

left=2cm, right=2cm, top=2cm, bottom=2cm

Hochschule Bremerhaven

Fachbereich II
Management und Informationssysteme
Wirtschaftsinformatik B.Sc.

Modul
Qualitätsmanagement

Semesteraufgabe

Entwicklung einer Hausverwaltung

Vorgelegt von:	Junior Lesage Ekane Njoh	MatNr. 40128
	Steve Aguiwo II	MatNr. 40088
	Franck Majeste Dogmo Silatsa	MatNr. 00000
Vorgelegt am:	20. Februar 2025	
Dozent:in:	Prof. Dr. Karin Vosseberg	

Inhaltsverzeichnis

1	Einleitung	5
2	Anforderungsanalyse	5
2.1	Review der Anforderungen	5
2.2	Verbesserung der Anforderungen	5
3	Testkonzept	5
3.1	Auswahl von Testverfahren	5
3.2	Teststufen und Testarten	5
3.3	Testumgebung und Testdaten	5
4	Entwicklung der Testfälle	5
4.1	Ableitung konkreter Testfälle	5
4.2	Erstellung einer Testfall-Dokumentation	5
5	Prototypische Umsetzung der Hausverwaltung	5
5.1	Software-Architektur und Technologien	5
5.2	Implentierung	5
5.3	Anwendung des Testkonzepts	5
6	Qualitätsmanagement-Methoden in der Softwareentwicklung	5
6.1	Relevanz der Qualitätssicherung	5
6.2	Anwendung von QS-Methoden im Projekt	5
7	Einleitung	5
	Literaturverzeichnis	6
	Listingverzeichnis	6
	Anhang	7
I	Review-Protokoll der Anforderungen an die Hausverwaltung	8
II	Verbesserte Anforderungen auf Review-Basis	11
III	Vorher-Nachher-Vergleich	11
IV	Testkonzept	16
IV.1	Einleitung	16
IV.2	Testziele und Strategie	16
IV.3	Ausgewählte Testverfahren und Begründung	17
IV.4	Testumgebung und Testfälle	17
IV.5	Fazit	18
V	Konkrete Testfälle für die Hausverwaltungssoftware	19

VI	Prototyp-Dokumentation	21
	Selbstständigkeitserklärung	22

1 Einleitung

2 Anforderungsanalyse

2.1 Review der Anforderungen

2.2 Verbesserung der Anforderungen

3 Testkonzept

3.1 Auswahl von Testverfahren

3.2 Teststufen und Testarten

3.3 Testumgebung und Testdaten

4 Entwicklung der Testfälle

4.1 Ableitung konkreter Testfälle

4.2 Erstellung einer Testfall-Dokumentation

5 Prototypische Umsetzung der Hausverwaltung

5.1 Software-Architektur und Technologien

5.2 Implementierung

5.3 Anwendung des Testkonzepts

6 Qualitätsmanagement-Methoden in der Softwareentwicklung

6.1 Relevanz der Qualitätssicherung

6.2 Anwendung von QS-Methoden im Projekt

7 Einleitung

Listingverzeichnis

Anhang

I Review-Protokoll der Anforderungen an die Hausverwaltung

Review der Anforderungen

Methode des Reviews:

- Es wurde ein technisches Review nach ISO 20246 durchgeführt.
- Die Überprüfung erfolgte anhand folgender Kriterien:
 - Vollständigkeit
 - Eindeutigkeit
 - Widerspruchsfreiheit
 - Testbarkeit der Anforderungen
- Zusätzlich wurden relevante Inhalte aus den Vorlesungsfolien zum Thema Qualitätsmanagement, Softwaretest und Anforderungsanalyse berücksichtigt.

Identifizierte Probleme und Unklarheiten

Anforderung 1: Gebäudestruktur (1..n Gebäude, Eingänge, Wohnungen, Zähler)

Problem/Unklarheit: Keine klare Definition von „Eingang“. Ist ein Eingang ein Gebäudeteil oder eine logische Struktur?

