



Raumüberwachung mit Smartphones

Software Modelling and Design

Project 1 (BTI3301) 21

Studiengang:

Informatik

Autor:

Mohammed Ali, Mac Müller

Betreuer:

Stefan Cotting - cis1

Datum:

03.05.2021

Inhaltsverzeichnis

	Software Modelling and Design	1
1	Einleitung	3
	1.1 Einleitung der Technologien für die Realisierung	3
	1.1.1 WebRTC (Web Real-Time Communication)	3
	1.1.2 Node.js	4
	1.1.3 HTTPS (Hyper Text Transfer Protocol Secure)	4
2	Anwedung-View	5
	2.1 Anwendungsfalldiagramm	5
	2.1.1 Use Case: Allgemeines Überwachungssystem	5
	2.1.2 Use Case: Das Smartphone als Kamera mit dem Server einbinden	6
	2.1.3 Use Case: Mit dem Smartphone das Live-Video überwachen	7
	2.1.4 Use Case: Mit dem Smartphone die aufgenommenen Videos abspielen	8
3	Komponente-View	9
	3.1 Anwendungsoberfläche mit SPA	9
	3.2 Domain Model	10
4	Dynamic View	11
	4.1 Sequenzdiagramm: Live-Video-Übertragung	11
	4.2 Sequenzdiagramm: Live-Video ansehen	11
5	Deployment View	12
6	Design Decisions	13
	6.1 User Interface	13
	6.2 Farben Schema	14
7	Test	14

1 Einleitung

Im Rahmen des "Projektes 1" an der Berner Fachhochschule arbeiten wir als Gruppe mit zwei Studenten. Ziel dieses Projektes umfasst ein System für die Raumüberwachung mit Smartphones zu realisieren.

In diesem Abschnitt werden das Software-Modelling und Software-Design für die zukünftige Realisierung des Systems zusammengefasst. Ziel dieser Dokumentation ist es, den Überblick der aufzubauen Anwendung ausführlich zu beschreiben. Die Softwareentwickler sollen anhand dieser Dokumentation die Raumüberwachung mit Smartphones verwirklichen. Die Anforderungen des Stakeholders, welche im Anforderungsdokument definiert wurden, müssen erfüllt werden.

1.1 Einleitung der Technologien für die Realisierung

1.1.1 WebRTC (Web Real-Time Communication)

Für die Realisierung des Webserver wird die Web-Echtzeitkommunikation Technologie verwendet.

Die WebRTC ermöglicht mittels JavaScript APIs eine webbasierte Echtzeitkommunikation (z.B. Videokonferenz) ohne die Notwendigkeit zusätzlicher Client-Software oder Plugins.

Die Technologie ist in allen modernen Browsern sowie auf nativen Clients für alle wichtigen Plattformen verfügbar, in anderen Worten ist sie Betriebssystem unabhängig.

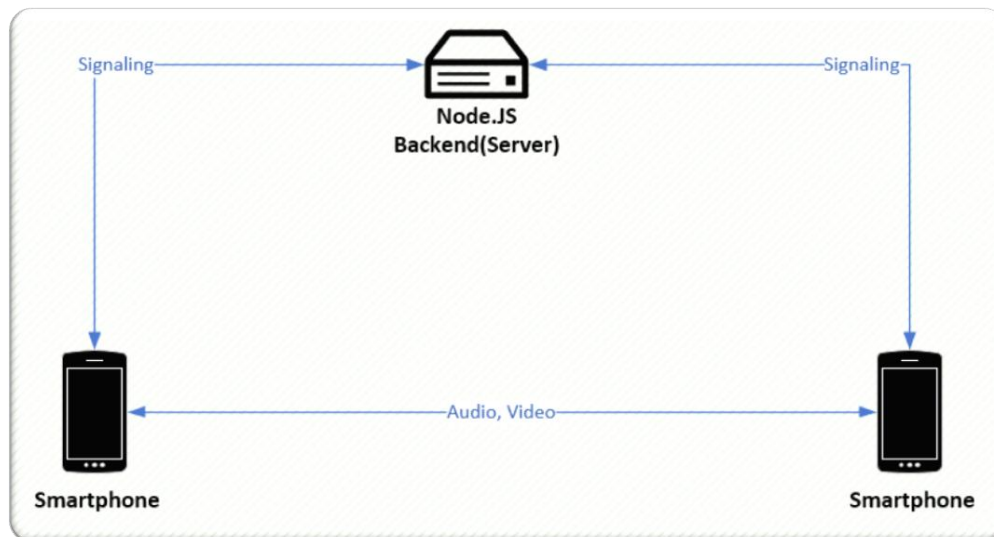
Smartphones «Clients» werden als Peers betrachtet, die erste Verbindung zwischen den zwei Peers wird über einen Server aufgebaut.

