

Mitarbeiterverwaltungssystem

1 Ausgangssituation

Die Übernahme der Firma Pfusch durch Herrn Pfuscher markiert einen bedeutsamen Schritt in der Unternehmensentwicklung. Im Zuge dieser Neuausrichtung erkannte Herr Pfuscher die Notwendigkeit einer zeitgemäßen und effizienten Mitarbeiterverwaltung. Die bislang ausschließliche Nutzung von Mitarbeiterakten seines Vaters offenbarte sich als zeitintensiver Prozess, der nicht den Anforderungen der modernen Geschäftswelt entspricht. Die Suche nach relevanten Mitarbeiterdaten gestaltete sich als mühsam und ineffizient.

Um diese Herausforderungen zu bewältigen und eine zeitgemäße Lösung zu implementieren, begab sich Herr Pfuscher auf die Suche nach innovativen Ansätzen zur elektronischen Verwaltung von Mitarbeiterinformationen. In diesem Kontext stieß er auf die Möglichkeit einer Zusammenarbeit mit Uns.

Im Rahmen eines Erstgesprächs erläuterte Herr Pfuscher seine klaren Vorstellungen hinsichtlich eines effektiven Verwaltungsprogramms. Dabei standen die Anlage neuer Mitarbeiter*Innen, die zügige Auffindbarkeit bereits existierender Daten sowie die umfassende Anzeige aller relevanten Informationen im Vordergrund. Aspekte wie Einstellungs- und Ausscheidungsdaten, eine Löschfunktion für veraltete Informationen, Gehaltsvorstellungen und die Berücksichtigung hierarchischer Strukturen, einschließlich Positionen wie Chef, Vorstände, Abteilungs- und Teamleiter sowie Angestellte, waren essenziell für die effiziente Unternehmensverwaltung.

Die Firma Pfusch beabsichtigt nun, ein Mitarbeiterverwaltungsprogramm einzuführen, das nicht nur den aktuellen Anforderungen gerecht wird, sondern auch flexibel genug ist, um zukünftigen Bedürfnissen standzuhalten. Dieses Pflichtenheft skizziert im Detail die Anforderungen, Etappenziele, Rahmenbedingungen, technischen Aspekte, Teams, Schnittstellen, mögliche Probleme, Qualitätsstandards und den Zeitplan für die Entwicklung des Mitarbeiterverwaltungsprogramms – ein Prototyp, der die Grundlage für die innovative Neugestaltung der internen Verwaltungsprozesse bei Pfusch bilden soll.

2 Aufgabenstellung

Die vorgesehene Anwendung basiert auf einer SQL-Datenbank, in der wichtige Mitarbeiterdaten abgelegt und verwaltet werden. Der Schutz dieser Daten steht im Vordergrund, weshalb das Verwaltungsprogramm eine sichere Anmeldung mit einer SHA256- und zusätzlichen Salt-

Mitarbeiterverwaltungssystem

Verschlüsselung vorsieht. Ein vorheriges Registrierungsverfahren mit Benutzername und Passwort ist notwendig, um sich anschließend einzuwählen. Die Anwendung bietet grundlegende Funktionen wie das Anlegen, Bearbeiten, Löschen von Mitarbeiterinformationen sowie eine effiziente Ein- und Ausgabe der gesamten Mitarbeiterliste in ein geeignetes Format (Export nach .csv und .txt)

Der Code zeichnet sich durch parametrisierte Abfragen zur Vermeidung von z.B. SQL-Injektionen aus, ist lesbar und strukturiert gestaltet und beinhaltet Plausibilitätsüberprüfungen zur Fehlervermeidung, insbesondere bei der Anlage oder Bearbeitung von Mitarbeiterdaten.

Eine Suchfunktion ermöglicht eine schnelle und einfache Durchsuchung der Mitarbeiterdaten, während die Benutzeroberfläche bewusst einfach gehalten wurde.

3 Bereits bestehende Systeme oder Produkte

Es existieren derzeit keine elektronischen Systeme oder Produkte zur Mitarbeiterverwaltung in der Firma Pfuscher. Herr Pfuscher hat erkannt, dass die Implementierung einer elektronischen Verwaltung notwendig ist, um den Verwaltungsaufwand, der zuvor manuell bewältigt wurde, zu reduzieren. Ziel ist es, den Zugriff auf Mitarbeiterinformationen zu erleichtern und effizienter zu gestalten.

