


 + Laura Strickmann	2194476
 + Nadine Stellmacher	2210151
 + Nadine Winter	2191707
 + Xuan Linh Do	2138196

# SUCHERGEBNISSE

➤ hardware	1
➤ usability	2
➤ software/programmierung	2-3
➤ problematik	4
➤ anhang	5-8

# ☑hardware

## Materialien

- 2 RIBBA Ikea Rahmen mit Glasscheibe (je 50x50 cm, Tiefe von 4,5 cm)
- ein alter Monitor von NEC 16:9 22" (DVI/VGA)
- DVI auf HDMI Adapter Kabel
- Kaltgerätekabel
- Raspberry Pi
- 2 Verschlüsse + 2 Scharniere
- D-c-fix Spiegel-Sichtschutzfolie

Die wichtigsten Bestandteile unseres Projektes „SMIRROR“ sind der Monitor und der halbdurchlässige Spiegel. Da es preislich nicht in unserem Budget lag einen richtigen halbdurchlässigen Spiegel zu kaufen, haben wir uns dazu entschieden diesen selber zu ‚bauen‘. Das Selbstbauen ist recht simpel. Dafür benötigt man nur die oben genannte Folie und eine Glasscheibe. Die Folie ist von der einen Seite lichtdurchlässig und auf der anderen Seite verspiegelt. Diese wird einfach auf die Glasscheibe geklebt. Bei dem Monitor haben wir uns für das entschieden, was uns zur Verfügung stand: ein NEC Monitor 16:9 22“. Hieran machten wir auch die Größe unseres Spiegels, sowie die Tiefe des Rahmens fest und entschieden uns für den RIBBA (50x50cm) Rahmen mit Glasscheibe von Ikea.

## Aufbau (siehe Anhang 1 & 2)

Mit Gehäuse war der NEC Monitor zu groß für unseren improvisierten Spiegel, weshalb wir diesen abmontierten. Die mit Spiegelfolie beklebte Glasscheibe wird mit der spiegelnden Seite nach außen in den Rahmen gelegt, die lichtdurchlässige Seite zum Monitor. Um den Monitor an seinem Platz zu befestigen, nutzten wir den im Rahmen befindlichen Innenrahmen als Stütze.

Da der Monitor keinen HDMI Ausgang besitzt, benötigten wir einen Adapter um diesen mit dem RaspberryPi zu verbinden. Der RaspberryPi hatte zusätzlich einen Wlan-Stick. Zur Stromversorgung schließen wir den Monitor über ein Kaltgerätekabel an das Stromnetz an. Da wir nicht den Display von den Anschlüssen und der Lüftung trennen wollten, hatte unser Monitor eine größere Tiefe als der eigentliche Rahmen. Deswegen haben wir uns dazu entschlossen auf der Hinterseite einen weiteren baugleichen Rahmen anzubringen. Dieser behält die beigelegte Glasscheibe um den Spiegel von hinten als eine Art „Schaukasten“ zu betrachten. Die beiden Rahmen haben wir auf der einen Seite mit Scharnieren und auf der anderen Seite mit Verschlüssen verbunden. So kann man die Rückseite wie eine Tür zur Seite hin öffnen. In den hinteren Rahmen haben wir seitlich zwei Löcher für die Kabel gebohrt. Da auf der Hinterseite eine Glasscheibe ist, kann von dort Licht hereinfallen. Dies ist jedoch hinderlich, da von der Seite nur das Licht des Displays durchscheinen sollte. Deswegen haben wir die eigentliche Rückwand des Bilderrahmens mit etwas weißem Gaffa - passend zum Rahmen - auf der Rückseite befestigt.

## Alternativen zu unserem Aufbau

Statt des Monitors hätte man einen deutlich flacheren Display mit externem Controller-Board verwenden können.

Statt der selbstklebenden Spiegelfolie auf der Glasscheibe hätte man ein Spiegelglas verwenden können. Diese sind qualitativ hochwertiger, weniger transparent und geben ein ebenmäßiges Spiegelbild. Dadurch sind diese auch deutlich teurer.

