HOCHSCHULE
MITTWEIDA
UNIVERSITY OF
APPLIED SCIENCES

# Bachelorarbeit

**Herr ➢**
**Eric Heising <**

## Entwicklung einer Software zur Verwaltung und Dokumentation von Zugriffsrechten auf Basis rollenbasierter Zugriffskontrolle

Mittweida, 2022

Fakultät Angewandte Computer- und Biowissenschaften

# BACHELORARBEIT

# Entwicklung einer Software zur Verwaltung und Dokumentation von Zugriffsrechten auf Basis rollenbasierter Zugriffskontrolle

Autor:
**Herr**

**Eric Heising**

Studiengang:
**Angewandte Informatik - IT-Sicherheit**

Seminargruppe:
**IF15wI-B**

Erstprüfer:
**Prof. Ronny Bodach**

Zweitprüfer:
**M. Sc. Dipl.-Inf. Knut Altroggen**

Einreichung:
**Mittweida, 23.05.2022**

Verteidigung/Bewertung:
**Mittweida, 2022**

Faculty Applied Computer Sciences &
Biosciences

# BACHELORTHESIS

# Development of a software for administration and documentation of access rights based on role-based access control

author:
**Mr.**

**Eric Heising**

course of studies:
**Applied computer science - IT-security**

seminar group:
**IF15wI-B**

first examiner:
**Prof. Ronny Bodach**

second examiner:
**M. Sc. Dipl.-Inf. Knut Altroggen**

submission:
**Mittweida, 23.05.2022**

defence/ evaluation:
**Mittweida, 2022**

**Bibliografische Beschreibung:**

Heising, Eric:
Entwicklung einer Software zur Verwaltung und Dokumentation von Zugriffsrechten auf Basis rollenbasierter Zugriffskontrolle. - 2022. - 4, 57, 0 S.
Mittweida, Hochschule Mittweida, Fakultät Angewandte Computer- und Biowissenschaften, Bachelorarbeit, 2022

**Kurzreferat:**

Diese Arbeit behandelt die Entwicklung einer Software zur Verwaltung und Dokumentation von Zugriffsrechten. Dafür wird gezeigt, wofür eine solche Verwaltung und Dokumentation von Zugriffen benötigt wird und wie diese mit Hilfe einer geeigneten Software umgesetzt werden kann. Es werden wichtige Anforderungen erläutert und die entwickelte Software hinsichtlich Aufbau und wesentlichen Funktionen beschrieben.

# Inhaltsverzeichnis

# Tabellenverzeichnis

# Abbildungsverzeichnis

# Glossar

| Begriff | Erklärung |
|---|---|
| ArrayList | Ein in Java genutztes Konstrukt zum dynamischen Erstellen von Listen zur Speicherung von Objekten. |
| Authentifizierungs-verfahren | Das Verfahren prüft, ob ein Benutzer Zugriff auf eine System erhält. Der Begriff wird vereinfacht genutzt als Zusammenfassung von Authentisierung, Authentifizierung und Autorisierung. |
| CSV-Datei | Der Begriff steht für „comma-separated values“, einem Dateiformat, bei dem einzelne Werte mit Komma getrennt hintereinander geschrieben werden. |
| Dokumentationsdaten | Dazu zählen alle Daten die aus einer Anwendung in eine externe Datei (z.B. CSV-Datei) exportiert werden können. |
| JLabel | Ein in Java genutztes Konstrukt zur grafischen Ausgabe von Texten. |
| JList | Ein in Java genutztes Konstrukt zur grafischen Ausgabe und Auswahl von Objekten in einer Liste. |
| JPanel | Dient in Java als Komponente auf der alle grafischen Elemente (z.B. JLabel, JList, JTextField) liegen. |
| JScrollPane | Ein in Java genutztes Konstrukt um eine Liste mit Bildlaufleisten zu erhalten. Kann mit einer JList verbunden werden. |
| JTextField | Ein in Java genutztes Konstrukt zur Eingabe von Text. |
| Provisionierung | Bezeichnet die Vergabe von Zugriffsrechten. |
| PrintWriter | Ein in Java genutztes Konstrukt um Daten aus dem Programm in eine externe Datei zu schreiben. |
| Pseudocode | Beispielcode der verwendet wird, um einen Ablauf logisch, mittels vereinfacht dargestellter und unvollständiger Programmfunktionen zu veranschaulichen. |
| Reportdatei | Eine aus Dokumentationsdaten bestehende exportierte Datei. |
| Security-Kernel | Steuert den Zugriff auf ein System oder Daten mit Hilfe von speziell definierten Regeln. |

# 1. Einführung

Die fortschreitende Digitalisierung in der heutigen Zeit hat dafür gesorgt, dass immer mehr Arbeitsprozesse in Unternehmen digital durchgeführt werden. Das Versenden und Empfangen von E-Mails, das Bearbeiten von Dokumenten oder die Nutzung von verschiedensten Softwarelösungen zählen zu den Standardtätigkeiten, die in nahezu jeder Firma genutzt werden. Der Beginn der Corona Krise 2020 und die damit einhergehende verstärkte Nutzung von Homeoffice-Angeboten haben diesen digitalen Arbeitswandel noch weiter vorangetrieben und viele Firmen dazu genötigt, noch verstärkter in ein digitales Arbeitsumfeld zu wechseln.

Die heutigen, fast ausschließlich digital durchgeführten Arbeiten in den Unternehmen sorgen damit auch für erhöhte Anforderungen an die Informationssicherheit. Erhöhte Datenschutzanforderungen, Absicherungen vor IT-Angriffen oder das Verhindern von internen Datendiebstählen und Datenmissbräuchen rückt immer mehr in den unternehmerischen Fokus. Dabei gelten vor allem die eigenen Mitarbeiter häufig zu den größten Risikofaktoren, wenn es um das Erreichen von Informationssicherheitszielen geht.

In einem Monitoringbericht zeigt das Bundeskriminalamt, welchen Schaden eigene Mitarbeiter in Unternehmen anrichten können und wie hoch der Anteil von Innentätern bei Sicherheitsvorfällen in Unternehmen ist. Auf die Frage, ob „Unternehmen innerhalb der letzten 2 Jahre von Datendiebstahl, Industriespionage oder Sabotage betroffen“ (Bundeskriminalamt, 2020, https://www.bka.de/SharedDocs/Downloads/DE/Publikationen/Publikationsreihen/Forschungsergebnisse/2020KKFAktuell_InnentaeterinUnternehmen.pdf?__blob=publicationFile&v=2) waren, ergab sich folgende Auswertung:

- „3 von 4 Unternehmen wurden Opfer von Sabotage, Datendiebstahl oder Spionage (75%). 13% vermuteten betroffen gewesen zu sein (2017: 53% betroffene Unternehmen und 26% vermutlich betroffene, 2015: 51% betroffene Unternehmen und 28% vermutlich betroffene) (Bundeskriminalamt, 2020, ebd.).

Weiterhin wird gezeigt, dass: „Etwa ein Drittel der Betroffenen (33%) sagt, dass sie von früheren Mitarbeitern vorsätzlich geschädigt wurden. Ein knappes Viertel (23%) sieht vormals Beschäftigte in der Verantwortung, ohne ihnen ein absichtliches Fehlverhalten zu unterstellen“ (Bundeskriminalamt, 2020, ebd.).

Hieraus wird deutlich, wie groß der Anteil von internen Sicherheitsvorfällen durch Mitarbeiter ist und wie wichtig es ist, geeignete Risikominimierungsmaßnahmen durchzuführen. Dabei gelten vor allem effektive Zugriffskontrollverfahren und eine genaue Dokumentation von Mitarbeiterzugriffen als wichtiger Bestandteil, um Sicherheitsvorfälle zu minimieren.

Damit die Mitarbeiter einer Firma bspw. Daten einsehen und bearbeiten können sowie verschiedenste Softwaretools nutzen können, müssen ihnen die entsprechenden Zugriffsmöglichkeiten zur Verfügung stehen. Nur wenn ein Mitarbeiter Zugriffe auf die Daten und Anwendungen hat, die er für die Ausübung seiner Tätigkeit benötigt, kann ein

produktiver Arbeitsablauf garantiert werden. Leider stellen erteilte Zugriffe auch immer ein erhöhtes Risiko für eine Firma dar. Ein Mitarbeiter, der bspw. Zugriff auf ein firmeninternes Laufwerk hat und damit wichtige Daten einsehen und bearbeiten kann, hat theoretisch auch die Möglichkeit, diese Daten zu manipulieren, zu stehlen oder zu löschen. Um diese Risiken zu minimieren, müssen geeignete Prozesse existieren, die eine genaue Dokumentation von erteilten Zugriffen ermöglichen. Dazu zählt zum einen die genaue Auflistung aller genutzten Anwendungen und Daten, die für alle Firmenprozesse relevant sind, sowie die Auflistung aller Mitarbeiter und derer erteilten Zugriffsberechtigungen auf genau diese Daten. Dies alles wird im Rahmen des Risikomanagements in einem Unternehmen realisiert. Um den Anforderungen an das Risikomanagement und der sich daraus ergebenden Dokumentation von Zugriffsberechtigungen gerecht zu werden, können sogenannte IAM-Tools, wie Identity Repositories, helfen eine vollständige Auflistung von zugriffsrelevanten Daten und Systemen zu erstellen.

Ein Identity Repository kann alle Systeme und Daten des Unternehmens abbilden, in denen Zugriffe vergeben werden. Daraus ergibt sich die Möglichkeit, Mitarbeiter und Zugriffsrechte miteinander zu verbinden, um eine Übersicht zu schaffen, welcher Mitarbeiter welche Zugriffsrechte hat und somit auf bestimmte Daten zugreifen kann. Um diese Verbindungen zu realisieren, müssen die Mitarbeiterdaten mit den Zugriffsrechten zentral im Identity Repository aufrufbar und verknüpfbar sein. Dann können einzelnen Mitarbeitern mittels dieser Software gezielt Zugriffsrechte vergeben und entzogen werden. So kann theoretisch zu jedem Zeitpunkt sichergestellt werden, welcher Benutzer welche Zugriffsrechte hat und welche Risiken sich ggf. daraus ergeben.

## 1.1. Zielsetzung

Um Mitarbeiter und deren Zugriffsberechtigungen zentral dokumentieren und verwalten zu können, wird eine Softwareapplikation benötigt, die eine übersichtliche Darstellung von Benutzerdaten, Anwendungen und Systemen sowie sich daraus ergebenen Zugriffsberechtigungen ermöglicht. Die Software muss die Verknüpfung von Mitarbeitern und deren Zugriffsrechten ermöglichen und Reports zur Verfügung stellen, um Daten aus dem Programm heraus exportieren zu können.

Diese Arbeit beschäftigt sich mit der Entwicklung einer solchen Software. Hierfür wird eine den Anforderungen entsprechende Applikation in Form eines Identity Repository entwickelt und im Folgenden hinsichtlich Aufbau und Funktionalität beschrieben. Die gezeigte Softwarelösung bezieht sich dabei ausschließlich auf rollenbasierte Zugriffskontrolle, da diese Form der Zugriffskontrolle in einer Vielzahl von Unternehmen genutzt wird und einige Vorteile gegenüber anderen Verfahren bietet (Kapitel 2, Seite 20). Diese Form der Zugriffsvergabe wird als „Best-Practice-Verfahren“ (IONOS, 2020, https://www.ionos.de/digitalguide/server/sicherheit/was-ist-role-based-access-control-rbac/) eingestuft sowie als das „beste Verfahren zur Verwaltung von Zugriffsberechtigungen“ (IONOS, 2020, ebd.) beschrieben.

Die entwickelte Softwarelösung zeigt, wie dieses Zugriffsverfahren mittels Implementierung von Berechtigungskonzepten eine Einbindung von Zugriffsrechten erlaubt und eine anschließende Zugriffsrechtevergabe an einzelne Mitarbeiter ermöglicht.

### 1.2. Aufbau der Arbeit

In Kapitel 2 werden alle Grundlagen beschrieben, die zum allgemeinen Verständnis des Themas benötigt werden. Hierzu wird ein Überblick zum Risikomanagement und Identity und Access Management gegeben. Zudem werden Begrifflichkeiten wie Zugriff und Identity Repository erläutert und verschiedene Zugriffsverfahren vorgestellt.

Kapitel 3 befasst sich mit den Anforderungen, die an eine Software zur Verwaltung und Dokumentation von Zugriffsrechten gestellt werden. Hierzu wird zuerst der Nutzen einer solchen Anwendung und aktuelle und potenzielle Einsatzmöglichkeiten beschrieben. Außerdem wird dargestellt, welche theoretischen Dokumentationsmöglichkeiten es gibt und wie diese mit dem Einsatz eines Identity Repository in Verbindung stehen. Anschließend werden die technischen Anforderungen an die Software erläutert und wie das Einbinden von Daten in die Anwendung aussehen kann.

In Kapitel 4 folgt die Vorstellung der entwickelten Software hinsichtlich Aufbau und Funktionsweise. Dafür werden technische Grundlagen, die Benutzeroberfläche und die wichtigsten implementierten Klassen beschrieben. Zudem wird gezeigt wie Daten in der Anwendung verarbeitet werden.

Kapitel 5 beschreibt die implementierte Reportfunktion der Anwendung zum Erstellen von Output-Files. Der Abschluss der Arbeit erfolgt mit der Zusammenfassung in Kapitel 6.

## 2. Grundlagen

In diesem Kapitel sollen grundlegende Begrifflichkeiten und deren Zusammenhänge erläutert werden, die sich aus der vorangegangenen Fragestellung ergeben. Um den Nutzen und die Funktion eines Identity Repositories zur Verwaltung und Dokumentation von Zugriffsberechtigungen zu verstehen, muss zuerst abgegrenzt werden, in welche Bereiche der Informationssicherheit die Thematik eingeordnet wird und welche Begriffe dabei häufig genutzt werden. Dazu muss z.B. definiert werden, was man unter einem Zugriff versteht und wie er mit Bereichen wie Risikomanagement und Identity und Access Management in Zusammenhang steht.

## 2.1. Risikomanagement

Um zu verstehen, worum es sich beim Risikomanagement und der sich daraus ergebenden Risikobewältigung handelt wird zuerst der Begriff des *Risikos* definiert.

Ein *Risiko* bezeichnet alle Ereignisse, die zukünftig eintreten und die Unversehrtheit eines Unternehmens beeinflussen können. Es bezeichnet alle theoretisch eintretbaren Schadensfälle und deren direkte Auswirkungen auf das Unternehmen. Das Unternehmen, welches entsprechenden Risiken ausgesetzt ist, wird dabei als Risikoträger bezeichnet. Risiken beeinflussen die Arbeit eines Unternehmens dahingehend, dass versucht, wird deren Eintrittswahrscheinlichkeit mittels geeigneter Risikobewältigungsmaßnahmen so gering wie möglich zu halten. Jedes eingetretene Risiko kann je nach Schwere für erhebliche wirtschaftliche Einbußen sorgen oder das Image eines Unternehmens negativ beeinflussen. Durch die verstärkte Nutzung von IT-Systemen und deren immer komplexeren Strukturen ergeben sich vermehrt Risiken, die mittels geeigneter Vorgänge analysiert und minimiert werden müssen.

Mit der Identifizierung, Analyse und Bewältigung solcher Risiken befasst sich der Bereich des Risikomanagements. Dabei wird versucht, sich einen Überblick über alle möglichen Risiken zu verschaffen und diese nach ihrer Kritikalität einzuordnen. Anhand dieser Analyse wird versucht geeignete Maßnahmen für einzelne Risiken zu definieren. Risiken, deren Eintrittswahrscheinlichkeit und Schadensauswirkung als sehr gering eingestuft werden, können als weniger relevant angesehen werden als solche mit sehr hohen Schadensauswirkungen und Eintrittswahrscheinlichkeiten. Diese Einordnung wirkt sich wiederum auf den weiteren Umgang mit den identifizierten Risiken aus.

Um einen detaillierten Überblick über das Risikomanagement zu geben, werden die einzelnen Phasen des Risikomanagements im folgenden Abschnitt wie folgt definiert und beschrieben:

- Phase 1 – Risikoidentifizierung
- Phase 2 – Risikoanalyse
- Phase 3 – Risikobewältigung

### Risikoidentifizierung

Die erste Phase im Risikomanagement bildet die Risikoidentifizierung. In dieser Phase wird versucht, sich einen detaillierten Überblick über mögliche Risiken zu verschaffen, die im Unternehmen oder einzelnen Unternehmensbereichen auftreten können. Hierfür werden alle für den Unternehmensbereich relevanten Bedrohungen benannt und aufgelistet. Es soll eine weitreichende Übersicht aller möglichen Schadensereignisse erstellt werden, um diese im weiteren Verlauf genauer zu analysieren.

Diese Phase bildet somit die Basis für den gesamten weiteren Risikomanagementprozess. Gerade im Bereich der Informationstechnik ist eine genaue Risikoanalyse sinnvoll, da es hier zu starken Auswirkungen im Unternehmen kommen kann.

Im Folgenden sind einige Beispiele für identifizierte Risiken aufgelistet, welche die Informationssicherheit bedrohen können. Tabelle 1 zeigt, wie mögliche Risiken aufgelistet und anhand ihrer möglichen Schadensauswirkung eingeordnet werden können:

| **Identifiziertes Risiko** | **Mögliche Schadensauswirkung** |
|---|---|
| 1. Unautorisierter Zugriff auf firmeninterne Laufwerke | Missbrauch oder Verlust von Firmendaten:<br>Wenn ein Mitarbeiter oder eine dritte Person Zugriff auf Daten auf dem Firmenlaufwerk hat, die für sie nicht vorgesehen sind, kann es zu Datenmissbrauch kommen. Auch können Daten absichtlich oder unabsichtlich verändert oder gelöscht werden. |
| 2. Nutzen von Endgeräten für private Zwecke | Datenschutzverstöße oder Download von Schadprogrammen:<br>Wenn Endgeräte für private Zwecke genutzt werden, besteht die Möglichkeit, dass firmeninterne Daten von unautorisierten Dritten (z.B. Freunde oder Familienmitglieder) eingesehen werden können. Auch können private Downloads dazu führen, Schadsoftware auf das Gerät zu laden und diese so ins Firmennetzwerk zu bringen. |
| 3. Öffnen von E-Mail Anhängen oder Phishing-Links | Einschleusen von Schadsoftware in das Firmennetzwerk:<br>Durch das Öffnen von E-Mail Anhängen oder Phishing-Links kann Schadsoftware – wie Ransomware – in das Firmennetzwerk gelangen und wichtige Firmendaten verschlüsseln oder ganze Firmenprozesse lahmlegen. |

*Tabelle 1: Identifizierte Risiken und deren potentielle Schadensauswirkungen (eigene Tabelle)*

**Risikoanalyse**

Nach dem in einem ersten Schritt potenzielle Risiken aufgelistet und grob eingeordnet worden sind, erfolgt nun eine detaillierte Risikoanalyse. Hierbei werden die zuvor aufgelisteten Risiken hinsichtlich ihrer Schadensauswirkungen auf die einzelnen Unternehmensbereiche und ihrer Eintrittswahrscheinlichkeiten eingeordnet. Nachdem die Risiken hinsichtlich ihrer Auswirkung und Eintrittswahrscheinlichkeit eingestuft worden sind, können sie genauer bewertet und priorisiert werden.

Als gängige Methode zum Einordnen von Risiken hat sich die *Risikomatrix* bewährt. Eine Risikomatrix ist eine visuelle Darstellung von identifizierten Risiken und stellt den Zusammenhang von Schadensauswirkung und Eintrittswahrscheinlichkeit grafisch dar. Diese grafische Einordnung hilft dabei, die Risiken anhand Ihrer Kritikalität zu beurteilen und im weiteren Risikomanagementprozess genauer zu bewerten.

