

# SynergyHub

Audit 3

1

- Willkommen zu unserer Präsentation zum Audit3

# Inhalt

1. Domänenmodelle
2. Szenarien
3. Use-Cases
4. Use-Case-Diagramm
5. Proof of Concepts
6. Datenbank-Modelle
7. Kommunikations-Diagramme
8. Design
9. Rapid Prototype
10. Artefakte für Audit 4

2

- Viele Modelle erneut zu sehen
- Überarbeitung vieler Modelle aufgrund einer Konzeptänderung und
- **Folgende Inhalte:**

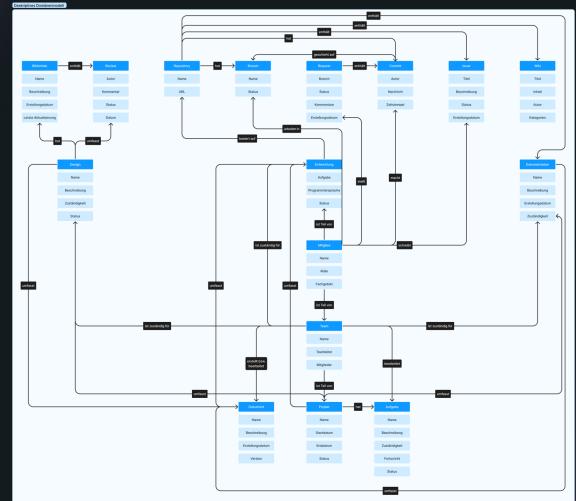
- Domänenmodelle
- Szenarien
- Use-Cases
- Use-Case-Diagramm
- Proof of Concepts
- Datenbank-Modelle
- Kommunikations-Diagramme
- Design
- Rapid Prototype (Code)
- Artefakte für Audit 4

# Domänenmodelle

3

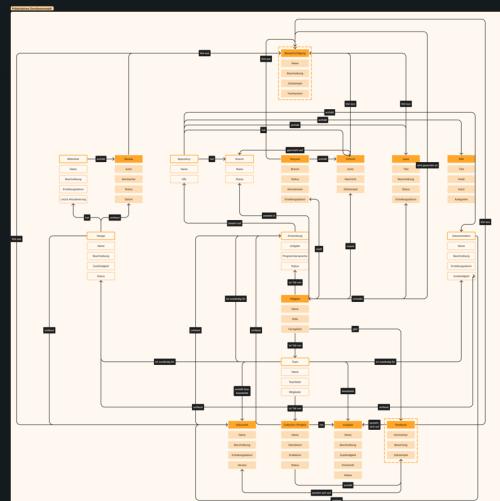
- Kommen wir zu den Domänenmodellen

# Deskriptives Domänenmodell



- **Erst das deskriptive Domänenmodell:**
    - Da wir im letzten Audit auf eine nicht Vollständige Domäne  
wurden, haben wir diese überarbeitet

## Präskriptives Domänenmodell



5

- **Jetzt das präskriptive Domänenmodell:**

- Da wir auch hier eine unvollständige Domäne hatten und das Modell anpassen mussten
- Hierbei hat es auch Änderungen gegeben, die durch Mario angebracht wurden, was unsere Perspektive verändert hat

# Szenarien

6

- Kommen wir zu den Szenarien, welche auf Basis der neuen Domäne erstellt wurden

## Anmelden mit Account

### Szenario Anmelden mit Account - Erfolgreiche Anmeldung

- Frau Bach besucht die Seite und hat bereits einen Account.
- Sie gibt für die Anmeldung einen Nutzernamen und ihr Passwort ein.
- Die Daten wurden richtig eingegeben.
- Sie wurde erfolgreich angemeldet.

Ende Anmelden mit Account - Erfolgreiche Anmeldung

## Account erstellen

### Szenario Account erstellen - Account erfolgreich erstellt

- Herr Schmitz besucht die Seite und möchte sich einen Account erstellen.
- Er gibt dafür seine Daten an und gibt einen Nutzernamen an.
- Ein Account mit diesen Daten existiert noch nicht.
- Der Account wurde erfolgreich erstellt.

Ende Account erstellen - Account erfolgreich erstellt

## Abmelden

### Szenario Abmelden - Erfolgreiche Abmeldung

- Frau Bach möchte sich abmelden.
- Dazu klickt sie in der Navigationsleiste auf "Ausloggen".
- Sie wurde erfolgreich ausgeloggt.

Ende Abmelden - Erfolgreiche Abmeldung

## Erstellen einer Collection

### Szenario Erstellen einer Collection - Erfolgreiche Erstellung einer Collection

- Herr Bauer hat sich auf der Seite angemeldet und möchte nun eine Collection erstellen.
- Er gibt in das Formular einen Namen und eine Beschreibung für die Collection an.
- Nach dem erfolgreichen Abschicken des Formulars wird er auf die Collection weitergeleitet.

Ende Erstellen einer Collection - Erfolgreiches Erstellen einer Collection

## Collection umbenennen

### Szenario Collection umbenennen - Collection umbenannt

- Herr Pythagoras möchte seine Collection umbenennen.
- Dazu wählt er unter den Einstellungen den Reiter Collection aus.
- Hier gibt er einen neuen Namen ein und bestätigt das Umbenennen.
- Die Collection wurde umbenannt.

Ende Collection umbenennen - Collection umbenannt

## Einbinden eines Projektes

### Szenario Einbinden eines Projektes - Projekt erfolgreich eingebunden

- Herr Pythagoras möchte eines seiner Projekte einer anderen Plattform mit einer Collection verbinden.
- Dazu wählt er unter den Einstellungen seine Zielplattform aus und erteilt den Zugriff.
- Anschließend wählt er ein Projekt aus.
- Das Projekt ist nun mit der Collection verknüpft.

Ende Einbinden eines Projektes - Projekt erfolgreich eingebunden

## Einladen eines Kollaborateurs

Szenario Einladen eines Kollaborateurs - Kollaborator erfolgreich eingeladen

- Frau Claassen möchte Kollaboratoren in ihre Collection einladen.
- Dazu wählt sie in den Einstellungen der Collection den passenden Button aus.
- In einem neu erschienenen Fenster gibt sie den Benutzernamen der Person ein, die sie einladen möchte.
- ⋮
- Anschließend schockt sie die Einladung ab.

Ende Einladen eines Kollaborateurs - Kollaborator erfolgreich eingeladen

## Einladung annehmen

Szenario Einladung annehmen - Einladung angenommen

- Herr Schmitz sieht in seinen Benachrichtigungen, dass er von Frau Claassen zu einer Collection eingeladen wurde.
- Er drückt auf den Annehmen-Button.
- ⋮
- Herr Schmitz wurde zu der Collection von Frau Claassen hinzugefügt.

Ende Einladung annehmen - Einladung angenommen

## Einladung ablehnen

Szenario Einladung ablehnen - Einladung abgelehnt

- Herr Schmitz sieht in seinen Benachrichtigungen, dass er von Frau Claassen zu einer Collection eingeladen wurde.
- Er drückt auf den Ablehnen-Button.
- Herr Schmitz wurde nicht zu der Collection von Frau Claassen hinzugefügt.

Ende Einladung ablehnen - Einladung abgelehnt

## Kollaborateur verwalten

Szenario Kollaborateur verwalten - Kollaborator aktualisiert

- Frau Claassen möchte Herr Schmitz aktualisieren.
- Dazu geht sie in den Einstellungen ihrer Collection unter den Reiter Kollaboratoren.
- Hier kann sie Herr Schmitz eine Rolle zuweisen oder diesen aus der Collection entfernen.
- Die Änderungen werden automatisch übernommen.

Ende Kollaborateur verwalten - Kollaborator aktualisiert

## Abschließen einer Task

Szenario Abschließen einer Task - Task wird in die nächste Status-Spalte verschoben

- Herr Bauer wurde von Frau Jones in einer Task zugewiesen.
- Dieser bearbeitet die Todos der Task und klickt sie als erledigt an.
- ⋮
- Nachdem Herr Bauer alle Todos der Task erledigt hat und diese auch als erledigt gekennzeichnet hat wird diese in die nächste Status-Spalte verschoben

Ende Abschließen einer Task - Task wird in die nächste Status-Spalte verschoben

## Erstellen einer Task

- Szenario Erstellen einer Task - Task wurde dem Taskboard hinzugefügt
- Frau Jones möchte eine Aufgabe hinzufügen und sie einem Teammitglied geben.
  - Dafür geht sie auf die Tasks-Seite der Collection und drückt Task hinzufügen.
  - In einem Fenster gibt sie nun Daten über den Status, die Plattform (GitHub, Figma, etc.), sowie den Namen und eine Beschreibung an.
  - Nun kann sie Todos für diese Task angeben. Hier gibt sie eine Beschreibung der Aufgabe ein.
  - Sie fügt Teammitglieder hinzu, die diese Task bearbeiten sollen.
  - Wenn die Form abgeschickt wird und die Daten korrekt sind, wird eine Benachrichtigung an die hinzugefügten Mitglieder geschickt.
  - Die Task wird dem Taskboard hinzugefügt.

