

Projektdokumentation

im Studiengang

AIN

PIPCO

Private IP Camera Observation

Referent : Prof. Dr. Elmar Cochlovius

Vorgelegt am : 13.01.2019

Abstract

This is the project documentation of a group from the course of studies Computer Science at the Hochschule Furtwangen University located in Germany. The project is taking place in the sixth semester and the group is consisting of four members. The project is about the implementation of a software for remote camera observation, while a special focus is placed on the interchangeability of the hardware in use. Furthermore, registered users of the product shall be informed automatically when a motion is detected by the software. Many providers of similar solutions are using cloud based services to tackle such tasks. By making use of more direct connections between the IP camera and the end consumer, this project aims to achieve lower latency.

Dies ist die Dokumentation zum Semesterprojekt einer vierköpfigen Gruppe aus dem sechsten Semester des Studienganges Allgemeine Informatik der Hochschule in Furtwangen. Bei dem Projekt geht es um die Implementierung einer Software zur Kameraüberwachung, wobei ein besonderer Fokus auf die Austauschbarkeit der Hardware gelegt wird. Zudem sollen durch eine Bewegungserkennung ausgelöste Benachrichtigungen automatisch an registrierte Nutzer versendet werden können. Viele Anbieter ähnlicher Softwarelösungen greifen bei der Umsetzung auf Cloud-Dienste zurück. Durch eine direktere Verbindung zwischen IP-Kamera und Endanwender sollen zudem geringere Latenzzeiten als bei zuvor genannten, kommerziellen Produkten erzielt werden.

Inhaltsverzeichnis

Abstract	i
Inhaltsverzeichnis	v
Abbildungsverzeichnis	vii
Tabellenverzeichnis	ix
Abkürzungsverzeichnis	xi
1 Einleitung	1
1.1 Rahmenbedingungen	1
2 Installation	3
2.1 System	3
2.2 Backend	3
2.3 Frontend	3
2.4 Run on Startup	4
3 Front-End	7
3.1 Angular	7
3.1.1 Begriffe	8
3.2 Bausteine	9
3.2.1 AppModule	9
3.2.2 RoutingModule	9
3.2.3 AppComponent	9
3.2.4 HeaderComponent	9

3.2.5	LoginComponent	10
3.2.6	MainPageComponent	10
3.2.7	RangeSliderComponent	10
3.2.8	VideoComponent	11
3.2.9	VideoSettingsComponent	11
3.2.10	TitleBarComponent	12
3.2.11	EventLogComponent	12
3.2.12	EmailNotificationComponent	12
3.2.13	StatusButtonComponent	13
3.2.14	SettingsPageComponent	13
3.2.15	AuthService	13
3.2.16	SettingsService	14
3.2.17	EmailService	14
3.2.18	EventService	14
3.2.19	AuthGuard	14
3.2.20	Model-Interfaces	14
3.3	Komponenten-Service-Diagramm	15
4	Tests	17
4.1	Ziele	17
4.2	Rahmenbedingungen	17
4.3	Teststrategie	17
4.4	Testen des Front-Ends	18
4.4.1	Unit-Tests	18
4.4.2	Manuelle Tests	20
4.5	Testen des Gesamtsystems	36
5	Ausblick	37

6	Fazit	39
	Literaturverzeichnis	41
	Eidesstattliche Erklärung	43

Abbildungsverzeichnis

Abbildung 1: Komponenten-Service-Diagramm zum Front-End	15
Abbildung 2: Ergebnisse der automatisierten Front-End-Tests	19

Tabellenverzeichnis

Tabelle 1: Input- und Output-Variablen der RangeSlider-Komponente	11
Tabelle 2: Beschreibung der Model-Interfaces	16
Tabelle 3: Testarten der unterschiedlichen Softwareteile	18
Tabelle 4: Eingestellte Werte vor jedem manuellen Test	20
Tabelle 5: Manuelle Front-End-Tests	36

Abkürzungsverzeichnis

IP Internet Protocol

CLI Command Line Interface

URL Uniform Resource Locator

MIT Massachusetts Institute of Technology

CSS Cascading Style Sheets

HTML Hyper Text Markup Language

MJPEG Motion Joint Photographic Experts Group

1 Einleitung

1.1 Rahmenbedingungen

Dieses Projekt stellt das Semesterprojekt von vier Studenten des Studienganges Allgemeine Informatik der Hochschule in Furtwangen dar. Es handelt sich dabei um das zweite Semesterprojekt, welches im sechsten Semester stattfindet.

Ziel des Projektes ist es, eine Software zur Überwachung mittels IP-Kamera zu implementieren, wobei die genutzte Hardware austauschbar bleiben soll. Die Anwendung soll die Fähigkeit besitzen, Bewegungen im Kameralivestream zu detektieren und zuvor hinterlegte Nutzer per E-Mail über die erkannten Bewegungen in Kenntnis zu setzen. Außerdem sollen Aufnahmen dieser Bewegungen erstellt und für den Endanwender einsehbar hinterlegt werden. Neben diversen Einstellungsmöglichkeiten für den Nutzer, wie zum Beispiel für die Sensitivität der Bewegungserkennung oder einer maximalen Anzahl an gespeicherten Aufnahmen soll die Anwendung über eine benutzerfreundliche Weboberfläche mit Login-Maske verfügen.

Unter der Betreuung von Prof. Dr. Elmar Cochlovius und Judith Jakob wurde das Projekt weitestgehend selbstorganisiert durchgeführt. Ein für Testzwecke erforderlicher Hardware-Aufbau konnte im Smart-Home-Labor am Campus in Furtwangen genutzt werden. Dort waren auch ähnliche Lösungen von kommerziellen Anbietern vorhanden, welche während dem Projekt als Referenzen gedient haben.

2 Installation

2.1 System

Die Installationsanweisungen wurden auf einem Ubuntu Server der Version 18.10 durchgeführt.

Hierzu wurde ein bereits installiertes Image für VirtualBox von <https://www.osboxes.org/ubuntu-server/> verwendet.

2.2 Backend

Für das Backend muss OpenCV, sowie der Flask-Webserver mit allen notwendigen Modulen installiert werden. Die Lightweight Installation von OpenCV, welche einfach mit pip installiert werden kann, enthält nicht den passenden Encoder für mp4, weshalb der aktuelle Stand selbst geladen und compiliert werden muss. Hierzu kann folgende Anleitung verwendet werden:

<https://www.pyimagesearch.com/2018/05/28/ubuntu-18-04-how-to-install-opencv/>
Zusätzlich zur Installation von OpenCV muss Flask mit pip installiert werden. Hierzu muss wie in der Anleitung beschrieben .bashrc mit

```
1 source ~/.bashrc
```

geladen werden. Anschließend kann mit

```
1 workon cv
```

in der Virtuellen Python-Umgebung gearbeitet bzw. flask wie folgt installiert werden:

```
1 pip install flask flask-cors
```

Nun muss nur noch das Repository ausgecheckt und ausgeführt werden.

```
1 git clone https://github.com/PIPC0-1819/PIPC0-Backend
  cd PIPC0-Backend
3 python Main.py
```

2.3 Frontend

Für das Frontend wird Node.js, npm, sowie Angular verwendet.

```
1 apt-get install npm nodejs
  sudo npm install -g npm@latest
3 sudo npm install -g @angular/cli
```

Repository auschecken und restliche dependencies installieren:

```
1 git clone https://github.com/PIPCO-1819/PIPCO-Frontend.git
  cd PIPCO-Frontend
3 npm install
```

