ERP MANAGEMENT

- **MODULES:-**
  1. STUDENT
  2. PARENT
  3. TEACHER
  4. STORE
  5. LIBRARY
  6. ADMINISTRATIVE
  7. SUPERADMIN

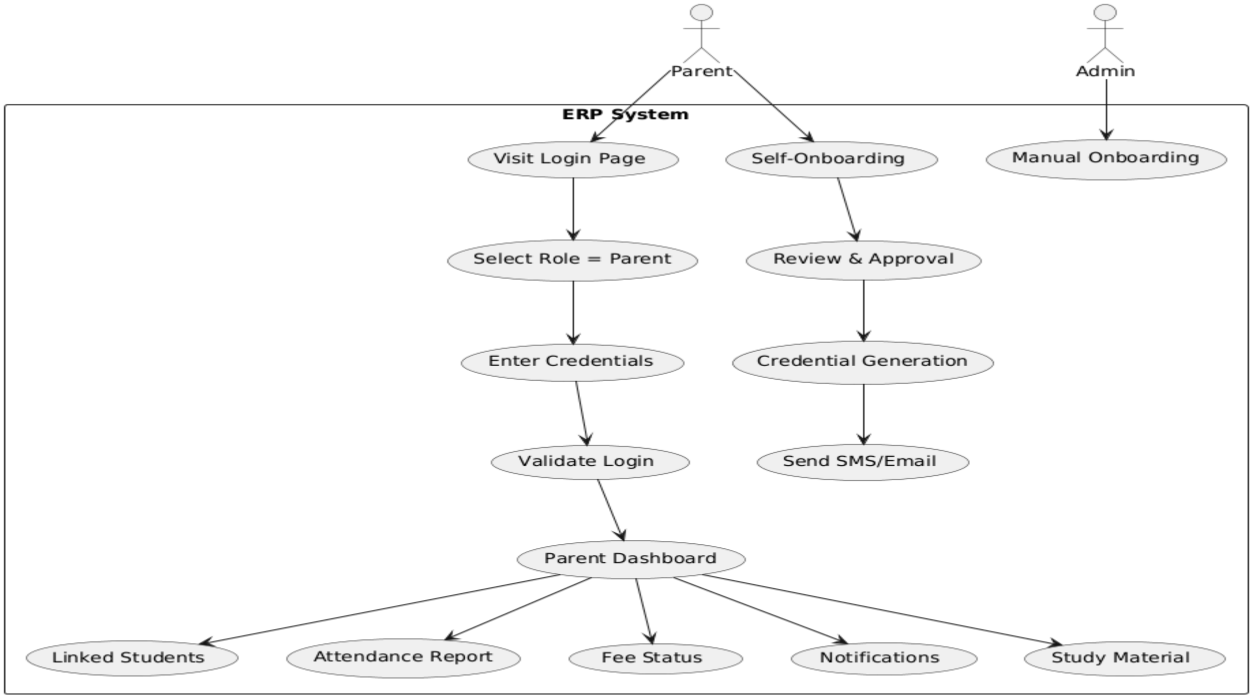
- **MODULES EXPLANATION:-**
  - **STUDENT MODULE:-**
    - **Basic Info:** name, email, gender, contact, DOB, religion, nationality, birth place etc
    - **Academic Info:** last institute, UDISC code, migration certificate, LC/TC, qualifying exam
    - **Notice & Session Info:**
      - Marksheet/Progress Report
      - Attendance
      - Study Materials
      - Tests, Assignments, Syllabus
  - **PARENT MODULE:-**
    - Monthly attendance reports
    - Study materials (based on child's class)
    - Fee tracking: Paid / Pending → Notification system
  - **TEACHER MODULE:-**
    - **Syllabus Management:** upload study materials, weekly schedule
    - **Notices:** post general or class-specific notices
    - **Attendance:** submit daily attendance
      - Notification to super admin if not submitted
    - **Leave Application:**
      - Select coverage teacher
      - Approval by both coverage teacher and superadmin
  - **STORE MODULE:-**
    - Non-consumables: fans, benches, etc.
    - Consumables: stationery, chemicals, chalk, food
    - Maintain inventory, update stock
  - **LIBRARY MODULE:-**
    - Book database: type, quantity

    - Book issue system: student/teacher
    - Return system
    - Notifications for return due
    - Auto update of quantity on issue/return

- **ADMINISTRATIVE MODULE:-**
    - Account section (fees, payments)
    - Admission form processing
    - Certificates/results management
    - Generate and export reports (attendance, fees, performance)
- **SUPERADMIN MODULE:-**
    - Approve teacher leaves (with coverage teacher)
    - View complete student/teacher data
    - Track attendance across teachers and students
    - Manage fee structure templates
    - View and audit store/library logs

PARENT AND STUDENT MODULE

- ❖ **PARENT MODULE**
  - ➢ **Features:-**
    1. **Manual Onboarding (Admin)**
       - Admin inputs Parent & Student info
       - Uploads documents (LC, ID, Address proof)
       - Credentials auto-generated → Sent via SMS/Email
    2. **Self-Onboarding (Optional)**
       - Parent fills online form
       - Uploads student & parent details + documents
       - Admin reviews → Approves/Rejects → Credentials sent
    3. **Login Flow**
       - Visit ERP Login Page
       - Select Role: Parent
       - Enter Username (mobile/email) + Password
       - Send otp and verify the otp
       - On Success → Redirect to Parent Dashboard
    4. **Parent Dashboard Includes:**
       - **Linked Students:-** View multiple enrolled children
       - **Attendance Reports:-** Daily/monthly summaries
       - **Fee Structure:-** View paid/pending amounts, make payments
       - **Notifications:-** Fee reminders, announcements
       - **Study Material:-** View child’s academic content

