Homework 1

von Amal Bolakhrif (7423946), Beyza Karaca (7422370), Lena Igoumrane (7424588)

1.) Stakeholders

1. Professors :
- they are highly interested as they need an efficient and fair way to assign students to exercise groups without the current manual workload
- they control the courses and exercise groups and can influence the acceptance and use of the system

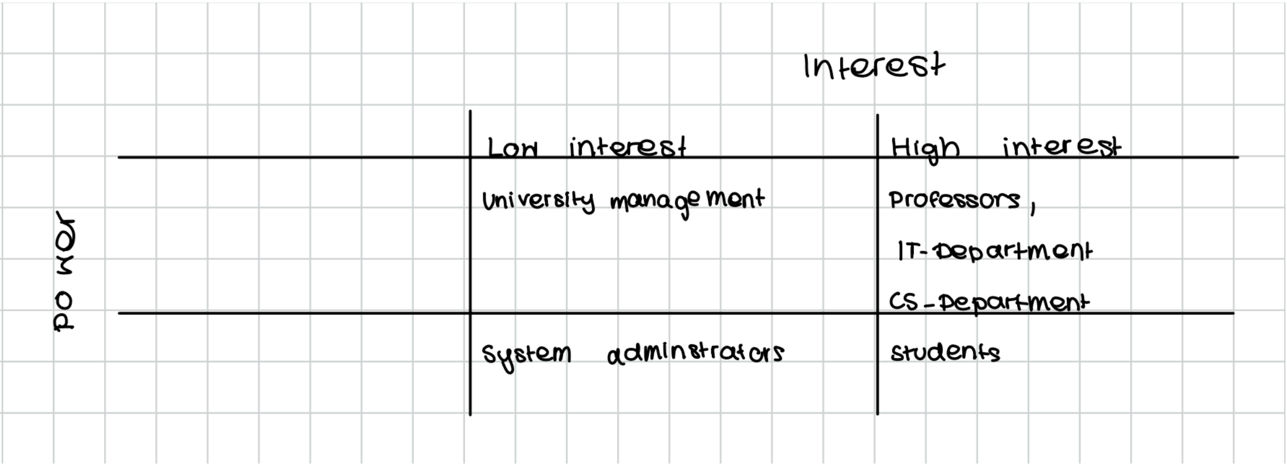
2. Students
- they want a fair distribution of exercise groups that align with their course schedules especially to avoid conflicts
- in general their satisfaction with the system impacts overall university experience
- they are the primary users but don't have direct decision-making authority in the development

3. IT-Department:
- resonsible for implementing and maintaining the system, ensuring it integrates well with other systems and functions reliably
- they control the technical resources , deployment and support

4. Computer science department:
- want a functional, fair system that can enhance the student experience and be expanded university-wide
- they provide funding and strategic direction for the system

5. System administrators:
- responsible for managing the technical operations of the EGD system, ensuring it is secure and functions smoothly
- they influence the system's performance and security but decisions are largely made by authorities

6. University managment:
- interested in the potential expansion of the system to the whole University, as it could improve scheduling and resource management across faculties
- if the system is adopted across the university, they provide support and resources for scaling the system



Interest

	Low interest	High interest
Power	University management	Professors , IT-Department CS-Department
	System adminstrators	students

2.)

a) Functional requirements

1. Group creation by professors
2. Student registration for their courses
3. Non-availability input by students
4. Automatic/ fair group assignments
5. Manual resolution of group assignments possible
6. Notify students of assignments

b) Quality requirements

1. Security
2. Usability
3. Scalability

c) Constraint

System must be developed in Java

d) Project requirement

System should be deployed in WS 2026/27, first test versions should be ready in beginning of WS 2025/26

e) Process requirement

Access controlled through university credentials (Shibboleth)

3.)

