Submitted by:

Shehriyar khan

Submitted to:

Sir Mukhtiar zamin

Course title:

Software Design and Architecture

Registration no:

SP23-BSE-162

❖ PROJECT TITLE:

Department and Discipline Management

⬥ Module:

Department Management:

Features:

- Add/Edit/Delete/View Departments.
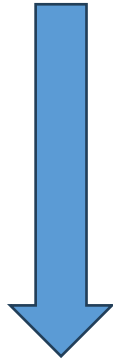- Assign Head of Department.
- View department states.

UML Deliverables:

- Class Diagram for Department.
- Activity Diagram for department creation flow.
- Sequence Diagram for "Assign HoD" process.
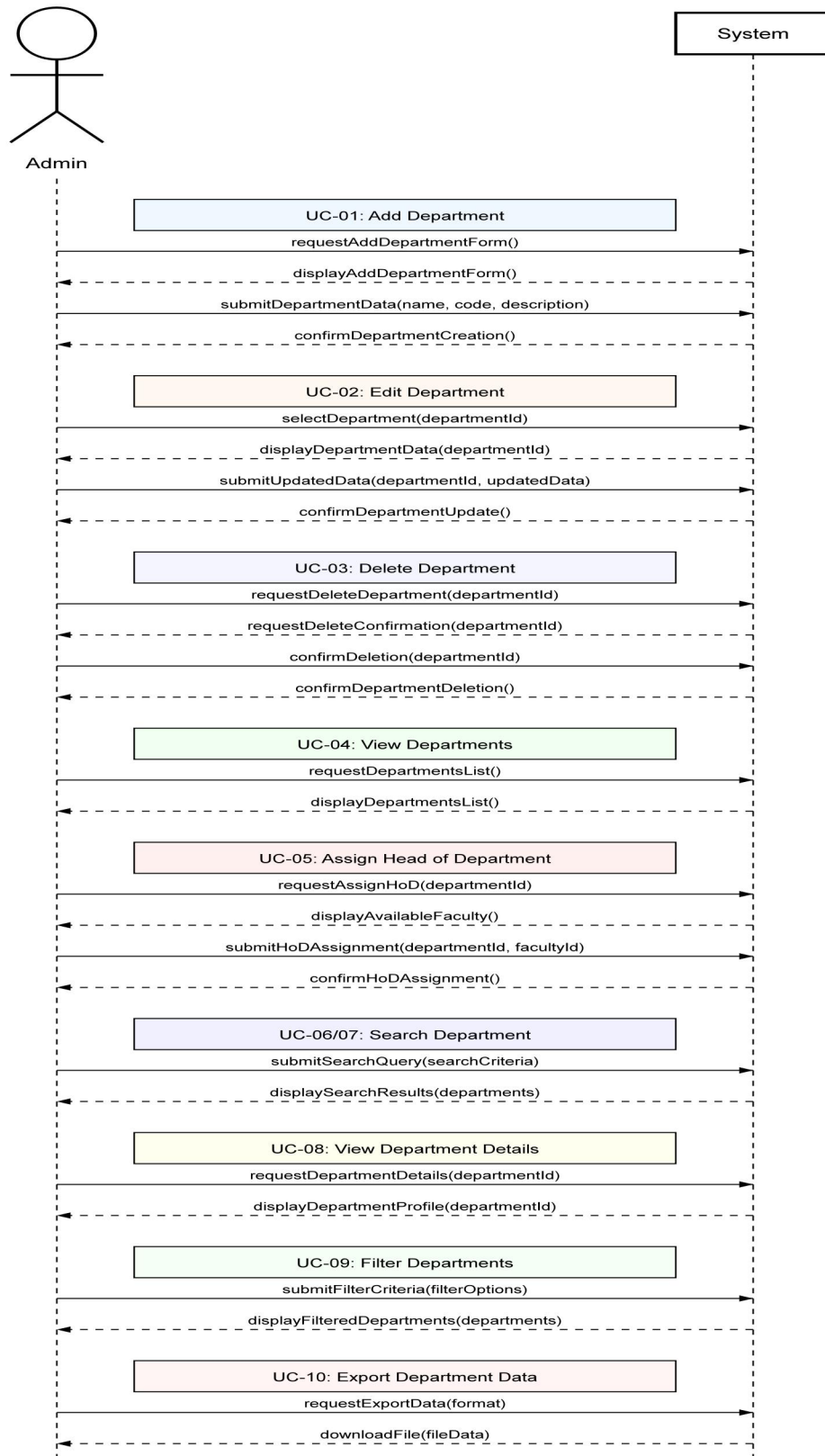
Java Code:

- GUI form to manage departments.

- Logic + file/DB storage.
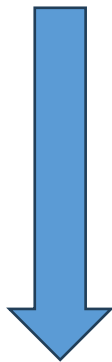
- <u>**SSD Diagram for Department Management Module:**</u>