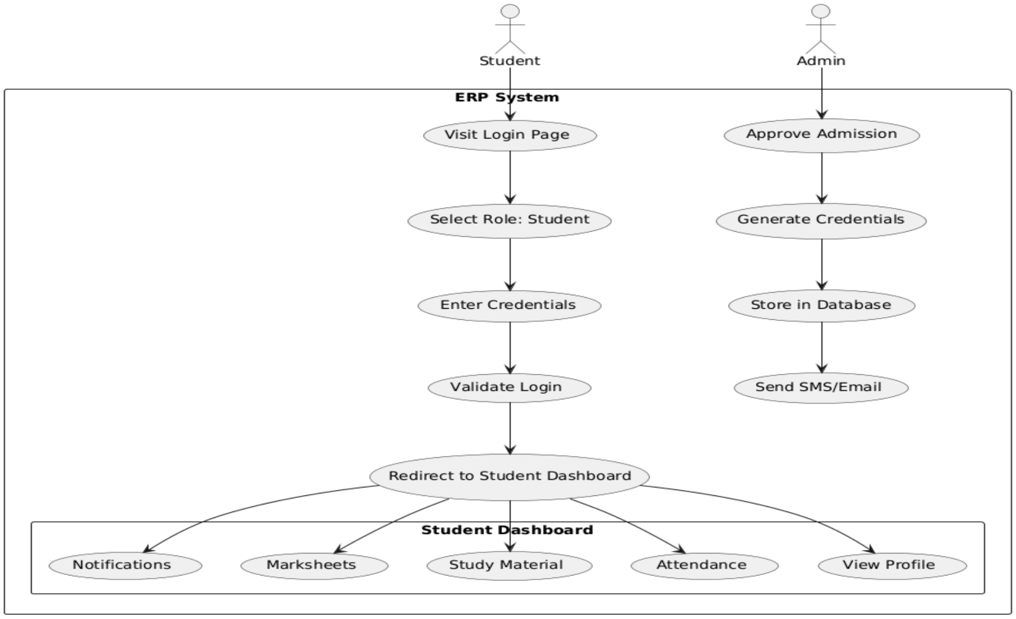


❖ STUDENT MODULE

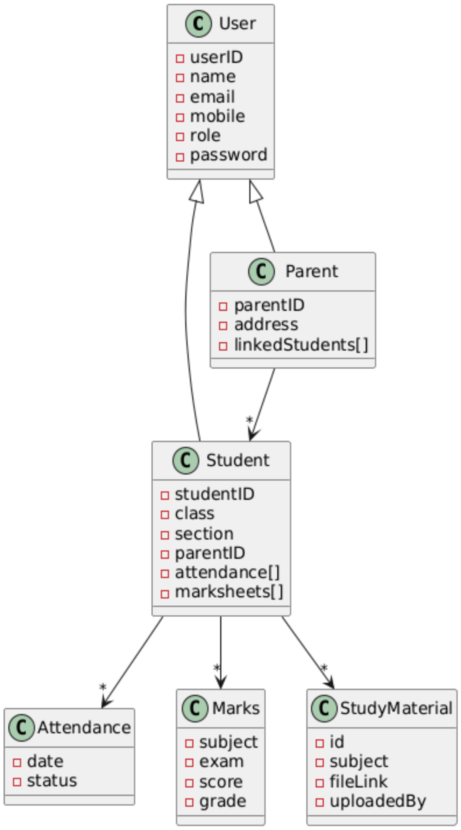
➢ **Features:**

1. **Credential Generation (Post Admission Approval)**
   - System generates unique credentials:
     - Username = Email/Mobile/Custom ID
     - Password = Auto-generated or set by Admin
   - Credentials sent via Email/SMS
2. **Login Flow**
   - Visit ERP Login Page
   - Select Role: Student
   - Enter Username (email/mobile/ID) + Password
   - Send otp and verify the otp
   - On Success → Redirect to Student Dashboard
3. **Student Dashboard Includes:**
   - Profile
     - Name, photo, class, contact info, guardian details
   - Attendance
     - Calendar view with filters
     - Statistics (% present, absent)
   - Study Material
     - Subject-wise files (PDFs, videos, links)
   - Marksheets

    - Exam-wise subject performance, grades, graphs
  - Notifications
    - Announcements, reminders, events



❖ CLASS DIAGRAM:-



OTP Email Verification System using Python and Gmail SMTP

Objective

To implement a secure One-Time Password (OTP) verification system that sends an OTP to the user's email using Gmail SMTP and verifies it upon user input.

Technologies Used

Technology	Description
Python	Programming language
Flask (optional)	For web framework
smtplib	Built-in Python module for sending emails
email.mime	For formatting and composing emails
Gmail SMTP	SMTP server used to send OTP emails
HTML	Frontend form input (email & OTP)

System Workflow (Flowchart Summary)

User → Enter Email → Generate OTP → Send Email (SMTP) → Enter OTP → Verify OTP → Success / Failure

Gmail SMTP Configuration

1. Go to your Google account settings.
2. Enable 2-Step Verification.
3. Create an App Password for 'Mail' and 'Other (Custom Name)'.
4. Save the 16-character password securely.

5. Use this app password in your Python code instead of your regular Gmail password.

Python Libraries Used

smtplib

Establishes connection with SMTP server and sends email using sendmail() function.