Anschließend den Server wie folgt starten:

```
1 ng serve --host 0.0.0.0
```

2.4 Run on Startup

start_backend.sh in PIPCO-Backend

```
1 #!/bin/sh
  printf "<LOGIN>\n<PASSWORD>\n" | \
3 /home/osboxes/.virtualenvs/cv/bin/python Main.py
```

start_frontend.sh in PIPCO-Frontend

```
1 #!/bin/sh
  printf "<LOGIN>\n<PASSWORD>\n" | \
3 /home/osboxes/.virtualenvs/cv/bin/python Main.py
```

start_frontend.sh in PIPCO-Frontend

```
1 #!/bin/sh
  ng serve --host 0.0.0.0
```

start_pipco.sh

```
#!/bin/bash
2 screen -dmS frontend bash -c \
  'cd /home/osboxes/PIPCO-Frontend; ./start_frontend.sh'
4 screen -dmS backend bash -c \
  'cd /home/osboxes/PIPCO-Backend; ./start_backend.sh'
```

rc.local bei Start des Systems ausführen

```
1 printf '%s\n' '#!/bin/bash' 'exit 0' | sudo tee -a /etc/rc.local
  sudo chmod +x /etc/rc.local
```

Skript zu rc.local hinzufügen

```
...  
2 /home/osboxes/start_pipco.sh  
exit 0
```


3 Front-End

Da zwei der vier Projektteilnehmer bereits im Praxissemester Erfahrungen damit gesammelt haben, fiel unsere Wahl bei den Technologien für unser Front-End auf Angular. Auf diese Weise konnten wir produktiver arbeiten und deutlich übersichtlicheren Code produzieren. Grundlegende Informationen rund um Angular sowie ein Tutorial zur Entwicklung mit Angular gibt es unter <https://angular.io/>, der offiziellen Website zum Framework.

3.1 Angular

Angular ist ein unter der sehr freizügigen MIT-Lizenz verfügbares, auf TypeScript basierendes Front-End-Framework für Webanwendungen, wobei die Entwicklung dieser Software von Google geleitet wird. Dieses Framework ist grundsätzlich Client-seitig, was bedeutet, dass unter anderem Darstellung sowie Strukturierung von Inhalten beim Anwender und nicht auf der Host-Maschine berechnet werden. Eine Kommunikation mit dem Server findet demnach nur dann statt, wenn neue Inhalte abgerufen werden, oder wenn ein weiterer Datenaustausch vom Entwickler vorgesehen ist. Das hat den Vorteil, dass die Kapazitäten des Servers geschont werden.

Neben den offensichtlichen Vorteilen eines Frameworks, wie zum Beispiel dem Steigern der Produktivität des Entwicklers durch die Abstraktion häufig auftretender Problemstellungen, bietet Angular den Vorteil einer komponentenorientierten Herangehensweise bei der Strukturierung von damit erstellten Webanwendungen. Durch diese Unterteilung semantisch zusammengehöriger Codebausteine wird eine ansonsten komplexe Anwendung übersichtlicher und damit wartbarer. Zudem können solche Komponenten aufgrund ihrer Kapselung deutlich einfacher getestet oder auch an anderer Stelle wiederverwendet werden. Einer der Hauptgründe dafür, dass in Angular eine so strikte Trennung einzelner Komponenten überhaupt möglich ist, stellt dabei die fundamentale Unterstützung von Dependency Injection dar.

Durch die Verwendung der JavaScript-Spracherweiterung TypeScript als Primärsprache des Frameworks profitieren Angular-Entwickler zudem von den Vorteilen der Objektorientierung. Zusätzlich wurde in TypeScript eine statische Typisierung für Variablen eingeführt, was dem Entwickler dabei unterstützt, dahingehende Fehler bereits beim Bauen der Anwendung aufzudecken. [OWA19]

3.1.1 Begriffe

Um Neulingen in Sachen Angular einen leichteren Einstieg zu bereiten, werden im folgenden einige Kernbegriffe im Bezug auf unser Projekt geschildert.

3.1.1.1 Components

Eine Angular-Component spiegelt in der Regel ein beliebig kleines Element in der Oberfläche einer Website dar. Eine Angular-Weboberfläche besteht ausschließlich aus einzelnen Components. Jede Component umfasst im Projekt drei Dateien, welche die Funktionalität der Komponente zur Verfügung stellen. Es gibt eine HTML-Datei für die HTML-Struktur, eine CSS-Datei für das Styling sowie eine TypeScript-Datei für die Dynamik der Inhalte.

3.1.1.2 Services

Angular-Services dienen in der Regel dazu, Daten mittels Http-Requests zu beschaffen und den Components der Anwendung zur Verfügung zu stellen. Dabei werden diese Services nicht direkt von den Komponenten erzeugt, sondern mittels dependency injection eingeschleust. Somit können unnötige Mehrfachinitialisierungen vermieden werden. Außerdem kann der Service damit zu einem für das Angular-Framework optimalen Zeitpunkt erzeugt werden. Ein Testen von Services nutzenden Komponenten kann durch das Verwenden der dependency injection ebenfalls besser umgesetzt werden, ohne auf die Implementierung der Services angewiesen zu sein, indem statt der eigentlichen Services Mock-Objekte injiziert werden.

3.1.1.3 Guards

Die Seitennavigation kann bei Angular, so wie es auch in diesem Projekt der Fall ist, mittels URL-Routen festgelegt werden. Sobald dann eine bestimmte URL aufgerufen wird, wird eine vordefinierte Komponente angezeigt. Damit manche Routen nur unter bestimmten Umständen erreicht werden können, kann man Guards verwenden. Diese prüfen dann beim Aufrufen einer Route, ob die benötigten Bedingungen erfüllt sind und leitet den Nutzer nur dann wirklich weiter. In dieser Anwendung kommt beispielsweise für die Login-Funktionalität ein Guard zum Einsatz.

3.1.1.4 Module

Angular-Module fassen eine inhaltlich sinnvoll vom Rest der Anwendung getrennte Sammlung von Programmelementen wie zum Beispiel Components oder Services zusammen. Services und Guards, welche innerhalb des Moduls mittels dependency

injection erhalten können werden sollen, müssen im entsprechenden Modul angegeben werden. In dieser Anwendung gibt es neben dem Routing-Module (dazu später mehr) nur ein richtiges Module, welches Komponenten und Services bündelt, das App-Module.

3.2 Bausteine

Hier werden in kurzer Form alle von uns erzeugten Bausteine des Front-Ends vorgestellt und erläutert.

3.2.1 AppModule

Die AppModule-Klasse stellt das einzige richtige Modul in unserer Anwendung dar. Hier werden alle Components deklariert, externe Module und damit deren Funktionalität importiert. Außerdem werden hier die für die dependency injection benötigten Services angegeben und damit bereitgestellt.

3.2.2 RoutingModule

Der Sinn dieses Moduls besteht ausschließlich darin, das Routing der Anwendung zu realisieren. Hier werden alle URL-Routen und die jeweiligen Komponenten als deren Gegenstück definiert. Durch das Verwenden von canActivate-Guards wird bei den Routen, die zur Hauptseite oder der Settings-Seite führen verhindert, dass diese ohne einen erfolgreichen Login erreicht werden können. Außerdem werden alle Routern, die nicht explizit von uns definiert wurden, durch die Nutzung einer Wildcard-Route auf die Login-Seite der Anwendung weitergeleitet.

