Analytical and Design Documentation

Purpose and Main Functionalities

The purpose of the JustGrades system is to support the academic process by managing student course registrations, grade tracking, and performance analysis. The system provides separate access and functionalities for users with different roles — students, lecturers, and administrators

Student functionalities:

- Viewing final grades for enrolled courses
- Viewing detailed scores earned in each course component
- Registering for available courses
- Deregistering from selected courses

Lecturer functionalities:

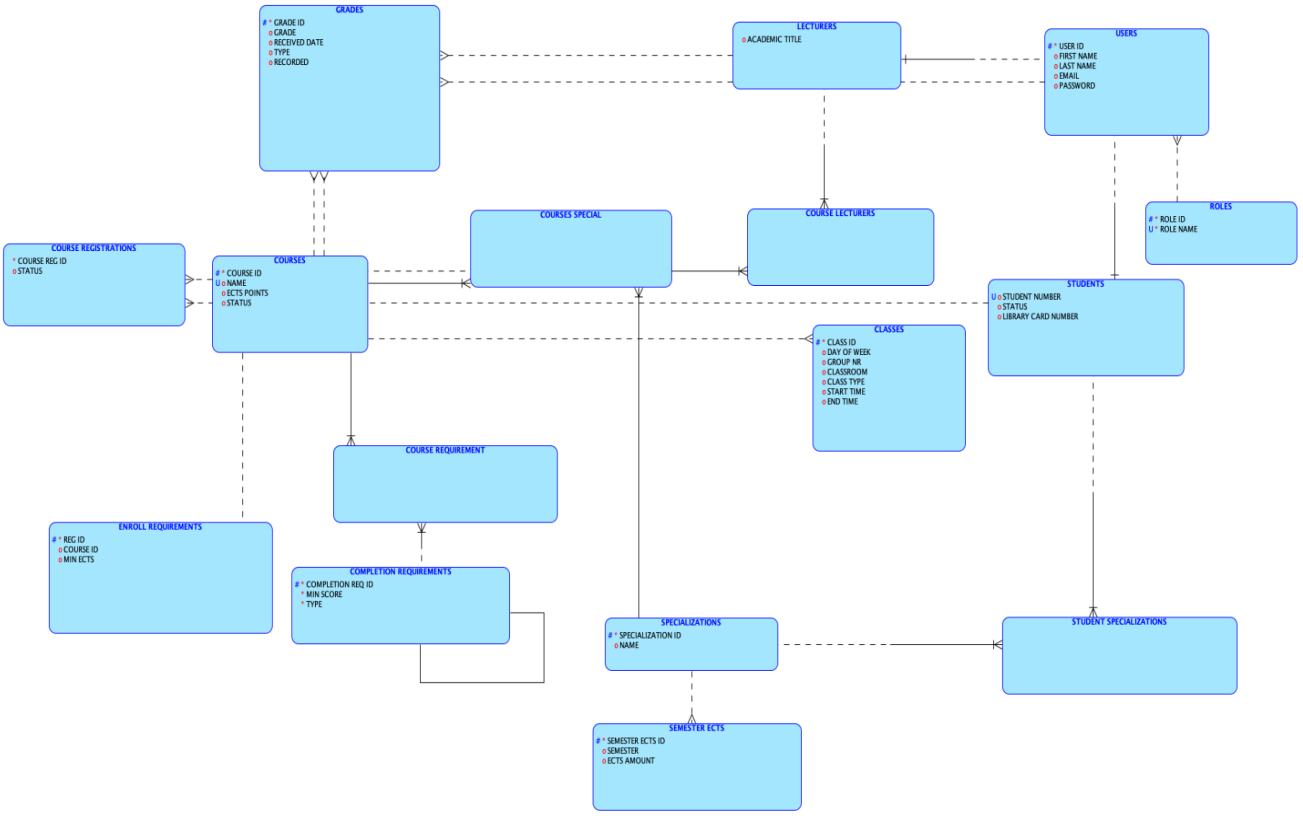
- Assigning grades to students
- Viewing student performance in a given course
- Defining course structure (e.g., number and weight of labs, presence and weight of final exam, minimum points required to pass)
- Closing a course and automatically calculating final grades
- Opening and closing the semester

