**EX NO: 5**

**DATE: 21.02.2025**

**USING THE IDENTIFIED SCENARIOS, FIND THE INTERACTION BETWEEN OBJECTS AND REPRESENT THEM USING UML SEQUENCE AND COLLABORATION DIAGRAMS**

**AIM :**

**Key Objects:**

* User (Admin, Faculty, Student)
* Account (User credentials, roles)
* Course (Course details, schedules)
* StudentRecord (Performance, attendance)
* FacultyRecord (Assigned courses, activities)
* Database (Backup, restore)
* Report (Performance, attendance, user activity)

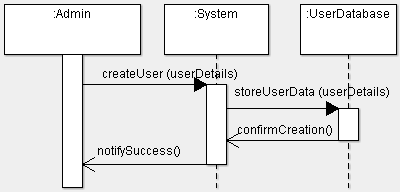
**Object Interactions & UML Diagrams:**

**1. User Account Management (R1)**

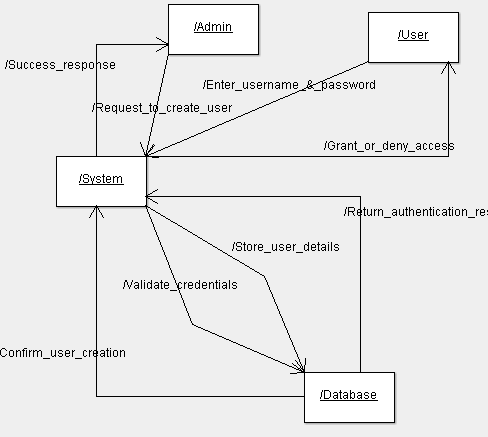
**Interactions:**

* Admin → System → createUser(userDetails)
* System → UserDatabase → storeUserData(userDetails)
* UserDatabase → System → confirmCreation()
* System → Admin → notifySuccess()

Sequence Diagram:



Collaboration Diagram:

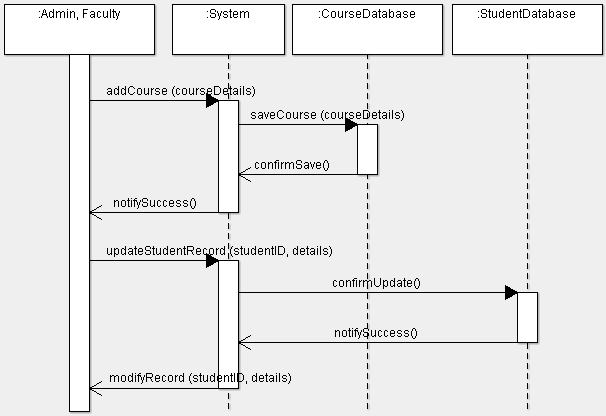


**2. Student, Faculty, and Course Management (R2)**

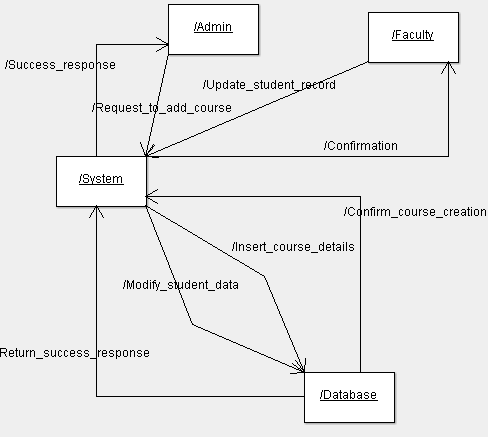
**Interactions:**

* Admin / Faculty → System → addCourse(courseDetails)
* System → CourseDatabase → saveCourse(courseDetails)
* CourseDatabase → System → confirmSave()
* Admin / Faculty → System → updateStudentRecord(studentID, details)
* System → StudentDatabase → modifyRecord(studentID, details)
* StudentDatabase → System → confirmUpdate()

Sequence Diagram:



Collaboration Diagram:

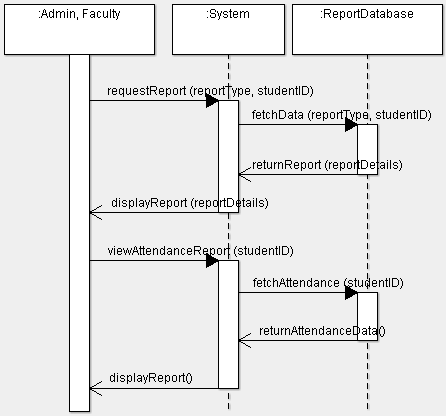


3. Report Generation (R4)

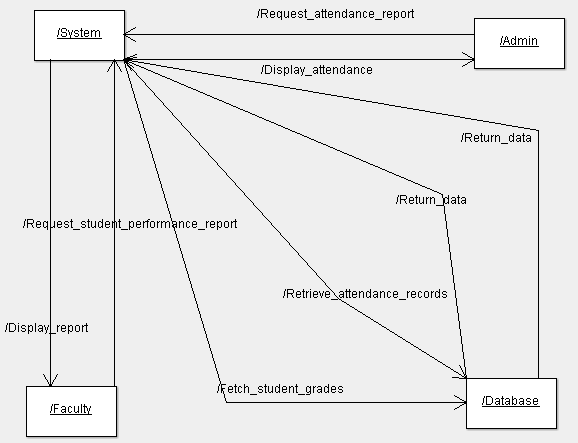
Interactions:

* Admin / Faculty → System → requestReport(reportType, studentID)
* System → ReportDatabase → fetchData(reportType, studentID)
* ReportDatabase → System → returnReport(reportDetails)
* System → Admin / Faculty → displayReport(reportDetails)
* Faculty → System → viewAttendanceReport(studentID)
* System → ReportDatabase → fetchAttendance(studentID)
* ReportDatabase → System → returnAttendanceData()

Sequence Diagram:



Collaboration Diagram:

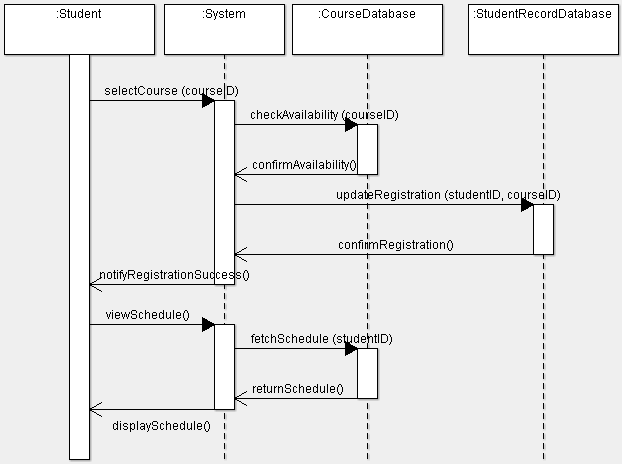


4. Course Registration (R6)

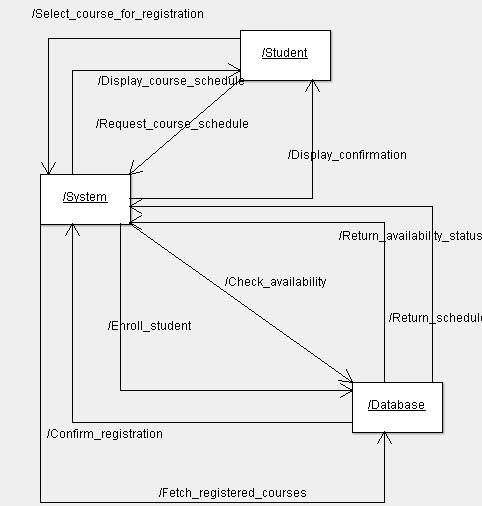
Interactions:

* Student → System → selectCourse(courseID)
* System → CourseDatabase → checkAvailability(courseID)
* CourseDatabase → System → confirmAvailability()
* System → StudentRecordDatabase → updateRegistration(studentID, courseID)
* StudentRecordDatabase → System → confirmRegistration()
* System → Student → notifyRegistrationSuccess()
* Student → System → viewSchedule()
* System → CourseDatabase → fetchSchedule(studentID)
* CourseDatabase → System → returnSchedule()

Sequence Diagram:



Collaboration Diagram:



**RESULT:**