

Dokumentation Programmierprojekt SS 16

Arbeitsbereich: Kommunikationsnetze

Finn Ickler, Steffen Lindner, Dominik Spoerle, Maximillian Pfister
Betreuer: Wolfgang Braun

24. Juli 2016

Inhaltsverzeichnis

1 Beschreibung der Software	4
1.1 Grundstruktur	4
1.1.1 Router	4
1.1.2 Main Controller	5
1.1.3 View	6
1.1.4 Model	6
1.1.5 Index-Datei	6
1.2 Core Klassen	7
1.2.1 Notify Klasse	7
1.2.2 Config Klasse	7
1.2.3 RestApi Klasse	8
1.3 Models	8
1.3.1 User	8
1.3.2 Kategorie	9
1.3.3 Video	10
1.3.4 Playlist	12
1.4 Umgesetzte Funktionalitäten	12
1.5 Verwendete Drittsoftware	13
2 Installationsanleitung	14
2.0.1 Systemanforderungen	14
2.0.2 Konfiguration	14
2.0.3 Installation	14
Apache	14
Nginx	15
2.0.4 Inbetriebnahme	15
3 Bedienungsanleitung	16
3.1 Kurzbeschreibung der Software	16
3.2 Hinweise zum ersten Start	16
3.3 Verfügbare Funktionen in Abhängigkeit der Benutzerrechte	16
3.3.1 Grundstruktur	16
Header	17
Content	18
Footer	18
Navigation auf der Seite	18
Navigationsmittel	18
3.3.2 Funktionen	19
Header Funktionen	19
Profil erstellen	19
Anmeldung	19
Suchfunktion	19
Profil	20
Videos verwalten	21
Videos ansehen	21
Video hochladen	22
Video bearbeiten	24
Playlists verwalten	25
Playlist Übersicht	25
Videos zu Playlist hinzufügen	28
Playlist Detailseite	29
Admin Funktionen	32

Nutzerverwaltung	32
Playlistverwaltung	32
Kategorienverwaltung	33
3.4 Probleme und Lösungen	33
3.5 Besonderheiten von WolfGang	34
3.5.1 Die Idee WolfGang	34
3.5.2 Navigation und Overlay Techniken	35
3.5.3 Wolf-Design	35
3.5.4 Tag Cloud Kategorien	35
3.5.5 RoboHash: automatische Benutzerbilder	35
3.5.6 Typeahead Suchfunktion	35

Beschreibung der Software

1.1 Grundstruktur

Die folgende Grafik illustriert das Grundsyste m der Software.

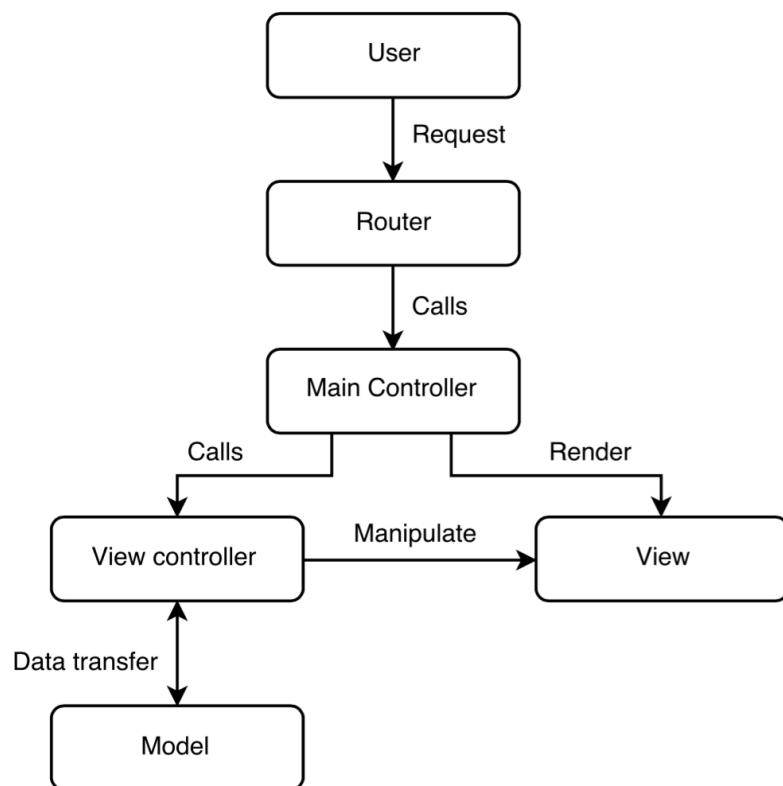


Abbildung 1.1: Software Grundstruktur

Das Grundsyste m ist nach dem MVC-Prinzip¹ gestaltet.

Dabei ist das System aufgespalten in verschiedene Templates (Views), die von ihrem zugehörigen Controller (View Controller) manipuliert werden. Der View Controller nutzt das Model um die nötigen Daten vom Server zu erhalten.

Das Model ist (nach dem MVC-Prinzip) für die Datenhaltung zuständig.

Anmerkung: Das komplette Grundsyste m (Struktur, Core-Klassen) wurde von Steffen Lindner implementiert / entwickelt.

1.1.1 Router

Der Router wird genutzt, um ein URL-Template matching durchzuführen.

Es lassen sich dadurch sehr leicht 'Pfade' entsprechenden Templates zuordnen. Dadurch sehen die URL's nicht nur schöner aus, es hat auch SEO² Vorteile.

¹MVC-Prinzip: Model View-Controller Abspaltung zur Strukturierung von Software

²SEO: Suchmaschinennoptimierung

```
Route::add("/home", function() {Controller::dispatch("home");});
```

Abbildung 1.2: Beispielhafter Announce für eine Route

Sobald die Route '/home' aufgerufen wird (beispielsweise: wolfgang.ne4y.de/home), wird die hinterlegte Funktion ausgeführt (in diesem Fall Controller::dispatch('home'));

Dabei wird über ein Patternmatching die aktuelle **Requested-URI**³ mit den 'angekündigten' Pfaden verglichen. Stimmt ein Pfad überein, wird die hinterlegte Funktion ausgeführt.

Der Router nutzt außerdem einen Eintrag der Config-Klasse (siehe dazu Abschnitt zur Config-Klasse). Dort wird festgelegt, wie der Basepath aussieht. Liegt das System beispielsweise unter wolfgang.ne4y.de/-meinDir/, so ist der Basepath meinDir. Dazu wird folgender Eintrag in einer Datei im config Ordner gesetzt:

```
Config::set('basepath', 'meinDir');
```

Abbildung 1.3: Beispiel Basepath für die Router-Klasse

1.1.2 Main Controller

Der Main-Controller implementiert die grundlegenden Steuerfunktionen unserer Software. Dazu zählt das Rendern der Templates, das Aufrufen der entsprechenden Template-Controller sowie das Umleiten des Users.

Er gibt außerdem die Möglichkeit HTTP-Statuscodes an den Nutzer zurückzugeben.

Der Main Controller wird in den meisten Fällen direkt vom Router aufgerufen. Er implementiert folgende Methoden / Attribute:

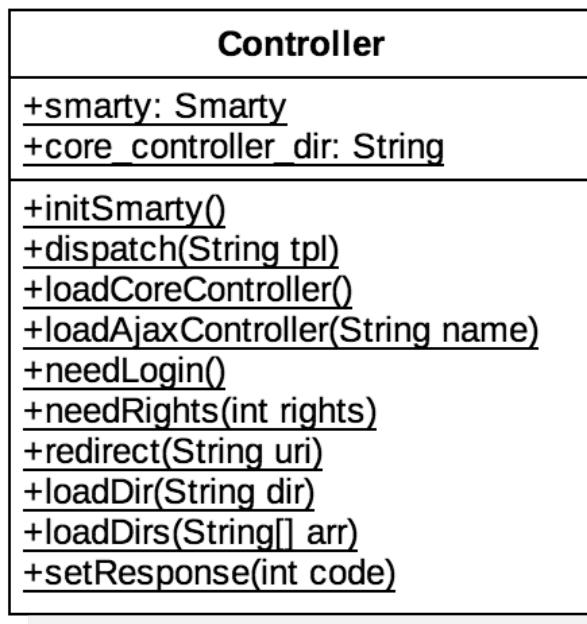


Abbildung 1.4: Main-Controller

Dabei sind die wichtigsten Funktionen das Initialisieren eines Smarty Objects (initSmarty()), sowie das Dispatchen eines Templates (dispatch(String tpl), wird vom Router aufgerufen). Er hält auch das Smarty-Objekt gespeichert, über das die Variablen für die Templates gesetzt werden.

Während des Dispatchprozesses wird der zugehörige Templatecontroller geladen.

³Requested-URI: Angefragte URL Zeile, die vom Webserver an den PHP-Interpreter weitergegeben wird

```
Controller::$smarty->assign("myVar", "Das_ist_meine_Variable")
```

Abbildung 1.5: Der Maincontroller dient dazu Smartyfunktionalität zu nutzen

1.1.3 View

Der View wird durch ein Templatingsystem umgesetzt. Dafür verwendet unsere Software die Templateengine Smarty⁴.

Smarty wird genutzt, um den HTML-Code von der Businesslogik zu trennen.

Dabei wird eine spezielle Templatesprache genutzt, die einem stark vereinfachtem PHP nahe kommt. Damit soll gewährleistet werden, dass im Template keine Businesslogik implementiert wird, um die klare Abtrennung der View vom Controller zu gewährleisten.

Ein kleines Beispiel soll die Interaktion von Controller und View verdeutlichen:

```
Controller::$smarty->assign("Author", "Steffen_Lindner");
```

Abbildung 1.6: Beispielcontroller

```
<p>Name des Autors: {$Author}</p>
```

Abbildung 1.7: Beispielview

Dadurch wird die gesamte Logik in den Controller verlagert und im Template lediglich durch einen 'Platzhalter' ersetzt.

Smarty erlaubt auch einfache logische Konstrukte wie

- If / Else / Elseif
- For / While / Foreach
- Isset / Empty

und viele Weitere.

1.1.4 Model

Die Models werden zur Datenhaltung genutzt. Da wir keinen direkten Datenbankzugriff haben, sondern lediglich über eine Rest-API mit dem Server kommunizieren führen die Model die meisten Rest-Calls aus. Das gewährleistet eine klarere Trennung des Codes nach dem MVC-Prinzip.

Dazu wurde eine separate Rest-Klasse geschrieben, die die Kommunikation mit dem Server verwaltet (dazu mehr im speziellen Abschnitt).

Die Models implementieren nun lediglich die speziellen Rest-Aufrufe, sowie einfache Verwaltungsaufgaben.

Eine genauere Beschreibung der Models folgt in den späteren Abschnitten.

1.1.5 Index-Datei

Die Index-Datei (index.php) ist der zentrale Ausgangspunkt unserer Software.

Alle Anfragen die der User stellt werden durch eine .htaccess-Datei unter Apache bzw. einer speziellen nginx-Konfiguration an die Index-Datei weitergeleitet. Dort lädt zunächst der Main-Controller alle Klassen (Core- und 'normale' Klassen), sowie alle Models (dazu später mehr). In der Index-Datei werden außerdem alle Routen aus dem Routen-Ordner geladen (angegeben durch einen Eintrag in der Config-Datei, siehe dazu den Abschnitt zur Config-Klasse).

Schließlich wird dort der User initialisiert (es wird erkannt, ob der Nutzer eingeloggt ist, was für Rechte er hat usw.) und der Routingprozess gestartet (und somit das angefragte Template samt zugehörigem Controller geladen).

⁴Smarty: Templatingengine, mehr Informationen auf: <http://smarty.net>

1.2 Core Klassen

Unsere Software ist eingeteilt in 'normale' Klassen und Core Klassen. Die Core Klassen implementieren allgemeinere Funktionen, die von der gesamten Software genutzt werden.

Zu den Core Klassen gehören:

- Notify Klasse
- Controller Klasse (Main Controller, bereits beschrieben)
- Config Klasse
- RestApi Klasse
- Route Klasse (bereits beschrieben)

So gut wie alle Core-Klassen sind statisch implementiert. Das heißt, es wird kein Objekt der einzelnen Klassen erzeugt. Das hat zum Einen den Grund, dass sie von der 'Logik' her keinen richtigen Zustand haben. Zum Andern ist es aus programmietechnischer Sicht einfacher, bzw. praktischer, direkt auf die Funktionen zugreifen zu können, ohne auf ein Objekt angewiesen zu sein.

