Proof-of-Concepts

Top Priority

* Upload eines Bildes
  + **Ablauf**
    - Aufruf einer Upload Funktion
    - Abfrage nach Metadaten (Datum, Adresse, Fotograf, Rechteinhaber – nur Datum und Rechteinhaber mandatory)
    - Abfrage nach Bild mit Unterstützung mehrerer Bildformate
    - Erzeugen eines Bildobjekts mit Bild und Metadaten
    - Abspeichern des Bildobjekts
    - Ansicht des Bildobjekts erzeugen
  + **Exit-Kriterien**
    - Funktion konnte aufgerufen werden und Eingaben wurden erfolgreich und korrekt übernommen.
    - Bilddaten konnten erfolgreich geladen werden.
    - Neues Bildobjekt wurde erfolgreich erzeugt und abgespeichert.
    - Ansicht wurde erfolgreich erzeugt.
  + **Fail-Kriterien**
    - Ungültige Eingabe wurde übernommen (Constraints für Datum- und Adressformat)
    - Bilddaten konnten gar nicht geladen werden
    - Nur begrenzte Bildformate konnten unterstützt werden
    - Neues Bildobjekt konnte nicht erzeugt werden
    - Bildobjekt konnte nicht abgespeichert werden
    - Ansicht wurde nicht erzeugt
  + **Fallback**
    - Bild wird ohne Metadaten gespeichert
    - Es werden begrenzte Bildformate gefordert
* Pub/Sub Funktionalität Trennen
  + **Ablauf**
    - Login als User A
    - Aufruf einer Funktion zum Abbonnement zu einer Stadt
    - Wechsel auf User B aus dieser Stadt
    - Upload eines Bildes
    - Wechsel zu User A
    - Ansicht mit Benachrichtigung, dass neuer Content erstellt wurde
    - Aufruf und Ansicht einer Timeline mit zuletzt gepostetem Content von allen abbonnierten Usern/Städten
    - Aufruf und Ansicht des Bildes
  + **Exit-Kriterien**
    - Jeweiliger Login war erfolgreich
    - Funktion zum abbonnieren konnte erfolgreich aufgerufen werden
    - Upload des Bildes war erfolgreich
    - Benachrichtigung, dass es einen neuen Post gab, wurde erfolgreich vermittelt.
    - Timeline wurde erfolgreich aufgerufen und korrekt mit dem neuem Bild angezeigt.
    - Bild wurde erfolgreich aufgerufen und korrekt angezeigt
  + **Fail-Kriterien**
    - Logins waren nicht erfolgreich
    - Funktion zum Abbonnement konnte nicht aufgerufen werden
    - Benachrichtigung, dass neuer Content gepostet wurde wurde nicht vermittelt
    - Timeline wurde gar nicht angezeigt
    - Hochgeladenes Bild wurde nicht auf der Timeline angezeigt
    - Bild konnte nicht aufgerufen werden
  + **Fallback**
    - Verzicht auf Pub/Sub Funktionalität

High Priority

* Login
  + **Ablauf**
    - Abfrage nach Username/Email - Passwort Dataset
    - Systemzugriff auf Registry mit allen Username/Email-PW Datasets
    - Vergleich mit Registry
    - Bei gefundener Übereinstimmung -> Ansicht des Heimatorts
    - Bei fehlender Übereinstimmung -> Ansicht mit Fehlermeldung
  + **Exit-Kriterien**
    - Eingabe wude erfolgreich und korrekt übernommen
    - Zugriff auf Registry war erfolgreich
    - Vergleich lieferte korrektes Ergebnis
    - Richtige Ansicht wurde erzeugt
  + **Fail-Kriterien**
    - Ungültige Eingabe wurde übernommen (Email constraints)
    - Zugriff auf Registry war nicht erfolgreich
    - Vergleich lieferte falsches Ergebnis
    - Vergleich lieferte kein Ergebnis
    - Falsche Ansicht wurde erzeugt
  + **Fallback**
    - Ein fester User account wird im Prototypen vorgelegt
    - Verzicht auf externe Registry, User Accounts werden fest in den Prototypen geschrieben
    - Upload eines Bildes
* Demonstration von Rechten Trennen
  + **Ablauf**
    - Login als Moderator
    - Zuweisung von Moderator Rechten durch Session Flags
    - Aufruf einer Funktion zum blockieren eines Users
    - Veränderung des Block-Status eines anderes Users
    - Logout als Moderator
    - Login als blockierter User
    - Prüfen des Block-Status
  + **Exit-Kriterien**
    - Login war erfolgreich
    - Moderator Rechte wurden erfolgreich zugewisen
    - Block-Funktion wurde erfolgreich aufgerufen
    - Login als blockierter User war erfolgreich
    - Ansicht mit „Du wurdest blockiert“ wurde erzeugt
  + **Fail-Kriterien**
    - Login war nicht erfolgreich
    - Moderator Rechte wurden nicht erfolgreich zugewiesen
    - Block-Funktion konnte nicht aufgerufen werden
    - Anderer User konnte nicht blockiert werden
    - Falsche Ansicht wurde erzeugt
  + **Fallback**
    - Eigene Clients für unterschiedlichen System Rollen schreiben. Funktionen die entsprechende Rechte benötigen sind exclusiv auf diesen Clients verfügbar.
    - Blockierten User komplett aus der Registry löschen.

