

FT (13): Notizverwaltung

Ziel / Zweck

Dieses Feature ermöglicht die Verwaltung von Notizen als eigenständige Domainobjekte, die sowohl Projekten als auch Kunden zugeordnet werden können. Notizen dienen der Dokumentation zusätzlicher Informationen, Hinweise oder Besonderheiten, die im Kontext eines Projekts oder Kunden relevant sind.

Fachliche Beschreibung

Notizen sind eigenständige Textobjekte mit Titel, formatierbarer Beschreibung und Zeitstempeln für Erstellung und letzte Bearbeitung. Sie werden über Relationstabellen entweder Projekten oder Kunden zugeordnet und ermöglichen eine flexible Dokumentation ohne strukturelle Abhängigkeiten. Eine Notiz gehört immer genau einem Parent-Objekt (Projekt oder Kunde). Eine Notiz existiert nie unabhängig - sie wird immer im Kontext ihres Parents erstellt, verwaltet und gelöscht.

Notizen werden in den Detailansichten von Projekt und Kunde als vertikale Kärtchenliste dargestellt. Die Bearbeitung erfolgt über einen schwebenden Richtext-Editor, der Textformatierung sowie Text- und Hintergrundfarben unterstützt.

Notizen haben keine fachliche Wirkung auf Termine, Status oder Planungslogik. Sie dienen ausschließlich der Information und Dokumentation. Das Löschen einer Notiz erfolgt direkt über die Detailansicht des zugehörigen Parents und ist endgültig.

Regeln & Randbedingungen

- Eine Notiz ist ein eigenständiges Domainobjekt mit eigener ID.
- Eine Notiz gehört immer genau einem Parent-Objekt (Projekt oder Kunde).
- Eine Notiz kann nie ohne Parent-Zuordnung existieren.
- Pflichtfelder einer Notiz:
 - Titel (Text)
 - Beschreibung (formatierter Text)

- Automatisch gepflegte Felder:
 - created_at (Erstellungszeitpunkt)
 - updated_at (letzter Bearbeitungszeitpunkt)
- Eine Notiz wird über Relationstabellen verknüpft mit:
 - genau 1 Projekt (über `project_note`) ODER
 - genau 1 Kunde (über `customer_note`)
- Das Löschen einer Notiz ist endgültig und entfernt automatisch die zugehörige Relation (CASCADE).
- Das Löschen eines Projekts oder Kunden entfernt automatisch alle zugehörigen Notizen und deren Relationen (CASCADE).
- Notizen werden ausschließlich in den Detailansichten von Projekt oder Kunde verwaltet.
- Es gibt keine separate Notizverwaltung in der Navigation.
- Notizen haben keine Versionierung oder Historie.
- Notizen sind rein informativ und haben keine Auswirkung auf Terminplanung oder Geschäftslogik.

▼ Datenmodell

Tabelle: note

Feld	Typ	Beschreibung
id	BIGSERIAL PRIMARY KEY	Eindeutige ID
title	TEXT NOT NULL	Titel der Notiz
body	TEXT NOT NULL	Formatierter Inhalt (inkl. Text- und Hintergrundfarben)
created_at	TIMESTAMPTZ NOT NULL DEFAULT NOW()	Erstellungszeitpunkt
updated_at	TIMESTAMPTZ NOT NULL DEFAULT NOW()	Letzter Änderungszeitpunkt

Relationstabelle: project_note

Feld	Typ	Beschreibung
project_id	BIGINT NOT NULL	Fremdschlüssel zu project
note_id	BIGINT NOT NULL	Fremdschlüssel zu note
PRIMARY KEY	(project_id, note_id)	Composite Key

Constraints:

- Foreign Key zu project(id) ON DELETE CASCADE
- Foreign Key zu note(id) ON DELETE CASCADE

Relationstabelle: customer_note

Feld	Typ	Beschreibung
customer_id	BIGINT NOT NULL	Fremdschlüssel zu customer
note_id	BIGINT NOT NULL	Fremdschlüssel zu note
PRIMARY KEY	(customer_id, note_id)	Composite Key

Constraints:

- Foreign Key zu customer(id) ON DELETE CASCADE
- Foreign Key zu note(id) ON DELETE CASCADE

Use Cases

▼ UC: Notiz zu Projekt hinzufügen

Akteur

Disponent

Ziel

Eine neue Notiz erstellen und einem Projekt zuordnen.

Vorbedingungen

- Das Projekt existiert.
- Der Disponent ist angemeldet und berechtigt.

Ablauf

1. Benutzer öffnet die Projektdetails.
2. Benutzer wählt „Notiz hinzufügen“.
3. Das System öffnet den Richtext-Editor.

4. Benutzer erfasst Titel und Beschreibung.
5. Benutzer bestätigt die Eingabe.
6. Das System erstellt die Notiz und verknüpft sie mit dem Projekt.
7. Das System zeigt die Notiz in der Notizenliste des Projekts an.

Alternativabläufe

- Pflichtfelder fehlen: Das System fordert zur Korrektur auf.
- Abbruch: Notiz wird nicht erstellt.

