

ISLAMABAD E-9

Submitted by: Rayan Nadeem, Abdul Moiz, Abdullah Farooq

Submitted to: Dr Sumaira Hayat

Course: Software Requirements Engineering

Roll number: 221114, 221028, 220996

Section: BSIT-B

***SRS (1)***

1. **Introduction**

**1.1 Purpose**

The purpose of this document is to outline the enhancements needed for the existing accounts system within the Campus Management System. It aims to detail the development of a new, comprehensive accounting module that encompasses all financial aspects of educational institutions, including admission fees, tuition fees, fines, scholarships, teacher fees, and overall financial management for schools, colleges, and universities

**1.2 Scope**

The SRS aims the enhancement of the existing accounts system to incorporate detailed functionalities for managing admission fees, tuition fees, fines, scholarships, teacher fees, and overall financial tracking for educational institutions. The goal is to provide a robust and user-friendly accounting module that integrates seamlessly with other CMS modules, ensuring efficient financial management.

**1.3 Definitions, Acronyms, and Abbreviations**

* **SRS/DS:** Software Requirements Specification and Design Specification
* **CMS:** Campus Management System
* **Fees:** Various charges associated with educational institutions, including admission fees, tuition fees, fines, scholarships, and teacher fees.
* **Profit and Loss:** Financial statement summarizing revenue, expenses, and net income/loss for the educational institution.
* **Software House:** The organization responsible for developing and enhancing the CMS.
* **Nonfunctional Requirements:** Quality attributes of the software, such as performance, security, and usability.

**1.4 References**

- ISLO Technologies (the softwarehouse that gave us the related software detailes)

**[The reference document is being attached as a second file on the GCR]**

**Description**

This document serves as a comprehensive guide for the enhancement of the accounts system within the Campus Management System (CMS) developed by [Software House]. This document outlines the requirements and design considerations necessary to develop a new, detailed accounting module tailored to the needs of educational institutions, including schools, colleges, and universities.

The SRS/DS begins by defining the purpose and scope of the project, detailing the specific functionalities and features to be included in the enhanced accounts system. It outlines the boundaries of the system and provides definitions for key terms, acronyms, and abbreviations used throughout the document.

Furthermore, the document references relevant resources and documents that inform the development process, ensuring alignment with existing standards, regulations, and best practices. Stakeholders can refer to these references for additional context and guidance.

The overview section also provides a roadmap for navigating the rest of the document, highlighting the organization and structure of the SRS/DS. It prepares stakeholders for the detailed descriptions of requirements, design constraints, nonfunctional requirements, and other factors essential for the successful development and implementation of the enhanced accounting module.

Throughout the document, stakeholders will find a thorough description of external behavior, design considerations, and other factors necessary to provide a complete and comprehensive understanding of the requirements for the software. The SRS/DS aims to facilitate effective communication between stakeholders, developers, and other project participants, ensuring a successful outcome for the enhancement of the accounts system within the CMS.

.

**2. Specific Requirements**

**2.1 Functional Requirements of the Enhanced Campus Management System - Accounting Module:**

1. **Admission Fees Management:**
   * Ability to define admission fees structures for different programs and courses.
   * Automated calculation of admission fees based on selected programs/courses and any applicable discounts or scholarships.
   * Generation of admission fee invoices for students upon enrollment.
2. **Tuition Fees Management:**
   * Flexible setup for tuition fees based on various parameters such as program, semester, credit hours, etc.
   * Integration with student records to automatically generate tuition fee invoices.
   * Option to apply scholarships, discounts, or waivers to tuition fees where applicable.
3. **Student Fines and Penalties:**
   * Tracking of fines for late payments, short attendence , library overdue fines, parking violations, etc.
   * Automated calculation of fines and penalties with configurable penalty rates.
   * Notification system for students regarding outstanding fines and penalties.
4. **Scholarship Management:**
   * Capability to define different scholarship programs with eligibility criteria.
   * Automated assessment of students' eligibility for scholarships based on predefined criteria.
   * Integration with student records to apply scholarships to applicable students' accounts.
5. **Teacher Fees Management:**
   * Recording and management of salary details for teachers and staff.
   * Flexible setup for salary components such as basic pay, allowances, deductions, etc.
   * Automated generation of salary slips or payment records for teachers.
6. **Profit and Loss Analysis:**
   * Comprehensive reporting module to analyze financial performance.
   * Calculation of total revenue from fees, fines, and other sources against expenses including salaries, overheads, etc.
   * Generation of profit and loss statements on a periodic basis (monthly, quarterly, annually).