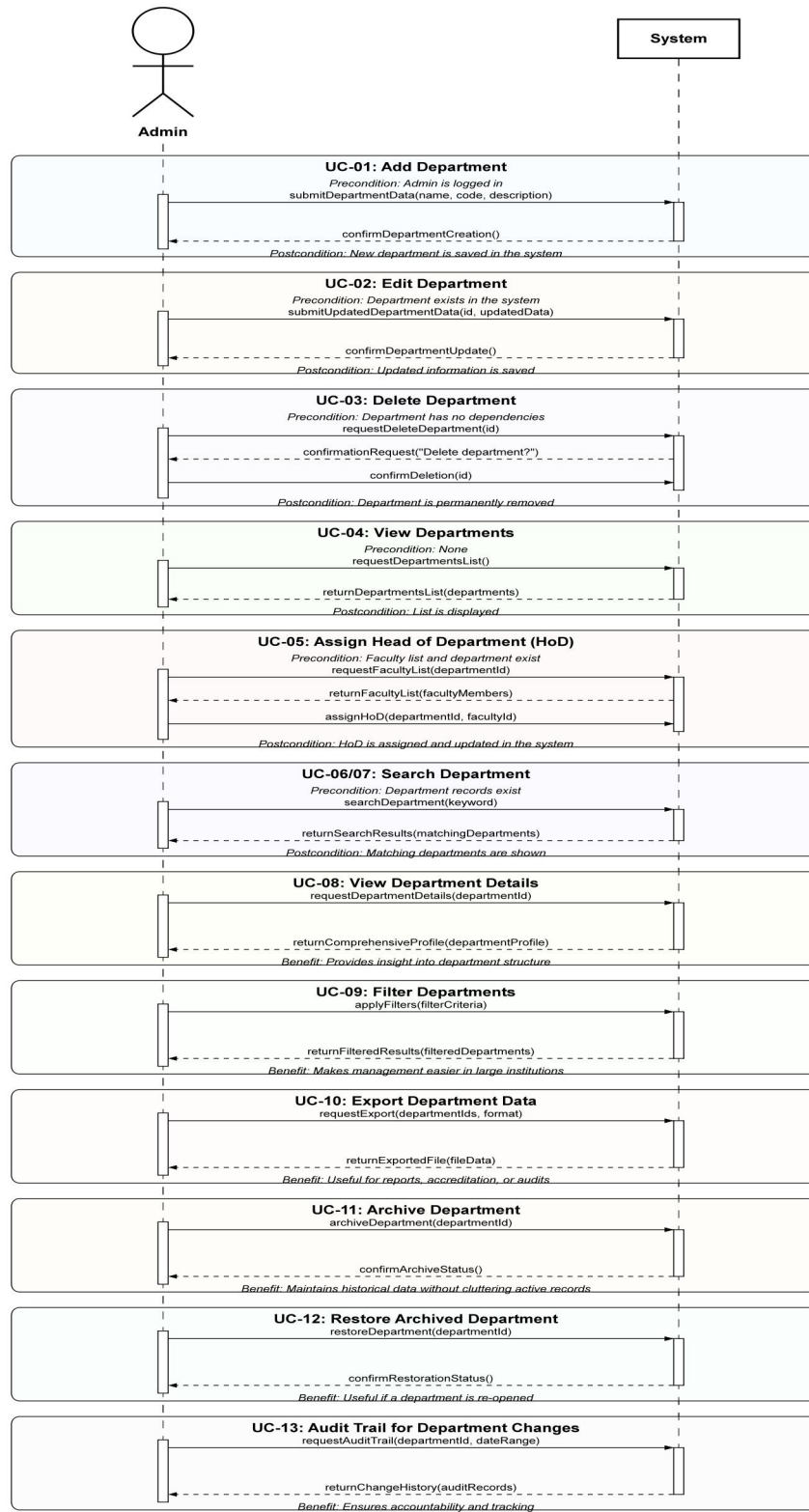**Department Management System - UML System Sequence Diagram**

Admin | System

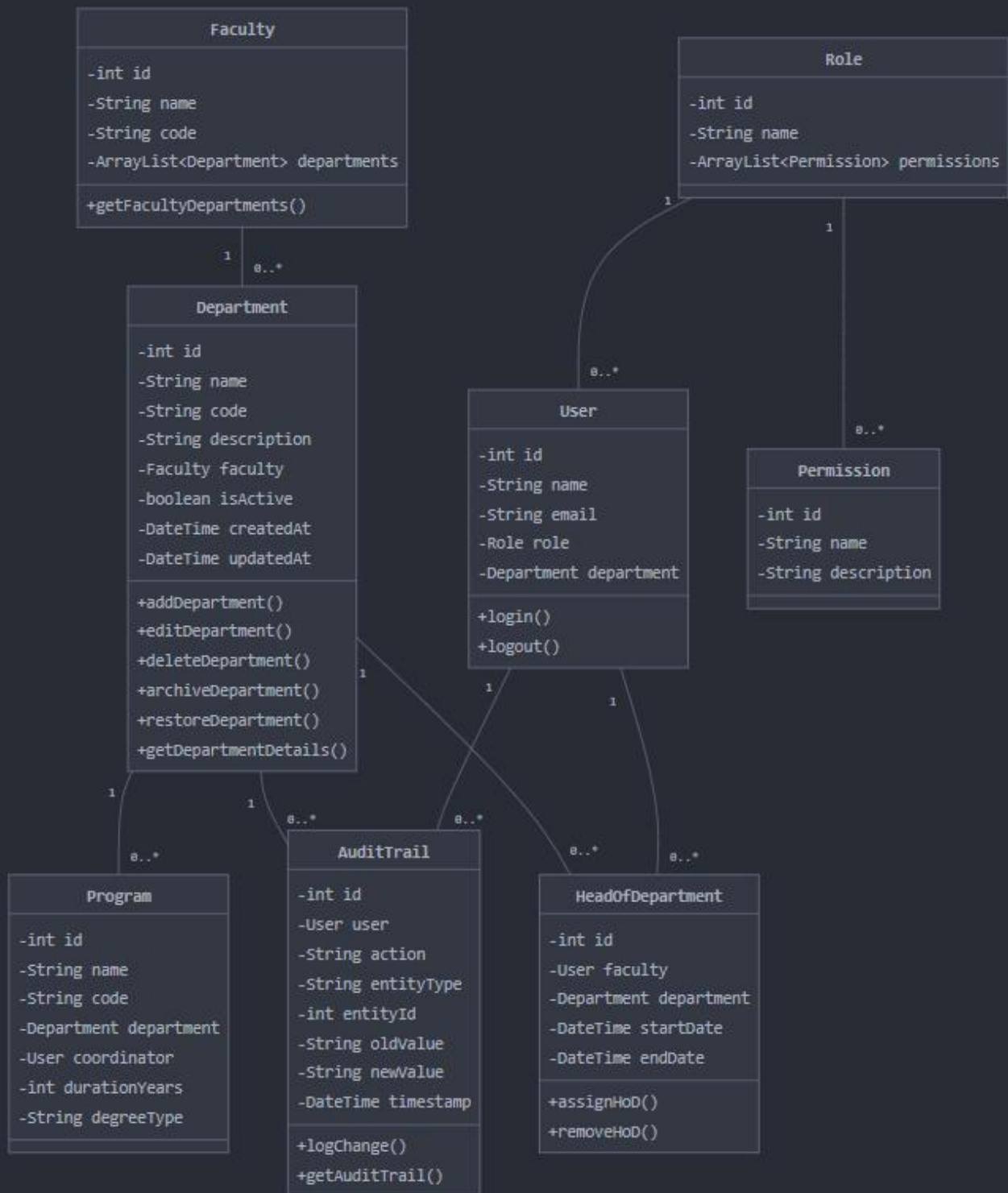**UC-01: Add Department**
- requestAddDepartmentForm()
- displayAddDepartmentForm()
- submitDepartmentData(name, code, description)
- confirmDepartmentCreation()

