

Immobilien- und Mietverwaltung der Hausbau GbR

Projekt Tenzur

1. Allgemeines.....	1
1.1. Historie der Dokumentenversionen.....	1
1.2. Ziel dieses Dokuments / Projektbezug.....	1
2. Produktkonzept und Rahmenbedingungen.....	1
2.1. Vision und Produktziel.....	1
2.2. Allgemeine Produktbeschreibung.....	2
2.3. Ziele des Produkts.....	2
2.4. Benutzer und Zielgruppe.....	3
2.5. Systemvoraussetzungen / Betriebsbedingungen.....	3
3. Konkretisierung der Anforderungen.....	3
3.1. Pflichtanforderungen.....	3
3.2. Optionale Anforderungen.....	4
3.3. Anforderungen an die Benutzerfreundlichkeit.....	4
3.4. Leistungsanforderungen.....	5
4. Projektorganisation.....	5
4.1. Projektteam und Verantwortlichkeiten.....	5
4.2. Vorgehensmodell.....	5
4.3. Projektmanagement.....	6
5. Qualitätssicherung.....	6
5.1. Testfälle für Pflichtanforderungen.....	6
5.2. Testfälle für optionale Anforderungen.....	7
6. Anhang / Ressourcen.....	7
6.1. UML (Unified Modeling Language).....	7
6.2. ERM (Entity Relationship Model).....	8
6.3. Frontend Mockup.....	8

1. Allgemeines

1.1. Historie der Dokumentenversionen

Version 1.0 (28.08.2023)	Erstellung des ersten Entwurfs mit grobem Aufbau.
Version 1.1 (29.08.2023)	Zieldefinition und Konzeptionierung des Vorgehensmodells.
Version 1.2 (30.08.2023)	Erster Entwurf des Vorgehensmodells erstellt inklusive Diagramm zum Ablauf. Feste Definitionen der Phasen.
Version 1.3 (31.08.2023)	Optimierung der Datenbankmodelle und Abnahme des Vorgehensmodells des Teams.
Version 1.4 (01.09.2023)	Finalisierung der Inhalte und Formatieren des Dokumentes.

1.2. Ziel dieses Dokuments / Projektbezug

Das vorliegende Pflichtenheft (System Specifications, nachfolgend referenziert als **SYS**) dient der Übersicht der Vision der Hausbau GbR bezüglich der Verwaltung von Immobilien und Mietern. Es beschreibt und erläutert die gewünschten Funktionalitäten des Produktes. Hierbei bezieht sich dieses Pflichtenheft auf die genannten Spezifikationen des Lastenheftes (Customer Specifications, im nachfolgenden referenziert als **CS**). Ziel des Projektes ist die Erstellung einer skalierbaren, benutzerfreundlichen und leistungs effizienten Software. Jegliche Anhänge und Ressourcen befinden sich am Ende des Dokumentes (innerhalb des Dokumentverlaufes referenziert als **AR**).

2. Produktkonzept und Rahmenbedingungen

2.1. Vision und Produktziel

Das Ziel des Auftraggebers ist, die Vermietungs- und Verwaltungsabläufe im Immobilienbereich zu digitalisieren und zu automatisieren. Das Ziel dieses Projektes ist dementsprechend die Erstellung einer geeigneten Lösung in Form einer Software unter Berücksichtigung von gesetzlichen Maßnahmen und technischen Rahmenbedingungen und Best-Practices.

2.2. Allgemeine Produktbeschreibung

Die geplante Web-Anwendung ermöglicht den Mitarbeitern von BauHaus GbR die effektive Verwaltung von Immobilienobjekten und Mietern, welche durch eine grafische Oberfläche Immobilien und Mieterdaten erfasst, bearbeitet und gelöscht werden können. Finanzielle Aspekte wie Nebenkosten, Zahlungen und Außenstände werden systematisch verwaltet. Die Dokumentenverwaltung unterstützt das Speichern und Organisieren von Dateien in verschiedenen Formaten. Die Anwendung wird in Java gemäß den Prinzipien der objektorientierten Programmierung entwickelt und nutzt PostgreSQL als zuverlässige Datenbank. Dieses Gesamtpaket bietet eine leistungsstarke Lösung zur Optimierung der Verwaltungsprozesse für Immobilien und Mieter.