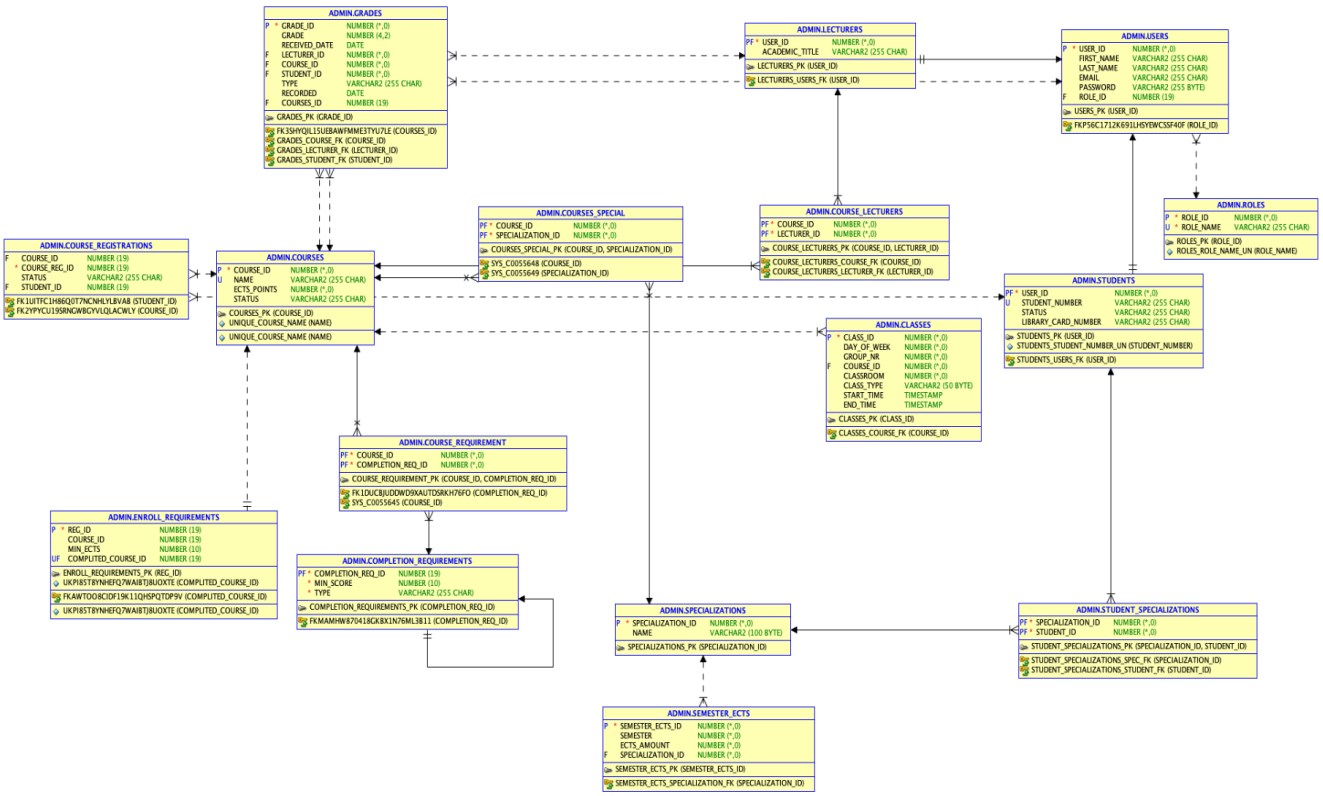
Shared functionalities:

- Generating visual reports:
    - Grade distribution charts
    - Cross-sectional statistics for specific assessments

Entity Descriptions

The logical model presents the structure of entities and their relationships in a way that is independent of the database implementation. The physical model includes data types and constraints as defined in the Oracle DB environment


GRADES
* GRADE ID
o GRADE
o RECEIVED DATE
o TYPE
o RECORDED
LECTURERS
o ACADEMIC TITLE
USERS
* USER ID
o FIRST NAME
o LAST NAME
o EMAIL
o PASSWORD
ROLES
* ROLE ID
U * ROLE NAME
COURSES SPECIAL
COURSE LECTURERS
COURSE REGISTRATIONS
* COURSE REG ID
o STATUS
COURSES
* COURSE ID
U o NAME
o ECTS POINTS
o STATUS
STUDENTS
U o STUDENT NUMBER
o STATUS
o LIBRARY CARD NUMBER
CLASSES
* CLASS ID
o DAY OF WEEK
o GROUP NR
o CLASSROOM
o CLASS TYPE
o START TIME
o END TIME
COURSE REQUIREMENT
ENROLL REQUIREMENTS
* REG ID
o COURSE ID
o MIN ECTS
COMPLETION REQUIREMENTS
* COMPLETION REQ ID
* MIN SCORE
* TYPE
SPECIALIZATIONS
* SPECIALIZATION ID
o NAME
STUDENT SPECIALIZATIONS
SEMESTER ECTS
* SEMESTER ECTS ID
o SEMESTER
o ECTS AMOUNT


