit und verhindert Konflikte bei der Integration von Änderungen im Code.[21] [26]

2.2.5 Github Desktop

GitHub Desktop ist eine kostenlose und Open-Source-App, die es dir ermöglicht, deine Git-Repositories von deinem Desktop aus zu verwalten. Mit GitHub Desktop können Sie mit Co-Autoren zusammenarbeiten, Pull-Anfragen überprüfen, Syntax hervorheben und mit deinem Editor und Shell integrieren.[22].

In diesem Projekt haben sich alle Teammitglieder für die Verwendung von GitHub Desktop entschieden, da es kompakter, übersichtlicher und leichter zu bedienen ist.

3 Technologien

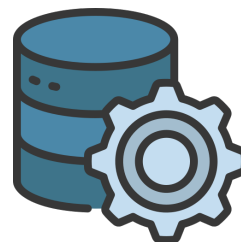
Im Folgenden werden zunächst die in unserer Diplomarbeit verwendeten Technologien erläutert. Anschließend gehen wir darauf ein, für welchen spezifischen Zweck diese Technologien in unserem Projekt eingesetzt wurden.

3.1 Windows Server 2022

3.1.1 Server Manager

Erläuterung

Der Server Manager ist eine Verwaltungskonsolle in Windows Server, die IT-Professionals dabei hilft, lokale und entfernte Windows-basierte Server von ihren Desktops aus zu versorgen und zu verwalten. Dafür ist weder physischer Zugriff auf die Server noch die Aktivierung von Remote-Desktop-Verbindungen zu jedem Server erforderlich. Obwohl der Server Manager in Windows Server 2008 R2 und Windows Server 2008 verfügbar ist, wurde er in Windows Server 2012 aktualisiert, um die entfernte, mehrfache Serververwaltung zu unterstützen und die Anzahl der Server zu erhöhen, die ein Administrator verwalten kann. [5]



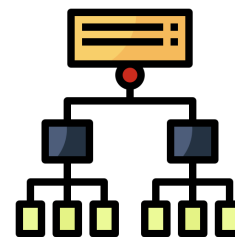
Verwendung

In unserem Projekt haben wir erfolgreich Rollen wie Active Directory Domain Services, DHCP, DNS, File and Storage Services und IIS implementiert. Diese ermöglichen eine effiziente Verwaltung von Benutzern, automatische IP-Zuweisung, DNS-Resolution, Dateispeicherung und Webserverfunktionalitäten über den zentralen Server Manager.

3.1.2 AD DS

Erläuterung

Ein Verzeichnisdienst, vergleichbar mit einer geordneten Struktur, dient dazu, Informationen über Dinge im Netzwerk zu speichern. Active Directory Domain Services (AD DS) ist ein Beispiel für einen Verzeichnisdienst, der sicherstellt, dass diese Informationen nicht nur gut gespeichert, sondern auch für berechtigte Personen im Netzwerk zugänglich sind.



AD DS spielt eine entscheidende Rolle bei der Speicherung von Benutzerinformationen, darunter Namen, Passwörter, Telefonnummern und weitere Attribute. Die strukturierte Organisation dieser Daten ermöglicht eine effiziente Verwaltung und gewährleistet, dass autorisierte Benutzer im Netzwerk darauf zugreifen können.

Durch die zentrale Verwaltung von Benutzerinformationen ermöglicht AD DS eine effektive Steuerung von Identitäten, Zugriffsberechtigungen und Gruppenmitgliedschaften. Dies erleichtert die Pflege und Aktualisierung von Benutzerkonten im gesamten Netzwerk. [6].

Verwendung

In unserem Projekt hat die Implementierung von Active Directory Domain Services (AD DS) eine entscheidende Rolle gespielt. Wir haben AD DS genutzt, um eine strukturierte und effiziente Verwaltung von Benutzern und Ressourcen zu ermöglichen. Hierbei haben wir speziell vier Gruppen erstellt, nämlich Benutzer, Vorgesetzte 1, 2, 3 und die Finanzabteilung.

Für jede dieser Gruppen haben wir gezielt Testprofile erstellt und diesen Profilen relevante Informationen zugewiesen. Dabei handelt es sich um grundlegende Angaben wie Vor- und Nachnamen, Abteilungszugehörigkeit, Adressdaten und weitere relevante Informationen. Diese individuellen Benutzerprofile ermöglichen nicht nur eine geordnete Organisation unserer Netzwerkbenutzer, sondern stellen auch sicher, dass die jeweiligen Benutzer die erforderlichen Zugriffsrechte und Informationen erhalten, um ihre Aufgaben effektiv zu erfüllen.

3.1.3 DNS

Erläuterung

In Windows Server 2022 ist DNS (Domain Name System) eine entscheidende Funktion, die sowohl mit dem Server-Manager als auch über PowerShell installiert werden kann. Bei der Einrichtung von Active Directory wird DNS automatisch als essentieller Service für die Struktur und Domäne



integriert. Diese nahtlose Integration gewährleistet eine effiziente Zusammenarbeit zwischen DNS und Active Directory Domain Services (AD DS).

AD DS nutzt DNS, um Computercontroller zu finden, und der DNS-Clientdienst spielt eine Schlüsselrolle bei der Auflösung von Computernamen zu IP-Adressen. Dies ist von besonderer Bedeutung, wenn sich Benutzer bei Active Directory anmelden und eine zuverlässige Namensauflösung erforderlich ist.

DNS in Windows Server 2022 arbeitet auf Anwendungsebene des TCP/IP-Protokolls, was eine robuste und effiziente Namensauflösung im Netzwerk sicherstellt. Die zentrale Rolle von DNS in der Infrastruktur macht es zu einem unverzichtbaren Bestandteil für die reibungslose Funktionsweise von Active Directory und anderen Netzwerkdiensten. [7].

Verwendung

In unserem Projekt, das auf Windows Server 2022 basiert, haben wir aktiv DNS genutzt, um die Namensauflösung zu ermöglichen. Diese Funktion ist besonders entscheidend, da wir die IP-Adressen unserer Netzwerkgeräte nicht auswendig kennen. Die Verwendung von DNS ermöglicht es uns, die Namen unserer Netzwerkgeräte zu verwenden und dadurch einfach auf sie zuzugreifen. Diese Herangehensweise erleichtert die Verwaltung und den Zugriff auf Ressourcen erheblich, indem sie uns von der Notwendigkeit befreit, IP-Adressen zu merken oder nach ihnen zu suchen.

3.1.4 IIS

Erläuterung

Der Internetinformationsdienst (IIS) in Version 7 und höher zeichnet sich durch eine vielschichtige Anforderungsverarbeitungsarchitektur aus, die für Flexibilität und Anpassbarkeit sorgt. Der Windows-Prozessaktivierungsdienst (WAS) integriert die Fähigkeit, Protokolle über HTTP und HTTPS hinaus zu verwenden, was die Kommunikation und Integration von verschiedenen Diensten erleichtert.



Diese Architektur ermöglicht eine hohe Anpassbarkeit durch die Webserver-Engine. Administratoren können Module nach Bedarf hinzufügen oder entfernen, um spezifische Funktionen zu implementieren. Dieser Ansatz erlaubt eine maßgeschneiderte Konfiguration des Servers entsprechend den individuellen Anforderungen einer Anwendung oder Website.

Die integrierten Pipelines stellen sicher, dass Anfragen sowohl für IIS als auch ASP.NET nahtlos verarbeitet werden. Dies schafft eine effiziente und flexible Webserverumgebung, die den reibungslosen Ablauf von Anfragen gewährleistet.

Zusätzlich zu den genannten Vorteilen bietet IIS 7 und höher umfassende Unterstützung für die sicheren Protokolle HTTP und HTTPS. Dies ermöglicht eine sichere Übertragung von Daten zwischen dem Webserver und den Clients, was insbesondere in Umgebungen mit sensiblen Informationen von großer Bedeutung ist. [9].

Verwendung

In unserem Projekt wurde IIS (Internet Information Services) eingesetzt, um die entwickelte Website auf anderen Clients zugänglich zu machen. Durch die Verwendung von IIS als Webserver konnten wir sicherstellen, dass die Website effizient gehostet und Nutzern auf unterschiedlichen Geräten und über diverse Webbrowser hinweg zur Verfügung gestellt wird. IIS hat es uns ermöglicht, die Anfragen von Client-Computern zu verarbeiten und ihnen die entsprechenden Inhalte unserer Website zu liefern. Dies war ein entscheidender Schritt, um die Reichweite und Verfügbarkeit unserer Webanwendung zu maximieren und den Zugriff für die Zielgruppe zu erleichtern.

3.1.5 Inbound Rules

Erläuterung

Inbound Rules sind spezifische Konfigurationen in einem Computernetzwerk, die festlegen, welche Art von Datenverkehr oder Verbindungen von externen Quellen akzeptiert werden sollen. Diese Regeln agieren wie Richtlinien für den Netzwerkverkehr, indem sie genau definieren, welche eingehenden Datenverbindungen erlaubt oder blockiert werden sollen. [10].



Verwendung

In Bezug auf die Verwendung von Inbound Rules haben wir in der Firewall eine Inbound Rule speziell für das Frontend erstellt. Diese Regel dient dazu, den eingehenden Datenverkehr für das Frontend-System zu steuern und zu regeln.

3.1.6 Outbound Rules

Erläuterung

Outbound Rules sind spezifische Konfigurationen in einer Firewall, die genau festlegen, welche Arten von ausgehendem Datenverkehr oder Verbindungen von einem Computer oder Netzwerk akzeptiert werden sollen. Im Gegensatz zu Inbound Rules, die den eingehenden Verkehr steuern, setzen Outbound Rules den Fokus darauf, welche Informationen oder Verbindungen von einem System nach außen zugelassen werden dürfen.



Die Bedeutung dieser Regelungen liegt in der Netzwerksicherheit, da sie ermöglichen, den ausgehenden Datenverkehr zu überwachen und zu filtern. Dies gewährleistet Datenschutz und schützt vor potenziellen Gefahren wie Datenlecks oder unautorisiertem Zugriff auf das Netzwerk. Durch die gezielte Steuerung von ausgehenden Verbindungen bieten Outbound Rules eine zusätzliche Sicherheitsebene, um das Verhalten von Anwendungen zu kontrollieren und Malware-Verbindungen zu verhindern. Die effektive Implementierung dieser Regeln ist entscheidend, um ein sicheres und geschütztes Netzwerkumfeld zu gewährleisten. [11].

Im Gegensatz zu Inbound Rules, die den eingehenden Verkehr steuern, setzen Outbound

Rules den Fokus darauf, welche Informationen oder Verbindungen von einem System nach außen zugelassen werden dürfen. [11].

Verwendung

In unserer Anwendung von Outbound Rules haben wir in der Firewall eine spezielle Outbound Rule für das Frontend erstellt. Diese Regel ist darauf ausgerichtet, den ausgehenden Datenverkehr des Frontend-Systems zu steuern und zu regulieren, indem genau festgelegt wird, welche Arten von ausgehenden Verbindungen oder Daten erlaubt sind. Dies ermöglicht eine präzise Kontrolle darüber, welche Informationen das Frontend-System nach außen senden kann, um die Sicherheit zu gewährleisten und nur autorisierten und sicheren Datenverkehr zuzulassen.

3.2 Frontend

3.2.1 JavaScript

Erläuterung

JavaScript ist eine vielseitige Programmiersprache, die vorwiegend für die Gestaltung von dynamischen und interaktiven Inhalten auf Webseiten verwendet wird. Sie ermöglicht es, Webanwendungen zu erstellen, die auf Benutzerinteraktionen reagieren und Inhalte dynamisch aktualisieren können. JavaScript wird clientseitig im Webbrowser ausgeführt und kann verwendet werden, um das Verhalten von HTML-Dokumenten zu steuern, auf Benutzeraktionen zu reagieren und Daten mit einem Webserver auszutauschen. Dank seiner breiten Unterstützung in modernen Webbrowsern ist JavaScript eine der wichtigsten Technologien für die Entwicklung von Webanwendungen. [12].



Verwendung

In unserem Fall haben wir JavaScript für die Anpassung des Website-Themas eingesetzt. Benutzer können aus verschiedenen Themen (Light, Dark oder System) wählen, und JavaScript wird verwendet, um die Benutzerauswahl zu verarbeiten und die entsprechenden Themen dynamisch auf der Website anzuzeigen. Darüber hinaus wurde JavaScript in Verbindung mit jQuery genutzt, um die Interaktivität der Website zu verbessern und spezifische Aktionen oder Effekte basierend auf Benutzerinteraktionen zu ermöglichen.

3.2.2 Bootstrap

Erläuterung

Bootstrap ist ein Open-Source-Framework für die Entwicklung von responsiven und ansprechenden Websites und Webanwendungen [13].



Verwendung

In unserem Fall haben wir Bootstrap für das Design unserer Website verwendet. Die Abhängigkeit unserer Website von Bootstrap ermöglicht eine effiziente Gestaltung, indem wir auf vordefinierte Layouts, Schriftarten, Farbpaletten und interaktive Komponenten zurückgreifen. Dadurch schaffen wir eine moderne und ansprechende Benutzeroberfläche, die auf bewährten Designprinzipien basiert.

3.2.3 Node.js

Erläuterung

Node.js ist eine serverseitige JavaScript-Laufzeitumgebung, die es ermöglicht, JavaScript-Code außerhalb des Browsers auszuführen [17].



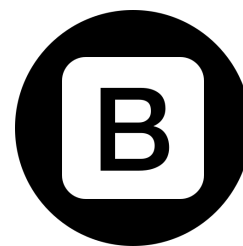
Verwendung

In unserem Fall haben wir npm genutzt, um verschiedene Pakete wie popper.js, rxjs, zone.js, htmlpdf.js usw. zu installieren. Diese Pakete sind im Verzeichnis `node_modules` abgelegt und stellen Funktionalitäten oder Erweiterungen dar, die in unserem Projekt verwendet werden können.

3.2.4 Bootstrap Icons

Erläuterung

Bootstrap Icons ist eine dedizierte Sammlung von Symbolen, die von Bootstrap bereitgestellt wird [14].



Verwendung

Auf unserer Website haben wir viele Icons aus der Bootstrap Icons-Sammlung integriert. Diese Icons dienen dazu, visuelle Elemente und Funktionen zu verbessern, indem sie eine klare und ansprechende Darstellung verschiedener Aktionen, Funktionen oder Informationen ermöglichen. Die Verwendung von Bootstrap Icons trägt dazu bei, dass unsere Website ein konsistentes und professionelles Erscheinungsbild beibehält, während gleichzeitig eine Vielzahl von Symbolen für unterschiedliche Anwendungsfälle zur Verfügung steht.

3.2.5 TTF

Erläuterung

TTF steht für TrueType Font, eine Schriftartentechnologie, die zur Darstellung von Schriftzeichen auf Computern verwendet wird [15].



Verwendung

In unserem Fall haben wir die Schrift für unsere PDF-Exportseite angepasst und dabei auf TrueType Fonts zurückgegriffen. Diese Anpassung ermöglicht eine präzise und gut lesbare Darstellung von Texten in den exportierten PDF-Dokumenten. TrueType Fonts werden oft für ihre Flexibilität in der Skalierung und Druckqualität geschätzt, was zu einer ansprechenden und professionellen Präsentation der textuellen Inhalte führt.

3.2.6 Angular

Erläuterung

Angular ist ein leistungsstarkes Open-Source-Framework von Google, das zur Entwicklung von modernen Single-Page-Anwendungen (SPAs) für die Webentwicklung verwendet wird. Es bietet eine umfangreiche Sammlung von Werkzeugen und Funktionen, die Entwicklern helfen, komplexe Anwendungen effizient zu erstellen. Angular basiert auf der Idee von Komponentennachitektur, bei der die Anwendung in einzelne Komponenten aufgeteilt wird, die jeweils ihre eigene Struktur, Stil und Logik haben. Das Framework bietet Funktionen wie Datenbindung, Dependency Injection, Routing und Formularverarbeitung, um die Entwicklung von SPAs zu vereinfachen. [16]



Verwendung

In unserem Fall haben wir uns mit Angular für die Clientseite unserer Anwendung beschäftigt, insbesondere für das Frontend. Dabei haben wir HTML für die Struktur der Benutzeroberfläche, CSS für das visuelle Layout und TypeScript für die Implementierung der Anwendungslogik genutzt. Angular ermöglichte es uns, eine dynamische und interaktive Benutzeroberfläche zu erstellen.

3.2.7 TypeScript Hero

Erläuterung

TypeScript Hero ist eine Erweiterung für Visual Studio Code, die dein Leben beim Codieren von TypeScript erleichtert. Wenn du viel TypeScript schreibst, möchtest du vielleicht, dass Visual Studio Code deine Imports organisiert und genau deswegen gibt es TypeScript Hero. [24]



Verwendung

Hier sind einige Gründe, warum wir TypeScript Hero in unserem Projekt verwenden haben:

- Organisation von Imports
- Codeübersicht
- Einfaches Hinzufügen von Imports

Insgesamt trägt TypeScript Hero dazu bei, die Entwicklungszeit zu verkürzen, indem es repetitive Aufgaben automatisiert und die Produktivität steigert. Es ist eine nützliche Ergänzung für jedes Angular-Projekt und kann dazu beitragen, deine Codequalität zu verbessern.

Die Installation dieser Erweiterung ist sehr einfach. . Führen Sie einfach den folgenden Befehl im Compiler von Visual Studio Code aus: `code --install-extension rbbbit.typescript-hero`. [24]

3.2.8 IntelliSense for CSS class names in HTML

Erläuterung

Diese Visual Studio Code-Erweiterung bietet IntelliSense für CSS-Klassen in HTML-Dateien. Sie vervollständigt automatisch CSS-Klassennamen basierend auf den Definitionen, die in deinem Arbeitsbereich oder in externen Dateien gefunden werden, die durch das Link-Element referenziert werden. [25].



Verwendung

Wir haben diese Erweiterung in unserem Projekt verwendet, um die Arbeit mit CSS-Klassen in HTML-Dateien zu erleichtern. Sobald HTML- oder JS-Dateien im Arbeitsbereich vorhanden sind, wird die Erweiterung automatisch aktiviert und beginnt, nach CSS-Klassendefinitionen zu suchen. Falls neue CSS-Klassen definiert werden oder neue CSS-Dateien hinzugefügt werden und automatische Vervollständigung gewünscht ist, genügt ein Klick auf das Blitzsymbol in der Statusleiste. Alternativ kann der Befehl durch Drücken von `Strg+Shift+P` (`Cmd+Shift+P` für Mac) ausgeführt und "Cache CSS-Klassendefinitionen" eingegeben werden.

Die Installation dieser Erweiterung ist sehr einfach.. Führen Sie einfach den folgenden Befehl im Compiler von Visual Studio Code aus: `code --install-extension zignd.html-css-class-completion`. [25].

3.2.9 HTML CSS Support

Erläuterung Die Erweiterung "CSS IntelliSense for HTML" bietet Autovervollständigung für HTML-IDs und -Klassenattribute in Visual Studio Code. Sie unterstützt das Navigieren zu Definitionen von CSS-Selektoren und validiert Klassenattribute. Die Konfiguration ermöglicht die Spezifizierung von lokalen und Remote-Stylesheets.[27]



Verwendung Die Verwendung dieser Erweiterung in unserem Projekt hat die Effizienz beim Schreiben von HTML-Code deutlich erhöht. Durch die Autovervollständigung von CSS-Klassennamen konnten wir Zeit sparen und Fehler reduzieren. Die Konfiguration erfolgte in der Datei `.vscode/settings.json` unseres Arbeitsbereichs, wo wir lokale Stylesheets spezifiziert haben, die berücksichtigt werden sollen. Diese Erweiterung wurde automatisch aktiviert und bot eine nahtlose Entwicklungsumgebung für HTML-Dateien in Visual Studio Code.

Die Installation dieser Erweiterung ist sehr einfach.. Führen Sie einfach den folgenden Befehl im Compiler von Visual Studio Code aus: `code --install-extension ecmel.vscode-html-css`.[27]

3.2.10 Angular Snippets

Erläuterung Diese Erweiterung bietet Code-Snippets für Angular-Anwendungen in TypeScript und HTML. Sie ermöglicht eine schnellere Entwicklung, indem sie häufig verwendeten Code automatisch generiert.[28]



Verwendung In unserem Projekt haben wir diese Snippets genutzt, um die Angular-Entwicklung zu beschleunigen. Durch Eingabe eines Teils des Snippet-Namens und Drücken der Tab-Taste konnten wir schnell auf häufig verwendeten Code zugreifen und ihn in unsere Dateien einfügen. Diese Erweiterung hat dazu beigetragen, unsere Produktivität zu steigern und die Entwicklung von Angular-Komponenten effizienter zu gestalten.

Die Installation dieser Erweiterung ist sehr einfach.. Führen Sie einfach den folgenden Befehl im Compiler von Visual Studio Code aus: `code --install-extension johnpapa.angular2`.[28]

3.2.11 Angular Language Service

Erläuterung Die Angular Language Service-Erweiterung bietet eine verbesserte Bearbeitungserfahrung für Angular-Templates in Visual Studio Code. Sie ermöglicht Funktionen wie Vervollständigungslisten, AOT-Diagnosemeldungen und Schnellinfo. [28]



Verwendung In unserem Projekt haben wir diese Erweiterung genutzt, um die Entwicklung von Angular-Anwendungen zu erleichtern. Durch die Installation konnten wir schnell auf Funktionen wie Vervollständigungslisten und Diagnosemeldungen zugreifen, was die Produktivität steigerte und die Qualität unserer Angular-Templates verbesserte.

Die Installation dieser Erweiterung ist sehr einfach.. Führen Sie einfach den folgenden Befehl im Compiler von Visual Studio Code aus: `code --install-extension angular.ng-template`.[28]

3.2.12 Move TS

Erläuterung Die Move TS-Erweiterung ermöglicht das Verschieben von TypeScript-Dateien und Ordnern innerhalb unseres Projekts. Dabei werden auch relative Importpfade innerhalb des Projekts automatisch aktualisiert. [29]



Verwendung In unserem Projekt haben wir diese Erweiterung genutzt, um TypeScript-Dateien und Ordner komfortabel zu verschieben, ohne dabei manuell Importpfade aktualisieren zu müssen.

Die Installation dieser Erweiterung ist sehr einfach.. Führen Sie einfach den folgenden Befehl im Compiler von Visual Studio Code aus: `code --install-extension stringham.move-ts` [29]

3.3 Backend

Das Backend bildet die Logik einer Software-Applikation ab. Beispiele hierfür sind zum Beispiel das Registrieren neuer Benutzer. Diese Abläufe finden nicht direkt im Frontend statt. Stattdessen werden die Daten mittels HTTP-Requests an das Backend übermittelt, wo die restliche Logik ausgeführt wird.

3.3.1 C#

Erläuterung

C# ist eine objektorientierte und typsichere Programmiersprache. Sie wurde von Microsoft im Jahr 2000 entwickelt. Diese Programmiersprache eignet sich besonders gut für die Entwicklung von Spielen. Unity basiert beispielsweise auf C# und wird weltweit millionenfach verwendet. Sie ist jedoch auch eine beliebte Wahl für die Entwicklung von Webseiten oder Android-Applikationen. Eine Besonderheit dieser Programmiersprache ist der "Garbage Collector". Der "Garbage Collector" überwacht während des Programmlaufs, welche Variablen nicht mehr benötigt werden. Sobald er solche Variablen findet, werden sie automatisch gelöscht, um Arbeitsspeicher freizugeben. Hinsichtlich der Syntax ähnelt sie C, C++ und auch Java, wobei C# die leistungsstärkste Programmiersprache von den dreien ist. [46]



Verwendung

Wir haben uns für diese Programmiersprache entschieden, weil sie ideal für unser Projekt ist und weil wir auch die meiste Erfahrung in dieser Sprache haben.

3.3.2 ASP.NET

Erläuterung

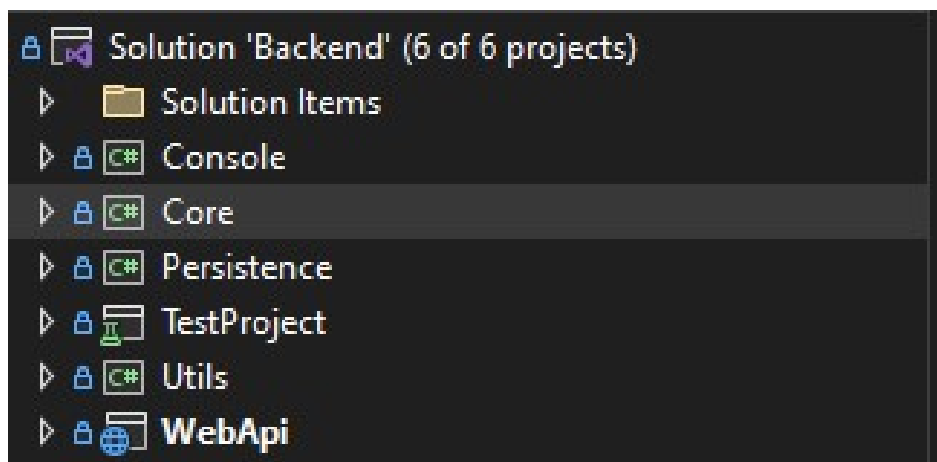
Active Server Pages .NET, kurz ASP.NET, ist ein Webframework, das von Microsoft entwickelt wurde und im Jahr 2002 erschien. Mit dieser Technologie ist vieles möglich, wie zum Beispiel die Erstellung von Websites oder die Nutzung der ASP.NET WebAPI, mit der es möglich ist, mit anderen Anwendungen mittels REST APIs zu kommunizieren [30].



Verwendung

In unserem Projekt haben wir die ASP.NET API verwendet, um mit dem Frontend mittels HTTP-Requests zu kommunizieren.

Im folgenden Bild ist der Aufbau unseres Backends dargestellt. Mehr Informationen dazu auf Seite 92:



3.3.3 Entity Framework Core

Erläuterung

Das Entity Framework Core ermöglicht es Benutzern, mit unterschiedlichen Datenbank-Engines und .NET-Objekten zu arbeiten. Was es außerdem macht, ist, dass es .NET-Objekte in SQL-Anfragen übersetzt und umgekehrt dasselbe tut. Der Zugriff auf Datenbanken und die Übersetzung werden auch als objektrelationales Mapping (ORM) bezeichnet.



Grundsätzlich ist es in Modell und Kontext aufgebaut. Das Modell umfasst die Entitäten, die dann auch in der Datenbank als Tabellen zu sehen sind. Der Kontext ist die Kommunikation zwischen dem Anwendungscode und der Datenbank. Das Abrufen und Speichern von Daten erfolgt mit LINQ, was für Language Integrated Query oder auf Deutsch sprachintegrierte Abfrage steht.

Außerdem ist es auch möglich, mithilfe von Annotationen, z. B. Beziehungen zwischen Entitäten zu erstellen. [31]

Verwendung

In unserem Fall haben wir das Entity Framework verwendet, um mit dem Anwendungscode und der Datenbank zu kommunizieren. Wir haben das Projekt ebenfalls so erstellt, dass die Entitäten und die Kommunikation mit der Datenbank getrennt sind.

