

Dokumentation & Projekttagebuch

Innovation Lab 2
Jahr 2025

Projekt: **IDERHA**

Team: Nr. **23**

Inhalt

1. Allgemeine Informationen	3
2. Projekt-Kurzbeschreibung	5
3. Spezifikation der Lösung	6
3.1 Systemumfeld	
3.2 Features (Funktionale Anforderungen)	
3.3 Schnittstellen	
3.4 Sonstige wesentliche Lösungsmerkmale	
3.5 Github	
4. Aufwandschätzung	9
4.1 Delphi-Methode zur Aufwandsschätzung	
5. Auslieferung	10
5.1 Fertige Lösung inklusive Source-Code	
5.2 Systemarchitektur und Datenhaltung	
5.3 Scrum-Management via GitHub	
6. Unser Projekt-Tagebuch	11
3. Semester	
4. Semester	

1. Allgemeine Informationen

Projektname: IDERHA

Supervisor: Lukas Rohatsch

Innovation Lab 2, SS 2025

Projektteam:

Erdem Mehdi, if22b202@technikum-wien.at - Teamleiter, Ansprechperson

Zehinovic Aldin, if22b130@technikum-wien.at - Stellvertretung

Dervisefendic Armin, if23b040@technikum-wien.at

Lamthi Mateo, if23b263@technikum-wien.at

Puka Benjamin, if23b197@technikum-wien.at

*Bitte beachten Sie: Um mögliche Urheberrechtsprobleme zu vermeiden, haben wir uns entschieden, unsere Lösung **eHealth Insights** anstelle von **IDERHA** zu nennen.*

Management-Summary des Projektes

Ziel ist die Entwicklung eines Prototyps für einen eHealth Data Space, der als sichere und benutzerfreundliche Plattform dient. Der Prototyp soll die Verarbeitung, Analyse und den Zugang zu Gesundheitsdaten revolutionieren, indem er einen transparenten und effizienten Datenfluss zwischen verschiedenen Akteur:innen im Gesundheitswesen ermöglicht, ohne die Sicherheit und den Datenschutz zu beeinträchtigen

Rahmenbedingungen und Projektumfeld

Rahmenbedingungen

Für das **eHealth Insights**-Projekt werden folgende Rahmenbedingungen festgelegt: Es werden Java mit Spring Boot für das Backend und React.js mit Vite für das Frontend verwendet. Die Daten werden in einer PostgreSQL-Datenbank gespeichert. GitHub dient als Versionskontrollsystem, Figma für das Webdesign. Zur Sicherstellung der Sicherheit und des Datenschutzes werden Spring Security für Authentifizierung und Autorisierung sowie CORS für die sichere Kommunikation zwischen Frontend und Backend implementiert.

Qualitätseigenschaften:

- **Benutzerfreundlichkeit:** Intuitive Bedienung, einfache Navigation und schneller Zugriff auf Datensätze und Analysewerkzeuge.
- **einfache Wartbarkeit/Flexibilität:** unkomplizierte Wartung und Weiterentwicklung möglich
- **Sicherheit und Datenschutz:** Höchste Sicherheitsstandards zum Schutz sensibler Gesundheitsdaten durch Verschlüsselung und Zugriffskontrollen.
- **Performance und Effizienz:** Schnelle und effiziente Datenverarbeitung für zeitnahe Ergebnisse in der Gesundheitsdatenanalyse.
- **Skalierbarkeit:** Effiziente Handhabung wachsender Datenmengen und einfacher Anschluss neuer Datenquellen.

Am Ende des vierten Semesters soll eine erste funktionierende Datenbank eines Hospitals vorhanden sein, aus der wir die Daten auf unserem Frontend veranschaulichen können.