4 Teams und Schnittstellen

Teammitglieder: Okcu, Dogukan (1), Akinalieva, Aidai (2), Cakir, Ahmet (3)

Aufgabenverteilung

Entwickler (1) -> Implementierung: Anlegen neuer Mitarbeiter / Bearbeiten / Entfernen von Mitarbeiterinformationen, Suchfunktion

Entwickler (2) -> Implementierung einer Datenbankstruktur und das Design der grafischen Oberfläche unserer Anwendung

Entwickler (3) -> Implementierung: Authentifizierungs- und Verschlüsselungsmethoden, Login- und Registrierungsmaske, Ein- und Ausgabefunktion (Import, Export)

Die Anwendungslogik des Systems bildet eine Schnittstelle zur zentralen Datenbank, in der Mitarbeiterdaten abgelegt werden.

Die Schnittstelle zwischen den Entwicklern besteht in der koordinierten Integration ihrer jeweiligen Funktionen im Mitarbeiterverwaltungssystem.

Kommunikationswege

- Die Teamkommunikation erfolgte über u.a. über Microsoft Teams, wobei insbesondere die Calling-Funktionen für Echtzeitbesprechungen genutzt wurden
- GitHub diente als zentrale Plattform für Code-Entwicklung, Versionierung und Projektmanagement. Die Zusammenarbeit erfolgte durch Push- und Pull-Anfragen, um Änderungen zu integrieren und den Projektfortschritt zu verfolgen

Nutzung von GitHub

GitHub fungierte als zentrale Plattform, die maßgeblich zur erfolgreichen Durchführung der Software beitrug.

Die Entscheidung für GitHub basiert auf den folgenden vorteilhaften Aspekten, die im Rahmen des Pflichtenhefts hervorgehoben werden:

1. Effiziente Code-Entwicklung

GitHub ermöglicht eine reibungslose Zusammenarbeit der Entwickler. Durch Funktionen wie Parallelentwicklung, Zweigverwaltung und Zusammenführung (Merge) kann das Team gleichzeitig an verschiedenen Teilen des Codes arbeiten und Änderungen nahtlos integrieren.

2. Versionskontrolle

Die ausgefeilte Versionskontrolle von GitHub bietet einen klaren Überblick über den Code-Fortschritt. Die systematische Verfolgung von Änderungen erleichtert die Rückkehr zu früheren Code-Zuständen, was die Stabilität und Qualität des Codes erhöht.

3. Integriertes Projektmanagement

GitHub stellt Tools für das Projektmanagement bereit, darunter Issues, Milestones und Projekttasks. Diese Funktionen unterstützen die Planung, Zuweisung von Aufgaben und bieten einen transparenten Überblick über den Projektfortschritt. Die Möglichkeit zur direkten Kommunikation über Kommentare und Diskussionen im Code fördert eine klare Teamkommunikation.

Die umfassenden Vorteile von GitHub machen es zu einer unverzichtbaren Plattform, die nicht nur die technische Umsetzung, sondern auch eine effektive Zusammenarbeit und zuverlässige Versionskontrolle gewährleistet.

Die klare Aufgabenverteilung innerhalb des Entwicklerteams und die definierten Schnittstellen gewährleisteten einen reibungslosen Entwicklungsprozess für das Mitarbeiterverwaltungssystem. Die Kommunikationswege über Teams (Calling) und GitHub förderten eine effiziente Zusammenarbeit und ermöglichten eine transparente Verfolgung des Projektfortschritts.

5 Rahmenbedingungen

Unterrichtstermine und Betriebszeiten

Datum	Unterrichtsstunden	Projektphase
30.01	4 Stunden	Definition- und Planungsphase
09.02	2 Stunden	Realisierungsphase, Testphase
20.02	2 Stunden	
01.03	2 Stunden	Abschlussphase
Gesamt	12,5 Stunden	2,5 Stunden (Betriebszeit)

Im Rahmen des Schulprojekts sind insgesamt 10 Unterrichtsstunden vorgesehen, verteilt auf verschiedene Termine im Zeitraum vom 30.01 bis 01.03. Die Unterrichtseinheiten dienen der praktischen Umsetzung und Vertiefung der im theoretischen Teil erworbenen Kenntnisse.

