

# Hochschule Bremerhaven

Fachbereich II  
Management und Informationssysteme  
Wirtschaftsinformatik B.Sc.

Modul  
Qualitätsmanagement

---

## Semesteraufgabe

Entwicklung einer Hausverwaltung

---

<b>Vorgelegt von:</b>	Junior Lesage Ekane Njoh	MatNr. 40128
	Steve Aguiwo II	MatNr. 40088
	Franck Majeste Dogmo Silatsa	MatNr. 00000
<b>Vorgelegt am:</b>	1. März 2025	
<b>Dozent:in:</b>	Prof. Dr. Karin Vosseberg	

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Anforderungsanalyse</b>	<b>5</b>
2.1	Review der Anforderungen	5
2.2	Verbesserung der Anforderungen	5
<b>3</b>	<b>Testkonzept</b>	<b>5</b>
3.1	Auswahl von Testverfahren	5
3.2	Teststufen und Testarten	5
3.3	Testumgebung und Testdaten	5
<b>4</b>	<b>Entwicklung der Testfälle</b>	<b>5</b>
4.1	Ableitung konkreter Testfälle	5
4.2	Erstellung einer Testfall-Dokumentation	5
<b>5</b>	<b>Prototypische Umsetzung der Hausverwaltung</b>	<b>5</b>
5.1	Software-Architektur und Technologien	5
5.2	Implentierung	5
5.3	Anwendung des Testkonzepts	5
<b>6</b>	<b>Qualitätsmanagement-Methoden in der Softwareentwicklung</b>	<b>5</b>
6.1	Relevanz der Qualitätssicherung	5
6.2	Anwendung von QS-Methoden im Projekt	5
<b>7</b>	<b>Einleitung</b>	<b>5</b>
	<b>Literaturverzeichnis</b>	<b>6</b>
	<b>Listingverzeichnis</b>	<b>6</b>
	<b>Anhang</b>	<b>7</b>
I	Review-Protokoll der Anforderungen an die Hausverwaltung	8
II	Verbesserte Anforderungen auf Review-Basis	11
III	Testkonzept	14
III.1	Einleitung	14
III.2	Testziele und Strategie	14
III.3	Ausgewählte Testverfahren und Begründung	15
III.4	Testumgebung und Testfälle	15
III.5	Fazit	15
IV	Konkrete Testfälle für die Hausverwaltungssoftware	16

<b>Selbstständigkeitserklärung . . . . .</b>	<b>18</b>
--	-----------

# 1 Einleitung

Die Verwaltung von Gebäuden und deren Energieverbrauch stellt in der Praxis eine zentrale Herausforderung dar. Insbesondere in Mehrfamilienhäusern oder Wohnanlagen ist eine effiziente und übersichtliche Erfassung von Zählerständen erforderlich, um Verbrauchsdaten transparent zu machen und eine gerechte Abrechnung zu ermöglichen. Im Rahmen dieses Projekts entwickeln wir als drei Studierende einen Prototyp für eine Hausverwaltungssoftware, die sich auf die digitale Erfassung, Verwaltung und Analyse von Zählerständen konzentriert.

Die Umsetzung erfolgt als webbasierte Anwendung, wobei der Fokus auf einer intuitiven Benutzeroberfläche und einer zuverlässigen Datenverarbeitung liegt. Die Hausverwaltung ermöglicht es, Gebäude, Zähler und Verbrauchsdaten zu verwalten, Zählerablesungen zu dokumentieren und historische Verbrauchswerte grafisch darzustellen. Dabei werden sowohl technische als auch organisatorische Aspekte berücksichtigt, um eine realitätsnahe und funktionale Lösung zu entwickeln.

Ein wesentlicher Bestandteil des Projekts ist das Review der Anforderungen sowie die Entwicklung eines fundierten Testkonzepts, um sicherzustellen, dass der Prototyp stabil, fehlerresistent und effizient arbeitet. Im Rahmen unserer Ausarbeitung dokumentieren wir die einzelnen Projektschritte detailliert und analysieren die gewonnenen Erkenntnisse. Unser Ziel ist es, ein möglichst praxisnahes und gut strukturiertes System zu entwerfen, das die wesentlichen Funktionen einer Hausverwaltung abbildet.