**2.2 Usability Requirements:**

1. **Intuitive user interface:** The system will have an interface that is easy to understand and navigate, ensuring that users can perform tasks without confusion. Buttons, menus, and navigation should be logically organized and labeled clearly to minimize the need for user training.
2. **Accessibility features:** The system will be accessible to users with disabilities, such as visual impairments. This may include features like screen readers, keyboard navigation options, and adjustable font sizes and contrast levels.
3. **User training and support:** The system will provide resources for users to learn how to use it effectively, such as user manuals, video tutorials, or interactive guides. Additionally, there should be a support system in place, such as helpdesk assistance or online forums, where users can seek help or report issues.

**2.3 Reliability Requirements:**

1. **Secure data encryption:** All sensitive data stored in the system, such as financial records and personal information, should be encrypted to prevent unauthorized access. This ensures that data remains confidential ,protected from cyber threats.
2. **Role-based access control:** Access to system features and data should be restricted based on users' roles and responsibilities. This prevents unauthorized users from accessing sensitive information or performing actions beyond their authority.
3. **Audit trails for tracking changes:** The system should maintain detailed logs of all user activities and changes made to data. This allows administrators to track who accessed the system, what actions they performed, and when these actions occurred, aiding in accountability and forensic analysis in case of security breaches or data discrepancies.

**2.4 Performance Requirements:**

1. **Seamless integration with existing modules:** The accounting system will seamlessly integrate with other modules of the campus management system, such as student acdamic records etc, to ensure smooth data flow and consistency across the platform.
2. **Compatibility with different platforms and devices:** The system should be compatible with various operating systems (e.g., Windows, macOS, Linux) and devices (e.g., desktop computers, laptops) to accommodate diverse user preferences and needs.
3. **Data import/export functionalities:** The system should support easy import of data from external sources, such as spreadsheets or databases, and allow for exporting data in standard formats for external analysis or backup purposes.

**2.5 Supportability Requirements:**

1. **Provision for ongoing support:** There will be a dedicated support team or helpdesk available to assist users with technical issues, answer questions, and provide guidance on system usage. This ensures that users can rely on prompt assistance whenever needed.
2. **Regular updates and maintenance:** The system will undergo regular updates and maintenance to address bugs, vulnerabilities, and performance optimizations. This ensures that the system remains reliable, secure, and compatible with evolving technologies.
3. **Compatibility with future technologies:** The system will be designed with flexibility and scalability in mind to adapt to future technological advancements and changes in user requirements. This prevents the system from becoming obsolete and ensures long-term viability.

**2.6 Design Constraints:**

1. **Adherence to data privacy regulations:** The system must comply with relevant data privacy laws and regulations to protect users' personal information and ensure legal compliance. This may include implementing data protection measures, obtaining user consent, and providing transparency regarding data handling practices.
2. **Modular and Scalable Architecture:**

The system should be designed with a modular architecture to facilitate scalability. This will allow the system to handle increased user loads, data volumes, and additional functionalities without significant disruption. The architecture should support the easy addition of new modules and features as needed.

1. **High Availability and Disaster Recovery:**

The system must be designed to ensure high availability, with minimal downtime. This includes implementing redundancy, failover mechanisms, and disaster recovery plans to protect against data loss and ensure continuous operation.

1. **Security Standards Compliance:**

The system must adhere to industry-standard security practices, including secure coding standards, regular security audits, and vulnerability assessments. This ensures the system is protected against common threats and vulnerabilities.