Um eine Migration zu erstellen, muss man in der Kommandozeile oder in der PowerShell folgende Befehle eingeben:

1. In das Persistence-Verzeichnis navigieren, wo sich die Migration befinden soll.
2. `dotnet ef migrations add [Name der Migration]`
3. `dotnet ef database update`

3.3.4 MYSQL

Erläuterung

MySQL ist eine Open-Source-relationale Datenbank, mit der auch die Verwaltung von Datenbanken möglich ist. Für einen einfachen Zugriff und Abfrage speichert sie Daten organisiert und strukturiert. Der Gründer ist Oracle, der sie entwickelt hat und bis heute unterstützt. Sie ist besonders beliebt bei Webanwendungen und CMS-Systemen (Content Management Systemen). Sie ist auch für alle anderen Betriebssysteme verfügbar, wie z. B. Windows, Linux und MacOS [32].



Verwendung

Im Projekt haben wir die MySQL-Datenbank verwendet, um Daten wie z. B. Bestellungen zu speichern und diese dann anzeigen und auswerten zu können. MySQL bietet auch eine sehr leichte Art und Weise, um Schemas zu erstellen und zwischen ihnen zu wechseln.

Bestellsystem-KUK in der MySQL-Datenbank:

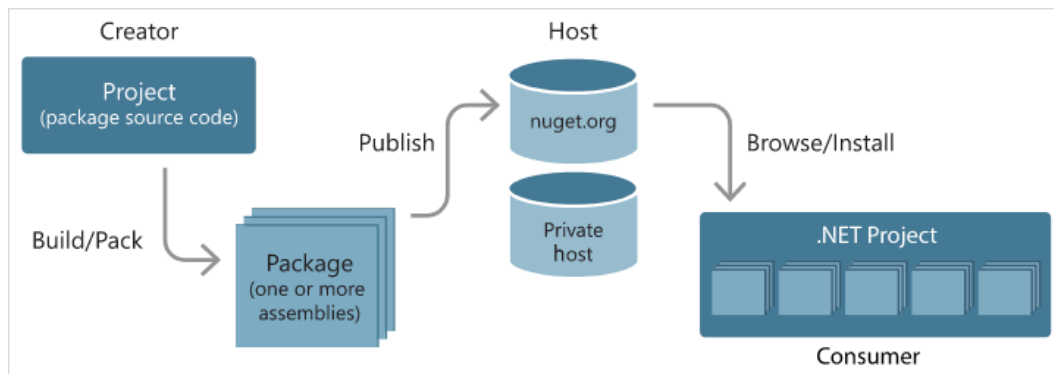


3.3.5 Nuget-Pakete

NuGet ist der Paketmanager für C# und andere .NET-Sprachen. Es ermöglicht Entwicklern, Bibliotheken aus Community-Quellen herunterzuladen und auch eigene Bibliotheken hochzuladen.

Grundsätzlich besteht ein NuGet-Paket aus einer ZIP-Datei mit der Endung ".nupkg". In dieser ZIP-Datei befinden sich kompilierte Codes (DLLs).

NuGet agiert als zentrales Repository mit über 100.000 Paketen auf nuget.org. Die Pakete können auch privat auf verschiedenen Plattformen gehostet werden, um den Zugriff zu steuern. Es verbindet Paketersteller und Paketverbraucher, sodass diese die Pakete herunterladen und in ihren Projekten verwenden können [34].



3.3.6 BouncyCastle.Cryptography

Erläuterung

BouncyCastle.Crypto ist eine kryptografische Open-Source-Bibliothek, die für die Java- und .NET-Umgebung entwickelt wurde und verwendet wird.

Diese Bibliothek bietet eine Vielzahl von Kryptografie-Algorithmen und Funktionen wie zum Beispiel: [35].



- Digests:
 - MD5 (Message-Digest Algorithm)
 - SHA-256 (Secure Hash Algorithm -> 256 für die Länge des Hashwerts in Bit)
 - Tiger
- Schlüsselvereinbarung:
 - Diffie-Hellman
 - EC-DH (Elliptic Curve Diffie-Hellman)

Verwendung

In diesem Projekt wurde zwar dieses NuGet-Paket installiert, aber es wurde noch nicht verwendet. Der Grund ist, dass am Anfang geplant war, die Passwörter beim Anlegen eines Kontos zu hashen und dann in die Datenbank zu speichern. Die Firma fand diese Option als nicht dringend und meinte, dass wir darüber nochmal sprechen werden, ob es nötig ist, wenn das Projekt abgeschlossen ist.

3.3.7 DotNetZip

Erläuterung

DotNetZip ist eine Open-Source-Bibliothek, die ZIP-Dateien lesen, erstellen und ändern kann [36].

Verwendung

Das .NET Framework hat nicht wirklich die Möglichkeit ZIP-Dateien zu lesen, erstellen und zu ändern. Da wir diese Funktionen für unseren PDF-Export benötigen, haben wir recherchiert und sind schließlich auf diese Bibliothek gestoßen, für die wir uns dann auch entschieden haben.

3.3.8 itext

Erläuterung

itext ist eine Bibliothek zum Erstellen und Manipulieren von PDF-Dateien, die von Bruno Lowagie, Paulo Soares und anderen Entwicklern für Programmiersprachen wie Java und C# entwickelt wurde. Mit dieser Bibliothek ist es möglich, per Code eine PDF-Datei zu schreiben oder zu ändern [37].



Verwendung

Diese Bibliothek haben wir ebenfalls für den PDF-Export benötigt, diesmal jedoch nicht für die Erstellung von ZIP-Dateien, sondern für die Erstellung von PDF-Dateien.

3.3.9 itext.bouncy-castle-adapter

Erläuterung

iText.BouncyCastle-Adapter ist eine zusätzliche Bibliothek, die iText benötigt. Zum Beispiel führt das Fehlen dieser Bibliothek zu einer Unkown **PdfException**, und auch für die digitale Signatur ist dieses NuGet-Paket erforderlich [38].

Verwendung

Wir haben dieses NuGet-Paket nicht für die digitale Signatur von PDF-Dateien verwendet, sondern nur, weil es ohne dieses NuGet-Paket zu Fehlern kommt.

3.3.10 Microsoft.AspNetCore.Authentication.Negotiate

Erläuterung

Mit diesem NuGet-Paket ist es möglich, Authentifizierungsanfragen, zum Beispiel im Unternehmen, auf das Active Directory mit Negotiate, Kerberos oder NTLM (NT Lan Manager) zu stellen. Es eignet sich besonders gut für ASP.NET-Anwendungen.

Verwendung

Wir haben dieses Paket benötigt, um die Benutzerinformationen aus dem Active Directory zu holen. Der Grund dafür ist, dass in unserem Projekt auch die Benutzer aus dem Active Directory anmelden können.

3.3.11 Microsoft.EntityFrameworkCore.Tools

Erläuterung

Mit dieser Bibliothek ist es möglich, Migrationen zu verwalten, DbContext und Entitätstypen zu generieren. Außerdem können mit den EF Core Tools im NuGet Package Manager Console Befehle ausgeführt werden wie:

- Add-Migration: Erstellt eine neue Migration
- Bundle-Migration: Bündelt mehrere Migrationen.
- Drop-Database: Löscht die Datenbank.
- Get-DbContext: Zeigt Informationen zum DbContext an.
- Get-Migration: Zeigt Informationen zu vorhandenen Migrationen an.
- Optimize-DbContext: Optimiert den DbContext.
- Remove-Migration: Entfernt eine Migration.
- Scaffold-DbContext: Generiert Code für den DbContext basierend auf einer vorhandenen Datenbank.
- Script-Migration: Erstellt ein Skript für eine Migration.
- Update-Database: Aktualisiert die Datenbank. [39]

Verwendung

Diese Bibliothek ist eines der Pakete, die installiert werden müssen, damit das Entity Framework funktionieren kann. Zum Beispiel sind die Erstellung von Migrationen notwendig.

3.3.12 Microsoft.Extensions.Configuration.Json

Erläuterung

Dieses NuGet-Paket ermöglicht das Lesen von Anwendungseinstellungen aus einer JSON-Datei. Das Paket enthält 4 Klassen:

- `JsonConfigurationProvider`
- `JsonConfigurationSource`
- `JsonStreamConfigurationProvider`
- `JsonStreamConfigurationSource` [40]

Verwendung

Auch dieses NuGet-Paket ist notwendig für die Funktionalität des Entity Frameworks.

3.3.13 MySql.Data

Erläuterung

`MySql.Data` ermöglicht die Verbindung und Interaktion mit MySQL-Datenbanken in .NET-Anwendungen. Die Hauptfunktionen sind: Herstellung der Datenbankverbindung, Ausführung von SQL-Abfragen, Verarbeitung von Datensätzen und Verwaltung von Transaktionen [41].

Verwendung

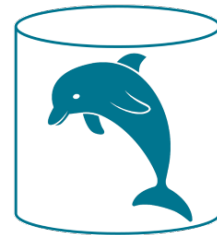
In unserem Projekt haben wir dieses NuGet-Paket für die SQL-Abfragen an unsere MySQL-Datenbank verwendet. Dies ermöglicht uns Dinge wie zum Beispiel das Anlegen einer neuen Bestellung.

3.3.14 MySqlConnection

Erläuterung

MySqlConnection ist eine Bibliothek, die einen C# ADO.NET-Treiber (eine Sammlung von Klassen, die den Zugriff auf relationale Datenbanken ermöglicht) für folgende Datenbanken bereitstellt:

- MySQL
- MariaDB
- Amazon Aurora
- Und andere MySQL-kompatible Datenbanken [42]



Verwendung

Wir haben dieses NuGet-Paket für die Verbindung zu unserer MySQL-Datenbank verwendet.

3.3.15 MySql.EntityFrameworkCore

Erläuterung

Das MySql.EntityFrameworkCore ist grundsätzlich wie das Entity Framework Core, nur dass dieses Paket MySQL-Datenbanken unterstützt. [43]

Verwendung

Da wir einen objektrelationalen Mapper benötigen, um Klassen als Tabellen in der Datenbank anzeigen zu können, und das für eine MySQL-Datenbank, benötigen wir dieses NuGet-Paket.

3.3.16 System.DirectoryServices.AccountManagement

Erläuterung

Mit diesem NuGet-Paket ist es möglich, mit Benutzer- und Gruppenkonten in einem Active Directory zu interagieren. Es bietet eine vereinfachte API für die Verwaltung von Benutzerkonten, Gruppen, Authentifizierung und Autorisierung in Active Directory-Umgebungen.

Die Hauptfunktionen dieser Bibliothek sind:

- Benutzer- und Gruppenverwaltung:
 - Mit den Klassen „UserPrincipal“ und „GroupPrincipal“ können Funktionen wie das Erstellen, Löschen, Bearbeiten und Suchen von Benutzern und Gruppen durchgeführt werden.
- Authentifizierung und Autorisierung:
 - Es ermöglicht die Überprüfung von Benutzeranmeldeinformationen und das Festlegen von Berechtigungen für Benutzer und Gruppen im Active Directory.
- Zugriff auf Benutzerinformationen:
 - Es ermöglicht den Zugriff auf Benutzerinformationen wie Benutzername, E-Mail-Adresse, Gruppenmitgliedschaften und andere Informationen, die im Active Directory gespeichert sind.
- Verwaltung von Benutzerprofilen:
 - Die Verwaltung von Benutzerprofilen ist möglich, einschließlich Passwortänderungen, Aktualisierung von Informationen und Sperren von Benutzerkonten.

Dieses Paket ist sehr nützlich für Anwendungen, die mit einem Active Directory arbeiten. Vor allem bietet es einen großen Vorteil, indem es die Verwaltung von Benutzer- und Gruppenkonten vereinfacht und automatisiert. [44]

Verwendung

Dieses NuGet-Paket war sehr wichtig für unser Projekt, weil wir dadurch die Möglichkeit hatten, z. B. die Benutzerinformationen aus dem ADS zu erhalten und sie dann für die Anmeldung zu verwenden.

3.3.17 Swagger

Erläuterung

Swagger UI ist ein Open-Source-Projekt, das API (Application Programming Interfaces) Methoden visualisiert. Es basiert auf der OpenAPI-Spezifikation. Ursprünglich wurde es von Reverb entwickelt, aber jetzt ist es herstellerneutral. Zusammengefasst beschreibt und dokumentiert Swagger APIs, insbesondere RESTful APIs. [45]



User		^
GET	/api/User/CheckEmailAddress	▼
GET	/api/User/GetActiveDirectoryFromUser	▼
GET	/api/User/CheckPassword	▼
GET	/api/User/LoginUser	▼
GET	/api/User/GetIdByEmailAddress	▼
GET	/api/User/GetUserByEmailAddress	▼
GET	/api/User/GetUserById	▼
GET	/api/User/GetPasswordFromEmail	▼
GET	/api/User/GetAllUsersByDepartmentId	▼
GET	/api/User/GetSixDigitCodeFromCurrentCode	▼
GET	/api/User/GetAuthenticatedUser	▼
POST	/api/User/RegisterUser	▼
DELETE	/api/User/DeleteUser	▼
PUT	/api/User/ChangePassword	▼
PUT	/api/User/GenerateSixDigitCode	▼
PUT	/api/User/ChangeAuthentication	▼

Wie auf dem Bild zu sehen ist, sind die 4 HTTP-Methoden farblich unterteilt, was zu einer besseren Übersicht führt. Außerdem sind die Klassen separat aufgeführt, wie im Bild zu sehen ist, wobei nur die HTTP-Methoden der Klasse „User“ angezeigt werden.

Verwendung

Wir haben Swagger UI verwendet, weil das Testen der HTTP-Methoden einfacher ist. Sollte z. B. das Frontend gerade nicht funktionieren und man möchte wissen, ob die Methode funktioniert, kann man dies leicht über den Swagger UI überprüfen.

4 Implementierung

4.1 Prozessablauf

4.1.1 Anmeldung


Auf unserer Website besteht die Möglichkeit, sich auf drei verschiedene Arten anzumelden. Diese Optionen sind:

Externe Anmeldung:

- Nur Benutzer, die von internen Benutzern erstellt wurden, können sich extern anmelden.
- Externe Benutzer haben beschränkten Zugriff und müssen von einem internen Benutzer genehmigt werden, bevor sie sich anmelden können.

Interne Anmeldung:

- Interne Benutzer können sich mithilfe von Active Directory Domain Services (AD DS) anmelden.
- AD DS ermöglicht eine zentrale Verwaltung von Benutzerkonten, Authentifizierung und Autorisierung.



Windows-
Benutzername
eingeben

Benutzername
Passwort

[Zurück zur Login-Seite](#)

Anmelden (Active Directory)

Abbildung 4: Anmeldung AD DS

Admin-Anmeldung:

- Die Anmeldedaten für Admin-Benutzer werden an die Gesundheitsholding übergeben, um den Zugang zum Admin-Dashboard zu ermöglichen.
- Die Admin-Benutzer loggen sich mit den bereitgestellten Anmeldeinformationen in das Admin-Dashboard ein.



Bitte melden Sie sich an

E-Mail Adresse
Passwort

[Anmeldung mit Active Directory](#)

Anmelden

Abbildung 5: Externe Anmeldung und Admin-Zugriff

In der dargestellten Abbildung wird der Anmeldeprozess für Administratoren und externe Benutzer illustriert. Es erfolgt eine sorgfältige Überprüfung der eingegebenen Anmeldedaten, um festzustellen, ob es sich um einen Administrator oder einen externen Benutzer handelt. Während Administratoren direkten Zugriff auf das Admin-Dashboard erhalten, werden externe Benutzer nach erfolgreicher Identifikation automatisch an das entsprechende Benutzer-Dashboard weitergeleitet.

Der folgende Codeabschnitt veranschaulicht die spezifische Behandlung der Anmeldung für Admin-Benutzer:

```
1 // Admin-Anmeldung
2 if (email.includes('admin@bestellsystem.local')) {
3     this.adminService.GetAdminUser(email, password).subscribe({
4         next: (data: any) => {
5             if (data) {
6                 this.toastr.success('Anmeldung erfolgreich');
7                 sessionStorage.setItem('email', email);
8                 this.router.navigate(['/user-dashboard']);
9             }
10            this.loading = false;
11        },
12        error: () => {
13            this.toastr.error('Fehler beim Laden der Benutzer Rechte');
14            this.loading = false;
15        }
16    });
17 }
```

Der nachfolgende Code beschreibt den Anmeldeprozess für externe Benutzer, inklusive der Überprüfung der 2FA-Aktivierung und des Aktivierungsstatus des Benutzerkontos:

```
1 // Externe Benutzer-Anmeldung
2 this.authenticationService.authenticate(email, password).then((isAuthenticated:
3     boolean) => {
4     if (isAuthenticated) {
5         this.userService.GetAuthenticatedUser(email).subscribe({
6             next: (data: any) => {
7                 if (data.isActive && data.is2FAEnabled) {
8                     this.toastr.success('Anmeldung erfolgreich');
9                     sessionStorage.setItem('email', email);
10                    this.router.navigate(['/authentication']);
11                }
12                this.loading = false;
13            },
14            error: () => {
15                this.toastr.error('Fehler beim Laden des Benutzers');
16                this.loading = false;
17            }
18        });
19    }
20    }).catch(() => {
21        this.toastr.error('Fehler bei der Anmeldung');
22        this.loading = false;
23    });
```

4.1.2 Authentifizierung über Active Directory

Innerhalb unserer Webanwendung ermöglicht die `LoginWithAD`-Methode die Überprüfung der Anmeldeinformationen eines Benutzers gegenüber dem Active Directory (AD).

Der folgende Codeabschnitt zeigt die Implementierung dieser Funktionalität:

```
1 [HttpGet("LoginWithAD")]
2 public async Task<IActionResult> LoginWithAD(string domain, string username,
3     string password)
4 {
5     using (var context = new PrincipalContext(ContextType.Domain, domain))
6     {
7         User? user = await
8             _unitOfWork.UserRepository.GetUserByEmailAddress(username +
9                 "@bestellsystem.local");
10        if (user == null)
11        {
12            return Ok(false);
13        }
14        bool isAuthenticated = context.ValidateCredentials(username, password);
15        return Ok(isAuthenticated);
16    }
17 }
```

Im ersten Schritt wird eine Instanz von `PrincipalContext` verwendet, um eine Verbindung zum AD aufzubauen, wobei der Domänenname als Parameter übergeben wird. Anschließend wird geprüft, ob ein Benutzer mit der angegebenen E-Mail-Adresse in der lokalen Datenbank existiert. Dies stellt sicher, dass nur bekannte Benutzer sich anmelden können.

Die Methode `ValidateCredentials` von `PrincipalContext` wird dann verwendet, um die übergebenen Anmeldeinformationen - Benutzername und Passwort - gegenüber dem AD zu validieren. AD prüft, ob die Kombination aus Benutzername und Passwort korrekt ist. Das Ergebnis der Überprüfung ist ein boolescher Wert, der angibt, ob die Authentifizierung erfolgreich war (`true`) oder nicht (`false`).

Diese Vorgehensweise gewährleistet, dass der Authentifizierungsprozess sicher und direkt gegen die vertrauenswürdige Quelle der Benutzerinformationen, das Active Directory, erfolgt.

4.1.3 Admin-Dashboard

Im Admin-Dashboard sind vier Sektionen verfügbar: Benutzer, Vorgesetzter 1, Vorgesetzter 2 und Vorgesetzter 3. Der Benutzer, der über den Admin-Account verfügt, hat die Befugnis, die Rollen der verschiedenen Gruppen zu bearbeiten. Folgende Rollen stehen zur Bearbeitung zur Verfügung:

Für alle Vorgesetzten-Gruppen:

- Benutzer bearbeiten
- Benutzer erstellen
- Lokale Lieferanten erstellen
- Benutzer aktivieren/deaktivieren
- Bestellungen und Statistiken einsehen

Für die Benutzergruppe:

- Bestellungen und Statistiken einsehen
- Lokale Lieferanten erstellen

Diese klar strukturierte Aufteilung der Berechtigungen innerhalb des Admin-Dashboards erleichtert eine effiziente Verwaltung für jede Gruppe. Durch die gezielte Zuteilung von Rechten für alle Vorgesetzten-Gruppen und spezifisch für die Benutzergruppe wird eine präzise Kontrolle über die Aktionen und Aufgaben ermöglicht. Diese klare Differenzierung fördert nicht nur die Sicherheit, sondern auch die Transparenz bei der Verwaltung von Benutzern und Ressourcen. Administratoren können somit die Zugriffsrechte feinabgestimmt anpassen und so den individuellen Anforderungen und Verantwortlichkeiten jeder Gruppe gerecht werden. Insgesamt trägt diese präzise Berechtigungsstruktur dazu bei, die Effizienz und Sicherheit des Administrationsprozesses im Dashboard zu optimieren.

Benutzergruppe: In der Benutzergruppe kann der Benutzer Bestellungen aufgeben, Lokale Lieferanten erstellen, Statistiken einsehen und allgemeine Benutzerverwaltungsaufgaben durchführen. Die Benutzergruppe ist für tägliche Aktivitäten und allgemeine Verwaltungsaufgaben konzipiert.

Administrations-Dashboard

Seien Sie vorsichtig bei der Bearbeitung der Rollen und Rechte der Gruppen.

Bevor Sie mit der Einreichung fortfahren, überprüfen Sie bitte alle Angaben, um die Richtigkeit sicherzustellen. Obwohl die Einreichung bei Bedarf korrigiert werden kann, ist es wichtig, alles gründlich zu überprüfen, um Unannehmlichkeiten zu vermeiden.

Benutzer [Vorgesetzter 1](#) [Vorgesetzter 2](#) [Vorgesetzter 3](#)

Administrative Zugriffssteuerung für Benutzergruppen

Durch die administrative Zugriffssteuerung können bestimmte Benutzergruppen im Bestellsystem gezielt autorisiert werden, auf bestimmte Funktionen und Ressourcen zuzugreifen.

Bestellen	Benutzer mit diesem Recht können Bestellungen im System aufgeben und Produktauswählen, um den Bestellvorgang abzuschließen	<input checked="" type="checkbox"/>
Lokale Lieferant Erstellen	Lokale Lieferanten können durch User erstellt werden.	<input checked="" type="checkbox"/>
Statistiken anzeigen	Benutzern mit diesem Recht wird die Möglichkeit geboten, ihre persönlichen Statistiken, einschließlich Bestellverlauf und Gesamtbestellmenge, einzusehen.	<input checked="" type="checkbox"/>

[Zurück zur Anmeldung](#)

Abbildung 6: Benutzergruppe im Admin-Dashboard

Vorgesetzter 1: Vorgesetzter 1 hat erweiterte Berechtigungen, einschließlich der Möglichkeit, Benutzer zu bearbeiten und zu erstellen, lokale Lieferanten zu erstellen, Benutzer zu aktivieren/deaktivieren, sowie Bestellungen und Statistiken einzusehen. Dies ermöglicht eine umfassendere Verwaltung auf höherer Ebene.

Administrations-Dashboard

Seien Sie vorsichtig bei der Bearbeitung der Rollen und Rechte der Gruppen.
 Bevor Sie mit der Einreichung fortfahren, überprüfen Sie bitte alle Angaben, um die Richtigkeit sicherzustellen. Obwohl die Einreichung bei Bedarf korrigiert werden kann, ist es wichtig, alles gründlich zu überprüfen, um Unannehmlichkeiten zu vermeiden.

Benutzer
Vorgesetzter 1
Vorgesetzter 2
Vorgesetzter 3

Effiziente Zugriffsverwaltung für Vorgesetzter 1-Benutzer
 Die effiziente Zugriffsverwaltung ermöglicht es, spezifische Rechte und Funktionen im Bestellsystem zu nutzen, um die Benutzer in ihren Abteilungen optimal zu verwalten und Bestellaktivitäten zu überblicken.

Benutzer bearbeiten	Vorgesetzter 1 kann die Informationen von Benutzern in einer bestimmten Abteilung bearbeiten und aktualisieren	<input type="checkbox"/>
Benutzer erstellen	Vorgesetzter 1 hat die Berechtigung, neue Benutzerprofile für eine spezifische Abteilung anzulegen.	<input checked="" type="checkbox"/>
Lieferant Erstellen	Lokale Lieferanten können durch Vorgesetzter 1 erstellt werden.	<input checked="" type="checkbox"/>
Benutzer aktivieren / deaktivieren	Vorgesetzter 1 kann die Aktivierungs- oder Deaktivierungsfunktion für Benutzer in der gesamten Abteilung verwalten	<input type="checkbox"/>
Bestellen	Vorgesetzter 1 ist berechtigt, Bestellungen über das System aufzugeben und Produkte auszuwählen.	<input checked="" type="checkbox"/>
Statistiken einsehen	Vorgesetzter 1 kann statistische Informationen einsehen, wie den Bestellverlauf und die gesamte Anzahl der getätigten Bestellungen.	<input checked="" type="checkbox"/>

Zurück zur Anmeldung

Abbildung 7: Vorgesetzter 1 im Admin-Dashboard

4.1.4 Rollenbasierte Zugriffssteuerung im Active Directory

In unserer Anwendung wird eine differenzierte rollenbasierte Zugriffssteuerung umgesetzt, die es ermöglicht, bestimmte Berechtigungen basierend auf der Benutzerrolle zuzuweisen. Der folgende C#-Code aus unserem Web-API-Controller illustriert den Prozess der Rollenzuweisung:

```

1  switch (groupName)
2  {
3      case "User":
4          Group? group1 = await _dbContext.Groups.FirstOrDefaultAsync(g => g.GroupName
5              == groupName);
6          if (group1 != null)
7          {
8              var rolesInGroup = await _dbContext.Roles.Where(role => role.Group_Id ==
9                  group.Id).ToListAsync();
10             var roleIds = rolesInGroup.Select(role => role.Id).ToList();
11
12             foreach (var role in rolesInGroup)
13             {
14                 switch(role.RoleName)
15                 {
16                     case "Bestellen":
17                         userRoles.Add(new UserRole { User_Id = adUser.Id, Role_id =
18                             role.Id, isActive = true }); // Add Bestellen role
19                         break;
20                     case "Statistik ansehen":
21                         userRoles.Add(new UserRole { User_Id = adUser.Id, Role_id =
22                             role.Id, isActive = true }); // Add Statistik ansehen role
23                         break;
24                     case "Deliverant erstellen (lokal)":
25                         userRoles.Add(new UserRole { User_Id = adUser.Id, Role_id =
26                             role.Id, isActive = true }); // Add Deliverant erstellen
27                             (lokal) role
28                         break;
29                 }
30             }
31         }
32         break;
33     case "Superior 1":
34
35     Group? group4 = await _dbContext.Groups.FirstOrDefaultAsync(g => g.GroupName
36         == groupName);
37     if (group4 != null)
38     {
39         var rolesInGroup = await _dbContext.Roles.Where(role => role.Group_Id ==
40             group4.Id).ToListAsync();
41         var roleIds = rolesInGroup.Select(role => role.Id).ToList();
42
43         foreach (var role in rolesInGroup)
44         {
45             switch (role.RoleName)
46             {
47                 case "Bestellen":
48                     userRoles.Add(new UserRole { User_Id = adUser.Id, Role_id =
49                         role.Id , isActive = true }); // Add Bestellen role
50                     break;
51                 case "User erstellen (lokal)":
52                     userRoles.Add(new UserRole { User_Id = adUser.Id, Role_id =
53                         role.Id }); // Add User erstellen (lokal) role
54                     break;
55                 case "User bearbeiten (lokal)":
56                     userRoles.Add(new UserRole { User_Id = adUser.Id, Role_id =
57                         role.Id }); // Add User bearbeiten (lokal) role
58                     break;
59                 case "Deliverant erstellen (lokal)":
60                     userRoles.Add(new UserRole { User_Id = adUser.Id, Role_id =
61                         role.Id }); // Add Deliverant erstellen (lokal) role
62                     break;
63                 case "User aktivieren / deaktivieren":
64                     userRoles.Add(new UserRole { User_Id = adUser.Id, Role_id =
65                         role.Id }); // Add User aktivieren / deaktivieren role
66                     break;
67                 case "Statistik ansehen":
68                     userRoles.Add(new UserRole { User_Id = adUser.Id, Role_id =
69                         role.Id, isActive = true });
70                     break; // Add Statistik ansehen role
71             }
72         }
73     }
74 }

```

```

57     }
58   }
59 }
60
61 break;
62 case "Superior 2":
63 case "Superior 3":
64   case "FinanceDepartment":
65     Group? group5 = await _dbContext.Groups.FirstOrDefaultAsync(g =>
66       g.GroupName == groupName);
67     if (group5 != null)
68     {
69       // Retrieve the "Statistik ansehen" role associated with the
70       // "FinanceDepartment" group from the database
71       var rolesInGroup = await _dbContext.Roles.Where(role => role.Group_Id
72         == group5.Id).ToListAsync();
73       var roleIds = rolesInGroup.Select(role => role.Id).ToList();
74
75       foreach (var role in rolesInGroup)
76       {
77         switch (role.RoleName)
78         {
79           case "Statistik ansehen":
80             userRoles.Add(new UserRole { User_Id = adUser.Id, Role_id
81               = role.Id, isActive = true });
82             // Add Statistik ansehen role
83             break;
84         }
85       }
86     }
87   }
88   break;
89 default:
90   // Default roles for other groups
91   break;
92 }

```

Der Code verwendet eine `switch`-Anweisung, um die Gruppenzugehörigkeit eines Benutzers zu überprüfen. Für jede Gruppe, beispielsweise 'User', 'Superior 1-3' oder 'FinanceDepartment', werden über eine Datenbankabfrage die zugehörigen Rollen ermittelt und dem Benutzer zugewiesen. Dies geschieht mithilfe des `UserRole`-Modells, das eine Verknüpfung zwischen dem Benutzer und seinen Rollen herstellt, wobei auch der Aktivierungsstatus der Rolle berücksichtigt wird.