Um WebRTC restlos verwenden zu können, müssen ein Paar Punkte berücksichtigen werden:

- Alle Clients müssen im gleichen Netzwerk sein.
- Smartphones verfügen über den Browser Google Chrom ab Version 91.0.4472.77
- Beim Starten der Kamera muss der Zugriff auf Audio- & Video-Hardware erlaubt werden

WebRTC funktioniert wie folgt:

- Client 1 sendet eine TCP-Session Anfrage an den Server
- Der Server speichert die Session von Client 1
- Client 2 sendet eine TCP-Session Anfrage an den Server
- Der Server speichert die Session von Client 2
- Der Server sendet die Sessions von Client 2 zu Client 1
- Der Server sendet die Sessions von Client 1 zu Client 2
- Somit können die zwei Clients unabhängig in gemeinsamen Sessions miteinander kommunizieren



WebRTC

1.1.2 Node.js

Um die Verbindung zwischen zwei Browser-Sessionen herzustellen, benötigt man einen Server für die Vermittlung. Das Node.js zusammen mit der JavaScript-Bibliothek Socket.IO ermöglichen die Web-Echtzeitkommunikation.

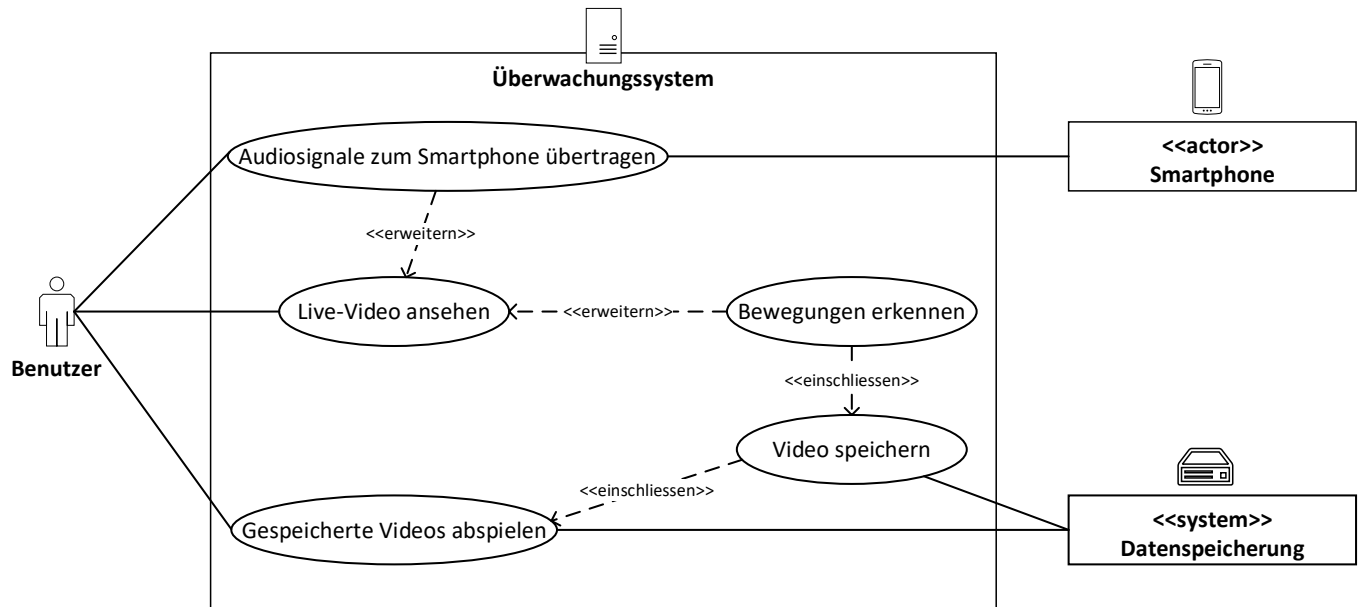
1.1.3 HTTPS (Hyper Text Transfer Protocol Secure)

Das Protokoll verschlüsselt die Kommunikation zwischen dem Browser des Smartphones und dem Server. Der Kamera- und Mikrofon-Zugriff ist ohne das HTTPS-Protokoll nicht möglich.