UC-02: Edit Department
- selectDepartment(departmentId)
- displayDepartmentData(departmentId)
- submitUpdatedData(departmentId, updatedData)
- confirmDepartmentUpdate()

UC-03: Delete Department
- requestDeleteDepartment(departmentId)
- requestDeleteConfirmation(departmentId)
- confirmDeletion(departmentId)
- confirmDepartmentDeletion()

UC-04: View Departments
- requestDepartmentsList()
- displayDepartmentsList()

UC-05: Assign Head of Department
- requestAssignHoD(departmentId)
- displayAvailableFaculty()
- submitHoDAssignment(departmentId, facultyId)
- confirmHoDAssignment()

UC-06/07: Search Department
- submitSearchQuery(searchCriteria)
- displaySearchResults(departments)

UC-08: View Department Details
- requestDepartmentDetails(departmentId)
- displayDepartmentProfile(departmentId)

UC-09: Filter Departments
- submitFilterCriteria(filterOptions)
- displayFilteredDepartments(departments)

UC-10: Export Department Data
- requestExportData(format)
- downloadFile(fileData)

x\xm\mxmskaxascjahdcjahscjhsajhcajsc

hashchHere is the Complete one:

Department Management System - UML System Sequence Diagram

Admin

System

UC-01: Add Department
Precondition: Admin is logged in
submitDepartmentData(name, code, description)
confirmDepartmentCreation()
Postcondition: New department is saved in the system

UC-02: Edit Department
Precondition: Department exists in the system
submitUpdatedDepartmentData(id, updatedData)
confirmDepartmentUpdate()
Postcondition: Updated information is saved

UC-03: Delete Department
Precondition: Department has no dependencies
requestDeleteDepartment(id)
confirmationRequest("Delete department?")
confirmDeletion(id)
Postcondition: Department is permanently removed

UC-04: View Departments
Precondition: None
requestDepartmentsList()
returnDepartmentsList(departments)
Postcondition: List is displayed

UC-05: Assign Head of Department (HoD)
Precondition: Faculty list and department exist
requestFacultyList(departmentId)
returnFacultyList(facultyMembers)
assignHoD(departmentId, facultyId)
Postcondition: HoD is assigned and updated in the system

UC-06/07: Search Department
Precondition: Department records exist
searchDepartment(keyword)
returnSearchResults(matchingDepartments)
Postcondition: Matching departments are shown

UC-08: View Department Details
requestDepartmentDetails(departmentId)
returnComprehensiveProfile(departmentProfile)
Benefit: Provides insight into department structure

UC-09: Filter Departments
applyFilters(filterCriteria)
returnFilteredResults(filteredDepartments)
Benefit: Makes management easier in large institutions

UC-10: Export Department Data
requestExport(departmentIds, format)
returnExportedFile(fileData)
Benefit: Useful for reports, accreditation, or audits

UC-11: Archive Department
archiveDepartment(departmentId)
confirmArchiveStatus()
Benefit: Maintains historical data without cluttering active records

UC-12: Restore Archived Department
restoreDepartment(departmentId)
confirmRestorationStatus()
Benefit: Useful if a department is re-opened

UC-13: Audit Trail for Department Changes
requestAuditTrail(departmentId, dateRange)
returnChangeHistory(auditRecords)
Benefit: Ensures accountability and tracking

Class Diagram:

Faculty

-int id
-String name
-String code
-ArrayList<Department> departments

+getFacultyDepartments()

Role

-int id
-String name
-ArrayList<Permission> permissions

Department

-int id
-String name
-String code
-String description
-Faculty faculty
-boolean isActive
-DateTime createdAt
-DateTime updatedAt

+addDepartment()
+editDepartment()
+deleteDepartment()
+archiveDepartment()
+restoreDepartment()
+getDepartmentDetails()

User

-int id
-String name
-String email
-Role role
-Department department

+login()
+logout()