Insbesondere wird die Methode `ValidateCredentials` des `PrincipalContext` verwendet, um die Anmeldeinformationen eines Benutzers zu validieren. Wenn die Validierung erfolgreich ist (das heißt, die Anmeldeinformationen stimmen mit denen im AD überein), wird `true` zurückgegeben, was bedeutet, dass der Benutzer die entsprechenden Rollen zugewiesen bekommt und entsprechend Zugang zu verschiedenen Teilen des Systems hat, abhängig von seinen Berechtigungen.

4.1.5 Bestellformular

Unser Bestellformular bildet das Herzstück unserer Webanwendung. Hier müssen Benutzer verschiedene Informationen eingeben, um eine Bestellung aufzugeben. Das Formular ist so gestaltet, dass es die erforderlichen Daten abfragt, einschließlich Pflichtfelder und optionale Informationen für eine präzise Bestellabwicklung.

Das Feld zum Anhängen von Dateien befindet sich nicht ganz unten im Formular. Es ermöglicht Benutzern, relevante Dateien wie Angebote oder andere unterstützende Dokumente anzuhängen. Diese Funktion bietet den Benutzern die Möglichkeit, zusätzliche Informationen oder Unterlagen hinzuzufügen, die für die Bearbeitung ihrer Bestellung relevant sein könnten.

Bestell-Formular

Datum:
29/03/2024

Besteller:
Emma Johnson

Abteilung:
Abteilung 1

Lieferanten wählen:
3Dee GmbH & Co KG

KreditorenNr:
28536

BBG-Shop-Bestellung:
 Ja
 Nein

Bekanntmachung nach BVergG 2018 nötig:
 Ja
 Nein

GZ-Nummer wenn BBG-Shop-Bestellung:
GZ

Betreff:

Anmerkungen:

Formale Grundlage wählen:

Lieferbedingungen:

Lieferadresse:
Holzstraße 2, 4040 Linz

Dateianhänge:
Choose Files | No file chosen

Abbildung 8: Bestellformular Teil 1

Nachdem die grundlegenden Informationen im Bestellformular eingegeben wurden, bietet das Formular einen Abschnitt, in dem spezifische Details zu den zu bestellenden Artikeln eingetragen werden können. Diese Informationen werden in einer Tabelle erfasst und umfassen Folgendes für jede Bestellposition:

- **Anzahl:** Die Menge des bestellten Artikels.
- **Komponentenbezeichnung:** Eine klare Bezeichnung des Artikels.
- **Nettopreis:** Der Nettopreis des Artikels.
- **Gesamtpreis:** Der Gesamtpreis für die Menge des bestellten Artikels.
- **Betroffenes Krankenhaus:** Das Krankenhaus, dem die Bestellung zugeordnet ist.
- **Budget:** Das Budget, das für die Bestellung verwendet wird.
- **Zuordnung:** Die Zuordnung der Bestellung zu einer bestimmten Abteilung oder Kategorie.
- **Auftrag/KST oder Invest:** Informationen über den spezifischen Auftrag, die Kostenstelle oder Investition.

Die Tabelle ist so strukturiert, dass zu Beginn fünf Zeilen für fünf verschiedene Produkte angezeigt werden. Wenn jedoch mindestens eine Zeile nicht vollständig ausgefüllt ist, wird eine Fehlermeldung angezeigt, die besagt, dass mindestens eine Bestellung getätigt werden muss. Wenn mehr als fünf Zeilen benötigt werden, kann der Benutzer über ein ”+“-Symbol zusätzliche Zeilen hinzufügen.

Am Ende des Formulars befinden sich zwei Schaltflächen. Eine Schaltfläche ermöglicht es dem Benutzer, zur Startseite zurückzukehren, wodurch der gesamte Fortschritt abgebrochen wird. Die andere Schaltfläche ist zum Speichern bestimmt. Beim Speichern werden alle eingegebenen Daten überprüft, um sicherzustellen, dass alle Pflichtfelder ausgefüllt wurden. Wenn nicht, wird eine entsprechende Fehlermeldung angezeigt.

ID	Anzahl	Komponentenbezeichnung	Nettopreis	Gesamtpreis	Betroffenes Krankenhaus	Budget	Zuordnung	Auftrag/KST oder Invest
1	<input type="text" value="10"/>	<input type="text" value="Medizinisches Thermomet"/>	<input type="text" value="15"/>	<input type="text" value="150"/>	<input type="text" value="IT-NMC"/>	<input type="text" value="MI Sonstig"/>	<input type="text" value="Wartungsverträ"/>	<input type="text" value="28000070"/>
2	<input type="text" value="0"/>	<input type="text"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="IT-NMC"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="0"/>
3	<input type="text" value="0"/>	<input type="text"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="IT-NMC"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="0"/>
4	<input type="text" value="0"/>	<input type="text"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="IT-NMC"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="0"/>
5	<input type="text" value="0"/>	<input type="text"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="IT-NMC"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="0"/>

+

Zurück zur Startseite
Speichern

Abbildung 9: Bestellformular Teil 2

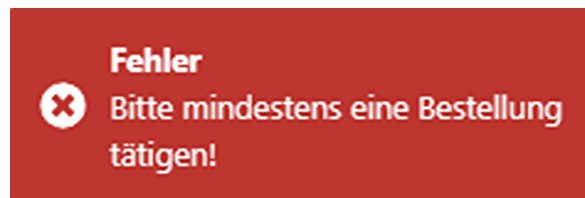
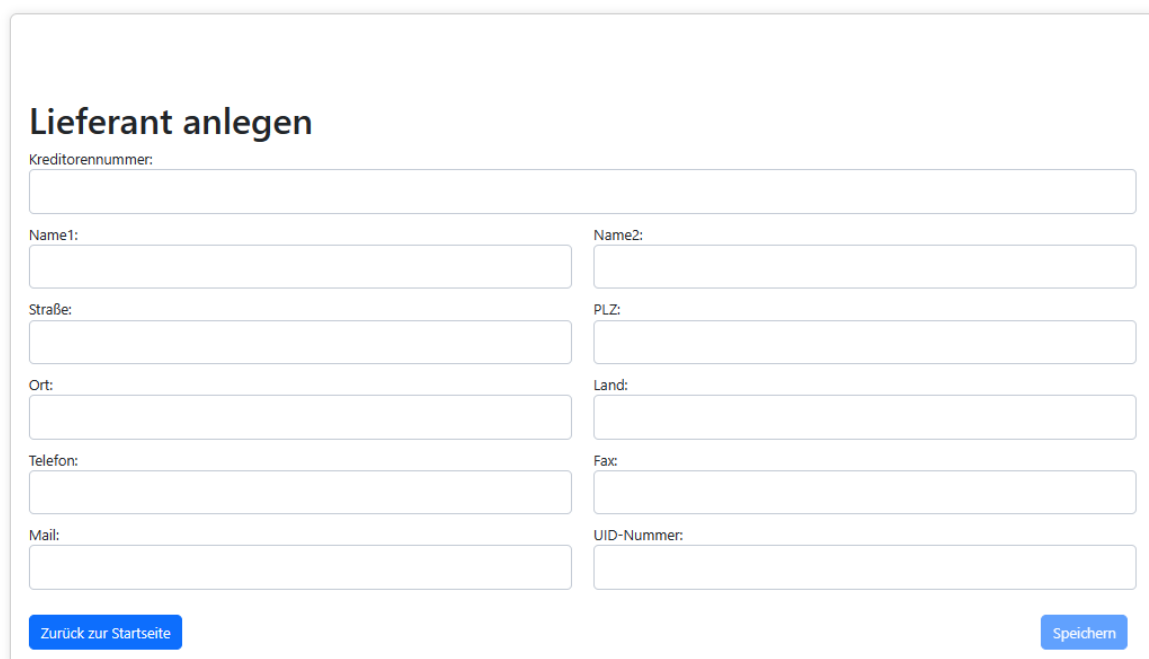


Abbildung 10: Fehlermeldung, wenn kein Produkt eingetragen wurde

4.1.6 Lieferant anlegen

Hier können die Benutzer einen neuen Lieferanten erstellen. Alle Felder auf dieser Seite sind Pflichtfelder, in die Informationen über den Lieferanten eingegeben werden müssen. Unten auf der Seite befinden sich zwei Schaltflächen: Eine ermöglicht es dem Benutzer, zur Startseite zurückzukehren, wodurch der Fortschritt verloren geht, und die andere ist die Schaltfläche "Speichern". Diese Schaltfläche ist standardmäßig nicht klickbar (ausgegraut), bis alle Felder vollständig ausgefüllt sind. Erst dann kann der Benutzer darauf klicken, um die Eingaben zu speichern.



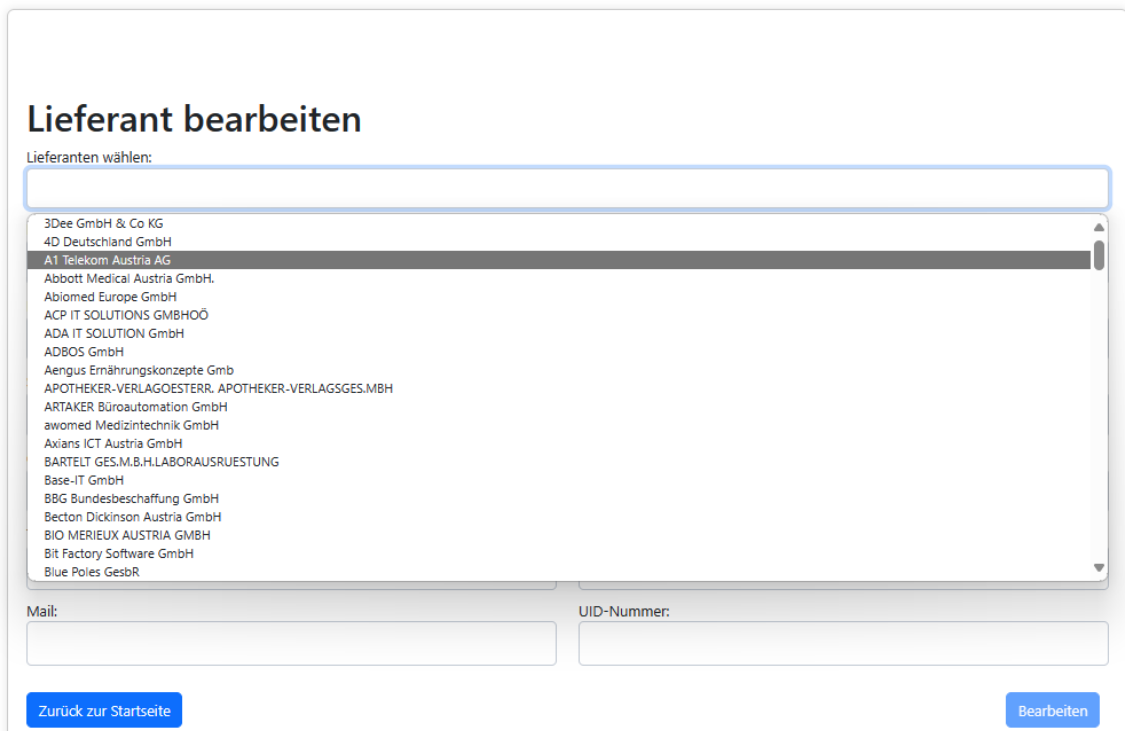
The screenshot shows a web form titled "Lieferant anlegen". It contains the following fields and buttons:

- Kreditorennummer:** A single-line text input field.
- Name1:** A single-line text input field.
- Name2:** A single-line text input field.
- Straße:** A single-line text input field.
- PLZ:** A single-line text input field.
- Ort:** A single-line text input field.
- Land:** A single-line text input field.
- Telefon:** A single-line text input field.
- Fax:** A single-line text input field.
- Mail:** A single-line text input field.
- UID-Nummer:** A single-line text input field.
- Zurück zur Startseite:** A blue button with white text.
- Speichern:** A blue button with white text, which is currently disabled (greyed out).

Abbildung 11: Lieferant anlegen

4.1.7 Lieferant bearbeiten

Hier haben Benutzer die Möglichkeit, Lieferanten zu bearbeiten. Zu Beginn befindet sich ein Dropdown-Menü, über das der gewünschte Lieferant ausgewählt werden kann. Anschließend werden alle relevanten Felder für den ausgewählten Lieferanten angezeigt und können bearbeitet werden. Am unteren Ende der Seite befinden sich zwei Schaltflächen: Eine ermöglicht es dem Benutzer, zur Startseite zurückzukehren, und die andere dient dazu, die vorgenommenen Änderungen zu speichern.



Lieferant bearbeiten

Lieferanten wählen:

- 3Dee GmbH & Co KG
- 4D Deutschland GmbH
- A1 Telekom Austria AG**
- Abbott Medical Austria GmbH.
- Abiomed Europe GmbH
- ACP IT SOLUTIONS GMBHOÖ
- ADA IT SOLUTION GmbH
- ADBOS GmbH
- Aengus Ernährungskonzepte Gmb
- APOTHEKER-VERLAGOESTERR, APOTHEKER-VERLAGSGES.MBH
- ARTAKER Büroautomation GmbH
- awomed Medizintechnik GmbH
- Axians ICT Austria GmbH
- BARTELT GES.M.B.H.LABORAUSTRUSTUNG
- Base-IT GmbH
- BBG Bundesbeschaffung GmbH
- Becton Dickinson Austria GmbH
- BIO MERIEUX AUSTRIA GMBH
- Bit Factory Software GmbH
- Blue Poles GesbR

Mail:

UID-Nummer:

[Zurück zur Startseite](#) [Bearbeiten](#)

Abbildung 12: Lieferant bearbeiten

4.1.8 Auswertungen

Auf der Seite für Auswertungen können Benutzer die Bestellungen einsehen. Es gibt drei verschiedene Ansichten: Ein normaler Mitarbeiter sieht nur seine eigenen Auswertungen, ein Vorgesetzter sieht seine eigenen Bestellungen sowie die Bestellungen der gesamten Abteilung, und das Finanzdepartement hat Zugriff auf alle Abteilungen und deren Bestellungen.

The screenshot shows the 'Auswertungen' interface. At the top, there are two date input fields labeled 'Von:' and 'Bis:', each with a calendar icon and a placeholder 'dd/mm/yyyy'. A 'Filtern' button is positioned to the right. Below this, a section titled 'Statistiken für' contains a button labeled 'Alle Bestellungen'. Underneath, there are two view selection buttons: 'Eigene Auswertungen' and 'Abteilung 1 Auswertungen', both with right-pointing arrows. The text 'Hier werden die Auswertungen angezeigt' is placed above the first button, and 'Hier werden die Auswertungen von deiner Abteilung angezeigt angezeigt' is placed above the second button.

Abbildung 13: Auswertungen als Vorgesetzter Ansicht

Zu Beginn der Seite kann der Benutzer den Zeitraum auswählen. Wenn nichts ausgewählt wird, werden alle Bestellungen angezeigt.

This screenshot shows the same 'Auswertungen' interface as in the previous image, but with a date picker open for the 'Bis:' field. The date picker is for 'März 2024' and shows a calendar grid. The date '30' is selected. The 'Von:' field contains 'TT.mm.jjjj' and the 'Bis:' field contains 'TT.mm.jjjj'. A 'Filtern' button is visible to the right. Below the date picker, there are buttons for 'Löschen' and 'Heute'. The 'Statistiken für' section shows 'Alle Bestellungen', and the view selection buttons are 'Eigene Auswertungen' and 'Abteilung 1 Auswertungen'.

Abbildung 14: Zeitraum auswählen

Danach gibt es ein Dropdown-Menü, über das nach verschiedenen Kriterien gefiltert werden kann:

- **Alle Bestellungen:** Zeigt einfach alle Bestellungen, sortiert nach dem Datum.
- **Budgetposition:** Zeigt die Bestellungen entsprechend ihrer Budgetposition an.
- **Zuordnung:** Zeigt die Bestellungen je nach Zuordnung an.
- **Krankenhaus:** Zeigt die Bestellungen je nach Krankenhaus an.
- **Betroffenes Krankenhaus:** Das Krankenhaus, dem die Bestellung zugeordnet ist.



Abbildung 15: Filter auswählen

Unten auf der Seite befinden sich Kärtchen, die beim Anklicken die jeweiligen Auswertungen öffnen.

Eigene Auswertungen

Sie haben bis jetzt insgesamt 4 Artikel bestellt.

Sie wären:

Datum	Anzahl	Komponentenbezeichnung	Nettopreis	Gesamtpreis	Betroffenes Krankenhaus	Budget	Zuordnung	Auftrag/KST oder Invest
28.02.2024	4	Druckermodell "PrintMaster 2000"	300	1200	IT-MC3	MIT-Tangente	Externe EDV Leistunge MIT	28000033

Hier werden die Auswertungen von deiner Abteilung angezeigt.

Abteilung 1 Auswertungen

Ihre Abteilung hat bis jetzt insgesamt 50 Artikel bestellt.

Sie wären:

Name-Nachname	Datum	Anzahl	Komponentenbezeichnung	Nettopreis	Gesamtpreis	Betroffenes Krankenhaus	Budget	Zuordnung
Emma Johnson	28.02.2024	10	Medizinisches Thermometer	15	150	IT-NMC	diverses	Wartungsverträge MIT
Emma Johnson	28.02.2024	3	Bürotisch "FlexDesk"	180	540	IT-MC3	KIS	Wartungsverträge MIT
Emma Johnson	28.02.2024	5	Bürostuhl "ErgoComfort"	120	600	IT-MC3	JKU sonst.	Externe EDV Leistunge MIT
Emma Johnson	28.02.2024	3	Schreibtischlampe "Lumina"	40	120	IT-MC3	Netzwerk	Externe EDV Leistunge MIT
Noah Brown	28.02.2024	4	Druckermodell "PrintMaster 2000"	300	1200	IT-MC3	MIT-Tangente	Externe EDV Leistunge MIT
James Davis	28.02.2024	20	Desinfektionsmittel (1 Liter)	8	160	IT-MC3	Infrastruktur	Externe EDV Leistunge MIT
James Davis	28.02.2024	5	Bürostuhl "ErgoComfort"	40	200	IT-NMC	MI Sonstiges	Externe EDV Leistunge MIT

Abbildung 16: Auswertungen

4.1.9 Bestell-Übersicht

In dieser Übersicht werden alle Bestellungen angezeigt, die von dem angemeldeten Benutzer aufgegeben wurden. Dies soll dem Benutzer einen einfachen Überblick über seine eigenen Bestellungen verschaffen, da bei wiederholtem Versenden von Bestellungen per E-Mail schnell die Übersicht über alle getätigten Bestellungen verloren geht.

Außerdem sollen nur die notwendigsten Informationen zur Bestellung angezeigt werden, damit der Benutzer sofort weiß, um welche Bestellung es sich handelt und was bestellt wurde.

Zum Abschluss gibt es auch eine Anzeige, die den Status einer bestimmten Bestellung anzeigt. Diese soll dem Benutzer zeigen, ob die Bestellung gerade von einem Vorgesetzten bearbeitet wird oder bereits in der Finanzabteilung liegt.



Abbildung 17: Bestell-Übersicht

Für Benutzer, die eher ein dunkles Design bevorzugen, haben wir die Möglichkeit unten rechts, das Aussehen von hell auf dunkel zu ändern. So sieht die Bestellübersicht im dunklen Design aus:

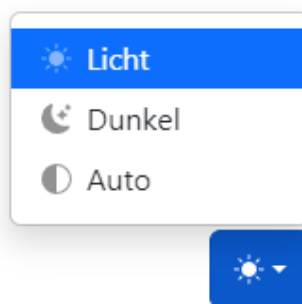


Abbildung 18: Button zum Wechseln des Designs

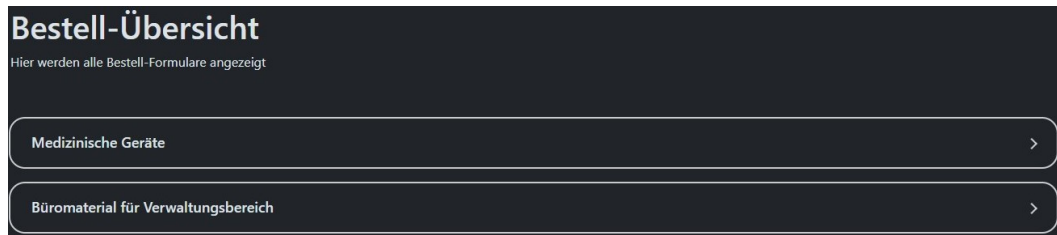


Abbildung 19: Bestell-Übersicht im dunklen Design

Um genauere Informationen über eine Bestellung zu sehen, muss man in dieser Übersicht auf eine „Bestell-Karte“ klicken. Dann erscheinen über dieser Bestellung weitere Informationen.

Medizinische Geräte
▼

Datum 28.02.2024	Besteller Emma Johnson	Abteilung Abteilung 1
Lieferant 3Dee GmbH & Co KG		
Bbg-Shop <input checked="" type="radio"/> Ja <input type="radio"/> Nein		
BvergG <input type="radio"/> Ja <input checked="" type="radio"/> Nein		
GZ-Nummer GZ		

Anzahl	Komponentenbezeichnung	Nettospreis	Gesamtpreis	Betroffenes Krankenhaus	Budget	Zuordnung	Auftrag/KST oder Invest
10	Medizinisches Thermometer	15	150	IT-NMC	diverses	Wartungsverträge MIT	28000072

Dateianhänge

Angebot.txt [Datei herunterladen](#)

Vertrag.txt [Datei herunterladen](#)

Formularfortschritt

Vorgesetzter 1
Vorgesetzter 2
Vorgesetzter 3
Finanzabteilung

Abbildung 20: Bestell-Informationen zu einer Bestellung

Welche Informationen sind gegeben?

- Ganz oben sind 3 Felder in derselben Reihe gegeben:
 - **Datum:** Hier wird das Datum angezeigt, wann die Bestellung getätigt wurde.
 - **Ersteller:** Hier wird angegeben, wer die Bestellung gemacht hat.
 - **Abteilung:** Hier wird die dazugehörige Abteilung des Benutzers angezeigt.
- Das Feld „Lieferant“ gibt an, wer der Lieferant (die Firma, bei der bestellt wird) für diese Bestellung ist.
- Der BBG-Shop (Bundesbeschaffung GmbH) ist ein Einkaufsdienstleister seit mehr als 20 Jahren. Es hat auch ein umfassendes Produkt- und Dienstleistungsportfolio. Hier wird angezeigt, ob es sich um eine BBG-Shop-Bestellung handelt oder nicht (ja/nein). [47]
- Das BVergG (Bundesvergabegesetz 2018) regelt die Vergabe von öffentlichen Aufträgen in Österreich. Genau wie oben wird angezeigt, ob es sich um eine Bekanntmachung nach dem BVergG 2018 handelt (ja/nein). [48]
- Im Feld „GZ-Nummer“ wird die GZ-Nummer der Bestellung angezeigt.
- Hier werden die bestellten Gegenstände aufgelistet. Wie im Bild zu sehen sind, gibt es 8 Spalten:
 - Anzahl
 - Komponentenbezeichnung
 - Nettopreis
 - Gesamtpreis
 - Betroffenes-Krankenhaus
 - Budget
 - Zuordnung
 - Auftrag/KST oder Invest
- Im Bereich Dateianhänge werden alle Dateien angezeigt, die der Benutzer beim Anlegen einer Bestellung hinzugefügt hat. Zu jeder Datei gibt es einen Button

„Datei herunterladen“. Mit diesem Button ist es möglich, die einzelnen Dateien herunterzuladen.

- Zum Schluss, ganz unten, ist der „Formularfortschritt“. Hier wird visualisiert, wo sich die Bestellung gerade befindet. Unten ist auch zu sehen, welche Schritte es beim Fortschritt gibt:
 - Vorgesetzter 1
 - Vorgesetzter 2
 - Vorgesetzter 3
 - Finanzabteilung

So sieht es aus, wenn sich im Formularfortschritt etwas ändert (z. B. wenn Vorgesetzter 1 akzeptiert):



Abbildung 21: Formularfortschritt wenn der Vorgesetzter 1 die Bestellung akzeptiert hat

Sobald jedoch ein weiterer Vorgesetzter die Bestellung danach ablehnt, ist der Fortschritt wieder leer.

