**Projekt Spezifikation** (ca 5 Seiten)

**Projektname:** Name

**1.Thema**

Abstimmungen und Wahlen in Vereinen, KMUs oder sonstigen kleineren Teams durchführen

**2. Umfeld, Ausgangslage**

Viele Institutionen wie zB Vereine konnten während der Pandemie ihre Mitgliederversammlungen nicht wie gewohnt ausführen und mussten auf improvisierte Alternativen ausweichen, zum Beispiel Abstimmungen auf dem Postweg – mit dem Nachteil, dass vor dem Entscheid keine Diskussion stattfand und keine Fragen gestellt werden konnten. Umgekehrt konnten die Mitgliederversammlungen auch nicht einfach per Zoom durchgeführt werden, weil ein einfaches, vertrauenswürdiges – und halbwegs rechtlich wasserdichtes – Tool zum Durchführen der Abstimmungen fehlte.

* 1. Die E-Voting-App verwendet soweit möglich das Backend des Projekts „VOKKO“ des MSc in Innovation and Entrepreneurship der HEC Paris (https://www.hec.edu/en/executive-education/executive-masters/msc-innovation-and-entrepreneurship) und bringt sein Frontend soweit sinnvoll in dieses Projekt ein. Im Backend fehlende Funktionen werden gemockt. Falls das API des Projekts „VOKKO“ sich als inkompatibel mit der E-Voting-App herausstellt, werden die Projekte entkoppelt.
  2. Die E-Voting-App sollte idealerweise sowohl als Mobile-App für die Wähler (zB Ionic React) als auch als Web-App für den Organisator und für die Resultatanzeige zur Verfügung stehen.

**3. Aufgabenstellung & Ziel des Projektes**

Die E-Voting-App füllt diese Lücke.

Damit kann man Versammlungen remote oder hybrid durchführen, so dass jedes mit einem Handy oder Browser bewehrte Mitglied bei den Abstimmungen und Wahlen teilnehmen kann und die Ergebnisse sehen kann.

Der Organisator kann im Voraus Wahlen bzw. Abstimmungsfragen erfassen.

Mit Online-Collaboration-Tools wie Mentimeter oder Miro kann man zwar auch Votings durchführen. Es fehlen aber Transparenz, Nachvollziehbarkeit und Unveränderbarkeit. Die E-Voting-App bietet dazu eine Visualisierung der im Backend laufenden Vorgänge.

Grobes Aufzeigen der Use-Cases (User, Rollen, Interaktionen mit dem System …).

**4. Scope des Projektes**

Die E-Voting-App soll folgende Features beinhalten:

* Erfassung von Abstimmungsfragen bzw. Wahlen mit Kandidaten
* Durchführung der Versammlung
  + Organisator-Sicht:
    - Der Organisator bestimmt Start, Dauer und Ende der Wahl
    - End-Resultatanzeige
  + Wähler-Sicht:
    - Stimmabgabe resp. Kandidatenauswahl
    - End-Resultatanzeige
    - Visualisierung der Blockchain (sichtbar machen, dass „meine Stimme gezählt wurde“)
* Nach der Versammlung: Reports + Charts über die einzelnen Entscheide (Ja, Nein, Enthaltungen etc.)
* Erweiterungsmöglichkeiten
  + Wähler verwalten
  + Kandidatenprofile pflegen
  + Laufende Resultatanzeige

Die Frage, ob eine solches Abstimmungsverfahren tatsächlich rechtlich gangbar wäre, ist out of scope und wird nicht geprüft.

Priorisierung der Features:

* Minimal Viable Product („Must Haves”)
* Nice to Haves – (Was wird als erstes weggelassen, wenn die Zeit nicht reicht?)

**5. Grob-Architektur und Design**

* Technischer System-Kontext, Systemgrenzen, Umsysteme, Deployment-Struktur
* Grob-Architektur der Lösung (Frontend, Backend, Layerung …)
* API-Endpoints, grober Überblick über die URLs der Applikation (Routing)
* Überblick über die zu realisierenden Screens (Screen-Flow)
* Überlegungen zum Statemanagement (welcher State im Client, welcher State im Backend, Client-seitige Persistenz, Offline-Fähigkeit …)
* Technologiewahl inkl Begründung
* Einbindung/Anbindung von relevanten 3rd Party APIs/Komponenten (z.B. Maps, Fahrpläne …)
* Mengengerüste

(Diagramme können hier sehr sinnvoll sein)

**6. Studierende:**

- Remo Peduzzi

- Walter Raaflaub