Die Models hingegen sind objektorientiert implementiert, d.h. um sie richtig nutzen zu können müssen zunächst Objekte erstellt werden.

Einige der Core-Klassen sollen im Folgenden etwas näher beschrieben werden.

1.2.1 Notify Klasse

Die Notify Klasse wird genutzt, um Meldungen an den Nutzer auszugeben.

Notify
<u>-notify: Array</u>
<u>+addNotify(String msg)</u>
<u>+getNotify(): return Array</u>

Abbildung 1.8: Notify Klasse

Der Main-Controller liest nach dem Abarbeiten der View-Controller die erstellten Nachrichten aus und speichert sie in einer Smarty-Variable, die anschließend vom Template ausgegeben wird. Dadurch können die View-Controller unabhängig voneinander Nachrichten an den Nutzer zurückgeben, ohne sich selbst um die Darstellung kümmern zu müssen.

1.2.2 Config Klasse

Die Config Klasse wird genutzt, um Konstanten anzulegen, wie beispielsweise der Basepath des Systems, wie viele Kategorien pro Seite angezeigt werden sollen o.ä.

Config
<u>-registry: Array</u>
<u>+set(String key, Mixed value)</u>
<u>+get(String key): return Mixed</u>
<u>+loadConfigs()</u>

Abbildung 1.9: Config Klasse

Es handelt sich dabei um einen einfachen Key-Value Store. Aufgrund der schwachen Typisierung von PHP kann jede Art von Value gespeichert werden.

1.2.3 RestApi Klasse

Da die gesamte Kommunikation mit dem Server über eine Rest-API⁵ abläuft, wurde eine zentrale Klasse geschrieben, die die Kommunikation mit dem Server erleichtern soll.

Die RestApi-Klasse implementiert dabei folgende Methoden:

RestApi
+String apiUrl
+get(String resource): return Resource
+post(String resource, Array body): return Resource
+patch(String resource, Array body): return Resource
+put(String resource, Array body): return Resource
+putEmpty(String resource): return Resource
+delete(String resource, Array body): return Resource
+deleteEmpty(String resource): return Resource
+formUpload(String path, String name, String mime, String description)

Abbildung 1.10: RestApi Klasse

Es können die grundlegenden HTTP-Requests get, post, patch, put, delete ausgeführt werden. Dafür wurde die PHP-Implementierung von cURL⁶ verwendet.

Beispielhafte Verwendung der RestApi-Klasse:

```
Last login: Fri Jul  8 22:34:48 on ttys002
Steffen@Steffens-MBP:~/ownCloud/Projects/www/ProgrammierProjektSS16\:$ php -a
Interactive shell

php > include("model/User.php");
php > User::initUser();
php > include("libs/Core/RestApi.php");
php > var_dump(json_decode(RestApi::get("/videos?limit=1")));
object(stdClass)#2 (7) {
    ["total"]=>
        string(3) "132"
    ["items"]=>
        array(1) {
            [0]=>
                object(stdClass)#3 (11) {
                    ["id"]=>
                        string(1) "4"
                    ...
                }
        }
}
php >
```

Abbildung 1.11: Beispielhafte Verwendung der RestApi-Klasse

(die Ausgabe wurde aus Platzgründen abgeschnitten).

Die RestApi-Klasse ist damit die Schnittstelle zwischen unserem System und dem zur Verfügung gestellten Backendsystem.

1.3 Models

1.3.1 User

Das User-Model verwaltet alle Daten des Users. Dazu zählt beispielsweise die UserID, die Rechte des Users, Name, Email etc.

⁵Rest: Representational State Transfer, Programmparadigma für verteilte Systeme

⁶cURL: CURL Url Request Library, Bibliothek zum Ausführen von HTTP-Requests

Das User-Model implementiert folgende Funktionalitäten:

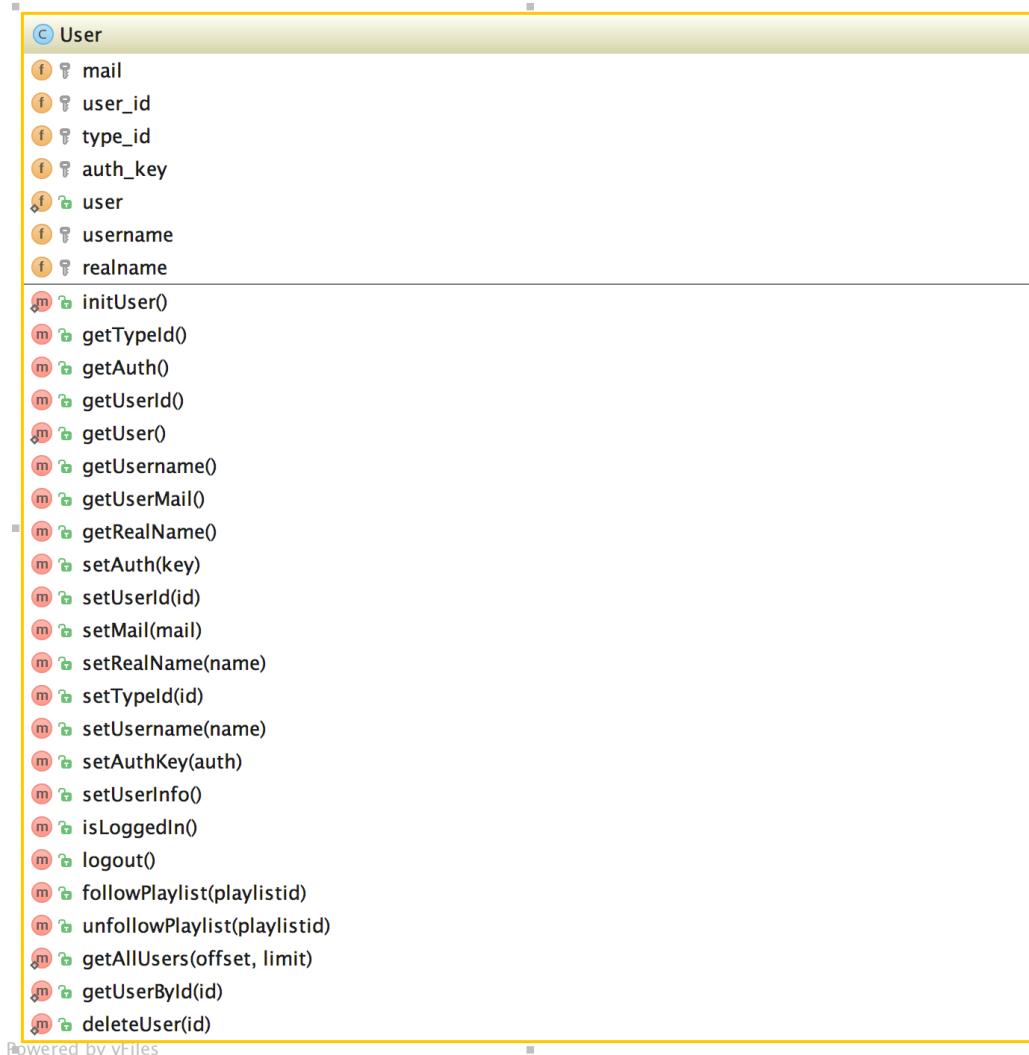


Abbildung 1.12: User Model

Das User-Model ist aus programmiertechnischer Sicht eine Mischung aus statischer und objektorientierter Implementierung. Um das User Objekt nicht durch alle Seiten zu 'tragen' wird zunächst in der Index-Datei der User initialisiert. Dabei wird ein neues (statisches) User Objekt erstellt und ggf. mit vorhandenen Daten (aus Sessions) initialisiert (notwendig, damit ein Nutzer eingeloggt bleibt und nicht bei jedem Seitenaufruf die Userdaten erneut vom Server abgefragt werden müssen).

Ab diesem Moment kann man in jedem Controller das aktuelle Userobjekt erhalten, um anschließend objektspezifische Operationen auszuführen.

```
User::getUser() -> getUserMail();
```

Abbildung 1.13: Beispielaufruf, um die Mail des aktuellen Benutzers zu erhalten

Innerhalb des User-Models sind die oben genannten Funktionalitäten implementiert, wobei die Kommunikation mit dem Backendsystem über die bereits beschriebene RestApi Klasse abläuft.

Anmerkun: Das User-Model wurde größtenteils von Steffen Lindner implementiert.

1.3.2 Kategorie

Das Kategorien Model verwaltet die Daten für eine Kategorie bzw. allgemeine Funktionen die Kategorien betreffen. Dabei wurden alle Methoden, die nicht speziell einer Kategorie zugeordnet werden (wie beispielsweise das Suchen nach Kategorien bezüglich eines Namens) statisch implementiert.

Aufgrund der unterschiedlichen Programmierer kommt es allerdings vor, dass Methoden, die eigentlich einer speziellen Kategorie zugeordnet werden können (wie beispielsweise das Hinzufügen eines Videos zu einer Kategorie) ebenfalls statisch implementiert wurden.

Das Kategorie-Model implementiert folgende Methoden:

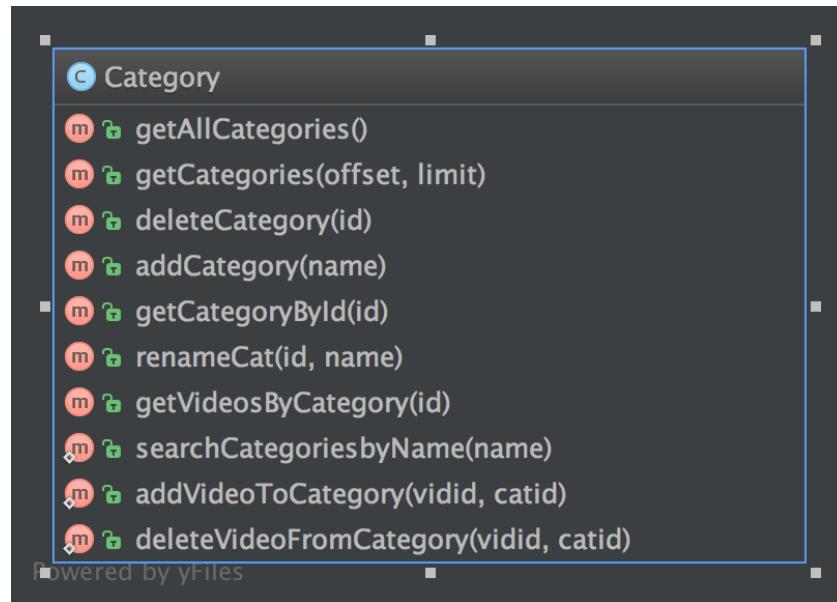


Abbildung 1.14: Kategorie Model

Anmerkung: Das Kategorien-Model wurde (größtenteils) von Steffen Lindner (erster Teil) und Maximilian Pfister (zweiter Teil) implementiert.

1.3.3 Video

Das Video Model, enthält zum einen alle Metadaten eines Videos und zum anderen statische Methoden um auf eine bestimmte Anzahl Videos zuzugreifen, sowie Kommentare, Ratings und Videos hinzufügen.

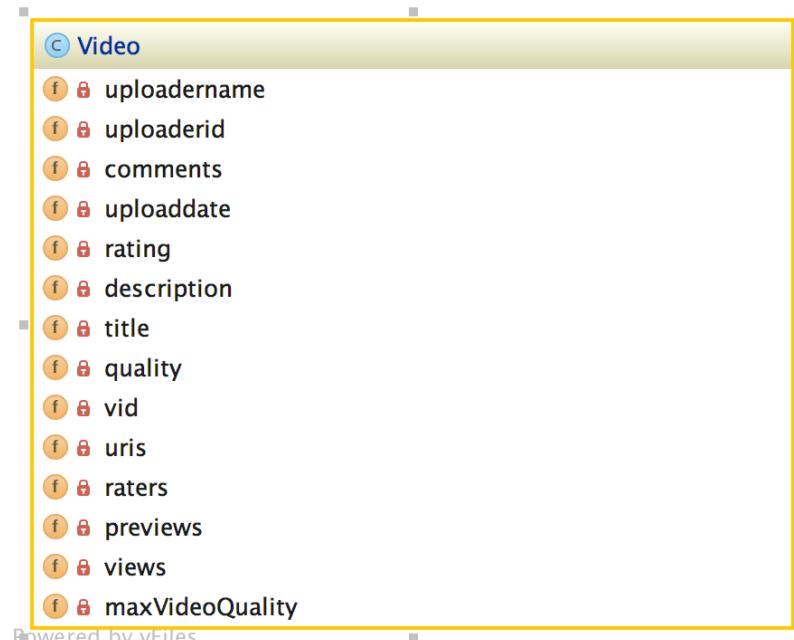


Abbildung 1.15: Metadaten eines Videos

Beim erstellen eines Video Objektes durch übergeben einer ID wird durch den Konstruktor der Klasse automatisch alle Metadaten durch ein REST-Call gesetzt.

```
public function __construct($id){
    $ret = json_decode(RestAPI::get("/videos/".$id, array("id" => $id)));

    if (isset($ret->error)){
        Controller::redirect("/404");
    }
    else{
        $this->views = $ret->views;
        $this->vid = $ret->id;
        $this->title = ($ret->title) ? 'Kein_Titel' : $ret->title;
        $this->description = $ret->description;
        $this->uploaderid = $ret->uploaderid;
        $this->uploadername = $ret->uploadername;
        $this->uploaddate = $ret->uploaddate;
        $this->rating = ($ret->rating) ? 0 : $ret->rating;
        $this->uris = $ret->uri;
        $this->previews = $ret->previews;
        $this->comments = $ret->comments;
        $this->raters = $ret->ratings->items;
        $this->maxVideoQuality = $ret->maxVideoQuality;
    }
}
```

Das sind alle Metadaten die der Backend Server liefert.

