

Fakultät 1

Fachgebiet Praktische
Informatik/Softwaresystemtechnik
Prof. Dr. rer. nat. Leen Lambers



Brandenburgische
Technische Universität
Cottbus - Senftenberg

Modul 12202 - Softwarepraktikum

Wintersemester 2025/2026

Gruppe A

Produktdokumentation

vorgelegt von:

Autor	Matrikelnummer
Marwa Al Siyamji Al Mousli	5003322
Valentina Aseeva	5006672
Bruno Schreier	5007043
Eric Staudt	5012775
Hai Hoang Du	5013508

Cottbus, den 10. Dezember 2025

Tutor: Leonie Ziechmann

Inhaltsverzeichnis

1	Anforderungsspezifikation und Beschreibung der Systemtests	4
1.1	Glossar	4
1.1.1	Fachbegriffe	4
1.1.2	Prozessbegriffe	5
1.2	Domänenstruktur	7
1.3	Geschäftsprozess	8
1.4	Systemfunktionen	9
1.4.1	Übersicht	9
1.4.2	Beschreibung pro Use Case	10
1.5	Beschreibung Systemtests	13
2	Architekturbeschreibung und Beschreibung Integrationstests	14
2.1	Bausteinschicht	14
2.1.1	Komponentendiagramm	14
2.1.2	Klassendiagramm	14
2.2	Laufzeitsicht	15
2.2.1	Anmeldung	15
2.2.2	Einwilligung erteilen	16
2.2.3	Einstellungen verwalten	17
2.2.4	Schlafprotokoll ausfüllen	18
2.2.5	Bericht ansehen	19
2.2.6	Bericht ansehen	19
2.3	Verteilungssicht	20
2.3.1	Deploymentdiagramm	20
2.4	Integrationstests	21
3	Dokumentation der Testdurchführung	24
3.1	Ziel	24
3.2	Testumgebung	24
3.3	Durchführung	24
3.3.1	AnswerTest	24
3.3.2	QuestionTest	24
3.3.3	QuestionTypeAdapterTest	24
3.4	Ergebnis	24
3.5	Fazit	25
4	Installations- und Bedienungsanleitung	26

Abbildungsverzeichnis

1	Domänenstruktur: Schlaftagebuch	7
---	---	---

2	Geschäftsprozessdiagramm: Schlaftagebuch	8
3	Use-Case-Diagramm: Schlaftagebuch	9
4	Systemtests	13
5	Komponentendiagramm	14
6	Klassendiagramm	14
7	Anmeldung 1	15
8	Anmeldung 2	15
9	Anmeldung 3	16
10	Einwilligung erteilen	16
11	Einstellungen verwalten Sprache	17
12	Einstellungen verwalten Zeit	17
13	Schlafprotokoll ausfüllen	18
14	Bericht ansehen Patient:in	19
15	Bericht ansehen Ärzt:in	19
16	Deploymentdiagramm	20

1 Anforderungsspezifikation und Beschreibung der Systemtests

Die Software dient der digitalen Erfassung des Schlafverhaltens von Jugendlichen ab 14 Jahren, um Ärzt:innen vor einer Untersuchung im Kinderschlaflabor verlässliche Daten bereitzustellen. Sie ersetzt unzuverlässige Papierfragebögen durch eine mobile App, in der Jugendliche täglich ein Morgenprotokoll ausfüllen. Optional können Smartwatch-Daten integriert werden. Die Ergebnisse werden automatisch ausgewertet und als PDF- oder Excel-Bericht an das medizinische Personal übermittelt.

Randbedingungen und Qualitätsziele:

- **Datenschutz:** Nutzung nur mit Einverständnis; Zugriff ausschließlich für Jugendliche und Ärzt:innen.
- **Mehrsprachigkeit:** Deutsch, Englisch, weitere Sprachen wählbar.
- **Benutzerfreundlichkeit:** Einfache, intuitive Bedienung (max. 5 Minuten täglich).
- **Zuverlässigkeit und Sicherheit:** Verschlüsselte Datenspeicherung, fehlerfreie Übertragung.

1.1 Glossar

1.1.1 Fachbegriffe

Patient:in — Nutzer:in der Anwendung, die/der täglich ein Schlafprotokoll ausfüllt. Verantwortlich für das optionale Koppeln einer Smartwatch.

Ärzt:in — Medizinisches Fachpersonal, das die exportierten Berichte zur Auswertung erhält.

Nutzer:in — Eine Person, die das System verwendet. Der Begriff umfasst sowohl Patient:innen als auch Ärzt:innen.

Patient:innen nutzen das System zur täglichen Eingabe und Verwaltung ihrer Schlafdaten. Ärzt:innen verwenden das System, um Berichte einzusehen und die Schlafdiagnostik zu unterstützen.

Nutzerkonto — Persönliches App-Konto einer Patient:in oder Ärzt:in, das mit einem einmaligen Zugangscode aktiviert wird.

Personenbezogene Daten (PbD) — Alle Informationen, die eine natürliche Person eindeutig identifizieren oder identifizierbar machen.

Im Kontext unseres Schlaf-Tagebuchs gehören dazu ausschließlich die Stammdaten der Nutzer:innen:

- Vorname
- Nachname
- Geburtsdatum

Diese Daten dienen ausschließlich der Zuordnung eines Nutzerkontos zu einer konkreten Person und sind gemäß DSGVO besonders zu schützen.

Erziehungsberechtigte — Gesetzliche Vertretung einer minderjährigen Patient:in; ab 18

übernimmt die/der Patient:in diese Rolle selbst.

Schlafstagebuch — Digitale Anwendung zur täglichen Dokumentation des Schlafverhaltens. Speichert, verwaltet und synchronisiert Schlafdaten und Protokolle und steuert den Anmeldevorgang. Leitet Daten an Berichte weiter und ermöglicht den Export.

Einwilligung zur Datenverarbeitung — Zustimmung zur Erhebung, Speicherung und Verarbeitung personenbezogener Gesundheitsdaten gemäß DSGVO.

Einmaliger Zugangscode — Von der Klinik ausgegebener Code zur erstmaligen Anmeldung und Aktivierung des Nutzerkontos.

Anmeldung — Zugang zur App und Aktivierung des Nutzerkontos; umfasst auch die Verwaltung und Änderung der Zugangscodes.

