

Prototyping Projektdokumentation

Name: Christo Cheruparambil

E-Mail: chechr01@students.zhaw.ch

URL der deployten Anwendung: <https://yugioh-deck-manager.netlify.app/>

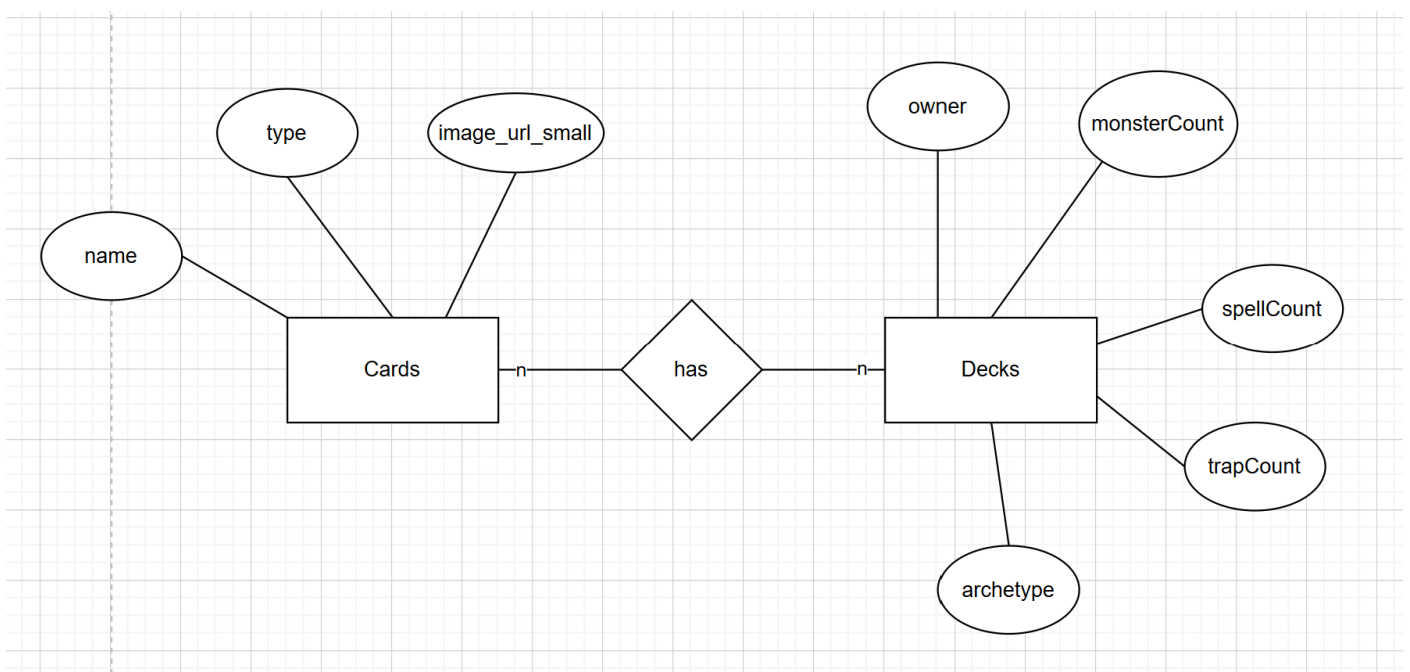
1. Einleitung

Die Webanwendung Yu-Gi-Oh! Deck Manager ermöglicht es Benutzern, ihre eigenen Yu-Gi-Oh!-Kartendecks digital zu verwalten. Ziel der App ist es, eine einfache und strukturierte Möglichkeit zur Pflege von Decks und Karten bereitzustellen. Die Benutzer können neue Karten hinzufügen, sowie bestehende Karten bearbeiten oder löschen.

Jedes Deck wird mit Informationen wie dem Namen des Besitzers und dem zugehörigen Archetype dargestellt. Für jedes Deck gibt es zudem eine Detailansicht, auf der die Kartenverteilung nach Typ (Monster, Zauber, Falle) übersichtlich angezeigt wird.

Die Daten werden in einer MongoDB-Datenbank gespeichert. Die Anwendung wurde mit SvelteKit für das Frontend und Backend sowie mit Bootstrap für ein responsives Design umgesetzt.

2. Datenmodell



Das gezeigte ERD-Diagramm (Entity-Relationship-Diagramm) beschreibt die Struktur der Datenbank für den Yu-Gi-Oh! Deck Manager. Es enthält zwei Hauptentitäten: Cards und Decks, sowie eine Beziehung has zwischen ihnen.

