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

Abbildungsverzeichnis

Tabellenverzeichnis

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. Rahmenbedinungen

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. Planung

2.1. Organisation

2.1.1. Vorgehensweise und Kommunikation

Zur Planung des Projekts haben wir uns von der agilen Softwareentwicklung und dem Scrum Prinzip inspirieren lassen. Ähnlich wie in Scrum haben wir Regelmäßige Treffen mit sogenannten Sprints verbunden. In den Treffen werden anhand des Projektplans oder Backlogs Aufgaben besprochen und für den Zeitraum bis zum nächsten Treffen entsprechend aufgeteilt. Die Treffen fanden wöchentlich Montags statt. Die Treffen wurden von einem sogenannten 'Srcum-Master' kuratiert, dieser achtet darauf, das die Besprechungen sachlich bleiben, dass jedes Teammitglied einen Statusbericht abgibt und dass anstehende Probleme und Aufgaben zugewiesen werden. Die Zeit zwischen den Meetings werden als Sprint bezeichnet und dienen dazu die zugewiesenen Aufgaben zu erledigen. Im reinen Scrum Modell gibt es zusätzlich noch Tägliche Meetings, die aber im Rahmen des Semesterprojektis nicht realisierbar waren. Die Kommunikation während des Sprints erfolgte Hauptsächlich über den Instant-Messenger WhatsApp. Bei Bedarf gab es auch noch weitere spontan organisierte Treffen um diverse Probleme zu besprechen und sich gegenseitig zu helfen.

2.1.2. Rollenverteilung

Zu allen Aufgabenbereichen des Projekts wurden Rollen erstellt und jeweils zwei Teammitglieder zugeordnet, ein Hauptverantwortlicher und ein Stellvertreter. Die so verteilten Rollen sollten somit einen Verantwortlichen für jeden Aspekt des Projekts bestimmen, damit bei Problemen oder nachfragen immer eine Ansprechperson gefunden werden kann.

Rolle	Beschreibung	Qualifikationen	Verantwortlichkeit	Person
Projektleiter	Operative Planung und Steuerung des Projekts	Beherrschen der Projektmanagementinstrumente / Führen der Gruppe	Erreichen der Sach- und Terminziele	1. Kevin Hertfelder 2. Florian Hauger

Dokumentationsbeauftragter	Dokumentation verwalten / Zusammentragen / Protokolle schreiben	Latexerfahren	Vollständigkeit und Termineinhaltung der Dokumentationsabgabe	1. Daniel Petrusic 2. Florian Hauger
Koordinator	Koordiniert Meetings und sonstige Termine		Termineinhaltung der Meetings	1. Patrick Kroner 2. Kevin Hertfelder
Architekt / Integrator	Erstellt Architektur, definiert Schnittstellen	Kenntnisse über Software Entwurfsmuster	Einhaltung der Architektur	1. Florian Hauger 2. Kevin Hertfelder
Testverantwortlicher	Prüft Testabdeckung	Testerfahren	Korrekte Tests	1. Daniel Petrusic 2. Patrick Kroner
OpenCV Verantwortlicher	Implementiert und plant Open CV Funktionalität	Gute Einarbeitung in OpenCV / Erfahrung mit Python	OpenCV Funktionalität	1. Patrick Kroner 2. Florian Hauger
Weboberflächen Verantwortlicher	Implementiert und plant Weboberfläche	Erfahrung mit Angular und Webtechnologien	Funktionsumfang und Funktionalität der Weboberfläche	1. Daniel Petrusic 2. Kevin Hertfelder
Softwarekommunikation und Schnittstellenbeauftragter	Sorgt für Kommunikation der Softwaremodule		Kommunikation zwischen Kamera und Python / Kommunikation zwischen Weboberfläche und Python	1. Kevin Hertfelder 2. Florian Hauger
Systemadmin	Kümmert sich um Ressourcen / Richtet notwendige Dienste ein		Ressourcen für Entwicklung vorhanden	1. Florian Hauger 2. Patrick Kroner

Tabelle 1.: Rollenverteilung

2.1.3. Tools und Technologien

Die Zentrale Bibliothek für das Projekt ist OpenCV, diese ist Open Source und ermöglicht es digitale Bildbearbeitung und Analyse einfach und effizient durchzuführen, in unserem Fall war dies notwendig um die Bewegung in einem Videostream zu erkennen.
OpenCV gibt es für mehrere Sprachen, die am besten unterstützten Sprachen sind jedoch C++ und Python. C++ hätte einen Prerformance Vorteil gebracht, hätte aber etwas komplizierteren Syntax verlangt. Auf Rat des Professors hin, haben wir uns schlussendlich für die Python Variante entschieden.
Für die Weboberfläche, haben wir das ebenfalls mit der Open Source Lizenz vertriebene Framework Angular ausgewählt, da dieses modernen Frontentwicklung mit einfacher Datenbindung erlaubt.
Die Sprache der Wahl für die Entwicklung eises Angular Clienten ist Typescript, welche ein Javascript Dialekt ist.
Für die Revisionssicherheit des Projekts wurden drei Git Projekte angelegt. Die drei Projekte teilten Das Gesamtprojekt in Frontend, Backend und Dokumentation auf. Diese klare Trennung erleichterte paralleles arbeiten am Projekt. Als Hostingprovider für diese Repositories wurde Github gewählt.
Die vewendete Entwicklungsumgebung wurde allen Teammitgliedern frei gestellt und es wurde dann für die Entwicklung am Backend für Python PyCharm von IntelliJ verwendet. Das Typescript Frontend wurde mit Microsoft Visual Studio Code entwickelt. für die Planung der REST-Schnittstelle haben wir Swagger verwendet.
Für Den Projektplan haben wir Microsoft Projekt verwendet, da wir dieses schon im Modul Projektmanagement kennen gelernt haben.
Für die Erstellung der Teilpräsentationen wurde Microsoft Powerpoint verwendet.
Schaubilder und Diagramme haben wir mit Microsoft Visio erstellt.
Die Plakate für die Abschlussmesse haben wir mit den Grafikprogramm Gimp entworfen.

2.2. Spezifikation

2.2.1. Anforderungsanalyse

Die Anforderungen für das Projekt haben wir nach dem Kick-Off Meeting mit dem Professor in einem Kollektiven Brainstorming gesammelt. Da bei diesem Projekt kein klassischer Kunde vorhanden ist, der die grundlegenden Anforderungen vorgibt, waren die Anforderungen recht flexibel und wir haben uns an einigen Referenzprodukten und deren Funktionsumfang orientiert.

2.2.2. Anforderungsdetails

Bewegungserkennung
Ein Videostream soll programmatisch auf Bewegungen überprüft werden. Die Überprüfung soll kleine Änderungen wie durch Wind bewegte Blätter und Kamerarauschen ignorieren und größere Bewegungen erkennen.

Log
Erkannte Bewegungen sollen in einem Log festgehalten und archiviert werden. Dabei soll der Zeitpunkt der Aufnahme, die Aufnahme selbst und ein dazugehörendes Thumbnail gespeichert werden. Der Log Wird automatisch erzeugt und kann durch manuelles löschen von Einträgen manipuliert werden.

Aufnahmefunktion
Wenn eine Bewegung erkannt wurde soll das System in der Lage sein den Videostream in einer persistenten Datei zu speichern. Die gespeicherten Aufnahmen müssen jederzeit abrufbar sein. In der Aufnahme sollen erkannte Bewegungen visuell hervorgehoben werden.

E-Mailbenachrichtigungen
Bei Einer erkannten Bewegung soll Das System in der Lage sein Automatisiert Eine E-Mailbenachrichtigung zu versenden.

lokales Verfahren
Die aufgezeichneten Daten sollen das lokale Netzwerk nicht verlassen. Alle Verfahrensschritte benutzen ausschließlich lokal ausgeführte Software.

Videofeed
Der live Videostream soll direkt angezeigt werden können.

Backup
Es soll die Möglichkeit die im Log gesammelten Daten als Datei zu exportieren.

Einstellungsmöglichkeiten
In der Weboberfläche sollen einige Benutzer spezifische Einstellungen für das Programm getroffen werden können. Diese umfassen:

- Einstellen der Empfindlichkeit der Bewegungserkennung
- Bildkorrektureigenschaften wie Kontrast und Helligkeit
- Einstellen der Streamquelle
- Limit für die maximale Länge einer zusammenhängenden Aufnahme bei konstanter Bewegung
- Limits für das Backup (maximale Anzahl Dateien / Dateigröße)
- Login Daten ändern

- festlegen der E-Mailbenachrichtigungs Empfänger.
- Ein-/ausschalten des Logs
- Ein-/ausschalten der E-Mailbenachrichtigungen

Weboberfläche
Es soll eine Weboberfläche als Benutzerschnittelle geschaffen werden die alle Funktionen des Systems einem Anwender exponiert. Die Funktionen der Weboberfläche sind wie Folgt definiert:

- Der zugriff soll durch einen login mit Benutzernamen und Passwort geschützte geschützt werden werden.
- Der Livefeed der Kamera soll prominent auf der dargestellt werden.
- Der Log soll mit allen oben definierten Eigenschaften präsentiert werden.
- Aufnahmen sollen angesehen werden können
- Alle oben definierten Einstellungen sollen über die Weboberfläche manipuliert werden können.
- Möglichkeit ein Backup zu erstellen und herunterzuladen.

Es gab noch weitere Anforderungen die, wie sich später herausstellte, nicht realisierbar waren. Eine Gesichtserkennung, die wegen der geringen Qualität und Größe der Gesichter in der Aufnahme technisch nicht umsetzbar war. Zuletzt eine Portierung auf einen Raspberry Pi, die wegen des hohen Leistungsanspruches der Software nicht umsetzbar war. Die Portierung des Produktivsystems erfolgt stattdessen auf einen lokalen Server.

2.2.3. User Stories

- Als Benutzer möchte ich mich mit einem Benutzernamen und Password in die Weboberfläche einloggen, um sicherzustellen, dass nur autorisierte Benutzer Zugriff auf die Weboberfläche haben.
- Als Benutzer möchte ich, dass ich Einstellung für die Software über die Weboberfläche ändern kann um das Softwareverhalten dynamisch zu beeinflussen.
- Als Benutzer möchte ich meine Emailadresse zum Verteiler für Benachrichtigungen hinzufügen und wieder entfernen können, um so einfach wie möglich auf erkannte Bewegungen aufmerksam gemacht zu werden.

