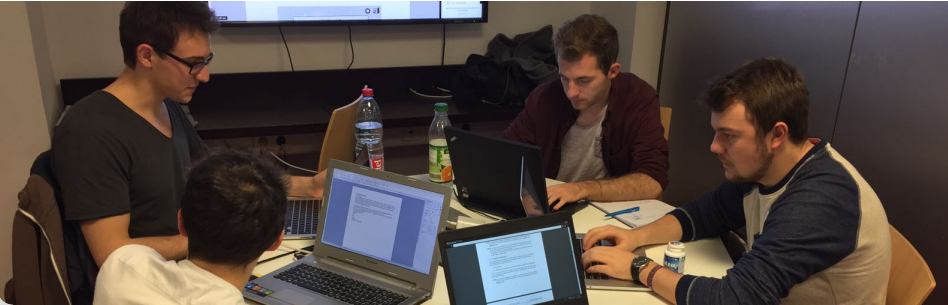


# Privacy Crash Cam: Implementierung

App, Web-Interface und Web-Dienst

Giorgio G., Christoph H., David L., Josh R., Fabian W. | 15. Februar 2017

KARLSRUHER INSTITUT FÜR TECHNOLOGIE, FRAUNHOFER INSTITUT FÜR OPTRONIK, SYSTEMTECHNIK UND BILDAUSWERTUNG





- Pflicht- (/Wunsch-) Kriterien Pflichtenheft
- Entwurf
- Einlesen und -arbeiten in Frameworks
- Auseinandersetzung mit den Entwicklungsumgebungen
- Schreiben von Demos

- Pflicht- (/Wunsch-) Kriterien Pflichtenheft
- Entwurf
- Einlesen und -arbeiten in Frameworks
- Auseinandersetzung mit den Entwicklungsumgebungen
- Schreiben von Demos

- Pflicht- (/Wunsch-) Kriterien Pflichtenheft
- Entwurf
- Einlesen und -arbeiten in Frameworks
- Auseinandersetzung mit den Entwicklungsumgebungen
- Schreiben von Demos

- Pflicht- (/Wunsch-) Kriterien Pflichtenheft
- Entwurf
- Einlesen und -arbeiten in Frameworks
- Auseinandersetzung mit den Entwicklungsumgebungen
- Schreiben von Demos

- Pflicht- (/Wunsch-) Kriterien Pflichtenheft
- Entwurf
- Einlesen und -arbeiten in Frameworks
- Auseinandersetzung mit den Entwicklungsumgebungen
- Schreiben von Demos





## ■ Planung

- Aufgaben sammeln (JIRA)
- Leitfaden bestimmen
- Implementierungsplan
- auf Styling (Layout, Javadoc, Formatter) einigen

## ■ Durchführung

- Projekte aufsetzen (IntelliJ/Android Studio)
- Implementierung
- Dokumentation
- Testen
- ausführbare Instanzen bauen

## ■ Planung

### ■ Aufgaben sammeln (JIRA)

- Leitfaden bestimmen
- Implementierungsplan
- auf Styling (Layout, Javadoc, Formatter) einigen

## ■ Durchführung

- Projekte aufsetzen (IntelliJ/Android Studio)
- Implementierung
- Dokumentation
- Testen
- ausführbare Instanzen bauen

## ■ Planung

- Aufgaben sammeln (JIRA)
- Leitfaden bestimmen
- Implementierungsplan
- auf Styling (Layout, Javadoc, Formatter) einigen

## ■ Durchführung

- Projekte aufsetzen (IntelliJ/Android Studio)
- Implementierung
- Dokumentation
- Testen
- ausführbare Instanzen bauen

## ■ Planung

- Aufgaben sammeln (JIRA)
- Leitfaden bestimmen
- Implementierungsplan
- auf Styling (Layout, Javadoc, Formatter) einigen

