

Visualisieren von 24h Daten des ActivPal Sensors

Studienarbeit HS 2025

Version: 1.0

Autor: Silvan Bollhalder

E-Mail: silvan.bollhalder@ost.ch

Autor: Quentin Weber

E-Mail: quentin.weber@ost.ch

Dozent: Prof. Dr. Markus Stolze

Experte: Dr. Martina Betschart

Praxispartner: MOVE-IT

Studiengang: BSc Informatik

Erstellt am: 25. September 2025

Letzte Änderung am: 19. Dezember 2025

Abstract

Die objektive Erfassung von Bewegungsverhalten mittels ActivPAL-Sensoren liefert wertvolle Daten für Forschung und klinische Praxis. Die bestehenden Visualisierungsmöglichkeiten der vom Sensor exportierten Daten durch die PAL Analysis Software sind jedoch wenig flexibel und auf die spezifischen Bedürfnisse von medizinischem Fachpersonal nicht zugeschnitten. Es fehlt an intuitiven, stundenbasierten 24-Stunden-Aktivitätsprofilen, automatisierten Durchschnittsprofilen über mehrere Messtage und Gruppenvergleichen. Dies erschwert nicht nur die effiziente Interpretation der Daten, sondern macht auch die Aufbereitung für wissenschaftliche Publikationen und Studien problematisch. Ohne qualitativ hochwertige Diagramme und aussagekräftige Vergleichsdarstellungen können Forschungsergebnisse nicht angemessen präsentiert werden, was die Verwertbarkeit der erhobenen Daten erheblich einschränkt.

Um diese Defizite zu beseitigen, wurde eine eigenständige Desktop-Applikation entwickelt, die von PAL Analysis exportierte CSV-Dateien importiert und vielfältige Visualisierungen bereitstellt. Die auf Electron und React basierende Software ermöglicht die Darstellung von Schrittdaten mit Kadenzfilterung, Sitzphasenanalysen, Aktivitätsprofilen über 24 Stunden und täglichen Zusammenfassungen. Patienten können in Gruppen organisiert werden, um kohortenweite Vergleiche durchzuführen. Die Visualisierungen unterstützen verschiedene Darstellungsformen (Liniendiagramme, Boxplots, Konfidenzintervalle) und ermöglichen den Export als PNG, PDF oder CSV für die direkte Verwendung in wissenschaftlichen Arbeiten. Zusätzlich können die Daten der Patienten und Gruppen gefiltert werden.

Neben der Kernfunktionalität wurden umfassende Qualitätssicherungsmaßnahmen implementiert, darunter Unit-Tests, E2E-Tests sowie Code-Qualitätsprüfungen mittels ESLint und JSCPD. Die Anwendung berücksichtigt Farbkontraste, läuft ohne Administratorrechte unter Windows 11 und gewährleistet Datenschutz, indem Rohdaten ausschliesslich temporär verarbeitet und nur Gruppendefinitionen persistent gespeichert werden. Der Quellcode wird unter MIT-Lizenz veröffentlicht.

Management Summary

Ausgangslage und Problemstellung

Die objektive Erfassung von Bewegungsverhalten mittels ActivPAL-Sensoren liefert wertvolle Daten für medizinische Forschung und klinische Praxis. Die bestehende PAL Analysis Software bietet jedoch nur eingeschränkte und wenig flexible Visualisierungsmöglichkeiten. Es fehlen intuitive 24-Stunden-Aktivitätsprofile, automatisierte Durchschnittsberechnungen über mehrere Messtage und aussagekräftige Gruppenvergleiche. Diese Defizite erschweren sowohl die effiziente Dateninterpretation als auch die Aufbereitung für wissenschaftliche Publikationen erheblich.

Zielsetzung

Ziel der Arbeit war die Entwicklung einer eigenständigen Desktop-Applikation zur Verarbeitung und Visualisierung von ActivPAL-Daten. Die Software sollte praxisorientiert, verständlich und flexibel einsetzbar sein – sowohl für individuelle Patientenanalysen als auch für Forschungsprojekte mit Gruppenstudien. Der Quellcode wird unter MIT-Lizenz veröffentlicht, um Transparenz und Weiterentwicklung zu ermöglichen.

Lösungsansatz

Die entwickelte Electron-basierte Desktop-Anwendung importiert von PAL Analysis exportierte CSV-Dateien und bietet vielfältige Visualisierungsmöglichkeiten:

- **Datenimport:** Automatische Erkennung von Patientenordnern mit Fehlerbehandlung
- **Gruppenverwaltung:** Organisation von Patienten in logischen Gruppen für Kohortenvergleiche
- **Visualisierungen:** Fünf unterschiedliche Diagrammtypen (Schritte pro Stunde, Aktivitäten über den Tag, Sitzphasen, Schritte nach Kadenz, Vergleich täglicher Durchschnittswerte)
- **Darstellungsmodi:** Liniendiagramme, Boxplots und Konfidenzintervalle für verschiedene Analyseperspektiven
- **Filteroptionen:** Kadenzfilterung und Datumsselektion mit Validitätsfiltern (Wochentage/Wochenende)
- **Export:** PNG, PDF und CSV für direkte Verwendung in wissenschaftlichen Arbeiten

Technologie und Architektur

Die Anwendung basiert auf Electron mit React und TypeScript, was moderne Webtechnologien für Desktop-Anwendungen nutzbar macht. Chart.js wurde als Visualisierungsbibliothek gewählt, da sie unter MIT-Lizenz verfügbar ist und umfangreiche Anpassungsmöglichkeiten bietet. Die Architektur folgt bewährten Patterns mit klarer Trennung zwischen Main-Prozess (Dateiverarbeitung, CSV-Parsing) und Renderer-Prozess (React-UI).

Qualitätssicherung

Umfassende Qualitätsmassnahmen wurden implementiert:

- **Unit-Tests:** 89.85% Statement-Coverage für die Kernlogik (1178/1311 Statements)
- **E2E-Tests:** 39 manuelle End-to-End-Tests zur Validierung aller Anforderungen
- **Usability-Tests:** Vier Tests mit Zielgruppe (MOVE-IT-Mitarbeitende) und externen Personen
- **Code-Reviews:** Zwei formelle Reviews mit externem Experten
- **CI/CD-Pipeline:** Automatisierte Tests, ESLint-Prüfung und Duplikations-Analyse (< 3% Duplikation)
- **Barrierefreiheit:** WCAG 2.1-konforme Farbkontraste

Projekterfolg

Von 28 funktionalen Anforderungen wurden 27 vollständig umgesetzt (96%), von 13 nicht-funktionalen Anforderungen 12 (92%). Das definierte MVP wurde nicht nur erreicht, sondern deutlich übertroffen. Alle Priorität-1- und Priorität-2-Anforderungen sowie mehrere Priorität-3-Features wurden erfolgreich implementiert.

Die einzige nicht erreichte funktionale Anforderung (Timeline-Darstellung) wurde durch eine funktional überlegene Alternative ("Aktivitäten über den Tag") ersetzt, die in Abstimmung mit dem Praxispartner als praxisgerechter eingestuft wurde. Die leicht erhöhte Startzeit (4-5 Sekunden statt geforderte ≤ 3 Sekunden) stellt keine wesentliche Einschränkung der Nutzbarkeit dar.

Nutzen und Ausblick

Die Anwendung bietet konkreten Mehrwert:

- **Klinische Praxis:** Bessere Interpretierbarkeit von Bewegungs- und Ruheverhalten durch intuitive Visualisierungen
- **Forschung:** Standardisierte, reproduzierbare Durchschnittsprofile für Studienanalysen und Gruppenvergleiche
- **Publikationsvorbereitung:** Direkt verwertbare Diagramme in publikationsreifer Qualität

Die modulare Architektur und offene Lizenzierung ermöglichen künftige Erweiterungen wie zentrale Datenverwaltung mit Rollenverwaltung, KI-gestützte Mustererkennung oder Unterstützung weiterer Sensoren.

Das Projekt wurde termingerecht abgeschlossen, alle geplanten Features wurden ohne Scope-Einschränkungen umgesetzt und die Qualität blieb durchgehend hoch. Der Praxispartner MOVE-IT hat bereits positives Feedback zur Anwendung gegeben.

Inhalt

Abstract.....	2
Management Summary.....	3
Inhalt.....	5
1 Ausgangslage.....	7
1.1 Hintergrund	7
1.2 Problemstellung.....	7
2 Aufgabenstellung / Ziel der Arbeit	8
2.1 Zielsetzung.....	8
2.2 Teilziele	8
2.3 Erwarteter Nutzen	8
3 Anwendungsübersicht.....	9
3.1 Domänenmodell	9
4 Anforderungen	11
4.1 Funktionale Anforderungen	11
4.2 Nicht-funktionale Anforderungen	19
4.3 Definition MVP	21
5 Architektur	22
5.1 Nutzwertanalyse Technologien.....	22
5.2 Entscheidung für Electron	26
5.3 C4 Diagramm	26
5.4 CSV-Dateien	29
5.5 Electron Store	29
5.6 Libraries und Frameworks	30
5.7 Auflistung verwendeten 3rd Party Libraries	31
5.8 Projekt Ordner Struktur.....	32
5.9 Wartung / Weiterentwicklung; Guide und Tools	32
6 UX/UI	33
6.1 Navigation	33
6.2 Import-Ansicht	33
6.3 Patienten-Ansicht	34
6.4 Gruppen-Ansicht	35
6.5 Visualisierungsauswahl	36
6.6 Diagramm-Ansichten	37
6.7 Export-Funktionen	44
6.8 Feedback und Fehlermeldungen	46
6.9 Responsive Design und Barrierefreiheit	47
6.10 Performance und User Experience.....	48
6.11 Zukünftige UX-Verbesserungen	48
7 Erreichte Ziele / Resultate.....	49
7.1 Funktionale Anforderungen	49
7.2 Nicht-funktionale Anforderungen	49

7.3	Zusammenfassung.....	50
7.4	Ausblick und Erweiterungsmöglichkeiten.....	50
8	Projektmanagement	51
8.1	Projektplan-SOLL	51
8.2	Zeitplan-IST.....	52
8.3	Vergleich & Diskussion Zeitplan	52
8.4	Zeiterfassung pro Person	54
8.5	Risikomanagement.....	55
9	Qualitätsmanagement.....	59
9.1	Regeln.....	59
9.2	GitLab Pipeline	60
9.3	Deployment	61
9.4	Testkonzept.....	61
9.5	Code Patterns	72
9.6	Code Metriken	73
10	KI-Reflexion	76
10.1	Positive Aspekte.....	76
10.2	Kritische Einschränkungen	76
10.3	Fazit	76
11	Literaturverzeichnis	77
12	Verzeichnisse	78
12.1	Abbildungen	78
12.2	Tabellen	79
13	Anhang.....	80
13.1	Aufgabenstellung.....	80
13.2	User Guide	83
13.3	Code Metriken.....	87
13.4	Usability Tests.....	94

1 Ausgangslage

Die objektive Erfassung von Bewegungs- und Schlafverhalten ist ein zentrales Instrument in Forschung und klinischer Praxis. Mittels Aktigraphie lassen sich Aktivitätsmuster kontinuierlich über längere Zeiträume hinweg dokumentieren, ohne die Patienten in ihrem Alltag wesentlich einzuschränken. Solche 24-Stunden-Profile liefern wertvolle Hinweise auf die körperliche Aktivität, Ruhephasen und deren Bedeutung für den Gesundheitszustand.

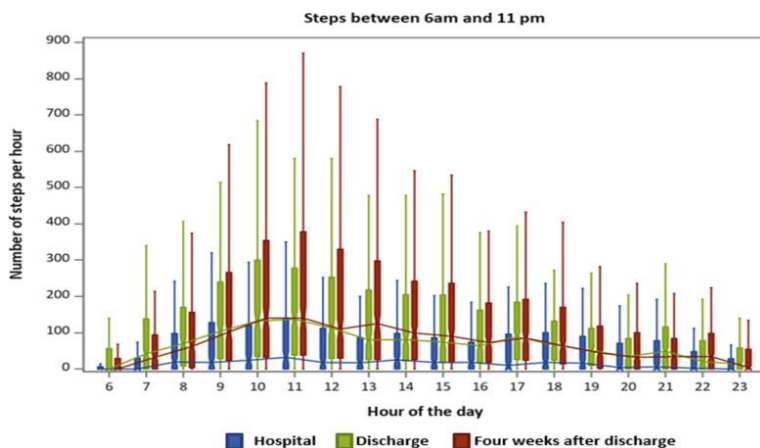


Abbildung 1 - ActivPAL Daten in aktuellen Studien [1]

1.1 Hintergrund

Der ActivPAL-Sensor ist ein etabliertes Messgerät, das zuverlässig Sitz-, Steh- und Gehzeiten erfasst. Mit der zugehörigen Software können die Sensordaten exportiert und in Programmen wie ActivPAL Analysis weiterverarbeitet werden. Es ist bereits möglich, erste Visualisierungen zu erstellen, diese sind jedoch standardisiert, wenig flexibel und nicht auf die spezifischen Bedürfnisse von Forschung und klinischer Anwendung zugeschnitten.

Die Erfahrung zeigt, dass die bestehenden Möglichkeiten zur Analyse und Darstellung für medizinisches Fachpersonal nur eingeschränkt nutzbar sind. Für eine effiziente Interpretation der Daten ist eine Software erforderlich, die speziell auf klinische Anforderungen zugeschnittene Darstellungen bereitstellt.

1.2 Problemstellung

Die aktuelle Situation lässt sich wie folgt zusammenfassen:

- **Vorhandene Visualisierungsmöglichkeiten:** ActivPAL-Daten können mit ActivPAL Analysis dargestellt werden, allerdings nur in einer sehr allgemeinen Form. Ohne Programmierkenntnisse ist die Weiterverarbeitung nur schwer möglich.
- **Mangelnde Anpassbarkeit:** Die Visualisierungen sind nicht flexibel genug, um individuelle klinische Fragestellungen oder unterschiedliche Studienanforderungen abzubilden.
- **Fehlende intuitive Darstellung:** Es gibt keine stundenbasierte Übersicht, die ein vollständiges 24-Stunden-Aktivitätsprofil leicht verständlich visualisiert.
- **Keine automatisierten Durchschnittsprofile:** Mittelwerte, Median, Standardabweichung über mehrere Messstage oder für ganze Kohorten sind unzureichend dargestellt.
- **Keine eigenständige Lösung:** Es existiert bisher keine leicht einsetzbare Applikation, die die spezifischen Anforderungen von Klinik und Forschung erfüllt.

Die Notwendigkeit ergibt sich daraus, eine Software zu entwickeln, die diese Defizite beseitigt und die ActivPAL-Daten in klinisch relevante Informationen transformiert.

2 Aufgabenstellung / Ziel der Arbeit

Die originale Aufgabenstellung befindet sich im Anhang.

2.1 Zielsetzung

Ziel der Arbeit ist die Entwicklung einer eigenständigen Applikation zur Verarbeitung und Visualisierung von ActivePAL-Daten. Damit soll ein Werkzeug geschaffen werden, das praxisorientiert, verständlich und flexibel einsetzbar ist - sowohl für die individuelle Patienten als auch für den Einsatz in Forschungsprojekten.

2.2 Teilziele

- **Stundenbasierte Darstellung:** Erstellung von 24-Stunden-Profilen, die Tagesmuster klar und übersichtlich darstellen.
- **Individuelle Durchschnittsprofile:** Berechnung und Visualisierung von Durchschnittswerten für einzelne Patienten über mehrere Messtage hinweg.
- **Kohortenweite Durchschnittsprofile:** Automatisierte Erstellung von Mittelwerten und Vergleichen für ganze Gruppen oder Studienkohorten.
- **Eigenständige Applikation:** Entwicklung einer unabhängigen Softwarelösung, die leicht bedienbar ist und eine klinisch relevante Darstellung ermöglicht. Der Quellcode wird unter einer MIT-Lizenz veröffentlicht, um Transparenz und Weiterentwicklung zu ermöglichen.

2.3 Erwarteter Nutzen

- **Für die klinische Praxis:** Bessere Interpretierbarkeit der Bewegungs- und Ruheverhalten durch intuitive Visualisierungen.
- **Für die Forschung:** Standardisierte und reproduzierbare Durchschnittsprofile, die für Studienanalysen und Gruppenvergleiche eingesetzt werden können.
- **Für die Weiterentwicklung:** Eine offene, modulare Grundlage, die künftige Erweiterungen, beispielsweise für weitere Sensoren, ermöglicht.

3 Anwendungsübersicht

In diesem Kapitel wird ein umfassender Einblick in die geplante Anwendung gegeben. Zunächst wird das Domänenmodell vorgestellt und erläutert. Anschliessend folgt eine Beschreibung der Anwendung, die mit Hilfe von unterstützenden Skizzen die zentralen Abläufe und Ansichten der Applikation verdeutlicht. Die Skizzen sind nicht als finales Designkonzept zu verstehen.

3.1 Domänenmodell

Das Domänenmodell beschreibt den Zusammenhang zwischen den Benutzereinstellungen, den Patientendaten und deren Visualisierung in Diagrammen.

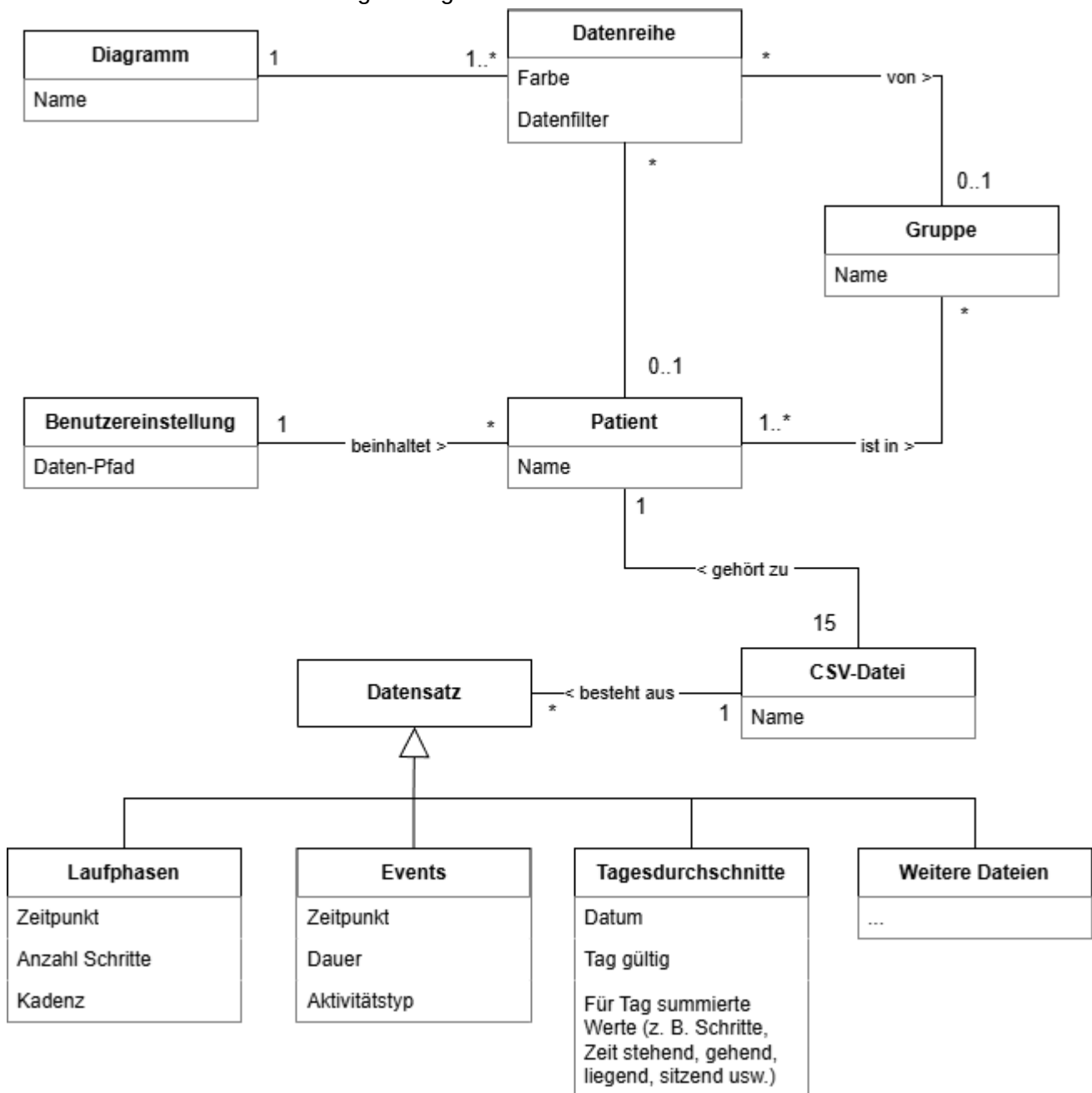


Abbildung 2 - Domänenmodell

Benutzereinstellung:

Der Benutzer gibt den Pfad an, in dem die Patientenordner gespeichert sind.

Patient:

Jeder Patient besitzt einen eigenen Ordner mit 15 verschiedenen CSV-Dateien. Die Applikation verwendet momentan drei Dateien. Diese sind im Domänenmodell dargestellt. Falls in Zukunft noch weitere Dateien dazukommen, sind diese ebenfalls im Domänenmodell zu ergänzen. Die in der Applikation nicht verwendeten Dateien werden nur mit «Weitere Dateien» beschrieben. Der Name des Patienten wird durch den Ordner seiner Daten definiert. Die Dateien enthalten die Messdaten, welche für die Visualisierung benötigt werden.

Gruppe:

Eine Gruppe besteht aus mehreren Patienten. Sie kann vom Benutzer erstellt werden und ihr wird ein Name gegeben. Dies ermöglicht die Darstellung oder den Vergleich mit mehreren Patienten in einer Ansicht.

CSV-Dateien und Datensätze:

Das Domänenmodell bildet die drei Dateitypen ab (Laufphasen, Events, Tagesdurchschnitte), welche in der Applikation verwendet werden. Die Datensätze sind nicht vollständig, da nur die verwendeten Spalten aufzeigt werden und zusätzlich bei Tagesdurchschnitte die Liste an für den Tag summierte Werte zusammenfasst wurde. Die exakten Informationen zu den verwendeten Dateien sind im Kapitel 5.4 beschrieben.

Diagramm und Datenreihe:

Aus den CSV-Dateien werden Datenreihen erzeugt, die in Diagrammen visualisiert werden. Es können mehrere Datenreihen enthalten sein, wobei jede Datenreihe eine eigene Farbe erhält. Eine Datenreihe ist entweder eine Gruppen- oder Patientenauswahl. Zusätzlich hat jede Datenreihe einen Filter:

- Valide Tage (Validität ist in Tagesdurchschnitte CSV für jeden Tag definiert)
- Valide Wochentage
- Valide Wochenendtage
- Alle Tage
- Alle Wochentage
- Alle Wochenendtage
- Benutzerdefiniert (nur bei Patientenauswahl, Daten können einzeln an- und abgewählt werden)

4 Anforderungen

Im Rahmen dieser Arbeit wird ein Minimum Viable Product (MVP) definiert. Das MVP stellt die kleinste lauffähige Version der Software dar, mit der bereits die Kernidee demonstriert werden kann. Konkret umfasst das MVP die Möglichkeit, einen Projektordner mit Patienten-CSV-Dateien auszuwählen (Import) und daraus ein erstes Diagramm zu erzeugen (Visualisierung).

Die Anforderungen sind nicht nach Standard definiert. Um die Kommunikation mit dem Auftraggeber zu vereinfachen, wurden die Anforderungen lösungsspezifisch und mit Sketches aus Figma formuliert.

Alle Anforderungen und Nicht-Anforderungen wurden vom Praxis-Partner (Martina Betschart) abgenommen.

4.1 Funktionale Anforderungen

Um den Entwicklungsfokus zu steuern, werden alle Anforderungen nach Prioritäten eingeteilt:

- **Priorität 1 (MVP):** Diese Anforderungen müssen zwingend umgesetzt werden, da sie die Grundfunktionalität sicherstellen.
- **Priorität 2 (Soll):** Diese Anforderungen sollen im Rahmen der Arbeit umgesetzt werden, sofern es der zeitliche Rahmen erlaubt. Sie sind wichtig für die Praxistauglichkeit.
- **Priorität 3 (Optional):** Diese Anforderungen sind wünschenswert, aber nicht zwingend notwendig. Sie werden nur umgesetzt, falls nach der Umsetzung der Priorität-1- und Priorität-2-Anforderungen genügend Zeit verbleibt.

Die Unterteilung der Anforderungen wurde nach FURPS gemacht. Nachfolgend sind die funktionalen Anforderungen beschrieben. Diese bilden das F in FURPS.

4.1.1 Import

Die Anforderungen werden zusammen mit zu Beginn definierten Figma-Visualisierungen (Sketches) erstellt. Dazu stehen jeweils die Priorität und die zugehörigen E2E-Tests.

- **FR-I1 (P1) E2E-I-1:** Der Benutzer soll einen Überordner/Projektordner auswählen können per Datei-Dialog. Jeder Unterordner entspricht einem Patienten.
- **FR-I2 (P2) E2E-I-1:** Das System durchsucht pro Patientenordner die enthaltenen ActivePAL-CSV und erstellt automatisch einen Patientendatensatz (Name = Ordnername).



Abbildung 3 - Import Page

- **FR-I3 (P2) E2E-PG-1:** Es gibt eine Übersichtsliste aller Patienten mit mindestens: Name, Zugehörige Gruppe, Ordnerpfad. Suche/Sortierung/Filter sind möglich.
- **FR-I4 (P2) E2E-PG-4:** Die Patienten können in der Ansicht aller Patienten aus-/eingebledet werden. Wenn ein Patient ausgeblendet ist, kann er in einem Diagramm nicht mehr ausgewählt werden. Die Namen aller ausgeblendeten Patienten werden gespeichert. Falls ein Patient einer Gruppe zugehörig war, wird dieser in der Gruppenaggregation übersprungen.
- **FR-I5 (P2) E2E-PG-1:** Es gibt eine Möglichkeit das Dateisystem neu einzulesen. Patienten werden dadurch neu geladen.

activePal

Patienten

Aktualisieren

Name	Group	Folderpath	
S01	G01	C:/Data/S01	...
S02	G02	C:/Data/S02	...
			Bearbeiten Löschen

Abbildung 4 - Patienten Page

4.1.2 Gruppen

- **FR-G1 (P2) E2E-PG-3:** Es gibt eine separate Ansicht «Gruppen». Benutzer können aus vorhandenen Patienten Gruppen definieren (Name, Mitgliederliste).
- **FR-G2 (P2) E2E-P-2:** Die Gruppen kann man anlegen, umbenennen, löschen. Zudem kann man Mitglieder hinzufügen/entfernen. Persistenz der Definitionen im Projekt (JSON im Projektordner).

activePal

Gruppen

Neu

Name	Membercount	Members
G01	3	S01,S02,S03
G02	3	S04,S05,S06

Bearbeiten
Löschen

Abbildung 5 - Gruppen Page

- **FR-G3 (P2) E2E-PG-3:** Patienten können mit einer Mehrfachauswahl in eine Gruppe hinzugefügt werden. Dabei kann ein Patient auch in mehreren Gruppen sein.

activePal

Gruppen

Neu

Name

G01

G02

Name

G01

G02

Name

Name eingeben...