- Als Benutzer möchte ich, Aufnahmen über die Weboberfläche einsehen können, um zu entscheiden, ob die erkannte Bewegung von einem unbefugten Zutritt herrührt.
- Als Benutzer möchte ich den Überwachungs-Videostream über eine Weboberfläche einsehen können, um das Blickfeld der Kamera zu sehen.
- Als Benutzer möchte ich, eine Übersicht über alle erfassten Ereignisse der Kamera mit Zeitstempel einsehen, um nach einem Event in einem gewissen Zeitraum zu suchen.
- Als Benutzer möchte ich, dass meine Daten nicht auf den Servern von Drittanbietern gespeichert werden, um meine Privatsphäre zu schützen.
- Als Benutzer möchte ich, die gesammelten Aufnahmedaten exportieren, um sie anderweitig zu verwenden.

2.2.4. Projektplanung und Meilensteine

Die Termine der Meilensteine wurden an den Zweiwöchentlichen Meetings mit den Betreuern ausgerichtet. Einzige Ausnahme war die Implementierungsphase, die für die Basisimplementierung vier Wochen beansprucht hat. Somit ergeben sich Meilensteine für die, unter anderem im klassischen Wasserfallmodell, vorgegebenen Phasen Spezifikation, Entwurf, Implementierung und Test. Ein letzter, zusätzlicher Meilenstein für die abschließende Präsentation und Abgabe. Die Implementierungsphase war jedoch in drei feingranulare Meilensteine aufgeteilt. Es wurden zwei Meilensteine für die Basisfunktionalität von Front- beziehungsweise Backend und die Featureimplementierung aufgestellt. Arbeit an den Meilensteinen von Front- und Backend konnten parallel bearbeitet werden. Zur Gewichtung der Projektphasen war uns bewusst, dass im Idealfall mehr Zeit für die Spezifikation und den Entwurf nötig gewesen wäre, jedoch war in der begrenzten Zeit die dem Semesterprojekt zur Verfügung stand die Implementierung im Vordergrund und nimmt in unserem Fall ungefähr die Hälfte der Gesamtzeit in Anspruch.

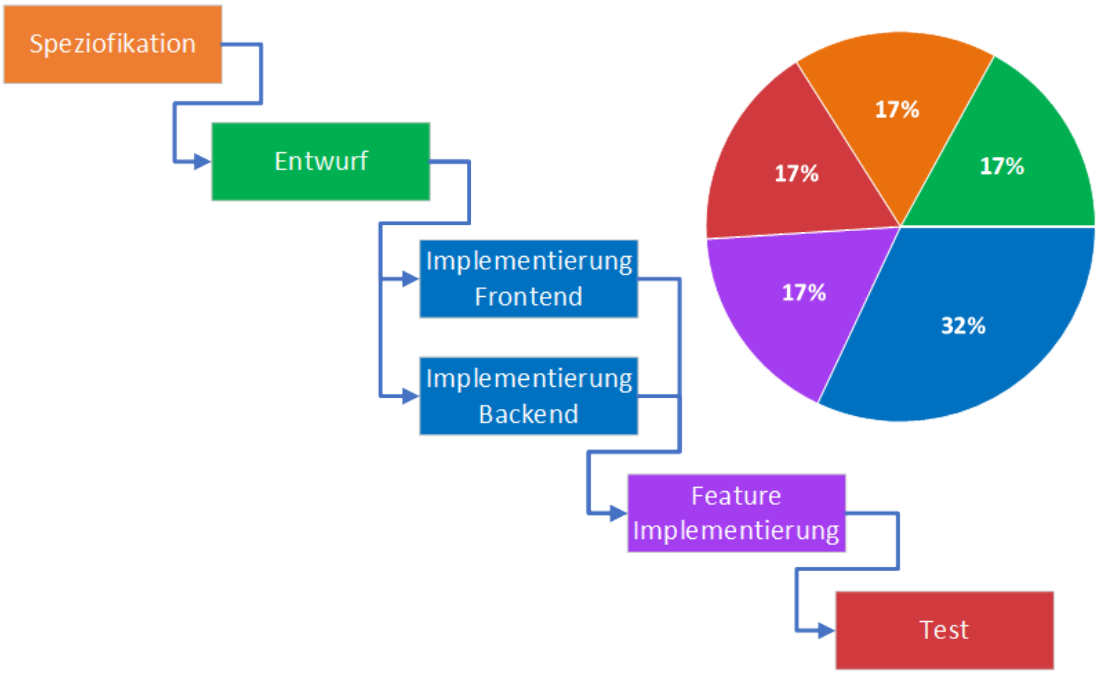

Abbildung 1.: Meilensteindiagramm

Die Meilensteine konnten im Projektverlauf ausnahmslos eingehalten werden und er gab keine Verschiebungen.

2.3. Entwurf

2.3.1. Systemarchitektur

Bei dem ersten Treffen mit dem 'Kunden' wurde von uns eine Software zur Überwachung eines Raumes mithilfe einer herkömmlichen IP-Kamera gewünscht. Da hierfür auch eine Möglichkeit zur Interaktion mit dem System gegeben sein sollte, entschieden wir uns für eine Webseite, mit welcher der Videofeed sichtbar und Einstellungen vorgenommen werden sollten. Da die Software im Gegensatz zu anderen kommerziellen Produkten im lokalen Netz bleiben sollte, spielte die Sicherheit keine große Rolle. Die Architektur sollte die Aufteilung der Schichten wie Logik, Datenhaltung sowie die Präsentation durch den Webserver darstellen. Hierfür wurde zuerst Abbildung 2 erstellt.

Da der Architekt im Gebiet der Webtechnologien kaum Erfahrung hatte, entschieden die verantwortlichen Entwickler sich selbst für Angular und sprachen sich bei der Entwicklung ab. Lediglich REST wurde für die Kommunikation zwischen Frontend und Backend definiert. Um die Aufteilung der Klassen und Kommunikation innerhalb des Backends darzustellen wurde ein vereinfachtes Klassendiagramm erstellt (Abbildung 3).
Die Klasse für die Bildverarbeitung sowie die Klasse für die REST Schnittstelle besit-

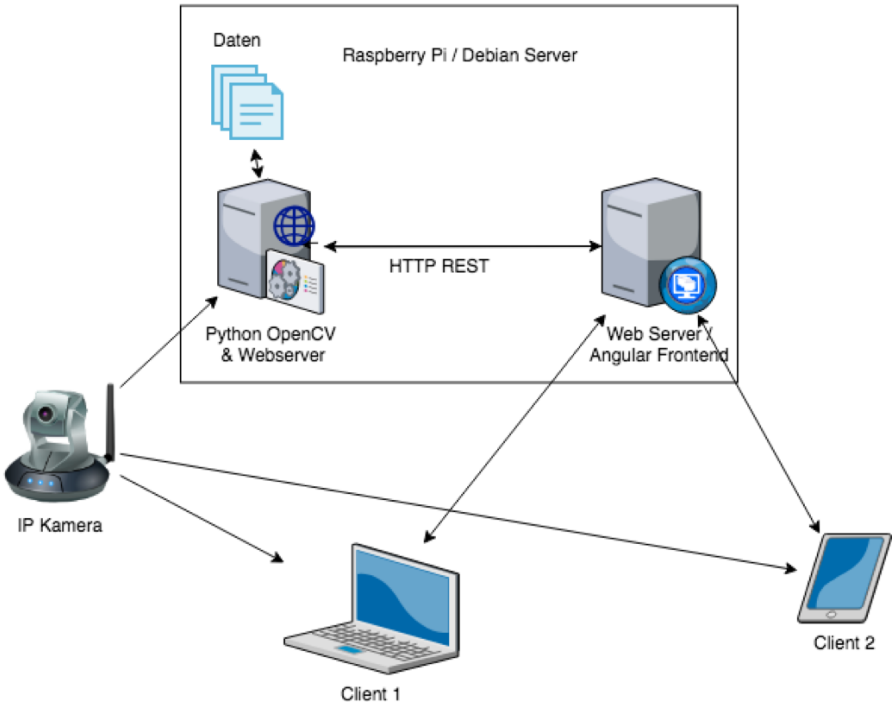

Abbildung 2.: Erste Planung der Architektur

zen eine run-Methode und laufen in einem eigenen Thread. Diese Methoden werden innerhalb der Mainklasse ausgeführt. Die Daten sind innerhalb der ImageProcessing-Klasse und des Webservers verfügbar. Es gibt außerdem jeweils eine weitere Klasse für das Versenden der Email-Benachrichtigung sowie für die permanente Ablage der Daten.

Während der Entwicklung stellte sich heraus, dass der Aufwand für die Implementierung der bidirektionalen Kommunikation zwischen Frontend und Backend zu hoch ist, weshalb wir uns dazu entschieden die Daten direkt vom Backend zu holen und die Logs mithilfe von Polling aktuell zu halten. Eine weitere Änderung war die Kamera, welche lediglich vom Backend angefragt wird. Der Client sollte sich ausschließlich die Bilder, mit eingezeichneten Bewegungen holen. Hierdurch änderte sich die Architektur wie in Abbildung 4 zu sehen.

Um den Ablauf der Implementierten Version besser nachvollziehen zu können wurde ein Sequenzdiagramm erstellt (Abbildung 5).