2 Anwendung-View

2.1 Anwendungsfalldiagramm

2.1.1 Use Case: Allgemeines Überwachungssystem



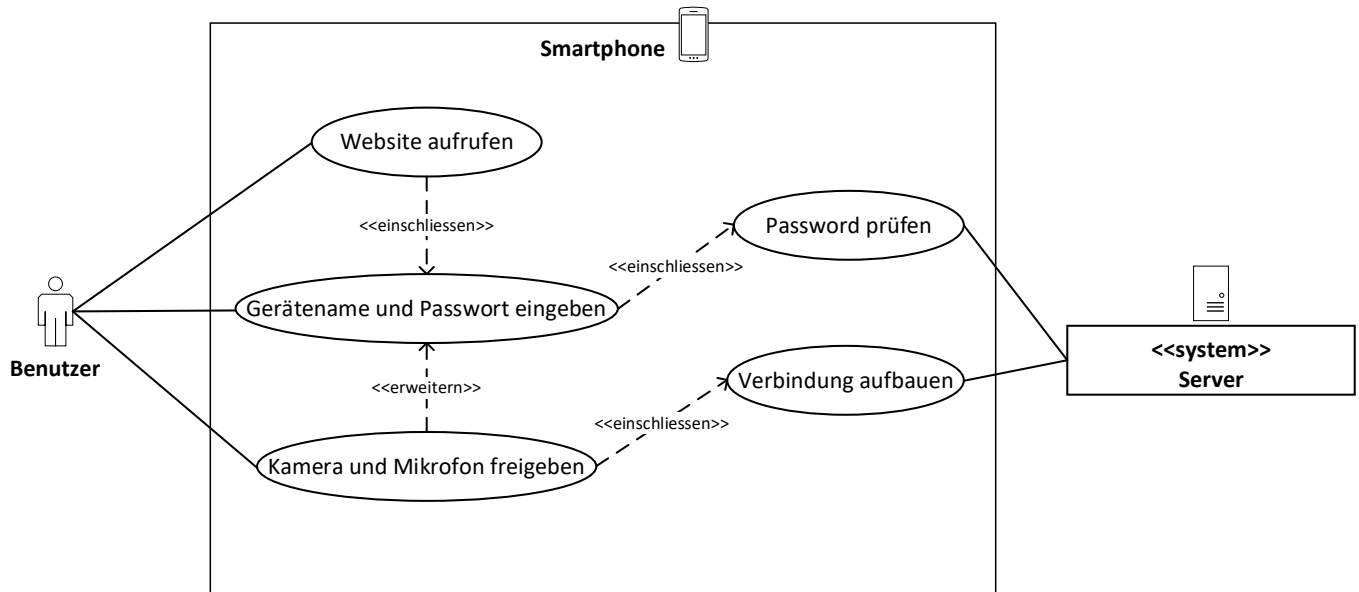
Das Überwachungssystem soll als einen Webserver dienen, welcher die Live-Übertragung von der Kamera aufnimmt. Der Benutzer kann das Video vom Serverdienst live streamen.

Bedingung: Alle Aktoren ausser Datenspeicherung müssen im gleichen LAN sein.

Der Benutzer hat bei dem Webserver die folgenden Funktionen zur Verfügung:

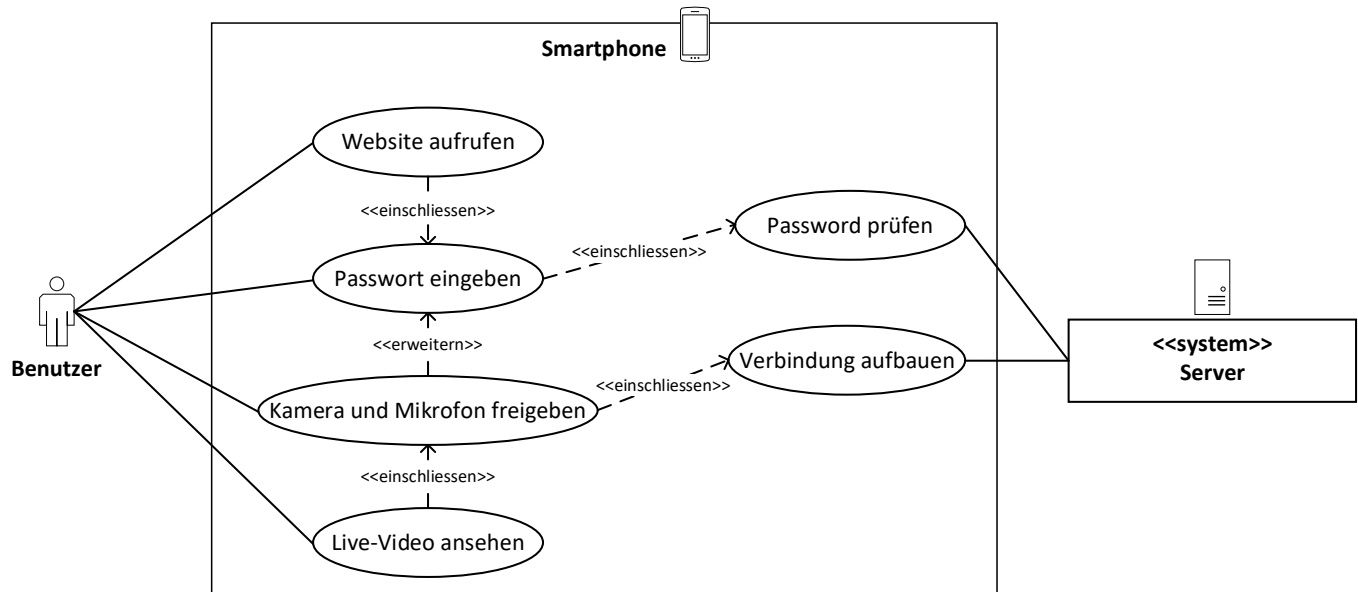
- Live-Video streamen
- Die Liste der aufgenommenen Videos abspielen
- Durch den Lautsprecher des überwachenden Smartphones sprechen

