Group - 8 (Monday) Batch **Approved by TA - Shreya Jaiswal**
Team Members -
Aksh Talati - 002342012
Bhargav Gandhi - 002310643
Varana Navadiya - 002308207

**Project Proposal**

**1. Problem Statement**

Students, faculty, staff, outside agencies (banking, police & pharmacy) and service providers are all part of the intricate ecosystem in which universities operate. These entities are frequently not completely integrated by current management systems, which results in inefficiencies, redundant data and delays in decision-making. In order to improve service delivery, streamline procedures, and increase coordination, this project suggests a comprehensive University Management System (UMS) that links internal university operations with external businesses (such as banks and police).

**2. Solution**

The following will be managed centrally by the University Management System:

-Core Operations of the University:

· Student Enrollment: Simplifying the registration and enrollment procedures for students.

· Course Management: effective scheduling and updating of curricula and courses.

· Course Management: effective scheduling and updating of curricula and courses.

· Grade Tracking: Making it easier to assess and report academic achievement.

-External Enterprise

· Police: Integration for document verification and emergency incident reporting.

· Bank: Simplifying loan processing, scholarship distribution and student fee payments.

· Pharmacy: Add and management of Medicines

-Key Features

· Data Security: Ensuring strong protection of private university data

· Role-Based Access: To guarantee controlled data access, role-based access assigns rights according to user roles.

· Automation of Routine Tasks: By automating repetitive tasks, manual involvement is reduced.

The solution will maximize the experience for both staff and students, increase departmental collaboration and improve operational efficiency.


**ORGANIZATION**

Enterprise 1: University

• Organization 1: Student Organization  o Role: Students

• Organization 2: Administration Organization o Role: Professors o Role: Staff

Enterprise 2: Police

• Organization 1: Crime Organization  o Role 1: Police Department

• Organization 2: Emergency Organization o Role 1: People

• Organization 3: Document Organization o Role 1: Employees

Enterprise 3: Bank

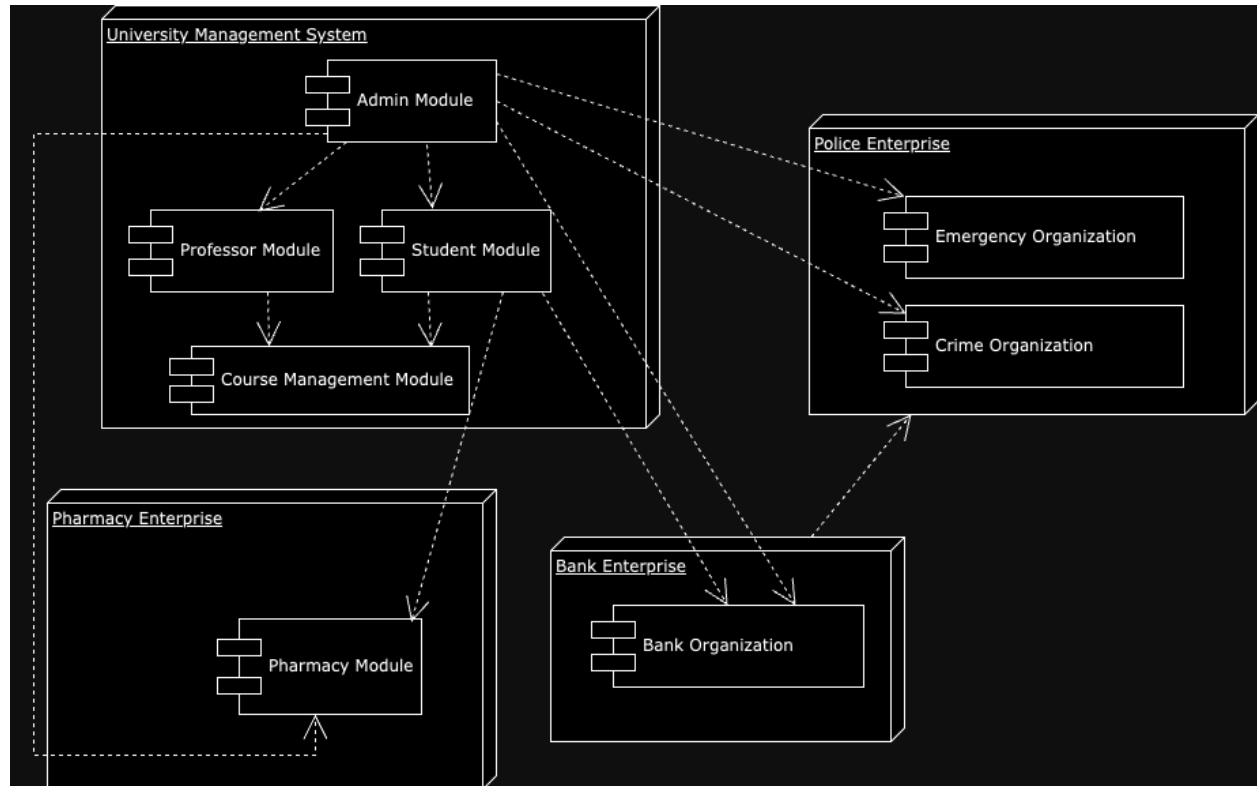• Organization 1: Bank Organization o Role 1: Student  o Role 2: Employees

