SRS On

"ACADEMIX"

Academic Management System

Course Title: Software Engineering and Information System Design

Course Code: CSE-3103

Submitted to

Md. Samsuddoha Sams
Assistant Professor
Department of CSE
University of Barishal

Submitted By

Team Vertex (Team-06)

Suria Hossain Bonna(22-CSE-006)
Nayim Sheikh(22-CSE-010)
Rasel Hossain(22-CSE-027)
A.B. Rahaman(22-CSE-034)
Biplob Hossain(22-CSE-037)
Yeatasim Billah(22-CSE-052)

Session: 2021-22

Year: 3rd Year 1st Semester

Department of Computer Science and Engineering

University of Barishal

Date of Submission: 19 January, 2026

Table of Contents

ACADEMIX Project Contribution Table

Serial No.	Task Name	Section	Member Name
1.	Inception	1.0	Yeatasim
2.	Elicitation	2.0	Yeatasim
3.	Activity Modeling	3.2	Yeatasim
4.	Swimlane Modeling	3.2.1	Yeatasim
5.	Use Case Modeling	3.1	Suraia Hossain Bonna
6.	Class Based Modeling	5.0	Nayim Sheikh
7.	Data Modeling (ERD)	4.0	Md. Biplob Hossain
8.	State Transition Modeling	7.1	Md. Rasel
9.	Behavioral Modeling	7.0	A.B. Rahaman
10.	System Sequence Diagrams	7.2	A.B. Rahaman

ACADEMIX

1. Inception

Introduction

Inception is the beginning phase of requirements engineering. It defines how a software project gets started and the scope and nature of the problem to be solved. The goal of the inception phase is to identify concurrence needs and conflict requirements among the stakeholders of the ACADEMIX project. To establish the groundwork, we have worked with identifying stakeholders, recognizing multiple viewpoints, and working towards collaboration.

Identifying Stakeholders

Stakeholder refers to any person or group who will be affected by the system directly or indirectly. To identify the stakeholders, we consulted with university management and asked questions regarding who pays for the project, who uses the outcomes, and whose work is affected .

We identified the following stakeholders for our ACADEMIX system:

1. **Super Admin (Project Owner):** The Super Admin has the final authority over system-level configurations and administrative oversight. Their position empowers them to manage global settings.
2. **Department Admin:** As the head of operational management, the Department Admin has direct authority over academic verification and scheduling within their specific department.
3. **Teacher:** Teachers interact directly with the software to manage courses, attendance, and assessments.
4. **Student:** Students will directly interact with the software to access materials and track performance.
5. **System Administrator (IT):** Responsible for infrastructure health and security protocols to ensure the system is maintained.
6. **University Management:** They require high-level strategic oversight and analytics to ensure the project meets organizational goals.

Recognizing Multiple Viewpoints

Each stakeholder has specific requirements and expectations:

1. Super Admin Viewpoints:

a) System Control: Needs to manage departments, academic years, and global configurations.

b) Account Management: Requires tools to oversee all user accounts and roles.

c) Audit Compliance: Needs access to system logs and activity records for security.

2. Department Admin Viewpoints:

a) Scheduling: Needs to create conflict-free class routines and exam schedules.

b) Academic Verification: Must verify and lock internal marks before publication.

3. Teacher Viewpoints:

a) Classroom Management: Needs real-time student lists and attendance marking tools.

b) Assessment Control: Requires a platform to post assignments and collect submissions.

4. Student Viewpoints:

a) Information Access: Needs easy access to routines, notices, and course materials.

b) Progress Tracking: Wants clear visibility of attendance and academic results.

c) Submission Utility: Needs a reliable portal for digital assignment hand-ins.

5. System Administrator (IT) Viewpoints:

a) Infrastructure Health: Needs to monitor XAMPP performance and database status.

b) Security Protocols: Ensures hashing, sanitization, and session security are active.

Working Towards Collaboration

Every stakeholder has their own requirement. In this step, we merged these requirements by identifying common and conflicting requirements and prioritizing them.

Common Requirements:

a) User-friendly and efficient system.

b) Role-based access to functionality.

c) Accessible via the Internet.

Conflicting Requirements:

a) Open access for students vs. Strict security protocols by IT.

b) Detailed analytics for Management vs. Simple interface for Students.

Final Requirements:

After prioritizing, we finalized the following requirements:

1. Error-free system for mark calculation.
2. Allow valid users to login and logout.
3. Restrict access to functionality based on user roles (Admin, Teacher, Student).
4. Allow Department Admins to configure class routines and verify marks.
5. Allow Teachers to create assignments and track attendance.
6. Allow Students to view results, attendance, and submit assignments.

2. Elicitation

Introduction

Elicitation is a task that helps the customer to define what is required. To help overcome problems of scope and understanding, we worked with the Eliciting requirements activity in an organized manner.

Collaborative Requirements Gathering

Many different approaches to collaborative requirements gathering have been proposed. We completed the following steps:

The meetings were conducted with the academic registrars and department heads. The stakeholders were questioned about their requirements and expectations from the ACADEMIX system. Finally, we selected our final requirement list from the meetings.

Quality Function Deployment (QFD)

Quality Function Deployment is a technique that translates the needs of the customer into technical requirements for software.

Normal Requirements

Normal requirements consist of objectives and goals that are stated during the meeting with the customers.

1. **Role-Wise Access:** Restrict access to functionality based upon user roles.
2. **CRUD Operations:** Complete Create, Read, Update, Delete for departments, users, and courses.
3. **Attendance:** Allow Teachers to track attendance and generate summary reports.
4. **Mark Management:** Allow automatic calculation of results and mark management.
5. **Notice Board:** Allow valid users to view official announcements.

Expected Requirements

These requirements are implicit to the system and may be so fundamental that the customer does not explicitly state them. Their absence will be a cause for dissatisfaction.

1. **Responsive Design:** The user interface should follow standard web practices and be fully responsive.
2. **Security:** The system shall ensure secure data handling (SQLi, XSS, CSRF protection).
3. **Navigation:** The user interface shall be easy to use and include intuitive navigation.
4. **Calculations:** The system shall automatically compute GPA/CGPA according to university rules.
5. **Authentication:** The system shall allow the user to log in based upon an assigned login ID and password.

Exciting Requirements

These requirements are features that go beyond the customer's expectations and prove to be very satisfying when present.

1. **Notifications:** The system shall automatically send notifications for attendance drops or result releases.
2. **Analytics:** Graphical analytics for department-wide performance comparison.
3. **Version Tracking:** Version tracking for student assignment submissions.
4. **Bulk Import:** The system shall enable Admins to add users via CSV/Excel import.

Usage Scenarios

At first, a user authenticates in our system. If a user already has an account, he/she will log in to the system with his/her own password and username.

In case of a Teacher:

If the user is a Teacher, the system will allow them to access the classroom management dashboard. The teacher selects a specific course and takes

attendance on their device; the system immediately updates the records in the database. The teacher can also create an assignment with a deadline.

In case of a Student:

If the user is a Student, the system will verify their enrollment status. The student checks their dashboard to see attendance percentage and eligibility. If an assignment is pending, the student uploads the file. The system changes the status of the submission to "Submitted" in our database.

In case of Department Admin:

The Department Admin logs in to verify marks. The system checks for any conflict in the exam schedule. If no conflict is found, the routine is published to the notice board database.

Elicitation Work Product

Our elicitation work product includes:

1. A statement of requirements for the ACADEMIX system.
2. A bounded statement of scope for our system.
3. A list of stakeholders who participated in requirements elicitation.
4. A set of usage scenarios.

3. Scenario Based Modeling

3.1 Use Case Modeling

3.1.1 Use Case Scenario Summary (Table)

Table 1. ACADEMIX system use cases by abstraction level and actors

Level - 0	Level - 1	Level - 2	Actors
ACADEMIX System	**Authentication**	Login / Logout	All Users
		Change Passwords	All Users
		Update Profile Data	All Users
	Governance	Manage Departments	Super Admin
		Manage Academic Years	Super Admin
		Configure System Settings	Super Admin
	Academic Setup	Create Course Catalog	Super Admin, Dept Admin
		Offer Courses	Dept Admin
		Assign Faculty	Dept Admin
		Generate Class Routine	Dept Admin
	Classroom	Mark Attendance	Teacher
		Post Course Materials	Teacher
		Create Assignments	Teacher
	Student Portal	Course Registration	Student
		Submit Assignments	Student
		View Attendance Stats	Student
	Assessments	Grade Submissions	Teacher
		Enter Mid/Final Marks	Teacher
		Verify & Lock Marks	Dept Admin
		Publish Results	Dept Admin
		Download Transcript	Student

3.1.2 Detailed Use Case Descriptions (Textual)

01. Authentication

i.

Use case: Login/Logout
Primary Actors: All users
context: To enter in the system
Precondition:

1. System has been programmed for add new user in database
2. System has interface for registration

Scenario:

1. Visit the register page
2. Input required information
3. Check availability for username & check validity of Password
4. Authentication and Robot checking
5. E-mail sent to user e-mail address
6. User confirm from his/ her e-mail address
7. Confirmation message showed

Exception:

1. User in not authorized for registration
2. Ambiguous Input
3. Authentication Fail

Priority: Essential, must be implemented
When Available: First increment

ii.

Use case: Sign In

Primary Actors: Student, Faculty, Administrator, Librarian

Goal in context: To enter the system.

Precondition: Must be registered

Triggers: Need to log in the system

Scenario:

1. Visit the login page
2. Input Username & Password
3. Proceed to the next activity

Exception:

1. Unrecognized Username
2. Wrong Password
3. User is blocked

Priority: Essential, must be implemented

When Available: First increment

iii.

Use case: Sign Out

Primary Actors: Student, Faculty, Administrator, Librarian

Goal in context: To exit from the system

Precondition: Must be logged in

Triggers: Need to log out from the

system

Scenario:

1. Click the logout button

Priority: Essential, must be implemented

When Available: First increment

iv.

Use case: Change Password

Primary Actors: All Users

Goal in context: To change the existing password to a new password

Precondition: System has been programmed for a password

Triggers: Super admin, Dept. admin, Teacher and Student have a need to change the existing password to a new one .

Scenario:

2. Visit the login page and login
3. Click on Edit button
4. Change Password
5. Proceed to the next activity

Exception: Weak Password: Password length is too short

Priority: Essential, must be implemented

When Available: First increment

02. Governance

i.

Use case: Manage Departments

Primary Actors: Super admin

Goal in context: To manage the Departments.

Precondition: Must be logged in as Super admin

Triggers: The Super admin has to modify and moderate everything of the departments

Scenario:

1. Visit login page and login
2. Click the Edit department button
3. Select the option
4. Click on the Edit button
5. Change the type for the selected option
6. Proceed to the next activity

Priority: Essential, must be implemented

When Available: First increment

ii.

Use case: Manage Academic Years

Primary Actors: Super Admin

Goal in context: To select the system for individual years and semesters

Precondition:

1. Must be logged in as Super Admin
2. System has been programmed for editing the academic functions

Triggers: The super admin has a need to configure the years and semesters.

Scenario:

1. Visit Login page and Login
2. Click on Manage session
3. Select the year
4. Click on required option
5. Change the function
6. Proceed to the next activity

Exception:

1. Option Unavailable: Requested item does not exist
2. Ambiguous Input

Priority: Expected

When Available: Second increment

iii.

Use case: Configure System Setting

Primary Actors: Super Admin

Goal in context: To configure internal systems

Precondition:

1. Must be logged in as Super admin
2. System has been programmed for editing the internal issues

Triggers: The Super admin has a need to configure the internal system.

Scenario:

1. Visit Login page and Log in
2. Click on System
3. Click on Configure the system for Department
4. Click on Update button
5. Proceed to the next activity

Exception: Ambiguous Input

Priority: Expected

When Available: Second increment

03. Academic Setup

i.

Use case: Create Course Catalog

Primary Actors: Super Admin, Dept. Admin.

Context: To create and delete course

Precondition:

1. System has been programmed for adding or deleting item in database
2. Must be logged in as Super admin or dept. admin

Trigger: The super admin or dept admin has a need to configure the courses

Scenario:

1. Visit Login page and Log in
2. Click on Create course or delete course
3. Click on Add Item button to add new item
4. Click on Delete Item button to remove old item
5. Enter the new Item data and confirm changes
6. Proceed to the next activity

Exception: While deleting: Item is not found in the database

Priority: Essential, must be implemented

When Available: First increment

ii.

Use case: Offer Courses

Primary Actors: Dept. Admin

Goal in context: To offer courses

Precondition:

1. System has been programmed for offering courses saved in database
2. Must be logged in as dept. admin

Trigger: The dept. admin has to exhibit the courses

Scenario:

1. Visit Login page and Log in
2. Click on Courses
3. Search and Select the Offer
4. Proceed to the next activity

Exception:

1. Does not exist: Requested item does not exist in the database
2. Ambiguous Input

Priority: Essential, must be implemented

When Available: First increment

iii.

Use case: Assign Faculty

Primary Actors: Dept. admin

Context: To assign course instructors and stuffs

Precondition:

1. System has been programmed to assign the needs as required
2. Must be logged in as dept. admin

Trigger: The dept. admin has a need to change and redesign the format of a department.

Scenario:

1. Visit Login page and Log in
2. Click on Maintain Character
3. Search and Select the Item to assign
4. Confirm changes
5. Proceed to the next activity

Exception: Does not exist: Requested item or character does not exist in the database

Priority: Essential, must be implemented

When Available: First increment

iv.

Use case: Generate Class Routine

Primary Actors: Dept.

admin

context: To change routine

according to needs and times

Precondition:

1.System has been programmed to assign the needs as required
2.Must be logged in as dept. admin

Trigger: The dept. admin has a need to change and redesign the format of a department.

Scenario:

1. Visit Login page and Log in
2. Click on Change Routines
3. Select courses and times
4. Confirm changes
5. Proceed to the next activity

Exception: Does not exist: Requested item does not exist in the database

Priority: Essential, must be implemented

When Available: First increment

04. Classroom

i.

Use Case: Mark Attendance

Primary Actors: Teacher

Goal in context: To stalk performance

Precondition: Users must be enroll to courses

Trigger: The Teacher has a need to track their students

Scenario:

1. Visit Login page and Log in
2. Click on Visit Courses
3. Click on Take Attendance
4. Mark the Present ones
5. Save the attendance
6. Proceed to the next activity

Priority: Essential, must be implemented

When Available: First increment

ii.

Use Case: Post Course Materials

Primary Actors: Teacher

Goal in context: To see the previous reports

Precondition:

1. The data must be saved in the database
2. Must be logged in as Teacher

Triggers: Teacher has a need to see the previous course recording.

Scenario:

1. Database is automatically checked
2. Share the course information with the next class
3. Select Change button
4. Update database
5. Proceed to the next activity

Exception: Error: Empty data

Priority: Essential, must be implemented

When Available: First increment

iii.

Use Case: Create Assignments

Primary Actors: Teacher.

Context: To create

Assignment

Precondition:

1. System has been programmed to add new assignments on saved courses
2. Must be logged in as Teacher

Trigger: The Teacher has a need to give assignments on the selected courses to the learners.

Scenario:

1. Visit Login page and Log in
2. Click on My Courses
3. Click Upload file
4. Confirm upload
5. Proceed to the next activity

Exception: Error: Systematically failed

Priority: Essential, must be implemented

When Available: First increment.

05. Student Portal

i.

Use Case: Course Registration

Primary Actors: Student

context: To apply for the registration of the courses

Precondition: System has been programmed for completing the registration

Triggers: The student needs to go through the registration process to be moderated by course instructors.

Scenario:

1. Visit the main page
2. Search the courses.
3. Enter data and information.
4. Click on the Submit button
5. Proceed to the next activity

Exception:

1. Search item does not exist
2. User is not eligible for searching that item

Priority: Essential, must be implemented

When Available: First increment

ii.

Use case: Submit Assignments

Primary Actors: Student

Context: To issue an item

Precondition:

1. User must be eligible for taking action
2. Assignments should be provided

Trigger: Students need to submit the assignment.

Scenario:

1. Visit Login page and Log in
2. Click on My Profile
3. Search for Due Assignment
4. Click on Upload file button
5. Click on Submit button
6. A message is displayed
7. Proceed to the next activity

Exception:

1. Invalid User: User status is not supported for this event
2. Time has ended

Priority: Essential, must be implemented

When Available: First increment

iii.

Use Case: View Attendance Stats

Primary Actors: Student

Context: To view the status of percentage over one's attendance

Precondition:

1. Valid User
2. Valid and Available data

Triggers: The student has the right to focus on classes.

Scenario:

1. Visit the login page and login
2. Go to My Profile
3. Select Course button
4. Click on My Attendance
5. Proceed to next activity

Exception:

1. Not approved for the course
2. Error : Systematically failed

Priority: Essential, must be implemented

When Available: First increment

06. Assesments

i.

Use Case: Grade Submissions

Primary Actors: Teacher

Context: To submit the grades for individual courses at that particular time.

Precondition:

1. Valid User
2. Availability of that courses in the database

Triggers: The teacher has a need to grade the students according to obtained marks.

Scenario:

1. Visit the login page and log in
2. Click on My Courses
3. Click on Students
4. Search the result
5. Click on Give Grade
6. Click on Submit
7. Proceed to the next activity

Exception: Absent

Priority: Essential, must be implemented

When Available: First increment

ii.

Use Case: Enter Mid/Final Marks

Primary Actors: Teacher

Context: To submit the marks for individual student at that particular time.

Precondition:

1. Valid User
2. Availability of that courses in the database

Triggers: The teacher has a need to grade the students according to obtained marks.

Scenario:

1. Visit the login page and log in
2. Click on My Courses
3. Click on Students
4. Search the result
5. Click on Give Marks
6. Click on Submit
7. Proceed to the next activity

Exception: Absent

Priority: Essential, must be implemented

When Available: First increment

iii.

Use Case: Verify & Lock Marks

Primary Actors: Dept. Admin

Context: To verify the marks for individual courses at that particular time.

Precondition:

1. Valid User
2. Availability of that courses in the database

Triggers: The Dept. admin has a need to recheck the marks of the students according to obtained marks.

Scenario:

1. Visit the login page and log in
2. Click on Courses
3. Click on Students
4. Search the result
5. Check the obtain marks
6. Click on Lock Mark
7. Changed the save
8. Proceed to the next activity

Exception: Absent

Priority: Essential, must be implemented

When Available: First increment

iv.

Use Case: Publish Results

Primary Actors: Dept. Admin

Context: To publish the results at that particular time.

Precondition:

1. Valid User
2. Availability of that courses in the database

Triggers: The Dept. Admin needs to publish the results at that particular time.

Scenario:

1. Visit the login page and log in
2. Click on Sessions
3. Select the Year
4. Click on Publish
5. Proceed to the next activity

Exception: Postponed

Priority: Essential, must be implemented

When Available: First increment

v.

Use Case: Download Transcript

Primary Actors: Student

Context: To get the hard copy at any places.

Precondition:

1. Valid User
2. Availability of that transcript in the database

Triggers: The student has a need to get the transcript for further purposes.