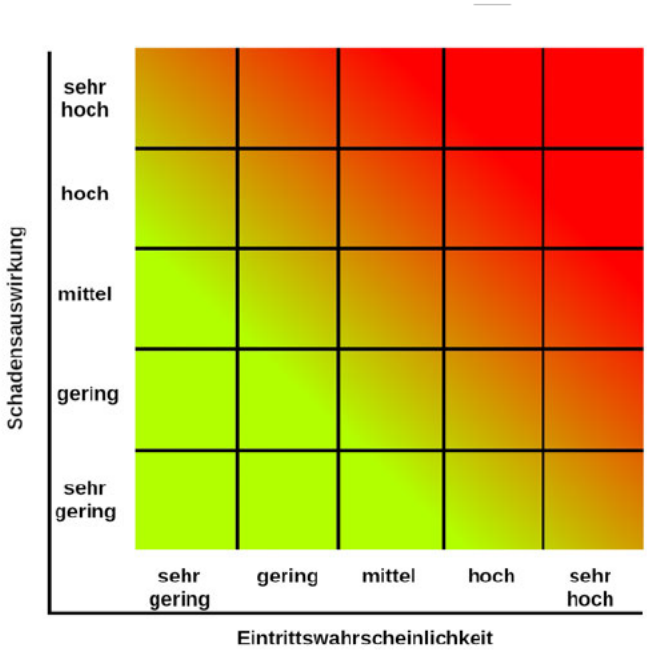


*Abbildung 1: Risikomatrix (eigene Abbildung)*

Die Risikomatrix aus Abbildung 1 ordnet die identifizierten Risiken aus Tabelle 1 beispielhaft nach ihrer Kritikalität ein. Entlang der x-Achse wird die Einordnung hinsichtlich der einzelnen eingeschätzten Eintrittswahrscheinlichkeiten vorgenommen, während die y-Achse die Einordnung bezüglich der Schadensauswirkungen symbolisiert. Je weiter rechts in der Matrix das Risiko eingeordnet wird, desto höher wird die Eintrittswahrscheinlichkeit angesehen. Je weiter oben die Einordnung eines Risikos

erfolgt, desto höher wird das Schadensausmaß bewertet. Damit lässt sich aussagen, dass Risiken, die gleichzeitig weit rechts und weit oben in der Matrix gelistet sind, als sehr kritisch angesehen werden müssen.

Für Risiko 3 (Öffnen von E-Mail Anhängen oder Phishing-Links) wird in der Matrix die Eintrittswahrscheinlichkeit als „hoch“ und das Schadensausmaß als „sehr hoch“ bewertet. Demnach kann dieses Risiko als kritisch angesehen werden und sollte in einer anschließenden Analyse genauer betrachtet werden. Dahingegen wird das Risiko 2 (Nutzen von Endgeräten für private Zwecke) mit seiner Einstufung als mittlere Eintrittswahrscheinlichkeit und geringem Schadensausmaß als weniger kritisch bewertet und bedarf im Vergleich zu Risiko 3, einer weniger ausführlichen Risikobewältigung.

**Risikobewältigung**

Unter der Risikobewältigung versteht man die Maßnahmen, die getroffen werden, um die zuvor identifizierten und analysierten Risiken zu minimieren, zu verlagern oder zu akzeptieren. Jedes Risiko bedarf eigener Maßnahmen und muss individuell betrachtet werden.

Bei der Risikominimierung steht das Senken der Eintrittswahrscheinlichkeit oder des Schadensausmaßes im Vordergrund. Hierfür können geeignete Sicherheitsmaßnahmen festgelegt werden, die helfen, die Kritikalität des Risikos zu senken. Im Beispiel von Risiko 3 (Öffnen von E-Mail Anhängen oder Phishing-Links) aus Tabelle 1 können z.B. Schulungen zur Sensibilisierung von Mitarbeitern als geeignetes Mittel zur Senkung der Eintrittswahrscheinlichkeit eingesetzt werden, was wiederum die Kritikalität verringert.

Die Risikoverlagerung ist eine weitere Möglichkeit, mit erkannten Risiken umzugehen. Dabei werden bestimmte Risiken auf Geschäftspartner oder Versicherungsgesellschaften übertragen, um eventuelle Ausfälle zu kompensieren. Da diese Form der Risikobewältigung aber meist mit erhöhten Kosten (z.B. für feste Versicherungszahlungen) einhergeht, muss geschaut werden, inwieweit der Kosten/Nutzen Faktor gegeben ist.

Schlussendlich bleibt die Risikoakzeptanz als letzte Form der Risikobewältigung. Ein Risiko wird akzeptiert, wenn es nach umfassender Analyse und Bewertung als tragbar angesehen wird. Für solche Risiken werden keine weiteren Maßnahmen durchgeführt. Das Risiko wird demnach als zu unwahrscheinlich angesehen oder dessen Schadensausmaß als zu gering bewertet, als das dafür weitere ggf. kostenintensive Maßnahmen ergriffen werden müssen.

## 2.2. Identity und Access Management

Identity und Access Management, kurz IAM, befasst sich mit der zentralen Verwaltung und Überwachung von Benutzeridentitäten und Zugriffsrechten. Diese zentrale Verwaltung ermöglicht es einem Unternehmen, jederzeit einen genauen Überblick über vorhandene Benutzeridentitäten (Mitarbeiter, externe Berater, Kunden etc.) und den damit verbundenen Zugriffsberechtigungen für Firmendaten und Anwendungen zu haben. Durch diese Verwaltungs- und Überwachungsfunktion von Benutzern und deren Zugriffen im Netzwerk deckt das IAM einen wichtigen Teil des Risikomanagements in einem Unternehmen ab. Nur wenn zu jedem Zeitpunkt klar ist, welche Person autorisierten Zugriff auf bestimmte Daten im Firmennetzwerk hat, können Sicherheitsrisiken in diesen Bereichen eingegrenzt und minimiert werden.

Um diesen Anforderungen gerecht zu werden, nutzt das IAM in der Regel verschiedene Softwareanwendungen und Tools, welche die Verwaltung, Vergabe oder Dokumentation von Benutzern und Zugriffsrechten vereinfachen sollen.

Folgende „Technologien des Identity und Access Managements“ (James A. Martin; Florian Maier, 2022, https://www.computerwoche.de/a/was-sie-ueber-iam-wissen-muessen) gelten dabei als relevante Mittel:

- „Passwort Management Tools
- „Provisioning Software“
- „Apps zur Durchsetzung von Security Policies“
- „Reporting und Monitoring Apps“
- „Identity Repositories“ (James A. Martin; Florian Maier, 2022, ebd.)

## 2.3. Identity Repository

Da sich der Kern dieser Arbeit um die Verwaltung und Dokumentation von Zugriffsrechten mittels einer geeigneten Softwarelösung dreht, soll im Folgenden der Begriff des Identity Repositoriy genauer erläutert werden..

Ein Repository ist ein „verwaltetes Verzeichnis zur Speicherung und Beschreibung digitaler Objekte für ein digitales Archiv.“ (Wikipedia, 16.02.2022, https://de.wikipedia.org/wiki/Repository). Es kann damit als eine Art Datenbank angesehen werden, welche die Auflistung und Bearbeitung von spezifischen Daten erlaubt. Ein Identity Repository bildet im IAM die zentrale Quelle von Benutzerdaten und Zugriffsrechten. Es ermöglicht unter anderem das Anlegen und Verwalten von Nutzerdaten (z.B. Mitarbeiterinformationen), das Implementieren von Benutzerrollen und Berechtigungskonzepten im Rahmen der rollenbasierten Zugriffskontrolle und die Verwaltung von Zugriffsrechten inklusive derer Zuordnung an Benutzer. Es stellt eine Softwareanwendung mit grafischer Oberfläche dar mittels derer relevante Daten eines Zugriffskontrollverfahrens initialisiert und verwaltet werden können.

Damit ist das Repository die zentrale Quelle, wenn es darum geht, sich eine Übersicht über alle vergebenen Zugriffsrechte zu verschaffen und kann als Basis für die spätere Provisionierung der erteilten Rechte in das System (z.B. per Active Directory unter Windows) dienen. Identity Repositories ermöglichen neben dem Verwalten von Benutzern, Berechtigungskonzepten oder Zugriffsrechten oft auch das Erstellen von Dokumentationsreports. Mittels dieser Reports lassen sich schnell strukturierte Übersichten im Bereich der Zugriffskontrolle erstellen.

Das kann z.B. eine Auflistung aller an einen bestimmten Mitarbeiter zugewiesenen Rechten inklusive des Zuweisungszeitpunktes oder von einem Mitarbeiter vergebene Zugriffe an andere Mitarbeiter sein. Damit lassen sich schnell gezielte Zugriffsinformationen herausfiltern, um diese genauer zu analysieren oder für Kontrollmaßnahmen zu nutzen.

## 2.4. Digitale Identität

Jede Person auf dieser Welt verfügt über eine einzigartige Identität. Diese Identität ist unverwechselbar und nur einmalig vorhanden. Sie setzt sich aus verschiedenen einzelnen Merkmalen zusammen und ergibt im Gesamtwert eine komplexe Merkmalsstruktur, anhand derer sich die Person eindeutig identifizieren lässt. Verschiedene persönliche Merkmale wie Name, Körpergröße, Geburtsdatum, Wohnort oder Haarfarbe ergeben zusammen eine einzigartige Identität.

Das System der Zusammenfassung einzelner Merkmale zu einer Identität lässt sich auch auf den digitalen Bereich abstrahieren, man spricht dann von einer digitalen Identität. Digitale Identitäten sind ähnlich wie persönliche Identitäten einzigartig und lassen sich aufgrund verschiedener Merkmale bestimmen. Statt Alter, Haarfarbe oder Wohnort werden hierbei digitale Merkmale zur Identifizierung einer Identität verwendet. Klassische Beispiele sind Benutzernamen, Passwörter und verschiedene weitere Authentifizierungsverfahren wie Fingerabdrücke oder Gesichtserkennung. Mit diesen Merkmalen lassen sich Benutzer digital identifizieren und authentifizieren. Solche Identitäten werden verwendet, um Zugriffe zu digitalen Systemen zu erhalten und sicherzustellen, dass nur autorisierte Benutzer Zugriff auf die entsprechenden System erhalten.

Diese Form der Identifizierung eines Benutzers auf ein System erfolgt in Form von Authentisierungs-, Authentifizierungs- und Autorisierungsmaßnahmen. Diese drei Schritte sind unmittelbar miteinander verbunden und ermöglichen einem Nutzer den Zugriff auf Daten. Bei der Authentisierung wird überprüft, ob ein Benutzer der ist, der er vorzugeben scheint. Damit dieser Nutzer Zugriff auf die gewünschten Daten erhält, muss er sich digital „ausweisen“. Dies geschieht in der Regel durch die Eingabe von Benutzernamen und Passwort. Wurden diese Daten eingeben, folgt die Authentifizierung. In diesem Schritt wird geprüft, ob die eingegebenen Daten korrekt sind und bspw. in der Datenbank des Zielsystems hinterlegt sind. Sind Benutzername und Passwort hinterlegt und in dieser Kombination richtig angegeben worden, erfolgt die

Autorisierung. Dieser letzte Schritt gewährt den eigentlichen Zugriff auf das Zielsystem. Ist die Autorisierung erfolgt, hat der Nutzer den Zugriff auf die Daten des Zielsystems erhalten und kann dort z.B. Dateien öffnen, Daten einsehen oder Anwendungen ausführen.

Im Bereich des IAM spielen digitale Identitäten und die damit verbundenen Authentifizierungsmaßnahmen eine zentrale Rolle. Jeder Benutzer, der sich im Firmennetzwerk bewegt, sollte eindeutig mittels seiner digitalen Identität (Benutzeraccount) identifizierbar sein. Dadurch kann nachvollzogen werden, um welche Person es sich handelt und ob diese Zugriff auf bestimmte Daten hat. Auch können anhand der Benutzeraccounts bestimmte Regeln für Datenzugriffe festgelegt werden. Durch das Zuweisen von Benutzern zu bestimmten Gruppen wird z.B. festgelegt auf welche Daten Benutzer dieser Gruppe Zugriff haben dürfen. Damit können genaue Regelungen im gesamten Netzwerk erstellt werden, um deren Verwaltung sich das Identity und Access Management kümmert.

## 2.5. Zugriffsrechte

Ein Zugriffsrecht bezeichnet eine Art Regel, die vorgibt, ob und wie ein Benutzer auf bestimmte Daten, Objekte oder Anwendungen in einem System zugreifen kann. Diese Regeln werden administrativ vergeben oder entzogen. Zugriffsrechte sind immer an eine digitale Benutzeridentität gebunden, für die sie die Regeln festsetzen. Ein Benutzer kann also nur die Daten einsehen oder bearbeiten, für die er die entsprechenden Zugriffsrechte hat. Die Zugriffsrechte bilden damit das zentrale Element der Rechtevergabe und der damit einhergehenden Zugriffskontrolle. In der Regel folgen die Zugriffsrechte dem sogenannten *CRUDE-Prinzip.* Das *CRUDE-Prinzip* beschreibt die Menge an Handlungen, die mit einem Zugriffsrecht durchgeführt werden können. CRUDE steht dabei als Abkürzung für die fünf Grundhandlungen der Zugriffsrechte. Diese Grundhandlungen sind in der nachfolgenden Tabelle 2 definiert:

| Zugriffshandlung | Beschreibung |
|---|---|
| **C**reate | Erstellen – es ist dem Benutzer erlaubt neue Daten anzulegen |
| **R**ead | Lesen – es ist dem Benutzer erlaubt bestehende Daten öffnen oder lesen zu können |
| **U**pdate | Bearbeiten – es ist dem Benutzer erlaubt bestehende Daten bearbeiten oder aktualisieren zu können |
| **D**elete | Löschen – es ist dem Benutzer erlaubt vorhandene Daten zu löschen |
| **E**xecute | Ausführen – es ist dem Benutzer erlaubt bestehende Daten oder Anwendungen zu öffnen bzw. auszuführen |

*Tabelle 2: Die Grundhandlungen des CRUDE-Prinzips (eigene Tabelle)*

Die Menge aller Zugriffsrechte, die ein Benutzer hat, bestimmt die Handlungen, die er in einem System durchführen kann. Ein Mitarbeiter einer Firma kann z.B. nur alle Dateien einsehen, für die er lesenden Zugriff erhalten hat. Möchte er weiterführende Zugriffsrechte erhalten, müssen diese im Bereich des IAM beantragt und geprüft werden. Die Möglichkeit, Zugriffsrechte zu vergeben und zu verwalten, wird über verschiedene Zugriffskontrollverfahren realisiert. Diese werden im anschließenden Absatz beschrieben.

## 2.6. Zugriffskontrollverfahren

Die Zugriffskontrolle ist der Grundstein des Risikomanagements in einem Unternehmen. Da Mitarbeiter ständig Zugriffe auf Systeme und Daten benötigen, ist eine zentral geregelte Zugriffskontrolle unerlässlich, um Sicherheitsrisiken zu minimieren. Mittels Zugriffskontrolle wird der Zugriff eines Benutzers auf bestimmt Ressourcen in einem System geregelt. Sie steuert und kontrolliert diesen Zugriff mittels verschiedener Authentifizierungsverfahren. Im Fokus der Zugriffskontrolle steht das Einhalten der Schutzziele der Informationssicherheit: Authentizität, Integrität, Vertraulichkeit und Verfügbarkeit. Diese Schutzziele lassen sich wie folgt definieren:

1. Authentizität

„Unter der Authentizität eines Objekts bzw. Subjekts (engl. authenticity) wird die Echtheit und Glaubwürdigkeit des Objekts bzw. Subjekts verstanden, die anhand einer eindeutigen Identität und charakteristischen Eigenschaften überprüfbar sind“ (Tobias Scheible, 14.12.2013, https://scheible.it/it-sicherheit-grundlagen-schutzziele).

2. Integrität

„Ein System gewährleistet die Datenintegrität (engl. integrity), wenn es Subjekten nicht möglich ist, die zu schützenden Daten unautorisiert und unbemerkt zu manipulieren“ (Tobias Scheible, 14.12.2013, ebd.).

3. Vertraulichkeit

„Ein System gewährleistet die Informationsvertraulichkeit (engl. confidentiality), wenn es keine unautorisierte Informationsgewinnung ermöglicht [...] “ und „erfordert in datensicheren Systemen die Festlegung von Berechtigungen und Kontrollen der Art, dass sichergestellt ist, dass Subjekte nicht unautorisiert Kenntnis von Informationen erlangen“ (Tobias Scheible, 14.12.2013, ebd.).

4. Verfügbarkeit

„Ein System gewährleistet die Verfügbarkeit (engl. availability), wenn authentifizierte und autorisierte Subjekte in der Wahrnehmung ihrer Berechtigungen nicht unautorisiert beeinträchtigt werden können“ (Tobias Scheible, 14.12.2013, ebd.).

Um diesen Sicherheitszielen gerecht zu werden, kommen unterschiedliche Zugriffsverfahren zum Einsatz. Diese Verfahren steuern die Zugriffskontrolle mittels verschiedener systematischer Ansätze. Folgende Zugriffskontrollverfahren werden häufig eingesetzt:

- Mandatory Access Control (MAC)
- Discretionary Access Control (DAC)
- Attribute-based Access Control (ABAC)
- Role-based Access Control (RBAC)

**Mandatory Access Control (MAC)**

Die Mandatory Access Control kann als „Sicherheitsstrategie zur strikten Steuerung von Zugriffsrechten“ (Sharon Shea, 06.2016, https://www.computerweekly.com/de/definition/Mandatory-Access-Control-MAC) beschrieben werden. Einzelne Kriterien werden „durch den Systemadministrator festgelegt, durch das Betriebssystem oder einen Security-Kernel durchgesetzt und können nicht durch Endanwender manipuliert werden.“ Diese Form der Zugriffskontrolle kommt laut Shea vorrangig bei sicherheitskritischen Systeme

wie bei Regierungen oder militärischen Institutionen zum Einsatz. Die Einordnung von Benutzern, Endgeräten und den vorhandenen Ressourcen in Sicherheitsstufen wie „ „vertraulich", „geheim" und „streng geheim" " (Sharon Shea, 06.2016, ebd.) ermöglicht einen späteren Vergleich dieser Stufen zwischen Benutzern und Zugriffsobjekten. Zugriff wird z.B. durch das Betriebssystem nur auf solche Daten gewährt, die der Sicherheitsstufe des Benutzers entsprechen. Diese Form der Zugriffskontrolle gehört zwar zu den sichersten Zugriffssystemen, allerdings ergibt sich daraus auch eine „sehr sorgfältige Planung und ein fortwährendes Überprüfen der zugewiesenen Rechte" (Sharon Shea, 06.2016, ebd.).

**Discretionary Access Control (DAC)**

Bei der Discretionary Access Control, zu deutsch Diskretionäre Zugriffskontrolle („dem Ermessen des Agierenden […] überlassen (Wiktionary, 10.05.2020, https://de.wiktionary.org/wiki/diskretion%C3%A4r)) werden Objektzugriffe (z.B. Dateizugriffe) über Zugriffsrichtlinien bestimmt, die von den Objekteigentümern festgelegt werden.

In einem Online-Artikel von Techopedia wird diskretionär in diesem Zusammenhang beschrieben als: „the subject (owner) can transfer authenticated objects or information access to other users." (Techopedia, o.D., https://www.techopedia.com). Das bedeutet, dass der Objekteigentümer eigenständig Berechtigungen für seine Objekte an andere Benutzer übertragen kann und gleichzeitig bestimmt, welche Zugriffsmöglichkeiten der Nutzer auf das Objekt erhält. Die Authentifizierung der Benutzer beim Zugriff auf ein Objekt erfolgt dann z.B. über Benutzernamen und Passwort.

**Attribute-based Access Control (ABAC)**

Bei der attributbasierten Zugriffskontrolle werden Zugriffe anhand von Attributen erteilt, die ein Benutzer und das Objekt (Daten), auf das der Nutzer zugreifen möchte, besitzt. Bei dieser Form der Zugriffskontrolle werden die einzelnen Attribute miteinander verglichen und anhand von Übereinstimmungen Zugriffe gewährt oder verweigert. Attribute können IP-Adressen, Uhrzeit, Ort oder Unternehmensabteilung sein.

Ein Objekt erlaubt den Zugriff nur, wenn bestimmte Attribute vorhanden sind, z.B. darf ein Mitarbeiter der Finanzabteilung lesenden Zugriff auf das Objekt erhalten, wenn er zur offiziellen Arbeitszeit aus dem Firmennetzwerk heraus die Daten des Objektes abruft.