Mitglieder auswählen

	Name	Group	Folderpath
☒	S01	-	C:/Data/S01
☐	S02	-	C:/Data/S02
☐			
☐			
☐			
☐			

Erstellen

Abbildung 6 - Gruppe erstellen

4.1.3 Visualisierung

- **FR-V1 (P1) E2E-V-2:** Die Ansicht «Schritte pro Tag» zeigt ein Liniendiagramm der Schritte, welche durchschnittlich absolviert wurden.

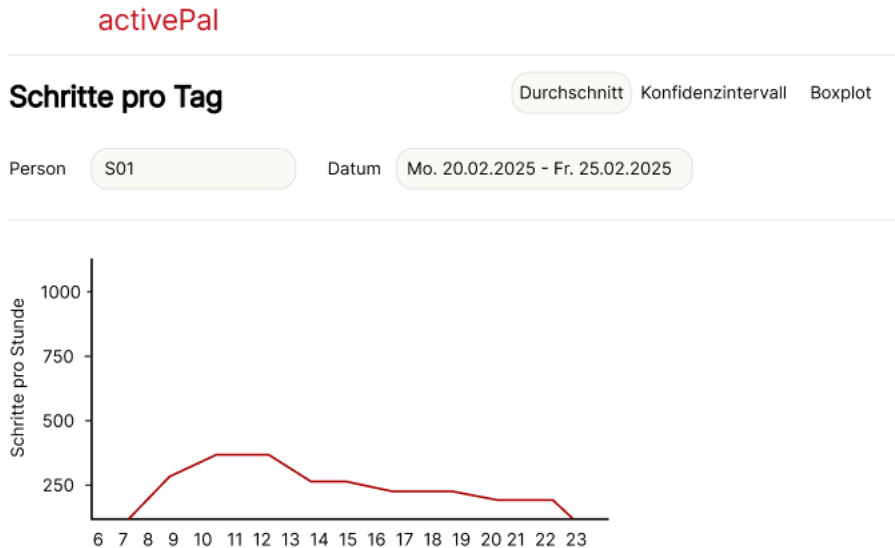


Abbildung 7 - Schritte pro Tag Diagramm

- **FR-V2 (P2) E2E-V-2:** Die Ansicht «Schritte pro Tag» zeigt zudem: «Totale Schrittzeit» in Minuten; «Totale Anzahl Schritte»; «Kadenz > 75 < 100»; «Kadenz > 100» in den Zusatzinformationen an. Im Diagramm selbst kann man die Schritte nach Kadenz filtern mit einem Kadenzfilter. Dieser Filter geht bei Verlassen der Visualisierung nicht verloren.
- **FR-V3 (P2) E2E-V-2:** Die Ansicht «Schritte pro Tag» hat drei Darstellungen. Die einfache Darstellung zeigt vereinfachte Informationen an mit einem Liniendiagrammen. Schaltet man auf die Standardabweichungsdarstellung um, werden die Daten mit einem 95% Konfidenzintervall angezeigt. Schaltet man auf die Boxplot Ansicht um, werden die Daten mit Hilfe von Boxplots dargestellt.

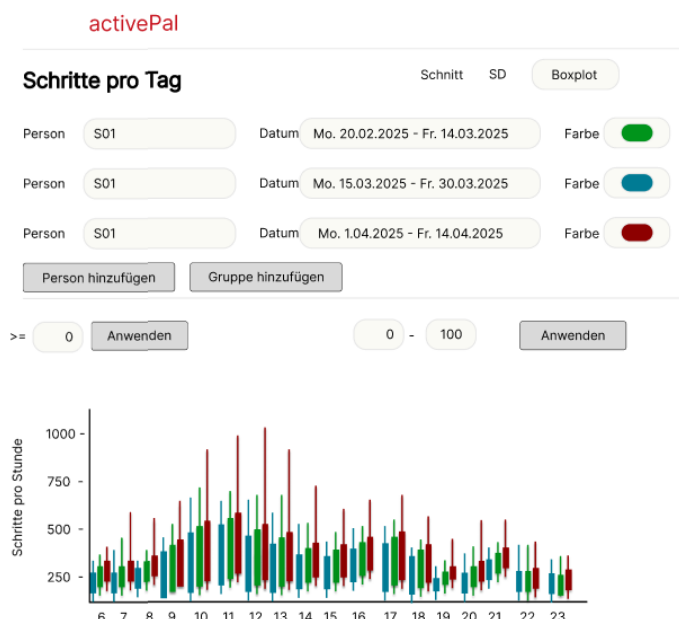


Abbildung 8 - Schritte pro Tag Diagramm – Boxplotsansicht

- **FR-V4 (P3) E2E-V-3:** Die Ansicht «Tagesverlauf» zeigt: «Inaktive Zeit»; «Aufrechte Zeit»; «Liegezeit»; «Sitzende Zeit»; «Anzahl der Übergänge vom Sitzen zum Stehen». Zudem werden die Daten mit einer Timeline über 24h visualisiert.

activePal

Tagesverlauf

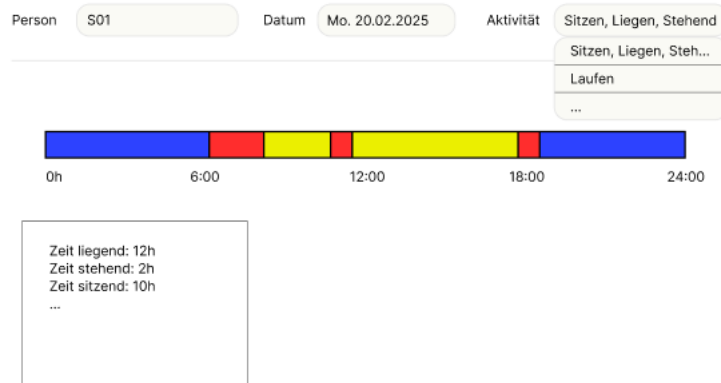


Abbildung 9 - Tagesverlauf Timeline

- **FR-V5 (P2) E2E-V-4:** Die Ansicht «Sitzphasen» zeigt: «Anzahl Sitzphasen (30min)»; «Anzahl Sitzphasen (60min)»; «Zeit sitzend 30 min»; «Zeit sitzend 60 min».

activePal

Sitzphasen

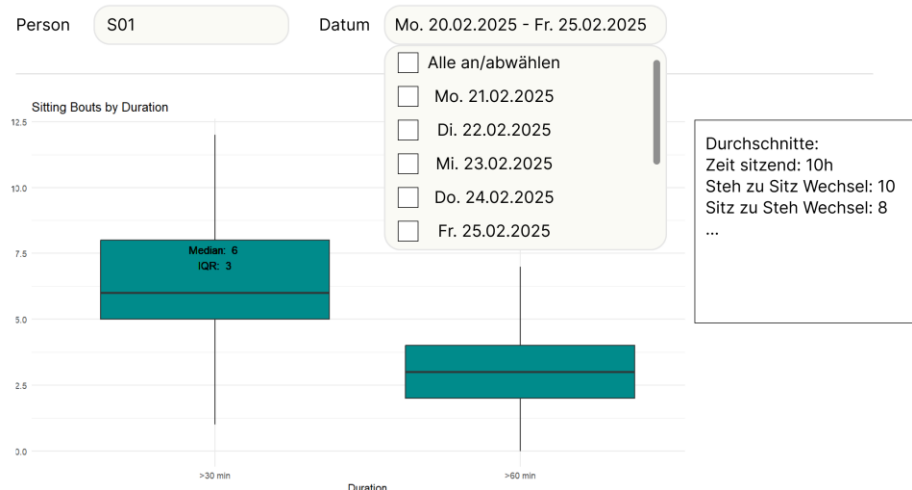


Abbildung 10 - Sitzphasen Diagramm

- **FR-V6 (P2) E2E-V-5:** Die Ansicht «Tages Zusammenfassung» ermöglicht es eine Gruppe oder Person mit anderen Gruppen oder Personen zu vergleichen.

activePal

Tages Zusammenfassung

Person

S01

Datum

Mo. 20.02.2025

Vergleich mit Durchschnitt (Overall aus Word-Table oder aus den Daten)	
Zeit liegend: 12h	+5% (2h)
Zeit stehend: 2h	-50% (2h)
Zeit sitzend: 10h	+20% (4h)
Zeit laufend:	
Anzahl Schritte:	...
Schritte mit Kadenz	
Cycling time	
...	

Abbildung 11 - Tages Zusammenfassung Page

- **FR-V7 (P2) E2E-V-6:** Die Ansicht «Schritte mit Kadenz» zeigt «Totale Anzahl Schritte», «Kadenz > 75 spm» und «Kadenz > 100 spm» an mit drei Boxplots (spm = Schritte pro Minute). Mit dem Kadenzfilter können eigene Boxplots erstellt werden, mit beliebigen SPM-Werten. Die Vergleichsoperatoren grösser- und kleiner-gleich stehen zur Verfügung. Alternativ kann ein Bereich definiert werden (z.B. 50-100).

activePal

Settings

Add Data

Schritte mit Kadenz

Person

S01

Datum

Mo. 20.02.2025 - Fr. 25.02.2025

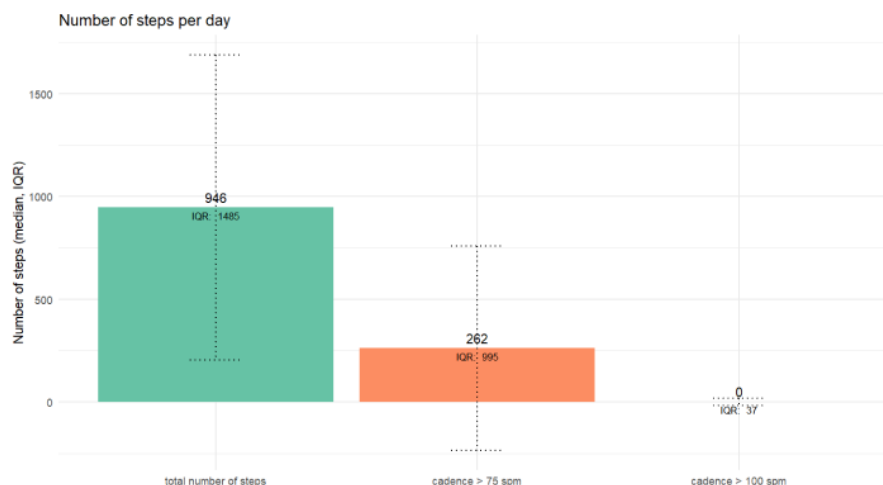


Abbildung 12 - Schritte mit Kadenz Diagramm

- **FR-V8 (P2) E2E-V-7:** Alle Diagramme können auch mehrere Personen bzw. Gruppen anzeigen. Dabei können Personen und Gruppen gleichzeitig angezeigt werden. Eine Farbauswahl soll möglich sein, um die verschiedenen Gruppen/Patienten zu kennzeichnen. Die Tagesverlauf Timeline ist davon ausgenommen.
- **FR-V9 (P2) E2E-V-1:** Alle Diagramme haben eine Datumsauswahl, mit welcher eine Zeitspanne ausgewählt werden kann, in welcher die Daten visualisiert werden sollen.
- **FR-V10 (P3) E2E-P-1:** Der Status eines Diagramms (Ausgewählte Farben, Datumsspanne, User/Gruppen) soll gespeichert werden, damit bei einem Neustart des Programms die Einstellungen eines Diagramms wieder hergestellt werden kann.
- **FR-V11(P2) E2E-E-1, E2E-E-2:** Der User kann ein Diagramm exportieren. Die Dateiformate sind PDF und CSV. Bei PDF wird eine Abbildung des Diagramms exportiert. Bei CSV werden die einzelnen Datenpunkte des Diagramms exportiert. Ist der Standardabweichungsmodus ausgewählt, wird zusätzlich die obere und untere Grenze des 95% Konfidenzintervalls exportiert.
- **FR-V12(P2) E2E-V-8:** Es gibt eine Auswahlpage, um zwischen den verschiedenen Diagrammen zu navigieren.

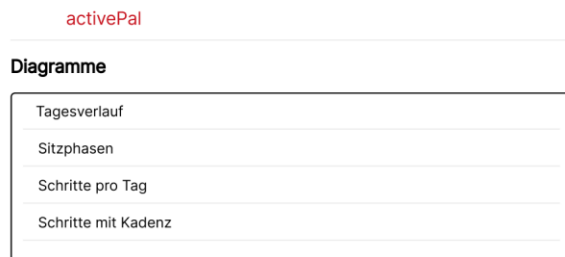


Abbildung 13 - Diagramme Auswahlpage

- **FR-V13(P2) E2E-V-9:** Es gibt eine Visualisierung Aktivitäten über den Tag, welche die Aktivitäten Gehen, Stehen, Sitzen, Liegen darstellt. Die Visualisierung an sich zeigt eine Aktivität über einen Tag und wie viele Minuten dieser Aktivität in einer Stunde jeweils ausgeführt wurden.

activePal

Aktivität über den Tag

Schnitt

SD

Boxplot

Person	S01	Datum	Mo. 20.02.2025 - Fr. 14.03.2025	Farbe	☒
Person	S01	Datum	Mo. 15.03.2025 - Fr. 30.03.2025	Farbe	☒
Person	S01	Datum	Mo. 1.04.2025 - Fr. 14.04.2025	Farbe	☒

Aktivität: ☒ Liegen ☐ Sitzen ☐ Stehen ☐ Gehen

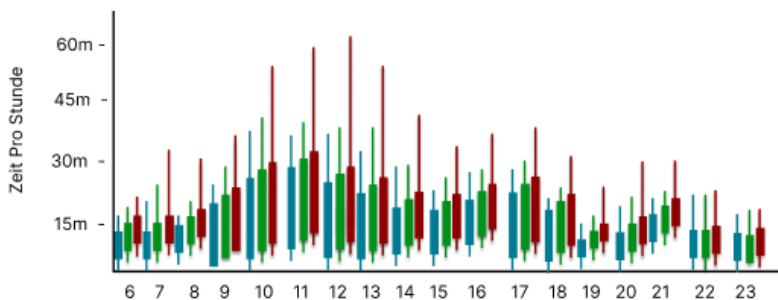


Abbildung 14 - Aktivitäten über den Tag

- **FR-V14(P2) E2E-V-10:** In einer Visualisierung sollen per Standard nur valide Tage angezeigt werden. Es gibt Schnellfilter für valide Wochentage, sowie valide Wochenendtage und alle Wochentage, sowie alle Wochenendtage.

4.1.4 Nicht-Anforderungen (Out of Scope)

- **NA-1:** Keine Cloud-Anbindung, kein Serverbetrieb.
- **NA-2:** Kein Direktimport vom Sensor/keine Rohdatenerfassung (nur bereits exportierte CSV).
- **NA-3:** Keine mobile App (iOS/Android).
- **NA-4:** Keine Integration in das frühere GENEActiv-System.
- **NA-5:** Keine Benutzer-/Rollenverwaltung, kein Login.
- **NA-6:** Keine automatische Verteilung/Updates (manuelle Installation pro Gerät).

4.1.5 Use Case Diagramm

Das Use Case Diagramm zeigt die Funktionalen Anforderungen vereinfacht auf.

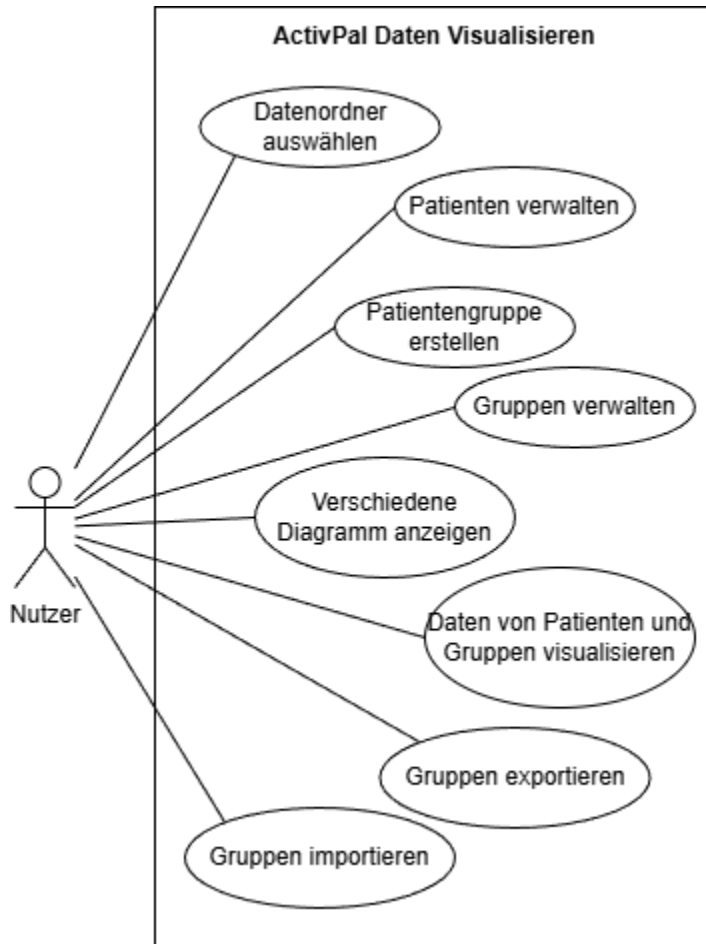


Abbildung 15 - Use Case Diagramm

4.2 Nicht-funktionale Anforderungen

Nachfolgend sind die Nicht-funktionalen Anforderungen zu sehen. Diese bilden das URPS in FURPS.

4.2.1 Usability

- **NFR-U1 - E2E-U-1:** Vom Projektordner-Wählen bis zur ersten Visualisierung sind nur wenige Schritte erforderlich. (≤ 3)
- **NFR-U2 - E2E-U-2:** Die Benutzeroberfläche reagiert auf Eingaben direkt mit einer Fortschrittsanzeige; langlaufende Prozesse laufen asynchron.
- **NFR-U3 - E2E-U-3:** Die Visualisierungen und Bedienelemente verwenden Farben, welche Punkt 1.4.1 der WCAG 2.1 Richtlinien zur Barrierefreiheit erfüllen.
- **NFR-U4:** Es ist eine Benutzeranleitung vorhanden, welche den User durch die Installation und Bedienung der Applikation führt.

4.2.2 Reliability

- **NFR-R1 - E2E-I-2:** Ungültige Dateien im Import werden übersprungen, die restlichen Daten werden verarbeitet. Dabei wird eine verständliche Fehlermeldung ausgegeben.
- **NFR-R2 - E2E-I-3:** Bei fehlenden Rechten oder gesperrten Dateien liefert die Anwendung verständliche Hinweise und bricht ohne Datenverlust ab.

4.2.3 Performance

- **NFR-P1 - E2E-P-1:** Indexierung eines Projektordners mit 10 Patienten erfolgt in 3 Sekunden, die Benutzeroberfläche bleibt responsiv.
- **NFR-P2 - E2E-P-2:** Analyse von sieben Tagen pro Patienten dauert höchstens fünf Sekunden auf einem Standard-Notebook (≥ 8 GB RAM, 2 Cores, SSD).
- **NFR-P3 - E2E-P-3:** Startzeit der Anwendung beträgt höchstens drei Sekunden; RAM-Spitzen bei Gruppenanalyse liegen unter 1.5 GB.

4.2.4 Security

- **NFR-SEC1 - E2E-SEC-1:** Die Anwendung speichert keine personenbezogenen Daten dauerhaft. Alle Analysen werden nur temporär verarbeitet, Arbeitsdateien beim Beenden automatisch gelöscht. Das einzige, was gespeichert wird, sind die Gruppenzusammenstellungen und der Zustand der Diagramme.
- **NFR-SEC2 - E2E-SEC-2:** Fehler- und Log-Dateien enthalten keine vertraulichen Daten (z. B. Patientenname, Dateipfad) enthalten.

4.2.5 Compatibility

- **NFR-C1 - E2E-C-1:** Die Anwendung kann parallel zu anderer Software auf den Zielgeräten betrieben werden, ohne bestehende Prozesse zu stören.

4.2.6 Portability

- **NFR-PO1 - E2E-PO-1:** Lauffähig auf Windows 11 (23H2+).
- **NFR-PO2 - E2E-PO-2:** Bereitstellung als .EXE.
- **NFR-PO3 - E2E-PO-3:** Installation im User-Kontext ohne Administratorrechte ist möglich. Konfigurations- und Arbeitsverzeichnisse liegen im Benutzerprofil.
- **NFR-PO4 - E2E-PO-4: Ersetzbarkeit:** Updates können durch Austausch der Anwendung erfolgen, ohne Datenmigration (Gruppen/Settings bleiben im Projektordner erhalten).

4.3 Definition MVP

Das Minimal Viable Product (MVP) umfasst die zentralen Funktionen, die für eine erste nutzbare Version der Anwendung notwendig sind. Im Fokus stehen dabei die priorisierten P1-Anforderungen. Der Benutzer soll über einen Datei-Dialog einen Überordner bzw. Projektordner auswählen können, wobei jeder Unterordner einem Patienten entspricht (FR-I1). Auf Basis dieser Daten stellt die Anwendung eine grundlegende Visualisierung bereit: In der Ansicht «Schritte pro Tag» wird ein Liniendiagramm angezeigt, das die durchschnittlich absolvierten Schritte darstellt (FR-V1). Die Daten dafür werden aus dem Steppingbouts.csv, Spalte NumSteps entnommen. Es soll möglich sein, ein Datumsrange auszuwählen. Zudem soll der MVP eine erste Version der Navigation beinhalten, welche dem Anwender ermöglicht von Dateiimport zu den Diagrammen zu navigieren.

5 Architektur

5.1 Nutzwertanalyse Technologien

Für die Umsetzung der Anwendung haben wir drei Technologien in die engere Auswahl genommen: Java, Electron.js und Python. Grundlage für diese Auswahl waren Übersichten zu Programmiersprachen für Datenvisualisierung [2]. Da wir keine Erfahrung mit R hatten, wurde ein Ersatz gesucht; Java wurde als Alternative gewählt, da es sich insbesondere für die Datenverarbeitung eignet und in entsprechenden Übersichten als geeignete Programmiersprache aufgeführt wird [3] und wir sehr viel Erfahrung mit Java haben. Um die Entscheidung zu treffen, wurden die Technologien anhand der folgenden Kriterien mit einer Punktzahl von 1-5 bewertet:

- Deployment (Einfachheit der Bereitstellung und Installation)
- Daten einlesen (Eignung zum CSV-Handling)
- Charts (Modernität & Flexibilität)
- Performance (Laufzeitgeschwindigkeit, Ressourcenverbrauch)
- Testbarkeit (Verfügbarkeit und Reife von Test-Frameworks)
- Wartbarkeit (Code-Struktur, Erweiterbarkeit, Stabilität)
- Community (Grösse, Support)
- Look & Feel (UI) (Modernität und Attraktivität der Benutzeroberfläche)
- Autonome Ausführbarkeit (Offline-Nutzung ohne externe Abhängigkeiten)
- Plattformunabhängigkeit (Lauffähigkeit auf Windows, Linux und macOS)

5.1.1 Java mit JavaFX

JavaFX ist ein GUI-Framework für Java-Anwendungen. Für Datenvisualisierung bietet JavaFX eingebaute Standard-Chart-Komponenten, die jedoch im Vergleich zu modernen Web-basierten Bibliotheken weniger interaktiv und flexibel sind. CSV-Daten können mit Bibliotheken wie Apache Commons CSV effizient eingelesen werden.

Kriterium	Beschreibung	Bewertung
Deployment	Java Runtime muss vorhanden sein, Packaging über JAR/Installer möglich	3
Daten einlesen	CSV-Verarbeitung mit Bibliotheken wie Commons CSV problemlos	4
Charts (Modernität & Flexibilität)	JavaFX Charts stabil aber weniger interaktiv und modern	3
Performance	Sehr gute Performance durch kompilierten Code, wenig Ressourcenverbrauch	5
Testbarkeit	Sehr gute Tools (JUnit, TestFX) für Unit- und UI-Tests	4
Wartbarkeit	Statische Typisierung und saubere Architektur möglich	4
Community	JavaFX Community kleiner, Java insgesamt gross	3
Look & Feel (UI)	Klassisch, wirkt im Vergleich weniger modern	3
Autonome Ausführbarkeit	Voll offline lauffähig, keine Internetabhängigkeit	5
Plattformunabhängigkeit	Cross-Plattform mit Java Runtime auf allen Systemen	4

5.1.2 Electron mit React

Electron kombiniert Node.js mit Chromium und ermöglicht so die Erstellung von Desktop-Anwendungen, die im Kern wie eine Web-App funktionieren. Für Datenvisualisierung kann auf moderne Chart-Bibliotheken zurückgegriffen werden, eine flexible und interaktive Chart-Bibliothek. CSV-Dateien lassen sich ebenfalls mit JavaScript-Bibliotheken verarbeiten. Zusätzlich kann React oder ein anderes Frontend-Framework integriert werden.

Kriterium	Beschreibung	Bewertung
Deployment	Grosse App-Grösse, Deployment über Node/NPM Pakete	3
Daten einlesen	Sehr einfach mit Bibliotheken, hohe Flexibilität	5
Charts (Modernität & Flexibilität)	Sehr modern, interaktiv, viele Optionen durch Bibliotheken	5
Performance	Mittlere Performance, hoher RAM/CPU-Verbrauch wegen Chromium	3
Testbarkeit	Gut mit Jest testbar	4
Wartbarkeit	Sauber strukturierbar mit modularen JS-Frameworks	4
Community	Sehr grosse JavaScript-Community, viele fertige Lösungen	5
Look & Feel (UI)	Sehr modern dank HTML/CSS, flexibel anpassbar	5
Autonome Ausführbarkeit	Offline möglich, benötigt aber eingebetteten Chromium	4
Plattformunabhängigkeit	Cross-Plattform durch Electron (Win/Linux/Mac)	4

5.1.3 Python mit customtkinter

Das Framework customtkinter bringt eine moderne Optik für Python-Desktopanwendungen. Für CSV-Verarbeitung ist Python durch Pandas und weitere Python-Bibliotheken besonders gut geeignet. Die Darstellung von Charts wird durch externe Bibliotheken wie matplotlib oder Plotly ermöglicht.

Kriterium	Beschreibung	Bewertung
Deployment	Sehr einfach mit PyInstaller, kleine kompakte Binaries möglich	4
Daten einlesen	Sehr stark dank Pandas und eingebauter CSV-Unterstützung	5
Charts (Modernität & Flexibilität)	Einfache Charts über matplotlib/plotly, begrenzte Integration in GUI	3
Performance	Langsamer als Java, für kleine/mittlere Datenmengen ausreichend	3
Testbarkeit	pytest stark für Logiktests, UI-Tests schwieriger	3
Wartbarkeit	Dynamische Sprache, bei grösseren Projekten komplexer	3
Community	Sehr grosse Data-Science-Community, GUI kleiner	4
Look & Feel (UI)	Moderner als Tkinter, aber im Vergleich eingeschränkt	3
Autonome Ausführbarkeit	Offline lauffähig, keine externen Abhängigkeiten	4
Plattformunabhängigkeit	Cross-Plattform mit Python und Tkinter (Win/Linux/Mac)	4

5.1.4 Vergleich mit Gewichtung

Kriterium	Gewicht	Java	Electron	Python
Deployment	7	3	3	4
Daten einlesen	8	4	5	5
Charts (Modernität & Flexibilität)	9	3	5	3
Performance	9	5	3	3
Testbarkeit	8	4	4	3
Wartbarkeit	8	4	4	3
Community	6	3	5	4
Look & Feel (UI)	6	3	5	3
Autonome Ausführbarkeit	10	5	4	4
Plattformunabhängigkeit	9	4	4	4
Gesamt		311	333	288

Der Vergleich zeigt, dass Electron die meisten Punkte erreicht hat. Es hat jedoch jede Technologie ihre eigenen Stärken. Python ist am besten für die CSV-Verarbeitung und schnelle Entwicklung. Java hat seine Stärken im Bereich Performance, Wartbarkeit und Testbarkeit. Electron überzeugt mit modernen Charts und User Interfaces.