2.3. Ziele des Produkts

Z1	Verwaltung von Immobilienobjekten
Z2	Verwaltung von Mietern (Objekt Mieter)
Z3	Verwaltung und Nebenkosten, Zahlungen und Außenständen
Z4	Dokumentenmanagement
Z5	Benutzerverwaltung
Z6	Dashboard Übersicht
Z7	Sortier-, Filter- und Suchfunktionen

Optionale Ziele							
Z8	Arbeitszeitdokumentation der Mitarbeiter						
Z9	Nachrichten-, Termin- und Notizfunktion						
Z10	Erstellung und Druck des Reports:						
	<table><tr><td>Z10.1</td><td>Objektbericht mit Objektedaten, Objektmieterbezug sowie Einnahme- bzw. Fehlbeträge für einen Zeitraum</td></tr><tr><td>Z10.2</td><td>Gesamtbericht aller Mietobjekte mit Ist- und Soll-Zahlungen (inkl. Fehlbeträgen) für einen Zeitraum</td></tr><tr><td>Z10.3</td><td>Übersicht über die geleisteten Arbeitsstunden von Mitarbeitern für einen Zeitraum.</td></tr></table>	Z10.1	Objektbericht mit Objektedaten, Objektmieterbezug sowie Einnahme- bzw. Fehlbeträge für einen Zeitraum	Z10.2	Gesamtbericht aller Mietobjekte mit Ist- und Soll-Zahlungen (inkl. Fehlbeträgen) für einen Zeitraum	Z10.3	Übersicht über die geleisteten Arbeitsstunden von Mitarbeitern für einen Zeitraum.
	Z10.1	Objektbericht mit Objektedaten, Objektmieterbezug sowie Einnahme- bzw. Fehlbeträge für einen Zeitraum					
	Z10.2	Gesamtbericht aller Mietobjekte mit Ist- und Soll-Zahlungen (inkl. Fehlbeträgen) für einen Zeitraum					
Z10.3	Übersicht über die geleisteten Arbeitsstunden von Mitarbeitern für einen Zeitraum.						
Z11	Import und Export Funktionen von Stammdaten						

2.4. Benutzer und Zielgruppe

2.4.1. Zielgruppe ist die HausBau GbR.

2.4.2. Benutzer der Software werden die Voll- und Teilzeitkräfte der HausBau GbR, Vermieter und Mieter sein.

2.5. Systemvoraussetzungen / Betriebsbedingungen

Datenbank		
	PostgreSQL	Version 15
Java	Version 17	
Maven	Version 3	
IDE	Netbeans	

3. Konkretisierung der Anforderungen

3.1. Pflichtenanforderungen

Anforderung ist, eine zeitgemäße, zentrale und komfortable Softwarelösung zur Verwaltung zu entwickeln.

SRF1	SRF1.1	Das System hat eine fest definierte Administrator-Gruppe welche nicht veränderbar ist, standardmäßig alle Berechtigungen besitzt und beim Deployment an einen User vergeben wird, welcher als Administrator in der Firma agieren wird.
	SRF1.2	Die Administrator-Gruppe hat Zugriff auf alle zu implementierenden Programm-Funktionalitäten inklusive der Benutzerverwaltung und Einsicht auf Arbeitszeiten, Termine, Notizen etc. .
	SRF1.3	Ein komplettes ERM (Entity Relation Model) und UML (Unified Modeling Language) ist zu finden im Anhang AR.6.1 und AR.6.2 unterhalb des Dokumentes
	SRF1.4	Normale Benutzernamen sind einzigartig, müssen mindestens 5 Zeichen (keine Leerzeichen) lang sein und dürfen nicht doppelt in der Datenbank vorkommen.
	SRF1.5	Benutzer Passwörter müssen mindestens 12 Zeichen lang sein, kann jegliche Zeichen enthalten und muss mindestens 3 Sonderzeichen und/oder Nummern enthalten.

