

# 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.

Zusätzlich bietet das Feature vordefinierte Notizvorlagen als Eingabehilfe sowie die Möglichkeit, wichtige Notizen anzupinnen, damit diese stets oben in der Notizliste erscheinen.

## 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.

**Angepinnte Notizen** werden in der Liste immer zuerst angezeigt, unabhängig von Erstellungs- oder Änderungsdatum. Innerhalb der gepinnten und nicht-gepinnten Gruppen erfolgt die Sortierung nach Änderungsdatum (neueste zuerst).

**Notizvorlagen** sind vordefinierte Textbausteine, die beim Erstellen einer neuen Notiz als Ausgangspunkt gewählt werden können. Vorlagen werden zentral in den Stammdaten verwaltet und stehen bei der Notizerstellung als Auswahlliste zur Verfügung. Die Vorlage wird beim Erstellen in die neue Notiz kopiert – danach besteht keine Verbindung mehr zwischen Vorlage und Notiz.

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

### Allgemeine Regeln für Notizen

- 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.

### Regeln für angepinnte Notizen

- Eine Notiz kann über das Feld `is_pinned` als angepinnt markiert werden.
- Angepinnte Notizen erscheinen in der Notizliste immer vor nicht-angepinnten Notizen.
- Innerhalb der gepinnten Gruppe erfolgt die Sortierung nach `updated_at` absteigend.
- Innerhalb der nicht-gepinnten Gruppe erfolgt die Sortierung ebenfalls nach `updated_at` absteigend.
- Das Pinning kann jederzeit aktiviert oder deaktiviert werden.






### Regeln für Notizvorlagen

- Notizvorlagen sind eigenständige Stammdatenobjekte mit Titel und vordefiniertem Inhalt.
- Vorlagen existieren unabhängig von Projekten und Kunden.
- Vorlagen werden in einem eigenen Stammdatenbereich verwaltet (z.B. unter Einstellungen oder Stammdaten).
- Beim Erstellen einer Notiz kann optional eine Vorlage ausgewählt werden.
- Bei Auswahl einer Vorlage werden Titel und Beschreibung in den Editor kopiert.
- Nach dem Kopieren besteht keine Verbindung zwischen Vorlage und erstellter Notiz.
- Änderungen an einer Vorlage wirken sich nicht auf bereits erstellte Notizen aus.
- Vorlagen können eine Sortierreihenfolge haben, um die Anzeige in der Auswahlliste zu steuern.
- Vorlagen können deaktiviert werden, ohne sie zu löschen.

## Aufgaben/Notizen/Fragen/Probleme

### MuGPlan

Aa Name	# Feature	🔗 Quelle	⚙ Status	📅 Ziel Datum	👤 zugewiesen
<u>Feature für Notizen</u>	13		Erledigt		

 Name	# Feature	 Quelle	 Status	 Ziel Datum	 zugewiesen
<u>Implementierung_</u> <u>FT (13).</u>	13		Nicht begonnen		

## ▼ 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)
is_pinned	BOOLEAN NOT NULL DEFAULT FALSE	Kennzeichnung ob Notiz angepinnt ist
created_at	TIMESTAMPTZ NOT NULL DEFAULT NOW()	Erstellungszeitpunkt
updated_at	TIMESTAMPTZ NOT NULL DEFAULT NOW()	Letzter Änderungszeitpunkt

**Tabelle: note\_template (NEU)**

Feld	Typ	Beschreibung
id	BIGSERIAL PRIMARY KEY	Eindeutige ID
title	TEXT NOT NULL	Titel der Vorlage
body	TEXT NOT NULL	Vordefinierter Inhalt
sort_order	INTEGER NOT NULL DEFAULT 0	Sortierreihenfolge in der Auswahlliste
is_active	BOOLEAN NOT NULL DEFAULT TRUE	Vorlage aktiv/inaktiv
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. Optional: Das System zeigt eine Auswahlliste verfügbarer Vorlagen an.
5. Optional: Benutzer wählt eine Vorlage – Titel und Beschreibung werden in den Editor übernommen.
6. Benutzer erfasst Titel und Beschreibung.
7. Benutzer bestätigt die Eingabe.
8. Das System erstellt die Notiz und verknüpft sie mit dem Projekt.
9. 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. Optional: Das System zeigt eine Auswahlliste verfügbarer Vorlagen an.
5. Optional: Benutzer wählt eine Vorlage – Titel und Beschreibung werden in den Editor übernommen.
6. Benutzer erfasst Titel und Beschreibung.
7. Benutzer bestätigt die Eingabe.

8. Das System erstellt die Notiz und verknüpft sie mit dem Kunden.
9. 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. Angepinnte Notizen erscheinen zuerst, gekennzeichnet durch ein Pin-Symbol.
4. 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. Angepinnte Notizen erscheinen zuerst, gekennzeichnet durch ein Pin-Symbol.
4. 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.