Scenario:

1. Visit the login page and log in
2. Click on My Profile
3. Click on Results
4. Search year/semester
5. Click on Show Grade
6. Click on Download
7. Proceed to the next activity

Exception: Data is not found

Priority: Essential, must be implemented

When Available: First increment

UML DIAGRAM

3.1.3 Use Case Diagrams (Level 0, Level 1 & Level 2)

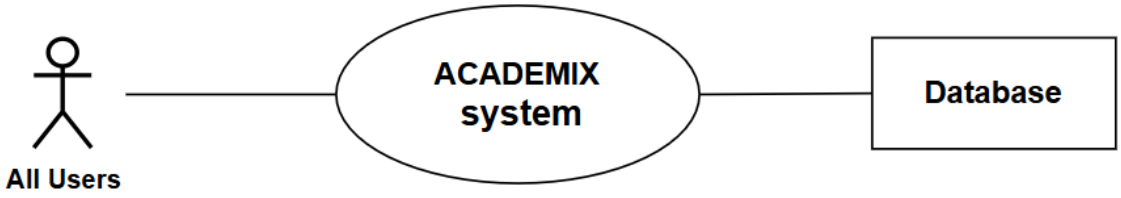


Fig : Level 0 for ACADEMIX system

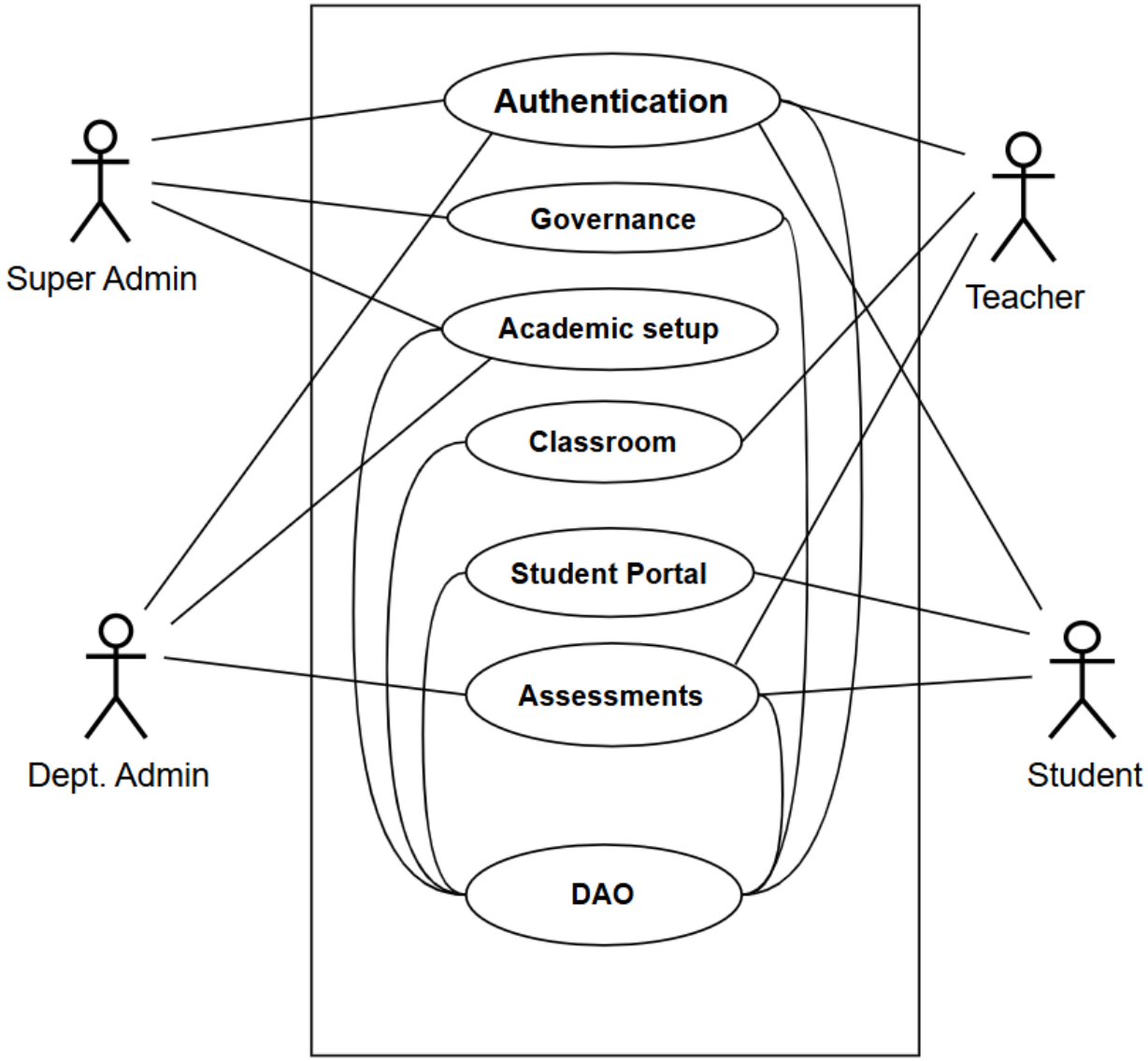


Fig : Level 1 for ACADEMIX system

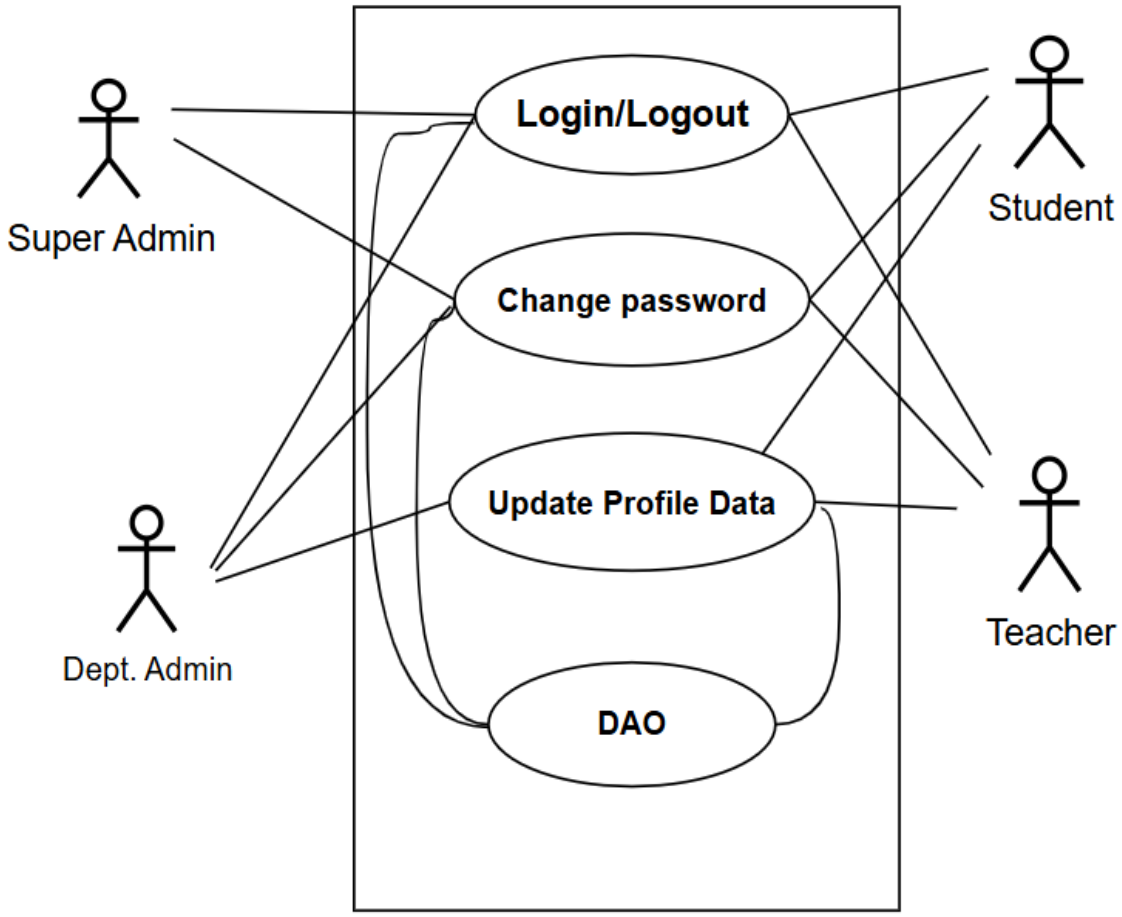


Fig : Level 2.1 for ACADEMIX system

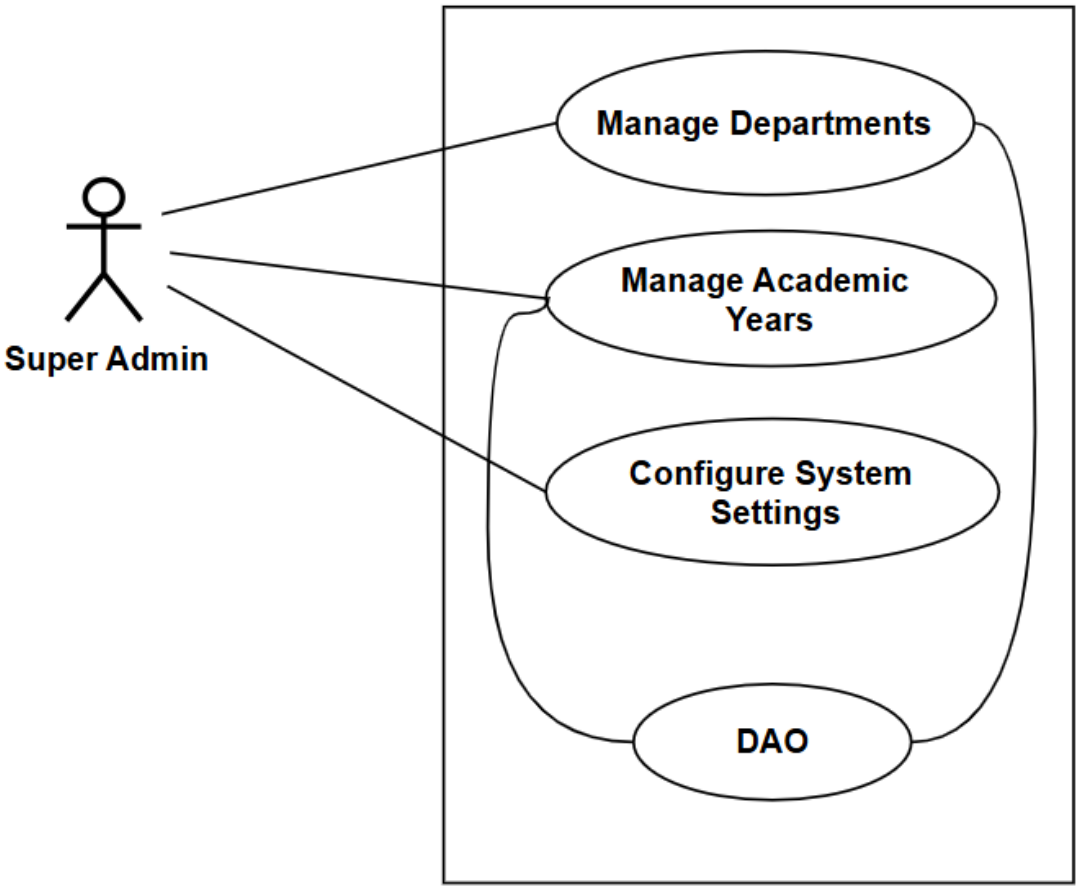


Fig : Level 2.2 for ACADEMIX system

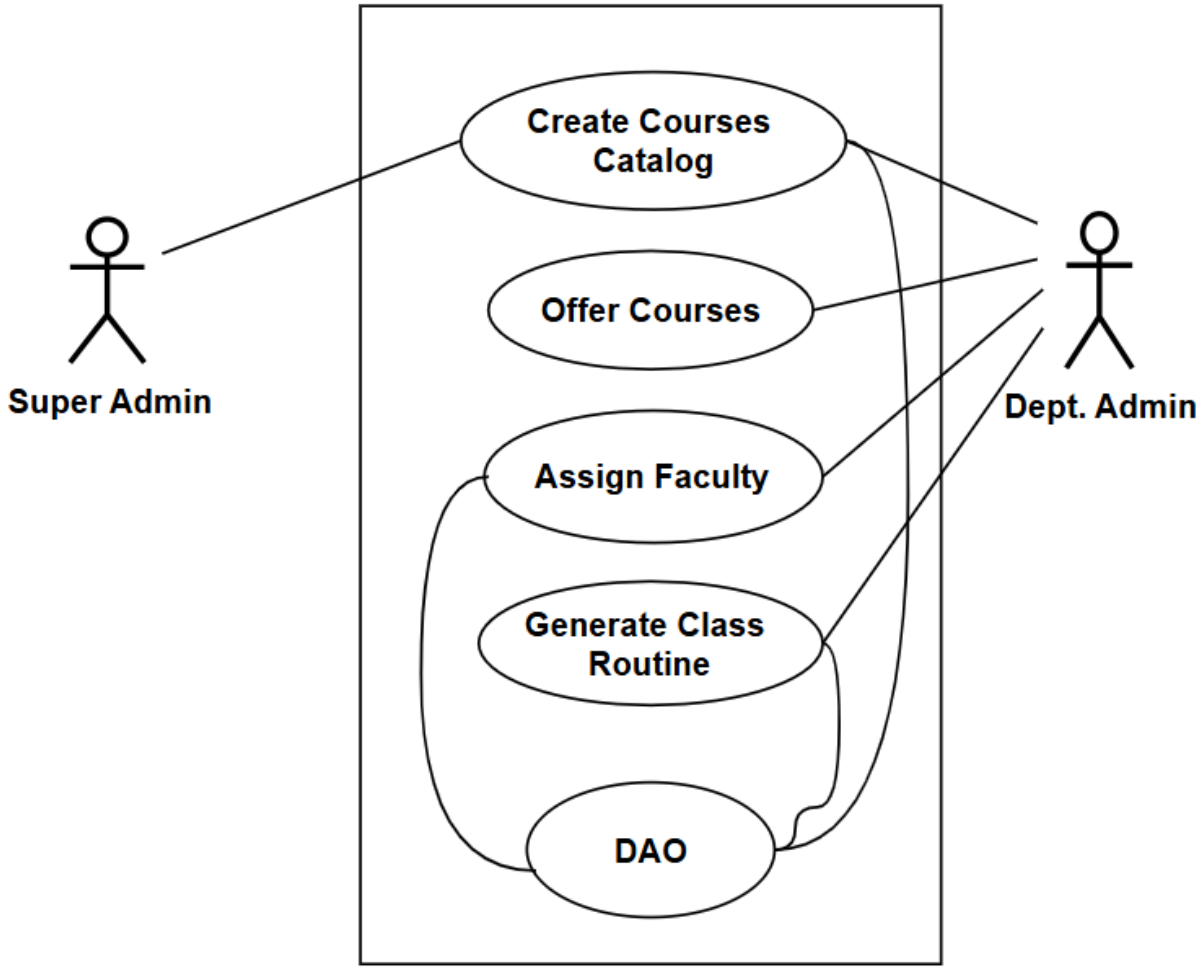


Fig : Level 2.3 for ACADEMIX system

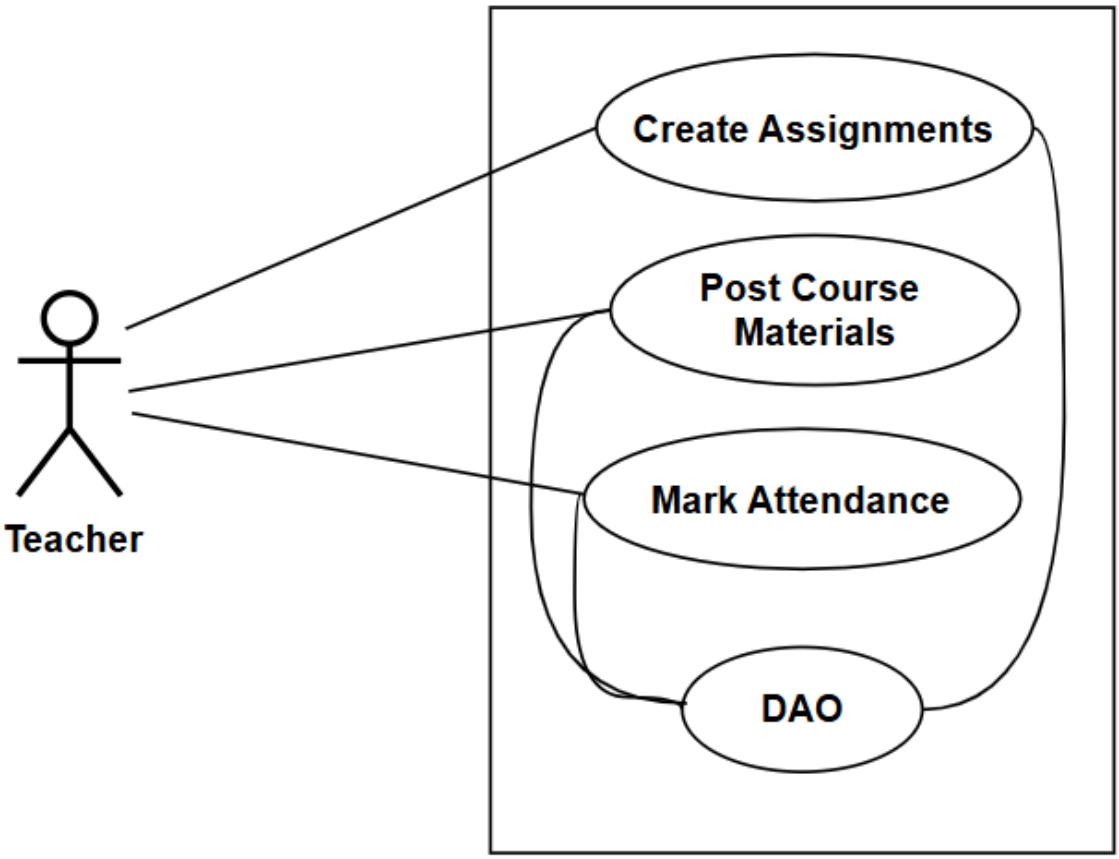


Fig : Level 2.4 for ACADEMIX system

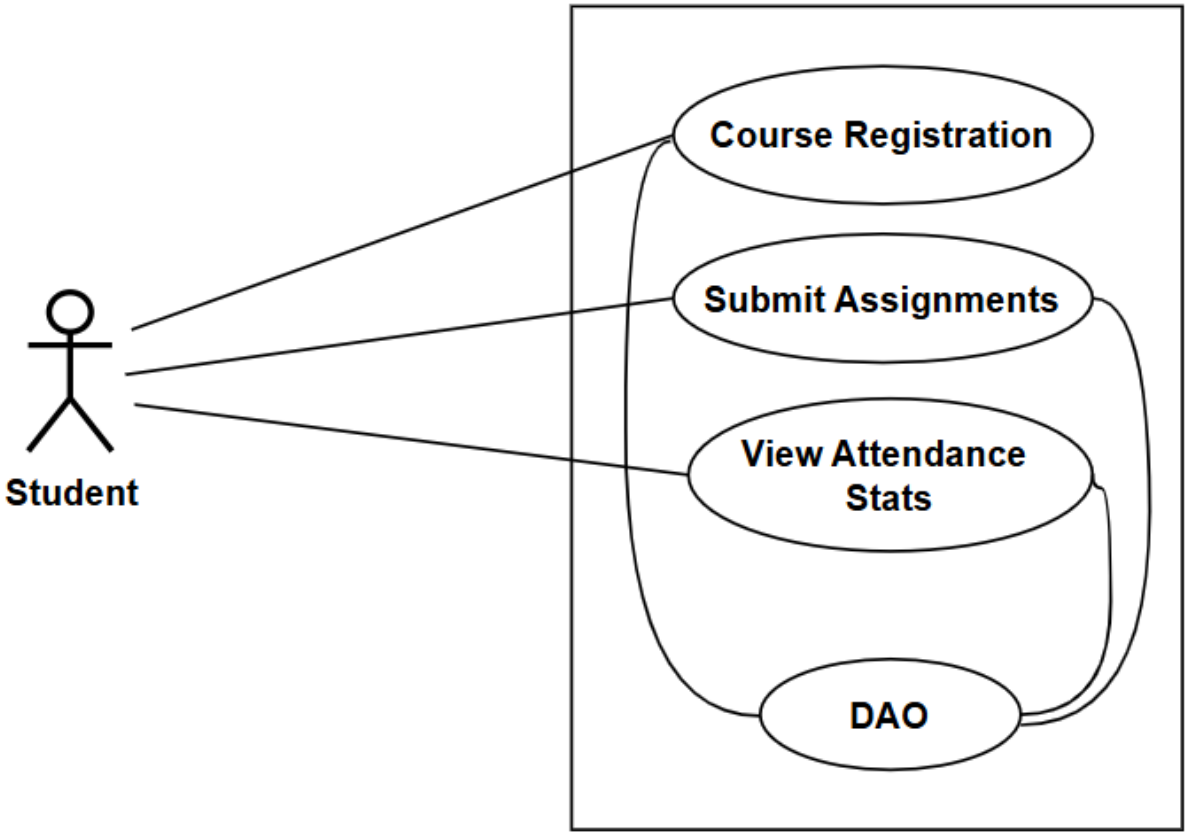


Fig : Level 2.5 for ACADEMIX system

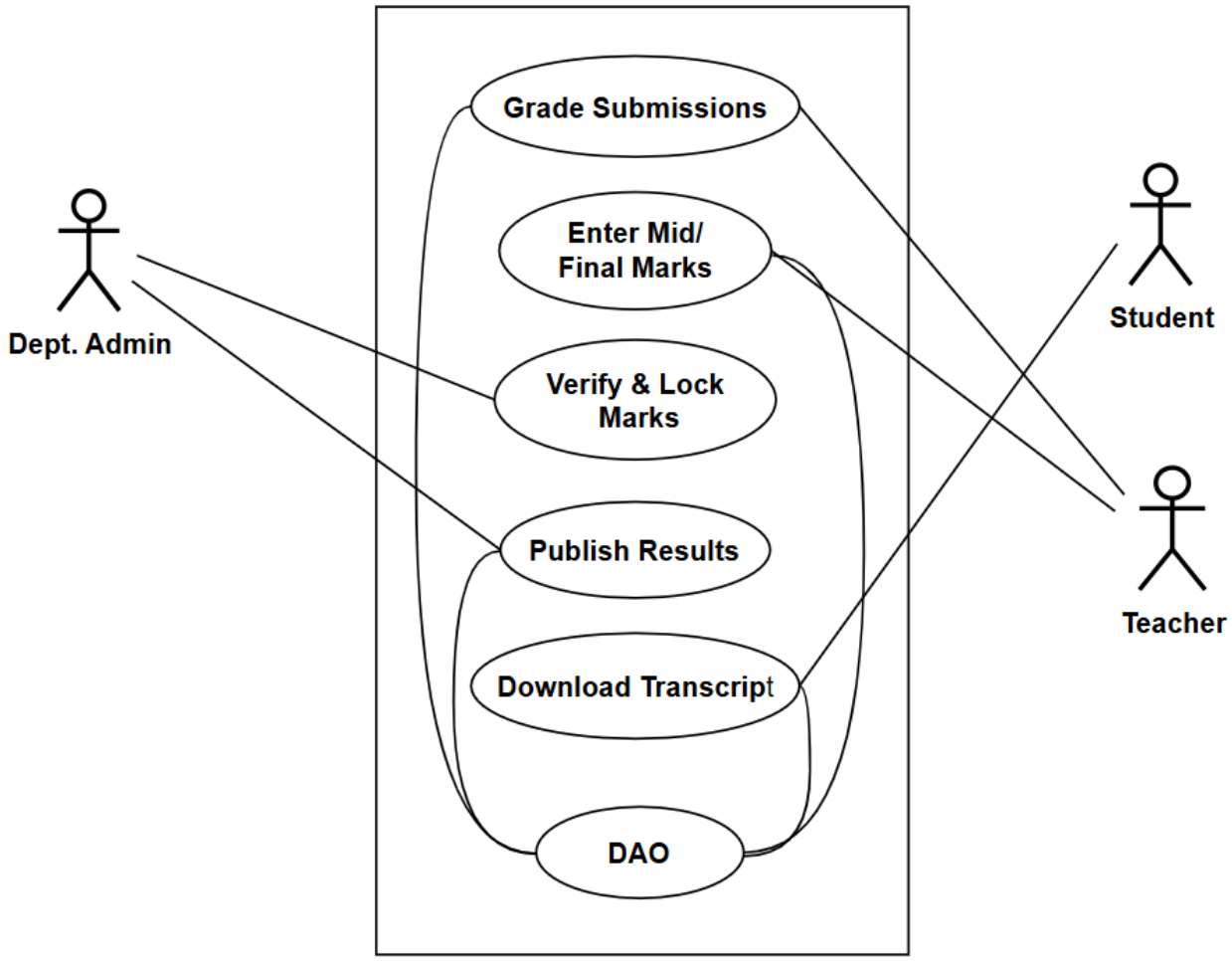


Fig : Level 2.6 for ACADEMIX system

3.2 Activity & Swimlane Modeling

3.2.1 Activity & Swimlane Diagrams (Process Flows)

Use case-1: Authentication & Session Management

Activity Diagram:

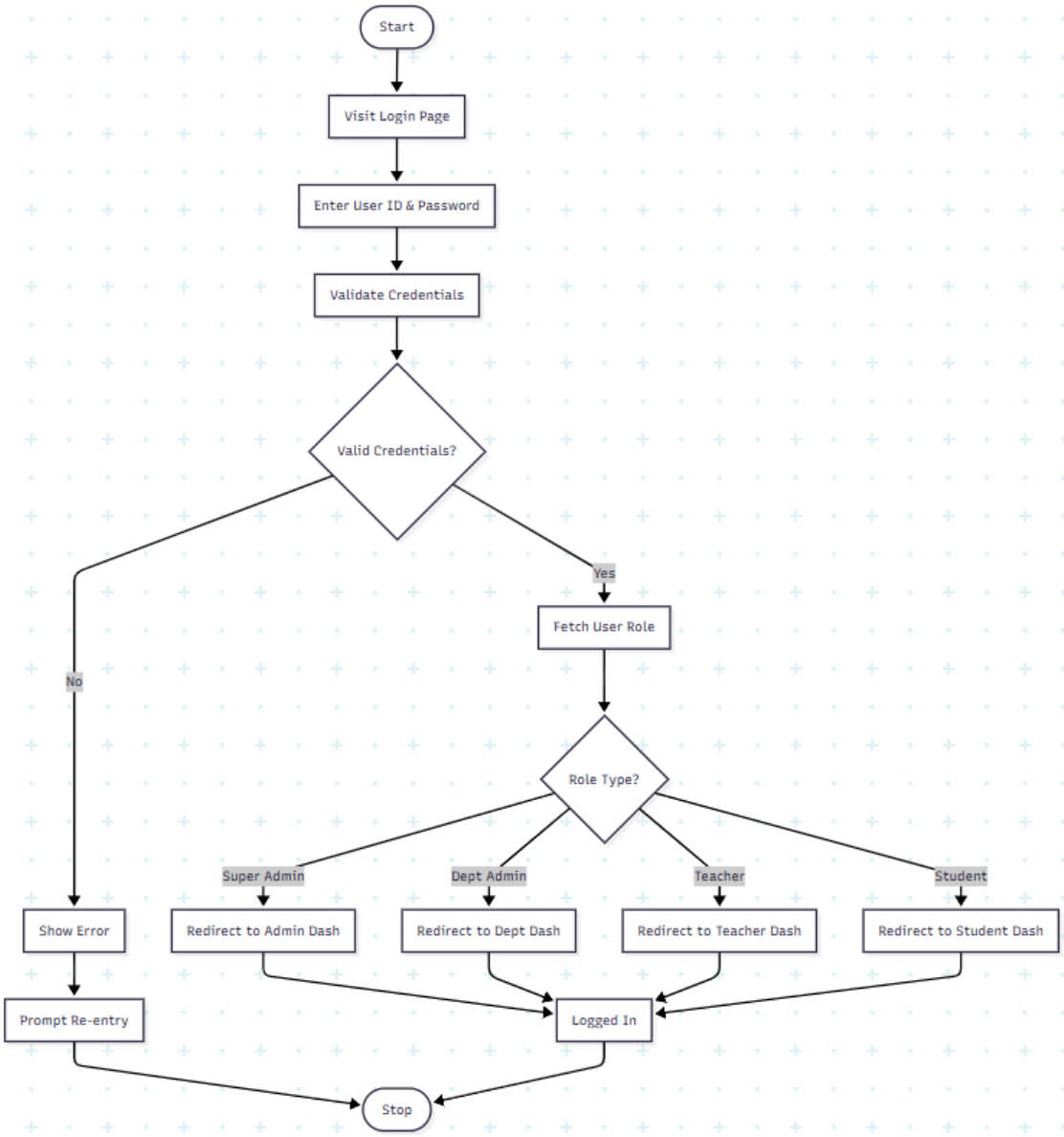


FIG-1: Authentication & Session Management

Swimlane Diagram:

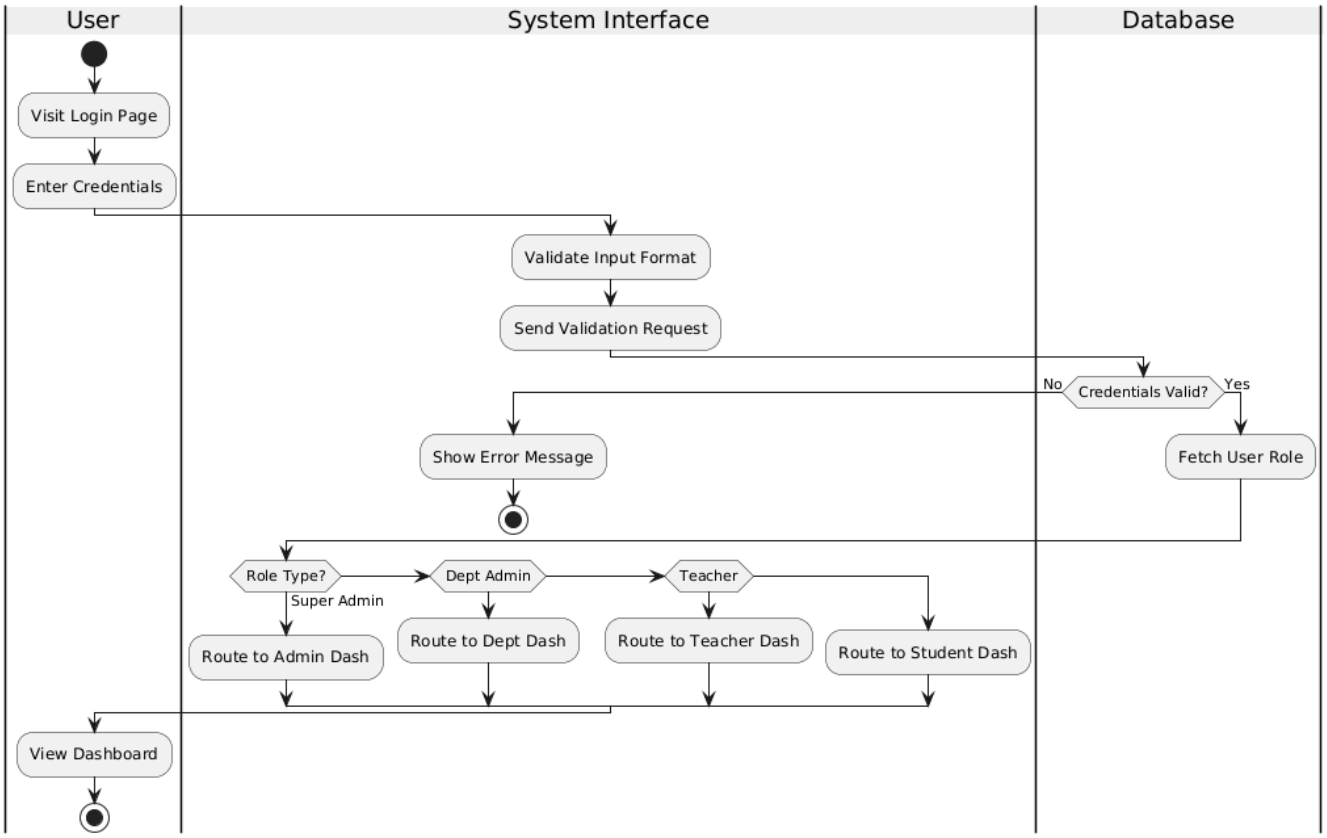


FIG-1: Authentication & Session Management

Use Case 2: Governance Management (Super Admin)

Activity Diagram:

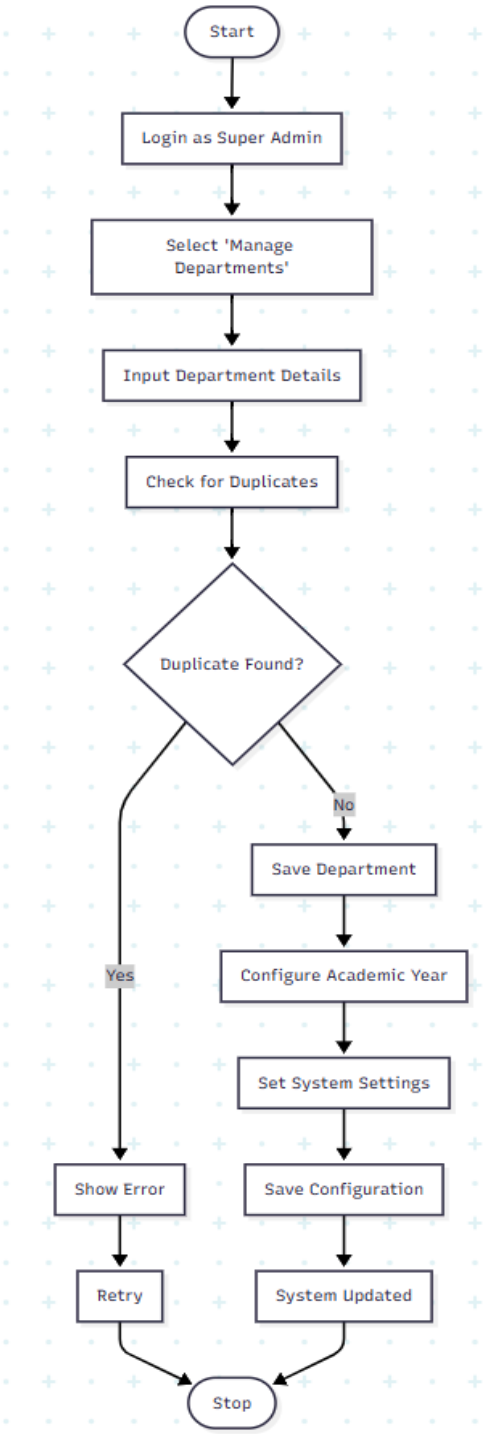


FIG-2: Governance Management (Super Admin)

Use Case 2: Governance Management

Swimlane Diagram:

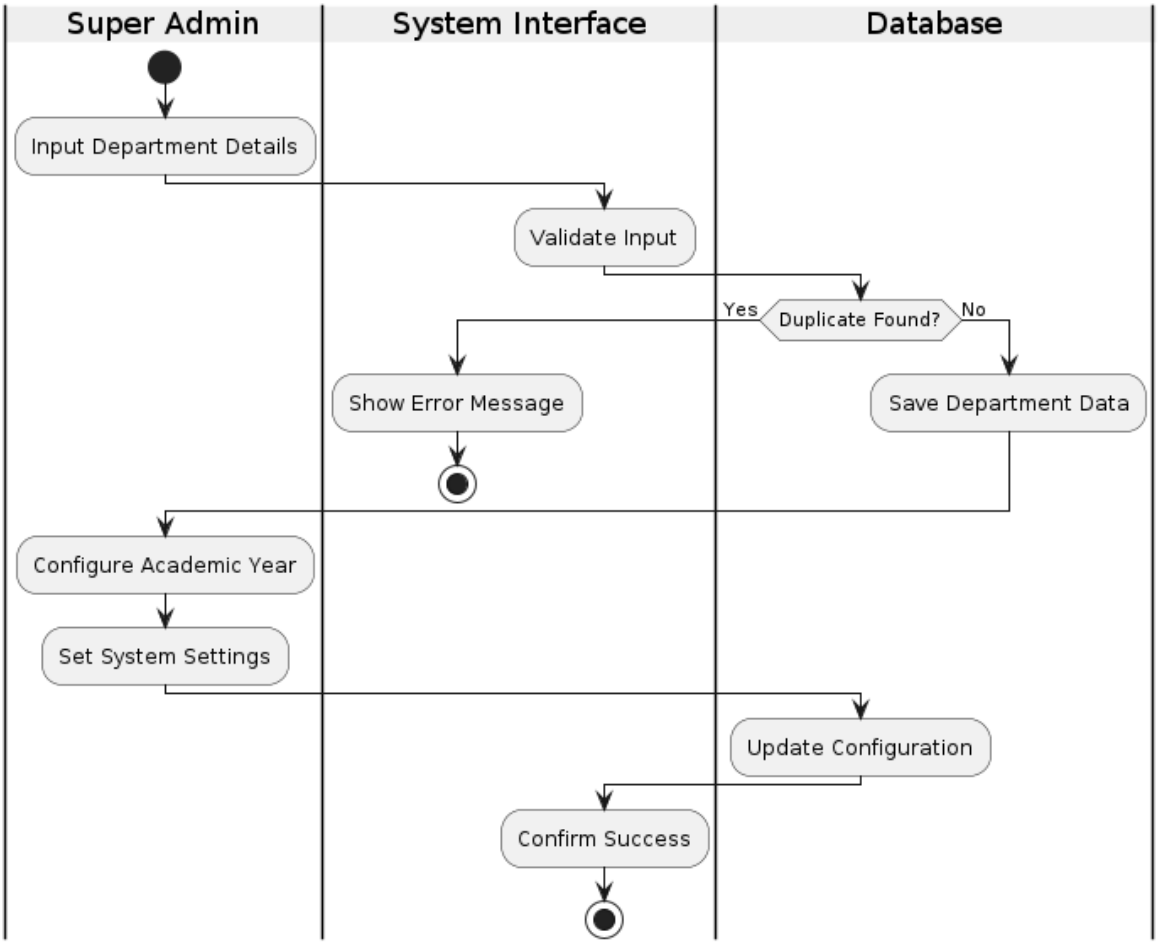


FIG-2: Governance Management (Super Admin)

Use Case 3: Academic Setup (Dept Admin)

Activity Diagram:

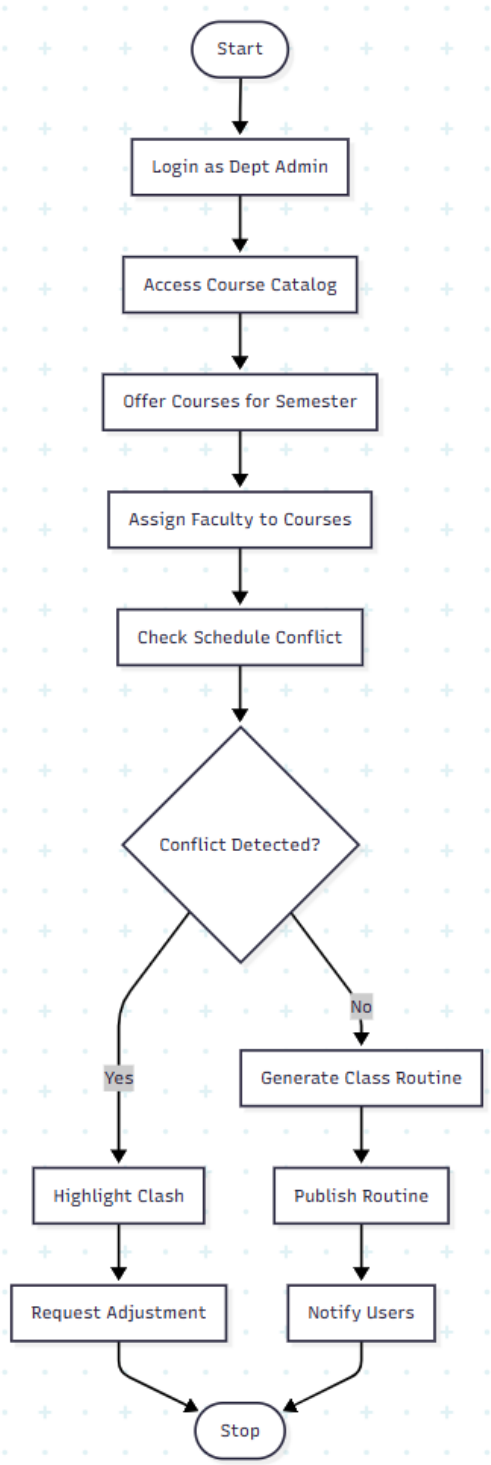


FIG-3: Academic Setup (Dept Admin)

Use Case 3: Academic Setup

Swimlane Diagram:

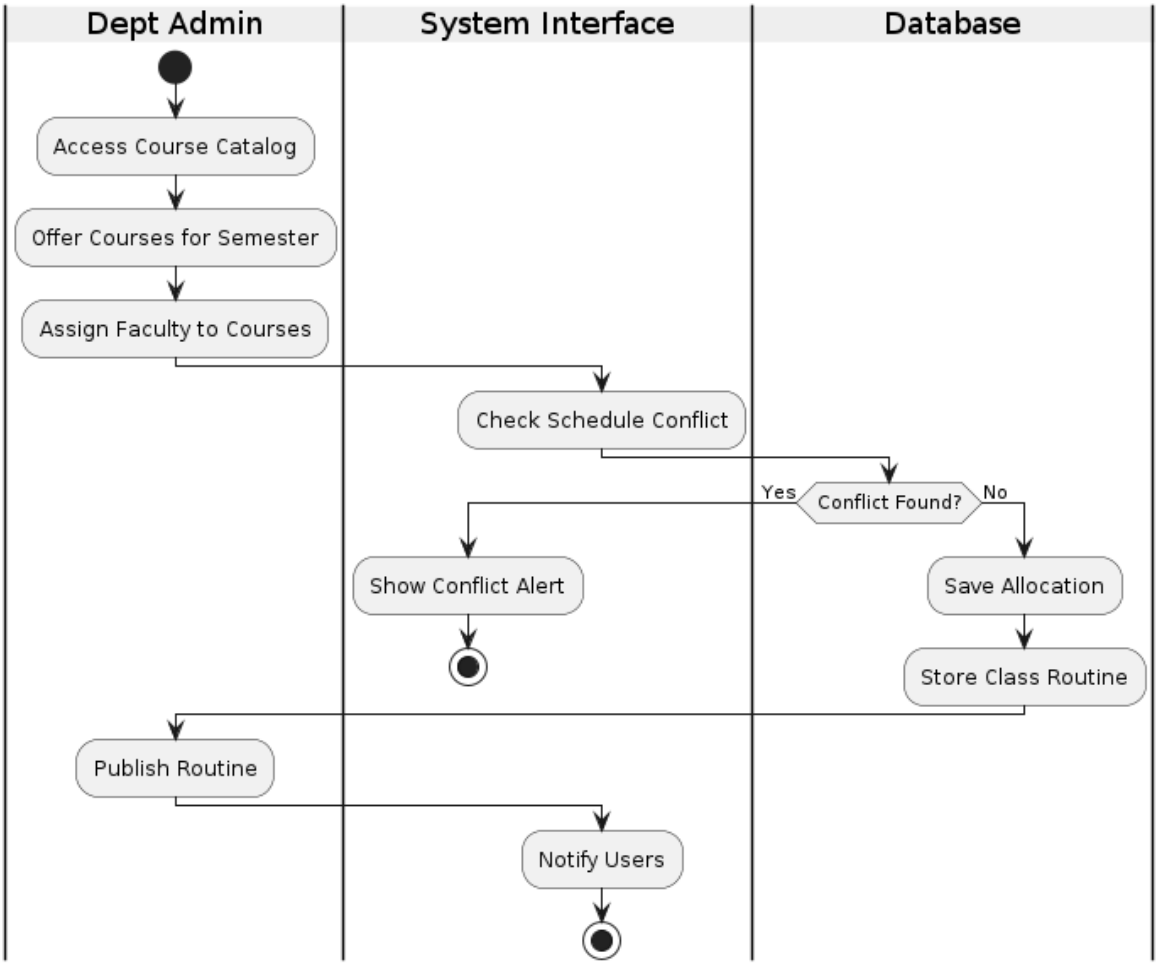


FIG-3: Academic Setup (Dept Admin)