## ■ Durchführung

- Projekte aufsetzen (IntelliJ/Android Studio)
- Implementierung
- Dokumentation
- Testen
- ausführbare Instanzen bauen

## ■ Planung

- Aufgaben sammeln (JIRA)
- Leitfaden bestimmen
- Implementierungsplan
- auf Styling (Layout, Javadoc, Formatter) einigen

## ■ Durchführung

- Projekte aufsetzen (IntelliJ/Android Studio)
- Implementierung
- Dokumentation
- Testen
- ausführbare Instanzen bauen

## ■ Planung

- Aufgaben sammeln (JIRA)
- Leitfaden bestimmen
- Implementierungsplan
- auf Styling (Layout, Javadoc, Formatter) einigen

## ■ Durchführung

- Projekte aufsetzen (IntelliJ/Android Studio)
- Implementierung
- Dokumentation
- Testen
- ausführbare Instanzen bauen

## ■ Planung

- Aufgaben sammeln (JIRA)
- Leitfaden bestimmen
- Implementierungsplan
- auf Styling (Layout, Javadoc, Formatter) einigen

## ■ Durchführung

- Projekte aufsetzen (IntelliJ/Android Studio)
- Implementierung
- Dokumentation
- Testen
- ausführbare Instanzen bauen

## ■ Planung

- Aufgaben sammeln (JIRA)
- Leitfaden bestimmen
- Implementierungsplan
- auf Styling (Layout, Javadoc, Formatter) einigen

## ■ Durchführung

- Projekte aufsetzen (IntelliJ/Android Studio)
- Implementierung
- Dokumentation
- Testen
- ausführbare Instanzen bauen

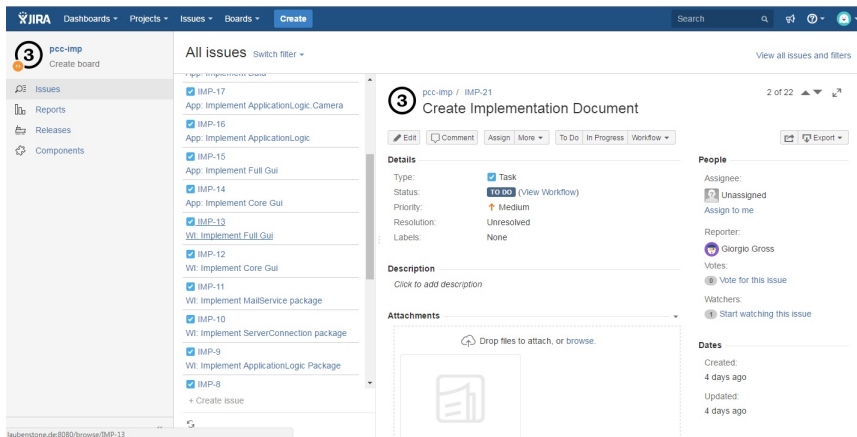


- Planung
  - Aufgaben sammeln (JIRA)
  - Leitfaden bestimmen
  - Implementierungsplan
  - auf Styling (Layout, Javadoc, Formatter) einigen
- Durchführung
  - Projekte aufsetzen (IntelliJ/Android Studio)
  - Implementierung
  - Dokumentation
  - Testen
  - ausführbare Instanzen bauen

- Planung
  - Aufgaben sammeln (JIRA)
  - Leitfaden bestimmen
  - Implementierungsplan
  - auf Styling (Layout, Javadoc, Formatter) einigen
- Durchführung
  - Projekte aufsetzen (IntelliJ/Android Studio)
  - Implementierung
  - Dokumentation
  - Testen
  - ausführbare Instanzen bauen

