

Homework 10

Aufgabe 1: User Stories

1. **Lehrer:**
Als Lehrer möchte ich für meine Lehrveranstaltungen Übungsgruppen erstellen, um die Studierenden effizient zu organisieren.
2. **Gruppenleiter:**
Als Gruppenleiter möchte ich die Übungsgruppen, einschließlich der Übungszeiten und der maximalen Anzahl von Studierenden, anpassen und definieren können, um den Ablauf optimal zu gestalten.
3. **Nutzer:**
Als Nutzer möchte ich meinen eigenen Account besitzen und mich sicher einloggen können, sodass ich sicher die App verwenden kann.
4. **Student (Anmeldung):**
Als Student möchte ich mich zuhause aus für alle gewünschten Kurse anmelden können, sodass ich dies bequem auch unterwegs machen kann und nicht in der Uni machen muss.
5. **Student (Verfügbarkeit):**
Als Student möchte ich Zeiten angeben können, an denen ich nicht an Übungen teilnehmen kann, sodass diese bei der Gruppenzuteilung berücksichtigt werden.
6. **Student (Fairness):**
Als Student möchte ich sicherstellen, dass die Gruppenzuteilung fair erfolgt, sodass ich eine faire Chance habe den Kurs zu bekommen der mir passt.

Aufgabe 2:

- a) **Analyse des Burn down Charts:** Das Burn down Chart zeigt, dass das Entwicklungsteam in den ersten fünf Tagen mit dem geplanten Fortschritt Schritt halten konnte. Jedoch verzögerte sich der Fortschritt zwischen Tag 5 und 10, wodurch am Ende des Sprints noch Restaufgaben übrigblieben. Diese verbleibenden Aufgaben müssen besprochen und in den nächsten Sprint mit eingeplant werden.
- b) **Wichtige Ereignisse im Sprint:**
 - **Tag 1–2:** Die Koordination des Teams erfolgte online.
 - **Tag 3–4:** Refactoring aufgrund neuen Updates von Spring.
 - **Tag 5:** Timothy hat das Team verlassen.
 - **Tag 9:** Es wurde an der Überarbeitung der Gruppenzuteilung gearbeitet (Rework Group Distribution).
- c) **Empfehlung zu Issue #1:** sollte nach Abschluss des aktuellen Sprints analysiert und im nächsten Sprint mit hoher Priorität eingeplant werden, um den Fortschritt sicherzustellen.

Aufg 3: Sprint for the Students Part of the System:

a) MoSCoW Method regarding the upcoming sprint:

Must Have:

- Students should have the possibility to apply as a group to exercise sessions
- Rechenzentrum demands Java Version 21 or higher.
- Als Student möchte ich mich zuhause aus für alle gewünschten Kurse anmelden können sodass ich dies bequem auch unterwegs machen kann und nicht in der Uni machen muss. (zwei Issues aus ex. 1 zusammengefasst)
 - Als Nutzer möchte ich meinen eigenen Account besitzen und mich sicher einloggen können sodass ich sicher die App verwenden kann.
- Als Student möchte ich Zeiten angeben können, an denen ich nicht an Übungen teilnehmen kann, sodass diese bei der Gruppenteilung berücksichtigt werden.
- Als Student möchte ich sicherstellen, dass die Gruppenteilung fair erfolgt sodass ich eine faire Chance habe den Kurs zu bekommen der mir passt.

Should Have:

- The maximum size of groups of students allowed to collectively register for exercise sessions should be decidable by the lecturer. Default is 3.

Could Have:

- All data stored in the database must be encrypted using state-of-the-art encryption standards
- Als Lehrer möchte ich für meine Lehrveranstaltungen Übungsgruppen erstellen, um die Studierenden effizient zu organisieren.
- Als Gruppenleiter möchte ich die Übungsgruppen, einschließlich der Übungszeiten und der maximalen Anzahl von Studierenden, anpassen und definieren können, um den Ablauf optimal zu gestalten.

Won't Have: (Won't be included in upcoming sprint, but maybe afterwards sprints)

- The system should crawl all course data from KLIPS on demand by administrators.

b) Goal: student's part of the system ready to be tested by real students.

Issues that would contribute:

- Students should have the possibility to apply as a group to exercise sessions
- Users should have their own Account and be able to log in from any place
- Students should be able to put in their unavailable Time
- Group distribution should be fair

c)/d)/f) Issues broken down into smaller parts and size Estimation:

- Students should have the possibility to apply as a group to exercise sessions **L => 8 Tage**
 - Offer "Group Apply" Option: Frontend => Rajesh (Frontend Entwicklung)
 - Option to search for other students

- Connect Students accounts => Tom (Backend, Experte in Spring)
- Tests => Benedikt (Software Tester)
- Users should have their own Account and be able to log in from any place **S => 4 Tage**
 - Log in option; Frontend (Rajesh)
 - Entwurf und Implementierung des Benutzerauthentifizierungssystem in der Datenbank. => Sara (Database Engineer)
 - Integration von Passwort-Wiederherstellung und E-Mail-Benachrichtigung => Rajesh (Frontend), Tom (Backend), Benedikt (Testing)
 - Connection to other University Systems for faster log in => Tom (Backend), Sara (Database Engineer)
 - Ensure Security => Ahmed (Sicherheit)
- Students should be able to put in their unavailable Time **M => 6 Tage**
 - Offer Option to put in time: Frontend => Rajesh(Frontend)
 - Change in/adapt the group algorithm; Change priority of the student for the other exercises => Tom (Backend)
 - Tests => Benedikt (Software Tester)
- Group distribution should be fair **L => 8 Tage**
 - Define fair for this system => Tom (Backend)
 - Write Algorithm => Sharon (Developer (frontend und backend))
 - Code Deployment => Juan (Dev Ops und Cloud engineer)
 - Tests => Björn (mobile Entwicklung)

e)

Die ersten drei genannten Issues sollten sich für den kommenden Sprint angeschaut und bearbeitet werden. Zum Schluss des Sprints sollte dennoch kontrolliert werden, ob der Algorithmus eine faire Gruppen Zuteilung berechnet hat.

f) siehe bei c und d