Enterprise 4: Pharmacy

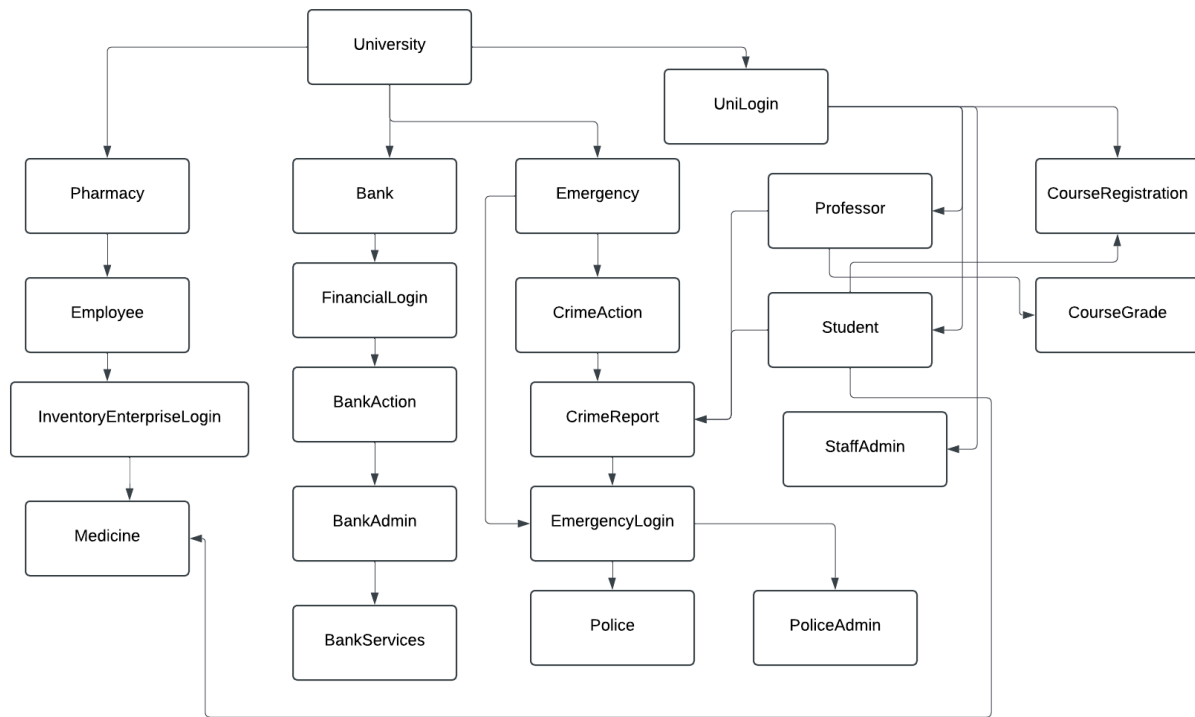• Organization 1: Pharmacy Organization  o Role : Suppliers

**ROLES:**

1. Students

2. Professors

3. University staff

4. Police Department

5. People

6. Employees bank

7. Admin

8. Suppliers

## 3-High-Level Component Diagram:

## 4. UML

# 5. CLASS DIAGRAM

**Pharmacy**

**Employee**
- id: int
- name: String
- age: String
- password: String
- phone: String
- reason: String
- systemAdmin()
- ViewEmployee()
- ViewEmployee()
- initComponents()

**Medicines**
- username: String
- password: String
- role: String
- Medicines()

**InventoryEnterpriseLogin**
- initComponents()
- Employee()
- Medicines()

**PharmacyAdminLogin**
- PharmacyAdminLogin()
- Medicines()
- Runnable()

**EmployeeModel**
- id: int
- name: String
- age: String
- password: String
- phone: String
- reason: String
- EmployeeModel()
- deleteEmployees()
- updateEmployees()
- insertEmployees()
- setGender()
- getGender()
- setPassword()
- getPassword()
- setPhone()
- getPhone()
- setAge()
- getAge()
- setName()
- getName()
- setId()
- getId()

