NATIONAL UNIVERSITY OF COMPUTING AND EMERGING SCIENCES

ISLAMABAD

Submitted to

DR. Shujaat Hussain

Department of Computing

Database Systems

PROJECT REPORT

GROUP MEMBERS:

ALI UMER (21I-0380)

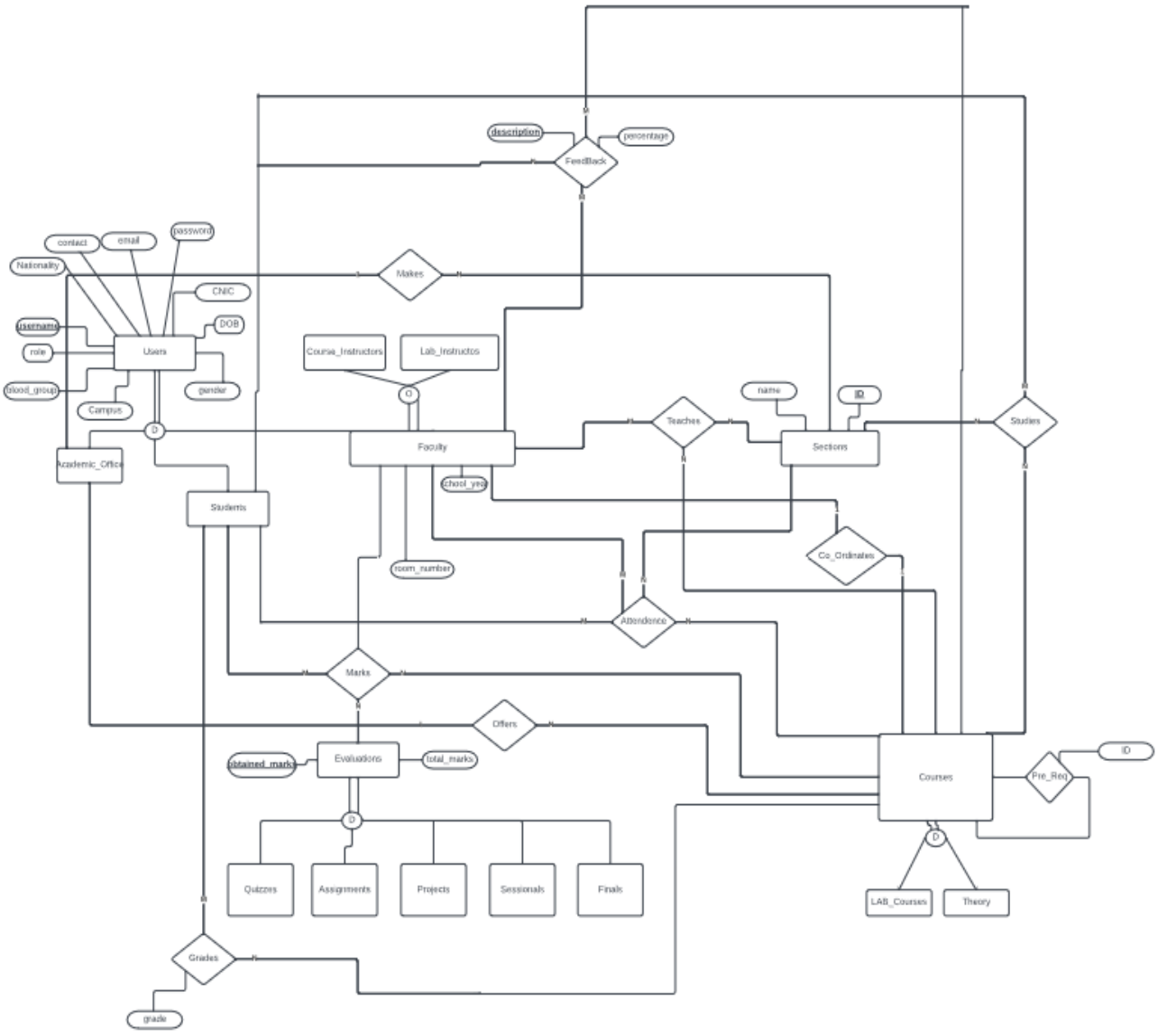
UMAIR KHALID (21I-0455)