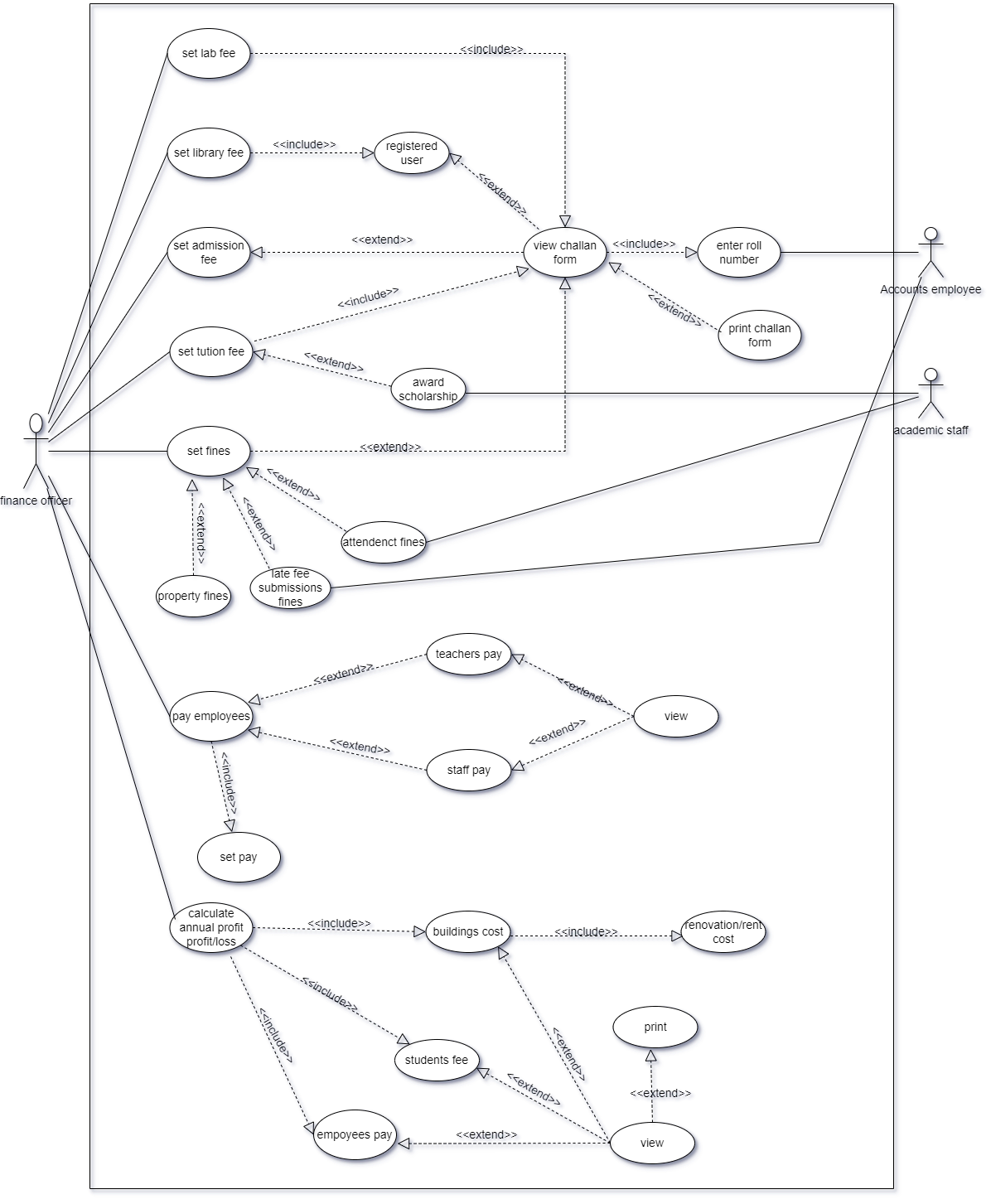
1. **Consideration of budget and resource constraints:** The system design should take into account budgetary limitations and resource availability, ensuring that development, deployment, and maintenance costs remain within acceptable limits. This may involve prioritizing essential features, optimizing resource utilization, and exploring cost-effective solutions.