SRF2	Der Zugang zu Programm Funktionalitäten ist zu beschränken.	
	SRF2.1	Beim Start der Anwendung erscheint ein Login-Fenster mit den Möglichkeiten, seine Login-Daten einzufügen, um sich anzumelden. Weitere Funktionalitäten sind nicht erreichbar.
	SRF2.2	Nach dem Login, wird das Hauptfenster der Anwendung geladen. Über die Menüpunkte und Buttons werden dem Anwender nur die zum Rollenprofil passenden Funktionalitäten angeboten.

3.2. Optionale Anforderungen

SRO1	Erstellung und Druck von Reports wird mithilfe eines Frameworks (JasperReports) implementiert.
SRO2	Die Stammdaten werden über das UI an den User ausgegeben im <code>.xlsx</code> oder <code>.csv</code> Format, je nachdem welches Format der User ausgewählt hat. Das gleiche gilt für den Import der Stammdaten. Die Import- und Export Funktion befindet sich bei der Verwaltung der Mieter.

3.3. Anforderungen an die Benutzerfreundlichkeit

SRB	Um die Benutzerfreundlichkeit für neue Nutzer zu garantieren, wird ein simples und auf den ersten Blick verständliches User-Interface gebaut, welches die Hauptfunktionen in einer linksbündigen "Navbar" (Navigationsleiste) präsentiert. Durch auswählen einer dieser Optionen ändert sich das "Content-Fenster" (Bereich rechts neben der Navbar) um die Ansicht der gewählten Funktion grafisch darzustellen.
------------	---

3.4. Leistungsanforderungen

LA1	Gemäß CS.5.1.RPQ2 wird für die Anwendung eine ausführbare JAR-Datei bereitgestellt.
LA2	In Bezug auf CS.5.1.RPQ1 erfolgt die Abgabe als gesamter NetBeans-Arbeitsbereich, inklusive der verwendeten Bibliotheken.
LA3	Es wird ein Datenbank-Dump mit geeigneten Testdaten gemäß CS.5.1.RPQ3 zur Verfügung gestellt.
LA4	Es wird eine Dokumentation zur Handhabung der erstellten Software, wie in CS.5.2.1.RDA1 erwähnt, erarbeitet.
LA5	Gemäß CS.5.2.2.RDT1-5 wird eine ausführliche technische Dokumentation in Form eines ER-Modells und Ableitung in ein relationales Schema, Dokumentation mittels JavaDoc und des Quellcodes und eine Anleitung zur Installation und Konfiguration der Software.

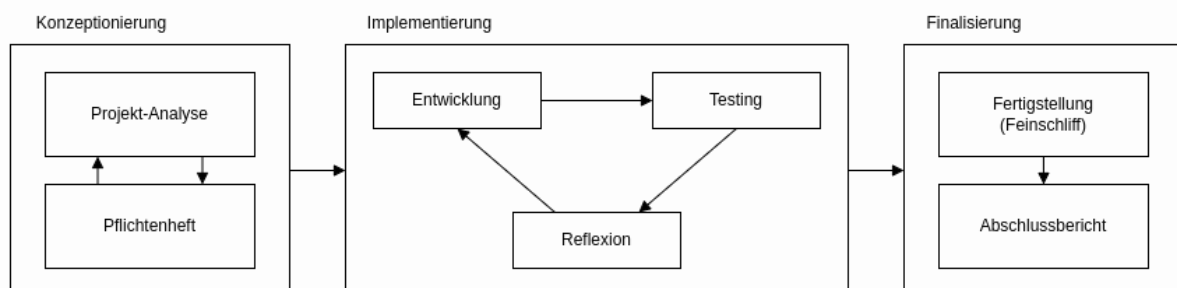
4. Projektorganisation

4.1. Projektteam und Verantwortlichkeiten

Projektleiter	Tobias Ochott	Backend
Stv. Projektleiter	Lars Belitz	Backend
Projektteam	Laura Boniecki Burak Kablan Marwin Parchainski Robin Walter Nils Rüttgers	Datenbank Datenbank Frontend Frontend n/a

4.2. Vorgehensmodell

Für eine tägliche Übersicht der Tasks und gegebenenfalls entstehenden Fehler wird eine vereinfachte Version des Scrum-Modells verwendet. Hierbei wird sowohl auf den Product Owner als auch auf die Sprints verzichtet.



