

**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 27.05.2014

Inhalt

[0 Versionsgeschichte 5](#_Toc388971979)

[1 Einführung 5](#_Toc388971980)

[1.1 Zweck des Dokuments 5](#_Toc388971981)

[1.2 Projektumfang (Scope) 5](#_Toc388971982)

[1.3 Definitionen, Akronyme und Abkürzungen 5](#_Toc388971983)

[1.4 Referenzen und Quellenangaben 6](#_Toc388971984)

[1.5 Übersicht über das Dokument 6](#_Toc388971985)

[2 Allgemeine Beschreibung 6](#_Toc388971986)

[2.1 Produktperspektive 6](#_Toc388971987)

[2.2 Produktfunktionen 7](#_Toc388971988)

[2.3 Benutzergruppen und Charakteristika 8](#_Toc388971989)

[2.4 Zielplattform 8](#_Toc388971990)

[2.5 Einschränkungen, Annahmen und Abhängigkeiten 9](#_Toc388971991)

[2.6 Benutzerhandbuch und Dokumentation 9](#_Toc388971992)

[3 Spezifische Anforderungen 9](#_Toc388971993)

[3.1 (Externe) Schnittstellen 9](#_Toc388971994)

[3.1.1 Benutzerschnittstellen 9](#_Toc388971995)

[3.1.2 Hardware-Schnittstellen 10](#_Toc388971996)

[3.1.3 Software-Schnittstellen 10](#_Toc388971997)

[3.1.4 Kommunikationsschnittstellen 10](#_Toc388971998)

[3.2 Funktionale Anforderungen 10](#_Toc388971999)

[3.3 Anwendungsfälle (Use Cases) 12](#_Toc388972000)

[4 Nicht-funktionale Anforderungen 14](#_Toc388972001)

[4.1 Performance 14](#_Toc388972002)

[4.2 Sicherheit 15](#_Toc388972003)

[4.3 Qualität 15](#_Toc388972004)

[4.3.1 Ausfallsicherheit 15](#_Toc388972005)

[4.3.2 Erreichbarkeit 15](#_Toc388972006)

[4.3.3 Wartbarkeit 15](#_Toc388972007)

[4.3.4 Erweiterbarkeit 15](#_Toc388972008)

[4.3.5 Usability 16](#_Toc388972009)

[4.4 Datenmodell 16](#_Toc388972010)

[4.5 Sonstige Anforderungen 16](#_Toc388972011)

Abbildungen

[Abbildung 1: Datenflussdiagramm 7](#_Toc388972012)

[Abbildung 2: Funktionsüberblick 8](#_Toc388972013)

Tabellen

Tabelle 1: Versionsgeschichte 5

Tabelle 2: Definitionen, Akronyme und Abkürzungen 5

Tabelle 3: Use Case (Durchsuchen der Plattform) 12

Tabelle 4: Use Case (Wiedergabe einer Sendung) 12

Tabelle 5: Use Case (Zusammenstellen des Abendprogrmms) 13

Tabelle 6: Use Case (Abos) 13

Tabelle 7: Use Case (TV-Programm) 14

# Versionsgeschichte

Tabelle : Versionsgeschichte

|  |  |  |
| --- | --- | --- |
| **Version** | **Datum** | **Beschreibung** |
| 0.1 | 27.05.2014 | Initialisierung |

# 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) ~~alle interessierten Personen~~. 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 abonniert, favorisiert, 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

Tabelle : 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 |
| RTMP | Real Time Messaging Protocol. Übertragung von VIdeo/Audio von einem Media-Server zu einem Flash Player |
| RTMPS | RTMP auf Basis von Https für sichere Verbindungen |
| RTSP | RealTime Streaming Protocol. Netzwerkprotokoll zur kontinuierlichen Übertragung von audiovisuellen Inhalten. |
| 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.