2.1.2 Use Case: Das Smartphone als Kamera mit dem Server einbinden



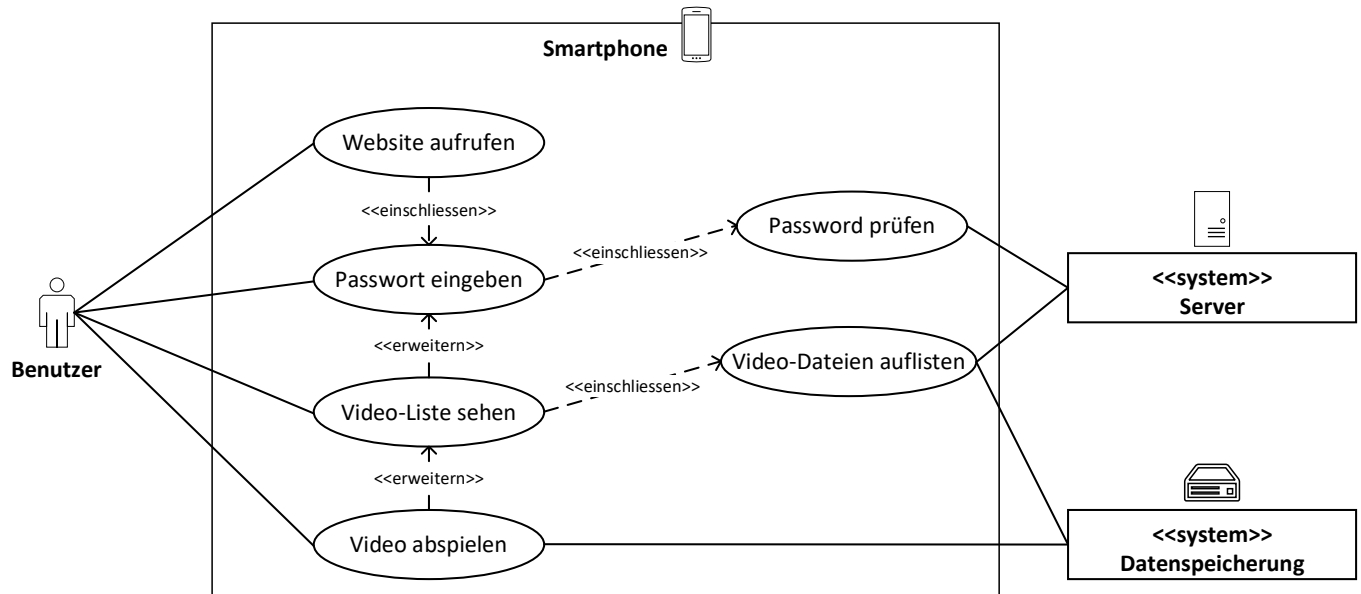
Trigger	Der Benutzer will sein Smartphone als Überwachungskamera verwenden.
Vorbedingung	Ein Geräte- und das Zugangspasswort für die Anwendung.
Szenario	1. Mit dem Browser des Smartphones die Webseite aufrufen 2. Den Gerätenamen und das Passwort eingeben 3. Das Passwort wird von Server geprüft 4. Der Benutzer erhält eine Anfrage von seinem Kamera- und Mikrofon-Zugriff 5. Der Benutzer erlaubt dem Browser den Zugriff
Error Szenario	Falsches Zugangspasswort, keine Verbindung, kein Zugriff auf die Kamera und das Mikrofon.
Resultat	Video-Übertragung wird gestartet. Der Benutzer sieht sein Video auf dem Webbrowser des Smartphones.
Nachbedingung	Der Benutzer muss seinen Kamera- und Mikrofon-Zugriff erlauben.

2.1.3 Use Case: Mit dem Smartphone das Live-Video überwachen



Trigger	Der Benutzer will das freigeschaltetes Live-Video ansehen.
Vorbedingung	Das Zugangspasswort für die Anwendung.
Szenario	<ol style="list-style-type: none"> 1. Mit dem Browser des Smartphones die Webseite aufrufen 2. Das Passwort eingeben 3. Das Passwort wird von Server geprüft 4. Der Benutzer erhält eine Anfrage von seinem Kamera- und Mikrofon-Zugriff 5. Der Benutzer erlaubt dem Browser den Zugriff
Error Szenario	Falsches Zugangspasswort, keine Verbindung, kein Zugriff auf die Kamera und das Mikrofon des Benutzers.
Resultat	Die Verbindung zu Live-Video wird aufgebaut. Der Benutzer sieht das Live-Video auf dem Webbrowser des Smartphones.
Nachbedingung	Der Benutzer muss seinen Kamera- und Mikrofon-Zugriff erlauben.