- **Konzeptionierung**

Es werden die zu realisierenden Tasks identifiziert, evaluiert und konzeptioniert. Hierbei wird iterativ vorgegangen und die Aufgaben in Bezug auf Kapitel 2 des Lastenheftes in das Entwicklungsteam aufgeteilt. Dabei wird durch die Verständigung innerhalb des Teams ein konstanter Feedback Loop aufgebaut, um die Funktionalitäten im Konzept zu optimieren.

- **Implementierung**

Während der Implementierungsphase werden in einem täglichen Intervall die zu realisierenden Funktionen besprochen, zugeteilt und entwickelt. Hierbei werden auch im Backlog vorhandene Tasks gemäß der vorher definierten Priorisierung und Größe berücksichtigt und mit erhöhter Priorität bearbeitet. Parallel wird das Produkt in Bezug auf 4.2 der Qualitätssicherung getestet.

- **Finalisierung**

Das Produkt wird dem Kunden vorgestellt und mitsamt der technischen Dokumentation übergeben.

4.3. Projektmanagement

Um die Planung und Durchführung des Projektes simpel und übersichtlich zu gestalten, nutzt das Team GitHub-Project für eine Übersicht und Statusanzeige der Tasks. Das Board ermöglicht dem Entwickler Team sämtliche Aufgaben und deren Status zu sehen und mit dem restlichen Team Absprache bezüglich Fehlern und Dringlichkeit zu halten. Für den Deployment Prozess wird ein in zwei Branches geteiltes GitHub Repository verwendet. Um die Stabilität des Endproduktes zu gewährleisten, wird ein `live` Branch verwendet, auf dem nur über den `dev` Branch gemerged werden kann. Dies garantiert, dass vorher auf dem `dev` Branch alle definierten und situationsabhängigen Nutzerbedingten Tests erfolgreich durchliefen mussten, sodass Fehler im Endprodukt ausgeschlossen werden.

5. Qualitätssicherung

5.1. Testfälle für Pflichtenforderungen

QS1	QS1.1	Alle Nutzer können sich einloggen und die grafische Oberfläche nach ihren festgelegten Rollen bedienen
	QS1.2	Administrator kann Benutzer anlegen und Rollen verwalten
	QS1.3	Mietobjekte, Mieter können angelegt, bearbeitet und gelöscht werden
	QS1.3.1	Prozess kann manuell oder über Import/Export ausgeführt werden

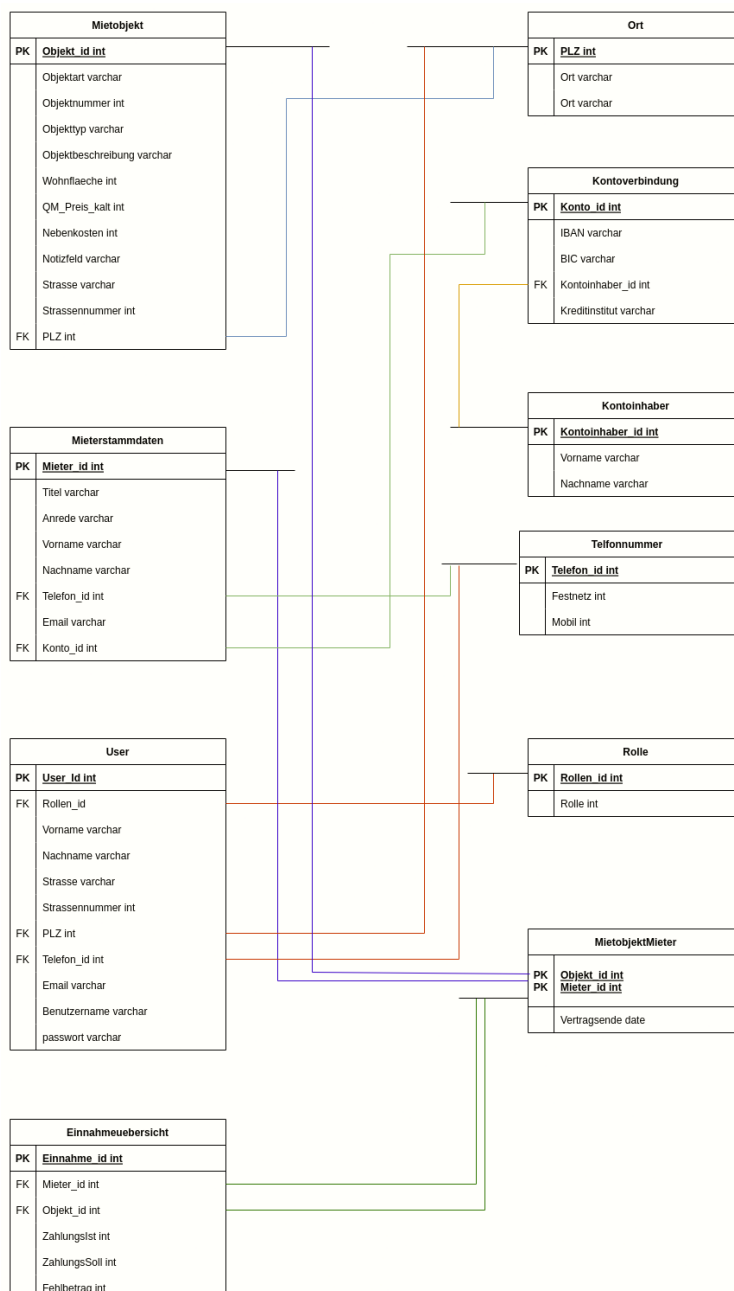
5.2. Testfälle für optionale Anforderungen