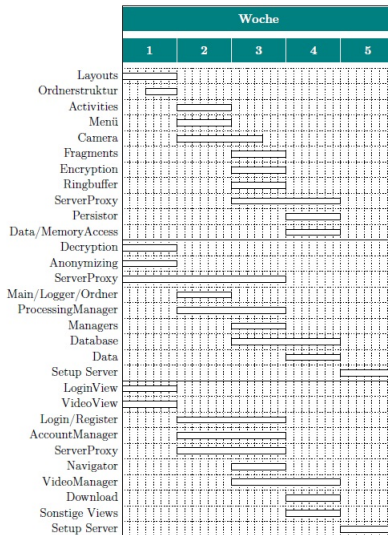
- Planung
  - Aufgaben sammeln (JIRA)
  - Leitfaden bestimmen
  - Implementierungsplan
  - auf Styling (Layout, Javadoc, Formatter) einigen
- Durchführung
  - Projekte aufsetzen (IntelliJ/Android Studio)
  - Implementierung
  - Dokumentation
  - Testen
  - ausführbare Instanzen bauen

# Aufgaben sammeln

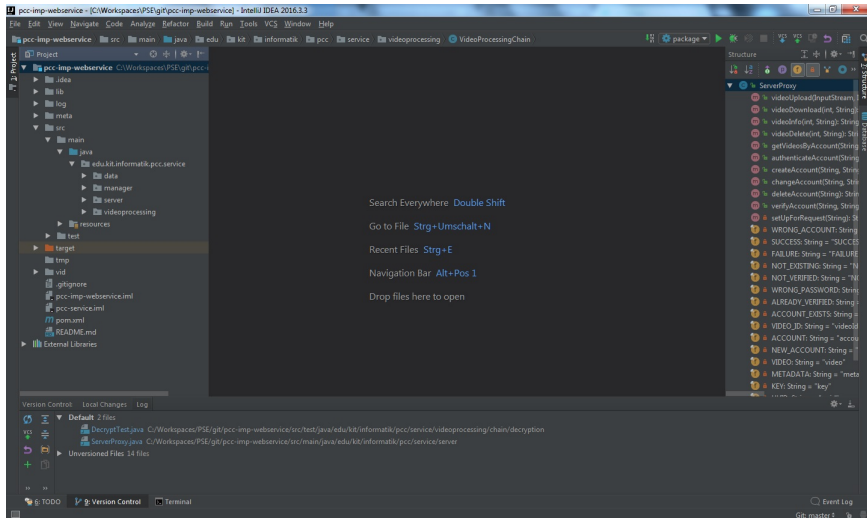


The screenshot shows the JIRA web interface. On the left, there's a sidebar with navigation links: Issues, Reports, Releases, and Components. The main area is divided into two panels. The left panel, titled 'All issues', shows a list of issues under the 'pcc-imp' project. The right panel shows the details of a specific issue, 'Create Implementation Document' (IMP-21). The issue details include a 'Details' section with fields like Type (Task), Status (To Do), Priority (Medium), Resolution (Unresolved), and Labels (None). There's also a 'Description' section with a placeholder 'Click to add description' and an 'Attachments' section with a placeholder 'Drop files to attach, or browse.'. On the right side of the issue details, there's a 'People' section showing the Assignee (Unassigned), Reporter (Giorgio Gross), and a 'Dates' section showing the Created and Updated dates (4 days ago).

# Implementierungsplan



# Projekte aufsetzen



Wir sind leider



→ Es sind Probleme aufgetreten

- Probleme mit Asynchronität des MediaRecorders

- Warten auf Fertigstellung der Videoschnipsel vor weiterer Bearbeitung

→ 1. Versuch: FileObserver auf Videoschnipseln

- Nicht alle Events wurden gefangen

→ Lösung: FileObserver auf Ordner



- Probleme mit Asynchronität des MediaRecorders
- Warten auf Fertigstellung der Videoschnipsel vor weiterer Bearbeitung

→ 1. Versuch: FileObserver auf Videoschnipseln

- Nicht alle Events wurden gefangen

→ Lösung: FileObserver auf Ordner

- Probleme mit Asynchronität des MediaRecorders
  - Warten auf Fertigstellung der Videoschnipsel vor weiterer Bearbeitung
- 1. Versuch: FileObserver auf Videoschnipseln
- Nicht alle Events wurden gefangen
- Lösung: FileObserver auf Ordner

