Projektarbeit 2
Programmierung einer Webbasierten Medikamentenakte in HTML 5

Fachhochschule Dortmund

Fachbereich Informationstechnik

Eingereicht von: Sophia Tholen
Matr.Nr.: 7205175

Betreuer: B.Sc. Kevin Köster

Prüfer: Prof. Dr. Burkhard Igel

Zweitprüfer: Dr. Georg Pietrek

Datum der Abgabe:

1. Abstract

Inhaltsverzeichnis

2. Einleitung

In der vorliegenden Arbeit wird die Entwicklung einer webbasierten Medikamentenakte vorgestellt, die als theoretisches Beispiel für den Einsatz in Krankenhäusern dient. Ziel ist es, bei einem vorliegenden Pneumonie-Erreger das passende Medikament sowie die korrekte Dosis und Therapiedauer anzuzeigen. Die Logik des Programms wurde in Java mithilfe von Spring Boot realisiert, während die Darstellung der Webseiten in HTML5 unter Verwendung des Thymeleaf-Frameworks implementiert wurde, um eine interaktive Web-Oberfläche zu bieten.

Die Verwendung des Spring-Frameworks in Kombination mit Thymeleaf unterstützt eine flexible und übersichtliche Strukturierung der Datenverarbeitung sowie die dynamische Aktualisierung der Web-Inhalte. Zur Effizienzsteigerung wurde während der Implementierung ein KI-gestützter Chatbot verwendet, um den Programmcodes zu optimieren und allgemeine Coding-Prozesse zu unterstützen. Das System demonstriert damit, wie moderne Entwicklungsansätze zur Automatisierung von Diagnoseprozessen beitragen und medikamentöse Empfehlungen im Klinikalltag vereinfachen können.

2.1. Antiinfektiva Therapie bei Pneumonie

Antiinfektiva-Therapien, insbesondere der rationale Einsatz von Antibiotika, sind essenziell für die effektive Behandlung bakterieller Infektionen wie der Pneumonie. Ein zentraler Bestandteil dieser Bemühungen ist das Antibiotic Stewardship (ABS), das darauf abzielt, die Qualität der Antiinfektiva-Behandlung hinsichtlich Auswahl, Dosierung, Applikation und Anwendungsdauer zu sichern, um das bestmögliche klinische Ergebnis zu erreichen[6].

In der vorliegenden Arbeit wurde eine Datenbank entwickelt, die die Daten eines ABS-Leitfadens nutzt, um Therapieempfehlungen für verschiedene Formen der Pneumonie in Abhängigkeit von klinischen Symptomen und mikrobiologischen Befunden bereitzustellen. Der Leitfaden unterscheidet zwischen ambulant erworbener Pneumonie (CAP) und nosokomialer Pneumonie (HAP), die jeweils unterschiedliche Erregerprofile und Therapieansätze erfordern.

Bei der ambulant erworbenen Pneumonie (CAP) erfolgt die Therapie unter Berücksichtigung des CRB-65-Scores oder der Minorkriterien des ATS-Scores. Ein CRB-65-Score von 0 ermöglicht in der Regel eine ambulante Behandlung mit einer Letalität von 0–2(%). Ein Score von 1 erfordert eine stationäre Behandlung auf der Normalstation (Letalität: 6–13(%)), während ein Score von über 2 eine Behandlung auf der Intensivstation indiziert (Letalität: 23–34(%)). Die nosokomiale Pneumonie (HAP) tritt häufig bei hospitalisierten Patienten auf und wird oft durch multiresistente gramnegative Erreger verursacht. Die kalkulierte Initialtherapie erfolgt je nach Risikofaktoren mit Antibiotika wie Piperacillin/Tazobactam, Cefepim oder in schweren Fällen mit Carbapenemen. Bei erhöhtem Risiko für Methicillin-resistenten Staphylococcus aureus (MRSA) wird zusätzlich eine Therapie mit Vancomycin oder Linezolid empfohlen[7].

