12.06.2026 - Bachelorarbeit Frühjahrssemester 2026

Isokinetische Kniemessungen

Datenerfassung und Analyse

Studenten:	**Tobias Keel** tobias.keel@ost.ch	**Simon Weibel** simon.weibel@ost.ch
Betreuer:	**Prof. Dr. Remo Frey** remo.frey@ost.ch	
Projekt-Partner:	**Timothy Grob** Rosenklinik Sportzentrum TGrob@swissmedical.net	**Janika Kessler** Rosenklinik Sportzentrum JKessler@swissmedical.net
Experte:	**Dr. Edward Ho** Google Schweiz edward.y.ho@gmail.com	
Gegenleser:	**Thomas Bocek** thomas.bocek@ost.ch	

Abstract

Isokinetische Kraftmessungen dienen der Bestimmung der maximalen Muskelkraft bei Flexion und Extension unter konstanter Winkelgeschwindigkeit. Das Sportzentrum der Rosenklinik in Rapperswil setzt diese Messungen zur Beurteilung der Oberschenkelmuskulatur im Rahmen von Rehabilitationsprogrammen nach Knieverletzungen sowie vor und nach Prothesenoperationen ein und nutzt sie zusätzlich für allgemeine Leistungsdiagnostiken. Dabei kommt der Prima Plus der italienischen Firma Easytech zum Einsatz.
Im Rahmen von Prothesenoperationen plant das Sportzentrum die Durchführung einer Studie zur Untersuchung des Einflusses der Ausprägung der Kniegelenksmuskulatur auf den Heilungsverlauf nach einer Operation. Die hierfür eingesetzte Gerätesoftware „Isocine3“ des Prima Plus erfüllt die Anforderungen der geplanten Datenauswertung jedoch nicht. Aufgrund ihres veralteten Softwaredesigns weist sie funktionale Einschränkungen im täglichen Einsatz auf, insbesondere im Bereich der Datenvisualisierung und des Exports. Pro Patient können lediglich die Graphen von maximal zwei Messungen direkt eingesehen werden, während der Export ausschliesslich über ein externes Hilfsprogramm erfolgt; eine direkte Speicherung als PDF ist nicht möglich, stattdessen steht lediglich eine Druckfunktion zur Verfügung.

Ziel dieser Arbeit ist die Entwicklung einer Softwarelösung zur Unterstützung des Sportzentrums bei der täglichen Verarbeitung von rund 21 Messungen pro Tag. Die Anwendung soll den Export von Messdaten anhand frei wählbarer Filter als CSV-, Excel- oder PDF-Datei ermöglichen und zusätzlich Analysefunktionen für einzelne sowie mehrere Patienten bereitstellen.
Der Fokus liegt dabei auf einer hohen Benutzerfreundlichkeit und einfachen Bedienbarkeit, um einen reibungslosen Arbeitsablauf zu ermöglichen unter Einhaltung der geltenden Datenschutzbestimmungen zu gewährleisten. Während der Umfang der Arbeit auf eine Messart, ein spezifisches Gerät sowie einen einzelnen Arbeitsplatz beschränkt ist, soll die Lösung eine spätere Erweiterung innerhalb des Kliniknetzwerks „Swiss Medical Network“, dessen Mitglied das Sportzentrum ist, ermöglichen. Die zentralen Herausforderungen des Projekts liegen in der Analyse und Decodierung proprietärer Datenstrukturen sowie im Design einer intuitiven und effizienten Benutzeroberfläche.

Im Rahmen dieser Arbeit wurde „IsoTrack“ entwickelt, eine .NET-basierte Softwarelösung, die eng in Microsoft-Technologien integriert ist und Mess- sowie Patientendaten automatisch mit einem lokal gehosteten Server synchronisiert.
Der Client für die Datensynchronisierung und der Server sind als getrennte Komponenten umgesetzt und werden über ein Installationspaket als Hintergrunddienste bereitgestellt. Der Server verwaltet eine SQLite-Datenbank über das Entity Framework und stellt zusätzlich eine Blazor-Weboberfläche zur Verfügung. IsoTrack ermöglicht den Export von Messungen anhand frei definierbarer Kriterien, die Darstellung von zeitlichen Verläufen einzelner Patienten sowie deren medizinische Einordnung anhand von Referenzwerten. Darüber hinaus werden patientenübergreifende Analysen mit umfangreichen Filtermöglichkeiten unterstützt.

Inhaltsverzeichnis

1 Einführung . . . 1
1.1 Projektpartner . . . 1
1.2 Ausgangslage . . . 1
1.3 Ziel . . . 4
1.4 Motivation . . . 4
1.5 Rahmenbedingungen . . . 6
1.6 Verwendete Technologien . . . 6
1.6.1 .NET . . . 6
1.6.2 SQLite . . . 8
1.6.3 OLE DB . . . 8
1.6.4 WiX Toolset . . . 8
1.7 Stand der Technologie . . . 8
1.7.1 Medizinische Messgeräte . . . 9
1.7.2 Smart Hospital . . . 10
1.7.3 Smartwatches . . . 10
2 Produkt Dokumentation . . . 12
2.1 Anforderungsanalyse . . . 12
2.1.1 Funktionale Anforderungen . . . 12
2.1.2 Epics . . . 12
2.1.3 User Stories . . . 12
2.1.4 Nicht-Funktionale Anforderungen . . . 13
2.2 Domänenanalyse . . . 16
2.2.1 Domänenmodell . . . 16
2.3 Architektur . . . 20
2.3.1 Softwarearchitektur . . . 20
2.3.2 Isocine3 . . . 22
2.3.3 Client . . . 28
2.3.4 Server . . . 30
2.3.5 Architectural Decision Records . . . 33
2.4 Implementation . . . 36
2.4.1 Synchronisierungsprozess . . . 36
2.4.2 Benutzeroberfläche . . . 37
2.5 Qualitätssicherung . . . 50
2.5.1 Arbeitsumgebung . . . 50
2.5.2 Tools . . . 52
2.5.3 Testkonzept . . . 53
3 Projekt Dokumentation . . . 57
3.1 Projektplan . . . 57
3.1.1 Ressourcen . . . 57
3.1.2 Prozesse . . . 57
3.1.3 Meilensteine . . . 58

3.1.4 Proof of Concept . . . 60
3.1.5 Fristen und Abgabetermine . . . 60
3.1.6 Teamrollen . . . 60
3.1.7 Meetings . . . 61
3.1.8 Planungstools . . . 61
3.2 Risikomanagement . . . 63
3.2.1 Risikomatrix Sprint 1 . . . 63
3.2.2 Risikomatrix Sprint 2 . . . 63
3.2.3 Risikomatrix Sprint 3 . . . 64
3.2.4 Risikomatrix Sprint 5 . . . 64
3.2.5 Risikomatrix Sprint 6 . . . 65
3.2.6 Risiken . . . 65
3.2.7 Probleme . . . 67
3.3 Ergebnisse und Erkenntnisse . . . 71
3.3.1 Aussicht . . . 71
3.4 Zeitmanagement . . . 73

A Anhang . . . 74
A.1 Modulbeschreibung Bachelorarbeit Informatik . . . 74
A.1.1 Hauptpunkte . . . 74
A.1.2 Lernziele . . . 74
A.1.3 Beschreibung . . . 75
A.2 Benutzeranleitung . . . 76
A.2.1 Technische Spezifikationen . . . 76
A.2.2 Installation . . . 76
A.2.3 Bedienung . . . 78
A.2.4 Fehlerbehandlung . . . 86

B Glossar . . . 89

C Tabellenverzeichnis . . . 92

D Abbildungsverzeichnis . . . 93

E Hilfsmittelverzeichnis . . . 95

F Literatur- und Quellenverzeichnis . . . 96

1 Einführung

Dieser Abschnitt bietet einen Überblick über die Rahmenbedingungen der Bachelorarbeit, die eingesetzten Technologien sowie eine Einführung in die zugrunde liegende Aufgabenstellung und Thematik.

1.1 Projektpartner

Das Sportzentrum Rosenklinik ist Teil der Rosenklinik, die 1995 in Rapperswil-Jona gegründet wurde. Es bietet ein umfassendes Angebot in den Bereichen Prävention, Leistungsdiagnostik und Rehabilitation. Für diese Programme sind unter anderem regelmässige Tests der Kniemuskelkraft erforderlich, insbesondere für Patienten vor und nach Knieprothesenoperationen.

Das Sportzentrum befindet sich derzeit in der Anfangsphase zur Aufstellung einer These, nach der die gezielte Kräftigung der Kniemuskulatur einen positiven Einfluss auf den Heilungsprozess nach einer Knieprothesenoperation haben könnte. Zur Validierung dieser These werden zahlreiche Messungen benötigt, die im aktuellen Prozess nicht realisierbar sind.

Die Rosenklinik stellte daher der OST die Anfrage für dieses Projekt, mit dem Ziel, die Datenerfassung so weit zu automatisieren, dass die erforderliche Datenmenge für die wissenschaftliche Untersuchung effizient erhoben werden kann.

1.2 Ausgangslage

Die italienische Firma Easytech bietet verschiedene medizinische Messgeräte an. Im Sportzentrum Rosenklinik kommt das Gerät Prima Plus zum Einsatz. Dieses ermöglicht die Beurteilung der Knie-, Fuss- und Schultermuskulatur mittels isokinetischer, isometrischer, isotonischer und hydrodynamischer Verfahren sowie durch kontinuierliche passive Bewegung.

Das Gerät wird über einen USB-Anschluss betrieben und durch die mitgelieferte Software „Isocine3“ gesteuert.

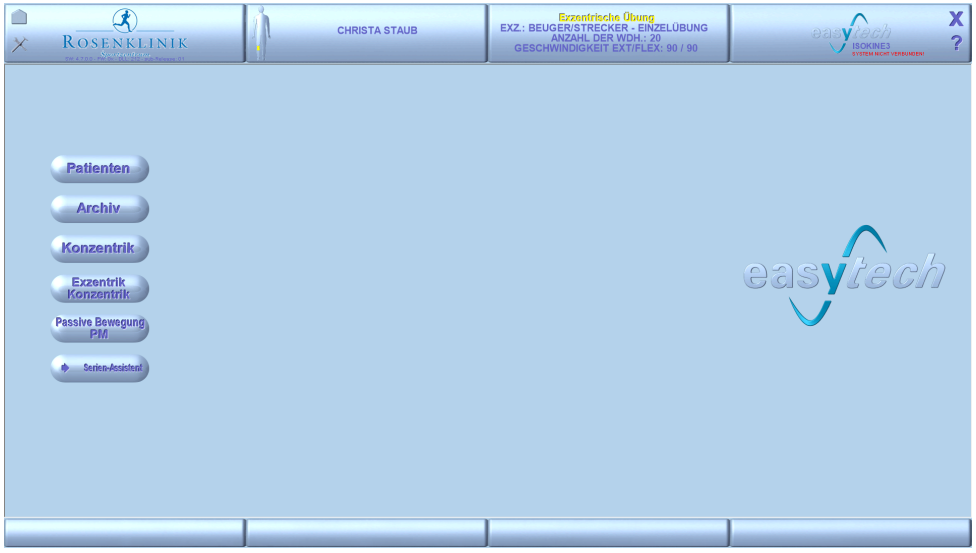


Abbildung 1: Hauptmenü der Isocine3-Software.

Der Schwerpunkt des Sportzentrums liegt auf isokinetischen Kraftmessungen des Knies. Diese werden insbesondere im Rahmen der Vorbereitung und Verlaufskontrolle vor und nach Operationen mit Knietotalprothesen (KTP) eingesetzt, finden jedoch auch Anwendung bei anderen Implantationen sowie bei sportmedizinischen Check-ups.

Bei isokinetischen Messungen wird die Gelenkkraft primär anhand des Drehmoments in Newtonmeter gemessen, da dieses neben der Muskelkraft auch den Hebelarm des Gelenks berücksichtigt und somit die tatsächliche Gelenkbelastung besser beschreibt. Dabei wird die Kraft des Quadrizeps beim Strecken (Extension) sowie der Kniebeuger beim Beugen (Flexion) jeweils unter konzentrischer Muskelarbeit gemessen.

Für die Beurteilung werden zusätzlich weitere Messgrössen herangezogen. Dazu gehören die Range of Motion (ROM), welche den Bewegungsradius eines Gelenks beschreibt, der Winkel des maximalen Drehmoments sowie der Ausdauerindex. Letzterer vergleicht die während der letzten Wiederholung verrichtete Arbeit in Joule prozentual mit jener der ersten Wiederholung und erlaubt Rückschlüsse auf die muskuläre Ermüdung.

Abbildung 2 zeigt eine Übersicht über zwei Messungen eines Patienten, jeweils für das linke und rechte Knie. Dabei sind das aktuelle Drehmoment als blaue Linie sowie der Gelenkwinkel als rot-orange Linie dargestellt. Die blauen Boxen entsprechen dem Spitzendrehmoment der jeweiligen Bewegung (Extension/ Flexion), wobei ihre Breite der Dauer der jeweiligen Bewegungsphase entspricht. In den Hilfsboxen sind die entsprechenden Metriken dargestellt.

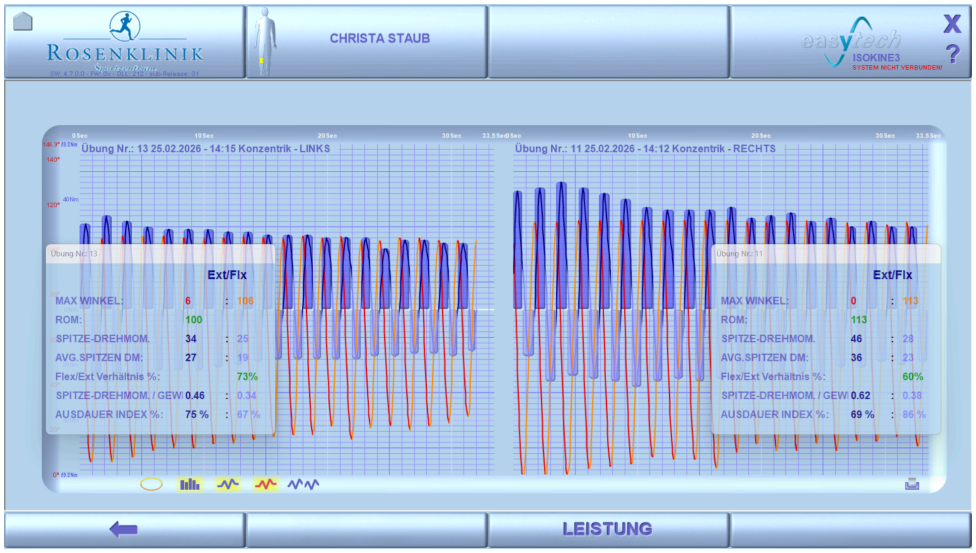


Abbildung 2: Messgraphen zweier isokinetischer Messungen in Isocine3 mit Metriken.

Bei der Beratung eines Patienten verwendet das Sportzentrum Rosenklinik primär diesen Graphen, aber auch eine überlappende Variante (siehe Abbildung 3).

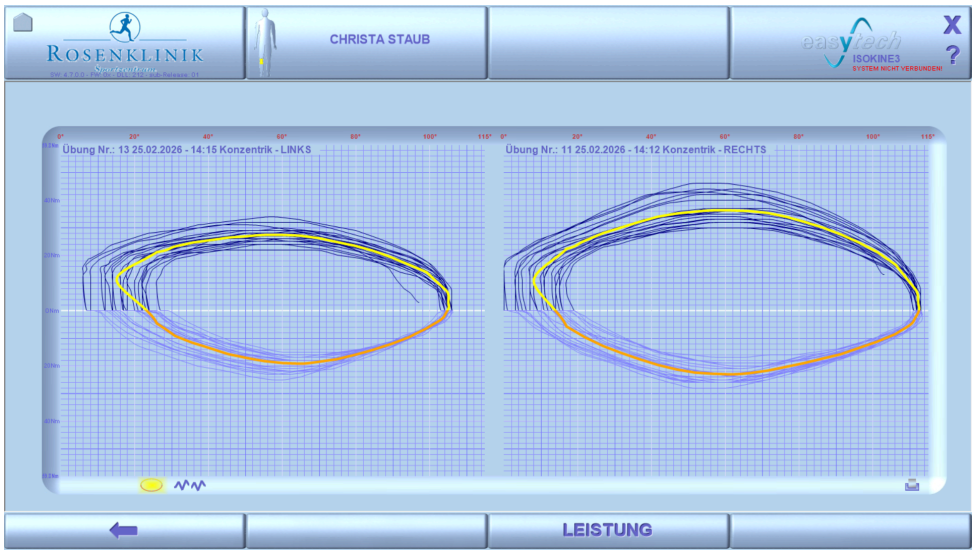


Abbildung 3: Messgraphen zweier isokinetischer Messungen in Isocine3 mit überlappenden Linien.

Der aktuelle Prozess des Exports der Messdaten im Anschluss an eine Messung ist ineffizient und stark manuell geprägt. PDF-Dokumente können jeweils nur bis zu zwei Messungen zusammenfassen, müssen einzeln erstellt werden und lassen sich ausschliesslich ausdrucken und nicht digital speichern.

Easytech stellt zusätzlich die Export-Software „IsoExportDB“ zur Verfügung (siehe Abbildung 4), welche den Export aller Messungen eines bestimmten Messtyps in eine Excel-Datei ermöglicht (siehe Abbildung 5).

Abbildung 4: IsoExportDB Zusatzsoftware für den Excel-Export.

-1-	-2-	-3-	-4-	-5-	-6-	-7-	-8-	-9-	-10-	-11-	-12-	-13-	-14-	-15-	-16-	-17-	-18-	-19-	-20-
Exercise n	Date/Time	Type	Series	Side	Velocity E	# Rep. Set	ROM limit	ROM max	Av.ge Pea	Peak Torc	Angle @P	Max Powe	Angle @M	Max Work	Av.ge Wo	Tot. Work	Endur. Ind	Time Eff./T	Flex/Ext Ratio %:
1	30.04.2025	CONCENTRIC		RIGHT	60/60	5/5	*/*	5/111	125/62	140/67	64/29	151/78	66/34	136/58	124/52	623/264	*/*	17/17	47
2	30.04.2025	CONCENTRIC		LEFT	60/60	5/5	*/*	3/100	113/49	117/52	68/26	128/58	68/25	118/35	107/33	539/168	*/*	17/17	44
3	30.04.2025	CONCENTRIC		RIGHT	180/180	20/20	*/*	-6/112	64/30	75/34	55/40	253/116	65/47	108/39	90/32	1800/653	93/100	32/32	45
4	30.04.2025	CONCENTRIC		LEFT	180/180	20/20	*/*	-2/107	66/30	79/35	67/34	264/122	60/34	105/36	87/29	1759/584	87/106	31/31	44

Abbildung 5: Excel-Export aller Messungen eines Patienten.

Bei durchschnittlich 21 Messungen pro Tag verursacht dieser Prozess einen erheblichen Zeitaufwand und erhöht zudem die Fehleranfälligkeit, da das anschliessende Zusammenführen der Daten manuell erfolgt.

1.3 Ziel

Das Ziel dieses Projekts ist die Entwicklung eines funktionsfähigen Produkts im Rahmen einer Bachelorarbeit für das Sportzentrum Rosenklinik, welches die Erfassung von Kniekraftmessungen erheblich beschleunigt und dadurch den Projektpartnern die Auswertung und Überprüfung ihrer Hypothesen ermöglicht.

Ein weiteres Ziel besteht darin, den Physiotherapeuten patientenübergreifende Analysen und Vergleiche zu ermöglichen.

Da die Rosenklinik die Durchführung einer Studienerhebung erst vor kurzem in Betracht gezogen hat, wurde der Datenqualität bislang nur begrenzte Aufmerksamkeit geschenkt. Für belastbare Analysen ist jedoch eine konsistente und möglichst vollständige Datenbasis erforderlich. Die Software soll deshalb Werkzeuge zur Identifikation und Korrektur von Datenmängeln bereitstellen und damit eine erste Datenbereinigung unterstützen.

1.4 Motivation

Die Vereinfachung und Beschleunigung der Datenverwaltung, insbesondere des Datenexports, sollen die Durchführung der beschriebenen Studie ermöglichen.

Nach den Erfahrungen der am Projekt beteiligten Physiotherapeuten der Rosenklinik besteht die Annahme, dass die Leistungsfähigkeit der Kniemuskulatur vor einer Operation sowohl mit dem Rehabilitationserfolg als auch mit der Rehabilitationsdauer korreliert. Zur Überprüfung dieser Hypothese ist jedoch eine ausreichend grosse und qualitativ bereinigte Datenbasis erforderlich.

Die Analyse isokinetischer Messverläufe erlaubt eine zeitliche Betrachtung der Muskelkraftentwicklung vor und nach operativen Eingriffen. Als primäre Kenngrösse dient dabei das maximale Drehmoment, welches in der klinischen Praxis häufig zur Beurteilung der Muskelkraft herangezogen wird. Ergänzend können weitere Messgrössen wie der Bewegungsumfang (Range of Motion) oder der Winkel beim maximalen Drehmoment berücksichtigt werden.

Im Rahmen einer Knietotalprothese (KTPT) erfolgen Messungen typischerweise etwa 5 Wochen präoperativ sowie 5, 12 und 24 Wochen postoperativ.

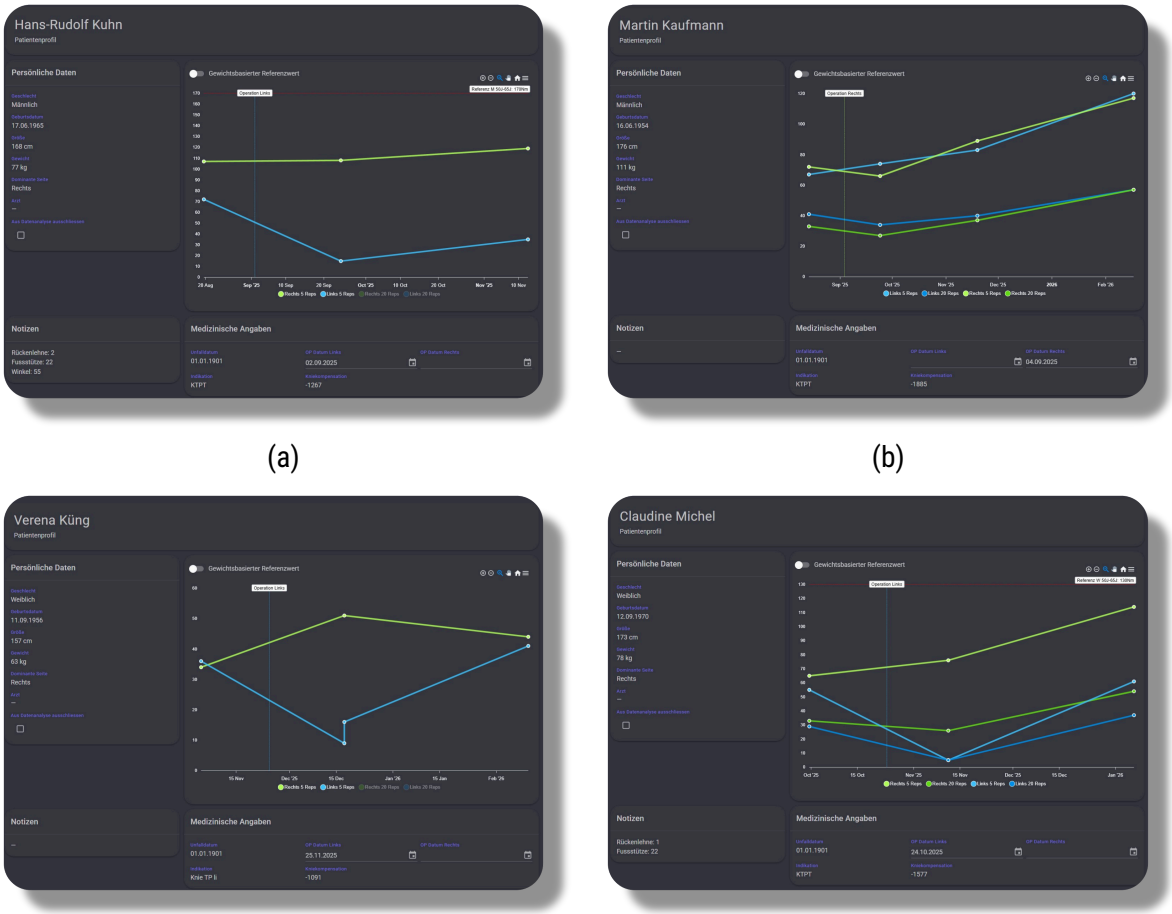


Abbildung 6: Vier beispielhafte Verläufe anonymisierter Patienten mit Knietotalprothese (KTPT), dargestellt in IsoTrack.

Abbildung 6 zeigt exemplarisch die zugrunde liegende Hypothese auf.

Die Graphen (a) und (b) stammen von männlichen Patienten, (c) und (d) von weiblichen Patienten. Alle Patienten sind zwischen 55 und 72 Jahre alt und wurden aufgrund einer KTPT operiert, wobei in drei Fällen die nichtdominante Seite betroffen ist. Graph (b) bildet als einziger Fall den vollständigen 24-Wochen-Verlauf ab, während die übrigen Patienten sich noch in früheren Nachsorgephasen befinden.

Alle dargestellten Verläufe zeigen einen deutlichen Kraftabfall der operierten Seite nach dem Eingriff. Im weiteren Verlauf ist in allen Fällen eine zumindest teilweise Wiederherstellung der Muskelkraft erkennbar. Zudem zeigt sich in zwei der vier Beispiele eine deutliche Kraftzunahme der nicht operierten Seite, was auf Anpassungs- und Kompensationseffekte während der Rehabilitation hindeuten könnte.

Für weiterführende und statistisch belastbare Aussagen sind jedoch deutlich umfangreichere Datensätze sowie eine konsistente und bereinigte Datenbasis erforderlich, deren Bereitstellung und Analyse durch dieses Projekt unterstützt werden soll.

Die Durchführung einer solchen Studie könnte dazu beitragen, Krankenkassen von präoperativen physiotherapeutischen Massnahmen zu überzeugen. Bei über 20'000 Knieprothesenoperationen pro Jahr in der Schweiz[1] könnten bereits moderate Verbesserungen der Rehabilitation einen spürbaren gesundheitli-

[1] Regula Heller, „Hüft- und Knieimplantationen: SIRIS Report 2024 bestätigt Trend zu sinkenden Revisionsraten“, preprint, ANQ, 10. Dezember 2024, https://www.anq.ch/de/medienmitteilungen/hueft-und-knieimplantationen-siris-report-2024-bestaetigt-trend-zu-sinkenden-revisionsraten/.

chen und ökonomischen Nutzen bewirken. Dadurch könnten Therapieeinheiten vor der Operation gezielt eingesetzt und potenziell durch eine reduzierte Nachbehandlung kompensiert werden. Dies hätte sowohl gesundheitliche Vorteile für die Patienten als auch mögliche ökonomische Effekte.

Unabhängig davon führt die Optimierung der Datenverwaltung zu einer Verkürzung und Vereinfachung der Arbeitsprozesse der Physiotherapeuten.

1.5 Rahmenbedingungen

Gemäss Modulbeschreibung[2] soll die Arbeit den Nachweis der Problemlösungsfähigkeit unter Anwendung ingenieurmässiger Methoden nachweisen und verfügt über einen konzeptionellen, theoretischen und einen praktischen Anteil. Die erfolgreiche Durchführung der Arbeit ergibt 12 ECTS-Punkte, was einem Arbeitsaufwand von ca. 360 pro Person entspricht.

Das Startdatum ist der 16. Februar 2026 und der Abgabetermin ist der 12. Juni 2026 um 17:00 Uhr. Die genaue Modulbeschreibung ist im Anhang Sektion A.1 zu finden.

Da im Rahmen des Projekts mit Kundendaten gearbeitet wird, sind die geltenden Datenschutzgesetze strikt einzuhalten. Alle in dieser Dokumentation abgebildeten Daten und Diagramme wurden anonymisiert, um die Vorgaben zu erfüllen. Zudem ist eine Verschwiegenheitserklärung zu unterzeichnen und entsprechend zu befolgen.

1.6 Verwendete Technologien

Im Rahmen des Projekts wurden primär Microsoft-Technologien eingesetzt, da diese für den vorgesehenen Einsatz in einer Windows-Umgebung sowohl in der Entwicklung als auch im Betrieb klare Vorteile bieten. Insbesondere die Integration in Windows-Systeme ermöglicht eine effiziente Umsetzung von Hintergrundservices sowie einen direkten Zugriff auf lokale Ressourcen.

Zudem deckt die Toolchain nahezu alle Aspekte des Projekts ab, wodurch der gesamte Entwicklungsprozess effizient gestaltet werden kann.

1.6.1 .NET

.NET/C#[3] bildet das Herzstück des Projekts und überzeugt durch eine moderne Syntax, gute Tool-Integration sowie eine ausführliche Dokumentation, weshalb die neueste Long-Term-Support (LTS) Version 10.0 verwendet wurde.

.NET wurde ursprünglich von Microsoft entwickelt und Anfang der 2000er-Jahre als .NET Framework eingeführt. C# wurde dabei als zentrale Programmiersprache unter der Leitung von Anders Hejlsberg konzipiert. Heute ist .NET eine plattformübergreifende Open-Source-Plattform, die kontinuierlich weiterentwickelt wird.

C# bietet eine konsequent weiterentwickelte Sprachgestaltung, bei der moderne Features wie Pattern Matching, Nullable Reference Types und asynchrone Programmierung tief in die Sprache integriert sind und nicht nur als Zusatzbibliotheken existieren. Besonders hervorzuheben ist zudem die nahtlose Integration in die Windows-Umgebung, insbesondere im Bereich von Hintergrundservices, wodurch sich systemnahe Anwendungen effizient umsetzen lassen.

[2] „Modulbeschreibung - Bachelor-Arbeit Informatik“, OST – Ostschweizer Fachhochschule, 27. November 2025, https://studien.ost.ch/allModules/41626_M_BAI21.html.

[3] „.NET - Build modern apps and powerful cloud services“, Microsoft Corporation, zugegriffen 4. Mai 2026, https://dotnet.microsoft.com/en-us/.

1.6.1.1 EntityFramework

Entity Framework[4] ist ein von Microsoft entwickelter Object-Relational Mapper (ORM), der den Zugriff auf relationale Datenbanken über objektorientierte Modelle ermöglicht. Im Projekt wird Entity Framework ausschliesslich im Server eingesetzt, um die SQLite-Datenbank im Code-First-Ansatz zu verwalten.

Dabei werden Datenbankstrukturen direkt aus den definierten C#-Modellen generiert, wodurch eine konsistente und wartbare Datenpersistenzschicht entsteht. Änderungen am Datenmodell können so zentral im Code vorgenommen und automatisch in die Datenbankstruktur übernommen werden. Dies reduziert den manuellen Aufwand bei der Datenbankverwaltung und minimiert potenzielle Inkonsistenzen zwischen Anwendung und Datenbank.

1.6.1.2 ASP.NET

ASP.NET[5] bildet das Web-Framework des Servers und wird im Projekt zur Umsetzung der serverseitigen API verwendet. Es ermöglicht die Entwicklung von HTTP-basierten Schnittstellen, über welche der Client mit dem Server kommuniziert.

Das Framework ist Teil des .NET-Ökosystems und bietet eine integrierte Unterstützung für Routing, Request-Handling und Middleware-basierte Verarbeitung von Anfragen. Dadurch lassen sich serverseitige Logik und Datenzugriffe strukturiert und effizient umsetzen.

Im Projekt wird ASP.NET ausschliesslich im Server eingesetzt, um die serverseitige HTTP-API bereitzustellen. Es dient der Umsetzung der Kommunikationsschnittstelle zwischen Client und Server und ermöglicht die strukturierte Verarbeitung eingehender Anfragen.

1.6.1.3 Blazor

Blazor[6] ist ein Webentwicklungs-Framework von Microsoft, das es ermöglicht, interaktive Webanwendungen mit C# und .NET statt hauptsächlich mit JavaScript zu erstellen. Dabei werden Benutzeroberflächen als wiederverwendbare Komponenten entwickelt, die entweder direkt im Browser über WebAssembly (Blazor WebAssembly) oder auf dem Server mit einer Echtzeitverbindung zum Client (Blazor Server) ausgeführt werden. Blazor übernimmt das Rendern und Aktualisieren der Benutzeroberfläche automatisch, sodass Entwickler sich stärker auf die Anwendungslogik konzentrieren können. In IsoTrack kommt Blazor serverseitig zum Einsatz und bildet die Grundlage für die gesamte Benutzeroberfläche der Anwendung.

Für die Gestaltung der Benutzeroberfläche wird zusätzlich MudBlazor[7] verwendet, welches vorgefertigte UI-Komponenten im Material-Design bereitstellt und die Entwicklung der Oberfläche vereinfacht. Zur Visualisierung von Messdaten wird ein Blazor-Wrapper für ApexCharts[8] eingesetzt, der die direkte Integration interaktiver Diagramme in Blazor-Komponenten ermöglicht.

Die Kombination dieser Technologien fügt sich nahtlos in den bestehenden .NET-Stack ein und ermöglicht eine konsistente Entwicklung von Logik und Benutzeroberfläche innerhalb derselben Technologieumgebung.

---

[4] „Entity Framework-Dokumentationshub | Microsoft Learn“, Microsoft Corporation, zugegriffen 4. Mai 2026, https://learn.microsoft.com/de-de/ef/.

[5] „ASP.NET Core, an open-source web development framework | .NET“, Microsoft Corporation, zugegriffen 4. Mai 2026, https://dotnet.microsoft.com/en-us/apps/aspnet.

[6] „Blazor | Build client web apps with C# | .NET“, Microsoft Corporation, zugegriffen 4. Mai 2026, https://dotnet.microsoft.com/en-us/apps/aspnet/web-apps/blazor.

[7] „MudBlazor - Blazor Component Library“, MudBlazor Project, zugegriffen 4. Mai 2026, https://mudblazor.com/.

[8] „Blazor-ApexCharts“, GitHub, zugegriffen 4. Mai 2026, https://github.com/apexcharts/Blazor-ApexCharts.

1.6.2 SQLite

SQLite[9] ist eine leichtgewichtige, dateibasierte relationale Datenbank, die ohne separaten Server betrieben wird und alle Daten in einer einzelnen Datei speichert.

IsoTrack verwendet SQLite als Datenbank für die erfassten Patienten- und Messdaten im Server. Durch die serverseitige Verwaltung in Kombination mit Entity Framework wird eine strukturierte und konsistente Datenhaltung ermöglicht, ohne dass ein separater Datenbankserver betrieben werden muss. Aufgrund des erwarteten Datenvolumens im aktuellen Anwendungsfall stellt die Skalierbarkeit von SQLite dabei keine Einschränkung dar.

1.6.3 OLE DB

OLE DB[10] (Object Linking and Embedding, Database; dt. „Objektverknüpfung und -einbettung Datenbank“) ist eine von Microsoft entwickelte Datenbank-API, die den Zugriff auf unterschiedliche Datenspeicherformate über sogenannte OLE DB Provider ermöglicht.

Dazu gehören unter anderem Excel-Dateien, Dateisysteme und Access-Datenbanken, wie sie von Isocine3 verwendet werden.

OLE DB wird im Client verwendet, um direkt auf die Access-Datenbank von Isocine3 zuzugreifen. Aufgrund der bestehenden Systemumgebung stellt OLE DB die zentrale Schnittstelle für diesen Anwendungsfall dar und wurde daher als einzige geeignete Lösung gewählt.

1.6.4 WiX Toolset

Das WiX (Windows Installer XML) Toolset[11] ist ein Open-Source-Framework zur Erstellung von Windows-Installationspaketen auf Basis des Microsoft Installer (MSI). Es ermöglicht die deklarative Beschreibung von Installationslogik mittels XML und deren anschliessende Kompilierung zu ausführbaren Setup-Dateien. Ergänzend bietet FireGiant mit HeatWave[12] eine Visual-Studio-Erweiterung, welche die Konfiguration und Erstellung der Installationspakete direkt innerhalb der Entwicklungsumgebung ermöglicht.

Die erstellten Windows-Installationspakete werden verwendet, um dem Projektpartner eine selbstständige Installation der Anwendung zu ermöglichen und somit die Verteilung von IsoTrack innerhalb der Klinikvereinigung zu vereinfachen.

1.7 Stand der Technologie

Die Geräte von Easytech ermöglichen die grundlegende Erfassung verschiedener Formen der Muskelkontraktion im Bereich von Knie- und Schultergelenk. Auf dem Markt existieren jedoch weitere Anbieter, wie beispielsweise das Biodex System 4 Pro der Firma Biodex Medical Systems oder der Humac Norm von CSMi.

Beide Systeme verfügen über umfangreichere Einstellmöglichkeiten sowie über fortgeschrittene, insbesondere automatisierte Funktionen zur Datenerfassung und -visualisierung. Diese ermöglichen es Physiotherapeuten, umfangreiche Studien[13] durchzuführen, was mit dem Easytech Prima Plus in dieser Form nicht möglich ist.

---

[9] „SQLite Home Page“, The SQLite Consortium, 23. April 2026, https://sqlite.org/.

[10] „Übersicht über die OLE DB-Programmierung“, Microsoft Corporation, zugegriffen 4. Mai 2026, https://learn.microsoft.com/de-de/cpp/data/oledb/ole-db-programming-overview?view=msvc-170.

[11] „WiX Toolset“, FireGiant, zugegriffen 4. Mai 2026, https://www.firegiant.com/wixtoolset/.

[12] „HeatWave“, FireGiant, zugegriffen 4. Mai 2026, https://www.firegiant.com/heatwave/.

[13] „Veröffentlichte Studien mit PHYSIOMED GROUP Produkten | PHYSIOMED GROUP“, PHYSIOMED GROUP, zugegriffen 25. März 2026, https://www.physiomed-group.com/de-ch/services/studien#c15277.

Dieser Vergleich dient nicht der Kritik an Easytech; deren Produkte haben einen klar definierten Anwendungsbereich mit einem entsprechenden Preis-Leistungs-Verhältnis und sind nicht primär für die Datenerhebung im Rahmen wissenschaftlicher Studien konzipiert. Gemäss Angaben des Projektpartners liegt der Preis für das Easytech Prima Plus bei etwa 15'000 CHF, während vergleichbare Systeme erfahrungsgemäss weitaus höhere Kosten aufweisen.