Für den Fall, dass mehr Stunden für das Projekt benötigt werden, können diese flexibel über die Betriebszeiten der Ausbildungsstätte kompensiert werden. Die regulären Betriebszeiten erstrecken sich von 8:00 Uhr bis 17:00 Uhr. Somit besteht die Möglichkeit, je nach Bedarf zusätzliche Stunden einzuplanen, um eine ausreichende Zeitressource für die Projektarbeit zu gewährleisten.

Die flexible Nutzung der Betriebszeiten ermöglicht eine bedarfsgerechte Anpassung an die Anforderungen des Schulprojekts und stellt sicher, dass ausreichend Zeit für die erfolgreiche Umsetzung des Projekts zur Verfügung steht.

6 Technische Anforderungen

Die erfolgreiche Umsetzung des Projekts erfordert bestimmte technische Voraussetzungen und Ausstattungen. Die wesentlichen Anforderungen im Überblick:

- **Entwicklungsumgebung**
 - Microsoft Visual Studio als primäre Entwicklungsumgebung
 - Erforderliche Erweiterungen (Extensions) für Visual Studio müssen installiert sein

Mitarbeiterverwaltungssystem

- **Software**
 - Microsoft Teams für die Kommunikation und Meetings im Team
 - Vertrautheit mit GitHub und Kenntnisse darüber, wie es für die Versionskontrolle und Zusammenarbeit genutzt wird
- **Hardware**
 - Ein funktionierendes Notebook, das den Anforderungen der Entwicklungsumgebung entspricht
 - Kopfhörer mit integriertem Mikrofon für klare Kommunikation während Team-Meetings
 - Optional: Eine funktionierende Kamera für Video-Meetings
- **Technisches Knowhow**
 - Fundierte Kenntnisse in C# (C-Sharp) als Programmiersprache
 - Grundlegende Kenntnisse in SQL für die Datenbankinteraktion
 - Nutzung des SQL Server Management Studios für die Verwaltung der Datenbank
- **Allgemeines technisches Verständnis**
 - Grundlegendes technisches Knowhow für die Fehlerbehebung und effiziente Nutzung der Entwicklungs- und Kommunikationstools

7 Problemanalyse

Datenverlust und Systemausfall

Ein zentrales Problem, dem wir begegnen könnten, ist der mögliche Datenverlust und Systemausfall. Um diesem vorzubeugen, planen wir die Implementierung regelmäßiger Backups und setzen auf eine robuste, redundante Datenbankstruktur. Dies ermöglicht nicht nur die Minimierung von Datenverlusten, sondern auch einen schnellen Wiederherstellungsprozess im Falle eines Systemausfalls.

Sicherheitslücken und Datenschutzverletzungen

Die Sicherheit sensibler Daten steht im Fokus unserer Betrachtungen. Mögliche Sicherheitslücken und Datenschutzverletzungen erfordern eine kontinuierliche Überwachung der Software, regelmäßige Updates und Schulungen für das Team, um Datenschutzrichtlinien strikt einzuhalten und die Integrität der Daten zu gewährleisten.

Mitarbeiterverwaltungssystem

Performance und Reaktionszeiten

Eine weitere Herausforderung könnte sich in Performance-Problemen und langen Reaktionszeiten manifestieren. Um diesen vorzubeugen, setzen wir auf regelmäßige Performance-Analysen, Code-Optimierungen und skalierbare Lösungen. Unser Ziel ist es, eine schnelle und effiziente Nutzung der Anwendung sicherzustellen.

Komplexität der Benutzeroberfläche

Die Gestaltung der Benutzeroberfläche ist von zentraler Bedeutung. Wir wollen mögliche Schwierigkeiten aufgrund von Komplexität minimieren. Durchdachte Gestaltung und kontinuierliches Benutzerfeedback tragen dazu bei, die Benutzeroberfläche einfach und intuitiv zu halten.

Erweiterbarkeit und Flexibilität

Um langfristig flexible und erweiterbare Lösungen zu gewährleisten, legen wir besonderen Wert auf die Planung unserer Softwarearchitektur. Die Struktur wird darauf ausgerichtet, zukünftige Anforderungen einfach aufnehmen zu können. Regelmäßige Überprüfungen und Anpassungen sorgen für eine anhaltende Erweiterbarkeit.

