Prototyping Projektdokumentation

Name: Anastasija Gojkovic

E-Mail: gojkoana@students.zhaw.ch

URL der deployten Anwendung: https://vaultofgames.netlify.app/

# Einleitung

GameVault ist eine webbasierte Anwendung zur Verwaltung und Darstellung einer umfangreichen Spielesammlung. Die Idee hinter der App ist es, eine übersichtliche und benutzerfreundliche Plattform zu bieten, auf der Informationen zu vielen Videospielen abgerufen und verwaltet werden können. Die Anwendung richtet sich an Nutzer, die sich einen Überblick über verfügbare Spiele verschaffen wollen.

Zu den Grundfunktionen gehören:

* Eine Übersicht aller Spiele mit Bild, Genre, Plattform und Preis
* Eine Detailansicht pro Spiel mit zusätzlichen Informationen
* CRUD-Funktionen für Spiele
* Eine eigene Entwickler-Übersicht, inklusive Detailansicht und Verknüpfung mit den zugehörigen Spielen
* Pagination zur strukturierten Navigation durch grosse Datenmengen

# Datenmodell

# Ein Bild, das Entwurf, Diagramm, Zeichnung, Lineart enthält. KI-generierte Inhalte können fehlerhaft sein.Beschreibung der Anwendung

## Startseite und Navigation

Route: /

Ein Bild, das Text, Screenshot, Reklametafel, Himmel enthält.

KI-generierte Inhalte können fehlerhaft sein.

Die Startseite der Anwendung zeigt einen Willkommenstext und ein grosses Hintergrundbild. Sie dient als visuelles Intro zur Anwendung. Sie enthält keine interaktiven Funktionen, sondern leitet Benutzer weiter zu den Bereichen Spiele oder Entwickler über die Navigation. Die Navigation ist von jeder Seite aus ersichtlich. Sie ist auch ausgelagert als Komponente Navbar.svelte und wird im +layout.svelte eingebunden. Die Navigations-Leiste enthält Links zur Startseite (GameVault oder ‘Jetzt Endecken’, «/»), Spieleübersicht (Spiele, «/games»), Entwicklerübersicht (Entwickler, «/developers»).

Dateien:

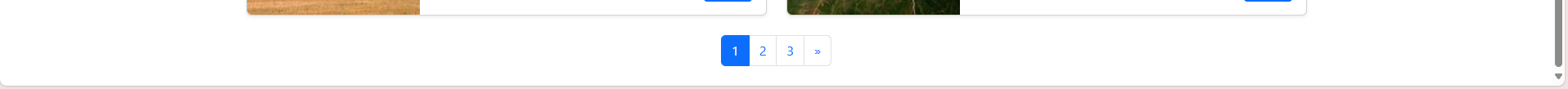
* lib/components/Navbar.svelte
* static/images/fallout\_wallpaper.jpg
* routes/+layout.svelte
* routes/+page.svelte

## Gameübersicht

Route: /games

Ein Bild, das Text, Screenshot, Software, Webseite enthält.

KI-generierte Inhalte können fehlerhaft sein.



Auf dieser Seite werden alle Spiele in einer Kartenbasierten Übersicht dargestellt. Die Darstellung erfolgt über die Komponente GameCard.svelte. Es werden jeweils 20 Spiele pro Seite angezeigt. Mithilfe der eingebauten Pagination kann durch verschiedene Seiten geblättert werden. Die Pagination ist als Komponente Pagination.svelte ausgelagert. Jede Karte enthält ein Bild (Platzhalterbild von <https://picsum.photos>), den Namen, Genre, Plattform, Preis sowie die durchschnittliche Spielzeit. Über den Button «Details» gelangt man auf die Detailansicht eines Spiels unter /games/[appid].   
Oben rechts befindet sich ein Button «Spiel hinzufügen», der zum Erstellungsformular unter /games/create weiterleitet.

Dateien:

* lib/components/GameCard.svelte
* lib/components/Pagination.svelte
* lib/server/games.js (Datenbankanbindung)
* routes/games/+page.svelte
* routes/games/+page.server.js

## Spiel hinzufügen

Route: /games/create

Auf dieser Seite kann ein neues Spiel erstellt werden. Das Formular ist vollständig clientseitig validiert und umfasst alle relevanten Felder eines Spiels. Die Eingabefelder für Plattformen und Genres sind als kommagetrennte Texteingabe gelöst, welche serverseitig in Arrays umgewandelt werden. Pflichtfelder sind App ID, Name, Entwickler ID und Preis. Nach erfolgreicher Speicherung wird der Benutzer zurück auf die Spieleübersicht geleitet.

Dateien:

* routes/games/create/+page.svelte
* routes/games/create/+page.server.js
* lib/server/games.js

# 

Auf dieser Seite werden alle Filme aufgelistet. Für die Darstellung der einzelnen Filme wird die Komponente MovieCard.svelte verwendet. Jeder Film kann zur Watchlist hinzugefügt oder von dieser entfernt werden. Per Checkbox «Show only movies on watchlist» können die Filme ausgeblendet werden, die nicht auf der Watchlist stehen. Über den Button «Add New Movie» gelangt man zur Seite /movies/create.

Dateien:

* lib/components/MovieCard.svelte
* routes/movies/+page.svelte
* routes/movies/+page.server.js

# Erweiterungen

*[Liste der Erweiterungen, die über die Grundanforderungen hinausgehen, siehe Punkt «Erweiterungen» in der Aufgabenstellung. Achten Sie auf Vollständigkeit.* ***Es werden nur die hier beschriebenen Erweiterungen bewertet****. Der Code muss grundsätzlich nicht beschrieben werden, aber es sollten Hinweise gegeben werden, wo die Erweiterung implementiert wurde (Dateiname, Funktionsname, etc.). Screenshots sind nur nötig, falls die Erweiterung nicht unter «Beschreibung der Anwendung» beschrieben wurde.]*

*Beispiel:*

## Watchlist-Filter

Auf der Seite /movies können via Checkbox Filme ausgeblendet werden, die nicht auf der Watchlist stehen. Dies wurde mit einem Formular und einer MongoDB-Query in der Funktion getMoviesOnWatchlist() in der Datei lib/db.js realisiert.

Datei(en):

* lib/db.js
* routes/movies/+page.svelte
* routes/movies/+page.server.js