QS2

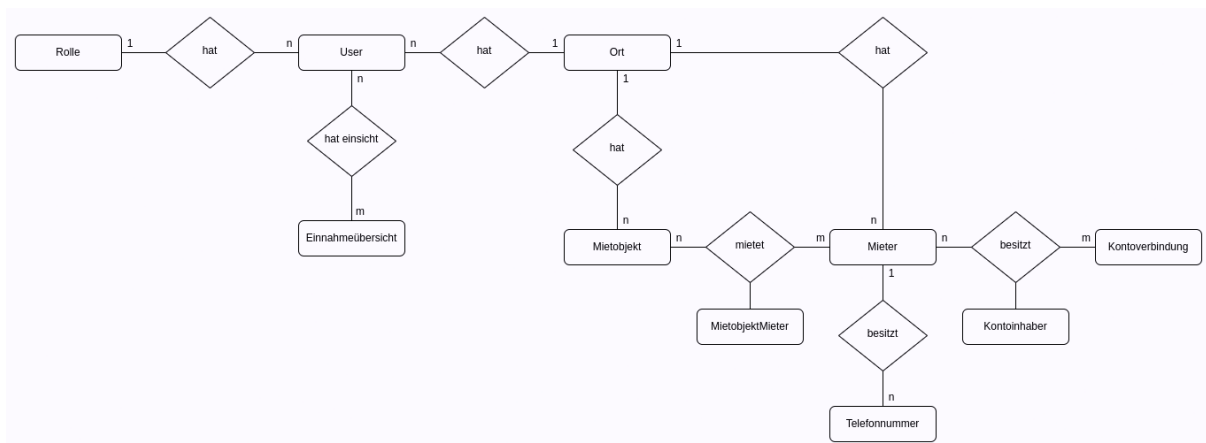
QS2.1	Reports werden korrekt erstellt und gedruckt
QS2.2	Exporte der Stammdaten beinhalten alle Daten unter den richtigen Bezeichnung