Die Entwicklung des Prototyps folgt einem iterativen Ansatz, bei dem wir zunächst die Anforderungen überprüft und überarbeitet haben, um Widersprüche oder Unklarheiten zu beseitigen. Anschließend wurden konkrete Testfälle definiert, um die Kernfunktionen zu validieren. Die Tests umfassen funktionale Prüfungen, negative Tests sowie Leistungstests, um sowohl korrekte Funktionalität als auch Systemgrenzen zu ermitteln. Schließlich wurde der Prototyp entsprechend der definierten Anforderungen und Testfälle umgesetzt.



## **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. Hier haben wir eine Unterscheidung zwischen funktionalen und nicht funktionalen Anforderungen gemacht.

### **Funktionale Anforderungen**

Tabelle II.1: Funktionale Anforderungen

Nr.	Anforderung	Beschreibung
F1	Gebäudestruktur verwalten	Gebäude können mehrere Eingänge haben, jede Wohnung hat eine eindeutige ID.
F2	Zählertypen verwalten	Unterstützte Typen: Strom, Gas, Wasser. Die Liste ist erweiterbar, indem neue Typen über eine Konfigurationsdatei hinzugefügt werden.
F3	Zählerverwaltung	Jeder Zähler hat eine eindeutige ID (Gebäude-Jahr-Nummer). Jeder Zähler gehört zu einer Wohnung und einem Zählertyp. Er speichert den letzten Ablesewert, das letzte Ablesedatum und die Ablesemethode.
F4	Datenfilterung	Filter nach Gebäude, Wohnung, Zählertyp und Zeitraum.
F5	Zählerablesung	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.
F6	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.“
F7	Verbrauchsanzeige	Historische Verbrauchswerte sind für die letzten 12 Monate abrufbar. Eine grafische Darstellung ist möglich.
F8	Ableser-Informationen	Ableser können Hauswart, Mieter oder Energieversorger sein. Falls keine Information vorhanden ist, wird „Unbekannt“ eingetragen.
F9	Bearbeiten und Löschen von Gebäuden	Gebäude können direkt bearbeitet oder gelöscht werden.
F10	Zurück-Buttons auf allen Seiten	Verbesserte Navigation in der Anwendung.
F11	Gebäude auswählen vor Verbrauchsanzeige	Nutzer müssen erst ein Gebäude wählen, bevor Verbrauchsdaten angezeigt werden.
F12	Direkte Weiterleitung bei nur einem Gebäude	Wenn nur ein Gebäude existiert, wird die Verbrauchsanzeige sofort geladen.
F13	Unterschiedliche Speicherung für aktuelle	historische Verbrauchsdaten: $aktuell_X.png$ und $historie_{XY}YYY-MM-DD.png$ werden getrennt gespeichert.

**Nicht-funktionale Anforderungen**

Tabelle II.2: Nicht-Funktionale Anforderungen

<b>Nr.</b>	<b>Anforderung</b>	<b>Beschreibung</b>
NF1	Zeitraum für die Verbrauchsanzeige im Diagramm sichtbar	Das Diagramm zeigt den Zeitraum der Messung an (z. B. „März 2024 - Februar 2025“).
NF2	Letzte 12 Monate immer anzeigen (auch ohne Werte)	Die Verbrauchsanzeige berücksichtigt automatisch die letzten 12 Monate. Fehlende Werte werden als „0“ dargestellt.
NF3	Farbliche Kennzeichnung der Zähler in der Verbrauchsanzeige	Jeder Zähler erhält eine eindeutige Farbe zur besseren Unterscheidung.
NF4	Optimierung der Antwortzeiten	Das System soll Verbrauchsdaten in unter 2 Sekunden berechnen und anzeigen.
NF5	Datenintegrität und Konsistenz	Ablesewerte dürfen nicht rückwirkend geändert werden (außer durch Admins).
NF6	Speicherung von Verbrauchsdaten gemäß Datenschutzbestimmungen	Verbrauchsdaten dürfen nur von autorisierten Nutzern eingesehen werden.
NF7	System skalierbar für große Datenmengen	Unterstützung für mindestens 100 Gebäude und 5000 Zähler.