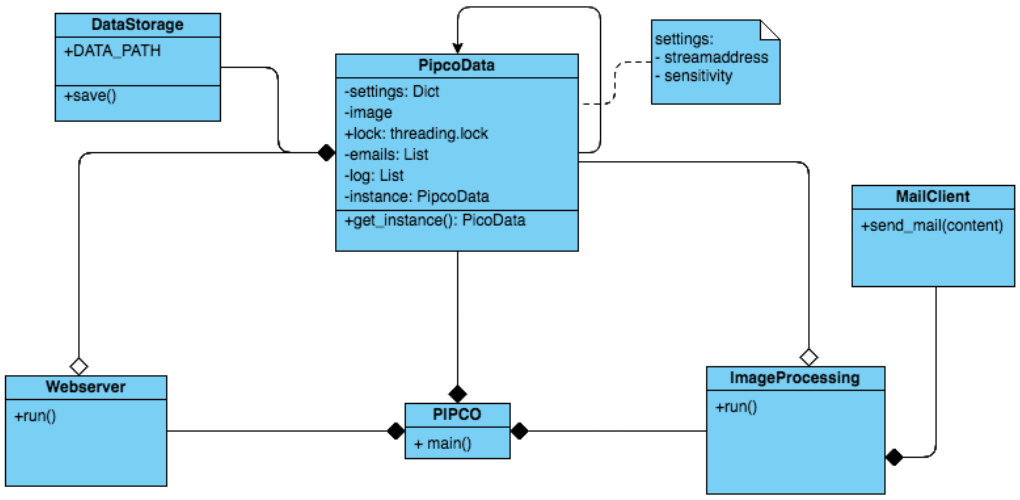

Abbildung 3.: Klassendiagramm: Aufbau des Backends

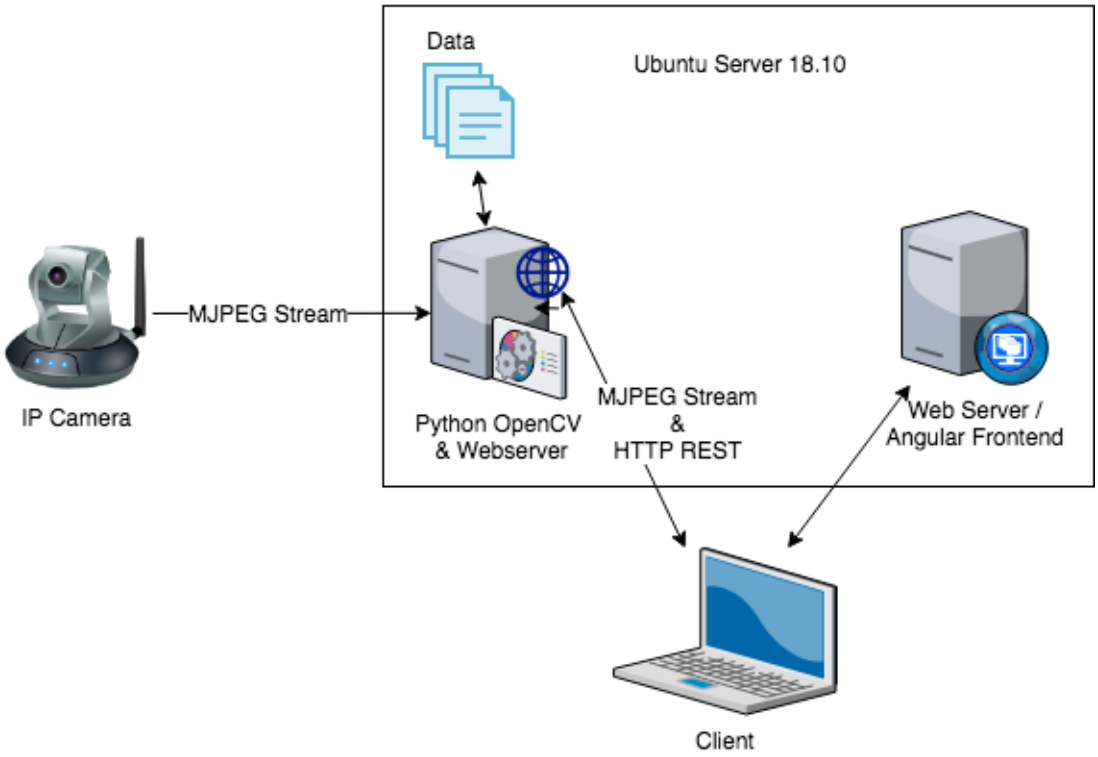

Abbildung 4.: Architektur: Direkter Zugriff auf das Backend

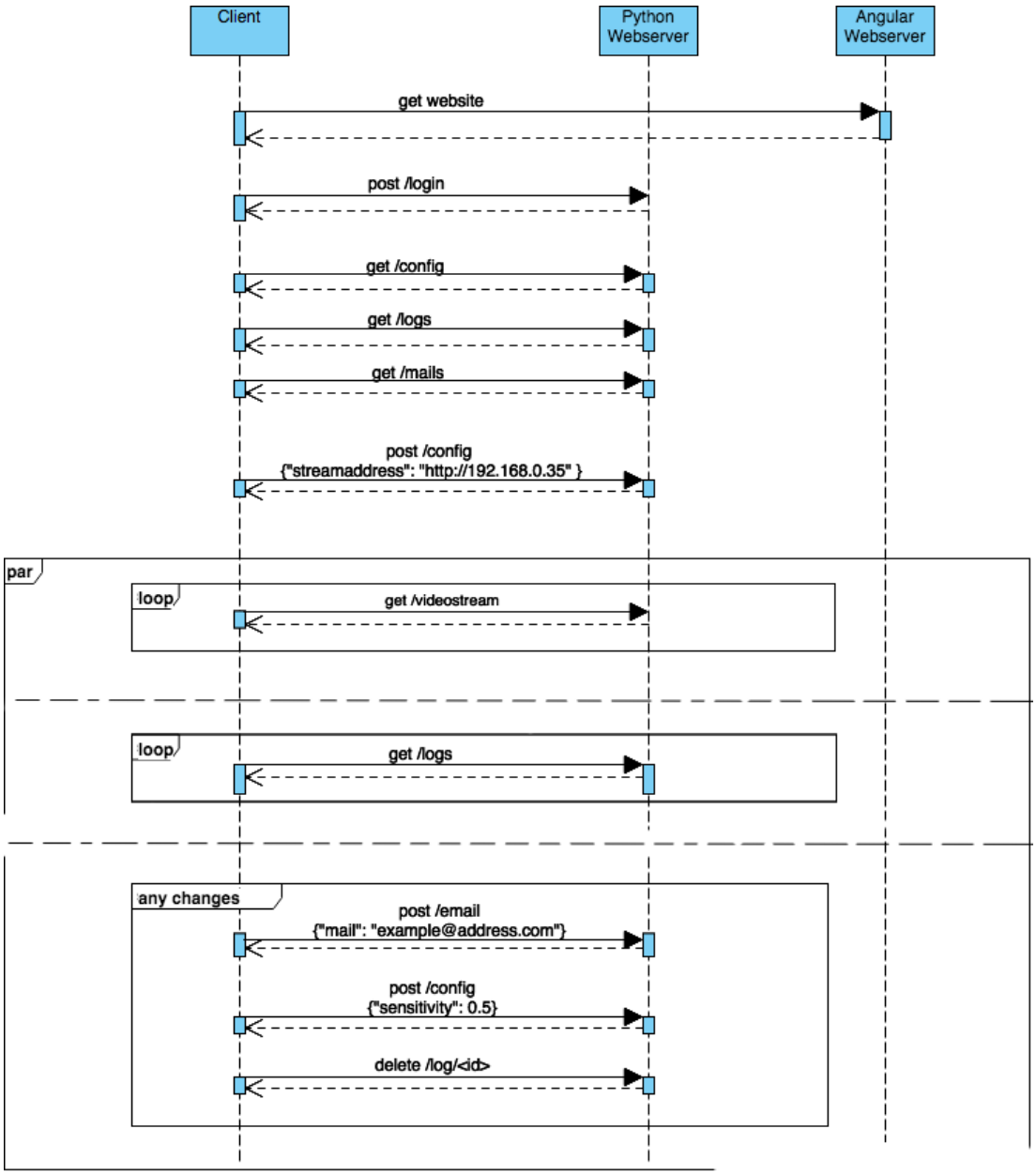

Abbildung 5.: Ablauf Login und Änderungen

Tabelle 2.: Kommunikation Backend und Frontend

Type	address	keys	returns
POST	/login	user, password	OK / Error 403
GET	/videostream		mjpeg stream
GET	/logs/<page_no>/<batch_size>		list with logs
DELETE	/log/<log_id>		id / Error 403
POST	/mail		id / Error 403
GET	/mails		list of mail addresses
DELETE	/mail/<mail_id>		id / Error 403
PUT	/mail/<mail_id>		notify status / Error 403
POST	/config	sensitivity(float), streamaddress,brightness (float), contrast (float), log_enabled(bool), global_notify(bool), cliplength (int in seconds), max_logs (int),max_storage (int in MB)	changed values / Error 403
GET	config		config
GET	/recording/<filename>		videofile
GET	/backup		backup.zip

2.3.2. Rest Schnittstelle

Für die Unabhängige Entwicklung der Kommunikation von Frontend und Backend wurde eine Liste mit den notwendigen REST-Nachrichten erstellt (Tabelle 2).

3. Front-End

Da zwei der vier Projektteilnehmer bereits im Praxissemester Erfahrungen damit gesammelt haben, viel 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. [?]

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. Am besten sollte jedoch die Einführung auf der offiziellen Seite zu Angular unter https://angular.io/ bearbeitet werden.

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 3 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 3.: 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 boolschen 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, das 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 mir 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 4 werden die in der Anwendung verwendeten Model-Interfaces aufgelistet und beschrieben.

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 Benachrichtungs-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 allokierbaren Speicherplatz im Megabyte an.

Tabelle 4.: Beschreibung der Model-Interfaces

3.3. Komponenten-Service-Diagramm

In der folgenden Abbildung 6 wird der Aufbau des Front-Ends im Bezug auf seine Komponenten und deren Nutzung von Services dargestellt.