ABDULLAH MASOOD (21I-0822)

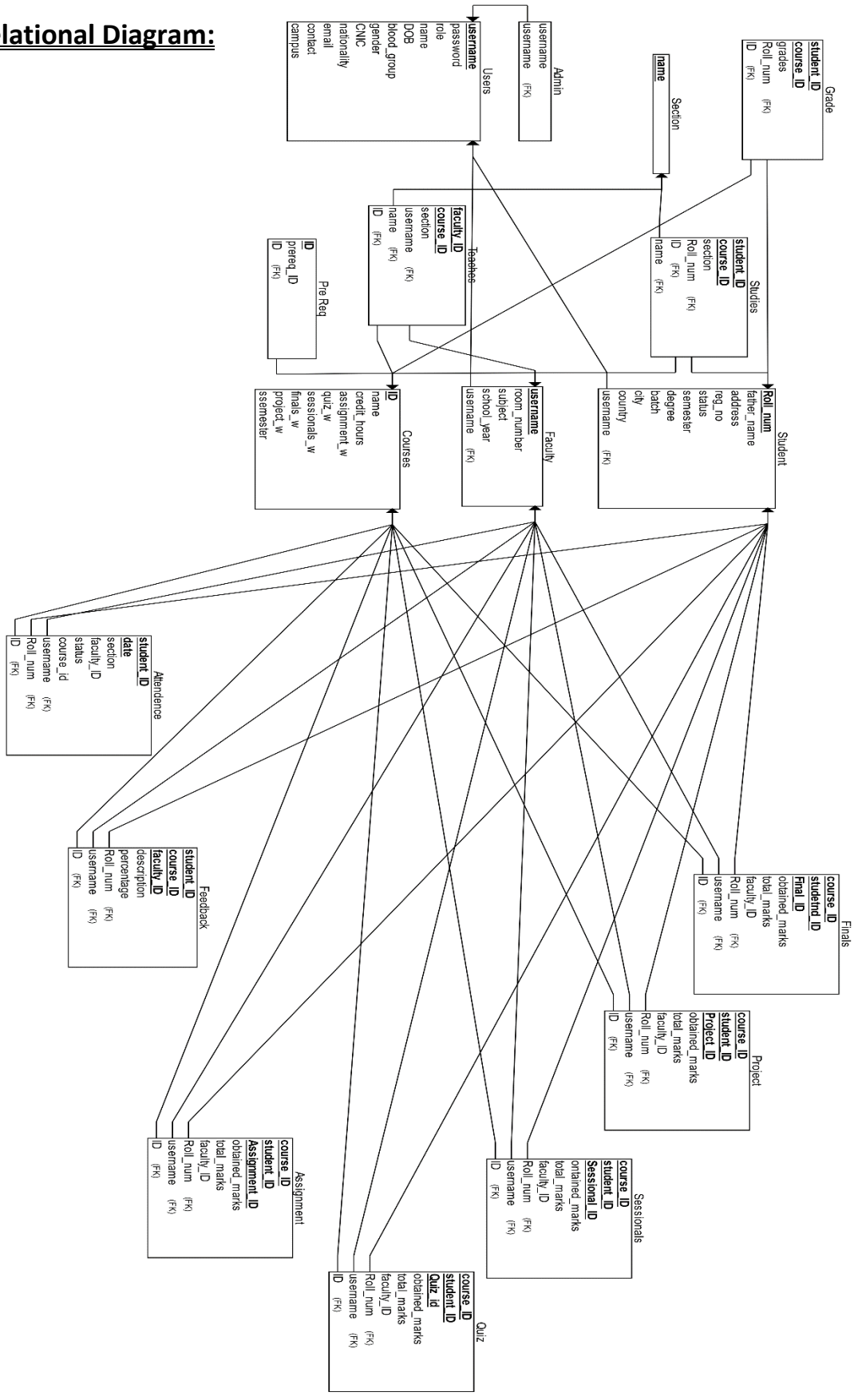
Introduction:

A FLEX Management System is a Database that is used to manage and schedule university students' academic information. There are 3 main types of users in FLEX **Student, Admin, and Faculty.** This database allows students faculty and admin to manage and interact with one another. This database consists of tables that represent different aspects of the university management system. The tables have been designed to store information related to students, faculty, courses, grades, attendance, quizzes, assignments, sessionals, projects, feedback, and sections.

