

# 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:

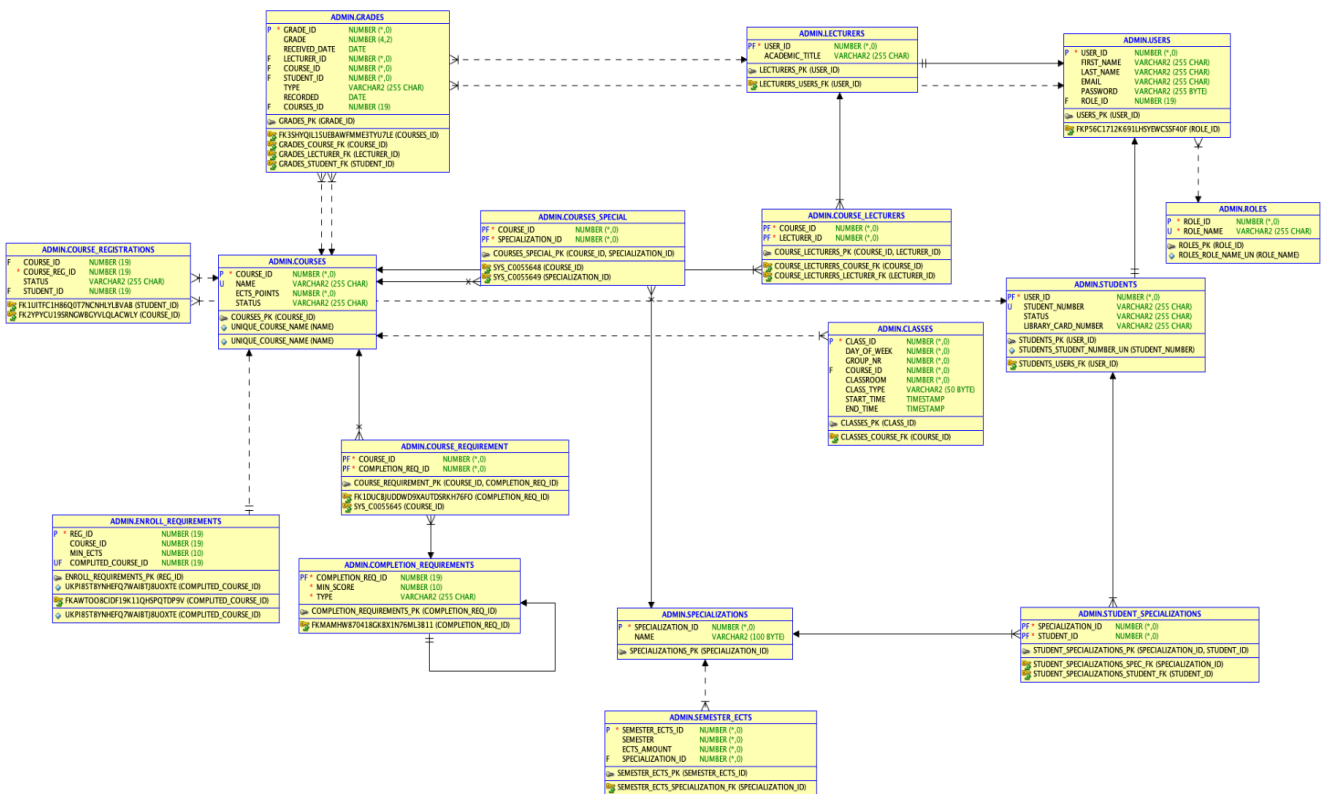
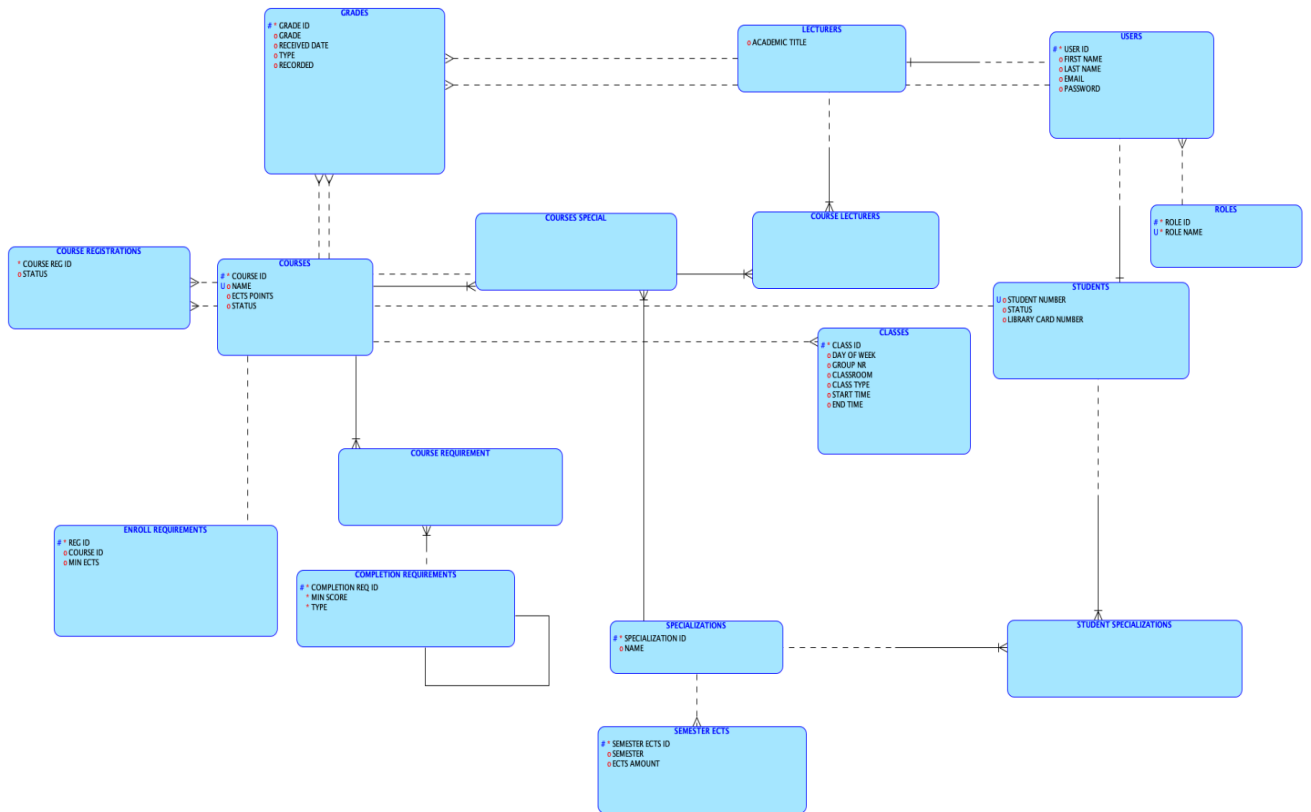
- Assigning grades to students
- Viewing student performance in a given course
- Defining course structure (e.g., number and weight of labs, presence and weight of final exam, minimum points required to pass)
- Closing a course and automatically calculating final grades
- Opening and closing the semester