1.7.1 Medizinische Messgeräte

Das System 4 Pro von Biodex bietet ein grösseres Anpassungsvermögen und ermöglicht das Messen weiterer Gelenke, wie zum Beispiel des Handgelenks. Zudem ist es Teil der „*Advantage BX*"-Domäne, welche bereits umfassende Möglichkeiten zur Datenvisualisierung für den Vergleich der Messungen eines Patienten über den Zeitverlauf bietet.

Vor allem erlaubt es dem Nutzer, mehrere Messdaten von mehreren Nutzern gleichzeitig anhand diverser Kriterien zu exportieren.

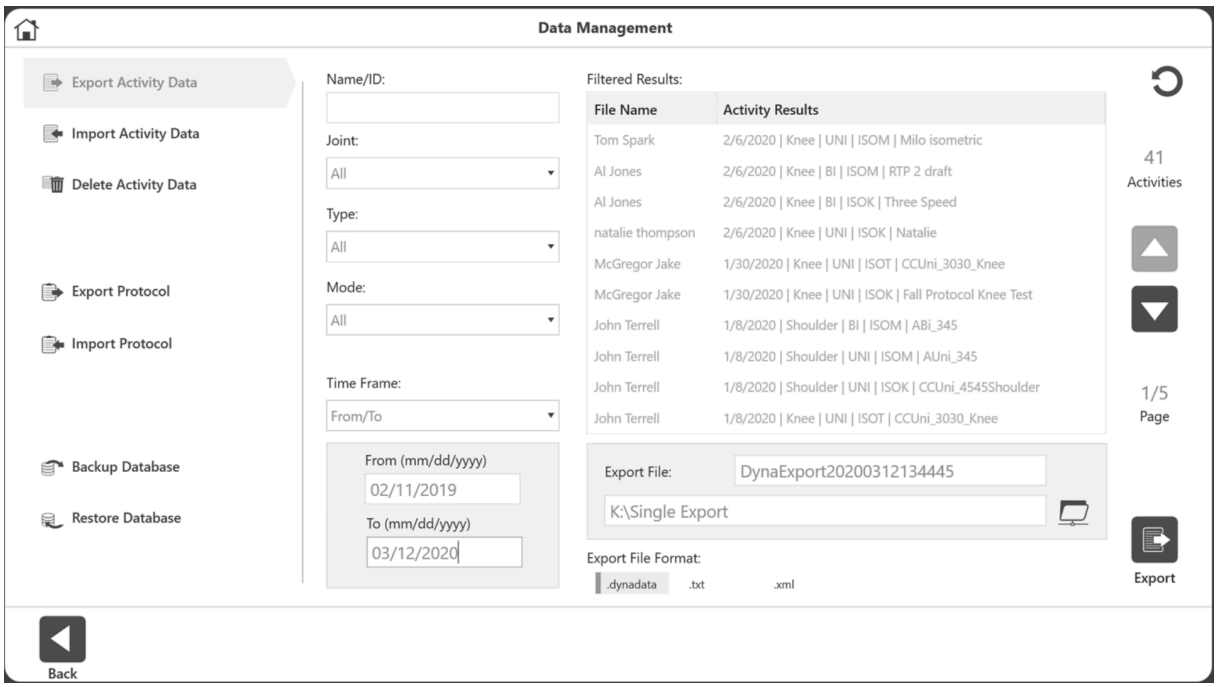


Abbildung 7: Multidatenexport in Biodex System 4 Pro[14]

Dabei stehen die Formate Extensible Markup Language (XML), Textdateien, welche so formatiert sind, dass ein Excel-Import nahtlos möglich ist, sowie ein proprietäres *.dynadata*-Format für die weitere Nutzung innerhalb der Biodex-Software-Domäne „*Advantage BX*" zur Verfügung.

Eine Möglichkeit, alle Messungen aller Patienten zentral einzusehen, existiert jedoch nicht und müsste über Software Dritter realisiert werden.

Alternativ bietet der Humac Norm von CSMi ebenfalls ein breites Spektrum an Gelenkmessungen und deckt mit 22 isolierten Bewegungsmustern Schulter, Ellbogen, Handgelenk, Hüfte, Knie und Sprunggelenk ab.

Bezüglich der Datenverwaltung bietet der Humac Norm einen besonders forschungsorientierten Ansatz: Das sogenannte *Group Summary*-Modul ermöglicht es, gespeicherte Daten gezielt zu durchsuchen und Patientengruppen anhand von Messmustern, Protokollen und demographischen Merkmalen zu filtern. Die

[14] „SYSTEM 4 (Advantage BX Software 5.3.X) INSTRUCTIONS FOR USE", Biodex Medical Systems, Inc., zugegriffen 31. März 2026, https://smti.co/hubfs/SportMedical_February2026/pdf/20-001-CLR-System-4-IFU-Rev-E.pdf?hsLang=co.

gefilterten Ergebnisse können anschliessend in eine Microsoft-Access-Tabelle exportiert oder als Gruppenvergleich ausgegeben werden.[15]

Für den Export einzelner Messungen stehen Tabellenkalkulations- sowie Textdateien zur Verfügung. Dabei werden Rohdaten automatisch gespeichert, sodass Berechnungen jederzeit mit aktuellen Einstellungen reproduzierbar sind. Berichte können zudem gedruckt, in der Vorschau angezeigt, per E-Mail versendet oder exportiert werden.

Eine zentrale Übersicht über alle Messungen aller Patienten ist jedoch, analog zum Biodex System 4 Pro, nicht nativ vorhanden und würde ebenfalls eine Drittlösung erfordern.

1.7.2 Smart Hospital

Eine weitere Arbeit, die eine ähnliche Problemstellung wie diese Bachelorarbeit adressiert, ist die Projektarbeit „Smart hospital: achieving interoperability and raw data collection from medical devices in clinical routine".[16] In dieser Arbeit wurden zahlreiche medizinische Messgeräte in ein sogenanntes Herz-Kreislauf-Informationssystem integriert, um eine einheitliche Datenerfassung und -verwaltung sowohl für den klinischen Betrieb als auch für Forschungszwecke zu ermöglichen. Der Ansatz entspricht damit grundsätzlich der Zielsetzung dieser Arbeit, jedoch im Enterprise-Kontext und mit einer stärker generalisierten Implementation.

Im Rahmen des Projekts wurden 39 Arten medizinischer Messungen, von der Abdomensonographie bis zur Bronchoskopie, durch mehr als 115 verschiedene Messgeräte in ein zentrales System integriert.

Als zentrale Lösungsansätze lassen sich insbesondere zwei Aspekte hervorheben. Zum einen die Integration der Messgeräte, welche durch die Entwicklung spezifischer Schnittstellen für die jeweiligen Geräte erreicht wurde. Dabei wurde darauf geachtet, die Schnittstellen möglichst generisch zu gestalten, um eine möglichst breite Abdeckung zu ermöglichen. Diese greifen auf eine zentral zugrunde liegende SQL-Datenbank (Structured Query Language) zu. Eine wesentliche Voraussetzung für die zentrale Verwaltung bildet dabei die Normalisierung der Daten, um die Interoperabilität sicherzustellen.

Zum anderen ist der Forschungsdatenexport von Bedeutung. Nutzer mit entsprechender Autorisierung können gezielt Daten anhand spezifischer Filter auswählen und diese mittels Drag-and-Drop exportieren. Zusätzlich stehen verschiedene Dashboards zur Visualisierung der Daten zur Verfügung.

Obwohl die betrachtete Lösung im Enterprise-Kontext angesiedelt ist, lassen sich zentrale Konzepte wie die standardisierte Datenintegration und der flexible Datenexport auf die vorliegende Problemstellung übertragen, vorwiegend im Hinblick auf eine mögliche zukünftige Erweiterung der Systemdomäne.

1.7.3 Smartwatches

Eine weitere Arbeit, die einen ähnlichen Lösungsansatz verfolgt, jedoch in einer anderen Domäne angesiedelt ist, ist die Arbeit „Development of a Research-Oriented Application for the Acquisition, Analysis, and Export of Sleep Metrics from Smartwatches" (dt. „Entwicklung einer forschungsorientierten Anwendung zur Erfassung, Analyse und zum Export von Schlafdaten aus Smartwatches").[17]

---

[15] „HUMAC® NORM™ TESTING & REHABILITATION SYSTEM User's Guide Model 502140", Computer Sports Medicine, Inc., 24. Februar 2025, https://humacnorm.com/wp-content/uploads/2025/03/300004O-409-HUMAC-NORM-User-Guide.pdf.

[16] „Smart hospital: achieving interoperability and raw data collection from medical devices in clinical routine", Bde. Volume6–2024, preprint, Frontiers in digital health, 6. März 2024, https://pmc.ncbi.nlm.nih.gov/articles/PMC10951085/pdf/fdgth-06-1341475.pdf.

[17] „Development of a Research-Oriented Application for the Acquisition, Analysis, and Export of Sleep Metrics from Smartwatches", Bd. 270, preprint, Elsevier B.V, o. J., zugegriffen 31. März 2026, https://www.sciencedirect.com/science/article/pii/S1877050925033253.

Im Rahmen dieses Projekts wurde eine Anwendung entwickelt, welche Schlafmetriken wie Herzfrequenz, Einschlaflatenz und Schlaffragmentierung als Rohdaten extrahierbar macht, erste Berechnungen und Kategorisierungen durchführt, in einer cloudbasierten Datenbank speichert und über eine mobile App visualisiert.

Der Fokus liegt dabei auf der Datenerfassung für Forschungszwecke: Ein Forscher stellt einer Testperson eine Smartwatch zur Verfügung, welche über einen gewissen Zeitraum Schlafdaten sammelt. Anschliessend können die Daten ausgewählt, zeitlich eingegrenzt und als JavaScript Object Notation (JSON) oder Comma-Separated Values (CSV) exportiert werden.

Die Motivation hinter diesem Ansatz liegt darin, dass herkömmliche Schlafmessverfahren kostenintensiv sind und den Einsatz von spezialisiertem Personal erfordern, während handelsübliche Smartwatches zu deutlich geringeren Kosten vergleichbare und bei neueren Modellen in einigen Fällen sogar genauere Ergebnisse liefern können.

Im Vergleich zu IsoTrack erfolgt die Datenerfassung nicht automatisch bei Änderungen der Stammdaten, sondern muss für jeden Patienten einzeln durchgeführt werden. Zudem werden bestimmte Schlafmetriken, wie beispielsweise die Einordnung in Schlafphasen, automatisiert mittels Python-Modulen berechnet, während bei IsoTrack die Analyse komplett durch den betreuenden Physiotherapeuten erfolgt.

Die Systemarchitektur weist dabei Parallelen zur Architektur von IsoTrack auf (siehe Kapitel 2.3.1), ergänzt jedoch um ein zusätzliches Modul zur Berechnung der Schlafmetriken.

2 Produkt Dokumentation

Dieser Teil dokumentiert das fertige Produkt detailliert und die Architektur anhand verschiedener Diagramme und Spezifikationen, insbesondere funktionale und nicht funktionale Anforderungen.

2.1 Anforderungsanalyse

Dieses Kapitel fasst alle beschreibenden Eigenschaften und Merkmale des Produkts zusammen, die in direkter Zusammenarbeit mit den Stakeholdern definiert wurden. Die erfassten Anforderungen dienen zugleich als Checkliste für die abschliessenden Abnahmetests (siehe Kapitel 2.5.3.4), wobei jede Anforderung einzeln überprüft wird. Dabei wird zwischen funktionalen Anforderungen (was) und nicht-funktionalen Anforderungen (wie) unterschieden.

2.1.1 Funktionale Anforderungen

Die funktionalen Anforderungen beschreiben die zentralen Features des Produkts. Sie beginnen mit den Epics, welche grössere Produktbereiche kategorisieren, und werden anschliessend in User Stories unterteilt. Jede User Story formuliert das jeweilige Feature in einem präzisen, in Alltagssprache formulierten Satz.

2.1.2 Epics

Die Funktionen des Produkts können grundsätzlich in Epics kategorisiert werden, also in funktionsübergreifende Kategorien, die jeweils ein gemeinsames übergeordnetes Ziel verfolgen. IsoTrack ist in folgende Epics aufgeteilt:

Jira Issue	Epic
ISOTRACK-26	Datenverwaltung
ISOTRACK-27	Datenvisualisierung

Tabelle 1: Epics.

2.1.3 User Stories

Eine User Story beschreibt einen konkreten, funktionalen Anwendungsfall aus Sicht eines Akteurs, ohne einen konkreten Umsetzungsvorschlag zu enthalten. Sie muss jedoch spezifisch genug sein, damit sie bei den abschliessenden Abnahmetests (siehe Kapitel 2.5.3.4) konkret überprüft werden kann.

Die folgenden Tabellen bieten eine Übersicht über alle User Stories, die ihren jeweiligen Epics zugeordnet und absteigend nach Priorität sortiert sind.

Datenverwaltung	
Jira Issue	**User Story**
ISOTRACK-30	Als User möchte ich alle Patientendaten als CSV- oder Exceldatei auf einmal exportieren können, um Zeit zu sparen.
ISOTRACK-31	Als User möchte ich die Patientendaten eines Patienten als CSV- oder Exceldatei exportieren können, um Zeit zu sparen.
ISOTRACK-32	Als User möchte ich meine Microsoft Access Datenbank als Quelle angeben, um das Exportieren zu automatisieren.
ISOTRACK-33	Als User möchte ich meine Daten automatisch synchronisieren, um Fehler zu vermeiden.
ISOTRACK-34	Als User möchte ich die Messdaten von Patienten anhand eines Referenzwertes vergleichen können.
ISOTRACK-35	Als User möchte ich bei Patienten zusätzliche Daten, wie das Operationsdatum, angeben können
ISOTRACK-91	Als User möchte ich bei einer Messung die falsch angegebene Beinseite korrigieren können, um fehlerhafte Daten zu vermeiden.

Tabelle 2: Datenverwaltung-User Stories absteigend nach Priorität sortiert.

Datenvisualisierung	
Jira Issue	**User Story**
ISOTRACK-36	Als User möchte ich einen Überblick über die Daten, z. B. Zeitverlauf oder mehrere Messungen, eines spezifischen Patienten erhalten, um Schlüsse daraus zu ziehen zu können.
ISOTRACK-37	Als User möchte ich die Patientenauswahl anhand bestimmter Kriterien, wie Alter oder Messdatum, filtern können.
ISOTRACK-38	Als User möchte den Graphen für den Messverlauf einer Messung visualisiert bekommen haben.
ISOTRACK-39	Als User möchte den Graphen für den Messverlauf mehrerer Messungen eines Patienten visualisiert bekommen haben.
ISOTRACK-40	Als User möchte ich den Graphen für den Messverlauf aller Messungen visualisiert bekommen haben.
ISOTRACK-41	Als User möchte ich eine Übersicht über alle Daten haben, um einen Gesamtüberblick zu erhalten.
ISOTRACK-66	Als User möchte ich eine Übersicht über die Datenqualität aller Daten haben.
ISOTRACK-86	Als User möchte ich zwei Messungen der beiden Beine eines Patienten als PDF exportieren können.

Tabelle 3: Datenvisualisierung-User Stories absteigend nach Priorität sortiert.

2.1.4 Nicht-Funktionale Anforderungen

Die nicht-funktionalen Anforderungen (NFA) definieren die konkreten Qualitätsansprüche des Produkts anhand eines spezifischen Ziels bzw. bei Performanzkriterien anhand von Erreichungsgraden (sogenannte

„*Landing Zones*“). Qualitative Anforderungen werden anhand des ISO/IEC-Standards 25010:2023[18] kategorisiert und erfüllen die SMART-Kriterien, um eine überprüfbare Kontrolle während der Abnahmetests (siehe Kapitel 2.5.3.4) zu ermöglichen.

Die SMART-Kriterien umfassen folgende Punkte:
- **S**pezifisch: Das Qualitätsattribut ist klar definiert und abgegrenzt.
- **M**essbar: Die Anforderung lässt sich reproduzierbar messen.
- **A**kzeptiert: Das Erreichen der Anforderung bietet einen relevanten Mehrwert für den Kunden.
- **R**ealistisch: Die Umsetzung und Prüfung der Anforderung ist im realistischen Rahmen möglich.
- **T**erminiert: Die Anforderung muss bis zu einem festgelegten Zeitpunkt erfüllt sein; im Rahmen dieses Projekts wird dafür der Abnahmetesttermin verwendet.

Neben den SMART-konkretisierbaren Anforderungen wird Skalierbarkeit als architektonisches Qualitätsziel berücksichtigt. Sie ist im vorliegenden Projekt nicht direkt messbar oder testbar, wurde jedoch im Systemdesign durch eine lose gekoppelte Client-Server-Architektur, eine ausführliche Dokumentation des Repositories sowie die Möglichkeit zur horizontalen Erweiterung des Clients berücksichtigt.

NFA 01 - Vertraulichkeit	
Kategorie	Sicherheit
Beschreibung	Die Patientendaten unterliegen strengster Vertraulichkeit und der Zugriff auf diese muss auf das ärztliche Personal beschränkt bleiben.
Ziele	Der Zugriff auf die Patientendaten ist nur innerhalb der Sportzentrum Infrastruktur möglich.
Priorität	Sehr hoch
Anmerkungen	Aufgrund dieser Anforderung fiel die Entscheidung auf ein lokales Deployment des Servers. Dadurch verlassen die Patientendaten nie die Räumlichkeiten der Rosenklinik.

Tabelle 4: NFA 01 - Vertraulichkeit

NFA 02 - Benutzerunterstützung	
Kategorie	Interaktionsvermögen
Beschreibung	Nach Ablauf des Projekts wird die Wartung und der Support voraussichtlich stark reduziert oder ganz entfallen. Damit der Kunde sich selbst helfen kann und sich zukünftige Mitarbeitende schnell einarbeiten können, müssen entsprechende Vorkehrungen getroffen werden.
Ziele	Dem Produkt wird eine ausführliche Bedienungsanleitung beigelegt, welche Architektur und Verhalten der Software beschreibt. Die Sprache richtet sich an Personen, die mit der Bedienung von Windows vertraut sind.
Priorität	Mittel
Anmerkungen	Die Benutzeranleitung ist im Anhang Sektion A.2 dokumentiert.

Tabelle 5: NFA 02 - Benutzerunterstützung

[18] *ISO/IEC 25010:2023* (International Organization for Standardization, 2023), https://www.iso.org/obp/ui/en/#iso:std:iso-iec:25010:ed-2:v1:en.

NFA 03 - Angemessenheit Erkennbarkeit / Selbstbeschreibbarkeit	
Kategorie	Interaktionsvermögen
Beschreibung	Beim Kunden handelt es sich primär um Physiotherapeuten mit begrenztem IT-Hintergrund. Um die Benutzung zu erleichtern, muss die Benutzeroberfläche dem gewohnten Arbeitsumfeld entsprechen.
Ziele	Die Benutzeroberfläche des Produktes ist simpel und effektiv gestaltet und richtet sich an medizinisches Fachpersonal. Sie orientiert sich in Aufbau und Bedienlogik an der bestehenden Easytech-Software.
Priorität	Hoch
Anmerkungen	-

Tabelle 6: NFA 03 - Angemessenheit Erkennbarkeit / Selbstbeschreibbarkeit.

NFA 04 - Installierbarkeit	
Kategorie	Flexibilität
Beschreibung	Die Installation, Konfiguration und allfällige Deinstallation von Services ist für technisch unversierte Personen aufwendig und fehleranfällig. Auch wenn davon ausgegangen wird, dass diese Vorgänge nur selten durchgeführt werden, sollen sie künftig ohne Unterstützung des Projektteams möglich sein.
Ziele	Für den Client steht eine ausführbare Windows-Installer-Datei (.msi) zur Verfügung, welche die Installation und Deinstallation des Clients übernimmt.
Priorität	Niedrig
Anmerkungen	-

Tabelle 7: NFA 04 - Installierbarkeit

2.2 Domänenanalyse

In diesem Kapitel wird die Problemdomäne im Sinne des Domain-driven Design analysiert und mithilfe einer Context Map dargestellt. Die daraus abgeleiteten Erkenntnisse fliessen in die Implementierung von IsoTrack ein und werden in einem Domänenmodell festgehalten.

Wie in Kapitel 3.2.7.1 beschrieben, bestehen Inkonsistenzen in der Isocine3-Datenbank, die auf Fehlübersetzungen oder spezifische Konfigurationsentscheidungen bei der ursprünglichen Einrichtung zurückzuführen sind. Diese führen dazu, dass das durch IsoTrack abgebildete Domänenmodell geringfügig vom ursprünglichen, durch Easytech definierten Domänenmodell abweicht.

Dies stellt ein potenzielles Risiko dar, falls zukünftig ein Update durch Easytech in der Infrastruktur der Rosenklinik durchgeführt würde und sich dabei das zugrunde liegende Datenmodell verändert. Aufgrund der sehr geringen Eintrittswahrscheinlichkeit dieses Szenarios wird dieses Risiko im Rahmen dieser Arbeit jedoch nicht weiter berücksichtigt.

In Abbildung 8 ist die Context Map des Projekts dargestellt. IsoTrack befindet sich hierbei downstream (*D*) von Isocine3 (*U*) und ist somit Empfänger des Datenflusses und hat keinen Einfluss auf den Upstream. Zusätzlich ist IsoTrack ein Conformist (*CF*) und passt sein Modell an das von Isocine3 an. Änderungen im Upstream haben daher einen direkten Einfluss auf IsoTrack.

Isocine3 definiert weder eine Published Language (*PL*) noch einen Open Host Service (*OHS*). Es ist klar ersichtlich, dass Isocine3 nicht für die Zusammenarbeit mit Dritten konzipiert wurde.

Der Datenfluss erfolgt über die bestehenden Persistenzartefakte des Isocine3-Systems, insbesondere Microsoft Access-Datenbanken und TRA-Dateien, die als Datenquelle für IsoTrack verwendet werden.

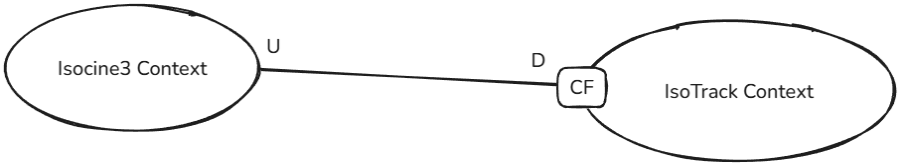


Abbildung 8: Context Map der beiden Bounded Contexts des Projekts.

Das Fehlen eines expliziten Datenvertrags führt dazu, dass IsoTrack auf implizite Annahmen über die Datenqualität von Isocine3 angewiesen ist. Dies betrifft insbesondere die Vollständigkeit und Konsistenz von Aufzeichnungseinträgen. Zudem müssen potenziell fehlende oder unvollständige Werte im Rahmen der Verarbeitung berücksichtigt werden, da keine Validierung im Quellsystem vorgenommen werden kann.

Die Analyse der Problemdomäne zeigt, dass nur eine begrenzte Anzahl expliziter fachlicher Konzepte und Regeln vorhanden ist. Entsprechend fällt das resultierende Domänenmodell vergleichsweise schlank aus und konzentriert sich auf die zentralen Entitäten und deren Beziehungen.

2.2.1 Domänenmodell

Im Domänenmodell (vgl. Abbildung 9) sind die Entitäten der Domäne dargestellt. Es bildet primär die Domäne von Isocine3 ab und erweitert diese um zusätzliche Konzepte wie Indikationen auf Messebene, Operationsdaten sowie die neue Entität der Drehmomentreferenzwerte.

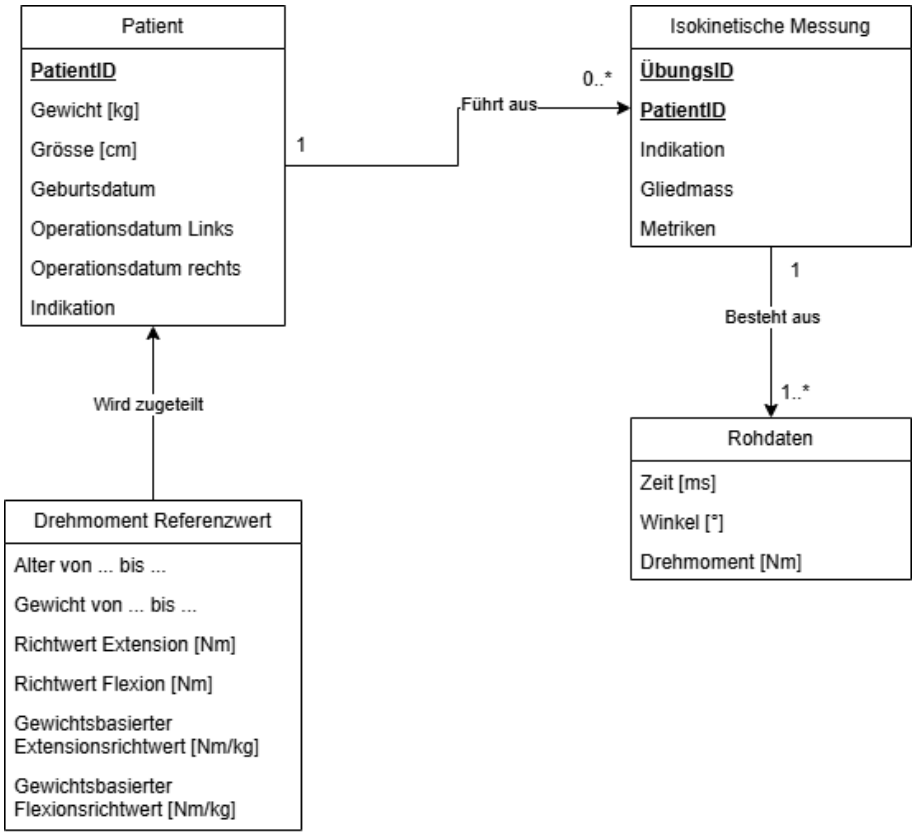


Abbildung 9: Domänenmodell von IsoTrack.

Patient

Der Patient besteht aus den folgenden, in Isocine3 angegebenen Daten:

- **PatientID** (Hauptschlüssel)
- Nachname
- Vorname
- Geschlecht
- Geburtsdatum
- Gewicht (kg)
- Grösse (cm)
- Dominante Seite
- Unfalldatum
- *Adresse*
- *Postleitzahl*
- *Ort*
- *Provinz*
- *Land*
- *Telefon*
- Therapeut
- Knieausgleich
- *Hüftausgleich*
- *Sprunggelenkausgleich*
- Anmerkungen
- Maximale Messanzahl
- Maximale Testanzahl

- Indikation
- Arzt

Die *kursiv* markierten Attribute werden von IsoTrack nicht verwendet. Die im Domänenmodell dargestellten Attribute (vgl. Abbildung 9) bilden hingegen die für IsoTrack relevanten Patientendaten ab.

Zusätzlich können in IsoTrack Operationsdaten getrennt für das linke und rechte Bein erfasst werden. Diese Informationen dienen insbesondere der zeitlichen Einordnung von Messungen innerhalb der Datenanalyse.

Isokinetische Messung

Die Entität der isokinetischen Messung verfügt über einen zusammengesetzten Primärschlüssel aus MessID und PatientID, wobei die PatientID gleichzeitig als Fremdschlüssel auf die Patientenentität verweist. Über diese Identifikatoren erfolgt zudem die Zuordnung zu den zugehörigen Rohdaten in den TRA-Dateien.

Sie besteht aus den folgenden, in Isocine3 angegebenen Daten. Messgrössen (Wert) wie Winkel, Drehmoment, Kraft und Arbeit werden jeweils getrennt für Flexion und Extension gespeichert.

- **MessID** (Hauptschlüssel)
- **PatientID** (Hauptschlüssel)
- Datum/Uhrzeit
- Gliedmass
- Seite
- *Position*
- *Messtyp*
- *Messmodus*
- *FMCompCos*
- Flexionsgeschwindigkeit
- Extensionsgeschwindigkeit
- Wiederholungen pro Satz
- Durchgeführte Wiederholungen
- *Flexionswinkel*
- *Extensionswinkel*
- Maximaler Wert
- Winkel bei Maximalem Wert
- Mittelwert Spitzenwert
- Maximaler Wert pro Wiederholung
- Mittlerer Wert pro Wiederholung
- Gesamtwert
- Widerstand
- Effektive Zeit (Sek.)
- Gesamtzeit (Sek.)
- *Attribut*
- *Arzt*
- *Bediener*
- *Berichtspfad*
- *Zusätzliche Daten*
- *Serienname*
- *Test-ID*

Zusätzlich wird für jede Messung eine Indikation gespeichert. Standardmässig wird dabei die Indikation des Patienten übernommen. Dadurch können auch Messungen von Patienten mit mehreren Indikationen eindeutig einer spezifischen Indikation zugeordnet werden.

Rohdaten

Die Rohdaten werden in TRA-Dateien gespeichert, welche die vom Messgerät aufgezeichneten Messwerte enthalten (siehe Kapitel 2.3.2.2). Sie bilden die Grundlage für die graphischen Darstellungen innerhalb von IsoTrack.

Drehmomentreferenzwert

Die Drehmomentreferenzwerte dienen der Einordnung der gemessenen Muskelkraft eines Patienten. Sie werden anhand von Alter, Geschlecht und Gewicht den jeweiligen Messungen gegenübergestellt und unterstützen die Therapeuten bei der Interpretation der Ergebnisse.

Die Referenzwerte können durch die Therapeuten entsprechend dem aktuellen medizinischen Kenntnisstand gepflegt und angepasst werden. Eine vergleichbare Funktionalität ist in Isocine3 nicht vorhanden.

2.3 Architektur

Dieses Kapitel umfasst die architektonische Dokumentation der im Projekt eingesetzten Softwarekomponenten sowie die daraus abgeleiteten architektonischen Entscheidungen.

2.3.1 Softwarearchitektur

Zur Visualisierung der Softwarearchitektur von IsoTrack wird die C4-Modell-Notation[19] verwendet. Dieses zeichnet sich durch einen hohen Abstraktionsgrad aus und ermöglicht die Darstellung der Architektur auf unterschiedlichen Detailebenen. Dabei wird von einer groben Kontextübersicht ausgegangen, die schrittweise bis hin zu einzelnen Komponenten verfeinert wird.

Der Begriff „C4“ steht für die vier Diagrammtypen Context, Container, Component und Code. Die Komponentendiagramme lassen sich in den jeweiligen Komponentenkapitel (siehe Kapitel 2.3.3 und Kapitel 2.3.4) auffinden. Das Code-Diagramm, welches eine nahezu 1:1-Abbildung der Klassenstruktur vorsieht, wird dabei in der Regel ausgelassen, da die Aussagekraft als Diagramm gering ist.

2.3.1.1 Kontextdiagramm

Im Kontextdiagramm werden die Systemumgebung von IsoTrack sowie dessen Beziehung zum Isocine3-System dargestellt. Dabei werden die Ergebnisse der Domänenanalyse (Kapitel 2.2) wiedererkennbar.

Es werden die beiden Systemkontexte explizit gegenübergestellt: der Systemkontext von IsoTrack sowie der externe Systemkontext des Isocine3-Systems. Dadurch werden die Interaktionen des Nutzers mit beiden Systemen sowie deren klare Abgrenzung sichtbar.

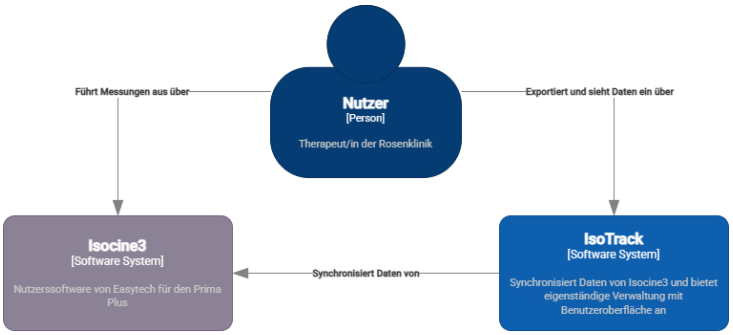


Abbildung 10: C4 Kontextdiagramm der IsoTrack Architektur.

2.3.1.2 Containerdiagramm

Das Containerdiagramm stellt die interne Aufteilung der IsoTrack-Architektur in einzelne Laufzeitkomponenten dar und zeigt den Client/Server-Split der Anwendung. Der Server ist dabei unabhängig vom Client konzipiert und kennt dessen interne Struktur nicht.

Zusätzlich werden die jeweiligen Datenhaltungssysteme der Container sowie deren Aufgaben innerhalb der Gesamtarchitektur dargestellt.

[19] Simon Brown, „C4 model“, zugegriffen 6. Mai 2026, https://c4model.com/.

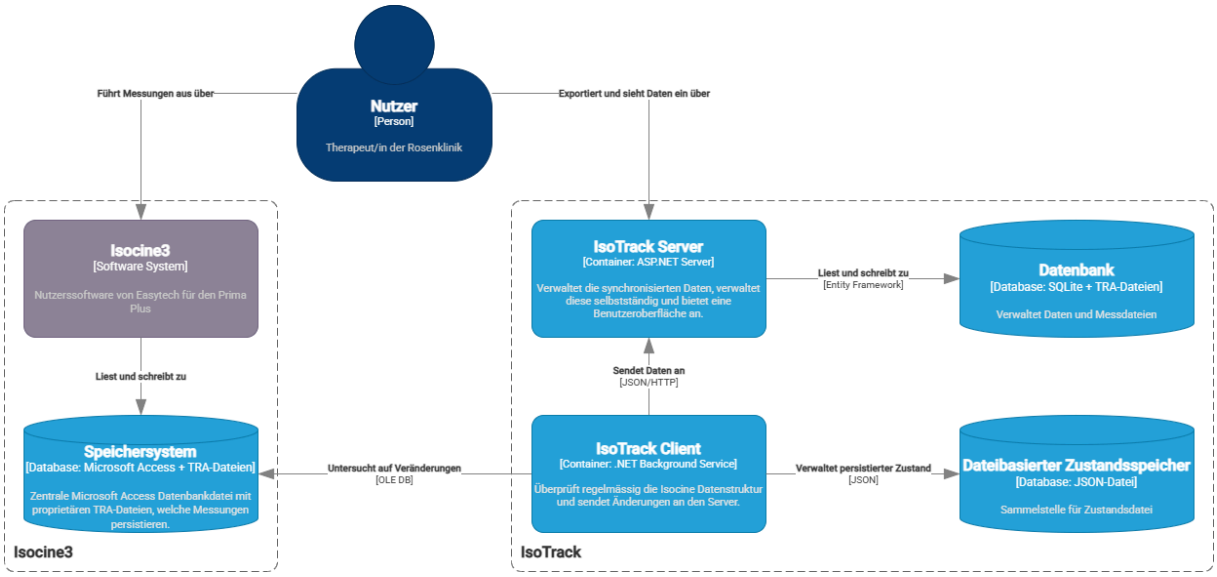


Abbildung 11: C4 Containerdiagramm der IsoTrack Architektur.

Es lassen sich dabei Parallelen zu den in Kapitel 1.7.2 und Kapitel 1.7.3 beschriebenen Architekturen erkennen.

Im Smart Hospital (vgl. Abbildung 12) entspricht das Cardio-Vascular-Information-System (CVIS) funktional dem IsoTrack Server. Einzelne Dienste werden dort jedoch zusätzlich auf eine separate HIS-Plattform ausgelagert. Die verschiedenen Abteilungen synchronisieren über eigene Interfaces mit dem CVIS, was der Rolle des Clients in IsoTrack entspricht. Hier wird ersichtlich, wie die IsoTrack-Lösung potenziell auf mehrere Messsysteme erweitert werden könnte und wie der Smart-Hospital-Ansatz dieses Architekturprinzip auf Enterprise-Ebene skaliert.

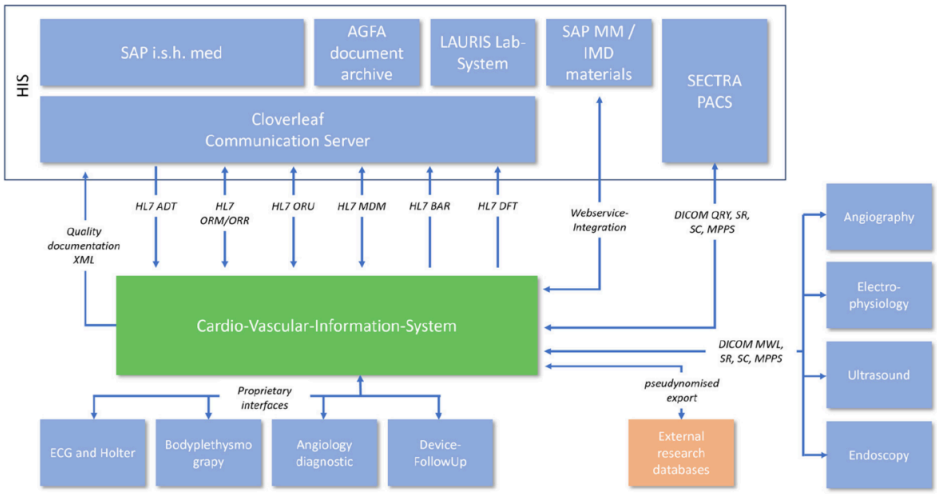


FIGURE 2
System interface overview.

Abbildung 12: Softwarearchitektur des Smart Hospitals[20]

Im Bereich der Smartwatches (vgl. Abbildung 13) zeigt sich eine noch stärkere strukturelle Ähnlichkeit. Die Smartwatch übernimmt dabei die Rolle des IsoTrack Clients, während die Kombination aus Smartphone beziehungsweise lokalem Server und Cloud-Infrastruktur funktional dem IsoTrack Server entspricht. Die

[20] „Smart hospital: achieving interoperability and raw data collection from medical devices in clinical routine“, Bde. Volume6–2024.

zugrunde liegende Kommunikationsstruktur lässt sich dabei nahezu direkt auf die IsoTrack-Architektur übertragen.