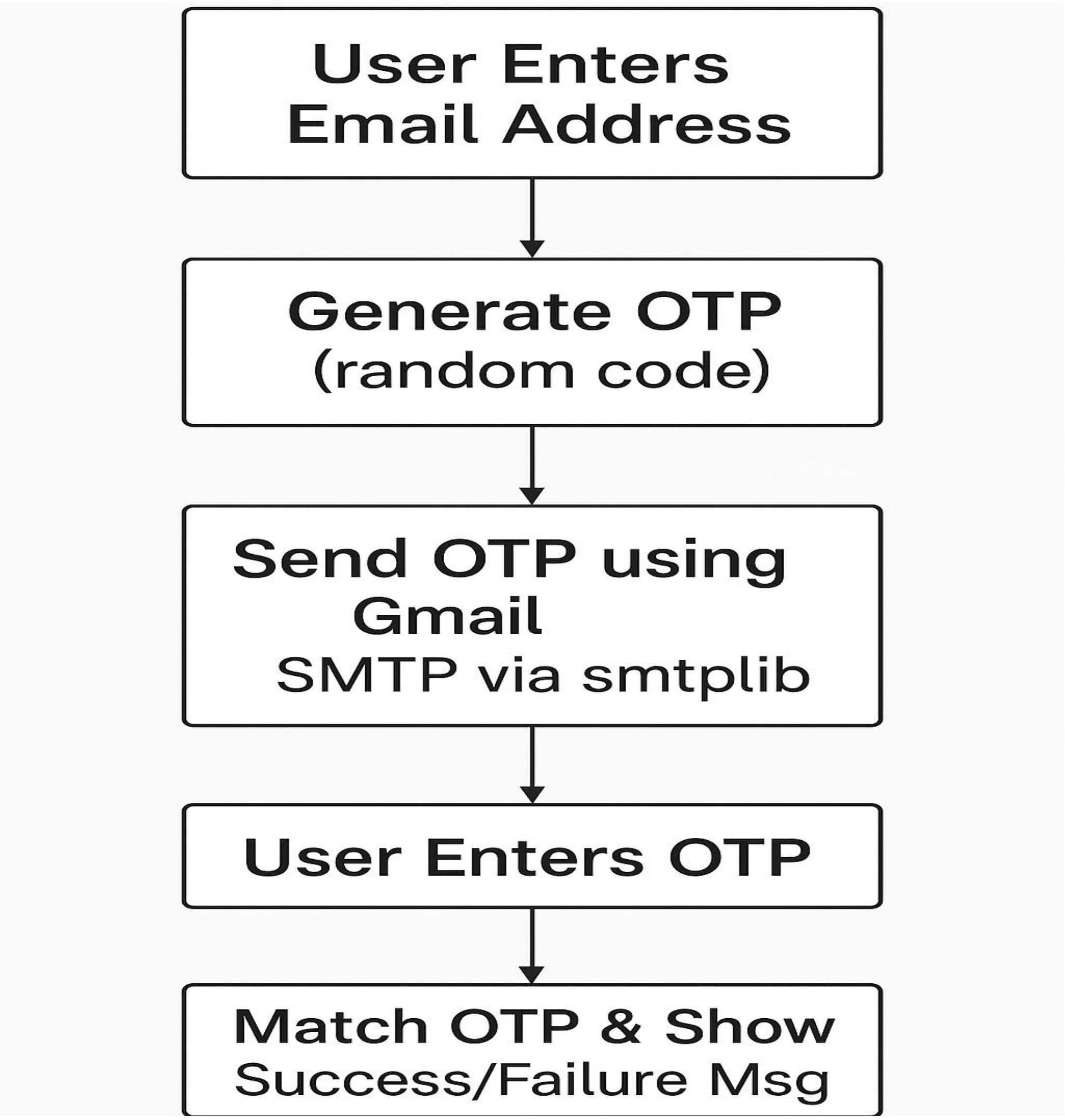
email.mime

Formats the email with subject, sender, receiver, and body (plain or HTML).

How the OTP System Works

1. User inputs their email.
2. OTP is generated using Python random.randint().
3. Email is formatted using email.mime.text.MIMEText.
4. SMTP connection is established using smtplib.
5. Email is sent with OTP to the user's email address.
6. User inputs OTP on the frontend.
7. System verifies OTP stored in session or memory.
8. Access granted if correct, else error message.

Flow Diagram


User Enters Email Address
Generate OTP (random code)
Send OTP using Gmail SMTP via smtplib
User Enters OTP
Match OTP & Show Success/Failure Msg

Security Tips

- Always use App Password for Gmail.
- Do not hardcode passwords. Use environment variables.
- Set OTP expiration time (e.g., 5 minutes).
- Use HTTPS in production environments.

Conclusion

This OTP system enables secure user verification through Gmail by using Python’s built-in libraries. It is cost-free, simple to integrate, and useful for login verification, password reset, or any user validation flow.