Semester-Roadmap:

Sprint 1 (Dienstag, 25 März) - Docker & Datenbank-Migration

- **Ziel:** Docker setup & UI verbessern

Sprint 2 (Dienstag, 4 April) - Patientenverwaltung & Rollenbasierter Zugriff

- **Ziel:** Patientenverwaltung implementieren & rollenbasierte Zugriffskontrolle (RBAC)

Sprint 3 (Dienstag, 29 April) - API-Sicherheit & Datenübertragung

- **Ziel:** Sichere Kommunikation & Krankenhaus-APIs

Sprint 4 (Dienstag, 13 Mai) - Graph-Datenbank & Visualisierung

- **Ziel:** Graph-Datenbank integrieren & erste Visualisierungen umsetzen

Sprint 5 (Dienstag, 27 Mai) - Skalierung & Performance

- **Ziel:** Mehrere Krankenhaus-Container & System optimieren

Sprint 6 (Dienstag, 10 Juni) - Tests, Fehlerbehebung & Abschluss

- **Ziel:** Finales Testing, Dokumentation & Abschlusspräsentation vorbereiten

Final Upload Documentation (Dienstag, 24 Juni)

- **Final Project Diary**
- **Source Code**
- **Presentation**
- **Video (life demo)**

2. Projekt-Kurzbeschreibung

Das Hauptziel von unserem Projekt besteht darin, einen innovativen eHealth Data Space zu entwickeln, der eine sichere, benutzerfreundliche Plattform zur Verarbeitung und Analyse von Gesundheitsdaten bietet. Diese Plattform soll den transparenten und effizienten Austausch von Daten zwischen verschiedenen Akteur:innen im Gesundheitswesen ermöglichen und gleichzeitig höchste Standards für Sicherheit und Datenschutz gewährleisten.

Basisfunktionalität (Mindestziel für das 4. Semester)

- Eine zentrale DB, dafür eigene Docker-Container die als Krankenhäuser mit Patient*innendaten (in DBs) und APIs angedockt werden können.
- Patient*innendaten werden in sicheren Datenbanken abgelegt und durch verschlüsselte Übertragungsprotokolle sowie rollenbasierte Zugriffskontrollen geschützt.
- Die gesammelten Daten werden mithilfe von Graphdatenbanken und Visualisierungstools in Graphen abgebildet, um eine effiziente und übersichtliche Analyse der Ergebnisse zu ermöglichen.
- Die Frontend-Oberfläche wird weiterentwickelt, um eine benutzerfreundliche und effiziente Interaktion mit komplexen Daten zu ermöglichen.

erweiterte Funktionalität (5.Semester):

- Voll funktionsfähige Webseite
- voll funktionsfähiger rollenbasierter Zugriff
- Optimierung der Performance
- Erweiterung der Sicherheitsmaßnahmen
- Testing & Qualitätssicherung
- Feedback sammeln
- Deployment & Dokumentation

Größte Herausforderungen:

Die zentralen Herausforderungen des Projekts bestehen darin, eine hohe Systemleistung und Sicherheit zu gewährleisten, die Interoperabilität zwischen verschiedenen Gesundheitsdienstleistern zu fördern und strenge Datenschutzvorgaben einzuhalten. Darüber hinaus ist eine intuitive Benutzeroberfläche für das Nutzer:innenportal entscheidend, um eine einfache Verwaltung und Interaktion mit Gesundheitsdaten zu ermöglichen.

Mehrwert für die Anwender:innen:

eHealth Insights steigert die Produktivität, indem es die Analyse und Interpretation von Gesundheitsdaten erleichtert, sodass Nutzer:innen ohne tiefgehende Kenntnisse in Datenvisualisierung schnell verwertbare Erkenntnisse gewinnen können. Zudem ermöglicht die Plattform den weltweiten Zugriff auf große Datenmengen für verschiedene Benutzergruppen und bietet die Möglichkeit, Daten in gängigen Formaten wie PDF oder DOC zu exportieren.

Projektumfang:

Das Hauptziel von **eHealth Insights** ist die Entwicklung eines innovativen eHealth Data Space, der eine sichere und benutzerfreundliche Plattform zur Verarbeitung, Analyse und Visualisierung von Gesundheitsdaten bereitstellt.