Permission

-int id
-String name
-String description

Program

-int id
-String name
-String code
-Department department
-User coordinator
-int durationYears
-String degreeType

AuditTrail

-int id
-User user
-String action
-String entityType
-int entityId
-String oldValue
-String newValue
-DateTime timestamp

+logChange()
+getAuditTrail()

HeadOfDepartment

-int id
-User faculty
-Department department
-DateTime startDate
-DateTime endDate

+assignHoD()
+removeHoD()

Fully Dressed Use Cases:

Use Case 1: Edit Department:

Field	Description
Use Case ID	UC-01
Use Case Name	Add Department
Primary Actor	Admin
Description	Admin adds a new department by entering the department name, code, and description.
Stakeholders and Interests	Admin – wants to successfully add departments with proper validation.
Preconditions	Admin is logged in and has the necessary permissions.
Postconditions	The department is successfully created and saved in the system database.
Exceptions:	If a department with the same name or code already exists, an error message is shown indicating a duplicate department name/code.
Basic Flow:	• Admin logs into the system and navigates to the "Add Department" section. • Admin enters the department name in the provided field. • Admin enters the department code. • Admin provides a description for the department. • Admin clicks the "Save" button. • The system checks for duplicates

	(department name or code). • If no duplicate is found, the system saves the department to the database. • A confirmation message is displayed to the Admin, indicating that the department was successfully added.
Alternative Flow:	If a duplicate department name/code is detected, the system prevents saving and displays an error message with a prompt to re-enter valid values.
Main Success Scenario	1. Admin logs into the system. 2. Navigates to the "Add Department" section. 3. Enters department name, code, and description. 4. Clicks "Save". 5. System validates and checks for duplicates. 6. If valid, department is saved. 7. Confirmation message is shown.
Extensions	3a. Required field is missing – Prompt admin to fill all fields. 5a. Duplicate name or code – Show error and ask for correction.
Special Requirements	Unique department code validation. Real-time field validation.
Frequency of Use	Occasionally when a new department is established.
Assumptions	Admin has accurate information and requires permissions.
Notes and Issues	Consider auto-generating codes to avoid duplication.

Use Case 2: Edit Department:

Field	Description
Use Case ID	UC-02
Use Case Name	Edit Department
Primary Actor	Admin
Stakeholders and Interests	Admin – wants to maintain accurate department data.
Description	The admin updates the details of an existing department (such as department name, code, and description).
Preconditions	The department exists and Admin is logged in with proper access.
Postconditions	Changes to the department are successfully saved in the system.
Exceptions:	Admin is restricted from editing certain fields like the department ID.
Basic Flow	• Admin logs into the system and navigates to the "Departments" section. • Admin selects the department they wish to edit from the list of existing departments. • Admin updates the department name, code, and/or description as needed. • Admin clicks the "Save" button to apply changes. • The system validates the changes, ensuring the new department code does not already exist. • The updated department details are saved in the system.

	• A confirmation message is displayed to the Admin, indicating the successful update.
Alternative Flow	If the Admin attempts to change a restricted field (such as the department ID), the system prevents saving and displays an error message indicating that the ID field cannot be changed.
Main Success Scenario	1. Admin logs into the system. 2. Navigates to the "Departments" section. 3. Selects a department to edit. 4. Makes necessary changes to name, code, or description. 5. Clicks "Save". 6. System validates changes. 7. Updates are saved. 8. Confirmation message is shown.
Extensions	4a. Attempt to edit department ID – Field is disabled. 6a. Duplicate code entered – Error message shown.
Special Requirements	ID field is non-editable. Unique constraint on department code.
Frequency of Use	Occasionally, based on administrative decisions.
Assumptions	Admin ensures data accuracy and consistency.
Notes and Issues	Include audit trail for edits in future iterations.

Use Case: Delete Department:

Field	Description
Use Case ID	UC-03
Use Case Name	Delete Department
Primary Actor	Admin
Description	Admin deletes a department if it's not associated with any active programs or users.
Goal in Context	To remove a department that is no longer in use and has no dependencies (e.g., linked users or programs).
Scope	Department Management System
Level	User goal
Stakeholders and Interests	Admin – needs to clean up inactive or deprecated departments. Institution – avoids data clutter and maintains system hygiene.
Preconditions	Admin is logged in. Department has no active users or programs associated.
Postconditions	Department is permanently deleted and no longer visible in lists.
Main Success Scenario	1. Admin logs into the system. 2. Navigates to department list. 3. Selects a department. 4. Clicks "Delete". 5. System checks for dependencies. 6. If none, department is deleted. 7. Confirmation is shown.
Alternative Flows	5a. Dependencies found: - System blocks deletion and displays dependency info. - Admin is prompted to remove dependencies first.

Exceptions	Database error, session timeout, or unexpected failure during dependency check.
Special Requirements	Confirmation prompt, deletion logs, secure access to delete functionality.
Frequency of Use	Occasional
Assumptions	Admin verifies department is truly no longer needed.
Open Issues	Should deletion be soft (archived) instead of permanent?

UC-04: View Departments

- **Use Case ID:** UC-04

- **Use Case Name:** View Departments

- **Actor(s):** Admin

- **Description:** Admin can view a list of all departments along with basic details.

- **Assumptions:**

  - The Admin has the necessary permissions to view the department data.

- **Preconditions:**

  - Admin is logged in to the system.

  - Admin has access to department data stored in the system.

- **Postconditions:**

  - A list of departments is displayed, including department name, department code, and HoD.