Statt des fertigen Ikea-Rahmens könnte man sich einen individuellen Rahmen anfertigen, wo man die Maße selbst bestimmen kann.

# ☑usability

## Was bietet der SMIRROR dem Verbraucher?

Der SMIRROR bietet dem Verbraucher zwei Abläufe in einem: Während man vor dem Spiegel steht und sich die Zähne putzt, kann man aktuelle Daten, wie Wetter, Uhrzeit und Kalendereinträge, checken. Damit der Tag auch gut beginnt, gibt es passend zum Sternzeichen einen Motivationsspruch.

## Anmeldung (siehe Anhang 3)

Damit sich ein Handy auf dem Smirror anmelden kann, muss er sich zunächst mit dem eigenen Wlan-Netzwerk des Raspberry Pis verbinden. Der SSID (Name des Wlan-Netzwerks) lautet in unserem Projekt My\_AP. Anschließend gibt es zwei Wege für die Kommunikation mit dem SmartMirror:

*Über die App:* Anfangs gelangt man in ein Fenster, in der man aufgefordert wird, die IP Adresse einzugeben. Damit ist die IP-Adresse des Raspberry Pis gemeint: 169.254.3.142

*Über den Handybrowser:* In die Adresszeile entweder 169.254.3.142:3000/mobile oder 169.254.3.142:3000/mKompli eingeben. Bei der Unterseite mKompli kann der Benutzer lediglich Komplimente zum Smirror hinzufügen. Bei der Unterseite mobile hat er zusätzlich die Möglichkeit Contentboxen entweder mit Kalenderdaten, mit Datum sowie Uhrzeitdaten oder Wetterdaten hinzuzufügen.

# ☑software

*Anmerkung zu diesem Teil: Es wird nur grob auf die Programmierung eingegangen. Es würde beispielsweise den Rahmen sprengen, die Funktionen zu erklären. Was eine Funktion tut, soll aus den Kommentaren im Quellcode entnommen werden.*

## Einrichten des Raspberry Pis

Zunächst musste der Raspberry Pi so eingerichtet werden, dass er beim Starten automatisch den NodeJS Server öffnet und den Browser Chromium im Vollbildmodus startet. Zudem steht bei diesem Projekt der Monitor hochkant, weswegen auch der Bildschirmmodus verändert werden musste.

Autostart im Raspberry Pi ändern:

siehe auf die Dateien /etc/rc.local und /etc/xdg/lxsession/LXDE-pi/autostart

Bildschirmmodus ändern:

siehe /boot/config.txt

## Erläuterungen zum UML Diagramm (siehe Anhang 4)

Da eher wenig objektorientiert programmiert wurde, wurde stattdessen das UML Diagramm vereinfacht nach dem WAE (Web Application Extensions) Prinzip erstellt. So kann ein grober Überblick behalten werden, wie die Dateien miteinander im Bezug stehen. Die Javascript-Dateien sind so aufgeteilt, dass möglichst jede Datei eine spezifische Aufgabe hat. So sind zum Beispiel in der Datei conBoxTypes.js, nur die Funktionen enthalten mit der man eine ContentBox hinzufügen kann. Für die ContentBox erscheint auf dem SmartMirror ein Kasten mit den erforderlichen Informationen (wie z.B. Wetter).

## Client-Server-Kommunikation

Zur Realisierung des Servers auf dem Raspberry Pi wurde die Plattform Node.js verwendet. Für die Kommunikation mit dem Server hat jeder Client eine Javascript-Datei, in dem über Websockets mit Hilfe der Javascript-Library SockJS die Daten ausgetauscht werden. Beim SmartMirror Spiegel ist es die Datei clientSM\_WithServer\_Communication.js und bei der SmartMirror App die Datei MainAppWithServerCommunication.js. Damit die App sich mit dem Server und so auch indirekt mit dem Spiegel verbinden kann, wurde zusätzlich ein Wlan-Stick an den Raspberry Pi angesteckt, um so den Raspberry Pi als Wlan-AccessPoint einrichten zu können. Dazu wurden größtenteils die Tutorials auf den Seiten <https://frillip.com/using-your-raspberry-pi-3-as-a-wifi-access-point-with-hostapd/> und <https://www.elektronik-kompodium.de/sites/raspberry-pi/2002171.htm> befolgt. Es wird ein eigenes, offenes Wlan-Netzwerk aufgebaut, mit der sich der Benutzer über sein Handy verbinden kann. Im Fall dieses Projektes ist die IP des Raspberry Pis bei der Methode immer fest 169.254.3.142.

