Business Requirements Document

1. Project Overview

Project Title: University Management Database System

Team Members:

- Mostafa Karam (23010064) **[Team Leader] -----> GUI**
- Yossef Hassan Ali (23010059) ------>Report
- Ziad Rami Fikry (23010057) ----> SQL FILE
- Seif El-Din Haytham Ahmed (23010053) ------> SQL FILE
- Mahmoud Samir Said (23010021) -----> SQL FILE
- Adam Mohamed Ibrahim (23010007) ------>ERD
- Mohamed Mokhtar Hussein (23010080) -------> SQL FILE
- Ismail Hesham Ismail (23011024) ---------> ERD

Project Objective:
To create a comprehensive and robust database system for managing university operations effectively. This system is designed to handle critical processes such as student enrollment, course management, professor assignments, faculty oversight, and department-faculty linkage. By leveraging this database, the university aims to improve operational efficiency, minimize redundancy, and ensure reliable, secure, and accessible data for all stakeholders, including administrative staff, faculty members, and students.

Scope:
The system will include modules for students, professors, courses, departments, and faculties, ensuring seamless interactions and relationships between these entities.

Target Audience:

- University administrative staff
- Academic faculty members
- Students

2. Business Processes

The database is designed to support the following core processes:

Process	Description
Student Management	Store and manage detailed student profiles, including personal details, academic records, and enrollment history.
Course Management	Maintain comprehensive course details such as credits, prerequisites, semester scheduling, and assigned professors.
Professor Management	Track professor information, including their qualifications, teaching schedules, and departmental responsibilities.
Department Oversight	Link departments to faculties, oversee department activities, and manage departmental leadership assignments.
Enrollment Tracking	Record and analyze student enrollment in courses, tracking academic performance, grades, and attendance.
Faculty Oversight	Manage faculties, their associated departments, and ensure proper allocation of resources and leadership roles.
Performance Reporting	Generate reports on academic performance, enrollment trends, course popularity, and faculty resource allocation.
Data Integrity Checks	Implement validation mechanisms to ensure accurate and consistent data entry across all processes.

3. Functional Requirements

Requirement ID	Description
FR_01	Store detailed student information, including ID, name, email, phone number, address, and department affiliation.
FR_02	Record detailed course information such as ID, name, credits, assigned professors, prerequisites, and semester availability.
FR_03	Manage student enrollment records, including student progress, academic history, and grades.
FR_04	Maintain comprehensive professor details, including their academic credentials, specializations, and teaching schedules.
FR_05	Establish hierarchical relationships between faculties, departments, courses, and administrative units.
FR_06	Enable secure role-based access control for database interaction, ensuring proper permissions for students, faculty, and administrators.
FR_07	Generate dynamic reports on academic performance, enrollment trends, and departmental activities.
FR_08	Support data auditing and tracking to maintain data integrity and compliance with university policies.

4. SQL Analysis

The provided SQL script implements the **University** database, featuring the following key components:

Entities and Tables

1. **Faculty**:
    - **FacultyID**: Primary Key, uniquely identifies each faculty.
    - **FacultyName**: Name of the faculty, capturing descriptive titles such as 'Engineering' or 'Humanities.'
    - **HeadProfessorID**: Foreign Key linking to the Professor table, designating the professor who oversees the faculty.
    - Additional Details: Faculties are administrative units overseeing several departments. Each faculty has a dedicated head professor and can share resources across departments to enhance collaboration and academic management.

2. **Department**:
    - **DepartmentID**: Primary Key, uniquely identifies each department.
    - **DepartmentName:** The official name of the department, reflecting its academic or operational focus.
    - **FacultyID:** Foreign Key linking to Faculty, defining its parent faculty.
    - **HeadProfessorID**: Foreign Key linking to Professor, identifying the head of the department.
    - Additional Details: Departments serve as specialized academic units within faculties, managing courses, professors, and student enrollment at a focused level.