Medium Priority

* Registrierung u. Verschlüsselung Trennen
  + **Ablauf**
    - Abfrage nach erforderlichen Userdaten (Username, Email, Heimatort, Passwort)
    - Speichern des neuen User Accounts
    - Verschlüsselung der Daten
    - Prüfen der Verschlüsselung
  + **Exit-Kriterien**
    - Eingaben werden erfolgreich und korrekt übernommen
    - create Funktion für neuen User Account ist erfolgreich & korrekt
    - User Account wurde erfolgreich abgepseichert
    - Daten wurden erfolgreich verschlüsselt
  + **Fail-Kriterien**
    - Ungültige Eingabe wurde übernommen (Constraints für email, Heimatort)
    - Gültige Eingabe ist nicht erfolgreich
    - User Account wurde nicht korrekt erzeugt
    - User Account wurde nicht abgespeichert
    - Daten wurden nicht verschlüsselt
  + **Fallback**
    - Verzicht auf Registrierung neuer Accounts, bestimmte Anzahl an Accounts wird fest erzeugt
    - Verzicht auf User Daten, die Verschlüsselung benötigen

Low Priority

* Chat mit neuem Freund Trennen
  + **Ablauf**
    - Login als User A
    - Aufruf einer Funktion zur Freundschaftsanfrage an User B
    - Wechsel auf User B
    - Ansicht mit Notification der Freundschaftsanfrage
    - Akzeptieren der Freundschaftsanfrage
    - Erzeugen einer Verbindung zwischen den Usern
    - Erzeugen eines Chatobjekts als Teil dieser Verbindung
    - Aufruf des Chats
    - Ansicht des Chats zwischen den Usern (leer)
    - Abfrage nach Texteingabe
    - Abspeichern des Textes im Chat
    - Ansicht des Chats mit dem eingegebenem Text
    - Wechsel auf User A
    - Ansicht mit Notification der neuen Chat Nachricht
    - Aufruf des Chats
    - Ansicht des Chats
  + **Exit-Kriterien**
    - Jeweiliger Login war erfolgreich
    - Freundschaftsanfrage wurde erfolgreich vermittelt
    - Verbindung und Chatobjekt wurden erfolgreich erzeugt
    - Aufrufe des Chats waren erfoglreich
    - Eingabe wurde erfolgreich und korrekt übernommen
    - Text konnte erfolgreich abgespeichert werden
    - Ansichten wurden erfolgreich erzeugt
  + **Fail-Kriterien**
    - Login war nicht erfolgreich
    - Freundschaftanfrage konnte nicht vermittelt werden
    - Es wurde keine Verbindung erzeugt
    - Chat konnte nicht aufgerufen werden
    - Eingabe wurde falsch übernommen
    - Text wurde nicht abgespeichert
    - Ansichten wurden nicht erzeugt
  + **Fallback**
    - Persönlicher Chat erfolgt über eigenes Subforum pro User

//Highly likely to be scrapped and Fallback to be used

* Zugriff auf Datenbank
  + Ablauf
    - Einrichten einer Datenbank
    - Zugriff auf einen Eintrag in der DB
    - Veränderung eines Wertes aus diesem Eintrag
    - Speichern des veränderten Eintrags in der DB
    - Erneuter Zugriff auf Eintrag zur Prüfung
  + Exit-Kriterien
    - Auslesen des Wertes ist erfolgreich
    - Veränderung des Wertes in der DB ist erfolgreich
  + Fail-Kriterien
    - Zugriff ist nicht erfolgreich
    - Wert kann aus Eintrag nicht ausgelesen werden
    - Speichern ist nicht erfolgreich
  + Fallback
    - Lokale persistente Datenspeicherung (z.B. über JSON)

Feedback:

PoCs kleiner gestalten: Beispiel Pub/sub: einen für Pub anderen für Sub