Ergebnis

Die Notiz ist erstellt und dem Projekt zugeordnet.

▼ UC: Notiz zu Kunde hinzufügen

Akteur

Disponent

Ziel

Eine neue Notiz erstellen und einem Kunden zuordnen.

Vorbedingungen

- Der Kunde existiert.
- Der Disponent ist angemeldet und berechtigt.

Ablauf

1. Benutzer öffnet die Kundendetails.
2. Benutzer wählt „Notiz hinzufügen“.
3. Das System öffnet den Richtext-Editor.
4. Benutzer erfasst Titel und Beschreibung.
5. Benutzer bestätigt die Eingabe.
6. Das System erstellt die Notiz und verknüpft sie mit dem Kunden.
7. Das System zeigt die Notiz in der Notizenliste des Kunden an.

Alternativabläufe

- Pflichtfelder fehlen: Das System fordert zur Korrektur auf.
- Abbruch: Notiz wird nicht erstellt.

Ergebnis

Die Notiz ist erstellt und dem Kunden zugeordnet.

▼ **UC: Notiz bearbeiten**

Akteur

Disponent

Ziel

Eine bestehende Notiz ändern.

Vorbedingungen

- Die Notiz existiert.
- Der Disponent ist berechtigt.

Ablauf

1. Benutzer wählt eine Notiz aus der Notizenliste (Projekt oder Kunde).
2. Das System öffnet den Richtext-Editor mit den aktuellen Inhalten.
3. Benutzer ändert Titel, Beschreibung oder Farben.
4. Benutzer bestätigt die Änderungen.
5. Das System aktualisiert die Notiz.
6. Das System setzt updated_at auf den aktuellen Zeitstempel.
7. Das System zeigt die aktualisierte Notiz an.

Alternativabläufe

- Pflichtfelder ungültig: Das System fordert zur Korrektur auf.
- Abbruch: Änderungen werden verworfen.

Ergebnis

Die Notiz ist aktualisiert und in allen zugeordneten Kontexten (Projekt/Kunde) sichtbar.

▼ **UC: Notiz löschen**

Akteur

Disponent

Ziel

Eine Notiz vollständig entfernen.

Vorbedingungen

- Die Notiz existiert.
- Der Disponent ist berechtigt.

Ablauf

1. Benutzer wählt eine Notiz aus der Notizenliste.
2. Benutzer wählt „Notiz löschen“.
3. Das System zeigt eine Sicherheitsabfrage an.
4. Benutzer bestätigt das Löschen.
5. Das System löscht die Notiz und die zugehörige Relation endgültig
6. Das Löschen erfolgt immer aus dem Parent-Kontext heraus
7. Das System entfernt die Notiz aus allen Notizenlisten.

Alternativabläufe

- Abbruch: Notiz bleibt erhalten.

Ergebnis

Die Notiz ist vollständig gelöscht und erscheint in keiner Ansicht mehr.

▼ UC: Notizen eines Projekts anzeigen

Akteur

Disponent, Leser

Ziel

Alle einem Projekt zugeordneten Notizen einsehen.

Vorbedingungen

- Das Projekt existiert.
- Der Nutzer besitzt Leserechte.

Ablauf

1. Benutzer öffnet die Projektdetails.
2. Das System zeigt alle verknüpften Notizen als vertikale Kärtchenliste an.
3. Jede Notiz wird mit Titel, Beschreibung und Farben dargestellt.

Alternativabläufe

- Keine Notizen vorhanden: Das System zeigt eine leere Liste an.

Ergebnis

Alle Notizen des Projekts sind sichtbar.

▼ **UC: Notizen eines Kunden anzeigen**

Akteur

Disponent, Leser

Ziel

Alle einem Kunden zugeordneten Notizen einsehen.

Vorbedingungen

- Der Kunde existiert.
- Der Nutzer besitzt Leserechte.

Ablauf

1. Benutzer öffnet die Kundendetails.
2. Das System zeigt alle verknüpften Notizen als vertikale Kärtchenliste an.
3. Jede Notiz wird mit Titel, Beschreibung und Farben dargestellt.

Alternativabläufe

- Keine Notizen vorhanden: Das System zeigt eine leere Liste an.

Ergebnis

Alle Notizen des Kunden sind sichtbar.

Zusammenfassung (fachlich klar)

- Notizen sind eigenständige Domainobjekte mit Titel, Beschreibung und Formatierung
- Notizen werden über n:m-Beziehungen entweder Projekten oder Kunden zugeordnet
- Eine Notiz gehört immer genau einem Parent (Projekt ODER Kunde)
- Notizen existieren nie unabhängig von ihrem Parent
- Notizen werden ausschließlich in den Detailansichten von Projekt und Kunde verwaltet

- Es gibt keine separate Notizverwaltung in der Navigation
- Notizen haben keine fachliche Wirkung auf Terminplanung oder Geschäftslogik
- Notizen dienen ausschließlich der Information und Dokumentation
- Das Löschen von Notizen ist endgültig und erfolgt aus dem Parent-Kontext
- Das Löschen von Projekten/Kunden löscht auch deren Notizen (CASCADE)