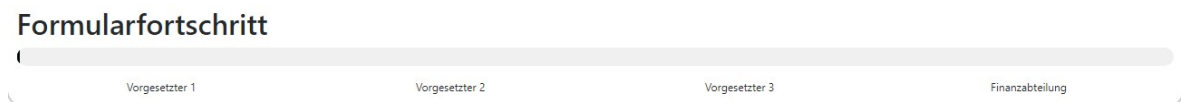


Abbildung 22: Formularfortschritt wenn ein Vorgesetzter abgelehnt hat

4.1.10 Benachrichtigungs-Seite

In dieser Ansicht werden sämtliche Fortschritte der Bestellformulare festgehalten. Das bedeutet, dass hier die Zustellung, Akzeptierung und Ablehnung zusammen mit den entsprechenden Gründen aufgeführt sind. Der Zweck dieser Seite besteht darin, einen zentralen Ort zu schaffen, an dem jeder Schritt dokumentiert wird und nicht irgendwo im E-Mail-Postfach verloren geht. Obwohl per E-Mail über den erfolgreichen Versand an den Vorgesetzten, dessen Akzeptierung oder Ablehnung informiert wird, dient dies lediglich als Hinweis darauf, dass sich ein Fortschritt im Formularprozess ergeben hat. Möchte man beispielsweise die Ablehnung genauer betrachten und Änderungen vornehmen, so kann man dies direkt auf dieser Seite vornehmen.



Abbildung 23: Benachrichtigungs-Seite (Mitarbeiter)

Wie im Bild zu sehen ist, sind 2 Benachrichtigungen zu erkennen, bei denen jeweils 2 Bestellungen an den Vorgesetzten Noah Brown gesendet wurden. Auf der rechten Seite ist zudem der genaue Zeitpunkt der Benachrichtigung ersichtlich.

Diese Seite ist unter dem Menüpunkt „Konto -> Benachrichtigungen“ erreichbar.

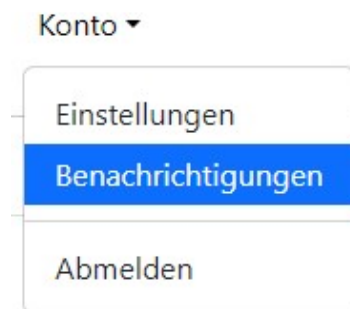


Abbildung 24: Dropdown-Menüpunkt „Konto -> Benachrichtigungen“

Arten von Benachrichtigungen

In der Website gibt es drei Arten von Benachrichtigungen, für die jeweils eine entsprechende E-Mail versendet wird:

1. Bestellzustellung und -empfang



Abbildung 25: Benachrichtigungs-Seite (Vorgesetzter)

Die erste Art der Benachrichtigungen ist die Bestellzustellung und den -empfang. In Abbildung 23 haben wir gesehen, dass 2 Bestellungen an den Vorgesetzten Noah Brown gesendet wurden. Im obigen Bild ist die Ansicht aus Sicht des Vorgesetzten zu sehen. Wie man hier auch erkennen kann, sind die beiden Bestellungen eingegangen. Bei jeder Bestellung, die der Vorgesetzte aus seiner Abteilung erhält, hat er die Möglichkeit, diese anzusehen.

Die Mail-Benachrichtigung nachdem der Mitarbeiter die Bestellung an den Vorgesetzten geschickt hat



Abbildung 26: Bestellzustellungs-Mail (Mitarbeiter)

E-Mail-Benachrichtigung zum Bestellempfang an den Vorgesetzten.



Abbildung 27: Bestellempfangs-Mail (Vorgesetzter)

2. Bestellformulargenehmigung

Eine weitere Benachrichtigung betrifft die Akzeptierung eines Bestellformulars. Nachdem ein Vorgesetzter ein Formular akzeptiert hat, wird der nächste Vorgesetzte über eine neue Nachricht im „Postfach“ der Website informiert, während der Bestellbeantragter per E-Mail benachrichtigt wird. Außerdem wird der Fortschritt in der Formularfortschrittleiste aktualisiert.

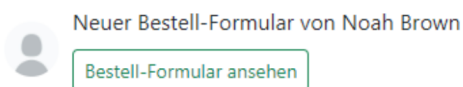


Abbildung 28: Neue Zustellungsbenachrichtigung für Vorgesetzten 2

Die Nachricht zur Bestellzustellung wird an den nächsten Vorgesetzten gesendet. Dabei wird jedoch der letzte Vorgesetzte als Zusteller angegeben. Der Grund dafür ist, dass Vorgesetzter 1 (Noah Brown) ein Mitarbeiter von Vorgesetzter 2 (James Davis) ist. Da Vorgesetzter 2 nicht weiß, wer der Mitarbeiter von Vorgesetzter 1 ist, macht es mehr Sinn, dass der Zusteller der Vorgesetzte des Mitarbeiters ist.

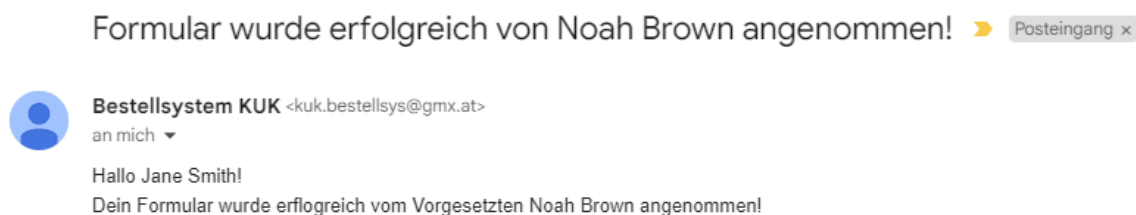


Abbildung 29: Formularfortschritt aktualisiert

Formularfortschritt



Abbildung 30: Formularfortschritt aktualisiert

3. Bestellformularablehnung

Die letzte Art der Benachrichtigung betrifft die Ablehnung von Bestellformularen. Nachdem ein Vorgesetzter (egal, ob Vorgesetzter 1, 2, 3 oder die Finanzabteilung) ein Formular abgelehnt und einen Grund dafür angegeben hat, wird der Bestellbeantrag auf der Website benachrichtigt. Außerdem wird eine E-Mail verschickt, in der angegeben ist, dass das Bestellformular abgelehnt wurde und von wem. Der Fortschritt des Bestellformulars wird zurückgesetzt.

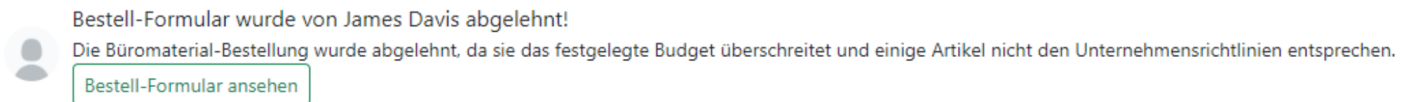


Abbildung 31: Bestellformular abgelehnt mit Grund

Zwischen dem Benachrichtigungstext und den Button „Bestell-Formular ansehen“ ist der Grund, wieso der Bestellformular abgelehnt wurde.



Abbildung 32: Bestellformular abgelehnt Benachrichtigung per E-Mail

Formularfortschritt

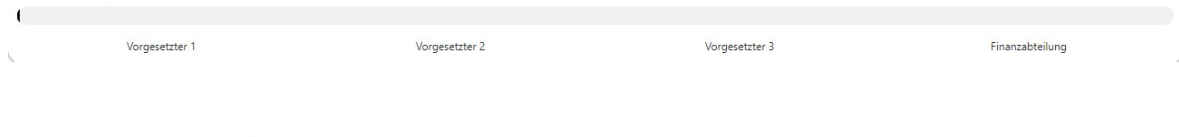


Abbildung 33: Formularfortschritt wurde zurückgesetzt

Bestellformular-Feedback

Wenn ein Mitarbeiter oder Vorgesetzter eine Bestellung bestätigt, erscheint beim nächsten Vorgesetzten eine Nachricht, wie in Abbildung 28 (Bei Vorgesetzten 3 ist der nächste „Vorgesetzte“ die Finanzabteilung).

Sobald man auf den Button „Bestell-Formular ansehen“ klickt, erscheint eine neue Seite.

Bestell-Formular Details

Datum:
22.11.2023

Besteller:
Jane Smith

Abteilung:
Abteilung 1

Lieferant
3Dee GmbH & Co KG

KNr:

BBG-Shop-Bestellung:
 Ja
 Nein

Bekanntmachung nach BVerG 2018 nötig:
 Ja
 Nein

GZ-Nummer wenn BBG-Shop-Bestellung:
GZ

Betreff:
Büromaterial

Anmerkungen:
Priorität auf umweltfreundlichen Produkten legen, Rechnung bitte elektronisch zusenden.

Abbildung 34: Bestell-Formular Details Part 1

Formale Grundlage wählen:
Angebot

Lieferbedingungen:
Lieferung innerhalb der nächsten 2 Wochen

Lieferadresse:
Holzstraße 2, 4040 Linz

Dateianhänge:
Vertrag.txt [Download File](#)
Angebot.txt [Download File](#)

ID	Anzahl	Komponentenbezeichnung	Nettopreis	Gesamtpreis	Betroffenes Krankenhaus	Budget	Zuordnung
1	3	Schreibtischlampe Lumina	40	120	IT-MC3	Netzwerk	Externe EDV Leist.

[Ablehnen](#) [Akzeptieren](#)

Abbildung 35: Bestell-Formular Details Part 2

Auf dieser Seite kann der Vorgesetzte die Bestellung genau anschauen und prüfen, ob alles passt. Nachdem er mit der Überprüfung fertig ist, hat er die Möglichkeit, unten auf der Seite eine Entscheidung zu treffen: Entweder die Bestellung abzulehnen oder zu akzeptieren.

Sollte sich der Vorgesetzte oder die Vorgesetzte dafür entscheiden, die Bestellung abzulehnen, klickt er oder sie auf den Button unten links „Ablehnen“. Dann erscheint eine Textbox, in der man den Grund für die Ablehnung eingeben kann.

[Ablehnen](#) [Akzeptieren](#)

Begründung für die Ablehnung:

[Absenden](#)

Abbildung 36: Bestellung ablehnen

Abgelehnte Bestellung

Wie bereits erwähnt, werden abgelehnte Bestellungen mit einer neuen Nachricht an den Bestellbeantragenden, zusammen mit einem Grund, warum die Bestellung abgelehnt wurde, gesendet (siehe Abbildung 31). Auch in diesem Fall gelangt man auf die Bestell-Details Seite. Allerdings sind in diesem Fall alle Felder nicht schreibgeschützt (read-only), sodass je nach Grund für die Ablehnung Änderungen an der Bestellung vorgenommen werden können.

Wenn der Bestellbeantragende die Bestellung ändern möchte, klickt er auf „Bestell-Formular ansehen“ und gelangt zur Bestellformularseite.

Bestell-Formular ansehen

Abbildung 37: Bestell-Formular ansehen Button

Dateianhänge:

Vertrag.txt [Download File](#) [Delete File](#)
 Angebot.txt [Download File](#) [Delete File](#)

ID	Anzahl	Komponentenbezeichnung	Nettospreis	Gesamtpreis	Betroffenes Krankenhaus	Budget	Zuordnung	Auftrag/KST oder Invest
1	3	Schreibtischlampe Lumina	40	120	IT-MC3	Netz	Externe EDV	2800007z

+

[Zurück zur Startseite](#) [Senden](#)

Abbildung 38: Bestellformular-Detailseite, aber die Felder sind nicht schreibgeschützt.

Wie im Bild zu sehen ist, ist auch der oder die Mitarbeiter:in in der Lage, Dateien zu löschen, falls beispielsweise das Angebot nicht passt. Zusätzlich können auch mehrere Dateien hinzugefügt werden.

Hat der oder die Bestellbeantragende die nötigen Änderungen gemacht, klickt er/sie unten rechts auf den Button „Senden“. Der gesamte Prozess beginnt dann wieder von vorne.

Budgetposition

Zu jedem bestellten Artikel muss auch die entsprechende Budgetposition angegeben werden. Dies erleichtert der Finanzabteilung die Verwaltung der Finanzen. Im Bestellformular stehen drei Spalten zur Verfügung, um die Budgetposition präzise anzugeben:

1. Budget
2. Zuordnung
3. Auftrag/KST oder Invest

Budget	Zuordnung	Auftrag/KST oder Invest
Netzwerk	Externe EDV Leistunge MIT	28000074

Abbildung 39: Die 3 Felder um die Budgetposition anzugeben

Die Felder „Budget“ und „Zuordnung“ werden mit einer Auswahlliste ausgewählt.

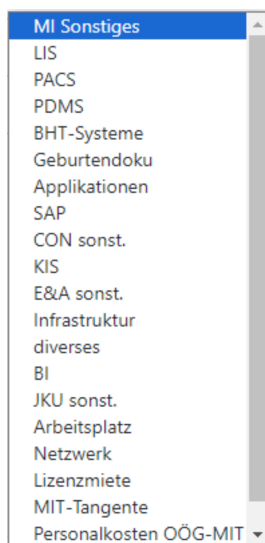


Abbildung 40: Auswahlliste für Budget

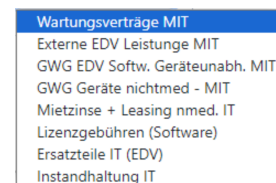


Abbildung 41: Auswahlliste für Zuordnung

Das Feld „Auftrag/KST oder Invest“ entsteht durch die Kombination von „Budget“ und „Zuordnung“. Wie in Abbildung 39 zu sehen ist, entsteht durch die Kombination von Netzwerk und Externe EDV Leistungen MIT die Nummer 28000074.

Es ist jedoch zu beachten, dass nicht alle Kombinationen eine Nummer ergeben. Zum Beispiel ergibt die Kombination JKU sonst. und Mietzinse + Leasing nmed. IT die Nummer 0.

Budget	Zuordnung	Auftrag/KST oder Invest
JKU sonst.	Mietzinse + Leasing nmed. IT	0

Abbildung 42: Auftrag/KST oder Invest Feld ist 0

Wenn wir eine andere Kombination betrachten, sehen wir, dass sich auch die entsprechende Nummer ändert.

Budget	Zuordnung	Auftrag/KST oder Invest
MI Sonstiges	Wartungsverträge MIT	28000070

Abbildung 43: Ein anderes Beispiel für eine Budgetposition.

Die Daten für die Auswahllisten und die Kombinationen wurden uns in Form einer CSV-Datei zur Verfügung gestellt. Mit dieser Datei haben wir die Informationen ausgelesen und anschließend in die Datenbank eingefügt. Sollten sich die Daten ändern, ist der einzige Weg, sie erneut in eine CSV-Datei zu exportieren und dann wieder in die Datenbank zu importieren.

4.1.11 PDF-Exportseite für Bestellungen mit Datei-Anhängen

Nachdem Benutzer Bestellungen erstellt haben, bietet die PDF-Exportseite eine fortschrittliche Funktion, um relevante Dokumente und Datei-Anhänge zu den Bestellungen hinzuzufügen. Diese Funktion ermöglicht es, wichtige Informationen und Unterlagen direkt mit der Bestellung zu verknüpfen.

Datei-Anhänge hinzufügen

Die Benutzeroberfläche der PDF-Exportseite enthält eine benutzerfreundliche Funktion zum Hinzufügen von Datei-Anhängen. Durch Klicken auf die entsprechende Schaltfläche kann der Benutzer Dateien auswählen und sie mit einer kurzen Beschreibung versehen. Dies könnte beispielsweise eine Rechnung, ein Vertrag oder ein Bericht sein, der die Bestellung begleitet.

Übersichtliche Darstellung

Jeder hinzugefügte Datei-Anhang wird übersichtlich in einer Liste dargestellt. Die Liste enthält Informationen wie den Dateinamen und die Beschreibung. Dies ermöglicht eine klare Identifizierung und Zuordnung der Dateien zu den entsprechenden Bestellungen.

Generierung der PDF-Datei

Nachdem alle erforderlichen Datei-Anhänge hinzugefügt wurden, kann der Benutzer die Option auswählen, eine PDF-Datei zu generieren. Die generierte PDF-Datei enthält die Bestelldetails sowie alle zugehörigen Datei-Anhänge. Dies erleichtert die Archivierung und den Austausch von relevanten Dokumenten im Zusammenhang mit den Bestellungen.



Abbildung 44: PDF-Exportseite mit Datei-Anhängen

4.1.12 PDF: Bestellansicht

Die PDF Bestellansicht bietet einen umfassenden Überblick über eine Bestellung und enthält alle relevanten Informationen, um eine schnelle und klare Einsicht in den Bestellprozess zu ermöglichen.

Bestellinformationen

Die Bestellansicht enthält die folgenden Informationen:

- **Datum:** Das Datum, an dem die Bestellung erstellt wurde.
- **Besteller:** Der Vorname und Nachname des Bestellers.
- **Abteilung:** Die Abteilung, zu der der Besteller gehört.

Lieferanteninformationen

Für jeden Lieferanten in der Bestellung werden die folgenden Details angezeigt:

- **Name des Lieferanten**
- **Adresse:** Straße, PLZ, Ort und Land des Lieferanten.

Bestellpositionen

Die Bestellpositionen zeigen eine detaillierte Liste der bestellten Artikel an. Jede Position umfasst:

- **Anzahl:** Die Menge des bestellten Artikels.
- **Artikelbeschreibung:** Eine klare Bezeichnung des Artikels.
- **Nettopreis:** Der Nettopreis des Artikels.



Oberösterreichische Gesundheitsholding GmbH
Medizininformatik und Informationstechnologie
Holzstraße 2, 4020 Linz
Tel.: 05 055463-21176

BESTELLUNG

Datum: 28/02/2024
Vorname und Nachname: Emma Johnson
Abteilung: Abteilung 1,
Lieferant: 3Dee GmbH & Co KG
KreditorenNr: 28536
Name1: 3Dee GmbH & Co KG
Name2:
Adresse: Margaretenstraße 101
PLZ: 1050
Ort: Wien
Land: AT
Telefonnummer: 1 7102041 10
Fax:
E-Mail: office@3dee.at
UidNr: ATU68075003
Ist BBG Shop Bestellung: Ja
Ist BVergG: Nein
Gz-Nummer: GZ
Notizen: Bitte um Lieferung bis spätestens 28.02.2024. Dringend benötigt für Patientenversorgung.
Formale Grundlage: Vertrag
Lieferadresse: Holzstraße 2, 4040 Linz
Lieferbedingungen: Standardlieferung, keine speziellen Anforderungen.
Betroffenes Krankenhaus: IT-NMC

Abbildung 45: PDF Bestellansicht

4.1.13 PDF: Bestellung Tabelle

Die PDF Bestellung Tabelle stellt eine übersichtliche Tabelle bereit, die alle relevanten Informationen zu den Bestellungen enthält, um eine einfache Referenz und Analyse zu ermöglichen.

Tabelle der Bestellpositionen

Die Tabelle enthält folgende Spalten für jede Bestellposition:

- Anzahl
- Komponentenbezeichnung
- Nettopreis
- Gesamtpreis
- Betroffenes Krankenhaus
- Budget
- Zuordnung
- Auftrag/KST oder Invest

Anzahl	Komponentenbezeichnung	Nettopreis	Gesamtpreis	Betroffenes Krankenhaus	Budget	Zuordnung	Auftrag/KST oder Invest
10	Medizinisches Thermometer	15	150	IT-NMC	diverses	Wartungsverträge MIT	28000072

Abbildung 46: PDF Bestellung Tabelle

4.1.14 PDF-Generierung für Bestellformulare

Der folgende C#-Code ist Teil unserer Webanwendung und zeigt, wie PDF-Dokumente für Bestellungen generiert werden. Die Methode `GeneratePDF` wird aufgerufen, um basierend auf einer spezifischen Bestell-ID zwei PDF-Dokumente zu erstellen und diese in einem ZIP-Archiv zu komprimieren.

```

1      [HttpPost("GeneratePDF")]
2      public async Task<IActionResult> GeneratePDF(int id)
3      {
4          OrderFormular? orderFormular = await
5              _unitOfWork.OrderFormularRepository.GetFormularById(id);
6          if (orderFormular == null)
7          {
8              return BadRequest("OrderFormular data not provided.");
9          }
10         try
11         {
12             using (var zip = new Ionic.Zip.ZipFile())
13             {
14                 // Create the first PDF document (portrait)
15                 using (var memoryStream = new MemoryStream())
16                 {
17                     var pdfDocument = new PdfDocument(new PdfWriter(memoryStream));
18                     var document = new iText.Layout.Document(pdfDocument,
19                         PageSize.A4);
20                     pdfDocument.AddEventHandler(PdfDocumentEvent.END_PAGE, new
21                         PageEventHandler($"{orderFormular.Subject}.pdf"));
22
23                     var imagePath = UserHelper.GetImageInTheTop();
24                     var image = new Image(ImageDataFactory.Create(imagePath))
25                         .ScaleToFit(200, 200)
26                         .SetFixedPosition(20, PageSize.A4.GetTop() - 90)
27                         .SetTextAlignment(TextAlignment.LEFT);
28                     document.Add(image);
29
30                     // Additional content based on OrderFormular data
31                     // Code continues...
32                 }
33                 // Create the second PDF document (landscape) and add to zip
34                 // Code continues...
35                 // Save the ZIP archive to a memory stream and return to client
36                 // Code continues...
37             }
38         }
39         catch (Exception ex)
40         {
41             return StatusCode(500, $"An error occurred: {ex.Message}");
42         }

```

Die Methode beginnt mit der Abfrage des Bestellformulars anhand seiner ID. Bei erfolgreicher Abfrage werden zwei PDF-Dokumente erstellt: eines im Hochformat mit allgemeinen Informationen zur Bestellung und ein weiteres im Querformat, das eine detaillierte Artikelliste enthält. Beide Dokumente werden anschließend in einem ZIP-Archiv zusammengefasst und zur Verfügung gestellt.

4.1.15 Einstellungen

Im Einstellungsabschnitt sind fünf Schlüsselbereiche verfügbar, die es dem Active Directory Benutzer ermöglichen, das Nutzererlebnis und die Sicherheit zu optimieren.

Profil

Im Profilbereich wird besonderer Wert auf die Kontrolle und Sicherheit der Benutzerdaten gelegt. Der Benutzer kann sein Profilbild nach Belieben ändern, um eine persönliche Note hinzuzufügen. Allerdings wurden bewusst Einschränkungen für bestimmte Felder wie Vorname, Nachname, E-Mail-Adresse, Abteilung und Manager eingeführt. Diese Felder können nicht direkt vom Benutzer bearbeitet werden, um die Integrität und Konsistenz der Benutzerdaten im Active Directory zu wahren.

Die strikten Einschränkungen dieser Schlüsselinformationen gewährleisten, dass grundlegende Identifikationsmerkmale stabil bleiben und nicht unbeabsichtigt verändert werden. Dies dient nicht nur der Sicherheit, sondern auch der Vermeidung von Fehlern in den organisatorischen Strukturen.

Neben diesen Einschränkungen gibt es jedoch eine Reihe optionaler Informationen, die der Benutzer nach Bedarf aktualisieren kann. Dazu gehören Telefonnummer, Geburtstag, Adresse, Postleitzahl, Stadt und Land. Diese optionalen Anpassungsmöglichkeiten bieten dem Benutzer eine gewisse Flexibilität, um persönliche Details zu pflegen und sicherzustellen, dass die Informationen stets aktuell sind.

Die ausgewogene Kombination aus strengen Kontrollen und optionalen Anpassungsmöglichkeiten zielt darauf ab, sowohl die Sicherheit als auch die individuelle Anpassungsfähigkeit der Benutzerprofile zu gewährleisten. Dies ermöglicht eine effektive Verwaltung der Benutzerdaten im Rahmen des Active Directory.

Abbildung 47: Profil-Einstellungen im Active Directory

Sicherheit

Im Sicherheitsbereich stehen dem Benutzer zwei wichtige Funktionen zur Verfügung, um die Sicherheit seines Kontos zu gewährleisten.

Zwei-Faktor-Authentifizierung (2FA)

Die Zwei-Faktor-Authentifizierung (2FA) ist eine wirksame Methode, um die Kontosicherheit zu erhöhen. Der Benutzer hat die Möglichkeit, die 2FA für sein Konto zu aktivieren oder zu deaktivieren. Bei der Aktivierung wird die Eingabe der E-Mail-Adresse des Benutzers erforderlich, um sicherzustellen, dass der Authentifizierungsprozess mit einer verifizierten Adresse verknüpft ist. Durch diesen zusätzlichen Sicherheitsschritt wird die Kontozugänglichkeit weiter geschützt.

```

1      [HttpPost("GenerateSixDigitCode")]
2      [ProducesResponseType(typeof(User), StatusCodes.Status200OK)]
3      [ProducesResponseType(typeof(User), StatusCodes.Status404NotFound)]
4      [ProducesResponseType(typeof(User), StatusCodes.Status400BadRequest)]
5      [ProducesResponseType(StatusCodes.Status500InternalServerError)]
6      public async Task<IActionResult> GenerateSixDigitCode([FromBody]
7          UserGenerateSixDigitCode userDto)
8      {
9          try
10         {
11             User? user = await
12                 _unitOfWork.UserRepository.GetUserByEmailAddress(userDto.Email);
13             if (user == null)
14             {
15                 return NotFound(); // User not found
16             }
17             if (user.CodeExpiration <= DateTime.Now)
18             {
19                 // Generate a new code using the UserHelper class
20                 string code = UserHelper.GenerateCode();

```

```

21         // Update the code in the user object
22         user.RowVersion = userDto.RowVersion;
23         user.Id = userDto.Id;
24         user.Code = code;
25         user.CodeExpiration = DateTime.Now.AddMinutes(5);
26         _unitOfWork.UserRepository.UpdateUser(user);
27         await _unitOfWork.SaveChangesAsync();
28
29         return Ok(user);
30     }
31     else
32     {
33         return BadRequest("Previous code has not expired");
34     }
35 }
36 catch (Exception ex)
37 {
38     return StatusCode(500, $"An error occurred: {ex.Message}");
39 }
40 }

```

Die Methode `GenerateSixDigitCode` ist ein HTTP-PUT-Endpoint, der für die Generierung und Zuweisung eines neuen sechsstelligen Codes zu einem Benutzerkonto zuständig ist. Der Prozess umfasst mehrere Schritte:

Benutzersuche: Zunächst wird versucht, den Benutzer anhand der im Anforderungskörper ([FromBody]) übergebenen E-Mail-Adresse zu finden. Wenn kein entsprechender Benutzer gefunden wird, antwortet der Endpoint mit einem 404-Statuscode, um anzuzeigen, dass der Benutzer nicht gefunden wurde.

