

**Philosophische** Fakultät III

Sprach- , Literatur- und Kulturwissenschaften

Institut für Information und Medien, Sprache und Kultur (I:IMSK)  
Lehrstuhl für Medieninformatik

Praxisseminar

Modul: MEI-M 26 (M. Sc.)

SS 2014

Leitung: Tim Schneidermeier

Mediathek Crawler

Software Requirements Specification –Anforderungsspezifikation

Version 0.1  
Lukas Lamm, David Lechler, Felix Müller, Tobias Semmelmann  
1511167, 1510484, 1535015, 1509311  
1. und 2. Semester M.Sc. Medieninformatik

Abgegeben am 30.08.2014

Inhalt

[0 Versionsgeschichte 5](#_Toc397095982)

[1 Einführung 5](#_Toc397095983)

[1.1 Zweck des Dokuments 5](#_Toc397095984)

[1.2 Projektumfang (Scope) 5](#_Toc397095985)

[1.3 Definitionen, Akronyme und Abkürzungen 5](#_Toc397095986)

[1.4 Referenzen und Quellenangaben 6](#_Toc397095987)

[1.5 Übersicht über das Dokument 6](#_Toc397095988)

[2 Allgemeine Beschreibung 7](#_Toc397095989)

[2.1 Produktperspektive 7](#_Toc397095990)

[2.2 Produktfunktionen 7](#_Toc397095991)

[2.3 Benutzergruppen und Charakteristika 8](#_Toc397095992)

[2.4 Zielplattform 8](#_Toc397095993)

[2.5 Einschränkungen, Annahmen und Abhängigkeiten 9](#_Toc397095994)

[2.6 Benutzerhandbuch und Dokumentation 9](#_Toc397095995)

[3 Spezifische Anforderungen 9](#_Toc397095996)

[3.1 (Externe) Schnittstellen 9](#_Toc397095997)

[3.1.1 Benutzerschnittstellen 9](#_Toc397095998)

[3.1.2 Hardware-Schnittstellen 10](#_Toc397095999)

[3.1.3 Software-Schnittstellen 10](#_Toc397096000)

[3.1.4 Kommunikationsschnittstellen 10](#_Toc397096001)

[3.2 Funktionale Anforderungen 11](#_Toc397096002)

[3.3 Anwendungsfälle (Use Cases) 13](#_Toc397096003)

[4 Nicht-funktionale Anforderungen 15](#_Toc397096004)

[4.1 Performance 15](#_Toc397096005)

[4.2 Sicherheit 15](#_Toc397096006)

[4.3 Qualität 16](#_Toc397096007)

[4.3.1 Ausfallsicherheit 16](#_Toc397096008)

[4.3.2 Erreichbarkeit 16](#_Toc397096009)

[4.3.3 Wartbarkeit 16](#_Toc397096010)

[4.3.4 Erweiterbarkeit 16](#_Toc397096011)

[4.3.5 Usability 17](#_Toc397096012)

[4.4 Datenmodell 17](#_Toc397096013)

[5 Risiken 18](#_Toc397096014)

Abbildungen

[Abbildung 1: Datenflussdiagramm 7](#_Toc397096706)

[Abbildung 2: Funktionsüberblick 8](#_Toc397096707)

[Abbildung 3: Datenbankschema 17](#_Toc397096708)

Tabellen

Tabelle 1: Use Case (Durchsuchen der Plattform) 13

Tabelle 2: Use Case (Wiedergabe einer Sendung) 13

Tabelle 3: Use Case (Zusammenstellen des Abendprogrmms) 14

Tabelle 4: Use Case (Registrieren) 14

# Versionsgeschichte

|  |  |  |
| --- | --- | --- |
| **Version** | **Datum** | **Beschreibung** |
| 0.1 | 27.05.2014 | Initialisierung |
| 0.2 | 04.07.2014 | Korrekturen & Ergänzungen nach T. Schneidermeier |
| 0.3 | 10.07.2014 | Aufnahme der Risiken |
| 0.4 | 29.08.2014 | Projektabschluss |

# Einführung

## Zweck des Dokuments

Dieses Dokument dient zur Spezifizierung der Anforderungen, die an das Projekt “Mediathek Crawler” gestellt werden. Die Spezifikation basiert auf dem IEEE 830-1998 Standard. Zu den Zielgruppen des Projekts zählen neben dem Projektteam und dem Auftraggeber (Tim Schneidermeier) insbesondere Personen, die ohne einen stationären Fernseher auf Inhalte der Mediatheken zugreifen wollen. Durch das Dokument werden die Anforderungen an das System schriftlich festgehalten und dient als Prüfstein für die Projektabnahme.

## Projektumfang (Scope)

Die umzusetzende Software stellt eine Online-Plattform zum Streaming von frei verfügbaren Videoinhalten im Internet dar. Das Projekt wird im Rahmen des Praxisseminars im Masterstudiengang Medieninformatik an der Universität Regensburg durchgeführt. Die Software umfasst das Durchsuchen und Streamen von Videoinhalten mehrerer Plattformen. Hinzu kommt ein persönlicher, Login-geschützter Bereich, in dem Inhalte vorgemerkt und in Playlists gesammelt werden können.

Sinn und Ziel des Projekt ist es eine Plattform zu entwickeln, welche die Mediatheken verschiedener Anbieter miteinander verbindet um so sein eigenes TV Programm unabhängig von den TV Sendern zusammenzustellen.