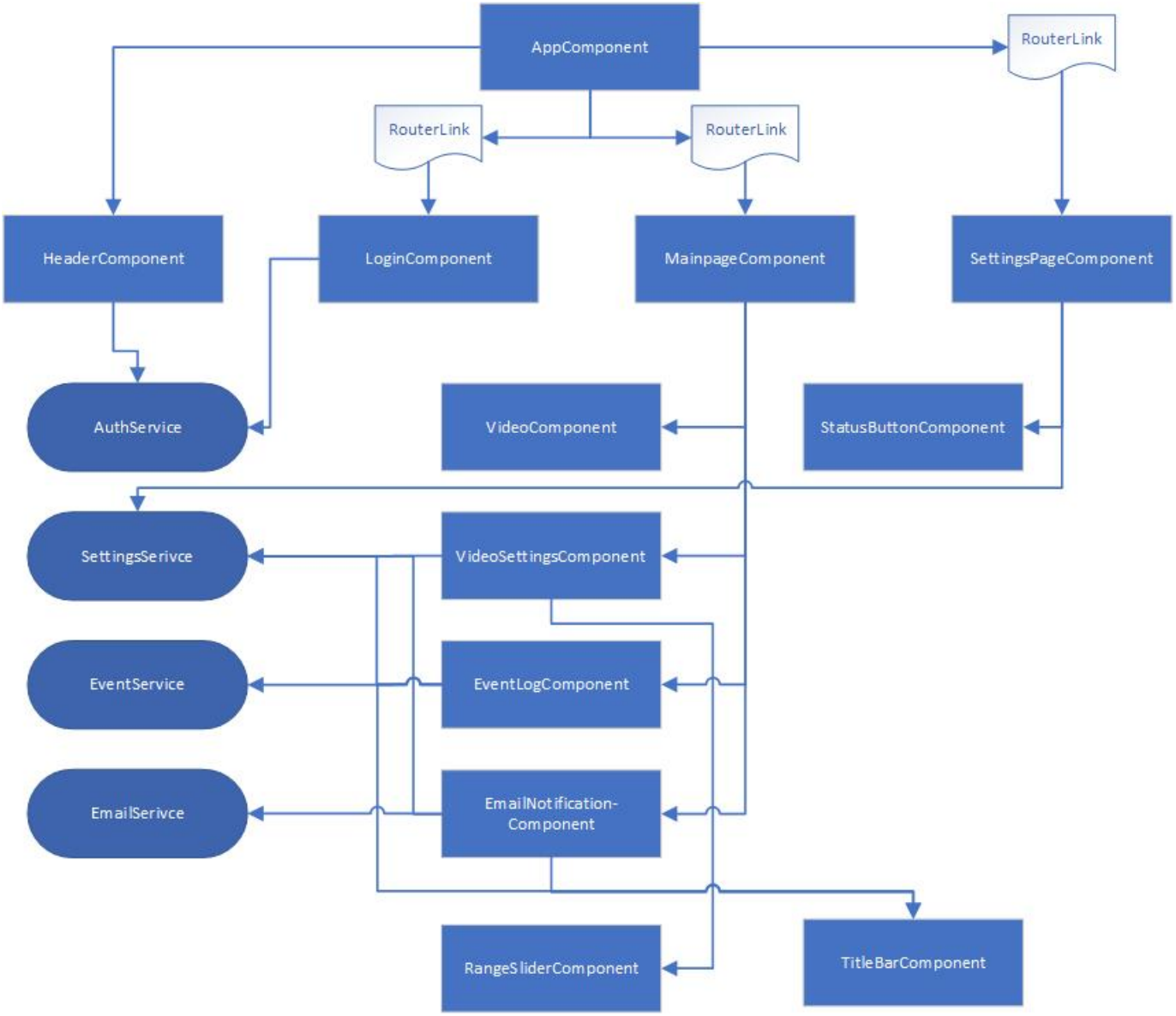

Abbildung 6.: Komponenten-Service-Diagramm zum Front-End

4. Back-End

Für die Entwicklung des Backends stand C++ und Python im Raum. Auf Empfehlung unseres Betreuers hin und aufgrund der einfachen Implementierung einiger Beispiele entschieden wir uns für die Entwicklung in Python.

4.1. Konzept

4.2. Verwendete Open-Source Bibliotheken

4.2.1. OpenCV

Bei OpenCV (Open Source Computer Vision Library) handelt es sich um eine Bibliothek zur Verarbeitung und Bearbeitung von Bildern und Video Formaten. OpenCV wurde in C/C++ entwickelt und unterliegt derzeit einer Lizenz die eine kostenlose Nutzung für akademische als auch kommerzielle ermöglicht. Die Bibliothek unterstützt nicht nur herkömmliche Betriebssysteme wie Windows, Linux und Max OS, sondern auch Mobile Betriebssysteme wie iOS und Android. OpenCV wurde so entwickelt, dass es effizient in Echtzeitsystemen genutzt werden kann. In unserem Projekt werden wir für die Bearbeitung des Video-Streams mehrere Funktionalitäten der Bibliothek aufgreifen und verwenden.

4.2.2. Numpy

Bei NumPy handelt es sich um eine Python exklusive Bibliothek. Diese wurde entwickelt, um mathematische Funktionen anzuwenden. Der Schwerpunkt hierbei liegt ins besonders auf Matrizen. Auch Numpy unterliegt der gleichen Lizenz wie OpenCV und erlaubt daher eine kostenlose Nutzung. In unserem System benötigen wir Numpy um ein paar Operationen auf unsere Bilder anzuwenden oder auch um exemplarisch Bilder zu erstellen.

4.3. Bildverarbeitungsprozess

Um eine Bewegung aus mehreren Bildern erkennen zu können, reicht es normalerweise aus, die jeweiligen Pixelpositionen voneinander zu subtrahieren. Hat man an diesem Punkt noch weiße bzw. farbige Pixel, kann man davon ausgehen, dass es einen Unterschied gibt und daher auch eine Bewegung vorhanden ist. Da wir nun

die Bewegungserkennungen einer Kamera umsetzen sollen, kann man nicht so simple vorgehen. Es gibt mehrere kleinere Aspekte die Beachtet werden können und es gibt auch keine einfache Lösung für einige Probleme. Im Laufe dieses Kapitels werden einige Probleme beleuchtet als auch Lösungen für diese aufgeführt, sofern diese umsetzbar sind.

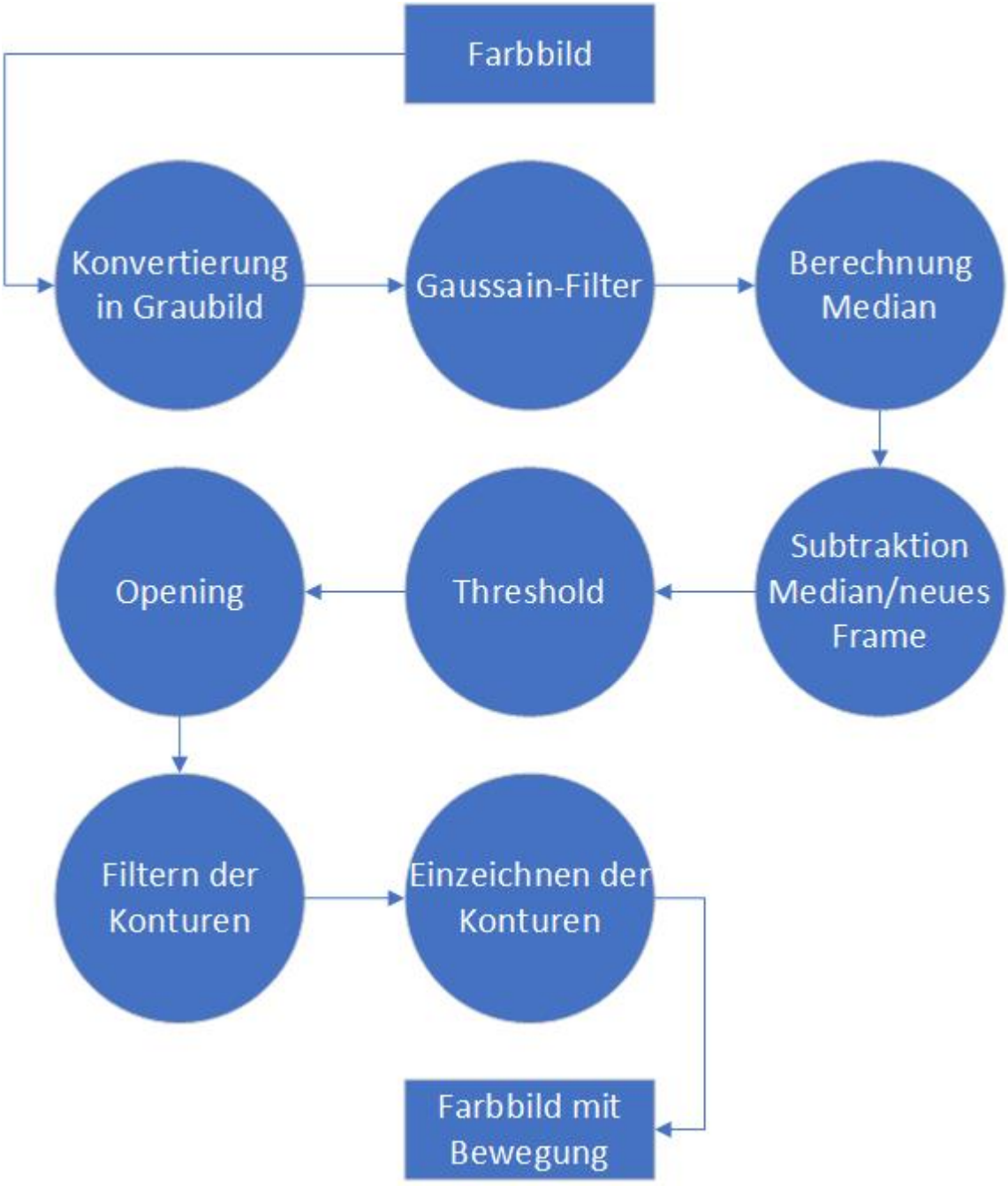

Abbildung 7.: Ablaufdiagramm der Bildverarbeitung

In dem Ablaufdiagramm kann man die Vorgehensweise unserer Bildverarbeitung sehen. Der erste Schritt beinhaltet die Konvertierung eines Farbwertbilds in ein Grauwertbild. Diese kann mit Hilfe von OpenCV leicht umgesetzt werden. Somit muss man