- **Main Flow (Basic Path):**

  1. Admin navigates to the "View Departments" section.

2. The system retrieves and displays a list of all departments with basic details such as department name, department code, and HoD.

- **Alternate Flow (if any):**

  o If there are no departments, the system displays a message: "No departments available."

- **Exceptions:**

  o If there is a database connection failure, the system displays an error message: "Unable to retrieve department data."

- **Frequency of Use:**

  o Typically used when the Admin wants to review the list of departments in the institution.

UC-05: Assign Head of Department (HoD)

- **Use Case ID:** UC-05

- **Use Case Name:** Assign Head of Department (HoD)

- **Actor(s):** Admin

- **Description:** Admin assigns or changes the HoD for a department.

- **Assumptions:**

  o The Admin has access to faculty records and department details.

- **Preconditions:**

  o Admin is logged in and authorized to make changes.

  o Faculty list and department records are available in the system.

- **Postconditions:**

  o The HoD is assigned to the selected department and updated in the system.

- **Main Flow (Basic Path):**

1. Admin selects the department to assign an HoD.
2. Admin chooses a faculty member from the available list.

3. The system validates if the faculty member is eligible to be assigned as HoD.
4. The system assigns the selected faculty member as the HoD for the department.
5. The system updates the department record with the new HoD information.
6. A success message is displayed confirming the HoD assignment.

- **Alternate Flow (if any):**

    o If the faculty member is already assigned as HoD to another department, the system displays an error message: "Faculty member already assigned as HoD in another department."

- **Exceptions:**

    o If no faculty members are available for assignment, the system displays a message: "No available faculty members."

- **Frequency of Use:**

    o Typically used when a new HoD needs to be assigned or when an existing HoD is replaced.

UC-06: Search Department

- **Use Case ID:** UC-06

- **Use Case Name:** Search Department

- **Actor(s):** Admin

- **Description:** Admin searches for departments by name or code.

- **Assumptions:**

    o Department data is available and up-to-date in the system.

- **Preconditions:**

    o Admin is logged in and has access to the department database.

    o Department records exist in the system.

- **Postconditions:**

    o Departments matching the search criteria (name or code) are displayed.

- **Main Flow (Basic Path):**

1. Admin enters the department name or department code in the search field.
2. The system searches for matching departments.
3. The system displays a list of matching departments.

- **Alternate Flow (if any):**

    o If no matching departments are found, the system displays: "No departments found."

- **Exceptions:**

    o If a database error occurs, the system displays: "Unable to search for departments."

- **Frequency of Use:**

    o This use case is used regularly to find a specific department based on its name or code.

UC-07: Advanced Department Search

- **Use Case ID:** UC-07

- **Use Case Name:** Advanced Department Search

- **Actor(s):** Admin

- **Description:** Admin can search departments using multiple criteria such as department name, department code, or faculty.

- **Assumptions:**

    o Department records are structured to include faculty and other attributes.

- **Preconditions:**

    o Admin is logged in and has access to the department and faculty data.

- **Postconditions:**

    o The system displays departments that match the search criteria (e.g., name, code, faculty).

- **Main Flow (Basic Path):**

  1. Admin enters search criteria such as department name, code, or faculty.
  2. The system processes the criteria and searches the department database.
  3. The system returns matching departments and displays them to the Admin.

- **Alternate Flow (if any):**

  - If no matching departments are found, the system displays: "No results found."

- **Exceptions:**

  - If an error occurs while processing the search, the system displays: "Search failed. Please try again later."

- **Frequency of Use:**

  - Typically used when Admin needs to search for specific departments based on detailed criteria.

UC-08: View Department Details

- **Use Case ID:** UC-08

- **Use Case Name:** View Department Details

- **Actor(s):** Admin

- **Description:** Admin can view the detailed profile of a department.

- **Assumptions:**

  - The department's full profile, including the faculty list and HoD, is available in the system.

- **Preconditions:**

  - Admin is logged in.

  - Department records are available in the system.

- **Postconditions:**

o The department's detailed information is displayed, including HoD, faculty members, and related disciplines.

- **Main Flow (Basic Path):**

1. Admin selects a department from the department list.
2. The system retrieves and displays the department's detailed profile.

- **Alternate Flow (if any):**

  o If the department's details cannot be retrieved, the system displays an error message: "Unable to retrieve department details."

- **Exceptions:**

  o If the system encounters a database failure, an error message is shown.

- **Frequency of Use:**

  o Used when Admin needs a full overview of a specific department.

UC-09: Filter Departments

- **Use Case ID:** UC-09

- **Use Case Name:** Filter Departments

- **Actor(s):** Admin

- **Description:** Admin can filter departments based on various attributes such as faculty, HoD assigned status, and department status (active/inactive).

- **Assumptions:**

  o Department and faculty records are maintained and categorized by status and other attributes.

- **Preconditions:**

  o Admin is logged in and authorized to filter department records.

  o Department data is available in the system.

- **Postconditions:**

  o Departments that meet the selected criteria are displayed.

- **Main Flow (Basic Path):**

1. Admin selects the filter criteria (faculty, HoD assigned, active/inactive).
2. The system filters the departments based on the criteria.
3. The system displays the filtered list of departments.

- **Alternate Flow (if any):**

  - If no departments meet the filter criteria, the system displays: "No results found."

- **Exceptions:**

  - If filtering fails, the system displays: "Unable to filter departments."

- **Frequency of Use:**

  - Used when Admin needs to quickly find departments based on specific criteria.

UC-10: Export Department Data

- **Use Case ID:** UC-10

- **Use Case Name:** Export Department Data

- **Actor(s):** Admin

- **Description:** Admin can export department details into formats such as PDF, Excel, or CSV.