Ende Erstellen einer Task - Task wurde dem Taskboard hinzugefügt

## Bearbeiten einer Task

- Szenario Bearbeiten einer Task - Task wurde Bearbeitet
- Herr Schweighöfer möchte eine der Tasks vom Taskboard überarbeiten und hierbei den Titel ändern und eine Todo hinzufügen.
  - Herr Schweighöfer ruft das Taskboard der Collection auf und Klickt auf die Task-Card und öffnet somit die Task.
  - Er Klickt auf den Titel und ändert diesen nach seinem Ermessen.
  - Er fügt eine Todo hinzu und gibt dieser Aufgabe eine Beschreibung.
  - Er Klickt auf Fertig um den Vorgang abzuschließen.
  - Die geänderte Task wird auf dem Taskboard angezeigt

Ende Bearbeiten einer Task - Task wurde Bearbeitet

## Feedback

- Szenario Einer Task Feedback geben - Feedback wird der Task hinzugefügt
- Herr Schweighöfer ist unzufrieden mit der Task und auch mit der Umsetzung von Herrn Bauer
  - Er öffnet den Reiter Task und Klickt auf die Task-Card.
  - Er Klickt auf Feedback geben und gibt in der Form ein Kommentar ab. Er fügt Herrn Pythagoras hinzu und fragt ihn nach seiner Meinung.
  - Er wählt aus, dass eine Benachrichtigung an die Zielpлатform geschickt werden soll (Diese Einstellung wird automatisch gespeichert).
  - Nach dem Bestätigen wird das Feedback zur Task hinzugefügt.
  - Eine Benachrichtigung wird an die Zielpлатform geschickt (Bei GitHub wird ein Issue erstellt)

Ende Einer Task Feedback geben - Feedback wird der Task hinzugefügt

## Reagieren auf Feedback

- Szenario Einem Feedback eine Bewertung geben - Das Feedback hat einen neuen Vote
- Herr Pythagoras hat eine Benachrichtigung bekommen.
  - Er wurde in einem Feedback erwähnt.
  - Er geht auf das Taskboard und liest sich das Feedback durch.
  - Da er dem Feedback zustimmt, gibt er diesem einen Upvote.
  - Der Upvote wird gespeichert und wird dem Feedback hinzugefügt.

Ende Einem Feedback eine Bewertung geben - Das Feedback hat einen neuen Vote

## Dokumente der Collection einsehen

- Szenario Dokumente der Collection einsehen - Dokumente werden angezeigt
- Frau Claassen möchte die mit der Collection verbundenen Dokumente einsehen.
  - Dazu klickt sie auf den Reiter "Inspektion".
  - Hier kann sie die verschiedenen Aufgabenbereiche betrachten.

Ende Dokumente der Collection einsehen - Dokumente werden angezeigt

## Benachrichtigungen verwalten

- Szenario Benachrichtigungen verwalten - Benachrichtigungen wurden aktualisiert
- Frau Claassen möchte ihre Einstellungen zu den Benachrichtigungen verwalten.
  - Dazu geht sie unter den Einstellungen auf den Reiter Benachrichtigungen.
  - Hier kann sie einzelne Benachrichtigungen verwalten oder generell alle Benachrichtigungen aktivieren oder deaktivieren.
  - Ihre Einstellungen werden automatisch gespeichert.

Ende Benachrichtigungen verwalten - Benachrichtigungen wurden aktualisiert

# Use-Cases

11

- Kommen wir zu den überarbeiteten Use-Cases, die auf Grundlage Szenarien erstellt wurden

## Einloggen

```
use case Einloggen
actors Nutzer
precondition Nutzer hat einen Account und ist nicht angemeldet
main flow
    1. Der Nutzer gibt seine E-Mail und sein Passwort ein.
    2. Das System prüft die eingegebenen Daten auf Korrektheit.
    3. Der Nutzer ist angemeldet und wird auf die Hauptseite geleitet.
postcondition
    Erfolgreiche Anmeldung des Nutzers
exceptional flow Fehlerhafte Daten
    2a. Das System stellt eine Inkonsistenz der Daten fest.
    3. Der Nutzer wird auf den Fehler hingewiesen.
postcondition Fehlerhafte Anmelddaten
    Der Nutzer wurde nicht angemeldet und wurde darüber benachrichtigt.
exceptional flow Kein Nutzer mit diesem Konto
    2a. Das System stellt eine Inkonsistenz der Daten fest.
    3. Der Nutzer wird auf den Fehler hingewiesen.
end Einloggen
```

## Registrieren

```
use case Registrieren
actors Nutzer
precondition Der Nutzer hat keinen Account und möchte sich registrieren
main flow
    1. Der Nutzer geht zur Seite für die Registrierung.
    2. Der Nutzer gibt seine Informationen ein, wie Vorname, Nachname, E-Mail-Adresse und ein Passwort seiner Wahl.
    3. Das System prüft, ob die Eingegebenen Daten okay sind, z.B. ob die E-Mail-Adresse korrekt ist und das Passwort sicher genug ist.
    4. Wenn die Daten in Ordnung sind, erstellt das System einen neuen Account für den Nutzer.
    5. Der Nutzer bekommt eine E-Mail zur Bestätigung, um sicherzustellen, dass die Registrierung echt ist.
    6. Nach erfolgreicher Bestätigung wird der Nutzer zur Anmeldeseite weitergeleitet.
postcondition Account erstellt
    Ein neuer Nutzer-Account wurde erstellt.
end Registrieren
```

```

Erstellung von Collection
use case Erstellung von Collection
actors Nutzer
precondition Der Benutzer hat ein Konto und ist angemeldet
+ :: main flow
    1. Der Benutzer navigiert zur Seite für die Erstellung einer Sammlung.
    2. Der Benutzer gibt einen Namen und optional eine Beschreibung für die Sammlung ein.
    3. Das System überprüft die eingegebenen Informationen auf Gültigkeit.
    4. Bei gültigen Informationen erstellt das System eine neue Sammlung für den Benutzer.
    5. Der Benutzer wird zur Seite der erstellten Sammlung weitergeleitet.
postcondition Collection erstellt
    Eine neue Collection wurde erstellt.
end Erstellung von Collection

Verwaltung von Collection
use case Verwaltung von Collection
actors Administrator
precondition Der Administrator ist angemeldet und hat Zugriff auf die Sammlungsverwaltung
main flow
    1. Der Administrator wählt die Option "Sammlung verwalten" aus.
    2. Das System zeigt eine Liste aller vorhandenen Elemente in der Sammlung an.
    3. Der Administrator kann folgende Aktionen ausführen:
        • Element hinzufügen
        • Element entfernen
        • Element bearbeiten
        • Sammlung durchsuchen
    4. Das System aktualisiert die Sammlung basierend auf den durchgeföhrten Aktionen.
postcondition Collection aktualisiert
+ :: Die Collection wurde aktualisiert.
end Verwaltung von Collection

```

**Autorisierung von anderen Plattformen**

```
use case Autorisierung auf anderer Plattform
actors Benutzer, externe Plattform
precondition Der Benutzer ist angemeldet und möchte die Anwendung mit einer externen Plattform autorisieren
main flow
    1. Der Benutzer wählt die Option "Externe Plattform autorisieren" in den Anwendungseinstellungen.
    + :: 2. Das System leitet den Benutzer zur Anmeldeseite der externen Plattform weiter.
    3. Der Benutzer gibt seine Anmeldeinformationen für die externe Plattform ein.
    4. Die externe Plattform überprüft die Anmeldeinformationen.
    5. Wenn die Anmeldung erfolgreich ist, fordert die externe Plattform den Benutzer auf, die Autorisierung für die Anwendung zu erteilen.
    6. Der Benutzer stimmt der Autorisierung zu.
    7. Die externe Plattform generiert einen Autorisierungscode und sendet diesen an die Anwendung.
    8. Das System verwendet den Autorisierungscode, um einen Zugriffstoken von der externen Plattform zu erhalten.
    9. Das System speichert den erhaltenen Zugriffstoken für zukünftige Interaktionen mit der externen Plattform.

postcondition Benutzer hat andere Plattformen Autorisiert
    Der Benutzer hat die Berechtigung für die Nutzung einer anderen Plattform erteilt.
end Autorisierung von anderen Plattformen
```

**Feedback geben**

```
use case Feedback geben
actors Benutzer, System
precondition Der Benutzer hat die Möglichkeit, Feedback zu geben, und das System unterstützt die Erfassung von Rückmeldungen
main flow
    1. Der Benutzer navigiert zu der Feedback-Seite oder -Funktion in der Anwendung.
    2. Das System präsentiert dem Benutzer ein Formular oder eine Schnittstelle zur Eingabe von Feedback.
    3. Der Benutzer gibt sein Feedback ein, das Informationen zu positiven Erfahrungen, Verbesserungsvorschlägen oder Problemen enthalten kann.
    4. Das System überprüft und speichert das eingegebene Feedback.

postcondition Benutzer hat ein Feedback gegeben
+ :: Der Benutzer hat neues Feedback hinterlassen.
end Feedback geben
```