- Probleme mit Asynchronität des MediaRecorders
  - Warten auf Fertigstellung der Videoschnipsel vor weiterer Bearbeitung
- 1. Versuch: FileObserver auf Videoschnipseln
- Nicht alle Events wurden gefangen
- Lösung: FileObserver auf Ordner

- Probleme mit Asynchronität des MediaRecorders
  - Warten auf Fertigstellung der Videoschnipsel vor weiterer Bearbeitung
- 1. Versuch: FileObserver auf Videoschnipseln
- Nicht alle Events wurden gefangen
- Lösung: FileObserver auf Ordner

- Probleme mit OpenCv, da OpenCv keine Maven-Schnittstelle besitzt.
- Zudem benötigt OpenCv native libraries.
- Lösung Java: Verwaltung über maven-dependency-plugin
- Lösung natives: manuelle Installation

- Probleme mit OpenCv, da OpenCv keine Maven-Schnittstelle besitzt.
- Zudem benötigt OpenCv native libraries.

→ Lösung Java: Verwaltung über maven-dependency-plugin

→ Lösung natives: manuelle Installation

- Probleme mit OpenCv, da OpenCv keine Maven-Schnittstelle besitzt.
  - Zudem benötigt OpenCv native libraries.
- Lösung Java: Verwaltung über maven-dependency-plugin
- Lösung natives: manuelle Installation

- Probleme mit OpenCv, da OpenCv keine Maven-Schnittstelle besitzt.
  - Zudem benötigt OpenCv native libraries.
- Lösung Java: Verwaltung über maven-dependency-plugin
- Lösung natives: manuelle Installation



- Probleme mit der Vor- bzw. Zurückfunktion der Browser.
  - Probleme mit dem FileDownloader
  - Probleme mit Layouts
  - Probleme mit Vaadin
- Das Interface verbietet Navigation zwischen Login-View und anderen Views durch Browserfunktionen.
- File Proxy, der das Video bei Anfragen lädt
- In Zukunft kein Vaadin für kleine Projekte

- Probleme mit der Vor- bzw. Zurückfunktion der Browser.
  - Probleme mit dem FileDownloader
  - Probleme mit Layouts
  - Probleme mit Vaadin
- Das Interface verbietet Navigation zwischen Login-View und anderen Views durch Browserfunktionen.
- File Proxy, der das Video bei Anfragen lädt
- In Zukunft kein Vaadin für kleine Projekte

- Probleme mit der Vor- bzw. Zurückfunktion der Browser.
  - Probleme mit dem FileDownloader
  - Probleme mit Layouts
  - Probleme mit Vaadin
- Das Interface verbietet Navigation zwischen Login-View und anderen Views durch Browserfunktionen.
- File Proxy, der das Video bei Anfragen lädt
- In Zukunft kein Vaadin für kleine Projekte

- Probleme mit der Vor- bzw. Zurückfunktion der Browser.
- Probleme mit dem FileDownloader
- Probleme mit Layouts
- Probleme mit Vaadin

- Das Interface verbietet Navigation zwischen Login-View und anderen Views durch Browserfunktionen.
- File Proxy, der das Video bei Anfragen lädt
- In Zukunft kein Vaadin für kleine Projekte

- Probleme mit der Vor- bzw. Zurückfunktion der Browser.
  - Probleme mit dem FileDownloader
  - Probleme mit Layouts
  - Probleme mit Vaadin
- Das Interface verbietet Navigation zwischen Login-View und anderen Views durch Browserfunktionen.
- File Proxy, der das Video bei Anfragen lädt
- In Zukunft kein Vaadin für kleine Projekte