Förderung der Akzeptanz und Schulungsbereitschaft

Die Akzeptanz der Lösung durch die Endanwender ist entscheidend. Um möglichen Widerständen entgegenzuwirken, binden wir die Endanwender frühzeitig in den Entwicklungsprozess ein. Zudem planen wir Schulungen für die Mitarbeiter*Innen und sehen kontinuierliche Anpassungen basierend auf deren Feedback vor.

Die frühzeitige Identifikation und Lösungsansätze für diese potenziellen Probleme gewährleisten eine robuste und widerstandsfähige Umsetzung des Projekts. Dieser proaktive Ansatz ermöglicht es uns, Zeit und Ressourcen effizient zu nutzen und die Qualität der Lösung sicherzustellen.

8 Qualität

Um höchste Qualität und Sicherheit im Projekt zu gewährleisten, haben wir folgende Qualitätsansprüche an unsere Anwendung definiert.

1. Sicherheit vor schädliche Datenbankabfragen

Implementierung von parametrisierten Abfragen, um potenzielle SQL-Injektionen zu verhindern und die Integrität der Datenbank zu schützen

Mitarbeiterverwaltungssystem

2. Kontoschutz durch starke Kryptografie

Anwendung von Verschlüsselungstechniken wie SHA256 und Salt für eine robuste Verschlüsselung, um die Sicherheit der Benutzerkonten zu gewährleisten und sensible Informationen zu schützen

3. Redundante Datenbank für Ausfallsicherheit

Einrichtung einer redundanten Datenbankstruktur, um Ausfallsicherheit zu gewährleisten und die sichere Speicherung der Daten zu garantieren

4. Schnelle Reaktionszeiten

Optimierung der Anwendung für eine schnelle Reaktionszeit, um die Effizienz der Mitarbeiterverwaltung zu maximieren

5. Benutzerfreundliches Design

Gestaltung einer benutzerfreundlichen Anwendungsoberfläche, um die Verwaltung von Mitarbeiterdaten intuitiv und effektiv zu gestalten

6. Erweiterbarkeit & Zuverlässigkeit

Konzeption der Softwarearchitektur mit Fokus auf Erweiterbarkeit, um zukünftige Anpassungen und Funktionalitätserweiterungen zu ermöglichen. Gleichzeitig wird hohe Zuverlässigkeit angestrebt

7. Effiziente Datenverwaltung

Implementierung von Import- und Exportfunktionen für Mitarbeiterdaten zur erleichterten Ein- und Ausgabe. Die Integration ermöglicht nicht nur einen formatierten Export für eine übersichtliche Datenausgabe, sondern auch einen strukturierten Import für eine nahtlose Datenaktualisierung und -erweiterung

Qualitätssicherung, -kontrolle und -abnahme

- Regelmäßige Code-Reviews und Testphasen, um sicherzustellen, dass die Qualitätsansprüche erfüllt werden
- Endabnahme durch das Projektteam, um sicherzustellen, dass die definierten Qualitätsstandards erfüllt sind, bevor die Anwendung in den produktiven Einsatz geht

Mitarbeiterverwaltungssystem

9 Projektentwicklung

Die Entwicklung dieses Projekts erfolgt gemäß einem sorgfältig geplanten Zeitplan und einer klaren Organisationsstruktur. Im Folgenden werden die Schritte beschrieben, die für die Projektentwicklung geplant sind, sowie die Nutzung eines Triggers für die automatische Berechnung von Alter und Rentenbeginn aus dem Geburtsdatum.

9.1 Zeitplan

Die Entwicklung des Projekts erfolgt in mehreren definierten Phasen:

Planungsphase: Hier werden die Anforderungen erfasst, der Umfang des Projekts festgelegt und die Datenbankstruktur entworfen. Diese Phase dauert voraussichtlich 2 Wochen.

Implementierungsphase: In dieser Phase wird die eigentliche Anwendung entwickelt und die Datenbankstruktur implementiert. Die Funktionen für die Benutzerregistrierung, Authentifizierung und Mitarbeiterverwaltung werden erstellt. Die Implementierung ist für 4 Wochen geplant.

Testphase: Die fertige Anwendung wird umfassend getestet, um sicherzustellen, dass sie den Anforderungen entspricht und fehlerfrei funktioniert. Diese Testphase erstreckt sich über 2 Wochen.