- **Assumptions:**

  - The system supports export functionality in multiple formats.

- **Preconditions:**

  - Admin is logged in and has permission to export data.

  - Department data is available in the system.

- **Postconditions:**

  - The department data is successfully exported to the selected file format.

- **Main Flow (Basic Path):**

1. Admin selects the "Export" option.

2. Admin chooses the file format (PDF, Excel, or CSV).

3. The system generates the selected file and allows the Admin to download it.

- **Alternate Flow (if any):**

   o If the export fails, the system displays: "Export failed. Please try again."

- **Exceptions:**

   o If there is a system failure during the export process, an error message is displayed.

- **Frequency of Use:**

   o This use case is typically used when reports or records need to be shared externally or archived.

UC-11: Archive Department

- **Use Case ID:** UC-11

- **Use Case Name:** Archive Department

- **Actor(s):** Admin

- **Description:** Admin can mark a department as inactive (archived) rather than deleting it permanently.

- **Assumptions:**

   o Admin is authorized to archive departments.

   o Archived departments are stored for future reference.

- **Preconditions:**

   o Admin is logged in.

   o The department exists in the system.

- **Postconditions:**

- o The department is marked as archived, making it inactive in the system.

- **Main Flow (Basic Path):**

1. Admin selects the department to archive.
2. Admin clicks "Archive" to mark the department as inactive.
3. The system updates the department's status to archived.
4. A success message is displayed confirming the department has been archived.

- **Alternate Flow (if any):**

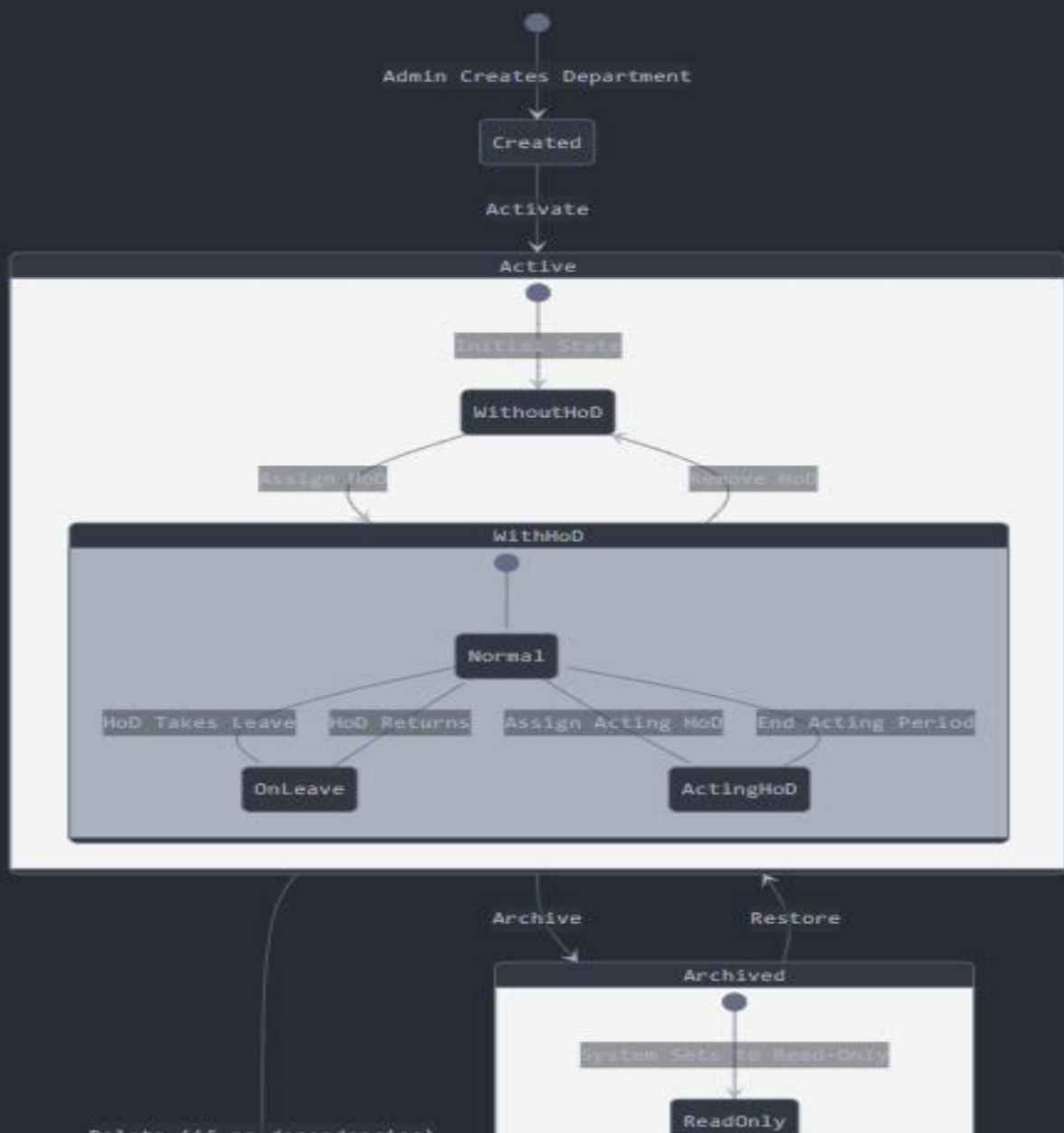 - o If the department cannot be archived, an error message is displayed: "Unable to archive department."

- **Exceptions:**

  - o If the department does not exist in the system, an error message is displayed.