Verbesserungsvorschlag: Definition eines Eingangs hinzufügen (z. B. „Ein Eingang ist eine physische oder logische Einheit, die Zugang zu Wohnungen ermöglicht.“).

Anforderung 2: Verschiedene Zählertypen (Strom, Gas, Wasser)

Problem/Unklarheit: Sind weitere Zählertypen möglich? Falls ja, wie werden sie erfasst?

Verbesserungsvorschlag: Klarstellung, ob die Liste erweiterbar ist und wie neue Zählertypen ergänzt werden können.

Anforderung 3: Zähler eindeutig identifizierbar (Zählernummer)

Problem/Unklarheit: Keine Angabe, welches Format oder welche Länge die Zählernummer haben muss.

Verbesserungsvorschlag: Definition des Formats der Zählernummer (z. B. „Die Zählernummer besteht aus einer eindeutigen 10-stelligen alphanumerischen ID.“).

Anforderung 4: Auswahl von Daten per Selektion in der Struktur

Problem/Unklarheit: Welche Filter- und Suchmöglichkeiten gibt es?

Verbesserungsvorschlag: Ergänzung der Anforderungen zur Filterung (z. B. Suche nach Gebäude, Wohnung oder Zählertyp).

Anforderung 5: Zähler haben einen Ablesewert (ganze Zahl)

Problem/Unklarheit: Was passiert bei fehlerhafter Eingabe? Kann der Wert korrigiert werden?

Verbesserungsvorschlag: Spezifikation einer Fehlerbehandlung für falsche Eingaben.

Anforderung 6: Zähler sind über ihre ID zu finden

Problem/Unklarheit: Was passiert, wenn eine ID nicht existiert?

Verbesserungsvorschlag: Definition einer Fehlermeldung für nicht gefundene IDs.

Anforderung 7: Zähler sollen abgelesen werden (Eingabe von Datum und Wert)

Problem/Unklarheit: Gibt es eine Validierung für vergangene/future Daten?

Verbesserungsvorschlag: Klarstellung, ob das Ablesedatum nur in der Vergangenheit oder auch in der Zukunft liegen darf.

Anforderung 8: Zähler und Datum laufen nur vorwärts

Problem/Unklarheit: Fehlt eine Angabe zu Testfällen (z. B. wie rückdatierte Werte behandelt werden).

Verbesserungsvorschlag: Testfälle für Grenzwerte (min/max Werte für Datum) spezifizieren.

Anforderung 9: Weitere Ableseinformationen eingeben (Ablesung, Schätzung)

Problem/Unklarheit: Müssen Nutzer einen Ablesetyp zwingend angeben oder gibt es Standardwerte?

Verbesserungsvorschlag: Standardwert oder Pflichtfeld definieren.

Anforderung 10: Ableser-Informationen eingeben (Hauswart, Mieter, Energieversorger)

Problem/Unklarheit: Können mehrere Ableser für einen Zähler existieren?

Verbesserungsvorschlag: Klärung, ob Mehrfachzuweisungen erlaubt sind.

Anforderung 11: Verbrauch berechnen und Anzeigen

Problem/Unklarheit: Sind historische Verbrauchswerte abrufbar?

Verbesserungsvorschlag: Definition, ob und wie Langzeitverbräuche gespeichert werden.

Verantwortliche Personen und Datum

- Junior Lesage Ekane Njoh
- Franck Majesté Silatsa Dogmo
- Datum: 20.02.2025

II Verbesserte Anforderungen auf Review-Basis

Nach der Überarbeitung der ursprünglichen Anforderungen haben wir die finalen Anforderungen für die Hausverwaltung definiert. Diese berücksichtigen die Ergebnisse des Reviews und wurden klarer formuliert, widerspruchsfrei gestaltet und um spezifische Validierungsregeln ergänzt. Die neuen Anforderungen bilden die Basis für die Implementierung des Prototyps und stellen sicher, dass alle relevanten Aspekte der Hausverwaltung praxisnah und technisch umsetzbar sind.