Optimierungsphase: Basierend auf den Testergebnissen werden eventuelle Fehler behoben und Verbesserungen vorgenommen, um die Leistung und Benutzerfreundlichkeit der Anwendung zu optimieren. Diese Phase ist für eine Woche geplant.

9.2 Datenbankstruktur- und Trigger

Die Datenbankstruktur umfasst zwei Haupttabellen:

Mitarbeitertabelle: Hier werden alle relevanten Informationen zu den Mitarbeitern gespeichert. Die Spalte „ID“ ist Primary Key, d.h. alle Mitarbeiter werden anhand ihrer einzigartigen ID identifiziert. Die Nummerierung erfolgt automatisch in 2er Schritten, beginnend ab der Zahl „100“.

Benutzer (dbo.Benutzer): Diese Tabelle speichert die Anmeldeinformationen der Benutzer, darunter Benutzername, Passwort-Hash und Salt.

Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
Vorname	nvarchar(50)	<input checked="" type="checkbox"/>
Nachname	nvarchar(50)	<input checked="" type="checkbox"/>
Strasse	nvarchar(50)	<input checked="" type="checkbox"/>
Nr	nvarchar(5)	<input checked="" type="checkbox"/>
PLZ	nvarchar(10)	<input checked="" type="checkbox"/>
Ort	nvarchar(20)	<input checked="" type="checkbox"/>
Telefon	nvarchar(50)	<input checked="" type="checkbox"/>
[E-Mail]	nvarchar(50)	<input checked="" type="checkbox"/>
Position	nvarchar(50)	<input checked="" type="checkbox"/>
EintrittDatum	date	<input checked="" type="checkbox"/>
Gehalt	int	<input checked="" type="checkbox"/>
Rentenbeginn	date	<input checked="" type="checkbox"/>
Geburtsdatum	date	<input checked="" type="checkbox"/>
[Alter]	int	<input checked="" type="checkbox"/>
Geschlecht	nvarchar(15)	<input checked="" type="checkbox"/>

Abbildung 1 dbo.Mitarbeiter

Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
Benutzername	nvarchar(255)	<input type="checkbox"/>
PasswortHash	nvarchar(64)	<input type="checkbox"/>
Salt	nvarchar(64)	<input type="checkbox"/>
AktuelleRolle	nvarchar(50)	<input checked="" type="checkbox"/>

Abbildung 2 dbo.Benutzer

Zusätzlich zu den Tabellen wird ein Trigger namens "tr_calcAgeAndRetirement" implementiert. Dieser Trigger wird jedes Mal ausgelöst, wenn ein neuer Datensatz in die Mitarbeiter-Tabelle eingefügt oder aktualisiert wird. Er berechnet automatisch das Alter des Mitarbeiters anhand seines Geburtsdatums und den Rentenbeginn basierend auf geltenden Regelungen. Diese automatische Berechnung erleichtert die Verwaltung der Datenbank und stellt sicher, dass die relevanten Informationen stets aktuell sind.

```

CREATE TRIGGER tr_calcAgeAndRetirement
ON dbo.Mitarbeiter
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    -- Aktualisierung des Alters und des Rentenbeginns basierend auf den geänderten Geburtsdaten
    UPDATE m
    SET
        [Alter] = DATEDIFF(YEAR, i.Geburtsdatum, GETDATE()) -
            CASE WHEN MONTH(i.Geburtsdatum) > MONTH(GETDATE()) OR
                (MONTH(i.Geburtsdatum) = MONTH(GETDATE()) AND DAY(i.Geburtsdatum) > DAY(GETDATE()))
            THEN 1
            ELSE 0
        END,
        Rentenbeginn = DATEADD(DAY, 22, DATEADD(YEAR, 67, i.Geburtsdatum))
    FROM dbo.Mitarbeiter m
    INNER JOIN inserted i ON m.ID = i.ID;
END;
    
```