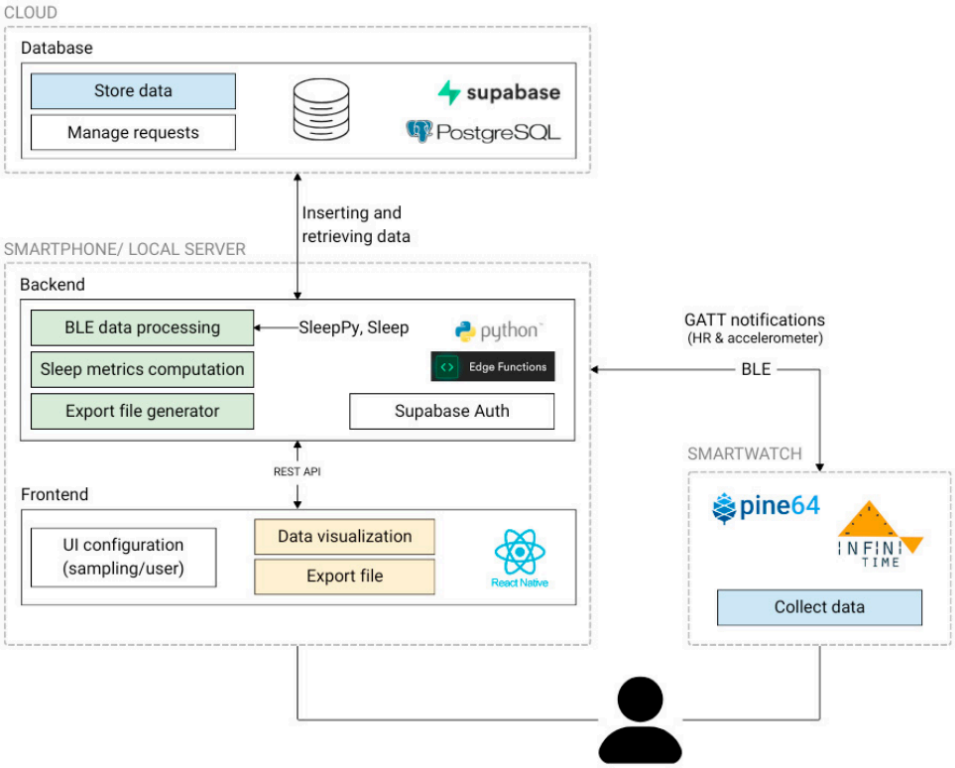


Abbildung 13: Softwarearchitektur der Smartwatches[21]

2.3.2 Isocine3

Isocine3 ist die von Easytech mitgelieferte, proprietäre Software zur Nutzung des Prima Plus. Sie ermöglicht die Aufnahme von Messdaten und speichert die Resultate sowohl in einer Microsoft-Access-Datenbank als auch in proprietären „TRA"-Dateien. Die Software wurde vermutlich in C++ entwickelt, was sich aus dem Alter der Access-Datenbank (Erstellungsdatum 2003) sowie der Verwendung von Dynamic Link Libraries (DLL) ableiten lässt.

Für die Software steht keine öffentlich zugängliche Dokumentation zur Verfügung, und der Hersteller reagierte nicht auf Kontaktaufnahmeversuche. Daher basieren alle gewonnenen Erkenntnisse auf einer eigenständigen Analyse der vorliegenden Daten und wurden nach bestem Wissen und Gewissen erarbeitet.

Die Verzeichnisstruktur von Isocine3 ist in mehrere funktionale Bereiche unterteilt (siehe Abbildung 14). Der Ordner „archivio" stellt dabei den zentralen Datenspeicher dar und enthält sowohl die Access-Datenbank als auch die zugehörigen TRA-Dateien. Die Unterordner DAT_* repräsentieren verschiedene Messarten und dienen der strukturierten Ablage der jeweiligen Messdaten. Weitere Verzeichnisse wie „Serie" und „Test" enthalten Konfigurationsdateien für Messungen, während „Utilities" verschiedene Hilfsprogramme umfasst, die vom Hauptsystem genutzt werden (z. B. für Datenexport oder Konfiguration).

```
Easytech_Isocine3
├──archivio
│   ├──DAT_CPM
│   ├──DAT_IDRODINAMICO
│   ├──DAT_INERZIALE
```

[21] „Development of a Research-Oriented Application for the Acquisition, Analysis, and Export of Sleep Metrics from Smartwatches", Bd. 270.

```
│   ├──DAT_ISOCINETICO
│   └──DAT_ISOMETRICO
├──Empty_archivio
│   ├──DAT_CPM
│   ├──DAT_IDRODINAMICO
│   ├──DAT_INERZIALE
│   ├──DAT_ISOCINETICO
│   └──DAT_ISOMETRICO
├──IsoCPU
├──Logo
├──Serie
│   ├──Genu
│   └──Prima
├──Sound
├──Test
│   ├──1_Limb
│   └──2_Limbs
├──Tutor_Files
├──Utilities
│   ├──BackupINI
│   ├──Firmware
│   ├──Iso2Txt
│   ├──IsoCfgIsocineINI
│   │   └──installer
│   ├──IsoDateExSync
│   │   ├──immagini
│   │   └──installer
│   ├──IsoExportDB
│   │   ├──immagini
│   │   └──installer
│   ├──IsoTest
│   │   ├──cypress_bootloader
│   │   ├──Installer
│   │   └──IsoTest3-CyUSB
│   └──RecoveryINI
└──Video_Tutorial
```

Abbildung 14: Verzeichnisstruktur der Software

2.3.2.1 Microsoft Access Datenbank

Die Access-Datenbank stellt den zentralen Speicherort für die Datenverwaltung der Software dar. Änderungen werden jeweils beim Schliessen der Anwendung persistiert. Die Tabellen sowie deren Spaltenbezeichnungen sind vollständig in italienischer Sprache gehalten (siehe Abbildung 15).

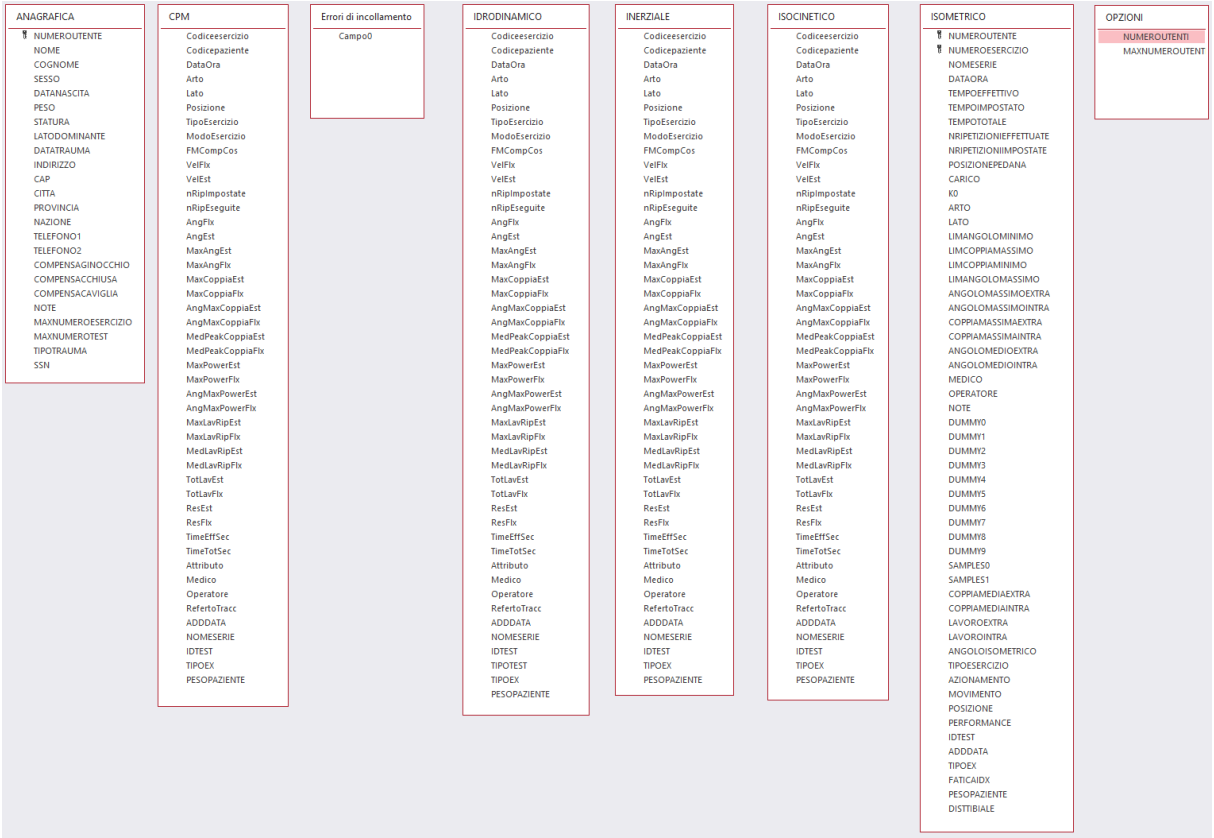


Abbildung 15: Tabellen der Datenbank mit italienischen Spaltennamen.

Die Datenbank ist in acht Tabellen unterteilt:

1 Patientendatentabelle	
ANAGRAFICA	Anzumerken ist, dass die Spalten für Vor- („NOME“) und Nachname („COGNOME“) vertauscht implementiert wurden. Diese Inkonsistenz wird in IsoTrack korrigiert. Zudem wird die Spalte für die italienische Provinz fälschlicherweise als „Pat.-Nr.“ (Patientennummer) interpretiert.
5 Messdatentabellen	
CPM	Für passive Bewegungsmessungen.
IDRODINAMICO	Für hydrodynamische Messungen.
INERZIALE	Für isotonische Messungen.
ISOCINETICO	Für isokinetische Messungen.
ISOMETRICO	Für isometrische Messungen.
1 Konfigurationstabelle	
OPZIONI	Enthält zwei Spalten mit der Anzahl Patienten („NUMEROUTENTI“) sowie die höchste vergebene Patientennummer („MAXNUMEROUTENTE“).
1 nicht eindeutig zuordenbare Tabelle	
Errori di incollamento	dt. „Einfügefehler“, mit einer einzigen Spalte „Campo0“ (dt. „Feld0“) und einer Zeile „DataRow dr“. Möglicherweise durch einen Implementationsfehler des Datenbankschreibers entstanden.

Tabelle 8: Tabellen der Access-Datenbank.

Die Datenbank verwendet ausschliesslich indirekte Beziehungen. So entspricht beispielsweise die Spalte „NUMEROUTENTE“ (dt. Nutzernummer) der Tabelle „ANAGRAFICA“ den „Codicepaziente“-Spalten der Messtabellen; eine explizite relationale Verknüpfung ist jedoch nicht implementiert.

Dennoch sind Ansätze einer rudimentären Strukturierung erkennbar. Dies zeigt sich unter anderem daran, dass die Tabelle „ISOMETRICO“ die Patientennummer, in den anderen Messtabellen als „Codicepaziente“ bezeichnet und hier als „NUMEROUTENTE“ geführt, als zweiten Primärschlüssel definiert. Generell unterscheidet sich die isometrische Tabelle zudem dadurch, dass alle Feldnamen in Grossbuchstaben vorliegen und mehrere Dummy-Spalten enthalten sind.

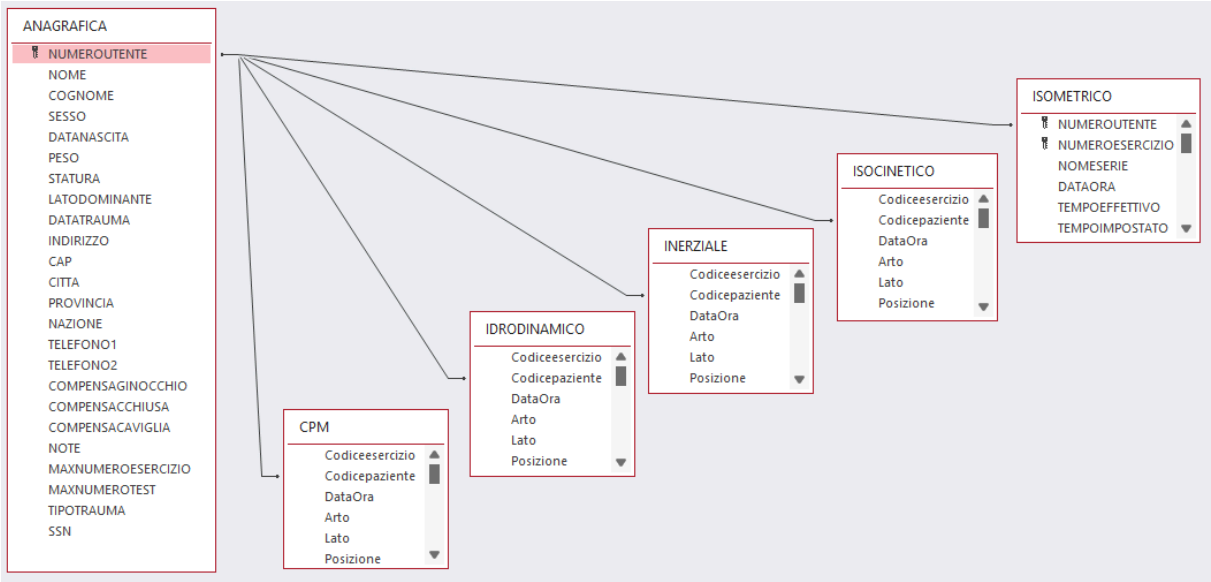


Abbildung 16: Abgeleitetes Beziehungsdiagramm der Tabellen, dargestellt mit 1:n-Beziehungen in Microsoft Access.

Wie bereits erwähnt, bietet das Prima Plus zusätzlich die Möglichkeit, isometrische, isotonische, hydrodynamische sowie passive Bewegungsmessungen durchzuführen. Das Sportzentrum Rosenklinik verwendet jedoch ausschliesslich isokinetische Kniemessungen, weshalb die übrigen Messtypen im Rahmen dieses Projekts keine Relevanz besitzen. Aufgrund der strukturellen Ähnlichkeit der Tabellen wäre eine spätere Integration jedoch grundsätzlich problemlos möglich.

2.3.2.2 TRA-Dateien

In der Datenbank werden lediglich Metadaten und Eckkennzahlen einer Messung gespeichert; die eigentlichen Messwerte liegen in separaten, proprietären Binärdateien vor. Eine solche Datei enthält jeweils sämtliche Messwerte einer Messung.

Wie bereits erwähnt, befinden sich die Dateien im Verzeichnis `/archivio/DAT\_<MESSART>` und tragen die Dateiendung `.TRA`. Der Dateiname setzt sich aus der fünfstelligen Patientennummer sowie der dreistelligen Messnummer zusammen und bildet damit den Primärschlüssel der entsprechenden Tabelle in der Access-Datenbank ab.

2.3.2.2.1 Struktur

Die TRA-Dateien verwenden ein nicht dokumentiertes, proprietäres Binärformat. Um die Struktur der Dateien zu rekonstruieren, wurde Python explorativ eingesetzt, um verschiedene Kombinationen aus Offsets, Datentypen und Endianness zu testen und die resultierenden Daten zu visualisieren (siehe Abbildung 17).

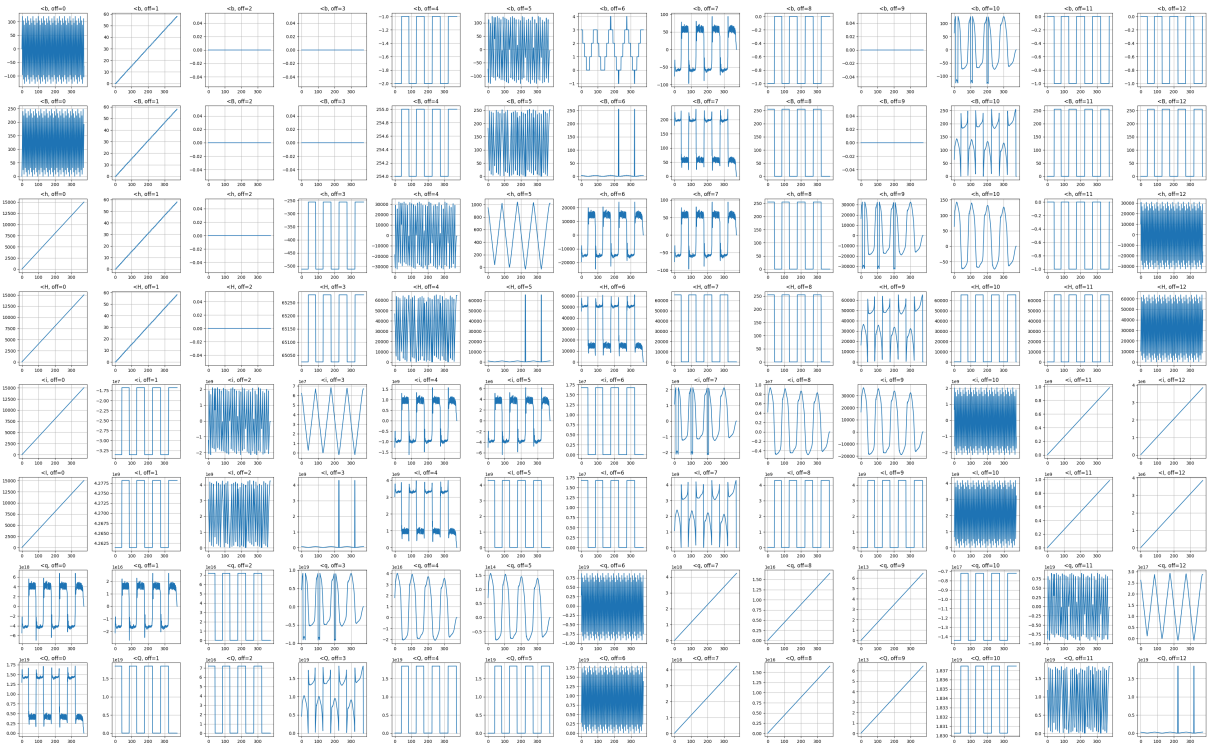


Abbildung 17: Visualisierung einer TRA-Datei mit verschiedenen Datentypen (Offset: 13 Bytes, Little-Endian).

In den resultierenden Darstellungen lässt sich beispielsweise in der dritten Zeile und elften Spalte ein Verlauf erkennen, der den Drehmomentwerten einer Messung entspricht. Daraus konnte abgeleitet werden, dass das Drehmoment jeweils mit einem Offset vom Zeilenbeginn von 10 Bytes (`off=10`) als Little-Endian (`<`) Signed Integer (`h`) gespeichert wird (vergrössert dargestellt in Abbildung 18).

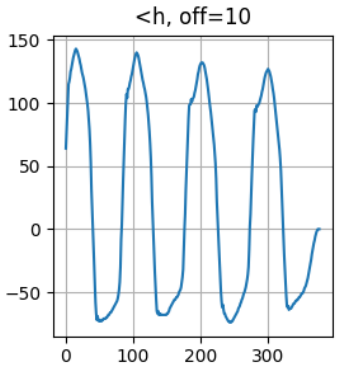


Abbildung 18: Ausschnitt: Zeile 3, Spalte 11 aus der Python-Visualisierung der TRA-Datei.

Auf Basis dieser Analysen konnte schliesslich folgende Datenstruktur rekonstruiert werden:

Zeit [ms] Winkel [10°] Drehmoment [Nm]

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C
00000000	00	00	00	00	FE	B6	03	B4	FF	00	40	00	00
0000000D	28	00	00	00	FE	A2	03	D5	FF	00	4D	00	00
0000001A	50	00	00	00	FE	8E	03	E2	FF	00	5B	00	00
00000027	78	00	00	00	FE	84	03	CB	FF	00	69	00	00
00000034	A0	00	00	00	FE	70	03	CE	FF	00	73	00	00
00000041	C8	00	00	00	FE	5C	03	CC	FF	00	75	00	00
0000004E	F0	00	00	00	FE	3E	03	C5	FF	00	78	00	00
0000005B	18	01	00	00	FE	2A	03	C6	FF	00	7D	00	00
00000068	40	01	00	00	FE	16	03	C1	FF	00	7F	00	00
00000075	68	01	00	00	FE	F8	02	C1	FF	00	82	00	00
00000082	90	01	00	00	FE	E4	02	C5	FF	00	85	00	00
0000008F	B8	01	00	00	FE	C6	02	C5	FF	00	88	00	00
0000009C	E0	01	00	00	FE	B2	02	C5	FF	00	8A	00	00
000000A9	08	02	00	00	FE	94	02	C5	FF	00	8C	00	00
000000B6	30	02	00	00	FE	80	02	C6	FF	00	8E	00	00
000000C3	58	02	00	00	FE	62	02	C1	FF	00	8F	00	00

Abbildung 19: Struktur einer TRA-Datei

Die Zeitachse inkrementiert in Schritten von 40 ms, was mit der 25 Hz Samplingrate des Easytech Prima Plus darauf schliessen lässt, dass die Zeit in Millisekunden gespeichert wird (1s / 25 = 40ms). Nicht beschriftete Spalten enthalten unter anderem Status-Flags, beispielsweise zur Bewegungsrichtung, sind für die vorliegende Arbeit jedoch nicht weiter relevant.

2.3.3 Client

Der IsoTrack Client ist ein .NET-basierter Windows-Hintergrundservice mit der Aufgabe, die lokalen Daten aus Isocine3 automatisch mit dem Server zu synchronisieren. Dabei wurde der Fokus auf eine schlanke und robuste Implementierung gelegt, wobei in gewissen Bereichen bewusst Redundanz in Kauf genommen wurde, da der Dienst permanent im Hintergrund ausgeführt wird.

So wird beispielsweise der Zustand des Clients über eine JSON-Datei persistiert, welche ausschliesslich die relevanten Identifikatoren von Patienten und Messungen enthält und auch nur beim Programmstart gelesen wird, also beim start des PCs. Durch diese Massnahmen belegte der Client im stabilen Betrieb einen Arbeitsspeicher im Bereich von rund 140 MB, was in etwa dem Speicherverbrauch eines einzelnen Chrome-Tabs im typischen Anwendungsfall entspricht[22].

Der Client ist derzeit spezifisch auf isokinetische Messungen aus Isocine3 ausgelegt. Die Architektur wurde jedoch so konzipiert, dass eine Erweiterung auf weitere Messarten oder Systeme grundsätzlich möglich ist, unter Verwendung eines modularen, Interface-basierten Designs auf Basis eines Dependency-Injection Containers. Der Zugriff auf Datenquellen erfolgt dabei ausschliesslich über Abstraktionen. Dadurch kann die Synchronisierungslogik unabhängig von konkreten Implementierungen entwickelt und bei Bedarf auf andere Datenquellen erweitert werden. Gleichzeitig ist der Synchronisierungsprozess Client jedoch eng an die spezifischen Eigenschaften und das Verhalten der Isocine3-Software gekoppelt. Bei einer Erweiterung auf andere Messsysteme müsste der Dictionaryvergleich wohl eher weichen, für ein anspruchsvolleres System mit klar definiertem Businessmodell.

Auf die Verwendung konkreter Data Transfer Objects (DTOs) wurde bewusst verzichtet und stattdessen mit generischen String-Dictionaries gearbeitet. Die Datenstruktur von Isocine3 weist mehrere Besonderheiten auf, darunter inkonsistente Feldbezeichnungen, fehlerhafte Übersetzungen sowie installationsspezifische

[22] „Chrome Tabs Using Too Much RAM? Here's What Actually Works", NEST, 25. Februar 2026, https://nestextended.com/blog/chrome-tabs-too-much-ram/.

Anpassungen der Datenbank. Durch die Verwendung von Dictionaries können Felder flexibel über Mapping-tabellen übersetzt und verglichen werden, ohne dass für jede Änderung des Datenmodells Anpassungen an Klassenstrukturen erforderlich sind. Da der Client primär als Synchronisierungskomponente dient und nur einen kleinen Teil der verfügbaren Daten verarbeitet, überwogen in diesem Fall die Vorteile der Flexibilität gegenüber den Vorteilen einer streng typisierten Modellierung.

Das folgende Diagramm spiegelt die Komponenten des Client-Containers wieder:

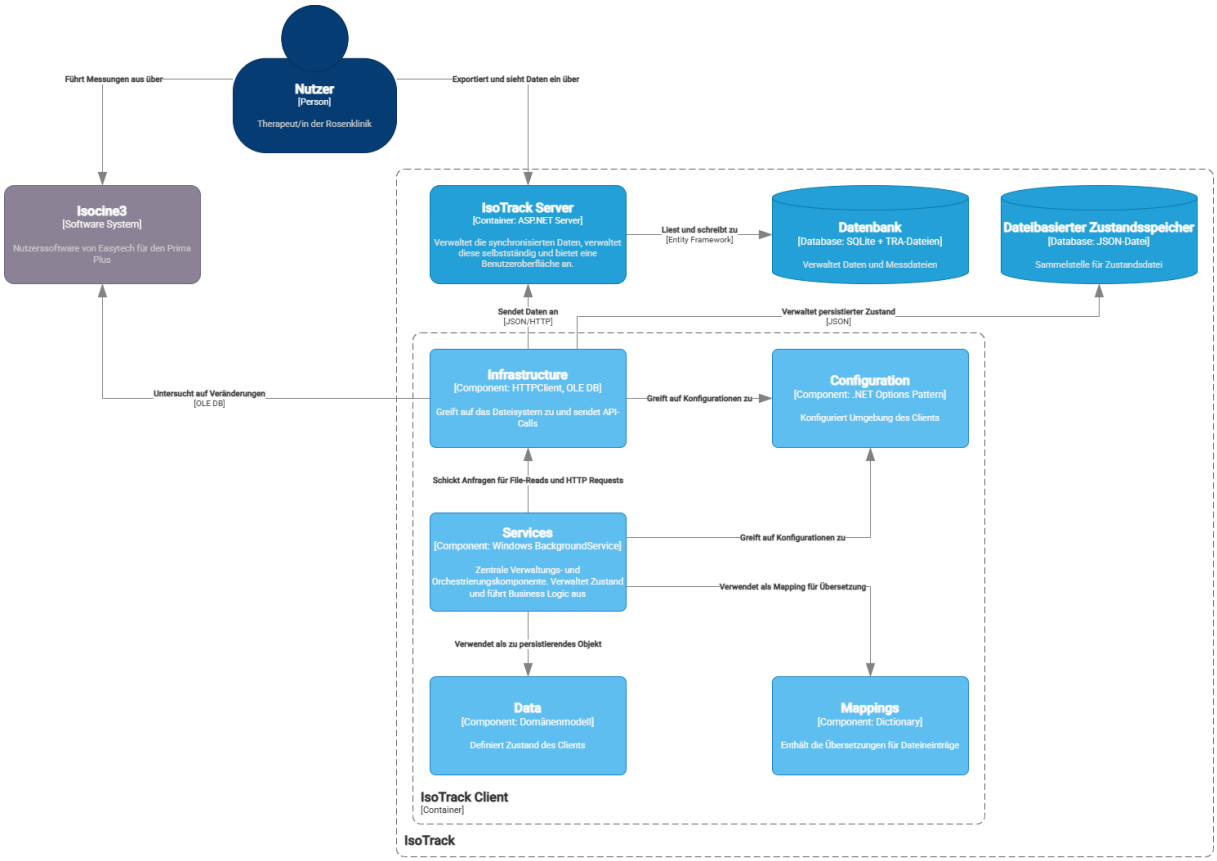


Abbildung 20: C4 Komponentendiagramm des IsoTrack Clients.

Infrastructure

Stellt die Schnittstelle des Clients zur Umgebung dar. Die Komponente enthält den Zugriff auf das Speichersystem von Isocine3, die Kommunikation mit den HTTP-API-Endpunkten des Servers sowie die Persistierung des Zustands.

Configuration

Verwaltet die Initialkonfiguration beim Programmstart, beispielsweise Pfade und Zustand.

Services

Die zentrale Steuerungskomponente des Clients. Sie verwaltet den Zustand, verarbeitet Fehler und initiiert den Synchronisierungsprozess.

Data

Enthält die Datenmodelle des Clients, aktuell primär ein Zustandsobjekt zur Persistierung relevanter Identifikatoren.

Mappings

Enthält die Dictionaries zur Übersetzung von Isocine3-Feldnamen. Dies umfasst insbesondere die Übersetzung von Italienisch nach Englisch sowie die Korrektur fehlerhafter Feldbezeichnungen (siehe Kapitel 3.2.7.1).

2.3.4 Server

Der IsoTrack Server basiert auf dem ASP.NET Core Framework und agiert als Windows Hintergrundservice. Zur Datenpersistierung wird eine SQLite[23] Datenbank verwendet, während parallel die TRA-Dateien als normale Dateien gespeichert werden.

Der Server ist lokal auf den Rechnern der Rosenklinik gemeinsam mit dem Client gehostet, weshalb auf eine HTTPS-Verschlüsselung verzichtet werden kann und stattdessen HTTP verwendet wird. Dies ist dadurch begründet, dass die Infrastruktur der Rosenklinik auf einen lokalen Betrieb mit einem einzelnen isokinetischen Messgerät ausgelegt ist.

Der Server wurde jedoch mit Blick auf eine potenziell verteilte Architektur konzipiert, sodass eine spätere Erweiterung grundsätzlich problemlos möglich ist. In diesem Fall könnten eine HTTPS-Konfiguration mit Zertifikatsverwaltung sowie eine Umstellung der Datenbank auf Microsoft SQL Server[24] oder PostgreSQL[25] erfolgen. Für Das Datenmodell wurde ein Code-First Ansatz mit EntityFramework genutzt. Dies vereinfacht einen potenziellen wechsel des DBMS in der Zukunft und erlaubte es schnell Anpassungen machen zu können während der Entwicklung.

Das Datenmodell ist bewusst pragmatisch und eng an die bestehende Isocine3-Struktur angelehnt. Dadurch werden die Lesbarkeit und die direkte Abbildbarkeit der bestehenden Daten erhöht, allerdings kann dies bei zukünftigen Erweiterungen zu zusätzlichem Anpassungsaufwand führen.

Im aktuellen Setup ist der Server unter `http://localhost:5115/` erreichbar. Die Installation legt zusätzlich eine Desktopverknüpfung zur vereinfachten Nutzung an.

Abbildung 21 stellt die Komponenten des Server dar.

[23] „SQLite Home Page“.
[24] „Microsoft SQL Server“, Microsoft Corporation, zugegriffen 4. Juni 2026, https://www.microsoft.com/de-de/sql-server.
[25] „PostgreSQL“, The PostgreSQL Global Development Group, zugegriffen 26. Mai 2026, https://www.postgresql.org/.

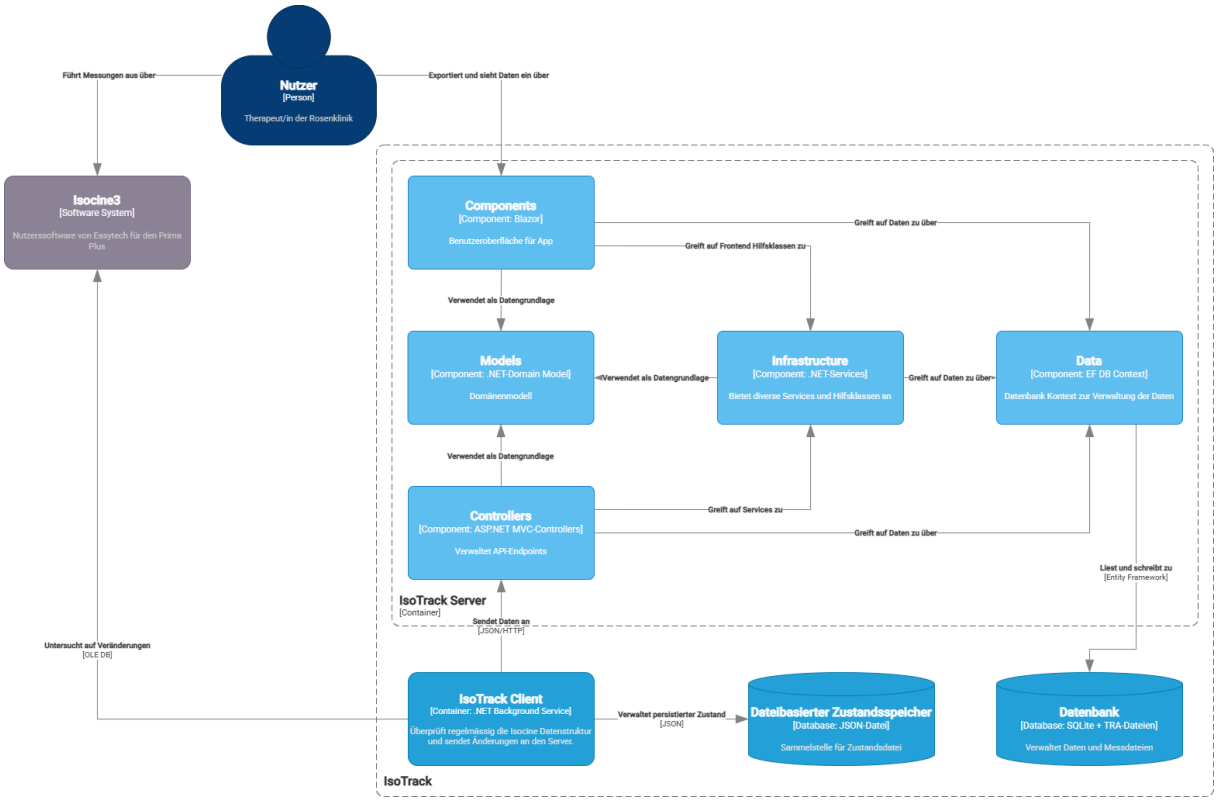


Abbildung 21: C4 Komponentendiagramm des IsoTrack Servers.

Components

Enthält die Blazor-Pages, wiederverwendbare UI-Komponenten sowie Layoutdefinitionen der Benutzeroberfläche.

Die Applikation verwendet ein Hauptlayout „MainLayout“ um das Grundgerüst der Applikation zu bilden. Das beinhaltet zum Beispiel die Navigationsleiste und Provider für Dialog-Overlays. Darin befinet sich ein „@Body“ Objekt das als Platzhalter für die gewählte Seite dient.

Jede Seite der Applikation ist eine eigene Blazor-Komponente die aus MudBlazor-Komponenten aufgebaut ist. Einige Komponenten wurden zudem nochmal separat abgekapselt.

Models

Definiert die zentralen Datenmodelle der Anwendung die von EntityFramework genutzt werden um das Datenbankschema zu generieren. Zudem befinden sind jegliche Datenklassen und Enums ebefalls hier.

Infrastructure

Stellt Services für die Businesslogik bereit, beispielsweise für PDF- und Graphikgenerierung sowie das Dekodieren der TRA-Dateien, und bildet damit die Service-Schicht der Anwendung. Des weiteren befinden sich hier ViewModels die genutzt werden um den Zugriff auf die Datenbank mit Filter zu kapseln.

Data

Definiert den EntityFramework DatabaseContext der in den Blazor-Komponenten und Controllern mittels Dependency-Injection genutzt wird um über die gemappten Objekte auf die Datenbank zuzugreifen.

Controllers

Die Controllers-Komponente verwaltet die API-Endpunkte des Servers, über welche der Client Patienten- und Messdaten als JSON-Objekte synchronisiert.

Die Kommunikation mit dem Server wird über folgende API-Endpunkte geregelt:

Methode	Pfad	Beschreibung
GET	/api/health	Simpler Healthcheck Endpoint, der dem Client erlaubt, die Verfügbarkeit des Servers abzufragen.
POST	/api/patients	Der mitgelieferte Patient wird hinzugefügt, falls er nicht existiert. Ansonsten wird der Patient im Server anhand der Daten im Payload aktualisiert. In beiden Fällen wird der Patient in der Response zurückgegeben.
DELETE	/api/patients/{systemId}/{patientId}	Der zu löschende Patient wird anhand der System- und PatientID aus der Datenbank gelöscht, mitsamt aller Messungen. Gibt nur einen HTTP-Status Code zurück anhand des Erfolgs und Misserfolgs des Requests.
POST	/api/isokinetic-exercises	Erwartet eine Multiform Request aus Messeintrag und TRA-Datei. Fügt den mitgelieferten Messeintrag in der Datenbank ein und legt die TRA Datei im entsprechenden Ordner ab.
DELETE	/api/isokinetic-exercises/{systemId}/{patientId}/{exerciseId}	Ermittelt den zu löschenden Messeintrag mitsamt respektiver TRA-Datei anhand IDs aus Datenbank.

Tabelle 9: API-Endpunkte von IsoTrack.

Bei den Requests wird bewusst nicht zwischen POST- und PUT-/PATCH-Methoden unterschieden, um den Client möglichst schlank zu halten, da ansonsten eine zusätzliche Datenhaltung zur Verwaltung des Synchronisierungsstatus von Patienten und Messungen benötigt würde. Der Server unterscheidet selbstständig anhand der SystemID sowie der Patienten- und ExerciseIDs, ob eine neue Ressource erstellt oder ein bestehender Datensatz aktualisiert werden muss.

Für die Implementierung werden ausschliesslich Open-Source-Bibliotheken eingesetzt, um Lizenzkosten zu vermeiden und die langfristige Wartbarkeit der Anwendung sicherzustellen.

Für die PDF-Generierung kommen PDFsharp[26] und MigraDoc[27] zum Einsatz, während ScottPlot[28] für die Visualisierung der Messkurven innerhalb der PDF-Dokumente verwendet wird.

Der Import und Export von Excel-Dateien erfolgt mittels ClosedXML[29]. Für das Einlesen von CSV-Dateien wird CsvHelper[30] eingesetzt, während der CSV-Export ohne zusätzliche Bibliothek direkt implementiert ist.

[26] „PDFsharp in detail“, empira Software GmbH, zugegriffen 5. Juni 2026, https://www.pdfsharp.com/PDFsharp.
[27] „MigraDoc in detail“, empira Software GmbH, zugegriffen 5. Juni 2026, https://www.pdfsharp.com/MigraDoc.
[28] Scott W Harden, „ScottPlot - Interactive Plotting Library for .NET“, zugegriffen 5. Juni 2026, https://scottplot.net/.
[29] ClosedXML Contributors, „ClosedXML“, GitHub, 4. Juni 2026, https://github.com/closedxml/closedxml.
[30] Josh Close, „CsvHelper“, zugegriffen 5. Juni 2026, https://joshclose.github.io/CsvHelper/.