- Probleme mit der Vor- bzw. Zurückfunktion der Browser.
  - Probleme mit dem FileDownloader
  - Probleme mit Layouts
  - Probleme mit Vaadin
- Das Interface verbietet Navigation zwischen Login-View und anderen Views durch Browserfunktionen.
- File Proxy, der das Video bei Anfragen lädt
- In Zukunft kein Vaadin für kleine Projekte

- Probleme mit der Vor- bzw. Zurückfunktion der Browser.
  - Probleme mit dem FileDownloader
  - Probleme mit Layouts
  - Probleme mit Vaadin
- Das Interface verbietet Navigation zwischen Login-View und anderen Views durch Browserfunktionen.
- File Proxy, der das Video bei Anfragen lädt
- In Zukunft kein Vaadin für kleine Projekte

- Bei der Implementierung sind Lücken/Ungenauigkeiten im Entwurf aufgefallen. Zudem entstanden Ideen, wie man Dinge verbessern kann.

→ es wurden einige Ergänzungen/ Änderungen nötig

- neue Methoden, z.B. `cleanUp()` in `VideoProcessingChain`
- neue Konstanten/Enumerations für Rückgabewerte oder Parameter, z.B. `VideoProcessingChain.Chain`, Konstanten im `ServerProxy`
- neue/andere Parameter, z.B. im Encryption-Modul der App
- Ergänzungen in der Datenbank, z.B. Salt für das Hashen
- Umbenennungen, z.B. `Ringbuffer` → `VideoRingBuffer`



- Bei der Implementierung sind Lücken/Ungenauigkeiten im Entwurf aufgefallen. Zudem entstanden Ideen, wie man Dinge verbessern kann.

→ es wurden einige Ergänzungen/ Änderungen nötig

- neue Methoden, z.B. `cleanUp()` in `VideoProcessingChain`
- neue Konstanten/Enumerations für Rückgabewerte oder Parameter, z.B. `VideoProcessingChain.Chain`, Konstanten im `ServerProxy`
- neue/andere Parameter, z.B. im Encryption-Modul der App
- Ergänzungen in der Datenbank, z.B. Salt für das Hashen
- Umbenennungen, z.B. `Ringbuffer` → `VideoRingBuffer`

- Bei der Implementierung sind Lücken/Ungenauigkeiten im Entwurf aufgefallen. Zudem entstanden Ideen, wie man Dinge verbessern kann.

→ es wurden einige Ergänzungen/ Änderungen nötig

- neue Methoden, z.B. `cleanUp()` in `VideoProcessingChain`
- neue Konstanten/Enumerations für Rückgabewerte oder Parameter, z.B. `VideoProcessingChain.Chain`, Konstanten im `ServerProxy`
- neue/andere Parameter, z.B. im Encryption-Modul der App
- Ergänzungen in der Datenbank, z.B. Salt für das Hashen
- Umbenennungen, z.B. `Ringbuffer` → `VideoRingBuffer`

- Bei der Implementierung sind Lücken/Ungenauigkeiten im Entwurf aufgefallen. Zudem entstanden Ideen, wie man Dinge verbessern kann.

→ es wurden einige Ergänzungen/ Änderungen nötig

- neue Methoden, z.B. `cleanUp()` in `VideoProcessingChain`
- neue Konstanten/Enumerations für Rückgabewerte oder Parameter, z.B. `VideoProcessingChain.Chain`, Konstanten im `ServerProxy`
- neue/andere Parameter, z.B. im Encryption-Modul der App
- Ergänzungen in der Datenbank, z.B. Salt für das Hashen
- Umbenennungen, z.B. `Ringbuffer` → `VideoRingBuffer`

- Bei der Implementierung sind Lücken/Ungenauigkeiten im Entwurf aufgefallen. Zudem entstanden Ideen, wie man Dinge verbessern kann.