### Auf Feedback reagieren

```
use case Auf Feedback reagieren
actors Benutzer
precondition Das System hat Feedback von Benutzern erhalten
main flow
    1. Das System überwacht kontinuierlich eingehendes Feedback von Benutzern.
    + :: 2. Wenn neues Feedback erfasst wird, wird es in einem Feedback-Verwaltungssystem
          gespeichert.
    3. Ein Administrator überprüft regelmäßig das eingehende Feedback.
    4. Der Administrator analysiert das Feedback, um Trends, Muster oder wiederkehrende
       Probleme zu identifizieren.
    5. Basierend auf der Analyse werden geeignete Maßnahmen ergriffen:
        • Positive Rückmeldungen: Diese können für Motivation und Anerkennung genutzt
          werden.
        • Verbesserungsvorschläge: Die werden in den Entwicklungsprozess integriert.
        • Probleme: Sofortige Lösungen oder Pläne zur Fehlerbehebung werden eingeleitet.
    6. Das System aktualisiert den Status des Feedbacks im Feedback-Verwaltungssystem.
postcondition Auf feedback reagiert
    Es wurde auf auf das Feedback reagiert.
end Auf Feedback reagieren
```

### Benachrichtigung senden

```
use case Benachrichtigung senden
actors System, Benutzer
precondition Das System unterstützt die Funktion, Benachrichtigungen zu senden.
Ein Auslöser tritt auf, der eine Benachrichtigung auslöst (z. B. Ereignis, Update, Nachricht)
main flow
    1. Das System identifiziert die betroffenen Benutzer oder Zielgruppe für die
       Benachrichtigung.
    2. Das System erstellt die Benachrichtigung mit relevanten Informationen, einschließlich
       Text, Medien oder Links.
    3. Die Benachrichtigung wird an die jeweiligen Benutzer gesendet.
    4. Die Benutzer erhalten die Benachrichtigung auf ihrem bevorzugten Endgerät oder
       Kanal (z. B. App, E-Mail, Push-Benachrichtigung).
    5. Die Benutzer können auf die Benachrichtigung reagieren oder weitere Aktionen
       ausführen.
+ :: postcondition Benachrichtigung wurde gesendet
    Der Benutzer hat eine Benachrichtigung vom System erhalten
end Benachrichtigung senden
```

### **Benachrichtigung für Collection aktivieren**

use case Benachrichtigung für Collection aktivieren

actors Nutzer, System

precondition Der Benutzer ist angemeldet und hat Zugriff auf die Einstellungen oder Verwaltungsfunktionen der Sammlung

+ main flow

1. Der Benutzer navigiert zu den Einstellungen oder der Verwaltungsseite der Sammlung.
2. Das System zeigt die Option "Benachrichtigungen aktivieren" für die ausgewählte Sammlung an.
3. Der Benutzer wählt die Option "Benachrichtigungen aktivieren" aus.
4. Das System fragt nach den Präferenzen des Benutzers für Benachrichtigungen (z. B. E-Mail-Benachrichtigungen, Push-Benachrichtigungen).
5. Der Benutzer wählt seine bevorzugten Benachrichtigungseinstellungen aus.
6. Das System speichert die ausgewählten Einstellungen für die Benachrichtigung der Sammlung.

postcondition Der Nutzer hat die Benachrichtigung für eine Collection aktiviert

Die Einstellungen des Nutzers wurden erfolgreich aktualisiert

end Benachrichtigung für Collection aktivieren

### **Ausloggen**

use case Ausloggen

actors Nutzer

precondition Nutzer ist aktuell angemeldet

main flow

1. Der Nutzer wählt die Option zum Ausloggen aus.
2. Das System führt eine Abmeldung durch, beendet die aktuelle Sitzung und leitet den Nutzer auf die Anmeldeseite.

postcondition Erfolgreich ausgeloggt

Der Nutzer befindet sich auf der Anmeldeseite oder einer Bestätigungsseite.

end Ausloggen

## Projekt verknüpfen

use case Projekt verknüpfen

actors Nutzer, System

precondition Nutzer ist angemeldet und auf der Projektverwaltungssseite

mainflow

1. Der Nutzer wählt die Option, ein neues Projekt zu verknüpfen oder ein bestehendes Projekt auszuwählen.
- + :: 2. Das System zeigt dem Nutzer eine Liste der verfügbaren Projekte oder ermöglicht ihm die Erstellung eines neuen Projekts.
3. Der Nutzer wählt das gewünschte Projekt aus oder erstellt ein neues Projekt und bestätigt die Auswahl.
4. Das System speichert die Verknüpfung zum ausgewählten Projekt und aktualisiert die Projektverwaltungssseite.

precondition Erfolgreiche Verknüpfung des Projekts

Das ausgewählte Projekt ist nun mit dem aktuellen Arbeitskontext verknüpft.

end Projekt verknüpfen

## Collection entfernen

use case Collection entfernen

actors Nutzer, System

precondition Nutzer ist angemeldet und auf der Seite zur Verwaltung von Sammlungen (Collections)

main flow

1. Der Nutzer wählt die Option, eine bestimmte Collection zu entfernen.
2. Das System zeigt dem Nutzer eine Bestätigungsfrage oder ein Dialogfeld, um sicherzustellen, dass er die ausgewählte Collection wirklich entfernen möchte.
3. Der Nutzer bestätigt die Aktion durch Klicken auf "Entfernen" oder eine ähnliche Bestätigungsaktion.
4. Das System entfernt die ausgewählte Collection und aktualisiert die Ansicht der verbleibenden Collections.

postcondition Nutzer hat eine Collection entfernt

Die Collection wurde erfolgreich entfernt.

end Collection entfernen

### Collection anzeigen

```
use case Collection anzeigen
actors Nutzer, System
precondition Nutzer ist angemeldet und auf der Hauptseite der Anwendung oder auf der Seite zur Verwaltung von Sammlungen
main flow
    1. Der Nutzer wählt die Option, eine bestimmte Collection anzuseigen.
    + 2. Das System zeigt dem Nutzer die ausgewählte Collection mit ihren Inhalten und relevanten Details an.
postcondition dem Nutzer wird eine Collection angezeigt
    Das System präsentiert die Inhalte der Collection.
end Collection anzeigen
```

### Kollaboratoren einladen

```
use case Kollaboratoren einladen
actors Nutzer, Kollaboratoren, System
precondition Nutzer ist angemeldet und befindet sich auf der Seite des Projekts oder der Zulassungsverarbeit, für die Kollaboratoren eingeladen werden sollen
main flow
    1. Der Nutzer wählt die Option, Kollaboratoren einzuladen.
    2. Das System zeigt dem Nutzer ein Formular oder ein Dialogfeld an, in dem er die E-Mail-Adressen der Kollaboratoren eingeben kann.
    3. Der Nutzer gibt die E-Mail-Adressen der Kollaboratoren ein und wählt die entsprechende Option, um eine Einladung für jeden Kollaborator aus.
    4. Der Nutzer bestätigt die Einladung, und das System sendet Einladungs-E-Mails an die angegebenen E-Mail-Adressen.
    + 5. Die eingeladenen Kollaboratoren erhalten die Einladungs-E-Mails mit einem Link zur Anmeldung.
postcondition Der Benutzer hat eine Einladung verschickt
    Ein neuer Kollaborator wurde hinzugefügt
end Kollaboratoren einladen
```

## **Content hinzufügen**

**use case** Content hinzufügen

**actors** Nutzer, System

**precondition** Nutzer ist angemeldet und befindet sich auf der Seite des Projekts oder Bereichs, wo neuer Content hinzugefügt werden soll

+ :: main flow

1. Der Nutzer wählt die Option, neuen Content hinzuzufügen.
2. Das System zeigt dem Nutzer ein Formular oder eine Eingabemaske, in der er relevante Informationen für den neuen Content eingeben kann (z. B. Titel, Beschreibung, Dateianhänge).
3. Der Nutzer gibt die erforderlichen Informationen ein und lädt bei Bedarf Dateien hoch.
4. Der Nutzer bestätigt die Eingabe, und das System speichert den neuen Content.
5. Das System zeigt eine Bestätigungsmeldung an und aktualisiert die Ansicht, um den neuen Content anzuzeigen.

**postcondition** Content hinzugefügt

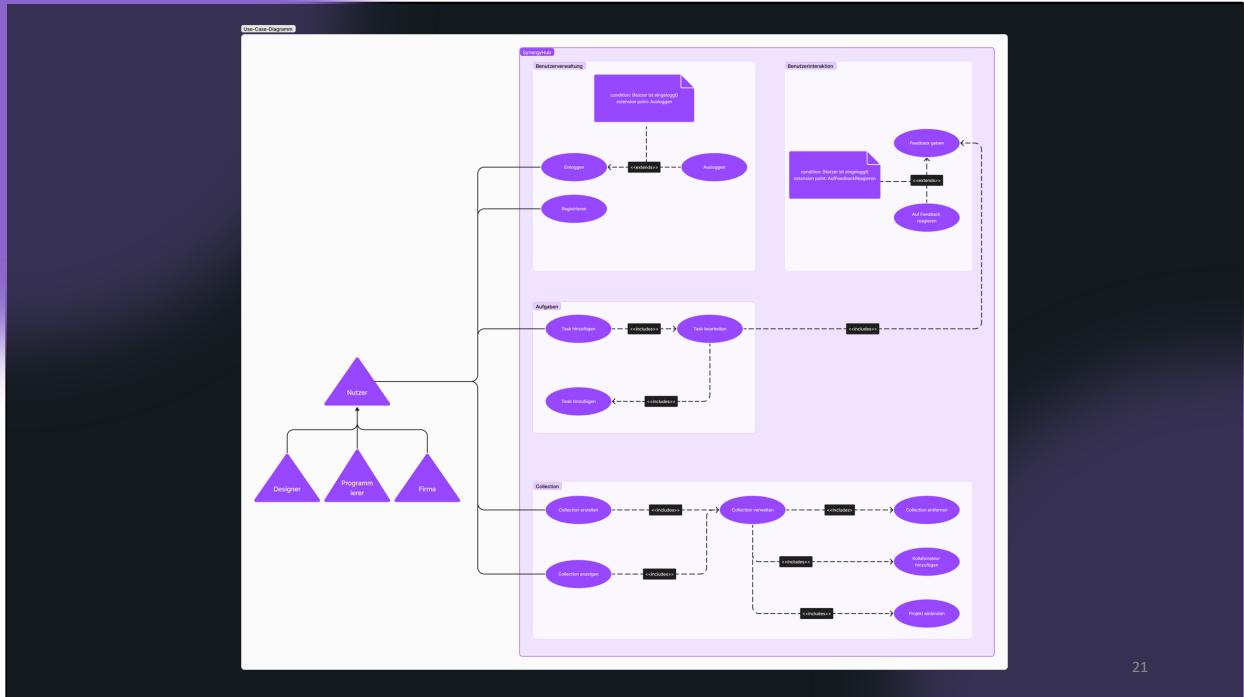
Es wurde neuer Content zu einer Collection hinzugefügt