## Kalender

Die Kalenderdaten werden aus dem Google Calendar vom Google Konto entnommen. Aufgrund der weiten Verbreitung des Google Calendars, musste das Eintragen der Termine praktischerweise nicht mehr programmiert werden. So kann man zum Beispiel entweder über die Seite <https://calendar.google.com>, über die Google Calendar App oder über zahlreiche weitere Kalender Apps Termine eintragen werden. Damit der SmartMirror Server die Kalenderdaten abrufen kann, wird die von Google bereitgestellte API benutzt. Es werden immer die kommenden 5 Events angezeigt, die im Calendar eingetragen sind.

## Zeit

Die Zeit Box zeigt das aktuelle Datum, die Uhrzeit, und den Wochentag an. Dabei werden die verschiedenen Daten aus dem Date Objekt gezogen und vor die Variablen für die Minuten und Sekunden eine 0 gesetzt, falls der Wert kleiner als 10 ist.

## Wetter

Die Wetterdaten werden von dem Online- Wetterdienst OpenWeatherMap über eine API (application programming interface) mit zugehöriger ID bezogen. Die API ist eine Schnittstelle zur Anwendungsprogrammierung. Der benötigte API Key ist für die aktuellen Wetterdaten kostenlos zu erhalten.

Gezeigt wird die aktuelle Temperatur in Grad Celsius, eine kurze Wetterbeschreibung, der Ort und ein Icon, dass die Wettersituation visuell verdeutlicht. Durch ein Intervall werden die Daten alle drei Sekunden aktualisiert.

## Komplimente

Hinter dem Komplimente Button verbirgt sich ein Eingabefeld für den eigenen Namen und ein anderes für seinen Geburtstag. Füllt man beide Felder aus, wird auf dem Spiegel der eingegebene Name und ein Kompliment ausgegeben. Je nachdem zu welchem Sternzeichen das Geburtsdatum gehört, wird ein anderes Kompliment angezeigt. Um das Sternzeichen zu ermitteln, wird aus dem eingegebenen Datum der Tag und der Monat mit einem Substring rausgezogen und mit Hilfe von if- Abfragen ermittelt.

Z.B. ist der Geburtstag der 13.07. Dann ist das Datum 713 und somit größer als 622 und kleiner als 722. Dementsprechend ist das Sternzeichen Krebs.