Benachrichtigung — Tägliche Erinnerung der App an das Ausfüllen des Schlafprotokolls.

Benachrichtigungszeit — Von der Patient:in gewählte Uhrzeit für die tägliche Erinnerung.

Schlafprotokoll — Fragebogen zur Selbsteinschätzung der gesundheitlichen und emotionalen Verfassung, zur Schlafsituation sowie zur Erfassung der Schlafdauer.

Schlafdauer — Automatisch berechnete Zeitspanne zwischen Einschlafzeit und Aufwachzeit.

Smartwatch — Tragbares Gerät, das Schlafdaten automatisch erfasst und mit der App synchronisiert; optional.

Schlafdaten — Von der Smartwatch gesammelte Daten der Nacht; optional und werden vom Schlafstagebuch verwaltet.

Synchronisation — Abgleich der in der App eingegebenen Daten mit automatisch erfassten Smartwatch-Daten.

Visualisierung — Grafische Darstellung der erfassten Schlafdaten.

Bericht / Dokument — Automatisch generiertes Dokument, das die gespeicherten Schlafdaten zusammenfasst.

Generierung / Erstellung — Automatischer Prozess der Berichtserzeugung.

Export — Bereitstellung des Berichts zum Herunterladen oder sicheren Übermitteln an die Klinik.

1.1.2 Prozessbegriffe

Einmaligen Zugangscode erhalten — Empfang des von der Klinik bereitgestellten Codes zur Aktivierung des Nutzerkontos.

Anmelden — Erstellung eines Nutzerkontos mit klinischen Zugangsdaten; umfasst auch deren Änderung.

Zugangsdaten ändern — Ändern des Anmeldepassworts durch die Patient:in.

Einwilligung erteilen — Zustimmung zur Datenverarbeitung innerhalb der App.

Einstellungen verwalten — Übergeordneter Begriff für Spracheinstellung und Benachrichtigungszeit.

Sprache auswählen — Festlegung der App-Sprache (Deutsch, Englisch, Russisch).

Benachrichtigungszeit auswählen — Festlegung der Uhrzeit für die tägliche Erinnerung.

Benachrichtigung erhalten — Empfang der Erinnerung zum Ausfüllen des Schlafprotokolls.

Smartwatch koppeln — Verbinden der App mit einer kompatiblen Smartwatch zur Synchronisation.

Daten synchronisieren — Abgleich der manuell eingegebenen Informationen mit Smartwatch-Daten.

Visualisierung ansehen — Öffnen und Betrachten der grafischen Darstellung der Schlafdaten.

Schlafprotokoll ausfüllen — Beantworten der Fragen zum Schlafverhalten und Tagesbefinden.

Bericht erstellen / generieren — Automatische Erstellung eines Berichts aus allen vorhandenen Daten.

Daten exportieren — Übermittlung des Berichts an die/den Ärzt:in und Speichern im Protokoll.

1.2 Domänenstruktur

Die Domänenstruktur des digitalen Schlaftagebuchs beschreibt die wichtigsten Beteiligten (Patient:in, Ärzt:in), zentrale Artefakte (Schlafprotokoll, Bericht) sowie deren Beziehungen. Abbildung 1 zeigt diese Struktur als Klassendiagramm.

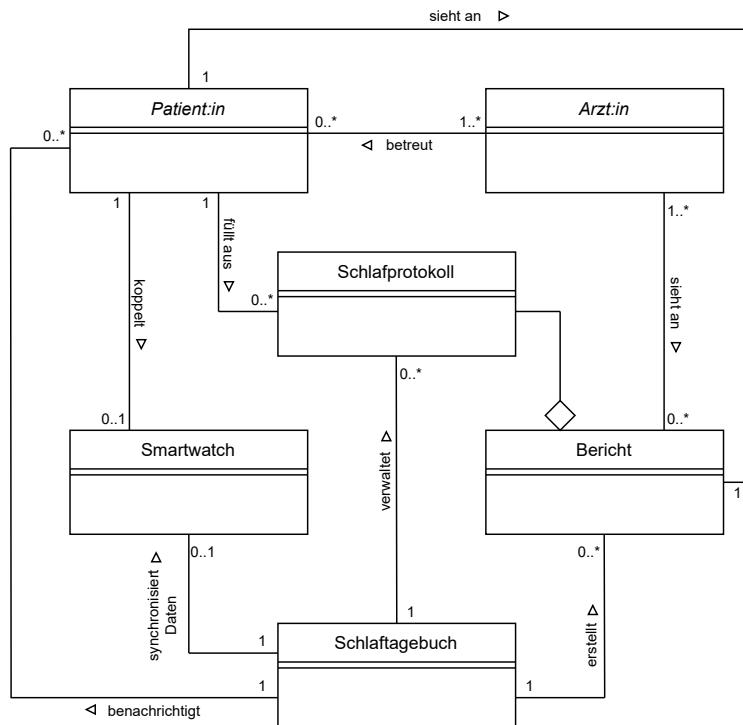


Abbildung 1: Domänenstruktur: Schlaftagebuch

Eine Patient:in füllt regelmäßig Schlafprotokolle aus; jedes Protokoll gehört genau einer Patient:in ($0..^* \rightarrow 1$), während eine Patient:in beliebig viele Protokolle besitzen kann (Im besten Fall aber nur 14). Diese Protokolle werden im Schlaftagebuch gesammelt.

Optional kann eine Smartwatch mit einer Patient:in gekoppelt werden (Patient:in $1 \rightarrow 0..1$ Smartwatch). Sie stellt automatisch erfasste Schlafdaten bereit, die mit dem Schlaftagebuch synchronisiert werden können.

Aus einem oder mehreren Schlafprotokollen wird ein Bericht erstellt. Die Aggregationsbeziehung zeigt, dass ein Bericht Protokolle zusammenfasst, die jedoch unabhängig weiterexistieren können. Eine Ärzt:in erhält diese Berichte zur Auswertung und betreut die jeweilige Patient:in (Ärzt:in $1 \rightarrow 0..^*$ Patient:innen).

1.3 Geschäftsprozess

Der Geschäftsprozess beschreibt den Ablauf der Schlafdiagnostik für Jugendliche im Schlaflabor. Ziel ist es, Schlafstörungen zu erkennen und eine Diagnose zu stellen. Die Software unterstützt diesen Prozess, indem sie digitale Schlafprotokolle sammelt, vervollständigt, validiert und in einem automatisch generierten Bericht zusammenfasst. Ärzt:innen nutzen diesen Bericht zur Auswertung und zur Erstellung einer Diagnose.