**2.7** **User Interfaces:**

1. **Intuitive User Interface (UI):**
   * The UI will feature navigation menus and clearly labeled buttons for easy interaction.
   * It will include visually distinct sections for different functionalities such as admission, tuition, fines, and financial reports.
   * Input fields will have appropriate validation and feedback mechanisms to guide users in entering correct data.
2. **Accessibility Features:**
   * The UI will support accessibility features such as screen reader compatibility and keyboard navigation.
   * Font sizes, color contrasts, and other visual elements should be adjustable to accommodate users with visual impairments.
   * Text alternatives will be provided for non-text elements such as images and icons.
3. **Hardware Interfaces:**
4. **Server Infrastructure:**
   * The software will be compatible with standard server hardware configurations, including CPU architectures, memory capacities, and storage configurations.
   * It will specify requirements for server-side components such as database management systems and web servers.
5. **Client Devices:**
   * The software will support a range of client devices including desktop computers, laptops.
   * It will be compatible with common operating systems such as Windows, macOS, Linux, Android, and iOS.
6. **Software Interfaces:**
7. **Integration with Campus Management System:**
   * + The accounting module will seamlessly integrate with other modules of the campus management system, such as student records, course management, and faculty information.
     + APIs or web services should be utilized to facilitate data exchange and communication between different components of the system.
8. **Data Import/Export:**
   * + The software will support standard data exchange formats such as CSV, XML, or JSON for importing and exporting data.
     + Integration with external databases or third-party applications will be possible through standardized APIs or data connectors.
9. **Security Interfaces:**
   * + Integration with security protocols and mechanisms such as SSL/TLS for encrypted communication and OAuth for secure authentication.
     + The software should adhere to industry-standard security practices to ensure data confidentiality, integrity, and availability.

**3.Diagrams**

**3.1 Use case diagram**



* 1. **Use Case Description:**

**Use Case ID: UC001 - Manage Admission Fees**

* **Actors:** Finance Officer
* **Feature:** Admission Fee Management
* **Pre-condition:**
  1. The Finance Officer must be logged into the institution's administrative system.
  2. The fee module must be accessible to the Finance Officer.

|  |  |  |
| --- | --- | --- |
| **Step** | **Action** | **Software Reaction** |
| 1 | Finance Officer navigates to the admission fee management section | System displays the admission fee management dashboard. |
| 2 | Finance Officer defines admission fee structures for various courses | System saves the defined fee structures and associates them with respective courses. |
| 3 | Finance Officer sets fee amounts for each defined structure | System updates the fee amounts according to the Finance Officer's input. |
| 4 | Finance Officer updates fee schedules for different academic terms | System modifies fee schedules as per the changes made by the Finance Officer. |
| 5 | Finance Officer manages discounts or waivers for special cases | System applies discounts or waivers to eligible cases as directed by the Finance Officer. |

**Alternate Scenarios:**

|  |  |  |
| --- | --- | --- |
| **Scenario** | **Action** | **Software Reaction** |
| 1a | If the Finance Officer encounters an error while defining fee structures | System prompts the Finance Officer to review and correct the input. |
| 1b | If the Finance Officer attempts to set an invalid fee amount | System alerts the Finance Officer and prevents the action until a valid amount is entered. |

**Post Conditions**

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Admission fee structures, amounts, and schedules are updated and saved in the system. |

**Use Case Cross-referenced:** None

**User Interface Reference:** Admission Fee Management Dashboard

**Use Case ID: UC002 - Manage Tuition Fees**

* **Actors:** Finance Officer
* **Feature:** Tuition Fee Management
* **Pre-condition:**
  1. The Finance Officer must be logged into the institution's administrative system.
  2. The tuition fee module must be accessible to the Finance Officer.