Use Case 4: Classroom Management (Teacher)

Activity Diagram:

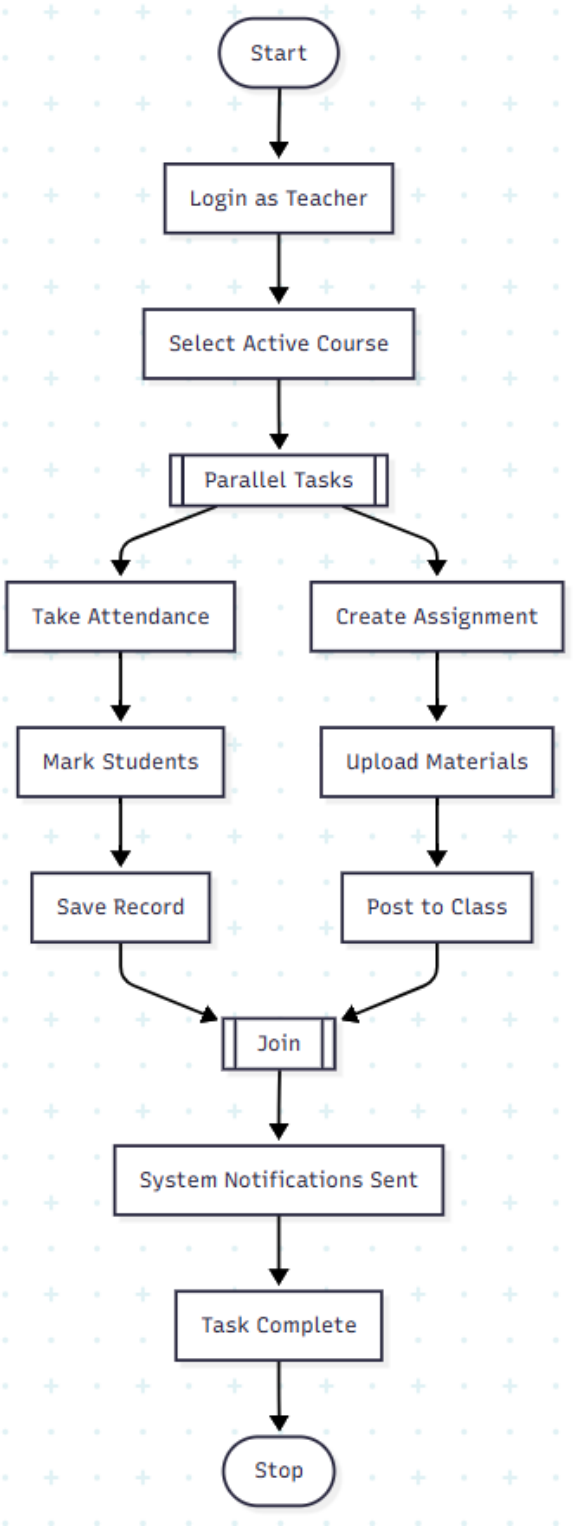


FIG-4: Classroom Management (Teacher)

Use Case 4: Classroom Management

Swimlane Diagram:

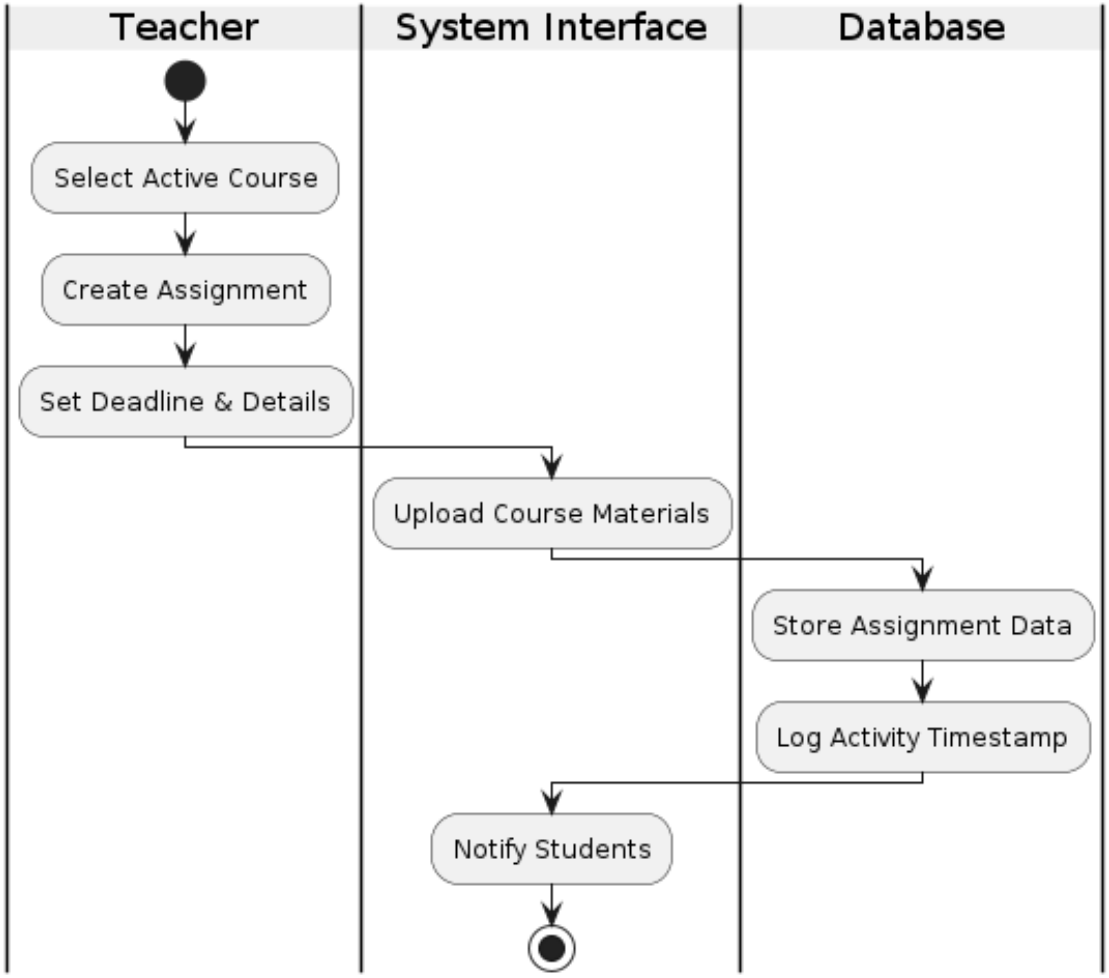


FIG-4: Classroom Management (Teacher)

Use Case 5: Student Portal (Student)

Activity Diagram:

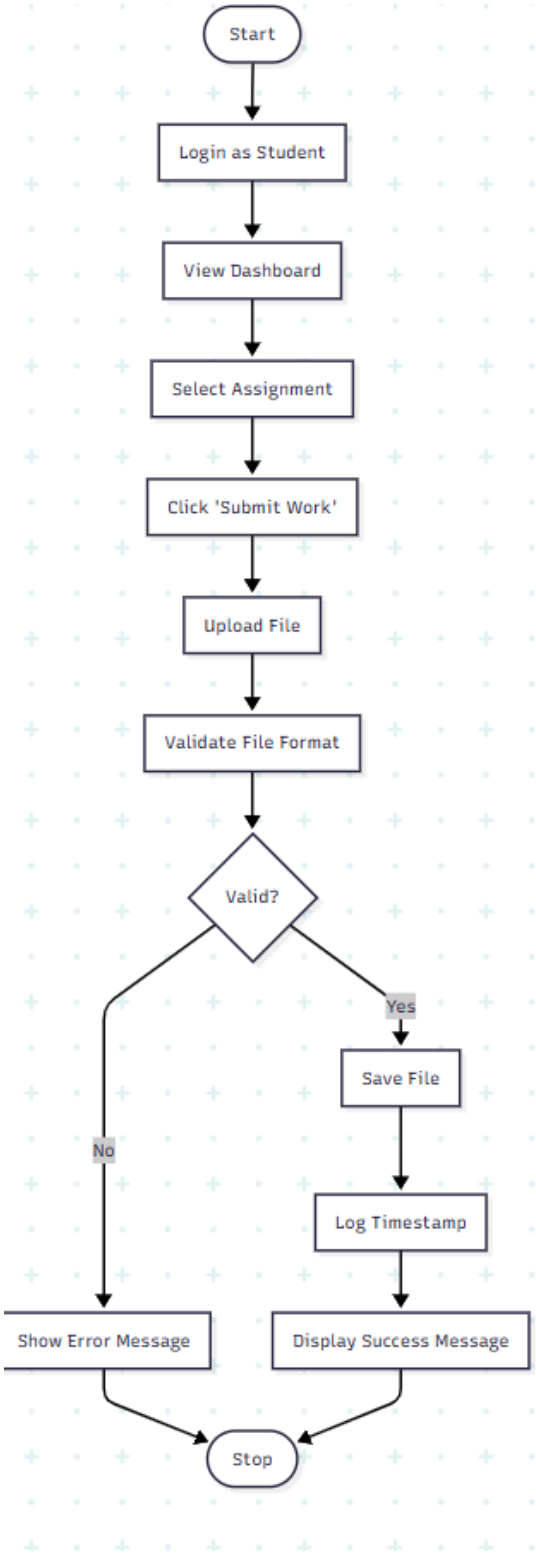


FIG-5: Student Portal (Student)

Use Case 5: Student Portal

Swimlane Diagram:

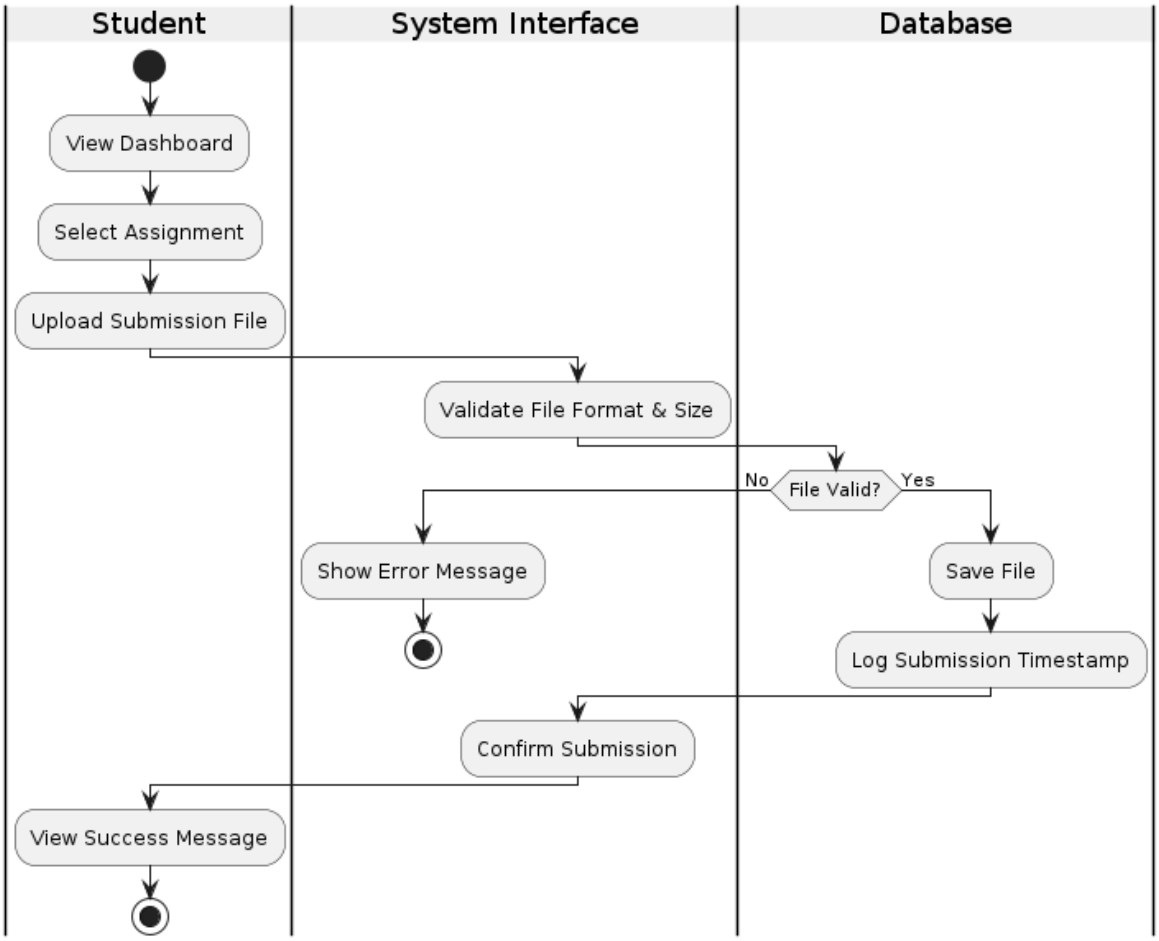


FIG-5: Student Portal (Student)

Use Case 6: Assessments & Result Processing

Activity Diagram:

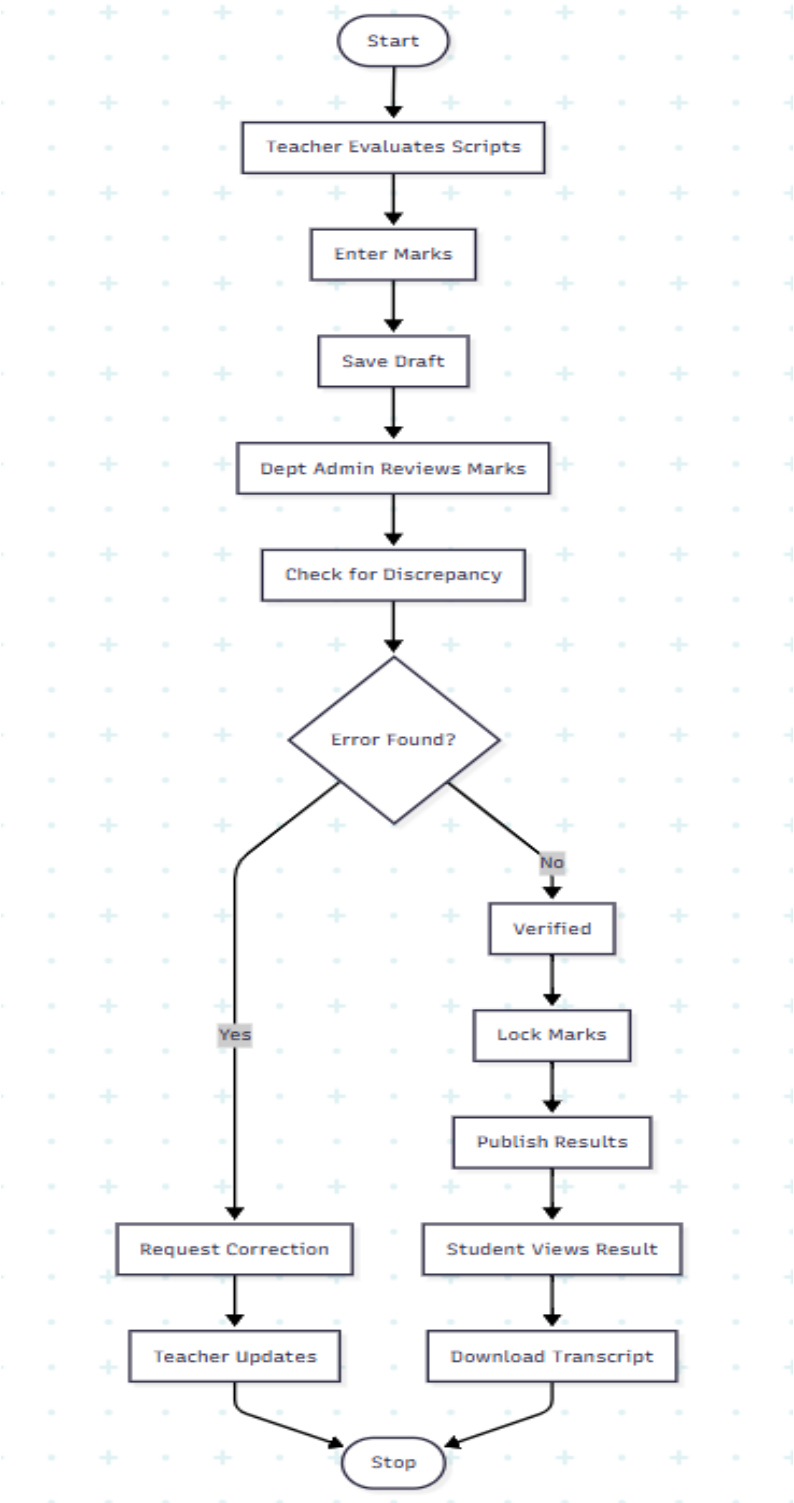


FIG-6: Assessments & Result Processing

Use Case 6: Assessments & Results

Swimlane Diagram:

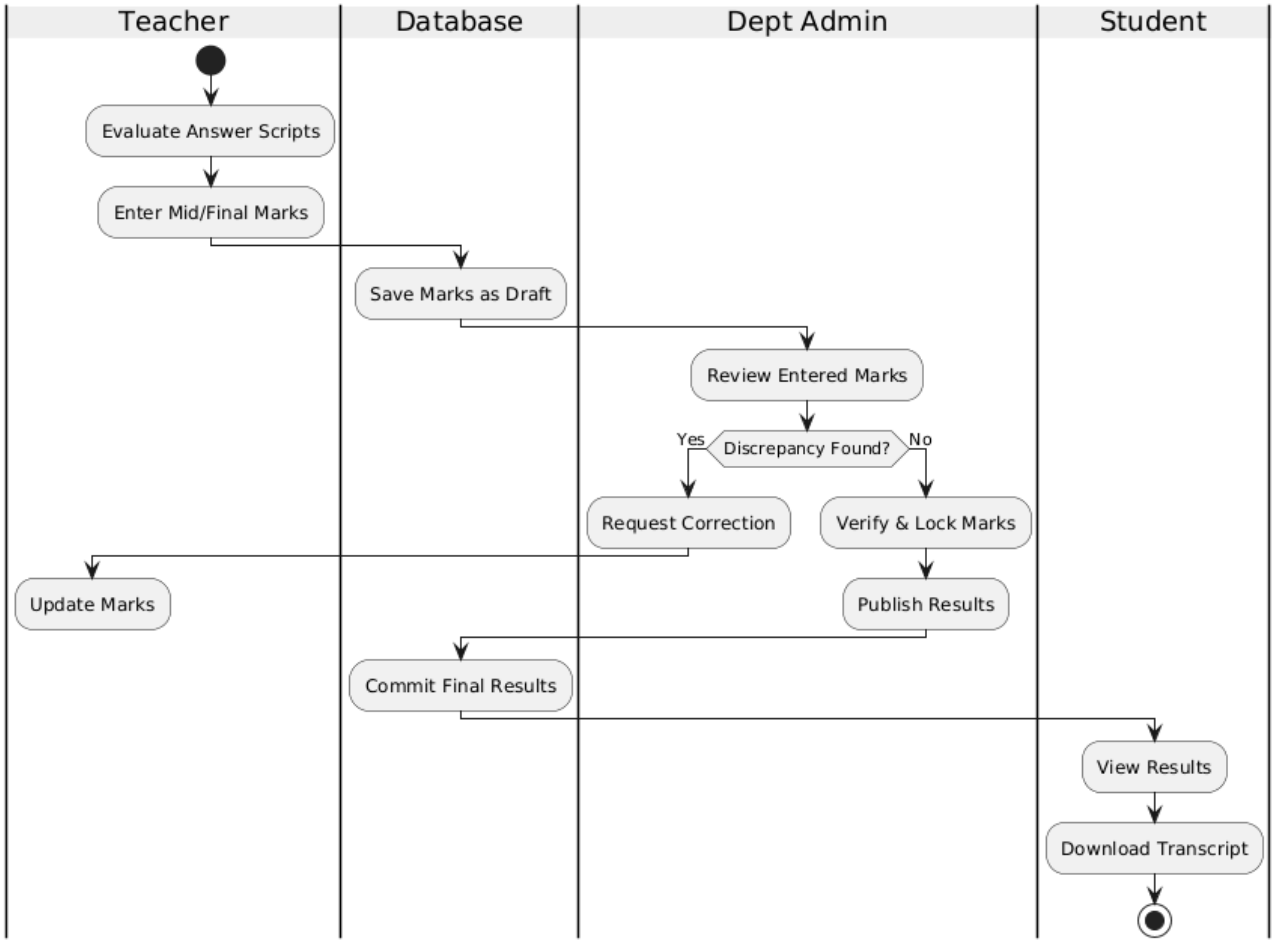
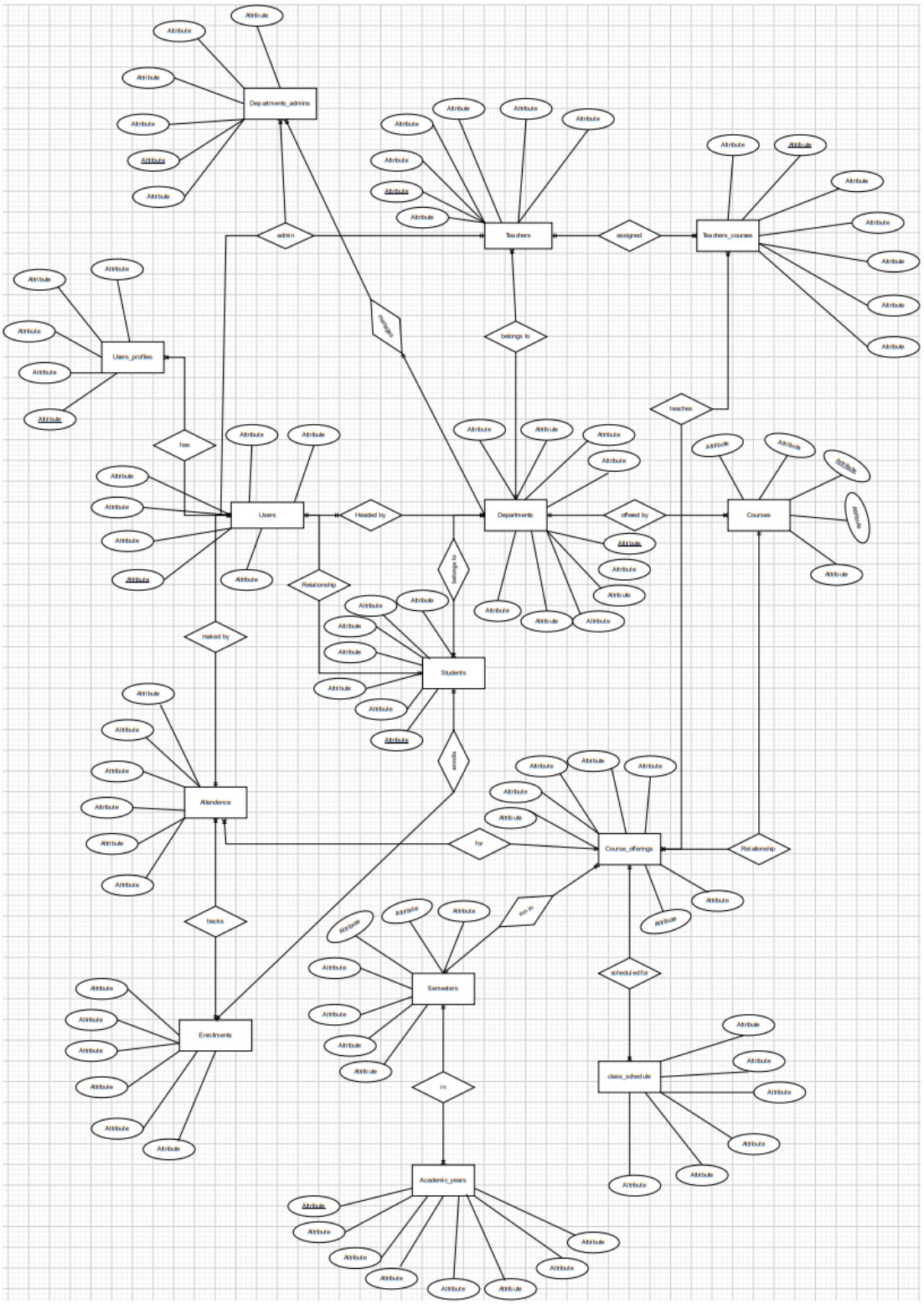