## Definitionen, Akronyme und Abkürzungen

|  |  |
| --- | --- |
| **Definition/Akronym/Abkürzung** | **Beschreibung/Erläuterung** |
| API (Application Programming Interface) | Schnittstelle zur Anbindung an ein Softwaresystem |
| Streaming / Stream | Gleichzeitiges Empfangen und Wiedergabe des Videodateien |
| Webapp | Anwendungsprogramm, welches in einem Browser abläuft |
| (Web-)Browser | Computerprogramm zur Darstellung von Webseiten im World Wide Web |
| Http-Request | Anfrage über das Hypertext Transfer Protocol |
| FTP | File Transfer Protocol |
| FLV | Flash Video. Offenes Containerformat für die Übertragung von Videoinhalten im Internet |

## Referenzen und Quellenangaben

Real Time Messaging Protocol. (n.d.). *In Wikipedia.* Retrieved from <http://de.wikipedia.org/wiki/Real_Time_Messaging_Protocol> - 27.05.2014

Real Time Streaming Protocol. (n.d.). *In Wikipedia.* Retrieved from <http://de.wikipedia.org/wiki/Real-Time_Streaming_Protocol> - 27.05.2014

File Transfer Protocol. (n.d.). *In Wikipedia.* Retrieved from <http://de.wikipedia.org/wiki/File_Transfer_Protocol> - 27.05.2014

Flash Video. (n.d.). *In Wikipedia.* Retrieved from <http://de.wikipedia.org/wiki/Flash_Video> - 27.05.2014

## Übersicht über das Dokument

Im Folgenden wird eine allgemeine Beschreibung des Projekts gegeben. Hierzu zählen insbesondere die Produktfunktionen, die Zielgruppe, die Zielplattform sowie Einschränkungen und Abhängigkeiten. Punkt 3 beschreibt die spezifischen Anforderungen an das Projekt. Hierbei werden u.a. die Schnittstellen, Anforderungen und Anwendungsfälle geklärt. Im letzten Punkt werden die nicht-funktionalen Anforderungen, wie z.B. die Performance, Qualität und das zugrunde liegende Datenmodell, beschrieben.

# Allgemeine Beschreibung

## Produktperspektive

Die herzustellende Plattform dient der Sammlung von Videoinhalten verschiedener Plattformen und Mediatheken in einem gemeinsamen Interface. Hierbei soll dem Anwender ermöglicht werden über eine Suchmaske mehrere Mediatheken zu durchsuchen und deren Inhalte an selbiger Stelle zu konsumieren. Ursprung des Projekts ist die Aufhebung des linearen Programmschemas von klassischen Fernsehsendern, sprich zu jedem Zeitpunkt verschiedenste Videoinhalte zur Verfügung zu haben.

Der Mediathek Crawler ist eine alleinstehende Webplattform im Rahmen des Masterkurses “Praxisseminar” des Lehrstuhls Medieninformatik an der Universität Regensburg.