**Scenarios**

|  |  |  |
| --- | --- | --- |
| **Step** | **Action** | **Software Reaction** |
| 1 | Finance Officer navigates to the tuition fee management section | System displays the tuition fee management dashboard. |
| 2 | Finance Officer defines tuition fee rates for different programs | System saves the defined fee rates and associates them with respective programs. |
| 3 | Finance Officer updates the fee rates for different programs | System updates the fee rates accordingly and maintains a record of adjustments. |

**Alternate Scenarios:**

|  |  |  |
| --- | --- | --- |
| **Scenario** | **Action** | **Software Reaction** |
| 1a | If the Finance Officer encounters an error while defining fee rates | System prompts the Finance Officer to review and correct the input. |
| 1b | If the Finance Officer attempts to set an invalid fee rate | System alerts the Finance Officer and prevents the action until a valid rate is entered. |

**Post Conditions**

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Tuition fee rates and payment schedules are updated and saved in the system. |

**Use Case Cross-referenced:** None

**User Interface Reference:** Tuition Fee Management Dashboard

**Use Case ID: UC003 - Manage Fines**

* **Actors:** Finance Officer
* **Feature:** Fine Management
* **Pre-condition:**
  1. The Finance Officer must be logged into the institution's administrative system.
  2. The fine management module must be accessible to the Finance Officer.

**Scenarios**

|  |  |  |
| --- | --- | --- |
| **Step** | **Action** | **Software Reaction** |
| 1 | Finance Officer accesses the fine management section | System displays the fine management dashboard. |
| 2 | Finance Officer defines fine policies and parameters | System saves the defined policies and parameters for fine imposition. |
| 3 | Finance Officer issues fines for late payments, damages, or disciplinary actions | System records the fines and notifies the respective parties involved. |
| 4 | Finance Officer tracks fine payments and updates the status | System maintains a log of fine payments and marks them as paid. |
| 5 | Finance Officer manages fine waivers or appeals | System allows the Finance Officer to review and process waiver or appeal requests. |

**Alternate Scenarios:**

|  |  |  |
| --- | --- | --- |
| **Scenario** | **Action** | **Software Reaction** |
| 1a | If the Finance Officer encounters an error while defining fine policies | System prompts the Finance Officer to review and correct the input. |
| 1b | If the Finance Officer attempts to issue a fine for an invalid reason | System alerts the Finance Officer and prevents the action until a valid reason is provided |

**Post Conditions**

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Fine policies, fines, and related activities are successfully managed and recorded. |

**Use Case Cross-referenced:** None

**User Interface Reference:** Fine Management Dashboard

**Use Case ID: UC004 - Manage Scholarships**

**Actors:** Finance Officer, Academic Staff

**Feature:** Scholarship Management

**Pre-condition:**

1. The Finance Officer and Academic Staff must be logged into the institution's administrative system.
2. The scholarship management module must be accessible to them.

**Scenarios**

|  |  |  |
| --- | --- | --- |
| **Step** | **Action** | **Software Reaction** |
| 1 | Finance Officer and Academic Staff access the scholarship management section | System displays the scholarship management dashboard. |
| 2 | Finance Officer defines scholarship criteria and eligibility requirements | System saves the defined criteria and requirements for scholarship applications. |
| 3 | Academic Staff processes scholarship applications | System records and reviews scholarship applications submitted by students. |
| 4 | Academic Staff awards scholarships to eligible students | System notifies the selected students and updates their scholarship status. |
| 5 | Finance Officer disburses scholarship funds to awarded students | System initiates the disbursement process and transfers funds to the students' accounts. |

**Alternate Scenarios:**

|  |  |  |
| --- | --- | --- |
| **Scenario** | **Action** | **Software Reaction** |
| 1a | If the Finance Officer or Academic Staff encounters an error while defining scholarship criteria | System prompts them to review and correct the input. |
| 1b | If there are insufficient funds available for scholarship disbursement | System notifies the Finance Officer and Academic Staff and holds the disbursement until funds are available. |

**Post Conditions**

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Scholarship criteria are defined, applications processed, scholarships awarded, and funds disbursed. |

**Use Case Cross-referenced:** None

**User Interface Reference:** Scholarship Management Dashboard

**Use Case ID: UC005 - Set Lab Fee**

* **Actors:** Finance Officer
* **Feature:** Lab Fee Management
* **Pre-condition:**
  1. The Finance Officer must be logged into the institution's administrative system.
  2. The lab fee module must be accessible to the Finance Officer.