sich keine Gedanken um den Datentyp des Bildes machen. Der Sinn hinter dieser Konvertierung bezieht sich stark auf die Performanz der kommenden Operationen, die zur Bewegungserkennung notwendig sind. Würden wir alle Operationen mit einem Farbwertbild durchführen, hätten wir mindestens eine dreimal längere Laufzeit für die Bildverarbeitung. Als nächsten Schritt entfernen wir das Rauschen aus dem Grauwertbild. Bei Rauschen in einem Bild handelt es sich um Pixelfehler die Werte enthalten, die dem eigentlichen Farbschema widersprechen. Diese entstehen meist direkt durch das Aufnahmegerät. Mithilfe des in OpenCV gebotenen Gaussain-Filters können diese bereinigt werden. Hierbei werden die umliegenden Pixel angeschaut und anhand denen entschieden, welchen Wert er erhalten wird. Sprich liegt der Pixel in einer dunklen grauen Fläche wird auch der Pixelwert den entsprechenden Grauwert erhalten. Da wir nun das neue Bild vorbereitet haben, benötigen wir noch das vorherige Bild für den Vergleich. In unserem Fall berechnen wir Pixelweise den Median aus den letzten 15 Bildern. Durch dieses Vorgehen sollen kleinere Bewegungen von der Bewegungserkennung ignoriert werden. Für das Speichern der Bilder nutzen wir eine zyklische Liste, die das älteste Objekt beim überschreiten der maximalen Anzahl entfernt. Für die Berechnung des Medians bietet OpenCV leider keine Funktion. Daher greifen wir auf die Bibliothek NumPy zurück. Diese ermöglicht uns sowohl die Addition als auch das Berechnen des Medians. Gibt es keinen Fehler bei der Berechnung wird uns das berechnete Bild zurückgegeben, ansonsten erhalten wir das zuletzt hinzugefügte Bild zurück. Haben wir nun das Durschnitts-Bild und das letzte Bild, können wir diese voneinander subtrahieren. Somit haben wir die Unterschiede aus den letzten Bildern. Die Subtraktion können wir direkt mit OpenCV ausführen. Als Ergebnis der Subtraktion erhalten wir ein Bild mit unterschiedlichen Grauwerten. Wenn Grauwerte voneinander abgezogen werden, kann es sein, dass ähnliche Grauwerte trotzdem einen Grauwert größer als „0“ zurückgeben. Für uns spielen aber größere Änderungen eine Rolle. Um die niedrigen Grauwerte zu filtern, kann ein Threshold angewandt werden. Durch die Threshold-Funktion von OpenCV ist es uns möglich die Grauwerte mit einer Grenze in Schwarz und Weiß aufzuteilen. Sprich alle Grauwerte bis Beispielsweise dem Wert von 30 werden zu schwarz und alle höheren Grauwerte werden weiß. Da wir kleine Bewegungen nicht als Bewegung zählen lassen wollen, müssen wir kleine Bereiche vorher entfernen. Durch die sogenannte Opening-Operation können kleine Bereiche gefiltert werden. Hierbei werden zuerst alle Bereiche erodiert und die übriggebliebenen Bereiche werden dilatiert. Dadurch bleiben die größeren Bereiche bestehen und kleine Bereiche werden entfernt. OpenCV bietet für die Erosion und Dilatation separate Funktionen an, die sich leicht verwenden lassen. Als letzten Schritt Extrahieren wir, falls vorhanden, übrig gebliebene Konturen. Gibt es eine Kontur, gab es auch eine Bewegung, und die Pixelkoordinaten der Konturen werden in das Farbwertbild übernommen, um die Bewegung kenntlich zu machen und gegeben falls werden Empfänger

per Email benachrichtigt.

4.4. Performanceprobleme durch GIL

Bei der Berechnung des Medians könnten sinnvoll mehrere Kerne benutzt werden. Dies müsste in Python jedoch über Umwege von Hand gemacht werden, da Multithreading durch den Global Interpreter Lock auf einen Kern beschränkt ist. Bei einem Versuch, bei dem für die Berechnung die Bilder in vier Teile aufgeteilt und eigene Prozesse erstellt wurden, erhöhte sich die Berechnungszeit sogar. Aufgrund des engen Zeitplans konnte dem Versuch der Performanceverbesserung nicht weiter nachgegangen werden.

4.5. Zusätzliche Features

4.5.1. Video, Log und Thumbnail

Es gibt mehrere Aspekte die wichtig für die Bildverarbeitung sind. Da wir über das Backend die Bilder für das Frontend zu Verfügung stellen und auch das Video erstellen, muss einiges in dieser Hinsicht beachtet werden. Zu Beginn haben wir sofort Video, Log und Thumbnail bei der ersten Bewegungserkennung zu Verfügung gestellt. Dies hatte zur Folge, dass beim Aktivieren des Videos die Aufnahme nicht fertiggestellt wird. Und kein Video angezeigt wird. Um dies zu umgehen, werden Video, Log und Thumbnail nur beim Abschluss der Bewegungssequenz weitergeleitet und das Thumbnail kann ein Bild mitten in der Bewegung darstellen.

4.5.2. FPS-Berechnung

OpenCV bietet eine Schnittstelle zum Erstellen von Videos. Hierbei muss vorab eine feste FPS (Frames Per Second) angegeben werden. Während den Tests ist uns aufgefallen, dass das Backend je nach System unterschiedlich schnell arbeitet. Zusätzlich dazu ist die Geschwindigkeit des Videos durch eine falsche Bearbeitungsrate verzerrt. Um dies zu vermeiden, berechnen wir die durchschnittliche FPS zur Laufzeit. Die ersten hundert benötigten Zeiten werden in einer Liste abgespeichert und am Schluss dividiert. Anhand dieses Wertes erhalten wir einen guten Wert, der auch das Video in einer angemessenen Geschwindigkeit darstellt.

4.5.3. Maximale Cliplänge

Um eine maximale Cliplänge gewährleisten zu können muss die Bearbeitungsrate aufgegriffen werden. Wurden genug Bilder gesammelt (FPS x maximale Cliplänge in Sekunden) wird das Video abgebrochen und eine neue Aufnahme wird gegebenenfalls gestartet.

4.5.4. Wait for Motion-end

Wurde gerade eine Bewegung erkannt und die Bewegungen haben aufgehört, wartet die Bearbeitung noch einen Moment ab, ob nicht doch noch eine Bewegung auftaucht. Dadurch wird die Aufnahme nicht direkt beendet, wenn keine Bewegung mehr erkannt wird und ein eventuell kurzer Aussetzer der Bewegung wird ignoriert.

4.6. Datenhaltung und Speicherung

Für die Datenhaltung wurde ein Singleton-Objekt mit den verschiedenen zu teilenden Daten erstellt. Um die Integrität der Daten zu gewährleisten musste darauf geachtet werden, dass keine Zugriffe gleichzeitig auf die jeweiligen Daten gemacht werden können. Hierfür wurden Locks für die jeweiligen Datenobjekte erstellt.
Bei der Erstellung eines Backups dürfen ebenfalls keine Änderungen vorgenommen werden, weshalb alle Locks gesperrt werden müssen.
Die Daten müssen für die Anfrage passend zurückgegeben werden. Damit die Daten nach dem Beenden des Programms noch vorhanden sind, sollten diese gespeichert werden. Da die Daten für die Kommunikation in JSON serialisiert werden müssen, wird auch für das Speichern ein JSONEncoder verwendet. Da die zu speichernden Attribute der Objekte bereits serialisierbar sind, musste kein eigener Encoder geschrieben werden. Für die Decodierung musste jedoch eine Funktion geschrieben werden, welche die Art der Objekte anhand eines Keys prüft und aus den JSON-Daten wieder Objekte erstellt. Es gibt drei JSON-Dateien mit denen die Emails, Logs sowie Einstellungen gespeichert werden.

4.7. Mailclient

Ein wesentlicher Teil eines Überwachungssystems ist die Benachrichtigung wie z.B. per Email. Hierfür haben wir mit Hilfe der umfrangreichen Standardbibliothek in Python einen SMTP-Client erstellt. Der Mailclient soll zum einen den Benutzer benachrichtigen, wenn eine Bewegung erkannt wurde und zum anderen wenn der maximal zu nutzende Speicher belegt ist. Beim Start des Backends werden Login und Passwort abgefragt. Für Testzwecke wurde der Anbieter auf unseren verwendeten Provider vorbelegt.

4.8. Webserver

Für die Kommunikation mit dem Frontend sollte eine REST-Schnittstelle definiert werden. Hierzu wurde zuerst mit einem in den Standardbibliotheken verfügbaren HttpServer experimentiert, welcher jedoch zu minimalisisch und dadurch für die An-

wendung ungeeignet war. Anschließend wurde ein Flask-Webserver verwendet, bei welchem recht einfach über das Web Server Gateway Interface die Ressourcen definiert, und auf diese Ressourcen bestimmte Funktionen registriert werden können. Am einfachsten ginge das mit Annotationen, wenn der Webserver als eigenes Programm läuft. Dies war in unserem Szenario jedoch nicht gegeben, weshalb der Webserver als Klasse definiert und die Pfade innerhalb des Konstruktors hinzugefügt wurden.
Der Webserver holt sich durch Anfragen die Daten vom Singleton-Objekt und sendet regelmäßig das aktuelle JPEG für den Videostream. Allerdings gilt zu beachten, dass der von uns verwendete Webserver nicht für Produktivsysteme gedacht ist.

4.9. Videoquelle

4.9.1. Voraussetzung

Als Videoquelle bieten die meisten IP-Kameras die Bilder als MJPEG oder RTSP-Stream an.
Beide Formate werden durch OpenCV und somit durch unsere Anwendung unterstützt. Ebenso kann OpenCV auch Videodateien unterschiedlicher Formate wie z.B. mp4 oder webm abspielen. Von einer Youtube-Url kann jedoch nicht gestreamt werden, da Youtube nicht das File direkt zur Verfügung stellt.
Der Stream sollte mindestens 15 FPS zur Verfügung stellen, da wir somit auch den Median über den Zeitraum von etwa. einer Sekunde bilden. Wenn der Stream weniger Bilder sendet, wird der Median über einen längeren Zeitraum gebildet und somit ändert sich das Verhalten der Bilderkennung.

4.9.2. Auflösung

Aufgrund der o.g. Performanceprobleme ist die Verwendung einer höheren Auflösung ein großes Problem.
Für die Verwendung einer einfachen Bewegungserkennung reicht die Auflösung von 640x368 aus, jedoch erreichen wir selbst hierbei je nach Rechner nur ca. 20 FPS.

5. Tests

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

5.1. Ziele

Ziel unseres Testprozesses ist es garantieren zu können, dass die in diesem Projekt geschaffene Software unter den von uns festgelegten Vorraussetzungen 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.

5.2. 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.

5.3. 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 5 dargestellt.

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

Tabelle 5.: 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.

5.4. Testen des Front-Ends

5.4.1. 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 automatiertes 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.

5.4.1.1. Ausführen der Unit-Tests

Nachdem das Projekt korrekt auf die in 6 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.