3.2.3 AppComponent

Diese Komponente ist die Root-Komponente der Anwendung. In ihr wird keine Funktionalität implementiert, sondern lediglich der Grundaufbau der Webseite durch HTML- sowie CSS-Code definiert. Im HTML-Teil ist dabei ein „router-outlet“ genanntes Element auffällig. Dabei handelt es sich um einen Platzhalter für die jeweilige Komponente der aktuellen Route (siehe 3.2.2)

3.2.4 HeaderComponent

Die Header-Komponente stellt den Header der Weboberfläche dar und zeichnet sich vor allem dadurch aus, dass verschiedene Buttons je nach aktiver Router angezeigt werden. Neben einem permanenten Refresh-Button als Logo wird nur wenn der Anwender eingeloggt ist ein Logout-Button angezeigt, welcher den Anwender über den in

3.2.15 beschriebenen AuthService ausloggt und anschließend zur Login-Seite weiterleitet. Zudem gibt es je nachdem, ob sich der Nutzer auf der Haupt- oder Settings-Seite der Webanwendung befindet, einen Button der zu der jeweils anderen Seite führt.

3.2.5 LoginComponent

Hier wird der Aufbau der Login-Seite definiert. Zudem wird die Eingabe von Login-Daten und die Abwicklung des Login-Prozesses durch den in 3.2.15 beschriebenen AuthService geregelt. Bei erfolgreichem Login wird der Anwender auf die Hauptseite der Anwendung weitergeleitet und bei einem fehlgeschlagenem Login wird eine Fehlermeldung ausgegeben.

3.2.6 MainPageComponent

In dieser Komponente wird die Hauptseite der Anwendung beschrieben. Dabei geht es vor allem um den groben Aufbau und das Verhalten der Anwendung bei verschiedenen Bildschirmgrößen. Bei einer kleineren Auflösung rutschen die nebeneinander dargestellten Teilbereiche der Anwendung in eine Darstellung, bei der sie untereinander angeordnet werden. Das soll die Nutzung auf Geräten mit einer geringeren Auflösung oder einem anderen Bildformat verbessern. Die eigentlichen Inhalte der Hauptseite selbst, sind hierbei in anderen Komponenten definiert, welche hier lediglich eingebunden werden.

3.2.7 RangeSliderComponent

Die RangeSliderComponent nutzt die Funktionalitäten des HTML-Input-Elementes des Typs Range und fügt dem ein ansprechendes Styling hinzu. Die Implementierung dieser Komponente ist sehr stark auf Wiederverwendbarkeit ausgelegt, da es sich um ein sehr unspezifisches Element handelt, was in komplett anderen Anwendungen ohne nennenswerte Änderungen sinnvoll sein kann. Aus diesem Grund können hier viele Werte zur Anpassung übergeben werden. In der folgenden Tabelle 1 werden alle Input- sowie Output-Parameter der Komponente beschrieben.

Art	Name	Typ	Beschreibung
Input	min	number	Minimalwert des Sliders. Kann größer sein als max. Kann eine Fließkommazahl sein.
Input	max	number	Maximalwert des Sliders. Kann kleiner sein als min. Kann eine Fließkommazahl sein.

Input	value	number	Initialisierungswert des Sliders. Muss innerhalb von min und max liegen. Kann eine Fließkommazahl sein.
Input	step	number	Schrittweite des Sliders. Kann eine Fließkommazahl sein.
Input	color1	string	Hexcode der Hintergrundfarbe des Sliders linkerhalb des Thumb-Elements (aktuelle Auswahl im Slider) in der gängigen Form eines Hexadezimal-Farbcodes (beispielsweise #fff).
Input	color2	string	Hexcode der Hintergrundfarbe des Sliders rechterhalb des Thumb-Elements (aktuelle Auswahl im Slider) in der gängigen Form eines Hexadezimal-Farbcodes (beispielsweise #000).
Output	value-Change	Event-Emitter- <number>	EventEmitter welcher bei Änderung des Slider-Wertes ein Event mit dem neuen Slider-Wert ausstößt. Kann dazu genutzt werden, um ein Data-Binding mittels (change)-Directive zu realisieren.

Tabelle 1: Input- und Output-Variablen der RangeSlider-Komponente

3.2.8 VideoComponent

In dieser Komponente wird sowohl der MJPEG-Livestream als auch die Wiedergabe der aufgezeichneten Video-Clips implementiert. Standardmäßig wird hier nur der Livestream angezeigt. Erst wenn der Anwender über die in beschriebene 3.2.11 EventLog-Component die Wiedergabe einer Aufzeichnung auslöst, wird der Livestream, welcher per HTML-img-Tag angezeigt wird, durch eben diese Clip-Wiedergabe ersetzt, welche per HTML-video-Tag angezeigt wird.

3.2.9 VideoSettingsComponent

Mithilfe dieser Komponente können die Bildeinstellungen des vom Back-End produzierten MJPEG-Streams durch mehrere RangeSlider (siehe 3.2.7) konfiguriert werden. Die neuen Einstellungen werden sobald der Anwender die Position eines Slider verändert hat, den Slider-Thumb also verschoben und losgelassen hat, über den in 3.2.16 beschriebenen SettingsService an das Back-End geschickt.

3.2.10 TitleBarComponent

Diese Komponente wird in der EventLogComponent (siehe 3.2.11) sowie der EmailNotificationComponent (siehe 3.2.12) als Titelzeile verwendet. In ihr gibt es neben der Möglichkeit einen Titel von außerhalb der Komponente einzuschleusen auch eine Möglichkeit, einen booleschen Wert an einen Toggle-Switch zu binden. Dieser Schalter ist dazu gedacht, die Features, welche die beiden Komponenten zur Verfügung stellen, aktivieren beziehungsweise deaktivieren zu können.

3.2.11 EventLogComponent

Die EventLogComponent dient dazu, dem Anwender alle durch das System aufgezeichneten Clips von detektierten Bewegungen aufzulisten und das Starten dieser Aufnahmen per Klick auf das jeweilige Thumbnail zu ermöglichen. Dazu wird ein Event ausgestoßen, welches dazu genutzt wird in der MainPageComponent (siehe 3.2.6) eine Funktion auszulösen, welche wiederum in der VideoComponent (siehe 3.2.8) das eigentliche Abspielen des Videoclips auslöst. Zudem sollen Aufnahmen permanent gelöscht werden können. Die Bewegungserkennung kann zudem über eine in dieser Komponente enthaltene Instanz der TitleBarComponent (siehe 3.2.10) in dieser Komponente deaktiviert beziehungsweise aktiviert werden. Die angezeigte Tabelle mit den Aufzeichnungen ist dabei so implementiert, dass nicht sofort alle Einträge angezeigt werden. Es werden zunächst immer nur bis zu zehn Einträge angezeigt. Erst wenn der Nutzer an das Ende der Tabelle gescrollt hat, werden ihm bis zu zehn weitere Einträge aufgelistet, bis alle Einträge in der Tabelle enthalten sind. Auf diese Weise müssen nicht immer alle Daten von Back-End abgerufen werden, obwohl der Anwender eventuell gar nicht an ihnen interessiert ist. Je nach Einstellungen und Situation könnte das Initialisieren der Liste andernfalls sehr lange dauern, wenn extrem viele Log-Einträge gespeichert sind. Über eine Polling-Funktion werden zudem im Abstand von wenigen Sekunden neue Listeneinträge vom Back-End abgefragt und in die Liste eingetragen.

3.2.12 EmailNotificationComponent

In dieser Komponente werden dem Anwender alle im System registrierten E-Mail-Adressen aufgelistet. Alle registrierten E-Mail-Adressen werden bei der Detektion einer Bewegung über diese informiert. Es besteht hierbei die Möglichkeit, einzelne E-Mails über eine Checkbox bei jedem Eintrag von den Benachrichtigungen auszuschließen. Das Feature der E-Mail-Benachrichtigungen kann zudem über eine in dieser Komponente enthaltene Instanz der TitleBarComponent (siehe 3.2.10) in dieser Komponente deaktiviert beziehungsweise aktiviert werden. Neue E-Mail-Adressen können über ein

Input-Feld eingetragen und gespeichert werden. Bereits eingetragene Adressen können über einen Button bei jedem Eintrag gelöscht werden.

