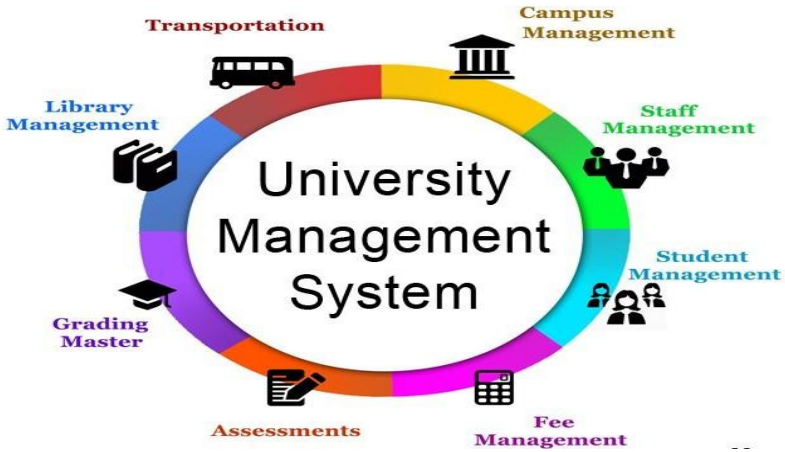
UNIVERSITY MANAGEMENT SYSTEM



LANGUAGE: JAVA

Project Overview:-

The University Management System (UMS) is a comprehensive Java-based application designed to streamline and automate various university processes. It manages students, faculty, courses, departments, and enrollments in a modular and scalable architecture. The system emphasizes clean object-oriented design, extensive file handling, and data persistence.

Technical Features

1. Object-Oriented Architecture

Inheritance:

- Superclass: Person (Subclasses: Student, Faculty, Administrator)

- Superclass: Course (Subclasses: Undergraduate Course, Graduate Course, Online Course)

Interfaces & Abstraction:- Grading System, Login Interface, Fee Payable - used to define common contracts for extensibility.

2. **Polymorphism**

- Method Overloading: enroll Course() method varies by course type and student eligibility.

- Method Overriding: calculate Grade() overridden in different course types based on evaluation criteria.

3. Data Structures

Uses Array List and HashMap to store:

- Registered students and faculty

- Course catalogues

- Enrollment lists

- Fee payment records

4. File Handling (Persistence Layer)

Files used to store data permanently:

- students.txt, faculty.txt, courses.txt, grades.txt

Supports adding/updating/deleting entries, searching for users or courses, and exporting reports (e.g., grade cards, fee receipts).

Modules of the System

MODULE	KEY FUNCTIONALITIES
Student Management	Registration, Login, Course Enrollment, Grade Viewing
Faculty Management	Course Assignment, Grade Submission, Schedule Management
Course Management	Create/Update/Delete Courses, Course Prerequisites
Enrollment System	Add/Drop Courses, Generate Timetable
Fees Management	Fee Calculation, Payment Status, Generate Receipts
Academic Reports	GPA Reports, Attendance Tracking, Transcripts

Advanced Features (Optional Enhancements)

- Login System with role-based access (Student, Faculty, Admin)
- Exception Handling for robust error management
- Swing/JavaFX GUI for graphical interface (optional)
- Unit Testing using JUnit

Expected Outcomes

- A structured, object-oriented university administration platform
- Automation of enrollment, academic tracking, and personnel management
- Secure, persistent storage of academic and personal records
- User-friendly CLI or GUI interface
- Scalable codebase for future expansion (e.g., hostel, transport modules)

TEAM -WARRIORS