5.4.1.2. Ergebnisse der Unit-Tests

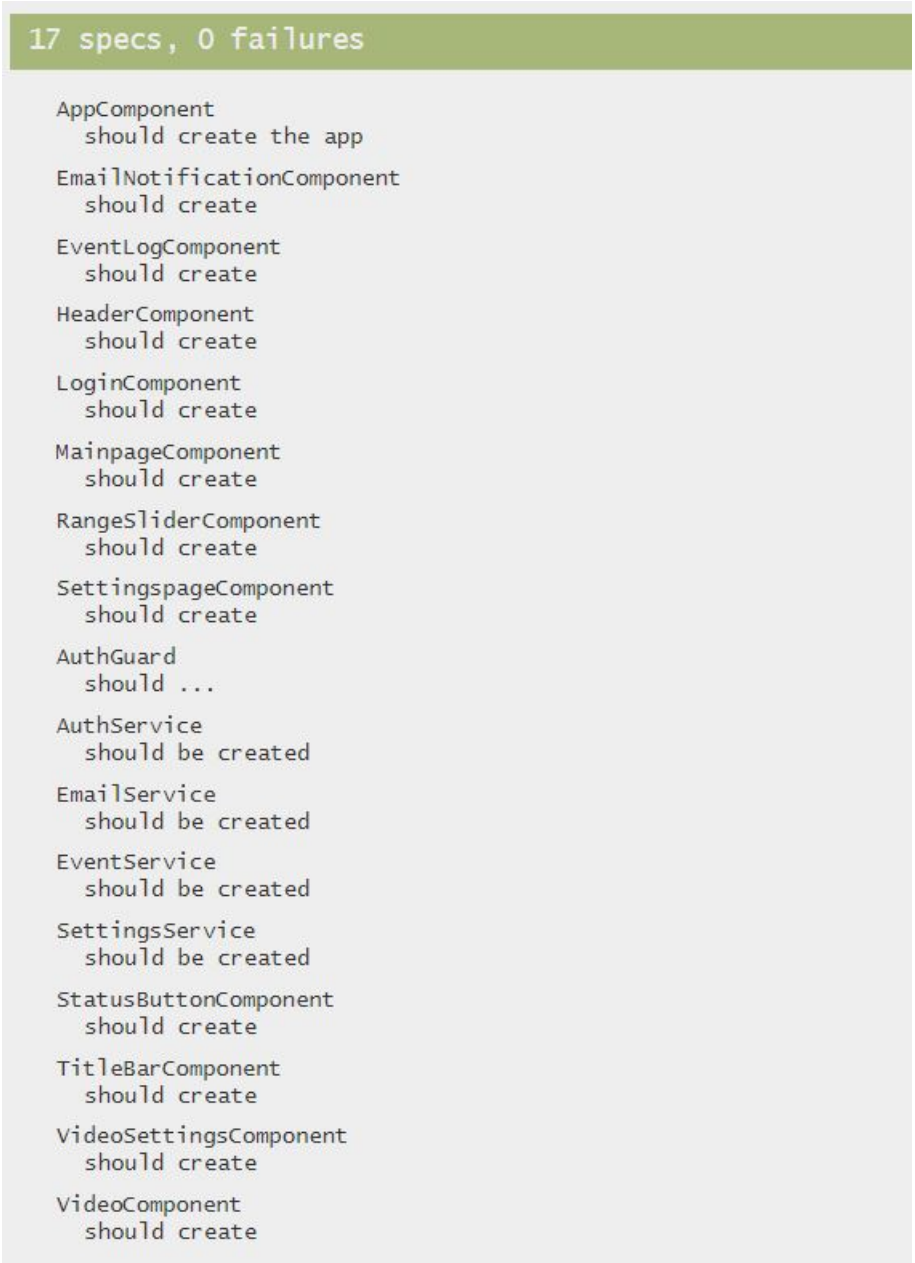

Abbildung 8.: Ergebnisse der automatisierten Front-End-Tests

5.4.2. 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 betriebenem Laptop mit einer Auflösung von 1920x1080 durchgeführt.

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 E-Mail-Adressen 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_storagee	1024

Tabelle 6.: Eingestellte Werte vor jedem manuellen Test

Alle manuellen Tests des Front-Ends inklusive deren Ergebnisse können im Anhand A eingesehen werden.

5.5. Testen des Back-Ends

Für das Testen der für das Back-End erstellten Tests wurde eine Testsuite angelegt, welche alle Testklassen auf einmal ausführt.

5.5.1. Webserver

Für alle Schnittstellen wurden Tests angelegt und mindestens auf Verfügbarkeit geprüft (Tabelle 7).

Tabelle 7.: Unittests für Webserver

Testname	Testmethode	Eingabe	Ausgabe
test_login_wrong	post /login	{'user': 'user', 'password': 'bla'}	Statuscode == 403
test_login_wrong_empty	post /login	{}	Statuscode == 403
test_login_correct	post /login	{'user': 'user', 'password': 'geheim'}	Statuscode == 200
test_videostream_available	get /videostream		Statuscode == 200
test_logs_length	get /logs/0/5		length == 4
test_logs_take_batch	get /logs/1/2		index 0 higher id than index 1
test_logs_take_not_existing_batch	get /logs/2/2		length == 0
test_log_delete_successful	delete /log/2		'log_id' == 2
test_log_delete_not_existing	delete /log/4		Statuscode == 403
test_mail_add_successful	post /mail	{'mail': 'bla@bla'}	'mail_id' == 4
test_mail_add_already_existing	post /mail	{'mail': 'bla@bla'}, {'mail': 'bla@bla'}	Statuscode == 403
test_mail_delete_successful	delete /mail/2		'mail_id' == 2
test_mail_delete_not_existing	delete /mail/4		Statuscode == 403
test_mail_toggle_notify	put /mail/2		'notify' == False
test_mails_length	get /mails		length == 4
test_change_config_all	post /config	{'sensitivity': 0.5, ...}	length == 9
test_change_config_specific	post /config	{'sensitivity': 0.5, 'streamaddress': 'blablubb', 'brightness': 0.0, 'contrast': 0.5}	length == 4
test_get_recording_successful	get /recording/test.mp4		'Content-Length' == 2107114
test_get_recording_not_existing	get /recording/bla.mp4		Statuscode == 404
test_get_backup_successful	get /backup		Statuscode == 200

5.5.2. Datenverwaltung

Für die Änderungen der Daten wurden ebenfalls Tests erstellt um mögliches Fehlverhalten auszuschließen (Tabelle 8).

Tabelle 8.: Unittests für Datenverwaltung

Testname	Testmethode	Eingabe	Ausgabe
test_toggle_mail_index_oor	toggle_mail_notify()	4	KeyError
test_toggle_mail	toggle_mail_notify()	0	False
test_add_email	add_mail(mail)	bla@bla	bla@bla
test_add_same_email_twice	add_mail(mail)	bla@bla, bla@bla	-1
test_remove_mail	remove_mail(id)	0	0
test_remove_not_existing_mail	remEove_mail(id)	4	KeyError
test_get_mails	get_mails()		len == 4
test_copy_of_mails	get_mails()	change mail1	mail1.address != mail2.address
test_get_settings	get_settings()	brightness = 0.83	brightness == 0.83
test_copy_of_settings	get_settings()	change contrast1	contrast1 != contrast2
test_change_settings	change_settings(...)	change all settings	all values changed
test_get_log_page	get_log_page(page,size)	both pages	order of ids correct
test_get_log_page_not_existing	get_log_page(page,size)	page 3	len(result) == 0
test_get_free_index	get_free_index()		4
test_add_log	add_log()		len(logs) == 5
test_add_log_exceed_limit	add_log()	add twice	len(logs) == 5 logs.get(0) == None
test_check_login_correct	check_login(user, password)	user, geheim	True
test_check_login_wrong	check_login(user, password)	user, asd	False
test_remove_log	remove_log(id)	0	0 logs.get(0) == None
test_remove_log_not_existing	remove_log(id)	4	KeyError

5.6. Testen des Gesamtsystems

Da bereits sehr viele Funktionen des Gesamtsystems durch Front-End- sowie Back-End-Tests abgedeckt wurden, wird hier nur noch bisher ungetestete Funktionen getestet.

5.6.1. Manuelle System-Tests

Da das Gesamtsystem aus zwei Teilprogrammen besteht, nämlich Back-End sowie dem Front-End, sind die Manuellen Tests an dieser Stelle am sinnvollsten. Automatisierte Tests, die auf beide Softwareteile zugreifen und diese auswerten, sind nur sehr schwer umzusetzen.
Bei den manuellen Systemtests wurde die Anwendung - wie es der Endanwender auch tun würde - über die Weboberfläche bedient. Es wurden auch hier die selben Softwareeinstellungen und Rahmenbedingungen wie bei den manuellen Front-End-Tests in 5.4.2 eingehalten.

Alle manuellen Tests des Gesamtsystems inklusive deren Ergebnisse können im Anhand B eingesehen werden.

5.6.2. Lasttests

Zusätzlich wurde das System mit verschiedenen Lasttests geprüft, die das Verhalten und die Performance unter verschiedenen Extrembedingungen testen sollten.

5.6.2.1. Stream Videoauflösung

In diesem Test wurden verschiedene Auflösungen des Videostreams mit deren Auswirkung auf die Berechnungszeit der Bewegungserkennung pro Bild ermittelt. Getestet wurde auf verschieden Systemen (Laptop/PC). Die Performance skaliert wie zu erwarten war sowohl mit der Leistung des Hostsystems, als auch mit der Anzahl an Pixeln, des Videostreams. Dabei ist zu beachten, dass, da der Prozess hauptsächlich einen einzelnen Kern belastet, die Anzahl der zur Verfügung stehenden Rechenkerne zu vernachlässigen ist.
Das Ergebnis des Tests ist, dass sich die Performace linear zur Anzahl der Pixel verhält und diese wiederum, bei linearer Steigerung der Auflösung, exponentiell wächst. Weiter konnten wir herausfinden, dass die Leistung bis zu einer Auflösung von 720p, auch mit laptop Hardware, noch im akzeptablen Bereich liegt. Alle höheren Auflösungen benötigen sehr starke Hardware, um akzeptable Bildraten zu erreichen. Bei einer Auflösung von 4k, welche sicher außerhalb der Spezifikation einer Überwachungskamera liegt, kam es sogar nach einiger Zeit zu Abstürzen.