5.2 Entscheidung für Electron

Wir haben uns für die Technologie Electron und React entschieden. Electron bietet die Möglichkeit, eine Desktop-Anwendung zu entwickeln, die trotz dessen auf Webtechnologien basiert. Durch React kann eine komponentenbasierte Architektur aufgebaut werden, welche die Wartbarkeit der Anwendung verbessert. Ein weiterer Vorteil dieser Technologie ist die Erweiterbarkeit.

Für die Datenvisualisierung wird auf Chart.js zurückgegriffen, da diese Bibliothek eine grosse Community hat und viel Möglichkeiten für eigene Anpassungen bietet. Die Chart.js Diagramme sind möglicherweise etwas weniger modern als andere kostenpflichtige Bibliotheken. Chart.js ist jedoch unter der MIT-Lizenz verfügbar, was maximale Freiheit in der Nutzung und Verteilung unserer Anwendung garantiert, ohne lizenzrechtliche Einschränkungen.

5.3 C4 Diagramm

Um die Softwarearchitektur der entwickelten Anwendung strukturiert darzustellen, wurde das C4-Modell angewendet. Dieses Modell ermöglicht eine hierarchische Visualisierung der Systemarchitektur in vier Ebenen (Context, Container, Component, Code), wobei für die Dokumentation die Ebenen L1 bis L3 relevant sind.

5.3.1 C4 – Level 1: Systemkontext



[System Context] ActivePAL Desktop App
Dienstag, 30. September 2025 um 17:40 Mittteleuropäische Sommerzeit

Abbildung 16 - C4 Level 1

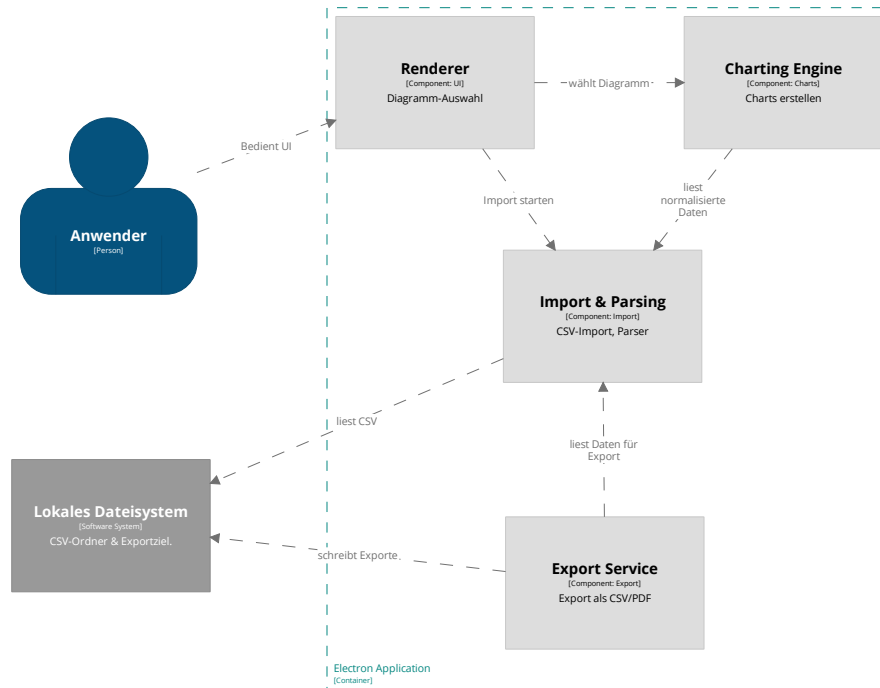
5.3.2 C4 – Level 2: Container



[Container] ActivePAL Desktop App
Dienstag, 30. September 2025 um 17:39 Mittteleuropäische Sommerzeit

Abbildung 17 - C4 Level 2

5.3.3 C4 – Level 3: Komponente



[Component] ActivePAL Desktop App - Electron Application
Dienstag, 30. September 2025 um 17:38 Mitteleuropäische Sommerzeit

Abbildung 18 - C4 Level 3

5.4 CSV-Dateien

Die durch den Benutzer ausgewählten CSV-Dateien werden nicht direkt in der Anwendung gespeichert, sondern über ihren Dateipfad referenziert. Die Anwendung greift bei Bedarf auf diese Pfade zu, um die Daten einzulesen. Alle CSV-Dateien von PALanalysis enthalten einen Header, welcher für die Applikation nicht relevant ist. Es wird immer bis zum Datenmarker (**data**) gesprungen und erst dann gelesen. Dabei werden in der Applikation drei verschiedene CSV-Dateien gelesen:

...DailySummaries.csv

In dieser Datei ist für jeden aufgezeichneten Tag eine Zusammenfassung verfügbar mit der Auswertung, ob genügend lange aufgezeichnet wurde, damit der Tag in der Applikation als gültiger Tag angezeigt wird. Die Spalten «Date» und «Day Valid» werden in jedem Fall für die Selektion gelesen. Die folgenden Spalten werden für Diagramme und Zusatzdaten verwendet:

- Total Num Steps
- Total Stepping Time(m)
- Num Steps (Cadence>=75)
- Num Steps (Cadence>=100)
- Num Sitting Bouts >30m
- Num Sitting Bouts >60m
- Time Spent In Sitting Bouts >30m
- Time Spent In Sitting Bouts >60m
- Num Sit To Stand Transitions & Num Stand To Sit Transitions
- Cycling Time(m)
- Upright Time(m)
- Primary Lying Time(m) & Secondary Lying Time(m)
- Num Cycling Steps
- Sitting Time(m)

...EventsEx.csv

Das EventsEx.csv ist die grösste Datei, wie verwendet wird. Sie enthält für jede Änderung im Aktivitätstyp einen Eintrag. Diese Datei wird für die Darstellung «Aktivität pro Stunde» benötigt. Dabei werden die Spalten «Time(approx)» (zur Bestimmung des Zeitpunktes), «Duration (s)» (Dauer der Aktivität) und «Event Type» (Aktivität) verwendet.

...SteppingBouts.csv

In dieser Datei sind alle Schrittdaten zusammengefasst. Diese Datei wird für die Diagramme «Schritte pro Stunde» und «Schritte nach Kadenz pro Tag» verwendet, um die Anzahl Schritte mit ihrer exakten Kadenz filtern zu können. Dazu werden «Time(approx)» (Zeitstempel), «Num Steps» (Anzahl Schritte) und «Cadence» (Frequenz der Schritte, Schritte pro Minute) gelesen. Danach werden die Daten so umgewandelt, dass für jede Stunde oder jeden Tag ersichtlich ist, wie viele Schritte in welcher Kadenz gemacht wurden.

5.5 Electron Store

Für die persistente Speicherung wird Electron Store verwendet. Electron Store speichert Daten als JSON-Datei im von Electron vorgesehenen benutzerspezifischen Konfigurationsverzeichnis des Betriebssystems (z. B. AppData unter Windows, Application Support unter macOS oder .config unter Linux).

Es werden die ausgewählten Pfade gespeichert, sowie der aktuelle Pfad, damit die Applikation auch nach einem Neustart noch weiss, welcher Pfad offen war. Diese Daten befinden sich im «general.json». Für jeden geöffneten Pfad wird eine zusätzliche JSON-Datei angelegt, welche die ausgeblendeten Patienten, sowie die Gruppeninformationen für den entsprechenden Pfad enthält.

5.6 Libraries und Frameworks

React

Für die Implementierung des Frontends wurde React gewählt. Diese Entscheidung basiert primär auf der bereits vorhandenen Erfahrung mit React. React bietet zudem eine modulare Komponentenstruktur, die eine klare Trennung von Logik und Darstellung erlaubt und die Wiederverwendbarkeit von UI-Komponenten unterstützt, was ein entscheidender Vorteil bei der Darstellung komplexer Diagramme aus CSV-Daten.

JavaScript / TypeScript

Ursprünglich wurde JavaScript als Programmiersprache verwendet, da im Team mehr Erfahrung mit JavaScript vorhanden war. Im Rahmen von Code Review 1 wurde jedoch darauf hingewiesen, dass eine Migration auf TypeScript viele Vorteile bietet, insbesondere durch die stärkere Typisierung. Durch die Umstellung auf TypeScript konnten Fehler bereits zur Entwicklungszeit erkannt und die Codequalität signifikant verbessert werden. Gleichzeitig konnten bestehende Entscheidungen bezüglich Frameworks und Libraries weiterhin beibehalten werden.

Chart Library

Für die Visualisierung der CSV-Daten in unterschiedlichen Diagrammen wurden die Chart-Bibliotheken Highcharts, Plotly und Chart.js evaluiert. Die Auswahl dieser Bibliotheken basierte auf aktuellen Übersichten zu den führenden JavaScript-Charting-Bibliotheken [4], [5], in denen sie regelmässig als besonders leistungsfähig, vielseitig und gut dokumentiert aufgeführt werden, insbesondere im Hinblick auf unterstützte Diagrammtypen, Interaktivität und Community-Akzeptanz. Ausserdem war das optische Erscheinungsbild ein zentrales Kriterium bei der Vorauswahl, weshalb nicht alle verfügbaren Bibliotheken berücksichtigt wurden. Die ausgewählten Bibliotheken zeichneten sich dabei durch moderne und ansprechende Standardvisualisierungen aus. Highcharts fiel aufgrund der Lizenzbedingungen aus, da die Bibliothek kommerziell ist und nicht unter der MIT-Lizenz verfügbar ist. Plotly wurde zwar ebenfalls geprüft, erfüllte jedoch nicht alle Projektanforderungen hinsichtlich Einfachheit und Anpassbarkeit.

Chart.js erwies sich schliesslich als die am besten geeignete Lösung. Die Bibliothek ist unter der MIT-Lizenz verfügbar, einfach in React integrierbar und bietet eine grosse Flexibilität bei der Darstellung von Diagrammen. Besonders vorteilhaft ist, dass Chart.js die dynamische Aktualisierung von Diagrammen unterstützt – ein entscheidender Faktor, da CSV-Daten häufig neu eingelesen oder gefiltert werden müssen. Zudem ermöglicht Chart.js eine einfache Darstellung von Mehrfach-Datensätzen in einem Diagramm, was für den Vergleich von Patienten- oder Gruppendaten essenziell ist. Zusätzlich unterstützt Chart.js die Darstellung von Boxplots, wodurch statistische Verteilungen, Medianwerte und Ausreisser in den Patientendaten anschaulich visualisiert werden können. Dies war eine wichtige Anforderung, damit die Daten auf verschiedenen Arten angezeigt werden können. Weitere Vorteile sind die umfangreiche Dokumentation, eine grosse Community für Unterstützung und eine gute Performance selbst bei grösseren Datensätzen. Zusammengefasst erfüllt Chart.js sowohl die funktionalen Anforderungen, die eine flexible, interaktive und leicht erweiterbare Visualisierung von CSV-Daten erfordern, als auch die technischen Anforderungen hinsichtlich Lizenzfreiheit und einfacher Integration in das bestehende React/TypeScript-Frontend.

UI

Für die Umsetzung der Benutzeroberfläche wurde zusätzlich MUI (Material-UI) verwendet. MUI bietet eine umfangreiche Sammlung vorgefertigter, responsiver Komponenten, die sich nahtlos in React integrieren lassen. Es wurde sich aufgrund von bestehender Erfahrung dafür entschieden. Die Vorteile liegen insbesondere in der einheitlichen Designsprache, der schnellen Entwicklungszeit durch wiederverwendbare UI-Komponenten, der Unterstützung von Theming und Responsivität sowie der umfassenden Dokumentation und Community-Unterstützung. Dadurch konnten komplexe UI-Elemente wie Tabellen, Buttons oder Dialoge effizient umgesetzt werden, ohne die Konsistenz oder Benutzerfreundlichkeit zu beeinträchtigen.

State Container

Für das State-Management unserer Electron-Anwendung wurde uns bei Code Review 2 @zubridge/electron vorgeschlagen. Bei unserer Recherche sind wir jedoch auf Zustand gestossen. Wir haben uns schliesslich für Zustand entschieden, da die Implementierung deutlich einfacher und schlanker ist, was die Entwicklung und Wartung des globalen States erheblich erleichtert.

Weitere Libraries

Neben den beschriebenen Libraries wurden weitere für CSV-Verarbeitung, PDF-Erstellung, Datei-Downloads, Logging, Build-Tools, Testing und Code-Qualität eingesetzt. Ihr Einfluss auf das Gesamtergebnis der Anwendung ist jedoch gering, sodass eine ausführliche Konkurrenzanalyse hierfür nicht erforderlich ist.

5.7 Auflistung verwendeten 3rd Party Libraries

Tabelle 1 - 3rd Party Libraries

Library	Beschreibung	Lizenz
React	Frontend-Bibliothek für komponentenbasiertes UI	MIT
React DOM	Bindings von React an den Browser	MIT
React Router DOM	Routing für Single-Page-Applications	MIT
Chart.js	Bibliothek für interaktive Diagramme	MIT
react-chartjs-2	React-Wrapper für Chart.js	MIT
chartjs-plugin-zoom	Zoom- und Pan-Plugin für Chart.js	MIT
@sgratzl/chartjs-chart-boxplot	Boxplot-Erweiterung für Chart.js	MIT
Material UI (@mui/material)	UI-Komponentenbibliothek für React	MIT
@mui/icons-material	Icon-Komponenten für Material UI	MIT
Zustand	State-Management für React	MIT
Electron	Framework zur Erstellung von Desktop-Apps mit Web-technologien	MIT
electron-store	Persistente Konfiguration/Store für Electron	MIT
electron-squirrel-startup	Helper für Windows-Installer mit Electron	MIT
electron-log	Logging-Library für Electron	MIT
FileSaver.js	Download von Dateien im Browser	MIT
PapaParse	CSV-Parsing im Browser/Node	MIT
jsPDF	Erstellung von PDF-Dateien im Browser	MIT
Vite	Build-Tool und Dev-Server	MIT
@vitejs/plugin-react	Vite-Plugin für React-Projekte	MIT
TypeScript	Statische Typisierung für JavaScript	Apache 2.0
Jest	Testframework für JavaScript/TypeScript	MIT
ts-jest	TypeScript-Preprocessor für Jest	MIT
ESLint	Linter für JavaScript/TypeScript	MIT
Prettier	Code-Formatter	MIT

5.8 Projekt Ordner Struktur

src	Source Code
common	Konstanten und Typen
main	Electron Main Prozess
import	Logik zum Einlesen der CSV-Dateien
parser	Parser zur Verarbeitung der eingelesenen CSV-Daten
ipc	IPC Handler
utils	Logger, Speichern von Gruppen etc.
main.ts	Einstiegspunkt Main (Fensteraufbau)
preload.ts	IPC Bridge
renderer	React App
api	Schnittstelle zu Main-Prozess
components	React Komponenten, die mehrmals verwendet werden
chart	Chart Komponenten
selection	Daten Selektion Komponenten
...	weitere Komponentenunterteilungen
hooks	Datenladen, Gruppen & Patienten Hooks
layouts	Grundlayout Komponenten
pages	Hauptseiten der App
store	State-Container (globaler Zustand)
styles	Styling Komponenten (CSS, MUI Theme)
renderer.tsx	Einstiegspunkt für das React-Frontend
test	Jest Unit Tests
assets	App-Icon
public	Weitere statische Medien

5.9 Wartung / Weiterentwicklung; Guide und Tools

Für die Wartung und Weiterentwicklung des Projekts ist lediglich Node.js (Version ≥ 16.4) und npm erforderlich. Während der Entwicklung wurde GitLab verwendet, dies ist aufgrund der Pipeline zu empfehlen. Alle weiteren Libraries werden über npm verwaltet. Wie das Projekt lokal zum Laufen gebracht wird, ist im README.md des Projektordners beschrieben. Als IDE wurde Visual Studio Code verwendet.

Anforderungen an Produktionsumgebung

Damit die App deployt werden kann wird ein entsprechendes Betriebssystem benötigt. Wird der Befehl `npm run make` im Projektordner ausgeführt, wird im out Ordner ein dem Rechner entsprechendes Setup deployt. Dem entsprechend ist für jedes Betriebssystem, das angeboten werden soll, ein Rechner / VM / Pipeline Runner notwendig.

Datenmigration

Die Applikation speichert nur die Pfade, Gruppen und ausgeblendeten Patienten in JSON-Dateien mit Electron-Store. Die Daten der Patienten werden immer aus den CSV-Dateien gelesen. Falls sich die CSV-Dateien ändern, muss der Code entsprechend angepasst werden. Bei Änderungen in den JSON-Dateien, muss darauf geachtet werden, dass auch veraltete Versionen noch eingelesen werden können und beispielsweise mit einem Default Wert gesetzt werden.

6 UX/UI

Die Benutzeroberfläche und User Experience der ActivPAL Data Visualizer Applikation wurden mit Fokus auf Effizienz, Intuitivität und Barrierefreiheit entwickelt. Das Design orientiert sich an den Bedürfnissen von medizinischem Fachpersonal und Forschenden, die schnell und zuverlässig zu aussagekräftigen Visualisierungen gelangen müssen.

6.1 Navigation

Die Anwendung verwendet eine persistente seitliche Navigationsleiste, über die zwischen den Hauptbereichen gewechselt werden kann:

- **Datenauswahl:** Auswahl und Import von Patientendaten
- **Patienten:** Verwaltung der importierten Patienten
- **Gruppen:** Erstellung und Verwaltung von Patientengruppen
- **Visualisierungen:** Zugriff auf verschiedene Diagrammtypen

6.1.1 UX-Überlegungen zur Navigation:

- Die Navigation bleibt stets sichtbar und ermöglicht schnellen Kontextwechsel
- Aktiver Bereich ist farblich hervorgehoben
- Icons unterstützen die Text-Labels für schnellere visuelle Erfassung
- Navigation kann per Drag verbreitert werden, damit Text zu den Labels erscheint

6.2 Import-Ansicht

Die Import-Ansicht ermöglicht die Auswahl eines Projektordners über einen Datei-Dialog. Nach der Auswahl werden alle Unterordner automatisch als Patienten erkannt und deren CSV-Dateien indexiert.

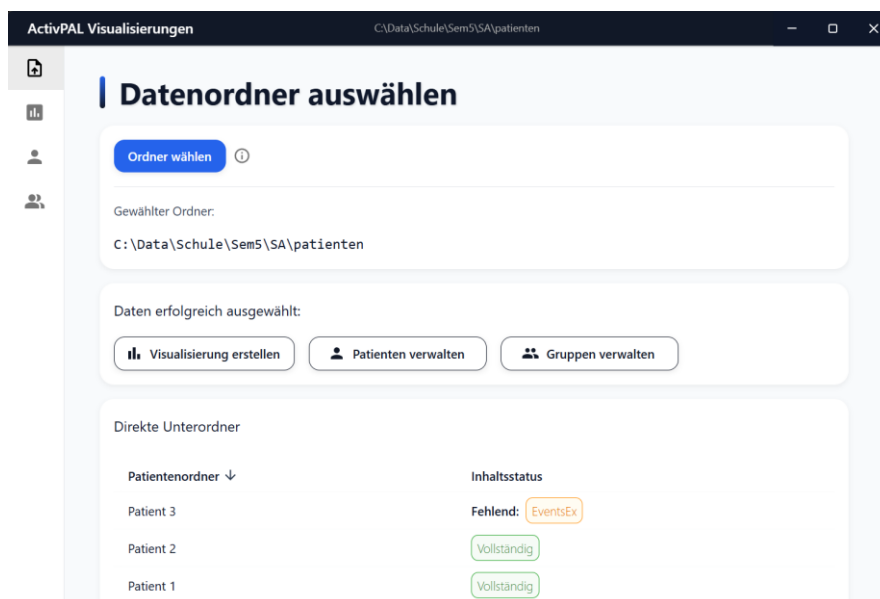


Abbildung 19 - Datenordner auswählen

6.2.1 Hauptelemente:

- Prominenter "Ordner auswählen"-Button als primäre Aktion
- Fehlermeldungen bei ungültigen oder unvollständigen Daten mit konkreten Hinweisen
- Direkte Navigationselemente

6.3 Patienten-Ansicht

In der Patienten-Ansicht werden alle importierten Patienten in einer tabellarischen Übersicht dargestellt.



Abbildung 20 - Patientenansicht

6.3.1 Funktionen:

- **Tabellenspalten:** Name, Gruppe, Anzeigen (ein-/ausgeblendet)
- **Sortierung:** Klick auf Spaltenüberschriften sortiert die Liste (auf-/absteigend)
- **Ein-/Ausblenden:** Toggle-Button zum Ausblenden von Patienten
- **Aktualisieren-Button:** Erneutes Einlesen des Dateisystems bei Änderungen
- **Bearbeiten:** Patienten könne umbenannt oder gelöscht werden, dies wirkt sich direkt auf das Dateisystem aus. Die jeweiligen Ordner werden umbenannt oder gelöscht.

Ausgeblendete Patienten erscheinen weder in Diagrammen noch in Gruppenaggregationen, bleiben aber im System erhalten und können jederzeit wieder eingeblendet werden.

6.4 Gruppen-Ansicht

Die Gruppen-Ansicht ermöglicht die Organisation von Patienten in logischen Einheiten für Vergleichsanalysen.



Abbildung 21 – Gruppenansicht

6.4.1 Funktionen:

- **Gruppe erstellen:** Button zur Erstellung neuer Gruppen
- **Gruppenverwaltung:** Kontextmenü zum Umbenennen und Löschen bestehender Gruppen
- **Mitgliederverwaltung:** Intuitive Mehrfachauswahl von Patienten
- **Gruppenübersicht:** Kompakte Darstellung aller Gruppen mit Mitgliederzahl
- **Import / Export:** Exportieren und Importieren von Gruppen JSON



Abbildung 22 - Gruppe erstellen Dialog

6.4.2 UX-Überlegungen:

- **Mehrfachauswahl:** Patienten können per Checkbox-Liste ausgewählt werden
- **Visuelle Gruppierung:** Gruppenmitglieder werden in einer scrollbaren Liste angezeigt
- **Validierung:** Gruppen ohne Namen können nicht erstellt werden
- **Überschneidungen erlaubt:** Ein Patient kann in mehreren Gruppen sein
- **Bestätigungsdialoge:** Vor dem Löschen einer Gruppe wird um Bestätigung gebeten

6.5 Visualisierungsauswahl

Die Visualisierungsauswahl bietet einen Überblick über alle verfügbaren Diagrammtypen.

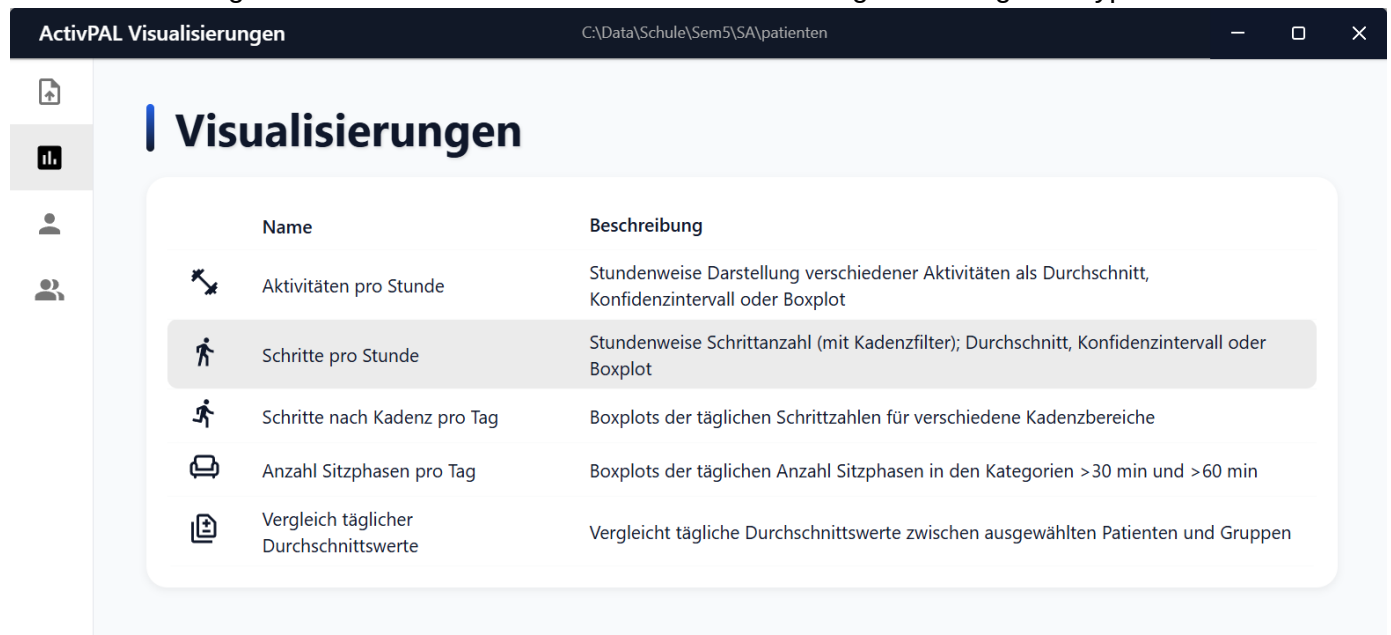


Abbildung 23 - Visualisierungen Übersicht

6.5.1 Verfügbare Visualisierungen:

- Aktivitäten pro Stunde
- Schritte pro Stunde
- Schritte nach Kadenz pro Tag
- Sitzphasen pro Tag
- Vergleich täglicher Durchschnittswerte

6.5.2 UX-Überlegungen:

- **Unterstützende Icons:** Visuell unterscheidbare Icons für schnellere Orientierung neben den Labels als Anschrift.
- **Kurzbeschreibungen:** Ein-Zeiler erklärt den Zweck jeder Visualisierung

6.6 Diagramm-Ansichten

6.6.1 Diagramm-Ansicht Layout

Alle Diagramm-Ansichten folgen einem konsistenten Layout-Muster:

6.6.1.1 Datenselektion

- **Patienten-/Gruppenauswahl:** Dropdown mit Suchfunktion und Mehrfachauswahl
- **Datumsspanne:** Datumsauswahl mit Schnellfilter (Valide Tage, Wochenendtage etc.)
- **Farbauswahl:** Color-Picker für jede ausgewählte Datenreihe
- **Gruppe Erstellen:** Für bessere Usability kann eine Gruppe direkt in der Diagramm Ansicht erstellt werden.