**Scenarios**

|  |  |  |
| --- | --- | --- |
| **Step** | **Action** | **Software Reaction** |
| 1 | Finance Officer navigates to the lab fee management section | System displays the lab fee management dashboard. |
| 2 | Finance Officer defines lab fee structures for various courses | System saves the defined lab fee structures and associates them with respective courses. |
| 3 | Finance Officer sets fee amounts for each defined structure | System updates the lab fee amounts according to the Finance Officer's input. |
| 4 | Finance Officer updates fee schedules for different academic terms | System modifies fee schedules as per the changes made by the Finance Officer. |
| 5 | Finance Officer manages discounts or waivers for special cases | System applies discounts or waivers to eligible cases as directed by the Finance Officer. |

**Alternate Scenarios:**

|  |  |  |
| --- | --- | --- |
| **Scenario** | **Action** | **Software Reaction** |
| 1a | If the Finance Officer encounters an error while defining fee structures | System prompts the Finance Officer to review and correct the input. |
| 1b | If the Finance Officer attempts to set an invalid fee amount | System alerts the Finance Officer and prevents the action until a valid amount is entered. |

**Post Conditions**

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Lab fee structures, amounts, and schedules are updated and saved in the system. |

**Use Case Cross-referenced:** None

**User Interface Reference:** Lab Fee Management Dashboard

**Use Case ID: UC006 - Set Library Fee**

* **Actors:** Finance Officer
* **Feature:** Library Fee Management
* **Pre-condition:**
  1. The Finance Officer must be logged into the institution's administrative system.
  2. The library fee module must be accessible to the Finance Officer.

**Scenarios**

|  |  |  |
| --- | --- | --- |
| **Step** | **Action** | **Software Reaction** |
| 1 | Finance Officer navigates to the library fee management section | System displays the library fee management dashboard. |
| 2 | Finance Officer defines library fee structures for various courses | System saves the defined library fee structures and associates them with respective courses. |
| 3 | Finance Officer sets fee amounts for each defined structure | System updates the library fee amounts according to the Finance Officer's input. |
| 4 | Finance Officer updates fee schedules for different academic terms | System modifies fee schedules as per the changes made by the Finance Officer. |
| 5 | Finance Officer manages discounts or waivers for special cases | System applies discounts or waivers to eligible cases as directed by the Finance Officer. |

**Alternate Scenarios:**

|  |  |  |
| --- | --- | --- |
| **Scenario** | **Action** | **Software Reaction** |
| 1a | If the Finance Officer encounters an error while defining fee structures | System prompts the Finance Officer to review and correct the input. |
| 1b | If the Finance Officer attempts to set an invalid fee amount | System alerts the Finance Officer and prevents the action until a valid amount is entered. |

**Post Conditions**

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Library fee structures, amounts, and schedules are updated and saved in the system. |

**Use Case Cross-referenced:** None

**User Interface Reference:** Library Fee Management Dashboard

**Use Case ID: UC007 - Pay Employees**

* **Actors:** Finance Officer
* **Feature:** Employee Payment Management
* **Pre-condition:**
  1. The Finance Officer must be logged into the institution's financial system.
  2. The employee payment module must be accessible to the Finance Officer.