- **Frequency of Use:**

  - o Typically used when a department is no longer operational but needs to be preserved for historical records.
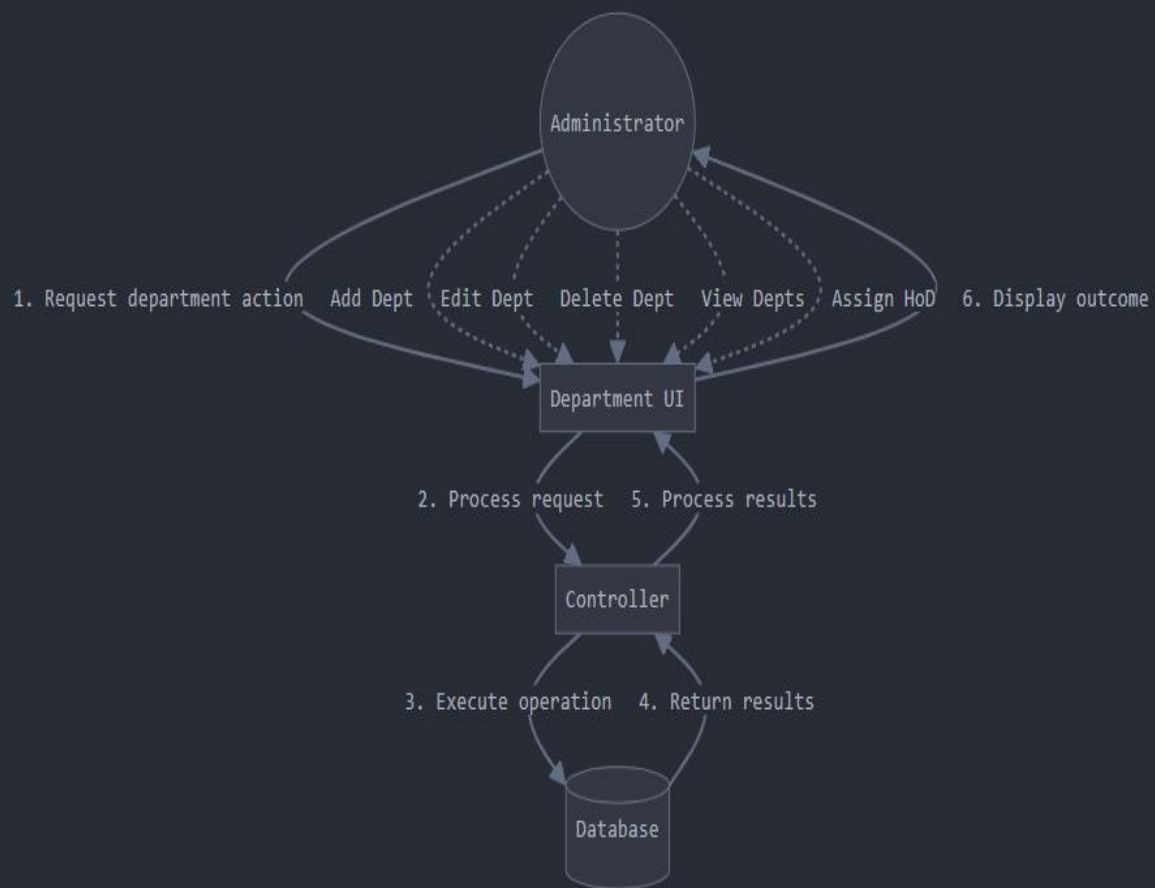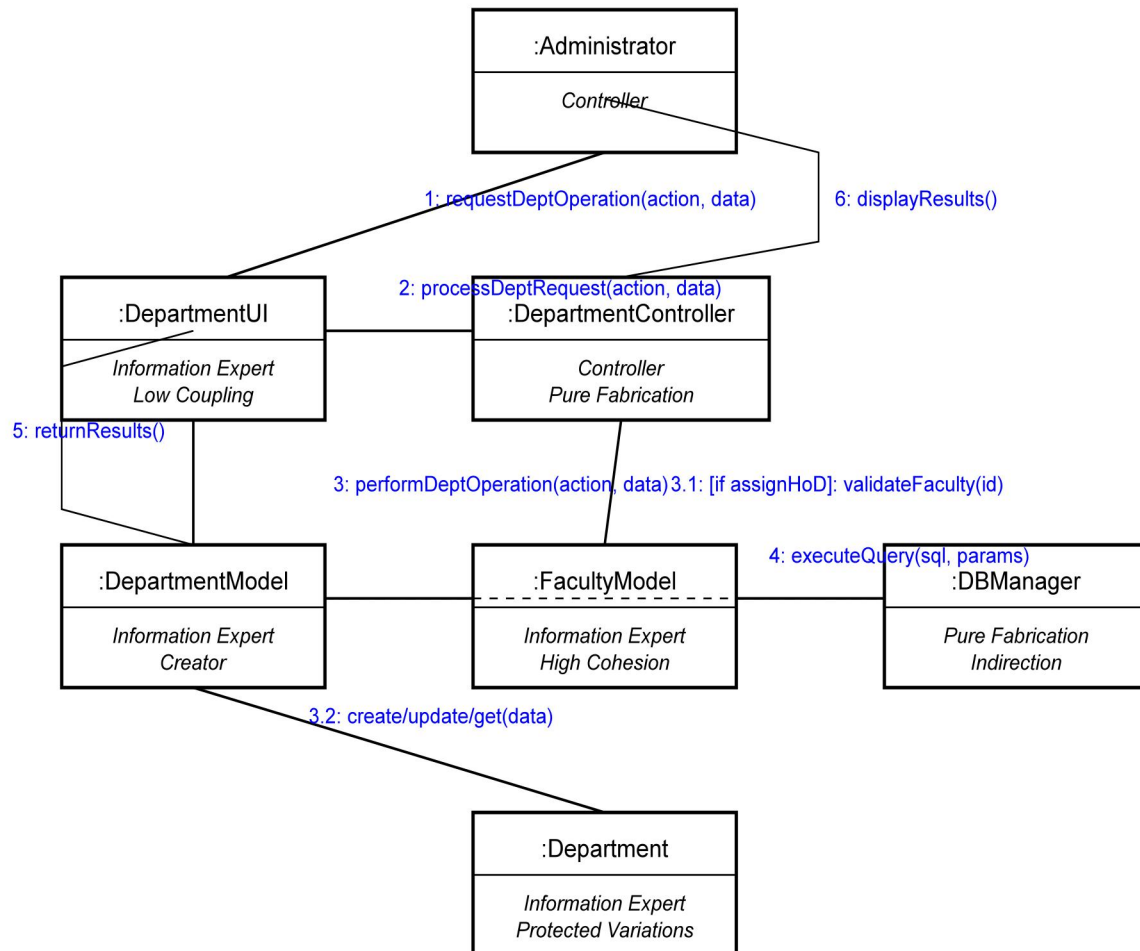
State Diagram:

USE Case diagram:

Communication Diagram:

Administrator

1. Request department action   Add Dept   Edit Dept   Delete Dept   View Depts   Assign HoD   6. Display outcome

Department UI

2. Process request   5. Process results

Controller

3. Execute operation   4. Return results

Database
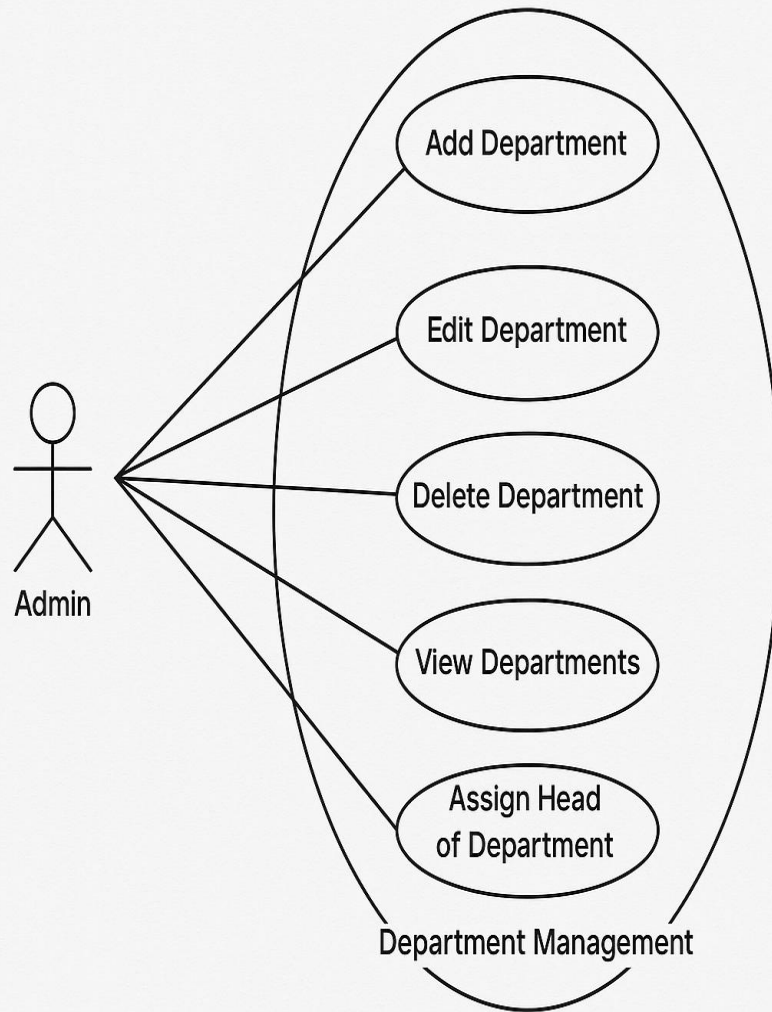
Department Management Communication Diagram with GRASP Principles

I've created a communication diagram that explicitly incorporates GRASP (General Responsibility Assignment Software Patterns) principles into the design. This

approach ensures proper assignment of responsibilities to objects in the Department Management Module.

Objects and Their GRASP Roles:

1.
   - **GRASP Role**: Controller (initiates system operations)
   - Initiates all department management operations

2.
   - **GRASP Roles**: Information Expert (for UI data), Low Coupling
   - Handles user interaction and displaying results
   - Maintains low coupling by delegating business logic to controllers

3.
   - **GRASP Roles**: Controller, Pure Fabrication
   - Coordinates operations between UI and domain objects
   - Pure Fabrication as it's created specifically to handle control tasks

4.
   - **GRASP Roles**: Information Expert, Creator
   - Maintains knowledge about department data
   - Creates Department objects (Creator pattern)

5.
   - **GRASP Roles**: Information Expert, High Cohesion
   - Handles faculty-specific data (focused responsibility)
   - High cohesion as it handles only faculty-related operations

6.
   - **GRASP Roles**: Pure Fabrication, Indirection

- Abstracts database operations (indirection)
- Shields other objects from database implementation details

7.

- **GRASP Roles**: Information Expert, Protected Variations
- Contains core department data and behavior
- Protects system from variations in department requirements