Code-Generierung und -Aktualisierung: Wenn der aktuelle Code des Benutzers abgelaufen ist (d.h., `user.CodeExpiration` liegt in der Vergangenheit), generiert die `UserHelper.GenerateCode`-Methode einen neuen sechsstelligen Code. Dieser Code, zusammen mit einem neuen Ablaufdatum (5 Minuten in der Zukunft), wird dann im Benutzerobjekt aktualisiert.

Datenbankaktualisierung: Die aktualisierten Benutzerdaten werden in der Datenbank gespeichert. Bei erfolgreicher Aktualisierung sendet der Endpoint eine HTTP 200-Antwort mit den aktualisierten Benutzerdaten zurück.

```

1     public static string GenerateCode()
2     {
3         Random random = new Random();
4         int code = random.Next(100000, 999999);
5
6         return code.ToString();
7     }

```

Die `GenerateCode`-Methode erzeugt einen zufälligen sechsstelligen Code. Dies geschieht durch die Auswahl einer zufälligen Zahl zwischen 100.000 und 999.999. Der generierte Code dient als der zweite Faktor in der 2FA-Überprüfung, der sicherstellt, dass der Benutzer tatsächlich Zugriff auf die mit dem Konto verknüpfte E-Mail-Adresse hat.

Passwort ändern

Das Ändern des Passworts ist eine grundlegende Sicherheitsmaßnahme. Der Benutzer kann sein Passwort nach Bedarf ändern. Es ist zu beachten, dass externe Benutzer Einsicht in den Änderungsprozess haben, während interne Benutzer diesen Vorgang im Active Directory durchführen können. Dieser Unterschied wurde implementiert, um sicherzustellen, dass interne Benutzer eine erhöhte Sicherheitskontrolle über ihre Konten haben.

The image shows two side-by-side web forms. The left form is titled 'Passwort ändern' (Change Password) and contains three input fields: 'Aktuelles Passwort' (Current Password) with the placeholder 'Bitte geben Sie Ihr aktuelles Passwort ein', 'Neues Passwort' (New Password) with the placeholder 'Bitte geben Sie ein neues Passwort ein', and 'Passwort bestätigen' (Confirm Password) with the placeholder 'Bitte bestätigen Sie das Passwort'. At the bottom are two buttons: 'Änderungen speichern' (Save Changes) in blue and 'Abbrechen' (Cancel) in red. The right form is titled 'Zwei-Faktoren-Authentifizierung' (Two-Factor Authentication). It contains a text block explaining that 2FA increases account security by adding an extra authentication level, and that a 6-digit number will be sent to the user's email address. Below this are two radio buttons: 'Ein' (On) and 'Aus' (Off), with 'Aus' selected. There is also an 'E-Mail Adresse' (Email Address) field containing 'karlof04gamer@gmail.com'. At the bottom are two buttons: 'Änderungen speichern' (Save Changes) in blue and 'Abbrechen' (Cancel) in red.

Abbildung 48: Sicherheitseinstellungen

Benutzer erstellen

Der interne Benutzer hat die Befugnis, einen neuen externen Benutzer im System anzulegen. Bei der Erstellung eines neuen Benutzers sind verschiedene Informationen erforderlich:

- Vorname
- Nachname
- Abteilungszugehörigkeit
- Manager (Vorgesetzter des neuen Benutzers)
- E-Mail-Adresse (für die Anmeldung)
- Passwort und Passwortbestätigung

Rollenzuweisung

Zusätzlich zur grundlegenden Benutzererstellung hat der interne Benutzer die Möglichkeit, spezifische Rollen für den neuen externen Benutzer festzulegen. Diese Rollen umfassen:

- Bestellen
- Statistik ansehen
- Lieferanten erstellen

Einen neuen Benutzer anlegen

Vorname	Nachname
<input type="text"/>	<input type="text"/>
Abteilung	
<input type="text"/>	
Manager	
<input type="text"/>	
E-Mail Adresse	
<input type="text"/>	
Neues Passwort	Passwort bestätigen
<input type="password"/>	<input type="password"/>

Rollen verwalten

Bestellen (Rolle)
 Statistik ansehen (Rolle)
 Lieferant erstellen (Rolle)

Abbildung 49: Benutzererstellung im internen Bereich

Benutzer-Management

Der interne Benutzer, sofern er die entsprechenden Rollen innehat, verfügt über umfassende Funktionen im Bereich des Benutzermanagements.

Aktivieren / Deaktivieren von Benutzern

Der interne Benutzer hat die Befugnis, sowohl interne als auch externe Benutzer zu aktivieren oder zu deaktivieren. Die Deaktivierung eines Benutzers bedeutet, dass dieser sich nicht mehr anmelden kann. Diese Funktion ermöglicht eine schnelle und präzise Steuerung der Benutzerzugriffe im System.

Bearbeiten von Benutzern

Zusätzlich zur Aktivierung und Deaktivierung kann der interne Benutzer auch Benutzer bearbeiten. Dies umfasst die Möglichkeit, sowohl interne als auch externe Benutzerdaten zu aktualisieren. Der Bearbeitungsvorgang ermöglicht es, Änderungen an Benutzernamen, E-Mail-Adressen, Abteilungszuordnungen und anderen relevanten Informationen vorzunehmen.

Diese umfangreichen Funktionen im Benutzermanagement gewährleisten eine effektive Verwaltung der Benutzer im System und ermöglichen es dem internen Benutzer, flexibel auf Änderungen und Anpassungen zu reagieren.

Interne User				
Vorname	Nachname	Abteilung	E-Mail	Aktionen
Emma	Johnson	Abteilung 1	e.johnson@bestellsystem.local	Aktivieren Deaktivieren
Noah	Brown	Abteilung 1	n.brown@bestellsystem.local	Aktivieren Deaktivieren
James	Davis	Abteilung 1	j.davis@bestellsystem.local	Aktivieren Deaktivieren
Benjamin	Clark	Abteilung 1	b.clark@bestellsystem.local	Aktivieren Deaktivieren

Externe User				
Vorname	Nachname	Abteilung	E-Mail	Aktionen
Abdalahman	EL-hamaoui	Abteilung 1	karlof04gamer@gmail.com	Aktivieren Deaktivieren Bearbeiten

Abbildung 50: Benutzermanagement im internen Bereich

Änderungen von Benutzerdaten

Der interne Benutzer kann die folgenden Informationen eines externen Benutzers bearbeiten:

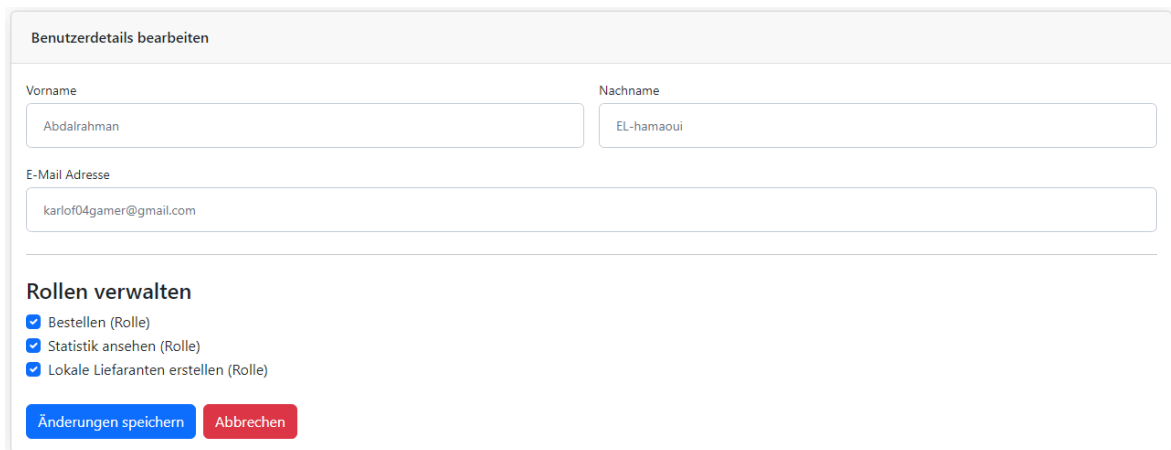
- Vorname & Nachname
- E-Mail-Adresse

Diese Funktion ermöglicht es dem internen Benutzer, wichtige Identifikationsmerkmale von externen Benutzern zu aktualisieren und sicherzustellen, dass die Benutzerdaten stets aktuell sind.

Bearbeitung von Rollen

Zusätzlich zu den grundlegenden Benutzerdaten hat der interne Benutzer auch die Möglichkeit, die Rollen eines externen Benutzers zu bearbeiten. Diese Rollen umfassen:

- Bestellen
- Statistik ansehen & Lokale Lieferanten erstellen



Benutzerdetails bearbeiten

Vorname: Abdalrahman

Nachname: EL-hamaoui

E-Mail Adresse: karlof04gamer@gmail.com

Rollen verwalten

- Bestellen (Rolle)
- Statistik ansehen (Rolle)
- Lokale Lieferanten erstellen (Rolle)

Änderungen speichern Abbrechen

Abbildung 51: Bearbeitung externer Benutzer im internen Bereich

E-Mail-Benachrichtigungen

Der Benutzer hat die Möglichkeit, E-Mail-Benachrichtigungen nach seinen Präferenzen zu konfigurieren.

Aktivieren / Deaktivieren

Der Benutzer kann die E-Mail-Benachrichtigungen nach Bedarf aktivieren oder deaktivieren. Diese Funktion ermöglicht es dem Benutzer, die Kontrolle darüber zu behalten, welche Benachrichtigungen er erhalten möchte.

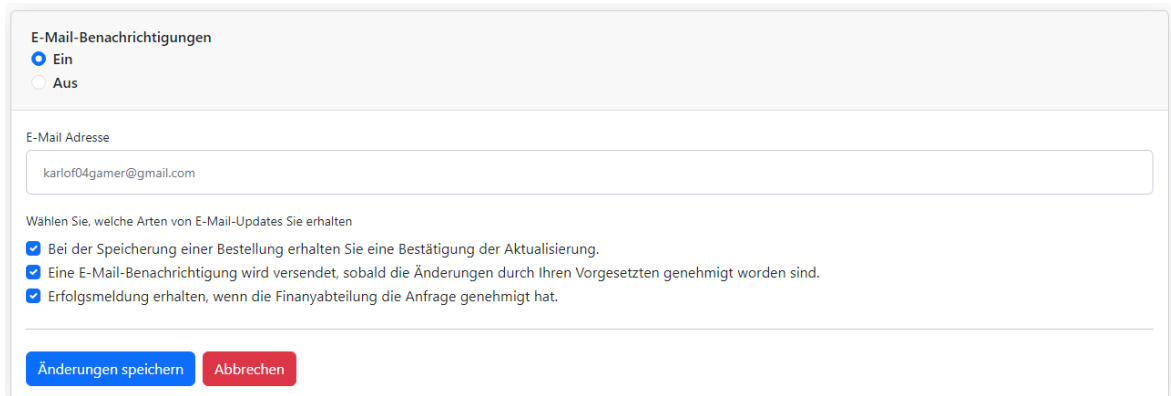
E-Mail-Adresse eingeben

Um die E-Mail-Benachrichtigungen zu personalisieren, kann der Benutzer seine E-Mail-Adresse eingeben. Dadurch werden alle relevanten Benachrichtigungen an die angegebene Adresse gesendet.

Benachrichtigungsarten auswählen

Der Benutzer kann aus verschiedenen Arten von Benachrichtigungen auswählen und festlegen, welche er erhalten möchte. Die verfügbaren Optionen umfassen:

- Bei der Speicherung einer Bestellung erhalten Sie eine Bestätigung der Aktualisierung.
- Eine E-Mail-Benachrichtigung wird versendet, sobald die Änderungen durch Ihren Vorgesetzten genehmigt worden sind.
- Erfolgsmeldung erhalten, wenn die Finanyabteilung die Anfrage genehmigt hat.



The screenshot shows a web form titled "E-Mail-Benachrichtigungen". At the top, there are two radio buttons: "Ein" (selected) and "Aus". Below this is a text input field labeled "E-Mail Adresse" containing the email address "karlof04gamer@gmail.com". Underneath, there is a section titled "Wählen Sie, welche Arten von E-Mail-Updates Sie erhalten" with three checked checkboxes: "Bei der Speicherung einer Bestellung erhalten Sie eine Bestätigung der Aktualisierung.", "Eine E-Mail-Benachrichtigung wird versendet, sobald die Änderungen durch Ihren Vorgesetzten genehmigt worden sind.", and "Erfolgsmeldung erhalten, wenn die Finanyabteilung die Anfrage genehmigt hat." At the bottom of the form, there are two buttons: "Änderungen speichern" (blue) and "Abbrechen" (red).

Abbildung 52: E-Mail-Benachrichtigungseinstellungen

4.2 Frontend

4.2.1 REST Implementierung

Frontend-Entwicklung in Webanwendungen umfasst die Erstellung der Benutzeroberfläche, mit der Benutzer interagieren, sowie die Integration dieser Oberfläche mit dem Backend, um Daten und Funktionalitäten bereitzustellen. Ein wichtiger Aspekt der Frontend-Entwicklung ist die Implementierung von REST (Representational State Transfer) in Angular, einem beliebten JavaScript-Framework zur Entwicklung von Single-Page-Anwendungen. REST ist ein Architekturstil für die Kommunikation zwischen Client und Server über das HTTP-Protokoll und wird oft verwendet, um Daten zwischen Frontend und Backend auszutauschen. In diesem Kontext werden wir uns die Implementierung von REST in Angular genauer ansehen und wie dies zur Entwicklung moderner Webanwendungen beiträgt.

Service

Um mit der Implementierung zu beginnen, erstellen wir zunächst einen Service mithilfe des folgenden Befehls: `ng g s services/deliverrant`. Ein Angular-Service ist eine Klasse, die in einer Angular-Anwendung spezifische Funktionen bereitstellt und häufig für die Kommunikation zwischen Frontend und Backend verwendet wird. Diese Services sind typischerweise Singleton-Objekte, was bedeutet, dass sie innerhalb der gesamten Anwendung nur einmal erstellt werden und von verschiedenen Komponenten geteilt werden können.

In unserem aktuellen Kontext erstellen wir einen Service mit dem Namen `DeliverrantService`. Dieser Service ist verantwortlich für die Kommunikation mit dem Backend, um Daten über Lieferanten abzurufen und zu verarbeiten.

Der Code des Services sieht wie folgt aus:

```
1 import { HttpClient, HttpHeaders } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3 import { Deliverrant } from '../models/deliverrant';
4
5 const httpOptions = {
6   headers: new HttpHeaders({
7     'Content-Type': 'application/json'
8   })
9 }
10
11 @Injectable({
12   providedIn: 'root'
```

```
13     })
14     export class DeliverrantService {
15
16         apiUrl = "https://localhost:5001/api/Deliverrant";
17
18         constructor(private http: HttpClient) { }
19
20         GetAllDeliverrants(){
21             return this.http.get<Deliverrant []>(this.apiUrl, httpOptions);
22         }
23     }
```

In diesem Service wird eine Methode `GetAllDeliverrants` definiert, die eine HTTP-GET-Anfrage an die API-URL (`apiURL`) sendet, um alle Lieferanten abzurufen. Das `HttpClient`-Modul wird verwendet, um die HTTP-Anfragen zu verarbeiten, und `httpOptions` werden definiert, um den Content-Type der Anfrage festzulegen.

Die Methode `GetAllDeliverrants` gibt ein `Observable` zurück, das die Liste der Lieferanten enthält.

Komponente

Die `DeliverrantFormComponent` ist eine Angular-Komponente, die für die Anzeige und Bearbeitung von Lieferantendaten verantwortlich ist. In diesem Fall wird der `DeliverrantService` in die `DeliverrantFormComponent` injiziert, indem er im Konstruktor der Komponente aufgerufen wird. Dadurch kann die Komponente auf die Methoden und Daten des Services zugreifen.

```
1
2         constructor(private deliverrantService: DeliverrantService){
3         }
```

Um die `DeliverrantFormComponent` zu erstellen, haben wir den folgenden Befehl verwendet: `ng g c components/deliverantForm`. Dadurch wird automatisch eine Angular-Komponente mit dem Namen `DeliverantFormComponent` im Ordner `components` erstellt.

```
1
2   ngOnInit(): void {
3       this.getAllDeliverrants();
4
5   }
6
7
8   getAllDeliverrants() {
9       this.deliverrantService
10          .GetAllDeliverrants()
11          .subscribe({
12              next: (data) => {
13                  this.deliverrants = data;
14                  console.log(this.deliverrants.length)
15              },
16              error: (error) => {
17                  alert('Liferanten konnten nicht geladen werden!');
18                  this.loading = false;
19              },
20          });
21    }
```

Die Methode `ngOnInit` ist ein Lifecycle-Hook in Angular, der automatisch aufgerufen wird, sobald die Komponente initialisiert wird. Hier wird die Methode `getAllDeliverrants` aufgerufen, um alle Lieferanten beim Laden der Komponente abzurufen.

Die Methode `getAllDeliverrants` ist für das Abrufen von Lieferantendaten verantwortlich. Es verwendet den `deliverrantService`, der ein Service in der Angular-Anwendung ist, um Daten über Lieferanten vom Server abzurufen. Die Methode `GetAllDeliverrants` des `deliverrantService` wird aufgerufen, um die Daten abzurufen.

Die Methode `subscribe` wird verwendet, um auf die Daten zu warten, die von der API zurückgegeben werden. Wenn die Daten erfolgreich abgerufen werden, wird die Funktion im `next`-Block ausgeführt, die die Daten in die Variable `deliverrants` speichert. Wenn ein Fehler auftritt, wird die Funktion im `error`-Block ausgeführt, die eine Benachrichtigung anzeigt, dass die Lieferanten nicht geladen werden konnten.

Zusammengefasst

Der Service fungiert als Bindeglied zwischen der Komponente und dem Backend, indem er Daten bereitstellt und die Kommunikation zwischen Frontend und Backend koordiniert.

4.3 Backend

4.3.1 Technologieentscheidungen für das Backend

Bei der Projektplanung haben wir uns für das Backend für eine MySQL-Datenbank und ASP.NET entschieden. Zusätzlich verwenden wir das Unit-of-Work-Pattern, da wir mit diesen Mustern und Technologien sehr vertraut sind und uns deshalb dafür entschieden haben.

4.3.2 Aufbau des Backends

Das Backend wurde in unterschiedliche Teile aufgeteilt. Hier sind die Aufteilungen:

- **Core:** Hier sind zwei Ordner: Die Entitäten wie z.B. User, OrderFormular, und die Contracts. Dort sind die Schnittstellen zu jeder Entität definiert.

Wir haben 19 Entitäten, von denen jede eine Rolle spielt:

- **Admin:** Der Admin ist in der Lage, die Rechte über die Gruppen (Mitarbeiter, Vorgesetzter 1, 2, 3 und Finanzabteilung) zu verwalten, zum Beispiel das Anlegen von Bestellungen.
- **AffectedHospital:** Wird beim Anlegen einer Bestellung benötigt: Man muss angeben, für welches Gebäude die Bestellung ist.
- **Article:** Ein Artikel mit folgenden Eigenschaften: Anzahl, Bezeichnung, Einzelpreis netto und hat eine Beziehung zu den Entitäten OrderFormular und BudgetPosition_Assignment.
- **Assignment:** Die Zuweisung, die ebenfalls beim Anlegen einer Bestellung benötigt wird. Siehe Abbildung 41.
- **BudgetPosition:** BudgetPosition, genau wie Assignment, wird auch für die Anlegung einer Bestellung benötigt. Siehe Abbildung 40.
- **BudgetPosition_Assignment:** Ist die Kombination von BudgetPosition und Assignment. Siehe Abbildung 43.
- **Deliverrant** Der Lieferant der Bestellung, mit wichtigen Eigenschaften wie Name, Adresse, E-Mail, Fax und Liefernummer.

- **Delivery** Die Lieferung, also wo die Bestellung geliefert werden soll und welche Lieferbedingungen die Bestellung haben soll.
- **Department:** Die Abteilung eines Mitarbeiters, wo der Name und der Ort der Abteilung gespeichert sind.
- **EntityObject:** Das ist die Basis aller Entitäten. Alle Entitäten erben von dieser Klasse. Enthält Felder wie die ID und die RowVersion.
- **FileAttachment:** Die angehängten Dateien, die bei einer Bestellung mitgegeben werden.
- **Group:** Jeder Mitarbeiter hat eine Gruppe. Gruppe 1 = User -> Mitarbeiter, Gruppe 2-4 = Vorgesetzter 1-3 und Gruppe 5 = Finanzabteilung.
- **Inbox:** Hier werden alle Benachrichtigungen gespeichert. Wichtige Felder sind Empfänger, Sender, Art der Bestellung und das Bestellformular.
- **Manager:** Alle Vorgesetzten. Diese Entität wird von der Entität Inbox verwendet.
- **OrderFormular:** Das Bestellformular, die größte Entität von allen. Besteht hauptsächlich aus Beziehungen zu anderen Entitäten wie User, Department, Deliverrant, Delivery, AffectedHospital. Hat aber auch Eigenschaften wie den Titel der Bestellung, Anmerkungen usw. Auch wichtig ist der Fortschritt des Formulars. Hier gibt es 4 Felder für die Vorgesetzten 1-3 und die Finanzabteilung. Wenn der Vorgesetzte 1 die Bestellung genehmigt, wird die Spalte auf 25 gesetzt, was 25% Fortschritt bedeutet. Es geht immer in 25-Schritten, d.h. (25, 50, 75 und 100).
- **Role:** Hier werden sozusagen den Gruppen jede Berechtigungsrolle zugewiesen, wie zum Beispiel Gruppe 1 (Mitarbeiter) werden der Berechtigungsgruppe „Bestellen“ zugewiesen. Das bedeutet, alle Mitarbeiter können eine Bestellung aufgeben.
- **User:** Das ist der Benutzer im System. Die Entität hat alle notwendigen Eigenschaften, die ein Benutzer hat, wie Vor- und Nachname, E-Mail, Passwort, Abteilung usw.
- **UserRole:** Diese Entität wurde für externe Benutzer erstellt, also für Benutzer, die nicht im Active Directory sind. Diese Benutzer benötigen ebenfalls

Rollen bzw. Rechte, wie zum Beispiel „Statistik ansehen“. Deshalb werden hier die Rollen von externen Benutzern verwaltet.

- **Persistence:** Hier verfolgt der Migrationsordner alle Änderungen am Datenbankschema und generiert entsprechende SQL-Skripte zur Anwendung dieser Änderungen. Zusätzlich enthält er die jeweiligen Repositories für jede Entität sowie das **UnitOfWork**.

Zusätzlich gibt es eine Konfigurationsdatei namens „appsettings.json“.

```
1 {
2     "ConnectionStrings": {
3         "DefaultConnection": "Server=localhost;
4             Port=3306;Database=BestellsystemKUK;
5             Uid=Bestellsystem;Pwd=bestellsystem;"
6     },
7     "Logging": {
8         "LogLevel": {
9             "Default": "Warning"
10        }
11    },
12    "AllowedHosts": "*"
13 }
```

- Beim „ConnectionStrings“ wird die Verbindung zur Datenbank definiert, die in diesem Fall auf eine MySQL-Datenbank verweist und Informationen wie Server, Port, Datenbankname, Benutzername und Passwort enthält.
- Im Logging wird das LoggingLevel definiert. In diesem Fall wurde es auf „Warning“ gesetzt, was bedeutet, dass nur Meldungen mit dem Logging-Level „Warning“ oder höher protokolliert werden.
- Und zum Schluss beim „AllowedHosts“ werden die Hosts definiert. Hier wurde es mit „*“ definiert, was bedeutet, dass Anfragen von allen Hosts akzeptiert werden.

- **Utils:** Hier sind die „Hilfsklassen“, die wir im Projekt verwendet haben, wie zum Beispiel:

- **EmailSender:** Weitere Details dazu auf Seite 108.
- **MyFile:** Hier sind Methoden enthalten, die nach Dateien mit einem bestimmten Namen suchen, beginnend im Arbeitsverzeichnis der Anwendung bis zum Wurzelverzeichnis des Laufwerks, und den vollständigen Pfad zurückgeben.

```

1 public static string? GetFullNameInApplicationTree(string fileName)
2 {
3     if (string.IsNullOrEmpty(fileName)) return null;
4     string path = Environment.CurrentDirectory;
5     // string path = AppDomain.CurrentDomain.BaseDirectory;
6     while (path != Directory.GetDirectoryRoot(path) &&
7         Directory.GetFiles(path, fileName).Length == 0)
8     {
9         path = Directory.GetParent(path)?.FullName ?? string.Empty;
10    }
11    if (Directory.GetFiles(path, fileName).Length > 0) // Datei
12        existiert
13    {
14        string fullName = Path.Combine(path, fileName);
15        return fullName;
16    }
17    return null;
18 }

```

Außerdem enthält sie eine Methode zum Einlesen von CSV-Dateien in ein zweidimensionales String-Array.

```

1 public static async Task<string[][]>
2     ReadStringMatrixFromCsvAsync(string fileName, bool
3     overreadTitleLine, char delimiter = ';')
4 {
5     int startLine = 0;
6     int subtractIndex = 0;
7     string? fullFileName = GetFullNameInApplicationTree(fileName); //
8     csv-Datei liegt im Projektverzeichnis
9     if (fullFileName == null)
10    {
11        throw new FileNotFoundException("File " + fileName + " not
12        found in applicationpath");
13    }
14    string[] lines = await File.ReadAllLinesAsync(fullFileName,
15    Encoding.UTF8);
16    int lineCount = lines.Length;
17    if (overreadTitleLine)
18    {
19        lineCount--;
20        startLine = 1;
21        subtractIndex = 1;
22    }
23    string[][] elements = new String[lineCount][];
24    for (int line = startLine; line < lines.Length; line++)
25    {
26        elements[line - subtractIndex] = lines[line].Split(delimiter);
27    }
28    return elements;
29 }

```

- **PageEventHandler:** Diese Klasse wurde für den PDF-Export verwendet.