## ▼ UC: Notiz anpinnen / Pinning aufheben (NEU)

**Akteur:** Disponent

**Ziel:** Eine Notiz dauerhaft oben in der Liste positionieren oder diese Positionierung aufheben.

**Vorbedingungen:**

- Die Notiz existiert.
- Der Disponent ist berechtigt.

**Ablauf:**

1. Benutzer wählt eine Notiz aus der Notizenliste.
2. Benutzer wählt „Anpinnen“ (bzw. „Pinning aufheben“ wenn bereits gepinnt).
3. Das System setzt das Feld `is_pinned` auf TRUE (bzw. FALSE).
4. Das System aktualisiert die Sortierung der Notizliste.
5. Angepinnte Notizen erscheinen nun oben in der Liste.

**Alternativabläufe:** Keine.

**Ergebnis:** Die Notiz ist angepinnt (oder nicht mehr angepinnt) und die Liste zeigt die neue Sortierung.

## ▼ UC: Notizvorlage erstellen (NEU)

**Akteur:** Administrator, Disponent (je nach Rollenkonzept)

**Ziel:** Eine neue Notizvorlage anlegen.

**Vorbedingungen:**

- Der Benutzer hat Zugriff auf die Stammdatenverwaltung.

**Ablauf:**

1. Benutzer öffnet die Vorlagenverwaltung (z.B. unter Stammdaten oder Einstellungen).
2. Benutzer wählt „Vorlage hinzufügen“.
3. Das System öffnet den Editor.
4. Benutzer erfasst Titel und vordefinierten Inhalt.

5. Optional: Benutzer legt Sortierreihenfolge fest.
6. Benutzer bestätigt die Eingabe.
7. Das System erstellt die Vorlage.

**Alternativabläufe:**

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

**Ergebnis:** Die Vorlage ist erstellt und steht bei der Notizerstellung zur Auswahl.

▼ **UC: Notizvorlage bearbeiten (NEU)**

**Akteur:** Administrator, Disponent (je nach Rollenkonzept)

**Ziel:** Eine bestehende Notizvorlage ändern.

**Vorbedingungen:**

- Die Vorlage existiert.
- Der Benutzer hat Zugriff auf die Stammdatenverwaltung.

**Ablauf:**

1. Benutzer öffnet die Vorlagenverwaltung.
2. Benutzer wählt eine Vorlage aus.
3. Das System öffnet den Editor mit den aktuellen Inhalten.
4. Benutzer ändert Titel, Inhalt oder Sortierreihenfolge.
5. Benutzer bestätigt die Änderungen.
6. Das System aktualisiert die Vorlage.

**Alternativabläufe:**

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

**Ergebnis:** Die Vorlage ist aktualisiert. Bereits erstellte Notizen bleiben unverändert.

▼ **UC: Notizvorlage deaktivieren/aktivieren (NEU)**

**Akteur:** Administrator, Disponent (je nach Rollenkonzept)

**Ziel:** Eine Vorlage aus der Auswahlliste entfernen, ohne sie zu löschen.

**Vorbedingungen:**

- Die Vorlage existiert.

**Ablauf:**

1. Benutzer öffnet die Vorlagenverwaltung.
2. Benutzer wählt eine Vorlage aus.
3. Benutzer wählt „Deaktivieren“ (bzw. „Aktivieren“).
4. Das System setzt das Feld `is_active` entsprechend.

**Ergebnis:** Deaktivierte Vorlagen erscheinen nicht mehr in der Auswahlliste bei der Notizerstellung, bleiben aber im System erhalten.

## ▼ UC: Notizvorlage löschen (NEU)

**Akteur:** Administrator

**Ziel:** Eine Vorlage vollständig entfernen.

**Vorbedingungen:**

- Die Vorlage existiert.

**Ablauf:**

1. Benutzer öffnet die Vorlagenverwaltung.
2. Benutzer wählt eine Vorlage aus.
3. Benutzer wählt „Vorlage löschen“.
4. Das System zeigt eine Sicherheitsabfrage an.
5. Benutzer bestätigt das Löschen.
6. Das System löscht die Vorlage endgültig.

**Alternativabläufe:**

- Abbruch: Vorlage bleibt erhalten.

**Ergebnis:** Die Vorlage ist gelöscht. Bereits erstellte Notizen bleiben unverändert.

## 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 können angepinnt werden und erscheinen dann immer oben in der Liste.
- Notizvorlagen sind vordefinierte Textbausteine als Eingabehilfe bei der Notizerstellung.
- Vorlagen werden in den Stammdaten verwaltet und haben keine Verbindung zu erstellten Notizen.
- 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).