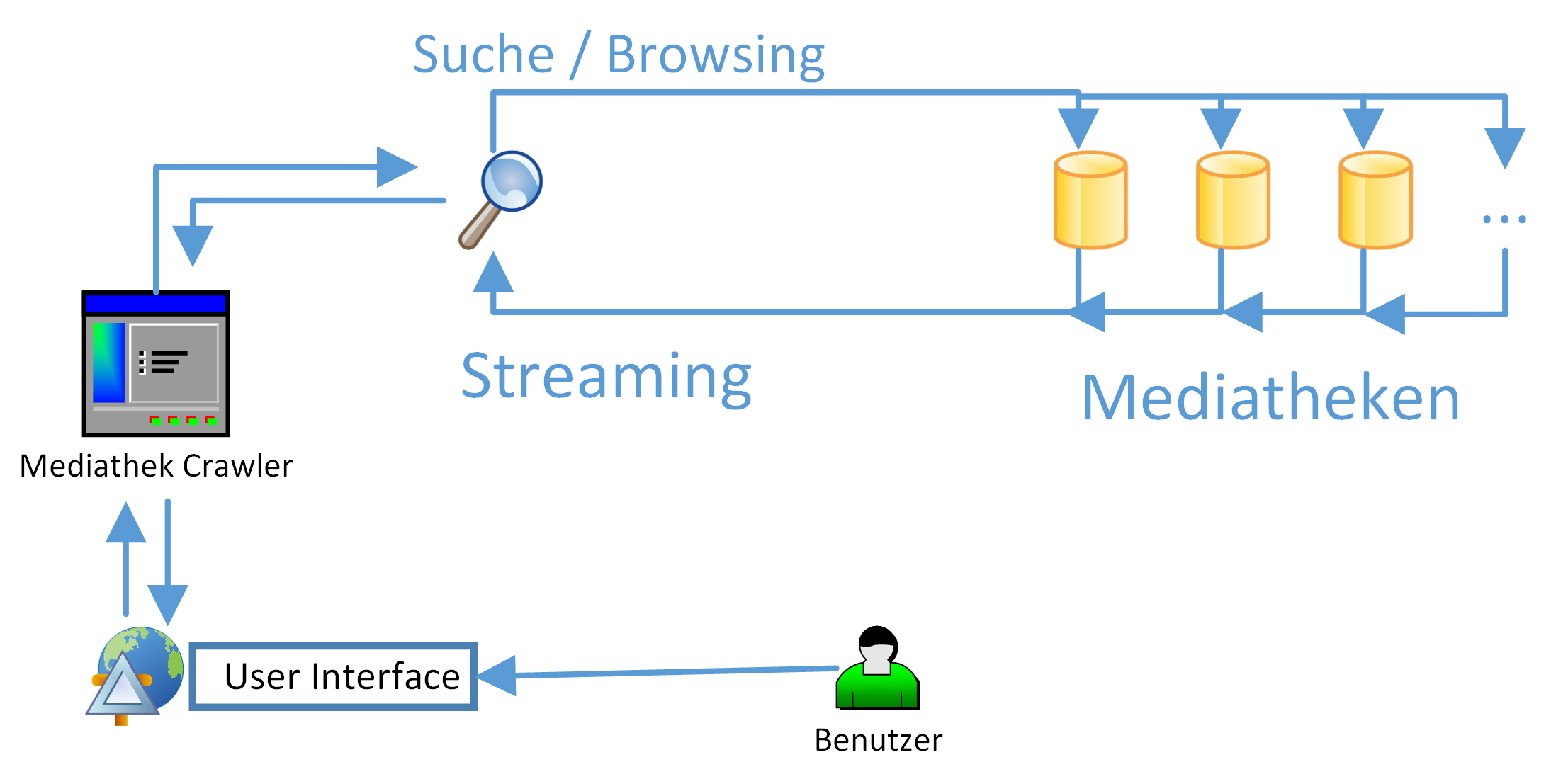


Abbildung 1: Datenflussdiagramm

## Produktfunktionen

Der Benutzer soll durch das Produkt in der Lage sein, ein persönliches TV-Programm zusammenzustellen und diese unabhängig von den eigentlichen Anbietern innerhalb der Webplattform streamen.

Eine Übersicht über aktuelle Videos der eingebundenen Mediatheken bietet dem Anwender Einblicke über verfügbare Inhalte. Diese Inhalte können je nach Bedarf gefiltert und sortiert werden. Der Login-geschützten Bereich bietet einige Funktionen, wie z.B. das Anlegen von Playlists, wodurch der Nutzer mehrere Inhalte am Stück abspielen, ohne dass weitere Interaktion nötig ist.

