

Projektdokumentation

im Studiengang

AIN

**PIPCO**

Private IP Camera Observation

Referent : Prof. Dr. Elmar Cochlovius

Vorgelegt am : 13.01.2019



## Abstract

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

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



## Inhaltsverzeichnis

Abstract . . . . .	i
Inhaltsverzeichnis . . . . .	iii
Abbildungsverzeichnis . . . . .	v
Tabellenverzeichnis . . . . .	vii
Abkürzungsverzeichnis . . . . .	ix
1 Einleitung . . . . .	1
1.1 Rahmenbedingungen . . . . .	1
2 Installation . . . . .	3
2.1 System . . . . .	3
2.2 Backend . . . . .	3
2.3 Frontend . . . . .	3
2.4 Run on Startup . . . . .	4
3 Tests . . . . .	7
3.1 Testen des Front-Ends . . . . .	7
3.1.1 Manuelle Tests . . . . .	7
4 [Eigene Kapitel] . . . . .	9
5 Ausblick . . . . .	11
6 Fazit . . . . .	13
Eidesstattliche Erklärung . . . . .	15



## Abbildungsverzeichnis





## Tabellenverzeichnis



## Abkürzungsverzeichnis

**IP** Internet Protocol



# 1 Einleitung

## 1.1 Rahmenbedingungen

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

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

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



## 2 Installation

### 2.1 System

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

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

### 2.2 Backend

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

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

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

geladen werden. Anschließend kann mit

```
1 workon cv
```

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

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

Nun muss nur noch das Repository ausgecheckt und ausgeführt werden.

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

### 2.3 Frontend

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

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

Repository auschecken und restliche dependencies installieren:

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

Anschließend den Server wie folgt starten:

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

## 2.4 Run on Startup

start\_backend.sh in PIPCO-Backend

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

start\_frontend.sh in PIPCO-Frontend

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

start\_frontend.sh in PIPCO-Frontend

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

start\_pipco.sh

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

rc.local bei Start des Systems ausführen

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



Skript zu rc.local hinzufügen

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



## 3 Tests

### 3.1 Testen des Front-Ends

#### 3.1.1 Manuelle Tests

Beim manuellen testen handelt es sich um eine Testmethode, bei dem der Tester ohne die Verwendung von Automatisierungstools vorgeht. Dabei können durch die systematische Verwendung der Software häufig Fehler aufgedeckt werden, die etwa bei Unit-Tests oft nicht gefunden werden. Insbesondere kleinere Fehler in Benutzeroberflächen fallen auf diese Weise schnell auf.

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



## 4 [Eigene Kapitel]



## 5 Ausblick





## 6 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