3.2.13 StatusButtonComponent

Diese Komponente stellt einen Button mit Text dar, der zusätzlich je nach Statuswert neben dem Button-Text ein Status-Symbol anzeigt. Dazu wird eine Variable vom Type boolean verwendet. Ist diese Variable nicht initialisiert, so wird eine Ladeanimation angezeigt. Enthält sie jedoch den Wert true, so wird statt der Ladeanimation ein grüner Haken angezeigt. Bei false hingegen wird ein rotes Kreuz angezeigt. Beide Variablen, jene die den Button-Text beinhaltet sowie die andere von Typ boolean, können von außen in die Komponente gereicht werden.

3.2.14 SettingsPageComponent

Hierbei handelt es sich ähnlich wie bei der MainPageComponent (siehe 3.2.6) um eine Komponente für ein Seitenlayout. Die Settings-Seite ist dabei so aufgebaut, dass es für die einzelnen Einstellungen jeweils eine Eingabemöglichkeit sowie einen Status-Button (siehe 3.2.13) gibt. Beim Aufrufen der Komponente werden alle Input-Felder mit den vom Back-End erhaltenen, gespeicherten Einstellungen gefüllt. Alle Status-Buttons zeigen dann einen grünen Haken an. Bis zur Initialisierung hingegen zeigen sie eine Ladeanimation an. Sobald der Anwender die gespeicherten Werte eines Input-Feldes verändert, wird im entsprechenden Status-Button ein rotes Kreuz angezeigt, was dem Anwender signalisiert, dass der dort eingegebene Wert von gespeicherten abweicht. Per Klick auf den Status-Button wird das Speichern des eingegeben Wertes über den in 3.2.16 beschriebenen SettingsService gestartet. Bei einer positiven Rückmeldung, nachdem der gespeicherte Wert mit der neuen Eingabe überschrieben wurde, zeigt der Status-Button wieder den grünen Haken an. Über zusätzlichen Button am unteren Ende der Komponente kann zudem ein Backup-File zum Back-End im zip-Format heruntergeladen werden.

3.2.15 AuthService

Der AuthService stellt die eigentlichen Login-Funktionalitäten zur Verfügung. Durch das Übergeben von LoginCredentials an die authenticate-Methode wird der Login-Prozess mit dem Back-End abgewickelt. Falls die eingegebenen Login-Daten korrekt waren, wird das öffentliche Attribut „isAuthenticated“ von Typ boolean der Klasse auf true gesetzt und signalisiert so, dass der Anwender eingeloggt ist.

3.2.16 SettingsService

In diesem Service können beim Back-End Systemeinstellungen gespeichert werden. Dabei wird ein Objekt vom Typ Settings die Methode „changeSettings“ übergeben. Bei diesem Typ müssen nicht alle Member vorhanden sein, wodurch ein Objekt nur ausschließlich den Attributen übergeben werden kann, welche auch wirklich geändert werden sollen. Zusätzlich werden Methoden zum Abrufen der gespeicherten Einstellungen und zum Herunterladen einer Backup-Datei im zip-Format zur Verfügung gestellt.

3.2.17 EmailService

Der EmailService stellt innerhalb der Anwendung alle E-Mail-Daten-bezogenen Service-Funktionalitäten zur Verfügung. Dazu werden Methoden zum Abfragen aller gespeicherten E-Mails, zum togglen des Notification-Statuses einer bestimmten E-Mail, oder zum Hinzufügen beziehungsweise Löschen von E-Mails angeboten.

3.2.18 EventService

Beim EventService können alle den Event-Log betreffenden Service-Funktionalitäten gefunden werden. Über die Methode „getEventLogEntries“ können durch die Übergabe von Seitenzahl sowie Seitengröße bestimmte Anteile der gespeicherten Event-Logs vom Back-End abgerufen werden. Zusätzlich gibt es eine Methode zum Löschen einzelner Event-Log-Einträge. Eine weitere Funktion dient zum Laden eines Videoclips zu einem solchen Eintrag. Der entsprechende Clip liegt dann als Blob vor.

3.2.19 AuthGuard

Bei der Klasse AuthGuard handelt es sich um eine Implementierung des „canActivate“-Interfaces. Sie wird im RoutingModule (siehe 3.2.2) dazu verwendet, um alle wesentlichen Routen der Anwendung zu sperren, falls der Nutzer nicht korrekt eingeloggt ist. Dazu wird im AuthService (siehe 3.2.15) geprüft, ob das Attribut „isAuthenticated“ den Wert „true“ aufweist. Andernfalls wird der Anwender zu der Route weitergeleitet, welche die LoginComponent darstellt.

3.2.20 Model-Interfaces

In der folgenden Tabelle 2 werden die in der Anwendung verwendeten Model-Interfaces aufgelistet und beschrieben.

3.3 Komponenten-Service-Diagramm

In der folgenden Abbildung 1 wird der Aufbau des Front-Ends im Bezug auf seine Komponenten und deren Nutzung von Services dargestellt.