Die in dieser Arbeit entwickelte Datenbank integriert diese evidenzbasierten Leitlinien

und ermöglicht eine dynamische Therapieplanung für Pneumonien. Durch die Berücksichtigung individueller Patientendaten unterstützt sie Ärzte bei der Auswahl des geeigneten Antibiotikums und trägt dazu bei, die Effektivität der Behandlung zu maximieren und die Entstehung weiterer Resistenzen zu minimieren.

2.2. Digitalisierung in Krankenhäusern

Die Digitalisierung im Gesundheitswesen spielt eine entscheidende Rolle bei der Verbesserung der medizinischen Versorgung und der Effizienz von Krankenhäusern. Durch den Einsatz digitaler Technologien können wichtige Dokumente zentral gespeichert werden, wodurch das Risiko des Verlusts von Unterlagen minimiert wird. Zudem ermöglichen Videosprechstunden eine flexible Patientenbetreuung, die den persönlichen Arztbesuch in bestimmten Fällen ersetzen kann. Ein zentrales Element der Digitalisierung ist die elektronische Patientenakte (ePA), die persönliche Gesundheitsinformationen digital bündelt und somit Transparenz sowie einen schnellen Zugriff für medizinisches Personal gewährleistet. Dies führt zu einer verbesserten Kommunikation zwischen verschiedenen Leistungserbringern und unterstützt eine koordinierte Behandlung[1].

Darüber hinaus bieten digitale Gesundheitsanwendungen (DiGA), auch als Äpps auf Rezept"bekannt, Unterstützung bei der Behandlung diverser Erkrankungen. Sie vermitteln Wissen, veranschaulichen Zusammenhänge oder leiten bei Übungen an, was zur Verbesserung des Gesundheitszustandes beitragen kann[4].
Telemedizinische Angebote ermöglichen es, medizinische Leistungen aus der Ferne in Anspruch zu nehmen, beispielsweise durch Videosprechstunden. Dies ist besonders vorteilhaft für Patienten mit eingeschränkter Mobilität oder in ländlichen Regionen[2]. In der vorliegenden Arbeit wurde eine Datenbank entwickelt, die die Daten eines ABS-Leitfadens nutzt, um Therapieempfehlungen für verschiedene Formen der Pneumonie in Abhängigkeit von klinischen Symptomen und mikrobiologischen Befunden bereitzustellen.
Durch die Digitalisierung dieser Informationen können Ärzte evidenzbasierte Entscheidungen treffen und die Behandlung individuell anpassen. Dies unterstreicht die Bedeutung der Digitalisierung im Krankenhausumfeld, da sie nicht nur die Effizienz steigert, sondern auch die Qualität der Patientenversorgung erheblich verbessert.

3. Verwendete Technologien

Folgende Technologien wurden bei der Entwicklung des Projektes genutzt:

Kategorie	**Technologie**
Containerdienst	Docker
Entwicklungsrechner	HP ZBook Firefly 15 inch G8 Mobile Workstation PC
Entwicklungsumgebung	IntelliJ IDEA 2024.3.1.1 Ultimate
Java Version	Java 21
Software-Projektmanagement	Maven
Framework	Spring AI 1.0.0-M1
Datenbank	PostgreSQL 42.7.3
Bibliotheken	Lombok 1.18.30, Thymeleaf 3.3.2
Webtechnologie	HTML5
Identitäts- und Zugriffsmanagement	Keycloak 26.1.4

Tabelle 1: Entwicklungsumgebung und verwendete Technologien

Einige der verwendeten Technologien, wie Keycloak, HTML5, Docker und PostgresSQL, spielen eine zentrale Rolle in diesem Projekt. In den folgenden Kapiteln werden diese Technologien besonders hervorgehoben und ihre Anwendung sowie ihr Mehrwert detailliert erläutert.