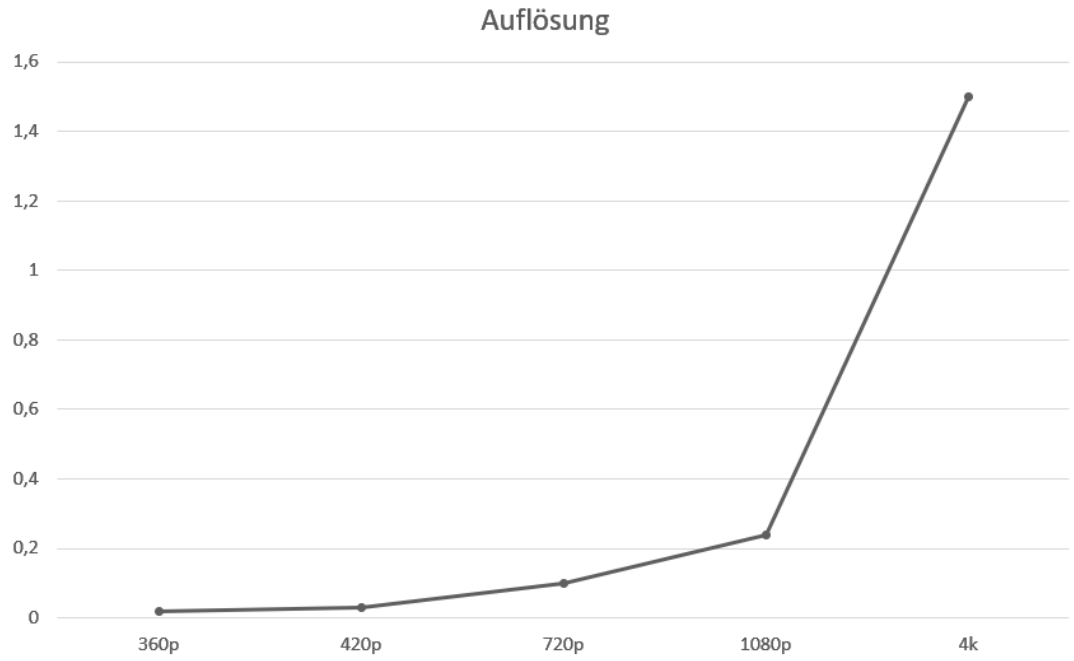

Abbildung 9.: Lasttest: Auflösungen

5.6.2.2. Dateigrößen Backup

Beim Erstellen eines Backups werden alle gesammelten Daten zusammengefasst und zum Download angeboten. Diese Lasttest soll die Abhängigkeit der benötigten Zeit zum erstellen eines eines solchen Backups mit der Dateigröße ermitteln.
Die benötigte Zeit zum erstellen des Backups steigt wie erwartet linear mit der Dateigröße, und es gibt keinen Limitierenden Faktor, wie zum Beispiel limitierter Arbeitsspeicher.

5.6.2.3. Aufnahmen

Auch die Aufnahmen wurden auf Anomalien unter extremen Bedingungen untersucht. So wurden beispielsweise eine Aufnahmedatei mit mehr als 5GB getestet. Diese verursachte keinerlei Probleme.

Als zweiten Test haben wir mehr als 500 einzelne Aufnahmen getestet. Auch hierbei gab es keine Probleme, jedoch stellte sich heraus, dass die Weboberfläche nicht optimal zum navigieren einer solchen Anzahl an Einträgen geeignet ist.

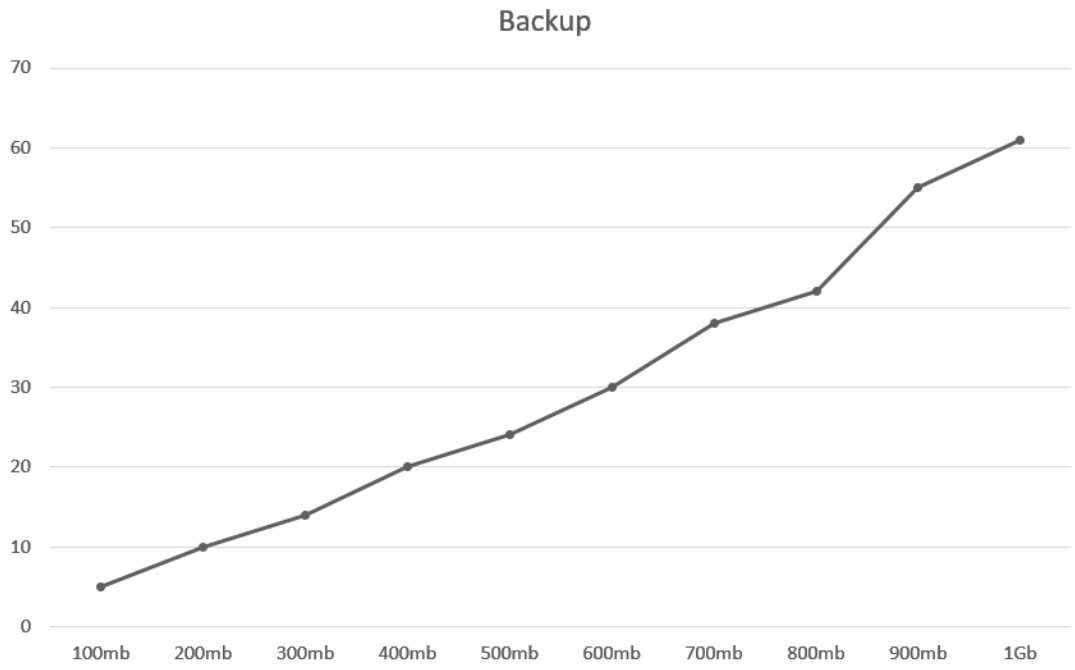

Abbildung 10.: Lasttest: Backup

6. Installation

6.1. System

Die Installationsanweisungen wurden auf einem Ubuntu Server der Version 18.10 sowie auf Wunsch des Kunden auch auf Ubuntu 16.04 durchgeführt.
Hierzu benötigte Images für Virtualbox können unter
`https://www.osboxes.org/ubuntu-server/` heruntergeladen werden. Die VM sollte über das Netzwerk erreichbar sein z.B. über die Netzwerkeinstellung „Bridge“. Des weiteren sollte darauf geachtet werden, dass genug Arbeitsspeicher (min. 4GB) sowie 4 Kerne der CPU zur Verfügung stehen.

6.2. 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.

6.2.1. Verwendete Versionen

- OpenCV 3.1.0
- Python 3.5.2
- Flask 1.0.2
- Flask-Cors 3.0.7

Für Ubuntu Server 18.10 wurde folgende Anleitung verwendet:
`https://www.pyimagesearch.com/2018/05/28/ubuntu-18-04-how-to-install-opencv/`

Für Ubuntu 16.04 wurde ein Installations-Skript basierend auf folgender Anleitung erstellt:
`https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/`
Dieses Skript ist innerhalb des Repositories zu finden, welches durch

```
1 git clone https://github.com/PIPCO-1819/PIPCO-Backend.git
```

heruntergeladen werden kann.
Anschließend wird das Installations-Skript ausgeführt.

```
./install_opencv.sh
```

Das Skript lädt alle notwendigen Pakete, erstellt eine virtuelle Umgebung und compiliert und installiert anschließend OpenCV.
Zusätzlich zur Installation von OpenCV muss Flask mit pip installiert werden. Hierzu muss wie in der Anleitung beschrieben .bashrc mit

```
source ~/.bashrc
```

ausgeführt werden. Daraufhin kann mit

```
workon cv
```

in der Virtuellen Python-Umgebung gearbeitet bzw. Flask mit CORS-Unterstützung wie folgt installiert werden:

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

Zuletzt muss nur noch das Back-End innerhalb des Repositories ausgeführt werden.

```
python3 Main.py
```

6.3. Frontend

6.3.1. Verwendete Versionen

- angular/cli 7.2.3
- npm 6.7.0
- nodejs 8.15.0

Für das Frontend wird Node.js, npm, sowie Angular verwendet.
Für Ubuntu 16.04 muss die Quelle für Node.js von Hand wie folgt aktualisiert und zuvor curl installiert werden:

```
sudo apt-get install curl
curl -sL https://deb.nodesource.com/setup_8.x -o nodesource_setup
    ↪ .sh
sudo bash nodesource_setup.sh
sudo apt-get install nodejs
```

Anschließend die aktuelle Version von npm sowie vom angular-cli installieren.

```
sudo npm install -g npm@latest
sudo npm install -g @angular/cli
```

Nach der Installation muss noch der Owner angepasst werden.

```
sudo chown -R $USER:$GROUP ~/.npm
sudo chown -R $USER:$GROUP ~/.config
```

Repository auschecken und restliche Abhängigkeiten installieren:

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

Nun muss die IP des Servers in /src/environments/environments.ts bei backendAddress eingetragen werden:

```
...
backendAdress: "http://192.168.0.125:8002",
...
```

Zuletzt kann der Server wie folgt gestartet werden:

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

6.4. Run on Startup

Für das Starten der Komponenten in einer eigenen Konsole wird **screen** verwendet, welches über apt-get installiert werden kann.
Die erstellten Skripte müssen mit

```
chmod +x skriptname.sh
```

ausführbar gemacht werden.

start_backend.sh in PIPCO-Backend

```
#!/bin/sh
printf "<MAIL_LOGIN>\n<PASSWORD>\n" | \
/home/<USER>/.virtualenvs/cv/bin/python3 Main.py
```

start_frontend.sh in PIPCO-Frontend

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

start_pipco.sh

```
#!/bin/bash
screen -dmS frontend bash -c \
'cd␣/home/<USER>/PIPCO-Frontend;␣./start_frontend.sh'
screen -dmS backend bash -c \
'cd␣/home/<USER>/PIPCO-Backend;.␣start_backend.sh'
```

rc.local bei Start des Systems ausführen

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

Skript zu rc.local hinzufügen

```
...
/home/<USER>/start_pipco.sh
exit 0
```

7. Ausblick

Eine einfache Bewegungserkennung und Aufnahme der Bewegung wurde erfolgreich umgesetzt. Des weiteren können bereits einige Einstellungen vorgenommen werden. Jedoch bietet dieses Projekt noch viele Möglichkeiten.
Die Berechnung des Medians kann in Zukunft auf mehrere Prozesse aufgeteilt werden oder es findet sich eine Möglichkeit die Erkennung der Bewegung über mehrere Frames einfacher zu gestalten. Mit einer besseren Performance könnten somit auch höhere Auflösungen verarbeitet werden. Außerdem wäre es praktisch mehrere Kamerastreams mit Tags (z.B. Wohnzimmer) der Website hinzuzufügen zu können und bei einer erkannten Bewegung zusätzlich darüber benachrichtigt zu werden, von welcher Kamera diese Bewegung erkannt wurde. Hierfür müsste ein eigener Prozess für jede Bildverarbeitunsroutine gestartet werden, wobei dann für die Datenhaltung auch eine neue Lösung (IPC) gefunden werden müsste, da diese Prozesse keinen gemeinsamen Bereich im RAM verwenden.
Eine weitere Verbesserung wäre die Kommunikation über den Front-End-Server zu leiten, welcher dann vom Backend über Änderungen informiert wird und diese aktualisierten Informationen an den Client weitergibt. Somit wäre das Back-End entlastet und könnte nach außen unsichtbar gemacht werden. Der Login ist momentan außerdem durch die jetzige Implementierung mit etwas Javascript-Änderungen zu umgehen. Es wird lediglich Client-seitig die Antwort des Back-Ends überprüft.
Die Benachrichtung könnte man vielleicht auch etwas moderner gestalten. Hierbei wäre eine App oder eine Progressive Web App für eine Push-Benachrichtigung interessant.
Mit OpenCV könnte auch in Richtung des Deep Learnings gearbeitet werden und damit vielleicht z.B. das Haustier mittels Object Detection erkannt und somit als Bewegung ignoriert werden.

8. Fazit

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

A. Testergebnisse der manuellen Front-End-Tests

Erklärung zur nachfolgenden Tabelle 9 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	Vorraussetz-ungen	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 hovert mit dem Cursor über das PIPCO-Logo auf der linken Siete 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 hovert 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 hovert 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 befinet 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 befinet 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 befinet 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 hovert 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 (Currentyl 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 (Currentyl 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 hovert 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 Maxixmalposition (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 hovert 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 hovert mit dem Cursor über abseits des Toggle-Switches der Event-Log-Komponente über die Titel-Leiste derselben Komponente.	Neben dem Cursor wir 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 hovert 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 hovert 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 hovert 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 neusten 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, einunzwanzigsten 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-Notifcation-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-Addresse (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-Addresse 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-Addresse 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-Addresse (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 6)	X	X
75	Settings-Page	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.		Das Input-Feld der Einstellungsmöglichkeit cliplength enthält den richtigen Wert (siehe Tabelle 6)	X	X
76	Settings-Page	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.		Das Input-Feld der Einstellungsmöglichkeit max_logs enthält den richtigen Wert (siehe Tabelle 6)	X	X
77	Settings-Page	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.		Das Input-Feld der Einstellungsmöglichkeit max_storage enthält den richtigen Wert (siehe Tabelle 6)	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 absofort die ganze breite der Browserfensters in Anspruch.	X	X

Tabelle 9.: Manuelle Front-End-Tests

B. Testergebnisse der manuellen Gesamtsystem-Tests

Erklärung zur nachfolgenden Tabelle 10 der manuellen Gesamtsystem-Tests:

Bedeutung der Spalte B*: Der Test wurde in durchgeführt und bestanden.

#	Vorraussetzungen	Aktion	Erwartetes Ergebnis	B*
1	Der Anwender befindet sich auf der Login-Seite und ist nicht eingeloggt.	Der Anwender versucht sich mit falschen Login-Daten einzuloggen (user: a und passwort: b).	Das Back-End erkennt die Login-Daten als falsch und sendet eine negative Antwort.	X
2	Der Anwender befindet sich auf der Login-Seite und ist nicht eingeloggt.	Der Anwender versucht sich mit falschen Login-Daten einzuloggen (user: a und passwort: b).	Der Login schlägt fehl.	X
3	Der Anwender befindet sich auf der Login-Seite und ist nicht eingeloggt.	Der Anwender versucht sich mit den richtigen Login-Daten einzuloggen (user: user und passwort: geheim).	Das Back-End erkennt die Login-Daten als richtig und sendet eine positive Antwort.	X
4	Der Anwender befindet sich auf der Login-Seite und ist nicht eingeloggt.	Der Anwender versucht sich mit den richtigen Login-Daten einzuloggen (user: user und passwort: geheim).	Der Login erfolgt.	X
5	Es wird eine neue Bewegung detektiert.		Das Back-End erzeugt und speichert eine neue Aufnahme.	X
6	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite. Es wird eine neue Bewegung detektiert.		Die erkannte Bewegung wird im im Output-Livestream hervorgehoben.	X
7	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite. Es wird eine neue Bewegung detektiert.		Ein neuer Eintrag wird in der EventLogComponent angezeigt.	X

8	Es wird eine neue Bewegung über 12 Sekunden detektiert.		Das Back-End erzeugt und speichert eine neue Aufnahme mit einer Länge von nur 10 Sekunden.	X
9	Es wurden bereits 20 Aufnahmen gespeichert. Es wird eine neue Bewegung detektiert.		Die neue Bewegung wird aufgezeichnet und gespeichert.	X
10	Es wurden bereits 20 Aufnahmen gespeichert. Es wird eine neue Bewegung detektiert.		Die älteste gespeicherte Aufzeichnung wird gelöscht.	X
11	Es wurden bereits 20 Aufnahmen gespeichert. Es wird eine neue Bewegung detektiert.		Die älteste gespeicherte Aufzeichnung wird gelöscht.	X
12	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.	Der Anwender betätigt den Button zum Downloaden eines Backups.	Das Back-End erzeugt eine einwandfreie zip-Datei aller Konfigurations- und Logdateien.	X
13	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.	Der Anwender betätigt den Button zum Downloaden eines Backups.	Eine einwandfreie zip-Datei aller Konfigurations- und Logdateien wird über den Browser heruntergeladen.	X
14	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite. Es wird eine neue Bewegung detektiert.		Die neue Aufnahme ist über die EventLogComponent abspielbar.	X
15	Es wird eine neue Bewegung detektiert.		Zur neuen Aufnahme wurde eine neue einzigartige ID erzeugt.	X
16	Es wird eine neue Bewegung detektiert.		Alle registrierten E-Mails wurden benachrichtigt.	X
17	Es wird eine neue Bewegung detektiert. Die E-Mail-Funktion wurde deaktiviert.		Es wurden keine E-Mail-Benachrichtigungen versendet.	X

18	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender fügt über die EmailNotificationComponent eine neue E-Mail-Adresse hinzu (test@test.com).	Die neue E-Mail-Adresse wurde richtig im Back-End gespeichert.	X
19	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender löscht eine gespeicherte E-Mail über die EmailNotificationComponent.	Im Back-End wird die richtige E-Mail erfolgreich aus dem Datenbestand gelöscht.	X
20	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender verschiebt in der VideoSettingsComponent den Regler zur Einstellung „sensitivity“ ganz nach rechts.	Im Back-End wird der neue Einstellungswert (1.0) korrekt gespeichert.	X
21	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender verschiebt in der VideoSettingsComponent den Regler zur Einstellung „sensitivity“ ganz nach rechts.	Die neue Einstellung wirkt sich korrekt auf den Output-Livestream des Back-Ends aus.	X
22	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender verschiebt in der VideoSettingsComponent den Regler zur Einstellung „brightness“ ganz nach rechts.	Im Back-End wird der neue Einstellungswert (1.0) korrekt gespeichert.	X
23	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender verschiebt in der VideoSettingsComponent den Regler zur Einstellung „brightness“ ganz nach rechts.	Die neue Einstellung wirkt sich korrekt auf den Output-Livestream des Back-Ends aus.	X
24	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender verschiebt in der VideoSettingsComponent den Regler zur Einstellung „contrast“ ganz nach rechts.	Im Back-End wird der neue Einstellungswert (1.0) korrekt gespeichert.	X
25	Der Anwender ist korrekt eingeloggt und befindet sich auf der Hauptseite.	Der Anwender verschiebt in der VideoSettingsComponent den Regler zur Einstellung „contrast“ ganz nach rechts.	Die neue Einstellung wirkt sich korrekt auf den Output-Livestream des Back-Ends aus.	X

26	Die Funktion der Bewegungserkennung wurde deaktiviert. Es wird eine Bewegung detektiert.		Es wird keine Aufnahme der erkannten Bewegung erzeugt.	X
27	Die Funktion der Bewegungserkennung wurde deaktiviert. Es wird eine Bewegung detektiert.		Die gespeicherten E-Mails werden nicht über die erkannte Bewegung in Kenntnis gesetzt.	X
28	Die Funktion der Bewegungserkennung wurde deaktiviert. Es wird eine Bewegung detektiert.		Die erkannte Bewegung wird noch immer im Output-Livestream hervorgehoben.	X
29	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.	Der Anwender verändert und speichert die Einstellung „streamaddress“.	Die neue Adresse wird korrekt im Back-End gespeichert.	X
30	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.	Der Anwender verändert und speichert die Einstellung „streamaddress“. Die neue Adresse verweist auf einen für die Anwendung brauchbaren Livestream.	Der Output-Livestream beinhaltet nun den Livestream der eingegebenen Adresse.	X
31	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.	Der Anwender verändert und speichert die Einstellung „streamaddress“. Die neue Adresse verweist auf einen für die Anwendung nicht brauchbaren Livestream.	Es werden keine neuen Frames über den Output-Livestream ausgegeben.	X
32	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.	Der Anwender verändert und speichert die Einstellung „cliplength“.	Die neue Einstellung wird korrekt im Back-End gespeichert.	X
33	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.	Der Anwender verändert und speichert die Einstellung „max_logs“.	Die neue Einstellung wird korrekt im Back-End gespeichert.	X
34	Der Anwender ist korrekt eingeloggt und befindet sich auf der Settings-Seite.	Der Anwender verändert und speichert die Einstellung „max_storage“.	Die neue Einstellung wird korrekt im Back-End gespeichert.	X

Tabelle 10.: Manuelle Gesamtsystem-Tests