Die Entität Cards umfasst folgende Attribute:

- name: der Name der Karte
- type: z. B. Monster, Spell oder Trap
- image_url_small: eine Bild-URL zur Darstellung der Karte

Die Entität Decks enthält:

- owner: der Besitzer des Decks
- archetype: z. B. "Blue-Eyes"
- monsterCount, spellCount, trapCount: Zählwerte für die Anzahl der jeweiligen Kartentypen im Deck

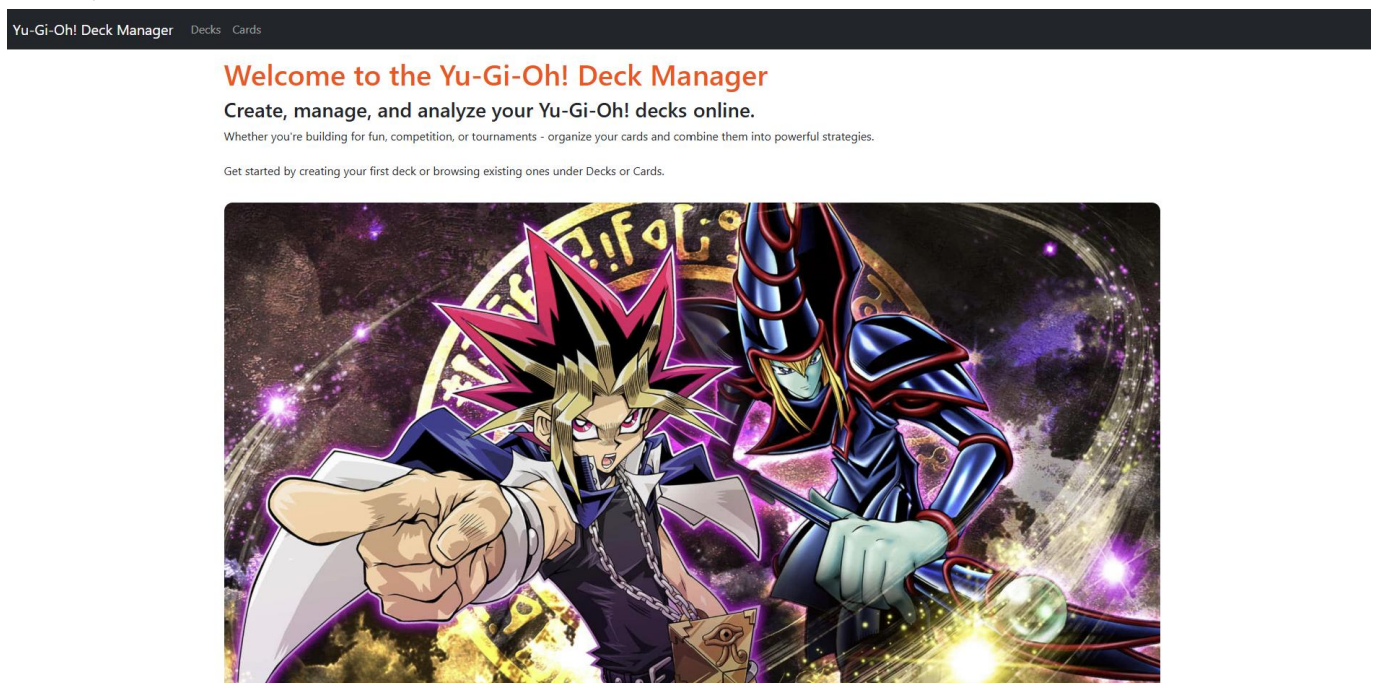
Die Beziehung has stellt eine n:m-Beziehung zwischen Cards und Decks dar. Das bedeutet, eine Karte kann in mehreren Decks vorkommen, und ein Deck besteht aus mehreren Karten.

Da MongoDB (wie NoSQL-Datenbanken allgemein) keine klassische relationale Struktur verwendet, würde man in einer relationalen Datenbank für diese n:m-Beziehung normalerweise eine Zwischentabelle benötigen (z. B. DeckCards). Aus zeitlichen Gründen wurde diese Zwischentabelle jedoch nicht umgesetzt, und stattdessen wird jede Karte nur einem Deck zugeordnet, was der aktuellen Implementierung entspricht. Damit ist die Beziehung in der Anwendung auf eine 1:n-Beziehung reduziert.