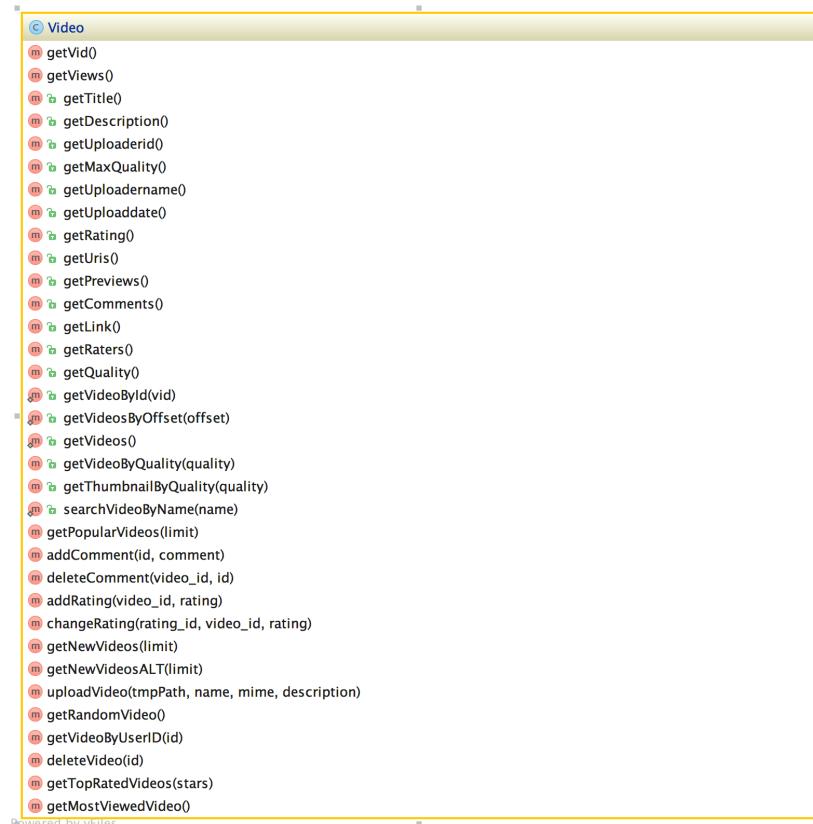


Abbildung 1.16: Funktionen der Video Klasse

Neben den trivialen Getter Methoden wird in der Video Klasse noch Funktionalitäten rund um das Video als statische Methoden implementiert.

`getTopRatedVideos`, `getRandomVideo`, `getNewVideos`, `getVideoByUserID` und `getPopularVideos` geben durch gezielte Suche bestimmte Arten von Videos zurück.

`addComment`, `deleteComment`, `addRating`, `uploadVideo`, `deleteVideo`, `searchVideoByName` und `changeRating` stehen als Schnittstellen zwischen ihren jeweiligen AJAX Implementationen und dem Backend.

Listing 1.1: Exemplarische Implementation einer Methode des Video Models

```
public static function getPopularVideos($limit){  
    $ret = RestAPI::get("/search?type=popularvideos&limit=" . $limit); //REST Call  
    return json_decode($ret); //Als PHP JSON (stdClass) Objekt zurückgeben  
}
```

Anmerkung: Die Klassenstruktur und die nicht statischen Methoden wurden von Finn Ickler implementiert. Die statischen Methoden vom Rest des Teams.

1.3.4 Playlist

Das Playlist Model verwaltet die Daten für eine Playlist bzw. allgemeine Funktionen die Playlisten betreffen. Da die Funktionen, die Playlisten betreffen, auch in der Suchfunktion oder z.Bsp. in der Video Wiedergabe genutzt werden, wurden diese alle statisch implementiert.

Das Playlist-Model implementiert folgende Methoden:

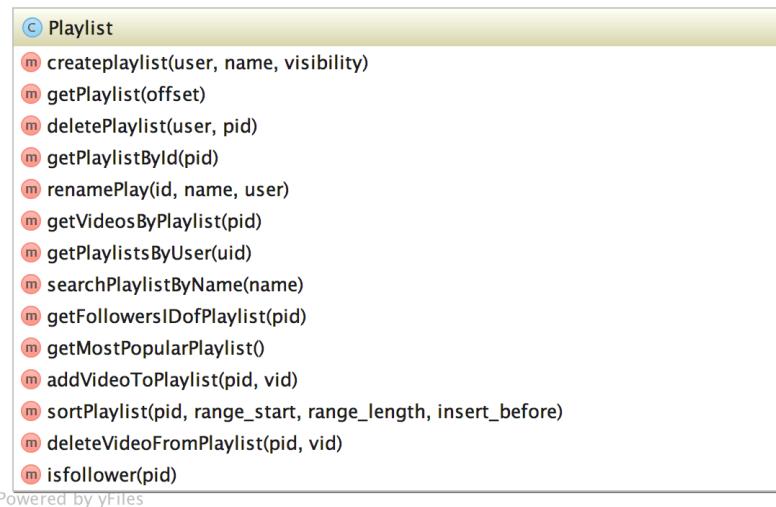


Abbildung 1.17: Kategorie Model

Anmerkung: Das Playlist-Model wurde von Dominik Spoerle, Finn Ickler und Maximillian Pfister (zweiter Teil) implementiert.

1.4 Umgesetzte Funktionalitäten

Das System implementiert folgende Funktionalitäten:

- Login (Steffen Lindner)
- Registrierung (Maximillian Pfister)
- Profil (ansehen / bearbeiten) (Steffen Lindner)
- Kategorienverwaltung (anzeigen, anlegen, bearbeiten) (Steffen Lindner)
- Playlistverwaltung (Dominik Spoerle)
- Playlistumsortierung (Steffen Lindner)
- Videosuche & Ergebnisanzeige (Finn Ickler)
- Videoansicht (Finn Ickler)
- Videos kommentieren / bewerten (Steffen Lindner)
- Video upload (Finn Ickler)
- Neuste Videos / Favoriten (Dominik Spoerle)
- Playlist abonieren / deabonieren, Videos zu Kategorien hinzufügen / entfernen (Maximillian Pfister)
- Rechtemanagment (Steffen Lindner & Finn Ickler)

1.5 Verwendete Drittsoftware

- Smarty PHP Template Engine
- JQuery
- Bootstrap Framework
- RoboHash
- VideoJS
- sortable.js
- fileinput.js

Installationsanleitung

2.0.1 Systemanforderungen

Unsere Software benötigt folgende Komponenten:

- Apache Version 2.0 (mit aktiviertem mod_rewrite) oder höher bzw. nginx 0.8.42 oder höher oder einen anderen Webserver der URL rewriting unterstützt
- PHP 5.2 oder höher (für Smarty)
- PHP-Erweiterung cURL

2.0.2 Konfiguration

Jegliche (vom Nutzer vornehmbare) Konfiguration wird in der *mainConfig.php* innerhalb des *config* Verzeichnisses vorgenommen.

Dabei ist besonders der bereits angesprochene *basepath* für den Nutzer interessant. Jenachdem, ob unsere Software in einem Unterverzeichnis aufgerufen wird (beispielsweise www.meineDomain.de/unterverzeichnis/) oder nicht, muss der *basepath* entsprechend angepasst werden.

```
Config::set('basepath', 'unterverzeichnis');
```

Abbildung 2.1: Beispiel Basepath zur Konfiguration unserer Software

Innerhalb der *mainConfig.php* lassen sich noch weitere Parameter einstellen, wie beispielsweise das Verzeichnis, in dem die Routen liegen (für das URL - Template-Matching), oder auch wie viele Kategorien / User pro Seite angezeigt werden sollen.

2.0.3 Installation

Die Installation gestaltet sich sehr einfach. Nachdem die Dateien in das gewünschte Verzeichnis kopiert wurden muss lediglich noch die Konfiguration des *basepaths* durchgeführt werden (siehe Abschnitt Konfiguration).

Damit alle Anfragen auch an die *index.php* weitergeleitet werden bedarf es allerdings noch einer Anpassung, die nun exemplarisch für Apache und nginx (unter unixoiden Systemen) beschrieben werden soll.

Apache

Zunächst muss geprüft werden, ob *mod_rewrite* geladen wurde. Dazu kann das folgende bash Kommando genutzt werden:

```
apachectl -M | grep 'rewrite_module'
```

Abbildung 2.2: Überprüfung ob *mod_rewrite* geladen wurde

Da es sich bei *mod_rewrite* um ein ASF¹ Modul handelt, ist es in der Regel automatisch installiert. Wird es nicht gefunden, so muss es für gewöhnlich lediglich aktiviert werden:

¹ASF: Apache software foundation: Standardmodul

```
a2enmod rewrite
```

Abbildung 2.3: Aktivierung von *mod_rewrite*

Nun muss in der httpd.conf (bzw. einer geladenen Konfiguration von Apache) für das Verzeichnis, in dem unsere Software liegt, das Umschreiben der URL erlaubt werden. Dazu genügt ein Eintrag wie folgt:

```
<Directory "/var/www/unterverzeichnis/">
    AllowOverride All
</Directory>
```

Abbildung 2.4: Erlaube das Umschreiben der URL für ein gewünschtes Verzeichnis

Anschließend muss in der .htaccess-Datei im Wunschverzeichnis die *RewriteBase* angepasst werden. Für das gegebene Beispiel würde folgen:

```
RewriteBase /unverzeichnis
```

Abbildung 2.5: Anpassung der RewriteBase in der .htaccess

Nginx

Wird nginx verwendet, so muss das '*ngx_http_rewrite_module*'² installiert sein. Anschließend muss in der nginx.conf folgender Eintrag ergänzt werden:

```
location /unterverzeichnis {
    alias /srv/ss16-web/htdocs;
    index index.php;
    try_files $uri $uri/ /ss16-web/trunk/index.php?$query_string;
}
```

Abbildung 2.6: Beispielhafte nginx.conf zur URL Umleitung

wobei die Pfade nach der tatsächlichen Struktur anzupassen sind.

2.0.4 Inbetriebnahme

Ab diesem Moment ist unsere Software einsatzbereit.

²NGINX Rewrite: http://nginx.org/en/docs/http/ngx_http_rewrite_module.html

Bedienungsanleitung

3.1 Kurzbeschreibung der Software

WolfGang ist eine Video-Streaming-Dienst, der im Rahmen des Software-Projekts 2016 an der Universität Tübingen entwickelt wurde. Das Videoportal soll den Benutzern das kostenloses Anschauen, Verwalten und Hochladen von Videos ermöglichen.

3.2 Hinweise zum ersten Start

Um alle Funktionen der Website nutzen zu können (Drop-Down-Menü für die Navigation bzw. für den Login), muss **Javascript** aktiviert sein.

Um alle Features unserer Seite zu erhalten, sollten sie sich bei ihrem ersten Besuch einen Account anlegen und sich mit diesem Anmelden. Dann können sie eigene Playlisten erstellen und Videos bewerten und kommentieren.

Falls sie sich gut verhalten, werden sie womöglich zu einem privilegierten Benutzer befördert und sie können Videos auf die Seite hochladen und dieses verwalten.

3.3 Verfügbare Funktionen in Abhängigkeit der Benutzerrechte

Die Benutzer der Website werden in Gruppen in Abhängigkeit ihrer Rechte eingeteilt.
Gäste, die keinen Account besitzen, sind User der Gruppe 1.