3. **Professor**:
    - **ProfessorID:** Primary Key, uniquely identifies each professor.
    - **ProfessorName**: Full name of the professor.
    - **ProfessorTitle:** Title or rank of the professor (e.g., Associate Professor, Lecturer).
    - **DepartmentID:** Foreign Key linking to Department, specifying the academic unit they are part of.
    - Additional Details: Professors are key stakeholders responsible for course instruction, research, and academic mentoring. They may also hold administrative roles such as department head or faculty dean.

4. **Course**:
    - **CourseID**: Primary Key, uniquely identifies each course.
    - **CourseName**: Title of the course.
    - **CourseCredit:** Number of credit hours assigned to the course.
    - **ProfessorID:** Foreign Key linking to Professor, specifying the instructor.
    - Additional Details: Courses are structured academic programs offered by departments, aligned with degree requirements and learning outcomes. Each course is linked to a professor who is responsible for teaching and evaluation.

5. **Student**:
    - **StudentID**: Primary Key, uniquely identifies each student.
    - **StudentName**: Full name of the student.
    - **Email**: Contact email of the student.
    - **DepartmentID:** Foreign Key linking to Department, specifying their academic unit.
    - Additional Details: Students represent the primary users of the university system, with attributes capturing academic progress, enrollment status, and personal data for effective management.

6. **Enrollment**:
    - **EnrollmentID:** Primary Key, uniquely identifies each enrollment record.
    - **StudentID:** Foreign Key linking to Student, specifying the participant.
    - **CourseID:** Foreign Key linking to Course, identifying the course enrolled in.
    - **Grade**: Grade received in the course.
    - Additional Details: Enrollment represents the transactional link between students and courses, tracking academic performance and participation over time.

Relationships

- **Faculty** ↔ **Department**: Faculties consist of multiple departments, establishing a hierarchical structure.
- **Department** ↔ **Professor**: Departments manage professors who contribute to academic and research activities.
- **Professor** ↔ **Course**: Professors are assigned to teach courses, ensuring proper delivery of academic content.
- **Student** ↔ **Department**: Students are associated with departments that guide their academic journey.
- **Student** ↔ **Course**: Students enroll in courses through the Enrollment table, tracking their academic engagement and performance.

5. ER Diagram Alignment

The provided ER diagram reflects the database design and highlights the following details:

- **Entities**:
  - Faculty: Represents the major organizational units within the university.
  - Department: Academic units operating under faculties.
  - Professor: Academic staff members teaching or managing departments.
  - Course: Subjects offered by departments for students.
  - Student: Individuals enrolled in the university, managed by departments.
  - Enrollment: Tracks which students are registered in specific courses.
- **Attributes**: Key details for each entity include:
  - FacultyName: Names of faculties like "Engineering" or "Arts."
  - ProfessorTitle: Ranks such as Lecturer or Professor.
  - CourseCredit: Indicates the weight or duration of a course.
  - StudentName: Full names of enrolled students.
- **Relationships**:
  - **Belongs To**: Links professors and students to their respective departments.
  - **Enroll**: Tracks student registration in courses and their grades.
  - **Teach**: Specifies which courses are taught by which professors.
  - **Have**: Shows departmental structure within faculties.
- **Normalization**:
  - Ensures minimal data redundancy by organizing tables into third normal form (3NF).
  - For example, the separation of Professor and Department avoids duplicating department details.
- **Data Flow**:
  - **Input**: Data entered for students, courses, and faculty members.
  - **Processing**: Validations such as checking prerequisites during enrollment.
  - **Output**: Reports on academic performance, faculty assignments, and student progress.

This alignment ensures the database adheres to the business requirements and facilitates efficient management of university operations.

The University Management Database System, as depicted in the ER diagram and implemented via the SQL script, meets the functional requirements for managing academic and administrative processes. The system ensures clear relationships between entities, efficient data management, and supports scalability for future enhancements. Potential improvements include indexing for faster query performance and additional constraints to enforce data integrity.