- **UserHelper:** Sie enthält Methoden wie die Generierung einer zufälligen 6-stelligen Zahl,

```

1 public static string GenerateCode()
2 {
3
4     Random random = new Random();
5     int code = random.Next(100000, 999999);
6
7     return code.ToString();
8 }

```

die Validierung von Passwörtern und E-Mails

```

1 public static bool IsValidEmail(string email)
2 {
3     try
4     {
5         var mailAddress = new MailAddress(email);
6         return true;
7     }
8     catch (FormatException)
9     {
10        return false;
11    }
12 }
13
14 public static bool IsValidPassword(string password)
15 {
16
17     // At least 8 characters, one uppercase letter, one lowercase
18     // letter, one number, and one special character
19     string passwordPattern =
20         @"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[^\da-zA-Z]).{8,}$";
21
22     return Regex.IsMatch(password, passwordPattern);
23 }

```

sowie eine Methode zum Laden eines bestimmten Bildes aus dem Projektverzeichnis. Dieses Bild wurde ebenfalls für den PDF-Export verwendet.

```

1 public static byte[] GetImageInTheTop()
2 {
3     // Get the base directory of the current application domain (usually the
4     // bin directory)
5     string baseDirectory = AppDomain.CurrentDomain.BaseDirectory;
6
7     // Construct the path to the Pictures directory
8     string picturesDirectory = Path.Combine(baseDirectory, "Pictures");
9
10    // Construct the full path to the image file
11    string imagePath = Path.Combine(picturesDirectory, "Image_Pfp.jpg");
12
13    // Check if the file exists
14    if (File.Exists(imagePath))
15    {
16        // Read the image file into a byte array
17        byte[] imageBytes = File.ReadAllBytes(imagePath);
18
19        return imageBytes;
20    }
21    else
22    {
23        // Handle the case where the file does not exist
24        throw new FileNotFoundException("Image file not found.");
25    }
26 }

```

- **WebApi:**

Der letzte Teil des Backends ist die WebApi. Hier können verschiedene Clients wie z.B. Webbrowser mittels REST und HTTP-Methoden (GET, POST, PUT und DELETE) interagieren. Mithilfe von Controllern können verschiedene Endpunkte definiert werden, auf die die Clients zugreifen können, um Daten abzurufen, zu aktualisieren oder zu löschen.

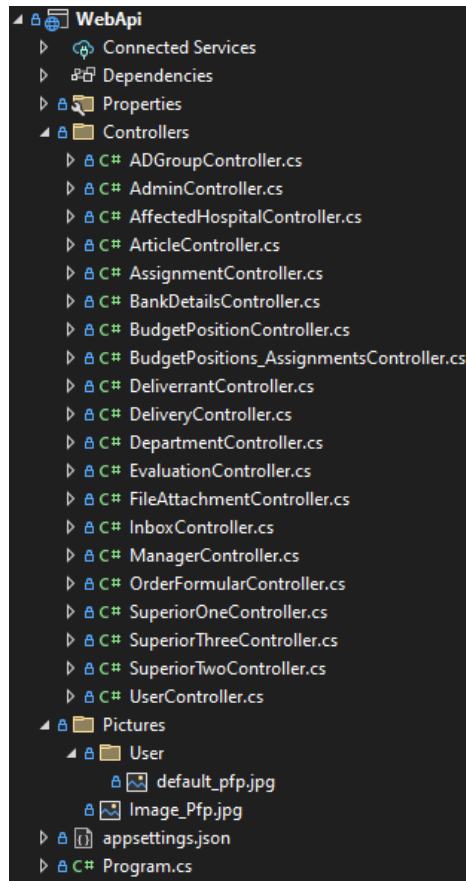


Abbildung 53: Aufbau WebApi

Außerdem haben wir auch einen Ordner „Pictures“. In diesem Ordner befindet sich ein weiterer Ordner namens „User“, wo das Standard-Profilbild aller Benutzer (default_pfp.jpg) gespeichert ist. Außerhalb des „User“-Ordners befindet sich ein weiteres Bild. Es handelt sich um das Firmenlogo, das oben links in der PDF-Bestellansicht angezeigt wird (Abbildung 45).

4.3.3 UnitOfWork

Das Unit of Work Pattern ist in der Softwareentwicklung weit verbreitet. Der Grund dafür ist, dass dadurch datenbankbezogene Operationen in einer einzelnen Transaktion gruppiert werden können. Das Ziel ist es, die Konsistenz und Integrität zu wahren. [49]

Ganz am Anfang des UnitOfWorks werden Repositories für jede Entität definiert, und zusätzlich werden im ersten Konstruktor die Repositories mit dem Datenbankkontext initialisiert.

```

1  public IUserRepository UserRepository { get; }
2  public IDeliverrantRepository LieferantRepository { get; set; }
3  public IOrderFormularRepository OrderFormularRepository { get; set; }
4  public IDepartmentRepository DepartmentRepository { get; set; }
5  public IDeliverrantRepository DeliverrantRepository { get; set; }
6  public IDeliveryRepository DeliveryRepository { get; set; }
7  public IArticleRepository ArticleRepository { get; set; }
8  public IAffectedHospitalRepository AffectedHospitalRepository { get; set; }
9  public IBudgetPoositionRepository BudgetPoositionRepository { get; set; }
10 public IAssignmentRepository AssignmentRepository { get; set; }
11 public IInboxRepository InboxRepository { get; set; }
12 public IFileAttachmentRepository FileAttachmentRepository { get; set; }
13 public IAdminRepository AdminRepository { get; set; }
14 public IManagerRepository ManagerRepository { get; set; }
15 public IGroupRepository GroupRepository { get; set; }
16 public IUserRoleRepository UserRoleRepository { get; set; }
17 public IRoleRepository RoleRepository { get; set; }
18 public IBankDetailsRepository BankDetailsRepository { get; set; }
19 public IBudgetPositionsAssignmentRepository BudgetPositionsAssignmentsRepository {
    get; set; }
20
21 public UnitOfWork(ApplicationDbContext applicationDbContext)
22 {
23     this._dbContext = applicationDbContext;
24     this.UserRepository = new UserRepository(_dbContext);
25     this.LieferantRepository = new DeliverrantRepository(_dbContext);
26     this.OrderFormularRepository = new OrderForumularRepository(_dbContext);
27     this.DepartmentRepository = new DepartmentRepository(_dbContext);
28     this.DeliverrantRepository = new DeliverrantRepository(_dbContext);
29     this.DeliveryRepository = new DeliveryRepository(_dbContext);
30     this.ArticleRepository = new ArticleRepository(_dbContext);
31     this.AffectedHospitalRepository = new AffectedHospitalRepository(_dbContext);
32     this.BudgetPoositionRepository = new BudgetPositionRepository(_dbContext);
33     this.AssignmentRepository = new AssignmentRepository(_dbContext);
34     this.InboxRepository = new InboxRepository(_dbContext);
35     this.FileAttachmentRepository = new FileAttachmentRepository(_dbContext);
36     this.AdminRepository = new AdminRepository(_dbContext);
37     this.ManagerRepository = new ManagerRepository(_dbContext);
38     this.GroupRepository = new GroupRepository(_dbContext);
39     this.UserRoleRepository = new UserRoleRepository(_dbContext);
40     this.RoleRepository = new RoleRepository(_dbContext);
41     this.BankDetailsRepository = new BankDetailsRepository(_dbContext);
42     this.BudgetPositionsAssignmentsRepository = new
        BudgetPositionsAssignmentRepository(_dbContext);
43 }

```

Im zweiten Konstruktor wird der erste Konstruktor aufgerufen, der eine Instanz von **ApplicationDbContext** mit den Konfigurationen als Parameter erstellt.

```
1 public UnitOfWork(IConfiguration configuration) : this(new
   ApplicationDbContext(configuration))
2 { }
```

Die `SaveChangesAsync`-Methode wird verwendet, um Änderungen zu speichern, die in der aktuellen Datenkontextinstanz (`dbContext`) vorgenommen wurden

```
1 public async Task<int> SaveChangesAsync()
2 {
3     var entities = _dbContext!.ChangeTracker.Entries()
4         .Where(entity => entity.State == EntityState.Added
5             || entity.State == EntityState.Modified)
6         .Select(e => e.Entity)
7         .ToArray(); // Geaenderte Entities ermitteln
8
9     // Allfaellige Validierungen der geaenderten Entities durchfuehren
10    foreach (var entity in entities)
11    {
12        ValidateEntity(entity);
13    }
14    return await _dbContext.SaveChangesAsync();
15
16 }
```

Als erstes werden die geänderten Entitäten ermittelt. Dies geschieht durch den **ChangeTracker**, der überprüft, ob sie den Zustand „Hinzugefügt“ oder „Geändert“ haben. Anschließend werden in der `foreach`-Schleife Validierungen für diese Entitäten durchgeführt. Schließlich werden sie asynchron in der Datenbank gespeichert.

Im `UnitOfWork` gibt es auch 2 „FillDb-Methoden“. Diese Methoden füllen die Datenbank, in unserem Fall mit Testdaten. Die **FillActiveDirectory()** Methode sieht folgendermaßen aus:

Ein großer unterschied zwischen den beiden Methoden ist die annotation die `FillActiveDirectory` hat und zwar `[SupportedOSPlatform("windows")]`

Die Annotation wird verwendet, um anzuzeigen, dass eine Methode, Klasse oder Assembly nur unter dem angegebenen Betriebssystem unterstützt wird, in diesem Fall Windows.

Als erstes wird die Datenbank gelöscht und dannach wir das Datenbankschema migriert.

```
1 await DeleteDatabaseAsync();
2 await MigrateDatabaseAsync();
```

Danach werden die Abteilungen in die Datenbank gespeichert.

```

1 var departments = new[]
2 {
3     new Department {Name = "Abteilung 1"},
4     new Department {Name = "Abteilung 2"},
5     new Department {Name = "Abteilung 3"},
6     new Department {Name = "FinanceDepartment"}
7 };
8
9 await _dbContext.Departments.AddRangeAsync(departments);

```

Der nächste Schritt ist das Einlesen der Lieferanten:

```

1 var csvLines = await
    MyFile.ReadStringMatrixFromCsvAsync("MIT_Kreditoren_20230706.csv", true);
2
3 List<Deliverrant> deliverrants = csvLines.GroupBy(line =>
4     new
5     {
6         KreditorenNr = (line[0]),
7         Name1 = line[1],
8         Name2 = line[2],
9         Address = line[3],
10        ZipCode = line[4],
11        Location = line[5],
12        Country = line[6],
13        PhoneNumber = line[7],
14        Fax = line[8],
15        Mail = line[9],
16        UidNr = line[10]
17    }).Select(grp =>
18        new Deliverrant()
19        {
20            KreditorenNr = grp.Key.KreditorenNr,
21            Name1 = grp.Key.Name1,
22            Name2 = grp.Key.Name2,
23            Address = grp.Key.Address,
24            ZipCode = grp.Key.ZipCode,
25            Location = grp.Key.Location,
26            Country = grp.Key.Country,
27            PhoneNumber = grp.Key.PhoneNumber,
28            Fax = grp.Key.Fax,
29            Mail = grp.Key.Mail,
30            UidNr = grp.Key.UidNr,
31        }).ToList();
32
33 await _dbContext.Deliverrants.AddRangeAsync(deliverrants);

```

Uns wurde eine CSV-Datei mit allen Lieferanten zur Verfügung gestellt. Hier ist ein Teil der CSV-Datei.

LiefNr	Name1	Name2	Straße	PLZ	Ort	Land
24	Dedalus HEALTHCARE Ges.m.b.H (Agfa)		Gertrude-Fröhlich-Sandner-Straße 1	1100	WIEN	AT
70	APOTHEKER-VERLAG	OESTERR. APOTHEKER-VERLAGSGES.MBH	SPITALGASSE 31A	1090	WIEN	AT
86	ARTAKER Büroautomation GmbH		Heumühlgasse 11	1040	WIEN	AT
132	BARTELT GES.M.B.H.	LABORAUSTRUSTUNG	NEUFELDWEG 42	8010	GRAZ	AT
149	Siemens Healthcare Diagnostics GmbH		SIEMENSSTRASSE 90	1210	WIEN	AT
187	BIO MERIEUX AUSTRIA GMBH		Harry-Glück-Platz 2/5	1100	WIEN	AT
198	Bit Factory Software GmbH		Fischergasse 22	4400	Steyr	AT
211	Roche Diagnostics GmbH		Engelhorngasse 3	1211	WIEN	AT
223	BRAUN AUSTRIA GMBH		OTTO BRAUN-STRASSE 3-5	2344	MARIA ENZEI	AT
346	DATO BERND MAIERHOFER		CORNELIUSGASSE 4/5	1060	WIEN	AT

Abbildung 54: CSV-Datei mit Lieferanten

Der nächste Teil besteht darin, die einzelnen Budgetpositionen zu speichern. Auch hier wurde uns eine CSV-Datei zur Verfügung gestellt (BudgetPositionenAssignmentNr.csv). Die CSV-Datei hat drei Spalten: „Konto-Bezeichnung“, „Innenauftrag“ und „Innenauftragsbezeichnung“. Die erste Spalte gehört zur Entität „Assignment“, die zweite Spalte zu „BudgetPosition“ und die letzte Spalte zu „BudgetPositionAssignment“, was die Kombination der beiden ist.

Konto Bezeichnung	Innenauftrag	Innenauftrag Bezeichnung
Wartungsverträge MIT	28000070	MI Sonstiges
Wartungsverträge MIT	28000069	LIS
Wartungsverträge MIT	28000068	PACS
Wartungsverträge MIT	28000067	PDMS
Wartungsverträge MIT	28000066	BHT-Systeme
Wartungsverträge MIT	28000065	Geburtendoku
Wartungsverträge MIT	28000064	Applikationen
Wartungsverträge MIT	28000063	SAP

Abbildung 55: CSV-Datei mit Lieferanten

```

1  var csvLinesBudgetPositionsAssignmentNr = await
    MyFile.ReadStringMatrixFromCsvAsync("BudgetPositionen_Assignment_Nr.csv",
    true);
2
3  List<Assignment> assignments = csvLinesBudgetPositionsAssignmentNr.GroupBy(line =>
4  new
5  {
6      Name = line[0]
7  }).Select(grp =>
8  new Assignment()
9  {
10     Name = grp.Key.Name
11  }).ToList();
12
13  List<BudgetPosition> budgetPositions =
    csvLinesBudgetPositionsAssignmentNr.GroupBy(line =>
14  new
15  {
16     Name = line[2]
17  }).Select(grp =>
18  new BudgetPosition()
19  {
20     Name = grp.Key.Name
21  }).ToList();
22
23  List<BudgetPosition_Assignment> budgetPosition_Assignments =
    csvLinesBudgetPositionsAssignmentNr.GroupBy(line =>
24  new
25  {
26     Number = Convert.ToInt32(line[1]),
27     Assignment = assignments.Single(a => a.Name == line[0]),
28     BudgetPosiotn = budgetPositions.Single(b => b.Name == line[2]),
29  }).Select(grp =>
30  new BudgetPosition_Assignment()
31  {
32     Number = grp.Key.Number,
33     Assignment = grp.Key.Assignment,
34     BudgetPosition = grp.Key.BudgetPosiotn
35  }).ToList();
36
37  await
    _dbContext.BudgetPositions_Assignments.AddRangeAsync(budgetPosition_Assignments);

```

Der nächste Schritt besteht darin, Gruppen hinzuzufügen, wie Benutzer, Vorgesetzte 1-3 und die Finanzabteilung. Zusätzlich werden den Gruppen auch gleichzeitig ihre Rollen bzw. Berechtigungen zugewiesen.

```

1  var groups = new[]
2  {
3      new Group
4      {
5          GroupName = "User"
6      },
7      new Group
8      {
9          GroupName = "Superior 1"
10     },
11     new Group
12     {
13         GroupName = "Superior 2"
14     },
15     new Group
16     {
17         GroupName = "Superior 3"
18     },
19     new Group
20     {
21         GroupName = "FinanceDepartment"
22     }
23 };
24
25 await _dbContext.Groups.AddRangeAsync(groups);
26 await _dbContext.SaveChangesAsync();
27
28 var groupRoleMapping = new Dictionary<string, string[]>
29 {
30     { "User", new[] { "Bestellen", "Statistik ansehen" } },
31     { "Superior 1", new[] { "Bestellen", "User erstellen (lokal)", "User
    bearbeiten (lokal)", "User loeschen (lokal)", "User aktivieren /
    deaktivieren", "Statistik ansehen" } },
32     { "Superior 2", new[] { "Bestellen", "User erstellen (lokal)", "User
    bearbeiten (lokal)", "User loeschen (lokal)", "User aktivieren /
    deaktivieren", "Statistik ansehen" } },
33     { "Superior 3", new[] { "Bestellen", "User erstellen (lokal)", "User
    bearbeiten (lokal)", "User loeschen (lokal)", "User aktivieren /
    deaktivieren", "Statistik ansehen" } },
34     { "FinanceDepartment", new[] { "Statistik ansehen" } },
35 };

```

Danach werden diese Gruppen mit den dazugehörigen Rollen mithilfe einer foreach-Schleife einzeln in die Datenbank gespeichert.

```

1  foreach (var groupRoleEntry in groupRoleMapping)
2  {
3      var groupName = groupRoleEntry.Key;
4      var rolesInGroup = groupRoleEntry.Value;
5
6      var group = await _dbContext.Groups.FirstOrDefaultAsync(g => g.GroupName ==
    groupName);
7      if (group == null)
8      {
9          throw new Exception($"Group '{groupName}' not found in the database. Make
    sure to seed groups first.");
10     }
11
12     var roles = rolesInGroup.Select(roleName => new Role { RoleName = roleName,
    Group_Id = group.Id }).ToList();
13
14     await _dbContext.Roles.AddRangeAsync(roles);
15     await _dbContext.SaveChangesAsync();
16 }

```

Zunächst werden die Mitglieder bzw. die Mitarbeiter zu jeder Gruppe geholt. Als erstes werden die einzelnen Gruppen, die im Active Directory sind, definiert. Dann werden die Mitarbeiter von einer Gruppe mithilfe der Methode **GetActiveDirectoryGroupMembers** geholt.

```

1 // Active Directory
2 var activeDirectoryGroupNameUser = "User";
3 var activeDirectoryGroupSuperior1 = "Superior 1";
4 var activeDirectoryGroupSuperior2 = "Superior 2";
5 var activeDirectoryGroupSuperior3 = "Superior 3";
6 var activeDirectoryGroupSuperiorFinaceDepartment = "FinanceDepartment";
7
8
9 await GetActiveDirectoryGroupMembers(activeDirectoryGroupNameUser);
10 await GetActiveDirectoryGroupMembers(activeDirectoryGroupSuperior1);
11 await GetActiveDirectoryGroupMembers(activeDirectoryGroupSuperior2);
12 await GetActiveDirectoryGroupMembers(activeDirectoryGroupSuperior3);
13 await GetActiveDirectoryGroupMembers(activeDirectoryGroupSuperiorFinaceDepartment);

```

Die **GetActiveDirectoryGroupMembers**-Methode hat als Parameter den Gruppennamen und gibt eine Liste von Benutzern zurück, die in diesem Fall die Mitarbeiter dieser Gruppe sind.

```

1 [SupportedOSPlatform("windows")]
2 public async Task<List<User>> GetActiveDirectoryGroupMembers(string groupName)
3 {
4     var membersList = new List<User>();
5
6     using (var principalContext = new PrincipalContext(ContextType.Domain,
7         "bestellsystem.local")) //
8     {
9         var groupPrincipal = GroupPrincipal.FindByIdentity(principalContext,
10             groupName); // Gets the members of the group
11
12         if (groupPrincipal != null)
13         {
14             var members = groupPrincipal.GetMembers();
15
16             foreach (Principal member in members)
17             {
18                 if (member is UserPrincipal user)
19                 {
20                     // Retrieve the Group from the database based on GroupName
21                     var group = await _dbContext.Groups.FirstOrDefaultAsync(g =>
22                         g.GroupName == groupName);
23
24                     // If the group does not exist, create it and add it to the
25                     // database
26                     if (group == null)
27                     {
28                         group = new Group { GroupName = groupName };
29                         await _dbContext.Groups.AddRangeAsync(group);
30                         await _dbContext.SaveChangesAsync();
31                     }
32                     // Retrieve the Manager from the database based on manager's
33                     // name
34                     var managerName = GetUserProperty(user, "manager");
35                     var managerNames = managerName.Split(new char[] { ' ' }, 2);
36                     // Split the name into two parts
37
38                     string firstName = managerNames.Length > 0 ? managerNames[0] :
39                         string.Empty; // Extract the first name
40                     string lastName = managerNames.Length > 1 ? managerNames[1] :
41                         string.Empty; // Extract the last name
42                     string resultFromFirstNameAndLastName =
43                         $"{firstName[0].ToString().ToLower()}.{lastName.ToLower()}";
44
45                     var manager = await _dbContext.Managers.FirstOrDefaultAsync(m
46                         => m.Email == resultFromFirstNameAndLastName +
47                         "@bestellsystem.local");
48
49                     // If the manager does not exist, create it and add it to the
50                     // database

```

```

39         if (manager == null)
40         {
41             manager = new Manager { Email =
                resultFromFirstNameAndLastName +
                "@bestellsystem.local" };
42             await _dbContext.Managers.AddRangeAsync(manager);
43             await _dbContext.SaveChangesAsync();
44         }
45
46         // Create the user object with properties from Active Directory
47         var adUser = new User
48         {
49             FirstName = user.GivenName,
50             LastName = user.Surname,
51             Email = $"{user.GivenName[0].ToString().ToLower()}
52             .{user.Surname.ToLower()}@bestellsystem.local",
53             Department_Id = GetDepartmentId(GetUserProperty(user,
                "department")),
54             Group_Id = group.Id, // Assign the Group directly
55             Manager_Id = manager.Id, // Assign the Manager directly
56             ProfilePicture = MyFile.defaultPicture,
57         };
58         await _dbContext.Users.AddRangeAsync(adUser);
59         membersList.Add(adUser);
60         await _dbContext.SaveChangesAsync();
61     }
62 }
63 }
64 }
65 }

```

Der Restliche Code wird auf der Seite 53 beschrieben

Die Zeile „var principalContext = new PrincipalContext(ContextType.Domain, "bestellsystem.local)“ erstellt eine neue Instanz der PrincipalContext-Klasse. Diese Klasse ermöglicht den Zugriff auf Active Directory-Operationen wie das Abfragen von Benutzerkonten, Gruppen usw.

Zwei Klassen werden in der Methode „GetActiveDirectoryGroupMembers“ verwendet, nämlich **GetDepartmentId** und **GetUserProperty**. „GetDepartmentId“ gibt anhand des Abteilungsnamens die Abteilungs-ID zurück. Die Methode „GetUserProperty“ ruft Informationen über einen Benutzer ab, z.B. den Abteilungsnamen.

```

1 private int GetDepartmentId(string departmentName)
2 {
3     Department? department = _dbContext.Departments.Where(d => d.Name ==
        departmentName).FirstOrDefault(); // Retrieve the department based on the
        name
4     return department?.Id ?? 0; // Return the department ID or 0 if not found
5 }
6
7 [SupportedOSPlatform("windows")]
8 private string GetUserProperty(UserPrincipal user, string propertyName)
9 {
10    DirectoryEntry? directoryEntry = (DirectoryEntry)user.GetUnderlyingObject();
11    if (directoryEntry.Properties.Contains(propertyName))
12    {
13        string managerPropertyValue =
            directoryEntry.Properties[propertyName]?.Value?.ToString() ??
            string.Empty;
14        // Extract the name from the manager property
15        int startIndex = managerPropertyValue.IndexOf('=') + 1;
16        int endIndex = managerPropertyValue.IndexOf(',', startIndex);
17        return (startIndex > 0 && endIndex > 0) ?
            managerPropertyValue.Substring(startIndex, endIndex - startIndex) :
            null;
18        return managerPropertyValue;
19    }
20    return string.Empty;
21 }

```

4.3.4 Funktionsweise des Workflows

Die gesamte Logik des Workflows befindet sich in den Web-API-Klassen „OrderFormularController“. Sie besteht aus 2 API-Methoden, wobei ein Teil der Logik beim Anlegen eines Bestellformulars in der Methode „AddOrderFormular“ liegt. Zudem gibt es eine Methode in der Web-API-Klasse „InboxController“, die für das Ablehnen der Benachrichtigung zuständig ist. Jede Methode spielt eine wichtige Rolle.

Die erste Methode ist die Benachrichtigung nachdem die Bestellung getätigt wurde.

Nach dem Anlegen wird zunächst überprüft, ob der Ersteller des Formulars ein Vorgesetzter ist. Wenn ja, wird entsprechend der Wert für den Vorgesetzten eingetragen, näher erläutert auf Seite 93 unter dem Punkt „OrderFormular“. Der Grund dafür ist, dass ein Vorgesetzter die Bestellung selbst anlegt. Daher kann der Status direkt auf „Akzeptiert“ gesetzt werden.

```

1  User? user = await _unitOfWork.UserRepository.GetByIdAsync(currentUserId);
2
3  if (user == null)
4  {
5      return NotFound();
6  }
7  else if (user.Group!.GroupName != "User")
8  {
9      switch (user.Group!.GroupName)
10     {
11         case "Superior 1":
12             newOrder.Superior1 = 25;
13             break;
14         case "Superior 2":
15             newOrder.Superior2 = 50;
16             break;
17         case "Superior 3":
18             newOrder.Superior3 = 75;
19             break;
20     };
21
22     if (user.Group!.GroupName != "FinanceDepartment")
23     {
24         isSuperiorOrder = true;
25     }
26 }

```

Nachdem festgestellt wurde, ob der Ersteller ein Vorgesetzter ist oder nicht, wird der nächste Vorgesetzte des Erstellers ermittelt. Anschließend wird eine neue Benachrichtigung (Inbox) mit dem Benachrichtigungstyp „new“ erstellt.