### 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



## 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\_LLECTURER, 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
- COMPLETED\_COURSE\_ID: optional prerequisite course

### COURSE\_REGISTRATIONS

Tracks which students are registered for which courses

- COURSE\_REG\_ID: primary key
- STATUS: e.g., "REGISTERED"

### 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

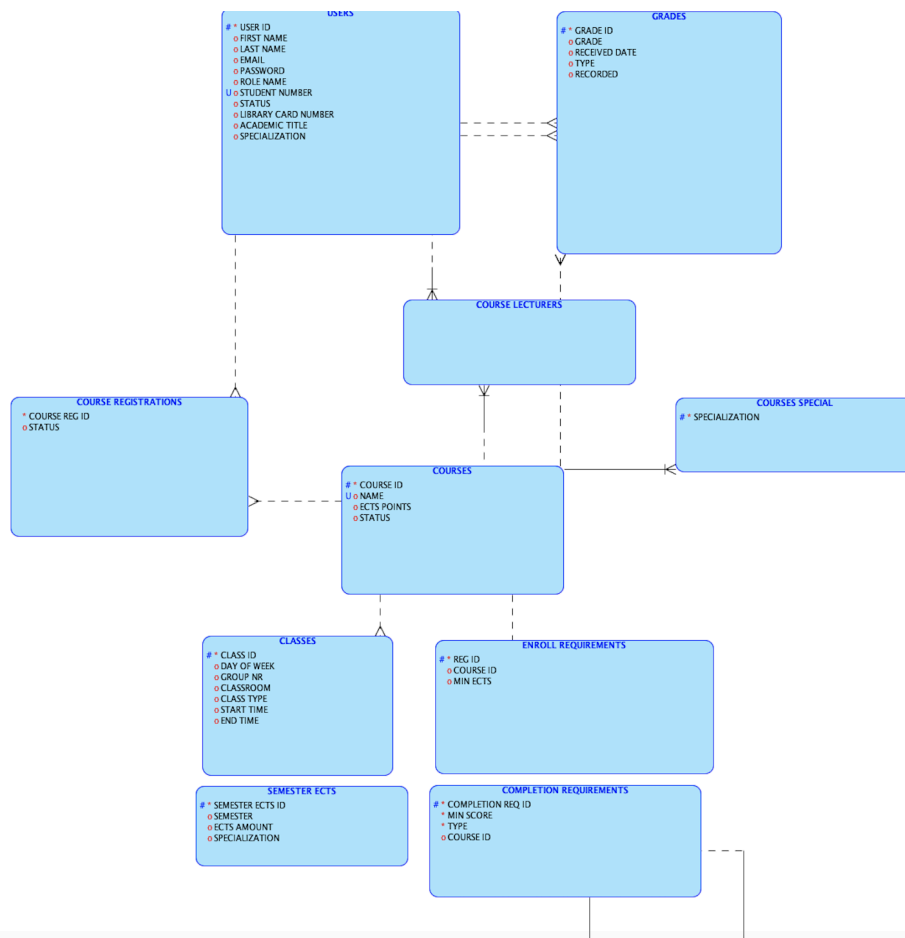
#### SPECIALIZATIONS – SEMESTER\_ECTS

- 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

### **Optimization**



## Indexes

Users:

idx\_users\_last\_name – speeds up searches and sorting by last name

idx\_users\_email — optimizes lookups by user email (login)

`idx_users_role_name` — improves filtering by user role

idx\_u\_email\_role — supports combined filtering by email and role

`idx_u uid role` — helps in queries joined by user ID and role

Grades:

`idx_grades_type` — improves queries filtering by grade type (e.g. “FINAL”)

idx grades course id — speeds up access to grades by course

`idx_grades_student_id` — used for retrieving grades by student

idx\_grades\_cid\_type — improves multi-condition filtering

idx\_grades\_sid\_cid — used in grade lookups per student/course pair

Course registrations:

idx\_c\_reg\_student\_id — for retrieving all course registrations of a student

idx\_c reg\_course\_id — for retrieving all registrations for a course

idx\_c\_reg\_status — supports filtering registrations by status (e.g. “OPEN”)

idx\_c reg\_sid status — efficient filtering by student and status together

Courses:

idx courses status — improves filtering courses by their status

`idx_courses_cid_status` — supports combined filtering by course ID and status

Completion requirements:

idx c req course id — optimizes access to completion requirements for a course

Enroll\_requirements:

idx e req course id — speeds up retrieval of enrollment conditions for a course

*The optimization introduced a set of targeted indexes that significantly reduced full table scans and improved query performance. Most queries now benefit from index range scans on key columns like course\_id, email, and student\_id, resulting in lower execution costs and faster response times. Composite indexes further enhance filtering efficiency in multi-condition queries*

## **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