Homework

Homework 1:

EGD = Exercise group Distribution

Department = Computer Science
Students feel unfair & lecturer handles the process differently
↓
register quick ⇒ get preferred groups
+
others end up with sessions that don't fit their schedules
↳ conflicts with other courses

New System:
- lecturers create exercise groups for their courses, set group details (session times & # of students in each group)
- Students will log in to view available sessions, can register for all courses

X priority System; but mark times where student are NOT available due to other circumstances → automated fair distribution (consideration: • available time slots • schedule conflicts)

Goal of system: minimize the # of students that could not be assigned a group.
↓
if a student couldn't be assigned → manual administration

Students get to know in which group they are through Notifications.

future versions: • register as a group for single lectures or multiple at once (low priority)

Scope: Start <1000 users / useful? system might be adopted university-wide
↓
be able to handle thousands of students (especially during registration periods)

Software should be easy to use for students / lecturers / system admins.
Priority = Security, with access controlled through uni credentials (Shibboleth) & personal data protected from unauthorized access.

Exercise 1:

a) Students
- Why: Users of system
- How: register for they'll be able to get a suitable exercice for their timetable & avoid conflicts with other courses
- Cause of interest: makes enrolment easier & simplifies the organization of timetable + they get to attend all exercices they want

Lecturers
- Why: Users of system
- How: create exercice groups for their courses, set group details (session times & # of students in each group) & ensure fair student distribution
- Cause of Interest: Overview of exercices & helping students to attend their courses

Department of Computer Science
- Why: responsible for registration Process
- How: Controlling of developing the system
- Cause of Interest: provide a better offer for students & avoid courses conflicts

system admins
- Why: User of system
- How: administrate the system & garanties that it works
- Cause of Interest: have a simplified system that is userfriendly

Chair of Software & Systems Engineering

Why: responsible for CS Department & for simplifying the registration process for students + provide a good education level for them

How: less complains for students

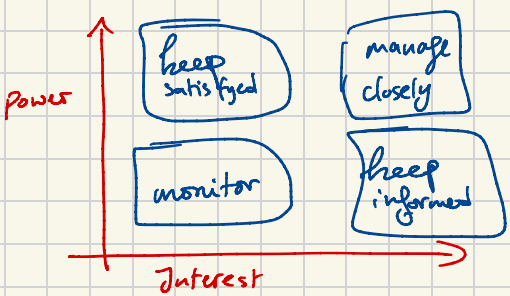
Security responsible

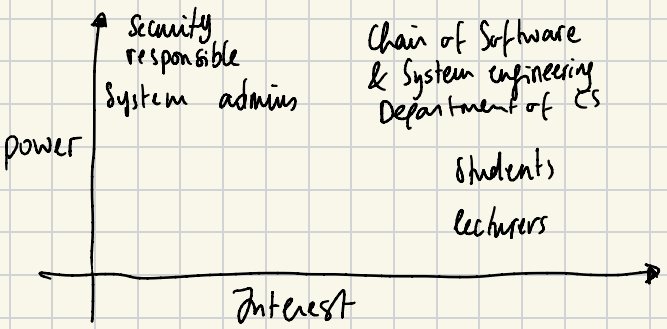
Why: responsible for the security of the system

How: the check if the system is according to DSGVO - Conformity

Cause of interest: having a secure system

b)





Exercice 2:

6 functional requirements:

- lecturers should be able to create exercice groups for their courses & set group details like session times & the # of students in each exercice
- The system should notify students of the exercice group they were assigned to
- Students will log in to view available sessions
- Students can register for all courses they want to attend exercises for in one semester
- " " should be able to mark times where they are NOT available due to other circumstances
- The system should distribute students across groups in a fair manner.

3 quality requirements & their respective quality attribute:

- System should be user-friendly (Usability)
- System should distribute students across groups in a fair manner
- System should be able to adapt to handle thousands of students (Scalability)
- System should be secure (priority) with access controlled through university credentials (Shibboleth) & personal data protected from unauthorized access (Security)

1 constraint:

- System should be programmed in Java.

1 Project requirement:

- Budget < 30 000 €

1 Process requirement:

- Students must participate in the system's development as both developers & testers.

Exercice 3:

Check for:
- precision
- consistency
- verifiability
- validity

- lecturers should be able to create exercice groups for their courses
  - → precision: yes, specific & clear
  - → consistency: yes, it aligns with the system's core functionality
  - → verifiability: yes, we can test by trying to create an exercice group & setting details
  - → validity: yes, lecturers need to manage group sessions for correctness & so that students can view them & their details.

- Students will log in to view available sessions
  - → Precision: Partially; it could be more specific about how the login works & what students see
  - → Consistency: Yes, it is consistent with the system's purpose
  - → verifiability: Yes, we can test it by trying to log in
  - → validity: yes; students need to view & register for groups.

- Students mark unavailable time slots
  - → Precision: yes, very clear defined
  - → validity: yes, the system should know, when students don't have time for a fair distribution
  - → Consistency: yes, there are no conflicts with other requirements
  - → verifiability: yes, by testing the input of time slots & checking if the students' distribution goes according to it.

- The system should notify students of the exercice group they were assigned to
  - → Precision: yes, clear that students will receive notifications
  - → Consistency: yes, it aligns with the system's goals
  - → verifiability: yes, it can be tested by running the algorithm & verifying fair group distribution
  - → validity: yes, students need to be informed about their group assignments

- Students can register for all courses they want to attend exercises for in one semester
  - → Precision: yes, it's clear
  - → Consistency: It aligns with the system's goal
  - → verifiability: yes, it can be checked by trying to registrate for a course.
  - → validity: yes, students should be able to register for group sessions.

- Students can register for all courses they want to attend exercises for in one semester
  - → Precision: Partially, it doesn't specify how the distribution works or handles conflicts.
  - → Consistency: yes, no conflicts
  - → verifiability: yes, can be tested by running the algorithm & verified fair group distribution
  - → validity: yes, it's the main goal of the system

Quality-requirements:

- System should be user-friendly:
  - → Precision: No, user-friendly is very broad, there is no further description of how this objective should be achieved nor what 'user-friendly' actually defines.
  - → Consistency: yes, there are no conflicts with other requirements
  - → verifiability: No, there is no clear process of how to test, how user-friendly the system is, as there is no clear goal
  - → yes, the system should be easy to use for students / lecturers