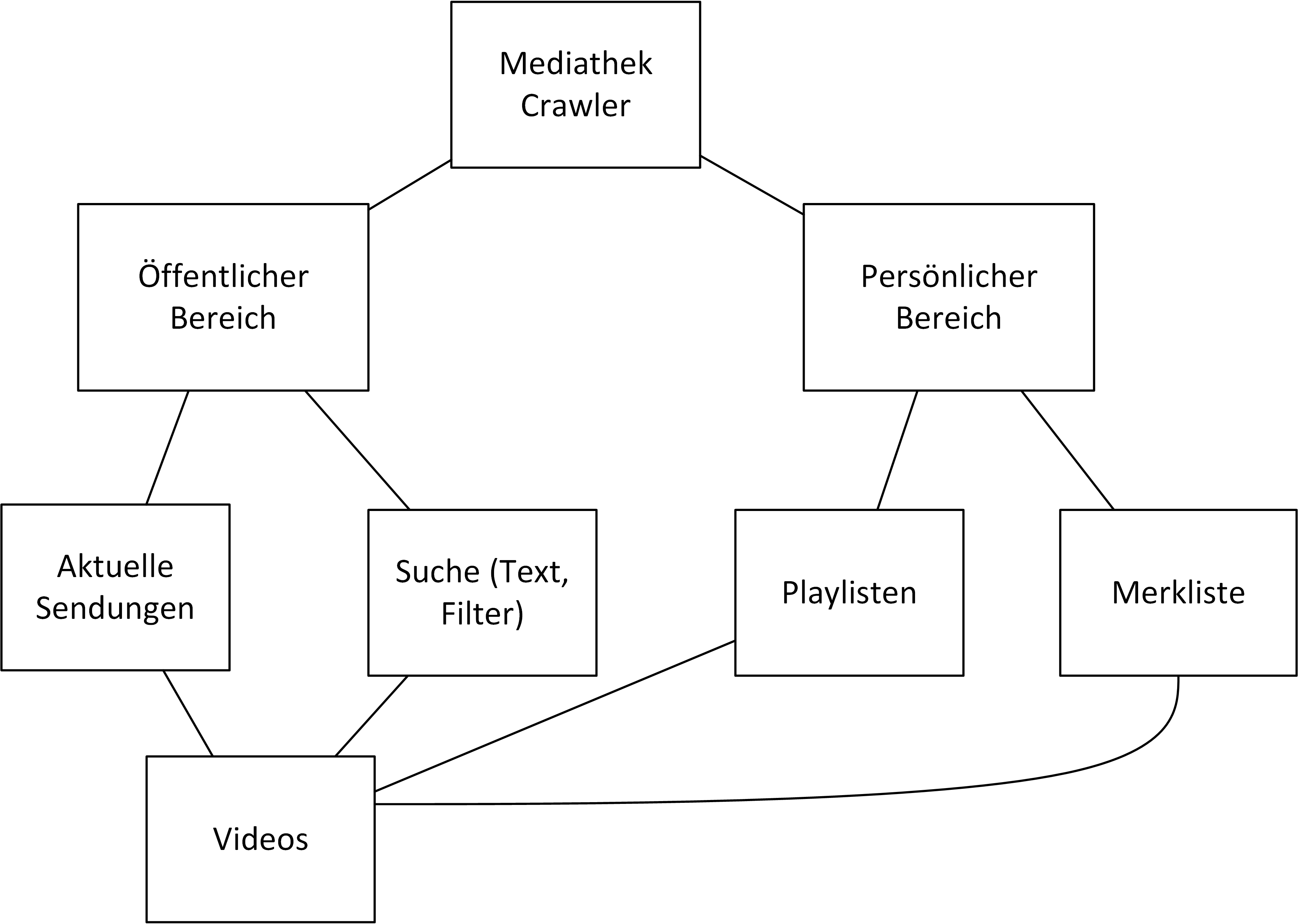


Abbildung 2: Funktionsüberblick

## Benutzergruppen und Charakteristika

Zu den Nutzergruppen der Anwendung zählen insbesondere Nutzer der eingebundenen Mediatheken und weitere Streaming-Begeisterte jeglichen Alters. Zu deren Charakteristika zählen das sichere Bedienen eines Web-Browsers als auch einschlägige Erfahrung mit ähnlichen Videostreaming-Angeboten.

Eine weitere große Zielgruppe bilden Personen, die keinen Fernseher besitzen, aber trotzdem an aktuellen Sendungen der öffentlich-rechtlichen Fernsehanstalten interessiert sind. Auch Personen, die aufgrund ihrer Tätigkeit oder anderer Gründe Sendungen nicht im programmgebundenen TV-Gerät ansehen können oder wollen. Stattdessen soll der Mediathek-Crawler eine einfache Möglichkeit bieten diese Abhängigkeit vom Fernsehprogramm zu umgehen.

Zusätzlich zu den bereits genannten Zielgruppen, ergibt sich durch den immer verbreiteteren Einsatz von Tablets eine weitere Nutzergruppe. Diesen soll es möglich sein, über ein mobiles Endgerät die Seite in vollem Umfang zu nutzen. Hierbei steht die Mobilität der Anwendung klar im Vordergrund. Ein adaptives Nutzerinterface sorgt für eine funktionale und ansprechende Oberfläche der Webseite auf Geräten mit geringeren Bildschirmgrößen.

## Zielplattform

Das Projekt wird als plattformunabhängige Webapplikation umgesetzt und ist damit in allen gängigen Browsern funktionsfähig. Weiterhin gibt es die Möglichkeit, die Anwendung auf mobilen Endgeräten wie Smartphone oder Tablet zu nutzen. Durch die Verfolgung des Cross-OS, mobile-first Implementierungs-Ansatzes wird ein sehr breites Publikum erreicht.

## Einschränkungen, Annahmen und Abhängigkeiten

Inwieweit das Einbinden verschiedener Mediatheken möglich ist, ist von den einzelnen Anbietern abhängig. Sollte das Streamen der Inhalte einzelner Mediatheken nicht möglich sein müssen eventuell andere Lösungen gefunden werden. Nicht streambare Videoinhalte werden nicht als Ergebnis eingeblendet, wodurch gewährleistet werden kann, dass der Anwender die Webseite nicht verlassen muss.

Eine weitere Einschränkung könnten Probleme mit dem Urheberrecht sein, da sich dieses Projekt zunächst ausschließlich im universitären Rahmen bewegt und nicht einer breiten Öffentlichkeit zur Verfügung gestellt wird, sind diese Bedenken vorerst zu vernachlässigen.

Das Streamen der Inhalte kann unter Umständen abhängig von weiteren Software sein, wie z.B. Adobe Flash. Hierdurch können sich insbesondere auf mobilen Geräten Einschränkungen ergeben, schlimmstenfalls ist das Betrachten von Streams auf mobilen Endgeräten nicht möglich.

Eine weitere Herausforderung stellt das Einbinden der Mediatheken dar, weil diese keinerlei öffentlich dokumentierte API’s besitzen. Das Abfragen der Videos, deren Titel, Länge, Ausstrahlungszeitpunkt und weitere Details erfolgt durch aufwändiges Crawlen und Parsen der Mediatheken. Lediglich die Mediatheken ARTE und ZDF bieten die Möglichkeit über präparierte Requests gezielt wohlgeformte Videoinhalte abzufragen.

Ein weiteres Problem stellt der Video-Player VideoJs dar. Dieser wird nicht von allen mobilen Browsers richtig erkannt. Unter Anderem kann hier der Browser “Google Chrome” für Android genannt werden. Diese Browser ersetzten den VideoJS-Player mit dem eigenen Standard Player.

## Benutzerhandbuch und Dokumentation

Nach Abschluss des Projekts wird neben der entwickelten Anwendung eine Dokumentation vorliegen, in der auf die einzelnen Bestandteile eingegangen wird. Ein zusätzliches Nutzerhandbuch ist nicht vorgesehen.

# Spezifische Anforderungen

## (Externe) Schnittstellen

### Benutzerschnittstellen

Als Benutzerschnittstelle dient eine Webseite, die durch den Benutzer über einen Webbrowser aufgerufen wird. Grundlegend soll hier, vor allem im Bezug auf die bereits erwähnten Benutzer von mobilen Geräten, der Ansatz des *mobile-first* verwendet werden. Für die Gestaltung der Nutzerschnittstelle werden bereits bewährte Design-Pattern verwendet. Diese sollen im weiteren Verlauf der Anforderungsanalyse noch näher spezifiziert werden. Als Frameworks für die Umsetzung sollen Laravel und Twitter Bootstrap 3 verwendet werden.