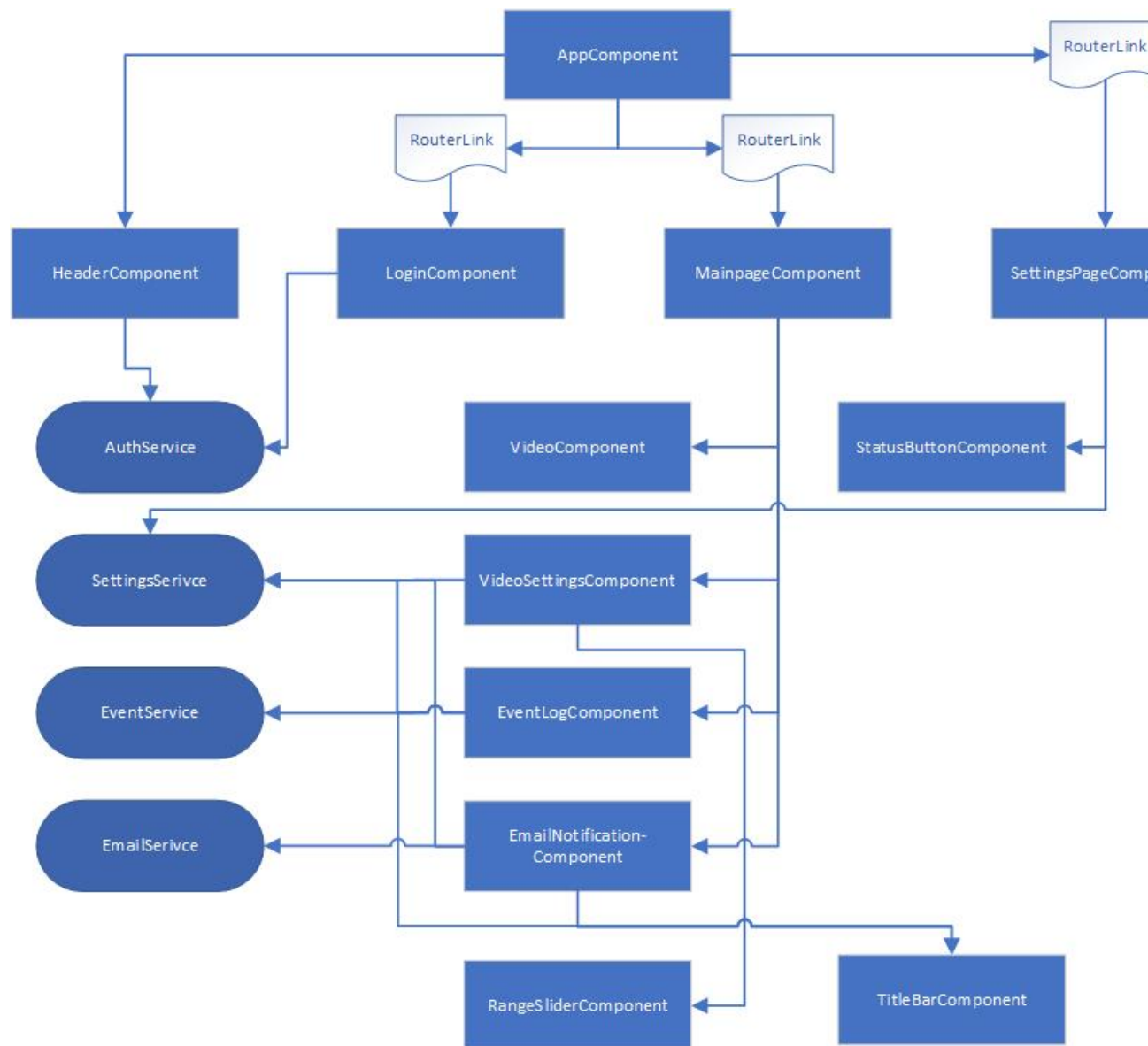


Abbildung 1: Komponenten-Service-Diagramm zum Front-End

Name	Parameter	Beschreibung
EventLogEntry	id	Id des Log-Eintrages. Vom Back-End generiert.
	message	Anzeigenachricht zum Log-Eintrag.
	timestamp	Zeitpunkt der Erstellung des Log-Eintrages.
	thumbnail	Erster Frame des aufgezeichneten Videos als Base64-Image.
	recording	Dateiname des aufgezeichneten Videos. Dient als Id der Videodatei.
LoginCredentials	user	Username zum Einloggen.
	password	Passwort zum Einloggen.
NotificationEmail	id	Id der Benachrichtigungs-E-Mail-Adresse. Vom Back-End generiert.
	address	Die tatsächliche E-Mail-Adresse
	notify	Sagt aus, ob die Benachrichtigung für diese E-Mail-Adresse aktiviert ist.
Settings	streamaddress	URL des Quelllivestreams.
	sensitivity	Sensitivität der Bewegungserkennung. Nimmt am Back-End Werte zwischen 0 und 1 an.
	brightness	Helligkeit des Ergebnislivestreams. Nimmt am Back-End Werte zwischen 0 und 1 an.
	contrast	Kontrast des Ergebnislivestreams. Nimmt am Back-End Werte zwischen 0 und 1 an.
	global_notify	Sagt aus, ob die E-Mail-Benachrichtigung im Allgemeinen aktiviert ist.
	log_enabled	Sagt aus, ob die Bewegungserkennung im Allgemeinen aktiviert ist.
	cliplength	Gibt die maximale Länge einer Aufzeichnung in Sekunden an.
	max_logs	Gibt die maximale Anzahl an Aufzeichnungen an. Bei Überschuss werden die ältesten Aufzeichnungen gelöscht.
	max_storage	Gibt den maximalen für Aufzeichnungen allozierbaren Speicherplatz im Megabyte an.

Tabelle 2: Beschreibung der Model-Interfaces

4 Tests

Diese Testdokumentation wurde erstellt, um die Herangehensweise, Durchführung sowie die Ergebnisse unseres Testprozesses festzuhalten.

4.1 Ziele

Ziel unseres Testprozesses ist es garantieren zu können, dass die in diesem Projekt geschaffene Software unter den von uns festgelegten Voraussetzungen annähernd bis vollständig fehlerfrei und mit möglichst guter Performance betrieben werden kann. Auffälligkeiten sowie nach dem Testprozess bekannte und nicht behobene Fehler sollen am Ende des Testprozesses dokumentiert sein.

4.2 Rahmenbedingungen

Grundsätzlich wurde während der Entwicklung der Anwendung stets darauf geachtet, dass die jeweils neu implementierten Features einwandfrei funktionieren und auch, dass durch die Implementierung jener Features keine der zuvor vorhandenen Teile beschädigt werden. Dennoch haben wir in unserer Projektplanung eine gesonderte Testphase geplant, bei der wir im Zeitraum von zwei Wochen alle nötigen Schritte abschließen möchten, um die von uns erstellte Anwendung ausgiebig zu testen. In dieser zweiwöchigen Testphase soll der Testprozess vollständig abgeschlossen werden. Alle einzelnen Tests von Fron-End sowie Back-End wurden dabei jeweils in der selben Testumgebung durchgeführt.

4.3 Teststrategie

Um unser Testziel zu erreichen greifen wir auf verschiedene Testmethoden zurück. Da die Testphase sowohl durch einen kurzen Zeitraum, als auch die Anzahl der Tester eingeschränkt ist, müssen wir diese Ressourcen bestmöglich nutzen. Nach längeren Diskussionen innerhalb des Entwicklungsteams haben wir uns dazu entschlossen, auf eine Kombination von automatisierten Unit-Tests, manuellen System- und UI-Tests, sowie Last-Test zu setzen. Auf diese Weise decken wir beim Testen nicht nur funktionale sondern auch qualitative Anforderungen der Software ab.

Welche Methodik bei den einzelnen Teilen der Anwendung verwendet wurde wird in der folgenden Tabelle 3 dargestellt.

Testobjekt	Art des Testens
Front-End	Unit-Tests, Manuelle Tests
Back-End	Unit-Tests. Manuelle Tests
Gesamtsystem	Manuelle Tests, Last-Tests

Tabelle 3: Testarten der unterschiedlichen Softwareteile

Trotz dessen, dass wir eine eigene Testphase geplant haben, ist es uns wichtig über den gesamten Entwicklungsprozess der Software für eine stets einwandfrei lauffähige Anwendung zu sorgen. Dies entspricht nicht nur unserem agilen Softwareentwicklungsprozess nach Scrum, sondern erleichtert auch die gemeinsame Arbeit durch mehrere Entwickler jeweils an Front-End sowie Back-End. Um dies gewährleisten zu können, haben wir abseits der Testphase jedes neu implementierte Feature sowie die Auswirkungen der Implementierung auf den Rest der Anwendung manuell getestet.

4.4 Testen des Front-Ends

4.4.1 Unit-Tests

Bei unserem Front-End sind wir zu dem Schluss gekommen, dass ein automatisiertes Testen nur bedingt sinnvoll ist. Ein großer Teil der Implementierungen dort bezieht sich rein auf die Darstellung der vom Back-End erhaltenen Daten im Webbrowser, oder um das Beschaffen und Versenden eben dieser Daten. Ein automatisiertes Testen der Weboberfläche ist dabei überproportional aufwändig und in unserem Falle in den meisten Fällen nicht sinnvoll, da es sich vor allem um statische Inhalte oder um Video- beziehungsweise Bildinhalte handelt. Außerdem muss beim Testen einer Weboberfläche auf Faktoren wie Browserkompatibilität geachtet werden, was durch manuelles Testen besser umsetzbar ist. Nichts desto trotz wurde für jede Komponente des Front-Ends ein eigener Unit-Test erstellt, der die vollständige Erzeugung eben dieser Komponente simuliert und testet. Dabei werden für die Komponente erforderliche Abhängigkeiten durch Mock-Objekte ersetzt, um ein unabhängiges Testen zu ermöglichen.