Die Benachrichtigungstypen dienen dazu, im Frontend die richtigen Benachrichtigungen auszugeben. Das bedeutet, wenn der Benachrichtigungstyp „new“ ist, sieht die Benachrichtigung wie in **Abbildung 28** aus.

```

1  int manager_Id = await
    _unitOfWork.UserRepository.GetManagerIdByUserId(orderformular.User_Id);
2  Manager? manager = await
    _unitOfWork.ManagerRepository.GetManagerEmailByManagerId(manager_Id);
3  User? userManager = await
    _unitOfWork.UserRepository.GetUserByEmailAddress(manager!.Email);
4
5  Inbox newInbox = new Inbox()
6  {
7      User_Id = orderformular.User_Id,
8      Manager_Id = manager_Id,
9      OrderFormular_Id = newOrder.Id,
10     NotificationType = "new",
11     Time = DateTime.Now.ToString("HH:mm:ss")
12 };
13
14 _unitOfWork.InboxRepository.Insert(newInbox);
15 await _unitOfWork.SaveChangesAsync();

```

Der Benutzer hat die Möglichkeit, eine „Benachrichtigungs-E-Mail“ in den Einstellungen anzugeben. Wenn diese E-Mail nicht festgelegt wurde, wird die E-Mail, die beim Login verwendet wurde, für die Benachrichtigungen genutzt (z. B. e.johnson@bestellsystem.local).

```

1  //Email to the user who send the formular
2
3  if (user.NotificationEmail != "" && userManager?.NotificationEmail != "")
4  {
5      emailSender.SendEmail("Formular wurde erfolgreich gesendet!",
6          $"Hallo {user.FirstName} {user.LastName}!\n" +
7          $"Dein Formular wurde erfolgreich beim Vorgesetzten
8              {userManager?.FirstName} {userManager?.LastName} gesendet und ist im
9              Dashboard zu sehen!", user.NotificationEmail);
10
11     emailSender.SendEmail($"Neuer Formular eingang von {user.FirstName}
12         {user.LastName}!",
13         $"Hallo {userManager?.FirstName} {userManager?.LastName}!\n" +
14         $"Ein neuer Formular wurde von {user.FirstName} {user.LastName}
15         eingereicht, siehe im Postfach in der Website nach.",
16         userManager!.NotificationEmail);
17 }
18 else
19 {
20     emailSender.SendEmail("Formular wurde erfolgreich gesendet!",
21         $"Hallo {user.FirstName} {user.LastName}!\n" +
22         $"Dein Formular wurde erfolgreich beim Vorgesetzten
23             {userManager?.FirstName} {userManager?.LastName} gesendet und ist im
24             Dashboard zu sehen!", user.Email);
25
26     emailSender.SendEmail($"Neuer Formular eingang von {user.FirstName}
27         {user.LastName}!",
28         $"Hallo {userManager?.FirstName} {userManager?.LastName}!\n" +
29         $"Ein neuer Formular wurde von {user.FirstName} {user.LastName}
30         eingereicht, siehe im Postfach in der Website nach.",
31         userManager!.Email);
32 }

```

Wenn es sich um eine Bestellung handelt, bei der ein Vorgesetzter etwas bestellt, wird eine Benachrichtigung an den nächsten Vorgesetzten mit dem Typ „superiorOrder“ erstellt und die Benachrichtigung wird auch per E-Mail versendet.

```

1  if (isSuperiorOrder)
2  {
3      string currentUserEmail = await
4          _unitOfWork.UserRepository.GetEmailAddressByUserId(currentUserId);
5      string currentUserAuthenticatedEmail = await
6          _unitOfWork.UserRepository.GetAuthenticatedEmailAddressByUserId(currentUserId);
7      int manager_IdSelfOrder = await
8          _unitOfWork.ManagerRepository.GetManagerIdByEmail(currentUserEmail);

```

```

1
2     Inbox newInboxSuperior = new Inbox()
3     {
4         User_Id = orderformular.User_Id,
5         Manager_Id = manager_IdSelfOrder,
6         OrderFormular_Id = newOrder.Id,
7         NotificationType = "superiorOrder",
8         Time = DateTime.Now.ToString("HH:mm:ss")
9     };
10
11
12     _unitOfWork.InboxRepository.Insert(newInboxSuperior);
13     await _unitOfWork.SaveChangesAsync();
14
15     //Email to the user who send the formular
16     if (user.NotificationEmail != "" && userManager?.NotificationEmail != "")
17     {
18         emailSender.SendEmail("Formular wurde erfolgreich gesendet!",
19             $"Hallo {user.FirstName} {user.LastName}!\n" +
20             $"Dein Formular wurde erflogreich beim Vorgesetzten
21                 {userManager?.FirstName} {userManager?.LastName} gesendet und ist
22                 im Dashboard zu sehen!", user.NotificationEmail);
23
24         emailSender.SendEmail($"Neuer Formular eingag von {user.FirstName}
25             {user.LastName}!",
26             $"Hallo {userManager?.FirstName} {userManager?.LastName}!\n" +
27             $"Ein neuer Formular wurde von {user.FirstName} {user.LastName}
28                 eingereicht, siehe im Postfach in der Website nach.",
29             userManager!.NotificationEmail);
30     }
31
32     else
33     {
34         emailSender.SendEmail("Formular wurde erfolgreich gesendet!",
35             $"Hallo {user.FirstName} {user.LastName}!\n" +
36             $"Dein Formular wurde erflogreich beim Vorgesetzten
37                 {userManager?.FirstName} {userManager?.LastName} gesendet und ist
38                 im Dashboard zu sehen!", user.Email);
39
40         emailSender.SendEmail($"Neuer Formular eingag von {user.FirstName}
41             {user.LastName}!",
42             $"Hallo {userManager?.FirstName} {userManager?.LastName}!\n" +
43             $"Ein neuer Formular wurde von {user.FirstName} {user.LastName}
44                 eingereicht, siehe im Postfach in der Website nach.",
45             userManager!.Email);
46     }
47 }

```

Die anderen API-Methoden, „NewInbox“, „UpdateOrderFormularProgressStatus“ und „DeclineOrderFormular“, sind im Grunde dasselbe, jedoch gibt es einige Unterschiede.

Die Methode **NewInbox** ist vereinfachter als die vorherige Methode, aber im Wesentlichen ist sie dasselbe. Der Zweck dieser Methode ist, dass diese Benachrichtigung nicht nach dem Anlegen eines Formulars, sondern nach dem Akzeptieren eines Formulars geschickt werden soll.

UpdateOrderFormularProgressStatus aktualisiert den Fortschritt des Formulars, wenn es angenommen oder abgelehnt wurde, und schickt auch entsprechend die passenden E-Mails.

Die letzte Methode, **DeclineOrderFormular**, macht grundsätzlich genau dasselbe wie „NewInbox“, nur wird hier statt einem neuen Formular eine Benachrichtigung geschickt, dass das Formular abgelehnt wurde.

4.3.5 Email Client

Zur Verbesserung der internen Kommunikation und um eine sichere Übermittlung von Informationen zu gewährleisten, wurde entschieden, die E-Mail-Kommunikation über den firmeneigenen Mail-Server der KUK zu leiten. Die speziell für diesen Zweck eingerichtete E-Mail-Adresse `kuk.bestellsystem@ooeg.at` dient als zentrale Anlaufstelle für alle kommunikationsrelevanten Aspekte im Zusammenhang mit dem Bestellsystem. Diese Maßnahme stellt sicher, dass alle Nachrichten effizient verarbeitet und in Einklang mit den firmeninternen Sicherheitsprotokollen behandelt werden.

Zielsetzung:

Die Einrichtung der firmeninternen E-Mail-Adresse diene speziell dazu, den Versand von Bestätigungs-E-Mails für Bestellungen und geschäftliche Mitteilungen zu automatisieren. Ziel war es, die Übermittlung essentieller Informationen zwischen Mitarbeitern und Vorgesetzten zuverlässig und sicher zu gestalten, um eine effektive Verfolgung und Bestätigung von Transaktionen innerhalb des Unternehmens zu gewährleisten.

C# Code für das Senden von E-Mails:

```
1 public void SendEmail(string subject, string body, string toEmail)
2 {
3     var fromAddress = new MailAddress(_fromEmail, "Bestellsystem KUK");
4     var toAddress = new MailAddress(toEmail);
5
6     var smtp = new SmtpClient
7     {
8         Host = _host,
9         Port = _port,
10        EnableSsl = true,
11        DeliveryMethod = SmtpDeliveryMethod.Network,
12        UseDefaultCredentials = false,
13        Credentials = new NetworkCredential(fromAddress.Address, _password),
14        Timeout = 20000
15    };
16
17    using (var message = new MailMessage(fromAddress, toAddress))
18    {
19        Subject = subject,
20        Body = body
21    })
22    {
23        smtp.Send(message);
24    }
25 }
```

Funktionsweise der SendEmail-Methode

Die `SendEmail`-Methode ist dafür verantwortlich, E-Mails zu versenden. Sie akzeptiert drei Parameter: den Betreff (`subject`), den Inhalt (`body`) und die E-Mail-Adresse des Empfängers (`toEmail`).

1. `var fromAddress = new MailAddress(_fromEmail, "Bestellsystem KUK");`
Hier wird die Adresse des Absenders erstellt. Der Absender hat einen Namen ("Bestellsystem KUK") und eine E-Mail-Adresse, die in der Variable `_fromEmail` gespeichert ist.
2. `var toAddress = new MailAddress(toEmail);` Es wird eine Adresse für den Empfänger erstellt. Die E-Mail-Adresse des Empfängers wird aus dem Parameter `toEmail` genommen.
3. `var smtp = new SmtpClient { ... };` Hier wird ein neues Objekt der Klasse `SmtpClient` erstellt, das für den Versand von E-Mails über das Simple Mail Transfer Protocol (SMTP) zuständig ist. Dabei werden verschiedene Einstellungen wie Host, Port und Anmeldeinformationen konfiguriert.
4. `using (var message = new MailMessage(fromAddress, toAddress) { ... };`
Ein neues E-Mail-Nachrichtenobjekt (`MailMessage`) wird erstellt und mit Absender, Empfänger, Betreff und Inhalt konfiguriert.
5. `smtp.Send(message);` Die erstellte E-Mail-Nachricht wird über den zuvor konfigurierten SMTP-Client (`smtp`) versendet.

5 Deployment

5.1 Deployment: Verbindung mit einer Domäne

Hier wird der Ablauf des Deployments beschrieben, speziell die Integration eines Windows Servers 2022 in die Domäne bestellsystem.local sowie die Verbindung einer Windows 11 Workstation mit diesem Server.

5.1.1 Windows Server 2022 Setup

Ein Windows Server 2022 wurde mit der Domäne bestellsystem.local eingerichtet. Hierbei wurde die Active Directory-Dienste installiert, einschließlich der DNS-Rollen.

5.1.2 Windows 11 Workstation Setup

Eine Windows 11 Pro Workstation innerhalb des Netzwerks soll durch die Integration in einen Windows Server 2022 in die Domäne eingebunden werden.

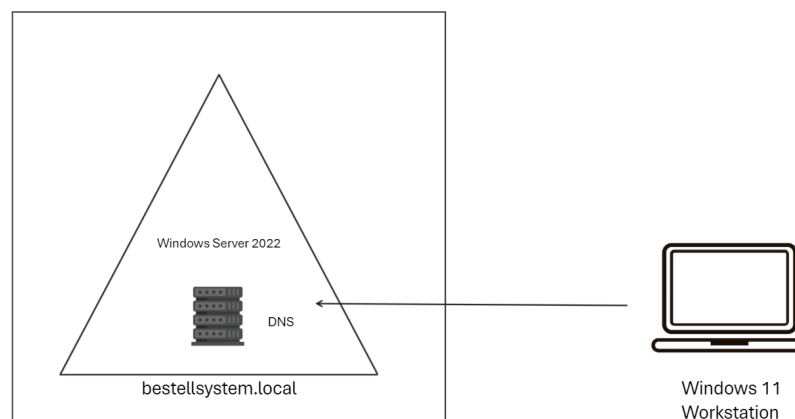


Abbildung 56: Verbindung mit einer Domäne

5.2 Netzwerkeinstellungen

5.2.1 Windows Server 2022

- **Hostname:** WindowsServer
- **IP-Adresse:** 192.168.199.146
- **Subnetzmaske:** 255.255.255.0
- **Gateway:** 192.168.199.2
- **DNS:** 127.0.0.1
- **Domäne:** bestellsystem.local

5.2.2 Windows 11 Pro Workstation

- **Hostname:** karlof
- **IP-Adresse:** 192.168.199.147
- **Subnetzmaske:** 255.255.255.0
- **Gateway:** 192.168.199.2
- **DNS-Server:** 192.168.199.101

Diese Konfigurationen ermöglichen eine klare Kommunikation und Integration zwischen dem Windows Server 2022 und der Windows 11 Pro Workstation im Netzwerk.

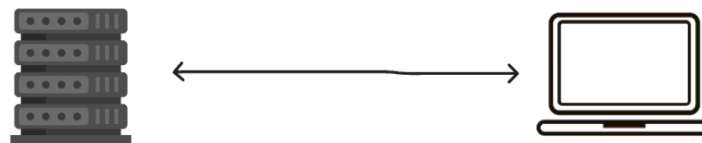
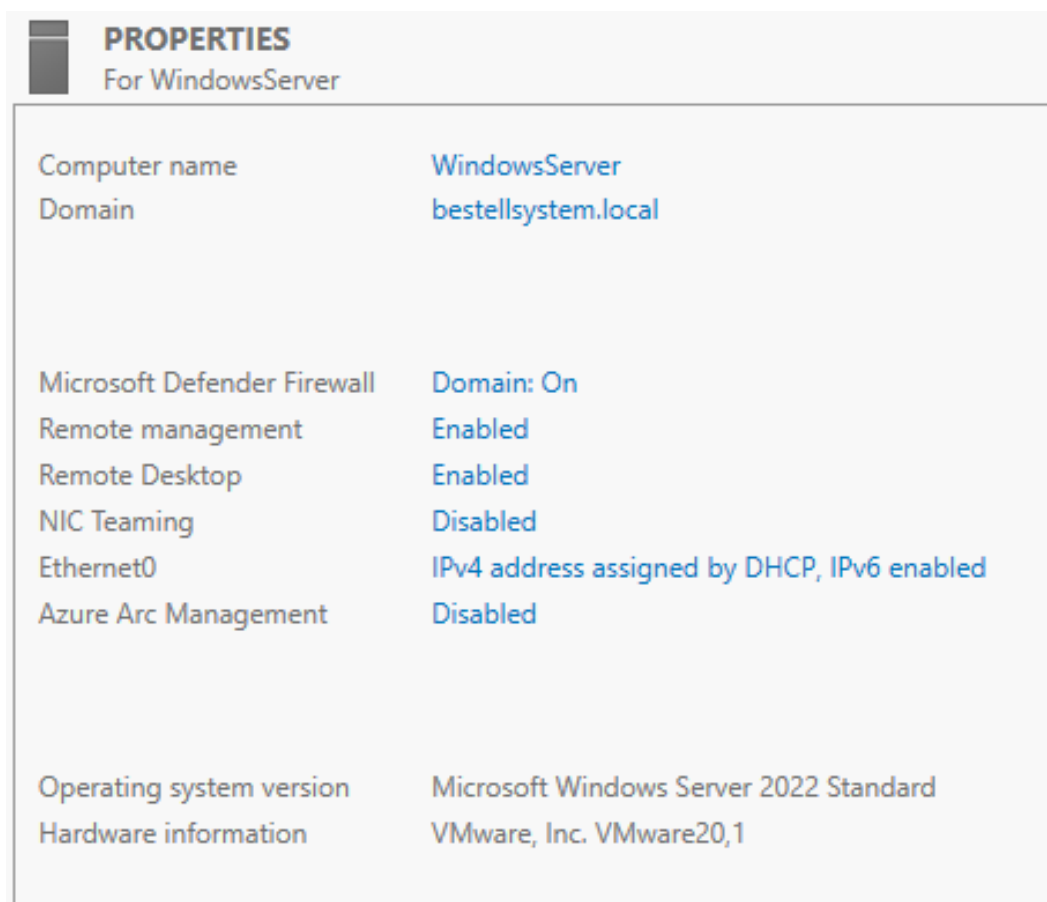


Abbildung 57: Netzwerkeinstellungen

5.3 Konfiguration von Windows Server 2022

Im Rahmen der Konfiguration des Servers im Server Manager-Programm wurden wichtige Parameter festgelegt. Der Server erhielt den Namen WindowsServer und wurde in die Domäne bestellsystem.local integriert. Zusätzlich wurden Remote Management und Remote Desktop aktiviert, um effiziente Verwaltung und Fernzugriffsmöglichkeiten zu gewährleisten. Die NIC-Teaming-Option wurde deaktiviert, um die Netzwerkkonfiguration anzupassen.

Nach der Erstellung des Servers und den grundlegenden Einstellungen erfolgte die Konfiguration der Ethernet-Schnittstelle. Dabei wurden spezifische Netzwerkeinstellungen festgelegt, um eine reibungslose Kommunikation im Netzwerk zu gewährleisten. Parallel dazu wurde die Zeitzone entsprechend der geografischen Lage eingestellt, um sicherzustellen, dass zeitabhängige Prozesse synchronisiert und korrekt ausgeführt werden.



Computer name	WindowsServer
Domain	bestellsystem.local
Microsoft Defender Firewall	Domain: On
Remote management	Enabled
Remote Desktop	Enabled
NIC Teaming	Disabled
Ethernet0	IPv4 address assigned by DHCP, IPv6 enabled
Azure Arc Management	Disabled
Operating system version	Microsoft Windows Server 2022 Standard
Hardware information	VMware, Inc. VMware20,1

Abbildung 58: Konfiguration von Windows Server 2022 im Server Manager

5.4 Serverrollen im Windows Server 2022

Die Abbildung(59) zeigt den Server-Manager von Windows Server 2022, in dem die manuell installierten Rollen zu sehen sind: Active Directory-Domänendienste (AD DS), DHCP, DNS, Datei- und Speicherdienste sowie Internet Information Services (IIS). Diese Dienste sind für die Netzwerkinfrastruktur und die Anwendungslogik von zentraler Bedeutung.

5.4.1 Gründe für die manuelle Installation

Windows Server 2022 bietet eine Vielzahl von Diensten, die je nach den Anforderungen und der Architektur des Unternehmensnetzwerks ausgewählt und konfiguriert werden müssen.

5.4.2 Zweck innerhalb des Projekts

Die Implementierung dieser Technologien trägt entscheidend zur Funktion und Sicherheit des Bestellsystems bei. AD DS und DNS sind die Grundpfeiler für eine effiziente Netzwerknavigation und Benutzerverwaltung. IIS bietet die notwendige Plattform für das Hosting der Webanwendung, während DHCP die Netzwerkkonfiguration und -verwaltung erleichtert.

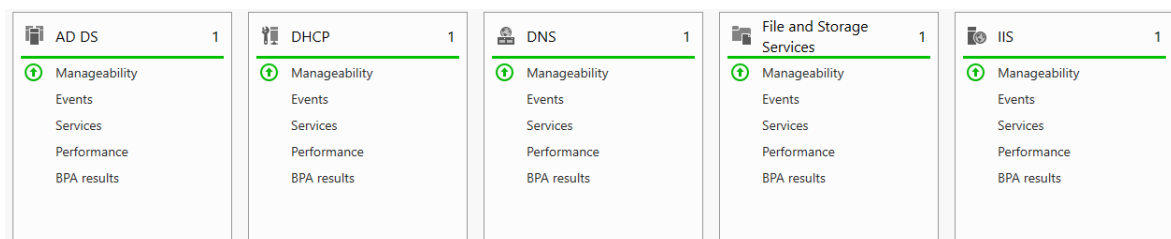


Abbildung 59: Installation von DNS und Active Directory Services

5.4.3 Active Directory

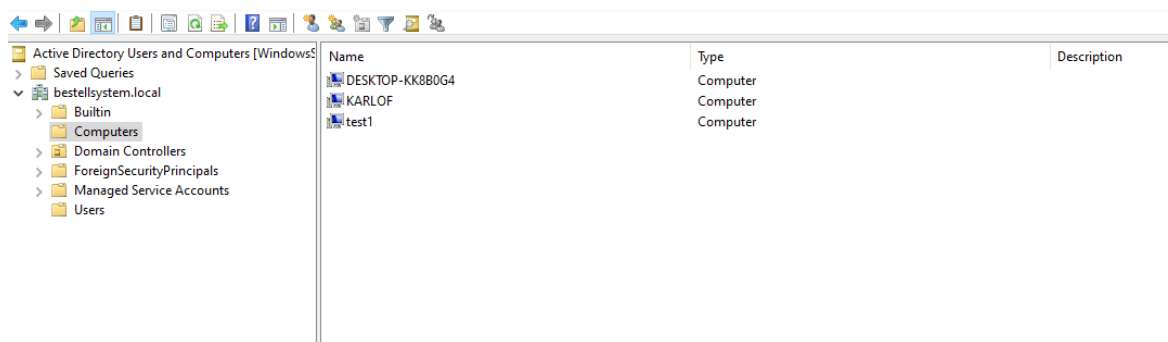
In unserem Projekt wurde das Active Directory (AD) verwendet, um eine realitätsnahe Testumgebung zu schaffen. Dafür haben wir zunächst Testgruppen und Benutzerkonten erstellt. Die Gruppen umfassten verschiedene Benutzerrollen, darunter 'User', 'Vorgesetzter 1-3' und 'Finanzabteilung'. Für jede Gruppe wurden exemplarisch zwei Testbenutzer mit vollständigen Attributen angelegt. Die initialen Testgruppen sind in Abbildung 60 dargestellt.



Schlüsseladministratoren	Security Group - Global	Mitglieder dieser Grupp...
Schreibgeschützte Domänencontroller	Security Group - Global	Mitglieder dieser Grupp...
Schreibgeschützte Domänencontroller der Organisation	Security Group - Universal	Mitglieder dieser Grupp...
Sophia Walker	User	
Superior 1	Distribution Group - Global	
Superior 2	Security Group - Global	
Superior 3	Security Group - Global	
Unternehmensschlüsseladministratoren	Security Group - Universal	Mitglieder dieser Grupp...
User	Distribution Group - Global	
William Wilson	User	
Zertifikatherausgeber	Security Group - Domain Local	Mitglieder dieser Grupp...
Zulässige RODC-Kennwortreplikationsgruppe	Security Group - Domain Local	Mitglieder dieser Grupp...

Abbildung 60: Active Directory - Windows 11 Pro Workstation

Zusätzlich zu den Benutzerkonten wurde ein Testcomputer im Active Directory angelegt, um die Anmeldeprozesse für die Testbenutzer zu simulieren. Dieser Schritt war notwendig, um sicherzustellen, dass die Testbenutzer sich an einem lokalen PC anmelden können, was die Konfiguration eines repräsentativen Testgeräts im AD erforderte. Die Konfiguration des Testcomputers im Active Directory ist in Abbildung 61 gezeigt.



Name	Type	Description
DESKTOP-KK8B0G4	Computer	
KARLOF	Computer	
test1	Computer	

Abbildung 61: Erstellung von Gruppen im Active Directory

5.5 Entwicklungsumgebung mit Visual Studio 2022

5.5.1 Einleitung

In unserer Diplomarbeit haben wir uns für Visual Studio 2022 entschieden, um eine Entwicklungsumgebung zu schaffen, die sowohl die Anforderungen moderner Softwareentwicklung erfüllt als auch eine nahtlose Zusammenarbeit im Team ermöglicht. Das Kernstück unserer Entwicklungsarbeit bildete ein lokaler Entwicklungsserver, der es uns erlaubte, Änderungen schnell umzusetzen und in einer Umgebung zu testen, die der Produktionsumgebung nachempfunden ist.

5.5.2 Öffentlicher Zugriff auf lokalen Entwicklungsserver

Eine zentrale Herausforderung in unserem Projekt war die Bereitstellung des Zugriffs auf das Backend für Teammitglieder, die remote arbeiten. Die Lösung boten die Dev Tunnels von Visual Studio 2022, die es ermöglichten, unseren lokalen Entwicklungsserver sicher für externe Zugriffe verfügbar zu machen. Diese Technologie war entscheidend für die Agilität unseres Entwicklungsprozesses.

5.5.3 Funktionsweise eines lokalen Entwicklungsservers

Der lokale Entwicklungsserver, unterstützt durch Dev Tunnels, spielte eine Schlüsselrolle in unserem Entwicklungsworkflow. Als private Instanz eines Web-Servers ermöglichte er es uns, Anwendungen unter realen Bedingungen zu debuggen und zu testen, ohne dass Änderungen direkt in der Produktionsumgebung vorgenommen werden mussten.

5.5.4 Integration mit Visual Studio 2022

Um den öffentlichen Zugriff zu aktivieren, konfigurierten wir den lokalen Server in Visual Studio 2022 entsprechend und passten die Firewall-Einstellungen an, um einen sicheren externen Zugriff zu gewährleisten. Die Einrichtung eines Dev Tunnels erforderte einige spezifische Schritte, darunter die Auswahl des 'Backend'-Tunnels in den Einstellungen von Visual Studio und die Generierung einer öffentlichen URL, über die unser Team auf das Backend zugreifen konnte.

5.5.5 Einsatz von Dev Tunnels für die Teamarbeit

Die Integration von Dev Tunnels in unseren Entwicklungsprozess bot einen entscheidenden Vorteil für die Teamarbeit. Durch die Bereitstellung eines öffentlich zugänglichen Links konnten alle Teammitglieder unabhängig von ihrem Standort in Echtzeit auf das Backend zugreifen und Änderungen testen. Diese Funktionalität war nicht nur ein bedeutender Fortschritt gegenüber traditionellen Entwicklungsmethoden, sondern förderte auch eine kollaborative und effiziente Arbeitsweise innerhalb unseres Teams.

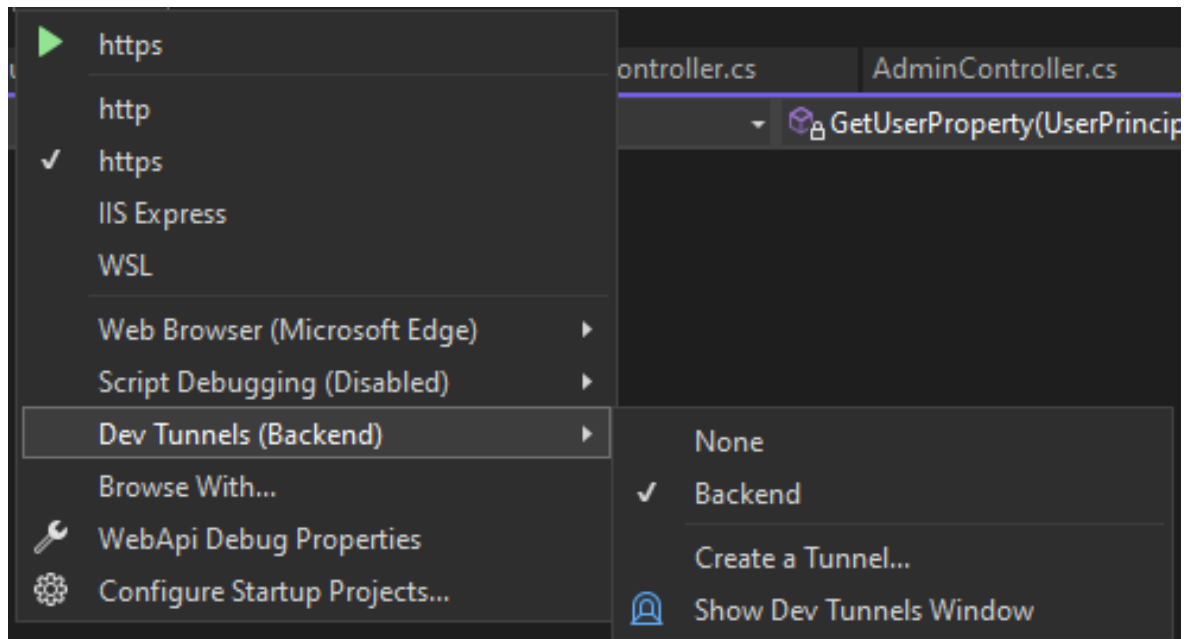


Abbildung 62: Nutzung von Dev Tunnels in Visual Studio 2022 zur Verbesserung der Teamkollaboration

5.5.6 Dev Tunnels - Erstellen eines Dev Tunnels

Die Abbildung 63 illustriert den Prozess der Erstellung eines Dev Tunnels in Visual Studio 2022, der für das Projekt unerlässlich war, um einen sicheren und stabilen Zugang zum lokalen Entwicklungsserver für das Team und autorisierte Nutzer zu ermöglichen.

5.5.7 Notwendigkeit eines Microsoft-Accounts

Zur Erstellung eines Dev Tunnels ist ein Microsoft-Account erforderlich, über den der Tunnel initialisiert wird. Dies gewährleistet eine nahtlose Integration in die Microsoft-Infrastruktur und erlaubt die Nutzung von Azure-Diensten.

