Hendrik Steinmetz Matrikelnr: 7384311

Exercise 1

(a)

1. Students
   - will be the main user of the system
2. Lecturers
   - will use the system to provide lectures and lecture times
3. Uni Köln Department of Computer Science
   - acts as the projects initiator and sponsor
4. Sysadmins
   - take care of maintenance and administration
5. Uni Köln Software and Engineering Chair
   - oversee and manage the project development
6. Uni Köln
   - provides budget and might adopt system university-wide

(b)

Exercise 2

- Functional Requirements
  1. lecturers can create and manage groups for their courses
  2. lecturers can set session times and session capacity
  3. students can view available groups and sessions
  4. students can mark time slots where they're unavailable
  5. the system assign students to slots based on their preferences
  6. the system notifies students once assigned
- Quality Requirements
  1. Usability: the system should be easy to use
  2. Scalability: has to be able to handle thousands of users during registration phases
  3. Security: data and access information has to be sufficiently secured
- Constraint
  1. java has to be used for the development of the system
- Project Requirement

    1. the system should be ready for deployment by WS 2026/27 with a test version to be made available a year before
- Process Requirement
    1. students should be allowed to participate in developing and testing then system

Exercise 3

- FR
    1. not precise, "manage" is rather vague, not verifiable because of that -> change "manage" to "create, edit and delete"
    2. precise, verifiable
    3. precise, verifiable
    4. verifiable, precision can be improved -> students can select 30 min intervals between XX:XX and YY:YY
    5. precise, verifiable
    6. not precise, specify notification method -> email, sms, etc.
- QR
    1. very vague, specify concrete functionality
    2. partially vague, specify target peak user load
    3. precise, verifiable
- Constraint
    1. precise, verifiable

Exercise 4

- Title: Exercise Group Selection
- Actor(s): Student, Lecturer
- Preconditions: Student has an account
- Trigger: Student views available time slots for a session
- Main Success Scenario:
    1. Student: logs in
    2. System: verifies login data
    3. Student: enters availability information and presses "assign"
    4. System: checks for potential conflicts
    5. System: assigns all courses that fit the students' time constraints
    6. System: notifies student of successful assignment
- Alternative Paths:

- 7a1: System: one or more courses cause conflicts and the student cannot be assigned as needed
- 7a2: System: assigns all other eligible courses
- 7a3: System: asks student to coordinate unassigned courses with lecturer