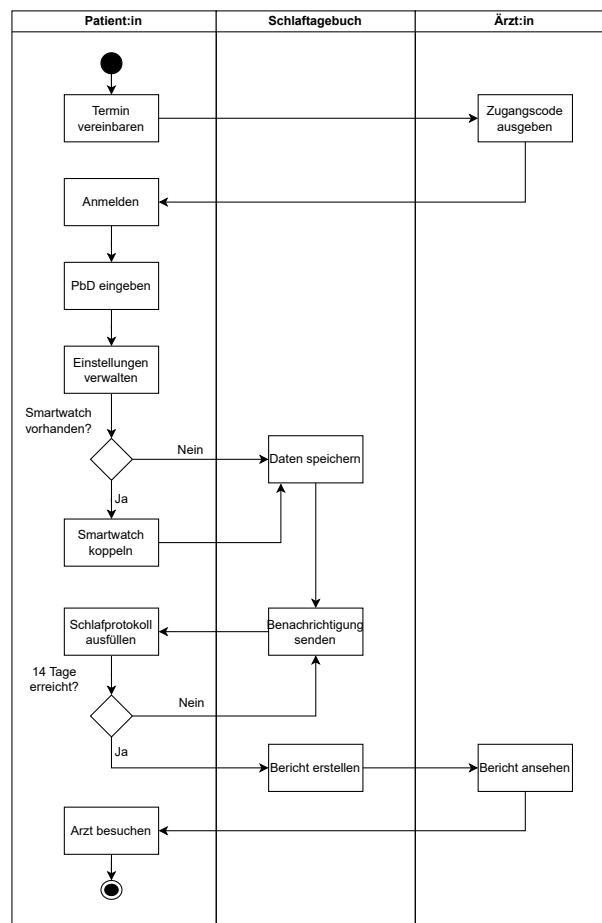


Abbildung 2: Geschäftsprozessdiagramm: Schlaftagebuch

1.4 Systemfunktionen

1.4.1 Übersicht

Die dargestellte Use-Case-Diagramm zeigt die zentralen Systemfunktionen des Schlaftagebuchs aus der Perspektive der beiden Hauptakteure: **Patient:in** und **Arzt:in**. Es veranschaulicht, welche Aktionen die jeweiligen Nutzergruppen mit dem System durchführen können und wie sie mit den Kernfunktionalitäten interagieren.

Patient:innen sind die primären Anwender:innen des Systems. Sie können sich im Schlaftagebuch anmelden, ihre Einstellungen verwalten und eine Smartwatch mit dem System koppeln, um Schlafdaten automatisiert zu erfassen. Darüber hinaus haben Patient:innen die Möglichkeit, täglich ein Schlafprotokoll auszufüllen, in dem subjektive und objektive Angaben zum Schlafverhalten erfasst werden.

Ärzt:innen nutzen das System vor allem zur Einsicht und Auswertung der protokollierten Daten. Sie können die von Patient:innen erstellten PDF-Berichte ansehen, um einen Überblick über Schlafmuster, Auffälligkeiten und relevante Gesundheitsinformationen zu erhalten.

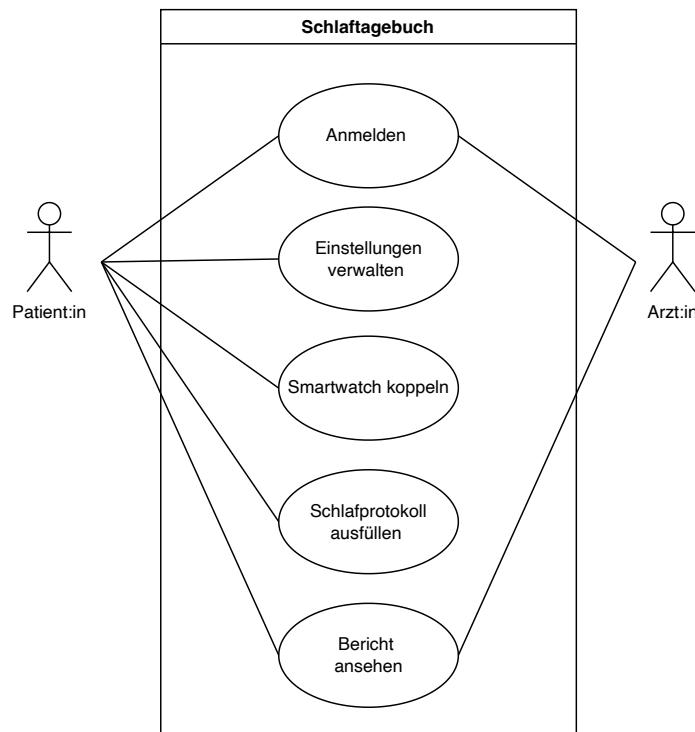


Abbildung 3: Use-Case-Diagramm: Schlaftagebuch

1.4.2 Beschreibung pro Use Case

Use Case: Anmelden

Titel: Anmelden

Kurzbeschreibung: Aktivierung und Anmeldung des Nutzerkontos mithilfe eines einmaligen Zugangscodes.

Aktoren: Nutzer:in (Patient:in oder Ärzt:in)

Vorbedingungen: Ein einmaliger Zugangscode liegt vor.

Ablauf:

1. Patient:in gibt den Zugangscode ein.
2. System aktiviert das Nutzerkonto.
3. Patient:in meldet sich an.

Auswirkungen: Nutzerkonto wird freigeschaltet und ist vollständig nutzbar.

Anmerkungen: Bei falschem Zugangscode erscheint eine Fehlermeldung.

Use Case: Einstellungen verwalten

Titel: Einstellungen verwalten

Kurzbeschreibung: Änderung der Sprache und der täglichen Benachrichtigungszeit.

Aktoren: Patient:in

Vorbedingungen: Nutzerkonto ist angemeldet.

Ablauf:

1. Öffnen des Einstellungsbereichs.
2. Auswahl der gewünschten Sprache.
3. Festlegung der täglichen Erinnerungszeit.
4. System speichert die neuen Einstellungen.