**Rollenbasierte Zugriffskontrolle (RBAC)**

Bei diesem Zugriffskontrollverfahren werden Zugriffe im Unterschied zur attributbasierten Zugriffskontrolle nicht durch Attribute von Benutzern und Objekten vergeben, sondern anhand von zuvor definierten Rollen. Benutzer bekommen je nach Abteilung oder Arbeitsaufgaben bestimmte Rollen zugewiesen, mit denen Sie alle Zugriffsrechte erhalten, die sie zum Ausüben Ihrer Tätigkeiten benötigen.

Da sich die in dieser Arbeit vorgestellte Softwarelösung auf dieses Zugriffsverfahren bezieht, soll dieses im folgenden Abschnitt ausführlicher beschrieben werden. Begriffe wie *Rolle* oder *rollenbasiertes Berechtigungskonzep*t müssen erläutert werden, um die rollenbasierte Zugriffskontrolle zu verstehen.

## 2.7. Rollen

Eine Rolle ist als eine Menge von Zugriffsrechten definiert. Rollen werden genutzt, um die Rechtevergabe an einzelne Benutzer zu erleichtern, da sie mehrere Zugriffsrechte unter einem Rollennamen vereinen. Statt der Zuweisung einzelner Zugriffsrechte an einen Benutzer werden diesem nur die einzelnen zuvor definierten Rollen zugewiesen. Damit erhält der Benutzer nach der Rollenzuweisung alle Rechte die der Rolle zugeordnet sind. Diese Art der Rechtezuweisung erleichtert den Rechtevergabeprozess, da jedem Benutzer nicht alle Zugriffsrechte einzeln vergeben oder entzogen werden müssen.

Im folgenden Beispiel wird gezeigt, wie eine Zusammenfassung von Zugriffsrechten zu Rollen aussehen kann. Dafür werden exemplarisch Zugriffsrechte von zwei Dateien beschrieben und wie diese zu jeweils einer Rolle mit unterschiedlicher Funktion zusammengeführt werden.

| **Datei** | **Zugriffsrecht** | **Beschreibung** |
|---|---|---|
| Finanzen.xlsx | Read | Lesender Zugriff auf die Excel-Datei |
| Finanzen.xlsx | Update | Die Excel Datei kann bearbeitet werden |
| Finanzen.xlsx | Delete | Die Excel Datei kann gelöscht werden |
| Mitarbeiterdaten.pdf | Read | Lesender Zugriff auf die PDF Datei |
| Mitarbeiterdaten.pdf | Delete | Die PDF Datei kann gelöscht werden |

*Tabelle 3: Zwei Dateien und deren mögliche Zugriffsrechte (eigene Tabelle)*

Tabelle 3 definiert zwei Dateien *Finanzen.xlsx* und *Mitarbeiterdaten.pdf*, wie sie in einem Unternehmen in unterschiedlichen Abteilungen vorkommen könnten. Die beiden Dateien haben unterschiedliche Zugriffsrechte und sollten nur von Mitarbeitern der jeweiligen Abteilungen (Finanzen und Personal) gelesen und bearbeitet werden können. Hierfür werden Rollen für die Abteilungen definiert, welche alle Zugriffsrechte der zugehörigen Dateien vereinen. Diese Rollen können dann an neue Mitarbeiter der Abteilung zugewiesen werden, damit diese Zugriff auf die Dateien erhalten. Folgende Rollen in Tabelle 4 werden dafür definiert:

| **Abteilung** | **Rolle** | **Zugewiesene Zugriffsrechte** |
|---|---|---|
| Finanzen | R.1. | Read, Update, Delete Finanzen.xlsx |
| Personal | R.2. | Read, Delete Mitarbeiterdaten.pdf |

*Tabelle 4: Definierte Rollen und deren zugewiesene Zugriffsrechte aus Tabelle 3 (eigene Tabelle)*

Dieses Beispiel zeigt, wie die schnelle Zuweisung der benötigten Zugriffsrechte an neue Mitarbeiter durch Rollen erfolgen kann. Wenn ein neuer Mitarbeiter der Finanzabteilung den vollen Zugriff auf die Datei „Finanzen.xlsx“ erhalten soll, muss ihm nun nur die Rolle R.1. zugewiesen werden.

Damit klar ist welche Rolle welche Zugriffsrechte beinhaltet und für welche Abteilungen oder Systeme diese Rollen benötigt werden, müssen diese eindeutig in einem zentralen Dokument definiert sein. Diese Definition der Rollen, inklusive der Systeme und Anwendungen, in denen sie zum tragen kommen, sowie die Zuweisung der Zugriffsrechte an die Rollen wird in rollenbasierten Berechtigungskonzepten beschrieben.

## 2.8. Rollenbasierte Berechtigungskonzepte

Rollenbasierte Berechtigungskonzepte stellen die Grundlage der Rechtevergabe in der rollenbasierten Zugriffskontrolle dar. Diese Konzepte enthalten alle Informationen über ein System, für das Zugriffsrechte vergeben werden können. Dafür werden Grundinformationen über das System, Verantwortlichkeiten, Sicherheitsrichtlinien, alle möglichen Zugriffsrechte, Rollen und deren gegenseitige Zuweisung detailliert erläutert. Folgende Informationen sollten in jedem Berechtigungskonzept enthalten sein:

### Grundlegende Informationen über das System

Um alle Zugriffsrechte auflisten zu können sowie dazugehörige Rollen zu definieren, muss zuerst klar sein, um was für ein System es sich handelt. Dateisysteme, einzelne Ordner oder genutzte Softwareanwendungen müssen erläutert werden, um eine Übersicht von genutzten Funktionen zu bekommen, für die später Zugriffsrechte benötigt werden.

### Verantwortlichkeiten

Jedes Berechtigungskonzept steht für ein System oder eine Anwendung, die Zugriffsrechte erlaubt. In einem Unternehmen sollte für jedes dieser Konzepte eine verantwortliche Person ausgewählt werden. Der Verantwortliche sollte gute Kenntnisse über das System haben und ist für die Aktualisierung des Berechtigungskonzeptes bei Änderungen im System verantwortlich.

### Zugriffsrechte

Im Konzept müssen alle möglichen Zugriffsrechte hinterlegt sein, die im späteren von Mitarbeitern beantragt und genutzt werden können. Nur diese Auflistung erlaubt eine zentrale Übersicht von vorhandenen Zugriffsmöglichkeiten auf das System und ist somit wichtiger Bestandteil für spätere Kontrollmöglichkeiten der Zugriffe im Bereich des IAM.

### Rollenstruktur

Die Rollen müssen so definiert sein, dass sie Zugriffsrechte sinnvoll und nach Vorgaben des Risikomanagements und IAM verbinden und die Vergabe der notwendigen Zugriffsrechte an die Mitarbeiter ermöglichen. Sie können z.B. nach Abteilung oder Aufgabenfeld der Mitarbeiter erstellt werden, um spätere differenzierte Zuweisungen von Rechten zu realisieren. Rollen sollten ggf. unterschiedliche Zugriffsrechte vereinen, um eine Trennung von Berechtigungen durch verschieden zuweisbare Rollen zu ermöglichen.

### Segregation of Duties

Segregation of Duties bezeichnet die Trennung von Aufgaben im Bereich des IAM und beschreibt die Verteilung von Zugriffen auf unterschiedliche Rollen, die in Kombination zu Sicherheitsrisiken führen könnten. Damit sollen mögliche Risiken minimiert werden. Ein Mitarbeiter der Finanzabteilung dessen Rollen es z.B. erlauben, Finanzmittel zu beantragen, sollte nicht gleichzeitig in dem Besitz der Rolle sein, welche die Genehmigung von Finanzmittel ermöglicht. Mögliche Konflikte die sich aus gewissen Rollenkombinationen ergeben, müssen im Konzept hinterlegt sein.

### Sicherheitsrichtlinien

Da Berechtigungskonzepte die absolute Grundlage für Rechte in einem System bilden, sollten hierbei auch mögliche Sicherheitsimplementierungen erwähnt werden. So können z.B. Authentifizierungsverfahren für ein System oder Passwortrichtlinien beschrieben werden, um Sicherheitsvorgaben für dieses System zentral zu dokumentieren.

Rollenbasierte Berechtigungskonzepte bilden alle wichtigen Eigenschaften eines Systems ab und bilden damit den Grundbaustein für die rollenbasierte Zugriffskontrolle. Die Nutzung von Rollen als Zusammenschluss von Zugriffsrechten und deren zentrale Hinterlegung in einem Berechtigungskonzept bildet das Hauptmerkmal dieser Form der Zugriffskontrolle. Daraus lassen sich einige Vorteile des Verfahrens ableiten:

**Geringerer Verwaltungsaufwand**

Durch die Zusammenfassung von Zugriffsrechten zu Rollen ergibt sich ein deutlich verringerter Verwaltungsaufwand, was die Vergabe und den Entzug von Zugriffsrechten angeht. Das Entfernen einer Rolle reicht aus, um alle damit verbundenen Zugriffsrechte zu entziehen. Das sorgt für eine schnellere Rollenverwaltung und verringert den administrativen Aufwand. Auch Mitarbeiter, die die Abteilung wechseln, können über einfache Rollenneuordnungen aktualisierte Zugriffsberechtigungen erhalten.

**Übersichtlichkeit**

Neben dem verringerten Verwaltungsaufwand ist die Ordnung von Zugriffsrechten zu Rollen auch deutlich übersichtlicher als alle Rechte einzeln zu hinterlegen. Soll ein Mitarbeiter z.B. Rechte für eine neue Anwendung erhalten, müssen lediglich die benötigten Rollen im Berechtigungskonzept herausgesucht und zugewiesen werden. Auch das Vorhandensein eines Konzeptes, in dem jederzeit Informationen zur Anwendung und zu den Zugriffen abgerufen und bearbeitet werden können, erleichtert die Arbeitsprozesse in einem Unternehmen.

**Erhöhte IT-Sicherheit**

Da Mitarbeiter nur Zugriffe auf die Daten haben, für die sie die jeweiligen Rollen besitzen, werden Sicherheitsrisiken minimiert. Durch die Einhaltung von *Segregation of Duties* können Mitarbeiter nur begrenzt in gewissen Bereichen bewusst oder unbewusst Daten manipulieren. So ist es nicht möglich, das ein Mitarbeiter der Finanzabteilung gleichzeitig Zahlungen anordnen und freigeben kann, wenn diese Aktionen per Segregation of Duties getrennt sind. Auch Angriffe von außen, z.B. durch die Übernahme eines Mitarbeiterkontos ermöglichen nur Zugriff auf die Daten, für die der Mitarbeiter Rollen und somit Rechte besitzt. Damit können schädliche Auswirkungen eingegrenzt werden.

# 3. Anforderungen

Damit Mitarbeiterdaten, Berechtigungskonzepte, Rollen und Zugriffsrechte zentral in einer Anwendung verwaltet werden können, muss diese Anwendung verschiedenen Anforderungen genügen. Diese Anforderungen beziehen sich u.a. auf die Handhabung des Programms, die Datenverarbeitung von implementierten Informationen, die einzelnen Funktionen zur Zuweisung von Rechten oder das Erstellen von Reports.

Bevor gezielte Anforderungen erläutert werden, muss zuerst festgestellt werden, wofür Funktionen, die diesen Anforderungen gerecht werden sollen, benötigt und eingesetzt werden und in welchen Bereichen das Nutzen von Dokumentationstools wie Identity Repositories überhaupt zum Tragen kommt.

## 3.1. Nutzen einer Software zur Verwaltung und Dokumentation von Zugriffsrechten

Softwareanwendungen, die sich mit der Verwaltung und Dokumentation von Zugriffsrechten befassen, erleichtern verschiedenste Arbeitsprozesse in einem Unternehmen und helfen den Anforderungen an das Risikomanagement im Bereich der Zugriffskontrolle gerecht zu werden. Im folgenden Kapitel sollen die wesentlichen Vorteile erläutert werden, welche den Einsatz einer Dokumentationssoftware bezüglich der Zugriffskontrolle betreffen. Neben den Vorteilen einer solchen Anwendung sollen auch rechtliche Vorgaben, die den Einsatz einer Verwaltungs- und Dokumentationssoftware fordern, betrachtet werden.

### Rechtliche Vorgaben

Um die Anforderungen des Risikomanagements zu erfüllen, existieren verschiedene Vorschriften, die den Umgang mit Zugriffsberechtigungen und dazugehörigen Informationen für bestimmte Unternehmensbereiche festlegen. Kreditinstitute stehen dabei besonders im Fokus von Aufsichtsbehörden.

Als Kreditinstitut zählen alle Firmen „die Bankgeschäfte gewerbsmäßig oder in einem Umfang betreiben, der einen in kaufmännischer Weise eingerichteten Geschäftsbetrieb erfordert“ (Prof. Dr. Oliver Budzinski, o.D., https://wirtschaftslexikon.gabler.de/definition/kreditinstitute-37317). Diese Unternehmen unterliegen strengen Auflagen, wenn es um die Umsetzung von Informationssicherheitsmaßnahmen geht. Kreditinstitute werden von der Bundesanstalt für Finanzdienstleistungsaufsicht (BaFin) regelmäßig hinsichtlich der Einhaltung dieser Vorgaben überprüft. Geeignete Zugriffskontrollverfahren und die Nutzung von IAM-Tools zur Dokumentation aller im Unternehmen vorhandenen Systemen mit Zugriffsberechtigungen stellen Prozesse dar, die der Einhaltung dieser Vorschriften dienen und dabei helfen können Risiken zu minimieren. In ihrem Rundschreiben stellt die BaFin wesentliche Punkte heraus, die ein Kreditinstitut zum Nutzen der Risikominimierung durchzuführen hat. Dazu zählen u.A. klar definierte Risikominimierungsstrategien, Einsatz von internen Kontrollsystemen und genaue Vorgaben hinsichtlich des Datenmanagements. In Absatz 4.3.4. der MaRisk heißt es

z.B.: „Datenstruktur und Datenhierarchie müssen gewährleisten, dass Daten zweifelsfrei identifiziert, zusammengeführt und ausgewertet werden können sowie zeitnah zur Verfügung stehen“ (BaFin, 16.08.2021, https://www.bafin.de/SharedDocs/Veroeffentlichungen/DE/Rundschreiben/2021/rs_1021_MaRisk_BA.html) und „Das Institut hat zu gewährleisten, dass Risikodaten genau und vollständig sind. Daten müssen nach unterschiedlichen Kategorien auswertbar sein und sollten, soweit möglich und sinnvoll, automatisiert aggregiert werden können“ (BaFin, 16.08.2021, ebd.).

Hinsichtlich der beschriebenen Anforderungen ist der Einsatz einer geeigneten Softwareanwendung fast unumgänglich, wenn es um die Umsetzung entsprechender Vorgaben geht. Ein Identity Repository kann helfen, alle Daten, die Informationen zu Berechtigungen enthalten aufzulisten sowie deren Vollständigkeit zu sichern. Mitarbeiterkonten und Zugriffsrechte stellen in diesem Bezug relevante Daten dar, die im Identity Repository aufgeführt werden. Die Möglichkeit, all diese Daten zentral mittels dieser Softwareanwendung zusammenzuführen, verwalten und auswerten zu können, hilft die Anforderungen der BaFin umzusetzen.

Neben Auflagen für Kreditinstitute durch die BaFin, gibt es auch allgemeine Vorgaben im Berechtigungsmanagement, an denen sich Firmen orientieren müssen. Dazu zählen zum Beispiel die Vorgaben des Bundesamt für Sicherheit in der Informationstechnik (BSI). Im Baustein ORP.4 des BSI werden z.B. genaue Regelungen zum Umgang mit Berechtigungen und deren Dokumentation beschrieben. In Absatz ORP.4.A.3 „Dokumentation der Benutzerkennungen und Rechteprofile“ (Bundesamt für Sicherheit in der Informationstechnik, 02.2021, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium_Einzel_PDFs/02_ORP_Organisation_und_Personal/ORP_4_Identitaets_und_Berechtigungsmanagement_Editon_2020.pdf?) heißt es unter Anderem: „Es MUSS dokumentiert werden, welche Benutzerkennungen angelegte Benutzergruppen und Rechteprofile zugelassen und angelegt wurden“ (Bundesamt für Sicherheit in der Informationstechnik, 02.2021, ebd.). Auch hier helfen Identity Repositories alle notwendigen Informationen zu Mitarbeiterkonten und Berechtigungen zu hinterlegen und zu dokumentieren. Weiterhin beschreibt der Absatz: „die Dokumentation der zugelassenen Benutzer, angelegten Benutzergruppen und Rechteprofile MUSS regelmäßig daraufhin überprüft werden, ob sie den tatsächlichen Stand der Rechtevergabe widerspiegelt“ (ebd.). Um eine regelmäßige Prüfung der in einem Unternehmen hinterlegten Berechtigungen vorzunehmen, hilft es alle dafür notwendigen Daten ins Repository zu überführen, um diese zu späteren Zeitpunkten – z.B. im Zuge einer Rezertifizierung – schnell und zentralisiert überprüfen zu können. Beim Rezertifizierungsprozess geht es darum alle aktuell vergebenen Berechtigungen aufzulisten und nach ihrer vorhandenen Gültigkeit zu prüfen. Mittels der Software kann die Rezertifizierung unterstützt werden, da sofort alle notwendigen Daten eingesehen und auf deren Aktualität geprüft werden können.

Nachdem die rechtlichen Vorgaben als Hauptfaktor für den Nutzen eines Identity Repository deutlich geworden sind, sollen im folgenden Abschnitt die allgemeinen Vorteile erläutert werden, die den Einsatz einer solchen Software rechtfertigen.

Der grundlegende Nutzen des Identity Repository besteht aus dem Einbinden einer Vielzahl von Daten in einem zentral abrufbaren Speicher, welche im Berechtigungsmanagement verwendet werden. Das Implementieren von Mitarbeiterdaten und Konten, Berechtigungskonzepten, Rollen, Zugriffsrechten und Verknüpfungen von Mitarbeitern und Rollen in digitaler Form sorgt nicht nur für eine geordnete Übersicht, in der jederzeit benötigte Informationen abgerufen werden können, sondern erleichtern auch den Verwaltungsaufwand, den das Auflisten von Berechtigungsinformationen mit sich bringt. Mithilfe der Anwendung können Informationen zielgerichtet implementiert, abgerufen oder geändert werden. Neue Mitarbeiter, Systeme oder Rechte können schnell in die Software eingebracht und somit direkt verwaltet und aktualisiert werden. So ist es z.B. für einen Identity-Manager einfach, ein neues Berechtigungskonzept im Repository anzulegen, die dazugehörigen Rollen zu erstellen, diese direkt im Anschluss an Mitarbeiter zu vergeben und alles direkt zentral hinterlegt zu haben. Auch kann für Mitarbeiter, welche die Abteilung wechseln, schnell eine zentraler Rollenumstrukturierung im Repository durchgeführt werden, in dem nicht mehr benötigte Rollen entfernt und neu benötigte Rollen hinzugefügt werden. Damit ist die neue Rollenordnung schnell in einer Anwendung hinterlegt und kann mittels weiterer Prozesse ins System eingebunden werden.

Zudem ist sofort ersichtlich, welcher Mitarbeiter zu welchem Zeitpunkt welche Rollen und damit Berechtigungen hatte. Im Falle von Sicherheitsvorfällen, die ggf. auf unautorisierte Zugriffe zurückzuführen sind, können diese Informationen helfen, herauszufinden welcher Mitarbeiter die Möglichkeit gehabt hat Daten zu manipulieren, zu stehlen oder zu löschen. Das hilft nicht nur dabei, solchen Sicherheitsvorfällen effizienter nachzugehen, sondern auch um geeignete Kontrollmaßnahmen festzulegen.