Tabelle II.1: Verbesserte Anforderungen

Nr.	Anforderung	Beschreibung
1	Gebäudestruktur	Ein Gebäude kann mehrere Eingänge haben. Ein Eingang ist eine physische oder logische Einheit, die Zugang zu Wohnungen ermöglicht. Jede Wohnung hat eine eindeutige ID.
2	Zählertypen	Unterstützte Typen: Strom, Gas, Wasser. Die Liste ist erweiterbar, indem neue Typen über eine Konfigurationsdatei oder Admin-Oberfläche hinzugefügt werden.
3	Zählerverwaltung	Jeder Zähler hat eine eindeutige 10-stellige ID (Format: 'ZZZ-YYYY-NNNN', z. B. '123-2024-4567'). Jeder Zähler gehört zu einer Wohnung und einem Zählertyp. Er speichert den letzten Ablesewert, das letzte Ablesedatum und die Ablesemethode.
4	Datenfilterung	Zähler können nach Gebäude, Wohnung, Zählertyp und Zeitraum gefiltert werden. Die Suche unterstützt Teilstringsuche.
5	Zählerableitung	Zählerwerte können nur mit aktuellem oder zukünftigen Datum erfasst werden. Korrekturen sind nur für Admins erlaubt. Negative Werte sind nicht zulässig. Falls der neue Wert kleiner als der vorherige ist, gibt es eine Fehlermeldung.
6	Fehlermeldungen	Falls eine Zähler-ID nicht existiert, erscheint „Die eingegebene ID existiert nicht. Bitte überprüfen Sie Ihre Eingabe.“ Falls eine Wohnung keiner ID zugeordnet ist, erscheint „Dieser Zähler ist keiner Wohnung zugeordnet.“
7	Verbrauchsanzeige	Historische Verbrauchswerte sind für die letzten 12 Monate abrufbar. Eine grafische Darstellung ist möglich. Monatliche Verbrauchswerte können exportiert werden.
8	Ableser-Informationen	Ableser können Hauswart, Mieter oder Energieversorger sein. Falls keine Information vorhanden ist, wird „Unbekannt“ eingetragen.

III Vorher-Nachher-Vergleich

Wir haben auch einen Vergleich der alten und neuen Anforderungen an der Hausverwaltung erstellt, um Unklarheiten zu beseitigen, Redundanzen zu vermeiden und eine bessere Umsetzbarkeit im Prototyp zu gewährleisten.

Nachfolgend haben wir versucht, es nachvollziehbar zu machen, weshalb die verschiedenen Änderungen da sind, weil wir weniger Anforderungen als ursprünglich haben.

Anforderung 1: Gebäudestruktur (1..n Gebäude, Eingänge, Wohnungen, Zähler)

Vorher: Die Gebäudestruktur enthält mehrere Gebäude, Eingänge und Wohnungen. Es war unklar, ob ein Eingang eine physische oder logische Einheit ist.

Nachher: Ein Gebäude kann mehrere Eingänge haben. Ein Eingang ist eine physische oder logische Einheit, die Zugang zu Wohnungen ermöglicht. Jede Wohnung hat eine eindeutige ID.

Begründung: Die Definition des „Eingangs“ wurde ergänzt, um Unklarheiten zu beseitigen.

Anforderung 2: Zählertypen (Strom, Gas, Wasser)

Vorher: Nur Strom, Gas und Wasser sind als Zählertypen definiert. Es war unklar, ob weitere Zählertypen ergänzt werden können.

Nachher: Zählertypen sind erweiterbar und können über eine Konfigurationsdatei oder Admin-Oberfläche hinzugefügt werden.

Begründung: Die Anforderung wurde verbessert, um zukünftige Erweiterungen zu ermöglichen.

Anforderung 3: Zähler eindeutig identifizierbar (Zählernummer)

Vorher: Die Zählernummer soll eindeutig sein, aber es gibt keine Spezifikation zum Format oder zur Länge.

Nachher: Zähler haben eine 10-stellige eindeutige ID (Format: ‘ZZZ-YYYY-NNNN’, z. B. ‘123-2024-4567’). Jeder Zähler gehört zu einer Wohnung und einem Zählertyp.