Ein eingeloggter Benutzer gehört zur Gruppe 2.

Von einem Admin privilegierte Benutzer gehören zur Gruppe 3,
die Admins bilden die Gruppe 4.

Daraus ergeben sich für die verschiedenen Funktionen der Website folgende Berechtigungen:

Usergruppe	1	2	3	4
Account erstellen	✓			
Video,Playlist abspielen	✓	✓	✓	✓
Playlist erstellen		✓	✓	✓
Playlist bearbeiten	(✓)	(✓)		✓
Playlist löschen	(✓)	(✓)		✓
Video bewerten und kommentieren		✓	✓	✓
Profil bearbeiten	(✓)	(✓)	(✓)	
Video hochladen		✓	✓	
Video zu Kategorie hinzufügen		(✓)	✓	
Video von Kategorie löschen		(✓)	(✓)	
Benutzer löschen				✓
Benutzer-Rechte verwalten				✓
Kategorie erstellen,bearbeiten,löschen				✓

Tabelle 3.1: Verfügbare Funktionen in Abhängigkeit von den Benutzerrechten

Eine eingeschränkte Berechtigung (✓) bedeutet,dass die Aktion nur für eigene Objekte”verfügbar ist.
Dies sind das eigene Profil, eigene Playlists oder eigene Videos.

3.3.1 Grundstruktur

Die Website ist in 3 Bereiche unterteilt:

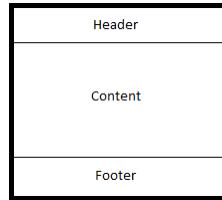


Abbildung 3.1: Grundstruktur

Der Header und der Footer sind dabei fest, der Content unterscheidet sich in Abhängigkeit der besuchten Unterseite.

Im weiteren werden die einzelnen Bereiche erklärt - möchten sie etwas genaueres über eine gewisse Funktion wissen, drücken sie hier

Header

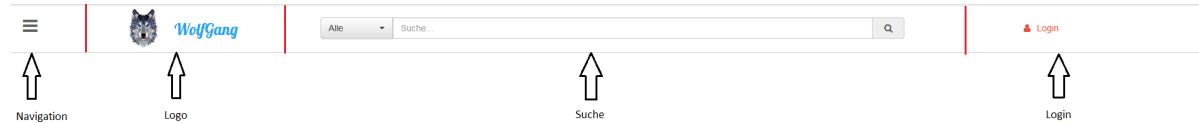


Abbildung 3.2: Header-Unterteilung

1. Navigation

Die Navigation ist ein Drop-Down-Menü und enthält eine Verlinkung auf alle wichtigen Seiten (Dabei werden nur die Seiten angezeigt, die der Benutzer auch besuchen kann).

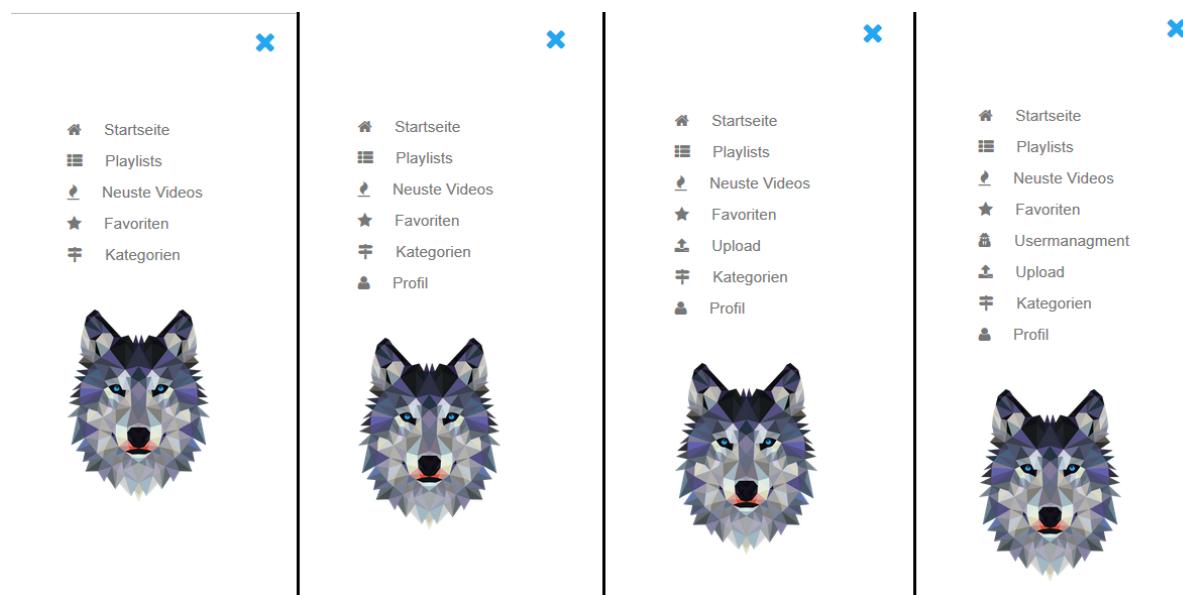


Abbildung 3.3: Ansicht der Navigationen mit verschiedenen Rechten (v.l. 1,2,3,4)

Versucht man, eine Seite über eine Manipulation der URL zu besuchen, für die man keine Berechtigung besitzt, so findet eine Weiterleitung zur Fehlerseite statt.

2. Schriftzug und Logo

Das Logo, dass natürlich auf keiner Website fehlen darf, ist gleichzeitig noch ein Link zur Startseite.

3. Suche

Im Suchfeld kann nach einem Video, einer Playlist oder einer Kategorie gesucht werden. Näheres wird im Unterpunkt Video suchen erläutert.

4. **Login** Der Log-In Bereich enthält Informationen darüber, ob man angemeldet ist und einen Button, mit dem man sich ggf. anmelden kann. Näheres zur Log-In Funktion finden sie unter Wie melde ich mich an?

Content

Der Content ist abhängig von der besuchten Unterseite und stellt zum Beispiel die Videoansicht, die Profilverwaltung oder die Playlistverwaltung dar.

Footer

Der Footer bietet neben dem farblichen Kontrast zur Seite einen Link zum Impressum (hierfür wurde das offizielle Impressum der Uni-Tübingen verwendet), einen Link zur Dokumentation der Software sowie einen Link zu den Social-Medias von **Wolfgang**.

Des Weiteren gibt es auf Unterseiten, welche durch die Darstellung vieler Videos eine große Höhe besitzen, einen Top-Button, der zum Anfang(nach oben) "zurückspringt".

Statusinformationen (wie Fehlermeldungen oder ähnlichem) des Servers werden rechts unten als Inhalt einer Notify-Box angezeigt, die nach einigen Sekunden wieder verschwindet.

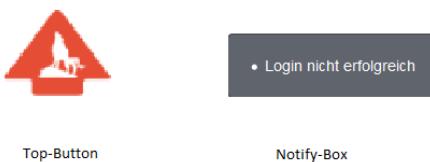


Abbildung 3.4: Top-Button und Notify-Box am Beispiel des fehlgeschlagenen Log-Ins

Navigation auf der Seite

Verfügbare Seiten und deren Inhalte:

- a) Startseite: Meist gesehene Videos (im Headerbereich), Zufälliges Video, Video of the Week, Kategorien Tag Cloud
- b) Playlists: Neuste Playlist, Top Rated Playlist, eigene Playlisten, abonnierte Playlists, alle Playlisten
- c) Neuste Videos: neuste Videos, meist gesehenes Video, alle Videos
- d) Favoriten: Populäre Videos, meist gesehenes Video, Videos mit bestem Rating, alle Videos
- e) Usermanagement: Übersicht aller User, Rechteänderungen von Usern, Löschen von Usern
- f) Upload: Video Upload
- g) Kategorien: Übersicht Kategorien, Kategorie und enthaltene Videos
- h) Profil: Nutzerprofil - Änderungen von Nutzerdaten, Übersicht eigene Videos, eigene Playlists

Navigationsmittel

- a) Hauptmenü links: kann per Klick auf das Menü-Icon links oben geöffnet werden
- b) Suchfunktion (suchen nach Videos, Playlisten, Kategorien)
- c) Logo: Link zur Startseite
- d) Video- und Playlisten-Play-Buttons: führen zu entsprechenden Detailseiten
- e) auf einer Seite: Top-Button/ Pfeil rechts unten, scrollt zum Seitenanfang

The diagram illustrates the user flow for account creation and login. It starts with a 'Login' button, which leads to a 'Login' form containing fields for 'Username' and 'Passwort'. Below the form is a 'Login' button. To the right of the form is a 'Registrieren' (Register) link. This link leads to a 'Registration' form with fields for 'Benutzername', 'Richtiger Name', 'Email', 'Passwort', and 'Passwort wiederholen'. A 'Registrieren' button is at the bottom of this form. An arrow points from the registration form back to the 'Registrieren' link in the login form, labeled 'zurück zum Login'.

Abbildung 3.5: LogIn-Button LogIn-Form mit Link zur Registrierung Registrierung-Form

3.3.2 Funktionen

Header Funktionen

Profil erstellen

1. Drücken sie rechts im Header auf den *Log-In Button*. Es öffnet sich das Log-In-Menü.
2. Drücken sie den Link *Registrierung*, der sich am unteren rechten Rand des Log-In-Menüs befindet. Es öffnet sich ein weiteres Fenster.
3. Tragen sie in die einzelnen Felder die geforderten Daten ein und klicken sie den Bestätigungsbutton. Bekommen sie als Antwort des Servers Ihre Registrierung war erfolgreich so wurde ihr Account erfolgreich registriert. Falls es eine andere bzw. keine Benachrichtigung gibt, versuchen sie die Lösungen hier



Abbildung 3.6: Login-Button Login-Felder Angemeldeter Benutzer mit Logout-Button

Anmeldung Besitzen sie einen Account, so können sie sich wie folgt anmelden:

(Wie sie einen neuen Account erstellen finden sie unter: Wie erstelle ich einen Account)

1. Drücken sie rechts im Header auf den *Log-In Button*. Es öffnet sich das Log-In-Menü.
2. Zum Einloggen geben sie bitte ihren Benutzernamen und das zugehörige Passwort ein.
3. War die Aktion erfolgreich, so wird die Seite neu geladen und sie werden im Header mit ihrem Namen begrüßt. Bekommen sie als Rückmeldung "Login nicht erfolgreich" so war entweder das Passwort falsch oder es existiert der Benutzer nicht.

Suchfunktion Die Suche eines Videos ist von überall über die Suchfunktion im Header möglich.

1. Klicken sie dazu auf das Eingabefeld.

2. Geben sie ihre Suche ein

3. Bestätigen sie per Mouseclick auf den Suche-Button (oder per Enter).

Auf der Ergebnisseite können sie oben zwischen den gefundenen Videos, Kategorien und Playlists wechseln:

Während dem Eingeben werden ihnen bereits zutreffende Resultate mit dem zugehörigen Bereich (also ob es sich bei den Ergebnissen um Videos, Kategorien oder Playlisten handelt) angezeigt - diese können sie per Mouseclick auswählen.

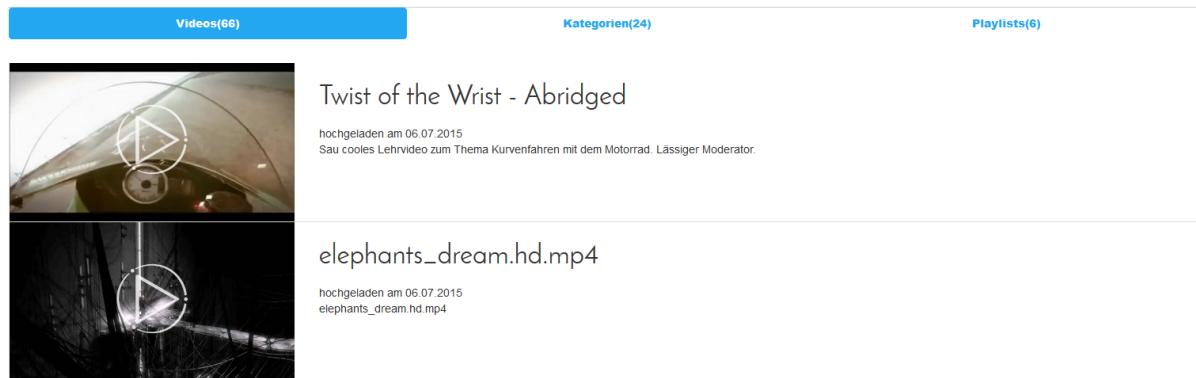


Abbildung 3.7: Suchergebnisse für den String "A"



Abbildung 3.8: Suchergebnisse des Suchstrings "G"

Rechts von dem Suchfeld könne sie vorab bereits ihre suche spezifizieren, wenn sie nach etwas Bestimmten suchen wollen.