❖ Onboarding Flow For Teacher

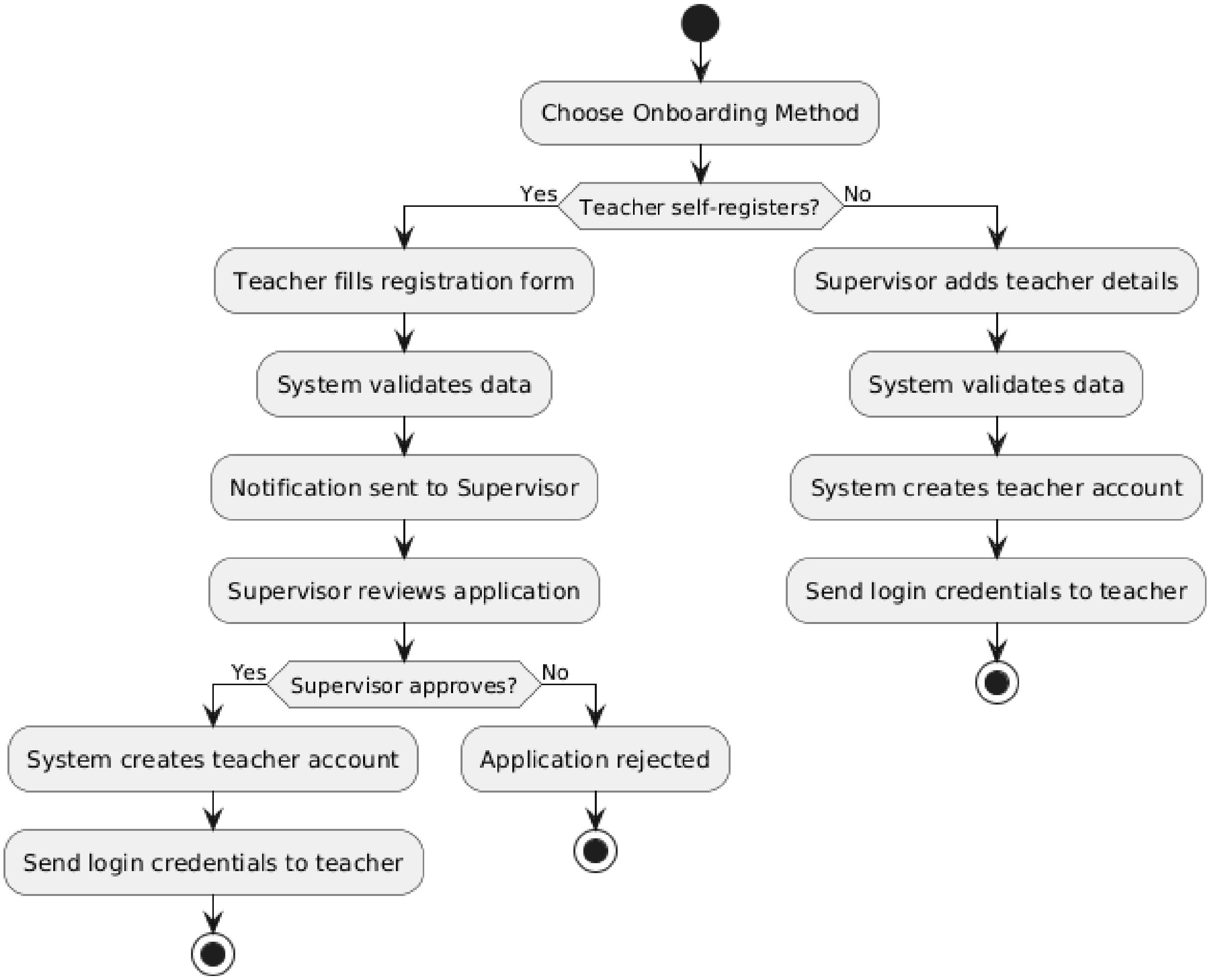
➢ Onboarding Methods:

Option 1: Teacher Self-Registration

- **The teacher chooses to register themselves on the ERP system.**
- **The teacher fills in the registration form with personal, academic, and professional details.**
- **The system automatically validates the entered data for completeness and correctness.**
- **Upon successful validation, the teacher's account is created in the system.**
- **Login credentials are sent to the teacher via email or SMS.**

Option 2: Superadmin Adds Teacher

- **The superadmin logs into the ERP system and accesses the 'Add Teacher' module.**
- **The superadmin fills in the teacher's details manually.**
- **The system validates the entered information.**
- **Upon successful validation, the teacher’s account is created in the system.**
- **Login credentials are sent to the teacher via email or SMS.**


Choose Onboarding Method
Teacher self-registers?
Yes
No
Teacher fills registration form
System validates data
Notification sent to Supervisor
Supervisor reviews application
Supervisor approves?
Yes
No
System creates teacher account
Send login credentials to teacher
Application rejected
Supervisor adds teacher details
System validates data
System creates teacher account
Send login credentials to teacher

Teacher Login Process

❖ Steps:

1. Teacher navigates to the ERP Login Page
2. Chooses the role as Teacher
3. Provides:
    - Username (email or ID)
    - Password
4. Clicks on Login button
5. System validates the credentials
    - OTP Verification
6. Upon successful verification:
    → Teacher is directed to the Dashboard featuring five key sections:
    - Subject Management
    - Weekly Planner
    - Announcements
    - Attendance Tracking
    - Leave Requests

A. Subject Management

❖ Steps:

1. Teacher selects the Subject Management section
2. Uploads syllabus documents for each class
3. Adds detailed topic content for each syllabus section
4. Uploads supporting files such as photos, videos, or PDFs linked with the content

B. Weekly Planner

❖ Steps:

1. Teacher opens the Weekly Planner section
2. Picks the class to plan for
3. Enters the subjects or topics to be covered in the week
4. Uploads a photo to complement the weekly plan

C. Announcements

❖ Steps:

1. Teacher accesses the Announcements section

2. Creates new notice or message
3. Attaches a photo if needed to enhance the notice

D. Attendance Tracking

❖ Steps:

1. Teacher goes to Attendance Tracking
2. Selects the relevant class
3. Marks students who are absent (all others are automatically marked present)
4. If attendance is not submitted by the end of the day, automatic reminders are sent to both the teacher and the superadmin