Begründung: Das Format der ID wurde spezifiziert, um die Implementierung zu erleichtern.

Anforderung 4: Auswahl von Daten per Selektion in der Struktur

Vorher: Benutzer sollen Daten über eine Selektion in der Struktur auswählen können. Es gibt keine genaue Beschreibung, welche Filter oder Suchoptionen existieren.

Nachher: Benutzer können Zähler nach Gebäude, Wohnung, Zählertyp und Zeitraum filtern. Die Suchfunktion unterstützt Teilstringsuche.

Begründung: Klarstellung der Filter- und Suchmöglichkeiten, um die Funktion für den Prototyp konkret umzusetzen.

Anforderung 5: Zähler haben einen Ablesewert (ganze Zahl)

Vorher: Die Ablesewerte sind Ganzzahlen, aber es war nicht definiert, ob falsche Eingaben möglich sind oder korrigiert werden können.

Nachher: Zählerwerte müssen aktuell oder in der Zukunft liegen. Negative Werte sind nicht zulässig. Admins können Werte korrigieren.

Begründung: Ergänzung von Validierungsregeln zur Fehlervermeidung.

Anforderung 6: Zähler sind über ihre ID zu finden

Vorher: Es war unklar, was passiert, wenn eine eingegebene ID nicht existiert.

Nachher: Falls eine ID nicht existiert, wird eine Fehlermeldung angezeigt: *,„Die eingegebene ID existiert nicht. Bitte überprüfen Sie Ihre Eingabe.“*

Begründung: Explizite Fehlermeldungen wurden definiert, um das Systemverhalten vorhersehbar zu machen.

Anforderung 7: Zähler sollen abgelesen werden (Eingabe von Datum und Wert)

Vorher: Es gab keine Validierung für vergangene oder zukünftige Ablesewerte.

Nachher: Ablesewerte werden nur akzeptiert, wenn sie größer als der vorherige Wert sind. Fehlermeldung bei ungültigen Eingaben.

Begründung: Logische Konsistenzregeln wurden ergänzt, um Fehler zu vermeiden.

Anforderung 8: Zähler und Datum laufen nur vorwärts

Vorher: Es war unklar, ob rückdatierte Werte erlaubt sind oder ob es Grenzwerte für Eingaben gibt.

Nachher: Diese Anforderung wurde mit „Zählerablesung“ zusammengeführt, da sie dieselbe Validierung betrifft.

Begründung: Diese Anforderung ist jetzt Teil von „Zählerablesung“, da eine rückdatierte Eingabe bereits als Fehler behandelt wird.

Anforderung 9: Weitere Ableseinformationen eingeben (Ablesung, Schätzung)

Vorher: Es war nicht spezifiziert, welche Ablesemethoden (manuell, automatisch, Schätzung) hinterlegt werden können.

Nachher: Ablesemethode (manuell, automatisch, Schätzung) wird gespeichert.

Begründung: Diese Information gehört zur Ablesung und wurde integriert.

Anforderung 10: Ableserinformationen eingeben (Hauswart, Mieter, Energieversorger)

Vorher: Es war nicht klar, ob mehrere Ableser für einen Zähler existieren können oder ob eine Standardzuweisung existiert.

Nachher: Ableser können Hauswart, Mieter oder Energieversorger sein. Falls keine Information vorhanden ist, wird „Unbekannt“ eingetragen.

Begründung: Ein Standardwert wurde definiert, um unvollständige Daten zu vermeiden.

Anforderung 11: Verbrauch berechnen und Anzeigen

Vorher: Es war unklar, ob historische Verbrauchswerte gespeichert oder angezeigt werden können.

Nachher: Historische Verbrauchswerte der letzten 12 Monate sind abrufbar. Grafische Darstellung und Export möglich.

Begründung: Die Anforderungen an die Verbrauchsanzeige wurden konkretisiert und die Möglichkeit eines Exports ergänzt.