3.1. Keycloak

In der vorliegenden Arbeit wurde Keycloak genutzt, um die Anmeldung eines Nutzers der Datenbank getrennt von der eigentlichen Datenbankverwaltung zu gestalten. Dies wurde umgesetzt, um eine höhere Sicherheit und Flexibilität im Authentifizierungsprozess zu gewährleisten. Keycloak bietet zahlreiche Vorteile, darunter die zentrale Verwaltung von Benutzeridentitäten, die Unterstützung für Single Sign-On (SSO), ein Verfahren, bei dem sich Benutzer einmal authentifizieren und anschließend ohne erneute Anmeldung auf verschiedene Anwendungen und Dienste zugreifen können, sowie eine einfache Integration in bestehende Systeme durch die Nutzung standardisierter Protokolle wie OpenID Connect, OAuth 2.0 und SAML. Single Sign-On (SSO) ist ein Authentifizierungsverfahren, das es ermöglicht, dass sich Benutzer einmal anmelden und danach auf mehrere Anwendungen oder Dienste zugreifen können, ohne sich erneut authentifizieren zu müssen. Dies reduziert die Anzahl der Passworteingaben und erleichtert die Verwaltung von Zugangsdaten erheblich[5].

Gleichzeitig erhöht es die Sicherheit, da weniger Anmeldevorgänge notwendig sind, wodurch das Risiko von Phishing und anderen Angriffen verringert wird. Durch diese Trennung der Benutzeranmeldung von der Datenbankverwaltung wird verhindert, dass Anwendungen direkt auf sensible Nutzerdaten zugreifen müssen, was das Risiko von Sicherheitslücken minimiert. Darüber hinaus erlaubt Keycloak eine granulare Rechtevergabe und erleichtert die Verwaltung von Berechtigungen über eine webbasierte Konsole. In der Implementierung wurde Keycloak in das HTML5-Programm eingebunden, indem eine neue Klasse erstellt wurde, die zur Konfiguration der sicherheitsrelevanten Aspekte dient[8]. Diese Klasse definiert die Authentifizierungs- und Autorisierungsmechanismen, die für die Kommunikation zwischen der Anwendung und dem Identitätsmanagementsystem genutzt werden.

Zusätzlich wurde in den ‘application.properties‘ des Programms die URL hinterlegt, die auf den Keycloak-Server verweist, sodass die Anwendung bei der Authentifizierung automatisch die hinterlegten Identitätsprovider und Benutzerverzeichnisse nutzt (siehe Abbildung 1).

application.properties

```
spring.application.name=ABS-Manager
server.port=8080

spring.ai.openai.api-key=sk-proj-XOEfAPtPX6qEU3sMfzb8EhpfAzvAvcR-HC4jv92l-xrTf_90D43mbED4BtI3egA5FAX8IMk91m
spring.ai.openai.chat.model=gpt-4o-mini
spring.ai.openai.chat.temperature=0.9
spring.ai.openai.chat.max-tokens=100

spring.ai.ollama.base-url=http://localhost:11434
spring.ai.ollama.chat.options.model=llama3
spring.ai.ollama.chat.options.temperature=0.7

spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:postgresql://localhost:5433/medizin_db
spring.datasource.username=test
spring.datasource.password=123
spring.datasource.driver-class-name=org.postgresql.Driver

## Keycloak einstellungen

spring:
    security:
        oauth2:
            client:
                registration:
                    keycloak:
                        client-id: example-client
                        scope: openid
                        authorization-grant-type: authorization_code
                        redirect-uri: http://localhost:8080/login/oauth2/code/keycloak
                        provider: keycloak
                provider:
                    keycloak:
                        issuer-uri: http://localhost:8082/realms/example-realm
            resourceserver:
                jwt:
                    jwk-set-uri: http://localhost:8082/realms/example-realm/protocol/openid-connect/certs
```

Abbildung 1: Ausschnitt der Klasse 'application.properties' des Programms

Keycloak stellt zudem eine einheitliche Schnittstelle bereit, über die Benutzer sich mit verschiedenen Methoden anmelden können, darunter Passwort-basierte Authentifizierung, soziale Logins oder Single Sign-On. Dies erhöht die Benutzerfreundlichkeit und ermöglicht eine einfache Integration mit anderen Diensten. Die Möglichkeit der föderierten Identitätsverwaltung erlaubt es, bestehende Benutzerverzeichnisse wie LDAP oder Active Directory zu nutzen, wodurch eine redundante Verwaltung von Benutzerkonten vermieden wird. Keycloak ist eine Open-Source-Lösung, die kontinuierlich weiterentwickelt wird und eine hohe Anpassbarkeit durch eigene Erweiterungen und Plugins bietet.