Auswirkungen: App-Sprache und Benachrichtigungen passen sich an.

Anmerkungen: Einstellungen können jederzeit geändert werden.

Use Case: Smartwatch koppeln

Titel: Smartwatch koppeln

Kurzbeschreibung: Verbindung der App mit einer kompatiblen Smartwatch zur automatischen Synchronisation von Schlafdaten.

Aktoren: Patient:in

Vorbedingungen: Eine Smartwatch ist vorhanden und eingeschaltet.

Ablauf:

1. Patient:in startet den Kopplungsvorgang.
2. App sucht verfügbare Geräte.
3. Patient:in wählt die Smartwatch aus.
4. System stellt die Verbindung her.

Auswirkungen: Nutzerkonto wird freigeschaltet und ist vollständig nutzbar.

Anmerkungen: Bei falschem Zugangscode erscheint eine Fehlermeldung.

Use Case: Schlafprotokoll ausfüllen

Titel: Schlafprotokoll ausfüllen

Kurzbeschreibung: Tägliche Eingabe der subjektiven und objektiven Angaben zum Schlaf.

Aktoren: Patient:in

Vorbedingungen: Nutzerkonto ist angemeldet.

Ablauf:

1. Öffnen des Formulars für den aktuellen Tag.
2. Eintragen von Schlafdauer, Einschlafzeit und Aufwachzeit.
3. Bewertung der Schlafqualität und des Wohlbefindens.
4. System speichert das Protokoll.

Auswirkungen: Das Tagesprotokoll ist vollständig gespeichert.

Anmerkungen: Pro Tag kann nur ein Protokoll abgegeben werden.

Use Case: Bericht ansehen

Titel: Bericht ansehen

Kurzbeschreibung: Ärzt:in öffnet den automatisch generierten Bericht über die Schlafdaten.

Aktoren: Ärzt:in

Vorbedingungen: Ein Bericht wurde erstellt und die Ärzt:in ist autorisiert.

Ablauf:

1. Ärzt:in öffnet die Berichtsübersicht.
2. Auswahl eines gewünschten Berichts.
3. System lädt und zeigt das Dokument.

Auswirkungen: Ärzt:in kann Schlafmuster analysieren und diagnostische Einschätzungen treffen.

Anmerkungen: Zugriff nur auf Berichte der zu betreuenden Patient:innen.

1.5 Beschreibung Systemtests

- Ein Systemtest pro Use Case für den Standardablauf
- Zusätzliche Tests für alternative Abläufe

Use-Case	Titel	Input	Output	Fehler
Anmelden	Zugangscode eingeben	App installieren und öffnen, dann Zugangscode vom Arzt eingeben	Nutzer wird zur Eingabe des Passworts weitergeleitet	Server-Probleme, falscher Zugangscode, Zugriff wird verweigert
	Passwort eingeben	Passwort eingeben	Passwort ist gültig und Nutzer wird eingeloggt/ein Nutzerkonto erstellt	Passwort ist falsch, fälschlicherweise eingeloggt
Einstellung verwalten	Sprache auswählen	Nutzer wählt die Sprache aus, die er/sie bevorzugt	Sprache in der App ist nun zur ausgewählten Sprache gewechselt	Sprachübersetzung funktioniert nicht richtig, Nutzer hat die falsche Sprache ausgewählt
Smartwatch koppeln	Smartwatch koppeln	Nutzer koppelt optional eine Smartwatch mit der App	Smartwatch ist nun gekoppelt mit App und nimmt Schlafdaten auf	Keine Smartwatch vorhanden, Verbindung schlecht
Schlafprotokoll ausfüllen	Schlafdaten speichern	Eingegebene Daten endgültig speichern	Daten werden final auf der Datenbank gespeichert	Datenbank ist nicht erreichbar, Serverfehler
	Zeitangaben tätigen	Uhrzeit mit Stunden und Minuten angeben	Zeitangabe wird in der App gespeichert	Es wird eine ungültige Zeitangabe getätig
	Schlafprotokoll auslassen	Schlafprotokoll wird ausgelassen, weil der Nutzer es vergessen hat, er/sie keine Zeit oder keine Lust hat	Daten für diesen Tag nicht vorhanden	Ungenügend Daten nach 14 Tagen für den Bericht
Bericht ansehen	Bericht wird erstellt	Bericht wird erstellt mit den Daten über die letzten 14 Tage	Bericht wird generiert und kann angesehen werden	Fehler bei Verarbeitung der Daten (ungültige Daten, nicht genügend Daten, ...)
	Bericht anzeigen	Nutzer ruft den Bericht in der App auf	Bericht wird dem Nutzer angezeigt (Schlafdaten und generierte Visualisierung)	Bericht wird nicht vollständig oder gar nicht generiert