2.3.5 Architectural Decision Records

Dieses Unterkapitel enthält eine strukturierte Auflistung aller im Verlauf des Projekts getroffenen Entscheidungen. Diese werden in Form von Architectural Decision Records (ADR) im Y-Statements-Schema[31] dokumentiert.

ADR 01		
Datum	26.03.2026	
Kontext	Repository	
Angesichts der Notwendigkeit	zur Definition des Repository Aufbaus	
entschieden wir uns für	ein Monorepo	
und verwarfen	die Idee mehrere Repositories zu erstellen,	
um	eine einfachere Verwaltung der Codebasis	**zu erreichen**
wobei wir	ein aufgeblähteres Repository	**in Kauf nahmen.**

Tabelle 10: ADR 01.

ADR 02		
Datum	27.03.2026	
Kontext	Deployment-Architektur	
Angesichts der Notwendigkeit	zur Definition der Deployment-Architektur	
entschieden wir uns für	eine Aufteilung der Software in eine Client-Server-Architektur	
und verwarfen	die Idee einer kombinierten Variante,	
um	eine mögliche verteilte Ausführung in Zukunft zu ermöglichen	**zu erreichen**
wobei wir	die erhöhte Komplexität bei der Schnittstellendefinition sowie zusätzliche Sicherheitsanforderungen	**in Kauf nahmen.**

Tabelle 11: ADR 02.

[31] Olaf Zimmermann, „Y-Statements“, Medium, 7. Mai 2020, https://medium.com/olzzio/y-statements-10eb07b5a177.

ADR 03		
Datum	27.03.2026	
Kontext	Lizenz	
Angesichts der Notwendigkeit	zur Definition einer Lizenz	
entschieden wir uns dafür	die MIT-Lizenz[32] zu verwenden	
und verwarfen	andere Open-Source-Lizenzen wie GPLv3 oder Apache 2.0	
um	dem Projektpartner nach Abschluss des Projekts die Möglichkeit zu bieten, selbstständig über die Weiterentwicklung der Software zu entscheiden	**zu erreichen**

Tabelle 12: ADR 03.

ADR 04		
Datum	30.03.2026	
Kontext	Deployment-Ort	
Angesichts der Notwendigkeit	zur Definition des Serverstandorts	
entschieden wir uns dafür	den Server lokal auf dem Computer zu deployen, der an das Messgerät angeschlossen ist	
und verwarfen	die Idee einer dedizierten Server-Hardware	
um	ein einfacheres Deployment, das der Skalierung der verfügbaren Messgeräte entspricht,	**zu erreichen**
wobei wir	mögliche Performanceeinbussen	**in Kauf nahmen.**

Tabelle 13: ADR 04.

ADR 05		
Datum	04.04.2026	
Kontext	Tests	
Angesichts der Notwendigkeit	zur Definition des Rahmens des Testkonzepts	
entschieden wir uns dafür	für den Client keine Komponententests zu implementieren	
und verwarfen	die Idee redundante, schon im Integrationstest integrierte, Tests durchzuführen,	
um	Redundanz	**zu verringern**
wobei wir	die Asymmetrik der Komponententests	**in Kauf nahmen.**

Tabelle 14: ADR 05.

[32] „The MIT License“, Open Source Initiative, zugegriffen 27. März 2026, https://opensource.org/license/MIT.

ADR 06		
Datum	19.05.2026	
Kontext	Indikation	
Angesichts der Notwendigkeit	zur Definition der Indikation bei Patienten mit mehreren Indikationen	
entschieden wir uns für	die Erfassung der Indikation pro Messung	
und verwarfen	die separate Erfassung von Patienten anhand ihrer Indikation,	
um	eine geringere Redundanz und eine höhere Genauigkeit der Datenanalyse	**zu erreichen**
wobei wir	einen zusätzlichen Aufwand bei der Messerfassung für den Therapeuten	**in Kauf nahmen.**

Tabelle 15: ADR 06.

2.4 Implementation

Dieses Kapitel beschreibt die Implementierung der funktionalen und nicht-funktionalen Anforderungen gemäss Kapitel 2.1.

2.4.1 Synchronisierungsprozess

Der Synchronisierungsprozess wird durch den Client initiiert. In einem Intervall von 20 Sekunden überprüft dieser die Verfügbarkeit des Servers, sofern die Access-Datenbank von Isocine3 seit der letzten Prüfung verändert wurde. Dieses Intervall hat sich durch heuristische Tests als geeigneter Kompromiss zwischen Reaktionszeit und Ressourcenverbrauch erwiesen.

Anschliessend werden sämtliche Patienten, isokinetischen Messungen sowie TRA-Dateien aus Isocine3 extrahiert. Dabei wird jedem Datensatz zusätzlich eine SystemID zugewiesen, welche eine eindeutige Identifikationszeichenfolge des Clients darstellt. Der Server verwendet diese zur Zuordnung eingehender Daten und zur Unterscheidung mehrerer Datenquellen, wodurch eine zukünftige verteilte Architektur grundsätzlich unterstützt werden kann.

Beim ersten Start des Clients nach der Installation erfolgt eine vollständige Synchronisierung aller Patienten und isokinetischen Messeinträge.

Der Lösch- bzw. Hinzufügungsstatus eines Patienten wird über ein Dictionary verwaltet, welches jede Patientennummer mitsamt den zugehörigen Messnummern speichert. Während des Synchronisierungsprozesses wird der aktuelle Zustand der Isocine3-Datenbank als Soll-Zustand betrachtet und mit dem lokal persistierten Ist-Zustand verglichen. Basierend auf diesem Vergleich werden neu hinzugefügte oder gelöschte Datensätze erkannt und über POST- (Hinzufügen) beziehungsweise DELETE-Requests (Löschen) synchronisiert. Das Dictionary wird als Zustand in einer JSON-Datei persistiert.

Da Patientendaten in Isocine3 veränderbar sind, wird ein Mechanismus zur Änderungserkennung benötigt. Hierzu werden die veränderbaren Eigenschaften eines Patienten zu einem Hashwert zusammengeführt, wodurch Änderungen zuverlässig erkannt werden können. Die Kollisionswahrscheinlichkeit des verwendeten SHA-256-Verfahrens gilt aufgrund des grossen Ausgaberaums (2^{256} mögliche Werte) als praktisch vernachlässigbar. Im Fehlerfall führt dies zu einer redundanten, jedoch konsistenzsicheren Synchronisierung. Auf die Persistierung der Hashwerte wird bewusst verzichtet, da dies insbesondere bei grösseren Datenbeständen zu einem unverhältnismässigen Wachstum der JSON-Datei führen würde. Stattdessen werden die Hashwerte beim Start des Clients neu berechnet, wodurch nach einem Neustart sämtliche Patienten erneut synchronisiert werden. Da dieser Vorgang in der Regel nur einmal täglich auftritt, wird die daraus resultierende Redundanz bewusst in Kauf genommen.

Abbildung 22 zeigt die Zustandsübergänge eines Patienten innerhalb des Synchronisierungsprozesses.

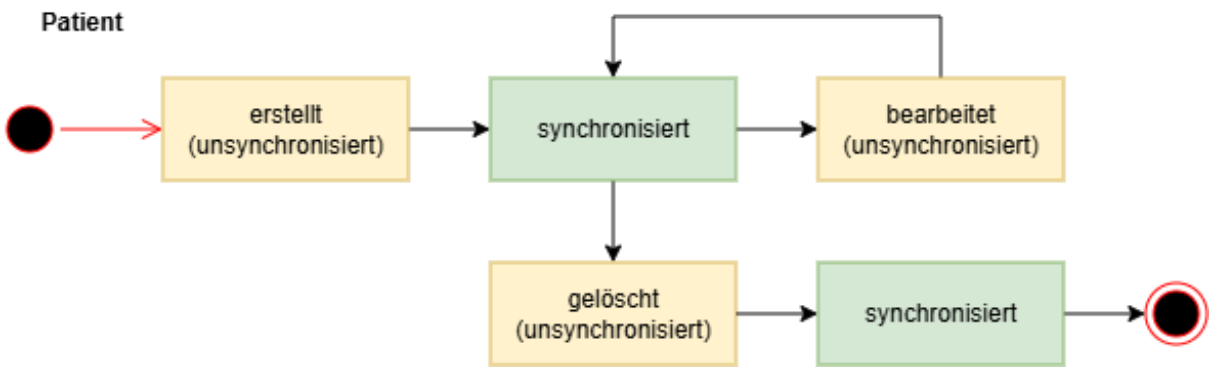


Abbildung 22: Zustandsübergangsdiagramm eines Patienten.

Für isokinetische Messungen genügt der Vergleich mit dem Dictionary, da diese unveränderlich sind. Zusätzlich werden die zugehörigen TRA-Dateien, welche jeweils einem isokinetischen Messeintrag entsprechen, gemeinsam mit dem Datensatz als Einheit an den Server übertragen.

Abbildung 23 zeigt den vereinfachten Zustandsübergang einer isokinetischen Messung.

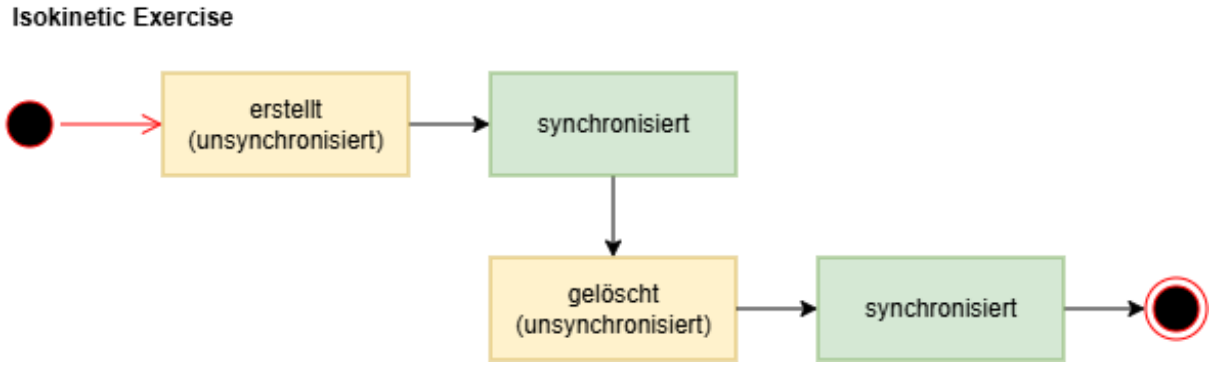


Abbildung 23: Zustandsübergangsdiagramm einer isokinetischen Messung.

Der Synchronisierungsprozess wurde bewusst idempotent ausgelegt. Mehrfach gesendete Datensätze führen dadurch zu keinen Inkonsistenzen, da der Server anhand der SystemID sowie Patienten- und Messidentifikatoren selbstständig zwischen Neuanlage und Aktualisierung unterscheidet.

In Isocine3 werden Felder geführt, beispielsweise sämtliche Adressdaten der Patientendaten, die keinen Einfluss auf die Funktionalität von IsoTrack haben. Diese Daten werden aus Gründen der Vollständigkeit dennoch übernommen und in IsoTrack persistiert, sind dort jedoch für den Nutzer nicht sichtbar.

2.4.2 Benutzeroberfläche

Die Benutzeroberfläche besteht aus vier Seiten plus einer zusätzlichen Einstellungsseite, die über eine Navigationsleiste auf der linken Seite erreicht werden kann. Es wurden praktisch ausschliesslich MudBlazor Komponenten verwendet, um ohne grossen Aufwand ein einheitliches Design und gleichzeitig ein responsives Layout zu erhalten. Die einzige Ausnahme bildet ApexCharts, da die Chart-Komponenten von MudBlazor einen zu beschränkten Funktionsumfang haben.

2.4.2.1 Neuste Messungen

Auf der Startseite „Neueste Messungen“ werden beim Öffnen der Anwendung direkt die zuletzt synchronisierten Messungen angezeigt. Dies ist eine bewusste Entscheidung, um den Therapeuten die Navigation zum Patientenprofil zu ersparen und die Resultate schneller dem Patienten vorführen zu können. Hier können die Messungen direkt als Kurven in Charts dargestellt werden.

Die Seite besteht in erster Linie aus zwei Komponenten. Oben eine Auswahl an Filtern und unten eine Tabelle von Messungen. Neben dem schnellen Zugriff auf neue Messungen dient die Seite auch dazu, die Messdaten als CSV- oder XLSX-Datei zu exportieren. Um mit einer bestimmten Teilmenge der Daten zu interagieren, steht dem Nutzer eine Auswahl an Filtern zur Verfügung:

- Geschlecht
- Nur Messungen von der betroffenen Seite
- Indikation
- Anz. Wiederholungen
- Alter (zum Zeitpunkt der Messung)

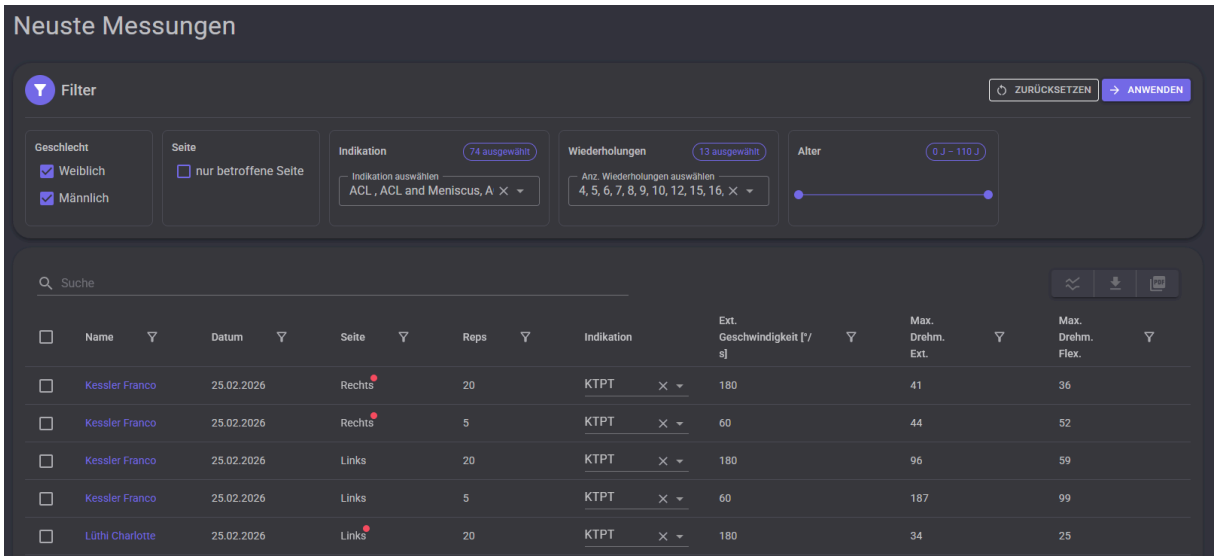


Abbildung 24: Screenshot Startseite "Neuste Messungen".

Die untere Tabelle bietet neben den wichtigsten Kennzahlen der Messungen einige Möglichkeiten, um mit den Daten zu interagieren (siehe Abbildung 25).

1. Suchfeld für den Namen/Vornamen eines Patienten.
2. Checkboxen, um Messungen an- und abzuwählen, um dann mit (6) diese weiterzuverarbeiten.
3. Link auf Patientenprofil.
4. Seite der Messung. Der rote Punkt signalisiert, dass diese Seite operiert wird/wurde (kurz die betroffene Seite).
5. Dropdown, um Indikation zu ändern, falls sie von der Indikation im Patientenprofil abweichen sollte.
6. Möglichkeiten, um die Daten weiterzuverarbeiten, v.l.n.r.
   1. Messwerte anzeigen.
   2. Als CSV- oder XLSX-Datei herunterladen.
   3. PDF-Bericht generieren

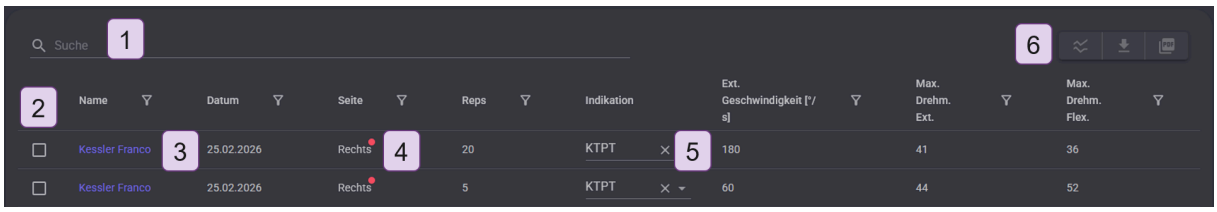


Abbildung 25: Tabelle der Messungen auf der Startseite.

2.4.2.1.1 Messwerte

Für die Visualisierung der Messwerte dient ein Diagramm (siehe Abbildung 26) mit dem Drehmoment auf der Y-Achse und der Zeit in Sekunden auf der X-Achse (2). Zudem werden Referenzwerte angezeigt (3). Diese zeigen, welche Kraft eine Person mit dem entsprechenden Geschlecht, Alter und Gewicht etwa erreichen sollte. Es können zwei verschiedene Arten von Referenzwerten eingepflegt werden, absolute (Nm) und gewichtsbasierte (Nm/kg Körpergewicht). Um zwischen den beiden Werten wechseln zu können, dient eine Schaltfläche oben am Diagramm (1).

Unterhalb des Diagramms findet man für jede der angezeigten Messungen eine Übersicht der wichtigsten Kennzahlen (4). Hier wird ebenfalls der rote Punkt verwendet, um die betroffene Seite des Patienten zu markieren.

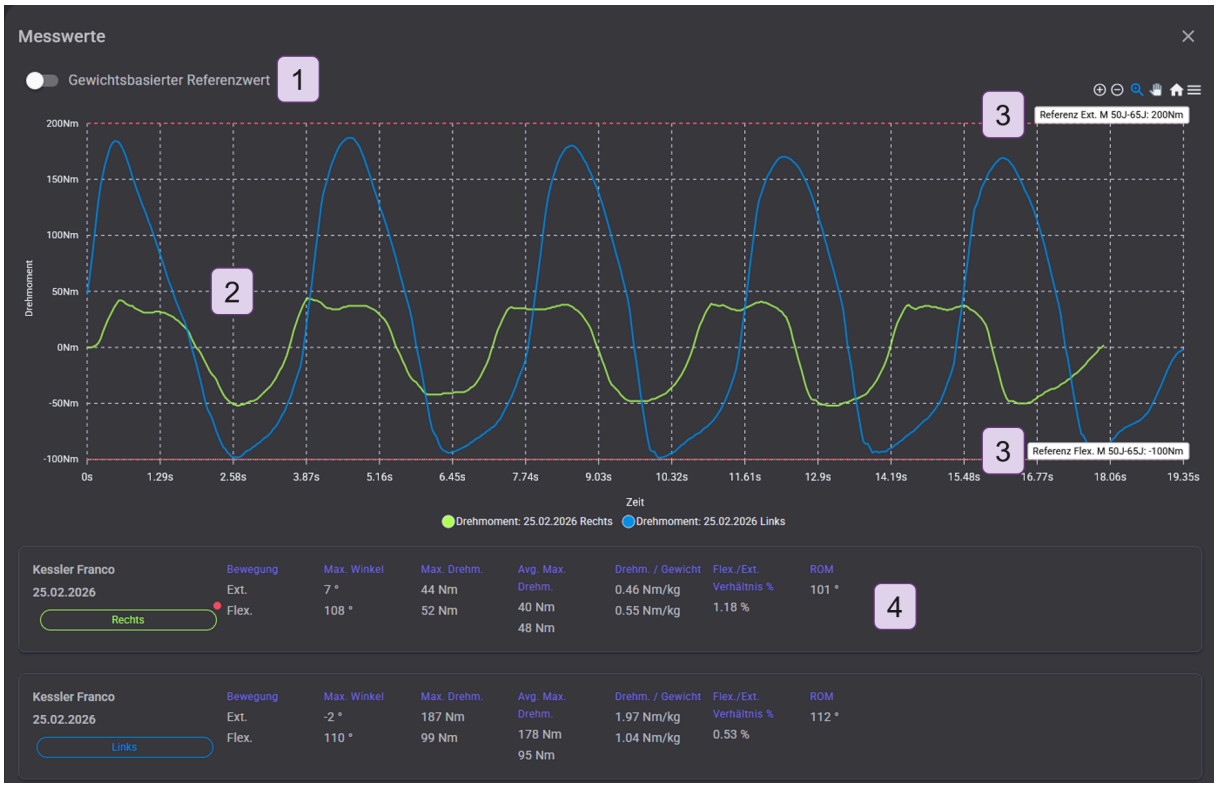
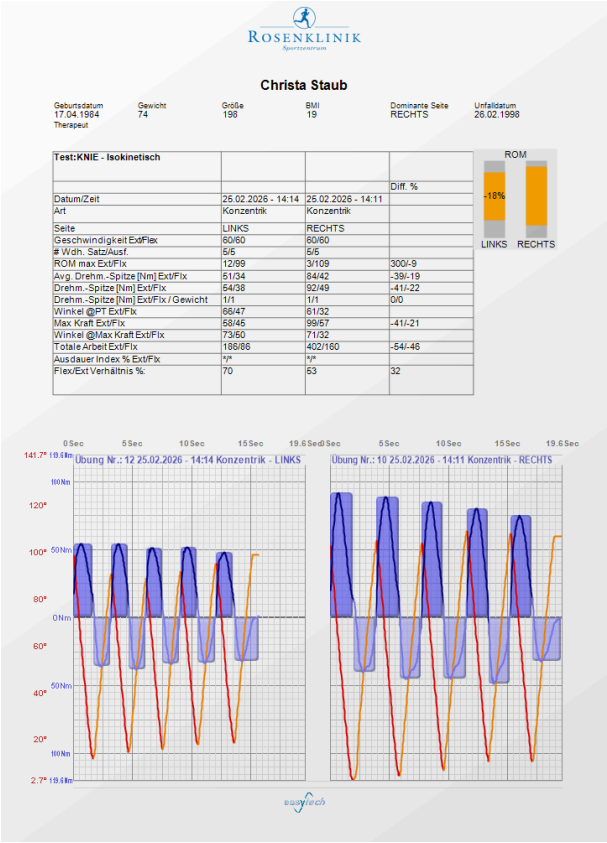


Abbildung 26: Visualisierung der Messwerte zweier Messungen.

2.4.2.1.2 PDF-Bericht

Das Ziel des PDF-Berichts ist es, eine schnelle Übersicht über den Stand eines Patienten zu bekommen. Diese Berichte können an weitere medizinische Institutionen weitergegeben werden, die keinen Zugriff auf die IsoTrack-Anwendung haben, z. B. den Hausarzt oder den Chirurgen des Patienten.

Es wurde eine konkrete Vorlage seitens der Rosenklinik definiert, basierend auf dem bestehenden Isocine3-PDF-Bericht (siehe Abbildung 27) sowie gängigen klinischen Standards, die entsprechend umgesetzt wurde.



ROSENKLINIK
Sportzentrum

Christa Staub

Geburtsdatum	Gewicht	Größe	BMI	Dominante Seite	Unfalldatum
17.04.1984	74	198	19	RECHTS	26.02.1998
Therapeut					

Test:KNIE - Isokinetisch			
			Diff. %
Datum/Zeit	25.02.2026 - 14:14	25.02.2026 - 14:11	
Art	Konzentrik	Konzentrik	
Seite	LINKS	RECHTS	
Geschwindigkeit Ext/Flex	60/60	60/60	
# Wdh. Satz/Ausf.	5/5	5/5	
ROM max Ext/Flx	12/99	3/109	300/-9
Avg. Drehm.-Spitze [Nm] Ext/Flx	51/34	84/42	-39/-19
Drehm.-Spitze [Nm] Ext/Flx	54/38	92/49	-41/-22
Drehm.-Spitze [Nm] Ext/Flx / Gewicht	1/1	1/1	0/0
Winkel @PT Ext/Flx	66/47	61/32	
Max Kraft Ext/Flx	58/45	99/67	-41/-21
Winkel @Max Kraft Ext/Flx	73/50	71/32	
Totale Arbeit Ext/Flx	186/86	402/160	-54/-46
Ausdauer Index % Ext/Flx	*/*	*/*	
Flex/Ext Verhältnis %:	70	53	32

Abbildung 27: Screenshot eines Isocine3 PDF-Berichts.

Die Schaltfläche für den PDF-Bericht ist nur aktiv, wenn der Nutzer zwei Messungen desselben Patienten ausgewählt hat, eine für die linke und eine für die rechte Seite.

Kessler Franco

Grösse	Gewicht	BMI	Betroffene Seite	Unfalldatum
184 cm	95 kg	28	Rechts	01.01.1901

Knie - Isokinetisch, kon/kon, 60/60, 5 Wdh

	Extension		Flexion		Ratio
	Wert	% Körpergewicht	Wert	% Körpergewicht	Flexion/Extension
Spitzendrehmoment (Nm)					
Rechts	44	46	52	55	118
Links	187	197	99	104	53
Referenzwert	200	220	100	110	
Seitendifferenz	24		53		
ROM (°)					
Rechts	7		108		
Links	-2		110		
Seitendifferenz	90		90		
Winkel bei Maximaldrehmoment (°)					
Rechts	90		54		
Links	68		28		

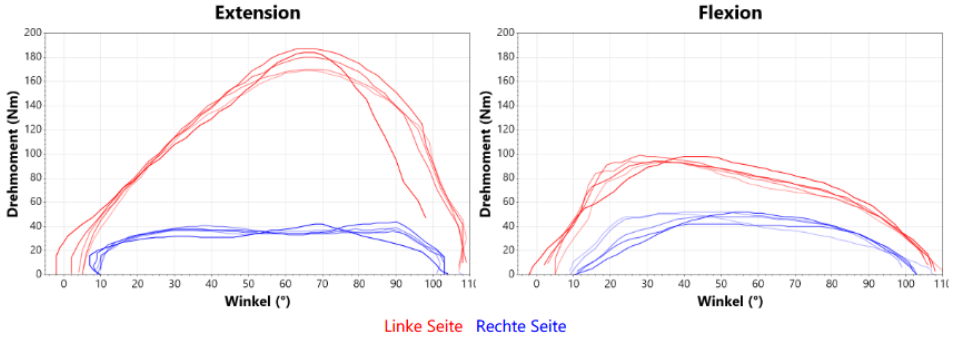


Abbildung 28: Screenshot PDF-Bericht.

Die in den beiden unteren Diagrammen dargestellten Messwerte entsprechen denjenigen des überlappenden Messwertdiagramms in Isocine3 (siehe Abbildung 29). Im Gegensatz zur Darstellung in Isocine3 werden die Werte jedoch nach Extension und Flexion getrennt ausgewiesen, was einen direkten Vergleich der beiden Seiten in den jeweiligen Kraftphasen ermöglicht. Zusätzlich wird im Vergleich zu den regulären Messwertdiagrammen von IsoTrack die Zeit auf der X-Achse durch den Winkel ersetzt.

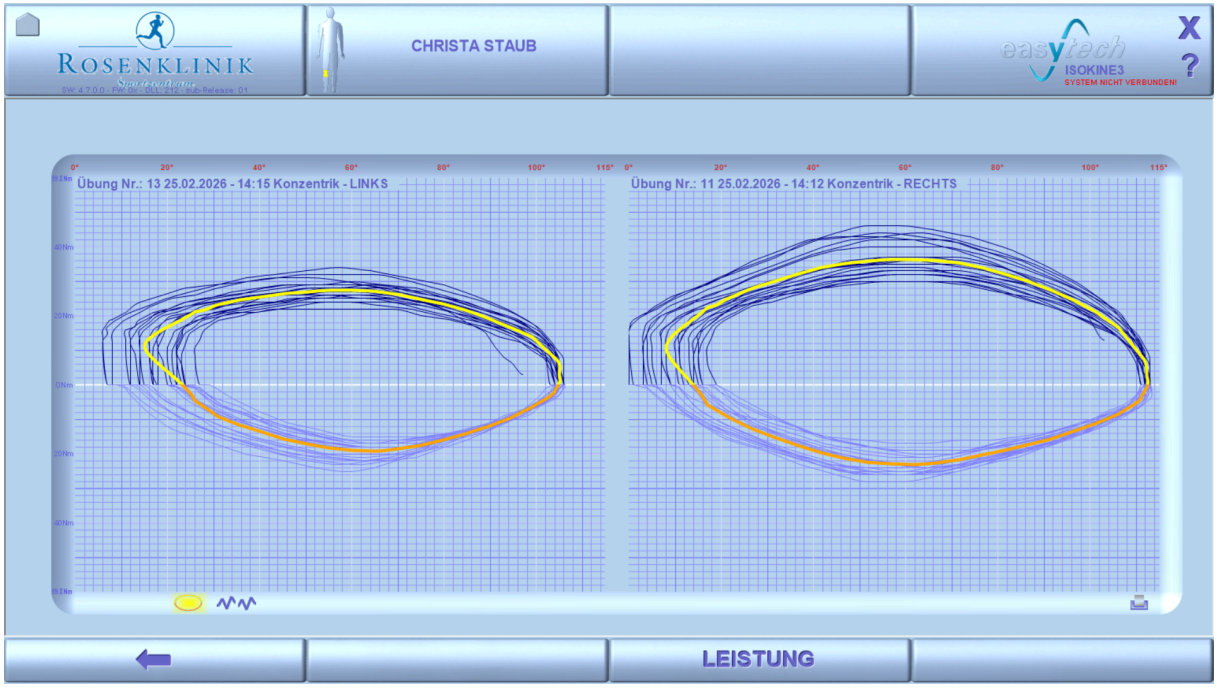


Abbildung 29: Messgraphen zweier isokinetischer Messungen in Isocine3 mit überlappenden Linien.

Für die Diagnose werden, wie bereits etabliert, primär der Spitzendrehmoment jeweils für Extension und Flexion herangezogen. Die Referenzwerte werden dabei direkt zum Vergleich unterhalb im Bericht angegeben.

Zusätzliche Informationen, die der Bericht bereitstellt, sind das Flexion/Extension-Verhältnis pro Seite sowie die Seitendifferenz.

Das Flexion/Extension-Verhältnis entspricht dem Quotienten aus Flexions- und Extensionsdrehmoment und ermöglicht einen direkten Vergleich zwischen Kniebeuger- und Kniestreckermuskulatur. Die Interpretation dieses Werts erfolgt durch den Therapeuten, der auf Basis klinischer Erfahrung beurteilt, welches Verhältnis als physiologisch einzustufen ist.

Die Seitendifferenz beschreibt das Verhältnis der beiden Beine, wobei das betroffene Bein als Referenz (Nenner) verwendet wird. Der Bericht bietet hierfür eine visuelle Einteilung: Werte unter 70 % werden rot dargestellt, Werte bis 90 % gelb und Werte ab 90 % grün. Dadurch erhält der Nutzer ein unmittelbares visuelles Feedback.

Das betroffene Bein wird, sofern vorhanden, anhand der Operationsdaten bestimmt. Falls diese nicht verfügbar sind, wird standardmässig die nicht-dominante Seite verwendet.

Der Bewegungsbereich (ROM) ergibt sich aus der Differenz zwischen den maximal erreichten Winkeln der jeweiligen Bewegung. Die Seitendifferenz bezieht sich somit ausschliesslich auf den Seitenvergleich und nicht auf Extension und Flexion.

2.4.2.2 Patienten

Das Patientenprofil erlaubt es den Therapeuten, alle Informationen zu einem Patienten auf einen Blick einzusehen können.

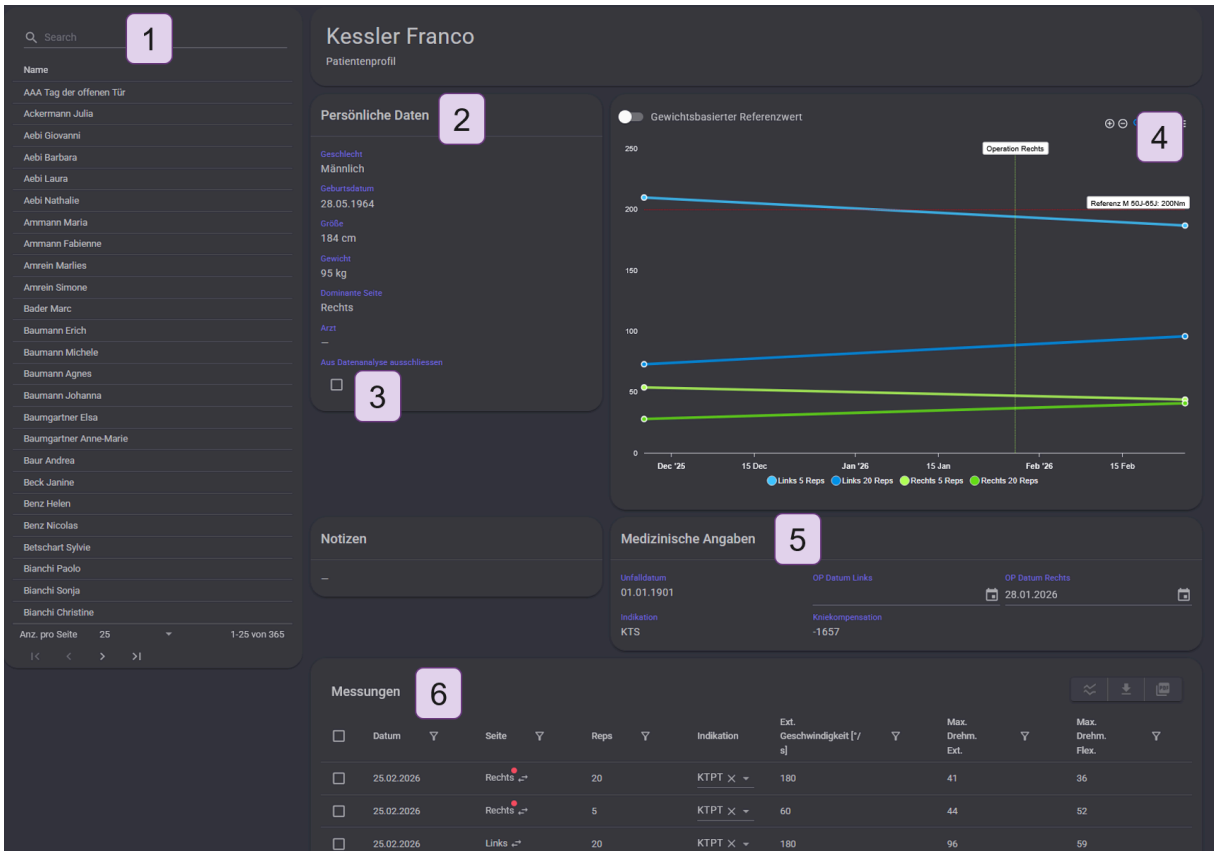


Abbildung 30: Screenshot Patientenprofil.

Die einzelnen Bereiche des Patientenprofils sind im Screenshot nummeriert und werden nachfolgend entsprechend dieser Nummerierung erläutert.

1. Liste aller Patienten mit Suchfeld
2. Stammdaten des Patienten
3. Checkbox, um Patienten aus Datenanalyse auszuschliessen. Zum Beispiel bei Test-Patienten oder den Patientenprofilen der Therapeuten.
4. Trenddiagramm, das jeweils das maximale Extensionsdrehmoment über die Zeitachse aller Messungen, meist mehrere Monate, des Patienten darstellt. So kann schnell ein Überblick über den Genesungsprozess gewonnen werden. Die Messungen sind nach Anzahl der Wiederholungen (Reps) und Seite gruppiert. Zudem wird auch hier der entsprechende Referenzwert mit einer roten horizontalen Linie eingeblendet und ebenfalls die Auswahl zwischen dem absoluten und dem gewichtsbasierten Referenzwert. Die vertikale Linie, die eingeblendet wird, stellt den Zeitpunkt der Operation dar.
5. Medizinische Angaben des Patienten. Hier lassen sich auch die Operationsdaten pro Seite eintragen, die für die Markierung im Trenddiagramm, den roten Punkt in den Tabellen sowie dann später in der Datenanalyse relevant sind.
6. Tabelle aller Messungen des Patienten, funktional weitgehend identisch mit der Tabelle auf der Startseite, jedoch mit der zusätzlichen Möglichkeit, die Seite einer Messung zu ändern. Es kommt vor, dass bei der Bedienung des Easytech Prima Plus die Therapeuten, die falsche Seite angeben, wobei Isocine3 keine Möglichkeit bietet, dies im Nachhinein zu korrigieren.

2.4.2.3 Datenqualität

Während der Entwicklung ist vermehrt aufgefallen, dass viele Stammdaten im Isocine3 relativ lückenhaft und inkonsistent gepflegt worden sind. Damit die Rosenklinik ihre Daten effektiv für wissenschaftliche oder statistische Zwecke nutzen kann, müssen diese Stammdaten korrigiert werden. Hierfür wurde der

Anwendung eine zusätzliche Seite hinzugefügt, um Patienten und Messungen mit mangelhaften Stammdaten zu erkennen und das gezielte Korrigieren im Isocine3 zu erleichtern.

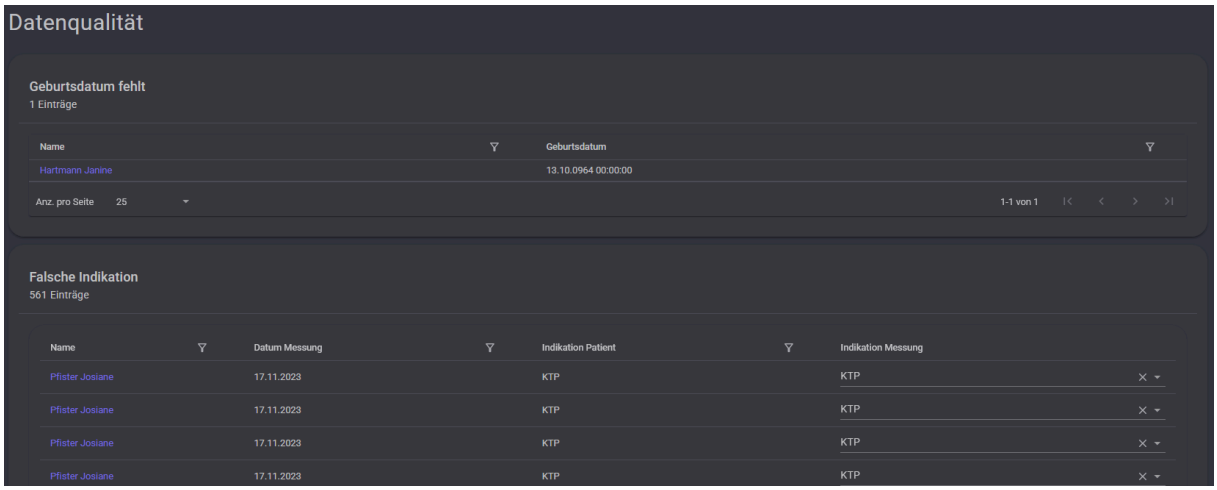


Abbildung 31: Screenshot Datenqualität.

Folgende Stammdaten werden geprüft:

- Geburtsdatum: Ist ein Geburtsdatum eingetragen und ist es plausibel?
- Indikation: Entspricht die Indikation der Messung einer der in den Einstellungen (siehe Kapitel 2.4.2.5) definierten Indikationen.
- Unfalldatum: Ist ein Unfalldatum eingetragen und ist es plausibel?
- Gewicht: Ist ein Gewicht eingetragen und ist es plausibel?
- Grösse: Ist eine Grösse eingetragen und ist sie plausibel?

Wenn bei einem Patienten das Häkchen „Aus Datenanalyse ausschliessen“ im Patientenprofil angewählt wurde, wird er hier nicht angezeigt.

2.4.2.4 Datenanalyse

Die Datenanalyseseite erlaubt es den Therapeuten, erste Datenanalysen durchzuführen sowie einen Überblick über den Umfang ihrer Messdaten zu kriegen. Sie basiert auf einer Auswahl an Filtern sowie einem Pivot-Diagramm.

Abbildung 32: Screenshot Datenanalyse.

Die einzelnen Elemente der Datenanalyseseite sind im Screenshot (siehe Abbildung 32) nummeriert und werden im Folgenden entsprechend dieser Nummerierung erläutert.

1. Filter
   - Geschlecht
   - Unfalldatum ≠ leer
     - Falls dieser Filter gesetzt wird, kann zusätzlich noch nach dem Alter beim Unfall gefiltert werden.

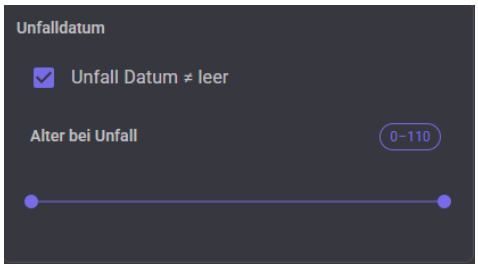


Abbildung 33: Filter wenn "Unfall Datum ≠ leer" ausgewählt ist.

   - OP-Datum ≠ leer
     - Falls dieser Filter gesetzt wird, kann zusätzlich noch nach dem Alter bei der Operation sowie nach Messungen auf der betroffenen Seite gefiltert werden.

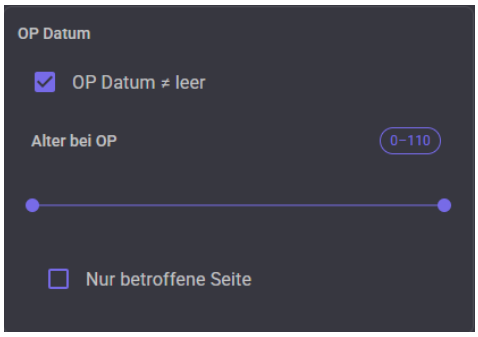


Abbildung 34: Filter wenn "OP Datum ≠ leer" ausgewählt ist.

- Alter, bei der Messung
- Gewicht
- Grösse
- Indikation
- Anz. Wiederholungen

Um die gefilterten Daten anschliessend darzustellen, hat der Nutzer folgende Optionen:

2. Chart Typ
   - Bar (Balkendiagramm)
   - Scatter (Streudiagramm)
3. X-Achse

   Eine Auswahl von Patienten sowie Messmerkmalen. Zu den Feldern, die ohnehin schon im Datenmodell vorhanden sind, werden während der Runtime zusätzliche Felder generiert wie:
   - Alter bei Unfall (als Zahl und gruppiert in 10-Jahresschritten)
   - Alter bei OP (als Zahl und gruppiert in 10-Jahresschritten)
   - Betroffene Seite
   - Tage zur Operation (als Zahl und gruppiert nach „Vor OP“, „5-“, „12-“ und „24-Wochen nach OP“) entsprechend dem Vorgehen der Rosenklinik vor und nach Operationen.

   Wenn das Streudiagramm ausgewählt wird, wird die Auswahl der X-Achsenmerkmale eingeschränkt, da das Streudiagramm lediglich numerische X-Achsen unterstützt.
4. Y-Achse

   Eine Auswahl von Messkennzahlen sowie Grösse und Gewicht des Patienten.
5. Aggregation
   - Durchschnitt
   - Min
   - Max
   - Anzahl, kann nützlich sein, um Lücken in Messdaten zu erkennen, bevor man falsche Schlüsse aus den Daten zieht.
6. Gruppierung

   Die gleichen Felder wie schon bei der X-Achse, jedoch werden diese nicht eingeschränkt bei der Auswahl vom Streudiagramm.

Das Pivot-Diagramm basiert auf einer generischen Analysearchitektur, die Rohdaten aus den Messungen zunächst in ein vereinheitlichtes Analysemodell transformiert und anschliessend dynamisch aggregiert und visualisiert.

Zentral dafür ist die Abstraktion der „PivotChartProperty“, welche jede analysierbare Eigenschaft, beispielsweise Alter, Geschlecht, Indikation oder Drehmomentwerte, über generische Accessor-Funktionen beschreibt. Dadurch muss die Diagramm-Logik keine konkreten Domänenfelder kennen, sondern kann beliebige Eigenschaften der X-Achse, Y-Achse oder Seriengruppierung zuordnen.

Die eigentliche Pivot-Funktionalität entsteht durch eine dynamische Projektion der Daten, gefolgt von einer Gruppierung nach den ausgewählten Dimensionen und einer frei wählbaren Aggregation.

Zusätzliche Abstraktionen wie „PivotSortKey“ ermöglichen eine korrekte Sortierung von numerischen, textuellen oder kategorisierten Werten (z. B. Altersgruppen oder Operationsdaten), während das Analysemodell bereits berechnete Kennzahlen wie Alter bei Operation, Alter bei Trauma oder Zeit seit der Operation bereitstellt. Durch diese Trennung von Datenaufbereitung, Metadatenbeschreibung und Visualisierung kann das Pivot-Diagramm flexibel erweitert werden, ohne die eigentliche Diagramm-Implementierung anpassen zu müssen.

2.4.2.5 Einstellungen

Auf der Einstellungsseite kann man den IsoTrack Server konfigurieren.

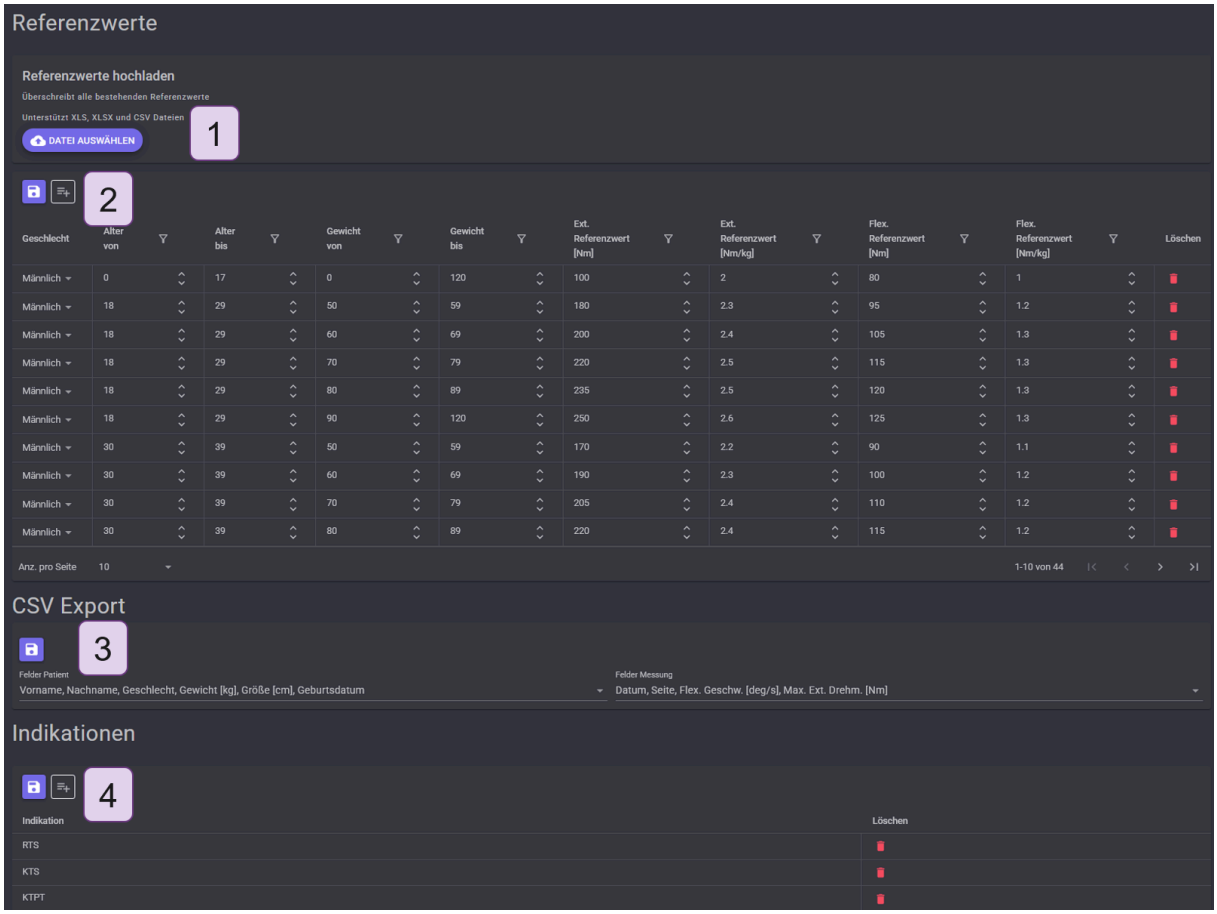


Abbildung 35: Screenshot Einstellungen.

Die Einstellungsseite ist in drei Einstellungen aufgeteilt.

Die im Screenshot dargestellten Elemente der Einstellungsseite sind nummeriert und werden nachfolgend erläutert:

1. Referenzwerte hochladen

   Hier können die Therapeuten die Referenzwerte eintragen, die danach bei der Visualisierung der Messwerte sowie im Trend-Diagramm rot angezeigt werden. Dafür können sie entweder eine XLSX- oder eine CSV-Datei hochladen. Das Hochladen von Referenzwerten mittels einer Datei überschreibt immer alle bestehenden Referenzwerte.

2. Referenzwerte manuell bearbeiten

   Für gezielte Anpassungen oder das Eintragen einzelner Messwerte gibt es hier eine Tabelle, die auch direkt in der Applikation editiert werden kann.

3. CSV-Export

   Damit im CSV-/Excel-Export nicht jedes Mal alle gesuchten Spalten einzeln ausgewählt werden müssen, kann man hier die Felder auswählen, die man im Export haben möchte.

4. Indikationen

   Um die Vereinheitlichung und Pflege der Daten zu vereinfachen, kann man hier „korrekte“ Indikationen angeben, also Indikationen, die man effektiv nutzen möchte. Messungen und Patienten, die eine Indikation haben, die nicht in dieser Liste aufgeführt ist, werden auf der Datenqualitätsseite als inkorrekt angezeigt.
   Wenn man auf einer Messung die Indikation editiert, erscheint jeweils eine Dropdown-Liste mit der Auswahl an Indikationen, die hier eingetragen ist.

2.4.2.6 Logo

Für das Favicon, die Desktopverknüpfung sowie die Benutzeroberfläche wurde ein einheitliches Logo entworfen.

Zunächst wurde ein einfacher Platzhalter in Inkscape erstellt (siehe Abbildung 36).

Abbildung 36: Initiales Platzhalter-Logo von IsoTrack.

Im weiteren Verlauf der Construction-Phase wurde ein finales Logo entworfen. Dabei wurde zur Unterstützung der Ideenfindung ein generatives Bildmodell eingesetzt und anschliessend eine finale Version basierend auf diesen Entwürfen erstellt.

Die Farbpalette basiert auf den Farbvorgaben der OST.

(a) (b)

Abbildung 37: Finales IsoTrack-Logo mit und ohne Schriftzug

2.5 Qualitätssicherung

Während nicht funktionale Anforderungen die Qualität des Produkts definieren, müssen zusätzlich Massnahmen spezifiziert werden, welche die Qualität des gesamten Projektablaufs sicherstellen, mit dem Ziel, diese auf das Endprodukt zu übertragen.

Dieses Kapitel beschreibt die Massnahmen, Richtlinien, Tools und Konzepte, welche den Entwicklungsprozess unterstützen und dessen Qualität gewährleisten.

2.5.1 Arbeitsumgebung

Dieses Unterkapitel regelt bestimmte Aspekte der Arbeitsumgebung des Teams, soweit dies erforderlich ist, und definiert Rahmenbedingungen, um Konsistenz sowie eine effiziente Zusammenarbeit zu gewährleisten.

2.5.1.1 Definition of Ready

User Stories jeglicher Art benötigen Kriterien, welche den Status kategorisieren. Im Falle einer Definition of Ready wird die Bereitschaft, dem Backlog hinzugefügt werden zu können, definiert.
Für Scrum-Projekte üblich sind hier die **INVEST**[33] Kriterien:

- **I**ndependent: Die User Story ist unabhängig von anderen Stories formuliert und umsetzbar.
- **N**egotiable: Keine konkreten Implementierungsdetails werden definiert.
- **V**aluable: Es wird ein erkennbarer Nutzen für Stakeholder gestiftet.
- **E**stimable: Der Aufwand kann realistisch geschätzt werden und ermöglicht eine verlässliche Planung.
- **S**mall: Der Umfang ist so begrenzt, dass die Umsetzung innerhalb eines Sprints möglich ist.
- **T**estable: Sie ist klar spezifiziert und damit überprüf- und testbar.

2.5.1.2 Definition of Done

Damit eine Aufgabe als abgeschlossen eingestuft werden kann, muss sie eine Reihe von Kriterien erfüllen. Diese Kriterien dienen dazu, eine konsistente Qualität sicherzustellen und die Aufgabe formell in den Projektablauf zu integrieren.

Eine Aufgabe gilt als abgeschlossen, wenn folgende Bedingungen erfüllt sind:

- Eine Codeabdeckung durch Tests von mindestens 75 Prozent wurde erreicht.
- Alle Unittests wurden erfolgreich bestanden.
- Ein Code Review wurde durchgeführt.
- Die Dokumentation wurde angepasst beziehungsweise erweitert.
- Die Pipeline ist erfolgreich durchgelaufen.
- Eine Demonstration für den Product Owner wurde vorbereitet.
- Die Freigabe durch den Product Owner liegt vor.

2.5.1.3 Dokumentationsrichtlinien

Zur Sicherstellung einer konsistenten und qualitativ konstanten Dokumentation, werden Richtlinien und Standards festgelegt.

- Jedes Teammitglied ist dafür verantwortlich, die eigene Arbeit vollständig und nachvollziehbar zu dokumentieren.
- Für Überschriften bis einschliesslich Level 2 ist zwingend ein Label zu vergeben.
- Für Überschriften ist folgendes Template zu verwenden:

[33] Bill Wake, „INVEST in Good Stories, and SMART Tasks“, XP123, 17. August 2003, https://xp123.com/invest-in-good-stories-and-smart-tasks/.

```
=== Kapitel Titel <kapitelTitel>

Text
```

- Querverweise sind ausschliesslich über Labels zu realisieren.
- Tabellen und Abbildungen müssen stets mit einer aussagekräftigen Bildunterschrift versehen werden.
- Die Ressourcenverwaltung erfolgt über das Verzeichnis `resources/` sowie dessen Unterordner.
- Angegebene Quellen sind vollständig und korrekt zu erfassen und über den Referenztag `@reference` zu zitieren. Ein korrekt erfasster Eintrag umfasst mindestens **Titel**, **Autor**, **Publisher**, **Dokumenttyp** und **Veröffentlichungsdatum**. Bei Online-Ressourcen sind zusätzlich **URL** und **Zugriffsdatum** anzugeben.
- Gemäss dem Leitfaden zum Umgang mit KI bei schriftlichen Arbeiten des Departments Informatik an der OST ist der Einsatz von KI grundsätzlich zulässig. Im vorliegenden Projekt wird dieser jedoch auf Orthographie- und Formatierungsunterstützung beschränkt.
- Zur Sicherstellung konsistenter Übersetzungen wird ausschliesslich DeepL verwendet.
- Bei Erweiterungen der Dokumentation ist stets das grundlegende Template einzuhalten und anzuwenden.

2.5.1.4 Coding-Richtlinien

Um Konsistenz und die Zusammenarbeit im Team zu fördern, werden Richtlinien für die Coding Standards definiert. Da der Tech-Stack (siehe Kapitel 1.6) hauptsächlich aus .NET mit C# besteht, wird sich an den offiziellen .NET-Codekonventionen von Microsoft[34] orientiert, welche eine ausführliche Liste von in der Industrie gängiger Konventionen bieten.

Diese Konventionen werden durch die Entwicklungsumgebung (siehe Kapitel 2.5.2.1) sowie durch Linter in der CI/CD-Pipeline (siehe Kapitel 2.5.2.2) überprüft und durchgesetzt.

2.5.1.5 Zeitmanagement-Richtlinien

Um jederzeit einen aktuellen Überblick über den Projektfortschritt gewährleisten zu können, ist eine sorgfältige Zeiterfassung erforderlich.

- Die Teammitglieder erfassen ihre aufgewendete Arbeitszeit zeitnah und vollständig.
- Zeitangaben werden auf 15 Minuten gerundet.
- Der Aufgabenstatus wird kontinuierlich aktuell gehalten.

2.5.1.6 Versionsverwaltungsrichtlinien

Die Richtlinien zur Versionsverwaltung legen den Ablauf von Commit bis Merge fest und gelten für alle Arten von Versionsverwaltung, sowohl für die Dokumentation als auch für Code.

- Die Teammitglieder arbeiten mit Feature-Branches, wobei Commits ausschliesslich Aufgaben betreffen, die zum jeweiligen Feature-Branch gehören.
- Sobald die Definition of Done (siehe Kapitel 2.5.1.2) einer User Story erfüllt ist, wird der Feature-Branch in den Main-Branch gemerged.
- Commits sollen so klein gehalten werden, dass die Änderungen in einem prägnanten Satz beschrieben werden können.

[34] „.NET Coding Conventions - C# | Microsoft Learn“, Microsoft Corporation, 18. Januar 2025, https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/coding-conventions.

2.5.2 Tools

In diesem Unterkapitel werden die verschiedenen Tools und Programme für die Entwicklung festgelegt. Die Standardisierung des Toolgebrauchs fördert die Konsistenz im Entwicklungsteam und erleichtert Integrationen.

2.5.2.1 Entwicklungsumgebung (IDE)

Für die Entwicklungsumgebung (eine sogenannte *IDE* „Integrated Development Environment“) werden Visual Studio[35] von Microsoft und Rider[36] von JetBrains verwendet. Den Entwicklerinnen und Entwicklern steht frei, welches der beiden Programme sie nutzen möchten. Beide IDEs bieten alle für die Produktentwicklung notwendigen Features, wobei Rider in Bezug auf Performance Vorteile bietet.

Von den bereitgestellten Features der Tools, insbesondere den „Quality-of-Life“-Funktionen wie automatischer Formatierung und Refactoring, soll aktiv Gebrauch gemacht werden.

Um für eine einheitliche Formatierung zu sorgen, wird zusätzlich das CSharpier-Tool[37] verwendet, welches strenge Formatierungsregeln definiert und auch selbstständig umsetzt.

2.5.2.2 CI/CD

Das CI/CD (engl. „Continuous Integration and Continuous Delivery“, dt. „kontinuierliche Integration und kontinuierliche Auslieferung“) der Software wird durch eine GitLab-Pipeline umgesetzt, die bei jedem Push in das Repository automatisch ausgeführt wird. Sie ermöglicht das automatische Testen, Linting und Builden der Software anhand selbst konfigurierter sowie vordefinierter Template-Jobs. Um einen Feature-Branch in den Main-Branch mergen zu können, muss die Pipeline erfolgreich durchlaufen.

Der Grossteil der Jobs wird innerhalb des offiziellen .NET-Docker-Images von Microsoft[38] auf den von der OST bereitgestellten Linux-Runnern ausgeführt. Ausnahmen bilden die Integrationstests, welche einen Microsoft-nativen Treiber benötigen, sowie der Build-Job für den Installer, der auf dem WiX Toolset basiert. Diese werden auf GitLab-Runnern ausgeführt, die auf einem von der OST bereitgestellten virtuellen Server registriert sind.

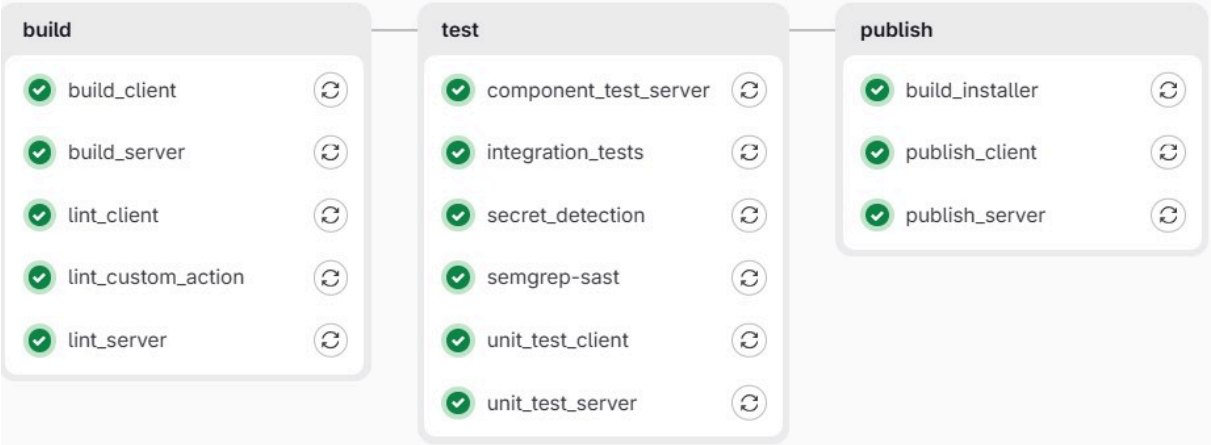


Abbildung 38: Überblick über alle Pipeline-Stages und Jobs.

[35] „Visual Studio: IDE und Code-Editor für Softwareentwicklung“, Microsoft Corporation, zugegriffen 8. Juni 2026, https://visualstudio.microsoft.com/de/.

[36] „Rider: the cross-platform .NET and game development IDE from JetBrains“, JetBrains, zugegriffen 8. Juni 2026, https://www.jetbrains.com/rider/.

[37] belav, „CSharpier“, zugegriffen 4. April 2026, https://csharpier.com/.

[38] „Official .NET Docker images“, Microsoft Corporation, zugegriffen 13. April 2026, https://learn.microsoft.com/en-us/dotnet/architecture/microservices/net-core-net-framework-containers/official-net-docker-images.

Die Pipeline ist in drei Stufen mit insgesamt 14 Jobs unterteilt:

build

Dieser Schritt umfasst drei Linting-Jobs, welche die drei C#-Projekte des Repositories auf ihre Formatierung per CSharpier überprüfen. Bei Abweichungen schlagen die entsprechenden Jobs fehl.

Zusätzlich enthalten sind zwei Build-Jobs, welche die Server- und Client-Projekte kompilieren.

test

In diesem Schritt werden alle automatisierten Tests durchgeführt (siehe Kapitel 2.5.3). Dazu gehören Unittests für Client und Server, ein Komponententest für den Server sowie ein Integrationstest für Client und Server.

Zusätzlich werden die beiden Template-Jobs „Secret Detection“[39] und „Static Application Security Testing“ (SAST)[40] ausgeführt. Diese überprüfen das Repository sowie Commits auf versehentlich veröffentlichte Geheimnisse beziehungsweise führen vordefinierte Sicherheitsanalysen auf dem Sourcecode durch.

publish

Im letzten Schritt werden Server und Client publiziert, wobei die resultierenden Kompilate erzeugt werden. Anschliessend wird aus den Artefakten der Publish-Jobs eine Windows Installer-Datei erstellt, mit der der Benutzer IsoTrack installieren kann. Diese Schritte werden nur im Main-Branch duchgeführt.

2.5.2.3 KI Tools

Das Projekt folgt der Richtlinie „Umgang mit KI bei schriftlichen Arbeiten“ des Departements für Informatik der OST. Diese regelt die Verwendung von KI-Tools unter Einhaltung von Datenschutz, akademischer Integrität, Korrektheit und korrekter Quellenangabe. Im Rahmen dieses Projekts wurden insbesondere ChatGPT, DeepL und Grammarly eingesetzt.

Eine vollständige Liste der genutzten Tools sowie deren Anwendungsbereiche findet sich im Hilfsmittelverzeichnis Anhang Sektion E.

2.5.3 Testkonzept

Das Testkonzept legt die Strategie zur Qualitätssicherung fest und definiert konkret, was, wie und wann getestet wird. Das erfolgreiche Bestehen aller Tests sowie eine hohe Codeabdeckung gelten als Indikatoren für ein qualitativ hochwertiges Produkt, weshalb ein sorgfältiges und umfassendes Testen angestrebt wird.

2.5.3.1 Unittests

Unittests sind White-Box-Tests. Dabei wird die interne Funktionsweise der Software, die Business-Logik, wird überprüft. Sie stellen die unterste Testebene dar und sind für jede Software unverzichtbar.

Dabei wird gezielt vorgegangen, indem nur ausgewählte Komponenten getestet werden, um den Einsatz von Mocks auf ein sinnvolles Mass zu beschränken. Komponenten der Business-Logik werden in der Regel umfassend getestet, während Komponenten wie API-Controller des Servers primär durch Integrationstests geprüft werden. Konkret werden die beiden Business-Logik-Namespaces `IsoTrack.Services` des Clients und `isotrack_server.Infrastructure` des Servers getestet.

[39] „Secret detection“, GitLab, zugegriffen 4. April 2026, https://docs.gitlab.com/user/application_security/secret_detection/.

[40] „Static application security testing (SAST)“, GitLab, zugegriffen 4. April 2026, https://docs.gitlab.com/user/application_security/sast/.

Idealerweise würde jede Verzweigung und jede Codezeile durch Tests abgedeckt, was einer Testabdeckung von 100 % entspräche. Da dies jedoch in vielen Fällen nicht notwendig ist und zu einem unverhältnismässigen Aufwand führen würde, wird stattdessen angestrebt, jede Funktion zu testen und eine Testabdeckung von etwa 80 % zu erreichen.

Die Testabdeckung der betroffenen Namespaces beträgt am Ende des Projekts ca. 85 %.

Name	Covered	Uncovered	Coverable	Total	Percentage	Covered	Total	Percentage
IsoTrackClient	**133**	**305**	**438**	**670**	**30.3%**	**31**	**68**	**45.5%**
+ -	**0**	**23**	**23**	**35**	**0%**	**0**	**0**	
+ IsoTrack.Configuration	**8**	**41**	**49**	**81**	**16.3%**	**0**	**16**	**0%**
+ IsoTrack.Data	**0**	**3**	**3**	**8**	**0%**	**0**	**0**	
+ IsoTrack.Infrastructure	**0**	**98**	**98**	**188**	**0%**	**0**	**12**	**0%**
+ IsoTrack.Mappings	**0**	**74**	**74**	**81**	**0%**	**0**	**0**	
– IsoTrack.Services	**125**	**66**	**191**	**277**	**65.4%**	**31**	**40**	**77.5%**
IsoTrack.Services.SyncService	125	37	162	234	77.1%	31	38	81.5%
IsoTrack.Services.WindowsBackgroundService	0	29	29	43	0%	0	2	0%
– IsoTrackServer	**852**	**2142**	**2994**	**5930**	**28.4%**	**167**	**745**	**22.4%**
+ -	**0**	**58**	**58**	**75**	**0%**	**0**	**2**	**0%**
+ isotrack_server.Analytics	**0**	**485**	**485**	**807**	**0%**	**0**	**118**	**0%**
+ isotrack_server.Components	**0**	**1339**	**1339**	**3040**	**0%**	**0**	**355**	**0%**
+ isotrack_server.Controllers	**0**	**134**	**134**	**205**	**0%**	**0**	**42**	**0%**
+ isotrack_server.Data	**81**	**0**	**81**	**103**	**100%**	**0**	**0**	
– isotrack_server.Infrastructure	**713**	**71**	**784**	**1377**	**90.9%**	**167**	**228**	**73.2%**
isotrack_server.Infrastructure.ChartColorService	32	0	32	85	100%	8	8	100%
isotrack_server.Infrastructure.ExerciseFilterSettings	13	0	13	138	100%	0	0	
isotrack_server.Infrastructure.ExerciseViewModel	56	31	87	138	64.3%	14	18	77.7%
isotrack_server.Infrastructure.ExportGeneratorService	93	3	96	162	96.8%	34	38	89.4%
isotrack_server.Infrastructure.FrontendUtils	21	1	22	85	95.4%	14	16	87.5%
isotrack_server.Infrastructure.GraphGeneratorService	92	0	92	139	100%	12	12	100%
isotrack_server.Infrastructure.PDFGeneratorService	359	27	386	547	93%	79	130	60.7%
isotrack_server.Infrastructure.PathProvider	17	0	17	24	100%	2	2	100%
isotrack_server.Infrastructure.TRAReaderService	30	9	39	59	76.9%	4	4	100%
+ isotrack_server.Models	**58**	**55**	**113**	**323**	**51.3%**	**0**	**0**	

Abbildung 39: Testabdeckung des Clients und Servers.

Zu den Unit-Tests zählen auch Komponententests, die für den Server durchgeführt werden. Dabei werden die Endpunkte des Servers gezielt auf die erfolgreiche Verarbeitung von Anfragen überprüft.

2.5.3.2 Integrationstests

Integrationstests prüfen das Zusammenspiel mehrerer Komponenten innerhalb eines Systems, insbesondere deren Kommunikation miteinander.

In IsoTrack wird dabei konkret die Kopplung zwischen Client und Server getestet. Zu diesem Zweck wurde ein xUnit-Testprojekt erstellt, das Client und Server startet, kurz wartet und anschliessend die vom Server erstellte SQLite-Datenbank darauf überprüft, ob neue Einträge vorhanden sind.

Die inhaltliche Richtigkeit der Daten wird dabei nicht geprüft. Dies erfolgt in den Komponenten- und Unit-Tests, die vor den Integrationstests durchgeführt werden.

2.5.3.3 Performancetests

Im Rahmen der Entwicklung wurden wiederholt manuelle Performancemessungen durchgeführt, vor allem nach architekturrelevanten Änderungen am Client. Ziel dieser Messungen war die Bewertung des Laufzeit- und Speicherverhaltens sowie die Identifikation potenzieller Speicherlecks im Dauerbetrieb.

Die Analyse konzentrierte sich primär auf den Arbeitsspeicherverbrauch des Prozesses. Als zentrale Metrik wurde dabei das Working Set verwendet. Dieses beschreibt den tatsächlich vom Betriebssystem einem Prozess zugewiesenen physischen Speicher inklusive Heap, nativer Ressourcen sowie geladener Assemblies und stellt damit eine geeignete Grösse zur Bewertung des realen Speicherbedarfs dar.

Für die Erhebung der Messwerte wurde das Tool dotnet-counters[41] eingesetzt.

[41] „dotnet-counters diagnostic tool - .NET CLI - .NET | Microsoft Learn“, Microsoft Corporation, 2. Oktober 2025, https://learn.microsoft.com/en-us/dotnet/core/diagnostics/dotnet-counters.

```
Name                                                                    Current Value
[System.Runtime]
    dotnet.assembly.count ({assembly})                                             85
    dotnet.exceptions ({exception})
        error.type
        -----------
        IOException                                                                 1
    dotnet.gc.collections ({collection})
        gc.heap.generation
        ------------------
        gen0                                                                        2
        gen1                                                                        2
        gen2                                                                        4
    dotnet.gc.heap.total_allocated (By)                                    57,109,784
    dotnet.gc.last_collection.heap.fragmentation.size (By)
        gc.heap.generation
        ------------------
        gen0                                                                        0
        gen1                                                                      768
        gen2                                                                    1,912
        loh                                                                        32
        poh                                                                         0
    dotnet.gc.last_collection.heap.size (By)
        gc.heap.generation
        ------------------
        gen0                                                                        0
        gen1                                                                   17,656
        gen2                                                                  883,168
        loh                                                                    98,384
        poh                                                                    24,528
    dotnet.gc.last_collection.memory.committed_size (By)                   10,768,384
    dotnet.gc.pause.time (s)                                                     0.035
    dotnet.jit.compilation.time (s)                                              2.053
    dotnet.jit.compiled_il.size (By)                                         559,494
    dotnet.jit.compiled_methods ({method})                                     6,101
    dotnet.monitor.lock_contentions ({contention})                                 0
    dotnet.process.cpu.count ({cpu})                                              12
    dotnet.process.cpu.time (s)
        cpu.mode
        --------
        system                                                                  1.422
        user                                                                        3
    dotnet.process.memory.working_set (By)                                  1.409e+08
    dotnet.thread_pool.queue.length ({work_item})                                  0
    dotnet.thread_pool.thread.count ({thread})                                     2
    dotnet.thread_pool.work_item.count ({work_item})                             515
    dotnet.timer.count ({timer})                                                   3
```

Abbildung 40: Erhobene Performance-Metriken des IsoTrack Clients mittels dotnet-counters.

Die Messung in Abbildung 40 zeigt ein Working Set von rund 140'900'000 Bytes (ca. 140 MB). Da der Client aufgrund möglicher Änderungen im Datensatz den relevanten Datenbestand jeweils vollständig analysieren muss, steigt der Speicherverbrauch bei der ersten Erkennung einer Änderung in der Datenbank von initial rund 50 MB auf etwa 140 MB. Im weiteren Verlauf bleibt der Speicherverbrauch jedoch stabil, wodurch Memory Leaks ausgeschlossen werden können.

Insgesamt ist der beobachtete Speicherverbrauch für einen .NET-basierten Hintergrunddienst mit Dependency Injection, HTTP-Kommunikation und kontinuierlicher Synchronisierung als unkritisch zu bewerten und liegt im erwartbaren Bereich vergleichbarer Anwendungen.

2.5.3.4 Abnahmetests

Während der Transition-Phase wurden gemeinsam mit dem Projektpartner am 28. Mai Abnahmetests durchgeführt, wobei Schritt für Schritt jede funktionale und nicht-funktionale Anforderung einzeln geprüft und vorgeführt wurde.

Ob eine Anforderung bestanden (✓) oder nicht bestanden (✗) wurde, entscheidet dabei ausschliesslich der Projektpartner.

Die Resultate sind im Kapitel 2.5.3.5 Testprotokoll einsehbar.

2.5.3.5 Testprotokoll

Am 28. Juni 2026 wurden beim Projektpartner vor Ort sämtliche definierten Anforderungen gemeinsam validiert. Dabei wurde jede funktionale und nicht-funktionale Anforderung einzeln überprüft und deren Erfüllung durch den Projektpartner bestätigt.

Mit Ausnahme der funktionalen Anforderung ISOTRACK-37 konnten alle Anforderungen erfolgreich umgesetzt werden. Diese wurde nachträglich implementiert und anschliessend separat getestet.

Die Anforderung ISOTRACK-91 wurde erst nach den Abnahmetests vom Projektpartner neu in Auftrag gegeben und daher nachträglich erfasst, implementiert und abschliessend getestet.

ID	Beschreibung	Resultat
ISOTRACK-30	Als User möchte ich alle Patientendaten als Excel-Datei auf einmal exportieren können, um Zeit zu sparen.	✓
ISOTRACK-31	Als User möchte ich die Patientendaten eines Patienten als Excel-Datei exportieren können, um Zeit zu sparen.	✓
ISOTRACK-32	Als User möchte ich meine Microsoft Access Datenbank als Quelle angeben, um das Exportieren zu automatisieren.	✓
ISOTRACK-33	Als User möchte ich meine Daten automatisch synchronisieren, um Fehler zu vermeiden.	✓
ISOTRACK-34	Als User möchte ich die Messdaten von Patienten anhand eines Referenzwertes vergleichen können.	✓
ISOTRACK-35	Als User möchte ich bei Patienten zusätzliche Daten, wie das Operationsdatum, angeben können	✓
ISOTRACK-36	Als User möchte ich einen Überblick über die Daten, z. B. Zeitverlauf oder mehrere Messungen, eines spezifischen Patienten erhalten, um Schlüsse daraus zu ziehen zu können.	✓
ISOTRACK-37	Als User möchte ich die Patientenauswahl anhand bestimmter Kriterien, wie Alter oder Messdatum, filtern können.	✓*
ISOTRACK-38	Als User möchte den Graphen für den Messverlauf einer Messung visualisiert bekommen haben.	✓
ISOTRACK-39	Als User möchte den Graphen für den Messverlauf mehrerer Messungen eines Patienten visualisiert bekommen haben.	✓
ISOTRACK-40	Als User möchte ich den Graphen für den Messverlauf aller Messungen visualisiert bekommen haben.	✓
ISOTRACK-41	Als User möchte ich eine Übersicht über alle Daten haben, um einen Gesamtüberblick zu erhalten.	✓
ISOTRACK-66	Als User möchte ich eine Übersicht über die Datenqualität aller Daten haben.	✓
ISOTRACK-86	Als User möchte ich zwei Messungen der beiden Beine eines Patienten als PDF exportieren können.	✓
ISOTRACK-91	Als User möchte ich bei einer Messung die falsch angegebene Beinseite korrigieren können, um fehlerhafte Daten zu vermeiden.	✓*
Total:		15/15

Tabelle 16: Testprotokoll der funktionalen Anforderungen

ID	Beschreibung	Resultat
NFA 01	Vertraulichkeit	✓
NFA 02	Benutzerunterstützung	✓
NFA 03	Angemessenheit Erkennbarkeit / Selbstbeschreibbarkeit	✓
NFA 04	Installierbarkeit	✓
Total:		4/4

Tabelle 17: Testprotokoll der nicht-funktionalen Anforderungen

3 Projekt Dokumentation

Dieses Kapitel hat das Ziel, Einsicht in den konkreten Projektablauf mitsamt Management und Aufbau zu gewähren.

3.1 Projektplan

Während des ersten Sprints in der Inception-Phase wurde ein initialer Projektplan erstellt, der im weiteren Projektverlauf kontinuierlich an die jeweils aktuellen Gegebenheiten und gewonnenen Erkenntnisse angepasst wurde.
Der Projektplan definiert Ressourcen, Prozesse, Rahmenbedingungen, den Ablauf des Projekts sowie weitere grundlegende Aspekte, um eine klare Struktur und eine reibungslose Durchführung sicherzustellen.

Abbildung 41 bildet den Zeitstrahl des Projekts ab, wobei die Einteilung der Phasen klar ersichtlich ist.

Semesterwoche:	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8	SW9	SW10	SW11	SW12	SW13	SW14	SW15	SW16	SW17
Datum:	16.2. - 22.2.	23.2. - 1.3.	2.3. - 8.3.	9.3. - 15.3.	16.3. - 22.3.	23.3. - 29.3.	30.3. - 5.4.	6.4. - 12.4.	13.4. - 19.4.	20.4. - 26.4.	27.4. - 3.5.	4.5. - 10.5.	11.5. - 17.5.	18.5. - 24.5.	25.4. - 31.5.	1.6. - 7.6.	8.6. - 12.6.
Sprint:	S1 (16.2. - 1.3.)		S2 (2.3. - 15.3.)		S3 (16.3. - 29.3.)		S4 (30.3. - 12.4.)		S5 (13.4. - 26.4.)		S6 (27.4. - 10.5.)		S7 (11.5. - 24.5.)		S8 (25.4. - 7.6.)		S9 (8.6. - 12.6.)
RUP Phase:	Inception		Elaboration				Construction								Transition		

Abbildung 41: Zeitstrahl des Projekts

3.1.1 Ressourcen

Der Leitfaden zur Bachelorarbeit sieht einen Arbeitsaufwand von 360 Stunden pro Person während des Semesters sowie zusätzlich eine Blockwoche nach der Unterrichtszeit vor. Es liegt in der Verantwortung der Studierenden, diesen Aufwand gleichmässig auf die 17 Wochen zu verteilen und eine entsprechende Zeiterfassung zu führen.

3.1.2 Prozesse

Das Projekt wurde in Anlehnung an Scrum+ konzipiert und geplant, einem Vorgehensmodell, das Scrum mit dem Rational Unified Process (RUP)[42] kombiniert. Aufgrund der Teamgrösse von lediglich zwei Mitgliedern wurden nicht alle Scrum-Rollen formal besetzt werden. Die vorgesehenen Scrum-Zeremonien wurden daher teilweise reduziert beziehungsweise in vereinfachter Form durchgeführt, um eine höhere Flexibilität zu gewährleisten und eine enge Zusammenarbeit innerhalb des Teams zu fördern.

Der 17-wöchige Projektzeitraum wurde in neun Iterationen, sogenannte Sprints, unterteilt. Acht dieser Sprints hatten eine Dauer von jeweils zwei Wochen, während der neunte Sprint mit einer Laufzeit von fünf Tagen bis zum Abgabetermin verkürzt angesetzt wurde.

3.1.2.1 Phasen

Im Rational Unified Process (RUP) wird ein Projekt in vier Phasen mit fliessenden Übergängen unterteilt, um den Fortschritt strukturiert darzustellen. Diese Phasen definieren einen übergeordneten Rahmen, der vorgibt, in welchem Zeitraum bestimmte Aufgaben schwerpunktmässig bearbeitet werden sollen, und erleichtern dadurch den Übergang zwischen den einzelnen Projektabschnitten.

[42] P. Kruchten, „Tutorial: introduction to the Rational Unified Process®“, preprint, IEEE Institute of Electrical and Electronics Engineers, o. J., zugegriffen 20. Februar 2026, https://ieeexplore.ieee.org/document/1008039.

Die Phaseneinteilung ist dabei bewusst generisch gehalten. Im vorliegenden Projekt umfasste die Inception-Phase eine Iteration, die Elaboration-Phase zwei Iterationen, die Transition-Phase anderthalb Iterationen und die Construction-Phase die verbleibenden vier Iterationen.

Inception	In der Inception Phase liegt der Fokus des Teams auf der Ausarbeitung der Aufgabenstellung, der Erstellung eines initialen Projektplans, der Bearbeitung grundlegender administrativer Angelegenheiten sowie der Einarbeitung in relevante Themengebiete.
Elaboration	In der Elaboration Phase werden die Anforderungen definiert und erste Prototypen erstellt.
Construction	Die Construction Phase konzentriert sich auf die Entwicklung der Software.
Transition	Die Transition Phase markiert den Beginn des Endes der Entwicklung. Es werden Abnahmetests durchgeführt und die Dokumentation fertiggestellt.

Tabelle 18: RUP Phasen.

Abbildung 42 beschreibt die grobe Einplanung aller Epics (siehe Kapitel 2.1.2) inklusive einer initialen Zeitabschätzung sowie der Zuordnung zu den jeweiligen Sprints und Phasen.

		Inception	Elaboration		Construction				Transition	
	720h	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Sprint 8	Sprint 9
Projektmanagement	140									
Projektplanung	30									
Tooling / Infrastruktur	40									
Testing	80									
Anforderungsanalyse	40									
Softwarearchitektur	10									
Datenverwaltung	200									
Datenvisualisierung	180									
Total:	720									

Abbildung 42: Übersicht über alle Epics mit grober Zeiteinteilung.

3.1.2.2 Sprints

Sprints sind zeitlich begrenzte Iterationen, in denen Aufgaben aus dem Product Backlog bearbeitet werden, um das Produkt schrittweise und inkrementell weiterzuentwickeln. Vor jedem Sprint werden die Aufgaben im Sprint-Planning anhand ihrer Priorität und des geschätzten Aufwands ausgewählt und eingeplant. Dabei muss jede Aufgabe die Definition of Done (siehe Kapitel 2.5.1.2) erfüllen, bevor sie in den Backlog aufgenommen werden kann.
Der Backlog wird mit Jira verwaltet.

Wie im Kapitel 3.1.2 erwähnt, ist der Projektzeitraum in neun Sprints aufgeteilt. Mit Ausnahme des letzten Sprints dauern diese jeweils zwei Wochen von Montag bis Sonntag. Der erste Sprint begann am Montag, 16. Februar, und endete am Sonntag, 1. März. Da der Abgabetermin in die Mitte des neunten Sprints fällt, wird dieser verkürzt durchgeführt.

3.1.3 Meilensteine

Meilensteine geben dem Team die Möglichkeit, Stakeholdern ausserhalb des Entwicklungsteams einen strukturierten Einblick in den Fortschritt des Projekts zu geben. Ein Meilenstein umfasst eine vordefinierte Liste von Aufgaben, die bis zum Ende des jeweils angegebenen Sprints erfüllt sein sollten. Die Meilensteine sind dabei so gestaltet, dass sie im Laufe des Projektes nicht angepasst werden müssen, und definieren gröbere Aufgabenbereiche.

Die Definition der Meilensteine orientierte sich in diesem Projekt an den Phasen des Rational Unified Process (RUP), an Abgaben sowie an den Epics.

M1 - Inception (Sprint 1)
- Die Projekprozesse wurden definiert.
- Nötige Richtlinien wurden festgelegt.
- Die Projektinfrastruktur wurde aufgesetzt.
- Alle Projektrollen wurden verteilt.
- Allfällige administrative Angelegenheiten wurden geklärt.
- Ein Langzeit Projektplan wurde definiert.
- Eine initiale Risikomatrix wurde definiert.
- Es wurde sich mit der Ausgangssituation auseinandergesetzt.

M2 - Elaboration (Sprint 3)
- Die Code Infrastruktur wurde aufgelegt.
- Die Risikomatrix wurde neu evaluiert.
- Die funktionalen Anforderungen wurden definiert.
- Die nicht-funktionalen Anforderungen wurden definiert.
- Eine initiale Softwarearchitektur wurde beschrieben.
- Ein Testkonzept wurde definiert.

M3 - Datenverwaltung (Sprint 5)
- Alle funktionalen Anforderungen zur Datenverwaltung (siehe Tabelle 2) wurden erfüllt.

M4 - Datenvisualisierung (Sprint 7)
- Alle funktionalen Anforderungen zur Datenvisualisierung (siehe Tabelle 3) wurden erfüllt.
- Das Produkt wurde dem Projektpartner mindestens einmal vorgestellt, und das daraus resultierende Feedback wurde umgesetzt.

M5 - Construction (Sprint 7)
- Die Architektur ist adäquat und dokumentiert.
- Alle User Stories erfüllen die Definition of Done und sind dokumentiert.
- Das Testkonzept wurde endgültig festgelegt.
- Die Zwischenpräsentation wurde erfolgreich durchgeführt.
- Der Bachelorprüfungstermin wurde festgelegt.

M6 - Transition (Sprint 8)
- Die Abnahmetests wurden erfolgreich durchgeführt.
- Das Abstract steht zur Abgabe bereit.
- Alle nicht-funktionalen Anforderungen sind erfüllt.

M7 - Abgabe (Sprint 9)
- Die Dokumentation wurde fertiggestellt und auf orthographische Korrektheit geprüft.
- Das Poster wurde fertiggestellt.
- Das Abstract wurde dem Feedback des Betreuers entsprechend angepasst.
- Die finale Abstract-Version wurde abgegeben.
- Alle Abgaben wurden im Arbeitsverwaltungstool (AVT) fristgerecht eingereicht.
- Alle definierten Abgaben wurden fristgerecht allen Stakeholdern im gewünschten Format abgegeben.

3.1.4 Proof of Concept

In der Elaboration-Phase wurden Machbarkeitsnachweise, sogenannte *Proofs of Concept*, erstellt, welche die ersten aufgestellten Konzepte testeten. Die Nachweise wurden in kleinem Rahmen umgesetzt und mussten keine besonderen Qualitätsanforderungen erfüllen. Im Vordergrund standen insbesondere die Integration der Access-Datenbankdatei in einen Windows-Dienst, die Überführung des Schemas der Access-Datenbank in eine SQLite-Datenbank sowie die Erstellung einer Windows-Installationsdatei.

Zum Entwickeln dieser kleinen Programme wurden die von Microsoft zur Verfügung gestellten Tutorials als Basis verwendet, zum Beispiel um einen Windows Hintergrundservice[43] oder eine benutzerdefinierte Windows-Installationsdatei[44] zu erstellen

3.1.5 Fristen und Abgabetermine

Das Bachelorarbeitsmodul schreibt eine Reihe verbindlicher Termine vor, die einzuhalten sind. Dazu gehören insbesondere:

- **Mitte des Projekts:** Durchführung einer Zwischenpräsentation mit Betreuungsperson und Gegenleser.
- **Bis 8. Juni:** Abgabe der ersten Version des Abstracts zur Korrektur durch den Betreuer.
- **Bis 12. Juni, 17:00 Uhr:** Abgabe der Bachelorarbeit inklusive aller erforderlichen Dokumente.

3.1.6 Teamrollen

Aufgrund der geringen Teamgrösse wurde auf eine detaillierte Rollenaufteilung innerhalb des Entwicklungsteams verzichtet. Stattdessen arbeiten beide Teammitglieder in allen Aufgabenbereichen mit und tragen gemeinsam die Verantwortung für sämtliche Projektaspekte.

Im Sinne agiler Vorgehensweisen erfolgt die Aufgabenverteilung primär anhand der verfügbaren zeitlichen Ressourcen sowie persönlicher Interessen und Fachkenntnisse. Sollten Wissenslücken oder fachliche Herausforderungen auftreten, wird eine gegenseitige Unterstützung innerhalb des Teams vorausgesetzt.

Rolle	Verantwortliche	Beschreibung
Betreuungsperson	Prof. Dr. Remo Frey	Unterstützt die Studierenden beim Projektablauf und ist Hauptverantwortlicher für die Notenfindung.
Projektpartner	Timothy Grob Janika Kessler	Ansprechpersonen für fachliche und auftragsspezifische Fragestellungen sowie Abnahmeinstanz der Anforderungen.
Experte	Dr. Edward Ho	Begutachtet das Projekt und erstellt eine unabhängige Bewertung.
Gegenleser	Thomas Bocek	Unterstützt die Notenfindung, indem eine alternative Sichtweise geboten werden kann.
Entwicklungsteam	Tobias Keel Simon Weibel	Verantwortlich für die Durchführung des Projekts, inklusive regelmässiger Präsentation des Projektstandes, Abgabe von Artefakten und Dokumentation.

Tabelle 19: Übersicht der Teamrollen mitsamt Beschreibung.

[43] „Create Windows Service using BackgroundService - .NET | Microsoft Learn“, Microsoft Corporation, 22. Oktober 2025, https://learn.microsoft.com/en-us/dotnet/core/extensions/windows-service.

[44] „Create a Windows Service installer - .NET | Microsoft Learn“, Microsoft Corporation, 20. Oktober 2025, https://learn.microsoft.com/en-us/dotnet/core/extensions/windows-service-with-installer.

3.1.7 Meetings

Im Verlauf des Projekts fanden sowohl regelmässige als auch anlassbezogene Meetings statt, die jeweils protokolliert wurden. In den regelmässig stattfindenden Beratungsgesprächen, welche alle zwei Wochen am Freitagmorgen durchgeführt wurden, informierten die Studierenden den Betreuer über den aktuellen Projektstand sowie über die Planung des kommenden Sprints. Bei Bedarf wurden einzelne Termine ausgesetzt oder zusätzliche Ad-hoc-Meetings angesetzt.
Darüber hinaus fanden Meetings mit dem Projektpartner statt, um Anforderungen zu definieren oder für die Abnahmetests.

Meeting	Kalender-woche	Semester-woche	Datum	Zeit	Dauer
Kickoff Meeting Betreuer	7	-	13.02.2026	10:00	45 Minuten
Kickoff Meeting Projektpartner	8	1	18.02.2026	10:10	1 Stunde
Beratungsgespräch 1	10	3	06.03.2026	09:00	30 Minuten
Anforderungsinterview	11	4	12.03.2026	13:30	1 Stunde
Beratungsgespräch 2	12	5	20.03.2026	09:00	30 Minuten
Beratungsgespräch 3	15	8	10.04.2026	09:00	30 Minuten
Demonstrationsmeeting 1	17	10	23.04.2026	11:00	30 Minuten
Zwischenpräsentation	18	11	01.05.2026	09:00	45 Minuten
Beratungsgespräch 4	20	13	15.05.2026	09:00	45 Minuten
Demonstrationsmeeting 2	21	14	19.05.2026	14:00	30 Minuten
Abnahmetests	22	15	28.05.2026	13:30	1 Stunde
Beratungsgespräch 5	22	15	29.05.2026	09:00	30 Minuten
Beratungsgespräch 6	24	17	12.06.2026	09:00	15 Minuten

Tabelle 20: Liste aller Meetings.

3.1.8 Planungstools

Ein sauberes Projektmanagement erfordert die Wahl und konsequente Nutzung geeigneter Tools. Dadurch ist es jederzeit möglich, eine klare Übersicht über den Projektverlauf zu behalten, da alle Aufgaben einsehbar sind und der aktuelle Zeitaufwand ersichtlich ist.

3.1.8.1 Atlassian Jira

Jira dient als zentrales Projektmanagement-Tool. Hier werden alle Aufgaben dokumentiert, Sprints zugeordnet, der Fortschritt festgehalten sowie das Zeitmanagement verwaltet.

3.1.8.2 OST GitLab

Die von der OST bereitgestellte GitLab-Plattform dient als zentrale Sammelstelle für den Sourcecode mit Versionsverwaltung. Für die Planung besonders relevant sind das Dokumentationsrepository sowie die Arbeit mit Feature-Branches, die den im Jira definierten Aufgaben entsprechen.

In der GitLab-Gruppe werden die Dokumentation sowie der Sourcecode in jeweils separaten Projekten verwaltet.

3.1.8.3 Microsoft Teams

Teams ist die zentrale Kommunikationsplattform, insbesondere zwischen dem Team und dem Betreuer. Über Teams werden zudem die zweiwöchentlichen Meetings abgehalten und Abgabedokumente verwaltet.

3.2 Risikomanagement

Zur frühzeitigen Erkennung von Risiken und zur Minimierung potenzieller Schäden wurde ein aktives Risikomanagement betrieben. Bestehende Risiken wurden dabei regelmässig neu bewertet und um neu identifizierte Risiken ergänzt.

In diesem Kapitel werden die Risikomatrizen der einzelnen Sprints dargestellt, sofern sich Änderungen gegenüber der vorherigen Bewertung ergeben haben.

3.2.1 Risikomatrix Sprint 1

Im ersten Sprint wurde versucht, erste Risiken zu ermitteln, einzuordnen sowie erste Massnahmen zu definieren und zu ergreifen.

Da viele Aspekte des Projekts noch nicht klar sind, stellen die ersten vier Risiken noch kein genaues Bild dar und sind eher als grobe Schätzung zu verstehen.

Schadensausmass / Wahrscheinlichkeit	Sehr gering	Gering	Mittel	Hoch	Sehr hoch
Sehr unwahrscheinlich					R3
Unwahrscheinlich				R1	
Möglich				R4	
Wahrscheinlich			R2		
Sehr Wahrscheinlich					

Tabelle 21: Risikomatrix Sprint 1.

3.2.2 Risikomatrix Sprint 2

Kontaktversuche mit Easytech schlugen fehl, weswegen die direkte Arbeit mit dem Hersteller sowie ein Zugriff auf die Dokumentation ausbleiben.

Die Chance des Risikos 2, dass nicht nachvollziehbare Eigenheiten der Software zu Problemen führen werden, steigt dadurch immens.

Schadensausmass / Wahrscheinlichkeit	Sehr gering	Gering	Mittel	Hoch	Sehr hoch
Sehr unwahrscheinlich					R3
Unwahrscheinlich				R1	
Möglich				R4	
Wahrscheinlich					
Sehr Wahrscheinlich			R2		

Tabelle 22: Risikomatrix Sprint 2.

3.2.3 Risikomatrix Sprint 3

Die TRA-Dateien konnten erfolgreich dekodiert werden, wodurch die Proprietarität der Isocine3-Software keine wesentlichen Einschränkungen mehr darstellt.

Für den Datenbankzugriff wurde OLE DB mit einem entsprechenden Access-Treiber gewählt. Da Isocine3 selbst auf dieselbe Access-Datenbank zugreift, ist davon auszugehen, dass dieser Treiber im Zielsystem verfügbar ist. Dennoch besteht ein Restrisiko, insbesondere im Hinblick auf weitere Systemkomponenten.

Schadensausmass / Wahrscheinlichkeit	Sehr gering	Gering	Mittel	Hoch	Sehr hoch
Sehr unwahrscheinlich				R5	R3
Unwahrscheinlich				R1	
Möglich		R4			
Wahrscheinlich					
Sehr Wahrscheinlich			R2		

Tabelle 23: Risikomatrix Sprint 3.

3.2.4 Risikomatrix Sprint 5

IsoTrack konnte auf dem Endgerät des Projektpartners installiert und auf Kompatibilität getestet werden. Dadurch konnte das Risiko gänzlich eliminiert werden.

Schadensausmass / Wahrscheinlichkeit	Sehr gering	Gering	Mittel	Hoch	Sehr hoch
Sehr unwahrscheinlich				~~R5~~	R3
Unwahrscheinlich				R1	
Möglich		R4			
Wahrscheinlich					
Sehr Wahrscheinlich			R2		

Tabelle 24: Risikomatrix Sprint 5.

3.2.5 Risikomatrix Sprint 6

Das Verhalten der Isocine3-Software konnte im Verlauf des Projekts schrittweise eruiert werden, wodurch die Hauptfunktionen und Interaktionen weitgehend identifiziert werden konnten. Dennoch besteht weiterhin die Möglichkeit, dass einzelne Eigenschaften oder Sonderfälle unentdeckt bleiben und gegebenenfalls auch bis zum Projektabschluss nicht erkannt werden. Solche Fälle könnten lediglich zufällig im praktischen Einsatz auffallen.

Schadensausmass / Wahrscheinlichkeit	Sehr gering	Gering	Mittel	Hoch	Sehr hoch
Sehr unwahrscheinlich					R3
Unwahrscheinlich				R1	
Möglich		R4	R2		
Wahrscheinlich					
Sehr Wahrscheinlich					

Tabelle 25: Risikomatrix Sprint 6.

3.2.6 Risiken

Nachfolgend werden alle erfassten Risiken aufgelistet, welche eine Identifikationsnummer, einen Namen, eine Beschreibung, eine Auswirkung sowie Massnahmen und den Sprint zum Zeitpunkt der Erfassung enthalten.

Risiko	R1
Name	Längerer Ausfall eines Teammitglieds
Beschreibung	Ein Teammitglied könnte aufgrund von Krankheit für längere Zeit ausfallen.
Auswirkungen	Das Projekt könnte in Verzug geraten und geplante Features müssten gestrichen werden.
Massnahmen	Ausführliche Dokumentation sowie konsequente Einhaltung aller Richtlinien und Prozesse ermöglichen es dem verbleibenden Teammitglied, die Arbeit reibungslos zu übernehmen.
Erfasst	Sprint 1 - Inception

Risiko 1: Längerer Ausfall eines Teammitglieds.

Risiko	R2
Name	Mangelhafte Dokumentation
Beschreibung	Die Softwaredokumentation des Geräts vom Projektpartner könnte unvollständig oder nicht detailliert genug sein.
Auswirkungen	Der Aufbau der Schnittstelle zwischen Server und Client würde dadurch zusätzlichen Zeit- und Arbeitsaufwand erfordern.
Massnahmen	Sorgfältige Analyse der vorhandenen Daten sowie direkter Kontakt mit dem Hersteller Easytech, um fehlende Informationen zu beschaffen.
Erfasst	Sprint 1 - Inception

Risiko 2: Mangelhafte Dokumentation.

Risiko	R3
Name	Sicherheitsbedenken
Beschreibung	Da der Projektpartner primär mit Gesundheitsdaten arbeitet, die strengsten Datenschutzgesetzen unterliegen, ist beim Umgang mit Kundendaten höchste Vorsicht geboten.
Auswirkungen	Ein Verstoß gegen Datenschutzgesetze könnte die rechtlichen Rahmenbedingungen verletzen und zu erheblichen Konsequenzen führen.
Massnahmen	Sorgfältige Recherche der geltenden Rechtsvorschriften, gewissenhafter Umgang mit den Daten, Implementierung von Sicherheitsmassnahmen in Pipeline und Server sowie Abstimmung mit dem Projektpartner über alle notwendigen Abklärungen.
Erfasst	Sprint 1 - Inception

Risiko 3: Sicherheitsbedenken.

Risiko	R4
Name	Proprietäre Software des Herstellers
Beschreibung	Aufgrund der proprietären Natur der Software könnte der Hersteller den Zugang zu detaillierter Dokumentation einschränken oder verweigern.
Auswirkungen	Der Aufbau des Clients würde dadurch zusätzlichen Zeit- und Arbeitsaufwand erfordern.
Massnahmen	Bestimmte funktionale Anforderungen könnten reduziert oder gestrichen werden, um den Arbeitsaufwand innerhalb des vorgesehenen Rahmens zu halten. Zusätzlich kann versucht werden, die proprietären TRA-Dateien zu dekodieren.
Erfasst	Sprint 1 - Inception

Risiko 4: Proprietäre Software des Herstellers.

Risiko	R5
Name	Inkompatibilität beim Kundengerät
Beschreibung	Die Umgebung des Kunden könnte die Installation von Applikationen unverifizierter Dritter aus Sicherheitsgründen unterbinden, wodurch eine Installation erschwert oder vollständig verhindert werden könnte. Da die Umgebung des Kunden realistisch nicht nachgestellt werden kann, ist ein Testlauf nicht möglich.
Auswirkungen	Es müsste erheblich viel Zeit aufgewendet werden, um eine Lösung zu finden beziehungsweise das Problem zu debuggen.
Massnahmen	Die Software muss so schnell wie möglich in einen verwendbaren Zustand gebracht werden, sodass eine Installation getestet werden kann.
Erfasst	Sprint 3 - Elaboration

Risiko 5: Inkompatibilität beim Kundengerät.

3.2.7 Probleme

Dieses Unterkapitel behandelt alle im Verlauf des Projekts aufgetretenen Probleme sowie deren Bearbeitung und Auswirkungen.

3.2.7.1 Access-Datenbank Mismatch

Die Access-Datenbank der Isocine3-Software weist mehrere Unstimmigkeiten in Bezug auf Übersetzungen und Zuordnungen auf.

Die Lokalisierung der Software ist sehr einfach umgesetzt: Für jede verfügbare Sprache existiert eine eigene Textdatei (z. B. „Isocine3GE.txt“ für Deutsch oder „Isocine3IT.txt“ für Italienisch), in der 311 Zeichenfolgen anhand der entsprechenden Sprache angegeben werden.

Diese Dateien folgen alle demselben Schema:

```
[Msg]
0 = ISOKINE3
1 = Patienten
2 = Archiv
3 = Übung
4 = Serien-Assistent
5 = ALLG. EINSTELLUNGEN
...
```

```
311 = WARNUNG! Mögliche interne Überhitzung. Warten Sie ungefähr eine halbe Stunde
und versuchen Sie es erneut, andernfalls Kundenservice kontaktieren
```

Ein erster Fehler zeigt sich bei der Übersetzung der Patientenfelder: Der italienische Begriff „Cognome“ (dt. „Nachname“) wird fälschlicherweise mit „Vorname“ übersetzt, während „Nome“ entsprechend inkorrekt gemappt wird.

```
Isocine3IT.txt            Isocine3GE.txt
...                       ...
17 = Cognome              17 = Vorname
18 = Nome                 18 = Name
...                       ...
```

Dieser Fehler könnte theoretisch durch einen einfachen Austausch der entsprechenden Einträge in der deutschen Lokalisationsdatei behoben werden. Da jedoch bereits mehrere hundert Patienten in der Datenbank erfasst sind, würde eine nachträgliche Anpassung der Daten erhebliche Risiken bergen. Aus diesem Grund wurde entschieden, die Korrektur auf Client-Seite vorzunehmen.

Ein weiterer Übersetzungsfehler betrifft das Feld „SSN“, welches in der deutschen Version als „Therapeut“ dargestellt wird:

```
Isocine3IT.txt            Isocine3GE.txt
...                       ...
279 = SSN                 279 = Therapeut
...                       ...
```

Die tatsächliche Bedeutung von „SSN“ konnte nicht eindeutig bestimmt werden. Es ist jedoch davon auszugehen, dass es sich um eine personenbezogene Identifikationsnummer handelt, beispielsweise im Sinne einer Sozialversicherungsnummer (engl. „Social Security Number“) oder im Kontext des italienischen Gesundheitssystems („Servizio Sanitario Nazionale“).

Die Zuordnung zum Feld „Therapeut“ ist in jedem Fall fachlich inkorrekt. In IsoTrack wird das Feld jedoch mittels eines Übersetzungs-Mappings verarbeitet und im User Interface des Servers als „Therapeut“ dargestellt, sodass für den Projektpartner keine Anpassungen erforderlich sind.

Neben fehlerhaften Übersetzungen bestehen auch inkonsistente Feldzuweisungen zwischen Datenbank und Benutzeroberfläche. So entspricht das Datenbankfeld „TELEFONO2“ (zweite Telefonnummer des Patienten in der Tabelle „ANAGRAFICA“, siehe Abbildung 43) dem Feld „Arzt“ in der Benutzeroberfläche (siehe Abbildung 44).

ANAGRAFICA

NUMEROUT	NOME	COGNOME	SESSO	DATANASCI	PESO	STATURA	LATODOMINANTE	DATATRAUM	INDIRIZZO	CAP	CITTA	PROVINCIA	NAZIONE	TELEFONO1	TELEFONO2	COMPENSA	COMPENSA	COMPENSA	NOTE	MAXNUMEROESERCIZI	MAXNUMEROTEST	TIPOTRAUM	SSN
49	Lydia	Ritter	0	03.01.1953	112	181	0	30.07.2005							Dr. Haus	0	0	0		21	1	Achillessehne	

Abbildung 43: Datenbankeintrag zum fiktiven Patienten "Lydia Ritter".

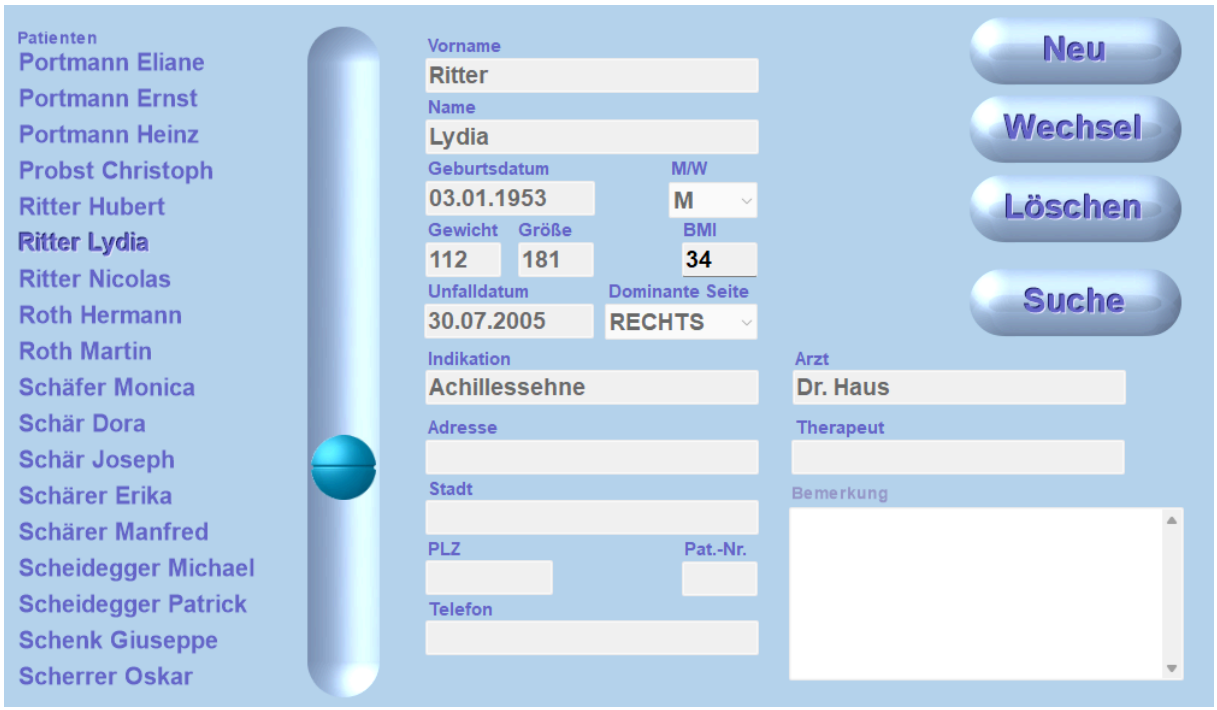


Abbildung 44: Patienteneintrag zum fiktiven Patienten "Lydia Ritter".

Dies ist besonders auffällig, da die Lokalisationsdateien für dieses Feld korrekt sind:

```
Isocine3IT.txt            Isocine3GE.txt
...                       ...
240 = Medico curante      240 = Arzt
...                       ...
```

Auch in diesem Fall erfolgt eine Korrektur im Client, sodass das Feld im Server als „Arzt“ dargestellt wird.

Ein weiterer, kleinerer Fehler betrifft die Patientennummer („Pat.-Nr.“). Dieses Feld wurde ursprünglich zur Speicherung der italienischen Provinz („PROV.“) verwendet:

```
Isocine3IT.txt            Isocine3GE.txt
...                       ...
29 = PROV.                29 = Pat.-Nr.
...                       ...
```

Da weder die Provinz noch die zusätzliche Patientennummer im aktuellen Nutzungskontext relevant sind und auch beide nicht von den Therapeuten geführt werden, hat diese Fehlbezeichnung keinen funktionalen Einfluss und wird ignoriert.

Eine weitere Eigenheit der Isocine3-Datenbank ist, dass die Kombination aus Vor- und Nachname innerhalb der Patientenliste eindeutig sein muss und keine Duplikate zulässig sind.

3.2.7.2 Kommunikation mit Hersteller

Im Rahmen des Projekts fanden während der Inception-Phase mehrere Kommunikationsversuche mit dem Gerätehersteller Easytech statt: ein direkter Versuch des Projektteams sowie zwei weitere Versuche der Rosenklinik über die offizielle Kundenkontaktadresse. Sämtliche Anfragen blieben erfolglos.

Die Kommunikationsversuche verfolgten primär das Ziel, technische Dokumentation zur Isocine3-Software zu erhalten, um deren Verhalten besser zu verstehen und zeitaufwändige Black-Box-Tests zu vermeiden.

Dieses Risiko wurde bereits sehr früh erkannt, weshalb schon im initialen Projektplan ausreichend Zeit für die explorative Analyse des Systemverhaltens sowie insbesondere für die Dekodierung der TRA-Dateien eingeplant wurde.

Die grösste Unsicherheit stellte dabei das Datenbankschreibverhalten von Isocine3 dar, insbesondere die Frage, zu welchen Zeitpunkten die Access-Datenbank beschrieben oder gelesen wird, vor allem aufgrund des Alters der Software.

Zur Untersuchung wurden verschiedene Hilfsskripte entwickelt, beispielsweise ein File Watcher zur Protokollierung von Dateizugriffen oder Skripte zur Überwachung laufender Datenbankzugriffe während der Benutzung von Isocine3.

In den Tests konnte abschliessend geklärt werden, dass die Datenbank zeitgleich gelesen und beschrieben wird. Auf Basis dieser Erkenntnisse fiel die Entscheidung für die Änderungserkennung im Client auf eine einfache Polling-Lösung, die das Änderungsdatum der Datei als Grundlage verwendet. Der Synchronisierungsprozess konnte somit robust gestaltet werden, sodass potenzielles Fehlverhalten vermieden beziehungsweise korrekt behandelt werden kann.

Durch den explorativen Einsatz von Python konnte die Struktur der TRA-Dateien (siehe Kapitel 2.3.2.2.1) in deutlich kürzerer Zeit als ursprünglich geplant entschlüsselt werden.

3.3 Ergebnisse und Erkenntnisse

Das Hauptziel des Projekts, der Rosenklinik eine Plattform zur erleichterten und modernisierten Verwaltung ihrer isokinetischen Messungen bereitzustellen, wurde klar erreicht. Bereits im ersten Demonstrationsmeeting erhielt das System sehr positives Feedback vom Projektpartner. Aufgrund der inkrementellen Vorgehensweise konnten regelmässig neue Versionen präsentiert, kontinuierlich verbessert und gezielt an die Bedürfnisse der Nutzer angepasst werden.

Die Patientenübersicht ermöglicht einen schnellen Überblick über den Verlauf eines Patienten. In Kombination mit den Referenzwerten erlaubt sie den Therapeuten, in kurzer Zeit eine fundierte Einschätzung vorzunehmen. Die PDF-Berichte unterstützen den Austausch mit externen medizinischen Institutionen und internen Abteilungen, während die Exportfunktion das flexible Exportieren ausgewählter Daten und Patienten ermöglicht und damit eine wichtige Grundlage für die vorgesehene Studie bildet.

Das System bietet eine übersichtliche und intuitive Benutzeroberfläche mit umfangreichen Konfigurationsmöglichkeiten sowie einen autarken Synchronisierungsdienst, der die Datenverarbeitung von Isocine3 effizient und robust handhabt und es den Therapeuten ermöglicht, sich auf ihre Kernaufgaben zu konzentrieren.

Die definierten Zusatzziele konnten insgesamt ebenfalls zufriedenstellend erreicht werden. IsoTrack stellt als Analyseplattform eine solide Grundlage für patientenübergreifende Auswertungen bereit und bietet vielfältige Visualisierungs- sowie Funktionen zur Sicherstellung der Datenqualität. Eine umfassende wissenschaftliche Studie allein auf Basis von IsoTrack ist jedoch nicht vollständig abbildbar und würde ein separates Projekt mit vertiefter medizinischer Forschung erfordern.

Die enge Zusammenarbeit mit der Rosenklinik ermöglichte eine zielgerichtete Anforderungsanalyse in einem für nicht-medizinische Fachpersonen komplexen Themengebiet und führte zu präzisen und praxisnahen Abstraktionen.

Damit wurde die Grundlage für ein zukunftsorientiertes System mit hohem Erweiterungspotenzial geschaffen, unterstützt durch saubere Code-Strukturen, umfassende Dokumentation und ein vorausschauendes Architekturkonzept.

IsoTrack kann somit als produktives System betrachtet werden, das bestehende Lücken in Isocine3 schliesst, mit marktüblichen Lösungen vergleichbar ist und die zukünftigen Prozesse der Rosenklinik nachhaltig unterstützt.

3.3.1 Aussicht

Die Zukunft von IsoTrack ist offen; Erweiterungen in alle Richtungen, sowohl vertikal als auch horizontal, sind möglich. Falls sich das Sportzentrum dazu entscheiden würde, weitere Messarten des Prima Plus zu verwenden, wäre dies die erste logische Erweiterung und zugleich eine der einfachsten.

Prinzipiell müssten im Client die Synchronisierungsmechanismen auf weitere Messarten erweitert, entsprechende Controller implementiert sowie Messartfilter eingeführt werden. Die Rosenklinik wird diese Erweiterung jedoch in absehbarer Zeit nicht anstreben.

Wahrscheinlicher ist hingegen eine Weiterentwicklung der Datenanalyseplattform; für das kommende Semester ist bereits eine entsprechende Studienarbeit ausgeschrieben, und interessierten Studierenden wurde bereits Einblick in das GitLab-Repository gewährt. Die Grundlage dafür ist bereits vorhanden, da sämtliche Daten im aktuellen System gespeichert werden.

Diese Arbeit hätte einen stark theoretischen Charakter, da die grösste Herausforderung in der Markt- und Literaturrecherche liegen würde. Informationen zu vergleichbaren industriellen Produkten zu finden

erwies sich bereits in dieser Arbeit als schwierig, da Proprietarität in dieser Branche eine grosse Rolle spielt. Ebenso wäre eine vertiefte Auseinandersetzung mit medizinischer Studiendatenanalyse erforderlich. Dabei können die Therapeuten der Rosenklinik zwar unterstützend wirken, ein erheblicher Teil der Einarbeitung müsste jedoch selbstständig erfolgen und anschliessend ingenieurmässig konzipiert und umgesetzt werden.

Eine kleinere, aber ebenfalls denkbare Erweiterung wäre eine stärkere Nutzung des Potenzials des PDF-Berichts. Darin könnten deutlich mehr, auch für Patientinnen und Patienten verständliche, Informationen dargestellt werden. Ebenso denkbar wäre das Aufheben der Limitierung auf zwei Messungen zugunsten der Darstellung des gesamten Verlaufs, ähnlich wie in der Patientenübersicht, jedoch in detaillierterer Form.

Das langfristige Ziel des Produkts wäre eine Ausweitung auf das Kliniknetzwerk „Swiss Medical Network“. Ein solches Projekt wäre ein Pendant zum Smart-Hospital-Ansatz (Kapitel 1.7.2) und wäre nicht im Rahmen einer einzelnen akademischen Diplomarbeit umsetzbar. Wie in dieser Arbeit mehrfach erläutert, wurde jedoch der notwendige Grundstein für ein solches Vorhaben gelegt, und es besteht die Überzeugung, dass darauf aufbauend ein derart umfangreiches System realisiert werden könnte.

3.4 Zeitmanagement

Durch die in den Kapitel 2.5.1.5 definierten Zeitmanagement-Richtlinien konnten die Zeitaufwände in Jira jederzeit realitätsnah nachgewiesen werden.

Die in den Modulvorgaben vorgegebenen 720 Stunden (360 Stunden pro Student) wurden mit 678 geleisteten Stunden leicht unterschritten (ca. 6 %).

Da trotz dieser Abweichung sämtliche definierten Projektziele erreicht wurden, ist das Ergebnis insgesamt als positiv zu bewerten.

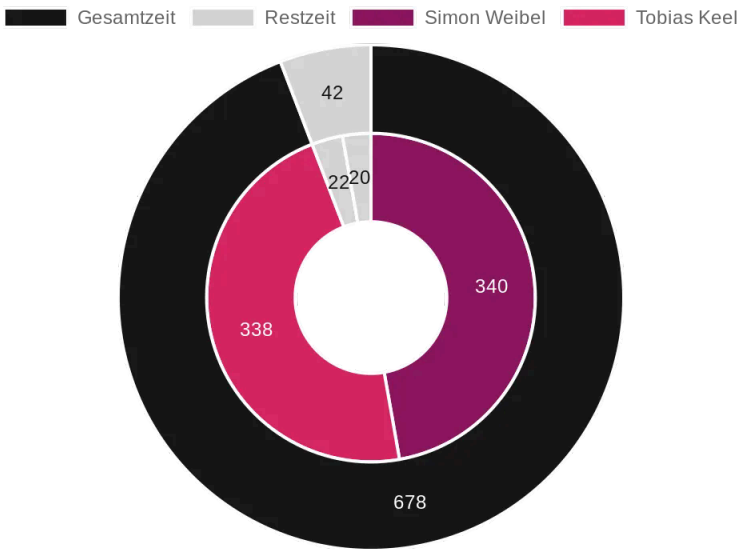


Abbildung 45: Donutdiagramm der Gesamtzeit, der Zeit pro Student sowie der Restzeit, dargestellt mittels quickchart.io @quickchartSrc.

A Anhang

Der Anhang enthält sämtliche Dokumente, die für das Verständnis der Arbeit nicht zwingend erforderlich sind, jedoch einen zusätzlichen Nutzen bieten und teilweise für die Bachelorarbeit obligatorisch sind.

A.1 Modulbeschreibung Bachelorarbeit Informatik

Auszug der Modulbeschreibung von https://studien.ost.ch/[45].

A.1.1 Hauptpunkte

Kürzel	M_BAI21
Durchführungszeitraum	HS/23-FS/27
ECTS-Credits	12
Verantwortliche Person	Prof. Stocker Mirko (STM7)
Standort	Rapperswil-Jona, St.Gallen (Informatik Raster)
Vorausgesetzte Module	Studienarbeit Informatik (M_SAI21, HS/22-FS/27)

Modulbeschreibung Semester Thesis Informatik.

A.1.2 Lernziele

Die Bachelorarbeit gibt den Studierenden die Gelegenheit zu zeigen, wie sie Probleme in der Informatik, insbesondere solchen in den Schwerpunkten des Studiengangs Informatik, unter Anwendung ingenieurmässiger Methoden und anderen Inhalten ihres bisherigen Studiums lösen können. Jede Arbeit enthält einen konzeptionellen, einen theoretischen und einen praktischen Teil.

Die konkreten Lernziele sind:

1. Die Studierenden können Informatik-Problemstellungen identifizieren, analysieren und umsetzen.
2. Die Studierenden können die in der «Studienarbeit Informatik» und den Lehrveranstaltungen «Software Engineering Practices 1 und 2», dem Software Engineering Projekt und anderen Lehrveranstaltungen erlernten Methoden und Praktiken anwenden und situativ anpassen, um den Projektauftrag systematisch und qualitätsgetrieben bearbeiten zu können:
   - Implizite und explizite Anforderungen und Rahmenbedingungen von Stake-/Interestholdern erheben und auf deren Basis innovative Lösungen konzipieren und umsetzen.
   - Den aktuellen Stand der Technik erheben, vorhandene wissenschaftliche Ergebnisse und graue Literatur (wie z. B. Blog-Posts oder Output generativer AI) beurteilen und im Kontext der gegebenen Problemstellung auswählen und anwenden.
   - Arbeitsergebnisse mit Hilfe geeigneter Analyse- und Design-Methoden erarbeiten und diese im Hinblick auf Korrektheit und Praxistauglichkeit initial validieren.
3. Ergebnisse zielgruppengerecht darstellen können, insbesondere mit den Zielgruppen:
   - Betreuungsperson und Mitglieder von Teams mit ähnlichen Aufgabenstellungen und Projektzielen,

[45] „Modulbeschreibung - Bachelor-Arbeit Informatik“.

- Anwendende (User) der Arbeitsergebnisse,
- ggfs. Maintainer/Extender der erstellten Software,
- Fachpublikum im Kontext der Studienarbeit,
- an ICT-Themen interessierte Öffentlichkeit.

Ein wichtiger Arbeitstyp im Informatik-Studium an der OST ist die Konzeption und Realisierung von Software unterschiedlicher Reifegrade (Bsp. Proof of Technology, Proof of Concept, Prototyp, Minimum Viable Product). Andere Arbeitstypen sind in Absprache mit der Betreuungsperson möglich.

Der «Leitfaden für Bachelor- und Studienarbeiten» des Studiengangs Informatik in seiner jeweils aktuellen Version regelt Zweck der Arbeit, Administratives, Themen, Aufgabenstellung, Durchführung und Bewertung im Detail.

A.1.3 Beschreibung

Auszug des Leitfadens für Bachelor- und Studienarbeiten des Studiengangs Informatik.

Die Bachelorarbeit wird in der Regel in Zweierteams bearbeitet und von einem Dozent der Fachhochschule betreut. Sie erstreckt sich über die 16-wöchige Unterrichtszeit des Semesters zuzüglich einer anschliessenden Blockwoche und ist auf einen Gesamtaufwand von 360 Stunden pro Person ausgelegt. Eine Bachelorprüfung muss anschliessend durchgeführt werden.

Die Arbeiten sollen die selbstständige Problemlösungsfähigkeit unter Anwendung ingenieursmässiger Methoden im Bereich Informatik nachweisen. Entsprechend umfasst die Arbeit einen konzeptionellen und theoretischen sowie einen praktischen Teil. Neben fachlichen und technischen Fähigkeiten nach den neuesten Erkenntnissen von Wissenschaft und Praxis sollen die Studierenden auch die Fähigkeit zur Reflexion, Vermittlung und Einordnung der eigenen Leistung in einen grösseren Kontext sowie ein Verantwortungsbewusstsein für wirtschaftliche, soziale und auf Nachhaltigkeit bezogene Fragestellungen nachweisen.

A.2 Benutzeranleitung

Diese Bedienungsanleitung beschreibt Installation und Bedienung des Patientendatensynchronisierungs- und Analyse-Systems **IsoTrack Version 1.0.7** für das isokinetische Messgerät Prima Plus von Easytech.

Material Icons werden für das UI verwendet[46].

A.2.1 Technische Spezifikationen

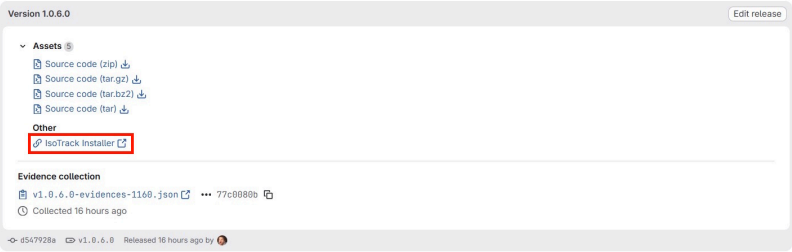
Die Software besteht aus zwei .NET-basierten Hintergrunddiensten: einem Client und einem Server. Die benötigte .NET-Runtime ist in den ausführbaren Dateien enthalten und muss nicht separat installiert werden. IsoTrack läuft ausschliesslich auf Windows-Systemen.

Für den Betrieb wird eine korrekt installierte Instanz der Software Isocine3 sowie die Microsoft 365 Access Runtime vorausgesetzt. Falls Isocine3 installiert ist, ist die Access Runtime in der Regel bereits vorhanden. Dies ist anhand der Unterstützung für Access-Dateien (.mdb, .accdb) erkennbar. [47]. Falls die Runtime nicht installiert ist, kann sie hier bezogen werden.

Zusätzlich muss die Umgebungsvariable *ISOTRACK_DBPATH* definiert sein. Diese muss auf die Datei *isocine.mdb* der Isocine3-Installation verweisen. Es wird empfohlen, diese Konfiguration automatisch durch das Installationspaket vornehmen zu lassen.

A.2.2 Installation

IsoTrack stellt ein Installationspaket bereit, das die Installation und Deinstallation erheblich vereinfacht. Das Installationspaket kann hier heruntergeladen werden. („IsoTrack Installer“ für die neuste Version)



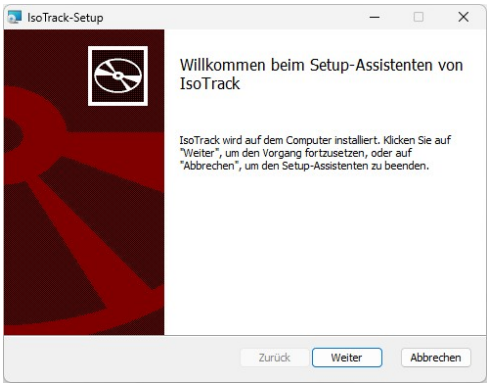
Nachdem das Installationspaket (.msi) heruntergeladen wurde, kann dieses ausgeführt werden. Unter Umständen verhindert das installierte Antivirenprogramm die Ausführung des Installers.

[46] Google, „Google Material Symbols“, https://fonts.google.com/icons.

[47] Microsoft Corporation,*.mdb icon (2000-03).svg*, o. J., https://commons.wikimedia.org/wiki/File:.mdb_icon_(2000-03).svg.

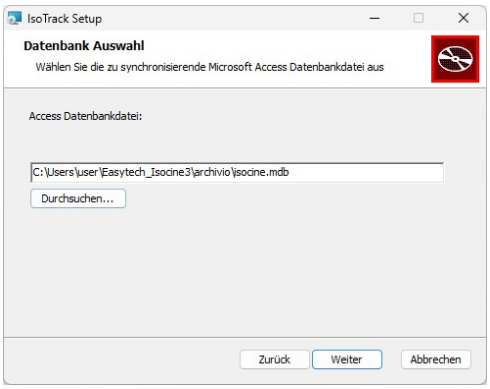
Bei Verwendung von Windows Defender kann die Ausführung über Weitere Informationen und anschliessend **Trotzdem ausführen** freigegeben werden. Bei anderen Antivirenprogrammen ist die Freigabe der .msi-Datei gegebenenfalls manuell zu konfigurieren.

Möglicherweise werden zusätzlich Administratorrechte angefordert; diese sind zu bestätigen.



Die einzige zwingende Benutzereingabe betrifft die Auswahl der Datenbank:

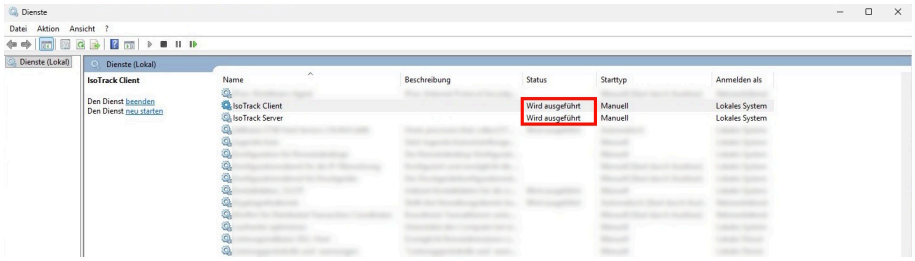
Hier muss die Datei *isocine.mdb* der Isocine3-Installation angegeben werden. Hierfür sollte die Funktion „Durchsuchen…“ verwendet werden. Die Datei befindet sich üblicherweise im Ordner: *Easytech_Isocine3\archivio*.



Nach Abschluss der Installation ist zu überprüfen, ob die Hintergrunddienste IsoTrack Client und IsoTrack Server erfolgreich gestartet wurden.

Die Überprüfung erfolgt über die Windows-Diensteverwaltung:

- Windows-Suche öffnen, „Dienste“ eingeben und öffnen.
- In der Liste nach IsoTrack Client und IsoTrack Server suchen.



Beide Dienste müssen den Status „Wird ausgeführt“ aufweisen.

Nach erfolgreicher Installation wird auf dem Desktop eine Verknüpfung mit dem Namen „IsoTrack“ erstellt. Diese öffnet die Anwendung im Standardbrowser. Es wird empfohlen, zusätzlich ein Lesezeichen anzulegen. IsoTrack kann alternativ direkt über folgende URL aufgerufen werden: *„http://localhost:5115/“*.

Der initiale Synchronisierungsprozess beginnt unmittelbar nach der Installation und kann einige Minuten in Anspruch nehmen. Es wird empfohlen, dem Client hierfür etwa 2 Minuten Zeit zu geben. Ein Zugriff auf die Anwendung ist währenddessen möglich, jedoch sind möglicherweise noch nicht alle Patienten- und Messdaten verfügbar.

Nach Abschluss des Synchronisierungsprozesses ist IsoTrack vollständig betriebsbereit.

A.2.2.1 Update

Zur Aktualisierung von IsoTrack kann die aktuelle Version von der Release-Seite heruntergeladen und wie im Kapitel „Installation“ beschrieben installiert werden.

Bei grösseren Versionssprüngen (Änderung der Haupt- oder Nebenversion, z. B. von 1.0.x auf 1.1.x) kann es erforderlich sein, die lokale Datenbank neu zu synchronisieren. Falls nach dem Update keine Daten mehr angezeigt werden, kann die Synchronisierung durch folgende Schritte zurückgesetzt werden:

1. IsoTrackClient und IsoTrackServer in der Windows-Diensteverwaltung beenden.
2. Die Inhalte im Verzeichnis `C:\ProgramData\IsoTrack` löschen.
3. Beide Dienste erneut starten.

Beim nächsten Start wird daraufhin eine vollständige Synchronisierung aller Patienten- und Messdaten durchgeführt.

Dabei gehen jedoch alle ausschliesslich in IsoTrack gespeicherten Informationen verloren. Dies betrifft insbesondere Operationsdaten sowie manuell erfasste Messindikationen.

A.2.3 Bedienung

Der Server und Client sind vollautomatisch und starten automatisch beim Hochfahren des PCs und fahren sich auch automatisch herunter beim Herunterfahren des PCs.

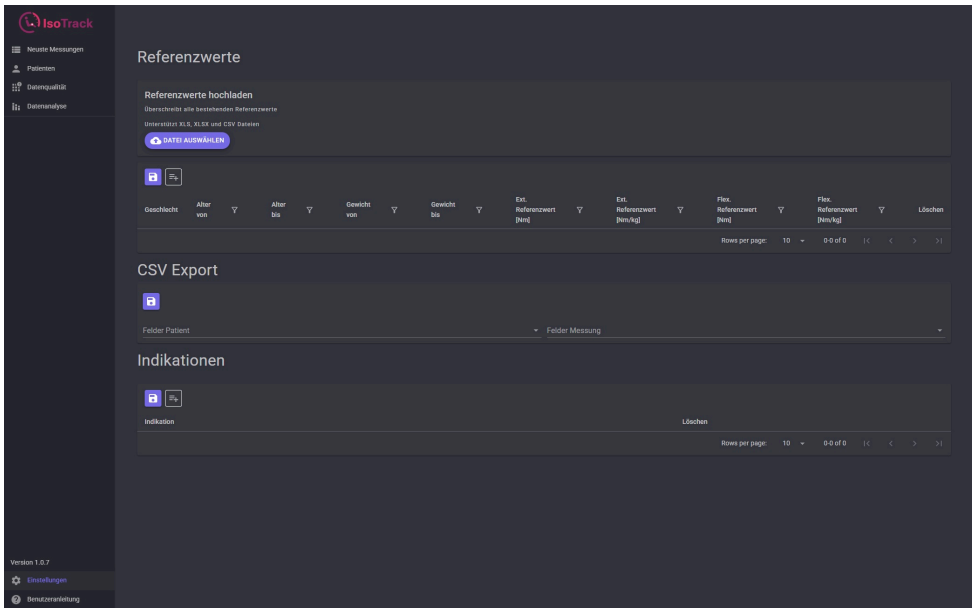
A.2.3.1 Datensynchronisierung

Der Synchronisierungsprozess wird vom Client automatisch alle 20 Sekunden ausgeführt. Nach einer Änderung kann es bis zu 20 Sekunden dauern, bis diese in IsoTrack sichtbar ist. Zur Aktualisierung der Anzeige muss der Browser manuell neu geladen werden über **F5**, **Ctrl + R** oder ⟳.

Die Zusatzangaben in IsoTrack zu Patienten- und Messdaten werden ausschliesslich in IsoTrack gespeichert und sind in Isocine3 nicht sichtbar.

A.2.3.2 Einstellungen

In den Einstellungen können Referenzwerte, Exportfelder und Indikationen verwaltet werden.



Jede Konfiguration muss separat über den jeweiligen 💾-Button gespeichert werden.

A.2.3.2.1 Referenzwerte

Referenzwerte können für Extension und Flexion sowie als gewichtsnormalisierte Werte (Nm/kg) erfasst werden. Die Zuordnung erfolgt anhand von Geschlecht sowie Alters- und Gewichtsbereichen.

Neue Einträge können manuell über das Hinzufügen eines leeren Datensatzes mittels ≡+ erstellt werden. Anschliessend werden die entsprechenden Werte und Faktoren erfasst.

Alternativ besteht die Möglichkeit, Referenzwerte über eine CSV- oder Excel-Datei zu importieren. Dies erfolgt über den Button **DATEI AUSWÄHLEN**. Die Datei muss dabei folgendem Schema entsprechen:

Sex,AgeFrom,AgeTo,WeightFrom,WeightTo,ExtReferenceValue,FlexReferenceValue,ExtReferenceValueByWeight,FlexReferenceValueByWeight

Für das Feld *Sex* ist ein binärer Wert zu verwenden, wobei 0 männlich und 1 weiblich entspricht.

A.2.3.2.2 Export

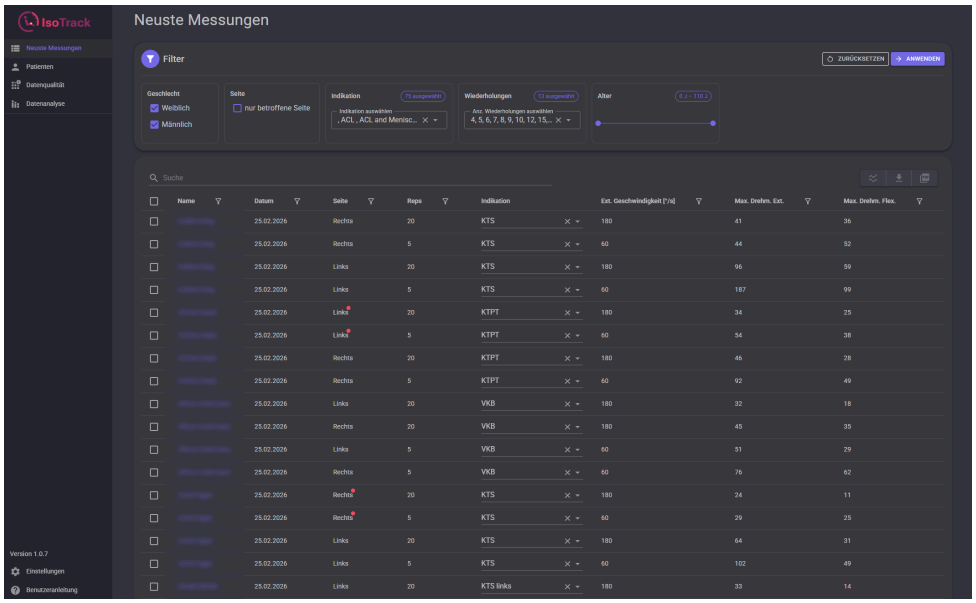
Hier können in zwei Dropdown-Menüs die Felder für Patienten- und Messdaten für den Excel/CSV-Export konfiguriert werden. Zur Auswahl stehen sämtliche verfügbaren Felder.

A.2.3.2.3 Indikationen

Die hier erfassten Indikationen stehen bei der Erfassung von Messungen zur Auswahl bereit.

A.2.3.3 Neuste Messungen

Beim Öffnen der Website gelangt man standardmässig auf die Seite der neusten Messungen. Dort werden alle Messungen in absteigender Reihenfolge nach Zeitpunkt angezeigt.



Ein roter Punkt bei einer Messung markiert die betroffene Seite des Patienten. Diese Information wird anhand der hinterlegten Operationsdaten ermittelt und steht daher erst nach der Angabe dieser zur Verfügung.

Über die Filtermöglichkeiten können Messungen nach Geschlecht, betroffener Seite, Indikation, Anzahl Wiederholungen sowie Alter gefiltert werden. Die Filter werden über „→ ANWENDEN“ angewendet und können über ⟳ zurückgesetzt werden.

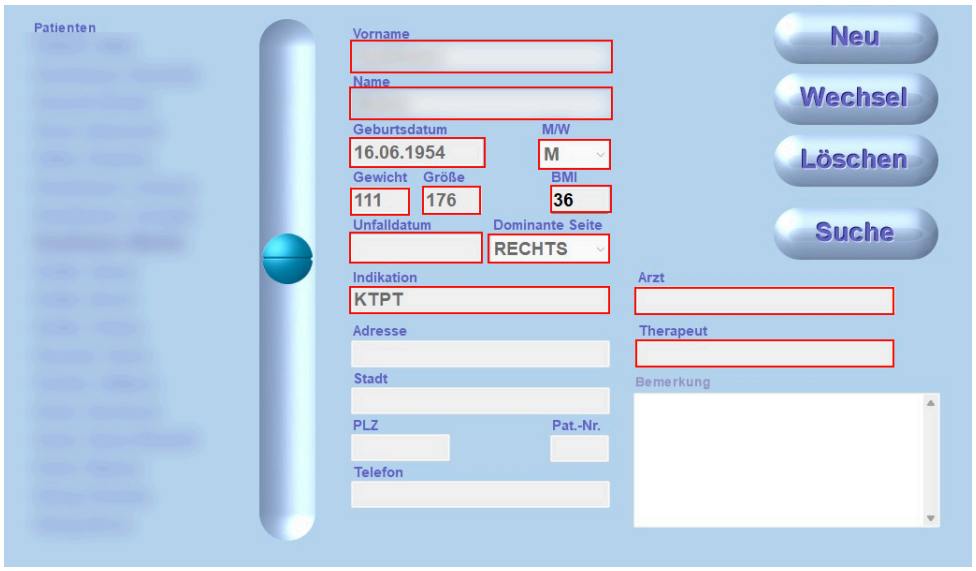
Zusätzlich können die Spalten beliebig sortiert werden. Beim Hovern über einer Spaltenüberschrift erscheint ein Pfeil, über den die Sortierung aktiviert wird. Ein erneutes Klicken wechselt zwischen aufsteigender und absteigender Sortierung.

Es können zudem mehrstufige Sortierungen definiert werden. Wird dabei die Ctrl</kbd>-Taste gedrückt gehalten, wird eine zusätzliche Sortierung zur bestehenden Sortierlogik hinzugefügt. Einzelne Sortierungen können über ⋮ im Spaltenmenü durch die Option „Unsort“ entfernt werden.

Über die Spaltenüberschrift können Felder zudem über ▼ manuell gefiltert oder über ⋮ mittels „Group“ gruppiert werden.

A.2.3.4 Patientenverwaltung

Für eine konsistente Datenanalyse ist eine vollständige und strukturierte Erfassung der Patientendaten erforderlich. Die in der folgenden Abbildung rot markierten Felder sollten, sofern verfügbar, ausgefüllt werden:



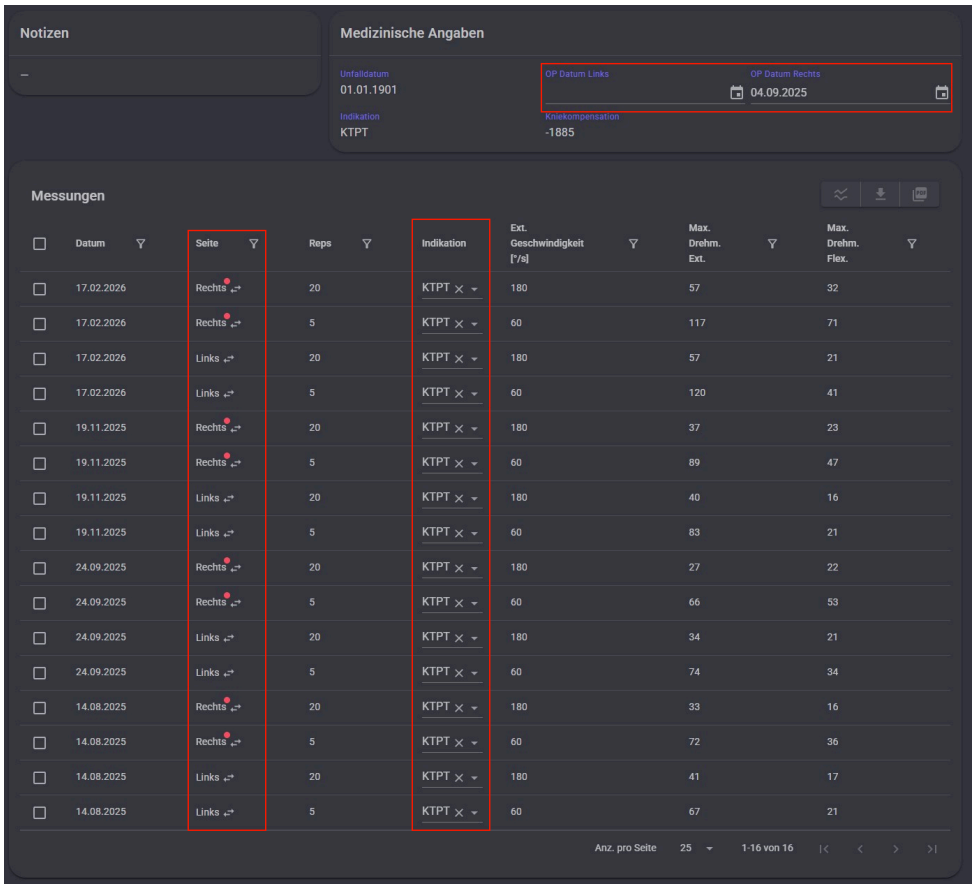
Die Indikation muss konsistent angegeben werden, da keine automatische Validierung erfolgt. Die Felder „Arzt“ und „Therapeut“ werden aktuell nicht für Auswertungen verwendet und sind optional aber dennoch empfohlen.

Der empfohlene Ablauf zur Patientenverwaltung ist:

1. Patient in Isocine3 erfassen
2. Messung durchführen
3. IsoTrack öffnen oder neu laden
4. Patient in IsoTrack suchen
5. Operationsdaten erfassen (falls vorhanden)
6. Indikation der Messung anpassen (falls nötig)

Operationsdaten können für beide Beine separat erfasst werden. Die Indikation kann manuell pro Messung angepasst werden, sofern sie von der in Isocine3 angegebenen Indikation abweichen sollte. Falls gewünscht, kann der Patient durch Aktivieren der Option „Aus Datenanalyse ausschliessen“ von der Auswertung ausgeschlossen werden.

Falls bei einer Messung die falsche Seite angegeben wurde, kann diese auf der Patientenseite über ↩ korrigiert werden. Die Änderung wird jedoch nur in IsoTrack gespeichert und nicht in Isocine3 übernommen.



Diese Vorgehensweise stellt eine konsistente Datenbasis für die weitere Auswertung sicher.

A.2.3.5 Patientenanalyse

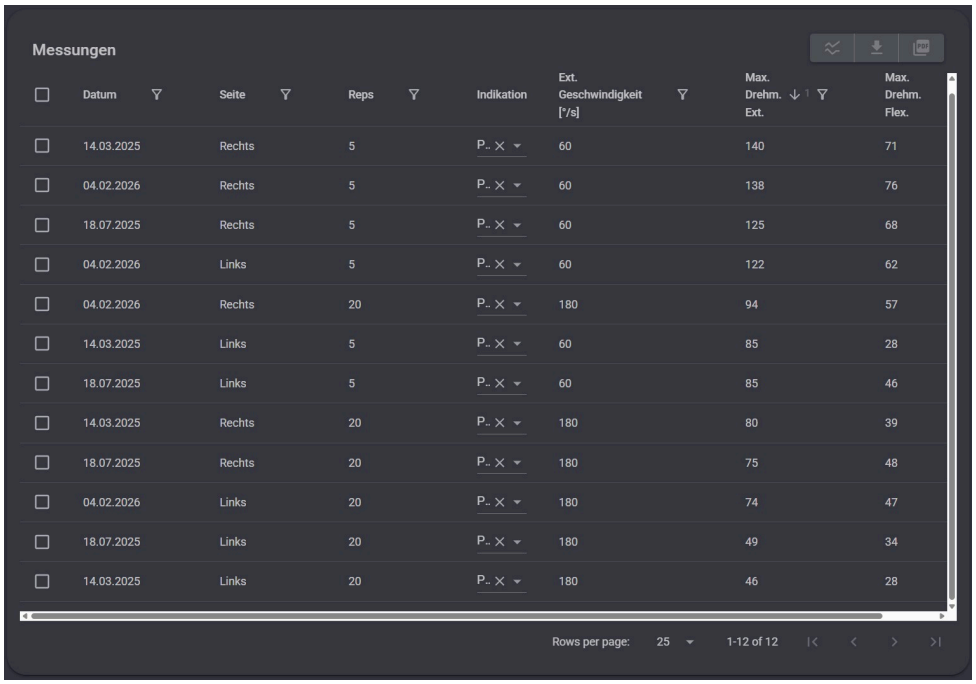
Auf der Patientenseite können erste Analysen pro Patient durchgeführt werden. Über die Suchfunktion lassen sich alle Patienten aufrufen.

Im zeitlichen Verlaufsgraph werden alle Messungen eines Patienten chronologisch dargestellt. Dabei werden folgende Informationen visualisiert:

- zugehörige Referenzwerte (umschaltbar zwischen normal und gewichtsbasierter Werte)
- Operationsdaten (sofern angegeben)

- maximale Drehmomente der Extension je Messung

In der Legende können Gruppen nach Seite und Wiederholungsanzahl durch Anklicken ein- und ausgeblendet werden.



Messungen

Datum	Seite	Reps	Indikation	Ext. Geschwindigkeit [°/s]	Max. Drehm. Ext.	Max. Drehm. Flex.
14.03.2025	Rechts	5	P..	60	140	71
04.02.2026	Rechts	5	P..	60	138	76
18.07.2025	Rechts	5	P..	60	125	68
04.02.2026	Links	5	P..	60	122	62
04.02.2026	Rechts	20	P..	180	94	57
14.03.2025	Links	5	P..	60	85	28
18.07.2025	Links	5	P..	60	85	46
14.03.2025	Rechts	20	P..	180	80	39
18.07.2025	Rechts	20	P..	180	75	48
04.02.2026	Links	20	P..	180	74	47
18.07.2025	Links	20	P..	180	49	34
14.03.2025	Links	20	P..	180	46	28

Rows per page: 25 1-12 of 12

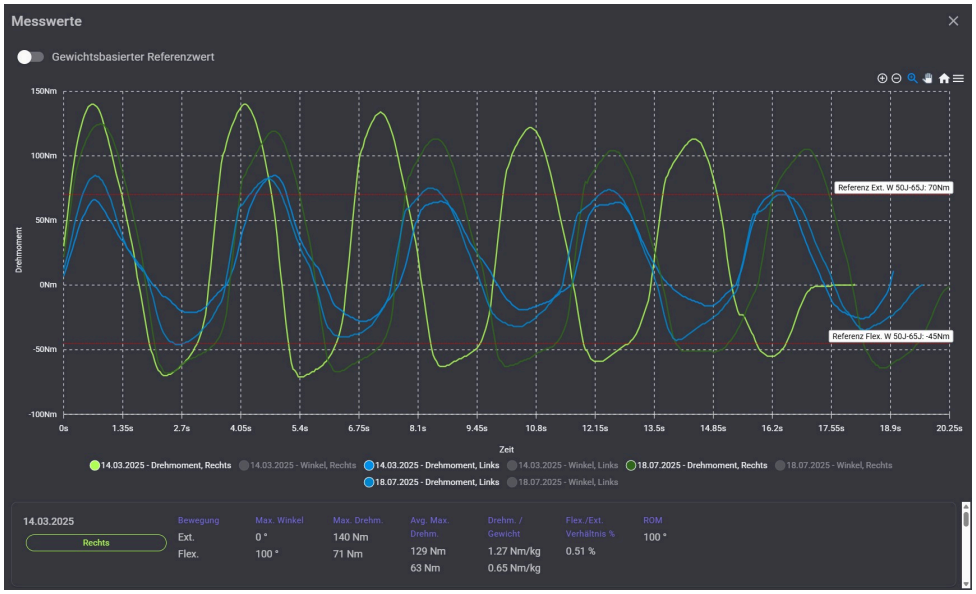
Für jede Messung wird standardmässig die beim Patienten hinterlegte Indikation verwendet. Falls mehrere Indikationen relevant sind, kann zusätzlich pro Messung eine abweichende Indikation in IsoTrack erfasst werden. Zur Auswahl stehen alle in den Einstellungen definierten Indikationen.

Messungen können anhand der verfügbaren Spalten (mit Ausnahme der Indikation) sortiert und gefiltert werden. Zusätzlich können Messungen für die Visualisierungsfunktionen ausgewählt werden.

A.2.3.5.1 Visualisierungsfunktionen

Für Messungen stehen drei Visualisierungsfunktionen zur Verfügung:

Über werden ausgewählte Messungen überlagert in einem Graphen dargestellt.



Auch hier können Messgruppen durch Anklicken ein- und ausgeblendet werden. Zusätzlich steht der Referenzwert für Flexion als separate Kurve zur Verfügung. Die referenzwerte lassen sich auch wieder auf gewichtsbasierte Werte umschalten.

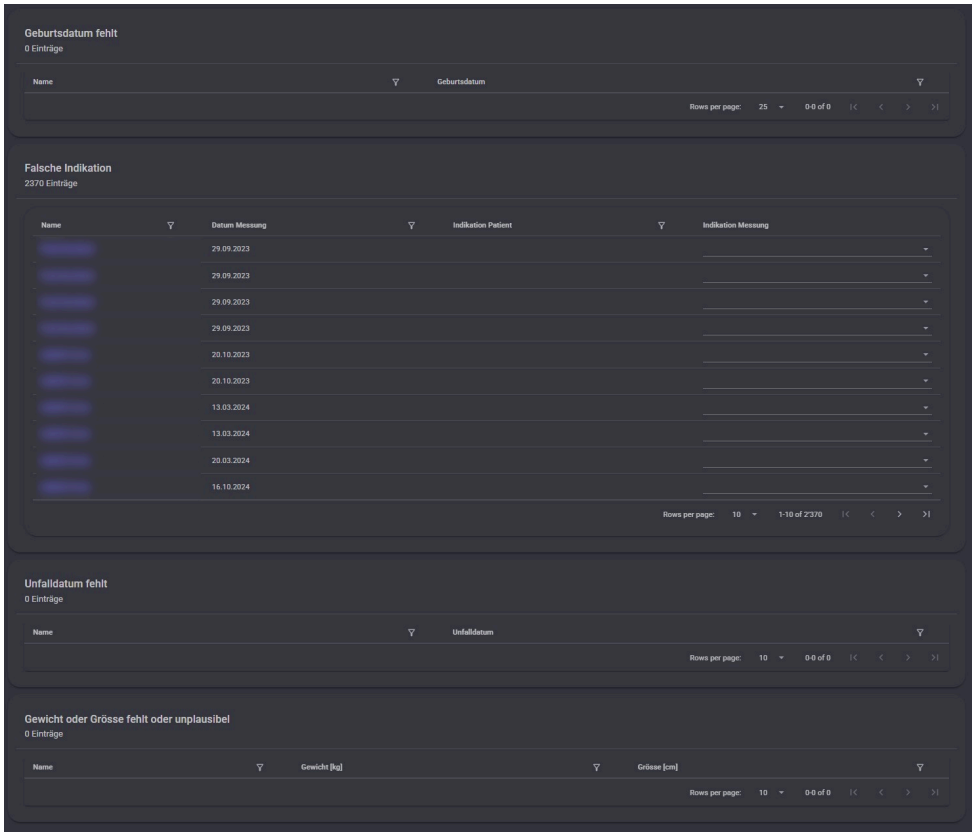
Unterhalb des Graphen werden die Eckdaten der jeweiligen Übungen aufgelistet.