# ☑problematik

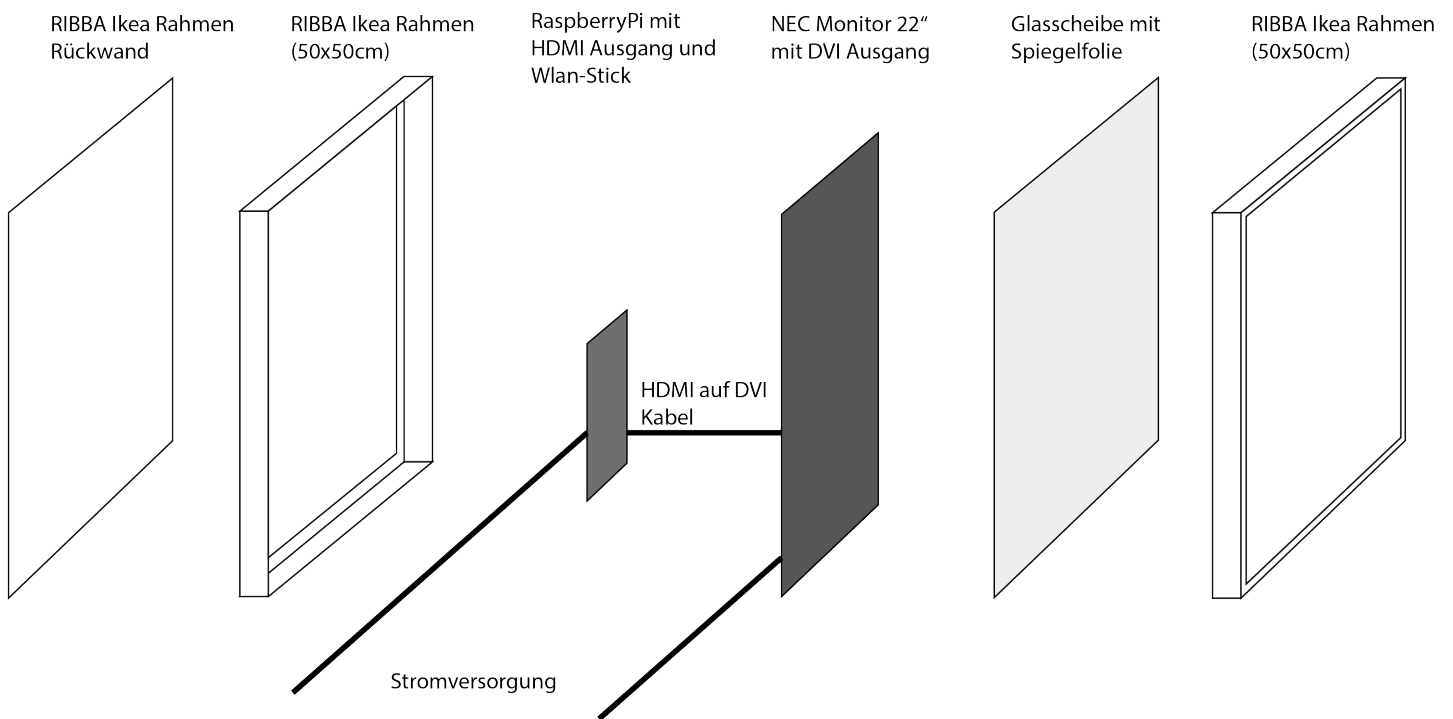
## Programmierung

Bei der Programmierung der Kommunikation des Clienten mit dem Server war eigentlich angedacht gewesen, dass die Kommunikation über MQTT und Broker und nicht über den Wlan-Access Point laufen soll. Das hätte erheblich den Programmieraufwand für den Datenaustausch zwischen dem Handy und dem Raspberry Pi erleichtert. Dadurch blieb weniger Zeit die anderen anfangs gesetzten Ziele des Projektes noch zu realisieren. Zum Beispiel fehlt, dass die Contentboxen des SmartMirror skalierbar sind oder Nachrichten auf dem Spiegel angezeigt werden können. Zudem ist es ein wenig unkomfortabler für den Benutzer, dass er sich mit seinem Handy extra im eigenen Wlan-Netzwerk des Raspberry Pis anmelden muss. Mit Blick auf die Ausstellung war dieser Weg dennoch der bessere gewesen. So haben wir bei einem Test festgestellt, dass der SmartMirror Spiegel deutlich schneller auf die Befehle vom Handy reagiert als über das uns zur Verfügung gestellte Diginet Netzwerk/Server/Wlan.

