

Anforderungsanalyse und Modellierung:

Programm zur Anonymisierung von Gesichtern

Grundlegende Idee: Eine Applikation, die in der Lage ist, hochgeladene Bilder so zu bearbeiten, dass die abgebildeten Personen anonymisiert werden, also nicht mehr identifizierbar sind. Die anonymisierte Datei soll dann zurückgegeben werden.

Lastenheft

Ziel	Priorität
Generell	
<u>Website lässt sich auf verschiedenen Geräten und Betriebssystemen öffnen</u>	Muss
Auf lokalem Rechner	Muss
Vom Internet aus zugreifbar	Kann
Light-Mode und Dark-Mode einstellbar	Kann
Startzustand: keine eingegebenen Dateien, weder bei Bildeingabe noch Bildausgabe. Bei neu-laden der Seite sind Bilder immer noch vorhanden	Soll
Titel	Soll
Impressum	Kann
Dateieneingabe	
<u>Nutzer kann Bild im PNG-Format hochladen</u>	Muss
Unterstützung von anderen häufigen Formaten: JPG, JPEG	Kann
Anzeige der unterstützten Formate	Soll
<u>Nutzer kann mehrere einzelne Bilder (der passenden Formate) hochladen</u>	Soll
Hochgeladenes Bild wird in einer Vorschau angezeigt	Soll
Bei mehreren hochgeladenen Bildern kann die Vorschau mehrere Bilder synchron anzeigen	Soll
Was zensiert wird, wird mithilfe von Bounding-boxes in der Vorschau angegeben	Kann
<u>Hochgeladene Datei kann wieder entfernt werden</u>	Soll

Bei mehreren hochgeladenen Bildern: Es können sowohl einzelne Bilder als auch alle Bilder gleichzeitig wieder entfernt werden	Soll
Anonymisierungsprozess	
<u>Art der Anonymisierung kann ausgewählt werden: Verpixelung (Standard)</u>	Soll
Schwarzer Balken	Soll
Blur	Kann
<u>Objekt der Anonymisierung kann ausgewählt werden: Gesicht (Standard)</u>	Muss (Standard)
Logos	Kann
Schuhe	Kann
Haare	Kann
Anonymisierungsprozess des Bildes kann manuell gestartet werden	Muss
Erweiterung bei mehreren hochgeladenen Bildern: der Anonymisierungsprozess für alle Objekte wird gleichzeitig gestartet	Soll
Bild wird entsprechend der Auswahlen verändert	Muss
Hinweis auf gestarteten Anonymisierungsprozess (als Pop-Up oder Text)	Kann
Bildausgabe	
<u>Anonymisiertes Bild kann heruntergeladen werden als PNG</u>	Muss
Bei mehreren hochgeladenen Bildern: Alle Bilder werden in einem Zip-Ordner zurückgegeben	Kann
<u>Anonymisiertes Bild wird in Vorschau angezeigt</u>	Soll
Bei mehreren hochgeladenen Objekten: Mehrere Vorschauen werden gleichzeitig angezeigt	Soll
<u>Titel des anonymisierten Bildes wird geändert: alterName_pixelated.endung</u>	Kann
Bei Ordner: gesamter Ordner wird "Anonymisiert" genannt	Kann
Bei wiederholter Ausführung des Programms, ersetzen die neuen anonymisierten Dateien die älteren. Diese werden nicht mehr angezeigt.	Soll
<u>Bei ungültigen Dateiformaten: Hinweis darauf, Keine Rückgabe</u>	Soll
Möglicher Hinweis auf oder Integrierung von ffmpeg	Kann
Bei keinen gefundenen Erkennungsmerkmalen auf einer Datei: Hinweismeldung	Kann