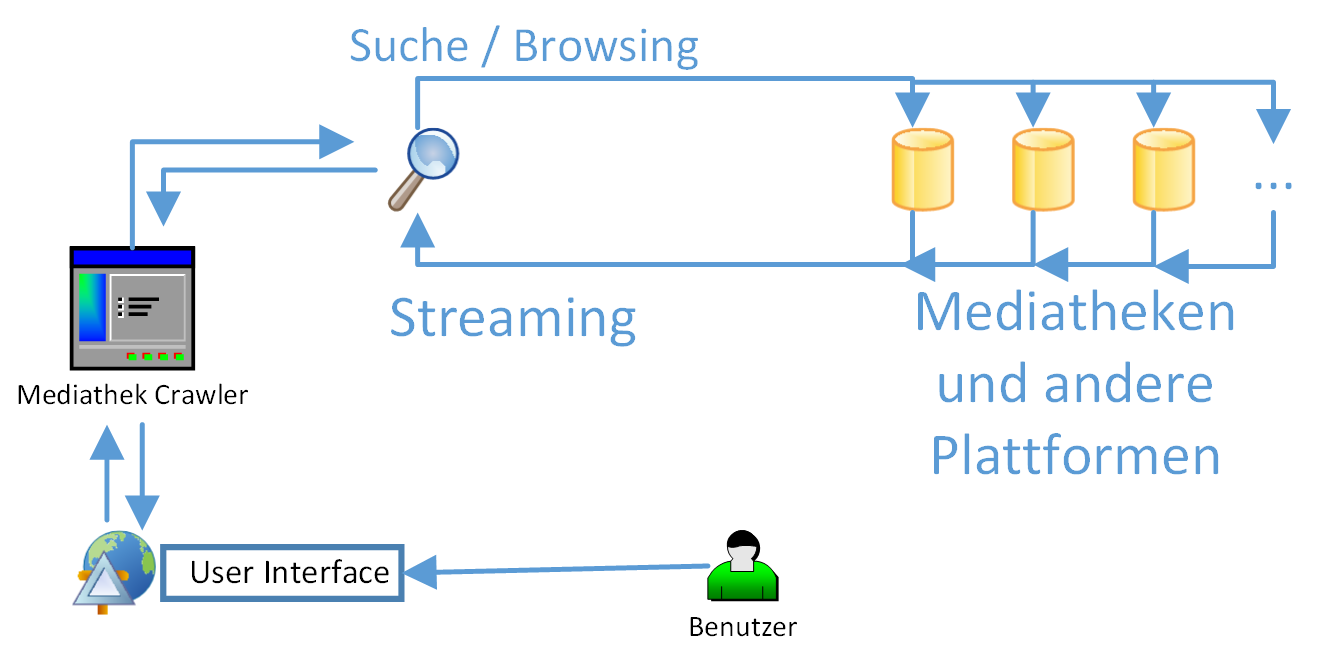


Abbildung 1: Datenflussdiagramm

## Produktfunktionen

Der Benutzer soll durch das Produkt in der Lage sein, ein persönliches TV-Programm zusammenzustellen, zu streamen 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 sowie favorisiert werden. Favorisierte Inhalte können in einem Login-geschützten Bereich verwaltet werden. Dieser Bereich bietet weiterhin einige Funktionen, wie z.B. Erinnerungen an neue Episoden einer Serie oder Hinweise zu Ausstahlungsterminen. Durch das Anlegen von Playlists, kann der Nutzer mehrere Inhalte am Stück abspielen, ohne dass weitere Interaktion nötig ist.

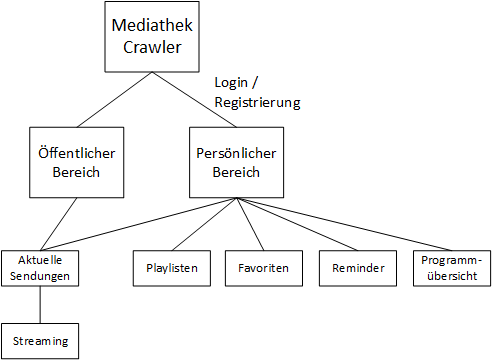


Abbildung 2: Funktionsüberblick

## 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 vermeiden.

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 Tablet die Seite in vollem Umfang zu nutzen.