9.3 Hauptanwendung

Anzahl Markierte Datensätze

Liste aktualisieren

Mitarbeiter anlegen

Import / Export-Funktion

Mitarbeiter bearbeiten

Mitarbeiter entfernen

Import

Export

Mitarbeiter hinzufügen

Ausgewählt: 3

ID	Vorname	Nachname	Geschlecht	Strasse	Nr	PLZ	Ort	Telefon	Email	Position	Eintrittsdatum	Gehalt	Rentenbeginn	Geburtsdatum	Alter
100	Max	Mustermann	männlich	Musterstrasse	12	12345	Musterstadt	123456789	max@example.com	Mitarbeiter	01.01.2022	3000	23.01.2057	01.01.1990	34
102	Ahmet	Cakir	männlich	Kaulbachweg	10	74074	Heilbronn	504991	ack@yahoo.com	Abteilungsleiter	01.01.2022	30000	01.07.2063	09.06.1996	27
104	Max	Schuhmacher	männlich	Musterstrasse	120	1234445	Musterstadt	1236789	max@example.com	Mitarbeiter	01.01.2022	3000	23.01.2057	01.01.1990	34
106	Ahmet	Berndt	männlich	Kaulbachweg	11	74074	Heilbronn	504991	ack@yahoo.com	Abteilungsleiter	01.01.2022	30000	01.07.2063	09.06.1996	27
108	Max	Mustermann	männlich	Musterstrasse	12	12345	Musterstadt	123456789	max@example.com	Mitarbeiter	01.01.2022	3000	23.01.2057	01.01.1990	34
110	Ahmet	Cakir	männlich	Kaulbachweg	10	74074	Heilbronn	504991	ack@yahoo.com	Abteilungsleiter	01.01.2022	30000	01.07.2063	09.06.1996	27
112	Max	Schuhmacher	männlich	Musterstrasse	120	1234445	Musterstadt	1236789	max@example.com	Mitarbeiter	01.01.2022	3000	23.01.2057	01.01.1990	34
114	Ahmet	Berndt	männlich	Kaulbachweg	11	74074	Heilbronn	504991	ack@yahoo.com	Abteilungsleiter	01.01.2022	30000	01.07.2063	09.06.1996	27
116	Max	Mustermann	männlich	Musterstrasse	12	12345	Musterstadt	123456789	max@example.com	Mitarbeiter	01.01.2022	3000	23.01.2057	01.01.1990	34
118	Max	Schuhmacher	männlich	Musterstrasse	120	1234445	Musterstadt	1236789	max@example.com	Mitarbeiter	01.01.2022	3000	23.01.2057	01.01.1990	34

Anzahl Datensätze: 10

Gesamtmitarbeiteranzahl

Mitarbeiterliste

Ansicht: Mitarbeiter bearbeiten

Import Export

Entfernen Bearbeiten Mitarbeiter hinzufügen

Profil bearbeiten

Persönlich

Vorname Geburtsdatum

Nachname Telefon

E-Mail Geschlecht

Adresse

Strasse Hausnummer

PLZ Ort

Beruflich

Eintritt

Position

Gehalt

Speichern

Mitarbeiterliste -> Export -> Textdokument

Mitarbeiterliste.txt

Datei Bearbeiten Ansicht

ID	Vorname	Nachname	Strasse	Nr	PLZ	Ort	Telefon	E-Mail	Position	EintrittDatum	Gehalt	Rentenbeginn	Geburtsdatum	Alter	Geschlecht
100	Max	Mustermann	Musterstrasse	12	12345	Musterstadt	123456789	max@example.com	Mitarbeiter	2022-01-01	3000	2057-01-23	1990-01-01	34	männlich
102	Ahmet	Cakir	Kaulbachweg	10	74074	Heilbronn	504991	ack@yahoo.com	Abteilungsleiter	2022-01-01	30000	2063-07-01	1996-06-09	27	männlich
104	Max	Schuhmacher	Musterstrasse	120	1234445	Musterstadt	1236789	max@example.com	Mitarbeiter	2022-01-01	3000	2057-01-23	1990-01-01	34	männlich
106	Ahmet	Berndt	Kaulbachweg	11	74074	Heilbronn	504991	ack@yahoo.com	Abteilungsleiter	2022-01-01	30000	2063-07-01	1996-06-09	27	männlich
108	Max	Mustermann	Musterstrasse	12	12345	Musterstadt	123456789	max@example.com	Mitarbeiter	2022-01-01	3000	2057-01-23	1990-01-01	34	männlich
110	Ahmet	Cakir	Kaulbachweg	10	74074	Heilbronn	504991	ack@yahoo.com	Abteilungsleiter	2022-01-01	30000	2063-07-01	1996-06-09	27	männlich
112	Max	Schuhmacher	Musterstrasse	120	1234445	Musterstadt	1236789	max@example.com	Mitarbeiter	2022-01-01	3000	2057-01-23	1990-01-01	34	männlich
114	Ahmet	Berndt	Kaulbachweg	11	74074	Heilbronn	504991	ack@yahoo.com	Abteilungsleiter	2022-01-01	30000	2063-07-01	1996-06-09	27	männlich
116	Max	Mustermann	Musterstrasse	12	12345	Musterstadt	123456789	max@example.com	Mitarbeiter	2022-01-01	3000	2057-01-23	1990-01-01	34	männlich
118	Max	Schuhmacher	Musterstrasse	120	1234445	Musterstadt	1236789	max@example.com	Mitarbeiter	2022-01-01	3000	2057-01-23	1990-01-01	34	männlich