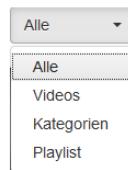


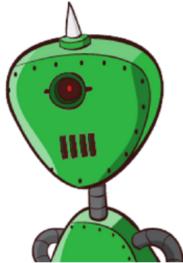
Abbildung 3.9: Eingrenzung der Suche

Tipp: Beginnt ihr Suchstring mit v: (bzw. c: oder p:), so wird automatisch nach dem zugehörigen Bereich (v - Video, c - Category, p - Playlist) gesucht, ohne dass sie dies im Reiter explizit auswählen müssen.

Profil

Das Profil bietet die Möglichkeit das Passwort und die persönlichen Daten (Email, Username, etc) zu ändern. Außerdem zeigt das Profil die von einem User hochgeladenen Videos und erstellten Playlisten an.

👤 Profil



Username:	User3
Realname:	Cooler Name1
Email:	some@mail.org

[Bearbeiten](#)
[Passwort ändern](#)

Abbildung 3.10: Ansicht Profilseite

Über die Buttons Bearbeiten bzw. Passwort ändern können die Nutzerdaten geändert werden.

🎥 Eigene Videos

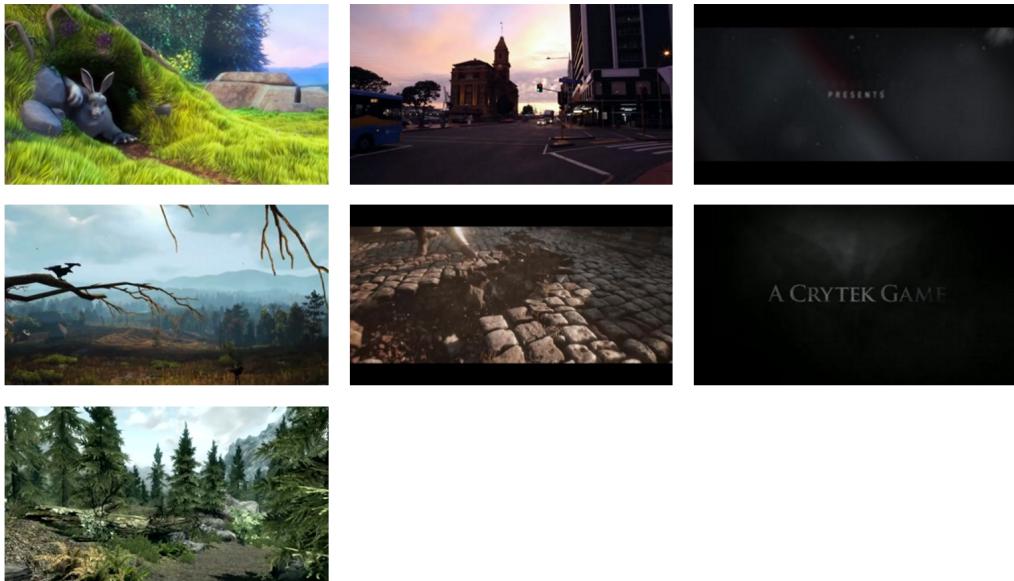


Abbildung 3.11: Ansicht Profilseite eigene Videos

Fährt man über eines seiner eigenen Videos auf dem Profil, so kann man diese löschen (der Admin kann auch auf fremden Profilen Videos löschen).

Videos verwalten

Die Verwaltung von Videos ist registrierten Benutzern vorerhalten. Als Besucher ohne Login Zugang kann man alle existierenden Videos ansehen, dabei hat man mehrere Möglichkeiten Videos zu finden oder auf der Seite zu stöbern.

Videos ansehen

Wenn Sie ein Video anschauen möchten, klicken Sie auf das Vorschaubild und sie werden zur Video-Detailansicht des Videos weitergeleitet. Dort stehen die normalen Funktionen eines Videoplayers, analog zum gängigen HTML5-Player, zur Verfügung:

- Der Play/Pause-Button spielt das Video ab oder hält es an.
- Mithilfe der Lautstärkeregelung können sie die Lautstärke einstellen.

- Daneben befindet sich ein Button, um das Video zu maximieren (Full Screen Modus).
- Unter dem Video befinden sich verschiedene Qualitätsstufen, in der sie das Video anschauen können (diese sind abhängig von der Qualität des Videos).

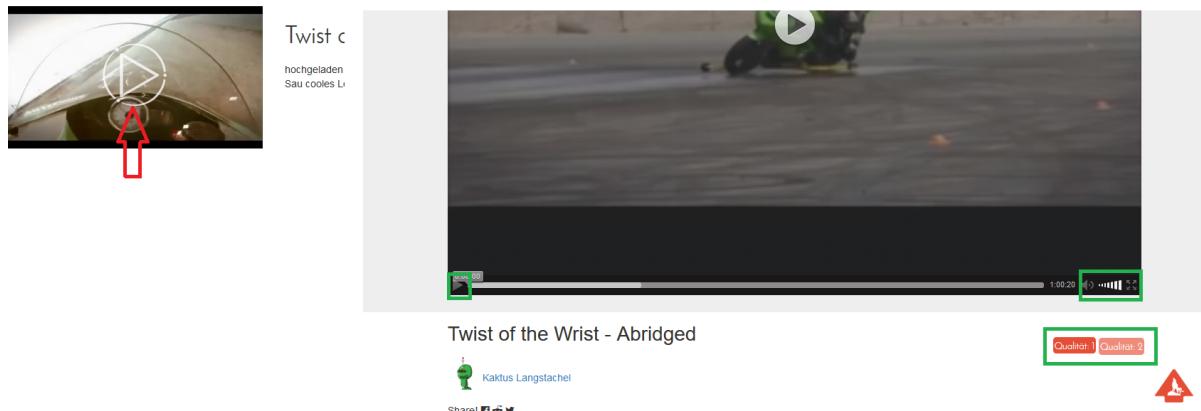


Abbildung 3.12: Videoansicht des Videos "Twist of the Wrist"

Video hochladen

Unter "Upload" können von einem eingeloggten Benutzern (ab Usergruppe 4) Videos hochgeladen werden.

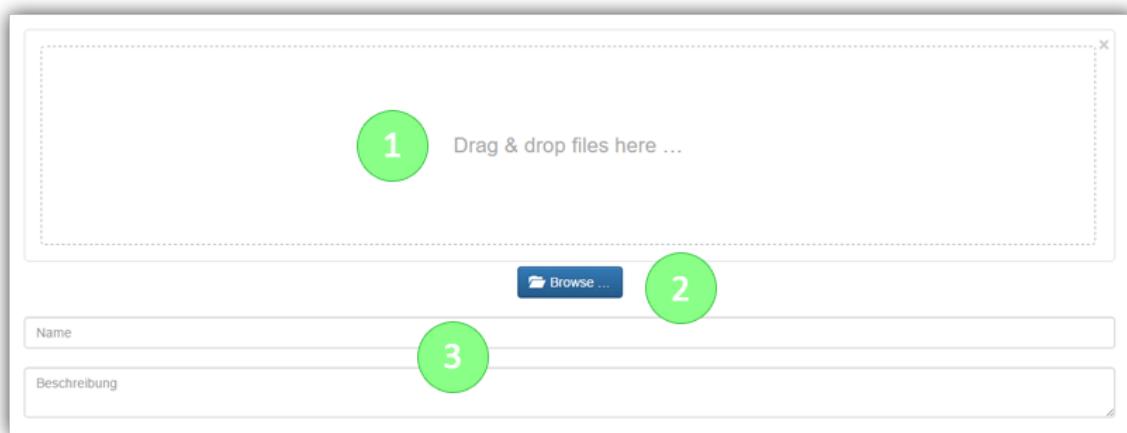


Abbildung 3.13: Ansicht Upload

Übersicht zu Abbildung 3.11

1. Feld für Drag'n Drop Eingabe von Video-Dateien, dabei kann immer nur 1 Datei hochgeladen werden. Zulässig sind dabei die Dateitypen: *mp4, mov, mpeg, wmv, avi, swf, flv, mpg*
2. "Browse-Button" zum suchen von Video Dateien auf dem Computer.
3. Name und Beschreibung des neuen Videos. Geben Wolfgang-Besuchern nähere Infos zum Video. *keine Pflichtfelder*

Übersicht zu Abbildung 3.12

1. Datei-Vorschau: hier ist das Abspielen des eingefügten Videos möglich, außerdem wird die Dateigröße angezeigt und das Element kann wieder aus dem Upload-Bereich gelöscht werden.

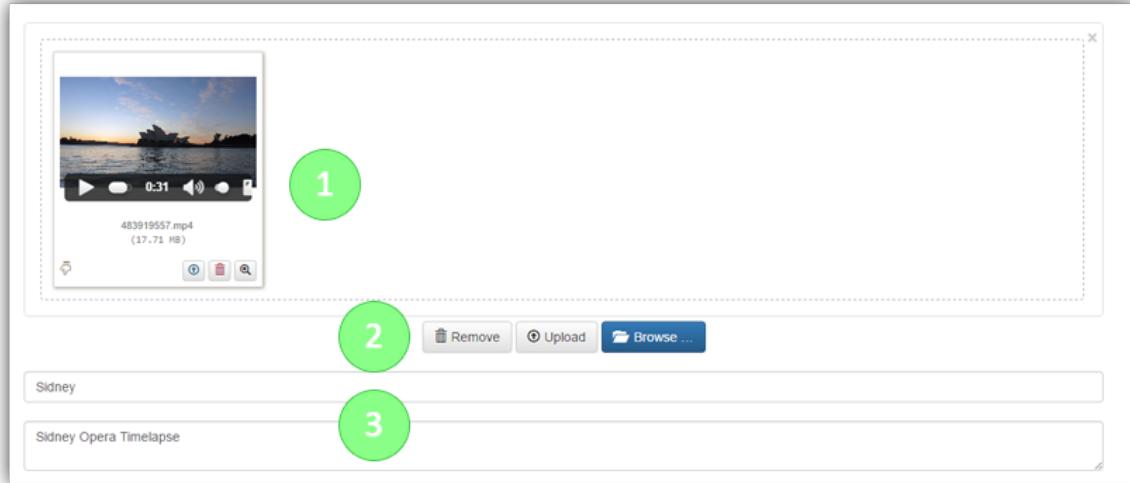


Abbildung 3.14: Ansicht Upload: nach dem Hinzufügen einer Videodatei

2. Mit dem "Remove"-Button kann die Datei aus dem Upload-Bereich gelöscht werden.
Mit "Upload" wird das Video hochgeladen.
Mit "Browse" wird erneut nach einer Datei auf dem Computer gesucht. Das ist nur dann möglich, falls die aktuelle Datei wieder aus dem Upload-Bereich gelöscht wurde.
3. Name und Beschreibung können auch in diesem Zustand noch ergänzt werden.

Wurde das Video hochgeladen, wird auf die Profilseite des Nutzers geführt. Hier können alle bereits hochgeladenen Videos eingesehen und auch wieder gelöscht werden. Mit einem Klick auf das entsprechende Video, kann es in der Video-Detailansicht angesehen werden.

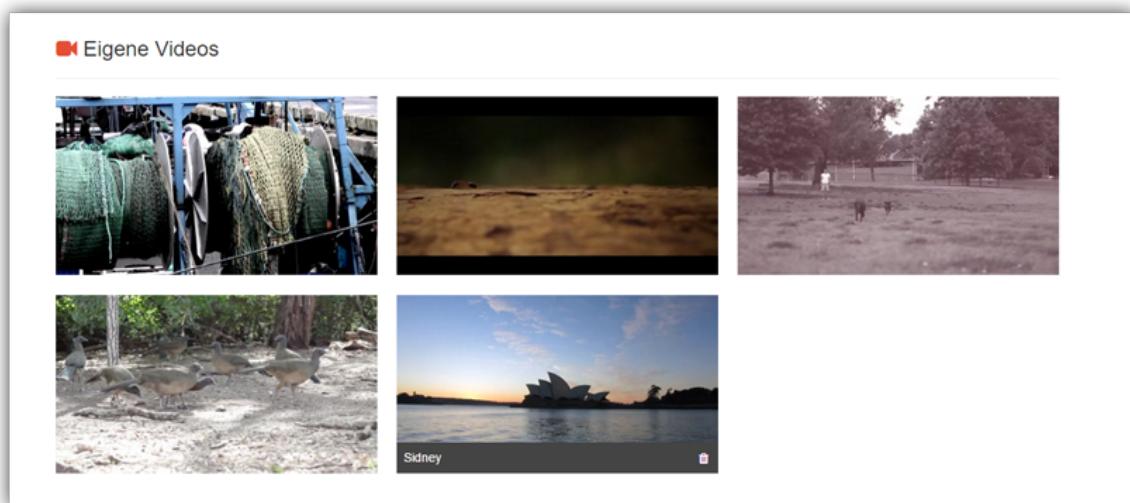


Abbildung 3.15: Ansicht Profil: nach dem Upload wird das neue Video hier mit allen bereits hochgeladenen Videos gelistet.