Neben der Benutzerverwaltung bietet Keycloak robuste Sicherheitsfunktionen wie Zwei-Faktor-Authentifizierung und Sitzungsmanagement, um unbefugten Zugriff zu verhindern. Administratoren können über eine zentrale Oberfläche Benutzergruppen verwalten, Sicherheitsrichtlinien definieren und detaillierte Protokolle über Anmeldevorgänge einsehen. Dies erleichtert die Wartung und Überwachung des Systems erheblich. Durch die Integration von Keycloak in das bestehende System konnte die Benutzerverwaltung effizienter und sicherer gestaltet werden, während gleichzeitig eine modulare und skalierbare Architektur erhalten blieb[10].

3.2. HTML5

3.3. Docker

Docker ist eine leistungsfähige Plattform zur Container-Virtualisierung, die es ermöglicht, Anwendungen mit all ihren Abhängigkeiten in isolierten Containern auszuführen. Im Gegensatz zu herkömmlichen virtuellen Maschinen, die ein vollständiges Betriebssystem emulieren, teilen sich Docker-Container den Kernel des Host-Systems. Dadurch sind sie erheblich ressourcenschonender und effizienter in der Nutzung von Speicher und Rechenleistung. Dies macht Docker zu einer beliebten Wahl für Entwickler, die Anwendungen konsistent über verschiedene Umgebungen hinweg bereitstellen möchten [9].

Die Nutzung von Docker bietet zahlreiche Vorteile. Einer der wichtigsten ist die Portabilität: Einmal erstellte Container können unabhängig vom Betriebssystem auf jeder Plattform mit Docker-Unterstützung ausgeführt werden. Dies ermöglicht eine nahtlose Migration von Anwendungen zwischen Entwicklungs-, Test- und Produktionsumgebungen. Darüber hinaus sorgt Docker für eine hohe Skalierbarkeit, da Container innerhalb von Sekunden gestartet oder gestoppt werden können. Dies erleichtert nicht nur das Deployment, sondern auch das effiziente Management von Ressourcen, insbesondere in Cloud-Umgebungen. Die Isolation der Container verhindert zudem Konflikte zwischen verschiedenen Anwendungen, da jede Anwendung in ihrer eigenen Umgebung läuft und keine unerwünschten Abhängigkeiten entstehen [9].

Ein häufiges Sicherheitsproblem bei der Nutzung von Docker ist, dass viele Befehle standardmäßig mit Root-Rechten ausgeführt werden müssen. Um Docker sicherer zu nutzen, kann der aktuelle Benutzer der Docker-Gruppe hinzugefügt werden, wodurch

bestimmte Aktionen ohne Administratorrechte möglich sind. Nach dem Hinzufügen zur Gruppe ist es notwendig, sich einmal ab- und wieder anzumelden, damit die Änderungen wirksam werden. Dies verbessert die Sicherheit und vereinfacht die tägliche Arbeit mit Docker erheblich [3]].

Zusammenfassend bietet Docker eine flexible, effiziente und skalierbare Lösung für die Bereitstellung und Verwaltung von Anwendungen. Die Kombination aus Ressourcenschonung, Plattformunabhängigkeit und schneller Bereitstellung macht Docker zu einem unverzichtbaren Werkzeug in der modernen Softwareentwicklung. Durch zusätzliche Sicherheitsmaßnahmen, wie die Verwaltung ohne Root-Rechte, kann Docker zudem sicher und effizient in verschiedensten Umgebungen eingesetzt werden.