Ze 13, Sp 1 2.628 Zeichen 100% Windows (CRLF) UTF-8

9.4 Profil anlegen

Mitarbeiterverwaltungssystem v1.0

Sie haben noch kein Profil? [Dann klicke hier](#)

Profil erstellen

Benutzername

Passwort ☐ Passwort anzeigen

Passwort bestätigen

Erstellen

9.5 Loginmaske

Mitarbeiterverwaltungssystem v1.0

Sie haben noch kein Profil? [Dann klicke hier](#)

Benutzername

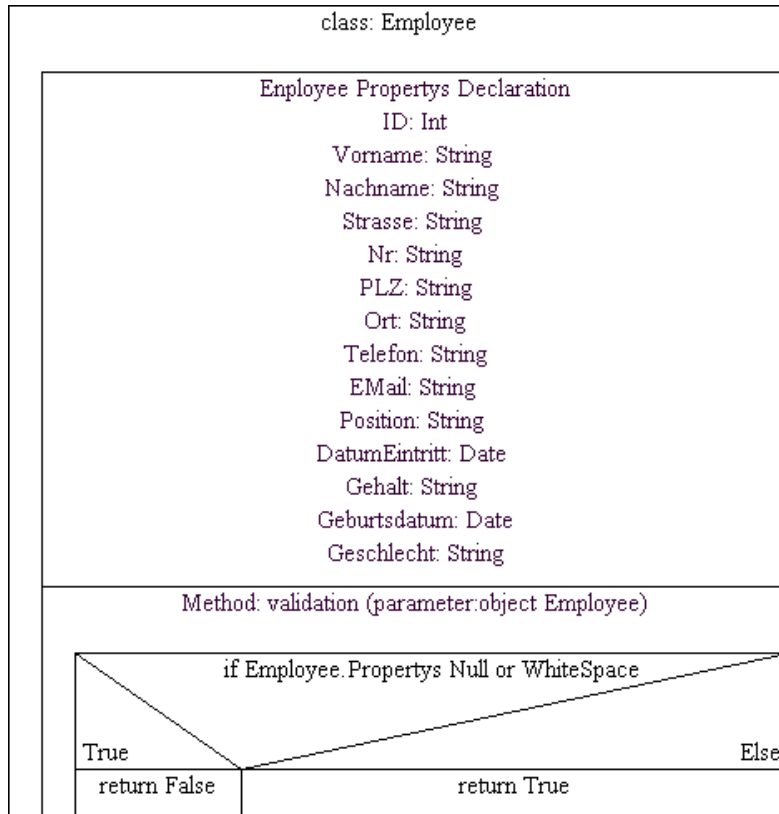
Passwort

Login

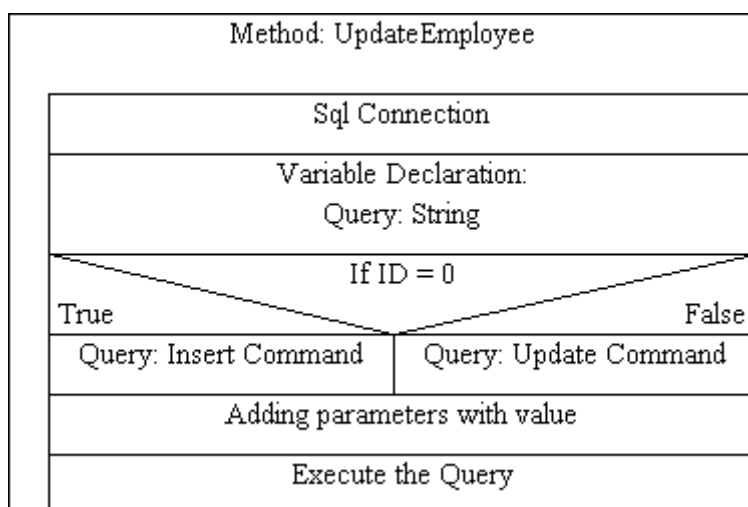
Mitarbeiterverwaltungssystem

Struktogramme

Methode: Validation

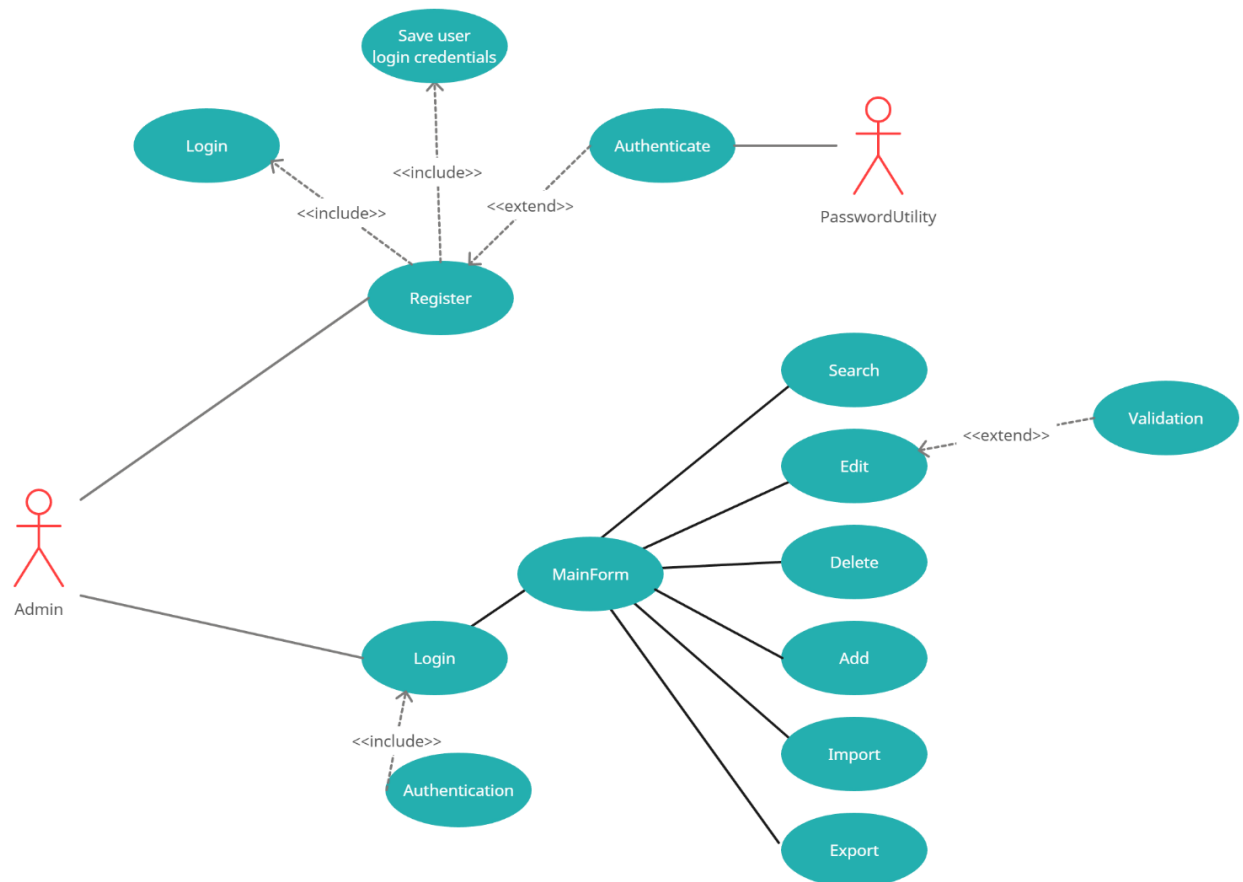


Methode: UpdateEmployee



Mitarbeiterverwaltungssystem

Anwendungsfall (Use-Case)



Gantt-Diagramm