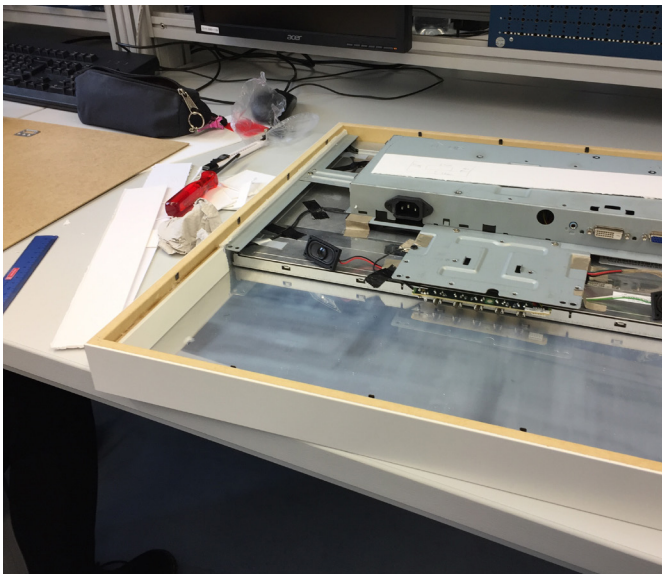
## Hardware

Die Spiegelfolie kann schnell Kratzer bekommen. Die Luftbläschen beim Aufkleben der Folie sind nur sehr schwer wegzubekommen und das Spiegelbild mit einer solchen Spiegelfolie ist nicht besonders eben. Außerdem muss man sehr darauf achten, dass der Spiegel nicht in einem sehr hellen Raum hängt, da sonst das Licht vom Monitor nicht stark genug ist um den gewünschten Effekt zu erzielen.

# ☑ anhang

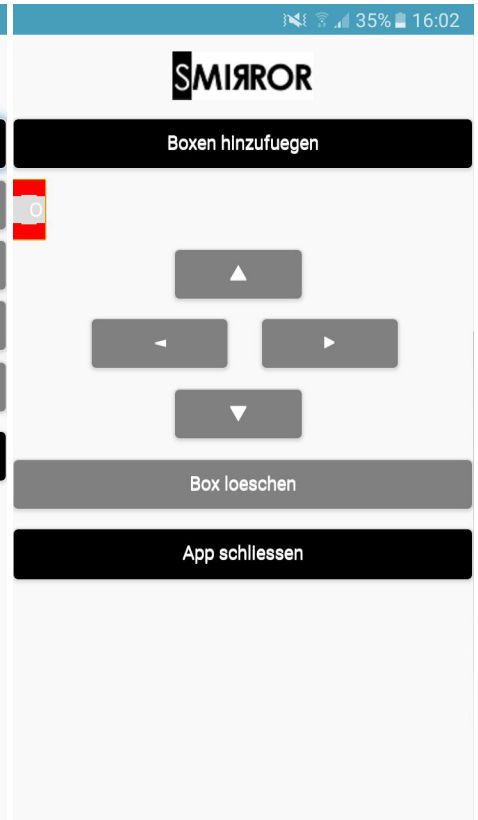
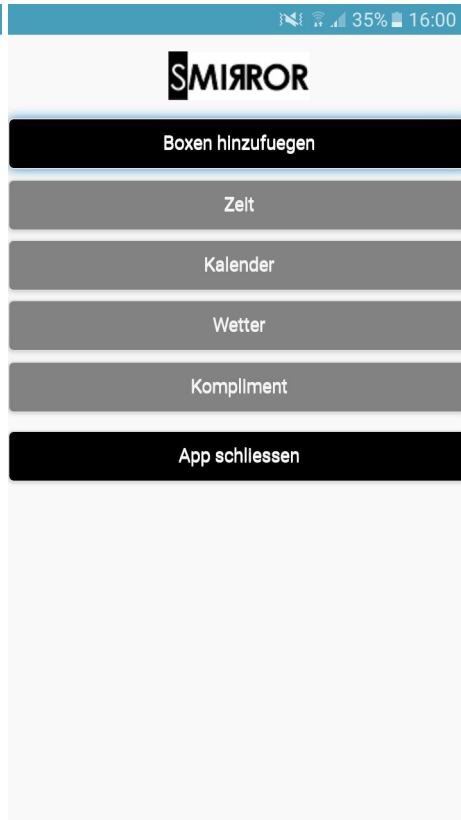