Datenselektion

Patient
Patient 1

Daten / Filter
Valide Tage

Patient
Patient 2

Daten / Filter
Valide Tage

Gruppe
Testgruppe 1

Anzuzeigende Tage
Valide Tage

Patient hinzufügen

Gruppe hinzufügen

Gruppe erstellen

Abbildung 24 - Datenselektion

6.6.1.2 Chart Container

- **Diagramm-Modi:** Auswahl zwischen drei verschiedenen Modi (Durchschnitt, Konfidenzintervall, Boxplot (Bei Sitzphasen und Schritte mit Kadenz, wird nur Boxplot angezeigt))
- **Export:** Export Buttons für PDF, PNG und CSV
- **Zoom:** Buttons für Drag-Zoom, Pan-Zoom und Scroll-Zoom
- **Fullscreen:** Mit dem Fullscreen-Button kann man den ganzen Chart-Container im vollen Bildschirm anzeigen.
- **Diagramm:** Das Diagramm selbst. Es zeigt alle Datenreihen an, im gewählten Modus.

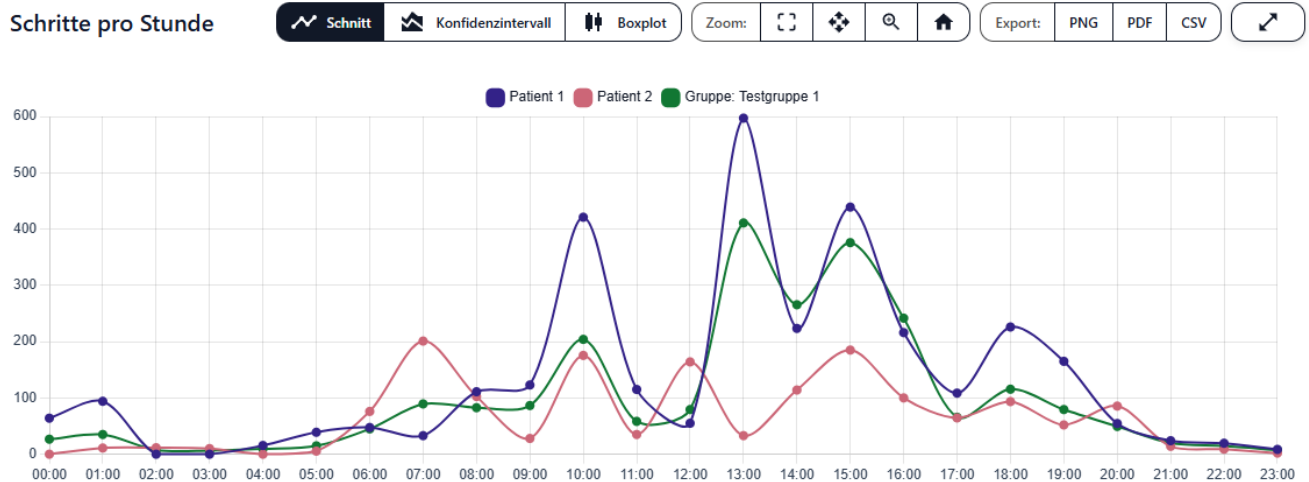


Abbildung 25 - Chart-Container Schritte pro Stunde

Alle Chart.js-basierten Diagramme bieten zusätzlich folgende Interaktionen:

6.6.1.3 Hover-Effekte:

- Tooltips zeigen detaillierte Werte beim Überfahren von Datenpunkten
- Anzeigen des Zeitwerts

6.6.1.4 Legende:

- Klickbare Legendeneinträge zum Ein-/Ausblenden einzelner Datenreihen (Standard von Chart.js). Zusätzlich wurden Ergänzungen in der Legende vorgenommen, um die Usability zu erhöhen (z.B. Cursor Pointer).
- Farbcodierung entspricht der im Color-Picker gewählten Farbe

6.6.1.5 Zusatzinformationen

Alle Diagramm Ansichten zeigen zusätzliche Informationen unterhalb des Chart-Containers an. Darin werden Informationen zu den Patienten oder Gruppen bereitgestellt, welche evtl. noch nicht im Diagramm zu sehen sind.



Abbildung 26 – Zusatzinformationen

6.6.2 Schritte pro Stunde

Diese Ansicht zeigt die durchschnittlich absolvierten Schritte über den ausgewählten Zeitraum.

Schritte pro
Stunde

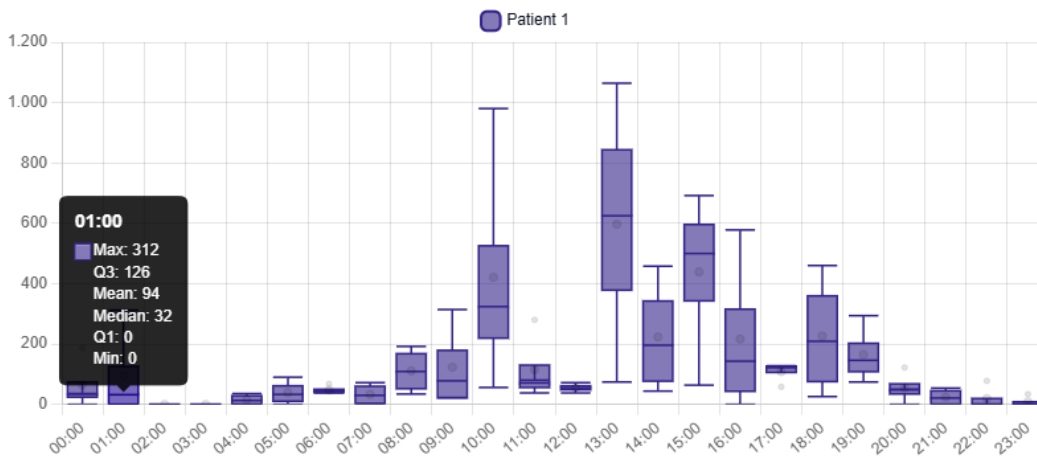
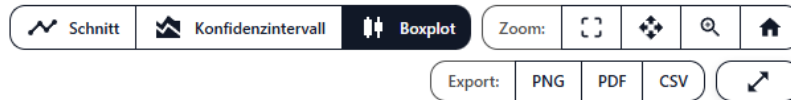


Abbildung 27 - Schritte pro Stunde - Boxplot

Die Boxplot-Darstellung zeigt Median, Quartile und Ausreisser für eine detailliertere statistische Analyse.

Schritte pro
Stunde

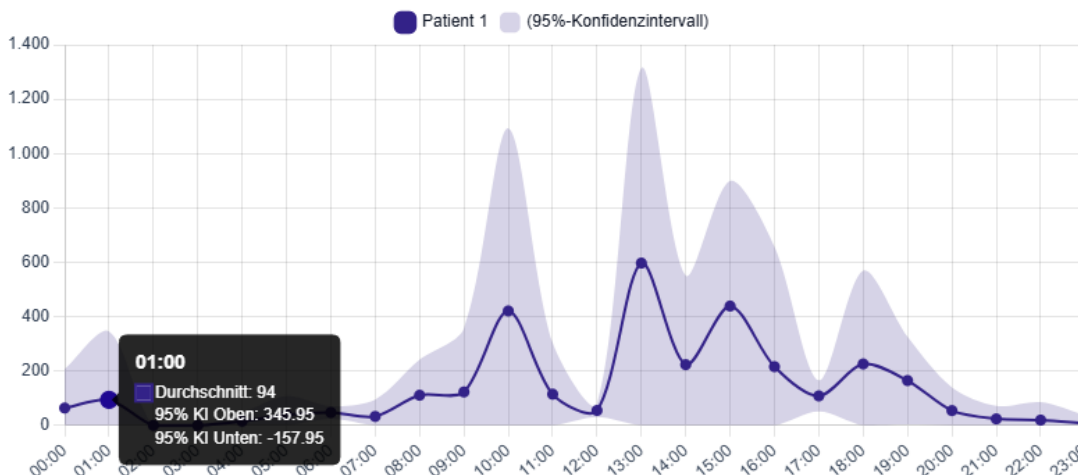


Abbildung 28 - Schritte pro Stunde - Konfidenzintervall

Die Konfidenzintervall-Darstellung zeigt den Durchschnitt mit 95%-Konfidenzintervall als schattierter Bereich.

6.6.2.1 Kadenzfilter:

- Vordefinierte Presets (75, 100 spm) als Schnellauswahl
- Custom-Filter mit Eingabefeldern für Min/Max-Werte
- Vergleichsoperatoren ($\geq$, $\leq$, Bereich) als Dropdown
- "Filter hinzufügen"-Button für mehrere gleichzeitige Filter

6.6.3 Aktivitäten über den Tag



Abbildung 29 - Liegen über den Tag

Diese Visualisierung zeigt die zeitliche Verteilung verschiedener Aktivitäten über 24 Stunden.

6.6.3.1 Dargestellte Aktivitäten:

- Gehen
- Stehen
- Sitzen
- Liegen
- Fahrrad
- Reisen
- Nicht getragen

Die Y-Achse zeigt die Anzahl Minuten pro Stunde (0-60), die X-Achse die Tagesstunden (0-23). Aktivitätstypen sind als Tabs dargestellt, nur eine Aktivität wird gleichzeitig visualisiert

6.6.4 Sitzphasen pro Tag



Abbildung 30 - Sitzphasen pro Tag

Diese Ansicht analysiert die Häufigkeit und Dauer von Sitzphasen. Anzeigen werden in Boxplots visualisiert, welche beim Hovern weitere Informationen, wie Median, Q1-Q3 etc. liefern.

6.6.4.1 Dargestellte Metriken:

- Anzahl Sitzphasen ≥ 30 Minuten
- Anzahl Sitzphasen ≥ 60 Minuten

6.6.5 Schritte mit Kadenz



Abbildung 31 - Schritte mit Kadenz

Diese Ansicht vergleicht die Schrittzahl in verschiedenen Kadenz-Bereichen mittels Boxplots.

6.6.5.1 Standard-Darstellung:

- Totale Schritte
- Kadenz ≥ 75 spm
- Kadenz ≥ 100 spm

6.6.5.2 Benutzerdefinierte Filter:

Über das Kadenzfilter-Panel können beliebige Kadenz-Bereiche definiert werden (z.B. 50-100 spm), die dann als zusätzliche Boxplots dargestellt werden.

6.6.6 Vergleich täglicher Durchschnittswerte

Tagesdurchschnittsvergleich

[PDF](#) [CSV](#)

Gruppe (Mitglieder) / Patient	Gruppe: Testgruppe 1 (3)	Patient 1
Zeit aufrecht (min/Tag)	238 min	350 min +47.0%, +112 min
Zeit liegend (min/Tag)	536 min	573 min +7.0%, +37 min
Zeit sitzend (min/Tag)	656 min	511 min -22.1%, -145 min
Sitz zu Steh Wechsel pro Tag	68	71 +4.0%, +3
Anzahl Sitzphasen > 30min pro Tag	5.4	2.8 -49.2%, -2.7
Anzahl Sitzphasen > 60min pro Tag	2.3	0.0 -100.0%, -2.3
Zeit Sitzend in Sitzphasen > 30min (min/Tag)	336 min	115 min -65.8%, -221 min
Zeit Sitzend in Sitzphasen > 60min (min/Tag)	195 min	0 min -100.0%, -195 min
Zeit gehend (min/Tag)	33 min	43 min +29.9%, +10 min
Schritte pro Tag	2391	3200 +33.8%, +809
Schritte mit Kadenz ≥ 75 pro Tag	1327	1817 +36.9%, +490
Schritte mit Kadenz ≥ 100 pro Tag	276	659 +138.7%, +383
Zeit Fahrrad (min/Tag)	0 min	1 min +200.0%, +1 min
Fahrradschritte pro Tag	38	115 +200.0%, +77

Abbildung 32 – Tagesdurchschnittsvergleich

Diese Ansicht ermöglicht den direkten Vergleich zwischen Patienten und/oder Gruppen in einer tabellari-schen Übersicht.

6.6.6.1 UX-Überlegungen:

- **Tabellen-Layout:** Übersichtliche Matrix mit Metriken in Zeilen, Patienten/Gruppen in Spalten
- **Färbung:** Werte werden farblich kodiert (Grün = gut, Rot = bedenklich)
- **Export optimiert:** Diese Ansicht eignet sich besonders für CSV-Export zur Weiterverarbeitung

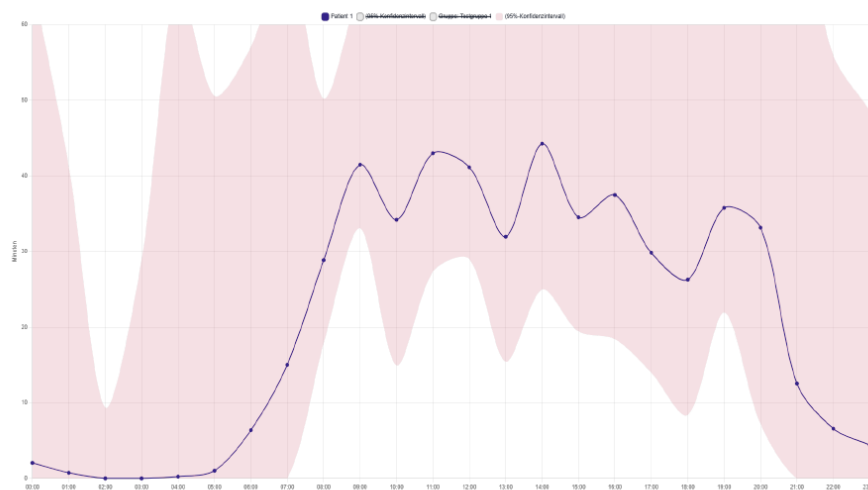
6.7 Export-Funktionen

Alle Diagramme bieten folgende Export-Optionen:

6.7.1 PDF-Export:

- Enthält das Diagramm als Grafik
- Zusatzinformationen als Liste
- Korrekter Umbruch bei mehreren Patienten

Sitzen über den Tag (Minuten pro Stunde)



Patient 1

Anzahl Tage: 4.0
Total Sitzen (Min): 2044.0
Total Sitzen (Std): 34.1
Durchschnitt pro Tag (Min): 511.0
Median pro Tag (Min): 506.0
Standardabweichung (Min): 71.0

Gruppe: Testgruppe 1

Anzahl Tage: 8.0
Total Sitzen (Min): 5638.0
Total Sitzen (Std): 94.0
Durchschnitt pro Tag (Min): 705.0
Median pro Tag (Min): 669.0
Standardabweichung (Min): 218.0

Abbildung 33 - PDF Export

6.7.2 PNG-Export:

- Diagramm als Grafik
- Weisser Hintergrund
- Keine Zusatzinformationen

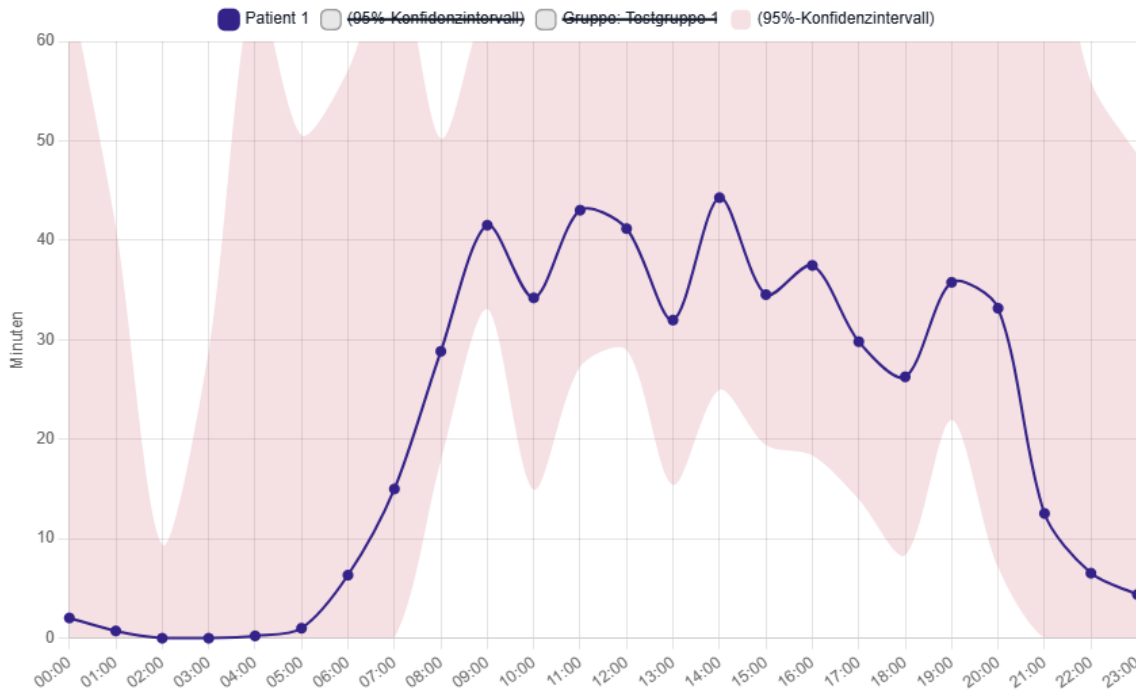


Abbildung 34 - PNG Export

6.7.3 CSV-Export:

- Rohdaten aller dargestellten Datenpunkte in tabellarischer Form
- Bei Konfidenzintervall-Modus: zusätzliche Spalten für obere/untere Grenze
- Keine Zusatzinformationen

Patient/Gruppe	Label	Min	25% Quantil	Median	Durchschnitt	75% Quantil	Max
Gruppe: Testgruppe 1	00:00	0	0	1.71333333	13.32833333	4.92833333	60
Gruppe: Testgruppe 1	01:00	0	0	0	1.56933333	0	7.84666667
Gruppe: Testgruppe 1	02:00	0	0	0	0.434	0	2.17
Gruppe: Testgruppe 1	03:00	0	0	0	0.36866667	0	1.84333333
Gruppe: Testgruppe 1	04:00	0	0	0	11.98633333	0.52833333	59.40333333
Gruppe: Testgruppe 1	05:00	0	0.45666667	0.785	1.37433333	2.06833333	3.56166667
Gruppe: Testgruppe 1	06:00	0.27666667	2.64833333	3.345	6.67133333	8.335	18.7516667
Gruppe: Testgruppe 1	07:00	0.53333333	4.07666667	32.0866667	27.572	50.3183333	50.845
Gruppe: Testgruppe 1	08:00	24.1066667	29.0583333	29.94	32.4063333	33.3266667	45.6
Gruppe: Testgruppe 1	09:00	37.4883333	40.87	44.4566667	45.9643333	49.6616667	57.345
Gruppe: Testgruppe 1	10:00	18.305	25.8933333	35.3933333	36.5573333	45.5116667	57.6833333
Gruppe: Testgruppe 1	11:00	38.27	41.985	50.9016667	48.4476667	51.4883333	59.5933333
Gruppe: Testgruppe 1	12:00	27.7933333	41.6483333	42.1033333	42.1856667	48.7366667	50.6466667

Abbildung 35 - CSV Export

6.8 Feedback und Fehlermeldungen

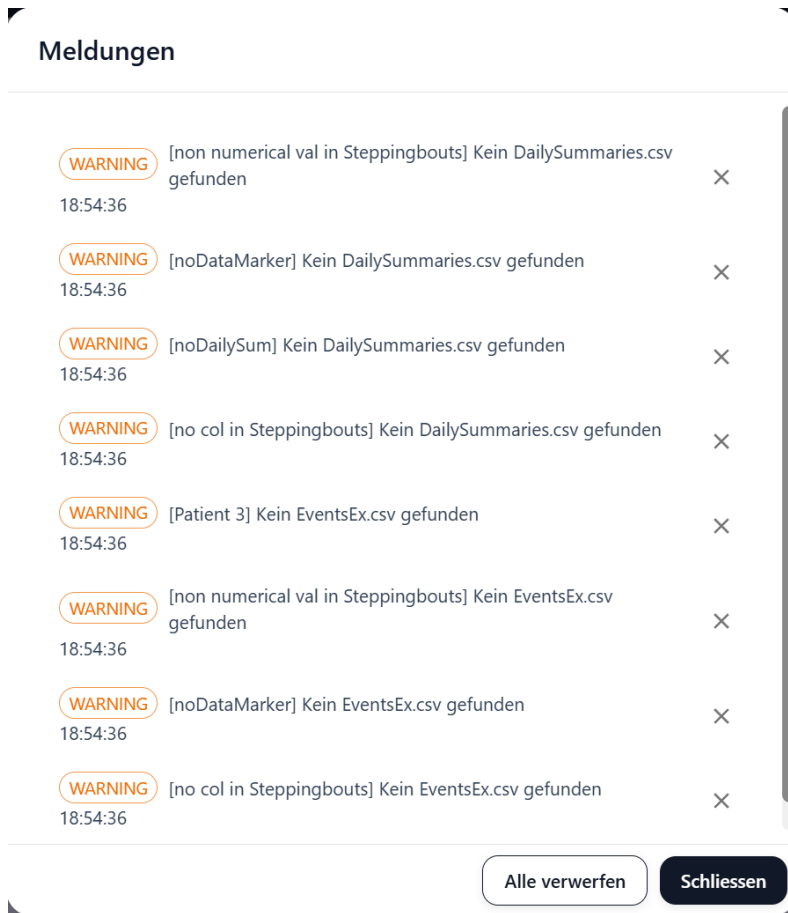


Abbildung 36 – Mitteilungen

Die Anwendung gibt klare Rückmeldungen bei:

- Erfolgreichen Operationen (Import, Export, Aktualisierung der Patienten)
- Fehlerhaften oder fehlende Dateien

6.8.1 Arten von Feedback:

Snackbar-Benachrichtigungen:

- Kurze Bestätigungen für erfolgreiche Aktionen (3-5 Sekunden sichtbar)
- Positioniert am unteren Rand mittig
- Farbcodierung: Grün = Erfolg, Blau = Info

Nachricht im Mitteilungscenter:

- Warnungen und Fehler
- Beschreibung wo, wann und wie das Problem/Fehler aufgetreten ist
- Nachrichten können wieder verworfen werden
- Das Mitteilungscenter ist von überall aufrufbar, wenn Warnungen/Fehler vorhanden sind

6.9 Responsive Design und Barrierefreiheit

Die Benutzeroberfläche passt sich verschiedenen Fenstergrößen an und gewährleistet:

- **Farbkontraste (WCAG 2.1, Punkt 1.4.1):** Gruppen und Patienten werden per Standard mit barrierefreien Farben dargestellt in Diagrammen. Petroff zeigt, dass optimal gewählte Farbsequenzen sowohl ästhetische als auch Accessibility-Anforderungen erfüllen müssen [6]. Um den unterschiedlichen Anforderungen medizinischen Fachpersonals gerecht zu werden, bietet die Anwendung zusätzlich einen Color-Picker, mit dem Benutzer die Farbzuoordnung individuell anpassen können.
- **Flexible Panels:** Navigation kann ein-/ausgeklappt werden für mehr Platz im Diagrammbereich
- **Draggable-Header:** Der gesamte Header kann für Verschiebung des Fensters genutzt werden
- **Vollbildmodus:** Diagramme können in Vollbild dargestellt werden
- **Mindestbreite:** Die minimale Darstellungsbreite der Anwendung ist abhängig vom Skalierungsfaktor und beträgt bei einer Skalierung von 100% 600 Pixel. Für diese Breite ist die Ansicht der Diagramme ohne horizontale Scroll Bar garantiert. Bei der Patienten- und Gruppenansicht kann eine horizontale Scroll Bar auftreten, falls lange Namen verwendet wurden.

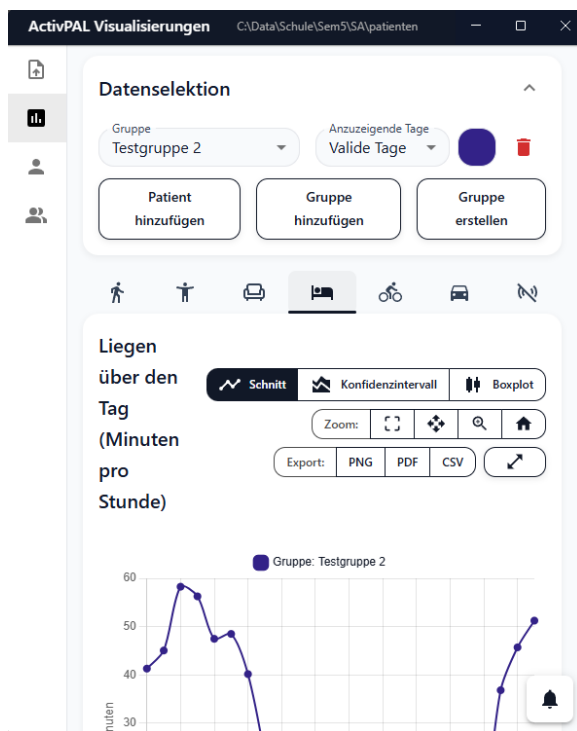


Abbildung 37 - Mindestbreite Layout

6.10 Performance und User Experience

Gefühlte Performance:

- Einfache Operationen wirken instantan (Filter ändern, Tabs wechseln)
- Längere Operationen zeigen Ladeanzeige (Komplexes Diagramm laden)

Optimierungen:

- **Lazy Loading:** Diagramme werden erst beim Öffnen gerendert
- **Memoization:** Teure Berechnungen werden gecacht und nur bei Änderung neu ausgeführt (z.B. CSV laden und parsen)

6.11 Zukünftige UX-Verbesserungen

Folgende Features könnten die User Experience weiter verbessern:

Onboarding:

- Interaktive Tour beim ersten Start
- Tooltips für neue Features
- Beispieldatensatz zum Ausprobieren

Personalisierung:

- Themes (Hell/Dunkel-Modus)
- Anpassbare Tastatur-Shortcuts

Chart:

- Der Scroll Zoom ist nicht ganz ausgereift und könnte verbessert werden. Es fehlt die Kontrolle, welche Achse gezoomt wird.
- Das Ausblenden der Datenreihen ist beim ersten Mal nicht intuitiv. Es könnte eine Komponente mit Checkbox erstellt werden, welche die Auswahl der angezeigten Datenreihen speichert. Zurzeit wird bei jeder Änderung in der Datenselektion immer alles wieder eingeblendet.

7 Erreichte Ziele / Resultate

Die Entwicklung des ActivPAL Data Visualizers wurde anhand der definierten funktionalen (FR) und nicht-funktionalen Anforderungen (NFR) strukturiert. Die Umsetzung wurde durch End-to-End-Tests (E2E) systematisch validiert, die direkt von den Anforderungen abgeleitet wurden.

7.1 Funktionale Anforderungen

7.1.1 Nicht erreichte Anforderungen

- **FR-V4 (P3):** Die ursprünglich geplante 24h-Timeline-Darstellung des Tagesverlaufs wurde nicht wie spezifiziert umgesetzt (E2E-V-3: nicht bestanden). Stattdessen wurde die Visualisierung "Aktivitäten über den Tag" (FR-V13) entwickelt, die eine detailliertere und praxisgerechtere Darstellung der Aktivitätsverteilung bietet. Diese Entscheidung wurde in Abstimmung mit dem Praxispartner getroffen und stellt eine funktionale Verbesserung dar.