Video bearbeiten

Nur als eingeloggter Nutzer (ab Usergruppe 2) möglich. Ohne Login kann das Video lediglich abgespielt werden. Dabei sind Rating und Views des Videos sichtbar.

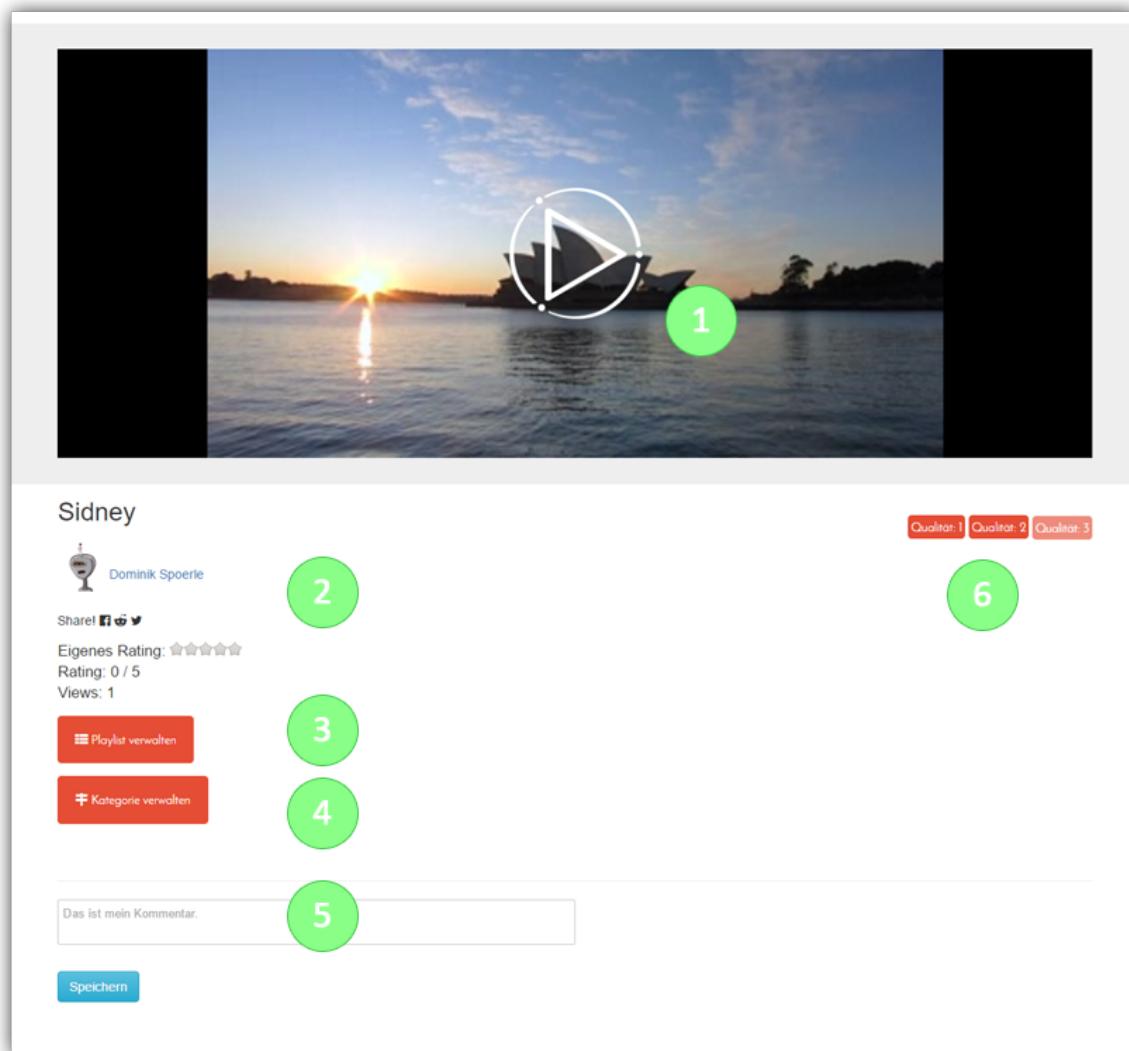


Abbildung 3.16: Ansicht Video: Video-Detailansicht

Übersicht zu Abbildung 3.14

1. Video abspielen
2. Video Informationen: Uploader-Name, Rating des eingeloggten Nutzers, Gesamtrating, Views
3. Playlist verwalten: das Video kann zu einer Playlist hinzugefügt oder von einer Playlist entfernt werden. **Hinweis:** das ist nur für Playlists des eingeloggten Nutzers möglich.
4. Kategorie verwalten: das Video kann zu einer Kategorie hinzugefügt oder von einer Kategorie entfernt werden.
5. Kommentar-Feld: hier können Kommentare zum Video hinzugefügt werden. Ein Kommentar zum Video wird zusammen mit dem Benutzernamen darunter angezeigt.
Falls ein Kommentar überarbeitet oder gelöscht werden soll, muss dieser mit dem (es ist auch möglich mehrere Kommentare zu einem Video zu verfassen).
6. Qualitätsstufen: Es wird angezeigt, in wie vielen Qualitätsstufen das Video zur Verfügung steht. Standardmäßig ist die höchste Qualitätsstufe des Videos vorab ausgewählt.

Playlists verwalten

Die Verwaltung sowie das Abonnieren von Playlisten ist registrierten Benutzern vorerhalten. Als Besucher ohne Login Zugang kann man existierenden Playlists ansehen und abspielen.

Playlist Übersicht

Auf der Seite "Playlist" findet man eine Übersicht zu existierenden Playlists. Besuchern ohne Login Zugang stehen dabei die Rubriken "Neuste Playlist", "Top Rated Playlist" und "Alle Playlisten" zur Verfügung. Eingeloggte Nutzer können zusätzlich die selbst erstellten und die abonnierten Playlists einsehen. Außerdem können Playlisten von Ihnen abonniert und ggf. ein bestehendes Abonnement für eine Playlist beendet werden. Diese Funktion wird über einen Like- / Dislike-Button dargestellt.

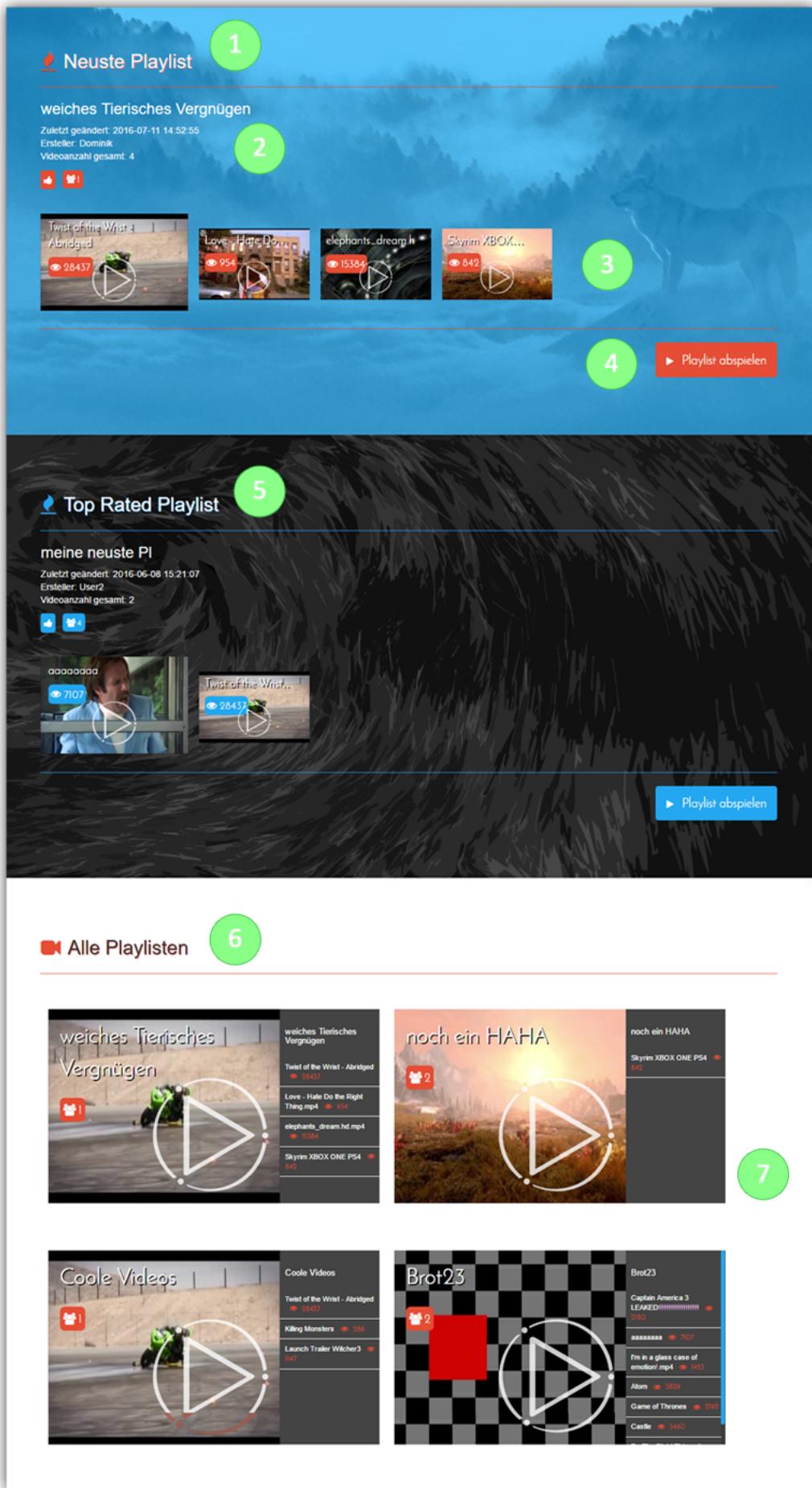


Abbildung 3.17: Ansicht Playlists: ohne Login

Übersicht zu Abbildung 3.15

1. Neuste Playlist: Darstellung der zuletzt angelegten Playlist.
2. Informationen zur Playlist: Name, Ersteller, Änderungsdatum, Videoanzahl, Abonnenten
3. Auflistung der maximal ersten 5 Videos der Playlist. Über den Play-Button sind diese Videos einzeln abspielbar.
4. Playlist abspielen: Link zur Playlist-Detailansicht der Playlist.
5. Top Rated Playlist: Darstellung der Playlist mit dem höchsten Rating durch Wolfgang-Nutzer. Der Aufbau dieser Ansicht verhält sich analog zu "Neuste Playlist", wie in den Punkten 2. - 4. beschrieben.
6. Alle Playlisten: eine Übersicht zu den Playlisten ohne weitere Filter. Über den Play-Button oder den Link "zur Playlist" wird man auf die entsprechende Playlist-Detailseite weitergeführt. Die Videos einer Playlist werden rechts neben dem Vorschaubild (1. Video der Playlist) mit Scrollfunktion gelistet.