## Zielplattform

Das Projekt soll als Webapp umgesetzt werden und ist damit plattformunabhängig. Die Webbapp soll in möglichst allen aktuellen Browsern funktionsfähig sein. Desweiteren soll es die Möglichkeit geben, die Plattform auch auf mobilen Geräten zu verwenden.

Durch die Verfolgung des Cross-OS, mobile-first Implementierungs-Ansatzes bietet sich die Möglichkeit, ein sehr breites Publikum zu erreichen.

## 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. Diese können neben dem Ausschluss dieser Mediatheken bis hin zum Verlinken auf die entsprechende Webseite reichen. 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.

Des Weiteren ist es zunächst noch offen in wie weit sich die API’s der Mediatheken unterscheiden und mit welchen API’s letztlich ein gemeinsamer Funktionsumfang angeboten werden kann. Ebenfalls ist es noch offen, ob API’s für die einzubindenen Mediatheken angeboten werden, oder ob diese über Http-Requests bzw. andere Protokolle erschlossen werden müssen.

## 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 Tablet-Benutzer, der Ansatz des *mobile-first* verwendet werden. Für die Gestaltung der Nutzerschnittstelle sollen bereits bewährte Design-Pattern verwendet werden. Diese sollen im weiteren Verlauf der Anforderungsanalyse noch näher spezifiziert werden. Als Framework für die Umsetzung soll Twitter Bootstrap 3 verwendet werden.

### Hardware-Schnittstellen

Die Suchanfragen werden lokal vom Nutzer formuliert, ggf. serverseitig geparst und über Http an die Server der Mediatheken gesendet. Die Antworten der Mediatheken an den Server, werden wiederum geparst und entsprechend beim Nutzer lokal 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 werden voraussichtlich die Protokolle Http, RTMP(S), RTSP, FTP oder weitere Streaming Protokolle benutzt - je nach Anforderungen und Möglichkeiten der Mediatheken.

Nähere Angaben zu den verwendeten Protokollen werden in späteren Versionen dieses Dokuments gemacht.

### Software-Schnittstellen

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

Weitere Abhängigkeiten bezüglich weiterführender Bibliotheken werden in späteren Versionen näher erläutert, wenn die Anforderungen daran weiter spezifiziert werden konnten.

### Kommunikationsschnittstellen

Um auf den Mediathek-Crawler zugreifen zu können wird ein Webserver benötigt. Dieser sollte möglichst schnell und zuverlässig sein. Neben den Standard-Protokollen FTP und HTTP sollte auch eine aktuelle Version der PHP-Engine vorhanden sein.

Zur Verschlüsselung der Anmeldeinformationen soll ein standardisiertes Verfahren des jeweiligen Frameworks angewendet werden.

## Funktionale Anforderungen

**Auffinden von Sendungen**

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

* **Suchfunktion**: Anhand der Eingabe von Schlüsselwörtern in einem Suchfeld sollen passende Inhalte gefunden und die Ergebnisse aufgelistet werden.
* **Suche anhand von Kategorien**: Es sollen zunächst Kategorien, als Ergebnis einer Wettbewerbsanalyse gebildet werden. Die Inhalte sollen dann anhand dieser Kategorien gefunden und die Ergebnisse aufgelistet werden.
* **Suche anhand TV-Programm:** Inhalten sollen auch mithilfe einer Art TV-Programm durchsucht und die Ergebnisse aufgelistet werden. Es soll so die Möglichkeit geben  anhand von Eingaben wie Tag, Zeit und Sender eine verpasste oder interessante Sendung anzusehen.

**Streamen**

* **Alte Sendungen:** Die Inhalte der Mediatheken sollen auf der Seite des Mediatheks-Crawlers gestreamt werden. Wenn möglich soll das Verlinken auf die Originalseite vermieden werden.
* **Livestream:** Ein Livestream sendet wenn vorhanden das aktuelle Programm der einzelnen Sender.

**Personalisierungsmöglichkeiten**

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

* **Playlists**: Videos sollen in Playlists verwaltet werden können.
* **Merklisten**: Anhand eines TV-Programms (s.u) sollen zukünftige Sendungen markiert werden können und den Nutzer zum Zeitpunkt der Ausstrahlung daran erinnern.
* **Favoritenlisten**:  Nutzer sollen ihre Lieblingssender und -Sendungen speichern können.

**Sonstiges**

* **Dashboard**: Es soll eine Übersichtseite geben, welche verschiedene Informationen übersichtlich darstellt.
* **TV-Programm**: Die Seite soll das vergangene, aktuelle und zukünftige TV-Programm anzeigen.

**Nice-to-haves:**

Folgende Funktionen werden je nach Projektverlauf und “Möglichkeiten” implementiert.

* **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.

## Anwendungsfälle (Use Cases)

Tabelle : 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 : 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 : 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 : Use Case (Abos)

|  |  |
| --- | --- |
| **Use Case 04** | ***Abos*** |
| **Priorisierung** | *\*\** |
| **Beschreibung** | *Der Nutzer soll die Möglichkeit haben regelmäßige Sendungen zu abonnieren.* |
| **Hauptakteur** | *Anwender* |
| **Auslöser (Trigger)** | *Anlegen einer Abos* |
| **Flow of Events**  **(Interaktionsschritte)** | *Der Nutzer sucht sich zunächst seine gewünschten Sendungen aus und abonniert diese (falls verfügbar). Das System erinnert den Nutzer per E-Mail, wenn eine weitere Folge der Sendung verfügbar ist.* |
| **Alternative Flow/ Erweiterungen**  **(Variierende Interaktionsschritte)** |  |
| **Nicht-funktionale Anforderung** | *Der Nutzer soll bei regelmäßigen Sendungen erinnert werden können.* |

Tabelle 7: Use Case (TV-Programm)

|  |  |
| --- | --- |
| **Use Case 05** | ***TV-Programm*** |
| **Priorisierung** | *\** |
| **Beschreibung** | *Der Nutzer soll die Möglichkeit haben das aktuelle TV-Programm einzusehen.* |
| **Hauptakteur** | *Anwender* |
| **Auslöser (Trigger)** | *Übersicht über TV-Programm* |
| **Flow of Events**  **(Interaktionsschritte)** | *Wenn der Nutzer die Funktion aufruft soll ihm das TV-Programm angezeigt werden. Über Filtermöglichkeiten soll er seine Anfrage spezifizieren können.* |
| **Alternative Flow/ Erweiterungen**  **(Variierende Interaktionsschritte)** |  |
| **Nicht-funktionale Anforderung** |  |

# Nicht-funktionale Anforderungen

## Performance

Performance-Probleme ergeben sich auf Entwicklerseite ~~maximal~~ 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 solange vernachlässigt werden kann, bis das Projekt nach Abschluss einer breiten Öffentlichkeit zugänglich gemacht wird.

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.

## Sicherheit

Die grundlegende Funktionalität des Systems soll auch nicht-authentifizierten Benutzern ermöglicht werden. Für den personalisierbaren Bereich, in dem unter anderem Playlists oder Favoritenlisten erstellt werden können, 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 mittels Message-Digest Algorithm 5 (MD5) 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, Favoriten und Erinnerungen verknüpft.

## Qualität

### Ausfallsicherheit

Aufgrund des Kontexts eines Universitäts-Projekts ist zunächst nicht davon auszugehen, dass die Anwendung für die breite Öffentlichkeit zugänglich gemacht wird. Somit ist von einer hohen Ausfallsicherheit auszugehen, da die anfallende Last für einen physikalischen oder virtuellen Server stark eingegrenzt sein wird.

### 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 gestaltet.

## Datenmodell

Für das Benutzerkonto und den persönlichen Bereich soll eine MySQL Datenbank verwendet werden. Neben Tabellen für die einzelnen Benutzer und deren Anmeldeinformationen und IDs werden weitere Tabellen für die Funktionen der Favoriten und Playlists benötigt.

Das genaue Datenbankschema kann allerdings erst in einer späteren Version dargelegt werden, da noch nicht alle Anforderungen daran genauestens ermittelt sind.

## Sonstige Anforderungen

Derzeit sind keine weiteren Anforderungen bekannt.