**MedicineModel()**
- company: String
- expiry: String
- production: String
- quantity: String
- price: String
- medicine: String
- getId()
- setId()
- getcompany()
- setcompany()
- getexpiry()
- setexpiry()
- getProduction()
- setProduction()
- getPrice()
- setPrice()
- getQuantity()
- setQuantity()
- insertMedicine()
- updateMedicine()
- deleteMedicine()
- MedicineModel()

**bankAdmin**
- age: int
- phone: int
- gender: String
- name: String
- bankAdmin()
- getUsername()
- setUsername()
- getGender()
- setPassword()
- getPassword()
- setPhone()
- getPhone()
- setAge()
- getAge()
- setName()
- getName()
- setId()

**bankResponse**
- name: String
- accountType: String
- operation: String
- amount: int
- tb1name: String
- tb1accountType: String
- tb1operation: String
- tb1amount: String
- bankResponse()
- initComponents()

**bankServices**
- username: String
- password: String
- accountType: String
- amount: int
- operation: String
- bankServices()
- Runnable()
- run()

**financialLogin**
- financialLogin()
- initComponents()
- Runnable()
- run()

**bankEmployee**
- name: String
- age: String
- username: String
- password: String
- phone: int
- gender: String
- id: int
- bankEmployee()
- getName()
- setName()
- getAge()
- setAge()
- getPhone()
- setPhone()
- setGender()
- getGender()
- getPassword()
- setPassword()
- setPassword()
- getPassword()

**bankResponse**
- name: String
- employee: String
- action: String
- accountType: String
- amount: int
- operation: String
- getName()
- setName()
- getAccountType()
- setAccountType()
- getOperation()
- setOperation()
- getAmount()
- setAmount()
- getEmployee()
- setEmployee()
- getAction()
- setAction()
- addResponse()

**bankServices**
- name: String
- accountType: String
- amount: int
- operation: String
- getName()
- setName()
- getAccountType()
- setAccountType()
- getAmount()
- setAmount()
- getAction()
- setAction()
- addServices()

**JDBCconnection**
- Connect()

**crimeAction**
- name: String
- phone: String
- phone: String
- phone: String
- address: String
- crimeDetails: String
- officer: String
- action: String
- crimeAction()
- Runnable()
- run()

**crimeReport**
- name: String
- phone: String
- action: String
- officer: String
- crimeDetails: String
- address: String
- crimReport()
- Runnable()
- run()

**emergencyLogin**
- username: String
- password: String
- role: String
- emergencyLogin()
- initComponents()
- Runnable()
- run()

**policeAdmin**
- name: String
- email: String
- age: int
- phone: int
- salary: int
- designation: String
- gender: String
- phone: String
- getModel()
- police_table()
- Runnable()
- run()

**police**
- id: int
- gender: String
- phone: int
- password: String
- username: String
- age: String
- name: String
- police()
- getName()
- setName()
- getAge()
- setAge()
- getPassword()
- setPassword()
- getGender()
- setGender()
- setPhone()
- getPhone()
- addPolice()

**crimeaction**
- name: String
- phone: String
- address: String
- crimeDetails: String
- officer: String
- action: String
- crimeaction()
- getName()
- setName()
- getPhone()
- setPhone()
- getAddress()
- setAddress()
- getcrimeDetails()
- setcrimeDetails()
- getOfficer()
- setOfficer()
- getAction()
- setAction()
- addAction()

**crimereport**
- name: String
- phone: String
- address: String
- crimeDetails: String
- crimereport()
- getName()
- setName()
- getPhone()
- setPhone()
- getAddress()
- setAddress()
- getcrimeDetails()
- setcrimeDetails()
- getAction()
- setAction()
- addCrime()

**signUp**
- signUp()
- initComponents()
- Runnable()
- run()

**ProfessorSinglePanel**
- currSubjectTeach: String
- currProfUsername: String
- professor()
- initComponents()
- setProfData()
- CourseGrade()
- CreateCourseGrade()
- Runnable()
- run()

**staffAdmin**
- name: String
- subjectTaken: String
- email: String
- age: int
- studUsername: String
- studPassword: String
- staffAdmin()
- initComponents()
- Students()
- createStudents()
- CreateCourseGrade()
- Runnable()
- run()

**student**
- pname: String
- subject: String
- email: String
- age: int
- Username: String
- profUsername: String
- course: String
- currStudentName: String
- student()
- initComponents()
- CourseReg()
- Runnable()
- run()

**uniLogin**
- role: String
- Username: String
- Password: String
- uniLogin()
- Runnable()
- run()