→ es wurden einige Ergänzungen/ Änderungen nötig

- neue Methoden, z.B. `cleanUp()` in `VideoProcessingChain`
- neue Konstanten/Enumerations für Rückgabewerte oder Parameter, z.B. `VideoProcessingChain.Chain`, Konstanten im `ServerProxy`
- neue/andere Parameter, z.B. im Encryption-Modul der App
- Ergänzungen in der Datenbank, z.B. Salt für das Hashen
- Umbenennungen, z.B. `Ringbuffer` → `VideoRingBuffer`

- Bei der Implementierung sind Lücken/Ungenauigkeiten im Entwurf aufgefallen. Zudem entstanden Ideen, wie man Dinge verbessern kann.
- es wurden einige Ergänzungen/ Änderungen nötig
- neue Methoden, z.B. `cleanUp()` in `VideoProcessingChain`
  - neue Konstanten/Enumerations für Rückgabewerte oder Parameter, z.B. `VideoProcessingChain.Chain`, Konstanten im `ServerProxy`
  - neue/andere Parameter, z.B. im Encryption-Modul der App
  - Ergänzungen in der Datenbank, z.B. Salt für das Hashen
  - Umbenennungen, z.B. `Ringbuffer` → `VideoRingBuffer`

- Einige Dinge liefen sogar schneller als ursprünglich geplant
  - App-Layout
  - App-Camera
  - Web-Service VideoProcessing-Modul
  - Web-Service Server setup
  - Web-Interface Video download



- Einige Dinge liefen sogar schneller als ursprünglich geplant
  - App-Layout
  - App-Camera
  - Web-Service VideoProcessing-Modul
  - Web-Service Server setup
  - Web-Interface Video download



- Einige Dinge liefen sogar schneller als ursprünglich geplant
  - App-Layout
  - App-Camera
  - Web-Service VideoProcessing-Modul
  - Web-Service Server setup
  - Web-Interface Video download





- Einige Dinge liefen sogar schneller als ursprünglich geplant
  - App-Layout
  - App-Camera
  - Web-Service VideoProcessing-Modul
  - Web-Service Server setup
  - Web-Interface Video download



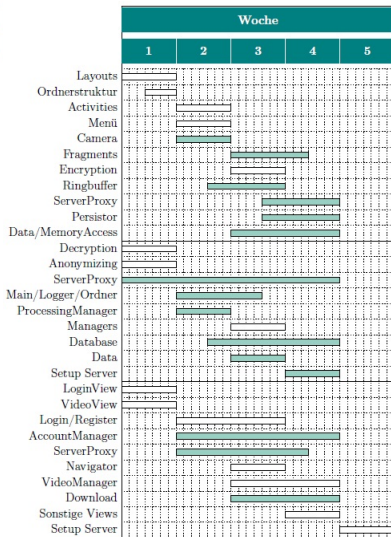
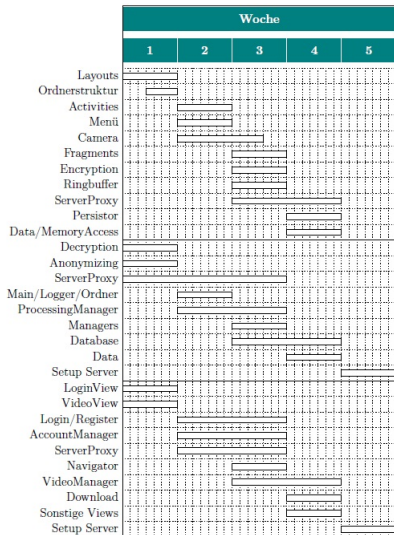
- Einige Dinge liefen sogar schneller als ursprünglich geplant
  - App-Layout
  - App-Camera
  - Web-Service VideoProcessing-Modul
  - Web-Service Server setup
  - Web-Interface Video download



- Einige Dinge liefen sogar schneller als ursprünglich geplant
  - App-Layout
  - App-Camera
  - Web-Service VideoProcessing-Modul
  - Web-Service Server setup
  - Web-Interface Video download