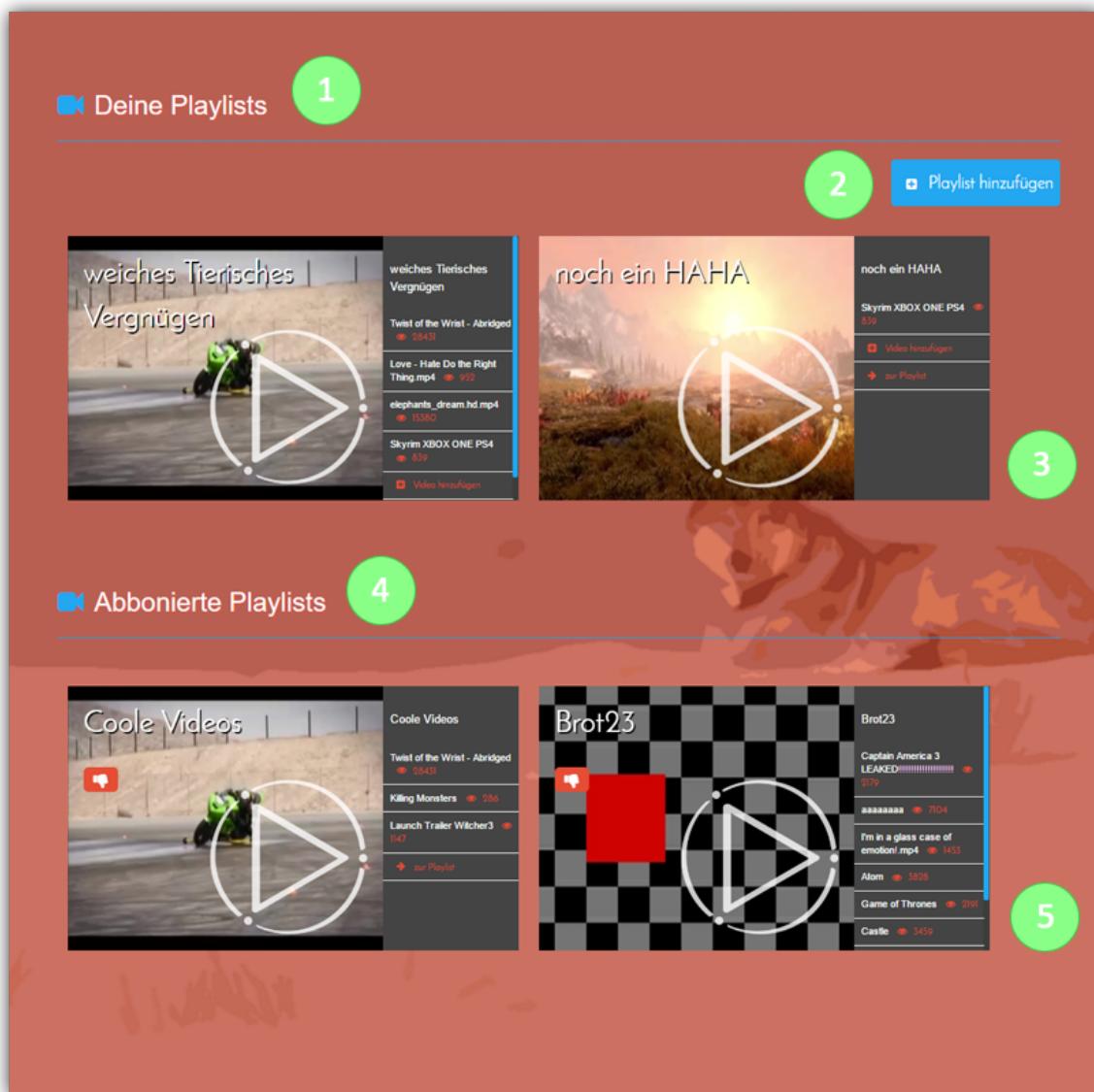


Abbildung 3.18: Ansicht Playlists: nach Login

Übersicht zu Abbildung 3.16

1. Deine Playlists: Auflistung der vom angemeldeten Nutzer erstellten Playlisten.
2. Playlist hinzufügen: öffnet ein Overlay zum Erstellen einer neuen Playlist.
3. Playlist Ansicht: über den Play-Button oder den Link "zur Playlist" wird man auf die entsprechende Playlist-Detailseite weitergeführt.
Die Videos der Playlist werden rechts neben dem Vorschaubild (1. Video der Playlist) mit Scrollfunktion gelistet.
Über den Link "Video hinzufügen" können Videos per Suchfunktion hinzugefügt werden (siehe dazu: Videos zu Playlist hinzufügen).
4. Abonnierte Playlists: Übersicht aller Playlisten, die vom angemeldeten Benutzer abonniert markiert wurden (per Like-Button).
5. Playlist Ansicht: über den Play-Button oder den Link "zur Playlist" wird man auf die entsprechende Playlist-Detailseite weitergeführt.
Die Videos der Playlist werden rechts neben dem Vorschaubild (1. Video der Playlist) mit Scrollfunktion gelistet.
Mit dem "Dislike"-Button kann das Abonnement für die Playiste beendet werden.

Videos zu Playlist hinzufügen

Videos können zu einer Playlist auf zwei Wegen hinzugefügt werden. Entweder in der Video Detailansicht (siehe "Video bearbeiten") oder unter Playlists, im Inhaltsbereich "deine Playlists" mithilfe des Link "Video hinzufügen".

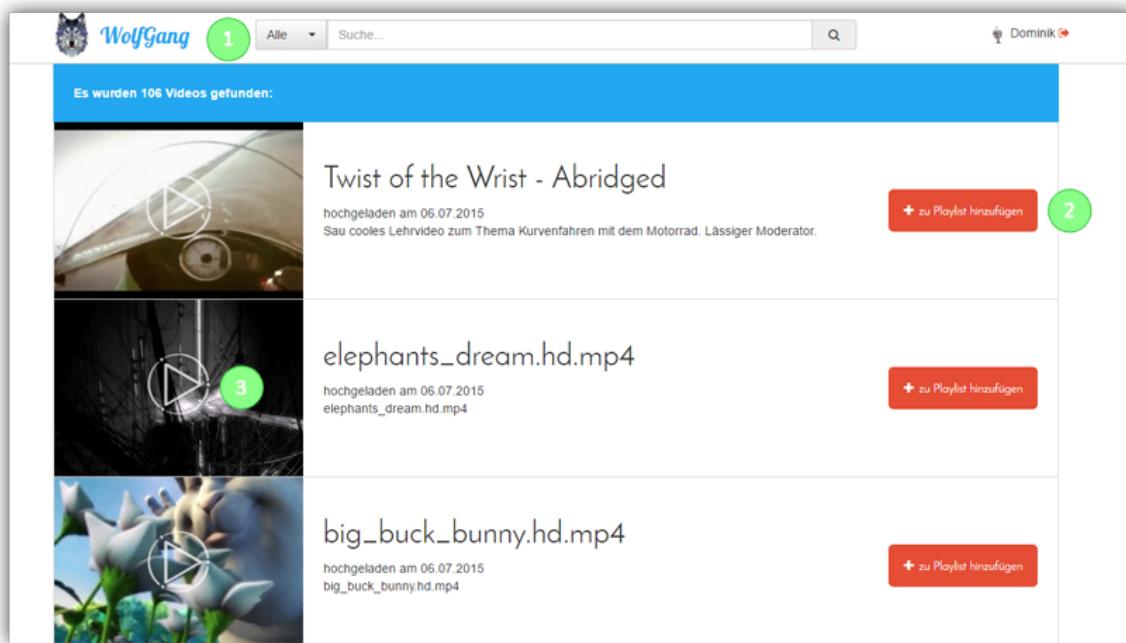


Abbildung 3.19: Ansicht Suchfunktion: Video zu Playlist hinzufügen

Übersicht zu Abbildung 3.17

1. Suchfunktion: um ein spezielles Video der gewählten Playlist hinzuzufügen kann es über die Suchfeld gesucht werden. Dabei muss bei der Auswahl entweder "Alle" oder "Video" hinterlegt sein (*diese Funktion ist aktuell leider fehlerhaft, da sie bis zum Abgabetermin nicht vollständig fertiggestellt werden konnte*).
2. Button "zu Playlist hinzufügen": fügt das Video der gewählten Playlist hinzu. Man wird auf die Seite Playlists weitergeleitet.
3. Video Wiedergabe

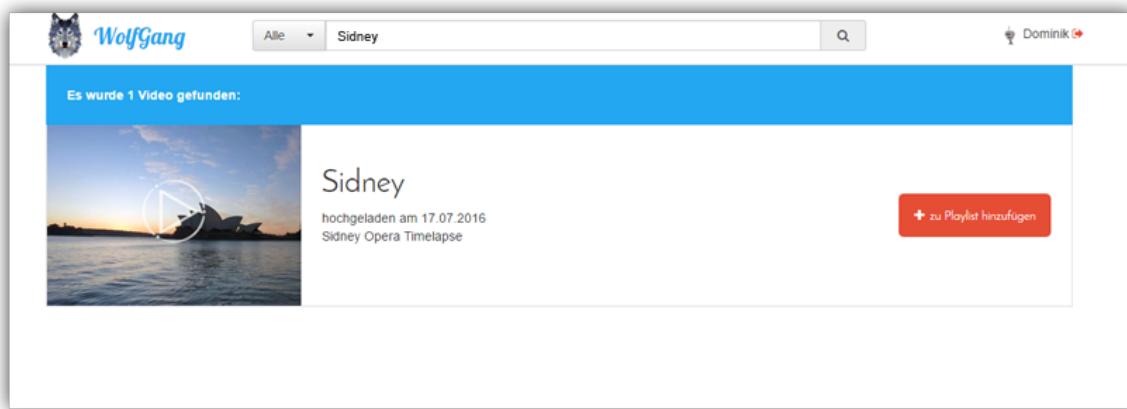


Abbildung 3.20: Ansicht Suchfunktion: Video zu Playlist hinzufügen, nach Suche nach Video Sidney

Playlist Detailseite

Auf der Detailseite einer Playlist kann diese abgespielt und bearbeitet werden.

weiches Tierisches Vergnügen

Playlist ändern

Playlist löschen

Playlist umsortieren

1

Twist of the Wrist - Abridged



Qualität 1 Qualität 2

Videos der Playlist

3



Twist of the Wrist - Abridged



Kaktus Langstachel

Share!

Rate: ★★★★☆

4

Das ist mein Kommentar.

Speichern



wolfgang2 2016-01-11 11:00:00
whay



StefanZuk 2016-01-11 10:04:10
neuer comment



Cooler Name1 2016-01-08 17:01:08
Sau cooles Lehrvideo!!

5

Abbildung 3.21: Ansicht Playlist Detailansicht

Übersicht zu Abbildung 3.19

1. Playlistname und Buttons zur Bearbeitung. Die Buttons können nur vom Ersteller Playlist oder einem User mit Adminrechten genutzt werden:

- (a) "Playlist ändern": Name der Playlist ändern
 - (b) "Playlist löschen": Playlist löschen
 - (c) "Playlist umsortieren": führt zur Ansicht zum Sortieren der Videos der Playlist
2. Wiedergabe der Playlist
 3. Anzeige der Videos der Playlist, beim Abspielen eines Videos wird die Playlist an dieser Stelle weiter abgespielt.
 4. Informationen zum aktuellen Video: Videoname, Uploader-Name, Rating
 5. Kommentarfunktion und Darstellung der Kommentare des aktuellen Videos.

Admin Funktionen

Admins haben Zugriff auch besondere Funktionen um die Verwaltung im Frontend zu ermöglichen.

Nutzerverwaltung

Der nur als Admin einsehbare Menüpunkt "Usermanagement" führt auf die entsprechende Unterseite. Hier kann ein Admin die registrierten User mit seiner UserId, sein Namen, seiner Email Adresse und der Anzahl der Playlisten/Videos einsehen (2) und folgende Funktionen nutzen:

- (1) Nach Usern suchen, dadurch kommt man auf eine Einzelansicht (5). Durch ein Suchen nach :all kommt man zurück zu allen Benutzern.
- (3) Alle User außer Admins löschen
- (4) Die User Gruppe von allen Usern außer Admins löschen

The screenshot shows a table of user management. The columns are: #, Name, Mail, Videos, Playlists, and Type. The Type column contains dropdown menus for Admin or Benutzer. There are also delete icons in each row. A search bar at the top right has a magnifying glass icon. A navigation bar at the bottom shows pages 1 through 8.

#	Name	Mail	Videos	Playlists	Type	
391	1	bla@blah.de	0	0	Admin	
288	1029	aldk@akdk.com	0	0	Admin	
451	111	111@111.com	0	0	Benutzer	
452	1113	dff@gdsfdf.com	0	0	Benutzer	
416	2	bla@blah.de	0	0	Benutzer	
419	3	bla@blah.de	0	0	Benutzer	
337	aaaaa	bla@blah.de	0	0	Benutzer	
317	Admin	some@mail.org	14	0	Admin	
582	alterknocken	asd@yahoo.com	0	0	Benutzer	
383	applefan42	isaac.newton@mail.com	0	1	Benutzer	
572	asd	asd@yahoo.com	0	0	Benutzer	
585	asd2	asd@yahoo.com	0	0	Benutzer	
332	baumstumpf1	baum@wald.de	0	0	Benutzer	
319	baumzweig	zweig@wald.de	0	0	Benutzer	
321	baumzweig2	zweig@wald.de	0	0	Benutzer	
550	Benutzer1	benutzer@provider.de	0	0	Benutzer	
590	carina	carina@irgendwo.de	8	0	Admin	
341	ccccccc	bla@blah.de	0	0	Benutzer	
335	coolguy	bla@blah.de	0	0	Benutzer	
481	dd3	test@google.de	0	0	Benutzer	

The screenshot shows a detailed view of a single user. The columns are: #, Name, Mail, Videos, Playlists, and Type. The Type column contains a dropdown menu for Admin. There is a back arrow at the top left and a magnifying glass icon in the search bar.