3. Spezifikation der Lösung

3.1 Systemumfeld

Die Lösung **eHealth Insights** ist ein eHealth Data Space zur sicheren Verarbeitung, Analyse und Darstellung von Gesundheitsdaten. Die Hauptkomponenten des Systems sind:

- **Frontend:** Eine webbasierte Benutzeroberfläche für unterschiedliche Nutzergruppen (z. B. Ärzt:innen, Administrator:innen).
- **Backend:** Eine Spring Boot API zur Datenverarbeitung, Authentifizierung und Kommunikation mit der Datenbank.
- **Datenbank:** Eine PostgreSQL-Datenbank, die über Docker-Container verwaltet wird und Patientendaten sowie Berechtigungen speichert.
- **Schnittstellen:** APIs zur Anbindung mehrerer „Krankenhaus“-Datenbanken und zur Datenauswertung mit Visualisierungstools.

Systemgrenzen:

- Datenverarbeitung und Speicherung erfolgt ausschließlich innerhalb der Plattform.
- Keine direkte Interaktion zwischen Nutzer:innen und der Datenbank – alle Operationen laufen über das Backend.
- Rollenbasierte Zugriffskontrollen zur Einschränkung der Datenverfügbarkeit.

3.2 Features (Funktionale Anforderungen)

Die Kernfunktionalitäten von **eHealth Insights** umfassen:

- Benutzerverwaltung

- Registrierung/Login (mit Datenbank-Anbindung)
- Rollenbasierte Zugriffskontrolle (Admin, Arzt, Forscher etc.)
- Sitzungsverwaltung und sichere Authentifizierung (Spring Security)

- Datenmanagement

- Speicherung von Patientendaten in sicheren Datenbanken
- Verschlüsselte Übertragung von Daten zwischen Krankenhäusern und der zentralen Datenbank
- Unterstützung für mehrere Krankenhäuser über separate Docker-Container

- Analyse & Visualisierung

- Darstellung von Patientendaten in einer übersichtlichen UI
- Integration von Graph-Datenbanken zur Analyse von Beziehungen in den Daten
- Export-Funktion für Berichte (PDF, DOC)

- API & Schnittstellen

- REST API zur Kommunikation mit Krankenhäusern
- Externe Schnittstellen für Datenanalyse-Tools

- Benutzeroberfläche (Frontend)

- Moderne UI mit React.js + Vite
- Dynamische Datenvisualisierung für medizinische Analysen
- Responsive Design für verschiedene Endgeräte

3.3 Schnittstellen

Die Hauptschnittstellen sind:

- **Frontend** → **Backend**: REST API für Datenabfragen und Benutzerinteraktionen.
- **Backend** → **Datenbank**: Kommunikation mit der PostgreSQL-Datenbank über Spring Boot.
- **Datenbank** → **Visualisierungstools**: Zugriff auf Graph-Datenbanken zur Analyse.
- **Krankenhäuser** → **Backend**: Sichere Datenübertragung über externe APIs.

3.4 Sonstige wesentliche Lösungsmerkmale

- Automatisierte Tests für API und Datenbank.
- Logging und Monitoring für Fehleranalyse.
- Einhaltung von Datenschutzstandards (DSGVO).