3.4. postgres

4. Datenbankdesign

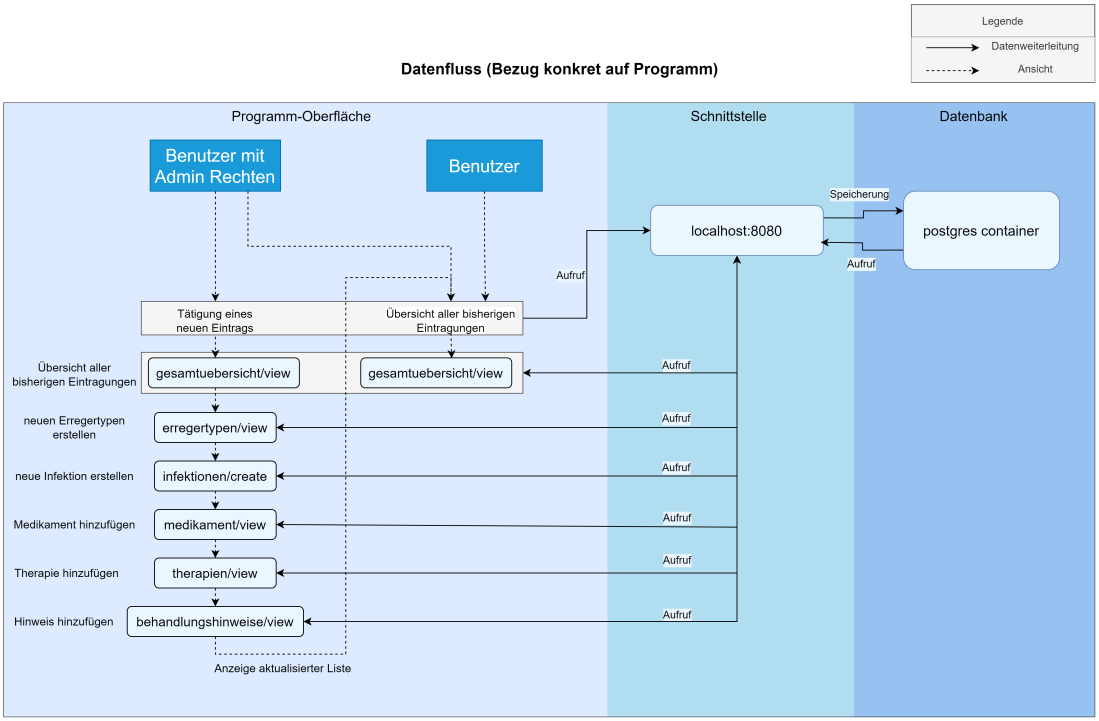

Abbildung 2: Darstellung des Datenflusses konkret innerhalb des Programms

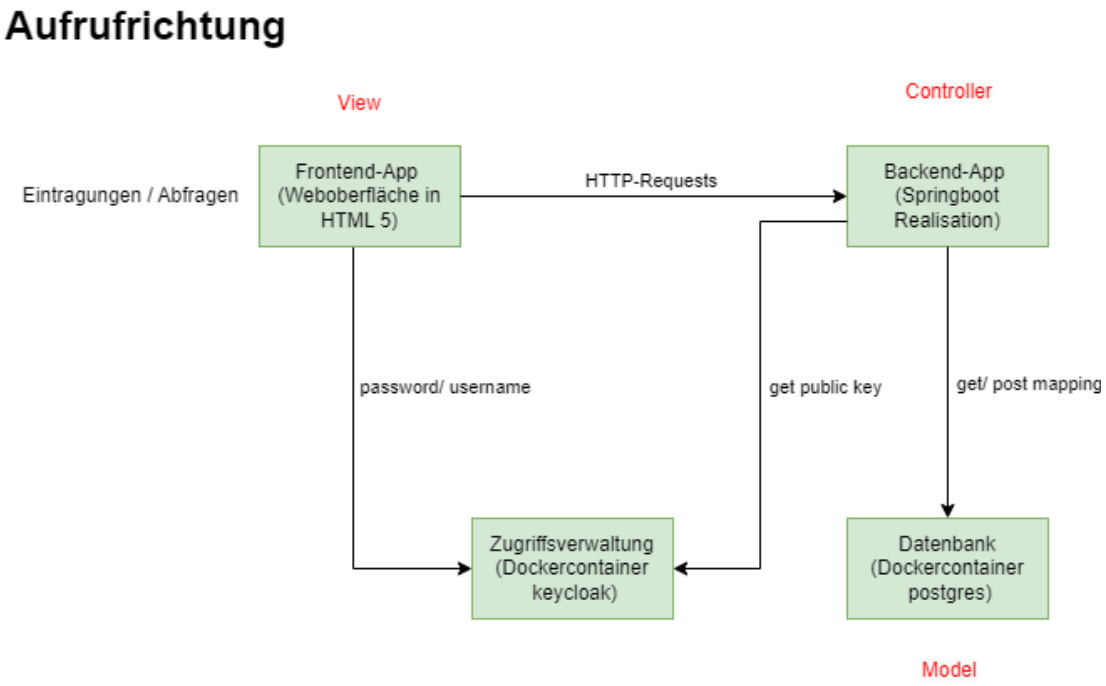

Abbildung 3: Darstellung der Aufrufrichtung der einzelnen Webseiten innerhalb der Datenbank