7.1.2 Erreichte Anforderungen

Alle übrigen funktionalen Anforderungen wurden erfolgreich umgesetzt. Besonders hervorzuheben ist:

- Das vollständige **MVP** (alle Priorität-1-Anforderungen) wurde erreicht
- Alle **Priorität-2-Anforderungen** zur Gruppenverwaltung, Visualisierung und Export wurden realisiert
- Mehrere **Priorität-3-Anforderungen** konnten ebenfalls implementiert werden, darunter die Persistenz des Diagrammzustands (FR-V10) und erweiterte Visualisierungsoptionen

7.2 Nicht-funktionale Anforderungen

7.2.1 Nicht erreichte Anforderungen

- **NFR-P3 (Performance):** Die Startzeit der Anwendung liegt inkonsistent bei 4-5 Sekunden statt der geforderten ≤ 3 Sekunden (E2E-P-3: nicht bestanden). Die RAM-Spitzen bei Gruppenanalysen bleiben jedoch wie gefordert unter 1.5 GB. Die verlängerte Startzeit ist primär auf den Electron-Framework-Overhead zurückzuführen und beeinträchtigt die Nutzbarkeit nur minimal.

7.2.2 Erreichte Anforderungen

Alle übrigen nicht-funktionalen Anforderungen wurden erfolgreich erfüllt, darunter:

- **Usability:** Der effiziente Workflow (≤ 3 Schritte vom Import zur Visualisierung) und die responsive Benutzeroberfläche mit WCAG 2.1-konformen Farbkontrasten
- **Reliability:** Robuste Fehlerbehandlung bei ungültigen Dateien und gesperrten Ressourcen
- **Security:** Keine dauerhafte Speicherung personenbezogener Daten und datenschutzkonforme Logs
- **Portability:** Lauffähigkeit auf Windows 11, Installation ohne Administratorrechte und verlustfreie Updates

7.3 Zusammenfassung

Von den insgesamt 28 funktionalen Anforderungen wurden 27 erfolgreich umgesetzt. Von den 13 nicht-funktionalen Anforderungen wurden 12 vollständig erfüllt.

Das Projekt kann als erfolgreich bewertet werden: Das MVP wurde vollständig erreicht und deutlich übertraffen. Die beiden nicht vollständig erfüllten Anforderungen (FR-V4, NFR-P3) stellen keine wesentlichen Einschränkungen der Funktionalität dar – im Fall von FR-V4 wurde sogar eine funktional überlegene Alternative implementiert.

Die systematische Validierung durch E2E-Tests hat sich als effektive Qualitätssicherungsmassnahme erwiesen und ermöglichte eine objektive Bewertung des Projekterfolgs.

7.4 Ausblick und Erweiterungsmöglichkeiten

Die entwickelte Anwendung bietet eine solide Grundlage für zukünftige Erweiterungen. Das Entwicklungsteam sieht folgende Funktionen als mögliche Erweiterungen in künftigen Versionen:

7.4.1 Zentrale Datenverwaltung

Ein zentraler Server mit Rollenverwaltung würde die Zusammenarbeit mehrerer Forschungsteams ermöglichen. Verschiedene Benutzerrollen (Administrator, Forscher, Viewer) könnten unterschiedliche Zugriffsrechte auf Patientendaten und Analysen erhalten. Dies würde besonders bei multizentrischen Studien die Datenverwaltung vereinfachen und gleichzeitig den Datenschutz durch granulare Berechtigungskonzepte verbessern.

7.4.2 KI-gestützte Analysen

Die Integration von Machine Learning könnte automatische Mustererkennung in Aktivitätsprofilen ermöglichen. Beispielsweise könnten Anomalien in Bewegungsmustern erkannt, Schlafqualität bewertet oder Vorhersagen über Gesundheitstrends getroffen werden.

7.4.3 Erweiterte Sensorunterstützung

Die Applikation könnte erweitert werden, um Daten von weiteren Sensoren zu verarbeiten. Ein einheitliches Datenmodell würde den Vergleich verschiedener Messgeräte ermöglichen und die Anwendung für ein breiteres Spektrum an Forschungsfragen nutzbar machen.

7.4.4 Activity Score Visualisieren

Diese Erweiterungsmöglichkeit kommt vom Praxispartner, wurde jedoch zu spät eingebracht und konnte aus Zeitgründen nicht mehr erledigt werden. Es geht darum einen Score zu visualisieren, welcher die körperliche Aktivität nach MET (Metabolisches Äquivalent) in den Kategorien leicht, moderat und hohe körperliche Aktivität zeigt. Im Events.csv der Patienten ist bereits ein Activity Score vorhanden, jedoch ist zum jetzigen Zeitpunkt nicht klar, ob dies der gewünschte Score ist. In Zukunft könnte dies abgeklärt werden und eine Visualisierung erstellt werden, welcher die gewünschten Kategorien anzeigt pro Patienten und Gruppe.

8 Projektmanagement

8.1 Projektplan-SOLL

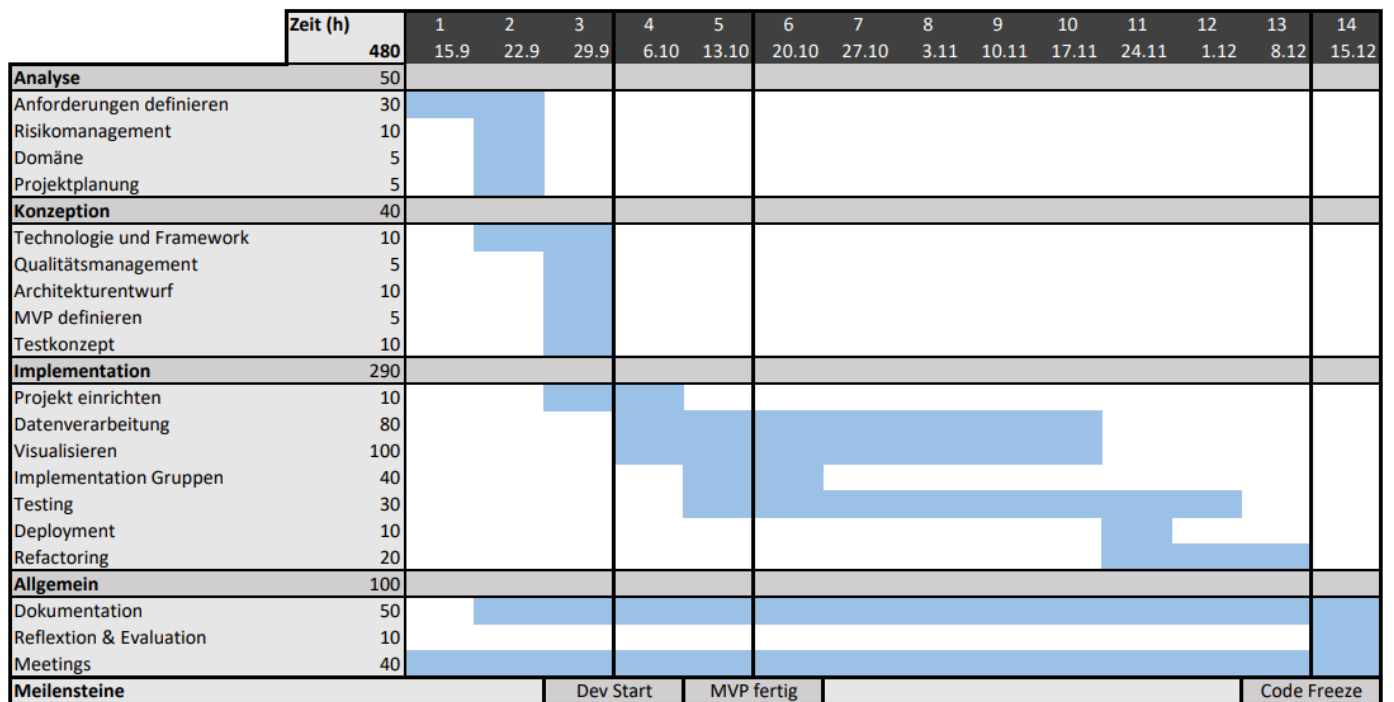


Abbildung 38 - Projektplan

Zusätzlich relevante Daten:

Kickoff: 16.09.2025

Abgabe: 19.12.2025

8.2 Zeitplan-IST

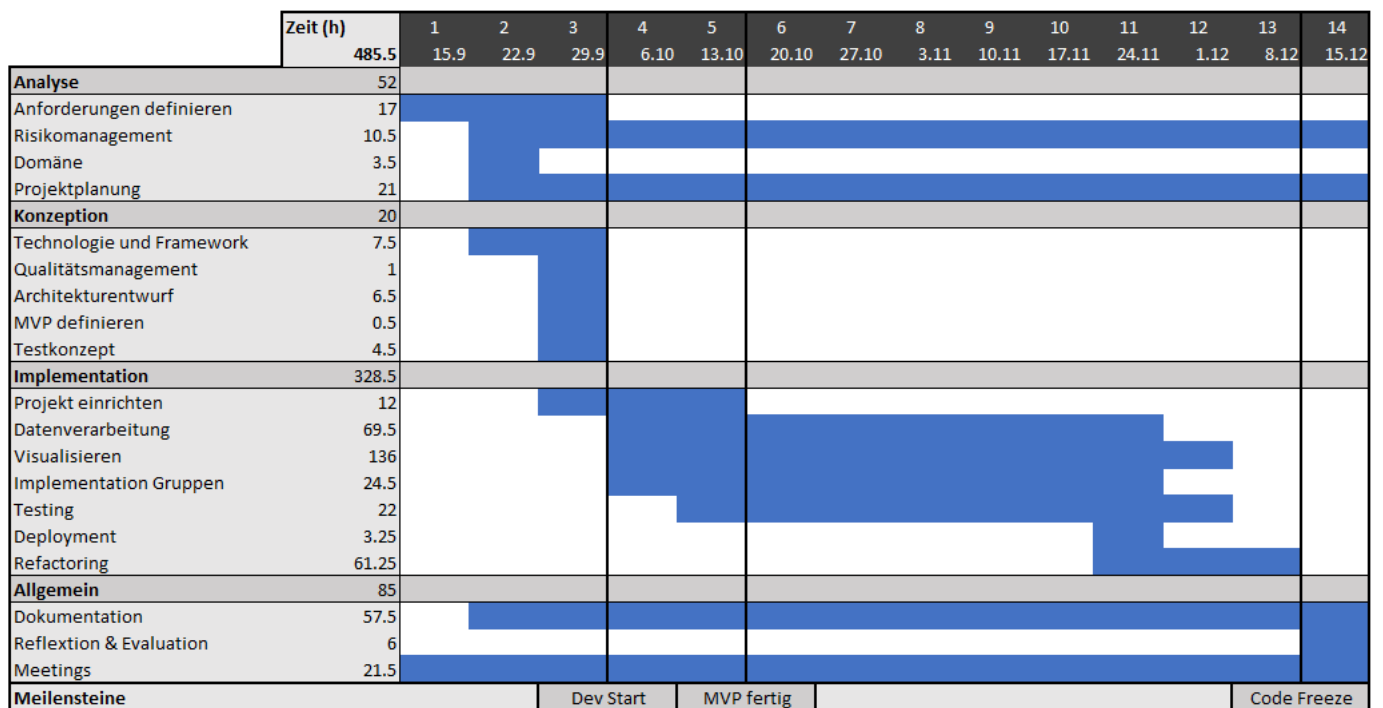


Abbildung 39 - Zeitplan-IST

8.3 Vergleich & Diskussion Zeitplan

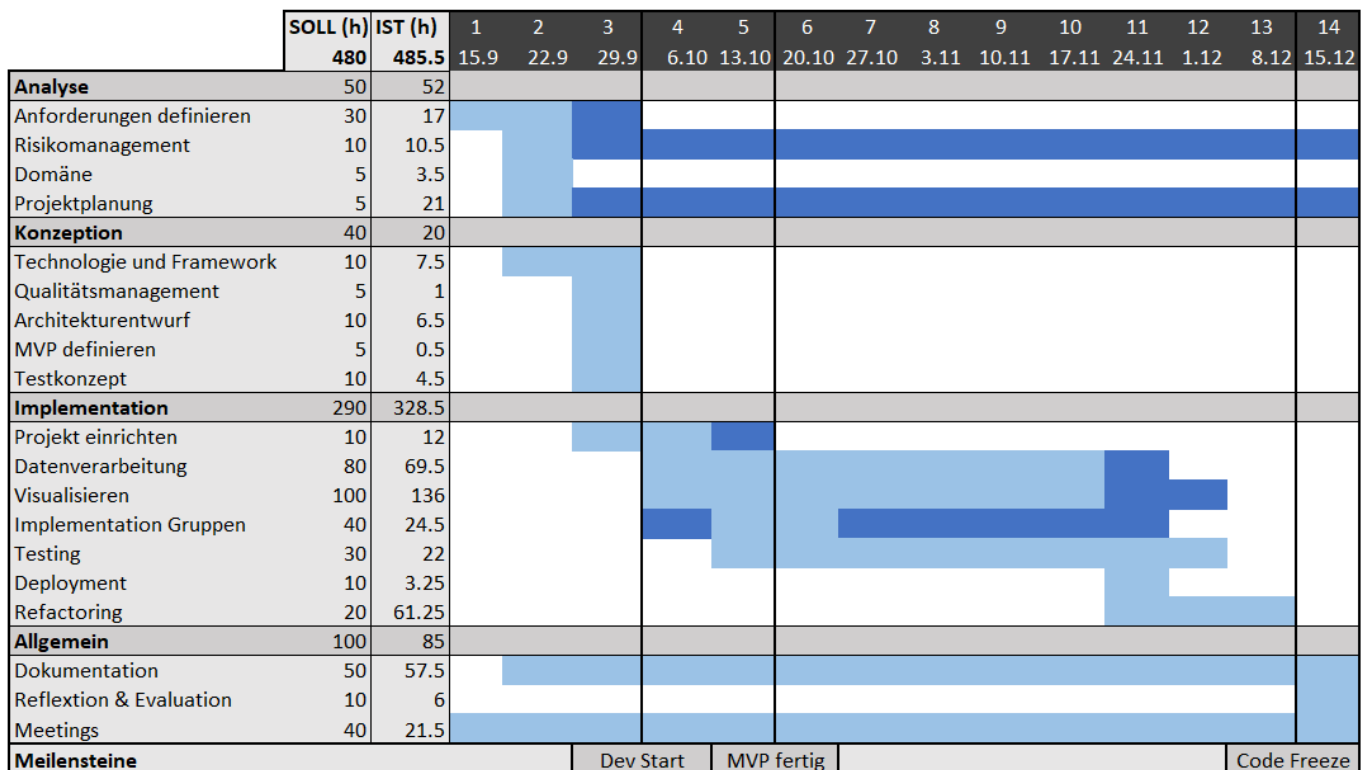


Abbildung 40 - Zeitplan Vergleich IST/SOLL

Die Gegenüberstellung von Soll- und Ist-Zustand zeigt insgesamt eine präzise Zeitplanung, wobei sich einzelne nachvollziehbare Abweichungen ergeben haben. Die Gesamtprojektdauer von 485 Stunden lag nur knapp über der geplanten Zeit von 480 Stunden, was einer Abweichung von lediglich 1% entspricht und auf eine realistische Einschätzung des Projektumfangs hindeutet.

Die grössten Unterschiede zeigen sich erwartungsgemäss beim Risikomanagement (10.5h statt 10h) und der Projektplanung (21h statt 5h). Diese Abweichungen sind darauf zurückzuführen, dass in der initialen Zeitplanung nicht berücksichtigt wurde, dass Sprint-Planning zur Projektplanung gehört und sich über alle Sprints erstreckt.

Ebenso wurde das Risikomanagement als kontinuierlicher Prozess durchgeführt, bei dem die Risiken zu Beginn jedes Sprints neu bewertet wurden, statt dies nur einmalig zu Projektbeginn durchzuführen.

Auch in der Implementierungsphase zeigen sich Verschiebungen: Die Visualisierung benötigte mit 136 Stunden deutlich mehr Zeit als geplant (100h), während die Datenverarbeitung mit 64.5 Stunden unter den veranschlagten 80 Stunden blieb.

Interessanterweise konnten diese Mehraufwände durch Effizienzgewinne in anderen Bereichen kompensiert werden, so wurde beim Testing (22h statt 30h) und bei der Implementation Gruppen (24.5h statt 40h) Zeit eingespart.

Insgesamt demonstriert die geringe Gesamtabweichung eine erfolgreiche Projektplanung und -durchführung, bei der die wesentlichen Meilensteine im vorgesehenen zeitlichen Rahmen erreicht wurden und Mehraufwände durch effiziente Arbeit in anderen Projektphasen ausgeglichen werden konnten.

8.4 Zeiterfassung pro Person

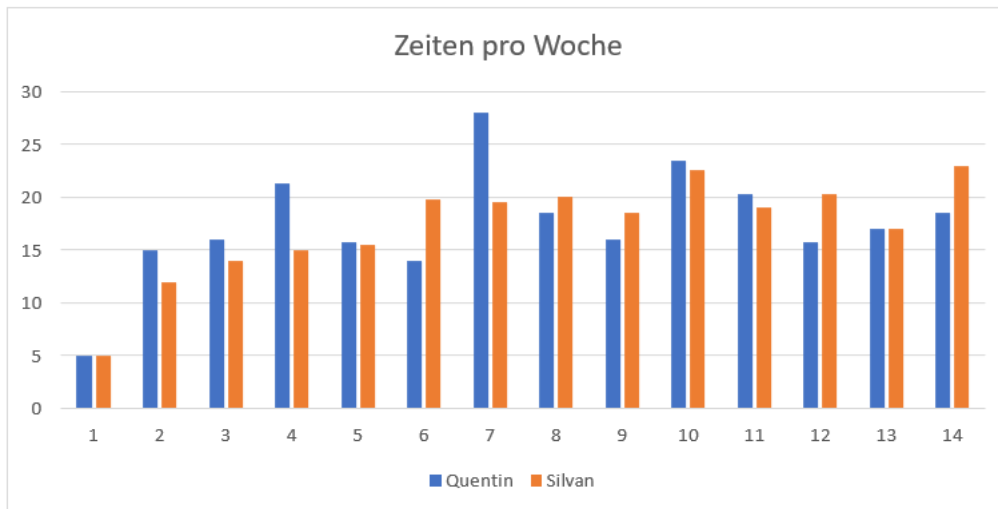


Abbildung 41 - Zeiten pro Person pro Woche

Tabelle 2 - Stunden pro Woche

Woche	Quentin	Silvan
1	5	5
2	15	12
3	16	14
4	21.25	15
5	15.75	15.5
6	14	19.75
7	28	19.5
8	18.5	20
9	16	18.5
10	23.5	22.5
11	20.25	19
12	15.75	20.25
13	17	17
14	18.5	23
Total	244.5	241

8.5 Risikomanagement

Die folgenden Risiken wurden zu Beginn des Projekts identifiziert und werden hier dokumentiert sowie fortlaufend aktualisiert. In dieser Abbildung sind die Änderungen aller Risiken zusammengefasst. Dabei repräsentiert der Wert das Produkt aus Schaden und Wahrscheinlichkeit des Risikos. Beide Werte werden im Bereich 0 (unwahrscheinlich, wenig Schaden) bis 10 (Sehr wahrscheinlich, grosser Schaden) eingestuft.

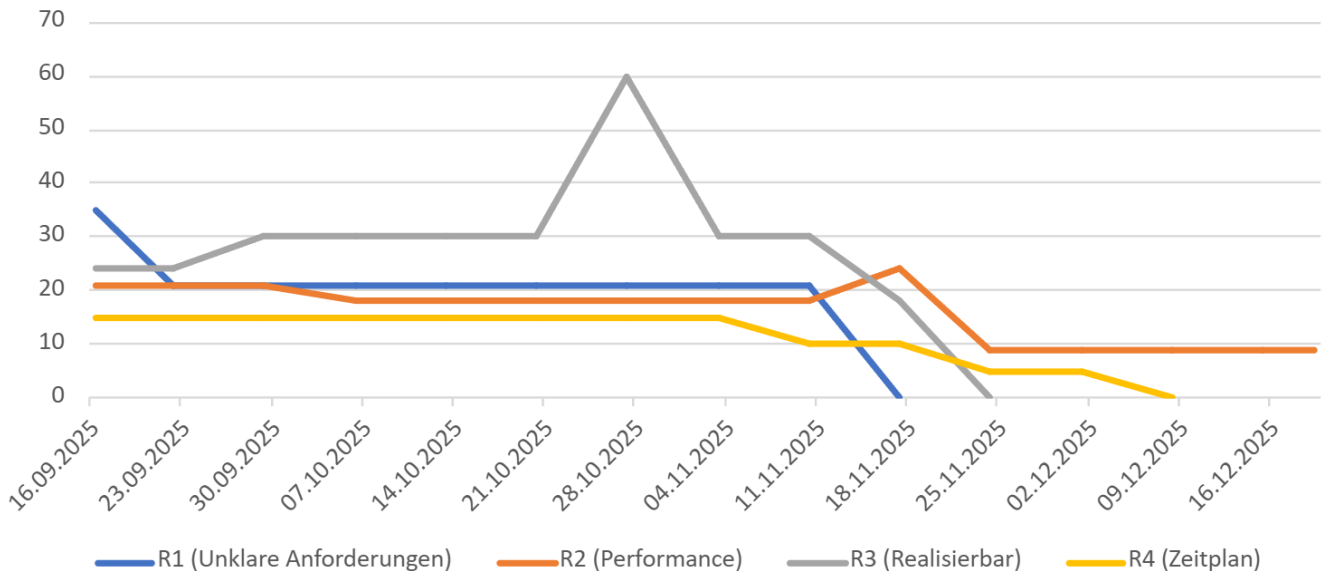


Abbildung 42 - Verlauf der Risiken

8.5.1 R1: Unklare Anforderungen

Risikolevel zu Beginn: Mittel

Risikolevel aktuell: Behoben

Auswirkung: Falls Anforderungen (z. B. zu den Visualisierungen, Gruppendefinitionen, Aggregationen) nicht präzise genug geklärt sind, können Verzögerungen durch Nachbesserungen entstehen.

Massnahmen:

- Anforderungen in Form von FR/NFR und MVP präzisieren und dokumentieren.
- Regelmässige Rücksprache mit Betreuer und Praxispartner halten.

Einschätzung zu Beginn: Schaden (S): 7, Wahrscheinlichkeit (W): 5

22.09.2025	Das zweite Meeting sorgte für einige Klarheit bezüglich der Aufgabenstellung. Fragen wurden mit dem Betreuer und Praxispartner geklärt.	W: 3 S: 7
17.11.2025	Die Anforderungen wurden nach der ersten Durchführung der Usability Tests mit Praxispartner erneut besprochen. Dabei wurden die Anforderungen neu spezifiziert.	W: 0 S: 7

8.5.2 R2: Performance bei grösseren Datensätzen

Risikolevel zu Beginn: Mittel

Risikolevel aktuell: Niedrig

Auswirkung: Die Analyse von Gruppen mit vielen Patienten oder langen Aufzeichnungszeiträumen könnte langsamer als geplant sein, was die Nutzerfreundlichkeit beeinträchtigt. Es ist schwierig zu planen, da noch keine Erfahrungen mit Parsing so grosser Dateien gesammelt wurden.

Massnahmen:

- Frühes testen mit evtl. synthetisch erzeugten Datensätzen
- Messung von Parsing- und Visualisierungszeiten
- Falls nötig Optimierungen machen mit Inkrementeller Anzeige.

Einschätzung zu Beginn: Schaden (S): 3, Wahrscheinlichkeit (W): 7

06.10.2025	Das Lesen der CSV-Dateien wurde analysiert. Die grösste Datei, die in diesem Projekt zur Verwendung geplant ist, braucht ca. 2-3 Sekunden, um zu lesen. Da diese Datei nur für die Tagesansicht eines Patienten auf einmal verwendet wird, sollte dies zu keinen Performance-Problemen führen, solange keine zu langen Datenansammlungen vorhanden sind. Die meisten zu verwendenden Dateien sind sehr klein und haben keine merkbare Verzögerung.	W: 6 S: 3
17.11.2025	Anstelle des Timeline Diagramms soll ein anderes Diagramm für die Aktivitäten verwendet werden. Da dieses ein grosses CSV lesen muss und Gruppen unterstützt, könnte es zu Performance-Problemen kommen.	W: 8 S: 3
24.11.2025	Das Diagramm Aktivitäten pro Stunde wurde umgesetzt. Die Daten werden schneller gelesen als gedacht. Bei ca. 75 Patienten mit Daten über 5 Tage dauert das erste Laden der Gruppe ca. 6 Sekunden. Das Risiko bleibt bestehen, falls grössere Mengen an Daten geladen werden müssen. Da dagegen nichts gemacht werden kann und der Nutzer weiss, dass er grosse Mengen an Daten lädt, wird das Risiko jedoch als niedrig eingestuft.	W: 3 S: 3

8.5.3 R3: Ändernde Anforderungen nicht mit Technologie / Framework realisierbar

Risikolevel zu Beginn: Mittel

Risikolevel aktuell: Behoben

Auswirkung: Die Anforderungen können sich im Laufe des Projekts ändern. Das Risiko besteht, dass die gewählte Technologie bzw. das gewählte Framework nicht in ihren Möglichkeiten ausreichend ist, um die neuen Anforderungen abzudecken.

Massnahmen:

- Flexibler Techstack mit loser Kopplung

Einschätzung zu Beginn: Schaden (S): 6, Wahrscheinlichkeit (W): 4

29.09.2025	Im Meeting wurde besprochen, das in dieser Arbeit keine Frameworks oder Libraries ohne MIT ähnliche Lizenz verwendet werden sollen. Daher wird die Auswahl etwas eingeschränkt, jedoch gibt es trotz dessen passende Optionen.	W: 5 S: 6
22.10.2025	Beim Code Review 1 teilte Mathias Lenz mit, dass React Applikationen bei der OST mit TypeScript entwickelt werden, weil dies besser wartbar ist. Somit ist dieses Risiko leider eingetroffen, da die gewählte Technologie JavaScript, die Anforderungen nicht länger erfüllen kann. Sofort wurde mit der Umstellung auf TypeScript begonnen.	W: 10 S: 6
03.11.2025	Die Umstellung auf Typescript wurde abgeschlossen. Der Technologie-Stack ist wieder in der Lage alle Anforderungen abzudecken.	W: 5 S: 6
17.11.2025	Da die Visualisierung mit Tagesverlauf (Timeline-Diagramm) durch ein neues Diagramm ersetzt wurde, sinkt die Wahrscheinlichkeit. Das neue Diagramm ist ebenfalls ein Linien und Boxplot Diagramm, was mit dem Techstack schon umgesetzt wurde.	W: 3 S: 6
24.11.2025	Alle Anforderungen und Visualisierungen konnten erfolgreich umgesetzt werden.	W: 0 S: 6

8.5.4 R4: Zeitplan / Umfang

Risikolevel zu Beginn: Mittel**Risikolevel aktuell:** Behoben

Auswirkung: Durch den vollen Funktionsumfang (Import, Gruppen, diverse Visualisierungen) besteht das Risiko, dass nicht alle Anforderungen in der verfügbaren Zeit realisiert werden können.

Massnahmen:

- Klare Priorisierung (MVP = P1)
- Fokus auf P1+P2, optionale Features (P3) nur bei Zeitüberschuss.
- Pufferzeiten einplanen

Einschätzung zu Beginn: Schaden (S): 5, Wahrscheinlichkeit (W): 3

23.09.2025	Im Bereich Implementation & Refactoring mehr Zeit eingeplant, welche als Puffer dient.	W: 3 S: 5
10.11.2025	Erste Usability Tests können durchgeführt werden, da mehrere Visualisierungen bereits fertiggestellt wurden.	W: 2 S: 5
24.11.2025	Alle geplanten Visualisierungen wurden umgesetzt.	W: 1 S: 5
09.12.2025	Die Letzte Usability Tests wurden durchgeführt und das Feedback ist ohne viel Aufwand umzusetzen.	W: 0 S: 5

9 Qualitätsmanagement

9.1 Regeln

In diesem Kapitel sind die Definitionen zu finden, die während der Entwicklung eingehalten werden. Dabei geht es um die Definition of Ready, Definition of Done, Code Guidelines und Regeln zur Verwendung von GitLab.

9.1.1 Definition of Ready

Ein Ticket kann begonnen werden, sobald es:

- Im Sprint Backlog liegt
- Mit Story Points geschätzt wurde
- Keine offenen Abhängigkeiten hat

9.1.2 Definition of Done

Ein Ticket kann abgeschlossen werden, sobald es:

- Die angegebenen Aufgaben erfüllt
- Tests dafür geschrieben wurden (sofern möglich)
- Die GitLab Pipeline durchläuft (Tests und ESLint)
- Die Anwendung fehlerfrei läuft
- Die benötigte Zeit auf dem Ticket erfasst wurde
- Folge-Tickets erstellt wurden

9.1.3 Code Reviews

Bei jedem Merge eines Feature-Branches in den main-Branch wird der Code jeweils von der anderen Person überprüft. Dadurch wird sichergestellt, dass alle Änderungen von einer zweiten Person kontrolliert werden. Der Fokus der Reviews liegt auf der Codequalität, der Lesbarkeit, der Einhaltung der Projektstruktur sowie auf möglichen Verbesserungen.

9.1.3.1 Formelle Code- und Architektur-Reviews

Zusätzlich wurden zwei formelle Code- und Architektur-Reviews mit Mathias Lenz durchgeführt. Mathias erhielt jeweils vor dem Termin Zugriff auf das GitLab-Repository und nahm sich etwa einen halben Tag Zeit, um den aktuellen Projektstand zu analysieren. Im anschliessenden Meeting wurden die Erkenntnisse besprochen und konkrete Verbesserungsvorschläge erarbeitet. Die vollständigen Protokolle befinden sich im Anhang unter Meetings → Code Review 1.docx und Code Review 2.docx.

Code Review 1 (22.10.2025) - Hauptkenntnisse:

- Migration von JavaScript auf TypeScript für bessere Wartbarkeit und Typsicherheit
- Einführung von Prettier für konsistente Code-Formatierung
- Etablierung klarer Import-Konventionen und Projektstruktur
- Trennung von Loading- und Compute-Logik bei Operationen

Code Review 2 (03.12.2025) - Hauptkenntnisse:

- Refactoring grosser Komponenten in kleinere, wiederverwendbare Teile
- Migration vom Context-Provider auf Zustand für effizienteres State-Management
- Verbesserung der Type Safety durch Union-Types
- Code-Vereinfachungen und Optimierung der Komponentenstruktur

9.1.4 Code Guidelines

Als Basis wurden die Airbnb Richtlinien für React genommen [7]. Daraus leiten sich unsere eigenen Richtlinien ab:

- Vermeide doppelten Code – Funktionen sollten wiederverwendbar gestaltet werden.
- Behandle Fehlerfälle immer angemessen.
- Halte dich an die bestehenden Code-Konventionen des Projekts.
- Schreibe Tests für deinen Code.
- Namenskonventionen:
 - Variablen und Funktionen: camelCase
 - Komponenten (React): PascalCase
 - Dateien und Ordner: kebab-case
- JSDoc wird erstellt für Komponenten und grössere Funktionen (Deutsch)

9.1.5 Versionskontrolle (GitLab)

- Es soll kein Code gepusht werden, der nicht funktioniert oder die Pipeline nicht besteht.
- Sollte die Pipeline fehlschlagen, ist dies so schnell wie möglich zu korrigieren.
- Erstelle immer einen neuen Branch, ausser Änderungen am main-Branch wurden explizit im Team abgesprochen.
- Commit-Messages sollten den Tag oder Titel des zugehörigen Jira-Tickets enthalten.

9.2 GitLab Pipeline

Die CI/CD-Pipeline wird verwendet, um Qualitätsmessungen automatisiert durchzuführen. Dadurch erhält der Entwickler ein direktes Feedback, nachdem er seine Änderungen pusht.

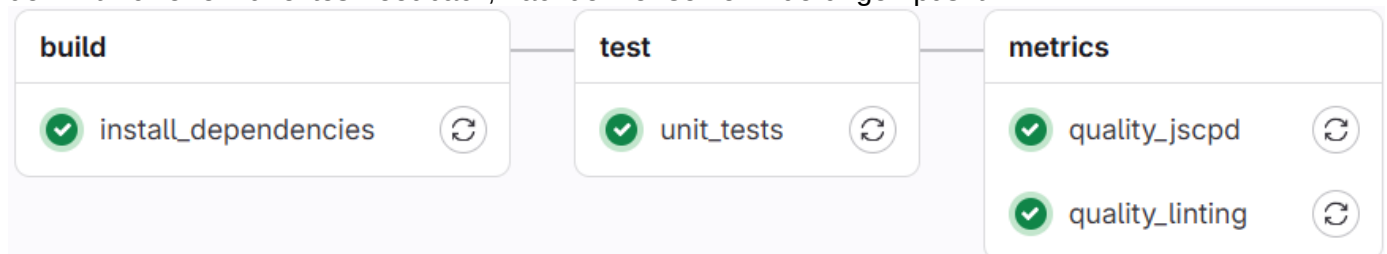


Abbildung 43 - Git Pipeline

Die Pipeline enthält die drei Stages build, test und metrics. In der ersten Stage werden die Abhängigkeiten (dependencies) des Projektes installiert. Diese werden von den anderen Stages benötigt.

9.2.1 Jest Unit-Tests

In der zweiten Stage test werden die Unit Tests ausgeführt. Diese sollen die Logik der Applikation testen. Damit überprüft werden kann, ob genügend Tests geschrieben wurden, gibt die Pipeline einen Coverage-Report aus, der angibt, zu welchem Prozentsatz die einzelnen Klassen durch Tests abgedeckt sind.

9.2.2 JSCPD

In der dritten Stage wird die Qualität des Codes geprüft. Dabei wird zum einen der JavaScript Copy/Paste Detector (jscpd) verwendet. Dieser prüft den Code auf identische Codezeilen und gibt ebenfalls in der GitLab CI einen Report aus. Mit diesem Report können doppelte Codestellen erkannt und zentralisiert werden. Dadurch soll die Qualität des Codes verbessert werden.

9.2.3 ESLint

Zusätzlich läuft ESLint über den JavaScript Code und prüft den Code auf Unschönheiten. So können beispielsweise unbenutzte Variablen erkannt und entfernt werden. Falls ESLint mindestens eine Stelle findet, schlägt der Pipeline Job fehl und der Entwickler erhält eine entsprechende E-Mail. Dadurch können diese Unschönheiten schnell entfernt werden und der Code bleibt konsistent und gut wartbar.

9.3 Deployment

Die Anwendung wurde zwar primär unter Windows entwickelt, kann jedoch plattformübergreifend genutzt werden. Für das Erstellen der Installationspakete einer Electron-Applikation wird jeweils eine Maschine mit dem entsprechenden Betriebssystem benötigt.

- **Windows-Installer** wurde über `npm run make` direkt auf den Entwicklungsrechnern (Windows) erstellt.
- **Linux-Installer** werden in der CI/CD-Pipeline erzeugt, da diese auf Linux läuft.
- **macOS-Installer** kann nur auf einem macOS-System erstellt werden und kann daher nicht bereitgestellt werden.

9.4 Testkonzept

Ziel des Testkonzepts ist es eine Absicherung des Kernflusses zu gewährleisten:

Ordner wählen → CSV importieren → Transformation/Normalisierung → Aggregation erstellen → Diagramme auswählen/anzeigen → Export.

9.4.1 Testpyramide

- **Unit-Tests (80%)** – mit Jest
Fokus: reine Logik ohne OS-Seitenwirkungen.
- **E2E, Usability & Performance (20%)** – manuell
Fokus: realistische Journeys, Messung der NFRs und Edge Cases.

9.4.2 Unit-Tests

Das Ziel der Unit Tests ist eine Sicherstellung der Korrektheit und Robustheit einzelner Bausteine.

- **CSV-Parsing-Funktionen:** Delimiter-/Encoding-Erkennung, Header, leere/fehlerhafte Zeilen, Randfälle.
- **Daten-Transformationen & Normalisierung:** Mapping der Spalten auf internes Schema, Typkonvertierungen, Aggregationen.
- **Gruppenverwaltung:** Logik der Gruppenerstellung und Verwaltung.
- **Chart-Konfigurationen:** Bucketing (Daily/Weekly), Nullbehandlung. Gruppenaggregationen.
- **Export-Logik:** CSV/PDF-Ausgabe und Fehlerfälle.
- **Einzelne React-Komponenten:** korrekte Darstellung der einzelnen Komponenten.

9.4.3 Usability-Tests

Die Usability-Tests dienen dazu, die Benutzerfreundlichkeit aus Sicht des Anwenders zu prüfen. Dazu bekommt ein Anwender der Zielgruppe eine Liste mit Funktionalitäten, welche zu testen sind. Die Zielgruppe beinhaltet Mitarbeiter der MOVE-IT. Folgende Punkte sind zu testen:

- Verständlichkeit Import-Dialog
- Navigation & Orientierung
- Gruppen- & Userverwaltung
- Filter & Datumsauswahl
- Export

Die genauen Spezifikationen der Tests werden in einem separaten Dokument aufgelistet, welches den Testpersonen überreicht wird. (Usability-Tests_ActivePAL.docx)

9.4.4 Ergebnisse Usability Tests

Die Usability-Tests stellten einen wichtigen Bestandteil der Qualitätssicherung dar und lieferten wertvolle Erkenntnisse zur Verbesserung der Benutzerfreundlichkeit. Insgesamt wurden Tests mit vier Personen durchgeführt, wobei ein Test als moderierter Test mit direkter Begleitung und drei als unmoderierte Tests konzipiert waren.

9.4.4.1 Testpersonen und Methodik

Die Testgruppe setzte sich bewusst heterogen zusammen, um verschiedene Perspektiven zu erfassen:

- **2 Mitarbeitende der MOVE-IT** (Zielgruppe): Personen mit direkter Erfahrung im Umgang mit ActivePAL-Sensoren und medizinischer Datenanalyse
- **1 Betreuer/Dozent** (Prof. Dr. Markus Stolze): Fachliche Perspektive aus dem akademischen Kontext
- **1 externe Person**: Testperson ohne Vorkenntnisse zur MOVE-IT oder dem Projekt, um die allgemeine Verständlichkeit zu prüfen

Bei den moderierten Tests wurde eine Testperson während der Durchführung begleitet, um direktes Feedback zu Gedankengängen und Entscheidungen zu erhalten. Die unmoderierten Tests erfolgten selbstständig anhand eines strukturierten Testleitfadens mit vordefinierten Szenarien (siehe Anhang)

9.4.4.2 Zentrale Erkenntnisse

Die Usability-Tests deckten mehrere wichtige Verbesserungspotenziale auf:

Import und Navigation:

- Die Ordnerauswahl beim Import war nicht eindeutig: Testpersonen wählten zunächst den falschen Ordner (Subject-Ordner statt übergeordneter Ordner)
- Nach Verlassen einer Visualisierung gingen ausgewählte Patienten und Gruppen verloren, was zu wiederholter Konfiguration zwang

Terminologie und Beschriftungen:

- Die Abkürzung "m" für Minuten wurde als "Meter" missverstanden
- "Zeit Laufen" sollte durch "Zeit gehend" ersetzt werden (Hochdeutsch vs. Schweizerdeutsch)
- Kadenz-Beschriftungen waren unklar bezüglich der Bedeutung (Anzahl Schritte vs. Zeit)

Feedback und Transparenz:

- Bei Gruppenvisualisierungen war nicht ersichtlich, welche Tage (alle/Wochentage/Wochenende) für die Aggregation verwendet wurden
- Fehlermeldungen waren nicht immer verständlich bezüglich der Konsequenzen (z.B. fehlende Dateien)

Funktionale Lücken:

- Export als PDF enthielt nur die Grafik, nicht aber die zugehörigen Statistiken
- Standardabweichung und Boxplot-Darstellung waren nicht für alle Visualisierungen verfügbar
- Farbkodierung in der Vergleichsansicht war nicht eindeutig bzgl. Referenzwerten

9.4.4.3 Integration des Feedbacks

Das gesammelte Feedback wurde systematisch in die Entwicklung integriert. Nach jedem Usability-Test wurden die identifizierten Probleme priorisiert und im darauffolgenden Sprint bearbeitet. Dieser iterative Ansatz ermöglichte eine kontinuierliche Verbesserung der Benutzerfreundlichkeit während der Entwicklung.

Umgesetzte Verbesserungen:

- Persistenz der Patienten-/Gruppenauswahl beim Wechsel zwischen Visualisierungen
- Korrektur der Terminologie (min statt m für Minute, gehend statt laufend)
- Ergänzung der Statistiken im PDF-Export
- Klarere Beschriftung der Kadenz-Filter mit Beispielen
- Hinzufügen von Validitätsfiltern zur transparenten Darstellung verwendeter Tage

Nicht umgesetztes Feedback:

- Checkbox bei Legende, denn die Legende der Library ist nicht erweiterbar. Man müsste die Legende komplett mit einer eigenen Komponente ersetzen.
- Wertebeschriftungen im PDF/PNG anzeigen, weil dies kann durch CSV-Export erreicht werden. Zudem wäre es aufwändig und unübersichtlich.

Die Usability-Tests haben sich als äusserst wertvoll erwiesen und trugen massgeblich zur Praxistauglichkeit der Anwendung bei. Das Feedback der MOVE-IT-Mitarbeitenden war besonders wichtig, da diese die primäre Zielgruppe darstellen und konkrete Anforderungen aus dem klinischen Alltag einbringen konnten.

9.4.5 E2E-Tests

Das Ziel der E2E-Tests ist es eine Vollständige Testung des Systems durchzuführen. Diese werden manuell durchgeführt und dokumentiert.

Alle E2E-Tests wurden vom Praxis-Partner (i.V. Caroline Yvonne Tanner) abgenommen.

9.4.5.1 Import**E2E-I-1: Grundlegender CSV-Import (FR-I1, FR-I2)**

- **Ziel:** Gültiges CSV wird fehlerfrei eingelesen und im Chart dargestellt.
- **Schritte:** App starten → Datenordner auswählen → Patient im Schritte/h Chart auswählen
- **Erwartung:** Import startet ohne Fehler; Spalten werden korrekt erkannt; Chart zeigt Zeitreihe «Schritte pro Stunde»; Keine Fehlermeldung erscheint.
- **Kriterium:** Wenn Chart mit sinnvollen Werten erscheint und keine Fehlermeldung angezeigt wird.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 18.10.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-I-2: CSV hat Fehler (NFR-R1)

- **Ziel:** Robuste Fehlerbehandlung bei fehlenden/fehlerhaften Spalten.
- **Schritte:** invalid_header.csv importieren.

- **Erwartung:** Klarer Fehlerhinweis (welche Spalten fehlen/unerkannt); App bleibt bedienbar; erneuter Import möglich.
- **Kriterium:** Wenn verständliche Fehlermeldung erscheint, kein Crash.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-I-3: Fehlende Rechte / gesperrte Datei (NFR-R2)

- **Ziel:** Anwendung reagiert robust auf Zugriffsfehler und zeigt verständliche Fehlermeldungen.
- **Schritte:** Eine CSV-Datei im Patientenordner sperren (durch geöffnetes Programm) → App starten → Import ausführen.
- **Erwartung:** Verständlicher Hinweis («Datei kann nicht gelesen werden / Zugriff verweigert») erscheint; Importvorgang bricht kontrolliert ab; bestehende Daten bleiben unverändert.
- **Kriterium:** Kein Crash oder Datenverlust; klare Fehlermeldung für den Benutzer.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: Die Applikation liest immer den letzten Stand der Datei aus, auch wenn diese gelockt ist. Es funktioniert also auch, wenn jemand die Datei gerade bearbeitet.

9.4.5.2 Patienten und Gruppen

E2E-PG-1: Patientenliste (FR-I3, FR-I5)

- **Ziel:** Patientenübersicht zeigt Pflichtspalten und ermöglicht Suche, Sortierung und Aktualisierung.
- **Schritte:** App starten → Seite «Patienten» öffnen → Sortierung nach Namen → Patient auf Dateiebene hinzufügen → Aktualisieren
- **Erwartung:** Spalten «Name», «Gruppe» vorhanden; Sortierung A→Z korrekt; Neuer Patient nach Aktualisierung in Liste.
- **Kriterium:** Funktionen arbeiten fehlerfrei und ohne UI-Absturz.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-PG-2: Patienten aus-/einblenden (FR-I4)

- **Ziel:** Patienten lassen sich in der Übersicht aus- und einblenden; ausgeblendete Patienten erscheinen weder in Diagrammen noch in Gruppenaggregationen.
- **Schritte:** App starten → Seite «Patienten» öffnen → Patient1 ausblenden → zu Diagramm-Ansicht wechseln → prüfen, dass Patient1 nicht auswählbar ist → zurück zur Übersicht → wieder einblenden.
- **Erwartung:** Ausgeblendeter Patient verschwindet aus Diagramm- und Gruppenansicht; Name wird gespeichert und nach Neustart weiterhin ausgeblendet; Einblenden stellt ursprünglichen Zustand wieder her.
- **Kriterium:** Sichtbarkeit synchron zwischen Patientenliste, Diagramm und Gruppenaggregation; keine Fehlermeldung oder Inkonsistenz.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025

- Status: Bestanden
- Bemerkung: -

E2E-PG-3: Gruppen-CRUD, Mehrfachauswahl (FR-G1, FR-G2, FR-G3)

- **Ziel:** Gruppen anlegen/umbenennen/löschen; Mitglieder per Mehrfachauswahl verwalten
- **Schritte:** Ansicht «Gruppen» → «Neue Gruppe» → Name «Studie A» → Mehrfachauswahl Patienten hinzufügen → Speichern → Umbenennen in «Studie Alpha» → Ein Mitglied entfernen, ein weiteres hinzufügen → Gruppe löschen.
- **Erwartung:** Alle Änderungen sichtbar und nach Neustart vorhanden; Mehrfachauswahl funktioniert; dieselbe Person kann in mehreren Gruppen sein; Löschung entfernt Gruppe dauerhaft.
- **Kriterium:** Kein Fehler; Datenzustand entspricht allen Aktionen vor und nach Neustart.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

9.4.5.3 Visualisierung**E2E-V-1: Datumsspanne (FR-V9)**

- **Ziel:** Änderung der Datumsspanne beeinflusst Chart korrekt.
- **Schritte:** stepping_bouts_medium.csv importieren → Datums-Range umschalten.
- **Erwartung:** Werte/Skalierung passen sich an; Achsenlabels korrekt; keine Artefakte.
- **Kriterium:** Wenn sichtbarer Bereich und Aggregationen konsistent sind.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-V-2: Schritte pro Tag – Chart, Infos & Ansichten (FR-V1, FR-V2, FR-V3)

- **Ziel:** Liniendiagramm, Zusatzkennzahlen und Umschalten auf Boxplot- und Konfidenzintervalldarstellung funktionieren.
- **Schritte:** Ansicht «Schritte pro Stunde» öffnen → prüfen: Chart sichtbar → Infos («Zeit gehend (min)», «Totale Schritte», «Kadenz >=75», «Kadenz >=100») prüfen → auf Boxplot umschalten.
- **Erwartung:** Chart zeigt durchschnittliche Schritte/Tag; alle Infos korrekt sichtbar; Boxplot-Ansicht zeigt präzisere Darstellung (Boxplots).
- **Kriterium:** Keine Fehler; Umschalten verändert Darstellung wie spezifiziert.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 18.10.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-V-3: Tagesverlauf – Metriken & 24h-Timeline (FR-V4)

- **Ziel:** Tagesverlauf visualisiert Metriken über 24h korrekt.
- **Schritte:** Ansicht «Tagesverlauf» öffnen.
- **Erwartung:** Sichtbar: «Inaktive Zeit», «Aufrechte Zeit», «Liegezeit», «Sitzende Zeit», «Übergänge Sitzen → Stehen»; 24h-Timeline wird angezeigt.
- **Kriterium:** Alle Metriken vorhanden; Timeline ohne Darstellungsfehler.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Nicht bestanden
 - Bemerkung: Die Timeline wurde durch Aktivitäten pro Stunde ersetzt

E2E-V-4: Sitzphasen – Kennzahlen (FR-V5)

- **Ziel:** Sitzphasen-Ansicht zeigt geforderte Kennzahlen.
- **Schritte:** Ansicht «Sitzphasen» öffnen.
- **Erwartung:** Sichtbar: «Anzahl Sitzphasen (30min)», «Anzahl Sitzphasen (60min)», «Zeit sitzend 30min», «Zeit sitzend 60min».
- **Kriterium:** Alle vier Kennzahlen sind vorhanden und plausibel befüllt.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-V-5: Tages Zusammenfassung – Vergleich Person/Gruppe (FR-V6)

- **Ziel:** Vergleich zwischen Personen und Gruppen funktioniert.
- **Schritte:** Ansicht «Vergleich täglicher Durchschnittswerte» → Person A + Gruppe X auswählen.
- **Erwartung:** Beide Statistiken erscheinen parallel und vergleichbar.
- **Kriterium:** Vergleichsansicht ohne Fehler; beide Einheiten klar unterscheidbar.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-V-6: Schritte mit Kadenz – Drei Boxplots und Filter(FR-V7)

- **Ziel:** Boxplots für Total, >75 spm, >100 spm.
- **Schritte:** Ansicht «Schritte nach Kadenz pro Tag» öffnen. → Eigenen Kadenzfilter setzen
- **Erwartung:** Drei Boxplots: «Totale Schritte», «Kadenz >75 spm», «Kadenz >100 spm». Eigener Kadenzfilter korrekt dargestellt.
- **Kriterium:** Genau drei Boxplots vorhanden per Standard; Mit Filter, Anzahl Boxplots = Anzahl Filter.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-V-7: Mehrfachanzeige & Farbauswahl (FR-V8)

- **Ziel:** Gleichzeitige Anzeige mehrerer Personen/Gruppen inkl. Farbauswahl
- **Schritte:** In «Schritte pro Stunde» Person A + Gruppe X auswählen → Farben zuweisen
- **Erwartung:** Im Diagramm erscheinen beide Serien mit gewählten Farben
- **Kriterium:** Farben/Legenden korrekt.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-V-8: Diagramm-Auswahlseite – Navigation (FR-V12)

- **Ziel:** Zwischen Diagrammen über Auswahlseite navigieren.
- **Schritte:** Auswahlseite öffnen → nacheinander «Schritte pro Stunde», «Vergleich täglicher Durchschnittswerte», «Anzahl Sitzphasen pro Tag» anwählen.
- **Erwartung:** Jeweils die passende Ansicht wird geladen.
- **Kriterium:** Navigation ohne Fehler/Reload-Schleifen; korrekte Zielseiten.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-V-9: Aktivitäten über den Tag – Stundendarstellung (FR-V13)

- **Ziel:** Visualisierung zeigt Gehen, Stehen, Sitzen, Liegen über 24h mit Minuten pro Stunde.
- **Schritte:** Aktivitätsdaten importieren → Ansicht «Aktivitäten über den Tag» öffnen → zwischen verschiedenen Aktivitäten wechseln.
- **Erwartung:** Chart zeigt 24 Stunden (X-Achse) → Y-Achse zeigt Minuten (0-60) → Wechseln der Aktivitäten aktualisiert Chart korrekt.
- **Kriterium:** Alle vier Aktivitäten sichtbar; Anzeige pro Stunde ≤ 60 Minuten; keine negativen Werte.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-V-10: Validitätsfilter - Standard & Schnellfilter (FR-V14)

- **Ziel:** Standard zeigt nur valide Tage; Schnellfilter für Wochentage/Wochenendtage funktionieren.
- **Schritte:** Datensatz mit validen/invaliden Tagen importieren → Schritte über Tag öffnen → Standard prüfen → Schnellfilter durchschalten.
- **Erwartung:** Standard zeigt nur valide Tage → «Valide Wochentage» zeigt nur valide Mo-Fr → «Valide Wochenendtage» zeigt nur valide Sa-So → «Alle Wochentage» zeigt Mo-Fr (valide+invalid) → «Alle Wochenendtage» zeigt Sa-So (valide+invalid) → Chart aktualisiert sofort.
- **Kriterium:** Standard-Filter aktiv beim Laden; alle Filter funktionieren; Datenfilterung korrekt; aktiver Filter in UI sichtbar.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

9.4.5.4 Persistenz

E2E-P-1: Persistenz Chart-Zustand (FR-V10)

- **Ziel:** Verlassen/Neustart erhält Sicht/Filter/Range.
- **Schritte:** Sicht/Filter/Datum setzen → App schliessen → neu starten.
- **Erwartung:** Zustände werden aus JSON geladen (Range, sichtbare Serien, Aggregation).
- **Kriterium:** Wenn Zustand nach Neustart identisch ist.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-P-2: Persistenz Gruppen (FR-G2)

- **Ziel:** Gruppen werden dauerhaft gespeichert.
- **Schritte:** Gruppen anlegen → App neu starten → Gruppenliste prüfen.
- **Erwartung:** Gruppen vollständig vorhanden, referenzierte Daten vorhanden.
- **Kriterium:** Wenn Gruppen konsistent geladen werden.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

9.4.5.5 Export

E2E-E-1: Chart-Export (PNG/PDF) (FR-V11)

- **Ziel:** Grafikexport mit Zusatzinformationen
- **Schritte:** Ansicht konfigurieren → «PDF/PNG exportieren» → Datei öffnen.
- **Erwartung:** Datei erzeugt; Beschriftungen korrekt; Auflösung ausreichend.
- **Kriterium:** Wenn Export lesbar ist und korrekte Daten enthält. Zusatzinformationen nur bei PDF.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-E-2: Daten-Export (CSV) (FR-V11)

- **Ziel:** Export von gefilterten/aggregierten Daten.
- **Schritte:** Filter/Gruppe setzen → «CSV exportieren» → CSV → Inhalt prüfen.
- **Erwartung:** CSV enthält Spalten + Werte wie im Chart (inkl. Aggregationen). Bei Konfidenzintervall Modus sind zusätzlich obere und untere Grenze des 95% Konfidenzintervalls enthalten.
- **Kriterium:** Wenn Struktur/Benennung korrekt und Werte plausibel sind.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

9.4.5.6 Usability

E2E-U-1: Effizienter Start bis zur ersten Visualisierung (NFR-U1)

- Ziel: Vom Projektordner-Wählen bis zur ersten Visualisierung sind höchstens drei Schritte nötig.
- Schritte: App starten → Projektordner auswählen → automatische Datenerkennung → erste Visualisierung öffnen.
- Erwartung: Erster Chart wird nach maximal drei Aktionen angezeigt.
- Kriterium: ≤3 Benutzeraktionen bis zur Darstellung; keine unnötigen Zwischenschritte.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-U-2: Responsives Verhalten & Fortschrittsanzeige (NFR-U2)

- Ziel: Oberfläche reagiert unmittelbar auf Eingaben mit einer Fortschrittsanzeige.
- Schritte: Import starten → Chart auswählen → während des Ladens UI beobachten → andere Bereiche anklicken.
- Erwartung: Fortschrittsanzeige sichtbar; UI bleibt bedienbar; keine Blockierung oder „Einfrieren“.
- Kriterium: Asynchrone Ausführung und sichtbare Rückmeldung bei längeren Operationen.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-U-3: Farbkontrast & Barrierefreiheit (NFR-U3)

- Ziel: Farben der Visualisierungen erfüllen WCAG 2.1 Kontrastanforderung 1.4.1.
- Schritte: Diagramm mit mehreren Serien öffnen → Farbkontraste prüfen (z. B. per WCAG-Tool).
- Erwartung: Text- und Diagrammelemente heben sich klar vom Hintergrund ab (Kontrast ≥ 4.5:1).
- Kriterium: Alle verwendeten Farben erfüllen WCAG 2.1 Punkt 1.4.1.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

9.4.5.7 Security

E2E-SEC-1: Temporäre Verarbeitung & automatische Löschung (NFR-SEC1)

- **Ziel:** Keine dauerhafte Speicherung personenbezogener Daten; temporäre Dateien werden beim Beenden gelöscht.
- **Schritte:** App starten → CSV mit Patientendaten importieren → App beenden → Projektverzeichnis prüfen.
- **Erwartung:** Nur Gruppendefinitionen und Diagrammzustände bleiben gespeichert; keine Rohdaten oder temporären Patientendateien vorhanden.
- **Kriterium:** Keine personenbezogenen Daten verbleiben nach Beenden der App.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-SEC-2: Datenschutz in Logs (NFR-SEC2)

- **Ziel:** Log-Dateien enthalten keine sensiblen Daten.
- **Schritte:** App ausführen → Fehler provozieren (ungültiges CSV) → Log öffnen.
- **Erwartung:** Keine Klarnamen im Logtext enthalten.
- **Kriterium:** Logs enthalten ausschliesslich technische Informationen, keine vertraulichen Inhalte.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

9.4.5.8 Compatibility

E2E-C-1: Paralleler Betrieb mit klinischer Software (NFR-C1)

- **Ziel:** Anwendung stört keine parallellaufenden Programme.
- **Schritte:** App starten, während Software aktiv ist → CPU/RAM/Port-Nutzung beobachten.
- **Erwartung:** Keine Ressourcenkonflikte oder Abstürze anderer Anwendungen.
- **Kriterium:** App läuft stabil neben anderer Software.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

9.4.5.9 Portability

E2E-PO-1: Lauffähigkeit auf Windows 11 (NFR-PO1)

- **Ziel:** Anwendung startet und funktioniert auf Windows 11 (23H2+).
- **Schritte:** Installation unter Windows 11 → App starten → Basisfunktionen prüfen.
- **Erwartung:** Keine Kompatibilitätsfehler oder Darstellungsprobleme.
- **Kriterium:** App voll funktionsfähig auf Windows 11.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-PO-2: Installierbarkeit als .EXE (NFR-PO2)

- **Ziel:** App lässt sich über .EXE-Installer installieren.
- **Schritte:** Setup ausführen → Installation abschliessen → App starten.
- **Erwartung:** Installation erfolgreich ohne Fehlermeldung.
- **Kriterium:** App startet nach Installation korrekt.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-PO-3: Installation ohne Adminrechte (NFR-PO3)

- **Ziel:** App kann im User-Kontext installiert werden.
- **Schritte:** Installation als normaler Benutzer durchführen → Pfade im Benutzerprofil prüfen.
- **Erwartung:** App funktioniert ohne Administratorrechte; alle Dateien im Benutzerprofil gespeichert.
- **Kriterium:** App lauffähig.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-PO-4: Update ohne Datenverlust (NFR-PO4)

- **Ziel:** Update ersetzt App-Version ohne Verlust der Gruppendaten.
- **Schritte:** Alte Version installieren → Gruppen anlegen → neue Version über alte kopieren → App starten.
- **Erwartung:** Gruppen- und Einstellungsdaten weiterhin vorhanden.
- **Kriterium:** Update-Prozess erfolgreich; Projektordner unverändert.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 28.11.2025
 - Status: Bestanden
 - Bemerkung: Einfach neue Setup.exe laufenlassen und die neue Version überschreibt die alte automatisch.

9.4.5.10 Performance**E2E-P-1: Import-Zeit von Daten (NFR-P1)**

- **Ziel:** Indexierung von 10 Patienten ≤ 3 s, UI responsiv.
- **Schritte:** Projektordner mit 10 Patienten laden → Indexierung starten.
- **Erwartung:** Fertig in ≤ 3 s, Oberfläche bleibt bedienbar.
- **Kriterium:** Wenn Zeit ≤ 3 s und keine UI-Blockade.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-P-2: Analyse 7 Tage (NFR-P2)

- **Ziel:** Analyse von 7 Tagen pro Patienten `stepping_bouts_big.csv` ≤ 5 s auf Standard-Notebook (≥ 8 GB RAM, 2 Cores, SSD).
- **Schritte:** 7-Tage-Datensatz laden $\rightarrow$ Analyse starten.
- **Erwartung:** Ergebnisse in ≤ 5 s im Chart sichtbar.
- **Kriterium:** Wenn Analysezeit ≤ 5 s.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Bestanden
 - Bemerkung: -

E2E-P-3: Startzeit und RAM (NFR-P3)

- **Ziel:** Startzeit ≤ 3 s; RAM-Spitzen bei Gruppenanalyse ≤ 1.5 GB.
- **Schritte:** App frisch starten $\rightarrow$ Zeit messen; Gruppenanalyse durchführen $\rightarrow$ RAM überwachen.
- **Erwartung:** Startzeit ≤ 3 s, RAM-Spitze ≤ 1.5 GB.
- **Kriterium:** Wenn beide Limits eingehalten.
- **Ausführung:**
 - Getestet von: Silvan Bollhalder
 - Datum: 6.12.2025
 - Status: Nicht bestanden
 - Bemerkung: Startzeit nicht konsistent unter 3s. Im Durchschnitt eher 4-5s.

9.5 Code Patterns

In diesem Kapitel sind die im Projekt verwendeten Code Patterns beschrieben und wie sie umgesetzt wurden.

9.5.1 State-Container

Für die Verwaltung des Anwendungszustands wird ein zentraler State-Container eingesetzt. Er dient dazu, benutzerbezogene Einstellungen – insbesondere die Auswahl von Patienten und Gruppen – während der Nutzung der Anwendung konsistent vorzuhalten. Dadurch bleiben zuvor getroffene Selektionen auch dann erhalten, wenn zwischen verschiedenen Ansichten gewechselt wird, was die Bedienung erleichtert.

Als Implementationsgrundlage kommt die State-Management-Bibliothek Zustand zum Einsatz. Sie bietet eine einfache API und verursacht nur geringe Laufzeitkosten, was sie für eine datenlastige Visualisierungsanwendung gut geeignet macht. Die gesamte Zustandslogik ist in der Datei

`/src/renderer/store/useAppStore.ts`

abgelegt. Dort werden die benötigten Teilzustände definiert und über entsprechende Setter-Funktionen aktualisiert.

Der Zugriff erfolgt komponentenübergreifend über den Hook `useAppStore((s) => s.setterFunc)` und `useAppStore((s) => s.zustandsVar)`. Auf diese Weise können sowohl Zustände ausgelesen als auch Änderungen vorgenommen werden, ohne dass komplexere State-Management-Strukturen erforderlich sind.

9.6 Code Metriken

In diesem Kapitel werden die Code Metriken, welche im Projekt verwendet wurden ausgewertet. Dabei sind die vollständigen Analysen im Anhang zu finden.

9.6.1 Lines of Code & Anzahl Files

In der Tabelle 3 - Lines of Code über 200 werden alle Dateien aufgezeigt, die über 200 Codezeilen haben. Grundsätzlich wurde darauf geachtet, dass grosse Files aufgeteilt werden und in kleinere Komponenten aufgeteilt werden. Die grössten Beispiele sind ChartContainer.tsx und DataSelectionList.tsx, welche ohne die Aufteilungen ca. 900 bis 1000 Codezeilen hätten. Die vollständige Tabelle, sowie die Tabelle der Testdateien ist im Anhang zu finden.

Tabelle 3 - Lines of Code über 200

Datei	Leer	Kom- mentar	Code
src/main/import/parser/dailySummariesParser.ts	21	13	346
src/main/ipc/parseHandlers.ts	52	13	328
src/renderer/components/chart/avgcomparison/DailyAverageComparisonTable.tsx	55	15	283
src/renderer/components/chart/chartcontainer/ChartContainer.tsx	27	29	277
src/renderer/pages/Patients.tsx	26	7	273
src/renderer/components/chart/activities/HourlyActivities.logic.ts	34	72	272
src/renderer/components/lists/GenericList.tsx	19	30	271
src/renderer/pages/Import.tsx	35	6	267
src/renderer/components/SnackbarProvider.tsx	23	24	266
src/renderer/components/chart/ChartUtils.ts	46	73	265
src/renderer/pages/Groups.tsx	32	7	264
src/renderer/components/CadenceFilterPanel.tsx	26	18	263
src/renderer/components/selection/parts/DataSelectionPatientRow.tsx	21	31	244
src/renderer/styles/theme.ts	6	1	240
src/main/utils/storeManager.ts	41	68	223
src/renderer/components/chart/activities/HourlyActivities.tsx	12	15	219
src/main/ipc/fileHandlers.ts	32	7	211
src/renderer/components/chart/chartcontainer/parts/ChartContainerExport.tsx	28	19	211
src/renderer/components/selection/useDataSelectionList.ts	26	0	211
Total	1064	1258	9084

Anzahl Files

Total gibt es im Ordner src 77 TypeScript und eine CSS-Dateien. Von diesen 77 Dateien sind 42 React Komponenten (.tsx) und die restlichen 35 TypeScript (.ts), welche für Logik und Hilfsfunktionen gebraucht wurden.

Testdateien

Zusätzlich wurden 15 Dateien (.test.ts) im Ordner test erstellt. Diese enthalten total 3938 Zeilen Code.

9.6.2 Code Komplexität

Strukturelle Komplexität

Die strukturelle Komplexität der Anwendung ergibt sich aus der Architektur von Electron. Die Trennung in Main-Prozess, Renderer-Prozess und IPC-Kommunikation führt zu einer klaren Trennung zwischen systemnaher Logik und Benutzeroberfläche.

Diese Struktur verbessert die Wartbarkeit einzelner Module, erhöht jedoch die Komplexität, da prozessübergreifende Aktionen definiert werden müssen. Insbesondere erschwert die Kommunikation zwischen den Prozessen die Fehlerbehandlung und das Debugging.

Logische Komplexität

Die logische Komplexität der Anwendung entsteht hauptsächlich durch die Verarbeitung und Verwaltung von Anwendungsdaten. Dazu zählen unter anderem CSV-Dateien lesen, validieren und parsen, Zustandsverwaltung und Benutzerinteraktionen.

Der komplexeste Teil des Projektes ist die Verarbeitung der Daten von der CSV-Datei zum entsprechenden Diagramm, sowie die dazu gehörende Datenselektion.

Gesamtbewertung

Insgesamt weist die Anwendung eine mittlere bis erhöhte Komplexität auf. Diese ist primär auf die Anforderungen bzw. Aufgabenstellung und die gewählte Architektur zurückzuführen und nicht auf unnötig komplexe Implementierungen.

Die strukturelle und logische Komplexität ist für den Funktionsumfang der Anwendung angemessen und nachvollziehbar. Durch eine klare Modultrennung und konsistente Architekturentscheidungen bleibt der Code trotz der Komplexität gut wartbar und erweiterbar.

9.6.3 Testabdeckung

Die Testabdeckung betrifft nur die Logik. Die React Komponenten wurden nicht mit Unit Tests abgedeckt. Der Fokus der Tests liegt auf der Datenverarbeitung von CSV-Dateien bis zum Datenset, welches in Chart.js angezeigt werden kann. Dazu gehören die folgenden Aufgaben: Auslesen, Parsen, Gruppieren, Filtern, Formatieren für entsprechendes Diagramm. Zusätzlich wurde das Speichern und Auslesen der Gruppen, Pfade und ausgeblendeten Patienten, sowie alle IPC-Handler getestet.

Für die getesteten Dateien wurde die folgende Coverage erreicht:

- 89.85% Statements 1178/1311
- 75.33% Branches 510/677
- 77.82% Functions 200/257
- 90.74% Lines 1079/1189

Um den vollständigen Bericht anzusehen, kann das «coverage/lcov-report/index.html» geöffnet werden. Dort kann für jede getestete Datei die exakte Coverage analysiert werden. Die Abbildung zeigt die generelle Übersicht der Test Coverage für Statements und Branches:

File ▲		Statements ▾	Branches ▾	
common		100%	1/1	100% 0/0
main/import		94.59%	35/37	85% 17/20
main/import/parser		93.89%	246/262	85.16% 132/155
main/ipc		87.59%	339/387	70.9% 78/110
main/utlis		98.54%	135/137	91.37% 53/58
renderer/api		2.63%	1/38	0% 0/1
renderer/components/chart		95.45%	126/132	82.64% 100/121
renderer/components/chart/activities		88.79%	103/116	55.93% 33/59
renderer/components/chart/avgcomparison		98.18%	54/55	61.7% 29/47
renderer/components/chart/cadence		82.14%	23/28	42.85% 9/21
renderer/components/chart/sedentarybouts		95.55%	43/45	66.66% 28/42
renderer/components/chart/steppingbouts		98.63%	72/73	72.09% 31/43

Abbildung 44 - Test Coverage

9.6.4 Code Duplikation

Die Code Duplikation wurde mit jscpd anaysiert. Durch Erstellen von Komponenten und zentralisierten Logikfunktionen, konnten die duplizierten Codezeilen unter 3% gehalten werden. Diese Metrik betrachtet den gesamten Projektordner. Die meisten Duplikationen sind durch die identische Verwendung einer bereits zentralisierten Funktion oder Komponente entstanden. Der vollständige Report mit den exakten duplizierten Codestellen kann über «jscpd/html/index.html» angeschaut werden.

10 KI-Reflexion

Im Rahmen dieses Projekts wurden KI-gestützte Werkzeuge gezielt zur Unterstützung der Entwicklung eingesetzt. Die primär eingesetzten KI-Tools waren Claude (Anthropic) und ChatGPT für konzeptionelle Fragestellungen, Code-Optimierung und insbesondere die TypeScript-Migration.

10.1 Positive Aspekte

Nach Code Review 1 wurde deutlich, dass eine Migration von JavaScript zu TypeScript notwendig war, um die Wartbarkeit und Code-Qualität zu verbessern. Die KI-Tools erwiesen sich hier als besonders wertvoll:

- **Type Definitions:** Unterstützung bei der Erstellung komplexer TypeScript-Typen, insbesondere Discriminated Unions für Diagramm-Modi und Chart-Datenstrukturen
- **Best Practices:** Vorschläge für Union Types, Generic Types und Interface-Strukturen, die zu typsicherem Code führten
- **Refactoring-Strategien:** Hilfe bei der Umstrukturierung von JavaScript-Code zu typsicherem TypeScript ohne Funktionalitätsverlust
- **Problemlösung:** Unterstützung bei der Fehleranalyse
- **Code-Optimierung:** Identifikation von Performance-Bottlenecks und Refactoring-Möglichkeiten in React-Komponenten
- **Dokumentation:** Hilfe bei der Strukturierung und Formulierung technischer Konzepte

10.2 Kritische Einschränkungen

Trotz der erheblichen Vorteile zeigte sich, dass kritische Überprüfung aller KI-Vorschläge unerlässlich ist:

10.2.1 Kontextverständnis

Die KI-Tools hatten keinen Zugriff auf den vollständigen Projektkontext. Dies führte zu Vorschlägen, die zwar theoretisch korrekt waren, aber nicht zu bereits getroffenen Architekturentscheidungen oder der bestehenden Komponentenstruktur passten.

10.2.2 Code-Qualität:

Vorgeschlagene Lösungen entsprachen manchmal nicht der etablierten Projektarchitektur oder verwendeten suboptimale Patterns. Ohne fundiertes Verständnis der verwendeten Technologien wären solche Fehler möglicherweise übernommen worden.

10.3 Fazit

Der Einsatz von KI-Werkzeugen erwies sich als wertvoller Faktor, insbesondere bei der herausfordernden Migration von JavaScript zu TypeScript. Die TypeScript-Umstellung konnte dank strukturierter KI-Unterstützung deutlich effizienter durchgeführt werden.

Entscheidend für den Erfolg war jedoch die kritische Auseinandersetzung mit allen KI-Vorschlägen. Die KI ersetzte nicht das fachliche Verständnis, sondern ergänzte es. Besonders bei projektspezifischer Fachlogik und Architekturentscheidungen war menschliche Expertise unerlässlich.

Die wichtigste Erkenntnis: KI ist ein kraftvolles Werkzeug zur Effizienzsteigerung bei klar definierten Aufgaben wie der TypeScript-Migration, aber kein Ersatz für fundiertes Fachwissen und kritisches Denken. Die Kombination aus menschlicher Expertise und KI-Unterstützung führte zu den besten Ergebnissen.

11 Literaturverzeichnis

- [1] B. N. Jawad, „Variations in physical activity and sedentary behavior during and after hospitalization in acutely admitted older medical patients: a longitudinal study,“ *BMC Geriatrics*, 2022.
- [2] GeeksforGeeks, „Top 10 libraries for data Visualization,“ GeeksforGeeks, 23 7 2025. [Online]. Available: <https://www.geeksforgeeks.org/data-visualization/libraries-for-data-visualization/>. [Zugriff am 18 12 2025].
- [3] J. Canales Luna, „Die 12 besten Programmiersprachen für Datenwissenschaftler im Jahr 2026,“ datacamp, 15 12 2025. [Online]. Available: <https://www.datacamp.com/de/blog/top-programming-languages-for-data-scientists-in-2022>. [Zugriff am 18 12 2025].
- [4] V. Baranwal, „Best JavaScript Chart Libraries for Data Visualization,“ DigitalOcean, 10 7 2025. [Online]. Available: <https://www.digitalocean.com/community/tutorials/javascript-charts#what-are-the-top-javascript-charting-libraries>. [Zugriff am 18 12 2025].
- [5] „Top 7 JavaScript Charting Libraries,“ [Online]. Available: <https://www.wearedevelopers.com/en/magazine/247/top-javascript-charting-libraries>. [Zugriff am 18 12 2025].
- [6] M. A. Petroff, „arXiv,“ 2024. [Online]. Available: <https://arxiv.org/abs/2107.02270>. [Zugriff am 19 12 2025].
- [7] Airbnb, „Airbnb React/JSX Style Guide,“ [Online]. Available: <https://airbnb.io/javascript/react/>. [Zugriff am 14 10 2025].

12 Verzeichnisse

12.1 Abbildungen

Abbildung 1 - ActivPAL Daten in aktuellen Studien [1]	7
Abbildung 2 - Domänenmodell.....	9
Abbildung 3 - Import Page	11
Abbildung 4 - Patienten Page	12
Abbildung 5 - Gruppen Page	13
Abbildung 6 - Gruppe erstellen	13
Abbildung 7 - Schritte pro Tag Diagramm	14
Abbildung 8 - Schritte pro Tag Diagramm – Boxplotansicht	14
Abbildung 9 - Tagesverlauf Timeline.....	15
Abbildung 10 - Sitzphasen Diagramm.....	15
Abbildung 11 - Tages Zusammenfassung Page	16
Abbildung 12 - Schritte mit Kadenz Diagramm.....	16
Abbildung 13 - Diagramme Auswahlpage	17
Abbildung 14 - Aktivitäten über den Tag	18
Abbildung 15 - Use Case Diagramm.....	19
Abbildung 16 - C4 Level 1.....	27
Abbildung 17 - C4 Level 2.....	27
Abbildung 18 - C4 Level 3.....	28
Abbildung 19 - Datenordner auswählen	33
Abbildung 20 - Patientenansicht	34
Abbildung 21 – Gruppenansicht.....	35
Abbildung 22 - Gruppe erstellen Dialog	35
Abbildung 23 - Visualisierungen Übersicht.....	36
Abbildung 24 - Datenselektion	37
Abbildung 25 - Chart-Container Schritte pro Stunde	38
Abbildung 26 – Zusatzinformationen.....	39
Abbildung 27 - Schritte pro Stunde - Boxplot	40
Abbildung 28 - Schritte pro Stunde - Konfidenzintervall	40
Abbildung 29 - Liegen über den Tag	41
Abbildung 30 - Sitzphasen pro Tag	42
Abbildung 31 - Schritte mit Kadenz	42
Abbildung 32 – Tagesdurchschnittsvergleich	43
Abbildung 33 - PDF Export	44
Abbildung 34 - PNG Export.....	45
Abbildung 35 - CSV Export	45
Abbildung 36 – Mitteilungen.....	46
Abbildung 37 - Mindestbreite Layout.....	47
Abbildung 38 - Projektplan	51
Abbildung 39 - Zeitplan-IST	52
Abbildung 40 - Zeitplan Vergleich IST/SOLL.....	52
Abbildung 41 - Zeiten pro Person pro Woche	54

Abbildung 42 - Verlauf der Risiken.....	55
Abbildung 43 - Git Pipeline	60
Abbildung 44 - Test Coverage	75

12.2 Tabellen

Tabelle 1 - 3rd Party Libraries.....	31
Tabelle 2 - Stunden pro Woche	54
Tabelle 3 - Lines of Code über 200	73

13 Anhang

13.1 Aufgabenstellung

Aufgabenstellung Semesterarbeit HS25/26 Informatik OST «Visualisierung von 24-Stunden Aktivitätsprotokollen basierend auf Daten des ActivPAL-Aktivitätssensors»

1. Betreuer

Prof. Dr. Markus Stolze, OST, I3, markus.stolze@ost.ch

2. Praxispartner

Dr. Martina Betschart, Institut für Gesundheitswissenschaften, MOVE IT

3. Studierende

Silvan Bollhalder

Quentin Weber

4. Ausgangslage

In der klinischen Praxis und der gesundheitsbezogenen Forschung wird die objektive Erfassung von Bewegungs- und Schlafverhalten immer wichtiger. Die Aktigraphie – ein nicht-invasives Verfahren – ermöglicht es, diese Verhaltensmuster über mehrere Tage oder Wochen im natürlichen Umfeld der Patient:innen kontinuierlich aufzuzeichnen. Die daraus resultierenden 24-StundenBewegungsmuster liefern wertvolle Hinweise auf den physischen und psychischen Gesundheitszustand.

Im Frühlingssemester 2025 (FS25) wurde im Rahmen einer Studienarbeit ein System zur Visualisierung von 24h-Aktivitätsprotokollen entwickelt. Es basiert auf Daten des GENEActivSensors und ermöglicht eine schnelle, benutzerfreundliche Darstellung individueller Aktivitätsmuster.

5. Aufgaben (zu validieren)

Ziel der aktuellen Arbeit ist es, dieses bestehende Visualisierungssystem so zu erweitern, dass auch Daten des ActivPAL-Sensors (<https://www.palt.com/>) verarbeitet und visualisiert werden können.

Die Daten werden derzeit mit der ActivPAL-Software extrahiert (<https://www.palt.com/software-suite/>), genügen jedoch in ihrer Standarddarstellung nicht den Anforderungen der klinischen Praxis. Es sollen deshalb folgende Erweiterungen umgesetzt werden:

Stundenbasierte Darstellung:

Die Aktivitätsdaten sollen nicht nur als Rohdaten mit Standardzeitwerten (z. B. in Spalte A) dargestellt werden, sondern in einer klar stundenbezogenen Darstellung, die ein 24-Stunden-Profil abbildet (siehe Abbildung 1).

Individuelles Durchschnittsprofil:

Für einzelne Patient:innen soll ein durchschnittliches 24-Stunden-Aktivitätsprofil berechnet und visualisiert werden, z. B. in Bezug auf die aufrechte Zeit („upright time“). Siehe Beispiel in Jawad et al. (2022) (Abbildung 2).

Kohortenweites Durchschnittsprofil:

Zusätzlich soll das System ein gemitteltes 24-Stunden-Aktivitätsprofil für eine ganze Patient:innenkohorte erstellen und visualisieren.

Die Visualisierung kann sich gestalterisch und technisch an den bereits umgesetzten Darstellungen für GENEActiv-Daten orientieren. Eine Integration in das bestehende System aus dem FS25 ist ausdrücklich erwünscht.

6. Rahmenbedingungen

Die von den Studierenden erstellte Software wird von den Studierenden mit einer Open-Source Lizenz versehen (MIT). Der Zeitpunkt und die Modalitäten der Verfügbarmachung des Codes wird zwischen den Studierenden und dem Betreuer abgesprochen. OST und die Studierenden können den Code auch vor der Veröffentlichung durch die Studierenden in Projekten einsetzen.

7. Schlussbestimmungen

Im Weiteren gelten die Bedingungen zu den Informatik Bachelor- und Studienarbeiten entsprechend dem aktuellen «Leitfaden für Bachelor- und Studienarbeiten» des Bachelorstudiengangs Informatik an der OST. Das Kriterien-Raster zur Bewertung wird den Studierenden zusammen mit dieser Aufgabenstellung verteilt. Allfällige Anpassungen müssen bis Mitte der Arbeit einvernehmlich beschlossen werden.