**end** Content hinzufügen

# Use-Case-Diagramm

20

- Jetzt das Use-Case-Diagramm
- Auch dieses musste aufgrund der neuen Use-Cases und des Perspektivs überarbeitet werden



21

- Bei der Konzipierung des Use-Case-Diagramms haben sich vier Benutzungsszenarien herauskristallisiert:

- Benutzerverwaltung
- Benutzerinteraktion
- Aufgaben / Tasks
- Collection

# Proof of Concepts

22

- Nun zu den Proof of Concepts
- Die Proof of Concepts basieren auf den Use Cases und wurden verwendet, um ein Rapid Prototype zu entwickeln

## 1. Registrieren

Name: Registrieren

Beschreibung: Der Nutzer erstellt einen neuen Account.

Ablauf:

1. Der Nutzer öffnet die Registrierungsseite.
2. Der Nutzer gibt valide Informationen (Nutzernname, E-Mail, Passwort) ein.
3. Das System prüft die Validität und Eindeutigkeit der Informationen.
4. Bei erfolgreicher Prüfung erstellt das System einen neuen Account und leitet den Nutzer zur Anmeldeseite weiter.

Exit-Kriterien:

1. Neuer Nutzer-Account erfolgreich erstellt.
2. Nutzer wurde auf die Anmeldeseite weitergeleitet.

Fall-Kriterien:

1. Fehlermeldung bei ungültigen Eingaben.
2. Fehlschlagen des Account-Erstellungsprozesses.

Fallbacks:

1. Anzeigen von Fehlermeldungen und Aufforderung zur erneuten Eingabe.
2. Bei technischen Problemen den Nutzer informieren und um erneuten Versuch bitten.

## 2. Einloggen

Name: Einloggen

Beschreibung: Der Nutzer loggt sich in seinen Account ein.

Ablauf:

1. Der Nutzer gibt Nutzernamen und Passwort ein.
2. Das System überprüft die eingegebenen Daten auf Korrektheit.
3. Bei erfolgreicher Überprüfung wird der Nutzer zur Übersichtsseite weitergeleitet.

Exit-Kriterien:

1. Erfolgreiche Anmeldung des Nutzers.
2. Weiterleitung des Nutzers auf die Übersichtsseite.

Fall-Kriterien:

1. Fehlermeldung bei falschen Anmelddaten.
2. Fehlschlagen des Anmeldevorgangs.

Fallbacks:

1. Anzeigen von Fehlermeldungen und Aufforderung zur erneuten Eingabe.
2. Bei technischen Problemen den Nutzer informieren und um erneuten Versuch bitten.

### 3. Ausloggen

Name: Ausloggen

Beschreibung: Der Nutzer loggt sich aus.

Ablauf:

1. Der Nutzer navigiert zur Übersichtsseite.
2. Der Nutzer klickt auf den Logout-Button.
3. Das System führt die Abmeldung durch und leitet den Nutzer auf die Anmeldeseite.

Exit-Kriterien:

1. Erfolgreiches Ausloggen des Nutzers.
2. Weiterleitung des Nutzers auf die Anmeldeseite.

Fail-Kriterien:

1. Fehlschlagen des Ausloggen-Prozesses.

Fallbacks:

1. Den Nutzer informieren und um erneuten Versuch bitten.

### 4. Collection erstellen

Name: Collection erstellen

Beschreibung: Der Nutzer erstellt eine neue Collection.

Ablauf:

1. Der Nutzer navigiert zur Übersichtsseite und klickt auf "Create Collection".
2. Der Nutzer gibt einen Namen und optional eine Beschreibung für die Collection ein.
3. Der Nutzer bestätigt das Erstellen und wird auf die Taskboard-Seite der neu erstellten Collection weitergeleitet.

Exit-Kriterien:

1. Erfolgreiche Erstellung einer neuen Collection.
2. Weiterleitung des Nutzers zur Taskboard-Seite der neu erstellten Collection.

Fail-Kriterien:

1. Fehlermeldung bei ungültigen Eingaben.
2. Fehlschlagen des Erstellungsprozesses.

Fallbacks:

1. Anzeigen von Fehlermeldungen und Aufforderung zur erneuten Eingabe.
2. Bei technischen Problemen den Nutzer informieren und um erneuten Versuch bitten.

### **5. Projekt einbinden (Code)**

Name: Projekt einbinden (Code)

Beschreibung: Der Nutzer bindet ein bestehendes Projekt von einer anderen Plattform in eine Collection ein.

Ablauf:

1. Der Nutzer navigiert zur Settings-Seite und klickt auf den passenden Button.
2. Hier autorisiert der Nutzer die Plattform, mit welcher eine Verbindung aufgebaut werden soll.
3. Der Nutzer wählt auf einer Übersichtsseite das Projekt für die Verbindung aus.
4. Das System verbindet die Collection mit diesem Projekt.

Exit-Kriterien:

1. Erfolgreiche Autorisierung und Verknüpfung des Projekts mit der Collection.
2. Der Benutzer hat die Berechtigung für die Nutzung einer anderen Plattform erteilt.

Fall-Kriterien:

1. Fehlermeldung bei ungültigen Projektdaten.
2. Fehlgeschlagen der Verknüpfung.

Fallbacks:

1. Anzeigen von Fehlermeldungen und Aufforderung zur erneuten Eingabe.
2. Bei technischen Problemen den Nutzer informieren und um erneuten Versuch bitten.

### **6. Projekt einbinden (Grafik)**

Name: Projekt einbinden (Grafik)

Beschreibung: Der Nutzer bindet ein bestehendes Projekt von einer anderen Plattform in eine Collection ein.

Ablauf:

1. Der Nutzer navigiert zur Settings-Seite und klickt auf den passenden Button.
2. Hier autorisiert der Nutzer die Plattform, mit welcher eine Verbindung aufgebaut werden soll.
3. Der Nutzer gibt in ein Feld den Link zu dem Projekt, das er verknüpfen möchte, ein.
4. Das System verbindet die Collection mit diesem Projekt.

Exit-Kriterien:

1. Erfolgreiche Autorisierung und Verknüpfung des Projekts mit der Collection.
2. Der Benutzer hat die Berechtigung für die Nutzung einer anderen Plattform erteilt.

Fall-Kriterien:

1. Fehlermeldung bei ungültigen Projektdaten.
2. Fehlgeschlagen der Verknüpfung.

Fallbacks:

1. Anzeigen von Fehlermeldungen und Aufforderung zur erneuten Eingabe.
2. Bei technischen Problemen den Nutzer informieren und um erneuten Versuch bitten.

**7. Collection verwalten**

Name: Collection verwalten  
Beschreibung: Der Nutzer möchte seine Collection anpassen.  
Ablauf:

- Der Nutzer ruft die Settings-Seite der Collection auf.
- Er klickt auf einen der Reiter.
- Er macht eine Änderung und bestätigt diese.

Exit-Kriterien:

- Erfolgreiche Aktualisierung der Collection nach vorgenommenen Änderungen.

Fall-Kriterien:

- Fehlschlagen des Aktualisierungsprozesses.

Fallbacks:

- Anzeigen von Fehlermeldungen und Aufforderung zur erneuten Eingabe.
- Bei technischen Problemen den Nutzer informieren und um erneuten Versuch bitten.

**8. Kollaborateur hinzufügen**

Name: Kollaborateur hinzufügen  
Beschreibung: Der Nutzer fügt einen Kollaborateur zu einer Collection hinzu.  
Ablauf:

- Der Nutzer ruft die Settings-Seite der Collection auf.
- Er klickt auf den Reiter "invite collaborator".
- Hier gibt der Nutzer den Nutzernamen eines Kollaborateurs ein und bestätigt seine Angabe.
- Das System prüft, ob dieser Kollaborateur existiert.
- Der Kollaborateur wird zur Collection hinzugefügt.

Exit-Kriterien:

- Erfolgreiches Hinzufügen eines neuen Kollaborateurs zur Collection.

Fall-Kriterien:

- Fehlermeldung, wenn der Kollaborateur nicht existiert.

Fallbacks:

- Anzeigen von Fehlermeldungen und Aufforderung zur erneuten Eingabe.
- Bei technischen Problemen den Nutzer informieren und um erneuten Versuch bitten.