ADMIN.GRADES
P * GRADE_ID NUMBER (*,0)
GRADE NUMBER (4,2)
RECEIVED_DATE DATE
F LECTURER_ID NUMBER (*,0)
F COURSE_ID NUMBER (*,0)
F STUDENT_ID NUMBER (*,0)
TYPE VARCHAR2 (255 CHAR)
RECORDED DATE
F COURSES_ID NUMBER (19)
GRADES_PK (GRADE_ID)
FK3SHYQIL15UEBAWFMMESTYU7LE (COURSES_ID)
GRADES_COURSE_FK (COURSE_ID)
GRADES_LECTURER_FK (LECTURER_ID)
GRADES_STUDENT_FK (STUDENT_ID)
ADMIN.LECTURERS
PF * USER_ID NUMBER (*,0)
ACADEMIC_TITLE VARCHAR2 (255 CHAR)
LECTURERS_PK (USER_ID)
LECTURERS_USERS_FK (USER_ID)
ADMIN.USERS
P * USER_ID NUMBER (*,0)
FIRST_NAME VARCHAR2 (255 CHAR)
LAST_NAME VARCHAR2 (255 CHAR)
EMAIL VARCHAR2 (255 CHAR)
PASSWORD VARCHAR2 (255 BYTE)
F ROLE_ID NUMBER (19)
USERS_PK (USER_ID)
FKP56C1712K691LHSYBWCSSF40F (ROLE_ID)
ADMIN.ROLES
P * ROLE_ID NUMBER (*,0)
U * ROLE_NAME VARCHAR2 (255 CHAR)
ROLES_PK (ROLE_ID)
ROLES_ROLE_NAME_UN (ROLE_NAME)
ADMIN.COURSES_SPECIAL
PF * COURSE_ID NUMBER (*,0)
PF * SPECIALIZATION_ID NUMBER (*,0)
COURSES_SPECIAL_PK (COURSE_ID, SPECIALIZATION_ID)
SYS_C0055648 (COURSE_ID)
SYS_C0055649 (SPECIALIZATION_ID)
ADMIN.COURSE_LECTURERS
PF * COURSE_ID NUMBER (*,0)
PF * LECTURER_ID NUMBER (*,0)
COURSE_LECTURERS_PK (COURSE_ID, LECTURER_ID)
COURSE_LECTURERS_COURSE_FK (COURSE_ID)
COURSE_LECTURERS_LECTURER_FK (LECTURER_ID)
ADMIN.COURSE_REGISTRATIONS
F COURSE_ID NUMBER (19)
* COURSE_REG_ID NUMBER (19)
STATUS VARCHAR2 (255 CHAR)
F STUDENT_ID NUMBER (19)
FK1UITFC1H86QOT7NCNHLYL8VAB (STUDENT_ID)
FK2YPYCU195RNGWBGYVLQLACWLY (COURSE_ID)
ADMIN.COURSES
P * COURSE_ID NUMBER (*,0)
U NAME VARCHAR2 (255 CHAR)
ECTS_POINTS NUMBER (*,0)
STATUS VARCHAR2 (255 CHAR)
COURSES_PK (COURSE_ID)
UNIQUE_COURSE_NAME (NAME)
UNIQUE_COURSE_NAME (NAME)
ADMIN.STUDENTS
PF * USER_ID NUMBER (*,0)
U STUDENT_NUMBER VARCHAR2 (255 CHAR)
STATUS VARCHAR2 (255 CHAR)
LIBRARY_CARD_NUMBER VARCHAR2 (255 CHAR)
STUDENTS_PK (USER_ID)
STUDENTS_STUDENT_NUMBER_UN (STUDENT_NUMBER)
STUDENTS_USERS_FK (USER_ID)
ADMIN.CLASSES
P * CLASS_ID NUMBER (*,0)
DAY_OF_WEEK NUMBER (*,0)
GROUP_NR NUMBER (*,0)
F COURSE_ID NUMBER (*,0)
CLASSROOM NUMBER (*,0)
CLASS_TYPE VARCHAR2 (50 BYTE)
START_TIME TIMESTAMP
END_TIME TIMESTAMP
CLASSES_PK (CLASS_ID)
CLASSES_COURSE_FK (COURSE_ID)
ADMIN.COURSE_REQUIREMENT
PF * COURSE_ID NUMBER (*,0)
PF * COMPLETION_REQ_ID NUMBER (*,0)
COURSE_REQUIREMENT_PK (COURSE_ID, COMPLETION_REQ_ID)
FK1DUCBJUDDWD9XAU7DSRKH76FO (COMPLETION_REQ_ID)
SYS_C0055645 (COURSE_ID)
ADMIN.ENROLL_REQUIREMENTS
P * REG_ID NUMBER (19)
COURSE_ID NUMBER (19)
MIN_ECTS NUMBER (10)
UF COMPLITED_COURSE_ID NUMBER (19)
ENROLL_REQUIREMENTS_PK (REG_ID)
UKPI8STBYNHEFQ7WAIBTJBUOXTE (COMPLITED_COURSE_ID)
FKAWTOO8CIDF19K11QH5PQTDP9V (COMPLITED_COURSE_ID)
UKPI8STBYNHEFQ7WAIBTJBUOXTE (COMPLITED_COURSE_ID)
ADMIN.COMPLETION_REQUIREMENTS
PF * COMPLETION_REQ_ID NUMBER (19)
* MIN_SCORE NUMBER (10)
* TYPE VARCHAR2 (255 CHAR)
COMPLETION_REQUIREMENTS_PK (COMPLETION_REQ_ID)
FKMAMHW870418CK8X1N76ML3B11 (COMPLETION_REQ_ID)
ADMIN.SPECIALIZATIONS
P * SPECIALIZATION_ID NUMBER (*,0)
NAME VARCHAR2 (100 BYTE)
SPECIALIZATIONS_PK (SPECIALIZATION_ID)
ADMIN.STUDENT_SPECIALIZATIONS
PF * SPECIALIZATION_ID NUMBER (*,0)
PF * STUDENT_ID NUMBER (*,0)
STUDENT_SPECIALIZATIONS_PK (SPECIALIZATION_ID, STUDENT_ID)
STUDENT_SPECIALIZATIONS_SPEC_FK (SPECIALIZATION_ID)
STUDENT_SPECIALIZATIONS_STUDENT_FK (STUDENT_ID)
ADMIN.SEMESTER_ECTS
P * SEMESTER_ECTS_ID NUMBER (*,0)
SEMESTER NUMBER (*,0)
ECTS_AMOUNT NUMBER (*,0)
F SPECIALIZATION_ID NUMBER (*,0)
SEMESTER_ECTS_PK (SEMESTER_ECTS_ID)
SEMESTER_ECTS_SPECIALIZATION_FK (SPECIALIZATION_ID)