# Tatsächlicher Implementierungsplan



Ausgangssituation

○○

Implementierung

○○○○○

Probleme

○○○○○

Ergebnis

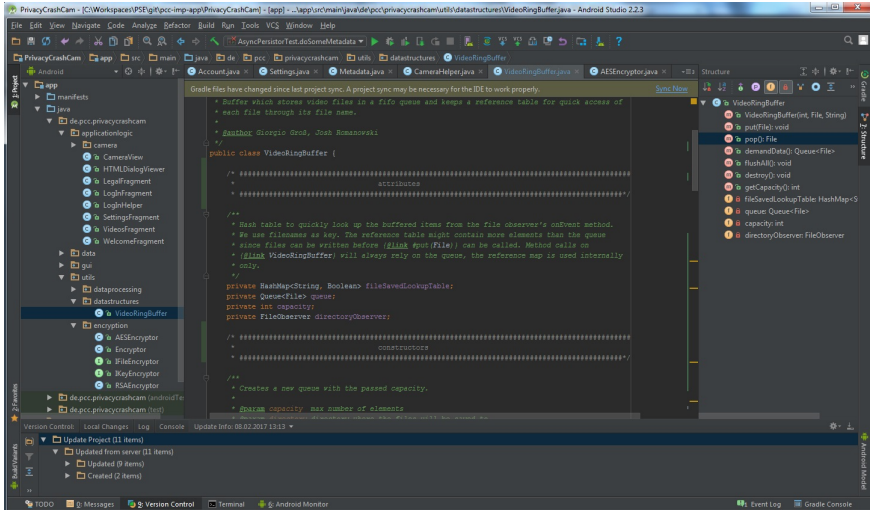
●○○○○

Organisation

○○

Gruppenarbeit

Quellen



```

/* Buffer which stores video files in a fifo queue and keeps a reference table for quick access of
 * each file through its file name.
 *
 * @author Giorgio Groß, Josh Romanowski
 */
public class VideoRingBuffer {

    /* =====
     * attributes
     * ===== */

    /**
     * Hash table to quickly look up the buffered items from the file observer's onEvent method.
     * We use filenames as key. The reference table might contain more elements than the queue
     * since files can be written before (link #put(File)) can be called. Method calls on
     * (link VideoRingBuffer) will always rely on the queue, the reference map is used internally
     * only.
     */
    private HashMap<String, Boolean> fileSavedLookupTable;
    private Queue<File> queue;
    private int capacity;
    private FileObserver directoryObserver;

    /* =====
     * constructors
     * ===== */

    /**
     * Creates a new queue with the passed capacity.
     *
     * @param capacity max number of elements
     */
    public VideoRingBuffer(int capacity) {
        this.capacity = capacity;
        queue = new Queue<File>(capacity);
        fileSavedLookupTable = new HashMap<String, Boolean>();
        directoryObserver = new FileObserver(".") {
            @Override
            public void onEvent(String path, boolean isDirectory) {
                // TODO: Implement file observer logic
            }
        };
        directoryObserver.startWatching();
    }

    /**
     * Adds a file to the queue and the lookup table.
     *
     * @param file File to be added
     */
    public void put(File file) {
        // TODO: Implement put logic
    }

    /**
     * Removes a file from the queue and the lookup table.
     *
     * @param file File to be removed
     */
    public void pop() {
        // TODO: Implement pop logic
    }

    /**
     * Returns the file at the given index.
     *
     * @param index Index of the file
     * @return File at the given index
     */
    public File demandData(int index) {
        // TODO: Implement demandData logic
    }

    /**
     * Flushes the queue.
     */
    public void flushAll() {
        // TODO: Implement flushAll logic
    }

    /**
     * Destroys the buffer.
     */
    public void destroy() {
        // TODO: Implement destroy logic
    }

    /**
     * Returns the current capacity.
     *
     * @return Current capacity
     */
    public int getCapacity() {
        return capacity;
    }







    /**
     * Returns the lookup table.
     *
     * @return Lookup table
     */
    public HashMap<String, Boolean> getFileSavedLookupTable() {
        return fileSavedLookupTable;
    }

    /**
     * Returns the queue.
     *
     * @return Queue
     */
    public Queue<File> getQueue() {
        return queue;
    }

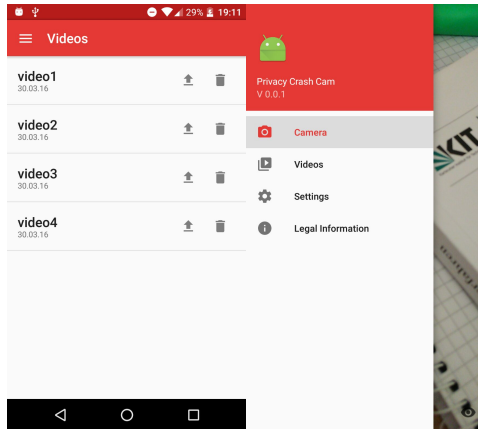
    /**
     * Returns the directory observer.
     *
     * @return Directory observer
     */
    public FileObserver getDirectoryObserver() {
        return directoryObserver;
    }
}