2.1.4 Use Case: Mit dem Smartphone die aufgenommenen Videos abspielen

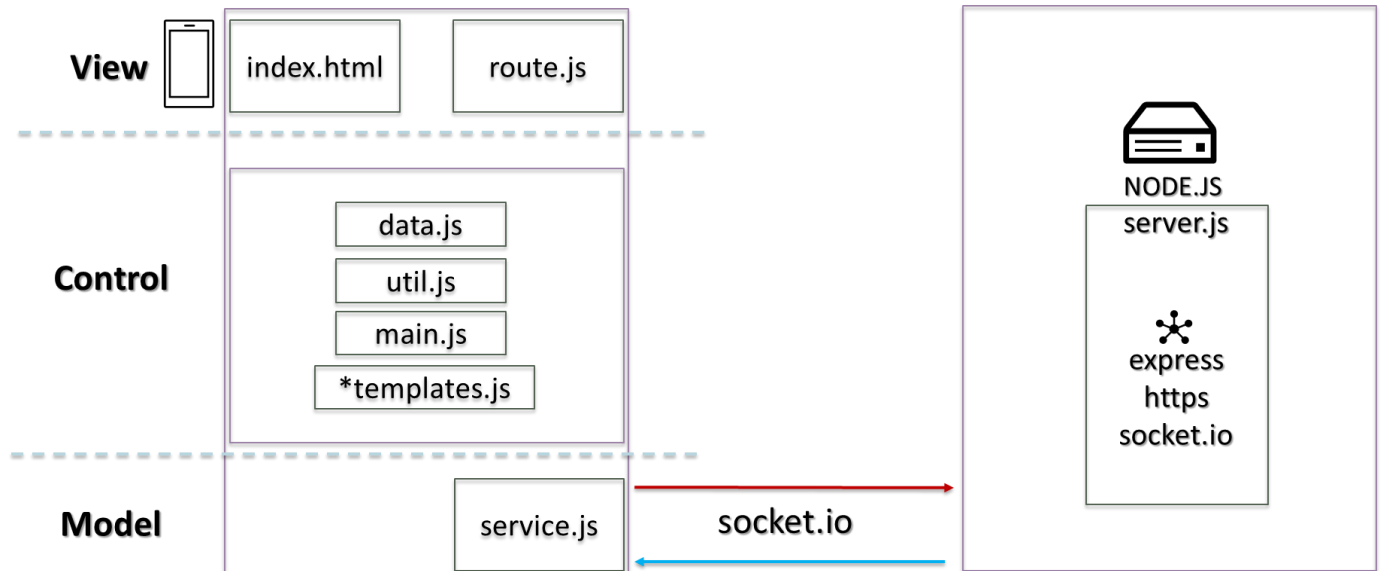


Trigger	Der Benutzer möchte das aufgenommene Video abspielen.
Vorbedingung	Das Zugangspasswort für die Anwendung
Szenario	1. Mit dem Browser des Smartphones die Webseite aufrufen 2. Navigation "Menü" und "Videos" auswählen 3. Das Passwort eingeben 4. Das Passwort wird vom Server geprüft 5. Der Benutzer kann die Liste der aufgenommenen Videos sehen
Error Szenario	Falsches Zugangspasswort, keine Verbindung.
Resultat	Der Benutzer kann die Liste der aufgenommenen Videos sehen, und kann dann ein Video auswählen und abspielen.
Nachbedingung	-

3 Komponente-View

3.1 Anwendungsoberfläche mit SPA

Das Paketdiagramm zeigt die Software-Komponente und die Assoziationen von der SPA.



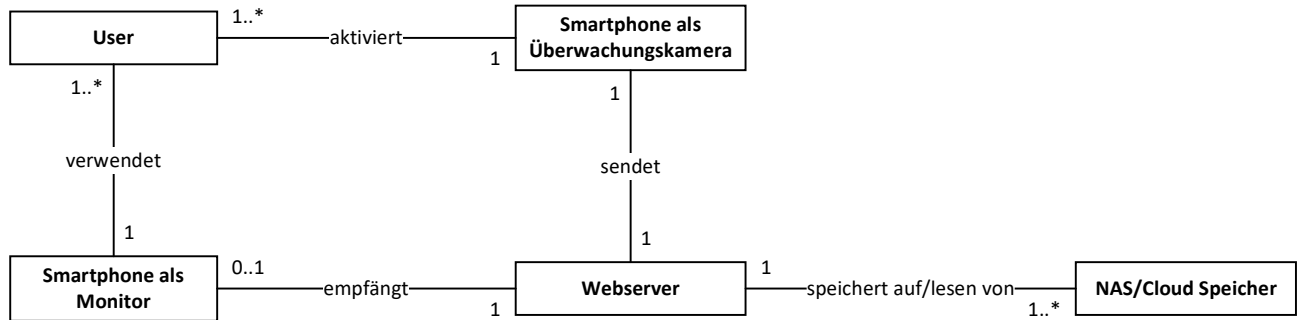
Für eine benutzerfreundliche Webseite wird eine Single-Page-Webanwendung (englisch: Single Page Application - **SPA**) sowie MVC-Pattern verwendet. Der Router in der View sucht von dem Benutzer ausgewählte Webpage einen entsprechenden Controller aus. Der Controller generiert die Inhalte der Webseite als HTML-Elemente und gibt diese an den Router zurück. Der Router überschreibt das "`<main>`"-Element in `index.html`. Somit muss die Seite bei jeder Anfrage der Benutzer nicht komplett neu geladen werden.