CLASSES

Represents specific classes (e.g. lectures or labs) associated with a course

- CLASS_ID: Unique identifier of the class
- DAY_OF_WEEK: Numeric value representing the day of the week
- GROUP_NR: Group number the class is assigned to
- COURSE_ID: Reference to the related course
- CLASSROOM: Numeric identifier of the classroom
- CLASS_TYPE: Type of class (e.g. lecture, lab)
- START_TIME, END_TIME: Time range when the class takes place

COMPLETION_REQUIREMENTS

Stores requirements needed to pass a course, like minimum scores for exams or labs

- COMPLETION_REQ_ID: Unique identifier for the requirement
- MIN_SCORE: Minimum score required
- TYPE: Type of requirement (e.g. "exam", "lab")

COURSES

Main entity representing an academic course

- COURSE_ID: Unique identifier of the course
- NAME: Course name
- ECTS_POINTS: Number of ECTS credits awarded for the course
- STATUS: Status of the course (e.g. "active", "archived")

USERS

Storing all user accounts in the system

- USER_ID: primary key
- FIRST_NAME, LAST_NAME: personal data
- EMAIL, PASSWORD: login credentials

LECTURERS

Specialized users who teach courses

- USER_ID: primary key (same as in USERS)
- ACADEMIC_TITLE: e.g., "Dr", "Prof."

