

Aufgabe 1: a) Identify 6 stakeholders of the project. Explain briefly, per stakeholder, why/how they are affected by the system and why they are interested in the development of this system.

1. Lecturers:

- **Interest:** High
- **Power:** High
- **Impact:** Lecturers will use the system to create and manage exercise groups for their courses. They are directly involved because they need the system to streamline the management of students, ensuring a fair distribution without manual intervention. They are interested in the system's functionality, usability, and reliability, as it will directly impact their workload and course organization.
- **Why Interested:** The system will simplify the allocation process and reduce administrative overhead.

2. Students:

- **Interest:** High
- **Power:** High
- **Impact:** Students are the primary users who will be assigned to exercise groups based on their availability. Their interest lies in ensuring that the system fairly considers their schedule conflicts and assigns them to suitable groups. Student satisfaction with the system will directly affect the system's acceptance and success.
- **Why Interested:** They want a fair, transparent process that minimizes scheduling conflicts and maximizes their choice flexibility.

3. System Administrators (IT Department):

- **Interest:** High
- **Power:** High
- **Impact:** The system administrators will be responsible for the ongoing maintenance, user management, and technical support of the system. They need to ensure the system runs smoothly, is secure, and can scale as required. Their interest is in having a robust, maintainable system with minimal downtime or technical issues.
- **Why Interested:** They need to ensure the system integrates with existing university infrastructure and provides a seamless experience for users.

4. University Leadership (Administration):

- **Interest:** Medium
- **Power:** High
- **Impact:** The university leadership may not be involved in day-to-day use but will be interested in the system's overall success and scalability. If the system works well, it could be rolled out to other departments, which would make administration smoother across the university. Their focus will be on budget management, system scalability, and the broader university reputation.
- **Why Interested:** A scalable and successful system reflects positively on the university and reduces administrative burden.

5. Software & Systems Engineering Chair:

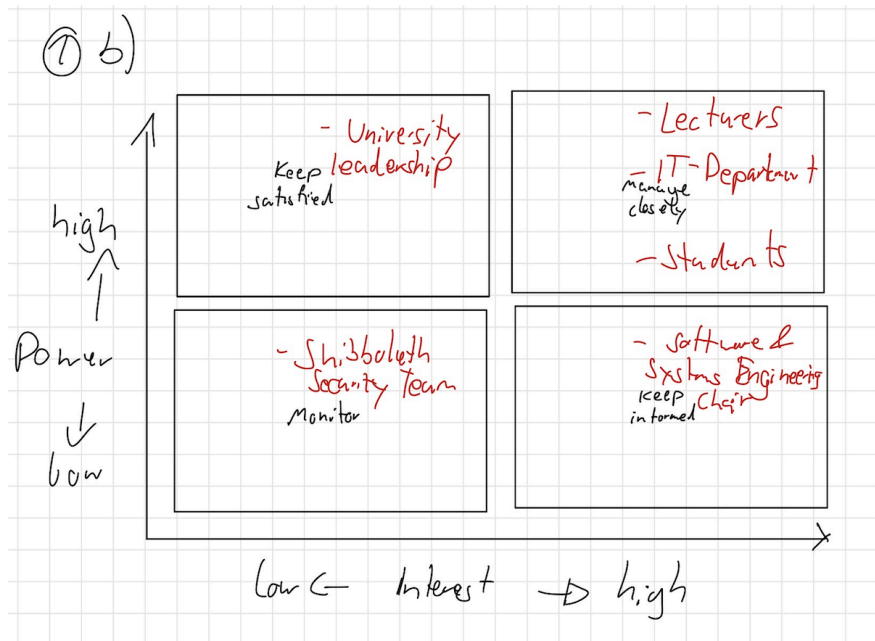
- **Interest:** High
- **Power:** Medium
- **Impact:** This chair initiated the project and will be overseeing the development process. They will also be managing the student developers and testers, so they have a vested interest in ensuring the project is successful, within budget, and meets the outlined requirements. They are deeply involved in technical aspects and quality assurance.

- **Why Interested:** The project is under their supervision, and its success is important for the academic and administrative goals of their department.