3. Beschreibung der Anwendung

3.1. Startseite

Route: /



Die Startseite begrüßt den Nutzer mit dem Titel „Welcome to the Yu-Gi-Oh! Deck Manager“ sowie einem einleitenden Text, der die Idee der Anwendung erklärt. Die Seite lädt dazu ein, eigene Yu-Gi-Oh!-Decks zu erstellen, zu verwalten und zu analysieren. Ausserdem motiviert sie zur Nutzung der Funktionen durch eine visuelle Darstellung bekannter Charaktere aus dem Yu-Gi-Oh!-Universum.

Funktionalität:

- Rein informativ
- Stellt die Anwendung vor
- Bietet eine einfache Navigation zu den Bereichen Decks und Cards über die Navigationsleiste

Diese Seite dient als Einstiegspunkt und vermittelt sofort das Thema und den Stil der Anwendung. Sie enthält keine interaktive Funktionalität, sondern leitet den Benutzer zu den Hauptbereichen der App weiter. Ganz zu oberst im Browser Tab sieht man auch das Favicon.

Dateien:

- routes/+page.svelte
- routes/+layout.svelte

- routes/styles.css
- src/app.html
- static/favicon2.png
- static/yugioh.png

3.2. Decks

3.2.1. Overview

Route: /decks

Yu-Gi-Oh! Deck Manager Decks Cards		
All Decks		
ID	Owner	Archetype
6840b8b415d0e9944bffe156	Kaiba	Blue-Eyes

Auf dieser Seite wird eine Liste aller vorhandenen Decks in Tabellenform angezeigt. Zu jedem Deck werden die folgenden Informationen dargestellt:

- ID: Die eindeutige Datenbank-ID des Decks (anklickbar, führt zur Detailansicht)
- Owner: Der Name des Besitzers (z. B. „Kaiba“)
- Archetype: Der übergeordnete Decktyp (z. B. „Blue-Eyes“)

In der Navigation oben ist „Decks“ hervorgehoben, was signalisiert, dass der Nutzer sich aktuell auf der Deck-Übersichtsseite befindet. Die Tabelle ist dynamisch und kann mehrere Einträge darstellen, je nach Anzahl gespeicherter Decks.

Dateien:

- routes/decks/+page.svelte
- routes/decks/+page.server.js

3.2.2. Details

Route: /decks/[deck_id]

Yu-Gi-Oh! Deck Manager Decks Cards	
Deck: Blue-Eyes	
Owner: Kaiba	
Card distribution:	
<ul style="list-style-type: none"> • Monsters: 30 • Spells: 8 • Traps: 2 	

Diese Seite zeigt die Detailansicht eines spezifischen Decks, in diesem Fall das „Blue-Eyes“-Deck. Angezeigt werden der Name des Decks (hier „Blue-Eyes“), der Owner (hier „Kaiba“) sowie eine statistische Übersicht der enthaltenen Karten.

Die Card distribution ist nach Typ aufgeschlüsselt:

- Monsters: 30 Karten
- Spells: 8 Karten
- Traps: 2 Karten

So erhält der Nutzer einen schnellen Überblick über die Deckzusammensetzung und kann besser analysieren, ob das Deck ausgewogen oder strategisch sinnvoll aufgebaut ist.

Dateien:

- routes/decks/[deck_id]/+page.svelte
- routes/decks/[deck_id]/+page.server.js

3.3. Cards

3.3.1. Overview

Route: /cards

Yu-Gi-Oh! Deck Manager Decks **Cards**

List of all Cards

[+ Add Card](#)