Aus den beschriebenen Punkten wird deutlich, welche Nutzen der Einsatz eines Identity Repository mit sich bringt. Die wichtigsten Punkte die für die Nutzung sprechen sind nachfolgend noch einmal kurz zusammengefasst:

- Umsetzung amtlicher Vorgaben (MaRisk oder ORP.4)
- Unterstützung von Rezertifizierungsprozessen
- Einbinden und aktualisieren aller notwendigen Zugriffsinformationen
- Rollenvergabe, Rollenentzug und Rollenumstrukturieren von Mitarbeitern zentral in einer Anwendung
- Nachverfolgung und Kontrolle von Zugriffen

### 3.2. Aktuelle und potentielle Einsatzmöglichkeiten

Das Spektrum an Einsatzmöglichkeiten, die ein Identity Repository mit sich bringt, ergibt sich aus dem vorangegangenen Abschnitt zum Nutzen einer solchen Anwendung. Es lässt sich festhalten, dass alle Kreditinstitute, die aufgrund der MaRisk-Bestimmungen der BaFin zur besonderen Umsetzung von Dokumentationen und Kontrollen im Bereich der Zugriffskontrolle gezwungen sind, ein Identity Repository nutzen können. Damit können sie Ihre Prozesse zur Risikominimierung erleichtern. Allerdings sind nicht nur Kreditinstitute als Einsatzgebiet von Repositories oder vergleichbaren IAM-Tools zu bewerten. Grundsätzlich kann jedes Unternehmen, in dem eine größere Anzahl von Mitarbeitern, Systemen und damit verbundenen Berechtigungen existiert, IAM-Tools nutzen, um verwaltende Tätigkeiten zu erleichtern und Risikomanagementprozesse für Berechtigungskontrollen durchzuführen. Nicht nur weil Vorgaben wie ORP.4 existieren, kann über den Einsatz einer geeigneten Softwarelösung nachgedacht werden. Ab einer gewissen Anzahl von Mitarbeitern ist die Auflistung von Benutzerkonten und den damit einhergehenden Zugriffen unerlässlich, um den Überblick über alle Zugriffsmöglichkeiten und sich daraus ergebenden Risiken zu behalten und ein effektives Berechtigungsmanagement zu betreiben.

Auch in öffentlichen Einrichtungen wie Schulen könnte der Einsatz von geeigneten IAM-Tools mehr und mehr Relevanz erhalten. Diese zählen zwar nicht als klassische Unternehmen, haben aber ebenso eine Vielzahl von digitalen Systemen für die eine Kontrollmöglichkeit in Form der Zugriffskontrolle hilfreich sein kann. Die vermehrte Nutzung von digitalen Lehr-/ und Lernmöglichkeit erweitert auch den Umgang mit digitalen Ressourcen, woraus sich wiederum Zugriffsberechtigungen ergeben. Der Anstieg von Homeschooling zu Beginn der Corona-Krise 2020 hat dafür gesorgt, dass immer mehr Lerninhalte digital vermittelt werden.

In einer Studie von Initiative D21 und der TU München wird aufgezeigt, in welcher Form Lehrinhalte während der Corona-Krise 2020 in digitaler Form vermittelt worden sind. Abbildung 2 zeigt einen Ausschnitt aus der Studie:

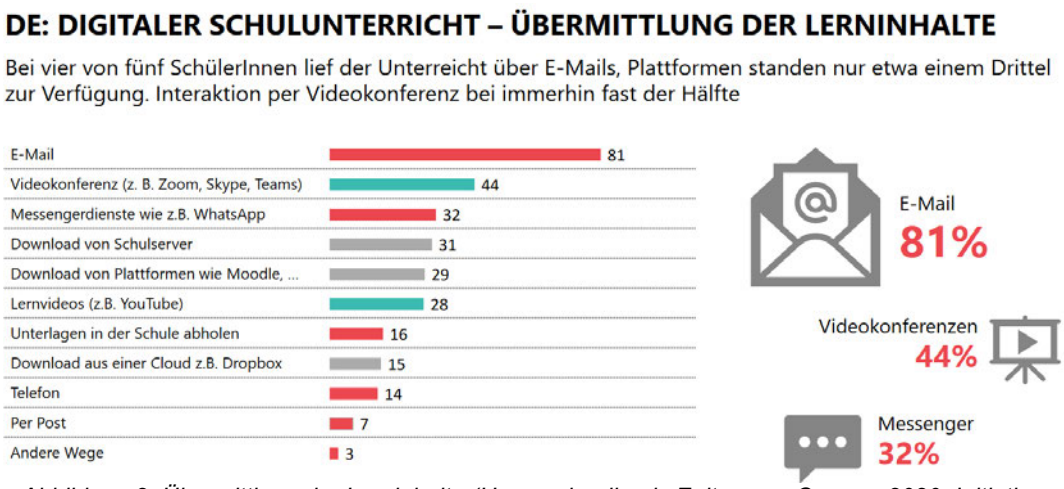


*Abbildung 2: Übermittlung der Lerninhalte (Homeschooling in Zeiten von Corona, 2020, Initiative D21, 2020, https://initiatived21.de/app/uploads/2020/08/homeschooling_ergebnisse_ egovern ment-monitor-2020.pdf)*

Die Abbildung mach deutlich wie stark sich die Übermittlung von Lerninhalten 2020 gewandelt hat. Nur 37% der Inhalte wurde persönlich übermittelt („Unterlagen in Schule abholen“ 16%,“Telefon“ 14%, „Per Post“ 7%). Es bleiben knapp zwei Drittel vermittelter Lerninhalte per digitalem Weg wie „E-Mail“ (81%), „Download vom Schulserver“ (31%) oder „Download aus Cloud, z.B. Dropbox“ (15%). Daraus ergeben sich große Datenmengen, die online verarbeitet und übertragen werden und somit auch ein starker Anstieg von Benutzern (Lehrern, Schülern, Eltern) die Zugriffe auf die jeweiligen Datenquellen benötigen und dementsprechend Zugriffsrechte erhalten. Diese Vielzahl von Zugriffsrechten könnte ebenfalls mittels eines Identity Repsoitories aufgelistet und überwacht werden, um Datenverluste oder Manipulationen in diesen Bereichen durch Kontrollmaßnahmen zu vermeiden.

Da sich dieser Trend zwar zu Krisenzeiten herausgestellt hat, jedoch tendenziell zukünftig weiterhin erhöht Richtung mehr digitalisiertem Lernen entwickeln könnte, kann es durchaus notwendig werden, auch an Schulen geeignete Risikominimierungsmaßnahmen zu treffen, wobei Identity Repositories eine wichtige Rolle einnehmen könnten.

### 3.3. Dokumentation von Zugriffsrechten

Um Zugriffsrechte und dazu relevante Informationen zu dokumentieren, gibt es verschiedene Möglichkeiten. In diesem Abschnitt sollen verschiedene Dokumentationsmöglichkeiten in der Theorie erläutert werden sowie deren Vor- und Nachteile gegenüber anderen Methoden.

Um eine effiziente Auflistung von Zugriffsrechten durchzuführen, muss zuerst klar sein, welche Daten überhaupt dokumentiert werden können. Je nach Zugriffskontrollverfahren ergeben sich verschiedene Daten, die bei einer Auflistung erfasst werden müssen. Diese

Daten müssen vollständig sein und alle notwendigen Informationen über die erteilten Zugriffsrechte, Daten und Systeme sowie Benutzer mit Zugriff auf diese Daten und Systeme, enthalten.

Um einen exemplarischen Überblick zu geben, welche Informationen bei der Dokumentation von Relevanz sein können, werden in Tabelle 5 Relevante Informationen für die rollenbasierte Zugriffskontrolle (RBAC) beschrieben. Es wird gezeigt, welche Punkte bei der Auflistung von rollenbasierten Zugriffsdaten berücksichtigt werden müssen und wie diese beschrieben werden können. Hierbei ist zu erwähnen, dass es sich um einen grundlegenden Überblick handelt. Welche Informationen genau dokumentiert werden müssen, ist von den einzelnen Unternehmensbereichen, Vorgaben, Strukturen und Zugriffsverfahren abhängig.

| Relevante Informationen | Beschreibung |
|---|---|
| Mitarbeiterdaten | Hierzu zählen alle grundlegenden Informationen, die über einen Mitarbeiter existieren und mit denen er eindeutig identifiziert werden kann (z.B. Name, Personalnummer, Abteilung, Vorgesetzer, Konten, Anmeldeinformationen). |
| Sonstige Benutzerdaten | Zu den sonstigen Benutzerdaten zählen Informationen zu allen weiteren Benutzern, welche direkt oder indirekt Zugriffe auf Daten der Firma haben können (z.B. Kundenkontos, externe Angestellte, Lieferanten).<br><br>Mitarbeiterdaten und sonstige Benutzerdaten können auch zusammengefasst werden, solange eine Differenzierung zwischen den beiden Typen bei der Dokumentation ersichtlich ist. |
| Systeme & Anwendungen (Berechtigungskonzepte) | Dazu zählen alle in einem Unternehmen genutzten Server, Dateisysteme oder Softwaretools, auf die ein Benutzer zugreifen kann. Bei der rollenbasierten Zugriffskontrolle werden diese Daten in Berechtigungskonzepte überführt, welche alle notwendigen Informationen enthalten. Dadurch ist für die Dokumentation die Auflistung aller aktuellen Berechtigungskonzepte inkl. wichtiger Grundinformationen (z.B. Name, interne ID, Verantwortliche) notwendig. |
| Rollen | Rollen bilden die Basis von RBAC und sind an ein Berechtigungskonzept gebunden. Daher müssen alle vorhandenen Rollen, die in einem Berechtigungskonzept definiert worden sind und an Benutzer vergeben werden können, aufgeführt werden. Informationen wie Rollenname, Rollen-ID und dem zugehörigen Berechtigungskonzept sind wichtige Bestandteile bei der Auflistung. Die Rollen müssen anhand der hinterlegten Informationen eindeutig identifizierbar und differenzierbar sein. |

| | |
|---|---|
| Zugriffsrechte | Die Zugriffsrechte sorgen für den eigentlichen Zugriff auf Daten oder Systeme. Daher müssen sie bei der Dokumentation ebenfalls aufgeführt werden. Zugriffsrechte sind im Rahmen der RBAC an Rollen geknüpft. Damit sind bei der Auflistung Informationen wie Zugriffsrechtsbezeichnung, zugehörige Rolle und Beschreibung der dadurch möglichen Zugriffe notwendig. |
| Verknüpfung von Benutzern und Rollen | Bei der Verknüpfung von Benutzern und Rollen wird erst deutlich, welche Person welche Zugriffe auf Daten in einem System hat. Um diese Zugriffe nachzuvollziehen, muss bei der RBAC aufgelistet werden, welchem Benutzer welche Rollen zugewiesen worden sind. So muss z.B. für jeden Mitarbeiter eine Liste hinterlegt sein, welche Rollen er besitzt und auf welche Daten er schließlich zugreifen kann. Dafür muss für alle Mitarbeiter eine eigenständige Liste geführt werden, welche regelmäßig geprüft (Rezertifizierung) und bei Änderungen angepasst werden kann. |

*Tabelle 5: Relevante Informationen zur Dokumentation bei rollenbasierter Zugriffskontrolle (RBAC) (eigene Tabelle)*

Um die in Tabelle 5 beschriebenen Informationen zu dokumentieren, existieren verschiedene Ansätze die sich in Form und Aufwand unterscheiden. Zwar existieren amtliche Vorgaben (wie MaRisk oder ORP.4) die eine Dokumentation von Zugriffsinformationen fordern, allerdings gibt es keine genauen Vorgaben, wie diese Dokumentation stattfinden muss. Aus diesem Punkt heraus ergeben sich folgende theoretischen Dokumentationsmöglichkeiten:

- 1. analoge schriftliche Dokumentation
- 2. digitale schriftliche Dokumentation
- 3. Nutzung eines Identity Repository
- 4. Kombination aus den verschiedenen Ansätzen

**analoge schriftliche Dokumentation**

Bei der analogen schriftlichen Dokumentation werden die benötigten Dokumentationsdaten handschriftlich auf Papier aufgeschrieben und physisch in einem Ordner geführt und gesammelt. Diese Form der Dokumentation ist zwar veraltet, jedoch können Zugriffsdaten auch in dieser Form verwaltet werden. Vorgaben der BaFin fordern u.A., dass Daten „soweit möglich und sinnvoll, automatisiert aggregiert werden können“ (BaFin, 16.08.2021, https://www.bafin.de/SharedDocs/Veroeffentlichungen/DE/Rundschreiben/ 2021/rs_1021_MaRisk_BA.html), allerdings wird keine Pflicht für dieses digitale Sammeln von Zugriffsinformationen beschrieben. Für kleine Unternehmen mit einer Handvoll Mitarbeitern und minimaler digitaler Infrastruktur wäre diese Form der Dokumentation zwar grundsätzlich denkbar, allerdings ist das Auflisten der gewünschten Informationen sehr unflexibel, veraltet und mit hohem Aufwand verbunden. Tabelle 6 zeigt einige Vor- und Nachteile, die sich aus der analogen schriftlichen Dokumentation ergeben:

| **Vorteile** | **Nachteile** |
|---|---|
| + keine Extrakosten für Software wie IAM-Tools | - veraltete Dokumentationsform |
| + kein Datenverlust bei digitalem Datendiebstahl oder Systemausfällen | - hoher Zeitaufwand |
| | - Änderungen schwer einzubinden |
| | - kein zentraler digitaler Zugriff auf Daten |
| | - hoher Papierverbrauch |
| | - unübersichtliche Dokumentation |
| | - Lagerungsort benötigt |

*Tabelle 6: Vor- und Nachteile der analogen schriftlichen Dokumentation (eigene Tabelle)*

**digitale schriftliche Dokumentation**

Diese Dokumentationsform nutzt digitale Möglichkeiten durch entsprechende Software (z.B. Microsoft Office Excel, Open Office Calc) um Zugriffsinformationen zu hinterlegen und zu bearbeiten. Hierbei können mittels Tabellenprogrammen Daten aufgelistet, bearbeitet und verknüpft werden. Dadurch ergibt sich ein deutlich verminderter Aufwand und eine übersichtlichere Struktur als bei der analogen schriftlichen Dokumentation. Änderungen sind schneller und effektiver einzupflegen und die Daten können zentral für mehrere Bearbeiter hinterlegt werden. Die Vor- und Nachteile dieses Verfahrens sind in Tabelle 7 Aufgeführt. Hierbei ergeben sich vor allem Vorteile gegenüber der analogen schriftlichen Dokumentation.

| Vorteile | Nachteile |
|---|---|
| + übersichtlichere Struktur als analog | - Änderungen müssen ggf. an mehreren Stellen in Tabelle manuell gemacht werden |
| + Änderungen schneller durchführbar als analog | - bei vielen Daten steigt die Unübersichtlichkeit und Fehleranfälligkeit bei der Bearbeitung |
| + schnellere Verarbeitung von Daten durch Kopieren, Verschieben oder Senden | - Verknüpfungen von Benutzern und Rollen (RBAC) müssen manuell bearbeitet werden |
| + schnelles Erstellen von Listen | |
| + zentraler Zugriff | |

*Tabelle 7: Vor- und Nachteile der digitalen schriftlichen Dokumentation (eigene Tabelle)*

**Nutzung eines Identity Repository**

Die Nutzung eines Identity Repositoriy stellt eine Möglichkeit dar, Zugriffsberechtigungen systematisch und übersichtlich zu dokumentieren sowie eine zentrale Verwaltung und Bearbeitung der eingetragenen Informationen durchzuführen. Dafür stellt ein solches Repository entsprechende Funktionen bereit, welche das Einbinden und Bearbeiten von Daten erleichtern. Das Repository ermöglicht es, die Daten in getrennten und klar strukturierten Bereichen anzulegen und zu bearbeiten. So stehen für einzelne Informationen (wie Mitarbeiterdaten, Berechtigungskonzepte oder Rollen) eigene Bereiche in der Anwendung zur Verfügung und ergeben somit eine klar getrennte Strukturierung der Daten. Dies verhindert Fehler bei der Bearbeitung durch das Vermischen von wichtigen Informationen oder durch fehlerhaftes Verknüpfen, wie es bei der Arbeit mit Tabellenprogrammen der Fall sein kann. Zwar ist das Nutzen von Identity Repositories in der Regel mit höheren Kosten verbunden, da geeignete Software oft kostenpflichtig in der Anschaffung ist, jedoch kann der dadurch verringerte Arbeitsaufwand bei der Bearbeitung der Informationen Arbeitszeit sparen und Bearbeitungsfehler minimieren. Durch die Erleichterung des administrativen Aufwands und der damit verbundenen Minimierung von Bearbeitungsfehlern können Risiken minimiert und mögliche Folgeschäden verringert werden.

Neben dem erleichterten administrativen Aufwand können Zusatzfunktionen Informationen Automatisiert generieren und speichern, die bei der Auflistung per Schreibprogramm sehr aufwändig und nur manuell durchführbar wären. Dazu zählt z.B. das automatische Aktualisieren von Informationen, nach dem Daten bearbeitet wurden. So können bei der Bearbeitung von Einträgen in der Software automatisch Aktualisierungen wie Datum oder bearbeitender Benutzer vorgenommen und gespeichert werden und aufzeigen, wer zu welchem Zeitpunkt Daten im System

geändert oder eingefügt hat. Solche Funktionen helfen den Anforderungen an das Risikomanagement gerecht zu werden, da Fehler in der Bearbeitung, aus denen z.B. falsche Rollenzuordnungen folgen, erkannt und zugeordnet werden können.

Das Zuweisen von Rollen zu Benutzern kann mittels Repository ebenfalls schneller und geordneter erfolgen als bei den schriftlichen Dokumentationsformen. Während in einem Tabellenprogramm große Listen mit Mitarbeitern und Rollen gepflegt werden müssen, können hier Rollen per Mausklick entzogen und vergeben werden. Auch dadurch wird der hohe administrative Aufwand und sich daraus ergebende Risiken verringert. Tabelle 8 fasst einige Vor- und Nachteile, die sich aus der Verwendung eines Identity Repository ergeben noch einmal zusammen:

| **Vorteile** | **Nachteile** |
|---|---|
| + Daten stehen in einer zentralen Anwendung zur Verfügung | - Abhängigkeit zur Anwendung |
| + strukturierte Trennung von Daten | - Daten müssen oft einzeln in der Anwendung angelegt werden |
| + schnelle und einfache Einbindung und Bearbeitung von Informationen per Mausklick | - erhöhte Kosten |
| + Kontrollfunktion durch Tracking veränderter Informationen (Datumsstempel, Bearbeiter) | |
| + Erweiterung von Funktionen und Updates möglich | |

*Tabelle 8: Vor- und Nachteile von Identity Repositories (eigene Tabelle)*

**Kombination aus den verschiedenen Ansätzen**

Eine weitere Möglichkeit, Zugriffsdaten zu dokumentieren, besteht in der Kombination von mehreren der bereits genannten Möglichkeiten. Hierbei soll vor allem auf die Kombination der digitalen schriftlichen Dokumentation mit der Nutzung eines Identity Repository eingegangen werden.

Diese Form der Dokumentation ermöglicht es sowohl die einfache und schnelle Auflistung von Daten mittels eines Tabellenprogramms als auch die strukturierte Verwaltungsform und Nutzung von Extrafunktionen des Identity Repository zu verbinden.

Der große Vorteil des Tabellenprogramms besteht in der schnellen Auflistung von Informationen in Listen. So können Mitarbeiterdaten oder vorhandene Berechtigungskonzepte schnell in einer entsprechenden Liste angelegt werden, während im Identity Repository die Daten zuerst einzeln in die Anwendung gebracht werden müssen. Beim Repository hingegen erleichtert sich der Folgeaufwand, nachdem die Daten implementiert worden sind. Durch die klaren Strukturen können Daten einfacher gefunden und bearbeitet werden. Zwar werden die Daten im Repository auch in einer Datenbank gespeichert und somit ähnlich gelistet wie im Tabellenprogramm, allerdings erlaubt die Softwareanwendung eine schnellere und gezieltere Bearbeitung durch entsprechende Programmfunktionen.