```

# Oberfläche (1)

Menu	Video Name	Video herunterladen	Meta-Info anzeigen	Video löschen
👤 josh.romanowski@gmail.com  Account Ansicht <b>Video Ansicht</b> Datenschutz Impressum Logout	1_myTestVideo_anonymVid			
	1_anotherTestVideo_anonymVid			

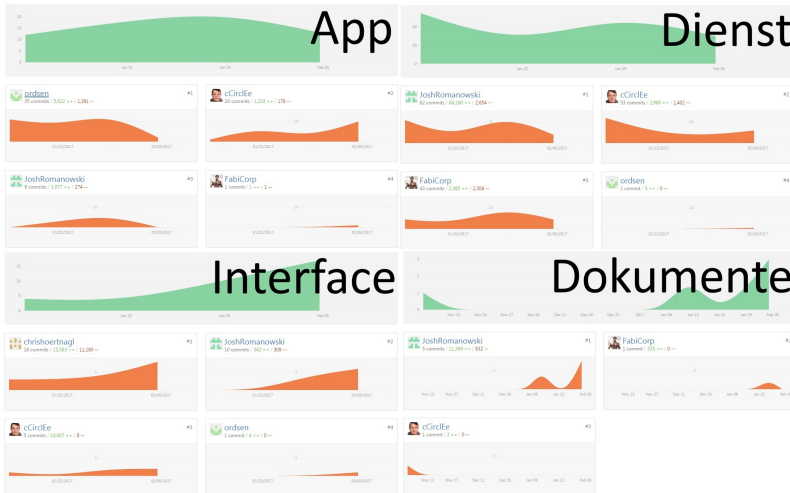
# Oberfläche (2)







- Christoph H.
  - Web-Interface
  - Präsentation
- David L.
  - Web-Dienst
  - App (Data)
- Fabian W.
  - Web-Dienst
  - ServerProxy App/Interface
- Giorgio G.
  - App
- Josh R.
  - Web-Dienst
  - App (Utility)





- <http://de.clipartlogo.com/free/rain-cloud-cartoon.html>
- <https://www.spreadshirt.de/sprechblase+-+comic+t-shirts>
- [https://de.wikipedia.org/wiki/Mona\\_Lisa](https://de.wikipedia.org/wiki/Mona_Lisa)
- <https://nebadonia.wordpress.com/2015/09/17/erkenne-die-matrix-und-wie-komme-ich-da-raus/>
- <http://ausmalbilder.kim/category/malvorlagen/zauberer/>
- <https://www.gratis-malvorlagen.de/phantasie/wahrsagerin-3/>
- <https://www.spreadshirt.de/daumen+geschenke>