FIG-6: Assessments & Result Processing

4. Data Modeling

4.1 Entity Relationship Diagram (ERD)



5. Class Based Modeling

5.1 Identify Class and Attributes

Step-1: Identifying and categorize all nouns

External Entities	Student, Teacher, Super Admin, Department Admin, User, University, Institute
Things	Course, Department, Batch, Semester, Class Routine, Attendance, Assignment, Grade, Result, Admit Card, Certificate, Document, Notice, Event, Syllabus, Lecture Note, Study Material, Profile, Notification, Report, Analytics, Dashboard
Occurrence or events	Login, Logout, Attendance Taking, Assignment Submission, Assignment Grading, Result Publishing, Document Issuing, Re-issuing, Revoking, Course Assignment, Class Reschedule, Notification Trigger
Roles	Super Admin, Department Admin, Teacher, Student
Organizational units	University, Department, Batch, Semester, Course
Places	Classroom, Department Office, Online System, Database
Structures	ACADEMIX System, Server, Database, Cloud Storage, Security Log, Backup System

Step-2: Selection of potential class

1. Retained information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations
6. Essential requirements

Potential Class	Characteristic Number That Applies
Student	Accepted: all apply
Teacher	Accepted: all apply
Super Admin	Accepted: all apply
Department Admin	Accepted: all apply
User	Accepted: all apply
Department	Accepted: all apply
Course	Accepted: all apply
Batch	Accepted: all apply
Semester	Accepted: all apply
Class Routine	Accepted: 1,2,3,5,6 apply
Attendance	Accepted: all apply
Assignment	Accepted: all apply
Result	Accepted: all apply
Grade	Accepted: 1,2,3,6 apply
Admit Card	Accepted: all apply
Certificate	Accepted: all apply
Document	Accepted: all apply
Notice	Accepted: 1,2,5,6 apply
Event	Accepted: 1,2,5,6 apply
Notification	Accepted: 1,2,5,6 apply
Syllabus	Accepted: 1,3,6 apply
Study Material	Accepted: 1,3,6 apply
Profile	Accepted: 1,3,4,6 apply
Analytics	Accepted: 1,2,6 apply
Report	Rejected: 3 fails
Interface	Rejected: 1,4,5 fails
Button	Rejected: 1,3,5,6 fails

Password	Rejected: 3 fails
Login	Rejected: 3 fails
Update	Rejected: 3 fails
Server	Rejected: 3 fails
Internet	Rejected: 3 fails
Database	Rejected:2,3, 4, 5, 6 fails
System	Rejected:1, 3, 4, 5, 6 fails