Mögliche Erweiterung I	
<u>Nutzbarkeit der Software als Command-Line Tool</u> Bilddateien können über Pfadangaben als Parameter übergeben werden Auswahl der Anonymisierungsart (z. B. Verpixelung, schwarzer Balken, Blur) erfolgt über Kommandozeilenparameter Auswahl des zu anonymisierenden Objekts (z. B. Gesicht, Logos, Schuhe, Haare) erfolgt über Kommandozeilenparameter Die anonymisierte Ausgabedatei wird im gleichen oder einem definierten Ausgabeverzeichnis gespeichert Fehlerhafte Eingaben (z. B. falsches Dateiformat, ungültiger Pfad) werden über eine Konsolenmeldung ausgegeben	Kann
Mögliche Erweiterung II	
<u>Nutzer kann Videos hochladen und anonymisieren</u> Nutzer kann Videos im Format MP4 hochladen Die Videoverarbeitung erfolgt auf Basis derselben Nutzereinstellungen wie bei Bildern Das Video wird Frame für Frame verarbeitet Die ausgewählte Art der Anonymisierung (z. B. Verpixelung) wird auf alle relevanten Frames angewendet Die ausgewählten Objekte der Anonymisierung (z. B. Gesicht, Logos) werden im gesamten Video erkannt Das anonymisierte Video wird dem Nutzer als MP4-Datei zurückgegeben Laufzeitorientierung	Kann

Vorgehen

1. Anforderungsanalyse und Modellierung erfolgen sequentiell, immer wieder mit Besprechungen und Überarbeitungen.
2. Implementierung erfolgt zuerst sequentiell, bis alle muss- (und die wichtigsten soll-) - Features implementiert wurden, so dass eine rudimentäre Version des Programms so schnell wie möglich vorhanden ist. Beide Teams führen hierbei durchgehend Testungen durch.

3. Testung der Zusammenarbeit von Frontend und Backend
4. Inkrementelle Entwicklung der Kann-Features
5. Weiteres, finales, Testen und beheben letzter Fehler

Aufgabenverteilung

Jakob: UI

Anastasia: Backend

Ludwig: Backend

Elisabeth: Koordination, Kommunikation UI und Backend

Christoph: Git, Koordination, Flexibel Unterstützung wo benötigt

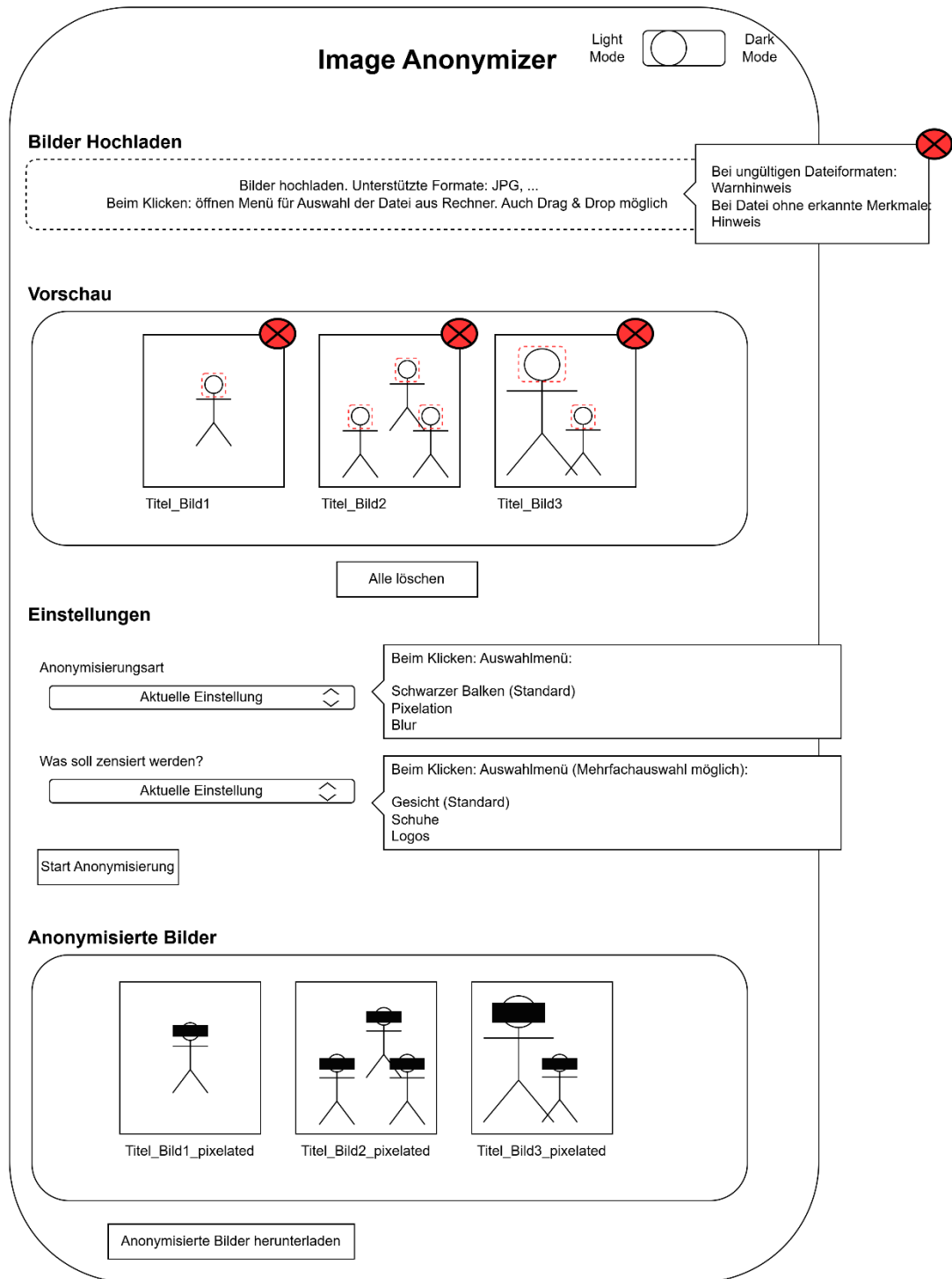
Zeitplan

Tag	Ziel
21. 01.	Erste Organisation und Abstimmung
23. 01. (Homeschooling)	Arbeit an Prototypen, an Anforderungsanalyse und Modellierung
26. 01. (Homeschooling)	
28. 01.	
30. 01. (Halber Block)	
01. 02. (Sonntag Beginn der Ferien)	Abgabe Anforderungsanalyse und Modellierung
09. 02.	Implementierung von Muss-Feature
11. 02.	
13. 02.	
18. 02. (Klausurvorbereitung)	
20. 02	Alle Muss- Features fertig
23. 02	Klausur
25. 02.	Möglichst alle Soll-Feature implementieren, ggf. Kann-Feature implementieren
27. 02.	
04. 03.	
06. 03.	
09. 03.	Abgabe der Projekte am Ende des Blockes
11. 03. (Homeschooling)	
13. 03.	Vorstellung der Projekte