## III Testkonzept

### III.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.

### III.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)

### III.3 Ausgewählte Testverfahren und Begründung

Tabelle III.1: Ausgewählte Testverfahren

Testverfahren	Einsatzbereich	Begründung
Unit-Tests	Einzelne Funktionen wie Datenvalidierung, ID-Format, Speicherung von Ablesewerten	Frühes Erkennen von Fehlern in einzelnen Modulen
Funktionstests	Überprüfung der gesamten Funktionalität wie Zählerverwaltung, Ablesungen, Filterung	Verifizierung der implementierten Anforderungen
Performance-Tests	Messung der Ladezeiten der Verbrauchsanzeige	Sicherstellen, dass das System auch mit vielen Gebäuden/Zählern performant bleibt
Negative Tests	Eingabe ungültiger Werte (z. B. leere Felder, falsche ID, negatives Datum)	Sicherstellen, dass das System Fehlersituationen richtig behandelt

### III.4 Testumgebung und Testfälle

#### Testumgebung

- Der Prototyp wird in einer lokalen Entwicklungsumgebung getestet.
- Es wird eine Testdatenbank mit Dummy-Daten erstellt. Die Persistenz der Daten wird durch die Nutzung von json-Datei-Format gewährleistet, da es sich bei uns um eine Testumgebung für einen Prototyp.

### III.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.

## **IV Konkrete Testfälle für die Hausverwaltungssoftware**

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



#### IV Konkrete Testfälle für die Hausverwaltungssoftware

Tabelle IV.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	Negativer Ablesewert	‘-10‘	Fehlermeldung: „Ungültiger Ablesewert“	Negative Test
TC-003	Zählerlänge	‘1-2024-4567823‘	Fehlermeldung: „Zähler-ID muss genau 14 Zeichen haben!“	Negative Test
TC-004	Ablesedatum rückdatiert	‘2000-01-01‘	Fehlermeldung: „Datum darf nicht in der Vergangenheit liegen!“	Negative Test
TC-005	Ablesewert kleiner als vorheriger Wert	Neuer Wert: ‘50‘, alter Wert: ‘100‘	Fehlermeldung: „Neuer Wert muss größer sein als der vorherige.“	Negative Test
TC-006	Gültige Zähler-ID über die Suchfunktion eingeben	‘1-2025-5487‘	Zählerdetails werden angezeigt	Funktionstest
TC-007	Korrekte Ablesung speichern	Alter Wert: 100, Neuer Wert: ‘250‘	Wert wird korrekt gespeichert	Funktionstest
TC-008	Ablesedatum in der Zukunft	Datum: ‘01.01.2030‘	Wert wird gespeichert	Funktionstest
TC-009	Standard-Ableser bei fehlender Eingabe	Ableser nicht eingetragen	Standardwert „Unbekannt“ wird gespeichert	Funktionstest
TC-010	Historische Verbrauchswerte anzeigen	Es wird die Schnittstelle für Verbrauchshistorie mit der Gebäude-ID “1” abgerufen	Ablesungen sollen als Liste zurückgegeben werden oder als Grafik in der Weboberfläche	Funktionstest
TC-011	Suchfunktion mit Teilstring	Eingabe: ‘123‘	Zeigt alle Zähler mit ‘123‘ in der ID	Funktionstests
TC-012	Massive Ablesungen	10000 Ablesungen	Es werden alle Ablesungen in maximal 60 Sekunden gespeichert und keine Daten gehen verloren	Performance-Tests
TC-013	Antwortzeit-Test	Es wird die index-Seite aufgerufen	Innerhalb von wenigen Millisekunden eine Antwort geliefert	Performance-Test
TC-014	Massive Zählererstellung	10000 Strom-Zähler	Alle Zähler werden hinzugefügt ohne zu lange Wartezeit	Performance-Test
TC-015	Massive Gebäudeerstellung	10000 Gebäude	Alle Gebäude werden hinzugefügt ohne zu lange Wartezeit	Performance-Test

## **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 1. März 2025

Unterschrift:

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