## 1 Aufbau Hardware



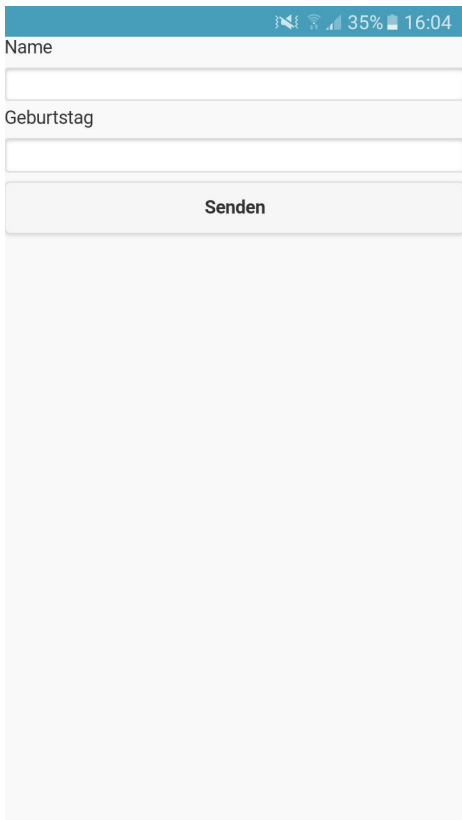
## 2 Aufbau Hardware



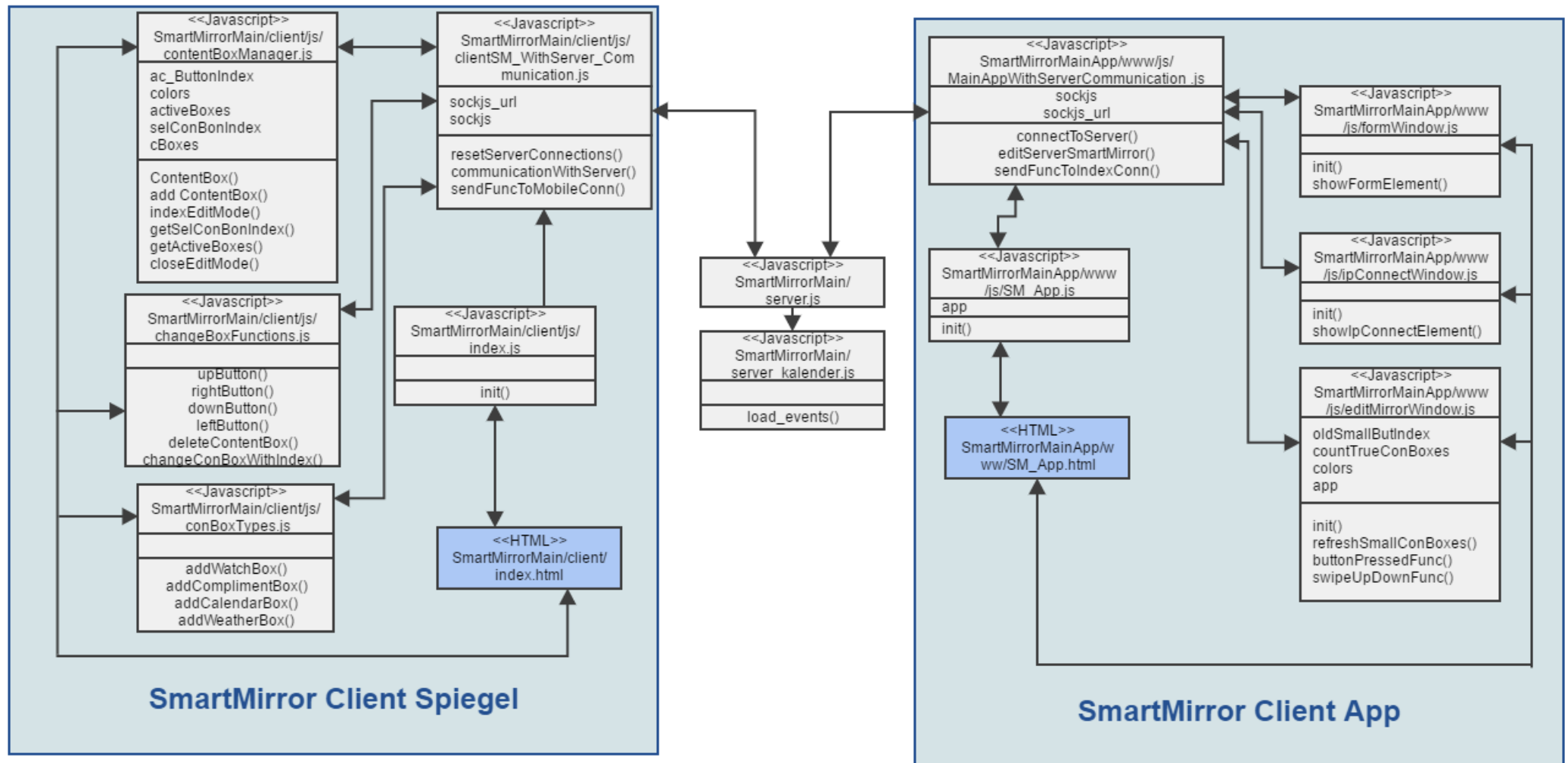


### 3 Smirror App