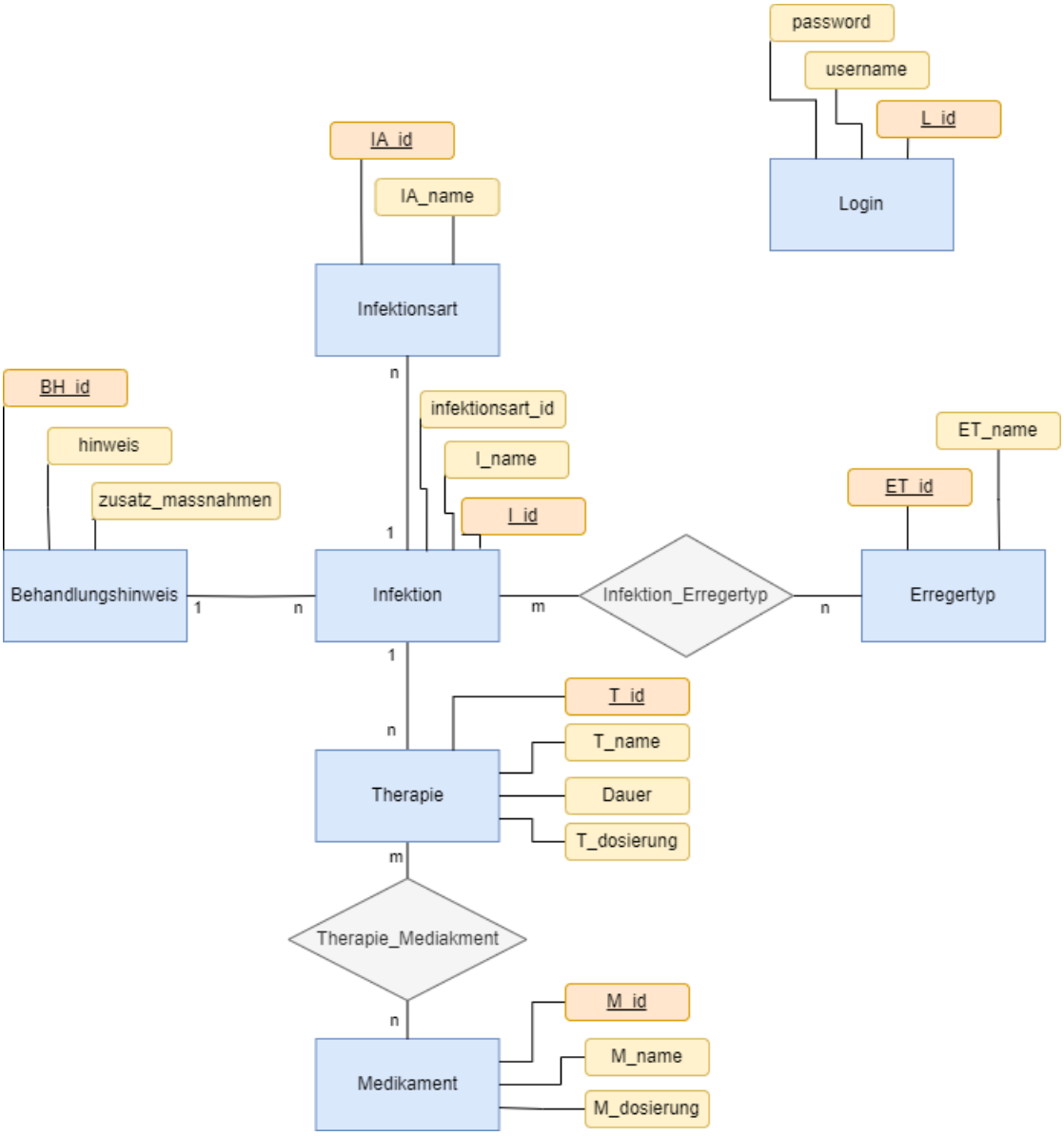

Abbildung 4: Darstellung der Entity-Relation

5. Zusammenfassung

Literaturverzeichnis

[1] Bundesministerium für Gesundheit. *Digitalisierung im Gesundheitswesen*. Bundesministerium für Gesundheit. 2024. URL: https://gesund.bund.de/digitalisierung-im-gesundheitswesen (besucht am 04. 03. 2024).

[2] Bundesministerium für Gesundheit. *Digitalisierung im Gesundheitswesen*. Bundesministerium für Gesundheit. 2024. URL: https://www.bundesgesundheitsministerium.de/themen/digitalisierung/digitalisierung-im-gesundheitswesen.html (besucht am 04. 03. 2024).

[3] Docker Documentation. *Post-installation steps for Linux*. Docker Docs. 2024. URL: https://docs.docker.com/engine/install/linux-postinstall/#manage-docker-as-a-non-rootuser (besucht am 18. 03. 2024).

[4] Fraunhofer IML. *Digitalisierung im Krankenhaus*. Fraunhofer IML. 2024. URL: https://www.iml.fraunhofer.de/de/abteilungen/b3/health_care_logistics/krankenhauslogistik/digitalisierung-krankenhaus.html (besucht am 04. 03. 2024).