ERD:



Relational Diagram:



Users Table:

The "users" table is the parent table, which stores the common attributes of all users, including their username, password, role, name, DOB, blood group, gender, CNIC, nationality, email, contact, and campus. The username column serves as the primary key for this table. The "admin" and "faculty" tables are child tables of the "users" table, with the username column serving as the foreign key in both tables. The Users table has a one-to-one relationship with Admin, Faculty, and Student.

Admin Table:

The “admin” table includes information about the administrators who manage the university's academic system. The primary key for this table is the Admins Username and also has a Foreign key that references with the Users table. Further, the Admin has a one-to-one relationship with the Users table.

Faculty Table:

The “faculty” table contains details about the university's faculty, including their usernames, room numbers, subjects, and school year. The primary key for this table is the Faculty’s Username, and also has a Foreign key that references with the Users table. Furthermore, the Faculty has a one-to-one relationship with the Users table.

Students Table:

The "students" table stores information related to the students, including their Roll_num, father_name, address, reg_no, section, status, semester, degree, batch, city, and country. The Roll_num column serves as the primary key, and the section column serves as the foreign key, referencing the "sections" table. The Students table has a one-to-one relationship with Section Table and a one-to-one Realtion with Users table.

Sections Table:

The "sections" table stores information related to the sections of different courses offered by the university. It has only one column, "name," which serves as the primary key.

Courses Table:

The "courses" table stores information related to the courses offered by the university, including their ID, name, credit_hours, assignment_w, quiz_w, sessionals_w, finals_w, project_w, and semester. The ID column serves as the primary key.

Pre-req Table:

The "prereq" table stores information related to the prerequisites of a course. It has two columns, "ID" and "prereq_ID," both of which serve as foreign keys referencing the "courses" table. The prereq table has a many-to-many relationship with the Courses table.

Studies Table:

The "studies" table stores information related to the enrollment of a student in a course, including their student_ID, course_ID, and section. The primary key is a combination of the student_ID and course_ID columns, and the section column serves as a foreign key referencing the "sections" table. The studies table has a many-to-many relationship with the Courses and Students tables.

Teaches Table:

The "teaches" table stores information related to the courses being taught by the faculty, including their faculty_ID, section, and course_ID. The primary key is a combination of the faculty_ID and course_ID columns, and both the section and course_ID columns serve as foreign keys referencing the "sections" and "courses" tables, respectively. The teaches table has a many-to-many relationship with the Courses and Faculty tables.

Grades Table:

The "grades" table stores information related to the grades obtained by the students in different courses, including their student_ID, course_ID, and grade. The primary key is a combination of the student_ID and course_ID columns, and both columns serve as foreign keys referencing the "students" and "courses" tables, respectively. The grades table has a many-to-many relationship with the Students and Courses tables.

Attendance Table:

The "attendance" table stores information related to the attendance of the students in different courses, including their student_ID, date, section, faculty_ID, status, and course_id. The primary key is a combination of the student_ID and date columns, and all other columns serve as foreign keys referencing the "students," "sections," "faculty," and "courses" tables, respectively. The attendance table has a many-to-many relationship with the Students, Faculty, Courses and Sections tables.

Quiz, Assignment, Sessionals, Projects, Final Table:

The "quiz," "assignment," "sessionals," "project," and "final" tables store information related to the quizzes, assignments, sessionals, projects, and finals of different courses, respectively. Each

table has columns for obtained_marks, total_marks, course_ID, student_ID, and faculty_ID. All these columns serve as foreign keys referencing the "courses," "students," and "faculty" tables, respectively. These tables has same types of relations with same tables; so the "quiz," "assignment," "sessionals," "project," and "final" table have a many-to-many relationship with the Students, Faculty, and Courses table.

Feedback Table:

Finally, the "feedback" table stores information related to the feedback provided by the students about different courses and faculty. It has columns for student_ID, course_ID, faculty_ID, description, and percentage, with the primary key being a combination of all these. The Feedback Table has one-to-many relation with Faculty, Students, and Course Table.