**Scenarios**

|  |  |  |
| --- | --- | --- |
| **Step** | **Action** | **Software Reaction** |
| 1 | Finance Officer accesses the employee payment management section | System displays the employee payment management dashboard. |
| 2 | Finance Officer calculates salaries for employees | System computes salaries based on predefined criteria and input data. |
| 3 | Finance Officer processes payments for employees | System initiates payment transactions and disburses salaries to the respective accounts. |
| 4 | Finance Officer manages deductions or bonuses for individual employees | System allows for adjustments to be made to salaries based on deductions or bonuses. |
| 5 | Finance Officer generates payment reports for record-keeping purposes | System produces reports summarizing payment transactions and financial activities. |

**Alternate Scenarios:**

|  |  |  |
| --- | --- | --- |
| **Scenario** | **Action** | **Software Reaction** |
| 1a | If there are discrepancies in salary calculations | System alerts the Finance Officer to review and correct the calculations. |
| 1b | If a payment transaction fails due to technical issues | System logs the error and prompts the Finance Officer to retry the transaction later. |

**Post Conditions**

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Employee payments are accurately processed, deductions or bonuses applied, and payment reports generated. |

**Use Case Cross-referenced:** None

**User Interface Reference:** Employee Payment Management Dashboard

**Use Case ID: UC008 - Manage Challan Forms**

* **Actors:** Finance Officer Accounts Employee
* **Feature:** Challan Form Management
* **Pre-condition:**

1. The user must be logged into the institution's administrative system.
2. The challan form management module must be accessible to the user.

**Scenarios**

Main Scenario: View Challan Form

|  |  |  |
| --- | --- | --- |
| **Step** | **Action** | **Software Reaction** |
| 1 | User navigates to the challan form section | System displays the challan form management dashboard. |
| 2 | User views the challan form for their respective fees | System retrieves and displays the challan form based on the user’s data. |
| 3 | Accounts Employee verifies the challan form details | System confirms the validity of the challan form. |

##### Main Scenario: Enter Roll Number

|  |  |  |
| --- | --- | --- |
| **Step** | **Action** | **Software Reaction** |
| 1 | Accounts Employee navigates to the student information section | System displays the student information dashboard. |
| 2 | Accounts Employee enters the roll number of a student | System retrieves and displays the student's information based on the roll number. |

##### Main Scenario: Print Challan Form

|  |  |  |
| --- | --- | --- |
| **Step** | **Action** | **Software Reaction** |
| 1 | Accounts Employee navigates to the challan form section | System displays the challan form management dashboard. |
| 2 | Accounts Employee selects a challan form to print | System retrieves and displays the selected challan form. |
| 3 | Accounts Employee initiates the print command | System sends the challan form to the printer and confirms successful printing. |

##### **Alternate Scenarios**

###### Error While Viewing Challan Form

|  |  |
| --- | --- |
| **Scenario** | **Action** |
| 1a | If the user encounters an error while viewing the challan form |

###### Invalid Roll Number

|  |  |
| --- | --- |
| **Scenario** | **Action** |
| 1a | If the Accounts Employee enters an invalid roll number |

#### **Post Conditions**

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | The challan form is accurately displayed and verified for the user. |
| 2 | The student's information is accurately retrieved and displayed. |
| 3 | The challan form is successfully printed and ready for distribution. |

**Use Case Cross-referenced:** None **User Interface Reference:** Challan Form Management Dashboard, Student Information Management Dashboard

**Use Case ID: UC009 - Calculate Annual Profit/Loss**

* **Actors:** Finance Officer
* **Feature:** Financial Management
* **Pre-condition:**
  1. The Finance Officer must be logged into the institution's financial system.
  2. The financial management module must be accessible to the Finance Officer.

**Scenarios**

|  |  |  |
| --- | --- | --- |
| **Step** | **Action** | **Software Reaction** |
| 1 | Finance Officer accesses the financial management section | System displays the financial management dashboard. |
| 2 | Finance Officer inputs financial data for the year | System calculates the annual profit or loss based on the input data. |
| 3 | Finance Officer reviews the calculated profit/loss | System displays a detailed report of the profit/loss calculation. |
| 4 | Finance Officer saves the report for record-keeping | System saves the report in the financial records database. |