Über ⬇ können ausgewählte Messungen als CSV- oder Exceldatei exportiert werden. Dabei werden alle in den Einstellungen definierten Felder verwendet.

Über 🖶 kann ein PDF-Bericht für zwei ausgewählte Messungen erstellt werden. Dabei muss eine Messung der linken und eine der rechten Seite zugeordnet sein. Beidseitige Messungen werden in einem PDF nicht unterstützt.

Die Funktionen ermöglichen eine strukturierte Patientenanalyse sowie die Weitergabe der Ergebnisse an medizinisches Fachpersonal.

A.2.3.6 Datenqualität



Die Datenqualitätsseite hilft Einblicke in unvollständige oder inkorrekte Datensätze zu geben.

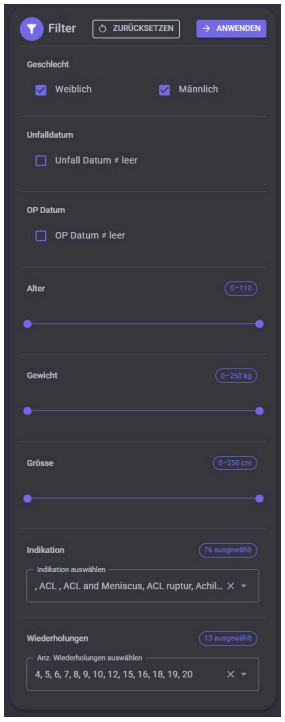
Es werden folgende Datenfehler angemerkt:

- Fehlendes Geburtsdatum
- Fehlende Indikation einer Messung (meistens als Resultat aus dem Fehlen der Indikation des Patienten)
- Fehlendes Unfalldatum
- Fehlende oder unplausibeles Gewicht und Grösse

Alle Änderungen, mit Ausnahme der fehlenden Indikation bei einer Messung, müssen in Isocine3 nachgereicht werden. Es empfiehlt sich also beide Apps offen zu halten und die Änderungen durch schnelles Wechseln über **Alt + Tab** ⇆ zu korrigieren.

Messindikationen können direkt auf dieser Seite nachgetragen werden. Wird eine Patientenindikation nachträglich in Isocine3 ergänzt, so wird diese automatisch auf alle bisher leeren Messindikationen dieses Patienten übernommen.

A.2.3.7 Datenanalyse



Die Datenanalyseseite bietet umfangreiche Visualisierungs- und Filtermöglichkeiten zur patientenübergreifenden Auswertung der Messdaten.

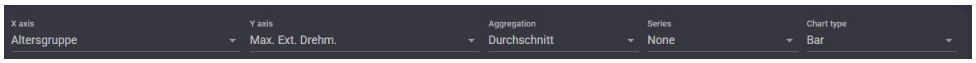
Die Konfiguration der Filter erfolgt über den entsprechenden Reiter.

Filter werden erst nach Betätigen des „→ ANWENDEN“-Buttons übernommen.

Für die Visualisierung stehen verschiedene analytische Darstellungsmöglichkeiten zur Verfügung. Die X-Achse kann anhand von Patienten-, Mess- oder Bewegungsmerkmalen konfiguriert werden, beispielsweise Alter, Geschlecht, Wiederholungsanzahl oder maximale Drehmomente. Die Y-Achse unterstützt verschiedene Mess-, Körper- und Zeitparameter, darunter Messergebnisse, Körpergewicht, Körpergrösse oder den zeitlichen Abstand zur Operation.

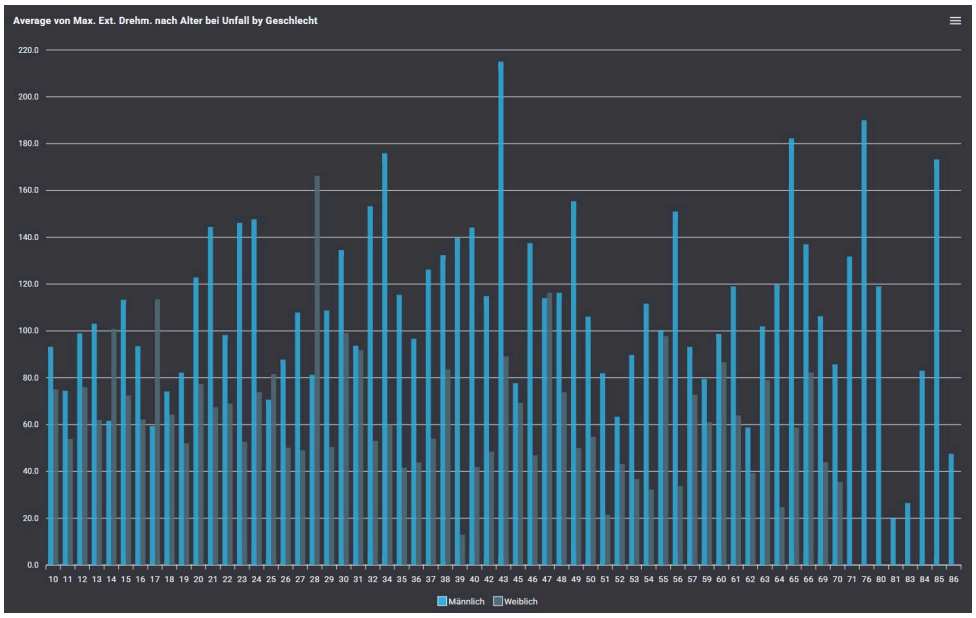
Zusätzlich kann die Aggregation der Daten über Minimum, Maximum, Durchschnitt oder Anzahl definiert werden. Über „Series“ lassen sich Daten innerhalb derselben Visualisierung anhand eines zusätzlichen Kriteriums unterteilen.

Abschliessend kann zwischen Balken- und Streudiagrammen als Darstellungsform gewählt werden.



Im resultierenden Diagramm können einzelne Werte durch Hovern eingesehen werden.

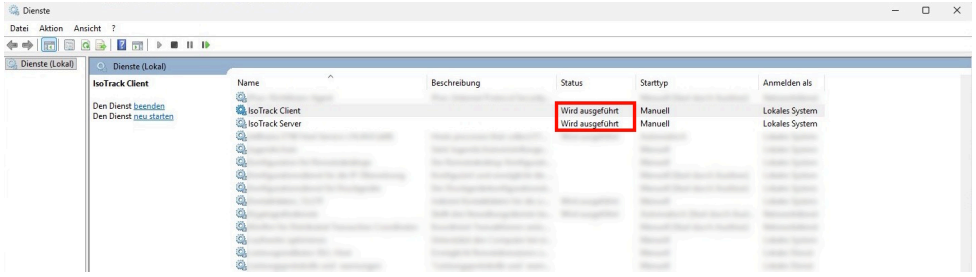
Zusätzlich kann das Diagramm über ≡ als SVG-, PNG- oder CSV-Datei exportiert werden.



A.2.4 Fehlerbehandlung

Falls dem Nutzer auffällt, dass Änderungen in Isocine3 nicht mehr in IsoTrack übernommen werden oder die Website nicht mehr aufgerufen werden kann, bietet dieses Kapitel erste Hilfe zur Fehlerbehebung.

In jedem Fall muss zunächst festgestellt werden, welche Komponente nicht korrekt funktioniert. Dazu ist, wie bei der Überprüfung nach der Installation, der Status der Dienste zu kontrollieren.



Falls einer der Dienste nicht läuft, sollte versucht werden, diesen manuell neu zu starten. Schlägt dies fehl, empfiehlt sich eine Neuinstallation.

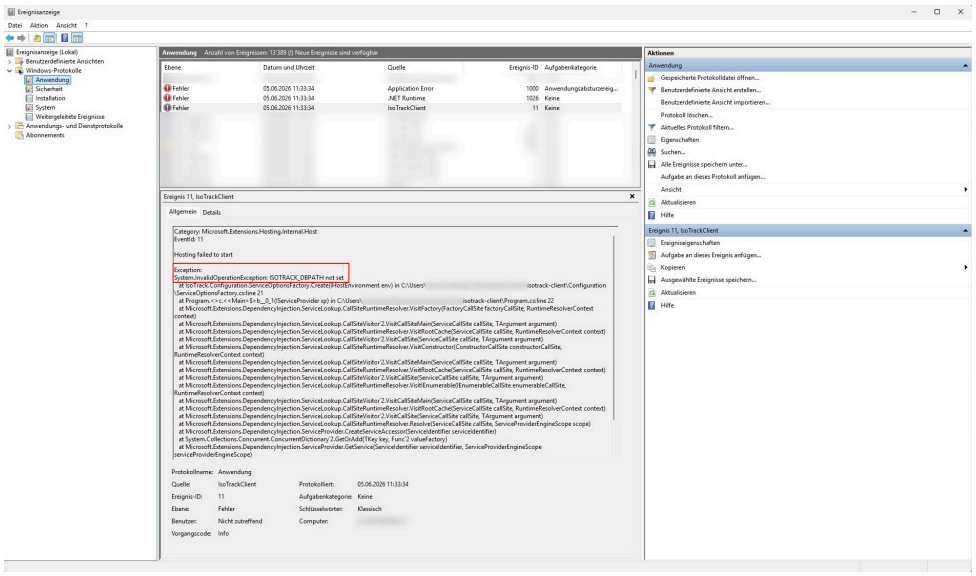
Hierfür wird der ursprüngliche Installer benötigt, welcher hier heruntergeladen werden kann.

Achtung: Es ist sicherzustellen, dass der Installer der gleichen Version entspricht wie die aktuell installierte Version. Die aktuelle Version ist im Kopf dieser Anleitung ersichtlich.

Mit dem Installer kann IsoTrack anschliessend repariert werden. Nach Abschluss der Reparatur sollte der Status der Dienste erneut überprüft werden.

Falls einer oder beide Dienste weiterhin nicht starten, können detaillierte Fehlermeldungen in den nachfolgenden Unterkapiteln überprüft werden. Dabei kann in der Windows-Ereignisanzeige unter „Ereignisanzeige (Lokal) ⟶ Windows-Protokolle ⟶ Anwendung“ nach entsprechenden Fehlern gesucht werden. Als Quelle sind dabei „IsoTrackClient“ oder „IsoTrackServer“ zu berücksichtigen.

A.2.4.1 ISOTRACK_DBPATH nicht gesetzt

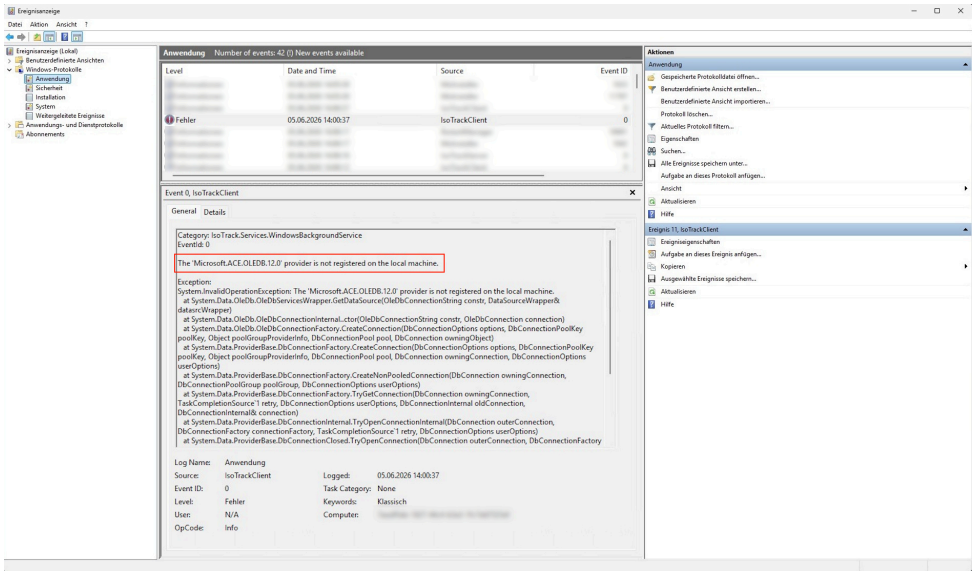


Falls dieser Fehler nach dem Installationsversuch auftritt, muss die Umgebungsvariable manuell gesetzt werden. Dies erfolgt über Windows-Suche ⟶ Systemumgebungsvariablen bearbeiten ⟶ Umgebungsvariablen… ⟶ Neu… (unter Systemvariablen).

Als Name ist ISOTRACK_DBPATH anzugeben. Als Wert kann über „durchsuchen…“ die isocine.mdb ausgewählt werden (siehe Installation).

Anschliessend können alle Fenster mit OK geschlossen werden. Nach einem Computerneustart sowie dem erneuten Start des IsoTrackClient über die Dienste sollte der Fehler behoben sein.

A.2.4.2 Fehlender Access Treiber

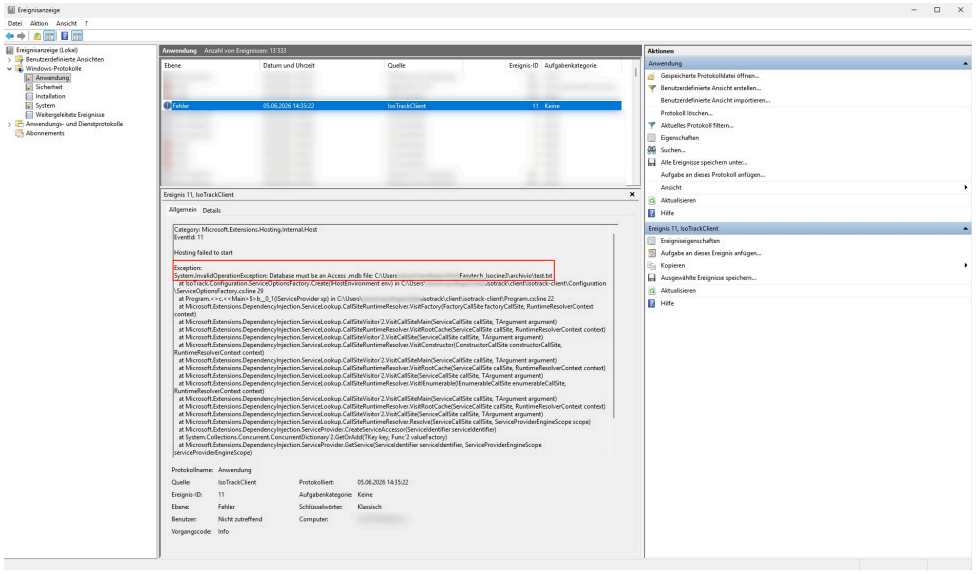


Dieser Fehler weist darauf hin, dass die Microsoft 365 Access Runtime beziehungsweise die darin enthaltenen Access-Datenbanktreiber nicht installiert sind.

Die Runtime ist im Technische Spezifikationen Kapitel verlinkt und kann alternativ auch direkt hier heruntergeladen werden.

Nach erfolgreicher Installation kann der IsoTrackClient über das Dienste-Fenster gestartet werden.

A.2.4.3 Inkorrekte Access-Datenbank



Dieser Fehler weist darauf hin, dass die konfigurierte Access-Datenbankdatei ungültig ist. Falls die angegebene Datei nicht existiert, wird stattdessen die Fehlermeldung `Database file does not exist` angezeigt.

In diesem Fall kann der Installer erneut ausgeführt und die Option **Reparieren** gewählt werden. Während des Reparaturvorgangs ist anschliessend die korrekte Access-Datenbankdatei auszuwählen.

Alternativ kann die Umgebungsvariable manuell angepasst werden, wie im Abschnitt „ISOTRACK_DBPATH nicht gesetzt“ beschrieben.

B Glossar

IT-Begriffe

API (Application Programming Interface)	Programmierschnittstelle, die die Kommunikation zwischen verschiedenen Softwarekomponenten über definierte Methoden und Datenstrukturen ermöglicht.
CI/CD	Sammelbegriff für Continuous Integration und Continuous Deployment/Delivery. Automatisierte Prozesse zum Bauen, Testen und Bereitstellen von Software.
DTO Data Transfer Object	Einfaches Datenobjekt ohne Geschäftslogik, das ausschliesslich zur strukturierten Datenübertragung zwischen Softwarekomponenten verwendet wird.
Endianness	Beschreibt die Reihenfolge der Bytes bei der Speicherung oder Übertragung mehrbyteiger Datenwerte. Es wird zwischen Little-Endian (Least Significant Byte zuerst) und Big-Endian (Most Significant Byte zuerst) unterschieden.
HTTP (Hypertext Transfer Protocol)	Netzwerkprotokoll zur Übertragung von Daten zwischen zwei Kommunikationspartnern auf Basis von Anfragen und Antworten.
IDE (Integrated Development Environment)	"Entwicklungsumgebung", Softwareplattform mit Werkzeugen zur Entwicklung von Software.
LTS (Long-Term Support)	Version einer Software mit langfristiger Unterstützung für Fehlerbehebungen und Sicherheitsupdates, ausgelegt auf Stabilität statt neue Funktionen.
ORM (Object-Relational Mapping)	Programmierparadigma bzw. Technik, die relationale Datenbanken über objektorientierte Strukturen abbildet und den Datenbankzugriff durch Objekte in der Programmiersprache ermöglicht.
UI (User Interface)	Benutzeroberfläche eines Softwaresystems, über die Nutzer mit der Anwendung interagieren können, insbesondere zur Eingabe von Daten und zur Darstellung von Informationen.
Wrapper	Softwarekomponente, die eine bestehende Bibliothek oder Schnittstelle kapselt und deren Nutzung über eine vereinfachte oder angepasste Schnittstelle in einer anderen Umgebung ermöglicht.
XML (Extensible Markup Language)	Textbasiertes Format zur strukturierten Darstellung und Speicherung von Daten, das sowohl für den Datenaustausch zwischen Systemen als auch zur Konfiguration von Software verwendet wird.

Microsoft Technologien

.NET	Softwareplattform von Microsoft zur Entwicklung und Ausführung von Anwendungen, insbesondere mit Sprachen wie C#.
ApexCharts	JavaScript-basierte Bibliothek zur Erstellung interaktiver und anpassbarer Diagramme zur Visualisierung von Daten in Webanwendungen. Für die Integration in Blazor-Anwendungen existiert ein entsprechender Wrapper, der die Nutzung der Bibliothek innerhalb von C#-basierten Komponenten ermöglicht.
ASP.NET	Web-Framework von .NET zur Erstellung serverseitiger Webanwendungen und APIs auf Basis des HTTP-Protokolls.
Blazor	UI-Framework von .NET zur Entwicklung interaktiver, komponentenbasierter Weboberflächen mit C#. Es ermöglicht die Erstellung von Webanwendungen ohne den direkten Einsatz von JavaScript als Hauptsprache.
COM (Component Object Model)	Von Microsoft entwickeltes Schnittstellenmodell zur Interaktion zwischen Softwarekomponenten, das die sprach- und plattformübergreifende Nutzung von Funktionen innerhalb von Windows ermöglicht.
EF (Entity Framework)	Object-Relational Mapper (ORM) von Microsoft für .NET, der den Zugriff auf Datenbanken über C#-Objekte ermöglicht und die Datenbankverwaltung im Code vereinfacht.
MudBlazor	Open-Source UI-Komponentenbibliothek für Blazor im Material-Design, die vorgefertigte Benutzeroberflächen-Elemente bereitstellt und die Entwicklung von Weboberflächen innerhalb von .NET-Anwendungen vereinfacht.
OLE DB (Object Linking and Embedding, Database)	Von Microsoft entwickelte Datenbank-API zur einheitlichen Anbindung unterschiedlicher Datenquellen über sogenannte Provider, unabhängig vom konkreten Speicherformat.
Windows Installerdatei (.msi)	Installationspaket für Windows-Programme, das die automatische Installation und Deinstallation einer Software ermöglicht.
WiX Toolset (Windows Installer XML)	Open-Source-Framework zur Erstellung von Windows-Installationspaketen (.msi) auf Basis des Microsoft Installer (MSI), das die deklarative Definition von Installationsprozessen mittels XML ermöglicht.

Versionsverwaltung

Branch	Abzweigung in einem Versionskontrollsystem wie Git, die es ermöglicht, unabhängig vom Hauptentwicklungszweig (Main) parallel Änderungen oder neue Funktionen zu entwickeln.
Feature Branch	Branch, der zur isolierten Umsetzung einer konkreten User Story verwendet wird und nach Abschluss der Entwicklung in den Hauptzweig integriert wird.

Git	Verteiltes Versionskontrollsystem zur Nachverfolgung und Verwaltung von Änderungen an Dateien, insbesondere Quellcode.
GitLab	Webbasierte Plattform zur Versionsverwaltung mit Git, die Funktionen für Zusammenarbeit, Codeverwaltung und CI/CD-Pipelines bereitstellt.
Merge	Vorgang in Versionskontrollsystemen wie Git, bei dem Änderungen aus einem Branch in einen anderen Branch (z. B. den Main-Branch) übernommen und zusammengeführt werden.

Medizinische Begriffe

ACL / VKB (Anterior Cruciate Ligament / Vorderes Kreuzband)	Wichtiges stabilisierendes Band im Kniegelenk, das vor allem das Vorwärtsgleiten des Schienbeins und Rotationsbewegungen begrenzt.
Extension	Streckbewegung eines Gelenks mit Vergrösserung des Gelenkwinkels.
Extension	Bewegung eines Gelenks, bei der der Gelenkwinkel verkleinert wird (Beugung).
Indikation	Medizinischer Grund oder Anlass für eine Untersuchung oder Behandlung.
Isokinetisch	Trainings- bzw. Messmethode, bei der die Bewegung mit konstanter Geschwindigkeit ausgeführt wird. Der Widerstand wird dabei während der Bewegung kontinuierlich angepasst.
Kontraktion	Aktive Verkürzung oder Spannungsentwicklung eines Muskels.
KTP / KTPT (Knietotalprothese)	Vollständiger Ersatz des Kniegelenks durch ein Implantat zur Wiederherstellung von Funktion und Schmerzreduktion, meist bei fortgeschrittener Arthrose.

C Tabellenverzeichnis

Tabelle 1 Epics. 12
Tabelle 2 Datenverwaltung-User Stories absteigend nach Priorität sortiert. 13
Tabelle 3 Datenvisualisierung-User Stories absteigend nach Priorität sortiert. 13
Tabelle 4 NFA 01 - Vertraulichkeit 14
Tabelle 5 NFA 02 - Benutzerunterstützung 14
Tabelle 6 NFA 03 - Angemessenheit Erkennbarkeit / Selbstbeschreibbarkeit. 15
Tabelle 7 NFA 04 - Installierbarkeit 15
Tabelle 8 Tabellen der Access-Datenbank. 25
Tabelle 9 API-Endpunkte von IsoTrack. 32
Tabelle 10 ADR 01. 33
Tabelle 11 ADR 02. 33
Tabelle 12 ADR 03. 34
Tabelle 13 ADR 04. 34
Tabelle 14 ADR 05. 34
Tabelle 15 ADR 06. 35
Tabelle 16 Testprotokoll der funktionalen Anforderungen 56
Tabelle 17 Testprotokoll der nicht-funktionalen Anforderungen 56
Tabelle 18 RUP Phasen. 58
Tabelle 19 Übersicht der Teamrollen mitsamt Beschreibung. 60
Tabelle 20 Liste aller Meetings. 61
Tabelle 21 Risikomatrix Sprint 1. 63
Tabelle 22 Risikomatrix Sprint 2. 64
Tabelle 23 Risikomatrix Sprint 3. 64
Tabelle 24 Risikomatrix Sprint 5. 65
Tabelle 25 Risikomatrix Sprint 6. 65

D Abbildungsverzeichnis

Abbildung 1 Hauptmenü der Isocine3-Software. 1
Abbildung 2 Messgraphen zweier isokinetischer Messungen in Isocine3 mit Metriken. 2
Abbildung 3 Messgraphen zweier isokinetischer Messungen in Isocine3 mit überlappenden Linien. . . 3
Abbildung 4 IsoExportDB Zusatzsoftware für den Excel-Export. 3
Abbildung 5 Excel-Export aller Messungen eines Patienten. 3
Abbildung 6 Vier beispielhafte Verläufe anonymisierter Patienten mit Knietotalprothese (KTPT), dargestellt in IsoTrack. 5
Abbildung 7 Multidatenexport in Biodex System 4 Pro[48] 9
Abbildung 8 Context Map der beiden Bounded Contexts des Projekts. 16
Abbildung 9 Domänenmodell von IsoTrack. 17
Abbildung 10 C4 Kontextdiagramm der IsoTrack Architektur. 20
Abbildung 11 C4 Containerdiagramm der IsoTrack Architektur. 21
Abbildung 12 Softwarearchitektur des Smart Hospitals[49] 21
Abbildung 13 Softwarearchitektur der Smartwatches[50] 22
Abbildung 14 Verzeichnisstruktur der Software 23
Abbildung 15 Tabellen der Datenbank mit italienischen Spaltennamen. 24
Abbildung 16 Abgeleitetes Beziehungsdiagramm der Tabellen, dargestellt mit 1:n-Beziehungen in Microsoft Access. 26
Abbildung 17 Visualisierung einer TRA-Datei mit verschiedenen Datentypen (Offset: 13 Bytes, Little-Endian). 27
Abbildung 18 Ausschnitt: Zeile 3, Spalte 11 aus der Python-Visualisierung der TRA-Datei. 27
Abbildung 19 Struktur einer TRA-Datei 28
Abbildung 20 C4 Komponentendiagramm des IsoTrack Clients. 29
Abbildung 21 C4 Komponentendiagramm des IsoTrack Servers. 31
Abbildung 22 Zustandsübergangsdiagramm eines Patienten. 36
Abbildung 23 Zustandsübergangsdiagramm einer isokinetischen Messung. 37
Abbildung 24 Screenshot Startseite "Neuste Messungen". 38
Abbildung 25 Tabelle der Messungen auf der Startseite. 38
Abbildung 26 Visualisierung der Messwerte zweier Messungen. 39
Abbildung 27 Screenshot eines Isocine3 PDF-Berichts. 40
Abbildung 28 Screenshot PDF-Bericht. 41
Abbildung 29 Messgraphen zweier isokinetischer Messungen in Isocine3 mit überlappenden Linien. . 42
Abbildung 30 Screenshot Patientenprofil. 43
Abbildung 31 Screenshot Datenqualität. 44
Abbildung 32 Screenshot Datenanalyse. 45
Abbildung 33 Filter wenn "Unfall Datum ≠ leer" ausgewählt ist. 45
Abbildung 34 Filter wenn "OP Datum ≠ leer" ausgewählt ist. 46

[48] „SYSTEM 4 (Advantage BX Software 5.3.X) INSTRUCTIONS FOR USE“.

[49] „Smart hospital: achieving interoperability and raw data collection from medical devices in clinical routine“, Bde. Volume6–2024.

[50] „Development of a Research-Oriented Application for the Acquisition, Analysis, and Export of Sleep Metrics from Smartwatches“, Bd. 270.

Abbildung 35 Screenshot Einstellungen. 47
Abbildung 36 Initiales Platzhalter-Logo von IsoTrack. 48
Abbildung 37 Finales IsoTrack-Logo mit und ohne Schriftzug 49
Abbildung 38 Überblick über alle Pipeline-Stages und Jobs. 52
Abbildung 39 Testabdeckung des Clients und Servers. 54
Abbildung 40 Erhobene Performance-Metriken des IsoTrack Clients mittels dotnet-counters. 55
Abbildung 41 Zeitstrahl des Projekts 57
Abbildung 42 Übersicht über alle Epics mit grober Zeiteinteilung. 58
Abbildung 43 Datenbankeintrag zum fiktiven Patienten "Lydia Ritter". 68
Abbildung 44 Patienteneintrag zum fiktiven Patienten "Lydia Ritter". 69
Abbildung 45 Donutdiagramm der Gesamtzeit, der Zeit pro Student sowie der Restzeit, dargestellt mittels quickchart.io @quickchartSrc. 73

E Hilfsmittelverzeichnis

Aufgabenbereich	Tools
Dokumentation	• ChatGPT • DeepL • Git/GitLab • LanguageTool • Typst • Visual Studio Code
Literaturrecherche	• Claude • Google Scholar
Projektmanagement	• Atlassian Jira • Microsoft Teams
Softwareentwicklung	• Blazor • C#/.NET • ChatGPT • Git/GitLab • Microsoft Access • MudBlazor • Windows Sandbox • WiX Toolset • Visual Studio • Visual Studio Code
Diagramme	• Diagrams.net • Excalidraw • QuickChart
Designen	• ChatGPT • Inkscape • Paint.NET

F Literatur- und Quellenverzeichnis

[1] Regula Heller, „Hüft- und Knieimplantationen: SIRIS Report 2024 bestätigt Trend zu sinkenden Revisionsraten“, 10. Dezember 2024, *ANQ*. Zugegriffen: 5. Juni 2026. [Online]. Verfügbar unter: https://www.anq.ch/de/medienmitteilungen/hueft-und-knieimplantationen-siris-report-2024-bestaetigt-trend-zu-sinkenden-revisionsraten/

[2] „Modulbeschreibung - Bachelor-Arbeit Informatik“. Zugegriffen: 18. Februar 2026. [Online]. Verfügbar unter: https://studien.ost.ch/allModules/41626_M_BAI21.html

[3] „.NET - Build modern apps and powerful cloud services“. Zugegriffen: 4. Mai 2026. [Online]. Verfügbar unter: https://dotnet.microsoft.com/en-us/

[4] „Entity Framework-Dokumentationshub | Microsoft Learn“. Zugegriffen: 4. Mai 2026. [Online]. Verfügbar unter: https://learn.microsoft.com/de-de/ef/

[5] „ASP.NET Core, an open-source web development framework | .NET“. Zugegriffen: 4. Mai 2026. [Online]. Verfügbar unter: https://dotnet.microsoft.com/en-us/apps/aspnet

[6] „Blazor | Build client web apps with C# | .NET“. Zugegriffen: 4. Mai 2026. [Online]. Verfügbar unter: https://dotnet.microsoft.com/en-us/apps/aspnet/web-apps/blazor

[7] „MudBlazor - Blazor Component Library“. Zugegriffen: 4. Mai 2026. [Online]. Verfügbar unter: https://mudblazor.com/

[8] „Blazor-ApexCharts“. Zugegriffen: 4. Mai 2026. [Online]. Verfügbar unter: https://github.com/apexcharts/Blazor-ApexCharts

[9] „SQLite Home Page“. Zugegriffen: 4. Mai 2026. [Online]. Verfügbar unter: https://sqlite.org/

[10] „Übersicht über die OLE DB-Programmierung“. Zugegriffen: 4. Mai 2026. [Online]. Verfügbar unter: https://learn.microsoft.com/de-de/cpp/data/oledb/ole-db-programming-overview?view=msvc-170

[11] „WiX Toolset“. Zugegriffen: 4. Mai 2026. [Online]. Verfügbar unter: https://www.firegiant.com/wixtoolset/

[12] „HeatWave“. Zugegriffen: 4. Mai 2026. [Online]. Verfügbar unter: https://www.firegiant.com/heatwave/

[13] „Veröffentlichte Studien mit PHYSIOMED GROUP Produkten | PHYSIOMED GROUP“. Zugegriffen: 25. März 2026. [Online]. Verfügbar unter: https://www.physiomed-group.com/de-ch/services/studien#c15277

[14] „SYSTEM 4 (Advantage BX Software 5.3.X) INSTRUCTIONS FOR USE“. Zugegriffen: 31. März 2026. [Online]. Verfügbar unter: https://smti.co/hubfs/SportMedical_February2026/pdf/20-001-CLR-System-4-IFU-Rev-E.pdf?hsLang=co

[15] „HUMAC® NORM™ TESTING & REHABILITATION SYSTEM User’s Guide Model 502140“. Zugegriffen: 31. März 2026. [Online]. Verfügbar unter: https://humacnorm.com/wp-content/uploads/2025/03/300004O-409-HUMAC-NORM-User-Guide.pdf

[16] „Smart hospital: achieving interoperability and raw data collection from medical devices in clinical routine“, 6. März 2024, *Frontiers in digital health*. Zugegriffen: 31. März 2026. [Online]. Verfügbar unter: https://pmc.ncbi.nlm.nih.gov/articles/PMC10951085/pdf/fdgth-06-1341475.pdf

[17] „Development of a Research-Oriented Application for the Acquisition, Analysis, and Export of Sleep Metrics from Smartwatches“, *Elsevier B.V*. Zugegriffen: 31. März 2026. [Online]. Verfügbar unter: https://www.sciencedirect.com/science/article/pii/S1877050925033253

[18] „ISO/IEC 25010:2023“, International Organization for Standardization, Nov. 2023. Zugegriffen: 9. März 2026. [Online]. Verfügbar unter: https://www.iso.org/obp/ui/en/#iso:std:iso-iec:25010:ed-2:v1:en

[19] Simon Brown, „C4 model“. Zugegriffen: 6. Mai 2026. [Online]. Verfügbar unter: https://c4model.com/

[20] „Chrome Tabs Using Too Much RAM? Here's What Actually Works“. Zugegriffen: 26. Mai 2026. [Online]. Verfügbar unter: https://nestextended.com/blog/chrome-tabs-too-much-ram/

[21] „Microsoft SQL Server“. Zugegriffen: 4. Juni 2026. [Online]. Verfügbar unter: https://www.microsoft.com/de-de/sql-server

[22] „PostgreSQL“. Zugegriffen: 26. Mai 2026. [Online]. Verfügbar unter: https://www.postgresql.org/

[23] „PDFsharp in detail“. Zugegriffen: 5. Juni 2026. [Online]. Verfügbar unter: https://www.pdfsharp.com/PDFsharp

[24] „MigraDoc in detail“. Zugegriffen: 5. Juni 2026. [Online]. Verfügbar unter: https://www.pdfsharp.com/MigraDoc

[25] Scott W Harden, „ScottPlot - Interactive Plotting Library for .NET“. Zugegriffen: 5. Juni 2026. [Online]. Verfügbar unter: https://scottplot.net/

[26] ClosedXML Contributors, „ClosedXML“. Zugegriffen: 5. Juni 2026. [Online]. Verfügbar unter: https://github.com/closedxml/closedxml

[27] Josh Close, „CsvHelper“. Zugegriffen: 5. Juni 2026. [Online]. Verfügbar unter: https://joshclose.github.io/CsvHelper/

[28] Olaf Zimmermann, „Y-Statements“. Zugegriffen: 25. Februar 2026. [Online]. Verfügbar unter: https://medium.com/olzzio/y-statements-10eb07b5a177

[29] „The MIT License“. Zugegriffen: 27. März 2026. [Online]. Verfügbar unter: https://opensource.org/license/MIT

[30] Bill Wake, „INVEST in Good Stories, and SMART Tasks“. Zugegriffen: 25. Februar 2026. [Online]. Verfügbar unter: https://xp123.com/invest-in-good-stories-and-smart-tasks/

[31] „.NET Coding Conventions - C# | Microsoft Learn“. Zugegriffen: 9. März 2026. [Online]. Verfügbar unter: https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/coding-conventions

[32] „Visual Studio: IDE und Code-Editor für Softwareentwicklung“. Zugegriffen: 8. Juni 2026. [Online]. Verfügbar unter: https://visualstudio.microsoft.com/de/

[33] „Rider: the cross-platform .NET and game development IDE from JetBrains“. Zugegriffen: 8. Juni 2026. [Online]. Verfügbar unter: https://www.jetbrains.com/rider/

[34] belav, „CSharpier“. Zugegriffen: 4. April 2026. [Online]. Verfügbar unter: https://csharpier.com/

[35] „Official .NET Docker images“. Zugegriffen: 13. April 2026. [Online]. Verfügbar unter: https://learn.microsoft.com/en-us/dotnet/architecture/microservices/net-core-net-framework-containers/official-net-docker-images

[36] „Secret detection“. Zugegriffen: 4. April 2026. [Online]. Verfügbar unter: https://docs.gitlab.com/user/application_security/secret_detection/

[37] „Static application security testing (SAST)“. Zugegriffen: 4. April 2026. [Online]. Verfügbar unter: https://docs.gitlab.com/user/application_security/sast/

[38] „dotnet-counters diagnostic tool - .NET CLI - .NET | Microsoft Learn“. Zugegriffen: 26. Mai 2026. [Online]. Verfügbar unter: https://learn.microsoft.com/en-us/dotnet/core/diagnostics/dotnet-counters

[39] P. Kruchten, „Tutorial: introduction to the Rational Unified Process®“, *IEEE Institute of Electrical and Electronics Engineers*. Zugegriffen: 20. Februar 2026. [Online]. Verfügbar unter: https://ieeexplore.ieee.org/document/1008039

[40] „Create Windows Service using BackgroundService - .NET | Microsoft Learn“. Zugegriffen: 9. März 2026. [Online]. Verfügbar unter: https://learn.microsoft.com/en-us/dotnet/core/extensions/windows-service

[41] „Create a Windows Service installer - .NET | Microsoft Learn“. Zugegriffen: 9. März 2026. [Online]. Verfügbar unter: https://learn.microsoft.com/en-us/dotnet/core/extensions/windows-service-with-installer

[42] Google, „Google Material Symbols“. [Online]. Verfügbar unter: https://fonts.google.com/icons

[43] Microsoft Corporation,*.mdb icon (2000-03).svg*. [Online]. Verfügbar unter: https://commons.wikimedia.org/wiki/File:.mdb_icon_(2000-03).svg