### Hardware-Schnittstellen

Die Suchanfragen werden lokal vom Nutzer formuliert, ggf. über einen serverseitigen Proxy geleitetund und an die Server der Mediatheken gesendet. Die Antworten der Mediatheken an den Server, werden an den Nutzer weitergeleitet und lokal geparst und entsprechend durch den Browser visualisiert.

Beim Streamen wird eine direkte Verbindung zwischen Anwender und Mediathek-Server vermittelt, um die audiovisuellen Inhalte direkt von der Quelle an den Browser des Nutzers zu leiten.

Zur Kommunikation wird das HTTP Protokoll benutzt. Die übermittelten Daten der Mediatheken liegen im String, HTML oder JSON Format vor und werden entsprechend behandelt.

### Software-Schnittstellen

Neben den Schnittstellen zu den einzelnen Mediatheken werden im Projekt weitere Frameworks und Bibliotheken verwendet. Für die Nutzerverwaltung und den persönlichen Bereich wird das PHP-Framework Laravel angewendet. Zur Unterstützung der visuellen Gestaltung, vor allem im Bezug auf die Herangehensweise des *mobile-first,* wird das Bootstrap Framework von Twitter verwendet.

Für die DOM-Naviagtion und Manipulation wird die Javascript Bibliothek jQuery und deren Erweiterung jQuery UI verwendet. Ein für die Webseite angepasster HTML5-Video-Player wird mithilfe der Javascript Bibliothek VideoJS implementiert. Für das Öffnen der Navigation durch seitliches Wischen auf mobilen Geräte wird die Bibliothek SnapJS verwendet, welche ebenfalls auf Javascript basiert. Für die Gestaltung der Beiträge auf der Startseite wird die auf jQuery basierende Bibliothek jQuery.Flipster verwendet. Diese erlaubt das manipulieren der einzelnen Elemente, sodass ein dreidimensionaler Effekt entsteht.

### Kommunikationsschnittstellen

Um das Projekt öffentlich zugänglich zu machen wird ein Webspace bei der Hosting-Firma Strato eingesetzt. Dieser nutzt die PHP-Engine in Version 5.5 und ist über FTP und HTTP erreichbar. Darüber hinaus wird auch eine im Paket integrierte MySQL-Datenbank von Strato genutzt um Daten persistent zu Speichern.

Zur Verschlüsselung der Anmeldeinformationen in der Datenbank wird der interne Hash-Algorithmus des eingesetzten PHP-Frameworks Laravel benutzt.

Da das Projekt zunächst als proof of concept umgesetzt wird ist der Zugang zum Server passwortgeschützt. Für die Anmeldung wird der Benutzername mediathekcrawler und das Passwort praxisseminar2014 benötigt.

## Funktionale Anforderungen

**Auffinden von Sendungen**

Den Nutzern des Mediathek-Crawlers wird die Möglichkeit gegeben gezielt und explorativ nach Inhalten zu suchen

* **Suchfunktion**: Anhand der Eingabe von Schlüsselwörtern in einem Suchfeld können passende Inhalte gefunden und die Ergebnisse aufgelistet werden.
* **Suche anhand von Kategorien**: Es wurden zunächst Kategorien, als Ergebnis einer Wettbewerbsanalyse gebildet. Die Inhalte werden anhand dieser Kategorien gefunden und die Ergebnisse aufgelistet.
* **Filterfunktion**: Neben Kategorien, kann die Suche auch durch Angabe eines bestimmten Zeitraums oder durch die Angabe einer Mindestlänge spezifiziert werden.

**Streamen**

* **Alte sowie aktuelle Videobeiträge:** Die Inhalte der Mediatheken werden auf der Seite des Mediatheks-Crawlers gestreamt.
* **Livestream:** Ein Livestream sendet wenn vorhanden das aktuelle Programm der einzelnen Sender.

**Personalisierungsmöglichkeiten**

Den Nutzern des Mediathek-Crawlers wird mithilfe einer einfachen Login-Funktion eine Möglichkeit gegeben, die Seite an ihre Bedürfnisse anzupassen. Hierfür wurden folgende Funktionen implementiert:

* **Playlists**: Videos werden in Playlists verwaltet werden können. Diese Funktion steht nur registrierten Nutzern zur Verfügung. Durch das Registrieren werden die Playlisten mit der angegebenen E-Mail-Adresse verknüpft und serverseitig gespeichert.
* **Merklisten**: Anhand eines TV-Programms (s.u) sollen zukünftige Sendungen markiert werden können und den Nutzer zum Zeitpunkt der Ausstrahlung daran erinnern. Die Merkliste dient der schnellen Markierung von potentiell interessanten Videos. Diese Funktion steht nur registrierten Nutzern zur Verfügung. Durch das Registrieren wird die Merkliste mit der angegebenen E-Mail-Adresse verknüpft und serverseitig gespeichert.
* **Favoriten**: Die Möglichkeiten Favoriten anzulegen, sowie ein TV-Programm mit Reminderfunktion wurden aufgrund des temporären Ausfalls eines Teammitgliedes und dem daraus resultierenden Mehraufwand für die restlichen Teammitglieder aus dem Funktionsumfang der Applikation entfernt.