 <p>BEACON OF WHITE [SPELL CARD]</p> <p>If you do not control another "Beacon of White", and you have 3 or more "Blue-Eyes" monsters in your G.V. Target 1 of them. Special Summon it, but this effect is negated, also equip it with this card. When this card leaves the field, banish the equipped monster. Other monsters you control cannot attack, also if you use any number of "Blue-Eyes" monster(s) in your G.V., the equipped monster can attack up to that number of times during each Battle Phase.</p>	 <p>BINGO MACHINE, GO!!! [SPELL CARD]</p> <p>Reveal cards from your Deck that each meets at least 1 of the criteria listed below, your opponent randomly picks 1 for you to add to your hand, and you shuffle the rest into your Deck. You can only activate 1 "Bingo Machine, Go!!!" per turn.</p> <ul style="list-style-type: none">● "Blue-Eyes" monster● Spell Trap that specifically lists the card "Blue-Eyes White Dragon" or "Blue-Eyes Ultimate Dragon", except 1 "Bingo Machine, Go!!!"	 <p>BLUE-EYES ABYSS DRAGON</p> <p>[DRAGON/EFFECT]</p> <p>If this card is Special Summoned: You can add 1 Ritual Spell or "Beyonder Dragon" from your Deck to your hand. During your End Phase: You can add 1 Level 11 or higher Dragon monster to any your Deck to your hand. You can banish this card from your G.O. at Level 11 or higher Dragon monster you control gain 1 x ATK. You can only use each effect of "Blue-Eyes Abyss Dragon" once per turn, and can only add one of them while "Blue-Eyes White Dragon" is on your field or in your G.O.</p> <p>ATK/2500 DEF/2500</p>
<p>Beacon of White</p> <p>Spell Card</p> <p>Delete Update</p>	<p>Bingo Machine, Go!!!</p> <p>Spell Card</p> <p>Delete Update</p>	<p>Blue-Eyes Abyss Dragon</p> <p>Effect Monster</p> <p>Delete Update</p>

Auf dieser Seite werden alle gespeicherten Yu-Gi-Oh!-Karten visuell dargestellt. Jede Karte wird mit Bild, Name und Kartentyp angezeigt. Darunter befinden sich zwei Schaltflächen: Delete (zum Löschen der Karte) und Update (zum Bearbeiten).

Der Titel „List of all Cards“ sowie der Link „+ Add Card“ am Anfang der Seite ermöglichen es, neue Karten hinzuzufügen. Das Layout ist responsiv und zeigt die Karten in einem Kartenraster (Card Grid), was eine übersichtliche Darstellung mehrerer Karten nebeneinander erlaubt.

Funktionalität:

- Visuelle Anzeige der Kartengalerie
- Löschen von Karten mit Bestätigungsdialog
- Weiterleitung zur Bearbeitungsseite über „Update“
- Neue Karte hinzufügen über „+ Add Card“

Dateien:

- Routes/cards/+page.svelte
- Routes/cards/+page.server.js

3.3.2. Update

Route: /cards/[card_id]/update

Yu-Gi-Oh! Deck Manager

Decks

Cards

Edit Card


Name

Beacon of White

Type

Spell Card

Current Image



Upload New Image (optional)

Datei auswählen

Keine ausgewählt

Update Card

Auf dieser Seite kann eine bestehende Karte bearbeitet werden. Der Nutzer kann den Namen und den Typ der Karte ändern sowie das aktuell verwendete Bild einsehen. Optional besteht auch die Möglichkeit, ein neues Bild hochzuladen, um die Darstellung der Karte zu aktualisieren.

Die bestehenden Felder sind beim Laden bereits mit den aktuellen Kartendaten vorausgefüllt, sodass Änderungen schnell vorgenommen werden können. Am Ende des Formulars befindet sich ein Button zum Speichern der Änderungen.

Dateien:

- Routes/cards/[card_id]/update/+page.server.js
- Routes/cards/[card_id]/update/+page.svelte

3.3.3. Create

Route: /cards/create

Yu-Gi-Oh! Deck Manager

Decks

Cards

[Back](#)

Add a Card

Name*

Type*

Upload Image

Datei auswählen

Keine ausgewählt

Add Card

Auf dieser Seite können neue Karten zur Anwendung hinzugefügt werden. Über ein einfach gehaltenes Formular werden der Name und der Typ der Karte erfasst. Zusätzlich besteht die Möglichkeit, ein Bild zur Karte hochzuladen, wobei das Bildfeld optional ist.

Pflichtfelder wie Name und Typ sind entsprechend markiert und müssen ausgefüllt werden, bevor die Karte gespeichert werden kann. Mit einem Klick auf den Button „Add Card“ wird die neue Karte erstellt und der Nutzer zurück zur Kartenübersicht geleitet.

4. Erweiterungen

4.1. Navigationsleiste

Eine besondere Funktion dieser Leiste ist die dynamische Hervorhebung des aktuellen Pfads: Der aktive Menüpunkt wird automatisch fett und hell dargestellt, sodass der Nutzer jederzeit erkennt, in welchem Bereich er sich befindet.