**9. Kollaborator verwalten**

Name: Kollaborator verwalten

Beschreibung: Der Nutzer kann einen Kollaborator verwalten.

Ablauf:

1. Der Nutzer ruft die Settings-Seite der Collection auf.
2. Er klickt auf den Reiter "Manage collaborators".
3. Hier kann der Nutzer Kollaboratoren entfernen oder die Rollen dieser ändern.
4. Das System führt die Aktionen aus.

Exit-Kriterien:

1. Erfolgreiches Verwalten der Kollaboratoren.

Fall-Kriterien:

1. Fehlermeldung, wenn der Kollaborator nicht existiert.
2. Fehlschlagen des Aktualisierungs-Prozesses.

Fallbacks:

1. Anzeigen von Fehlermeldungen und Aufforderung zur erneuten Aktion.
2. Bei technischen Problemen den Nutzer informieren und um erneuten Versuch bitten.

**10. Task hinzufügen**

Name: Task hinzufügen

Beschreibung: Der Nutzer fügt eine Task zum Taskboard einer Collection hinzu.

Ablauf:

1. Der Nutzer ruft die Taskboard-Seite einer Collection auf.
2. Der Nutzer klickt auf "Add Task" unter dem jeweiligen Status, bei dem er die Task hinzufügen möchte.
3. Der Nutzer gibt Daten wie Namen, ein verknüpftes Projekt und freiwillig eine Beschreibung zu dieser Task ein.
4. Der Nutzer kann außerdem die Kollaboratoren, die an dieser Task arbeiten sollen, hinzufügen.
5. Der Nutzer bestätigt diese Daten und erstellt somit eine Task.

Exit-Kriterien:

1. Erfolgreiche Erstellung einer neuen Task auf dem Taskboard.

Fall-Kriterien:

1. Fehlermeldung bei ungültigen Eingabedaten.
2. Fehlschlagen des Erstellungsprozesses.

Fallbacks:

1. Anzeigen von Fehlermeldungen und Aufforderung zur erneuten Eingabe.
2. Bei technischen Problemen den Nutzer informieren und um erneuten Versuch bitten.

<p><b>11. Task bearbeiten</b></p> <p>Name: Task bearbeiten</p> <p>Beschreibung: Der Nutzer aktualisiert eine Task.</p> <p>Ablauf:</p> <ol style="list-style-type: none"> <li>Der Nutzer ruft die Taskboard-Seite einer Collection auf und klickt auf die Task die er bearbeiten möchte.</li> <li>Er ändert bzw. den Namen der Task, fügt eine Beschreibung hinzu oder aktualisiert die Kolaboratoren.</li> <li>Der Nutzer bestätigt die Änderungen.</li> </ol> <p>Exit-Kriterien:</p> <ol style="list-style-type: none"> <li>Erfolgreiche Aktualisierung der Task nach vorgenommenen Änderungen.</li> </ol> <p>Fall-Kriterien:</p> <ol style="list-style-type: none"> <li>Fehlschlagen des Aktualisierungs-Prozesses.</li> </ol> <p>Fallbacks:</p> <ol style="list-style-type: none"> <li>Fehlermeldungen anzeigen und den Nutzer zur erneuten Eingabe auffordern.</li> <li>Bei technischen Problemen den Nutzer informieren und um erneuten Versuch bitten.</li> </ol>	<p><b>12. Todo hinzufügen</b></p> <p>Name: Todo hinzufügen</p> <p>Beschreibung: Der Nutzer fügt eine Todo zu einer Task hinzu.</p> <p>Ablauf:</p> <ol style="list-style-type: none"> <li>Der Nutzer ruft die Taskboard-Seite einer Collection auf und klickt auf die Task, der er Todo's hinzufügen möchte.</li> <li>Der Nutzer klickt auf "Add Subtask" um ein Todo hinzuzufügen.</li> <li>Hier gibt der Nutzer eine Beschreibung ein und bestätigt diese Eingabe um das Todo der Task hinzuzufügen.</li> </ol> <p>Exit-Kriterien:</p> <ol style="list-style-type: none"> <li>Erfolgreiches Hinzufügen eines Todos zur Task.</li> </ol> <p>Fall-Kriterien:</p> <ol style="list-style-type: none"> <li>Fehlermeldung bei ungültigen Todo-Daten.</li> <li>Fehlschlagen des Hinzufügen-Prozesses.</li> </ol> <p>Fallbacks:</p> <ol style="list-style-type: none"> <li>Anzeigen von Fehlermeldungen und Aufforderung zur erneuten Eingabe.</li> <li>Bei technischen Problemen den Nutzer informieren und um erneuten Versuch bitten.</li> </ol>
---	--

### 13. Feedback geben

Name: Feedback geben

Beschreibung: Der Nutzer gibt ein Feedback zu einer Task.

Ablauf:

1. Der Nutzer geht auf eine Task aus dem Tektopboard.
2. Er scrollt unter zur Sektion "Feedback" und gibt dort einen Kommentar ins Feld ein.
3. Um diesen Kommentar hinzuzufügen bestätigt er das Posten mit einem Klick auf den Button.
4. Das System schickt eine Benachrichtigung an die Verantwortlichen dieser Task.

Exit-Kriterien:

1. Erfolgreiches Hinzufügen von Feedback zu einer Task.

Fall-Kriterien:

1. Fehlermeldung bei leerem Feedback-Feld.
2. Fehlgeschlagen des Hinzufügen-Prozesses.

Fallbacks:

1. Anzeigen von Fehlermeldungen und Aufforderung zur erneuten Eingabe.
2. Bei technischen Problemen den Nutzer informieren und um erneuten Versuch bitten.

### 14. Auf Feedback reagieren

Name: Auf Feedback reagieren

Beschreibung: Der Nutzer möchte seine Meinung zu einem Feedback zum Ausdruck bringen.

Ablauf:

1. Der Nutzer navigiert auf die Task, in der er auf ein Feedback reagieren möchte.
2. Hier klickt der Nutzer bei dem Voting-Symbol entweder auf Upvote oder Downvote, um seine Meinung zu äußern.

Exit-Kriterien:

1. Erfolgreiche Reaktion auf gegebenes Feedback.

Fall-Kriterien:

1. Fehlschlagen des Hinzufügens der Reaktion.

Fallbacks:

1. Fehlermeldungen anzeigen und den Nutzer zur erneuten Eingabe auffordern.
2. Bei technischen Problemen den Nutzer informieren und um erneuten Versuch bitten.

### 15. Collection anzeigen

Name: Collection anzeigen

Beschreibung: Der Nutzer navigiert auf eine bestimmte Collection.

#### Ablauf:

1. Der Nutzer navigiert auf die Übersichtsseite.
  2. Der Nutzer klickt auf den Namen der Collection, die er sehen möchte.
  3. Der Nutzer wird auf das Taskboard der ausgewählten Collection geleitet.
- Exit-Kriterien:
- + 1. Erfolgreiches Anzeigen der ausgewählten Collection.
- Fall-Kriterien:
- 1. Fehlschlagen des Anzeigeprozesses.
- Fallbacks:
- 1. Fehlermeldungen anzeigen und den Nutzer informieren.
  - 2. Den Nutzer ggf. wieder auf die Übersichtsseite zurückleiten.
  - 3. Bei technischen Problemen den Nutzer informieren und um erneuten Versuch bitten.

### 16. Collection entfernen

Name: Collection entfernen

Beschreibung: Der Nutzer entfernt eine Collection.

#### Ablauf:

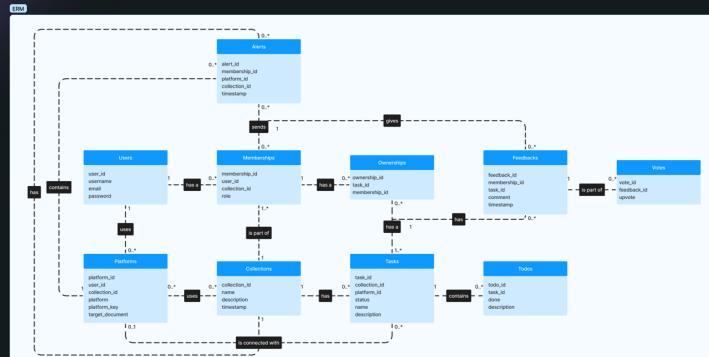
1. Der Nutzer geht auf die Settings-Seite einer Collection.
  2. Er wählt den Reiter "Delete collection".
  3. Der Nutzer gibt den Namen der Collection ein, um zu verifizieren, dass er diese wirklich entfernen möchte.
  4. Er bestätigt die Eingabe mit einem Klick auf den Button.
- Exit-Kriterien:
- 1. Erfolgreiches Entfernen der Collection.
- Fall-Kriterien:
- + 1. Fehlermeldung bei falschem Collection-Namen.
  - 2. Fehlschlagen des Entfernungsprozesses.
- Fallbacks:
- 1. Anzeigen von Fehlermeldungen und Aufforderung zur erneuten Eingabe.
  - 2. Bei technischen Problemen den Nutzer informieren und um erneuten Versuch bitten.