Durch das Einbinden einer Import-Funktion im Repository können Daten noch schneller in die Anwendung implementiert werden. Dafür muss eine geeignete Import-Funktion im Repository zur Verfügung stehen, mit der es möglich ist, zuvor mittels Tabellenprogramm erstellte Listen in die Datenbank des Repository zu laden. So können neue Mitarbeiter, Berechtigungskonzepte oder Rollen jeweils in separaten Listen erstellt und anschließend importiert werden, ohne die Daten direkt einzeln im Repository anzulegen. Aus dieser Kombination ergeben sich weiter Vorteile (Tabelle 9):

| **Vorteile** | **Beschreibung** |
|---|---|
| Erstellen von Datensätzen | Datensätze können schneller mittels Listen/Tabellen erzeugt werden statt diese einzeln im Repository einzutragen. |
| Einbinden von Datensätzen | Die Daten aus den erstellten Listen können per Mausklick direkt per Funktion in die Repository-Datenbank geladen werden und stehen sofort zur Verfügung. |
| Segregation of Duties | Die Aufgabentrennung kann in diesem Fall hilfreich, sein Risiken zu minimieren und Kontrollen durchzuführen. So können einzelne Mitarbeiter der Personalabteilung z.B die Listen für Mitarbeiter erstellen, die ins Repository implementiert werden müssen, ohne selber darauf Zugriff zu erhalten. Der Administrator oder Identity-Manager kann dann die vorgelegten Listen prüfen und anschließend ins Repository importieren. Dadurch entsteht eine weitere Kontrollfunktion und der Zugriff von unautorisierten Benutzern auf das Repository wird verhindert. |

*Tabelle 9: Vorteile der Kombination von Tabellenprogramm und Repository (eigene Tabelle)*

## 3.4. Technische Anforderungen

Bei der Entwicklung einer Software zur Verarbeitung und Dokumentation von Zugriffsrechten müssen einige Kernaspekte festgelegt werden, die erfüllt werden müssen, damit die Anwendung allen Anforderungen gerecht wird. Eine spezialisierte Software zur Speicherung von Daten im Bereich der rollenbasierten Zugriffskontrolle muss gewisse technische Aspekte berücksichtigen, welche unablässig für einen effektiven Einsatz sind.

Um eine entsprechende Software zu entwickeln, muss zuerst klar sein, wie die Software grafisch aussehen soll, welche Funktionen sie erfüllen muss und wie diese implementiert werden können. Neben der richtigen Programmiersprache und einer geeigneten Entwicklungsumgebung muss definiert werden, wie Eingabedaten gespeichert und externe Daten in die Anwendung gebracht werden können. So ist festzulegen, ob eine Datenbankanbindung benötigt wird und wie diese mit der Anwendung verbunden wird.

Nachfolgend sind alle Bereiche beschrieben in denen eine entsprechende Software Anforderungen gerecht werden muss:

- Grafische Benutzeroberfläche und Handhabung
- Funktionen im Rahmen der RBAC
- Datenverarbeitung und Datenbankanbindung
- Sicherheit

### Grafische Benutzeroberfläche und Handhabung

Die Grafische Benutzeroberfläche (GUI) stellt das Programm optisch auf dem Bildschirm dar und ist somit der Bereich, mit dem Benutzer der Software direkt in Kontakt kommen. Über die GUI werden alle Benutzerinteraktionen gesteuert, um sie im Hintergrund mittels verschiedenster Programmfunktionen zu verarbeiten. Die GUI stellt alle Fenster und Interaktionsmöglichkeiten für den Benutzer bereit und sollte daher möglichst intuitiv und einfach zu handhaben sein. Eine intuitive Handhabung kann durch eine bestimmte Anordnung von Elementen in der GUI erreicht werden, die sich an gängigen Softwareanwendungen orientiert. Als Elemente zählen dabei z.B. aufrufbare Fenster, Buttons, Textfelder, Texteingabefelder oder Menüleisten. Damit eine möglichst einfache Handhabung und Interaktion mit den einzelnen Elementen erreicht werden kann, sollten die Elemente schlicht und gut erkenntlich sein. Die Menüleiste sollte sich im oberen Bereich des Fensters anordnen. Darunter sollten Elemente wie Suchleisten, Buttons und Listen erzeugt werden ,um eine klare Struktur zu schaffen. Eine mögliche Anordnung von Elementen einer Access Management Anwendung ist in Abbildung 3 dargestellt.

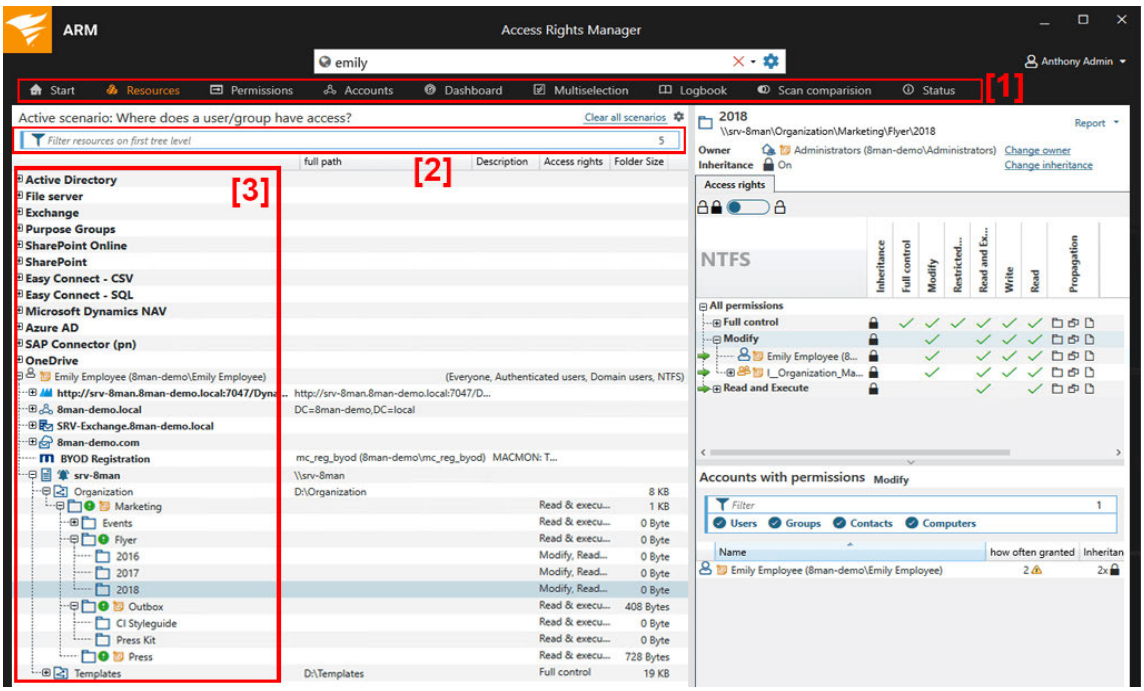


*Abbildung 3: IAM-Tool „solarwinds“ grafische Oberfläche (Solarwinds, 2022, https://www.solarwinds.com/de/access-rights-manager/access-management-system)*

Im Beispiel der Anwendung *solarwinds (siehe Abbildung 3)* erstreckt sich die Menüleiste [1] im oberen Bereich des Fensters. Darunter werden weitere Elemente wie Suchfelder [2], Listen und andere Auswahlmöglichkeiten [3] angezeigt, mit denen man die einzelnen Funktionen des Programms steuern kann. Dadurch ergibt sich eine klare Struktur, die eine einfache Handhabung ermöglicht und dem Benutzer deutlich macht, an welcher Stelle im Programm er welche Benutzerinteraktionen ausführen kann.

Neben der Anordnung der Elemente spielen auch die optischen Reaktionen des Programms auf Benutzereingaben eine Rolle. Optisches Feedback der GUI nach dem Klicken eines Buttons oder dem Auswählen eines Elementes einer Liste geben dem Benutzer die Möglichkeit nachzuvollziehen, ob eine gewisse Eingabe erfolgreich war und welche weiteren Eingaben sich ggf. daraus ergeben.

### Funktionen im Rahmen der RBAC

Neben der Grafischen Benutzeroberfläche als Bindeglied zwischen Benutzer und Programmfunktionen muss die Software dem Zugriffskontrollverfahren entsprechende Funktionen bereitstellen. Bei der RBAC sind das neben dem Anlegen und Bearbeiten von Mitarbeitern und Benutzern, Berechtigungskonzepten, Rollen und Zugriffsrechten auch die Funktionen zum Vergeben und Entziehen von Rollen sowie das Erstellen von Reports aus der Anwendung heraus. Im Programm müssen dazu verschiedene Funktionen implementiert werden. Für das Anlegen von Mitarbeitern muss nach der Eingabe von Daten über die GUI z.B. eine Datenbankanbindung hergestellt werden, in der der ausgewählte Mitarbeiter hinterlegt werden kann. Dazu müssen die Eingaben wie Namen oder Mitarbeiternummer aus dem Eingabefeld geholt und in die passende Datenbanktabelle geschrieben werden. Bei der Bearbeitung bestehender Mitarbeiterdaten müssen ebenfalls die Daten aus den Eingabefelder zwischengespeichert werden. Danach erfolgt z.B. mittels eindeutiger Mitarbeiternummer das Update in der Datenbanktabelle, in dem der dort passende Mitarbeitereintrag mittels ID zugeordnet und angepasst wird. Gleiche Vorgehensweise gilt beim Anlegen und Bearbeiten von Berechtigungskonzepten oder Rollen.

Bei der Zuweisung von Rollen an im Programm hinterlegte Mitarbeiter muss eine zusätzliche Funktion aufgerufen werden, welche jedem Mitarbeiter die Rollen in der Datenbank zuweist. Dabei muss zuvor für jeden Mitarbeiter eine eigene Tabelle erzeugt werden, in der die Rollen eingefügt werden können.

Für das Erstellen von Reports wird eine Funktion benötigt, welche die vorhandenen Daten aus der Datenbank holt und diese z.B. in eine Textdatei schreibt. Um z.B. einen Report zu erstellen, der alle im System hinterlegten Berechtigungskonzepte in einer Datei ausgibt muss eine separate Datenbankverbindung aufgebaut und die benötigten Daten aus der richtigen Tabelle geholt und in die Textdatei geschrieben werden. Die Einträge in der Textdatei müssen übersichtlich sein, um eine spätere Auswertung oder Verarbeitung zu vereinfachen.

### Datenbankanbindung und Datenverarbeitung

Um die Menge an Daten, die in einem Identity Repository anfallen, speichern zu können, empfiehlt sich das Verwenden einer Datenbank. Dafür können z.B. SQL-Datenbanken wie MySQL und SQLite oder alternative Datenbanksysteme wie MS Access genutzt werden. Die Auswahl einer geeigneten Datenbank richtet sich dabei u.a. nach der Verarbeitungsgeschwindigkeit oder der Menge an zu verarbeitenden Daten.

Im Programmcode müssen Funktionen zum Einfügen oder Auslesen von Daten aus der Datenbank implementiert sein. Um Daten mittels einer MySQL-Datenbank zu verarbeiten, werden u.a. folgende Klassenfunktionen benötigt (Tabelle 10):

| Funktion | Beschreibung |
|---|---|
| CreateTable() | Erzeugt eine neue Tabelle in der Datensätze eingefügt werden können. |
| DropTable() | Löscht eine bestehende Tabelle inklusive aller bereits vorhandenen Datensätze. |
| AlterTable() | Ermöglicht die Anpassung von Tabellenfeldern bereits erzeugter Tabellen. |
| InsertIntoTable() | Ermöglicht das Einfügen von Datensätzen in eine bestehende Tabelle. |
| UpdateTable() | Ermöglicht das Ändern von Datensätzen in einer bestehenden Tabelle. |
| SelectFromTable() | Holt Datensätze aus einer Tabelle um sie auszugeben, zu speichern oder weiterzuverarbeiten. |

*Tabelle 10: Programmfunktionen zum Verarbeiten von Daten mittels SQL-Datenbank (eigene Tabelle)*

**Sicherheit**

Da es sich beim Identity Repository um eine Anwendung handelt, die im Bereich des Risikomanagements zum Einsatz kommt, spielen Sicherheitsimplementierungen eine entscheidende Rolle. Die Software muss u.a. nicht nur vor unautorisiertem Zugriff geschützt werden, sondern auch Datenverluste verhindern oder Systemabstürze abfangen. Um diese Sicherheitsanforderungen zu erfüllen, müssen geeignete Funktionen existieren, die im Programm selbst implementiert sind.

Ein Identity Repository verfügt über sensible Daten, wie Mitarbeiterdaten oder kritische Zugriffsberechtigungen und muss daher vor unautorisierten Zugriffen geschützt werden. Nicht genehmigte Zugriffe bieten die Möglichkeit, diese sensiblen Daten zu lesen, zu kopieren oder zu manipulieren. Die Verletzung von Datenschutzrichtlinien ist dabei ebenso kritisch wie der Missbrauch oder Verlust dieser Daten. Ein Angreifer könnte durch den Zugriff auf diese Informationen z.B. schnell herausfinden, welche Anwendungen in der Firma genutzt werden, welche kritischen Berechtigungen es gibt und welcher Mitarbeiter welche Zugriffsrechte hat. Zudem könnte er die vorliegenden Einträge im Repository überschreiben oder verändern und damit wichtige Informationen unbrauchbar machen.

Um solche Manipulationen zu verhindern, sollte die Software geeignete Maßnahmen zum Zugriffsschutz beinhalten. Die wohl einfachste und effektivste Maßnahme ist das Einsetzen einer Log-In Funktion mittels Benutzernamen und Passwort. Um Zugriff auf die Daten in der Software zu erhalten, muss sich der Benutzer zuerst mit seinem hinterlegten Benutzernamen und Passwort autorisieren. Erst dann erhält er Zugriff auf

die sensiblen Daten und kann Änderungen im Repository durchführen. Zwar können Benutzerdaten und Passwörter ebenfalls angegriffen werden, allerdings bilden sie eine einfach zu implementierende Sicherheitsfunktion, die sich ggf. noch mittels Mehrfaktorauthentifizierung erweitern lässt. Um damit verbundene Risiken noch weiter zu senken sollte darauf geachtet werden, dass nur wenige ausgewählte Mitarbeiter entsprechende Zugangskonten bekommen und damit Zugriff auf die Software erhalten. Auch sollten die verwendeten Passwörter gewisse Eigenschaften erfüllen. So sollte ein Passwort eine Mindestlänge von 12 Zeichen haben und Groß- /Kleinbuchstaben, Zahlen und Sonderzeichen beinhalten. Diese Passwörter sollten zudem nicht im Klartext in einer Datenbank hinterlegt sein und mittels Verschlüsselungsfunktionen gesichert sein.

Neben Zugriffen durch nicht berechtigte Benutzer können auch andere Aspekte die Sicherheit der Software beeinflussen. So ist beim Aufbau der Software darauf zu achten, dass eine geeignete Fehlerbehandlung bei den einzelnen Funktionen stattfindet, um Abstürze und damit verbundene Datenverluste zu vermeiden. Wenn ein Benutzer z.B. eine Eingabe tätigt, die vom Programm nicht vorgesehen ist, muss die Fehlerbehandlung dafür sorgen, dass das Programm nicht abstürzt und bisher eingetragene Daten in diesem Moment nicht verloren gehen. Gerade beim Aufbau von Datenbankverbindungen oder dem Lesen oder Schreiben von Daten in eine bestehende Datenbank können Abstürze schnell zu Datenverlusten oder fehlerhaften Einträgen führen. Daher ist hier besonders auf eine strukturierte und weitreichend abdeckende Fehlerbehandlung zu achten.

### 3.5. Einbindung zu verwaltender Datensätze

Wie in Abschnitt 3.3. beschrieben, existieren verschiedene Ansätze Zugriffsdaten zu speichern. Um diese Daten speichern zu können, müssen sie allerdings erst in die Anwendung gebracht werden. Im Identity Repository erfolgt die Einbindung von Informationen hauptsächlich über das manuelle Einfügen durch einen autorisierten Benutzer. Alle Informationen zu Mitarbeitern, Berechtigungskonzepten oder Zugriffsrechten müssen initial erstellt und gespeichert werden, damit sie im Weiteren zur Verfügung stehen. Dafür existieren einzelne Eingabefelder, in denen die Informationen eingetragen und gespeichert werden können. Diese Eingabefelder stellen einzelne Elemente innerhalb der grafischen Oberfläche der Software dar und ermöglichen eine einfache Implementierung durch Benutzerinteraktionen. Über Textfelder können z.B. Namen, ID's oder Beschreibungen vom Bearbeiter eingegeben und anschließend über zugehörige Buttons gespeichert werden. Durch das Klicken des Speichern-Buttons erfolgt im Hintergrund der Datenbankaufbau und die Daten werden in der Datenbanktabelle hinterlegt. Damit sind sie in der Anwendung hinterlegt und können eingesehen und bearbeitet werden.

Neben der manuellen Einbindung von Daten stellt auch das Einbinden von bereits erstellten Dateien mit zugehörigen Informationen eine Möglichkeit dar Daten, in eine Anwendung zu heben. Ebenfalls wie in Abschnitt 3.3. beschrieben können z.B. CSV-

Dateien mit Mitarbeiterinformationen erstellt werden und über eine Importfunktion direkt in die Datenbank der Anwendung geladen werden. Damit lassen sich schnell größere Datenmengen mit einmal implementieren, ohne diese händisch eingeben zu müssen. Die Software muss dafür allerdings die entsprechende Import-Funktion bereitstellen und die Form der Importdateien genau festlegen.

# 4. Funktionsweise und Aufbau

Die in diesem Kapitel vorgestellte Softwarelösung stellt ein Identity Repository dar, mit den wichtigsten Funktionen zur Verwaltung und Dokumentation von Zugriffsrechten im Rahmen der rollenbasierten Zugriffskontrolle. Die Software soll zeigen, wie ein solches Repository hinsichtlich grafischer Oberfläche, Funktion, Handhabung und Datenverarbeitung aufgebaut sein kann und wie es die in Kapitel 3 beschriebenen Anforderungen umsetzt. Dafür wird zuerst gezeigt, wie die Software grafisch aufgebaut ist und welche wichtigen Klassen implementiert wurden. Anschließend werden die genutzten Mittel zur Datenverarbeitung und -speicherung und die Nutzung der Reportfunktion beschrieben.

## 4.1. technische Grundlagen

Die Software wurde mit der Programmiersprache Java entwickelt. Java ist eine objektorientierte Programmiersprache, das heißt, sie erlaubt es Attribute, zu einzelnen Objekten zusammenzufassen, mit denen innerhalb der Sprache interagiert werden kann. Diese Form der Programmierung eignet sich besonders für größere Softwareanwendungen und ermöglicht eine Art modularen Aufbau beim Entwickeln von einzelnen Klassen.

Für die Software wurde die Java-Version 17 verwendet. Um die Entwicklungsarbeit zu erleichtern, wurde eine Entwicklungsumgebung genutzt. Diese ermöglicht eine strukturierte Programmentwicklung durch die übersichtliche Darstellung von Klassen sowie dem direkten Testen der Funktionen in der Umgebung. Als Entwicklungsumgebung wurde Eclipse ausgewählt. Dafür muss das Java Software Development Kit (JDK) installiert sein. Folgende Funktionen waren bei der Auswahl von Eclipse ebenfalls von Bedeutung:

- Freeware
- Erweiterung mittels verschiedener Plug-Ins
- Import von Projekten und Erzeugen von ausführbaren Dateien
- Kompatibilität mit allen Windows-Plattformen

Beim Entwicklungssystem wurde ein Windows 10 Betriebssystem ausgewählt, da es eine einfache Nutzung der Eclipse Entwicklungsumgebung erlaubt. Das System wurde zum Erstellen der Software und zum Testen verwendet. Tabelle 11 listet alle Systemvoraussetzungen des Entwicklungssystems auf:

| **Komponente** | **Details** |
|---|---|
| Betriebssystem | 64-bit Windows 10 Pro v21H2 |
| CPU | AMD Ryzen 7 2700X Eight-Core Processor 3.70 GHz |
| RAM | 16 GB DDR4 |
| Java Version | Java SE JDK 17.0.2 |
| Eclipse Version | Eclipse IDE 2022-03 |

*Tabelle 11: Systemvoraussetzungen des Entwicklungssystems (eigene Tabelle)*

## 4.2. Grafische Oberfläche

Die komplette Benutzerinteraktion bei der entwickelten Anwendung findet in einem zentralen Fenster statt. Alle Benutzereingaben werden hierüber gesteuert und verarbeitet. Alle Daten, die in der verknüpften Datenbank hinterlegt sind, werden entsprechend ihrer Kategorie aufgelistet und ausgegeben. Die GUI ermöglicht das Einfügen und Ändern von Einträgen in der Datenbank und deren anschließende grafische Ausgabe in Textform.