5.2 Class Diagram

5.2.1 Class Structure & Relationships

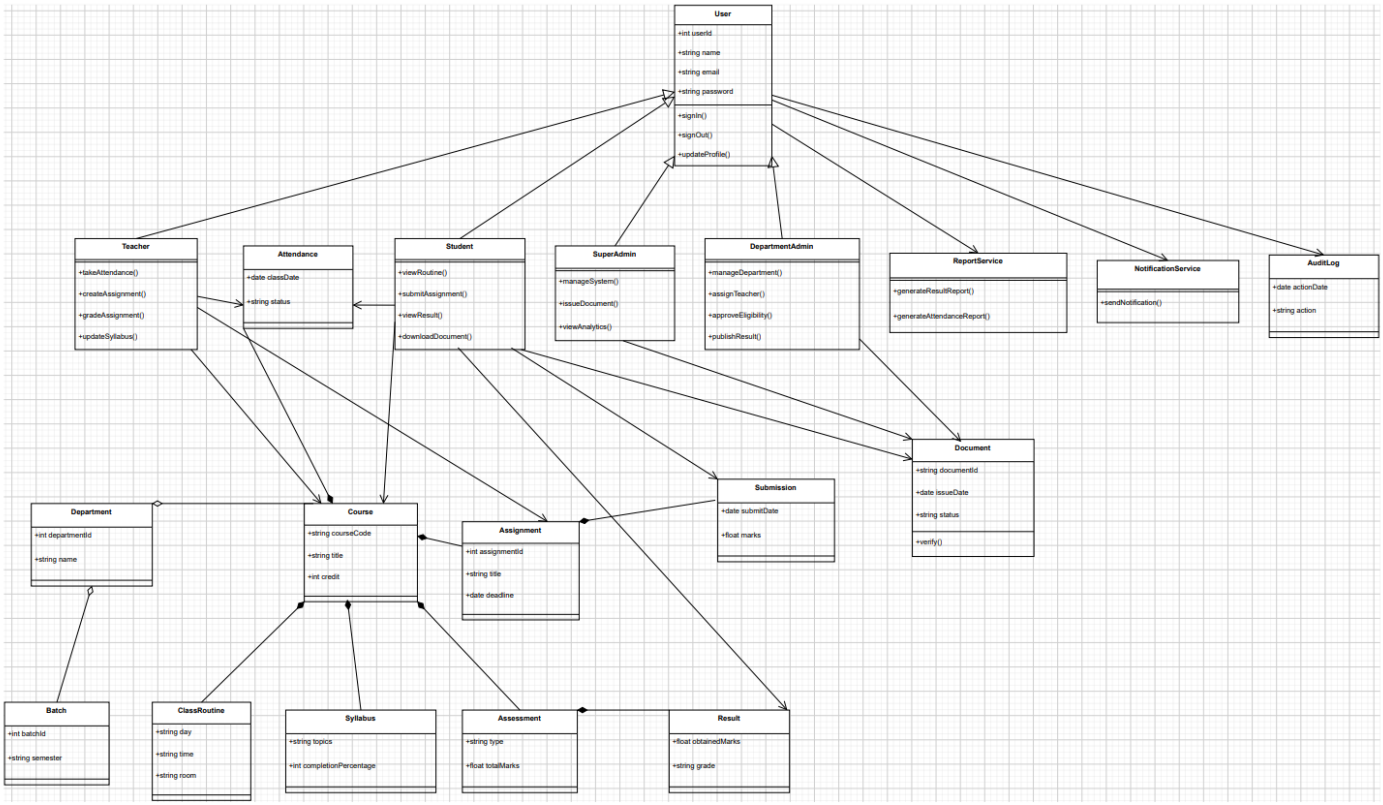


Fig: Class Based Model for ACADEMIX System

6. Flow Oriented Modeling

6.1 Data Flow Diagrams (DFD)

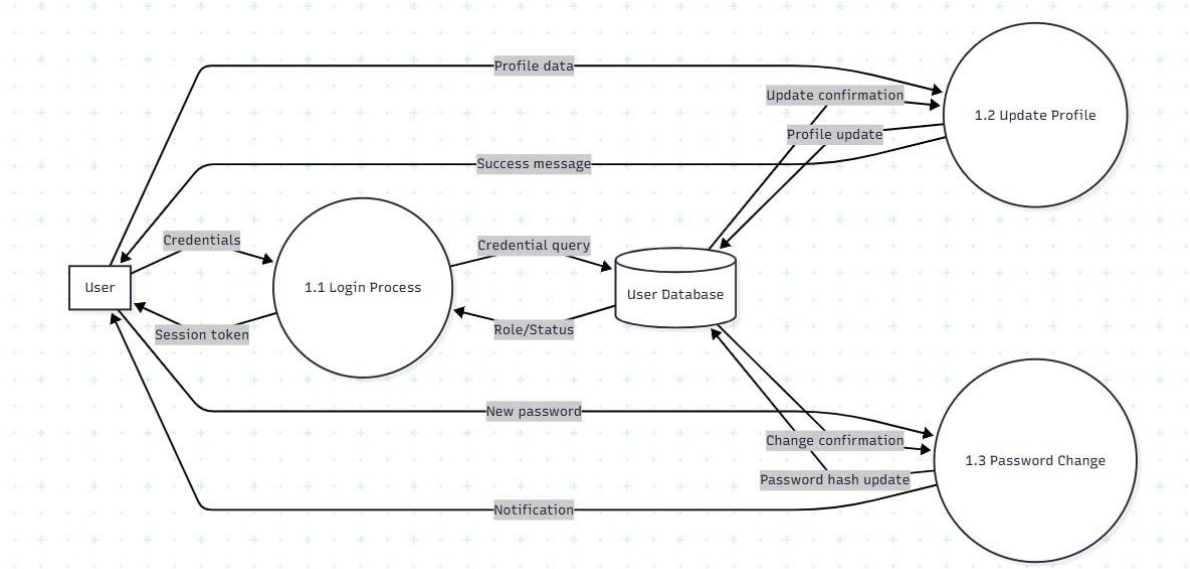


Figure-1: Level 0 for ACADEMIX system

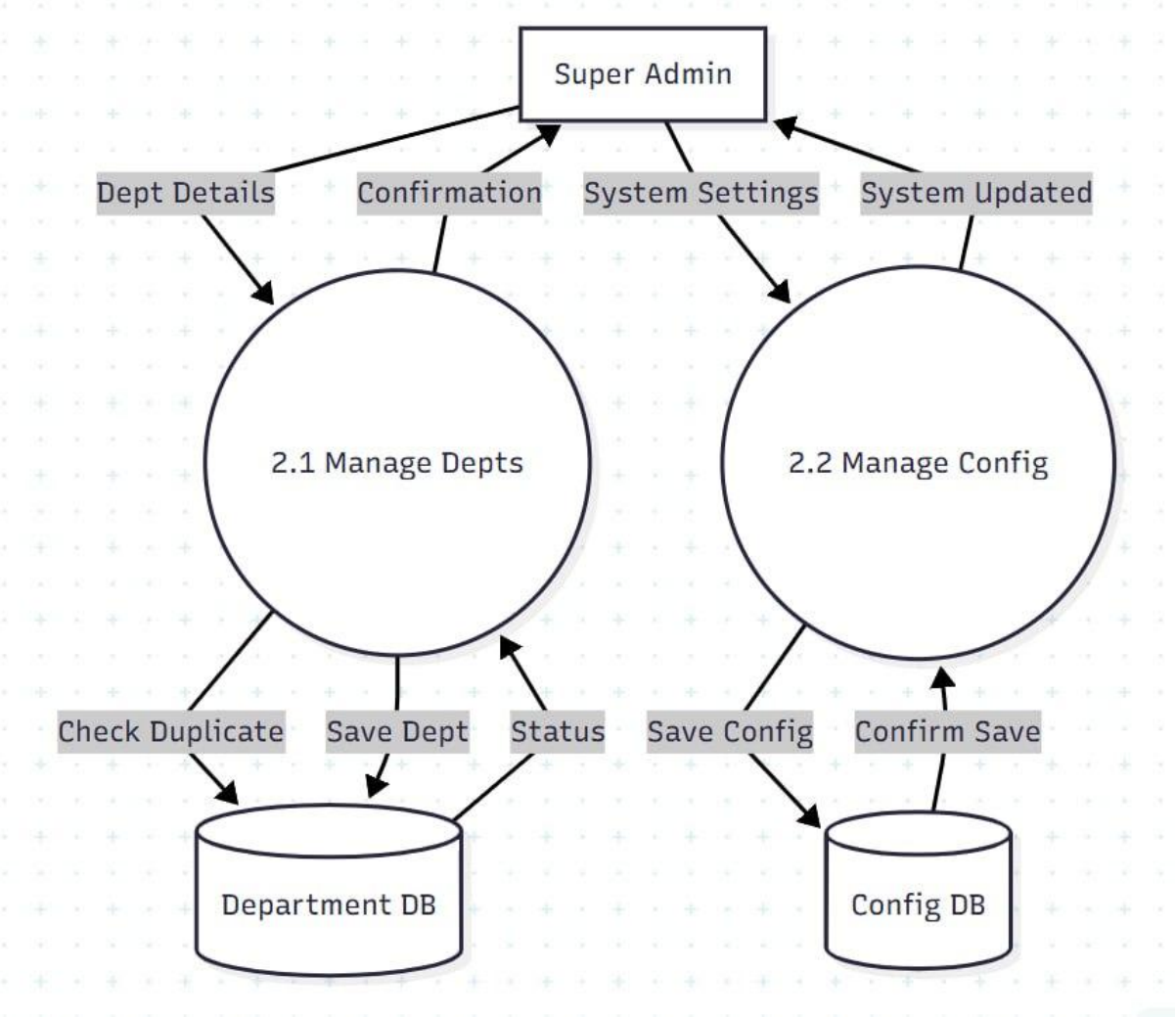


Figure-2: Level 1.1(Super Admin) for ACADEMIX system

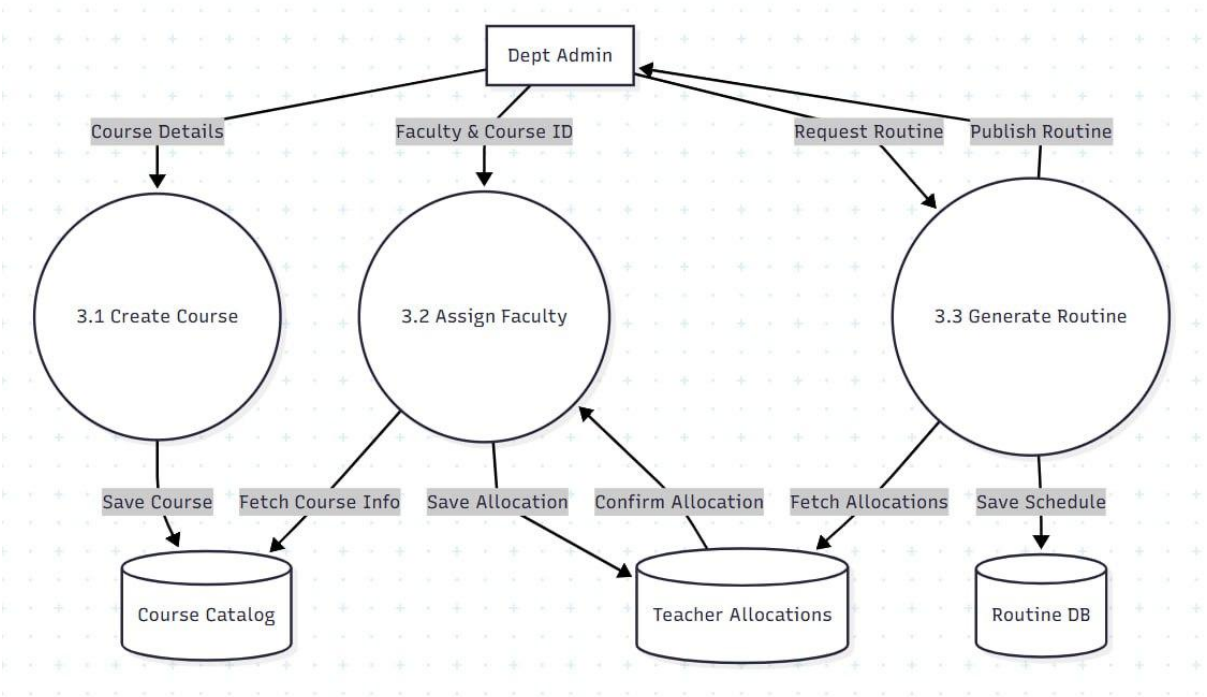


Figure-3: Level 1.2(Dept Admin) for ACADEMIX system

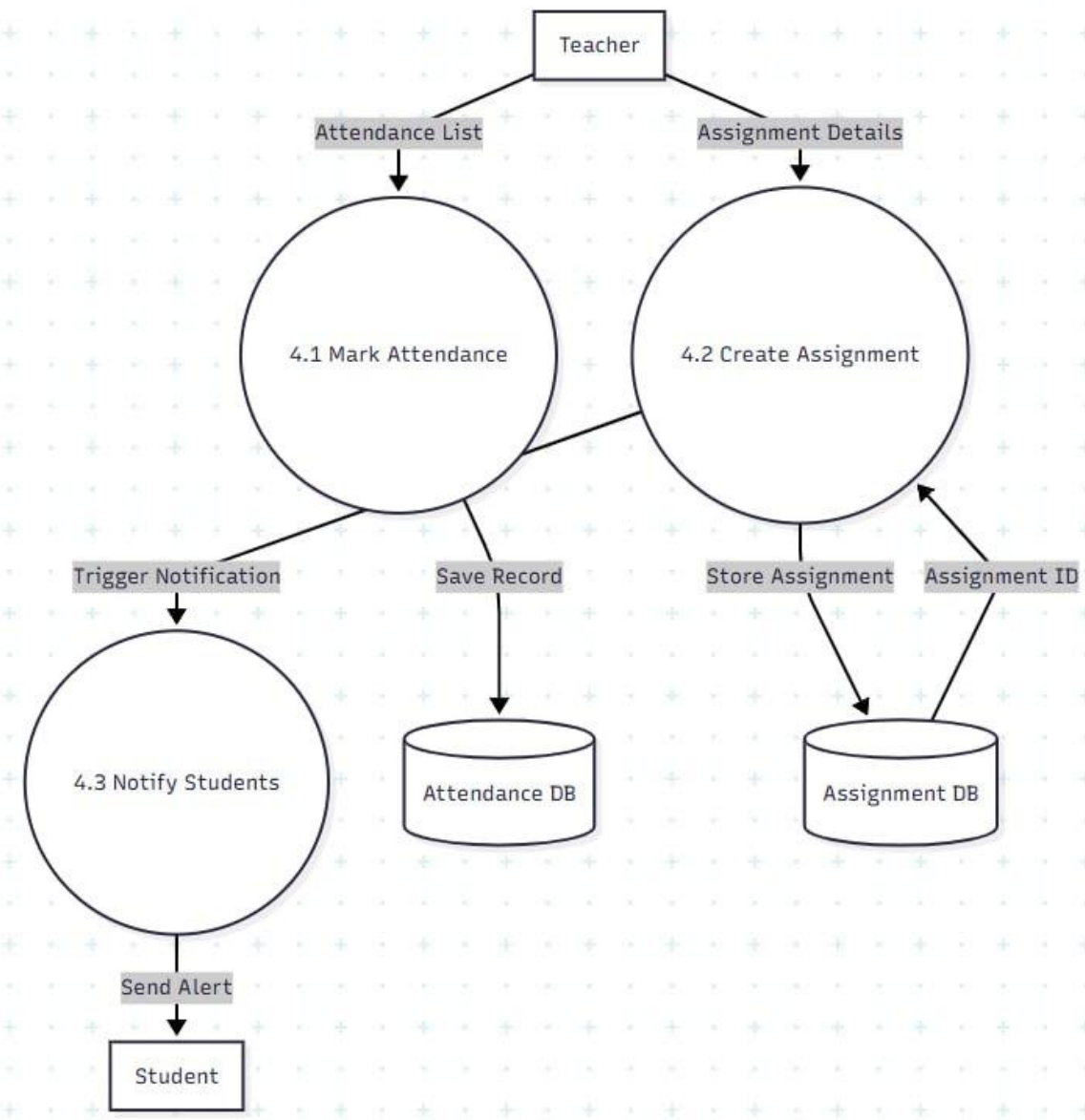


Figure-4: Level 1.3(Teacher) for ACADEMIX system

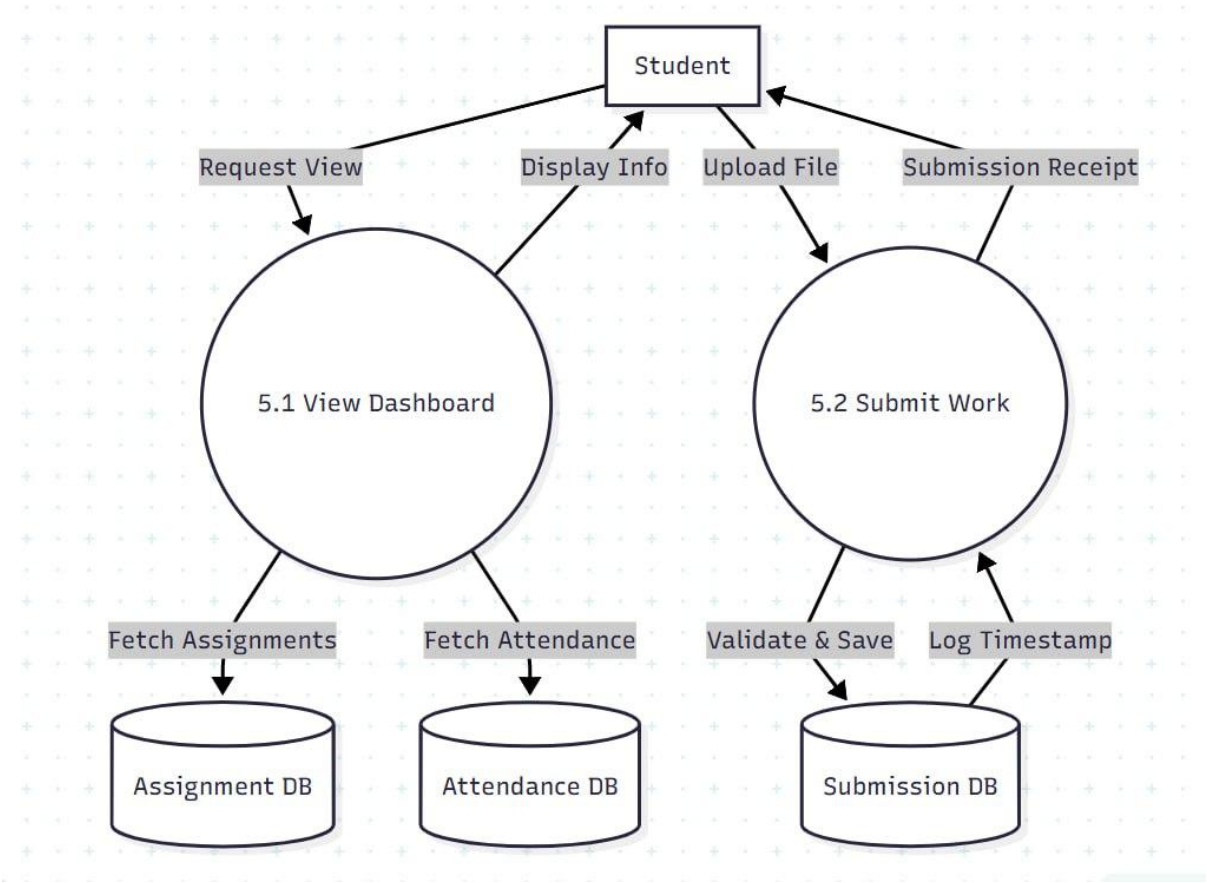


Figure-5: Level 1.4(Student) for ACADEMIX system

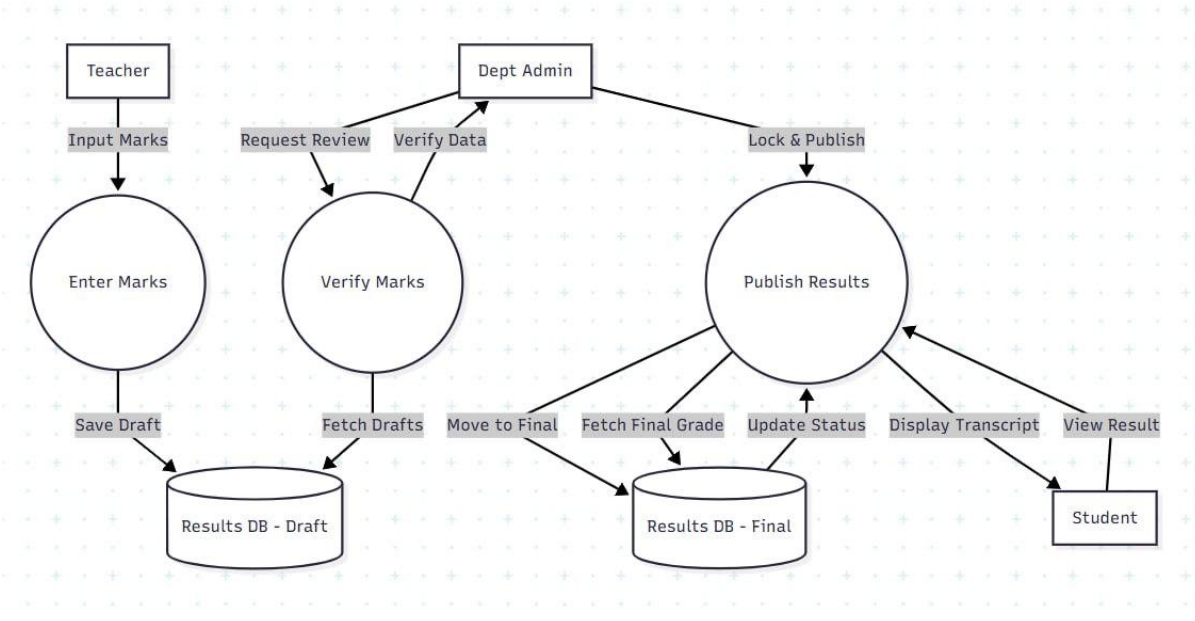


Figure-6: Level 2.1(Teacher) for ACADEMIX system

7. Behavioral Modeling

7.1 State Transition Modeling

7.1.1 State Transition Diagram

ID: 01
Name: Authentication

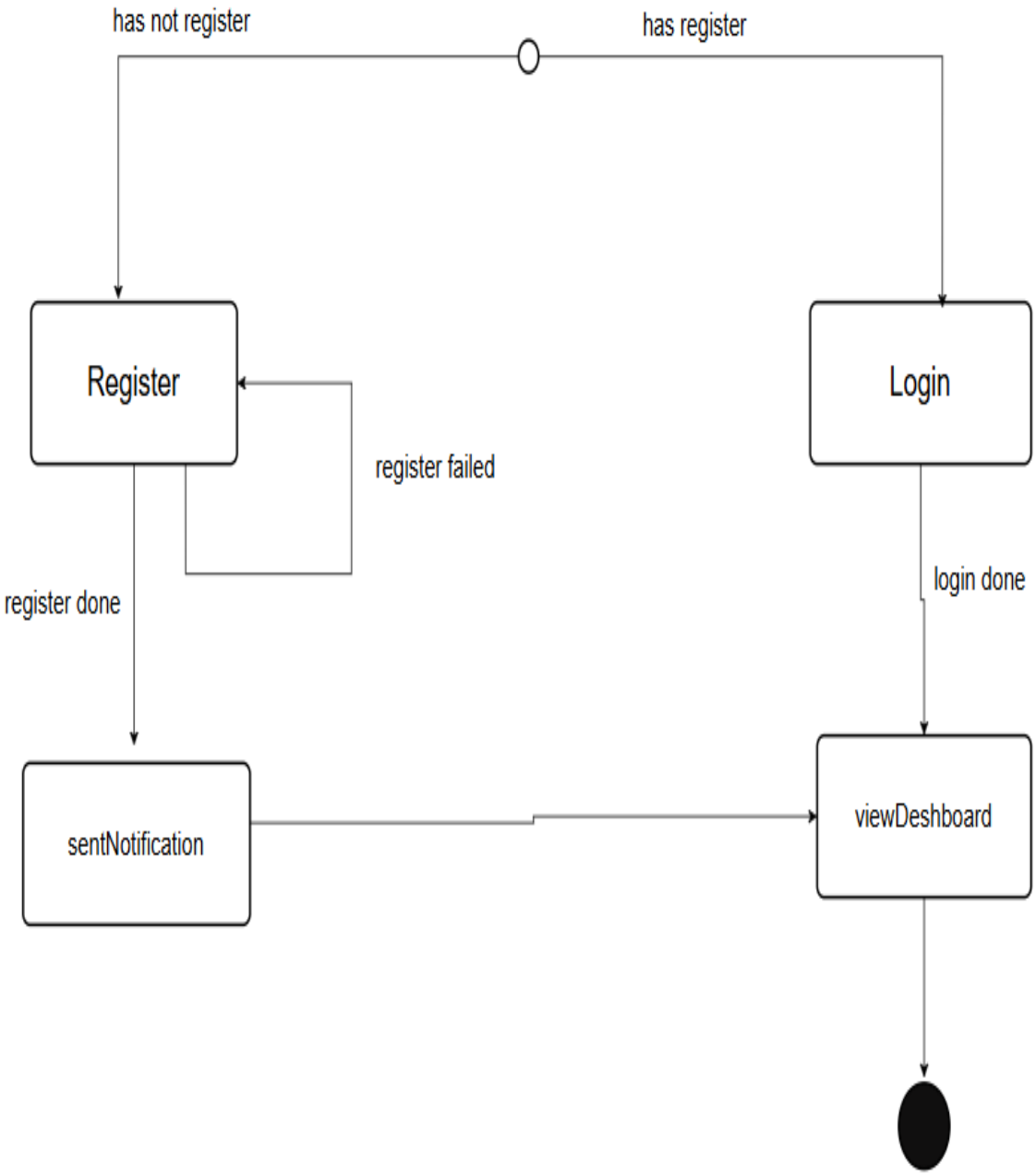


Fig-1: Authentication

ID: 02

Name: Super Admin