- `data.js`: ist ein Dienst für nicht persistente Daten z.B. eine Liste von Namen der Video-Dateien
- `util.js`: hier werden alle globalen Funktionen zur Verfügung gestellt, diese können alle Controller gemeinsam verwenden z.B. Anzeigen der Fehler-Meldungen

Das Model ist zuständig für die zusammen nutzbaren Ressourcen mit `service.js`:

- Ist ein Dienst für die Übertragungen von Videos sowie die Verbindungsaufbau
- Ist ein Dienst für die Abfragung über persistente Daten z.B. Login-Passwort

3.2 Domain Model



Das Domain-Model stellt die Übersicht der wichtigen Akteure in der Realität von der Systems-Umgebung dar.

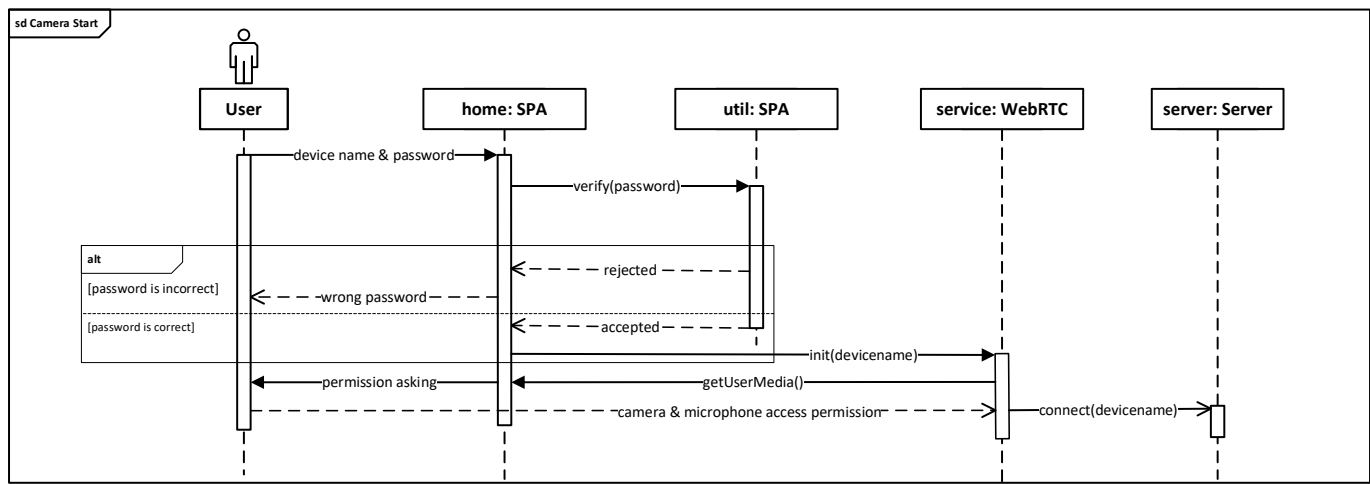
Für die Anwendung benötigt der Benutzer zwei Smartphones. Das erste Smartphone kann der Benutzer als Überwachungskamera aktivieren. Das zweite Smartphone kann von dem User als Monitor verwendet werden.

Das Smartphone als Überwachungskamera hat eine 1:1-Verbindung zum Server. Das Smartphone als Monitor kann das Video ansehen. An den Webserver wurde mindestens einen NAS/Cloud Speicher angeschlossen.