#	Name	Mail	Videos	Playlists	Type	
294	wolfgang	wolfgang.braun@uni-tuebingen.de	1	0	Admin	

Playlistverwaltung

Wenn ein Admin auf den Menüpunkt "Playlist" klickt, dann gelangt er statt zur normalen Useransicht,

bei der Playlisten erstellt und angesehen werden können, zur Adminansicht, die zur Verwaltung der Playlisten verwendet wird (1). Neben einer Auflistung aller Playlisten werden folgende Funktionen zu Verfügung gestellt:

- (2) Der Name der Playlist geändert werden. Vergisst der Admin danach den Edit Knopf(3) zu drücken warnt ihn eine rote Umrandung vor nicht gespeicherten Änderungen (4)
- Jede Playlist kann gelöscht werden, egal wer sie erstellt hat (3)

ID	Owner	public?	Letzte Änderung	Name	Videos	
272	Dominik	1	2016-07-22 16:08:42	Tierisches Vergnügen	5	Edit Löschen
274	Dominik	1	2016-07-06 15:28:32	noch ein HAHA	1	Edit Löschen
271	User3	1	2016-07-04 12:39:43	Cooles Videos	3	Edit Löschen
194	User2	1	2016-07-03 20:21:12	Brot23	7	Edit Löschen
214	steffen	1	2016-06-27 14:39:53	Steffen Pannen 1	4	Edit Löschen
266	User3	1	2016-06-27 14:39:06	Offeneliste 1	0	Edit Löschen
260	stefan	0	2016-06-24 13:22:54	EigeneTestListe	1	Edit Löschen
259	stefan	0	2016-06-24 13:01:10	StefanListe1	14	Edit Löschen
196	User3	1	2016-06-22 14:41:26	playtest	0	Edit Löschen
1	hans	0	2016-06-22 14:40:47	Lieblingsfilm	0	Edit Löschen

1 2 3

ID	Owner	public?	Letzte Änderung	Name	Videos	
272	Dominik	1	2016-07-22 16:08:42	Tierisches Vergnügen 3000	5	Edit Löschen
274	Dominik	1	2016-07-06 15:28:32	noch ein HAHA	1	Edit Löschen
271	User3	1	2016-07-04 12:39:43	Cooles Videos	3	Edit Löschen

Kategorienverwaltung

Wenn ein Admin auf den Menüpunkt "Kategorien" klickt, erhält er ein leicht anderes Layout (1) als Nutzer der anderen Gruppen.

Hier wird die UserIds des Erstellers, der Name der Kategorie und die Anzahl der Videos angezeigt. Außerdem kann ein Admin Kategorien umbenennen, löschen (3) und erstellen (2).

3.4 Probleme und Lösungen

Grundsätzlich werden alle (Fehler-) Meldungen über eine Notify-Box rechts unten unmittelbar nach dem Ereignis eingeblendet um so über Erfolg oder Misserfolg der Aktion zu informieren.

1. *Buttons funktionieren nicht(Drop-Down-Menü erscheint nicht)* :
Überprüfen sie, ob Javascript aktiviert ist.
2. *Videos können nicht abgespielt werden*:
Überprüfen sie ob sie die aktuellen Treiber zur Wiedergabe eines .mp4 Videos über den HTML5 Player installiert haben.

The screenshot shows a table of categories with the following data:

ID	Name	Videos	
231	Alles Gute kommt von Oben	6	Edit Löschen
164	Blaaa	2	Edit Löschen
124	Blender	7	Edit Löschen
178	cField	3	Edit Löschen
85	Deutscher Heimatfilm	5	Edit Löschen
95	dhdhdhdh	1	Edit Löschen
221	Die Testkategorie	1	Edit Löschen
232	Doku	3	Edit Löschen
175	DominiksKategorie	0	Edit Löschen
69	Drama	1	Edit Löschen

Page navigation: 1 2 3 4 5 6

3. Elemente werden nicht dargestellt (Seite ist fast leer) :

Vermutlich ist der Backendserver, der die meisten Daten liefert, gerade nicht erreichbar. Diesen Fehler können sie selbst nicht beheben, versuchen sie es später nochmal.

4. Ich kann ein Video/eine Playlist/eine Kategorie nicht finden:

Die Suche erkennt leider keine ähnlichen Begriffe (oder ähnliche Schreibweisen):

Wollen sie eine Kategorie "Fischäufufen", suchen aber nach "Fische", so werden sie die Kategorie nicht finden. Versuchen sie in diesem Fall die Suche mit einem Teilbegriff (und wählen das richtige Ziel über die Suchvorschläge aus). Finden sie trotzdem kein Ergebnis, so gibt es kein Ziel, wo den Suchbegriff als Teil ihres Namens besitzt.

5. Registrierung geht nicht :

1. Bekommen sie die Fehlermeldung "Passwörter stimmen nicht überein", so geben sie ihr Passwort und die Bestätigung erneut ein (achten sie auf Groß/Kleinschreibung).

2. Bekommen sie eine Fehlermeldung über die Notify-Box, so ist entweder die angegebene E-Mail Adresse ungültig oder ihr Username wird bereits verwendet (genaue Informationen stehen in der Benachrichtigung). In diesem Fall müssen sie sich einen anderen Usernamen /eine gültige E-Mail Adresse eingeben.

6. Ich kann meinen Kommentar nicht bearbeiten:

Eine Bearbeitung des Kommentars ist derzeit leider nicht möglich. Sie können allerdings ihren alten Kommentar löschen und einen neuen schreiben.

7. Ich sehe die in der Anleitung beschriebenen Knöpfe nicht :

Sind die Knöpfe nicht sichtbar, so haben sie nicht die benötigten Rechte, um die Aktionen, die mit den Knöpfen assoziiert sind, auszuführen. Schauen sie sich dafür die Rechtetabelle an.

8. Video Upload funktioniert nicht:

Bekommen sie keine Fehlermeldung und das Video wird trotzdem nicht hochgeladen, so ist vermutlich die Größe des Videos größer als die vom Server erlaubte maximale Größe. Komprimieren sie ihr Video oder versuchen sie es mit einem kleineren Video.

3.5 Besonderheiten von WolfGang

3.5.1 Die Idee WolfGang

Um dem Projekt einen Namen und ein Thema für das Design der Website zu geben, war beim Projektstart wichtig, eine passende Idee zu finden. Dabei stand die Namensfindung im Mittelpunkt.

Der Name "WolfGang" entstand mit der Anlehnung an den Trend, Software nach einem deutschen Namen zu benennen, wie z.B. beim Messenger "Franz". Als Name wurde der, des Projekt Betreuers Wolfgang Braun gewählt - auch da sich damit ein Themenfeld für Design und Konzeption anboten. Das Thema "Wolf" ist das Design-Thema und grafisch überall auf der Website zu finden, wobei "Gang" für einen Rundgang durch die Videos der Plattform stehen soll.



Abbildung 3.22: Logo WolfGang

3.5.2 Navigation und Overlay Techniken

Durch die Overlay Technik konnten viele Funktionen der Website grafisch sinnvoll und einfach eingebunden werden, ohne dass der Übersicht für den Nutzer der Website gestört wird. Beispiele hierfür sind die Hauptnavigation, sowie Buttons für Video- und Playlist-Verwaltung.

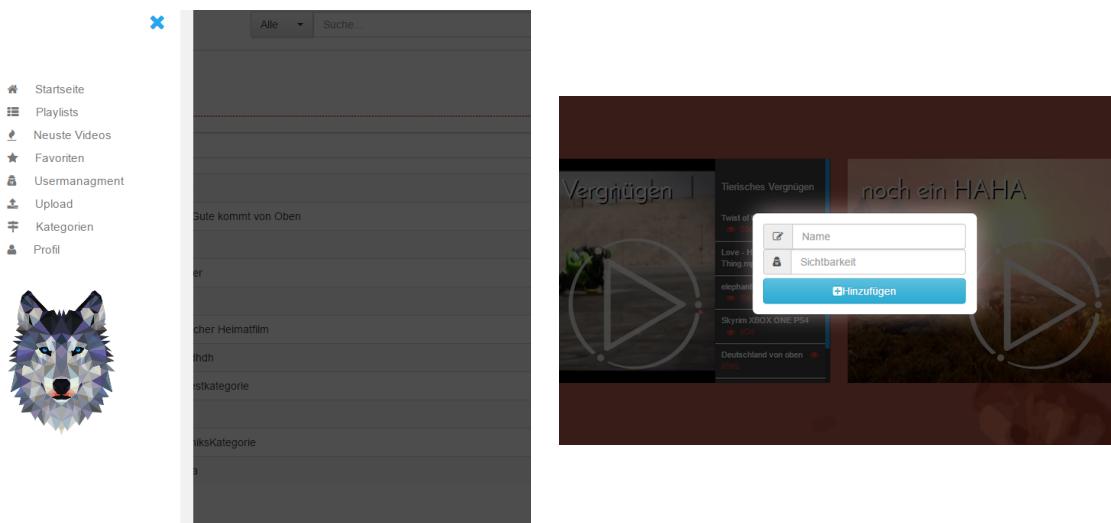


Abbildung 3.23: Overlay Ansicht Hauptnavigation / Button-Funktion

3.5.3 Wolf-Design

Angelehnt an den Projektnamen ist der Wolf, als Logo und Symbol für das Projekt an zahlreichen Stellen der Website zu finden und kennzeichnet das Design der Website. Genutzt wurden dafür ausschließlich freie Grafiken (Bilder, Gifs, Vektorgrafiken), die ggf. noch nachträglich abgewandelt, bearbeitet und für die Seite aufbereitet wurden. Beispiele finden sich dafür unter anderem im Logo, der Navigation, in den Hintergrundgrafiken des Content-Bereichs und der 404-Seite.

3.5.4 Tag Cloud Kategorien

Eine Begriffs-Wolke mit zufälligen Kategorien-Namen. Kategorien mit vielen Videos werden groß dargestellt, Kategorien mit wenigen Videos kleiner.

3.5.5 RoboHash: automatische Benutzerbilder

Da das Backend keine Avatar-Funktionen anbietet, wurde RoboHash.org für die Generation zufälliger Benutzerbilder für jeden User verwendet.

3.5.6 Typeahead Suchfunktion

Durch ein Bootstrap Addon ist die Suche mit dem "Typeahead" Feature ausgestattet. Beim tippen wird direkt die Suche im Hintergrund ausgeführt.



Abbildung 3.24: Logo Image



Abbildung 3.25: Hintergründe (rot, blau, schwarz)

Cloud Tag

Alles Gute kommt von Oben

Blaaa

Blender

Deutscher Heimatfilm

dhdhdhdh

Die Testkategorie

Drama

eke

epe

ete

Französische Nouvelle Vague

Französischer Surrealismus

gtgggggggggggg

Abbildung 3.26: Category Cloud

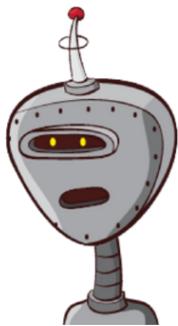


Abbildung 3.27: RoboHash Profilbild

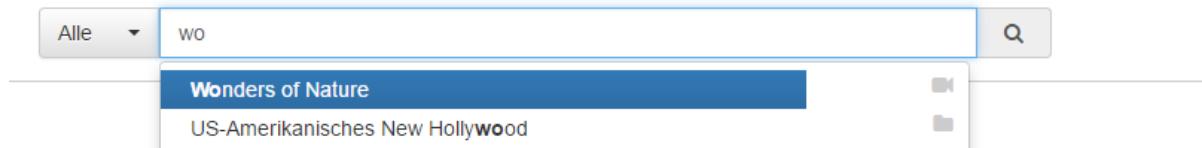


Abbildung 3.28: Autocomplete Beispiel der Suchfunktion