Standardgemäß verwenden wir beim automatisierten Testen unseres Angular-Front-Ends das Testframework Karma. Dieses ist bereits beim Erzeugen eines neuen Angular-Projektes per Angular-CLI integriert und vorkonfiguriert.

4.4.1.1 Ausführen der Unit-Tests

Nachdem das Projekt korrekt auf die in 2 Dargestellte Art und Weise installiert wurde und lauffähig ist, können die automatisierten Tests durch das Aufrufen eines Konsolenbefehls gestartet werden. Dazu muss im Projektordner ein Terminal geöffnet werden und der Befehl „ng test“ausgeführt werden.

4.4.1.2 Ergebnisse der Unit-Tests



```
17 specs, 0 failures

AppComponent
  should create the app
EmailNotificationComponent
  should create
EventLogComponent
  should create
HeaderComponent
  should create
LoginComponent
  should create
MainpageComponent
  should create
RangeSliderComponent
  should create
SettingspageComponent
  should create
AuthGuard
  should ...
AuthService
  should be created
EmailService
  should be created
EventService
  should be created
SettingsService
  should be created
StatusButtonComponent
  should create
TitleBarComponent
  should create
VideoSettingsComponent
  should create
VideoComponent
  should create
```

Abbildung 2: Ergebnisse der automatisierten Front-End-Tests

4.4.2 Manuelle Tests

Beim manuellen testen handelt es sich um einen Testprozess, bei dem der Tester ohne die Verwendung von Automatisierungstools vorgeht. Dabei können durch die systematische Verwendung der Software und das Nutzen von Diagnosetools oft Fehler aufgedeckt werden, die etwa bei Unit-Tests häufig nicht gefunden werden. Insbesondere Benutzeroberflächen können auf diese Weise unkompliziert getestet werden.

Im folgenden wird tabellarisch festgehalten, welche Aktionen getestet wurden und von welcher Ausgangssituation aus getestet wurde. Alle Tests wurden in den beiden Browsern Google Chrome (64-Bit Version 71.0.3578.98 Offizieller Build) und Mozilla Firefox (64-Bit Version 63.0.1 Offizieller Build) auf einem mit Windows 10 betriebenen Laptop mit einer Auflösung von 1920x1080 durchgeführt.

4.4.2.1 Ergebnisse der Manuellen Tests

Vorraussetzung für alle Tests ist selbsterklärend, dass Front-End sowie Back-End korrekt installiert und gestartet sind. Zudem sind alle Einstellungen sinnvoll gewählt. Das bedeutet beispielsweise, dass ein funktionierender MJPEG-Stream hinterlegt ist. Es sind außerdem fünf beliebige Clips mit allen nötigen Werten korrekt gespeichert sowie abrufbar. Es sind auch zwei gespeichert.

Folgende Einstellungen waren bei den folgenden manuellen Tests vorhanden:

Einstellung	Wert
sensitivity	0.0
brightness	0.5
contrast	1.0
global_notify	true
log_enabled	true
streamaddress	https://webcam1.lpl.org/axis-cgi/mjpg/video.cgi
cliplength	10
max_logs	20
max_storage	1024

Tabelle 4: Eingestellte Werte vor jedem manuellen Test

Erklärung zur nachfolgenden Tabelle 5 der manuellen Front-End-Tests:

Bedeutung der Spalte C*: Der Test wurde in der zuvor genannten Version von Google Chrome erfolgreich durchgeführt.

Bedeutung der Spalte F*: Der Test wurde in der zuvor genannten Version von Mozilla

Firefox erfolgreich durchgeführt.

#	Komponente	Vorraussetzungen	Aktion	Erwartetes Ergebnis	C*	F*
1	Login	Der Anwender befindet sich auf der Login-Seite und ist demnach nicht eingeloggt.	Der Anwender gibt beim Einloggen den richtigen Usernamen (user) und das richtige Passwort (geheim) ein.	Der Anwender wird auf die Hauptseite der Anwendung weitergeleitet und ist korrekt eingeloggt.	X	X
2	Login	Der Anwender befindet sich auf der Login-Seite und ist demnach nicht eingeloggt.	Der Anwender gibt beim Einloggen einen falschen Usernamen (Verwendet: test) und das richtige Passwort (geheim) ein.	Rechts neben dem Login-Button erscheint eine Nachricht (Login failed) in roter Schrift.	X	X
3	Login	Der Anwender befindet sich auf der Login-Seite und ist demnach nicht eingeloggt. Das Back-End ist nicht erreichbar.	Der Anwender versucht sich einzuloggen.	Rechts neben dem Login-Button erscheint eine Nachricht (Login failed) in roter Schrift.	X	X
4	Login	Der Anwender befindet sich auf der Login-Seite und ist demnach nicht eingeloggt.	Der Anwender gibt beim Einloggen den richtigen Usernamen (user) und ein falsches Passwort ein. (Verwendet: test)	Rechts neben dem Login-Button erscheint eine Nachricht (Login failed) in roter Schrift.	X	X

5	Login	Der Anwender befindet sich auf der Login-Seite und ist demnach nicht eingeloggt.	Der Anwender gibt beim Einloggen sowohl einen falschen Usernamen (Verwendet: test1) als auch ein falsches Passwort (Verwendet: test2) ein.	Rechts neben dem Login-Button erscheint eine Nachricht (Login failed) in roter Schrift.	X	X
6	Login	Der Anwender befindet sich auf der Login-Seite und ist demnach nicht eingeloggt.	Der Anwender gibt beim Einloggen einen falschen Usernamen (Verwendet: test) und das richtige Passwort (geheim) ein.	Zwischen dem Absenden der Logindaten und dem Empfangen einer Antwort durch das Back-End wird rechts neben dem Login-Button eine Ladeanimation angezeigt.	X	X
7	Login	Der Anwender befindet sich auf der Login-Seite und ist demnach nicht eingeloggt.	Der Anwender gibt beim Einloggen den richtigen Usernamen (user) und das richtige Passwort (geheim) ein.	Zwischen dem Absenden der Logindaten und dem Empfangen einer Antwort durch das Back-End wird rechts neben dem Login-Button eine Ladeanimation angezeigt.	X	X
8	Header	Der Anwender befindet sich auf der Login-Seite und ist demnach nicht eingeloggt.	Der Anwender klickt auf das PIPCO-Logo auf der linken Seite des Headers.	Die Webseite wird neu geladen.	X	X

9	Header	Der Anwender befindet sich auf der Settings-Seite und ist demnach bereits eingeloggt.	Der Anwender klickt auf das PIPCO-Logo auf der linken Seite des Headers.	Die Webseite wird neu geladen. Der Anwender ist nicht länger eingeloggt und wird daher auf die Login-Seite weitergeleitet.	X	X
10	Header		Der Anwender hovort mit dem Cursor über das PIPCO-Logo auf der linken Seite des Headers.	Ein Tooltip (Refresh Page) wird neben dem Cursor angezeigt. Der Cursor ändert sein Styling zu Pointer.	X	X
11	Header	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite		Auf der rechten Seite des headers befinden sich ein Settings-Button sowie ein Logout-Button (in dieser Reihenfolge)	X	X
12	Header	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender hovort mit dem Cursor über den Settings-Button auf der rechten Seite des Headers.	Ein Tooltip (Settings) wird neben dem Cursor angezeigt. Der Cursor ändert sein Styling zu Pointer.	X	X
13	Header	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender hovort mit dem Cursor über den Logout-Button auf der rechten Seite des Headers.	Ein Tooltip (Logout) wird neben dem Cursor angezeigt. Der Cursor ändert sein Styling zu Pointer.	X	X
14	Header	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender klickt auf den Settings-Button auf der rechten Seite des Headers.	Der Anwender wird auf die Settings-Seite weitergeleitet.	X	X