E. Leave Requests

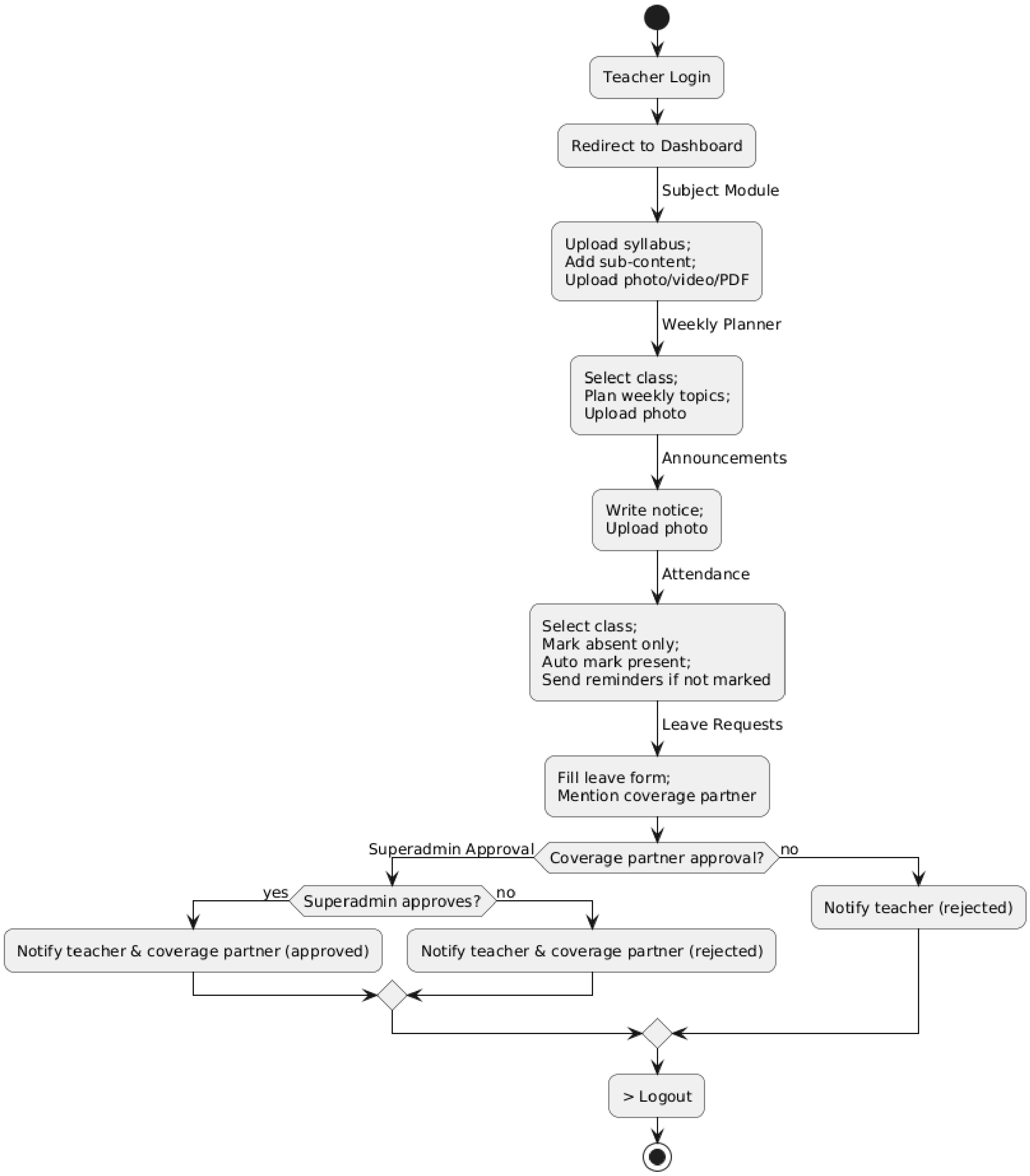
❖ Steps:

1. Teacher opens the Leave Requests section
2. Completes the leave application form
3. Specifies a coverage partner who will manage duties during absence
4. Submits the leave request to the coverage partner for initial approval
5. Coverage partner reviews and either accepts or declines the request
    - If accepted, the request moves to the superadmin
    - If declined, the teacher is notified immediately
6. Superadmin reviews and either approves or rejects the leave request
7. Final decision notifications are sent to both the teacher and the coverage partner

F. Logout

❖ Steps:

Teacher logs out from the ERP system securely


Teacher Login
Redirect to Dashboard
Subject Module
Upload syllabus;
Add sub-content;
Upload photo/video/PDF
Weekly Planner
Select class;
Plan weekly topics;
Upload photo
Announcements
Write notice;
Upload photo
Attendance
Select class;
Mark absent only;
Auto mark present;
Send reminders if not marked
Leave Requests
Fill leave form;
Mention coverage partner
Superadmin Approval
Coverage partner approval?
no
yes
Superadmin approves?
no
Notify teacher (rejected)
Notify teacher & coverage partner (approved)
Notify teacher & coverage partner (rejected)
> Logout