6. Anhang / Ressourcen

6.1. UML (Unified Modeling Language)

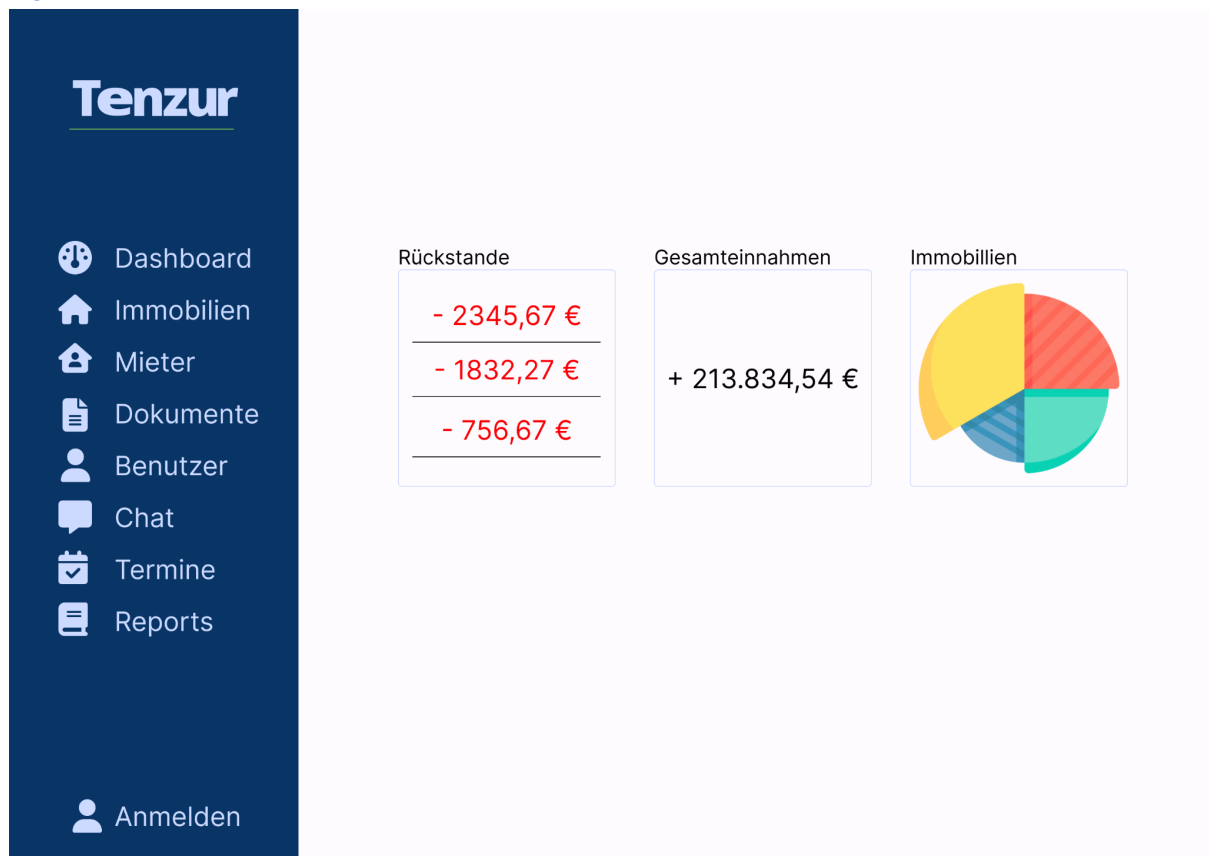


6.2. ERM (Entity Relationship Model)



6.3. Frontend Mockup

[Figma Interaktives Mockup](#)



[+ Hinzufügen](#)

ID	Vorname	Nachname	Email
1	Max	Muster	mm@mm.de
1	Max	Muster	mm@mm.de
1	Max	Muster	mm@mm.de
1	Max	Muster	mm@mm.de
1	Max	Muster	mm@mm.de
1	Max	Muster	mm@mm.de

[+ Hinzufügen](#)

ID	Dateiname	Dateityp	Größe
1	examplefile	png	1,2 mb
1	examplefile	png	1,2 mb
1	examplefile	png	1,2 mb
1	examplefile	png	1,2 mb
1	examplefile	png	1,2 mb
1	examplefile	png	1,2 mb

[+ Hinzufügen](#)

ID	Benutzername	Nachname	Email
1	xXxMartiPrincexXx	Prinz	Martin.Prinz@...
1	xXxMartiPrincexXx	Prinz	Martin.Prinz@...
1	xXxMartiPrincexXx	Prinz	Martin.Prinz@...
1	xXxMartiPrincexXx	Prinz	Martin.Prinz@...
1	xXxMartiPrincexXx	Prinz	Martin.Prinz@...
1	xXxMartiPrincexXx	Prinz	Martin.Prinz@...

[+ Hinzufügen](#)

ID	Objektnummer	Anschrift	Objekttyp
1	11111	Kö	Privat
1	11111	Kö	Privat
1	11111	Kö	Privat
1	11111	Kö	Privat
1	11111	Kö	Privat
1	11111	Kö	Privat