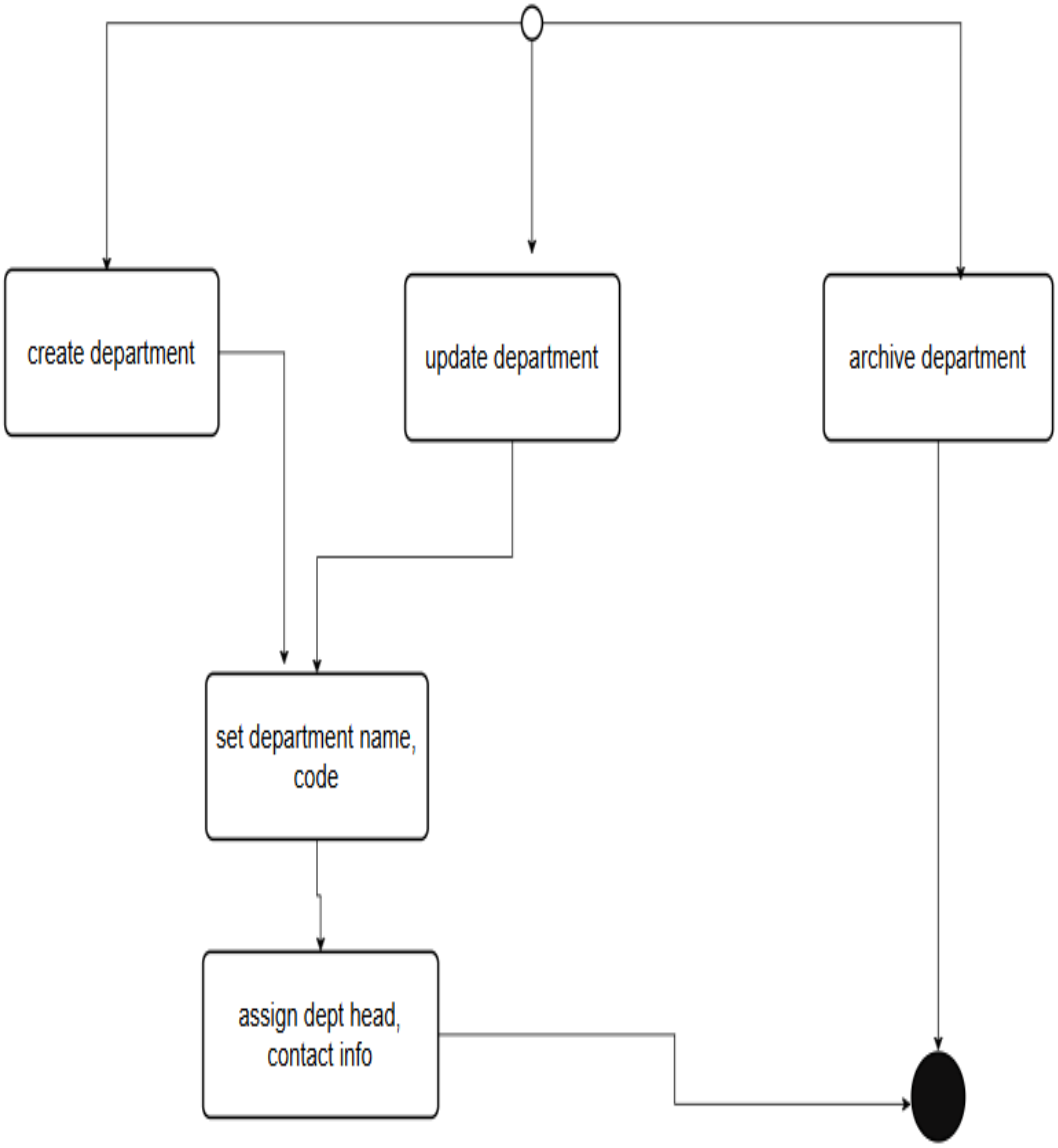


Fig-2: Super Admin

ID: 03
Name: Schedules

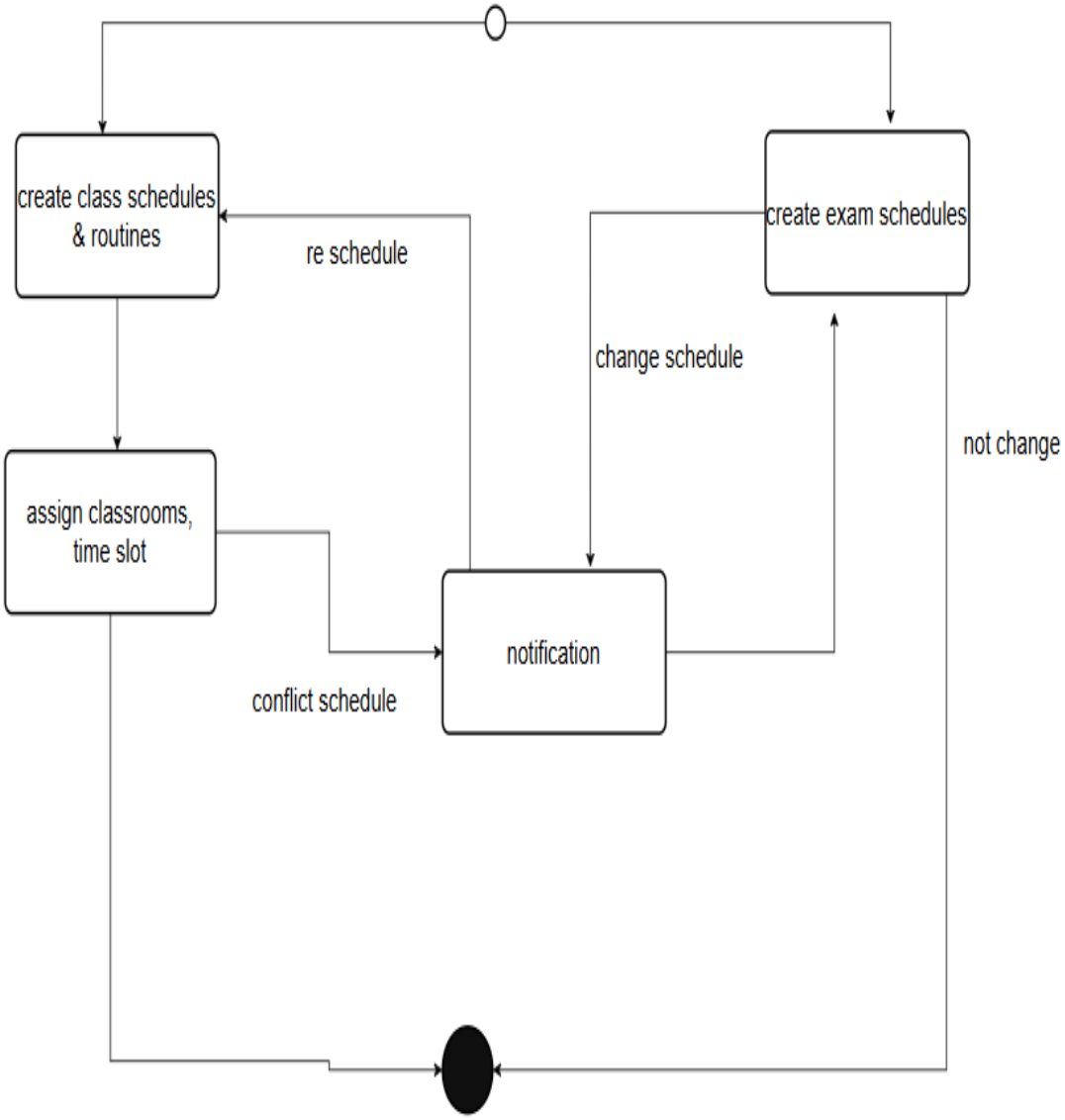


Fig-3: Schedules

ID: 04

Name: Syllabus

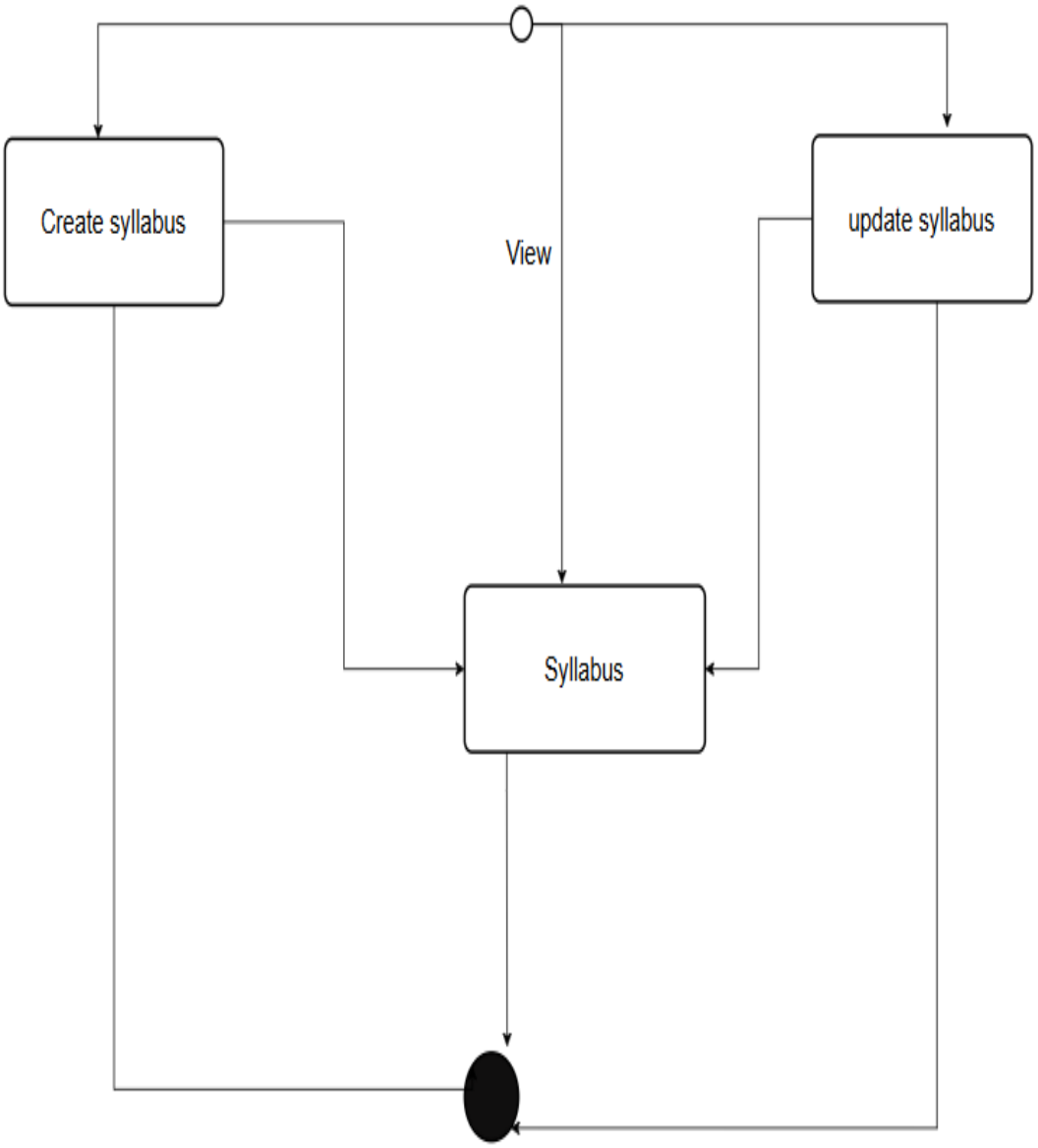


Fig-4: Syllabus

ID: 05

Name: Student Performance

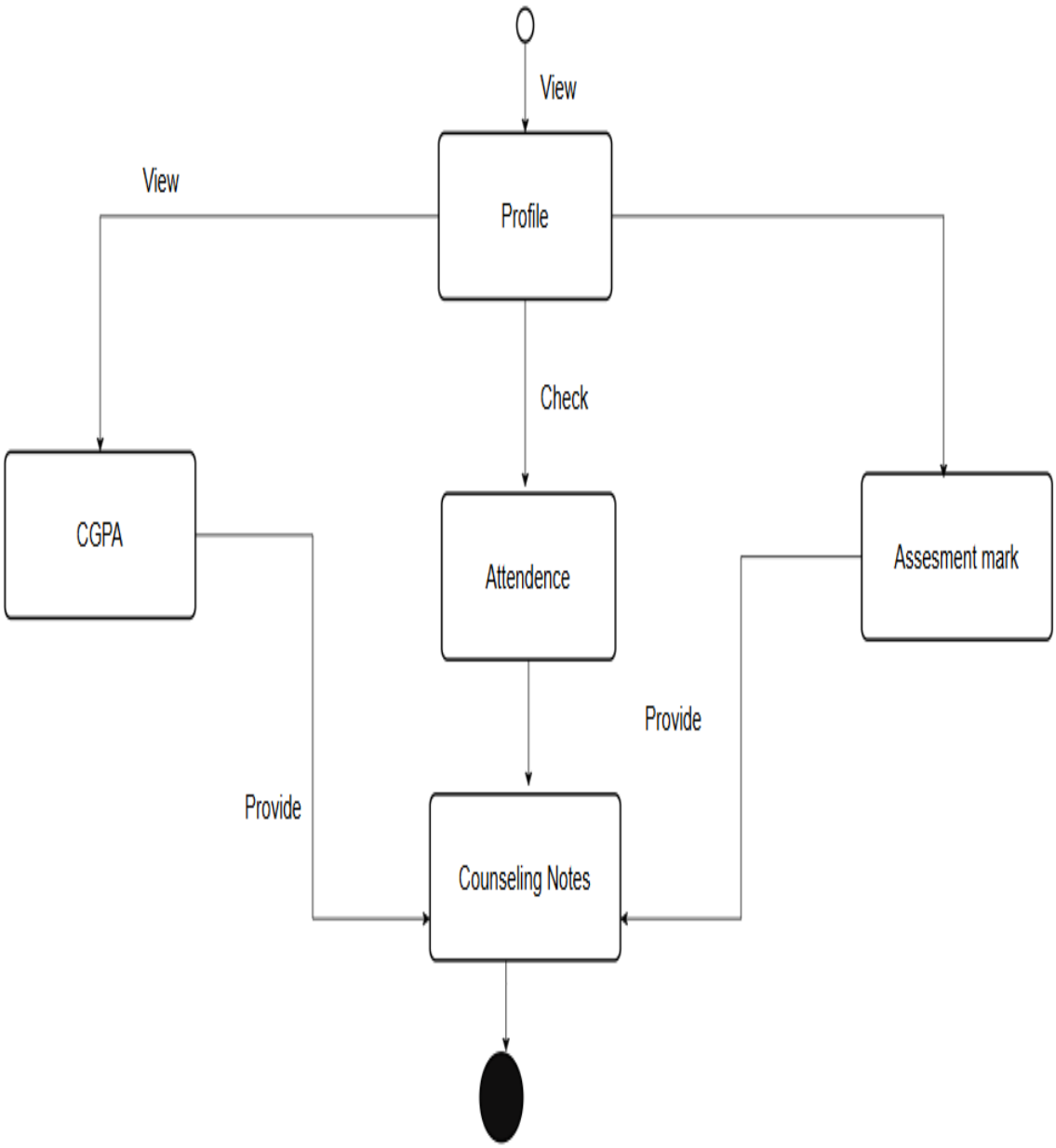


Fig-5: Student Performance

ID: 06

Name: Course Information

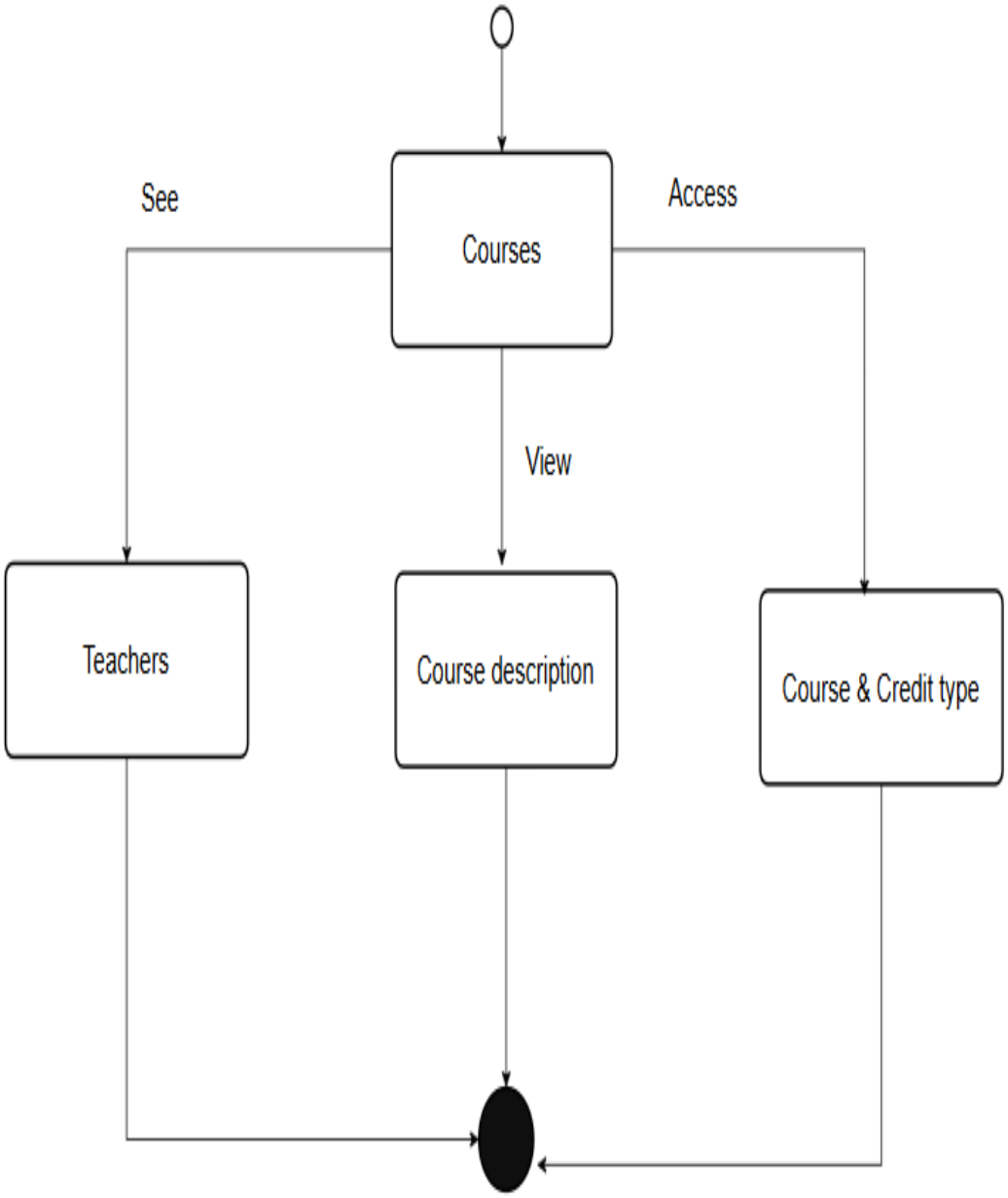


Fig-6: Course Information

ID:07

Name: Assignments

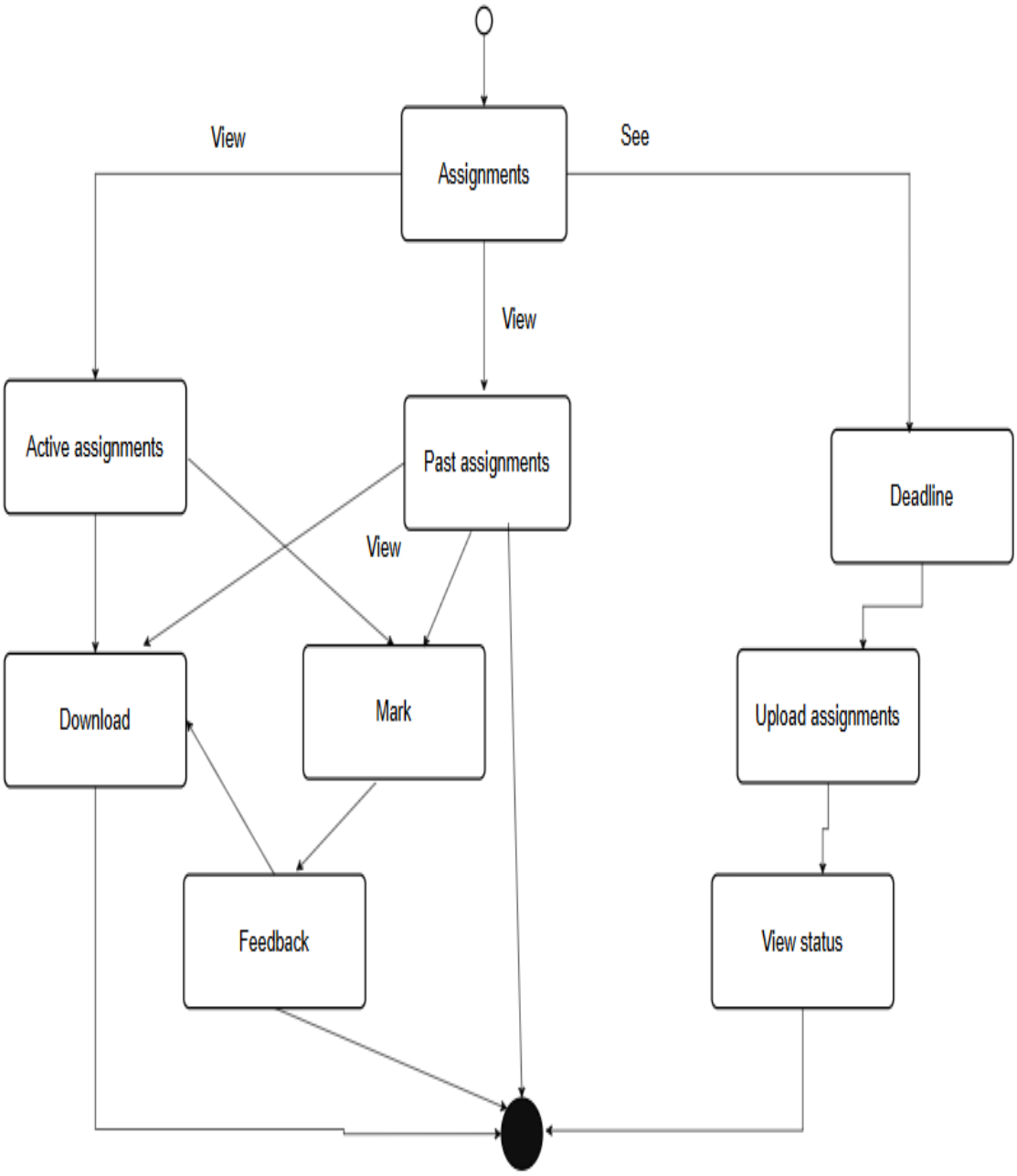


Fig-7: Assignments

ID:08

Name: Marks

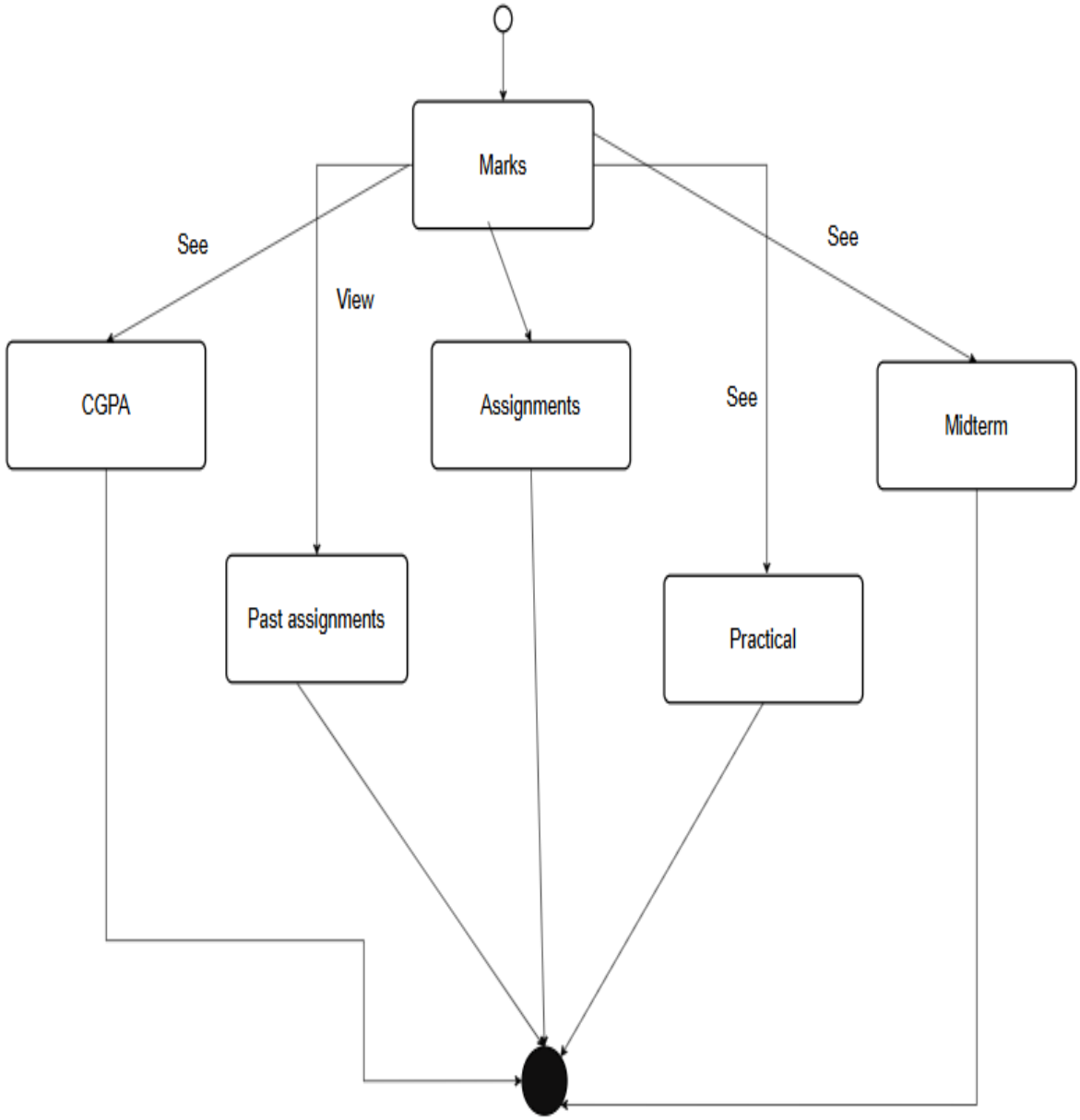


Fig-8: Marks

ID:09

Name: Examination

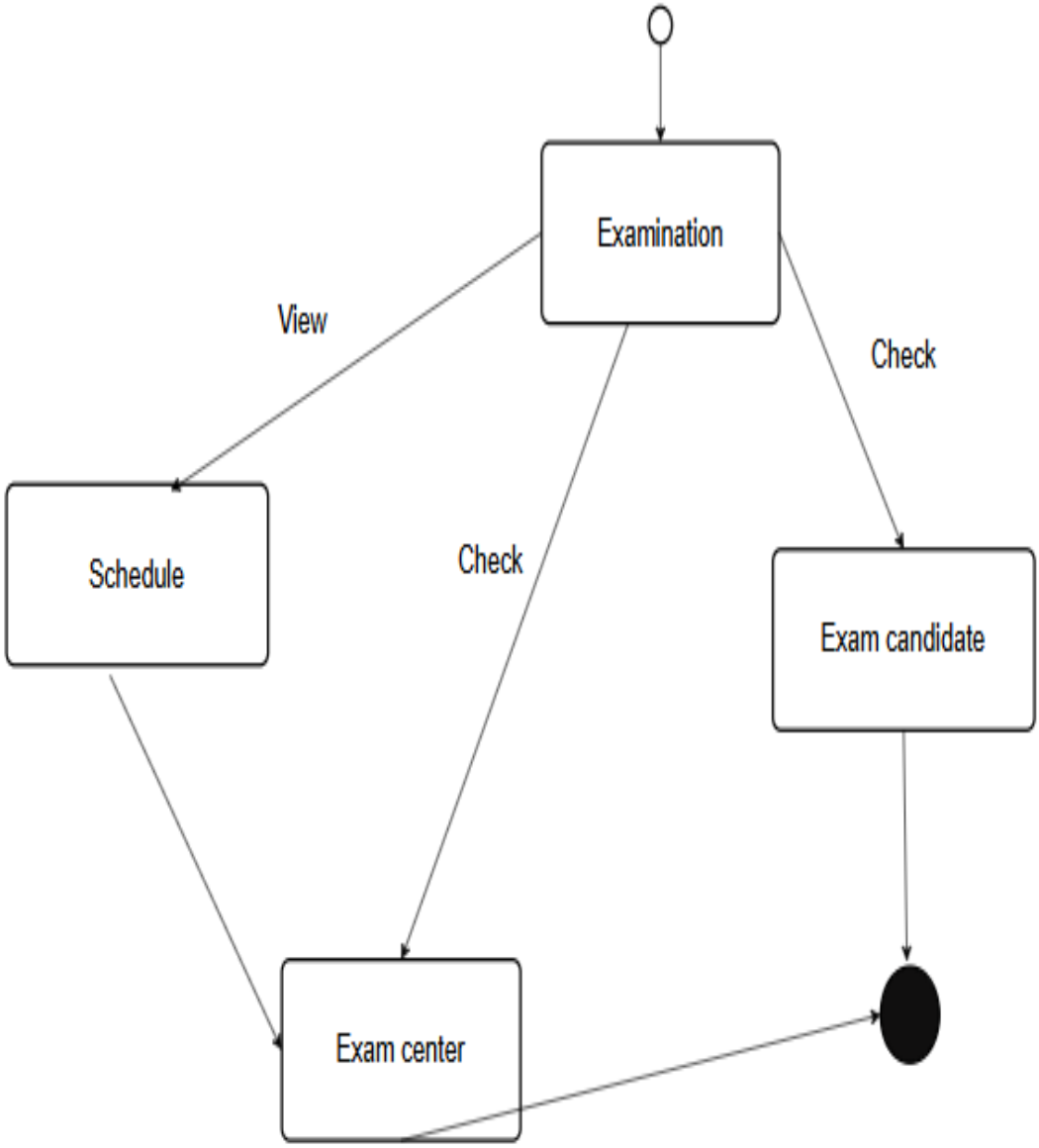


Fig-9: Examination

7.2 Sequence Modeling

7.2.1 System Sequence Diagrams (Interaction Flows)

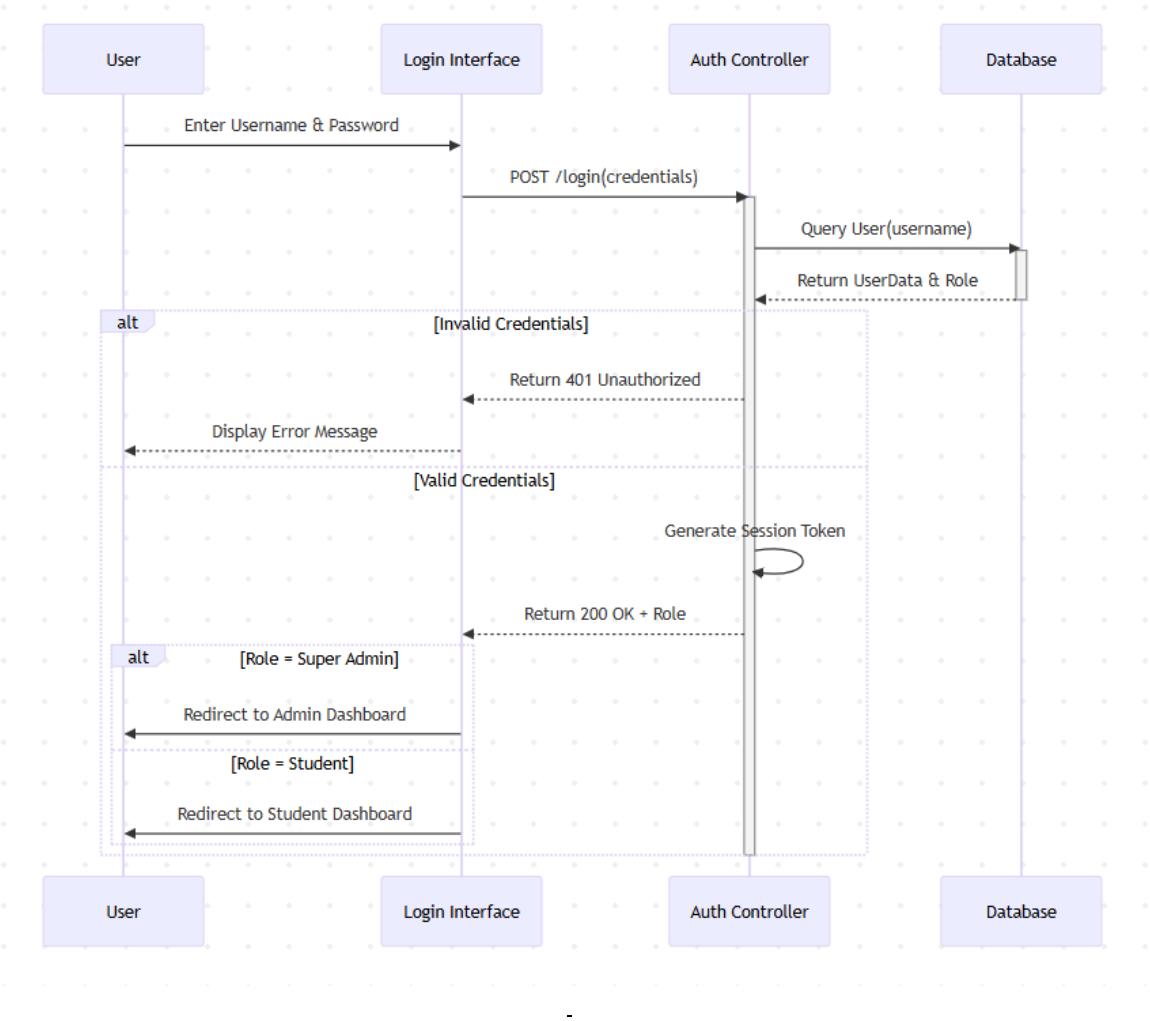


Figure-1:Sequence diagram(Authentication)

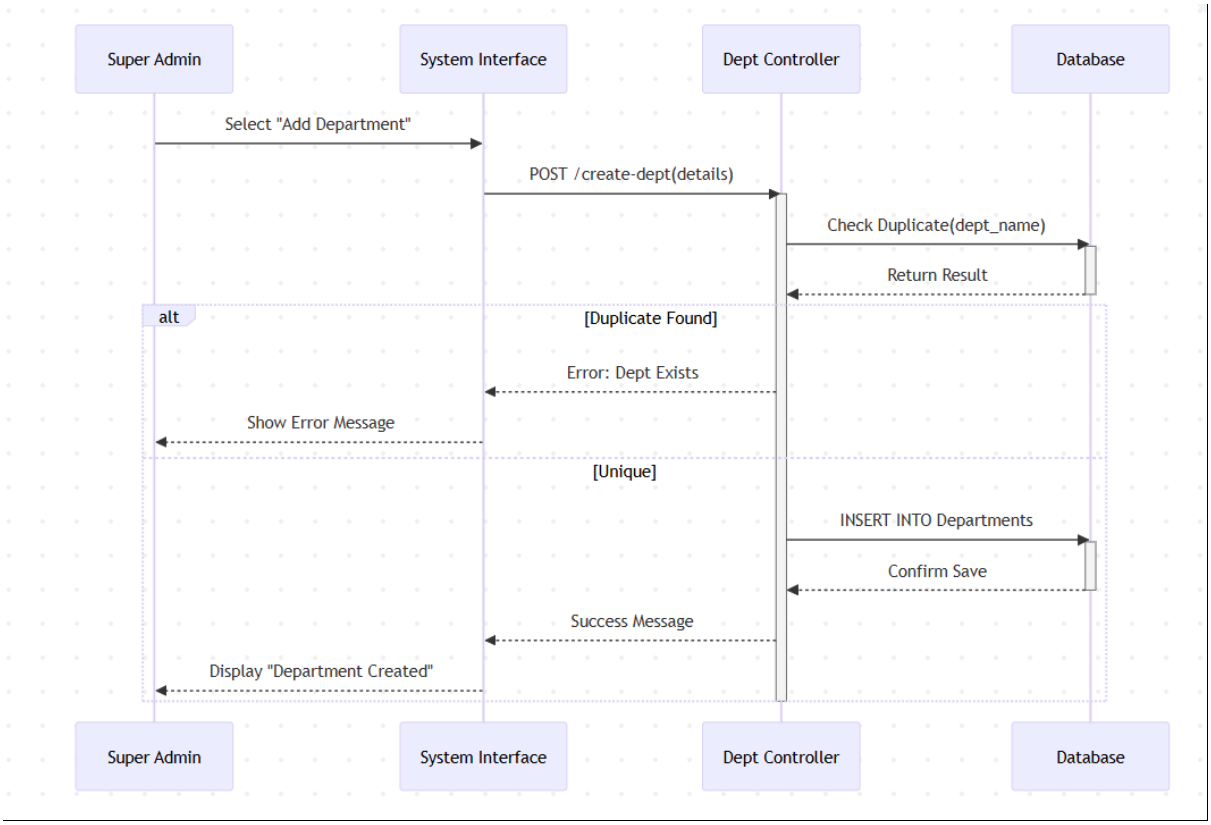


Figure-2: Sequence diagram (Governance Management)