Durch die Entwicklung der Software hinsichtlich rollenbasierter Zugriffskontrolle ist die grafische Oberfläche in verschiedene Reiter (Tabs) unterteilt, welche die Unterteilung zwischen den einzelnen Kategorien optisch und funktional trennt. Für jeden Bereich existiert ein einzelner Tab der am oberen Bereich des Fensters ausgewählt werden kann. Die Reiter kategorisieren folgende Bereiche, welche kurz hinsichtlich ihres Aufbaus und ihrer Funktion erläutert werden:

- Log-In
- User Management
- Access Concepts
- Roles
- Access Rights
- Assignment User-Roles
- Create Reports

### Log-In

Um die gespeicherten Informationen vor unerwünschten Zugriffen zu schützen, ist die Anwendung mit einer Log-In Funktion ausgestattet (Abbildung 4). Beim Programmstart muss sich der Benutzer zuerst mit Benutzernamen und Passwort autorisieren. Die Benutzerdaten sind in einer separaten Datenbanktabelle gespeichert und werden bei der Eingabe der Daten in die Felder „Username“ und „Password“ abgefragt und auf Richtigkeit geprüft. Wurden die Benutzerdaten korrekt eingegeben, öffnet sich das eigentliche Hauptfenster.

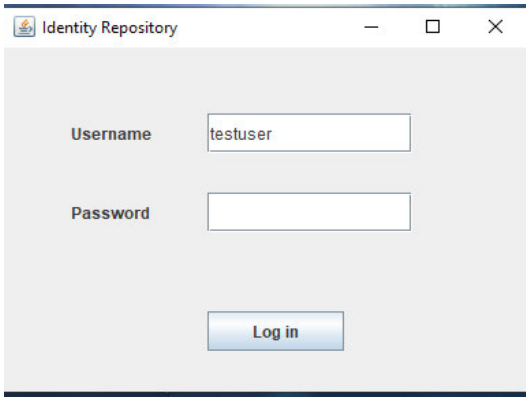


*Abbildung 4: Fenster zum Log-In (eigene Abbildung aus der Software)*

### User Management

Dieser Bereich befasst sich mit dem Anlegen und Verwalten von Benutzerdaten wie Mitarbeiterkonten. Hier können Mitarbeiterkonten angelegt und bearbeitet werden. Abbildung 5 zeigt den zugehörigen Ausschnitt aus der Software.

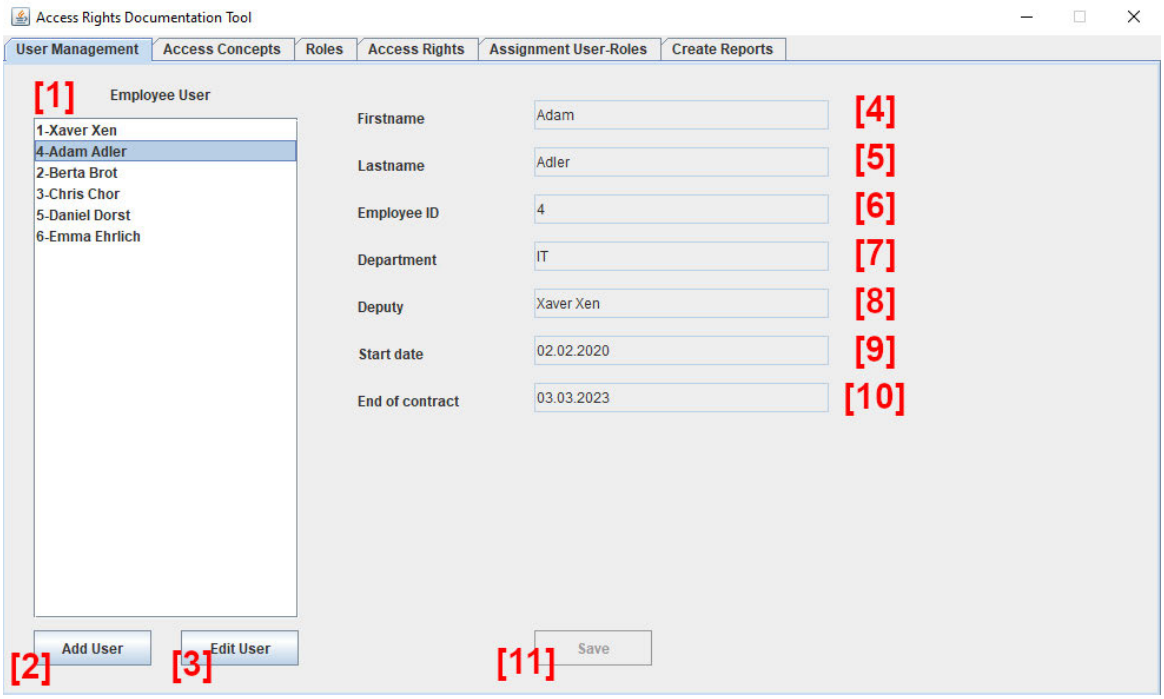


*Abbildung 5: Tab „User Management“ zum Anlegen und Bearbeiten von Benutzerdaten(eigene Abbildung aus der Software)*

Die Abbildung zeigt den Aufbau des User Management Tabs. Bereits angelegt Nutzer werden in der Liste auf der linken Seite [1] chronologisch nach ihrem Anlegedatum aufgelistet. Hierfür wird eine JList verwendet. Über den Button „Add User“ [2] können neue Benutzer angelegt werden. Dafür sind die Felder „Firstname“ [4], „Lastname“ [5], „Employee-ID“ [6], „Department“ [7], „Deputy“ [8], „Start date“ [9] und „End of contract“ [10] auszufüllen und per „Save“ Button [11] zu speichern. Das Anpassen von bereits hinterlegten Benutzerinformationen erfolgt über den „Edit User“ Button [3]. Damit können die zuvor eingefügten Daten aktualisiert werden. „Firstname“, „Lastname“ und „Employee-ID“ sind Pflichtfelder und müssen ausgefüllt werden, damit die Anwendung die Benutzer anlegen kann und keine Fehlermeldung ausgegeben wird. Die „Employee-ID“ muss zwingend einem ganzzahligen Wert entsprechen.

### Access Concepts

Abbildung 6 zeigt den Bereich „Access Concept“ in der GUI. Hierüber erfolgt, wie beim Tab „User Management“, das Anlegen und Bearbeiten von Berechtigungskonzepten über die zugehörigen Buttons „Add AC“ [1] und „Edit AC“ [2]. In diesem Falls sind die notwendigen Informationen zum Berechtigungskonzept zu hinterlegen. Dazu gehören „AC ID“ [3], „AC Name“ [4], „AC Owner“ [5] und “Description“ [6]. Im Feld „AC ID“ ist eine einmalige Identifikationsnummer für das Konzept anzugeben. „AC Name“ sollte den ausgeschriebenen Namen des Konzeptes enthalten. Das Feld „AC Owner“ legt die Verantwortlichkeit des Konzeptes fest und sollte den Namen des Mitarbeiters enthalten, der für das Berechtigungskonzept verantwortlich ist. Im Feld „Description“ kann eine kurze Beschreibung zum Inhalt des Konzeptes angelegt werden. Dieses Feld ist als einziges kein Pflichtfeld und muss nicht zwingend beim Anlegen befüllt werden.

Der nicht editierbare Bereich „Last Change“ [7] wird bei einer Neuanlegung oder Anpassung automatisch mit dem aktuellen Datum und dem Benutzer überschrieben, der die Änderung zuletzt vorgenommen hat. Es wird immer der Benutzer unter „Last Change“ hinterlegt, der sich aktuell in der Anwendung über den Log-In angemeldet und die entsprechende Änderung durchgeführt hat. In der Abbildung wäre dies der Benutzer mit dem Namen „admin“.

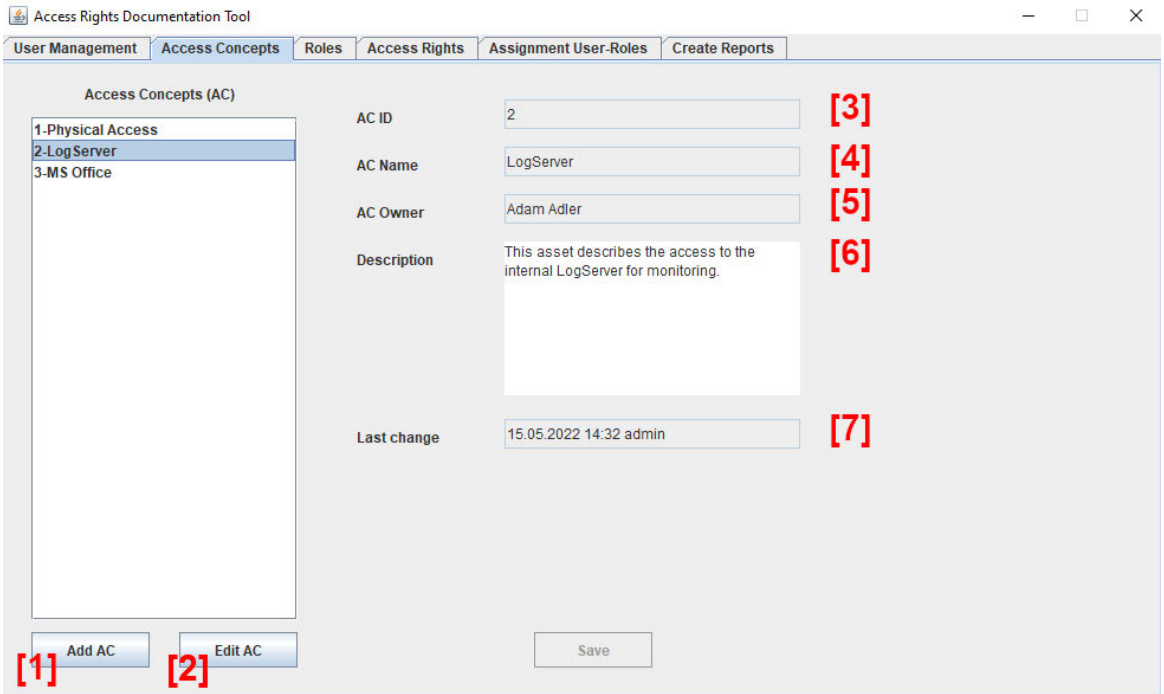


*Abbildung 6: Tab „Access Concept“ zum Anlegen und Bearbeiten von Berechtigungskonzepten (eigene Abbildung aus der Software)*

### Roles

In diesem Bereich der GUI werden Rollen erstellt und den zugehörigen Berechtigungskonzepten zugeordnet (Abbildung 7). Um eine Rolle zu erstellen, muss zuerst ein Berechtigungskonzept in der oberen Liste [1] ausgewählt werden. Danach kann über den „Add Role“ Button [2] das Anlegen einer neuen Rolle erfolgen, welche anschließend automatisch dem ausgewählten Berechtigungskonzept zugeordnet wird. Alle für eine Konzept erstellte Rollen werden in der unteren Liste unter „Roles from AC“ [3] angezeigt, sobald ein Konzept ausgewählt wurde. Beim Erstellen einer Rolle sind die Bereiche „Role ID“ [4], „Role Name“ [5], „RoleType“ [6] und „Description“ [7] auszufüllen. Wie im Reiter „Access Concepts“ wird im Bereich „Last Change“ [8] beim speichern automatisch die aktuelle Datums- und Benutzerangabe aktualisiert.

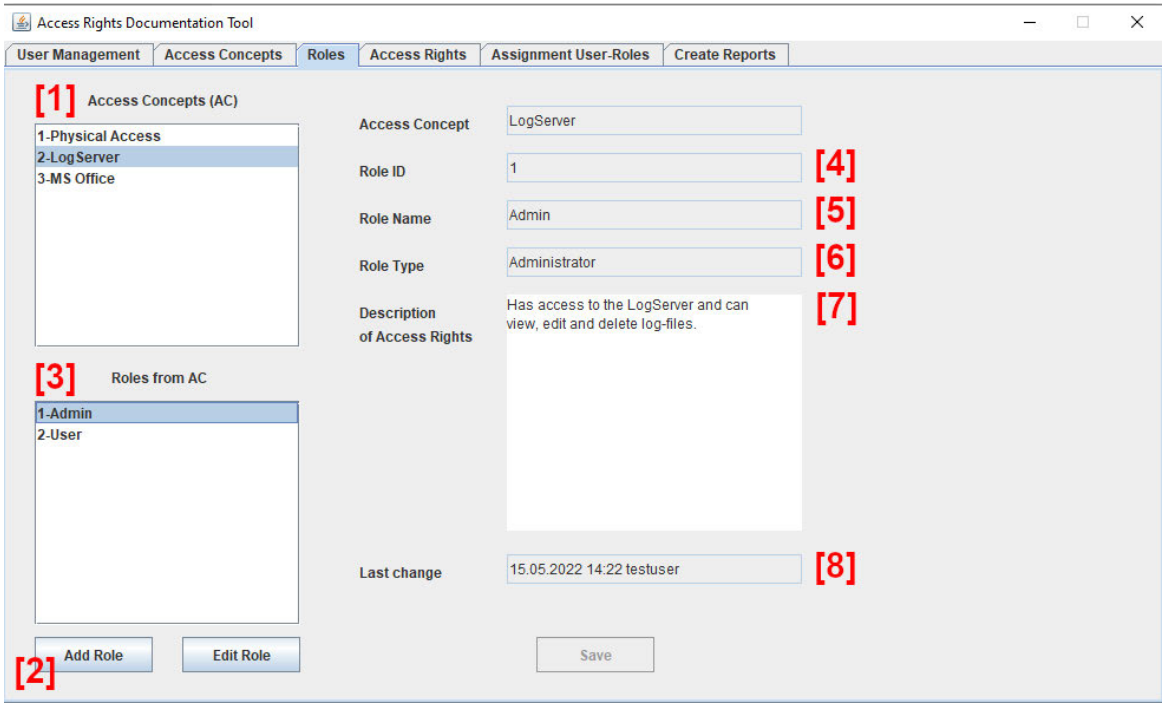


*Abbildung 7: Tab „Roles“ zum Anlegen und Bearbeiten von Rollen (eigene Abbildung aus der Software)*

### Access Rights

Dieser Reiter (siehe Abbildung 8) ermöglicht das Anlegen und Entfernen von Zugriffsrechten. Um ein neues Zugriffsrecht unter „Add Right“ [1] anzulegen, muss zuerst ein Berechtigungskonzept aus der Liste „Access Concept (AC)“ [2] und eine zugehörige Rolle unter der Liste „Roles from AC“ [3] ausgewählt werden. Das Anlegen von Zugriffsrechten erfolgt immer nur über eine ausgewählte Rolle aus einem Konzept. Um ein Zugriffsrecht erfolgreich zu erzeugen, werden Angaben unter „Right ID“ [4], „Access Right“ [5] und „Description“ [6] benötigt. Über „Right ID“ kann jedem Recht eine einmalige Nummer für die zugehörige Rolle zugewiesen werden, während unter „Access Right“ der eigentliche Name des Rechtes einzutragen ist. Die Anzeige der Zugriffsrechte wird bei der Auswahl eines Konzeptes oder einer Rolle aktualisiert. Änderungen an einem Dateneintrag werden wie gehabt unter „Last Change“ [7] angezeigt. Angelegte Access Rights können über den Button „Remove“ [8] wieder entfernt werden.

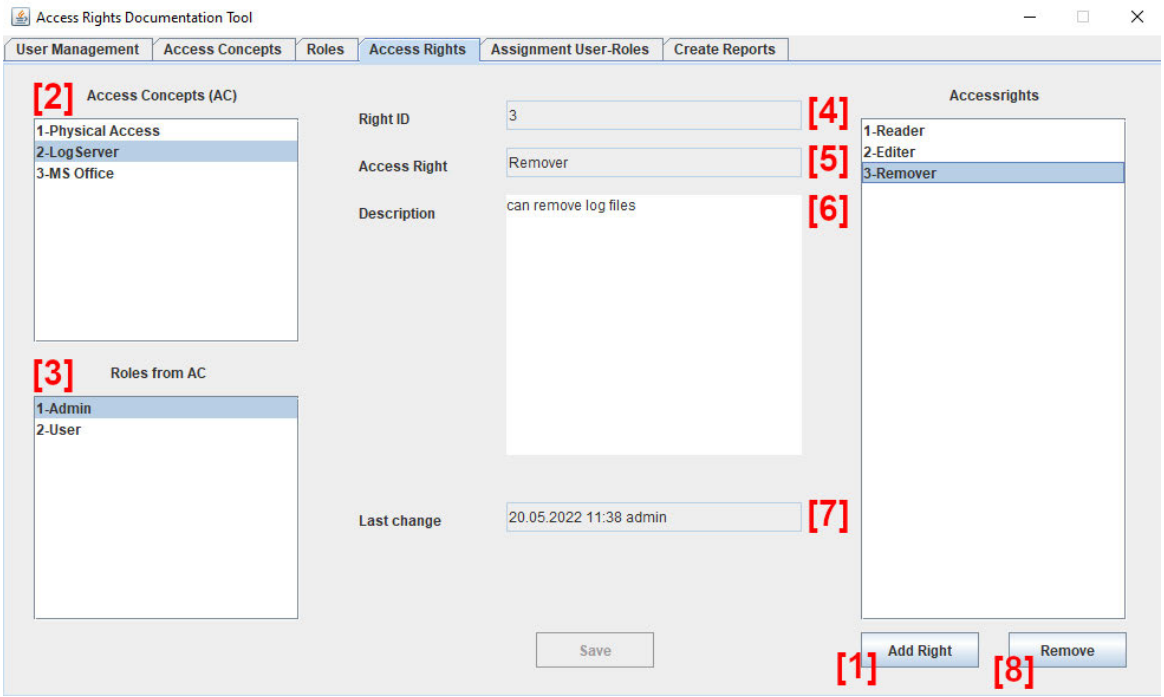


*Abbildung 8: Tab „Access Rights“ zum Anlegen und Löschen von Zugriffsrechten (eigene Abbildung aus der Software)*

### Assignment User-Roles

Abbildung 9 zeigt den Reiter „User Roles“, in dem die Zuordnung von angelegten Benutzern und Rollen erfolgt. Die GUI zeigt hier alle Listen aus den zuvor beschriebenen Reitern auf, da diese für die eindeutige Zuordnung notwendig sind.

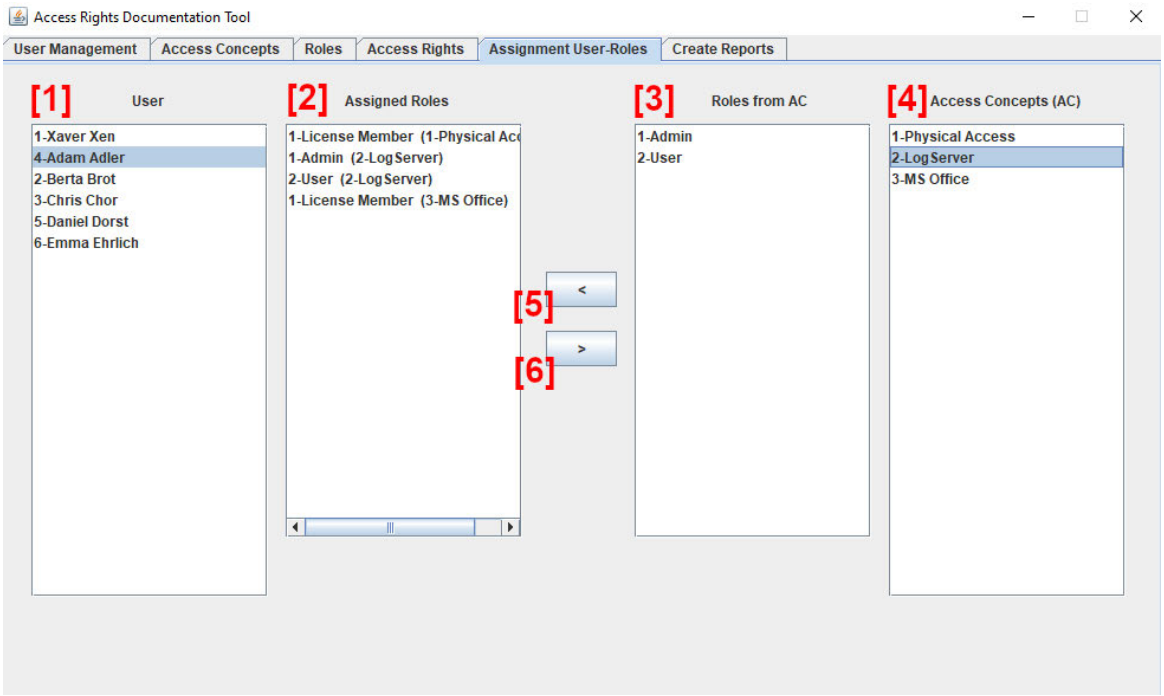


*Abbildung 9: Tab „Assignment User-Roles“ zum Zuweisen von Rollen an Benutzer (eigene Abbildung aus der Software)*

Die Liste „User“ [1] listet alle Benutzereinträge aus dem Reiter „User Management“ auf und gibt somit alle erstellten Benutzer aus. Unter „Assigned Roles“ [2] sind alle Rollen gelistet, welche der zuvor ausgewählte Benutzer bereits zugeordnet bekommen hat. Die Liste „Access Concepts (AC)“ [3] ermöglicht die Auswahl aller zuvor unter dem Reiter „Access Concepts“ angelegten Berechtigungskonzepte. Die zugehörigen Rollen eines aus dieser Liste ausgewählten AC werden in der Liste „Roles from AC“ [4] angezeigt.