Diese visuelle Rückmeldung verbessert die Orientierung in der Anwendung deutlich und sorgt für eine benutzerfreundliche Navigation – insbesondere bei grösseren Projekten mit mehreren Unterseiten. Die Umsetzung erfolgt technisch über einen Abgleich des aktuellen URL-Pfads mit dem jeweiligen Menüeintrag.

4.2. Bild Hochladen

Die Anwendung bietet beim Erstellen und Bearbeiten einer Karte die Möglichkeit, ein eigenes Kartenbild hochzuladen. Dies erfolgt über ein Eingabefeld mit dem Typ file, das den Upload einer Bilddatei vom lokalen Gerät erlaubt.

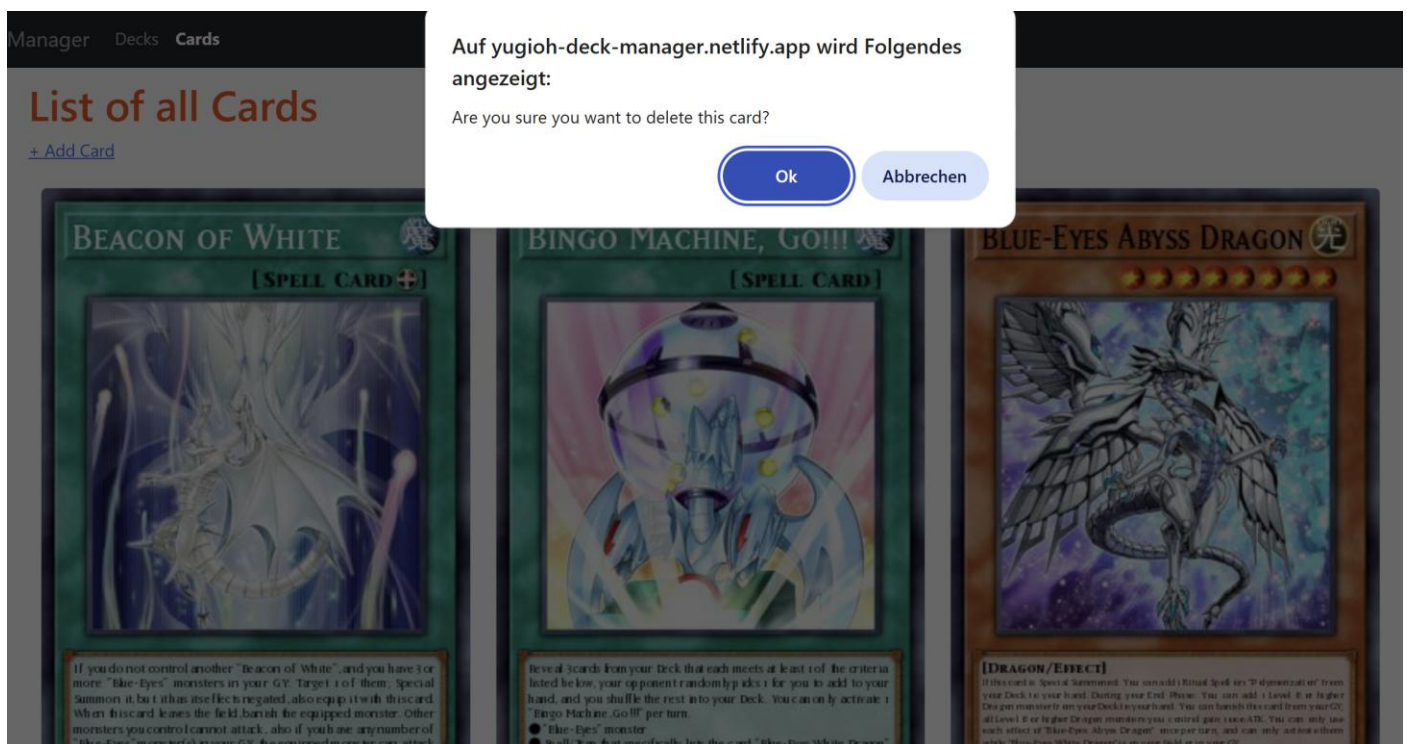
In der Bearbeitungsansicht wird das derzeit verwendete Bild unter „Current Image“ angezeigt. Darunter befindet sich das Feld „Upload New Image (optional)“, über das der Benutzer ein neues Bild auswählen kann. Falls kein neues Bild ausgewählt wird, bleibt das bestehende Bild erhalten.