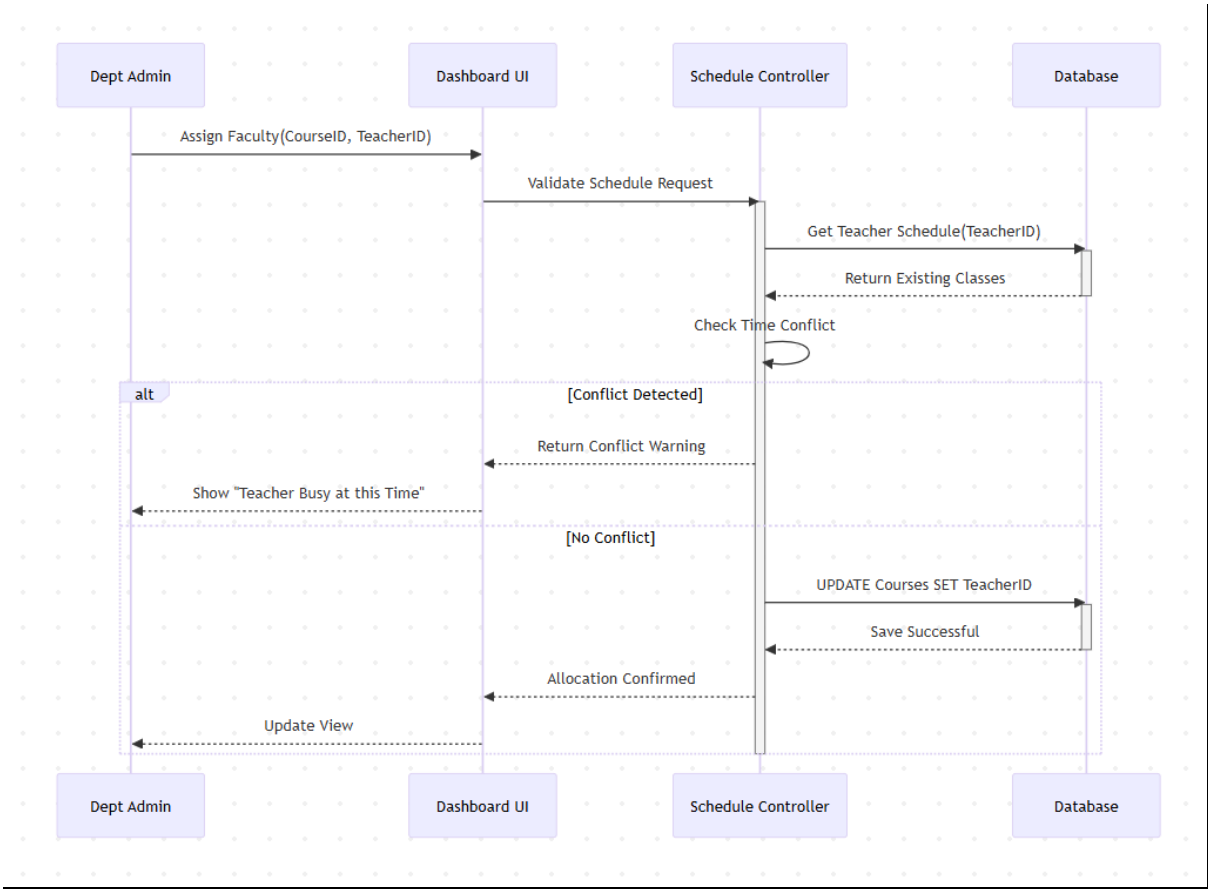


Figure-3: Sequence diagram (Academic Setup)

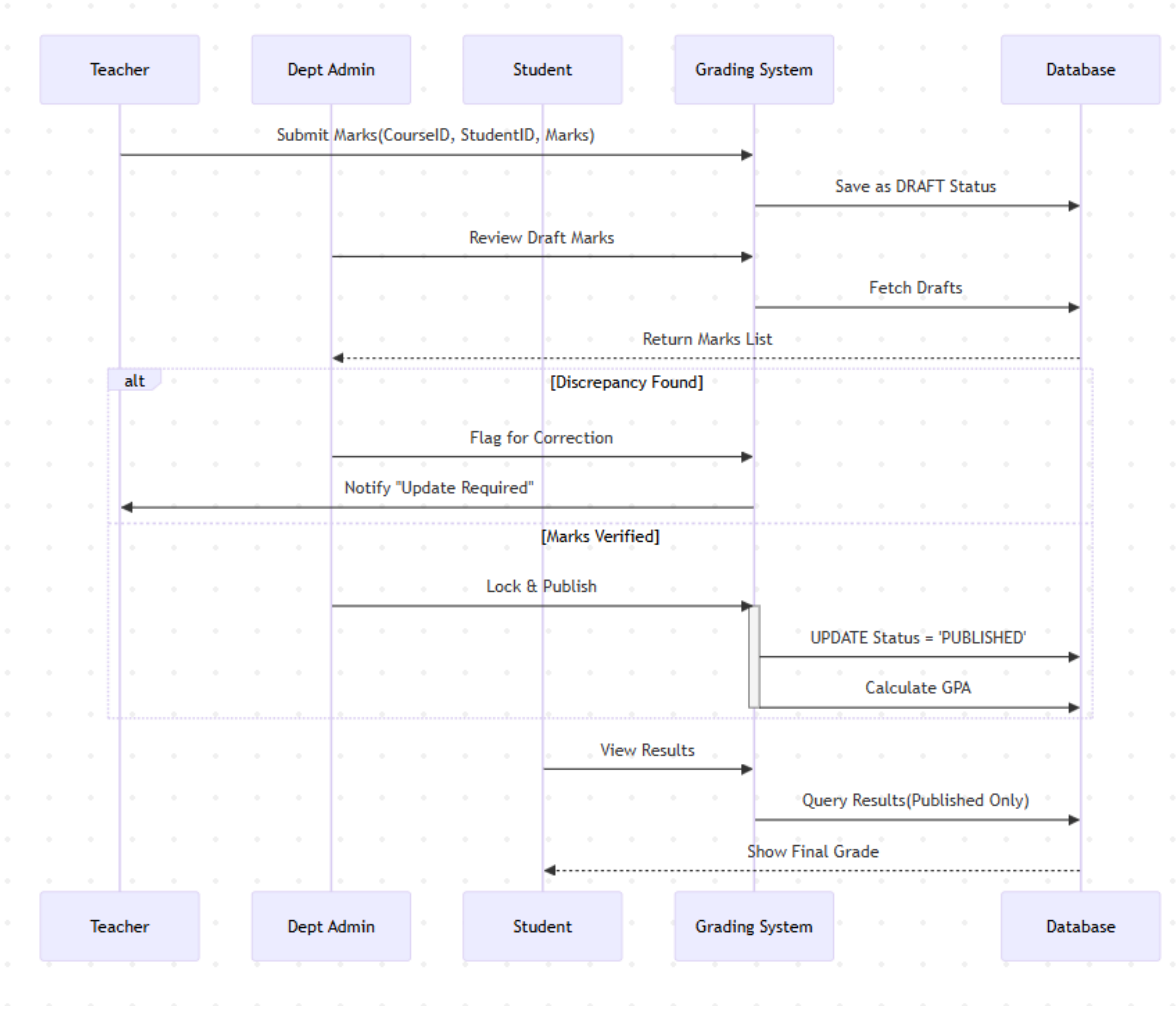


Figure-4: Sequence diagram (Classroom management)

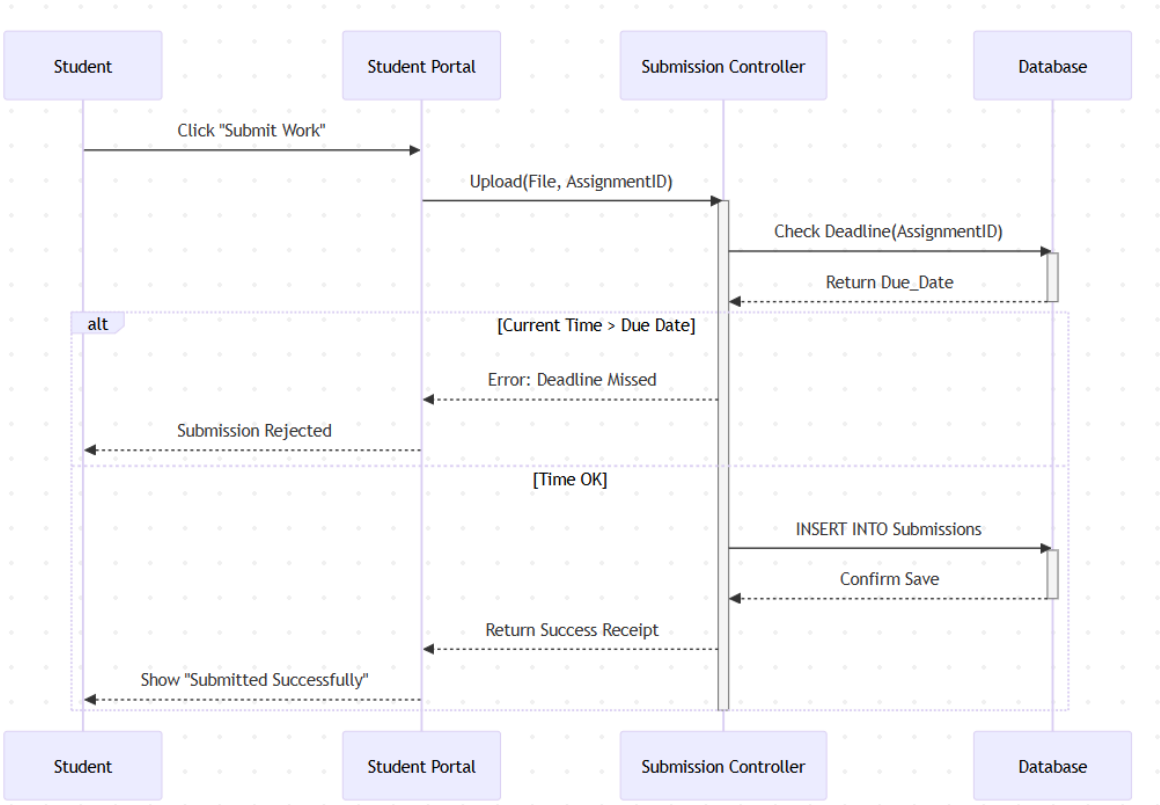


Figure-5: Sequence diagram (Student portal)

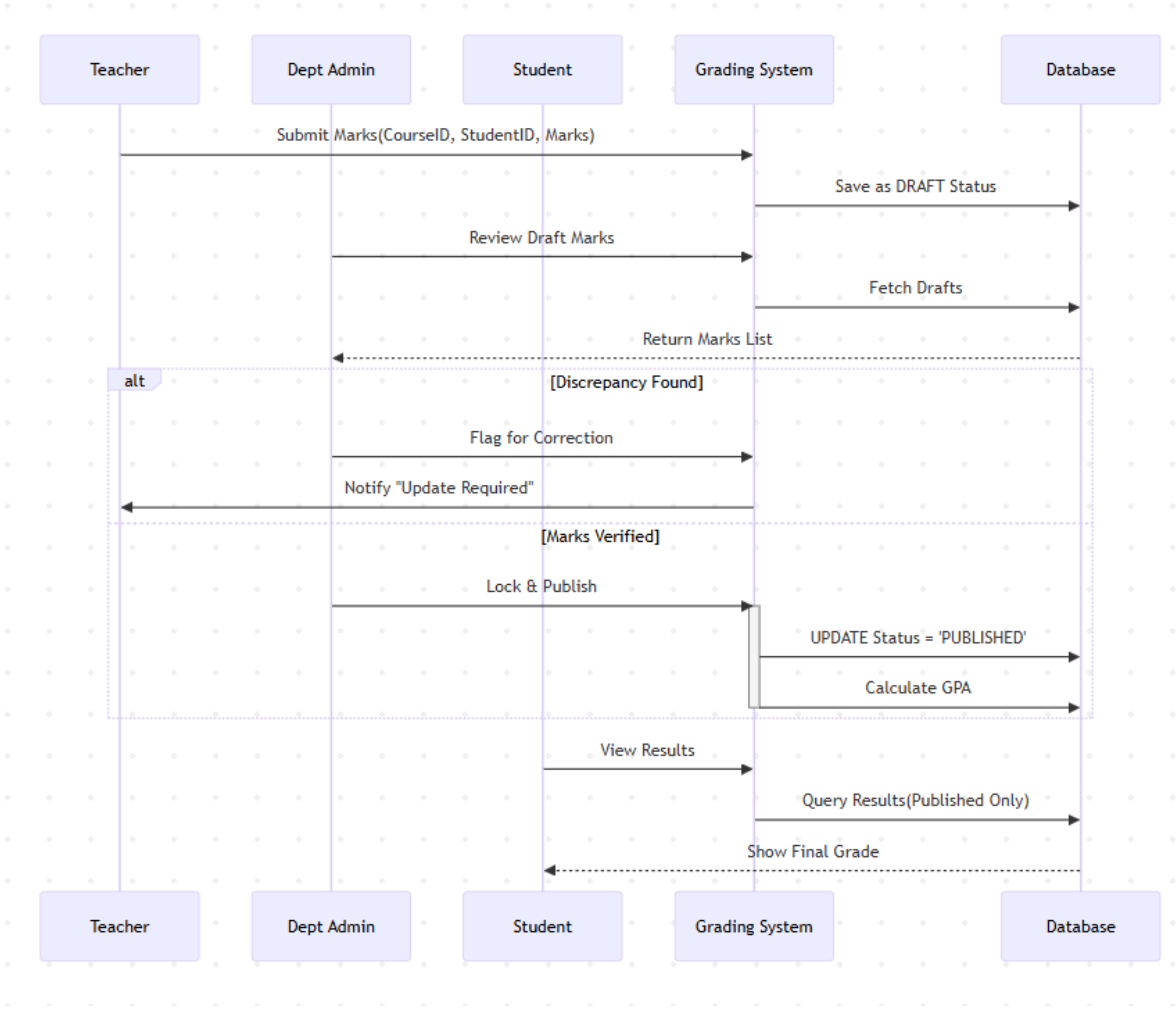


Figure-6: Sequence diagram (Assesments & Results)

Conclusion

The **ACADEMIX** SRS document establishes a solid foundation for a comprehensive University Management System, grounded in the rigorous **Inception** and **Elicitation** phases. By integrating detailed **Activity**, **Data**, and **Behavioral Modeling**, the system successfully translates complex stakeholder needs—such as automated scheduling and result processing—into clear technical requirements. This specification provides a complete roadmap for developing a secure, user-friendly platform that streamlines academic operations and enhances collaboration between students, teachers, and administration.