# Datenbank-Modelle

31

- Kommen wir nun zu den Datenbank-Modellen

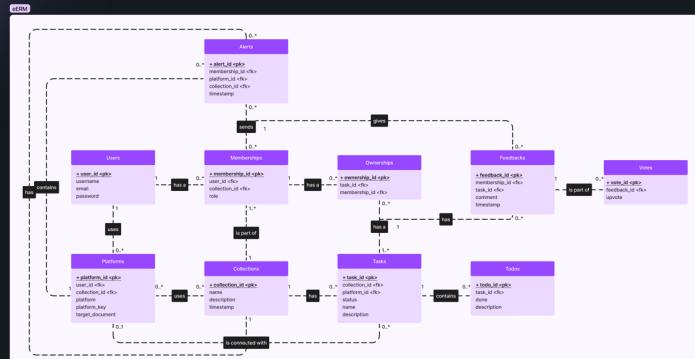
## ERM-Modell



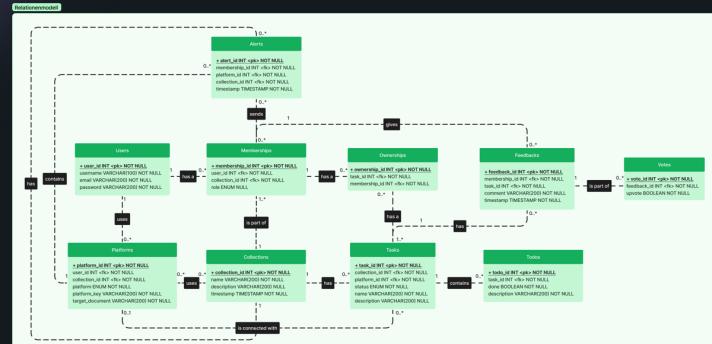
32

- Die Konzipierung der Datenbank hat sich im Vergleich zum letzten Modell verändert
- Hat den Grund, dass das neue Konzept andere Schwerpunkte hat, was zu einer Anpassung der neuen Tabellen abgedeckt werden müssen
- Außerdem war Erstellung diesmal einfacher, da durch Szenarien und Use Cases die relevanten Informationen hervorgehoben wurden

## eERM-Modell



# Relationen-Modell



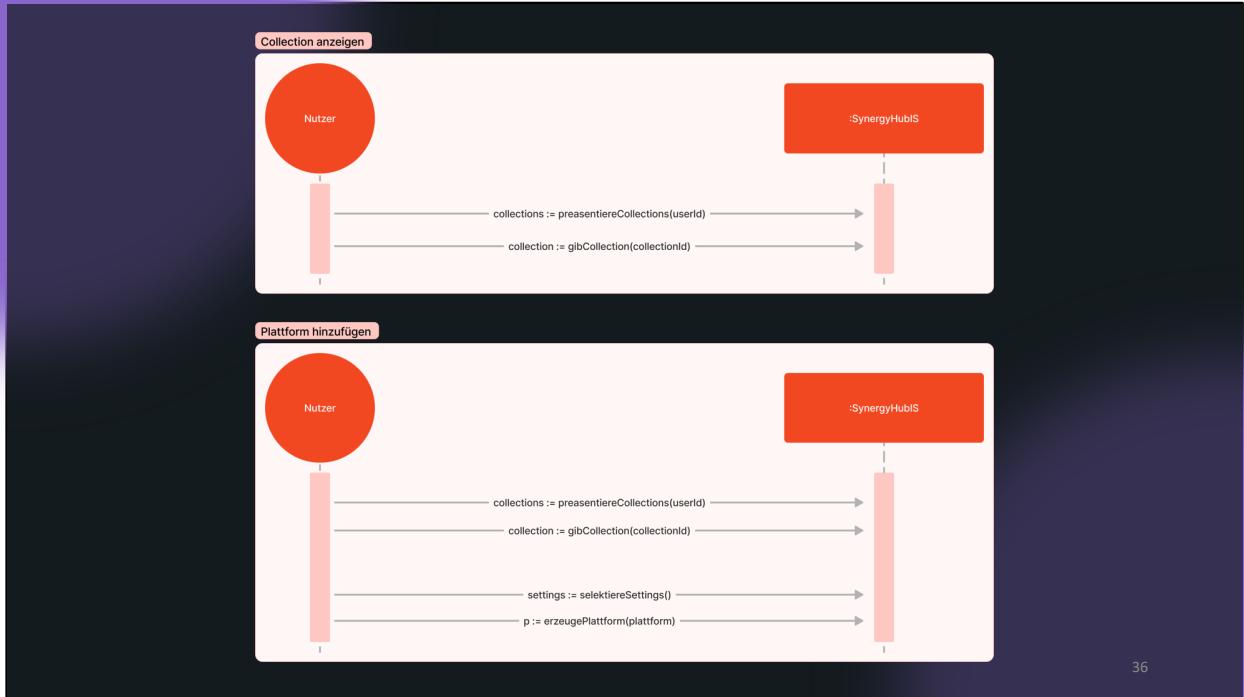
34

- Wichtig hierbei ist, dass diese Modelle sich noch bei ein paar Punkten ändern werden, da die Umsetzung des Rapid Prototypes Fehler in der Datenbank enthalten haben
- Darunter ist Beispielsweise, dass eine Task auch ohne eine Plattform existieren kann

# Kommunikations-Diagramme

35

- Nun die Kommunikations-Diagramme
- Hierbei haben wir nicht jeden Use-Case abgebildet, da das zum einen zu viele wären und zum anderen, die Kommunikation innerhalb der Use-Cases so komplex ist, als dass man diese abbilden müsste



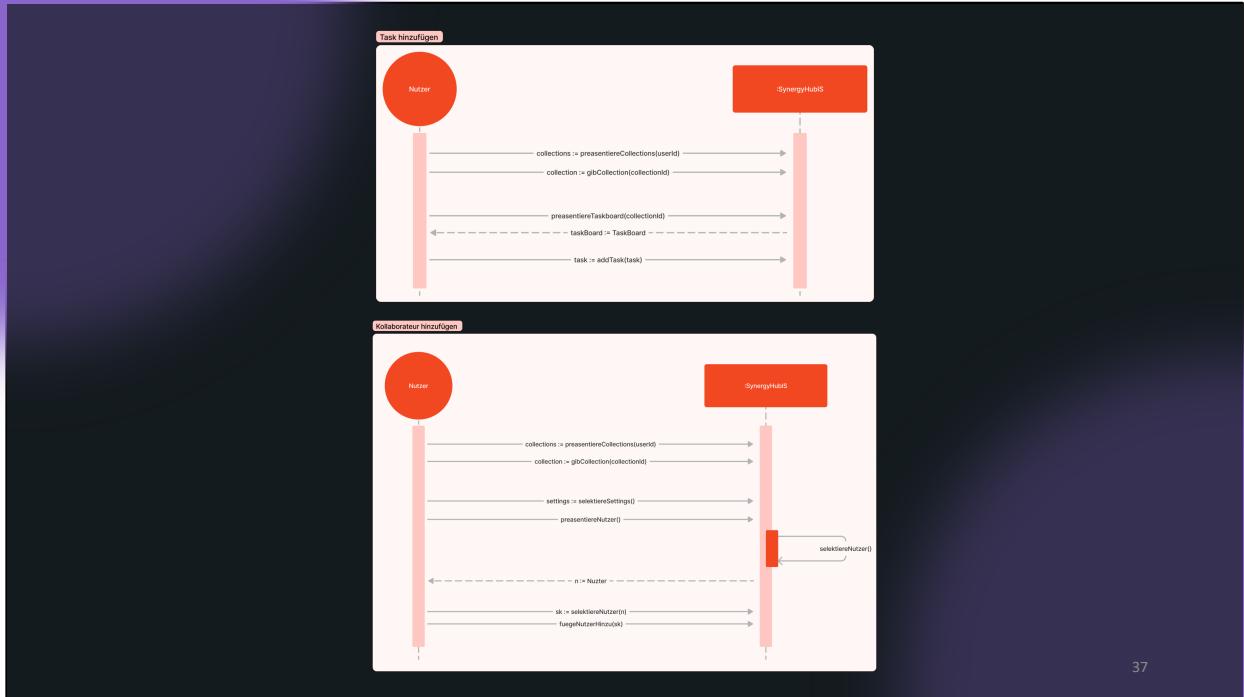
36

### Collection anzeigen:

- Wir haben einen Nutzer der das System für die eigenen Collections nutzt
- Das System liefert hierfür diese Collections zurück
- Der Nutzer wählt nun die Collection aus die sehen möchte
- Das System zeigt diese Collection nun dem Nutzer an

### Plattform hinzufügen:

- Auch hier präsentiert das System dem Nutzer alle Collections und dieser wählt die gewünschte Collection um diese angezeigt zu bekommen
- Der Nutzer selektiert die Settings dieser Collection und bekommt diese angezeigt
- Durch einen Knopfdruck gibt der Nutzer dem System den Befehl eine neue Plattform zu verbinden



37

### Task hinzufügen:

- Auch hier präsentiert das System dem Nutzer alle Collections und dieser wählt die gewünschte Collection um diese angezeigt zu bekommen
- Der Nutzer selektiert das Taskboard der Collection und bekommt dieses angezeigt
- Der Nutzer klickt auf Add Task und füllt ein Formular aus und schickt dieses an das System welches die Task zum Board hinzufügt

### Kollaborateur hinzufügen:

- Auch hier präsentiert das System dem Nutzer alle Collections und dieser wählt die gewünschte Collection um diese angezeigt zu bekommen
- Der Nutzer selektiert die Settings dieser Collection und bekommt diese angezeigt
- Der Nutzer Sucht nach einem Nutzer mit dem Namen und bekommt diesen angezeigt
- Durch eine Bestätigung schickt der Benutzer den gewählten Nutzer an das System und fügt diesen zur Collection hinzu

# Design

38

- Kommen wir zum Design, das sowohl für den Rapid Prototype, als auch für SynergyHub verwendet werden soll
- Es haben sich vom Design zur Umsetzung ein paar Elemente verändert. Diese Folgenden auch erläutert werden sollen

**Sign in**

Please enter your data to continue.