**Alternate Scenarios:**

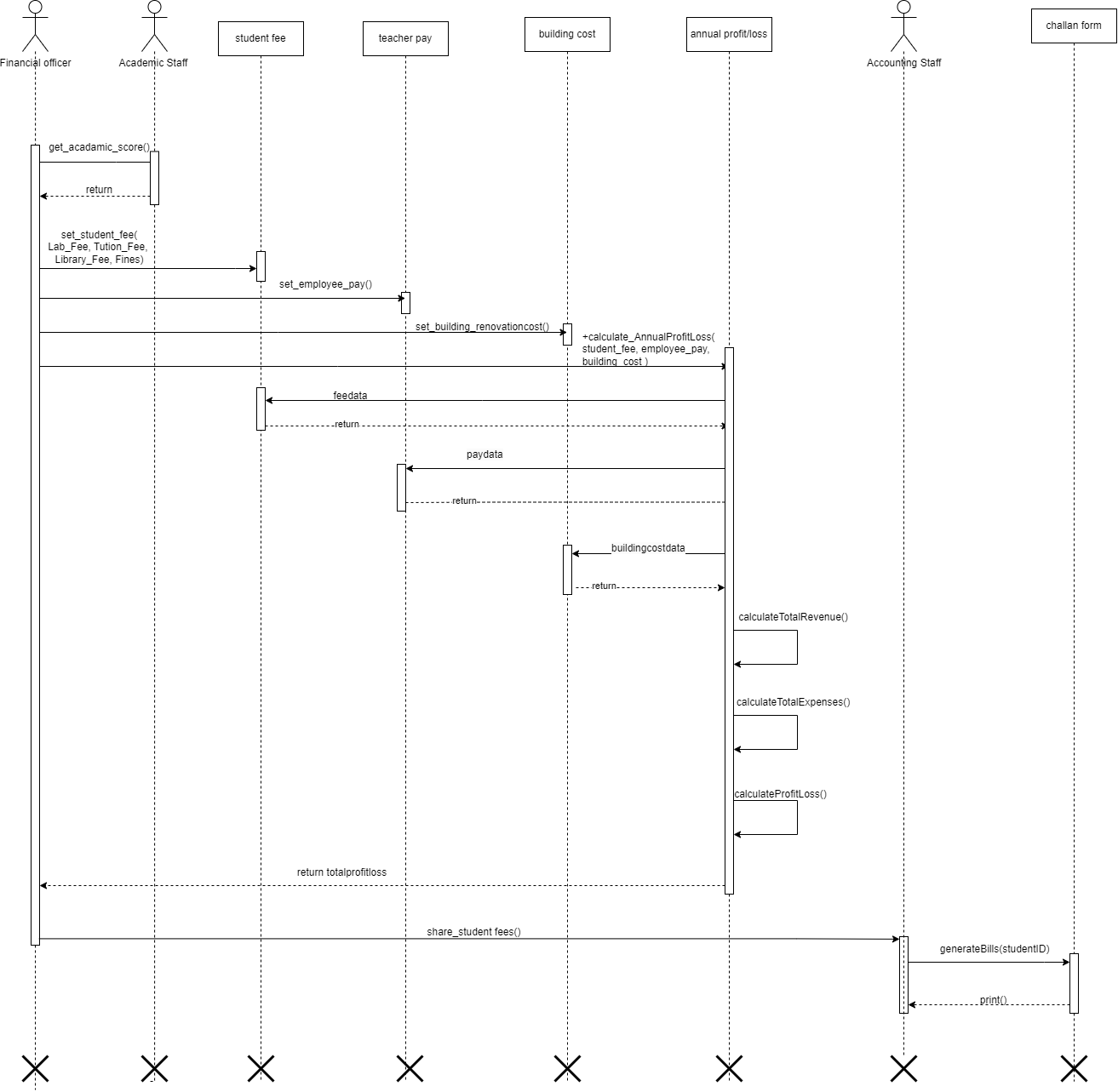
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario** | | **Action** | | **Software Reaction** |
| 1a | If there are errors in the input data | | System prompts the Finance Officer to review and correct the data. | |

**Post Conditions**

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | The annual profit or loss is accurately calculated and recorded in the system. |

**Use Case Cross-referenced:** None

**User Interface Reference:** Financial Management Dashboard

**3.3 Sequence diagram** 

* 1. **Class diagram**