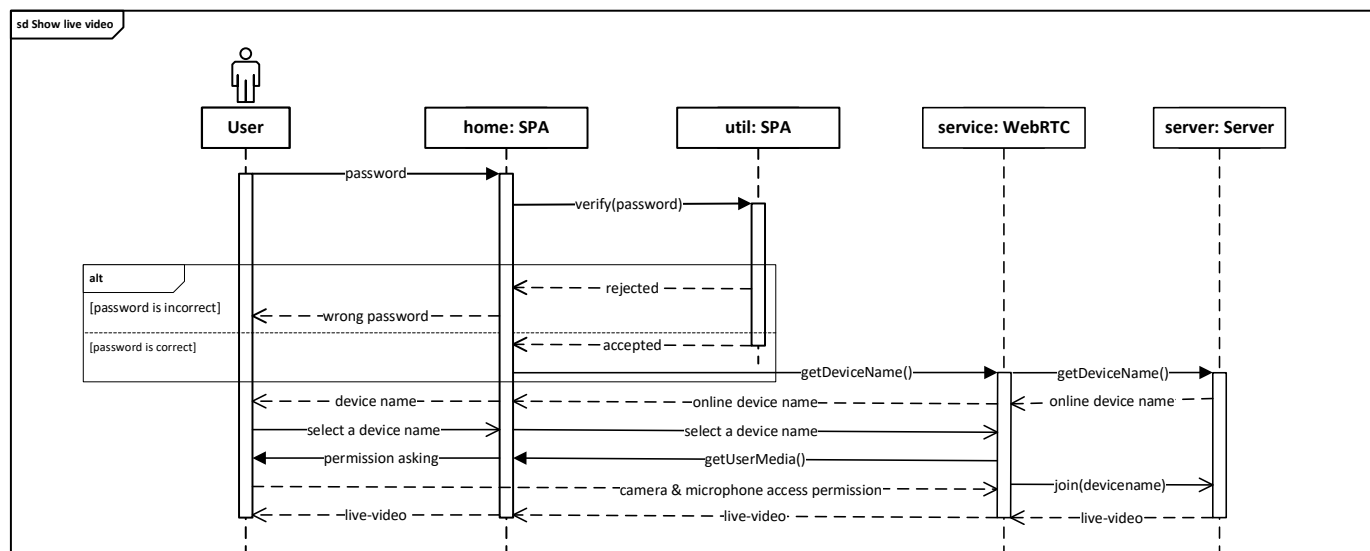
4 Dynamic View

Die Sequenzdiagramme zeigen die Reihenfolgen der Kommunikation zwischen den Komponenten.

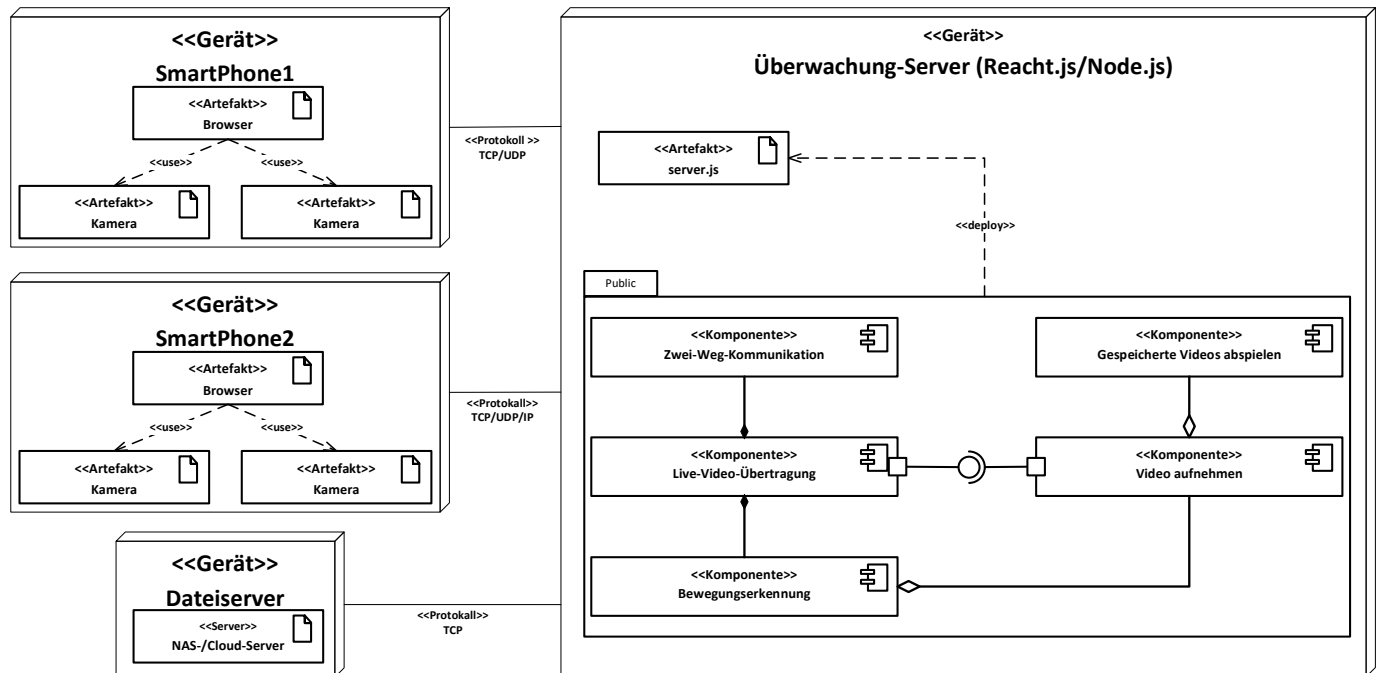
4.1 Sequenzdiagramm: Live-Video-Übertragung



4.2 Sequenzdiagramm: Live-Video ansehen



5 Deployment View



Das Verteilungsdiagramm (englisch deployment diagram) stellt die Architektur der physischen Hardware dar.