Abbildung 4: Systemtests

2 Architekturbeschreibung und Beschreibung Integrationstests

2.1 Bausteinschicht

2.1.1 Komponentendiagramm

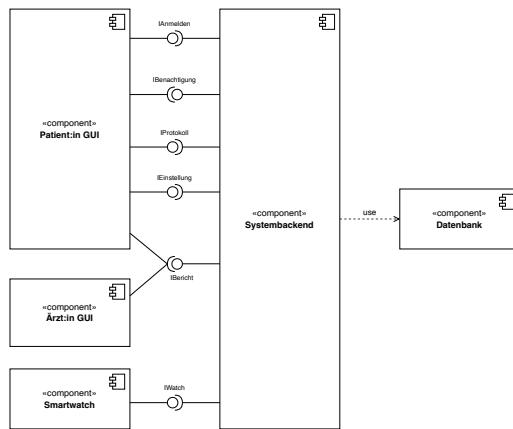


Abbildung 5: Komponentendiagramm

2.1.2 Klassendiagramm

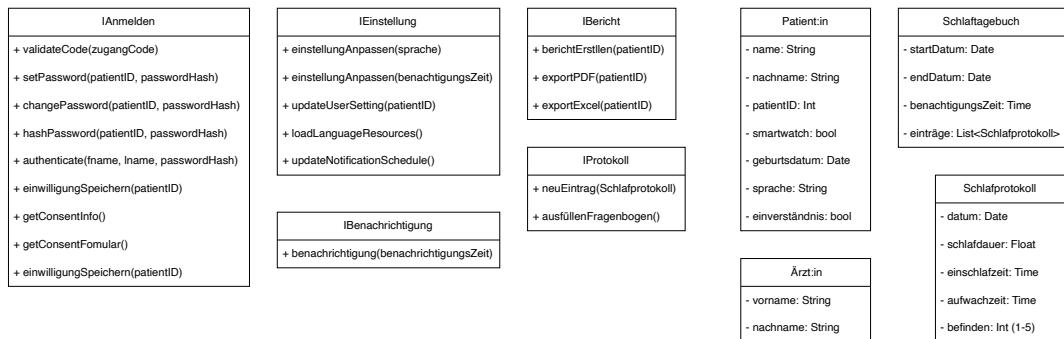


Abbildung 6: Klassendiagramm

2.2 Laufzeitsicht

2.2.1 Anmeldung

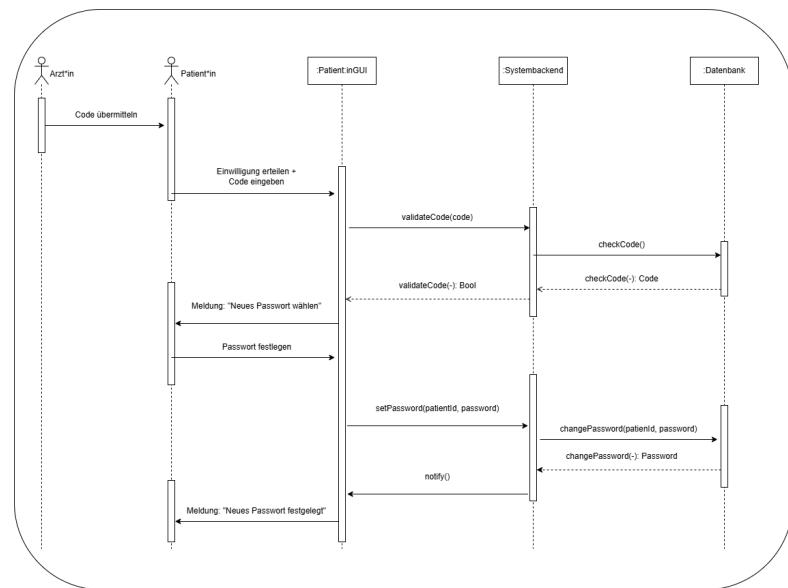


Abbildung 7: Anmeldung 1

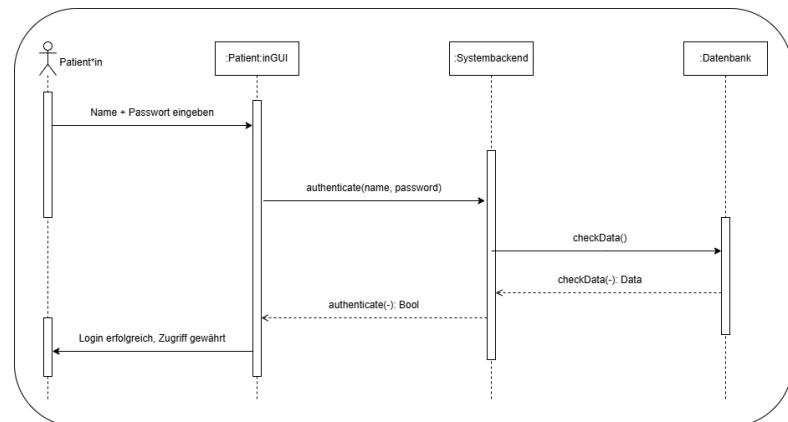


Abbildung 8: Anmeldung 2

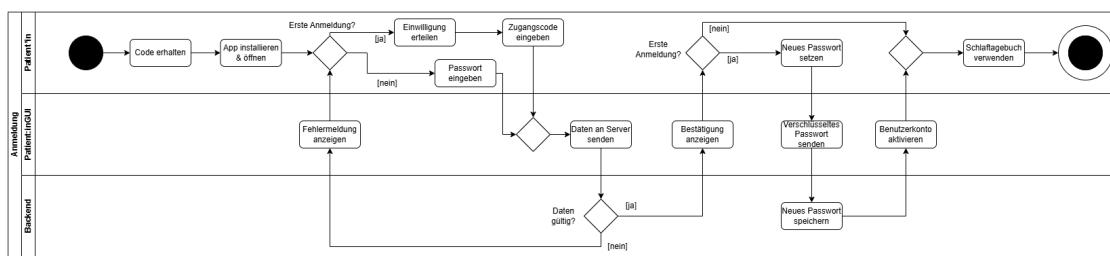


Abbildung 9: Anmeldung 3

2.2.2 Einwilligung erteilen

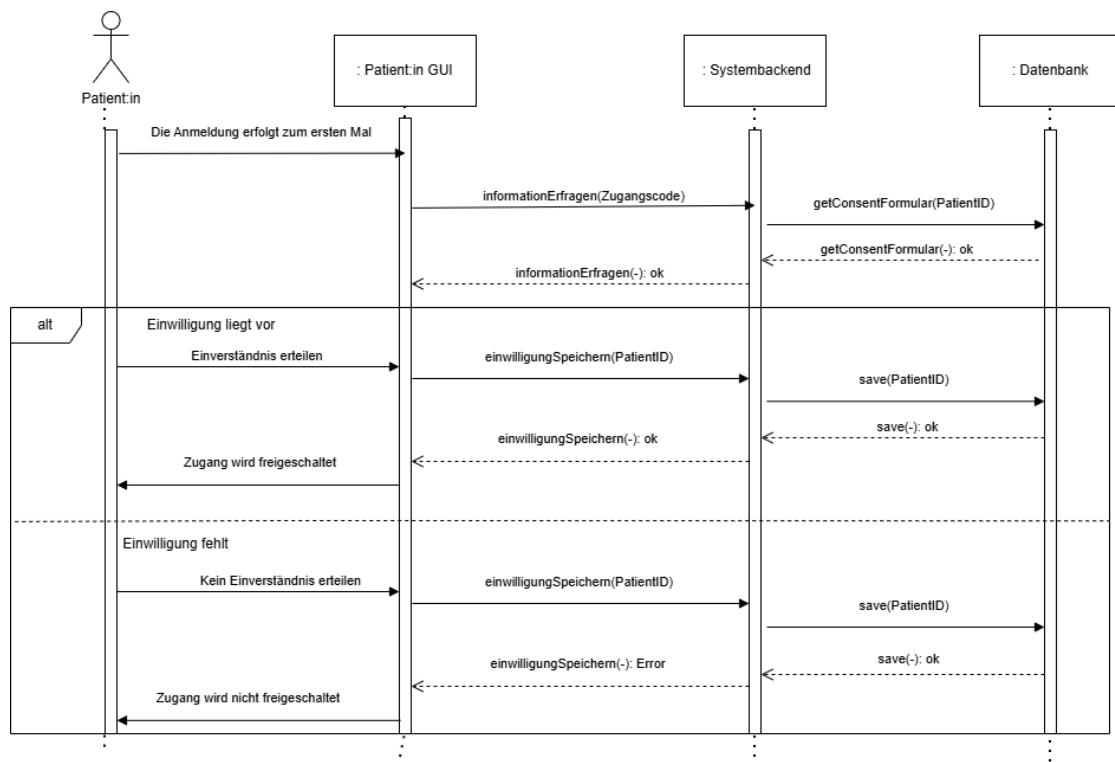


Abbildung 10: Einwilligung erteilen

2.2.3 Einstellungen verwalten

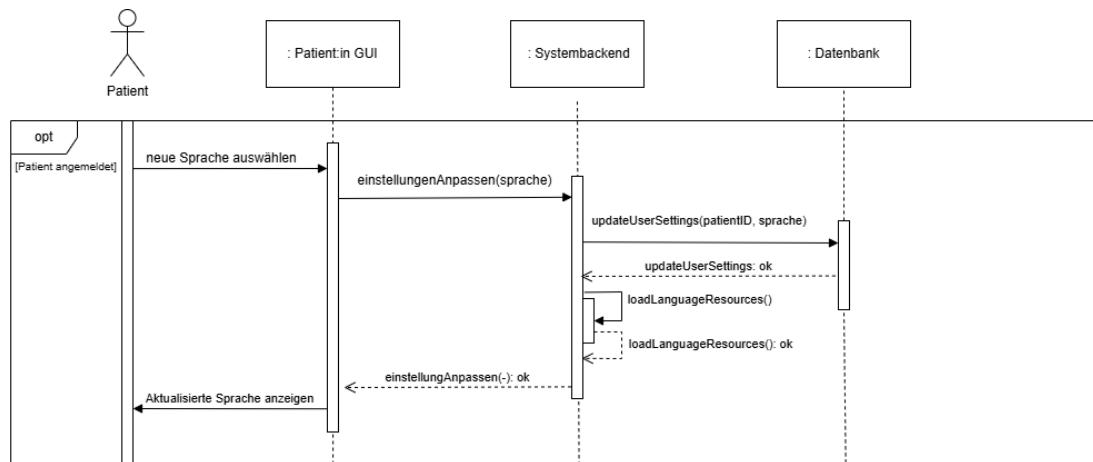


Abbildung 11: Einstellungen verwalten Sprache