STUDENTS

Specialized users who take courses

- USER_ID: primary key (same as in USERS)
- STUDENT_NUMBER: unique student ID
- STATUS: e.g., “ACTIVE”
- LIBRARY_CARD_NUMBER: student’s library card

ROLES

Defines available system roles

- ROLE_ID: primary key
- ROLE_NAME: ROLE_STUDENT, ROLE_LECTURER, ROLE_ADMIN

GRADES

Stores grades assigned to students

- GRADE_ID: primary key
- GRADE: numeric value
- RECEIVED_DATE, RECORDED: timestamps

- TYPE: e.g., "Exam", "Project
- Foreign keys: LECTURER_ID, COURSE_ID, STUDENT_ID

SPECIALIZATIONS

Represents fields of study

- SPECIALIZATION_ID: primary key
- NAME: specialization name

SEMESTER_ECTS

Defines ECTS quotas per specialization and semester

- SEMESTER_ECTS_ID: primary key
- SEMESTER: number
- ECTS_AMOUNT: required points

ENROLL_REQUIREMENTS

Describes requirements to enroll in a course

- REG_ID: primary key
- COURSE_ID: the target course
- MIN_ECTS: required ECTS
- COMPLITED_COURSE_ID: optional prerequisite course

COURSE_REGISTRATIONS

Tracks which students are registered for which courses

- COURSE_REG_ID: primary key
- STATUS: e.g., "REGISTRED"

USERS – ROLES

- Relation: Many-to-One
- Description: Each user has exactly one role. A role can be assigned to many users.

USERS – LECTURERS / STUDENTS

- Relation: One-to-One (specialization / inheritance)
- Description: A user can be either a lecturer or a student

STUDENTS – COURSE_REGISTRATIONS

- Relation: One-to-Many
- Description: A student can register for multiple courses

STUDENTS – STUDENT_SPECIALIZATIONS – SPECIALIZATIONS

- Relation: Many-to-Many
- Description: A student can belong to multiple specializations; a specialization can have multiple students

STUDENTS – GRADES

- Relation: One-to-Many
- Description: A student can have multiple grades

COURSES – COURSE_REGISTRATIONS

- Relation: One-to-Many
- Description: A course can have many student registrations

COURSES – GRADES

- Relation: One-to-Many
- Description: A course can have many grades

COURSES – CLASSES
- Relation: One-to-Many
- Description: A course can have multiple scheduled classes

COURSES – COURSE_LECTURERS – LECTURERS
- Relation: Many-to-Many
- Description: A course may be taught by several lecturers; a lecturer may teach many courses

COURSES – COURSE_REQUIREMENT – COMPLETION_REQUIREMENTS
- Relation: Many-to-Many
- Description: A course may require multiple grading components (e.g., exam, project)

COURSES – ENROLL_REQUIREMENTS
- Relation: One-to-Many
- Description: A course may define several enrollment conditions (e.g., min ECTS or prerequisite course)

COURSES – COURSES_SPECIAL – SPECIALIZATIONS
- Relation: Many-to-Many
- Description: A course can belong to one or more specializations

SPECIALIZATIONS – SEMESTER_ECTS
- Relation: One-to-Many
- Description: Each specialization defines ECTS quotas per semester

LECTURERS – GRADES
- Relation: One-to-Many
- Description: A lecturer may assign many grades

Use Case Scenarios

Student

The student logs in to the system to view a summary of final grades for all enrolled courses. They can also access detailed scores from individual assignments, labs, or exams within each course. Before the registration deadline, the student may withdraw from a selected course. Additionally, students have access to graphical reports presenting their academic performance

Lecturer

The lecturer enters or updates grades for students enrolled in their courses. They can view a breakdown of each student's progress and earn points within a specific course. During course setup, the lecturer defines completion requirements (e.g., labs, exams, and their weights). After the semester ends, the lecturer closes the course and initiates automatic final grade calculation. Additionally, lecturers can access graphical reports to analyze overall performance and grade distribution

Admin

Opens or closes the current semester and course registration period

Logic Rules

- Only registered users can access the system after successful login
- A student can register for a course only if the registration period is open and the course prerequisites are met
- A student can deregister from a course only before the registration closes
- A lecturer can assign or modify grades only for students enrolled in their courses
- A course is considered complete only if all completion requirements are met (e.g., lab scores, exam passed)
- Final grades are calculated automatically based on predefined course structure and achieved scores
- Only users with Admin role can open/close semesters
- Students can view only their own grades. Lecturers can view grades of their assigned students