Die Kommunikation zwischen den Geräten basieren auf TCP und UDP. Der Überwachungsserver kann entweder ein Server mit Reacht.js oder ein Server mit Node.js sein.

Bei den beiden Smartphones werden zwei Kamera-Artefakte zur Verfügung stehen. Ein Artefakt ist der Stream von der eigenen Kamera. Das andere Artefakt ist der Stream des anderen Smartphones.

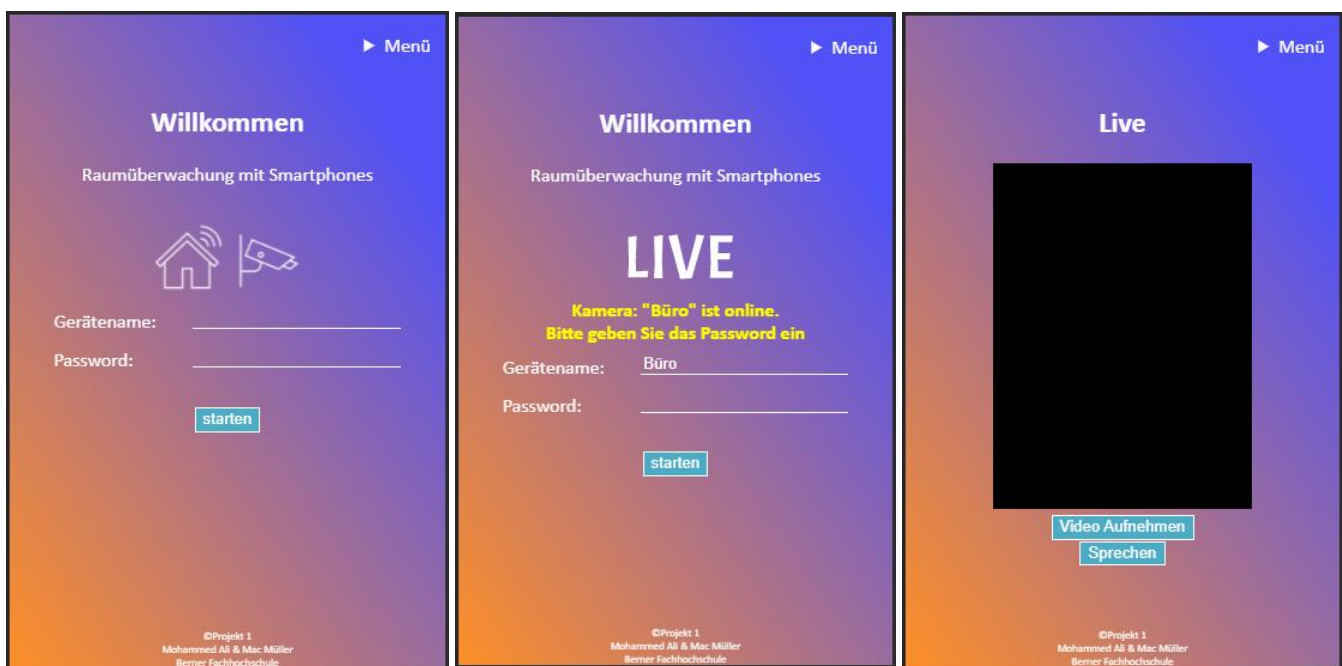
Ein Smartphone als Überwachungskamera zeigt auf seinem Webbrowser nur den Stream der eigenen Kamera. Ein anderes Smartphone als Überwachungsmonitor zeigt auf seinem Webbrowser nur den Stream des anderen Smartphones.

Die Voraussetzungen für die Hardware sind unter Kapitel 7 zu finden.

6 Design Decisions

Der Zugriff von der Anwendung erfolgt über einen Webbrowser und dem HTTPS-Protokoll. Das Design wurde wie unter Punkt 6.1 definiert. Des Weiteren sind die Farbschemen für die GUI unter Punkt 6.2 festgehalten. Die Schriftart ist frei wählbar, und kann während der laufenden Entwicklung geändert werden.

6.1 User Interface



6.2 Farben Schema

Haupt-Farben

HEX: #243FFF

HEX: #FF9022

Text-Farben

WHITE

Button-Farben

HEX: #4CABC5

7 Test

In diesem Abschnitt listen wir die verschiedenen Geräte und die Betriebssysteme, welche mit dem implementierten System getestet werden.

1. Xiaomi Redmi 9 (Android 10)
 - Google Chrom (Version 90.0.4430.66)
2. Huawei P20 Pro (Android 10)
 - Google Chrom (Version 90.0.4430.210)
3. iPhone10 und iPad Air 2 (iOS 14)
 - Safari Browser
4. Ein Laptop mit Microsoft Windows
 - Firefox Version 89
 - Google Chrome Version 91.0.4472.114

Funktionen zu testen:

1. Login
2. Live-Überwachen
3. Zwei-Weg-Kommunikation
4. Die Aufzeichnung Starten/Stoppen
5. Die Aufzeichnung herunterladen