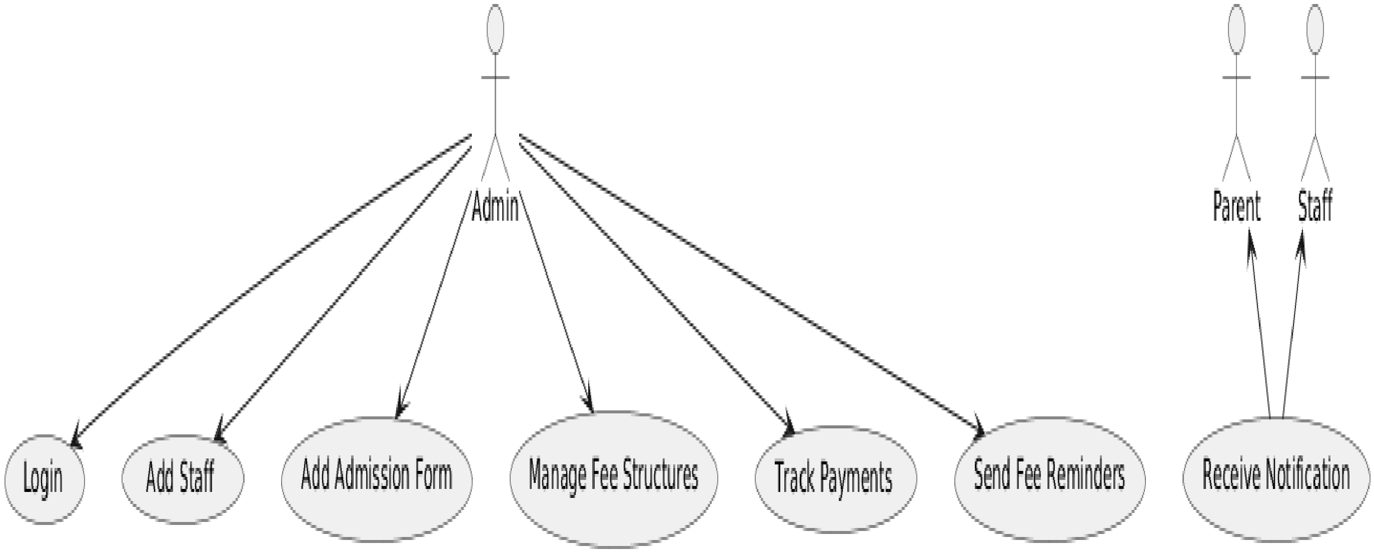
ADMINISTRATIVE MODULE

Overview

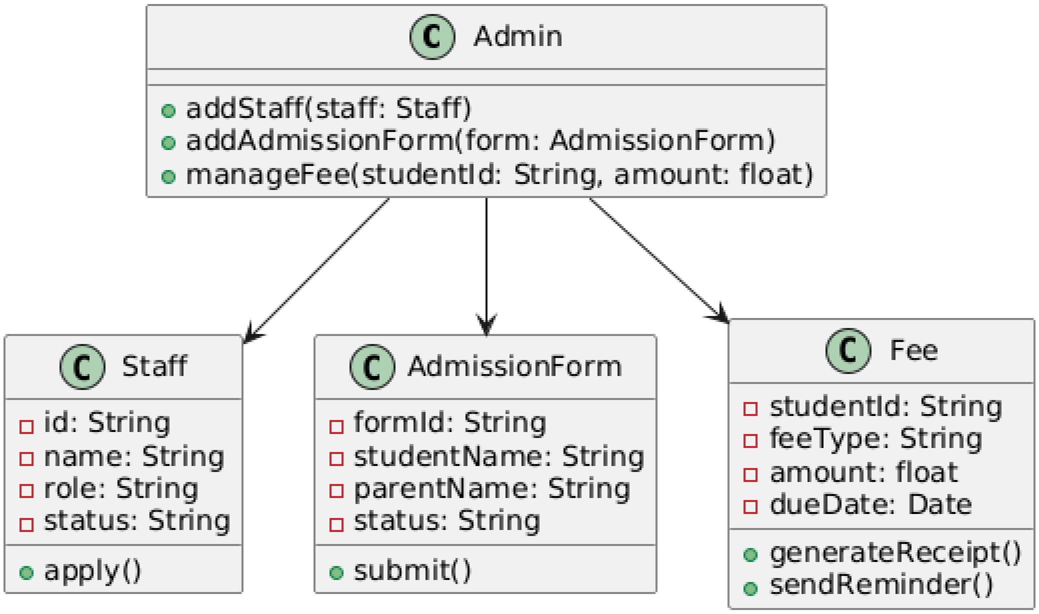
The Administrative Module provides core control for managing staff, admissions, and accounting operations. This module is primarily used by the Admin (Supervisor) role in the ERP system.

- **Features**
  - **Staff Management**
    - **Add Staff:** Admin can directly create records for staff members (Teacher, Librarian, etc.).
  - **Admission Management**
    - **Add Admission Form:** Admin can manually add admission forms or manage those submitted by parents/students.
  - **Fee/Account Management**
    - **Manage Fee Structures:** Define fee heads, durations (monthly, quarterly).
    - **Track Payments:** View payment history and dues.
    - **Generate Receipts:** Downloadable receipt per transaction.
    - **Send Reminders:** Notify parents/students about dues.

2. System Flow Diagram



3. Class Diagram



4. Notification System (Optional)

- Triggered on status updates for:
  - Fee dues
- Can be implemented using Email or Push Notification services.