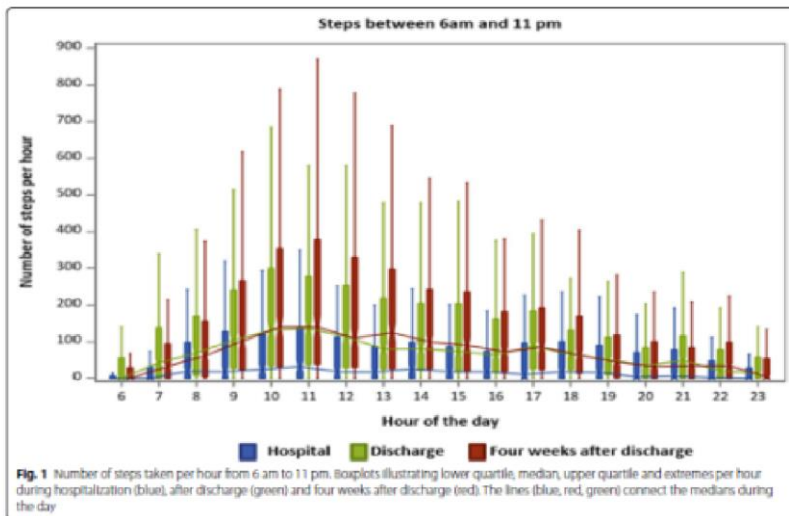
Prof. Dr. Markus Stolze
(Dokument ohne Unterschrift)

Anhang

Abbildung 1. Auszug aus einem .csv file output «epochs»; Werte von einem Patienten

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
4	validation_alg	MORA													
5	validation_alg	v1.0													
6	validation_alg		24												
7	analysis_algor	CREA													
8	analysis_algor	v1.3													
9	analysis_algor	yes													
10	analysis_algor	10													
11	analysis_algor	10													
12															
13	**data**														
14	Time	Time(approx)	StepCount	Activity Score	Sedentary Time	Upright Time	Stepping Time	Cycling Time	Primary Lying	Secondary Lying	Nonwear Time	Seated Time	Trans Data	Errors (s)	Sedentary to Upright to
15	45400.4791	#####	0	1.25	1	0	0	0	0	0	0	0	0	0	0
16	45400.4791	#####	0	1.25	1	0	0	0	0	0	0	0	0	0	0
17	45400.4791	#####	0	1.25	1	0	0	0	0	0	0	0	0	0	0
18	45400.4791	#####	0	1.25	1	0	0	0	0	0	0	0	0	0	0
19	45400.4791	#####	0	1.25	1	0	0	0	0	0	0	0	0	0	0
20	45400.4791	#####	0	1.25	1	0	0	0	0	0	0	0	0	0	0
21	45400.4791	#####	0	1.25	1	0	0	0	0	0	0	0	0	0	0
22	45400.4792	#####	0	1.25	1	0	0	0	0	0	0	0	0	0	0
23	45400.4792	#####	0	1.25	1	0	0	0	0	0	0	0	0	0	0
24	45400.4792	#####	0	1.25	1	0	0	0	0	0	0	0	0	0	0
25	45400.4792	#####	0	1.25	1	0	0	0	0	0	0	0	0	0	0
26	45400.4792	#####	0	1.25	1	0	0	0	0	0	0	0	0	0	0
27	45400.4792	#####	0	1.25	1	0	0	0	0	0	0	0	0	0	0
28	45400.4792	#####	0	1.25	1	0	0	0	0	0	0	0	0	0	0
29	45400.4792	#####	0	1.25	1	0	0	0	0	0	0	0	0	0	0
30	45400.4792	#####	0	1.25	1	0	0	0	0	0	0	0	0	0	0
31	45400.4793	#####	0	1.25	1	0	0	0	0	0	0	0	0	0	0
32	45400.4793	#####	0	1.25	1	0	0	0	0	0	0	0	0	0	0

Abbildung 2. aus *Jawad et al. BMC Geriatrics (2022) 22:209*; <https://doi.org/10.1186/s12877-022-02917-8>



13.2 User Guide

Installation

Die Installation ist für jedes Betriebssystem unterschiedlich.

Windows: Das in der Abgabe bereitgestellte Setup.exe Datei ausführen und die Applikation wird auf dem Rechner installiert.

Linux*: Mit der .deb Datei aus der Abgabe, kann die Applikation auf Linux installiert werden.

Mac*: Bei verantwortlicher Person des IFS nachfragen, ob eine Mac-Installation bereitgestellt werden kann.

*Die Applikation wurde für Windows entwickelt. Sie sollte auch auf Mac und Linux funktionieren, dies wurde aber nicht ausführlich getestet.

Datenauswahl

Die Applikation braucht gewisse CSV-Dateien, um funktionieren zu können. Wichtig ist, dass die Endung der Dateien mit diesen übereinstimmt:

- ...DailySummaries.csv
- ...SteppingBouts.csv
- ...EventsEx.csv

Damit Daten visualisiert werden können, muss der Datenordner ausgewählt werden. Dabei ist wichtig, **dass der Überordner der Patienten-Ordner ausgewählt wird** und die CSV-Dateien müssen direkt in einem Patienten-Ordner liegen.

Beispiel:

Es gibt einen Ordner «Patient1». In diesem Ordner befinden sich die CSV-Dateien

...DailySummaries.csv, ...SteppingBouts.csv, ...EventsEx.csv und noch einige weitere Dateien. Der Ordner «Patient1» befindet sich neben anderen Patienten-Ordnern im Ordner «Patienten-ActivPAL».

Somit muss der Ordner «Patienten-ActivPAL» bei Ordner Auswahl gewählt werden, um die Daten von «Patient1» und den anderen Patienten-Ordnern visualisieren zu können.

Wenn ein Pfad ausgewählt ist, werden unter «Direkte Unterordner» alle gefundenen Patientenordner aufgelistet. Zusätzlich wird angegeben, ob passende CSV-Dateien gefunden wurden oder welche fehlen.

Hat ein Ordner keine CSV-Dateien wird er automatisch ausgeblendet.

Jeder ausgewählte Ordner speichert seine Gruppen und Patienteninformationen für sich. Wenn zwischen verschiedenen Ordnern gewechselt wird, sind immer nur die Gruppen zu sehen, welche in diesem Ordner erstellt wurden. Diese bleiben auch durch das Wechseln zwischen Ordner bestehen.

Patienten

Auf der Patienten-Ansicht können die Patienten-Ordner über die 3 Punkte am Ende der Zeile umbenannt und gelöscht werden. Durch das Löschen / Umbenennen, wird der Ordner auf dem Dateisystem vollständig gelöscht / umbenannt. Zusätzlich können die Patienten aus eingeblendet werden, wodurch sie in Visualisierungen und Gruppen sichtbar werden. Ausgeblendete Patienten, die in Gruppen werden in den Visualisierungen nicht berücksichtigt.

Oben können über den Button «Aktualisieren» die Patienten neu ausgelesen werden. Zusätzlich werden die gelesenen Dateien aus dem Cache gelöscht.

Name ↑	Gruppen	Anzeigen
Exports		☐ ⋮
Patient1	Patienten	☒ ⋮
Patient2	Patienten	☒ ⋮
Patient3	Patienten	☒ ⋮

und
sind,
dadurch

Gruppen

In der Gruppen-Ansicht können Gruppen für die zurzeit ausgewählten Daten erstellt werden. Oben können Gruppen erstellt als auch exportiert und importiert werden. Beim Exportieren werden alle Gruppen in eine Datei geschrieben. Wird diese Datei dann importiert werden die Gruppen automatisch erstellt. Jedoch ist wichtig, dass die Patientennamen übereinstimmen, ansonsten werden sie nicht in die Gruppe hinzugefügt. Die Gruppe wird nicht erstellt, falls kein übereinstimmender Patient gefunden wird.

Gruppen können über die 3 Punkte am Ende der Tabelle gelöscht oder bearbeitet werden. Beim Erstellen und Bearbeiten erscheint der Dialog. Hier muss immer ein Name gesetzt werden und es können die angezeigten Patienten an und abgewählt werden, um sie der Gruppe hinzuzufügen. Zusätzlich können alle Patienten über das Icon rechts über der Auswahl an- / abgewählt werden.

Neue Gruppe erstellen

Gruppenname

Weiblich

Patienten auswählen (0/3)

Suchen...

☐ Patient1
☐ Patient2
☐ Patient3

Abbrechen

Erstellen

Visualisierungen

Unter Visualisierungen sind die verschiedenen Diagramme aufgelistet, welche dargestellt werden können. Jede Visualisierung hat eine Datenselektion:

Datenselektion



The interface shows two columns of dropdown menus. The first column has 'Patient' (selected 'Patient1') and 'Gruppe' (selected 'Patienten'). The second column has 'Daten / Filter' (selected 'Valide Tage') and 'Anzuzeigende Tage' (selected 'Valide Tage'). To the right of these are two colored circles (blue and red) with trash icons. Below the dropdowns are three buttons: 'Patient hinzufügen', 'Gruppe hinzufügen', and 'Gruppe erstellen'.

Hier kann gewählt werden, welche Patienten / Gruppen im Diagramm dargestellt werden. Dabei können alle Gruppen und die eingblendeten Patienten ausgewählt werden. Über die Farbe kann die Farbe verändert werden und der Eimer entfernt die entsprechende Selektion. Es können Patienten und Gruppen-selektionen hinzugefügt werden und es können auch direkt Gruppen erstellt werden.

Zusätzlich können Patienten und Gruppen mit einem Klick auf die Legende für das aktuelle Diagramm aus und wieder eingblendet werden:

☒ Patient1 ☐ (95%-Konfidenzintervall) ☐ Gruppe: Patienten ☐ (95%-Konfidenzintervall)

In den Diagrammdarstellungen können folgende Aktionen durchgeführt werden.

Zoom:

Drag-Zoom:  Bereich markieren und dann wird dieser Bereich auf dem Chart dargestellt.

Beispiele: Nur Daten von X-Y Uhr anzeigen, Y-Skalierung von 0 -60min auf 0 - 10min ändern (Aktivitäten pro Stunde)

Pan:  Diagramm verschieben, um andere Bereiche zu sehen.

Scroll:  Durch Scrollen, kann in das Diagramm oder aus dem Diagramm herausgezoomt werden (schwer zu bedienen)

Zoom zurücksetzen 

Export: PNG exportiert die aktuelle Darstellung als .png Datei. PDF erstellt ein PDF mit dem Bild sowie den Zusatzdaten, die sich unterhalb des Diagramms befinden. CSV erstellt eine .csv Datei, die in Excel geöffnet werden kann. Dabei werden die ausgewählten Patienten und Gruppen in Form (Schnitt, Boxplot, Konfidenzintervall) der aktuellen Darstellung exportiert.

Vollbild:  Über diesen Button, kann die Darstellung über den gesamten Bildschirm geöffnet werden.

Aktivitäten pro Stunde

Verwendete Dateien: EventsEx und DailySummaries



Hier können die Aktivitäten Gehen, Stehen, Sitzen, Liegen, Fahrrad, Reisen und Sensor nicht getragen dargestellt werden. Daten werden pro Stunde berechnet. Es wird also beispielsweise bei 12:00 angezeigt, wie viele Minuten der Patient / die Gruppe die gewählte Aktivität durchschnittlich von 12:00-12:59 ausübt. Die Daten können als Durchschnitt mit oder ohne Konfidenzintervall (95%) oder als Boxplot dargestellt werden.

Schritte pro Stunde

Verwendete Dateien: SteppingBouts und DailySummaries

Kadenzfilter: $\geq$ Schritte pro Minute

Hier können die Schritte pro Stunde ausgewertet werden. Zusätzlich gibt es einen Kadenzfilter, wodurch nur die Schritte mit passender Kadenz (Schritte pro Minute) in die Berechnung einfließen. Auch hier wird für 12:00 alle Schritte gezählt, die zwischen 12:00 und 12:59 gemacht wurden. Die Daten können als Durchschnitt mit oder ohne Konfidenzintervall (95%) oder als Boxplot dargestellt werden.

Schritte nach Kadenz pro Tag

Verwendete Dateien: SteppingBouts und DailySummaries

Kadenzfilter

$\geq$ Schritte pro Minute $\leq$ Schritte pro Minute Bereich

Von (SPM) - Bis (SPM)

Die Schrittdaten werden pro Tag in einem Boxplot dargestellt. Dabei können hier auch mehrere Kadenzfilter verwendet werden.

Anzahl Sitzphasen pro Tag

Verwendete Dateien: DailySummaries

Zeigt die Anzahl Sitzphasen, die mit einer mindestdauer von 30 bzw 60 Minuten pro Tag durchgeführt wurden. Die Darstellung ist ein Boxplot.

Vergleich täglicher Durchschnittswerte

Verwendete Dateien: DailySummaries

Hier werden die Tagesdurchschnitte der Patienten und Gruppen miteinander verglichen und in Textform dargestellt. Dabei ist immer die erste Selektion der Referenzwert.

13.3 Code Metriken

13.3.1 Lines of Code (src)

language	filename	blank	comment	code
TypeScript	src/main/import/parser/daily-SummariesParser.ts	21	13	346
TypeScript	src/main/ipc/parseHandlers.ts	52	13	328
TypeScript	src/renderer/components/chart/avgcomparison/DailyAverageComparisonTable.tsx	55	15	283
TypeScript	src/renderer/components/chart/chartcontainer/ChartContainer.tsx	27	29	277
TypeScript	src/renderer/pages/Patients.tsx	26	7	273
TypeScript	src/renderer/components/chart/activities/HourlyActivities.logic.ts	34	72	272
TypeScript	src/renderer/components/lists/GenericList.tsx	19	30	271
TypeScript	src/renderer/pages/Import.tsx	35	6	267
TypeScript	src/renderer/components/SnackbarProvider.tsx	23	24	266
TypeScript	src/renderer/components/chart/ChartUtils.ts	46	73	265
TypeScript	src/renderer/pages/Groups.tsx	32	7	264
TypeScript	src/renderer/components/CandenceFilterPanel.tsx	26	18	263
TypeScript	src/renderer/components/selection/parts/DataSelection-PatientRow.tsx	21	31	244
TypeScript	src/renderer/styles/theme.ts	6	1	240
TypeScript	src/main/utils/storeManager.ts	41	68	223
TypeScript	src/renderer/components/chart/activities/HourlyActivities.tsx	12	15	219
TypeScript	src/main/ipc/fileHandlers.ts	32	7	211
TypeScript	src/renderer/components/chart/chartcontainer/parts/ChartContainerExport.tsx	28	19	211
TypeScript	src/renderer/components/selection/useDataSelection-List.ts	26	0	211

TypeScript	src/renderer/components/chart/steppingbouts/SteppingBouts.logic.ts	23	42	195
TypeScript	src/renderer/api/electronAPI.tsx	12	25	181
TypeScript	src/renderer/layouts/AppLayout.tsx	30	14	173
TypeScript	src/renderer/components/GroupDialog.tsx	16	16	163
TypeScript	src/main/import/parser/steppingBoutsParser.ts	39	18	151
TypeScript	src/renderer/components/chart/avgcomparison/DailyAverageComparison.logic.ts	23	32	151
TypeScript	src/renderer/components/selection/DataSelectionList.tsx	12	19	142
TypeScript	src/renderer/components/chart/steppingbouts/SteppingBouts.tsx	13	14	141
TypeScript	src/main/import/parser/eventsParser.ts	27	8	138
TypeScript	src/renderer/components/chart/sedentarybouts/SedentaryBouts.logic.ts	16	35	123
TypeScript	src/renderer/components/selection/Utils.ts	13	45	112
TypeScript	src/renderer/components/selection/parts/DataSelectionGroupRow.tsx	9	20	109
TypeScript	src/renderer/components/selection/parts/ColorSelection.tsx	10	20	105
TypeScript	src/renderer/components/chart/cadence/SteppingCadence.logic.ts	12	38	101
TypeScript	src/main/ipc/groupHandlers.ts	9	3	99
TypeScript	src/renderer/components/chart/cadence/SteppingCadence.tsx	12	16	99
TypeScript	src/renderer/components/chart/Charts.tsx	2	1	96
TypeScript	src/renderer/hooks/useSelectionLoader.ts	11	19	87
TypeScript	src/renderer/components/chart/avgcomparison/DailyAverageComparison.tsx	8	14	81

TypeScript	src/main/import/csvReader.ts	11	4	77
TypeScript	src/renderer/components/chart/sedentarybouts/SedentaryBouts.tsx	6	15	77
TypeScript	src/common/types.ts	15	26	76
TypeScript	src/renderer/components/chart/chartcontainer/parts/ZoomControls.tsx	6	19	76
TypeScript	src/renderer/components/lists/AdditionalDataCard.tsx	5	16	76
TypeScript	src/renderer/components/selection/parts/PatientSelection.tsx	6	21	74
TypeScript	src/renderer/components/navigation/NavItem.tsx	6	17	71
TypeScript	src/renderer/store/useAppStore.ts	6	12	68
TypeScript	src/renderer/components/InfoBox.tsx	5	12	64
TypeScript	src/renderer/pages/Charts.tsx	7	11	64
TypeScript	src/renderer/components/chart/chartcontainer/hooks/useChartFull-screen.ts	12	20	62
TypeScript	src/renderer/components/chart/chartcontainer/hooks/useChartCursor.ts	11	13	60
TypeScript	src/renderer/layouts/MenuList.tsx	3	12	58
TypeScript	src/renderer/components/chart/chartcontainer/parts/ChartHeader.tsx	9	17	57
TypeScript	src/renderer/layouts/ChartDetailLayout.tsx	6	13	57
TypeScript	src/main/main.ts	11	2	55
TypeScript	src/renderer/layouts/HeaderLayout.tsx	3	11	52
TypeScript	src/renderer/layouts/PageTitle.tsx	4	11	48
TypeScript	src/renderer/components/chart/chartcontainer/parts/ChartContent.tsx	3	23	47
TypeScript	src/renderer/components/chart/chartcontainer/parts/ModeToggle.tsx	5	25	44
TypeScript	src/renderer/components/FolderPicker.tsx	5	13	41
TypeScript	src/renderer/renderer.tsx	3	10	38

TypeScript	src/renderer/components/StyledPaper.tsx	4	8	32
TypeScript	src/renderer/components/lists/AdditionalDataList.tsx	4	15	30
TypeScript	src/renderer/components/chart/chartcontainer/parts/FullscreenButton.tsx	2	14	29
TypeScript	src/renderer/components/selection/SelectionTypes.ts	5	0	29
TypeScript	src/renderer/components/chart/chartcontainer/hooks/useChartZoomPanDrag.ts	4	13	27
CSS	src/renderer/styles/index.css	5	0	27
TypeScript	src/renderer/components/chart/chartcontainer/parts/ExportControls.tsx	5	15	26
TypeScript	src/renderer/hooks/useEntityChartLoader.ts	3	4	26
TypeScript	src/renderer/hooks/usePatientsAndGroups.ts	4	4	25
TypeScript	src/renderer/layouts/VerticalFlowLayout.tsx	3	9	24
TypeScript	src/main/preload.ts	7	0	23
TypeScript	src/main/ipc/index.ts	3	0	14
TypeScript	src/renderer/styles/chartColors.ts	0	1	13
TypeScript	src/main/import/dataGroup.ts	2	5	12
TypeScript	src/main/utils/schema.ts	3	0	12
TypeScript	src/common/constant.ts	1	0	6
TypeScript	src/main/utils/logger.ts	2	0	4
TypeScript	src/renderer/components/chart/ChartConstants.ts	0	0	2
SUM		1064	1258	9084

13.3.2 Lines of Code (test)

language	filename	blank	comment	code
Type-Script	test/main/import/eventsParser.test.ts	106	0	486
Type-Script	test/steppingBouts.logic.test.ts	39	3	369
Type-Script	test/main/utils/storeManager.test.ts	101	4	367
Type-Script	test/hourlyActivities.logic.test.ts	71	0	364
Type-Script	test/main/import/dailySummaries-Parser.test.ts	45	0	336
Type-Script	test/main/ipc/parseHandlers.test.ts	58	8	327
Type-Script	test/main/ipc/fileHandlers.test.ts	60	0	323
Type-Script	test/chartUtils.test.ts	45	12	280
Type-Script	test/main/ipc/groupHandlers.test.ts	73	0	242
Type-Script	test/sedentaryBouts.logic.test.ts	22	0	237
Type-Script	test/dailyAverageComparison.logic.test.ts	28	0	199
Type-Script	test/main/import/steppingBoutsParser.test.ts	39	3	157
Type-Script	test/main/import/csvReader.test.ts	26	0	108
Type-Script	test/steppingCadence.logic.test.ts	34	4	99
Type-Script	test/main/import/dataGrouper.test.ts	8	0	44
SUM		755	34	3938

13.3.3 Coverage

All files

89.85% Statements 1178/1311 75.33% Branches 510/677 77.82% Functions 200/257 90.74% Lines 1079/1189

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

File		Statements	Branches	Functions	Lines	
common		100%	1/1	100%	0/0	100%
main/import		94.59%	35/37	85%	17/20	100%
main/import/parser		93.89%	246/262	85.16%	132/155	100%
main/lpc		87.59%	339/387	70.9%	78/110	87.32%
main/utlis		98.54%	135/137	91.37%	53/58	100%
renderer/apl		2.63%	1/38	0%	0/1	0%
renderer/components/chart		95.45%	126/132	82.64%	100/121	100%
renderer/components/chart/activities		88.79%	103/116	55.93%	33/59	89.28%
renderer/components/chart/avgcomparison		98.18%	54/55	61.7%	29/47	66.66%
renderer/components/chart/cadence		82.14%	23/28	42.85%	9/21	62.5%
renderer/components/chart/sedentarybouts		95.55%	43/45	66.66%	28/42	85.71%
renderer/components/chart/steppingbouts		98.63%	72/73	72.09%	31/43	91.3%

13.3.4 Duplication

Format	Files analyzed	Total lines	Total tokens	Clones found	Duplicated lines	Duplicated tokens
javascript	39	2012	14983	0	0 (0%)	0 (0%)
tsx	42	6366	46721	7	102 (1.6%)	875 (1.87%)
typescript	37	4975	40583	24	279 (5.61%)	2921 (7.2%)
Total:	118	13353	102287	31	381 (2.85%)	3796 (3.71%)

13.4 Usability Tests

Ziel

Dieser Test soll manuell von Mitarbeiter der MOVE-IT ausgefüllt werden und dient der Überprüfung der Benutzerfreundlichkeit und Funktionalität der Software ActivPAL Visualisierungen. Der Fokus liegt dabei auf Datenimport, Gruppen- und Patientenverwaltung sowie Darstellung 3 verschiedener Diagramme (Sitzphasen-, Kadenz-, Schritte-pro-Tag-Diagramme) und der Vergleichsansicht für Tageszusammenfassungen.

Zielgruppe

Die Tests werden manuell von Mitarbeitenden der MOVE-IT durchgeführt, welche mit dem ActivPAL Sensor arbeiten. Ziel ist es, Beobachtungen zum Verhalten der Anwendung, zur Verständlichkeit der Benutzeroberfläche und zur Effizienz der Abläufe zu sammeln. Diese Rückmeldungen dienen dazu, mögliche Schwachstellen in der Interaktion zu identifizieren und gezielt Verbesserungen für eine spätere Version der Software abzuleiten.

Voraussetzung

- Windows 11 23H2+
- .CSV-Dateien aus dem erhaltenen ZIP in einem lokalen Verzeichnis abgelegt.
- Evtl. eigene CSV-Dateien aus dem ActivPAL Sensor ebenfalls in ein lokales Verzeichnis abgelegt, um eigene Tests durchzuführen.

Szenarien

Spiele diese Szenarien mit der Applikation durch und dokumentiere deine Beobachtungen jeweils in der zugehörigen Tabelle. Schreibe vor allem Dinge auf, welche unklar, unübersichtlich oder nicht intuitiv waren.

Szenario 0 – Installation

1. Entpacke das ZIP und installiere die Applikation.
2. Starte die Applikation vom Desktop aus.

Schritt	Beobachtung
1) Installation	
2) Starten	

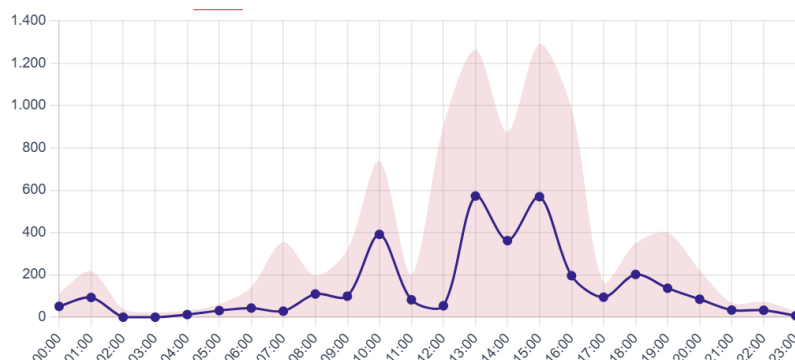
Szenario 1 - Import, Patienten und Sitzphasen Diagramm:

1. Öffne die Anwendung und wähle den entsprechenden Ordnerpfad mit den Patientendaten aus.
2. Du möchtest die Sitzphasen des Patienten S24 anzeigen und überprüfen, wie viele Sitzphasen der Patient an Wochentagen hat (>60m).
3. Vergleiche nun S24 mit dem Patienten «non numerical val in dailySum»
4. Du merkst, dass ein Patient Fehler hat. Überprüfe was fehlgeschlagen ist und blende den Patienten aus, damit der Patient nicht mehr in der Patientenauswahl zu sehen ist.

Schritt	Beobachtung
1) Ordnerpfad auswählen	
2) Sitzphasen S24	
3) Vergleich mit fehlerhaftem Patient	
4) Patient ausblenden	

Szenario 2 – Gruppen und Schritte pro Tag Diagramm:

1. Du möchtest eine Gruppe «Testgruppe A» erstellen. Diese Gruppe soll die Patienten S24 und S36 beinhalten
2. Benenne die Gruppe um in «Testgruppe Valide» und füge S40 hinzu.
3. Vergleiche die Testgruppe mit dem Patienten S24. Du möchtest dir ansehen, wie viele Schritte diese über den Tag verteilt haben.
4. Du möchtest die Standardabweichung der Gruppe «Testgruppe Valide» mit dem Schnitt des Patienten S24 vergleichen, damit es so aussieht wie im Bild unten.



5. Du möchtest nun die Boxplots anzeigen lassen, aber nur für die Stunden von 09:00 -23:00.
6. Erstelle nun eine neue Gruppe «Testgruppe Alle» direkt in der Visualisierungsansicht, welche alle verfügbaren Mitglieder beinhaltet.
7. Lösche die zuvor erstellte Gruppe und überprüfe, ob sie vollständig aus der Gruppenauswahl verschwindet.

Schritt	Beobachtung
1) «Testgruppe A» erstellen	
2) Gruppe umbenennen und S40 hinzufügen	
3) Vergleich mit S24	
4) Vergleich mit Standardabweichung	

5) Boxplots anzeigen von 9-23 Uhr	
6) Neue Gruppe «Testgruppe Alle» direkt erstellen	
7) Gruppe wieder löschen	

Szenario 3 – Kadenz Diagramm:

1. Vergleiche die Schritte pro Tag mit Kadenz von Patient S24 mit S36.
2. Du möchtest überprüfen welcher Patient mehr Schritte pro Tag mit Kadenz <60 getätigt hat und wer mehr Schritte mit einer Kadenz zwischen 50 und 100 hat.
3. Du möchtest die Farbe von Patient S24 ändern.

Schritt	Beobachtung
1) Kadenz von S24 und S36 vergleichen	
2) Vergleich von Kadenz <60 und 50 bis 100	
3) Farbe von S24 ändern	

Szenario 4 – Tageszusammenfassung Vergleich:

1. Vergleiche Gruppe «Testgruppe Valide» mit Patient S24 und S40. Du möchtest wissen, wer im Vergleich zur Gruppe mehr Zeit aufrecht verbracht und wer mehr gelaufen ist.
2. Vergleiche eine Gruppe mit S24 und S36 mit «Testgruppe Valide» Du möchtest wissen, wie viel mehr Zeit «Testgruppe Valide» mit Fahrradfahren verbringt. Der Prozentwert des Vergleichs muss bei «Testgruppe Valide» angezeigt werden.

Schritt	Beobachtung
1) Vergleich S24 und S40	
2) Vergleich Gruppe mit S24 und S36 mit Testgruppe Valide	

Szenario 5 – Eigene Dateien / sonstiges Feedback

Falls du eigene Dateien des ActivePAL Sensors hast, kannst du diese ebenfalls ausprobieren und deine Beobachtungen hier dokumentieren. Auch kannst du hier allgemeines Feedback festhalten.

Feedback	Beschreibung
1)	
2)	
3)	
4)	
5)	