Mit dieser Überarbeitung haben wir die Anforderungen an die Hausverwaltung strukturiert, verständlich und technisch realisierbar formuliert. Die neue Version stellt sicher, dass alle wichtigen Funktionalitäten klar definiert sind, notwendige Validierungsregeln integriert wurden und das System sowohl für Nutzer als auch für

Administratoren effizient funktioniert. Durch die Analyse der ursprünglichen Anforderungen und die darauf basierenden Verbesserungen wurde eine solide Grundlage für die Entwicklung des Prototyps geschaffen. Diese Anforderungen dienen als Leitfaden für die Implementierung und ermöglichen eine zielgerichtete und fehlerfreie Umsetzung.

IV Testkonzept

IV.1 Einleitung

Durch dieses Testkonzept haben wir versucht die grundlegenden Testverfahren zu beschreiben, die zur Überprüfung der Kernfunktionalitäten unseres Prototyps verwendet werden.

Da es sich lediglich um ein kleines Projekt handelt, liegt der Fokus bei uns auf die technischen Tests zur Funktionsprüfung, anstatt systemweit deckende Funktionalitäten oder System- oder Usability-Tests.

Unser Ziel ist es, die wichtigsten Funktionen zu validieren, um eine fehlerfreie und konsistente Prototyp-Umsetzung sicherzustellen.

IV.2 Testziele und Strategie

Testziele

- Sicherstellen, dass die Kernfunktionen korrekt arbeiten
- Prüfen, ob Module korrekt interagieren
- Fehlermeldungen und ungültige Eingaben testen

Teststrategie

- Zuerst einzelne Komponenten testen (Unit-Tests)
- Danach prüfen, ob die Module zusammenarbeiten (Integrationstests)
- Überprüfung der Systemfunktionen (Funktionstests)
- Bewusst falsche Eingaben ausprobieren (Negative Tests)

IV.3 Ausgewählte Testverfahren und Begründung

Tabelle IV.1: Ausgewählte Testverfahren

Testverfahren	Einsatzbereich	Begründung
Unit-Tests	Einzelne Funktionen wie Datenvalidierung, ID-Format, Speicherung von Ablesewerten	Frühes Erkennen von Fehlern in einzelnen Modulen
Integrationstests	Zusammenspiel der Module, z. B. Verknüpfung von Zähler, Wohnung und Gebäude	Sicherstellen, dass die Module korrekt miteinander arbeiten
Funktionstests	Überprüfung der gesamten Funktionalität wie Zählerverwaltung, Ablesungen, Filterung	Verifizierung der implementierten Anforderungen
Negative Tests	Eingabe ungültiger Werte (z. B. leere Felder, falsche ID, negatives Datum)	Sicherstellen, dass das System Fehlersituationen richtig behandelt

IV.4 Testumgebung und Testfälle

Testumgebung

- Der Prototyp wird in einer lokalen Entwicklungsumgebung getestet.
- Es wird eine Testdatenbank mit Dummy-Daten erstellt.

Wichtige Testfälle

Tabelle IV.2: relevante Testfälle

Testfall	Erwartetes Ergebnis
Zähler-ID existiert nicht	Fehlermeldung: „Die eingegebene ID existiert nicht.“
Ablesewert ist negativ (-10)	Fehlermeldung: „Ungültiger Ablesewert.“
Eingabe eines zu langen Zähler-Codes	Fehlermeldung: „Zähler-ID muss 10 Zeichen haben.“
Korrekte ID eingeben	Zähler wird erfolgreich gefunden
Eingabe einer gültigen Ablesung	Wert wird korrekt gespeichert

IV.5 Fazit

Mit diesem Testkonzept wollen wir sicherstellen, dass die wichtigsten Funktionen des Prototyps getestet werden, ohne unnötig viel Zeit in realistische (wir meinen hier eine produktive Umgebung.)

oder nicht notwendige Tests zu investieren. Die Kombination aus Unit-Tests, Integrationstests, Funktionstests und Negative Tests reicht aus, um die Qualität und Stabilität des Prototyps sicherzustellen.