Library Management System - Documentation

Objective

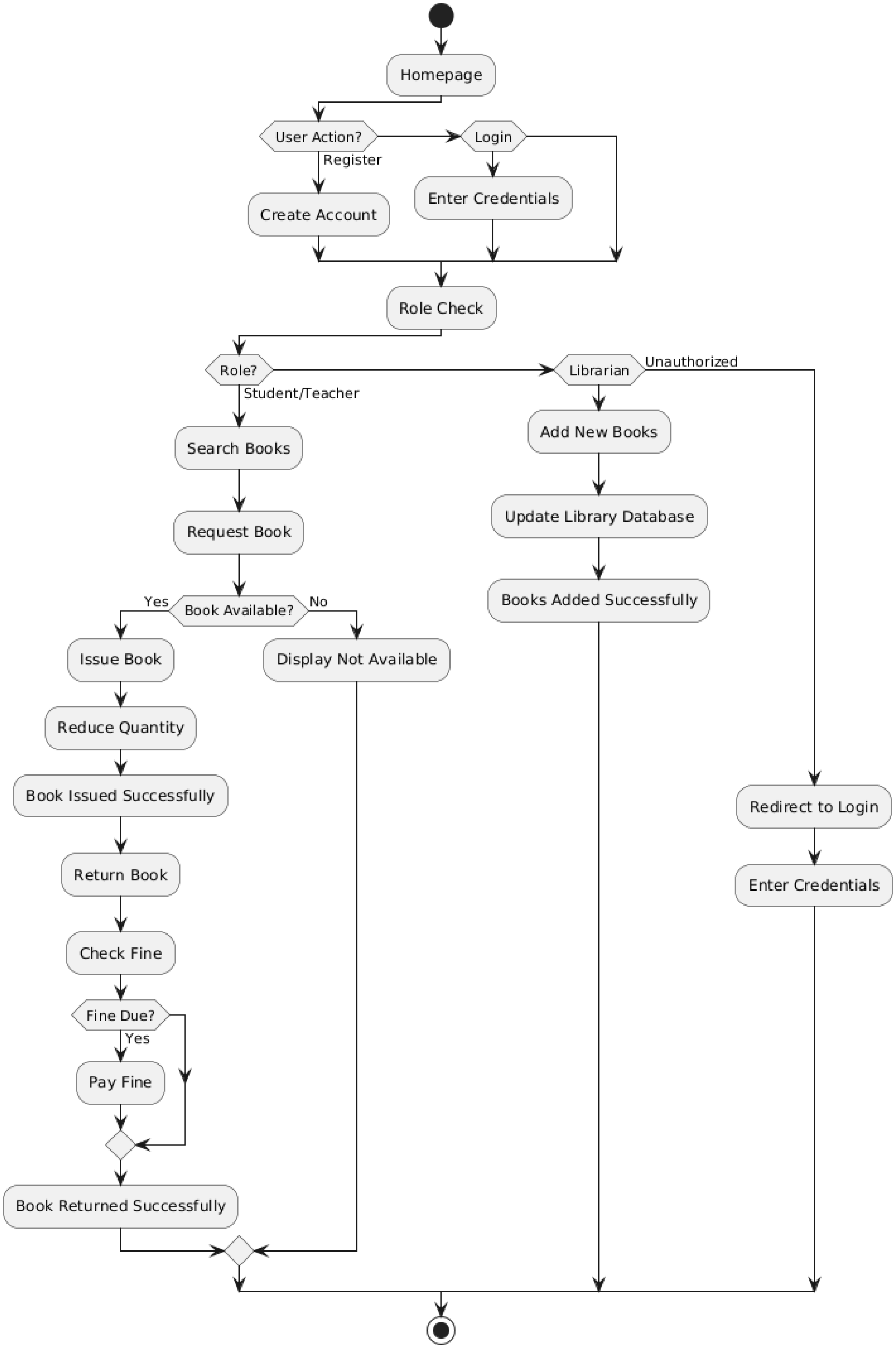
To develop a Library Management System using Python (Flask) that provides registration, login, book search, book request, and return functionality for students, teachers, and librarians.

Technologies Used

Technology	Description
Python	Programming language for backend logic
Flask	Web framework for Python
HTML/CSS	Frontend form and dashboard layout
MySQL / SQLite	Database for storing users, books, and requests
Mermaid	Used to generate system flowcharts

System Workflow Summary

Homepage → Register/Login → Role Check (Student/Teacher/Librarian) → Role-Based Dashboard → Book Search → Book Request → Auto-Issue Book → Return Book → Fine Calculation (if late)


Homepage
User Action?
Register
Login
Create Account
Enter Credentials
Role Check
Role?
Student/Teacher
Librarian
Unauthorized
Search Books
Request Book
Book Available?
Yes
No
Issue Book
Display Not Available
Reduce Quantity
Book Issued Successfully
Return Book
Check Fine
Fine Due?
Yes
Pay Fine
Book Returned Successfully
Add New Books
Update Library Database
Books Added Successfully
Redirect to Login
Enter Credentials

Conclusion

The Library Management System enables efficient book tracking and management with role-based access for students, teachers, and librarians. The system supports search, book issuing, and return workflows, along with basic fine calculation.

Book Quantity Management Logic

To ensure accurate tracking of book availability, the system updates the book quantity in the database when a user requests or returns a book. The logic is implemented as follows:

1. Book Request:

- When a student or teacher requests a book, the system checks if the book quantity is greater than 0.
- If available, the book is issued and quantity is reduced by 1.
- If quantity is 0, the user is informed that the book is not available.

2. Book Return:

- When the user returns the book, the system identifies the corresponding book.
- The book quantity is increased by 1 in the database.
- This ensures real-time inventory tracking and prevents over-issuing.

This logic is essential to maintain accurate inventory and support transparent book issuance.

Store Module Documentation

1. Introduction

The Store Module in the School ERP System manages all stock-related operations including tracking of consumable and non-consumable items. It ensures efficient stock management, alerts for low stock, and comprehensive reporting for transparency and accountability.

2. Technology Stack

Backend Framework: Flask (Python) **Frontend:** HTML/CSS with responsive design **Database:** MySQL **Architecture:** MVC (Model-View-Controller) pattern

3. Functional Requirements

3.1 Item Categories

- **Consumable Items:** Pens, chalk, papers, cleaning supplies, stationery
- **Non-Consumable Items:** Desks, computers, projectors, chairs, furniture