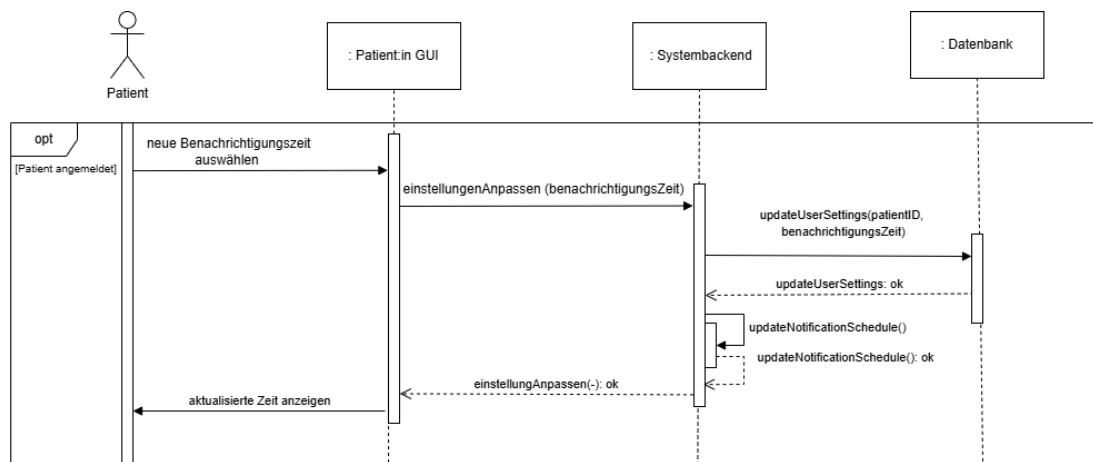


Abbildung 12: Einstellungen verwalten Zeit

2.2.4 Schlafprotokoll ausfüllen

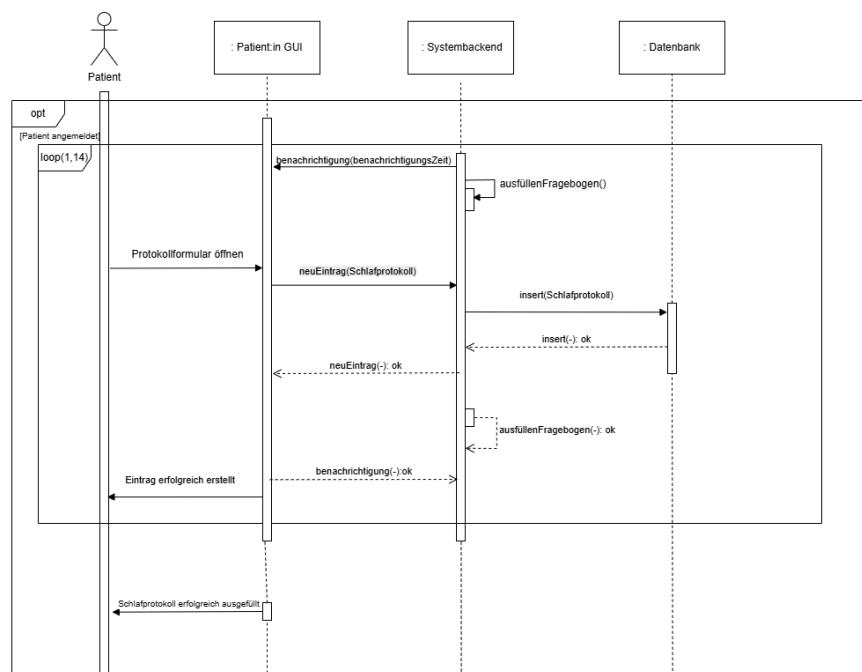


Abbildung 13: Schlafprotokoll ausfüllen

2.2.5 Bericht ansehen

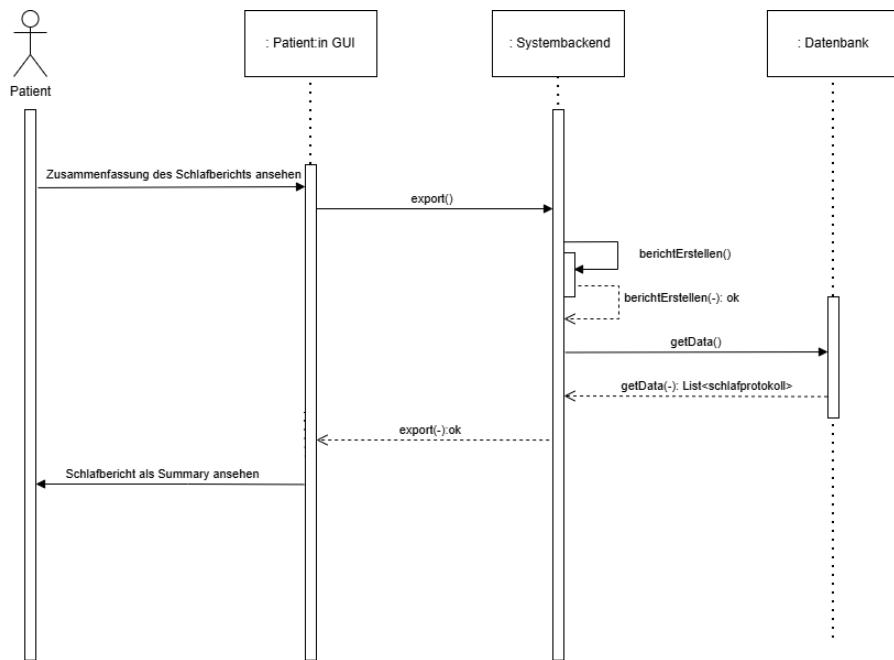


Abbildung 14: Bericht ansehen Patient:in

2.2.6 Bericht ansehen

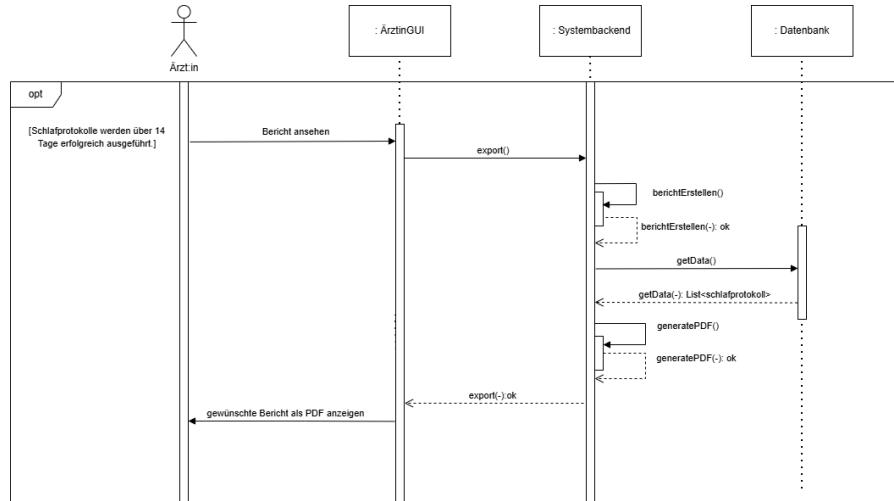


Abbildung 15: Bericht ansehen Ärzt:in

2.3 Verteilungssicht

2.3.1 Deploymentdiagramm

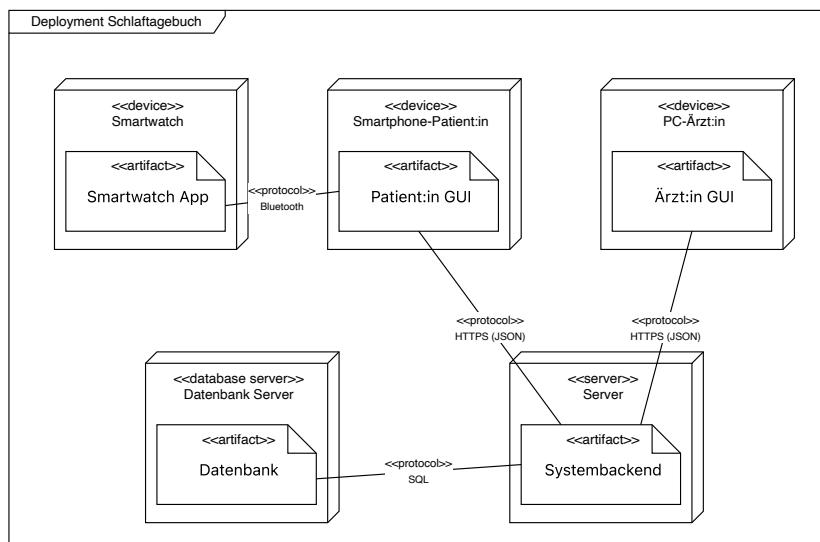


Abbildung 16: Deploymentdiagramm

2.4 Integrationstests

1. *Schlafprotokoll ausfüllen*

Test 1.1: Erfolgreich neues Schlafprotokoll speichern

Ziel	Prüfen, ob ein neuer Protokolleintrag korrekt von GUI
Vorbedingungen	Patient ist angemeldet
Eingaben (GUI)	Einschlafzeit: 22:30, Aufwachzeit: 07:10, Befunden: 4 und Datum wird automatisch eingetragen
Testschritte	1. Patient öffnet Protokollformular, GUI ruft <i>ausfüllenFrageboegen()</i> auf 2. GUI sendet Daten an Backend via <i>neuenEintrag()</i> 3. Backend validiert Daten 4. Backend ruft DB-Methode <i>insert(Schlafprotokoll)</i> auf
Erwartete Backend-DB Interaktionen	<i>insert()</i> wird einmal mit vollständigem Datensatz ausgeführt
Erwartete Ausgabe (GUI)	„Eintrag erfolgreich erstellt“
Fehler	Keine

Test 1.2: Ungültige Schlafzeiten

Ziel	Überprüfung von eingetragenen Daten
Eingabe	Einschlafzeit: 23:00, Aufwachzeit: 22:00 (am selben Tag)
Erwartetes Ergebnis	Backend erkennt den Fehler, sendet eine Fehlermeldung an GUI wie z.B. „ungültige Zeiten“ und <i>insert()</i> wird nicht ausgeführt.