**Sonstiges**

* **Mein Konto**:Es gibt eine Übersichtseite, welche für eingeloggte Benutzer verschiedene Informationen übersichtlich darstellt. Die Seite zeigt bis zu fünf Playlisten und bis zu fünf Einträge aus der Merkliste an. Weiterhin ist hier die Möglichkeit gegeben sein persönliches Passwort zu ändern.
* **Menü**: Sobald sich der Benutzer mit einer validen Mailadresse registriert hat und angemeldet ist, kann über den Menübutton zum eigenen Konto, zu den Playlists und zur Merkliste navigiert werden. Durch einen Klick auf “Abmelden” wird der Benutzer auf der Seite abgemeldet und zurück auf die Startseite geleitet.
* **Navigation**:
  + Desktop: Auf der Desktop-Version der Applikation befindet sich die Navigation im Footer jeder Seite. Von dort aus kann auf alle Unterseiten zugegriffen werden. Diese sind in die Kategorien “Rubriken”, “Sender” und “Konto” unterteilt. Durch einen Klick auf die Rubrik “Sport” wird der Benutzer auf die Ergebnisseite umgeleitet und dort werden ihm alle Videos der Mediatheken angezeigt, die der ausgewählten Rubrik entsprechen.
  + Mobil: In der mobilen Version wurde die Navigation mittels Menübutton gelöst. Dieser befindet sich konstant im linken oberen Bereich der Seite neben dem Logo. Die Navigation kann entweder durch Klick auf den Button oder durch wischen über den Bildschirm vom linken Rand aus zur Mitte hin geöffnet werden. Über die mobile Navigation kann ebenfalls auf Konto, Playlisten und Merkliste zugegriffen werden. Eine Übersicht der Sender und Rubriken wird jeweils auf einer neuen Seite angezeigt, um das mobile Menü nicht zu überladen.

**Nice-to-haves:**

Folgende Funktionen wurden von Anfang an als Nice-to-haves geplant schlußendlich aber nicht umgesetzt.

* **Googles Chromecast und Apples AirPlay**: Das gestreamte Video soll auch auf einem TV-Gerät dargestellt werden können.
* **Einbindung weiterer Anbieter** wie beispielsweise  private Sender, Youtube, Vimeo und weitere.
* **Abos**: Mit dieser Funktion sollte der Nutzer die Möglichkeit haben Sendungsreihen zu abonnieren um im persönlichen Kundenkonto einen Überblick über aktuelle Videos der Abos zu erhalten.
* **TV-Programm**: Dem Nutzer sollte ermöglicht werden das aktuelle TV-Programm der eingebundenen Mediatheken einzusehen, um gezielt nach Videos zu suchen.

## Anwendungsfälle (Use Cases)

Tabelle 1: Use Case (Durchsuchen der Plattform)

|  |  |
| --- | --- |
| **Use Case 01** | ***Durchsuchen der Plattform*** |
| **Priorisierung** | *\*\*\** |
| **Beschreibung** | *Der Nutzer soll die Möglichkeit haben eine Sendung auszusuchen.* |
| **Hauptakteur** | *Anwender* |
| **Auslöser (Trigger)** | *Aufruf der Startseite* |
| **Flow of Events**  **(Interaktionsschritte)** | *Der Nutzer klickt sich durch die verschiedenen Kategorien, bis er ein Video findet, das ihm gefällt.* |
| **Alternative Flow/ Erweiterungen**  **(Variierende Interaktionsschritte)** | *a) Alternativ kann der Nutzer für die Suche nach einem konkreten Video auch die Suchfunktion benutzen.*  *b) Eingeloggte Nutzer erhalten zusätzlich Empfehlungen zu möglicherweise interessanten Videos* |
| **Nicht-funktionale Anforderung** |  |

Tabelle 2: Use Case (Wiedergabe einer Sendung)

|  |  |
| --- | --- |
| **Use Case 02** | ***Wiedergabe einer Sendung*** |
| **Priorisierung** | *\*\*\** |
| **Beschreibung** | *Der Nutzer soll die Möglichkeit haben eine Sendung über die Plattform anzusehen* |
| **Hauptakteur** | *Anwender* |
| **Auslöser (Trigger)** | *Aufruf der Startseite* |
| **Flow of Events**  **(Interaktionsschritte)** | *Der Nutzer sucht sich zunächst eine Sendung aus, klickt diese an und startet die Wiedergabe.* |
| **Alternative Flow/ Erweiterungen**  **(Variierende Interaktionsschritte)** | *Falls die Sendung aus technischen Gründen nicht aufrufbar ist, wird dem Nutzer eine Fehlermeldung angezeigt* |
| **Nicht-funktionale Anforderung** | *Die Wiedergabe der Sendung soll direkt auf der Plattform stattfinden.* |

Tabelle 3: Use Case (Zusammenstellen des Abendprogrmms)

|  |  |
| --- | --- |
| **Use Case 03** | ***Zusammenstellen des Abendprogramms*** |
| **Priorisierung** | *\*\** |
| **Beschreibung** | *Der Nutzer soll die Möglichkeit haben sich sein Abendprogramm mit der Anwendung zusammenzustellen* |
| **Hauptakteur** | *Anwender* |
| **Auslöser (Trigger)** | *Anlegen einer Playlist* |
| **Flow of Events**  **(Interaktionsschritte)** | *Der Nutzer sucht sich zunächst seine gewünschten Sendungen aus und fügt diese einer Playlist hinzu. Anschließend kann die Playlist abgespielt werden, wodurch keine zusätzliche Interaktion nötig ist.* |
| **Alternative Flow/ Erweiterungen**  **(Variierende Interaktionsschritte)** |  |
| **Nicht-funktionale Anforderung** | *Der Nutzer soll auch bei Sendungen unterschiedlicher Herkunft seine Playlist möglichst unterbrechungsfrei ansehen können.* |

