

Exercise(1):

a) Students

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

lecturers

- Why: Users of system
- How: create exercise groups for their courses, set group details (session times & # of students in each group) & ensure fair student distribution
- Cause of interest: Overview of exercises & 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 & guarantees 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 Prozess for students & provide a good education level for them
- How: less complaints for students

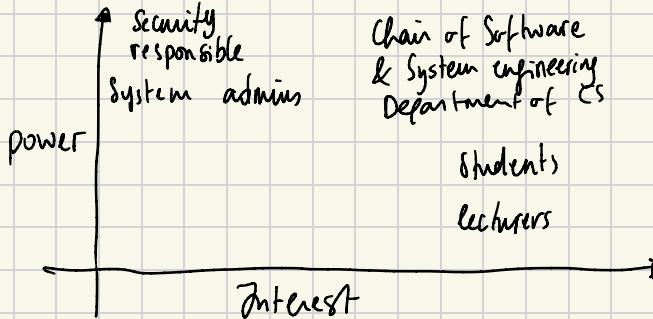
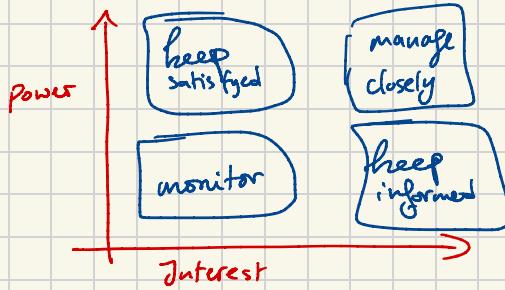
Security responsible

- 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)



Exercise 2:

6 functional requirements:

- lecturers should be able to create exercise groups for their courses & set group details like session times & the # of students in each exercise
- The system should notify students of the exercise 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 adopt 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.

Exercise 3:

- Check for:
- precision
 - consistency
 - verifiability
 - validity

- lecturers should be able to create exercise 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 exercise 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 exercise 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 register 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
 - Validity: yes, the system should be easy to use for students / lecturers