3.5 Github

- <https://github.com/mide553/IDERHA>

4. Aufwandschätzung

4.1 Delphi-Methode zur Aufwandsschätzung

Aufgabe	Teammitglied 1 (h)	Teammitglied 2 (h)	Teammitglied 3 (h)	Teammitglied 4 (h)	Teammitglied 5 (h)	Durchschnitt (h)	Erläuterung
Frontend-Entwicklung (UI-Verbesserungen)	10	15	10	13	15	12,6	Verbesserung der Benutzeroberfläche basierend auf Nutzerfeedback. Unterschiedliche Schätzungen aufgrund der Komplexität der Designänderungen.
Datenbank Setup & Docker Container	9	6	10	8	12	9	Einrichtung der Docker-Container und Datenbank, umfasst auch die Konfiguration der Umgebung. Abweichungen aufgrund unterschiedlicher Erfahrungslevel.
API-Integration	22	25	20	20	30	23,4	Integration von Frontend und Backend über APIs, unterschiedliche Einschätzungen hinsichtlich der API-Komplexität.
Rollenbasierte Zugriffskontrolle	15	10	10	14	12	12,2	Implementierung der rollenbasierten Zugriffskontrolle. Unterschiede aufgrund verschiedener Erfahrungen mit Sicherheitsprotokollen.
Frontend-Testing & Bugfixing	7	7	5	6	8	6,6	Tests und Fehlerbehebung im Frontend. Ähnliche Schätzungen aufgrund vergleichbarer Erfahrung mit Frontend-Tests.
Backend-Entwicklung (API Endpunkte)	25	23	20	22	20	22	Entwicklung von Backend-API-Endpunkten. Unterschiede basierend auf der Komplexität der Logik und Erfahrung in Backend-Technologien.
Sicherheitsimplementierung (Verschlüsselung)	18	15	14	12	15	14,8	Implementierung von Sicherheitsprotokollen und Verschlüsselung. Abweichungen aufgrund unterschiedlicher Vertrautheit mit Verschlüsselungsstandards.
Team Meetings	12	12	13	14	12	12,6	Gesamtstundenzahl für Teambesprechungen im Semester
Summe aller Stunden						91,6	

5. Auslieferung

5.1. Fertige Lösung inklusive Source-Code

Unsere Lösung umfasst die folgenden Hauptkomponenten:

- **Frontend:** Eine benutzerfreundliche Web-Oberfläche, die den Zugriff und die Visualisierung von Gesundheitsdaten ermöglicht.
- **Backend:** Ein RESTful API-Service, der Daten verarbeitet und mit der zentralen Datenbank sowie externen Services kommuniziert.
- **Datenbank:** Eine zentrale PostgreSQL-Datenbank, die alle gesammelten Daten speichert und eine effiziente Verarbeitung ermöglicht.
- **Docker-Container:** Die Lösung wird in Docker-Containern bereitgestellt, die es ermöglichen, die verschiedenen Komponenten wie Frontend, Backend und Datenbank unabhängig voneinander zu betreiben.
- **API-Integration:** Mehrere APIs zur Anbindung an andere Systeme und zum Abrufen von Gesundheitsdaten.

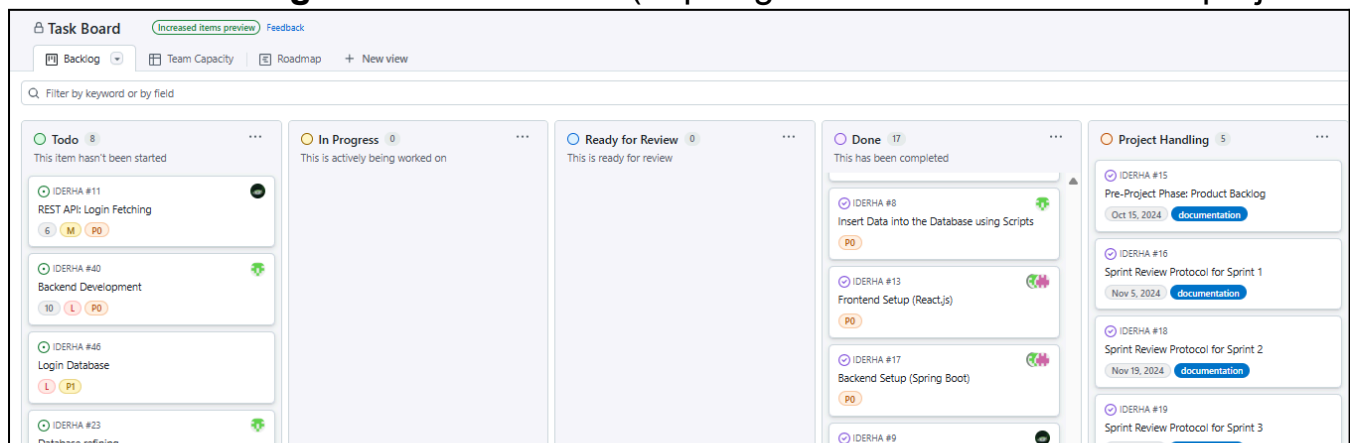
Der vollständige Source-Code der Lösung wird im GitHub-Repository des Projekts (<https://github.com/mide553/IDERHA>) abgelegt und ist zur Weiterentwicklung und Wartung zugänglich.

5.2. Systemarchitektur und Datenhaltung

Die Systemarchitektur basiert auf einer Microservice-Architektur, die es ermöglicht, alle Komponenten der Lösung unabhängig zu entwickeln und zu betreiben. Die Architektur sieht wie folgt aus:

- **Frontend:** Ein React-basiertes Web-Frontend, das mit dem Backend über HTTP-Requests kommuniziert.
- **Backend:** Ein Spring Boot-basierter Service, der die API-Endpunkte bereitstellt und mit der PostgreSQL-Datenbank interagiert.
- **Datenbank:** Eine PostgreSQL-Datenbank speichert alle Daten, mit einer strukturierten Architektur, die auf der OMOP-CDM (Common Data Model) basiert.
- **Docker:** Alle Komponenten werden in Docker-Containern betrieben, um eine flexible Bereitstellung und Skalierung zu ermöglichen.

5.3 Scrum-Management via GitHub (<https://github.com/users/mide553/projects/1>)



6. Unser Projekt-Tagebuch

3. Semester

24.09.2024 14:30 erstes Meeting mit Prof. Lukas Rohatsch über Zoom: Besprechung der Projektanforderungen, Fragerunde zur Klärung von Unklarheiten.

02.10.2024 17:30 Teammeeting über Discord: Festlegung der Technologie (Java, Spring Boot, React, Postgresql). Einigung der Personen, welche das Projekt nach außen vertritt (Mehdi Erdem) und Stellvertreter (Aldin Zehinovic). Besprechung und Beginn der Überführung der Projektspezifikationen in die Dokumentation für Punkt 1 & 2.

10.10.2024 15:00 Teammeeting zur Fertigstellung von Punkt 1 & Punkt 2 der Dokumentation für die Abgabe. Des Weiteren wurde eine grobe Aufgabenverteilung für den ersten Sprint festgelegt, um den Arbeitsfortschritt zu organisieren. Eine E-Mail wurde an Herrn Rohatsch gesendet, um die Terminabstimmung für weitere Treffen zu ermöglichen.

18.10.2024 17:30 Das Meeting mit Herrn Prof. Rohatsch fand statt, um die ausgearbeiteten Punkte 1 & 2 des Project Diaries zu besprechen. (Projektkurzbeschreibung, Spezifikation der Lösung...). Weiters haben wir das Datenbankmodell OMOP besprochen und die hilfreichen Links bezüglich des Projekts bekommen.

25.11.2024 Besprechung der UI Oberfläche und des Architecture Design des Central Nodes (Aldin, Armin, Mehdi). Datenbank aufsetzen (pgadmin).

05.12.2024 11:00 Das Meeting mit Herrn Prof. Rohatsch fand statt, es wurden die nächsten Schritte besprochen.

12.12.2024 14:00 Teammeeting über Discord, hier haben wir die Aufgaben für Frontend und Backend geteilt.

10.01.2025 16:00 Uhr, war unser letztes Meeting mit Herrn Prof. Rohatsch, wir haben besprochen, mit was wir in der 4. Semester weiter machen werden: - Docker images für die Datenbanken und einen central node - Entwurf von Daten in Graphen

18.01.2025 17:00 Wir haben uns getroffen und an der der Dokumentation gearbeitet (Aldin, Armin, Mehdi)

26.01.2025 16:00 Wir haben uns getroffen und an der der Dokumentation gearbeitet und das Video gedreht (Aldin, Armin, Mehdi)

28.01.2025 21:00 letzte Kontrolle der Dokumentation und Präsentation bevor der Abgabe (Aldin, Armin, Mehdi)

4. Semester

09.03.2025 Projektbesprechung

10.03.2025 Dokumentation & Projekttagbuch und Aufwandschätzung