3.2 Stock Management Features

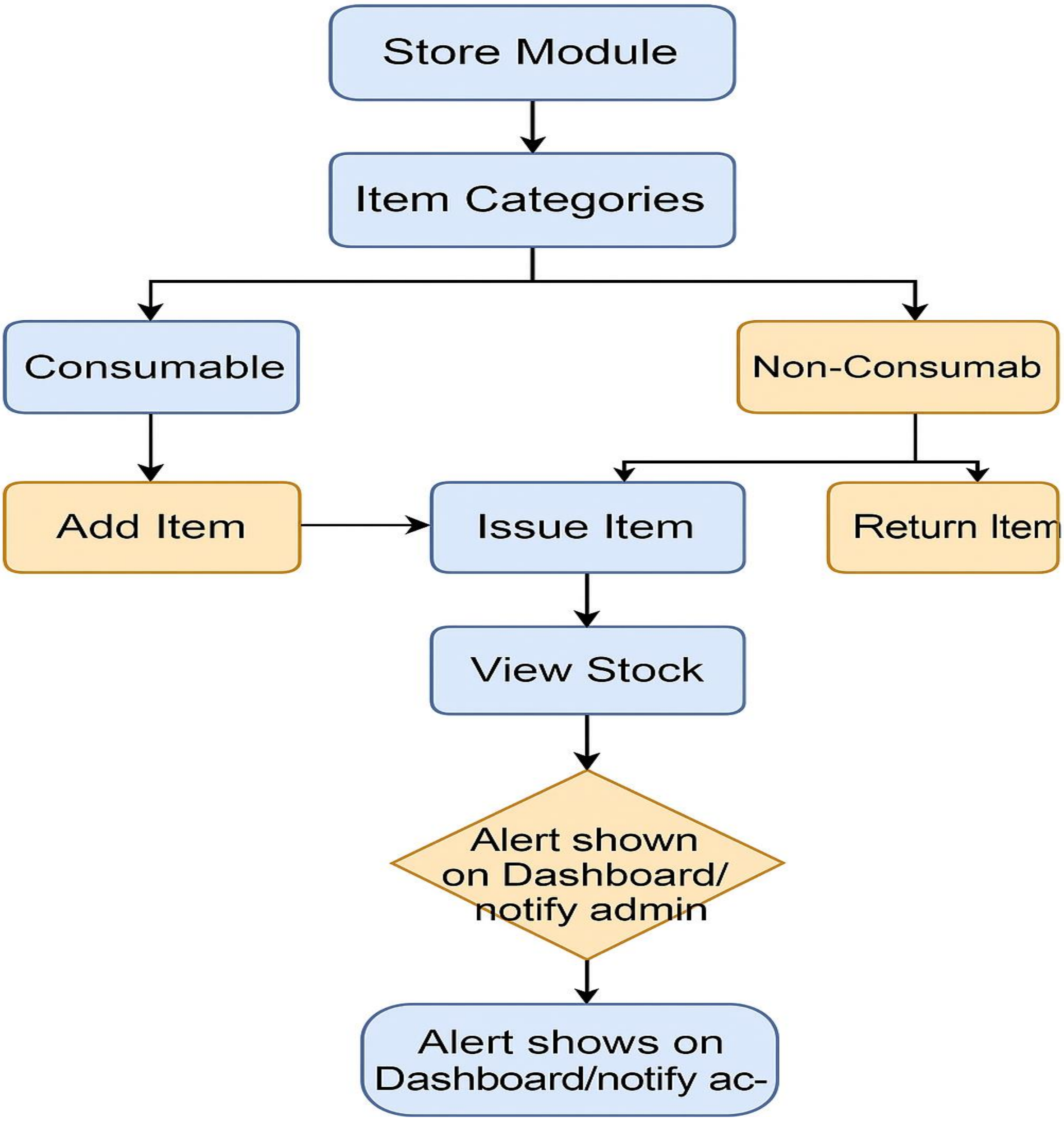
- **Item Management:** Add, edit, delete items with detailed specifications
- **Stock Operations:** Issue items to departments/staff/students
- **Return Management:** Process returns (non-consumables only)
- **Quantity Tracking:** Real-time stock level monitoring
- **Alert System:** Automated low stock notifications
- **Transaction History:** Complete audit trail of all stock movements
- **Multi-level Access:** Role-based permissions for different users

3.3 Reporting & Analytics

- **Stock Summary Report:** Current inventory status across all categories
- **Item Usage Report:** Consumption patterns and trends
- **Department-wise Report:** Items issued to specific departments
- **Date-wise Transaction Report:** All issued/returned items by date range
- **Low Stock Alert Dashboard:** Items requiring immediate attention

4. System Workflow

The following flowchart illustrates the complete workflow of the Store Module, showing how items are categorized, managed, and how the alert system functions:



Workflow Process:

1. **Item Categorization:** All items are first classified as Consumable or Non-Consumable
2. **Item Addition:** New items are added to the system with complete details
3. **Stock Operations:** Items can be issued to departments, staff, or students
4. **Return Processing:** Non-consumable items can be returned and tracked
5. **Stock Monitoring:** System continuously monitors stock levels
6. **Alert Generation:** When stock falls below minimum threshold, alerts are generated
7. **Dashboard Notification:** Alerts are displayed on admin dashboard for immediate action

5. Database Design (MySQL)

Table: ItemCategory

Sample Data:

id	name	description
1	Consumable	Items that are used up and need regular replenishment
2	Non-Consumable	Durable items that can be returned and reused

Table: Item

Sample Data:

id	name	category_id	quantity	unit	min_stock_alert	description
1	Whiteboard	2	10	pcs	2	Interactive whiteboards for classrooms
2	Chalk	1	200	box	50	White chalk boxes for blackboards

Table: StockTransaction

Sample Data:

id	item_id	transaction_date	transaction_type	quantity	issued_to	returned_by	remarks
1	2	2024-01-10	ISSUE	20	Staff ID 3	NULL	Daily classroom usage
2	1	2024-01-11	RETURN	1	NULL	Staff ID 3	Equipment in good condition

6. Flask API Endpoints

Action	Endpoint	Method	Description
Item Management			
Add Item	/add_item	POST	Create new inventory item
View Items	/items	GET	Display all items with pagination
Edit Item	/edit_item/<id>	POST	Update item details
Delete Item	/delete_item/<id>	DELETE	Remove item from inventory
Stock Operations			
Issue Item	/issue_item	POST	Record item issuance
Return Item	/return_item	POST	Process item return
Reporting & Alerts			
View Reports	/reports	GET	Generate various reports
Low Stock Alerts	/low_stock_alerts	GET	Display items below threshold
Stock History	/stock_history/<id>	GET	View transaction history

Action	Endpoint	Method	Description
Dashboard			
Dashboard	/dashboard	GET	Main overview page
Search Items	/search	GET	Search functionality