2. Einwilligung erteilen

Test 2.1: Einwilligung vorhanden

Ziel	Prüfen, dass Einwilligung korrekt gespeichert wird
Vorbedingung	Patient gibt Zugangscode eine
Schritte	1. GUI → Backend: <i>informationErfragen()</i> 2. Backend → DB: <i>getConsentFormular(patientID)</i> 3. Patient klickt „Einverstanden“ → GUI → Backend: <i>einwilligungSpeichern()</i> 4. Backend → DB: <i>save(patientID)</i>
Erwartet	DB speichert <i>einverständnis=true</i> , GUI zeigt „Zugang freigeschaltet“

Test 2.2: Einwilligung verweigert

Ziel	Verhindern, dass Minderjährige ohne Zustimmung starten
Schritte	Gleich wie oben, aber Patient „Ablehnen“
Erwartet	DB speichert <i>false</i> , GUI zeigt „Zugang nicht möglich“

3. Bericht ansehen

Test 3.1: PDF wird Korrekt generiert

Ziel	Prüfen, ob Backend die Richtigen Daten holt und PDF generiert.
Schritte	1. Arzt wählt Patient 2. GUI → Backend: <i>berichtErstellen(patientID)</i> 3. Backend → DB: <i>getReport(patientID)</i> 4. Backend generiert PDF via <i>generatePDF()</i> 5. PDF an GUI senden
Erwartet	PDF enthält aller Einträge der Schlafprotokoll einer Patient

Test 3.2: Keine Einträge vorhanden

Kategorie	Beschreibung
Eingabe	PatientID existiert, aber keine Protokolle
Erwartet	Backend liefert eine PDF ohne Datensätze

4. Anmelden

Test 4.1: Login möglich

Ziel	Prüfen, dass Registrierung funktioniert
Schritte	1. GUI → Backend: <i>validateCode(code)</i> 2. Backend → DB: <i>checkCode()</i> 3. GUI → Backend: <i>setPassword()</i> 4. Backend → DB: <i>changePassword()</i>
Erwartet	Passwort gehasht gespeichert, Login möglich

Test 4.2: Kein Login möglich

Kategorie	Beschreibung
Eingabe	Ungültiger Code als Passwort eingeben
Erwartet	Fehlermeldung: „Ungültiger Code“ kein Passwort darf gespeichert werden

5. Einstellung verwalten

Test 5.1: Sprache erfolgreich ändern

Schritte	GUI → Backend: <i>einstellungAnpassen("EN")</i> Backend → DB: <i>updateUserSettings()</i> Backend lädt <i>loadLanguageResources()</i>
Erwartet	GUI wird vollständig auf Englisch aktualisiert werden

Test 5.2: SprachPaket fehlt

Schritte	Gleich wie oben
Erwartet	Backend-Fehler → GUI: „Sprache nicht verfügbar“

Test 5.3: Erinnerung erfolgreich ändern

Schritte	GUI → Backend: <i>einstellungAnpassen(benachrichtigungsZeit)</i> Backend → DB: <i>updateUserSettings()</i> Backend setzt Timer
Erwartet	Neue Zeit erscheint im GUI

Test 5.4: Ungültige Zeit auswählen

Eingabe	Uhrzeit: 99:99
Erwartet	Fehlermeldung: „Ungültiger Code“ kein Passwort darf gespeichert werden

3 Dokumentation der Testdurchführung

3.1 Ziel

Die Unit-Tests sollen zeigen, ob unsere Datenklassen richtig funktionieren und ob der `QuestionTypeAdapter` die JSON-Daten richtig einliest.

3.2 Testumgebung

Alle Tests wurden in Android Studio unter `src/test/java` ausgeführt. Die Tests laufen ohne Handy, nur auf der JVM.

3.3 Durchführung

3.3.1 *AnswerTest*

Hier wurde geprüft, ob die `Answer`-Klasse die Werte richtig speichert. Das hat funktioniert.

3.3.2 *QuestionTest*

Es wurde getestet, ob die verschiedenen Frageklassen (z. B. Slider, Yes/No, Multiple Choice) die Daten korrekt übernehmen. Auch das funktioniert.

3.3.3 *QuestionTypeAdapterTest*

Hier wurde geprüft, ob der Adapter aus dem `type`-Feld im JSON die richtige Frageart macht. Beispiele:

- "QuestionSlider" wird zu `QuestionSlider`
- "QuestionTime" wird zu `QuestionTime`

Außerdem wurden zwei Fehler getestet:

- wenn `type` im JSON fehlt
- wenn ein falscher `type` drinsteht

In beiden Fällen wurde richtig eine Fehlermeldung ausgelöst.

3.4 Ergebnis

Alle Tests waren erfolgreich. Die Daten werden richtig eingelesen und falsches JSON wird erkannt.

3.5 Fazit

Die wichtigsten Funktionen für den Fragebogen arbeiten wie erwartet. Damit ist die Grundlage für die weitere Entwicklung gelegt.

Testname	Was wurde getestet?	Status
AnswerTest	Prüft, ob die Answer-Klasse Werte wie <code>id</code> und <code>result</code> korrekt speichert.	Bestanden
QuestionTest	Testet, ob die verschiedenen Frageklassen ihre Felder korrekt speichern.	Bestanden
QuestionTypeAdapterTest	Prüft, ob der Adapter den richtigen Fragetyp aus der JSON-Datei erkennt und Fehler bei fehlendem oder falschem <code>type</code> auslöst.	Bestanden

Tabelle 1: Tabelle der durchgeföhrten Unit-Tests

4 Installations- und Bedienungsanleitung

Literatur