[5] Keycloak Community. *Keycloak – Open Source Identity and Access Management*. Keycloak Community. 2024. URL: https://www.keycloak.org/ (besucht am 04. 03. 2024).

[6] LADR Labor. *Antibiotika – ja oder nein? Procalcitonin und CRP als Entscheidungshilfe*. LADR Labor. 2024. URL: https://www.ladr.de/news/detail/antibiotika-ja-oder-nein-procalcitonin-und-crp-als-entscheidungshilfe (besucht am 04. 03. 2024).

[7] LADR Labor. *Antiinfektivatherapie-Leitfaden (ABS)*. LADR Labor. 2024. URL: https://www.ladr.de/fileadmin/user_upload/116831_LADR_Antiinfektiva_web.pdf (besucht am 04. 03. 2024).

[8] Login Master. *Keycloak – Vorteile und Nachteile*. Login Master. 2024. URL: https://login-master.com/keycloak-vorteile-nachteile/ (besucht am 04. 03. 2024).

[9] Michael Schnepf. *Was sind Docker Container?* Systempilot. 2021. URL: https://systempilot.net/docker-container/ (besucht am 18. 03. 2024).

[10] Security Insider. *Keycloak – Open Source IAM-Verwaltung*. Security Insider. 2024. URL: https://www.security-insider.de/keycloak-open-source-iam-verwaltung-a-8d59caf4f95508e43263040e71a36c91/ (besucht am 04. 03. 2024).

A. Anhang

B. Anhang