Tabelle 4: Use Case (Registrieren)

|  |  |
| --- | --- |
| **Use Case 06** | ***Registrierung*** |
| **Priorisierung** | *\*\** |
| **Beschreibung** | *Der Nutzer soll die Möglichkeit haben sich zu registrieren* |
| **Hauptakteur** | *Anwender* |
| **Auslöser (Trigger)** | *Registrieren* |
| **Flow of Events**  **(Interaktionsschritte)** | *Wenn der Nutzer die Funktion aufruft soll ihm ein E-Mail- und Passwort-Feld angezeigt werden. Durch Klick auf den “registrieren”-Button werden diese Daten überprüft und gespeichert. Der Nutzer wird über den (Miss-)Erfolg informiert.* |
| **Alternative Flow/ Erweiterungen**  **(Variierende Interaktionsschritte)** |  |
| **Nicht-funktionale Anforderung** |  |

# Nicht-funktionale Anforderungen

## Performance

Performance-Probleme ergeben sich auf Entwicklerseite durch den zum Einsatz kommenden Server. Aufgrund des universitären Kontexts ist allerdings nicht mit einem hohen Andrang während der Entwicklung und den Tests zu rechnen, weshalb dieser Punkt im Rahmen des Masterkurses “Praxisseminar” zu vernachlässigen ist.

Da es sich um eine Webapp handelt ist seitens der Nutzer eine stabile, angemessen schnelle Internetverbindung Grundvoraussetzung für das Streamen von Videos. Ebenfalls sollte die genutzte Grafikkarte in der Lage sein die Streams zu verarbeiten und ruckelfrei darzustellen, da sonst das Nutzererlebnis stark leidet. Hier kann es, je nach Hardwareausstattung und temporärer Auslastung, bei mobilen Endgeräten zu Problemen kommen. Allerdings trifft dies wohl nur auf ältere Modelle zu und sollte nur in Einzelfällen auftreten, da die Hardware von Streaming-tauglichen mobilen Endgeräten potent genug sein sollte, unterstützte Technologien zu verarbeiten.

Defacto dauert das Crawlen und Parsen der Videoinhalte der Mediatheken unterschiedlich lange. Dies ist zum einen Abhängig von der Komplexität der Suchanfrage und deren Umfang. Die Suche nach einem Schlüsselwort wie “Tatort” in einer einzigen Mediathek wird schneller abgeschlossen, als die Kombination eines Schlüsselworts mit mehreren Mediatheken und weiteren Filtereinstellungen. Selbstredend können die Filtereinstellungen die Suchanfrage allerdings auch gezielt einschränken und somit stark beschleunigen. Die Dauer der Suche und Darstellung der Suchergebnisse ist letztlich Abhängig von der zugrundeliegenden Hardware des Endanwenders. Insbesondere Mobilgeräte oder ältere Desktophardware bieten keine optimale Grundvorraussetzung für komplexere Suchvorgänge und sind deshalb längeren Lade-/Wartezeiten ausgesetzt.

Zur Minderung dieses Problems wurde Entwicklerseitig hohen Wert auf performanten Code gelegt, was letztlich anhand von Usability Tests auf Desktop- wie mobiler Hardware kontrolliert wurde. Zusätzlich werden in der Production Environment auf dem Server minimierte Dateien benutzt. Dabei wird die Dateigröße von Skripten und Stylesheets durch Entfernung von Zeilenumbrüchen und Leerzeichen, sowie der Verkürzung von häufigen Zeichenketten verkleinert.

## Sicherheit

Die grundlegende Funktionalität des Systems wird auch nicht-authentifizierten Benutzern ermöglicht. Für den personalisierbaren Bereich sollen sich die Nutzer mit Email-Adresse und Passwort anmelden. Bei der Registrierung neuer Nutzer muss lediglich eine gültige Email-Adresse und ein Passwort festgelegt werden.

Die Anmeldedaten werden mit dem im eingesetzten PHP-Framework Laravel integrierten Hash-Algorithmus verschlüsselt. Diese grundlegende Verschlüsselung sollte ausreichend Sicherheit bieten, da weder persönliche Daten (mit Ausnahme einer E-Mail-Adresse) noch andere sensible Daten erfasst werden. Mit dem Benutzerkonto werden lediglich Daten wie die Playlist und gemerkte Videos verknüpft.

## Qualität

### Ausfallsicherheit

Aufgrund des Kontexts eines Universitäts-Projekts ist von einer hohen Ausfallsicherheit auszugehen, da die anfallende Last für einen physikalischen oder virtuellen Server stark eingegrenzt ist. Im Verlauf des Projekts ist noch zu klären, ob das Projekt einer breiten Öffentlichkeit zugänglich gemacht werden kann. Dies hängt von, zu klärenden, rechtlichen Fragen ab. Die Ausfallsicherheit hängt darüber hinaus stark von den Servern der eingebundenen Mediatheken ab, da diese kontaktiert werden um Videoinhalte anzubieten. Allerdings kann deren Erreichbarkeit nicht gewährleistet werden. Ebenfalls kann eine Umstrukturierung der Webseiten, bzw. der zugrundeliegenden Infrastruktur dieser Mediatheken zu Ausfällen führen, da in einem solchen Falle der Crawl- und Parse-Vorgang angepasst, sprich neu gecodet werden muss.