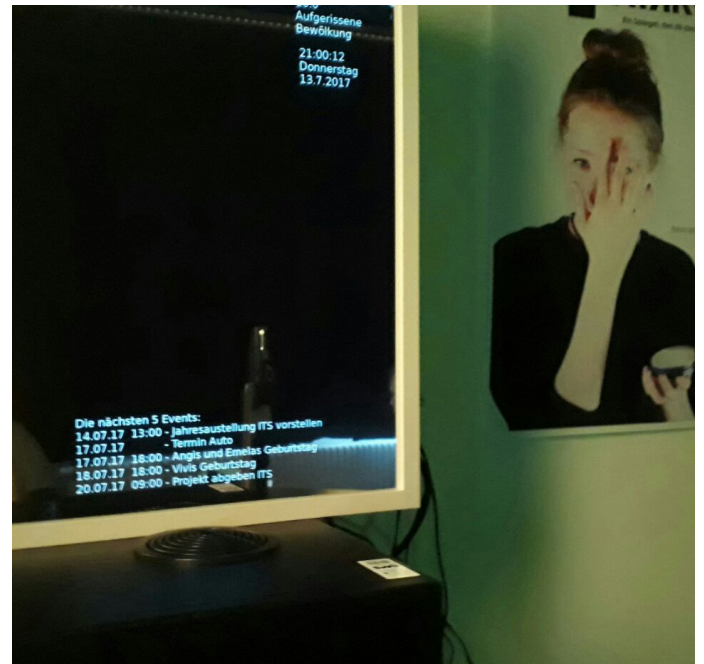
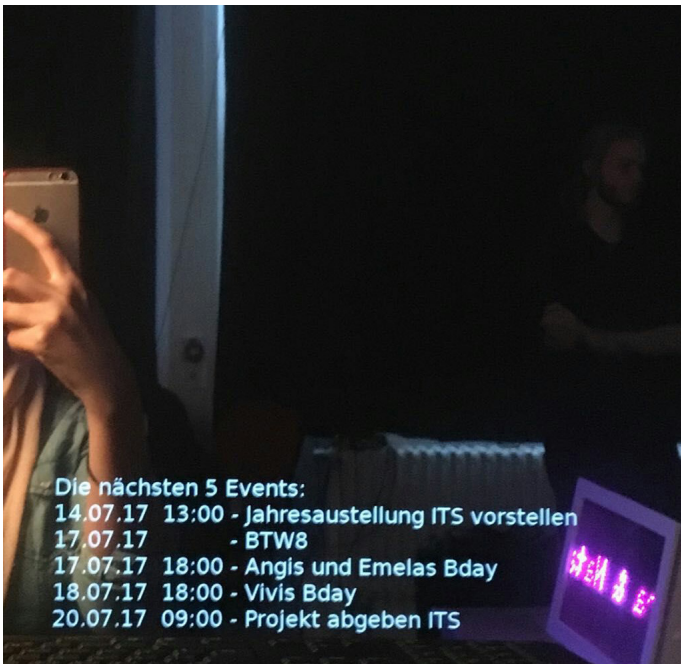
oben: Hauptapp  
unten: Unterseite mKompli







4 UML Diagramm



## 5 SMIRROR in Action