Username

Password

Don't have an account? [Sign up](#)

**Sign up**

Please enter your details to create a free account and use our service.

Username

Email

Password

Confirm password

Already have an account? [Sign in](#)

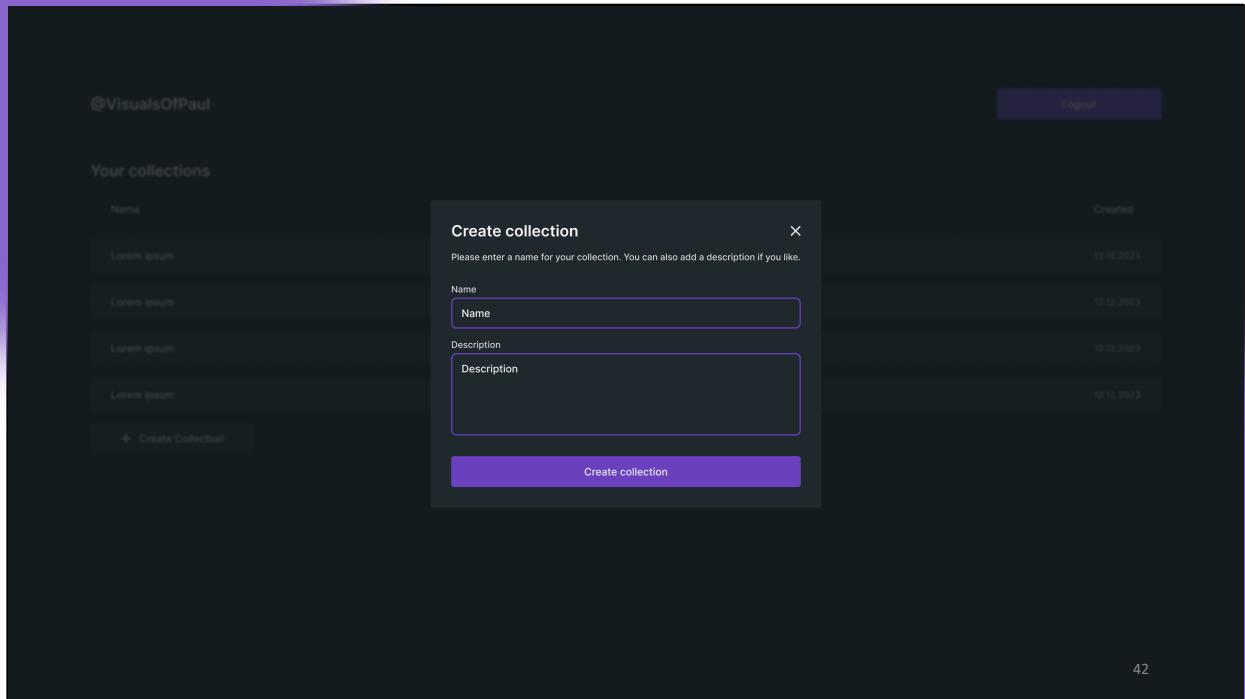
The screenshot shows a dark-themed application interface. At the top left is the handle '@VisualsOfPaul'. At the top right is a purple 'Logout' button. Below this, the heading 'Your collections' is displayed. A table lists four collections, each with a name and a creation date. At the bottom left of the table is a button labeled '+ Create Collection'.

Name	Created
Lorem ipsum	12.12.2023

+ Create Collection

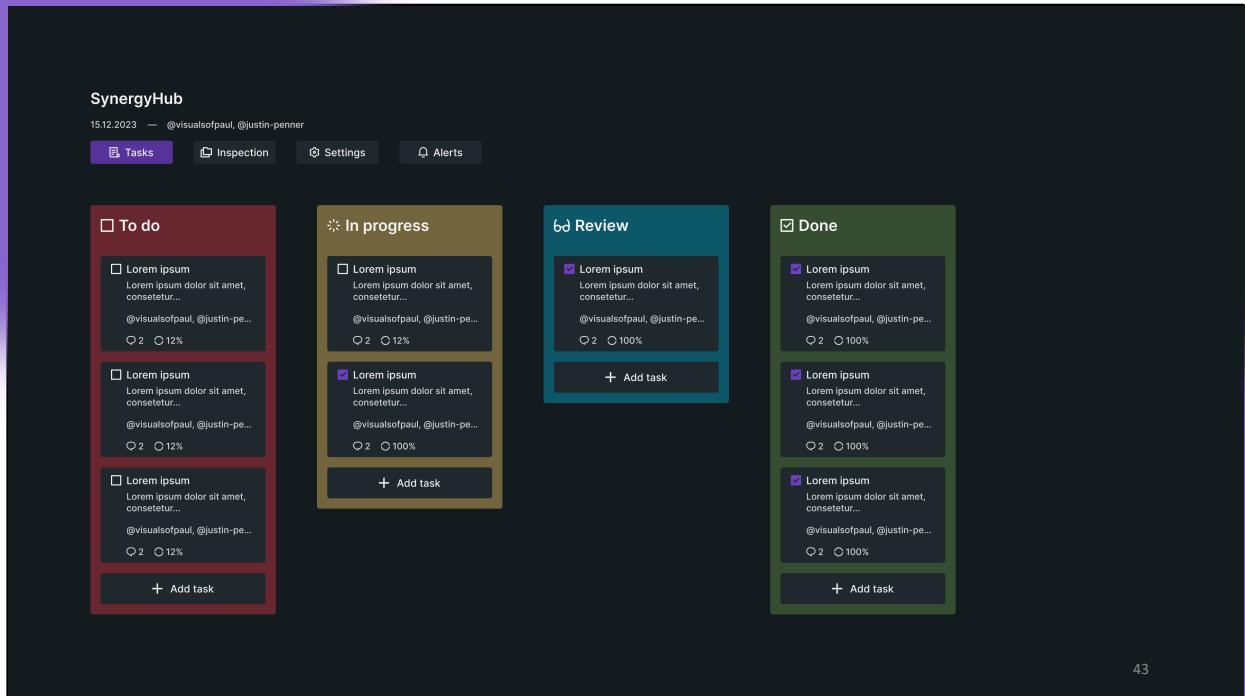
41

- Hier hat sich der „Create Collection“-Button in das normale Button verändert
- Dies wurde so entschieden, da hierdurch das System mehr vereinheitlicht wird



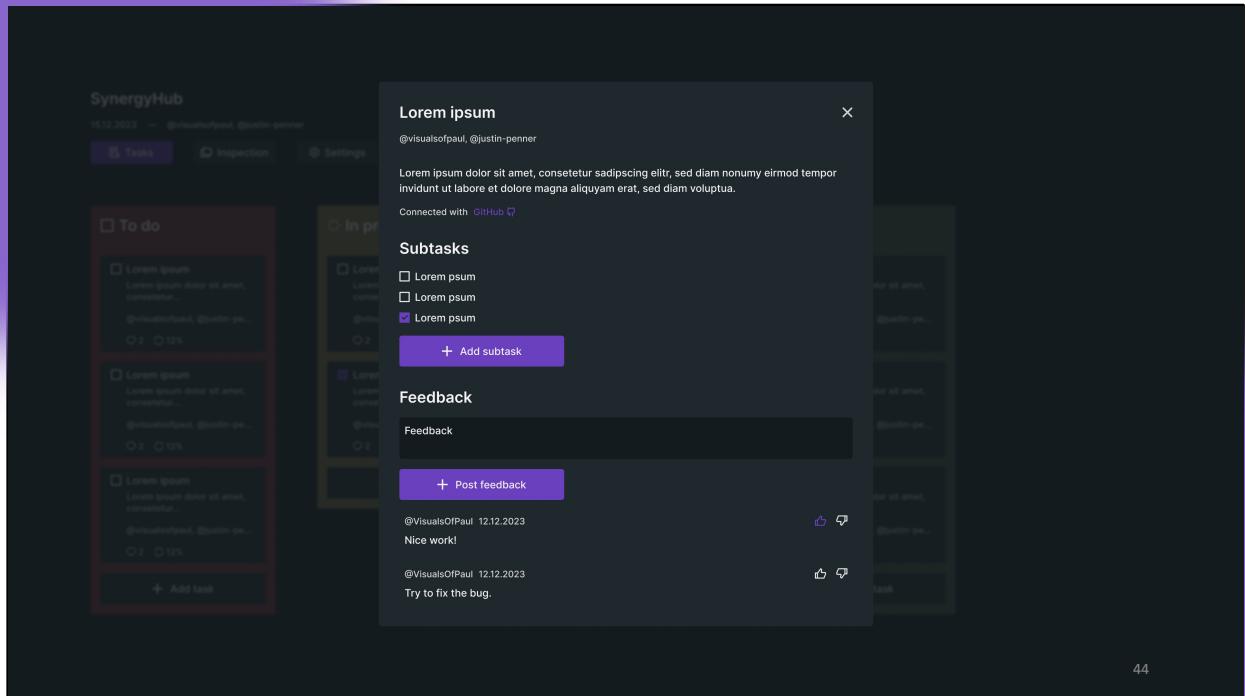
42

- Dieses Design wurde an sich exakt so umgesetzt, wie hier zu sehen
- Lediglich der Close-Button wurde nicht umgesetzt, da hierfür noch ein Konzept erarbeitet werden soll