### Erreichbarkeit

Wie im Punkt “Ausfallsicherheit” geschildert, wird es bezüglich der Erreichbarkeit ebenfalls kaum Einschränkungen geben, da die Anwendung zunächst im universitären Rahmen betrieben wird und die Erreichbarkeit des Servers somit weniger von der anfallenden Last als von externer Seite (Stromanbieter, Internetprovider) abhängig ist.

### Wartbarkeit

Durch eine umfassende und verständliche Dokumentation, sowie durch ausreichend annotierten Code wird eine solide Basis für die Wartbarkeit der Software gewährleistet.

Des Weiteren wird auf einen modularen Aufbau gesetzt, wodurch sich bspw. Änderungen am Design schnell und effizient durchführen lassen.

### Erweiterbarkeit

Auf Grundlage der Dokumentation und des annotierten Codes ist eine grundsätzliche Erweiterung der Software möglich. Das Hinzufügen von Mediatheken, sprich die Anbindung weiterer API’s soll ohne größere Änderungen am bestehenden Code ermöglicht werden. Auch durch den bereits erwähnten modularen Aufbau, ist eine grundsätzliche Erweiterbarkeit gewährleistet.

### Usability

Die Plattform soll leicht verständlich und einfach zu Bedienen sein. Dazu werden in der Designphase Usability-Engineering Methoden genutzt und die Webapp nach aktuell geltenden Usability-Standards (ISO 9241-210) gestaltet.

## Datenmodell

Für das Benutzerkonto und den persönlichen Bereich soll eine MySQL Datenbank verwendet werden. Neben einer Tabelle für die einzelnen Benutzer und deren Anmeldeinformationen und IDs und einer Tabelle in der die Playlisten aller Benutzer gespeichert werden gibt es eine Tabelle, welche alle Videos speichert, die in Playlisten oder Merklisten eingetragen sind. Die Abhängigkeiten ob sich solch ein Video in der Merkliste oder ein oder mehrerer Playlisten befindet ergibt sich aus je einer Spalte für die Benutzer-Id und der Playlisten-Id. Das genaue Datenbankschema wird in Abbildung 3 dargelegt.

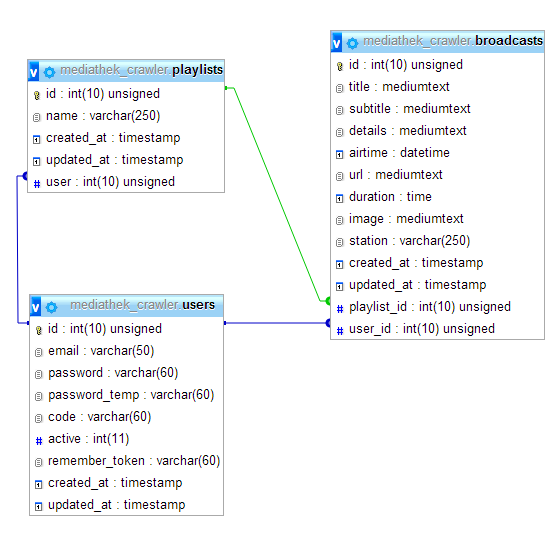


Abbildung 3: Datenbankschema

# Risiken

Es bestehen Risiken hinsichtlich der Betrachtung von Videos auf mobilen Geräten, da nicht sichergestellt werden kann, dass die Wiedergabefunktion aufgrund der unterschiedlichen Dateitypen von allen Geräten unterstützt wird. Ein weiteres Risiko stellt die verfügbare Bandbreite bei der Verbindung mit dem System dar. Insbesondere im mobilen Kontext können hier erhöhte Ladezeiten auftreten, da das Durchsuchen der Mediatheken live von statten geht. Auch das Streamen von Videos kann bei schlechter Verbindung problematisch werden. Dem wird versucht durch die Auslieferung von Videos in kleineren Formaten entgegen zu wirken. Zudem stellt die Kompatibilität mit den einzelnen verfügbaren Browsern, sowie die Sicherstellung einer plattformübergreifenden Funktionalität, ein Risiko dar.

Als problematisch stellt sich auch die Umsetzung des responsive design dar, dabei müssen durch die teils hohe Komplexität der grafischen Elemente auf den einzelnen Seiten Kompromisse für kleinere Displaygrößen eingegangen werden. Das Durchsuchen der einzelnen Mediatheken kann, insbesondere im Hinblick auf die Verfügbarkeit der Inhalte und die Zugriffsmöglichkeiten vom System, zu Problemen führen. Weitere Probleme können durch fehlende Daten entstehen, welche von den Mediatheken bereitgestellt werden, da dadurch geplante Funktionen möglicherweise garnicht oder nur bedingt umgesetzt werden können. Beispielhaft ist hier die Mediathek des Senders SRF zu erwähnen, von welcher keinerlei Informationen über die Länge eines Videobeitrags zu entnehmen ist. Lediglich durch das Anspielen eines Videos, kann dessen Länge festgestellt werden. Diese Information steht dem Nutzer deswegen nicht wie sonst bereits vorher zur Verfügung.

Aus persönlichen Gründen konnte ein Projektmitarbeiter während des Projekts für einen Zeitraum von zwei Wochen nicht eingesetzt werden. Dadurch mussten teilweise anstehende Aufgaben und Arbeitspakete unter den übrigen Mitarbeitern aufgeteilt werden