V Konkrete Testfälle für die Hausverwaltungssoftware

Die folgenden Testfälle überprüfen die wichtigsten Funktionen des Prototyps. Dabei werden **Unit-Tests, Integrationstests, Funktionstests und Negative Tests** berücksichtigt.

Tabelle V.1: Testfälle für die Hausverwaltungssoftware

Test-ID	Beschreibung	Eingabe	Erwartetes Ergebnis	Testtyp
TC-001	Zähler-ID existiert nicht	‘999-9999-9999‘	Fehlermeldung: *„Die eingegebene ID existiert nicht.“*	Negative Test
TC-002	Gültige Zähler-ID eingeben	‘123-2024-4567‘	Zählerdetails werden angezeigt	Funktionstest
TC-003	Ablesewert negativ	‘-10‘ als Ablesewert	Fehlermeldung: *„Ungültiger Ablesewert.“*	Negative Test
TC-004	Ablesewert kleiner als vorheriger Wert	Neuer Wert: ‘50‘, alter Wert: ‘100‘	Fehlermeldung: *„Neuer Wert muss größer sein als der vorherige.“*	Negative Test
TC-005	Korrekte Ablesung speichern	Neuer Wert: ‘250‘	Wert wird korrekt gespeichert	Funktionstest
TC-006	Eingabe einer zu langen Zähler-ID	‘123-2024-45678‘ (11 Zeichen)	Fehlermeldung: *„Zähler-ID muss genau 10 Zeichen haben.“*	Negative Test
TC-007	Filtern nach Gebäude und Zählertyp	Gebäude: ‘Haus A‘, Zählertyp: ‘Strom‘	Liste zeigt nur Stromzähler von ‘Haus A‘	Integrationstest
TC-008	Ablesedatum in der Zukunft	Datum: ‘01.01.2030‘	Wert wird gespeichert	Funktionstest
TC-009	Ablesedatum rückdatiert	Datum: ‘01.01.2000‘	Fehlermeldung: *„Datum darf nicht in der Vergangenheit liegen.“*	Negative Test
TC-010	Standard-Ableser bei fehlender Eingabe	Ableser nicht eingetragen	Standardwert „Unbekannt“ wird gespeichert	Funktionstest
TC-011	Historische Verbrauchswerte anzeigen	Monat: ‘Januar‘	Diagramm zeigt Verbrauchswerte für Januar	Funktionstest
TC-012	Suchfunktion mit Teilstring	Eingabe: ‘123‘	Zeigt alle Zähler mit ‘123‘ in der ID	Integrationstest

VI Prototyp-Dokumentation

Passend zu diesem Template existieren noch zwei weitere Vorlagen für Poster. Diese unterscheiden sich lediglich durch die Ausrichtung der Seite und sind ansonsten identisch. Der Aufbau der Dateien und Projektstruktur ist weitestgehend identisch zu dieser Vorlage, es gilt lediglich auf einige kleine Besonderheiten zu verweisen.

In der sog. `multicols` Umgebung ist es nicht möglich, mit Floatumgebungen zu arbeiten, dies ist für fast alle Beispiele aus dieser Anleitung irrelevant, lediglich Grafiken müssen anders eingebunden werden. Dazu finden sich jedoch Beispiele in den Vorlagen. Ebenso ist ein Teil der Präambel in eine separate Datei ausgelagert, in dieser sollten auch nicht ohne weiteres Änderungen vorgenommen werden. Wie in diesem Template, müssen innerhalb der Präambel Zitierstil, sowie Autor:innen und Titel gesetzt werden.

Die Vorlagen können auf meiner Webseite sowohl angeschaut, als auch heruntergeladen werden, alle Hinweise zum Kompilieren gelten für die Poster ebenso. Ergebnis dieser Vorlage ist z. B. folgendes Poster:

Selbstständigkeitserklärung

Wir versichern, die von uns vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die wir für die Arbeit benutzt haben, sind angegeben. Die Arbeit haben wir mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegt.

Bremerhaven, den 20. Februar 2025

Unterschrift:

Junior Leage EKane Njoh, Franck Majeste
Silatsa Dogmo, Steve Aguiwo II