15	Header	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender klickt auf den Logout-Button auf der rechten Seite des Headers.	Der Anwender wird korrekt ausgeloggt und auf die Login-Seite weitergeleitet.	X	X
16	Header	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.		Auf der rechten Seite des Headers befinden sich ein Home-Button sowie ein Logout-Button (in dieser Reihenfolge)	X	X
17	Header	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.	Der Anwender klickt auf den Home-Button auf der rechten Seite des Headers.	Der Anwender wird auf die Hauptseite weitergeleitet	X	X
18	Header	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.	Der Anwender klickt auf den Logout-Button auf der rechten Seite des Headers.	Settings-Button sowie Logout-Button im Header sind nicht mehr da.	X	X
19	Header	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.	Der Anwender hovers mit dem Cursor über den Home-Button auf der rechten Seite des Headers.	Ein Tooltip (Home) wird neben dem Cursor angezeigt. Der Cursor ändert sein Styling zu Pointer.	X	X
20	Video	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Der in den Einstellungen hinterlegte MJPEG-Stream wird angezeigt.	X	X
21	Video	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Über dem MJPEG-Stream wird eine Überschrift dargestellt (Currently Watching: IP Camera Live Stream)	X	X

22	Video	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Aus der Event-Log-Komponente wird das Thumbnail einer Aufnahme angeklickt.	Der MJPEG-Stream wird durch eine Video-Wiedergabe des ausgewählten Clips ersetzt.	X	X
23	Video	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Aus der Event-Log-Komponente wird das Thumbnail einer Aufnahme angeklickt.	Der Titel über der Clip-Wiedergabe ändert sich (Currently Watching: Motion Detection Clip)	X	X
24	Video	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Aus der Event-Log-Komponente wird das Thumbnail einer Aufnahme angeklickt.	Neben dem Titel über der Clip-Wiedergabe erscheint rechts ein Button (RETURN TO LIVESTREAM)	X	X
25	Video	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite. Über die Event-Log-Komponente wurde die Wiedergabe eines Clips gestartet.	Es wird auf den Return-Button (RETURN TO LIVESTREAM) rechts oben in der Komponente geklickt.	Die Clip-Wiedergabe wird durch den MJPEG-Stream ersetzt.	X	X
26	Video	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite. Über die Event-Log-Komponente wurde die Wiedergabe eines Clips gestartet.	Es wird auf den Return-Button (RETURN TO LIVESTREAM) rechts oben in der Komponente geklickt.	Die Überschrift über der Wiedergabe wird zurückgesetzt (Currently Watching: IP Camera Live Stream)	X	X

27	Video	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite. Über die Event-Log-Komponente wurde die Wiedergabe eines Clips gestartet.	Es wird auf den Return-Button (RETURN TO LIVESTREAM) rechts oben in der Komponente geklickt.	Der eben betätigte Return-Button verschwindet.	X	X
28	Range-Slider	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Der Range-Slider für die Einstellung Contrast weist die richtige Hintergrundfarbe auf (#431ede)	X	X
29	Range-Slider	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Der Griff des Range-Sliders für die Einstellung Contrast weist die richtige Hintergrundfarbe auf (#c7c7c7)	X	X
30	Range-Slider	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender hovers mit dem Cursor über den Griff des Range-Sliders für die Einstellung Contrast	Der Cursor verändert sein Styling zu grab	X	X
31	Range-Slider	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender greift den Griff des Range-Sliders für die Einstellung Brightness und zieht diesen komplett nach rechts.	Der Griff des Range-Sliders für die Einstellung Brightness bewegt sich in 5 Sprüngen an den rechten Rand des Range-Sliders.	X	X
32	Range-Slider	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender greift den Griff des Range-Sliders für die Einstellung Brightness und zieht diesen komplett nach rechts.	Der Hintergrund des Range-Sliders für die Einstellung Brightness bleibt stets bis hinter seinen Griff mit der richtigen Farbe ausgefüllt.	X	X

33	Video-Settings	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Der Regler für die Einstellung Sensitivity befindet sich auf seiner Minimalposition (0.0)	X	X
34	Video-Settings	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Der Regler für die Einstellung Brightness befindet sich genau auf der mittleren Position (0.5)	X	X
35	Video-Settings	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Der Regler für die Einstellung Contrast befindet sich auf seiner Maximalposition (1.0)	X	X
36	Title-Bar	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Die Titel-Leiste der Event-Log-Komponente zeigt den richtigen Titel an (Motion Detection)	X	X
37	Title-Bar	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Der Toggle-Switch der Event-Log-Komponente weißt den richtigen Wert auf (true).	X	X
38	Title-Bar	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite. Der Toggle-Switch der Event-Log-Komponente weißt den richtigen Wert auf (true).	Der Anwender klickt auf den Toggle-Switch der Event-Log-Komponente.	Der Toggle-Switch der Event-Log-Komponente ändert seinen Wert (false).	X	X

39	Title-Bar	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender hovers mit dem Cursor über den Toggle-Switch der Event-Log-Komponente	Der Cursor verändert sein Styling zu pointer.	X	X
40	Title-Bar	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender hovers mit dem Cursor über abseits des Toggle-Switches der Event-Log-Komponente über die Titel-Leiste derselben Komponente.	Neben dem Cursor wird ein Tool-Tip mit dem Titel der Komponente angezeigt (Motion Detection).	X	X
41	Event-Log	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		In der Event-Log-Komponente werden fünf Einträge dargestellt.	X	X
42	Event-Log	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Der Slider auf der rechten Seite ist ausgegraut, da er noch nicht benötigt wird.	X	X
43	Event-Log	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Jeder ungerade Eintrag in der Event-Log-Komponente hat eine graue Hintergrundfarbe. Jeder gerade Eintrag hat eine weiße.	X	X
44	Event-Log	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Bei jedem Eintrag in der Event-Log-Komponente wird in der Spalte Thumbnail das richtige Thumbnail zum Clip angezeigt.	X	X

45	Event-Log	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Bei jedem Eintrag in der Event-Log-Komponente wird in der Spalte Timestamp der richtige Zeitpunkt der Aufnahme angezeigt.	X	X
46	Event-Log	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Bei jedem Eintrag in der Event-Log-Komponente wird in der Spalte Message nichts angezeigt.	X	X
47	Event-Log	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Bei jedem Eintrag in der Event-Log-Komponente wird in der Spalte Delete ein Delete-Button als rotes Kreuz angezeigt.	X	X
48	Event-Log	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender hovers mit dem Cursor über den Delete-Button eines angezeigten Eintrages in der Event-Log-Komponente	Der Cursor verändert sein Styling zu pointer	X	X
49	Event-Log	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender hovers mit dem Cursor über das Thumbnail eines angezeigten Eintrages in der Event-Log-Komponente	Das Thumbnail vergrößert sich etwas.	X	X
50	Event-Log	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender hovers mit dem Cursor über das Thumbnail eines angezeigten Eintrages in der Event-Log-Komponente	Der Cursor verändert sein Styling zu pointer.	X	X