43

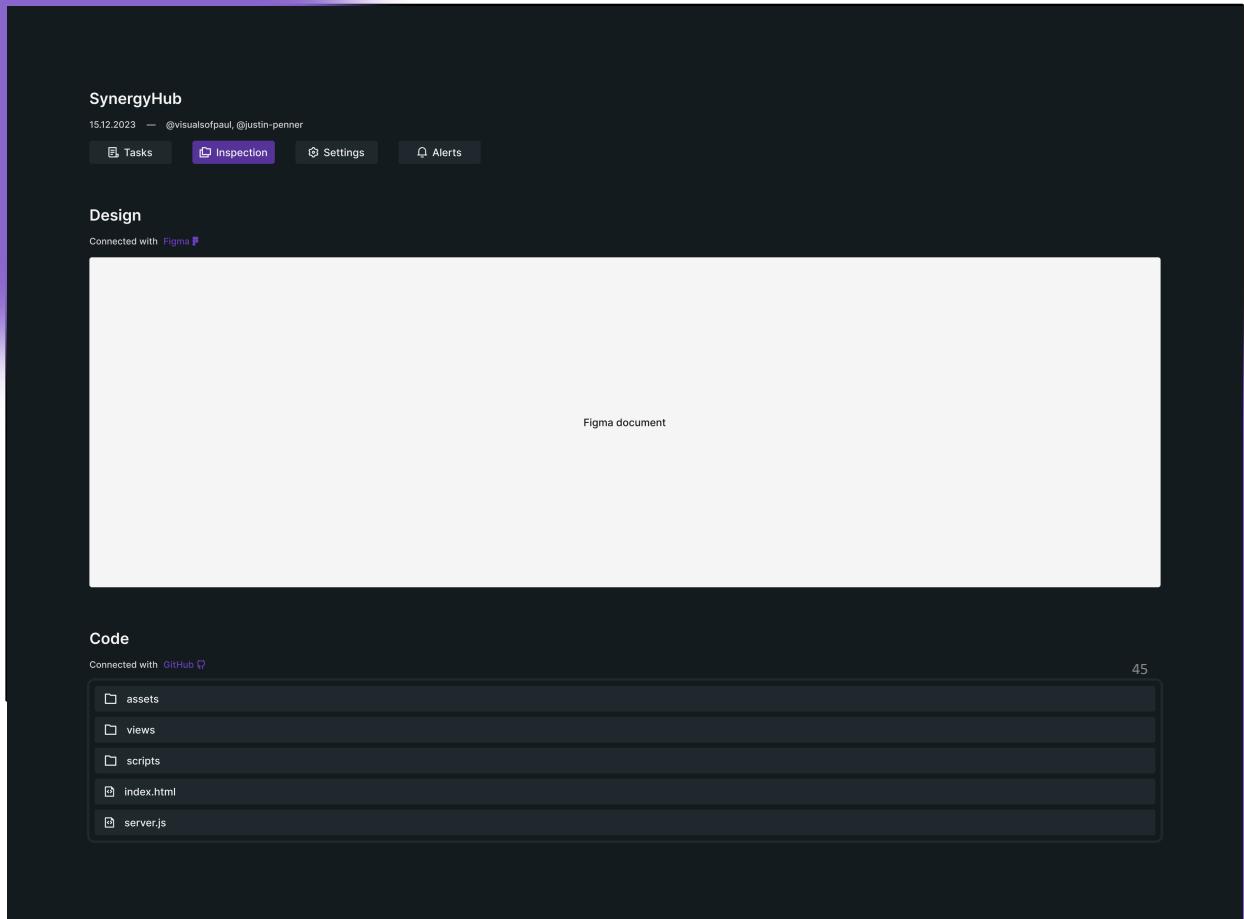
- Hier wurden die Icons der Navigations-Elemente und der einzelnen entfernt
- Dies hat den Hintergrund, dass das System mit den Icons zu überladen war, die Icons keine wirkliche Funktion hatten

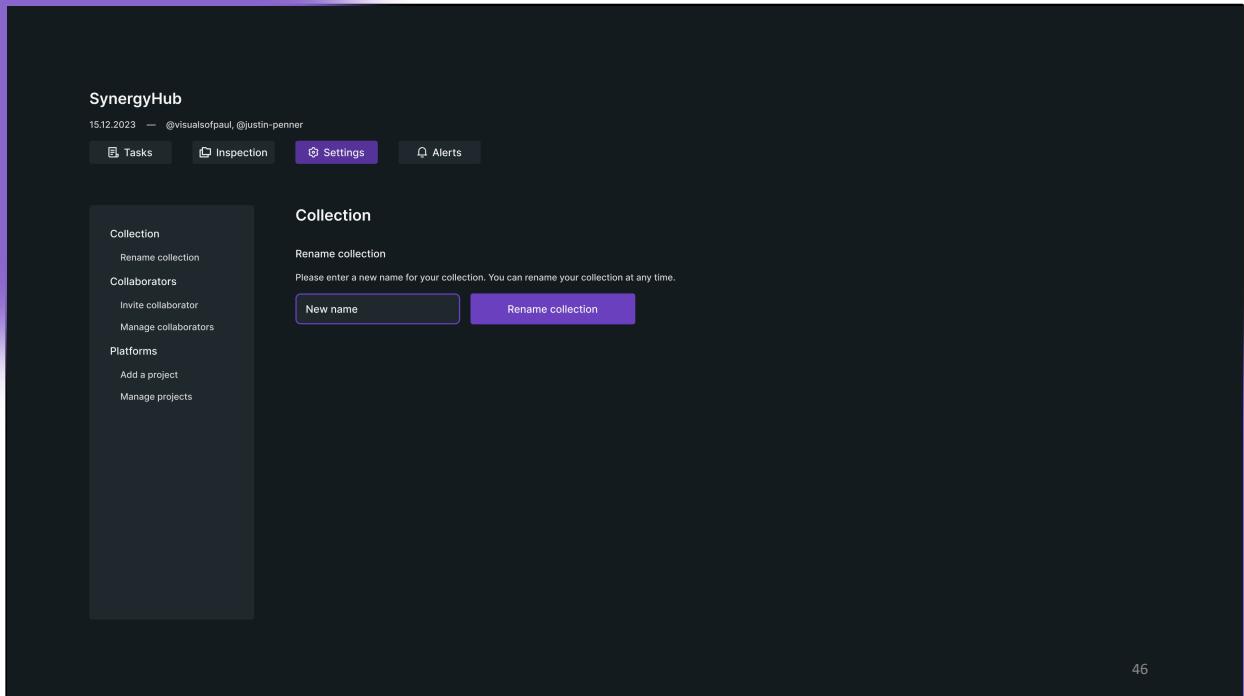


44

- **Hier hat sich das Design grundlegend verändert:**

- Die Überschriften und Informationen wurden vom Style vereinfacht.
- Der "Add subtask"-Button wurde entfernt und durch eine intuitivere Methode ausgetauscht (Mehr dazu im Rapid Prototype)
- Die Feedbacks wurden unter Berücksichtigung von Reaktionen auf die einzelnen Feedbacks angepasst, um eine klare Struktur zu gewährleisten





46

The screenshot shows a dark-themed user interface for SynergyHub. At the top, there is a header bar with the text "SynergyHub" and the date "15.12.2023 — @visualsofpaul, @justin-penner". Below the header are four navigation buttons: "Tasks", "Inspection", "Settings", and "Alerts" (which is highlighted in purple). The main content area is a table with the following columns: "Description", "Platform", "Caused by", and "Date". There are five rows in the table, each representing an alert. All alerts have the same description: "Add files via upload". The "Platform" column shows "GitHub" with a dropdown arrow. The "Caused by" column shows "@VisualsOfPaul". The "Date" column shows "12.12.2023". Each row has a small "X" icon on the right side.

Description	Platform	Caused by	Date
Add files via upload	GitHub ▾	@VisualsOfPaul	12.12.2023
Add files via upload	GitHub ▾	@VisualsOfPaul	12.12.2023
Add files via upload	GitHub ▾	@VisualsOfPaul	12.12.2023
Add files via upload	GitHub ▾	@VisualsOfPaul	12.12.2023
Add files via upload	GitHub ▾	@VisualsOfPaul	12.12.2023

- Dieses Design wurde noch nicht umgesetzt

# Rapid Prototype

Code-Inspection

48

- Nun kommen wir zum Rapid Prototype
- Hier sehen wir gleich die umgesetzten Modelle, Designs und POCs

## Artefakte für Audit 4

1. Poster
2. SynergyHub
3. Kritische Reflexion

49

- **Für den Folgenden Audit gibt es folgende Elemente:**

- Poster
- SynergyHub als umgesetztes Code-Endprodukt
- Und eine Kritische Reflexion zu unserem Prozess und der Teamarbeit