functional requirements

	1. FR: Group creation by professors	**2. FR: Student registration for their courses**	**3. FR: Non-availability input of students**	**4. FR: Automatic/ fair group assignments**	**5. FR: Manual override possible**	**6. FR: Notify students of assignments**
Precision	Could be more detailed and specify whether professors can edit or delete created groups	Clear communication that students registrate for all courses they want to attend exercises for and not for the exercise groups themselves, like they used to do	Precise enough	Specifies that system considers students available time slots and any schedule conflicts with other courses; Could specify details on conflict resolution and algorithm logic	Not explained precise enough, description only mentions that manual administration will be necessary, If a student could not be assigned	Precise, but could specify notification method (e.g., email, system notification, both)
Consistency	Consistent with system's goal to allow professors to manage exercises, does not cause inefficiencies	Consistent with the project goals, to later on have a fair group distribution (no first come, first serve), does not cause any inefficiencies	Consistent with system's goal of fair assignment based on availability, could cause ineffiency through dishonest students and interfere with goal of fairness	Consistent with the goal of fairness in group distribution	Consistent with the goal of fairness in group distribution whilst considering special cases	Consistent with the system's objective to keep students as stakeholders informed, does not cause any inefficiencies
Verifiability	Verifiable through testing	Verifiable through testing	Verifiable through testing, although it does not verify students' honesty	Likely verifiable with test cases, although absolute fairness can never be achieved and it also is a rather subjective aspect (some students may still find it unfair)	Verifiable, can be tested by simulating unresolved cases	Verifiable, notifications can be tested
Validity	Valid, important function as lecturers and their respective co-workers decide on group details (date, time, place, etc.); hence, have the knowledge oft the input of available groups into the system	Valid, as students are a main stakeholder of the project, this part is core to the system	Valid, as availability is key to a fair group distribution, but function cannot necessarily check how honest students are being	Valid	Valid, as some conflicts will require manual intervention	Valid, as notifications are essential for informing students of their assigned groups as an important part of their syllabus'.
Improvement	"Professors shall be able to create, edit, and delete exercise groups for their courses."	/	/	No improvement, maybe add, that it uses a pre-defined fairness algorithm	"Lecturers may have ability to adjust group distributions manually if conflicts are not resolved automatically"	"System shall notify students of their group assignments via email and system notifications"

	1.QR: Security	2. QR: Usability	3. QR: Scalability	Constraint: Development in Java	Project Req.: Project and test version deadlines	Process Req.: Access control through uni credentials
Precision	“Security will be a priority, with access controlled through university credentials (Shibboleth) and personal data protected from unauthorized access”, is precise	Precise, but could add more specifics about e.g., (graphical) user interface design (standards)	General Project description mentions details on how many users the system should handle at different phases (first versions vs. future)	Precise	Precise, time schedule also broken down with the planned test versions, that should be ready approx. one year earlier, for more clarity	Precise, but could specify how exactly Shibboleth will be integrated
Consistency	Consistent with the goal of protecting personal information of stakeholders/ students, does not cause issues	Consistent with the project’s goal of being user-friendly, leads to efficiency	Consistent with the requirement to handle large user bases, should not cause any issues or unsolvable interferences with other requirements	Consistent with scalability req.	Consistent with project’s timeline and potentially leads to more efficiency in developing system	Consistent with the goal of secure, university-based authentication
Verifiability	Verifiable, system/ security measures can be tested	Verifiable, usability tests can measure ease of use (surveys), even though usability can be subjective	Verifiable through load testing	Verifiable	Verifiable, progress can be tracked	Verifiable by testing Shibboleth integration
Validity	Valid, as protecting student data is critical	Valid, as usability impacts system’s success	Valid, as system needs to scale potential university-wide use	Valid, as system needs to scale with potential university-wide adoption and as the chair of Software & Systems Engineering wants to let students participate in the development of the system	Valid, as project should not take forever and timely deployment is necessary	Valid, necessary for protection and secure login
Improvement	/	“The system shall adhere to usability standards such as ISO 9241, ensuring ease of use for both students and lecturers.”	/	/	/	“Access to system shall be controlled through the university’s Shibboleth system, ensuring secure authentication for students and lecturers”

4.)
<u>Use case</u>: Student interaction with EGD
<u>Actors</u>: Students, Lecturers, System Admin, Course & exercise Coordinator
<u>Goal:</u> Assign students to exercise groups based on availability and preferences

<u>Pre-Conditions</u>: Valid registration in courses and exercise groups & access to an active university account

<u>Main scenario:</u>

1. Student logs into system
2. Enters (non) availability
3. The System assigns groups based on this data
4. Student receives a confirmation e-mail

<u>Alternative Scenario:</u>

- If no groups are available, the student is informed to contact lecturers for manual assistance
- Students can adjust their availability or preferences before the registration deadline
- Automatic scheduling with conflicts

<u>Post-Conditions:</u> Student is assigned to groups or notified if conflicts arise

<u>Trigger:</u> Student logs into the EGD system and submits their availability and preferences for exercise group

<u>Assumptions:</u>

- The system can handle student availability & preference data efficiently
- Lectures are accessible for conflict resolution