51	Event-Log	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite. Der Cursor befindet sich über einem Thumbnail eines angezeigten Eintrages der Event-Log-Komponente.	Der Anwender bewegt den Cursor vom Thumbnail weg.	Das Thumbnail verkleinert sich auf seine ursprüngliche Größe.	X	X
52	Event-Log	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite. Es sind zwanzig Clips verfügbar.		In der Event-Log-Komponente werden zehn Einträge angezeigt.	X	X
53	Event-Log	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite. Es sind zwanzig Clips verfügbar.		Der Slider auf der rechten Seite der Event-Log-Komponente ist nicht ausgegraut. Er kann benutzt werden um durch die Einträge zu scrollen.	X	X
54	Event-Log	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite. Es sind zwanzig Clips verfügbar.	Der Anwender Scrollt über das Ende der angezeigten Einträge in der Event-Log-Komponente hinaus.	Es werden weitere zehn Einträge nachgeladen und angezeigt.	X	X

55	Event-Log	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite. Es sind zwanzig Clips verfügbar. Es wurde bis zum Ende gescrollt, wodurch alle zwanzig Einträge aufgelistet werden.	Der Anwender Scrollt über das Ende der angezeigten Einträge in der Event-Log-Komponente hinaus.	Es passiert nichts.	X	X
56	Event-Log	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender betätigt den Delete-Button des ersten Eintrags in der Event-Log-Komponente.	Der Eintrag wird aus der Tabelle gelöscht.	X	X
57	Event-Log	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender Klickt auf das Thumbnail des ersten Eintrages in der Event-Log-Komponente.	Die Video-Komponente schaltet zur Wiedergabe des entsprechenden Videoclips um.	X	X
58	Event-Log	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite. Es sind zwanzig Clips verfügbar.		Alle fünf Sekunden werden von Back-End die neuesten Event-Logs abgerufen.	X	X
59	Event-Log	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite. Es sind zwanzig Clips verfügbar.	Das Back-End liefert einen neuen, einundzwanzigsten Event-Log Eintrag zurück.	Der älteste Eintrag wird aus der Liste aller Einträge gelöscht und der neue Eintrag wird am anderen Ende eingefügt.	X	X
60	Email-Notification	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		In der Email-Notification-Komponente werden zwei E-Mail-Einträge angezeigt.	X	X

61	Email-Notification	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Der Slider auf der rechten Seite ist ausgegraut, da er noch nicht benötigt wird.	X	X
62	Email-Notification	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Jeder ungerade Eintrag in der Email-Notification-Komponente hat eine graue Hintergrundfarbe. Jeder gerade Eintrag hat eine weiße.	X	X
63	Email-Notification	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Im Input-Feld der Email-Notification-Komponente wird ein Platzhaltertext (Add a new E-Mail address) angezeigt.	X	X
64	Email-Notification	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Links vom Input-Feld der Email-Notification-Komponente befindet sich ein runder Submit-Button mit einem Plus in der Mitte.	X	X
65	Email-Notification	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender schreibt etwas in das Input-Feld der Email-Notification-Komponente.	Der Platzhaltertext des Input-Feldes wird durch die Eingabe ersetzt.	X	X

66	Email-Notification	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender schreibt eine neue E-Mail-Adresse (test1@test2.com) in das Input-Feld der Email-Notification-Komponente und klickt auf den Add-Button.	Es wird ein neuer Eintrag in der Tabelle angezeigt. Die angezeigte E-Mail-Adresse stimmt mit der Eingabe überein und die Notification-Checkbox zeigt true an.	X	X
67	Email-Notification	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender schreibt eine bereits vorhandene E-Mail-Adresse in das Input-Feld der Email-Notification-Komponente und klickt auf den Add-Button.	Es passiert nichts.	X	X
68	Email-Notification	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender schreibt eine ungültige E-Mail-Adresse (iaintanemail) in das Input-Feld der Email-Notification-Komponente und klickt auf den Add-Button.	Es passiert nichts.	X	X
69	Email-Notification	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite. Es sind zehn Email-Einträge verfügbar.		Der Slider auf der rechten Seite der Email-Notification-Komponente ist nicht ausgegraut. Er kann benutzt werden um durch die Einträge zu scrollen.	X	X

70	Status-Button	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.		Der Save-Button der neben dem Input-Feld der Einstellungsmöglichkeit für die Maximale Cliplänge enthält den richtigen Text (SAVE)	X	X
71	Status-Button	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.		Der Save-Button der neben dem Input-Feld der Einstellungsmöglichkeit für die maximale Cliplänge zeigt einen grünen Haken an.	X	X
72	Status-Button	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.	Der Anwender verändert den Inhalt des Input-Feldes der Einstellungsmöglichkeit für die maximale Cliplänge (30).	Der Save-Button der neben dem Input-Feld der Einstellungsmöglichkeit für die Maximale Cliplänge zeigt ein rotes Kreuz an.	X	X
73	Status-Button	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite. Der Inhalt des Input-Feldes der Einstellungsmöglichkeit für die maximale Cliplänge wurde verändert (30).	Der Anwender betätigt den Save-Button neben diesem Input-Feld.	Das rote Kreuz innerhalb des Save-Buttons wird durch eine Ladeanimation ersetzt. Nachdem der neue Wert erfolgreich gespeichert wurde, wird diese wiederum durch einen grünen Haken ersetzt.	X	X
74	Settings-Page	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.		Das Input-Feld der Einstellungsmöglichkeit streamaddress enthält den richtigen Wert (siehe Tabelle 4)	X	X

75	Settings-Page	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.		Das Input-Feld der Einstellungsmöglichkeit <code>cliplength</code> enthält den richtigen Wert (siehe Tabelle 4)	X	X
76	Settings-Page	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.		Das Input-Feld der Einstellungsmöglichkeit <code>max_logs</code> enthält den richtigen Wert (siehe Tabelle 4)	X	X
77	Settings-Page	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.		Das Input-Feld der Einstellungsmöglichkeit <code>max_storage</code> enthält den richtigen Wert (siehe Tabelle 4)	X	X
78	Settings-Page	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.	Der Anwender klickt auf den Download-Backup-Button.	Bis die Backup-Datei vom Back-End erhalten wird, erscheint neben dem Download-Backup-Button eine Ladeanimation.	X	X
79	Settings-Page	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.	Der Anwender klickt auf den Download-Backup-Button.	Sobald die Backup-Datei vom Back-End erhalten wurde wird diese automatisch durch den Browser heruntergeladen.	X	
80	Main-Page	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Am oberen Bildschirmrand wird die Header-Komponente vollständig angezeigt.	X	X

81	Main-Page	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Unterhalb der Header-Komponente wird auf der linken Seite erst die Video-Komponente und darunter die Video-Settings-Komponente korrekt angezeigt.	X	X
82	Main-Page	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.		Unterhalb der Header-Komponente wird auf der rechten Seite erst die Event-Log-Komponente und darunter die Email-Notification-Komponente korrekt angezeigt.	X	X
83	Main-Page	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender verringert die Breite der Browserfensters auf weniger als 1200px.	Die beiden rechten Komponenten rutschen unter die beiden linken Komponenten. Alle vier dieser Komponenten nehmen ab sofort die ganze breite der Browserfensters in Anspruch.	X	X

Tabelle 5: Manuelle Front-End-Tests

4.5 Testen des Gesamtsystems

5 Ausblick

6 Fazit

Literaturverzeichnis

[OWA19] *Offizielle Website zu Angular*. <https://angular.io/>. Version: Januar 2019

Eidesstattliche Erklärung

Ich versichere, dass ich die vorstehende Arbeit selbständig verfasst und hierzu keine anderen als die angegebenen Hilfsmittel verwendet habe. Alle Stellen der Arbeit die wörtlich oder sinngemäß aus fremden Quellen entnommen wurden, sind als solche kenntlich gemacht.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt oder an anderer Stelle veröffentlicht.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben kann.

Furtwangen, den 13.01.2019