

Software Engineering Module Project

Project Context: University Management System “SmartCampus - UBMA”

Badji Mokhtar University - Annaba is taking steps to bring all its academic and administrative processes together into one modern, unified platform: **SmartCampus - UBMA**. The goal is to replace scattered tools with a single, easy-to-use system that supports every part of university life. This platform will handle:

- **Academic Registration:** Signing up for courses and programs
- **Schooling and Assessment:** Recording and checking grades, tracking student progress
- **Library Management:** Borrowing and returning books and resources
- **Space Reservation:** Booking classrooms, lecture halls, and study areas
- **Planning:** Creating and viewing timetables for students and instructors

Work Required

Rather than keeping five separate systems, we need one integrated solution. Using **UML 2** and **OCL**, you'll model the entire system, covering:

- **Structural aspects:** class diagrams, components, deployment
- **Dynamic behavior:** use cases, sequences, activities, state transitions
- **Functional needs:** business rules, constraints, user interfaces

Session 1: Use Case Diagram

Goal: Pinpoint the system's users and core functions

Tasks:

1. List every type of user (actor) in the system
2. Identify key use cases for each module:
 - Registration Module: Register for a course, View available courses
 - School Module: Enter grades, Check report card
 - Library Module: Borrow a book, Reserve a document
 - Reservation Module: Reserve a room, Cancel a reservation
 - Planning Module: Generate a schedule, Modify a time slot
3. Define relationships between use cases (include, extend)
4. Write a full scenario for "Register for a course"

Session 2: Class Diagram

Goal: Organize key concepts and show how they relate

Tasks:

- Identify main classes for: People, Pedagogy, Library, Reservation, Schedule
- Draw the corresponding class diagram

Session 3: OCL Constraints

Goal: Write clear business rules using OCL

Tasks:

Pick 5 of the following 8 constraints and write them in OCL. Make sure they fit well with your class diagram.

Registrations:

- A student can't register for more than 6 courses per semester
- A course can't have more than 30 students

Library:

- A student may not borrow more than 5 books at once

- An unavailable book cannot be borrowed

Reservation:

- A room can only be reserved during opening hours
- A teacher can only reserve one room at a time

Education:

- Only the course teacher can enter grades
- A score must be between 0 and 20

Session 4: Sequence and Activity Diagrams

Goal: Model how users interact with the system and how processes flow

Tasks:

1. Create sequence diagrams for:
 - “Borrow a book” (library module)
 - “Register for a course” (registration module)
 - Include what happens when things go wrong (refusals, errors)
2. Draw activity diagrams for these processes:
 - The full room booking process
 - How grades are managed (from entry → validation → publication)
 - Use decision points and parallel actions where they make sense

Session 5: State-Transition and Deployment Diagrams

Goal: Show how objects change over time and how the system is set up physically

Tasks:

1. Draw state-transition diagrams for:
 - Course statuses: Scheduled, Open, Full, Closed, Cancelled
 - Loan statuses: Ongoing, Late, Returned, Lost
 - Reservation statuses: Pending, Confirmed, Cancelled, Completed
2. Create a deployment diagram:
 - Show all components and how they connect
 - List the deployed artifacts (files, applications) for each module