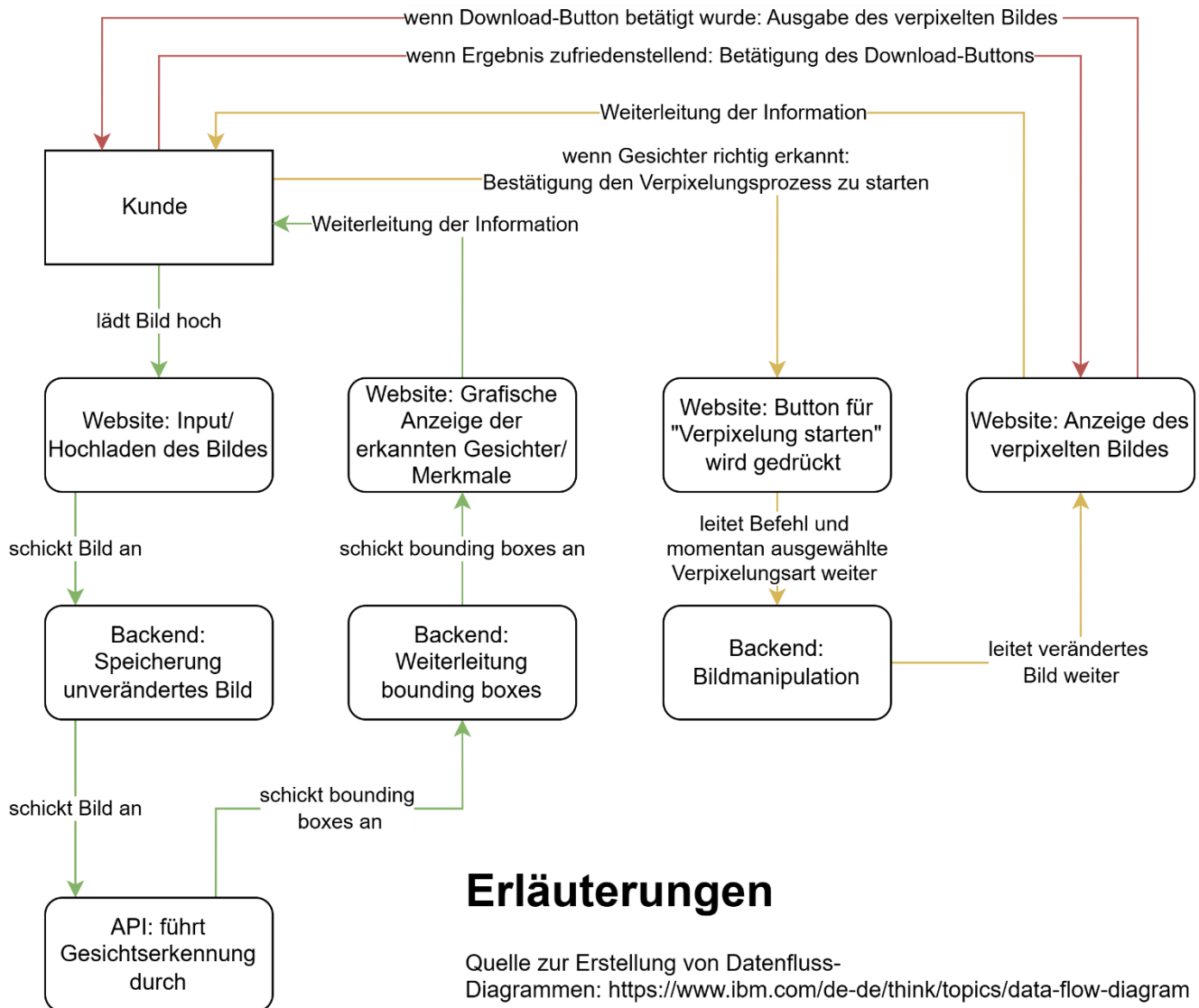
Modellierung

Modell 1: UI

Stand des Modells: alle muss, soll und kann- Features dargestellt außer den Erweiterungen als Comman-Line-Tool und für Videos



Modell 2: Datenfluss



Erläuterungen

Quelle zur Erstellung von Datenfluss-Diagrammen: <https://www.ibm.com/de-de/think/topics/data-flow-diagram>
 Externe Entitäten: Rechtecke
 Prozesse: abgerundeten Rechtecken
 Datenflüsse: Pfeile

Zur Reihenfolge:
 Grün: -> Gelb -> Rot

Die in der Anforderungsanalyse beschriebene Funktion "Auswahl des Objektes der Anonymisierung" (bzw. die Übermittlung dieser Information an das Backend) wird im Datenflussmodell ignoriert. Dies liegt daran, dass diese Funktion unter "kann" eingestuft wird und ihre Umsetzung vermutlich nicht mehr geschaffen wird, da dafür auch eine neue API herausgesucht werden muss, die die Erkennung anderer Objekte wie Schuhe oder Logos gewährleistet. Vorerst beschränken wir uns jedoch auf die Gesichtserkennung und -anonymisierung.

Im Sinne der Übersichtlichkeit, der Vereinfachung und der Darstellung der Kernfunktionen wurde diese Erweiterung, die Auswahl des zu anonymisierten Objektes, daher weggelassen.

Modell 3: Backend-Struktur