6. Shibboleth Security Team:

- **Interest:** Medium
- **Power:** Medium
- **Impact:** The team responsible for managing the university's authentication system will ensure that the EGD system integrates with Shibboleth for secure logins. They will be involved primarily during the development and deployment stages to ensure security protocols are followed. Their interest is in maintaining the security and privacy standards required for university systems.
- **Why Interested:** Ensuring that authentication is secure and compliant with university policies.

b)



Aufgabe 2:

a) 6 Functional Requirements:

1. The system shall allow lecturers to create exercise groups for their courses, specifying session times and maximum group sizes.
2. The system shall allow students to log in and view available exercise groups for all courses they are registered for in a semester.
3. The system shall allow students to mark times when they are unavailable due to schedule conflicts.
4. The system shall automatically distribute students to exercise groups based on their availability and group capacities.
5. The system shall notify students of their assigned exercise groups.
6. The system shall allow manual administration for students who could not be automatically assigned to any group.

b) 3 Quality Requirements and their respective quality attributes:

1. **Scalability:** The system should be able to scale to support thousands of students during peak registration periods without performance degradation.
2. **Security:** The system should ensure access control using Shibboleth and protect personal data from unauthorized access.
3. **Usability:** The system should be easy to use for both students and lecturers, with an intuitive interface that requires minimal training.

c) 1 Constraint:

- The system must be developed in Java to allow student developers and testers to participate in the development process.

d) 1 Project Requirement:

- The total budget for the development of the system shall not exceed 70,000 euros.

e) 1 Process Requirement:

- The first test version of the system must be ready by the beginning of the winter semester 2025/26.

Aufgabe 3:

a) Functional Requirements

1. The system shall allow lecturers to create exercise groups for their courses, specifying session times and maximum group sizes.

- **Precision:** Partially fulfilled. While it states the main functionality, it lacks details about how this will be done
- **Consistency:** Fulfilled. No conflicts with other requirements.
- **Verifiability:** Partially fulfilled. It can be tested if lecturers can create groups, but the process details need to be clearer.
- **Validity:** Fulfilled. The need for this function is clear.

Improvement: The system shall provide lecturers with a form to create exercise groups, specifying session times and group size limits.

2. The system shall allow students to log in and view available exercise groups for all courses they are registered for in a semester.

- **Precision:** Fulfilled. Clear in its purpose.
- **Consistency:** Fulfilled. No conflicts.
- **Verifiability:** Fulfilled. It can be easily tested if students can log in and view the groups.
- **Validity:** Fulfilled. Essential for students' ability to access course groups.

Improvement: None necessary, as the requirement is clear and testable.

3. The system shall allow students to mark times when they are unavailable due to schedule conflicts.

- **Precision:** Partially fulfilled. It is not clear how students will mark times
- **Consistency:** Fulfilled. No contradictions.
- **Verifiability:** Partially fulfilled. Needs more detail about how unavailability is recorded to be testable.
- **Validity:** Fulfilled. It's a necessary feature for group assignments.

Improvement: The system shall provide students with a calendar interface to mark times when they are unavailable due to schedule conflicts.

4. The system shall automatically distribute students to exercise groups based on their availability and group capacities.

- **Precision:** Partially fulfilled. It's clear in intent, but there's no explanation of the distribution logic or constraints.
- **Consistency:** Fulfilled. No conflicts.
- **Verifiability:** Partially fulfilled. The automatic distribution process needs more clarity to define how the system's fairness and success will be tested.
- **Validity:** Fulfilled. Addresses the key goal of fair distribution.

Improvement: The system shall use an algorithm to automatically distribute students to exercise groups based on their availability and group capacities, ensuring that the maximum group size is not exceeded.

5. The system shall notify students of their assigned exercise groups.

- **Precision:** Fulfilled. Clear in its intent.
- **Consistency:** Fulfilled. No contradictions.
- **Verifiability:** Fulfilled. Testable by checking if notifications are sent.
- **Validity:** Fulfilled. Critical for informing students of their group assignments.

Improvement: None necessary; the requirement is clear and testable.