Um einem Benutzer eine neue Rolle zuzuweisen, muss zuerst ein Benutzer unter „User“ und anschließend ein Berechtigungskonzept ausgewählt werden. Aus der sich aktualisierenden Rollen Liste kann dann die gewünschte Rolle ausgewählt und mittels dem „<“ Button [5] zugewiesen werden. Daraufhin wird die „Assinged Roles“ Liste des Benutzer aktualisiert und ihm die Rolle hinzugefügt. Das Entfernen einer bereits

existierenden Rolle bedarf hingegen ausschließlich der Auswahl des Benutzer und der Rolle sowie dem Klicken des „>“ Buttons [6]. Damit wird die Rolle sofort aus der Auflistung des Benutzers entfernt.

### Create Reports

Die Software ermöglicht auch das Erstellen von Reports aus den hinterlegten Datensätzen der Datenbank. Hierfür existiert der Reiter „Create Report“ (Abbildung 10). In diesem Bereich der Anwendung ist eine Auswahlliste und ein „Report Button“ eingefügt. In der Auswahlliste können die gewünschten Reports ausgewählt und mittels des „Report“ Buttons erzeugt werden. Mehr zur Reportfunktion wird in Kapitel 5 beschrieben.

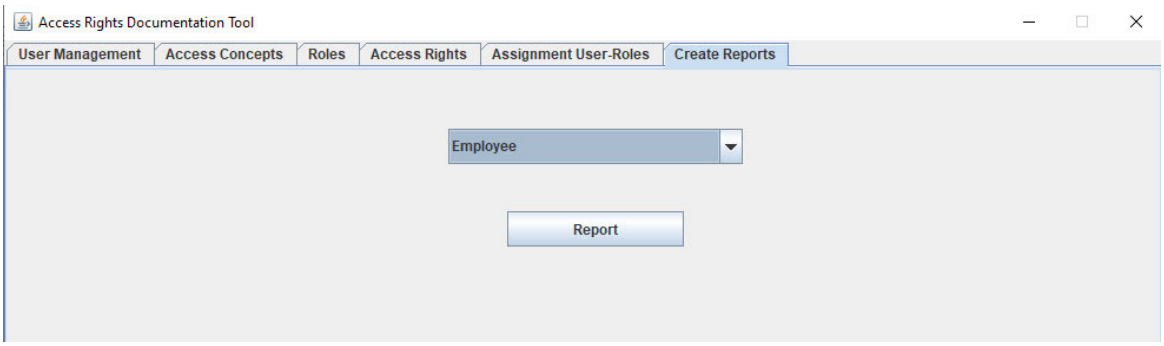


*Abbildung 10: Tab „Create Report“ zum Erstellen von Reportdateien (eigene Abbildung aus der Software)*

### 4.3. Vorstellung wichtiger Klassen

Durch die Nutzung von objektorientierter Programmierung wird eine übersichtliche und strukturierte Programmentwicklung ermöglicht. Abbildung 11 zeigt, wie die einzelnen Klassen mittels geordneter Packagestruktur in Eclipse angelegt und strukturiert wurden.

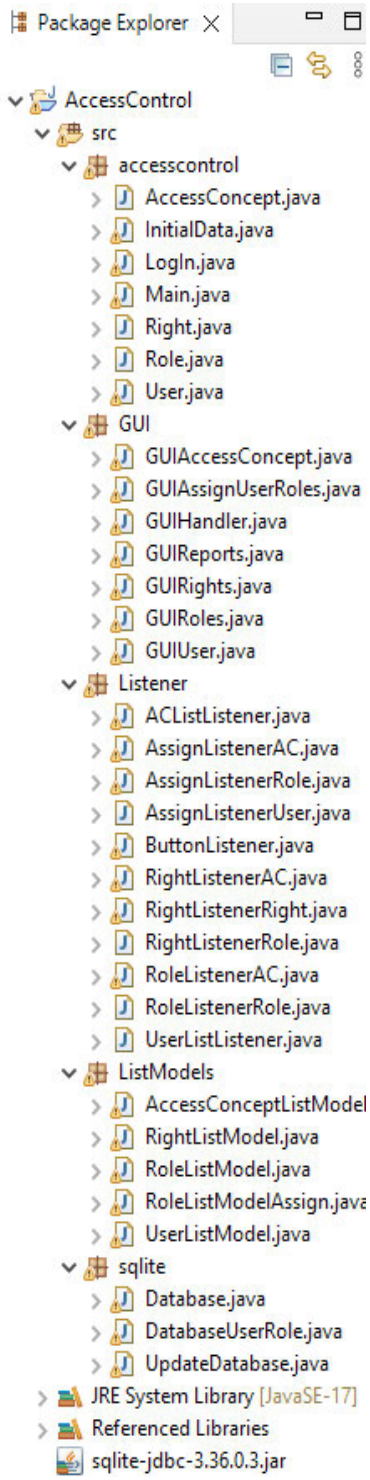


*Abbildung 11: Packagestruktur in der Entwicklungsumgebung (eigene Abbildung aus der Software)*

Die Strukturierung der Klassen erfolgt durch einzelne Packages, welche diese nach ihrem funktionalen Inhalt trennen. Im Package „accesscontrol“ werden grundlegende Objekt-Klassen gelistet sowie die Initialisierungs- und Main-Klasse zum Start des Programms. Das „GUI“ Package vereint alle Klassen, die für die Erzeugung der grafischen Benutzeroberfläche zuständig sind. Benutzerinteraktionen werden über die Klassen im „listener“ Package gesteuert. Im Package „ListModels“ werden alle erstellten ListModels verwaltet. Diese werden benötigt, um mit den erstellten Listen (JLists) zu interagieren. Über die einzelnen Models können die komplexeren Objekte (User, Access Concepts etc.) zu Listen hinzugefügt oder entfernt werden, damit diese später in der GUI ausgegeben werden können. Zuletzt folgt die Zusammenfassung der Klassen mit Datenbankinteraktionen unter „sqlite“. Diese regeln die Kommunikation mit der angebundenen Datenbank, indem sie Datenbankverbindungen herstellen und Daten mit SQL-Befehlen einfügen oder abrufen.

**UML-Klassendiagramm**

Um die Funktionalität der Klassen besser zu verstehen, werden nachfolgend einige der wichtigsten Klassen genauer vorgestellt. Dafür werden sie bezüglich ihres Aufbaus in Form eines UML-Diagrammes dargestellt und kurz in ihrer Funktion erläutert. Da sich einige Klassen in ihrem Aufbau ähneln, wird für diesen Fall jeweils eine Beispielklasse ausgewählt und vorgestellt. Die Funktionen der Klasse lassen sich auf andere Klassen mit ähnlichem Aufbau adaptieren. Für einige Klassen werden außerdem Funktionen mittels Pseudocode dargestellt um den Programmablauf zu verstehen.

**GUIUser**

In dieser Klasse wird die grafische Oberfläche unter dem Reiter „User Management“ erzeugt, in der neue Benutzer angelegt und verwaltet werden können. Hier werden alle Elemente wie Listen, Buttons und Textfelder erstellt und dem Hauptfenster hinzugefügt. Das JPanel dient als Container der einzelnen Elemente. Darauf lassen sich u.a. die JLabels zur Textanzeige und JTextFields zur Texteingabe einfügen. Die JList „userlist“ gibt alle erstellten User aus. Um eine geordnete Auflistung der User zu ermöglichen wird zudem ein JScrollPane verwendet. Dieses fügt automatisch Bildlaufleisten ein, sobald die Anzahl der User die Größe der Liste überschreibt. Zudem enthält die Klasse eine Methode *updateList()* zum Aktualisieren der Benutzerliste nach einer Änderung in der Datenbanktabelle. Folgende Klassenstruktur ergibt sich daraus (siehe Abbildung 12):

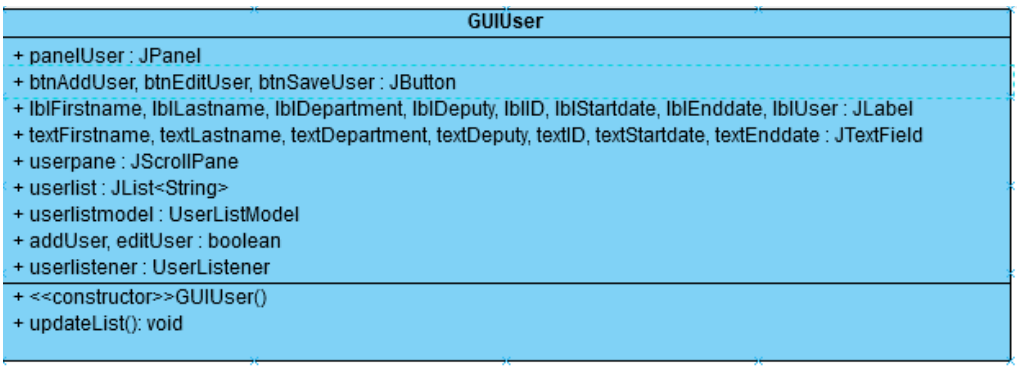


*Abbildung 12: UML Struktur Klasse GUIUser.java (eigene Abbildung)*

### ButtonListener

Diese Klasse steuert alle Aktionen, die nach Klicken eines in der GUI vorhandenen Buttons in den Reitern „User Management“, „Access Concepts“, „Roles“ und „Access Rights“ ausgelöst werden (siehe Abbildung 13). Dafür gibt es für jeden Button einzelne Methoden, mit denen unterschiedliche Aktionen ausgelöst werden. Die Methode *saveUser*() speichert z.B. die Daten, die beim Anlegen oder Editieren eines Benutzers in die Textfelder geschrieben wurden und ruft wiederum Methoden auf, welche die Datenbank und die GUI aktualisieren.

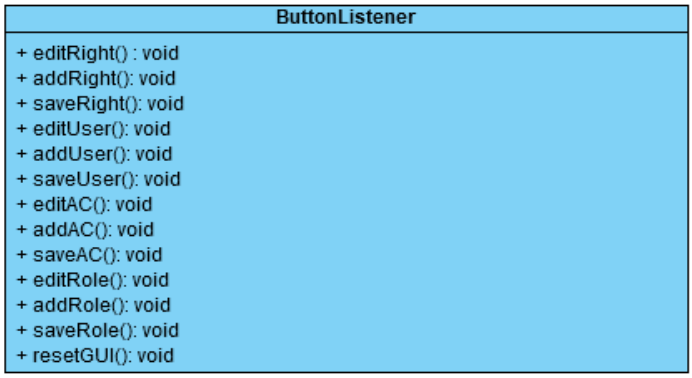


*Abbildung 13: UML Struktur Klasse ButtonListener.java (eigene Abbildung)*

### InitialData

In dieser Klasse wird die Initialisierung von Daten zum Programmstart und das Aktualisieren von ArrayLists über spezielle Methoden gesteuert (siehe Abbildung 14). Alle Einträge zu Usern, Access Concepts, Roles oder Access Rights, die in der Datenbank hinterlegt sind, werden zur Verarbeitung in eigene ArrayLists geschrieben, Listen, welche das Speichern von ganzen Objekten und ihrer Attribute erlauben. Eine ArrayList enthält immer nur Objekte eines Types (z.B. User). Die Einträge der ArrayList

werden dann in der zugehörigen Liste in der GUI ausgegeben. Sie dient in diesem Fall als eine Art Vermittler zwischen Datenbank und grafischer Ausgabe der Datenbankeinträge.

Das Aktualisieren einer ArrayList nach Änderung in der Datenbank wird vereinfacht in Abbildung 15 dargestellt. Alle Einträge aus der ArrayList werden gelöscht. Anschließend erfolgt die Datenbankabfrage für die Tabelle, in der neue Einträge eingefügt wurden. Die Daten der Abfrage werden erneut in die ArrayList geschrieben und können danach weiterverarbeitet werden (z.B. um die Änderung in der GUI anzuzeigen).

Weiterhin enthält diese Klasse die Funktion *getCurrentDate*(), um das aktuelle Datum zu speichern. Das Datum wird als Information verwendet, wenn ein eingeloggter Benutzer bspw. einen neues Access Concept anlegt oder editiert.

Dann wird die Methode aufgerufen und das aktuelle Datum unter „LastChange“ ausgegeben. Dies soll helfen nachzuvollziehen, welcher Nutzer zu welchem Zeitpunkt Änderungen im Repository vorgenommen hat.

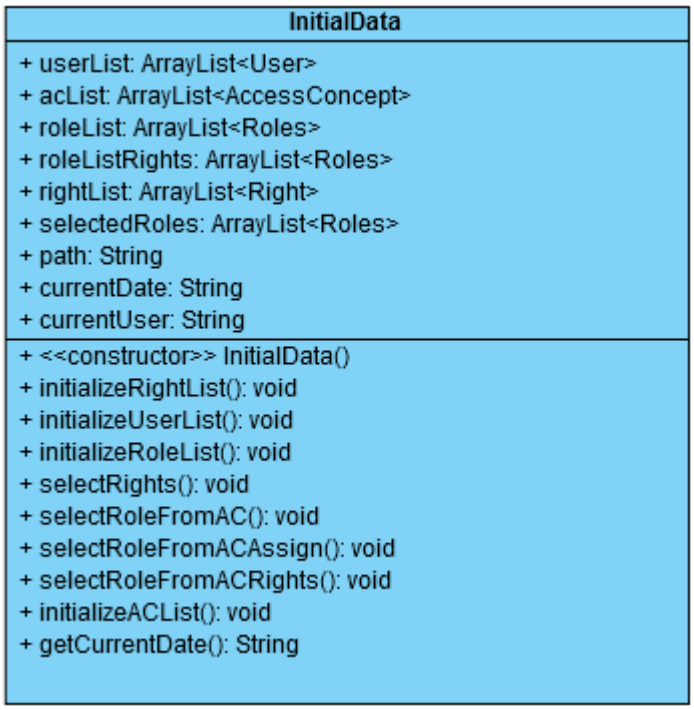


*Abbildung 14: UML Struktur Klasse InitialData.java (eigene Abbildung)*

```
public static void initializeUserList()
{
	//Lösche alle Einträge aus ArrayList "userlist"
	userlist.clear();

	Connection c = null;
	Statement stmt = null;

	try
	{
		createConnection();

		stmt = c.createStatement();
		ResultSet rs = stmt.executeQuery("SELECT * FROM user");

		while(rs.next())
		{
			//Füge "userlist" alle Einträge aus Datenbank hinzu
			userlist.add(new User(rs.getString(name), rs.getInt(id), .... ));
		}

		closeConnection();

	}catch(Exception e)
	{
		//...
	}
}
```

*Abbildung 15: Vereinfachtes Codebeispiel zum Aktualisieren einer ArrayList (eigene Abbildung)*

### UpdateDatabase

Diese Klasse steuert sowohl das Übermitteln neuer Daten in die Datenbank als auch das Aktualisieren bereits vorhandener Einträge durch unterschiedliche Methoden (siehe Abbildung 16). Ein neuer oder aktualisierter Datenbankeintrag erfolgt dabei immer, wenn Benutzer, Access Concepts, Rollen oder Access Rights angelegt oder bearbeitet werden. Für jede Möglichkeit steht eine Funktion bereit, mit der die Datenbankanbindung hergestellt und die Daten schließlich eingepflegt werden. Der Datenbankaufbau und das Einfügen von Daten in die Tabelle mittels Java-Code wird in vereinfachter Form für die Methode *addUser()* dargestellt (Siehe Abbildung 17).

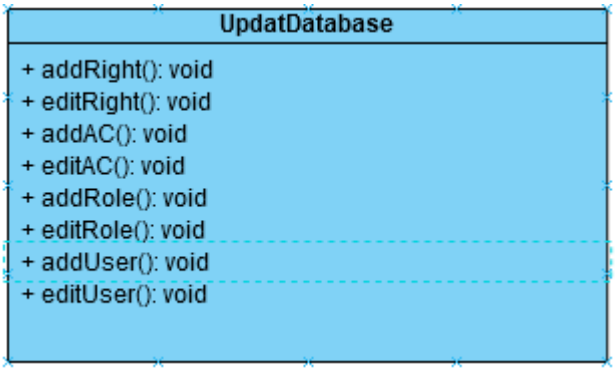


*Abbildung 16: UML Struktur Klasse UpdateDatabase.java (eigene Abbildung)*

```
public static void addUser()
{
        Connection c = null;
        Statement stmt = null;
        try
        {
                Class.forName(database);
                c = DriverManager.getConnection(path)
                stmt = c.createStatement();

                String sql = "INSERT INTO user (a1, a2, ....)   VALUES(v1, v2, ...);";
                stmt.executeUpdate(sql);

                stmt.close();
                c.commit();
                c.close();
        }catch(Exception e)
        {
                //...
        }
}
```

*Abbildung 17: Vereinfachtes Codebeispiel zum Einfügen neuer Daten in eine Datenbanktabelle (eigene Abbildung)*

## 4.4. Datenverarbeitung

Bei der gezeigten Software erfolgt die Datenverarbeitung auf zwei verschiedene Arten: während der Laufzeit im Programm selbst und mittels Datenübertragung über eine SQL-Datenbank.

Während der Laufzeit werden Daten innerhalb des Programms mithilfe von aufgerufenen Methoden übertragen. Dazu gehören alle Methoden, die in separaten Klasse aufgerufen und abgehandelt werden. Objekte und Methoden, die zum Aufruf in anderen Klassen benutzt werden, sind im Code als *static* deklariert. Dieser Ausdruck ermöglicht es, Daten von einer Klasse in einer anderen Klasse oder Methode aufzurufen. In den Objektklassen werden Benutzerobjekte mit allen benötigten Attributen erzeugt und Methoden bereitgestellt, die den Aufruf der erzeugten Daten in anderen Klassen ermöglichen. Dafür werden entsprechende get()- und set()-Methoden deklariert. Über die get()-Methoden können die gewünschten Daten des Objektes in anderen Klassen und Methoden aufgerufen werden. Über die set()-Methoden können Daten des Objektes über andere Klassen und Methoden verändert werden. Abbildung 18 zeigt beispielhaft die get()- und set()-Methoden der Klasse „User.java“ für das Attribut „Firstname“ eines angelegten Benutzers. In der set()-Methode wird beim Aufruf ein String übergeben, welcher den aktuellen Eintrag überschreibt. Die get()-Methode liefert über das *return* Statement den Wert zurück, der aktuell unter „firstname“ hinterlegt ist.

```
public String firstname;

public User()
{
        //....
}


public void setFirstName(String f)      //set()-Methode
{
        firstname = f;
}


public String getFirstName()            //get()-Methoden
{
        return firstname;
}
```

*Abbildung 18: Codebeispiel für get()- und set()-Methoden (eigene Abbildung)*

Der Großteil der Datenverarbeitung in der Software erfolgt über die zugehörige SQL-Datenbank. Der Einsatz einer solchen Datenbank ermöglicht es, Daten über die Laufzeit des Programms hinaus zu speichern und wieder aufzurufen. Hierfür wird ein relationales Datenbanksystem verwendet, ein System, welches die Verbindung von Daten unterschiedlicher Tabellen mittels eindeutiger Schlüssel ermöglicht.

Die gezeigte Software verwendet zur Speicherung der Daten eine SQLite Datenbank. SQLite zählt als „das verbreitetste und meistverwendete Datenbanksystem der Welt“ (Wikipedia, 26.02.2021, https://de.wikipedia.org/wiki/SQLite), da es „einen Großteil der im SQL-92-Standard festgelegten SQL-Sprachbefehle“ (Wikipedia, 26.02.2021, ebd.) unterstützt. Bei der Auswahl von SQLite spielte vor allem die vollständige Integrierbarkeit der Datenbank in die Anwendung eine Rolle. Im Gegensatz zu anderen Datenbanklösungen wie MySQL muss hierfür keine laufende Serververbindung hergestellt werden. Damit ist die Datenbank flexibel und kann ohne erhöhten Aufwand in die Anwendung integriert werden.

Für die Speicherung der verschiedenen Objekte wurden einzelne Datenbanktabellen angelegt. Diese werden beim Anlegen und Bearbeiten von Objekten mit Informationen der zugehörigen Eingabefelder in der GUI gefüllt. Als Grundlage für die Datenbankstruktur der Objekte dient das aufgezeigte Modell in Abbildung 19. Das Model beschreibt die einzelnen Tabellen für User, Access Concepts, Roles und Access Rights mit ihren Attributen und ihrer gegenseitigen Zuordnung.

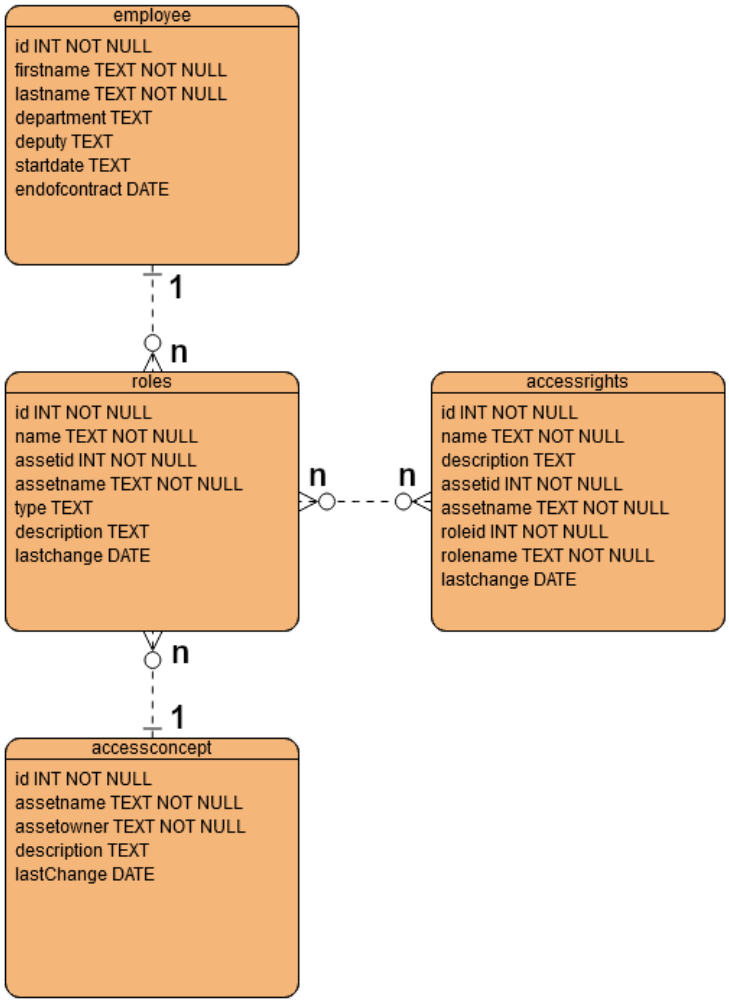


*Abbildung 19: Datenbankmodell (eigene Abbildung)*

Zusätzlich wird für jeden neu angelegten Benutzer eine separate Datenbanktabelle erstellt, mit derer alle zugewiesenen Rollen des Nutzers gespeichert werden. Die Tabelle enthält Informationen zum Access Concept und der zugehörigen Rolle und wird beim Zuweisen oder Entfernen einer Rolle im Reiter „Assignment User-Roles“ befüllt. Daraus ergibt sich die gezeigte Tabellenstruktur (Abbildung 20):

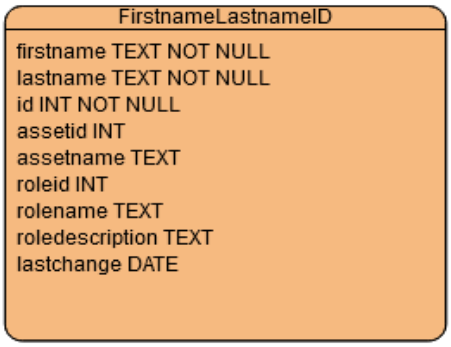


*Abbildung 20: Tabellenstruktur Rollenzuweisung Benutzer (eigene Abbildung)*

Das Abrufen und Einfügen von Datensätzen erfolgt mittels SQL-Befehlen. Diese erlauben es, neue Tabellen zu erzeugen und diese mit Datensätzen zu füllen oder zu verändern. Um die benötigten SQL-Befehle im Programm zur Datenverarbeitung zu verwenden, muss zuerst die Verbindung von Java und SQLite erfolgen. Dafür muss der aktuelle *SQLite JDBC* Treiber in das Projekt importiert werden. Für die Software wurde die Version *jdbc-3.36.0.3* verwendet.

Im Programm existieren eigene Klassen zur Erzeugung der Datenbankanbindung und dem Einfügen oder Ändern von Einträgen. Der Aufbau der Klasse „UpdateDatabase.java“ wurde bereits in Abschnitt 4.3. (Seite 58) vorgestellt. In dieser Klasse finden sich Methoden zum Einfügen und Ändern von Datensätzen. Neben dem Einfügen und Ändern müssen auch Tabellen erzeugt, bearbeitet oder gelöscht werden sowie gezielte Abfragen stattfinden. Folgende vereinfachte Programmausschnitte (Abbildung 21) zeigen weitere SQL-Befehle, die in einzelne Java-Methoden integriert wurden:

```
//Erzeuge neue Tabelle
public static void createTable()
{
	Connection c = null;
	Statement stmt = null;

	try
	{
		getConnection();
		stmt = c.createStatement();

		//erzeuge Tabelle tablename mit Ganzzahlwert id und Textwert text
		String sql = "CREATE TABLE tablename (id INT, text TEXT)";

		stmt.executeUpdate(sql);
		closeConnection();

	}catch(Exception e)
	{
		//...
	}

}


//Update Tabelle
public static void editEntry()
{
	Connection c = null;
	Statement stmt = null;

	try
	{
		getConnection();
		stmt = c.createStatement();

		//Überschreibe Inhalt aus text mit Ihhalt aus textfield1 für Eintrag mit id = konvertierter Inhalt aus textfield2
		String sql = "UPDATE tablename SET text = " + Classname.textfield1.getText() +
			     "WHERE id = " + Integer.parseInt(Classname.textfield2.getText())+";";

		stmt.executeUpdate(sql);
		closeConnection();

	}catch(Exception e)
	{
		//...
	}

}
```

*Abbildung 21: Codebeispiele für SQL-Aufrufe in Java-Code (eigene Abbildung)*

## 5. Reportfunktion zum Erstellen von Dokumentationsdaten

Die Software ermöglicht das Erstellen von Reportdateien zur systematischen Ausgabe von Daten aus der Anwendung. Im Reiter „Create Report“ können über die Auswahlliste verschiedene Reports erstellt werden. Diese Reports dienen als Beispiel für das erzeugen von Dokumentationsdaten und könnten beim zweckmäßigen Einsatz der Software noch um weitere Reports erweitert werden. Folgende in Abbildung 22 gezeigte Reports stehen standardmäßig zur Verfügung:

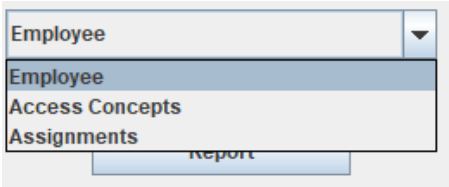


*Abbildung 22: Auswahlmenü zur Erstellung von Reports (eigene Abbildung)*

Reports ermöglichen es, Daten aus der Anwendung in separate Dateien zu schreiben, wie Text- oder CSV-Dateien. Die Reports der entwickelten Software werden in Form von CSV-Dateien gespeichert, damit sie in Tabellenkalkulationsprogrammen wie MS Excel oder OpenOffice Calc, weiter bearbeitet werden können. Um eine Übersicht aller vergebenen Rollen an Benutzer zu erhalten, wird in der Auswahlliste der Report „Assignments“ ausgewählt. Dieser erzeugt eine CSV-Datei und listet alle Benutzer chronologisch inklusive aller zum Zeitpunkt des Reports zugewiesenen Rollen untereinander auf. Der CSV-Datei wird zu Beginn ein Datumsstempel hinzugefügt, um den Zeitpunkt der Erstellung zu dokumentieren. Eine erstellte CSV-Datei kann wie folgt aussehen (Abbildung 23):

| | |
|---|---|
| 1 | Date of Report: 17.05.2022 13:51 |
| 2 | ID,Firstname,Lastname,AssetID,Assetname,RoleID,Rolename |
| 3 | 1,Xaver,Xen,1,Physical Access,1,License Member |
| 4 | 4,Adam,Adler,1,Physical Access,1,License Member |
| 5 | 4,Adam,Adler,2,LogServer,1,Admin |
| 6 | 4,Adam,Adler,2,LogServer,2,User |
| 7 | 4,Adam,Adler,3,MS Office,1,License Member |
| 8 | 2,Berta,Brot,3,MS Office,1,Licence Member |
| 9 | 2,Berta,Brot,1,Physical Access,2,Personal-Member |
| 10 | 3,Chris,Chor,3,MS Office,1,Licence Member |
| 11 | 3,Chris,Chor,1,Physical Access,2,Personal-Member |
| 12 | 5,Daniel,Dorst,3,MS Office,1,Licence Member |
| 13 | 5,Daniel,Dorst,1,Physical Access,1,IT-Member |
| 14 | 5,Daniel,Dorst,2,LogServer,2,User |
| 15 | 6,Emma,Ehrlich,3,MS Office,1,Licence Member |
| 16 | 6,Emma,Ehrlich,1,Physical Access,2,Personal-Member |
| 17 | |

*Abbildung 23: CSV-Datei zum Report der Rollenauflistung*

*(eigene Abbildung)*

Über die Funktion *reportAllAssignments()* werden mit Hilfe eines PrintWriters, die Daten aller Benutzer und derer Rollen in die Datei geschrieben. Hierfür wird eine Datenbankverbindung hergestellt, die abgefragten Daten in eine ArrayList geschrieben und diese wiederum in die CSV-Datei überführt. Der Beispielcode aus Abbildung 24 stellt die dafür genutzte Funktion vereinfacht dar:

```
public void reportAllAssignments() throws FileNotFoundException
{
        ArrayList<String> list = new ArrayList<>();
        File file = new File(ouputfile.csv);
        PrintWriter pw = new PrintWriter(file);

        Connection C = null;
        Statement stmt = null;

        try
        {
                getConnection();

                stmt c.createStatement();

                //Hole alle Einträge aus der Tabelle tabelAssignments
                ResultSet rs = stmt.executeQuery("SELECT * FROM tableAssignments");

                //Hole für jedes Element aus der Tabelle die Attribute und schreibe sie in die ArrayList mit "," getrennt
                while(rs.next())
                {
                        int value1 = rs.getInt("value1");
                        String value2 = rs.getString("value2");
                        //...

                        list.add(new String(value1 + "," + value2 + "," + ...));
                }

                closeConnection();

        }catch(Exception e)
        {
                //...
        }

        //schreibe die Inhalte der ArrayList in die Datei outputfile.csv
        for(int i = 0; i< list.size(); i++)
        {
                pw.println(list.get(i).toString());
        }

        pw.close();
        list.clear();
}
```

*Abbildung 24: Beispielfunktion zum Erstellen einer CSV-Datei (eigene Abbildung)*

# 6. Zusammenfassung

Die Arbeit geht der Frage nach, warum es notwendig sein kann, Zugriffsrechte zu verwalten und zu dokumentieren und wie dies mit Hilfe einer erstellten Software erleichtert werden kann. Dafür wird beschrieben, in welchen Bereich die Thematik einzuordnen ist und welche Grundlagen dafür definiert werden müssen. Anschließend wird gezeigt, welchen Anforderungen eine solche Software erfüllen muss und wie sich diese hinsichtlich der beschriebenen Anforderungen erstellen und einsetzen lässt.

Zusammenfassend lässt sich feststellen, dass eingesetzte Software im Gebiet der Berechtigungsvergabe hilft, den Risikomanagementprozess in einem Unternehmen zu unterstützen. Der Einsatz eines Identity Repositories ermöglicht es, Daten zu Zugriffsberechtigungen zu erfassen und zentralisiert zu verwalten. Diese Dokumentation vereinfacht nicht nur den Rechtevergabeprozess, sondern unterstützt durch die Nachvollziehbarkeit von vorhandenen Benutzern und Rechten die Kontrollmöglichkeiten rund um alle hinterlegten Zugriffsberechtigungen. Eine geordnete Dokumentation von Benutzern, Berechtigungskonzepten, Rollen und Zugriffsrechten im Rahmen der RBAC stellt zu jedem Zeitpunkt sicher, für welche Daten und Systeme in einem Unternehmen Zugriffe benötigt werden. Für diese Daten können dann gezielt alle Zugriffe beschrieben und für jeden Benutzer hinterlegt werden. Damit werden nicht nur die amtlichen Vorgaben von Behörden zur Dokumentationspflicht umgesetzt, sondern auch der Überblick über alle Berechtigungen und deren Verteilung an die Benutzer behalten.

Die entwickelte Software zeigt, wie eine solche Anwendung zur Dokumentation und Verwaltung von Zugriffsrechten aussehen kann. Eine geordnete grafische Oberfläche mit einer einfachen Handhabung sind ebenso wichtiger Bestandteil wie die Implementierung von Funktionen zum Anlegen und Bearbeiten von Benutzern, Berechtigungskonzepten oder Rollen. Sicherheitsmaßnahmen, wie Log-In Funktionen, sichern die Anwendung vor unautorisierten Zugriffen und schützen dadurch sensible Informationen vor unberechtigten Einblicken oder Manipulationen. Es wird gezeigt, wie die Zuweisung von Rollen an einzelne zuvor angelegte Benutzer erfolgt und wie Dokumentationsdaten mittels Reportfunktion zur weiteren Verarbeitung und Kontrolle aus der Anwendung heraus exportiert werden können. Dafür wird der Aufbau einzelner Klassen dargestellt und einzelne Beispiel-Methoden zur Datenverarbeitung beschrieben. Damit die Software alle notwendigen Daten hinterlegen und bereitstellen kann, muss eine passende Datenbankstruktur erstellt und mit der Anwendung verbunden werden. Dafür werden Beispielfunktionen mit Pseudocode aufgezeigt und die funktionalen Abläufe zu verdeutlichen. Datenbanklösungen wie SQLite ermöglichen, durch ihre einfache Integrierbarkeit, eine schnelle Einbindung von Daten in die Software.

Es ist festzuhalten, dass es sich bei der vorgestellten Software um eine exemplarische Anwendung eines Identity Repository handelt, welche lediglich grundlegende Funktionen implementiert hat. Um die Software gezielt in einem Unternehmen zur Verwaltung und Dokumentation einsetzen zu können, müssten noch weitere Funktionen eingefügt werden. Dazu zählt z.B. die Erweiterung der Reportfunktion, um weitere Auswahlkriterien und die Möglichkeit, noch mehr Informationen (z.B. zu Mitarbeiterkonten) in der

Datenbank hinterlegen zu können. Des Weiteren ist zu vermerken, dass kein ausgiebiges Testverfahren eingesetzt wurde, um das Programm auf alle Fehler zu prüfen oder Laufzeitanalysen durchzuführen. Für den endgültigen Einsatz müssten mögliche Fehler gründlicher ausgeschlossen werden und noch mehr Fehlerbehandlungen implementiert sein. Außerdem würde der Einsatz einer verschlüsselten Datenbank zu Sicherung der Datenbankinhalte Sinn machen. Hierauf wurde aufgrund der exemplarischen Darstellung verzichtet, wohl wissend, dass eine solche Absicherung im richtigen Einsatz von Vorteil ist.

## Literaturverzeichnis


Bundeskriminalamt: Innentäter in Unternehmen 2. URL: https://www.bka.de/SharedDocs/Downloads/DE/Publikationen/Publikationsreihen/Forschungsergebnisse/2020KKFAktuell_InnentaeterinUnternehmen.pdf?__blob=publicationFile&v=2, aufgerufen am 27.04.2022

IONOS: Role Based Access Control (RBAC): Wie funktioniert die rollenbasierte Zugriffskontrolle. URL: https://www.ionos.de/digitalguide/server/sicherheit/was-ist-role-based-access-control-rbac, aufgerufen am 25.04.2022

BaFin: Mindestanforderungen an das Risikomanagement (MaRisk). URL: https://www.bafin.de/SharedDocs/Veroeffentlichungen/DE/Rundschreiben/2021/rs_1021_MaRisk_BA.html, aufgerufen am 26.04.2022

Sharon Shea: Mandatory Acces Control (MAC). URL: https://www.computerweekly.com/de/definition/Mandatory-Access-Control-MAC, aufgerufen am 28.04.2022

James A. Martin; Florian Maier: Was Sie über IAM wissen müssen. URL: https://www.computerwoche.de/a/was-sie-ueber-iam-wissen-muessen, aufgerufen am 28.04.2022

Wiktionary: Diskretionär. URL: https://de.wiktionary.org/wiki/diskretion%C3%A4r, aufgerufen am 01.05.2022

Techopedia: Discretionary Access Control (DAC). URL: https://www.techopedia.com/definition/229/discretionary-access-control-dac, aufgerufen am 01.05.2022

Bundesamt für Sicherheit in der Informationstechnik: ORP.4: Identitäts- und Berechtigungsmanagement (Edition 2020). URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium_Einzel_PDFs/02_ORP_Organisation_und_Personal/ORP_4_Identitaets_und_Berechtigungsmanagement_Editon_2020.pdf?, aufgerufen am 02.05.2022

Tobias Scheible, IT-Sicherheit Grundlagen: Schutzziele. URL: https://scheible.it/it-sicherheit-grundlagen-schutzziele, aufgerufen am 01.05.2022

Prof. Dr. Oliver Budzinski: Kreditinstitute. URL: https://wirtschaftslexikon.gabler.de/definition/kreditinstitute-37317, aufgerufen am 02.05.2022

Initiative D21: Homeschooling in Zeiten von Corona. URL: https://initiatived21.de/app/uploads/2020/08/homeschooling_ergebnisse_egovernment-monitor-2020.pdf, aufgerufen am 03.05.2022

Solarwinds: Access Management System. URL: https://www.solarwinds.com/de/access-rights-manager/access-management-system, aufgerufen am 08.05.2022

Wikipedia: SQLite. URL: https://de.wikipedia.org/wiki/SQLite, aufgerufen am 15.05.2022

## Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe. Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach Publikationen oder Vorträgen anderer Autoren entnommen sind, habe ich als solche kenntlich gemacht. Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt oder anderweitig veröffentlicht.

| Mittweida, 23.05.2022 | Eric Heising | |
|---|---|---|
| Ort, Datum | Vollständiger Name | Unterschrift |

**Nutzungs- und Verwertungsrechte**

Ich übertrage zusätzliche Nutzungs- und Verwertungsrechte für die vorliegende Arbeit und allen damit in Zusammenhang stehenden Daten auf Grundlage der Creative Commons Lizenz "CC0" an alle genannten Betreuer dieser Arbeit.

| Mittweida, 23.05.2022 | |
|---|---|
| Ort, Datum | Unterschrift |