Proof-of-Concepts

Top Priority

* Upload eines Bildes
  + **Ablauf**
    - Aufruf einer Upload Funktion
    - Auslesen des Bild mit Unterstützung gängiger Bildformate
    - Automatisches Auslesen der erforderlichen Metadaten
    - Erzeugen eines Bildobjekts mit Bild und Metadaten
    - Abspeichern des Bildobjekts
  + **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.
  + **Fail-Kriterien**
    - Bilddaten konnten gar nicht geladen werden
    - Nur begrenzte Bildformate konnten unterstützt werden
    - Neues Bildobjekt konnte nicht erzeugt werden
    - Bildobjekt konnte nicht abgespeichert werden
  + **Fallback**
    - Aus Eindeutige Fehlermeldung wenn technische Fehler bei der Ausführung der Upload Funktion erfolgen
    - Bild wird ohne Metadaten gespeichert
    - Es werden begrenzte Bildformate gefordert
* Aufruf eines gespeicherten Bildes
  + **Ablauf**
    - Bild wird aus persistentem Speicher geladen
    - Ansicht des Bildobjekts erzeugen
  + **Exit-Kriterien**
    - Bild wurde erfolgreich aus persistentem Speicher geladen
    - Ansicht wurde erfolgreich erzeugt.
  + **Fail-Kriterien**
    - Bild wurde nicht aus persistentem Speicher geladen
    - Ansicht wurde nicht erzeugt
  + **Fallback**
    - Eindeutige Fehlermeldung anzeigen
    - Aufforderung zum neu laden mitteilen
* Pub/Sub Funktionalität - subscribe
  + **Ablauf**
    - Aufruf einer Funktion zum Abonnement zu Stadt A
    - Aufruf Abonnenten der Stadt A
    - Abgleich nach User Name A /Kontrolle im System ob User A subscribed in Stadt A
  + **Exit-Kriterien**
    - Funktion zum Abonnieren konnte erfolgreich aufgerufen werden
    - Upload des Bildes war erfolgreich
    - Abgleich auf User A in Abonnenten Stadt A ist erfolgreich/ Kontrolle im System, ob User A Subcribe Stadt A ist erfolgreich
  + **Fail-Kriterien**
    - Funktion zum Abonnement konnte nicht aufgerufen werden
  + **Fallback**
    - Fehlermeldung
* Pub/Sub Funktionalität - publish
  + **Ablauf**
    - Upload eines Bildes …
    - Kontrolle System Benachrichtigung an Subscriber von Stadt A
  + **Exit-Kriterien**
    - Funktion zum Abonnieren konnte erfolgreich aufgerufen werden
    - Upload des Bildes war erfolgreich
    - Benachrichtigung, dass es einen neuen Post gab, wurde erfolgreich vermittelt.
  + **Fail-Kriterien**
    - Funktion zum Abonnement konnte nicht aufgerufen werden
    - Benachrichtigung, dass neuer Content gepostet wurde, nicht vermittelt
    - Bild konnte nicht aufgerufen werden
  + **Fallback**
    - -

High Priority

* Login art(hashes), gegebenenfalls feste account für PoC
  + **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
* Login als User mit Sonderrechten
  + **Ablauf**
    - Login als Moderator
    - Zuweisung von Moderator Rechten durch Session Flags
    - Aufruf einer Funktion zum Blockieren eines Users
    - Veränderung des Block-Status eines anderen Users
    - Logout als Moderator
  + **Exit-Kriterien**
    - Login war erfolgreich
    - Moderator Rechte wurden erfolgreich zugewiesen
    - Block-Funktion wurde erfolgreich aufgerufen
  + **Fail-Kriterien**
    - Login war nicht erfolgreich
    - Moderator Rechte wurden nicht erfolgreich zugewiesen
    - Block-Funktion konnte nicht aufgerufen werden
    - Anderer User konnte nicht blockiert werden
  + **Fallback**
    - Eigene Clients für unterschiedlichen System Rollen schreiben. Funktionen die entsprechende Rechte benötigen sind exklusiv auf diesen Clients verfügbar. (Rechteverwaltung)
    - Blockierten User komplett aus der Registry löschen.
* Login in als User ohne Plattform zugriff
  + **Ablauf**
    - Login als blockierter User
    - Prüfen des Block-Status
  + **Exit-Kriterien**
    - Login als blockierter User war erfolgreich
    - Ansicht mit „Du wurdest blockiert“ wurde erzeugt
  + **Fail-Kriterien**
    - Falsche Ansicht wurde erzeugt
  + **Fallback**
    - ???

Medium Priority Keine Umsetzung ab hier

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

Low Priority

* „Befreunden“ mit anderem User
  + **Ablauf**
    - Aufruf einer Funktion zur Freundschaftsanfrage an User B
  + **Exit-Kriterien**
    - Freundschaftsanfrage wurde erfolgreich vermittelt
  + **Fail-Kriterien**
    - Freundschaftanfrage konnte nicht vermittelt werden
  + **Fallback**
    - ToDo
* Annahme „Befreundung“
  + **Ablauf**
    - Akzeptieren von Freundschaftsanfragen
  + **Exit-Kriterien**
    - Freundschaftsanfrage wurde erfolgreich vermittelt
  + **Fail-Kriterien**
    - Freundschaftanfrage konnte nicht vermittelt werden
  + **Fallback**
    - Befreundung erfolgt über eigenes Subforum pro User
* Chat Funktion
  + **Ablauf**
    - Imitieren einer Verbindung zwischen den Usern
    - Erzeugen eines Chatobjekts als Teil dieser Verbindung
    - Aufruf des Chats
    - Ansicht des Chats zwischen den Usern (vorstand)
    - Abfrage nach Texteingabe
    - Abspeichern des Textes im Chat
    - Ansicht des Chats mit dem eingegebenen Text
  + **Exit-Kriterien**
    - 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**
    - 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)