6. The system shall allow manual administration for students who could not be automatically assigned to any group.

- **Precision:** Partially fulfilled. It's unclear what kind of manual administration will be needed.
- **Consistency:** Fulfilled. No conflicts.
- **Verifiability:** Partially fulfilled. Needs more detail to test what manual administrative features are provided.
- **Validity:** Fulfilled. Manual intervention is necessary for unassigned students.

Improvement: The system shall provide administrators with a manual override interface to assign students to exercise groups if they could not be assigned automatically.

b) Quality Requirements

1. Scalability: The system should be able to scale to support thousands of students during peak registration periods without performance degradation.

- **Precision:** Partially fulfilled. It lacks specific numbers or performance metrics.
- **Consistency:** Fulfilled.
- **Verifiability:** Partially fulfilled. Specific metrics would make it more testable.
- **Validity:** Fulfilled. Scalability is essential as the system may be adopted university-wide.

Improvement: The system shall be able to scale to support up to 10,000 students during peak registration periods, with a maximum response time of 2 seconds per request.

2. Security: The system should ensure access control using Shibboleth and protect personal data from unauthorized access.

- **Precision:** Fulfilled.
- **Consistency:** Fulfilled.
- **Verifiability:** Fulfilled. Security can be verified through penetration testing and audits.
- **Validity:** Fulfilled. Ensuring security is essential for any university system.

Improvement: None necessary; the requirement is precise and verifiable.

3. Usability: The system should be easy to use for both students and lecturers, with an intuitive interface that requires minimal training.

- **Precision:** Partially fulfilled. "Intuitive" and "minimal training" are vague terms.
- **Consistency:** Fulfilled.
- **Verifiability:** Partially fulfilled. It's unclear how "intuitive" or "minimal training" will be measured.
- **Validity:** Fulfilled. Usability is a key quality attribute for the system.

Improvement: The system shall have a user-friendly interface that allows students and lecturers to complete tasks with less than 30 minutes of training and no more than 3 clicks per action.

c) Constraint

The system must be developed in Java to allow student developers and testers to participate in the development process.

- **Precision:** Fulfilled.
- **Consistency:** Fulfilled.
- **Verifiability:** Fulfilled. It's clear and testable whether Java is used.
- **Validity:** Fulfilled. The constraint is clear and makes sense given the project's educational context.

Improvement: None needed; the constraint is clear.

d) Project Requirement

The total budget for the development of the system shall not exceed 70,000 euros.

- **Precision:** Fulfilled.
- **Consistency:** Fulfilled.
- **Verifiability:** Fulfilled. It's easy to verify the budget limit.
- **Validity:** Fulfilled. A budget limit is essential for project planning.

Improvement: None needed; the requirement is clear and precise.

e) Process Requirement

The first test version of the system must be ready by the beginning of the winter semester 2025/26.

- **Precision:** Fulfilled.
- **Consistency:** Fulfilled.
- **Verifiability:** Fulfilled. The deadline is clear and easy to track.
- **Validity:** Fulfilled. The deadline aligns with the project's timeline.

Improvement: None needed; the requirement is clear and testable.

Aufgabe 4:

Title: Exercise group scheduling

Actors: Primary Actor: User (student), EGD system, Secondary Actor: lecturer

Precondition:

- user is enrolled in the courses
- the schedules of the courses are entered in the system
- user is registered to the system and has access to it

Trigger: User opens the EGD system (application or web)

Main success scenario:

1. User: Enters the log-in data and enters into the system
2. System: **Verifies authority and the log-in**
3. System: Shows the starting page and the available exercise groups
4. User: **Select the courses which the user wants to attend**
5. System: **Asks the user in which times the user is not available**
6. User: Enters the times, in which the user is not available
7. System: Creates a suitable schedule and assigns the user into the fitting groups
8. System: **Shows the user the schedule** and sends it as an confirmation to the users email

Alternative

Paths:

- 2a: [User enters wrong log-in data]
System: Shows error message -> return to step 1
- 7a: [System is unable to assign the user without any conflict]
System: Shows an error message and informs the user which group(s) are affected
- 7a2: System: Schedules the user into a available group (with conflict) and tells the user to contact the lecturer to solve the conflict manually
- 7a3: **System: Shows the contact details of the lecturer**

Postcondition: The user is now enrolled in the exercise groups without any conflicts