5.5.8 Name und Tunnel-Typ

Im Konfigurationsdialog, dargestellt in Abbildung 63, muss zuerst der Name des Tunnels festgelegt werden. In unserem Fall wurde der Tunnel als 'Backend' benannt, was seine Verwendung für das Backend des Projekts kennzeichnet.

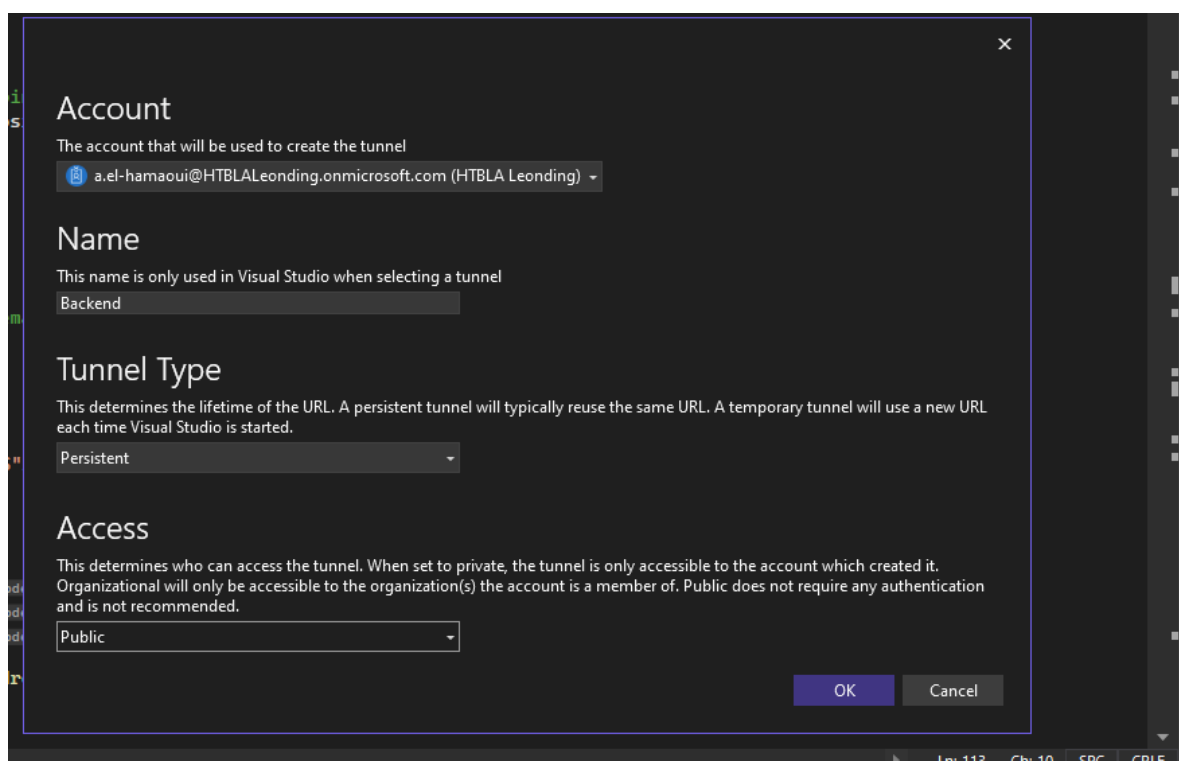


Abbildung 63: Erstellen eines Dev Tunnels in Dev Tunnels

Der *Tunnel-Typ* definiert die Lebensdauer der URL. Ein *persistenter* Tunnel behält dieselbe URL bei jedem Neustart von Visual Studio bei, während ein *temporärer* Tunnel

eine neue URL generiert, was zusätzliche Sicherheit bietet, da der Zugang nicht dauerhaft gleich bleibt.

5.5.9 Zugangsberechtigung

Die *Zugangsberechtigung* bestimmt, wer den Tunnel verwenden kann. Mit der Einstellung *Public* kann jeder auf den Tunnel zugreifen, was für die gemeinsame Nutzung mit dem Team oder für Präsentationszwecke nützlich sein kann. Die Einstellung *Private* beschränkt den Zugang auf den Ersteller des Tunnels oder auf spezifische Mitglieder einer Organisation, was für interne Testzwecke vorteilhaft ist.

5.6 Angular-Konfiguration für Externen Zugriff

Um sicherzustellen, dass die Angular-Anwendung auf anderen Computern zugänglich ist, ist eine spezifische Konfiguration erforderlich. Der Befehl `ng serve --host 0.0.0.0` spielt hierbei eine entscheidende Rolle. Durch die Angabe von `--host 0.0.0.0` wird der Angular-Entwicklungsserver so konfiguriert, dass er auf Anfragen von allen IP-Adressen lauscht, anstatt nur auf lokale Verbindungen zu reagieren.

Diese Einstellung ist unerlässlich, wenn die Angular-Anwendung über das Netzwerk freigegeben und von verschiedenen Computern aus darauf zugegriffen werden soll. Der Befehl wird in der Kommandozeile ausgeführt und öffnet den Entwicklungsserver für externe Verbindungen.

```
C:\Users\Administrator\Documents\GitHub\Bestellsystem-KUK\Frontend>ng serve --host 0.0.0.0
Warning: This is a simple server for use in testing or debugging Angular applications
locally. It hasn't been reviewed for security issues.

Binding this server to an open connection can result in compromising your application or
computer. Using a different host than the one passed to the "--host" flag might result in
websocket connection issues. You might need to use "--disable-host-check" if that's the
case.

✓ Browser application bundle generation complete.

Initial Chunk Files | Names          | Raw Size
vendor.js           | vendor        | 3.09 MB
main.js             | main          | 938.75 kB
styles.css, styles.js | styles       | 458.93 kB
polyfills.js        | polyfills     | 333.06 kB
scripts.js          | scripts       | 288.30 kB
runtime.js          | runtime       | 6.51 kB
                    | Initial Total | 5.07 MB

Build at: 2024-02-21T14:25:56.573Z - Hash: 00f1873839d60c6a - Time: 11203ms
** Angular Live Development Server is listening on 0.0.0.0:4200, open your browser on http://localhost:4200/ **
```

Abbildung 64: Angular-Konfiguration für externen Zugriff

5.7 Netzwerkkonfiguration

Um eine erfolgreiche Netzwerkverbindung zum Windows Server 2022 zu etablieren, mussten spezifische Einstellungen im Windows-Client vorgenommen werden, wie in Abbildung 65 dargestellt. Diese Konfiguration erfolgte im Eigenschaftenbereich des Internetprotokolls Version 4 (TCP/IPv4) und umfasste:

- **IP-Adresse:** Eine statische IP-Adresse wurde zugewiesen, um eine eindeutige Identifikation des Clients im Netzwerk zu ermöglichen.
- **Subnetzmaske:** Die Subnetzmaske definiert das lokale Subnetz und die Zugehörigkeit des Clients zu diesem Netzwerksegment.
- **Standardgateway:** Das Standardgateway wurde eingetragen, um die Kommunikation mit Geräten außerhalb des lokalen Netzwerks zu ermöglichen.
- **Bevorzugter DNS-Server:** Die manuelle Konfiguration des DNS-Servers war entscheidend, um die Namensauflösung innerhalb des Netzwerks sicherzustellen und Zugriff auf die Domain `bestellsystem.local` zu ermöglichen.

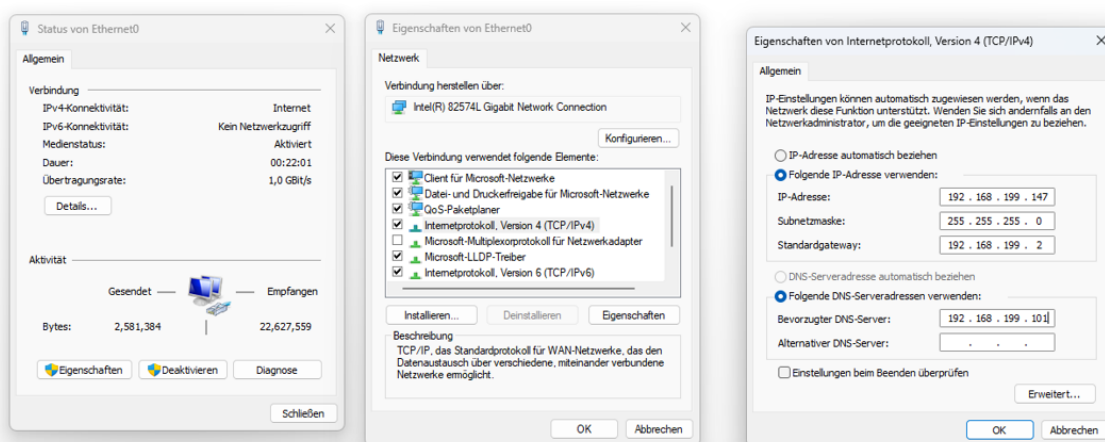


Abbildung 65: Windows Client Netzwerkeinstellungen

Nach der Eingabe der Netzwerkparameter wurde die Verbindung durch das Senden eines Ping-Befehls an den DNS-Server (z.B. `ping 192.168.199.101`) überprüft. Die erfolgreiche Antwort auf das Ping bestätigte die Funktionsfähigkeit der Einstellungen. Ein Neustart des Clients war der letzte Schritt, der notwendig war, um die Konnektivität vollständig herzustellen und eine zuverlässige Verbindung mit der Domain `bestellsystem.local` zu ermöglichen.

5.7.1 Integration in die Domäne `bestellsystem.local`

Der Beitritt unserer Client-Computer zur Domäne `bestellsystem.local` war ein entscheidender Schritt, um eine effektive Verwaltung und Steuerung der Netzwerkressourcen zu ermöglichen. Indem jeder Client Teil der Domäne wird, profitiert das System von den Vorteilen der zentralisierten Kontrolle durch das Active Directory, wie zum Beispiel einer einheitlichen Benutzerauthentifizierung und -autorisation sowie der Anwendung von Gruppenrichtlinien. Dies ist insbesondere wichtig für die Gewährleistung der Sicherheit und die Einhaltung der Unternehmensrichtlinien.

Die Abbildung 66 zeigt die Benutzeroberfläche, auf der die Einstellungen zur Domänenintegration vorgenommen wurden.

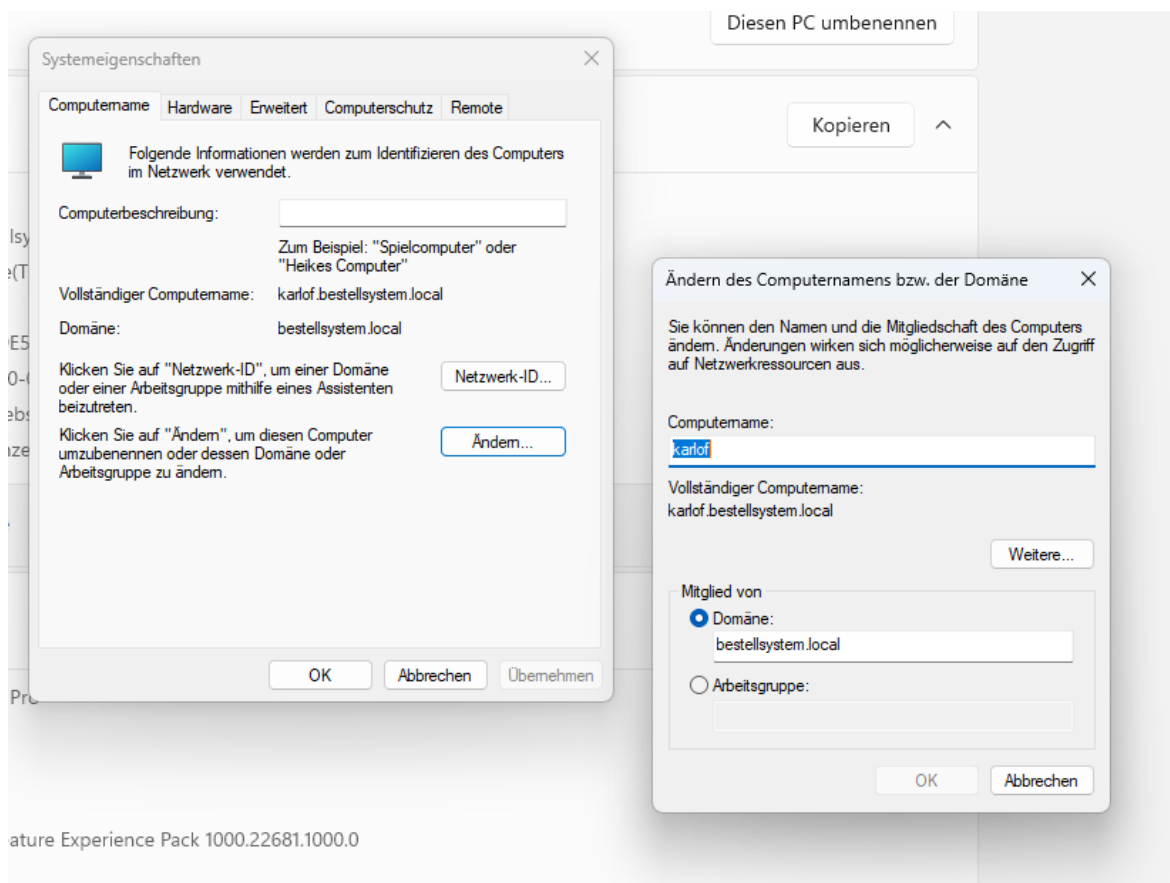


Abbildung 66: Benutzeroberfläche für den Beitritt zur Domäne

Nach Abschluss der Netzwerkkonfiguration und der Bestätigung einer funktionierenden Verbindung zum Domänencontroller, wurde der Beitrittsprozess über die Systemeinstellungen des Clients durchgeführt. Dieser Vorgang verknüpfte den Client mit der Domäne und ermöglichte die Nutzung des vollen Funktionsumfangs von Active Directory, was eine nahtlose Verbindung und Interaktion mit dem Bestellsystem sicherstellte.

Abbildungsverzeichnis

1	Hierarchieebene	4
2	Excel Datei	5
3	Systemarchitektur	8
4	Anmeldung AD DS	46
5	Externe Anmeldung und Admin-Zugriff	47
6	Benutzergruppe im Admin-Dashboard	51
7	Vorgesetzter 1 im Admin-Dashboard	52
8	Bestellformular Teil 1	55
9	Bestellformular Teil 2	57
10	Fehlermeldung, wenn kein Produkt eingetragen wurde	57
11	Lieferant anlegen	58
12	Lieferant bearbeiten	59
13	Auswertungen als Vorgesetzter Ansicht	60
14	Zeitraum auswählen	60
15	Filter auswählen	61
16	Auswertungen	62
17	Bestell-Übersicht	63
18	Button zum Wechseln des Designs	63
19	Bestell-Übersicht im dunklen Design	64
20	Bestell-Informationen zu einer Bestellung	64
21	Formularfortschritt wenn der Vorgesetzter 1 die Bestellung akzeptiert hat	66
22	Formularfortschritt wenn ein Vorgesetzter abgelehnt hat	66
23	Benachrichtigungs-Seite (Mitarbeiter)	67
24	Dropdown-Menüpunkt „Konto -> Benachrichtigungen“	67
25	Benachrichtigungs-Seite (Vorgesetzter)	68
26	Bestellzustellungs-Mail (Mitarbeiter)	68
27	Bestellempfangs-Mail (Vorgesetzter)	69
28	Neue Zustellungsbenachrichtigung für Vorgesetzten 2	69

29	Formularfortschritt aktualisiert	69
30	Formularfortschritt aktualisiert	70
31	Bestellformular abgelehnt mit Grund	70
32	Bestellformular abgelehnt Benachrichtigung per E-Mail	70
33	Formularfortschritt wurde zurückgesetzt	71
34	Bestell-Formular Details Part 1	71
35	Bestell-Formular Details Part 2	72
36	Bestellung ablehnen	72
37	Bestell-Formular ansehen Button	73
38	Bestellformular-Detailseite, aber die Felder sind nicht schreibgeschützt.	73
39	Die 3 Felder um die Budgetposition anzugeben	74
40	Auswahlliste für Budget	74
41	Auswahlliste für Zuordnung	74
42	Auftrag/KST oder Invest Feld ist 0	75
43	Ein anderes Beispiel für eine Budgetposition.	75
44	PDF-Exportseite mit Datei-Anhängen	77
45	PDF Bestellansicht	78
46	PDF Bestellung Tabelle	79
47	Profil-Einstellungen im Active Directory	82
48	Sicherheitseinstellungen	84
49	Benutzererstellung im internen Bereich	85
50	Benutzermanagement im internen Bereich	86
51	Bearbeitung externer Benutzer im internen Bereich	87
52	E-Mail-Benachrichtigungseinstellungen	88
53	Aufbau WebApi	97
54	CSV-Datei mit Lieferanten	100
55	CSV-Datei mit Lieferanten	101
56	Verbindung mit einer Domäne	110
57	Netzwerkeinstellungen	111
58	Konfiguration von Windows Server 2022 im Server Manager	112
59	Installation von DNS und Active Directory Services	113
60	Active Directory - Windows 11 Pro Workstation	114
61	Erstellung von Gruppen im Active Directory	114
62	Nutzung von Dev Tunnels in Visual Studio 2022 zur Verbesserung der Teamkollaboration	116

63	Erstellen eines Dev Tunnels in Dev Tunnels	117
64	Angular-Konfiguration für externen Zugriff	118
65	Windows Client Netzwerkeinstellungen	119
66	Benutzeroberfläche für den Beitritt zur Domäne	120

Literaturverzeichnis

- [1] Linzer Wiki https://www.linzwiki.at/wiki/Ober%C3%B6sterreichische_Gesundheitsholding/
- [2] Wikipedia https://de.wikipedia.org/wiki/Ober%C3%B6sterreichische_Gesundheitsholding
- [3]
- [4] Oberösterreichische Gesundheitsholding GmbH <https://www.ooeg.at/>
- [5] Microsoft. *Server Manager*. <https://learn.microsoft.com/en-us/windows-server/administration/server-manager/server-manager>
- [6] Microsoft. *Active Directory Domain Services – Überblick*. <https://learn.microsoft.com/de-de/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>
- [7] Microsoft. *DNS in Windows Server - Overview*. <https://learn.microsoft.com/de-de/windows-server/networking/dns/dns-top>
- [8] Microsoft. *File and Storage Services Overview*. [https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831487\(v=ws.11\)](https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831487(v=ws.11))
- [9] Microsoft. *Introduction to IIS Architecture*. <https://learn.microsoft.com/de-de/iis/get-started/introduction-to-iis/introduction-to-iis-architecture#introduction>
- [10] ANSYS. *Adding Inbound Rules to Windows Defender Firewall*. <https://optics.ansys.com/hc/en-us/articles/7144748040467-Adding-inbound-rules-to-Windows-defender-firewall>
- [11] Kaspersky. *About Outbound Firewall Rules*. <https://support.kaspersky.com/KSWS/11/en-US/181677.htm>

- [12] Mozilla Developer Network. *JavaScript - A high-level, interpreted programming language in the 1990s*. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Language_overview
- [13] Bootstrap. *Overview - Bootstrap*. <https://getbootstrap.com/docs/4.0/about/overview/>
- [14] Bootstrap Blog. *Bootstrap Icons*. <https://blog.getbootstrap.com/2019/11/26/bootstrap-icons/>
- [15] Typografie.info. *TrueType*. <https://www.typografie.info/3/wiki.html/t/truetype-r162/>
- [16] Angular. *Architecture Overview - Angular*. <https://angular.io/guide/architecture>
- [17] CODE Magazine. *Introduction to Node Modules*. <https://www.codemag.com/article/1709061/Introduction-to-Node-Modules>
- [18] Business-Wissen. *Introduction to Scrum*. <https://www.business-wissen.de/hb/so-funktioniert-scrum-rollen-regeln-methoden/>
- [19] Visual Paradigm. *BurnDownChart*. <https://www.visual-paradigm.com/scrum/scrum-burndown-chart/>
- [20] Zoho Sprints. *SprintReview*. <https://www.zoho.com/sprints/sprint-reviews.html#:~:text=A%20sprint%20review%20is%20a,enable%20them%20to%20provide%20feedback>
- [21] Fecher. *GitHub*. <https://www.modernizing-applications.de/it-glossar/github-definition/>
- [22] GitHub. *GitHub Desktop*. <https://docs.github.com/en/desktop/overview/about-github-desktop#about-github-desktop>
- [23] Microsoft. *Zentralverwaltung*. <https://docs.microsoft.com/de-de/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>
- [24] TypeScriptHero. *TypeScriptHero*. <https://marketplace.visualstudio.com/items?itemName=rbbbit.typescript-hero>
- [25] IntelliSense. *IntelliSense*. <https://marketplace.visualstudio.com/items?itemName=Zignd.html-css-class-completion>

- [26] Git. *Git Branching*. <https://git-scm.com/book/de/v2/Git-Branching-Branches-auf-einen-Blick>
- [27] HtmlCssSupport. <https://marketplace.visualstudio.com/items?itemName=eemel.vscod-html-css>
- [28] AngularSnippets. <https://marketplace.visualstudio.com/items?itemName=johnpapa.Angular2>
- [29] MoveTs. <https://marketplace.visualstudio.com/items?itemName=stringham.move-ts>
- [30] Microsoft *ASP.NET Web Forms*. <https://learn.microsoft.com/en-us/aspnet/web-forms/what-is-web-forms> Microsoft *ASP.NET WebAPI*. <https://dotnet.microsoft.com/en-us/apps/aspnet/apis>
- [31] Microsoft *Entity Framework Core*. <https://learn.microsoft.com/en-us/ef/core/> Microsoft *Sprachintegrierte Abfrage (Language-Integrated Query, LINQ)* <https://learn.microsoft.com/de-de/dotnet/csharp/linq/>
- [32] Kinsta *Was ist MySQL? Eine anfängerfreundliche Erklärung* <https://kinsta.com/de/wissensdatenbank/was-ist-mysql/>
- [33] Microsoft. *Zentralverwaltung*. <https://docs.microsoft.com/de-de/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>
- [34] Microsoft. *An introduction to NuGet*. <https://learn.microsoft.com/en-us/nuget/what-is-nuget>
- [35] Bouncycastle *The Legion of the Bouncy Castle* <https://www.bouncycastle.org/csharp/> Avast *Was ist der MD5-Hash-Algorithmus und wie funktioniert er?* [https://www.avast.com/de-de/c-md5-hashing-algorithm#:~:text=MD5%20\(Message%20DDigest%20Algorithm\),und%20digitalen%20Signaturen%20verwendet%20wird](https://www.avast.com/de-de/c-md5-hashing-algorithm#:~:text=MD5%20(Message%20DDigest%20Algorithm),und%20digitalen%20Signaturen%20verwendet%20wird) Google Ads-Hilfe *SHA-256-Algorithmus: Definition* <https://support.google.com/google-ads/answer/9004655?hl=de#:~:text=SHA%2D256%20ist%20eine%20kryptologische,Hashfunktionen%20erzeugen%20irreversible%2C%20eindeutige%20Hashwerte> Wikipedia *Elliptic-curve Diffie–Hellman* https://en.wikipedia.org/wiki/Elliptic-curve_Diffie%E2%80%93Hellman
- [36] Github *DotNetZip* <https://github.com/haf/DotNetZip.Semverd>
- [37] Wikipedia *iText* <https://de.wikipedia.org/wiki/IText>

- [38] Stackoverflow *How do I resolve Unknown PdfException when using itext7 in net maui* <https://stackoverflow.com/questions/76260218/how-do-i-resolve-unknown-pdfexception-when-using-itext7-in-net-maui> Viewbag *PDF DIGITAL SIGNATURES WITH ITEXT7, BOUNCY CASTLE AND .NET CORE* <https://viewbag.wordpress.com/2019/12/24/pdf-digital-signatures-itext7-bouncy-castle-net-core/>
- [39] Microsoft *Entity Framework Core tools reference* <https://learn.microsoft.com/en-us/ef/core/cli/> Microsoft *ntity Framework Core tools reference - Package Manager Console in Visual Studio* <https://learn.microsoft.com/en-us/ef/core/cli/powershell>
- [40] Microsoft *CMicrosoft.Extensions.Configuration.Json Namespace* <https://learn.microsoft.com/de-de/dotnet/api/microsoft.extensions.configuration.json?view=dotnet-plat-ext-7.0>
- [41] NuGet *MySql.Data* <https://www.nuget.org/packages/MySql.Data/> *MySQL Installing Connector/NET Using NuGet* <https://dev.mysql.com/doc/connector-net/en/connector-net-installation-binary-nuget.html>
- [42] MySqlConnection *MySqlConnection: High Performance .NET MySQL Driver* <https://mysqlconnector.net/> Wikipedia *ADO.NET* <https://de.wikipedia.org/wiki/ADO.NET>
- [43] Entity Framework Core *EF Core MySQL* <https://entityframeworkcore.com/providers-mysql>
- [44] Microsoft *System.DirectoryServices.AccountManagement Namespace* <https://learn.microsoft.com/en-us/dotnet/api/system.directoryservices.accountmanagement?view=dotnet-plat-ext-8.0> NuGet *System.DirectoryServices.AccountManagement* <https://www.nuget.org/packages/System.DirectoryServices.AccountManagement/>
- [45] Ionos *Swagger: Mehr Komfort bei der API-Entwicklung* <https://www.ionos.de/digitalguide/websites/web-entwicklung/was-ist-swagger/>
- [46] Microsoft *Überblick über C#* <https://learn.microsoft.com/de-de/dotnet/csharp/tour-of-csharp/>
- [47] Bundesministerium Finanzen *Bundesbeschaffung GmbH (BBG)* [https://www.bmf.gv.at/ministerium/aufgaben-und-organisation/bundesbeschaffung.html#:~:text=Die%20Bundesbeschaffung%20GmbH%20\(%20BBG%20\)%20bietet,ein%20umfassendes%20Produkt%2D%20und%20Dienstleistungsportfolio.](https://www.bmf.gv.at/ministerium/aufgaben-und-organisation/bundesbeschaffung.html#:~:text=Die%20Bundesbeschaffung%20GmbH%20(%20BBG%20)%20bietet,ein%20umfassendes%20Produkt%2D%20und%20Dienstleistungsportfolio.)
- [48] Bundesbeschaffung *Beschaffung Grundlagen des Vergaberechts* [https://www.bbg.gv.at/information/vergaberecht#:~:text=Das%20Bundesvergabegesetz%](https://www.bbg.gv.at/information/vergaberecht#:~:text=Das%20Bundesvergabegesetz%20)

202018%20regelt%20die,sowie%20ein%20Gleichbehandlungs%2D%20und%
20Transparenzgebot.

- [49] Medium *Unit of Work in C#: A Practical Guide for Developers*
<https://medium.com/c-sharp-programming/unit-of-work-design-pattern-in-c-a-practical-guide-for-developers-948474156837#:~:text=The%20Unit%20of%20Work%20design,and%20integrity%20in%20the%20database.>