Beim Erstellen einer neuen Karte ist dieses Bildfeld ebenfalls verfügbar. In dem Fall dient es dazu, das erste Bild der Karte zu definieren. Der Upload ist dabei optional, sodass die Karte auch ohne Bild angelegt werden kann (es wird dann ein Platzhalter verwendet).

Die hochgeladenen Bilder werden automatisch im Ordner /uploads gespeichert und sind direkt auf der Website sichtbar, sobald der Upload abgeschlossen ist.

4.3. Erweiterungen Update- und Delete button

4.3.1. Delete Dialog



Das Bild zeigt die Kartenübersicht mit einer eingeblendeten Bestätigungsdialogbox, die erscheint, sobald ein Nutzer versucht, eine Karte zu löschen. Im Vordergrund ist ein typischer JavaScript-Dialog zu sehen, der die Frage stellt:

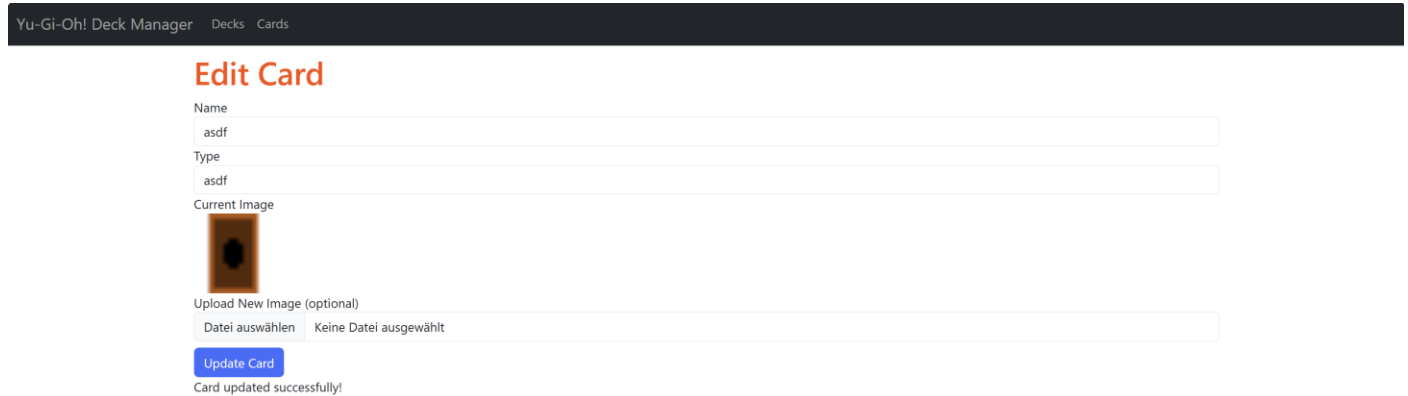
„Are you sure you want to delete this card?“

Dieser Dialog dient der Sicherheitsabfrage, damit Karten nicht versehentlich gelöscht werden. Der Nutzer hat zwei Optionen:

- „OK“: Die Löschung wird bestätigt und ausgeführt.
- „Abbrechen“: Die Aktion wird abgebrochen, und die Karte bleibt erhalten.

Oben im Dialog ist klar erkennbar, dass die Seite über Netlify bereitgestellt wurde („yugioh-deck-manager.netlify.app“). Im Hintergrund sieht man die Kartenliste leicht abgedunkelt, was darauf hinweist, dass die Seite im Modus der Benutzerinteraktion pausiert, bis der Nutzer sich für „OK“ oder „Abbrechen“ entscheidet.

4.4. Update und Create Meldung



Auf diesem Screenshot ist zu sehen, wie die Bearbeitung einer Karte mit einem neuen Bild durchgeführt wurde. Nachdem der Nutzer Änderungen am Namen oder Typ vorgenommen hat, kann zusätzlich über das Upload-Feld „Upload New Image (optional)“ ein neues Bild ausgewählt werden.

Sobald das Formular über den Button „Update Card“ abgeschickt wird, wird das Bild sofort aktualisiert und in der Datenbank ersetzt. Im Anschluss zeigt die Seite das neue Bild unter „Current Image“ direkt an, ohne dass ein zusätzlicher Schritt notwendig ist.

Der Hinweis „Card updated successfully!“ bestätigt, dass alle Änderungen einschliesslich des neuen Bilds erfolgreich übernommen wurden.

Ein anderer Hinweis für Bestätigung wird auch bei der Erstellung einer Karte angezeigt wie auf dem folgenden Bild.

Add Card
Card created