Aufgabe 3 - Systemarchitektur

1. EventStorming durchführen:

- Bürger registriert: Ein Bürger registriert sich im System
- Feedback(Anliegen/Beschwerde) eingereicht: Ein Anliegen/eine Beschwerde wird in dem System angelegt
- Feedback bearbeitet: Eine Feedback wird von einem Mitarbeiter bearbeitet
- Status des Feedbacks aktualisiert: Der Feedback-Status wird durch den Mitarbeiter geändert
- Feedback-Status abgefragt: Der Feedback-Status wird vom Bürger abgefragt.
- Update des Status' an den Bürger gesendet: Der Bürger wird per Email über eine Status-Änderung seines Anfrage informiert

2. Domänenmodell erstellen:

Kern-Entitäten:

- Bürger
- Mitarbeiter
- Feedback (Anliegen/Beschwerde)
- Status

3. Bounded Contexts

- **Nutzerverwaltung**: Zuständig für die Verwaltung von Nutzerdaten (Bürger / Mitarbeiter)
- **Feedbackverwaltung**: Zuständig für das Erfassen, Verwalten (z.B. updaten, abrufen) von Bürger-Feedback
- **Statusverwaltung**: Zuständig für die Verwaltung von Feedbackstatus (z.B. verwalten, abrufen, anzeigen)
- **Benachrichtigungssystem**: Zuständig für die Benachrichtigung des Bürgers bei Statusupdates

4. Entitäten und Aggregates definieren

- Nutzerverwaltung**:**
    - Entität: Bürger, Mitarbeiter
    - Aggregate: Bürger (Name, Email), Mitarbeiter (Name, Email)
- Feedbackverwaltung:
    - Entität: Feedback
    - Aggregate: Feedback (Enthält alle Daten zu einem Feedback, wie z.B. Titel, Text, Bürger, aktueller Bearbeiter, aktueller Status)
    - Entität: Status
    - Aggregate: Status (Enthält den Namen des aktuellen Status eines Bürgerfeedbacks)
- Benachrichtigungssystem:

    - Entität: Bürger, Feedback, Status
    - Aggregate: Feedback (Enthält alle Daten zu einem Feedback, wie z.B. Titel, Text, Bürger, aktueller Bearbeiter, aktueller Status); Bürger (Name, Email); Status (Enthält den Namen des aktuellen Status eines Bürgerfeedbacks)
- Statusverwaltung:
    - Entität: Feedback, Status
    - Aggregate: Feedback (Enthält alle Daten zu einem Feedback, wie z.B. Titel, Text, Bürger, aktueller Bearbeiter, aktueller Status); Status (Enthält den Namen des aktuellen Status eines Bürgerfeedbacks)

5. Domain Services und Repositories

Services:

- Feedbackservice:
    - Zuständig für die Verwaltung von Bürgerfeedback (Erstellen, updaten, Mitarbeiter zuweisen etc.)
- Statusservice:
    - Verwaltung von Beschwerdestatus (Methoden zur Definition von Statusbedingungen), setzen, abrufen des Status' bei Feedbacks
    - Zuständig für das Abrufen des Status' von Feedback auf Wunsch des Bürgers
- Benachrichtigungsservice:
    - Zuständig für die Benachrichtigung des Bürgers bei Statusupdates
- Benutzerverwaltungsservice:
    - Verwaltung von Benutzern (Nutzer finden, speichern, evtl. löschen)

Repositories:

- FeedbackRepository
    - findFeedbackById(Long id)
    - saveFeedback(Feedback feedback)
    - updateFeedback(Feedback feedback)
- UserRepository
    - findUserById(Long id)
    - saveUser(User user)
- StatusRepository
    - findStatus(String id)

6. Implementierungsstrategie

- **Umsetzung von Entitäten / Aggregates:** Realisierung der einzelnen Entitäten als je eine eigene Java-Klasse mit Attributen und Methoden. Diese sind für die Verwaltung von Beziehungen untereinander notwendig.

- **Domain Services:** Für jeden Service wird eine entsprechende Service-Klasse eingeführt, welche mit den Repositories in Verbindung steht, um die Datenverarbeitung untereinander zu unterstützen.
- **Repositories:** Jedes Repository wird als eigenes Interface definiert mit eigenen Methoden zur Speicherung und zum Abrufen der einzelnen Entitäten.

UML Diagramm: