ASSIGNMENT 5

Software Engineering Lab

Member	Enrollment No
Ramesh Chandra Soren	2022CSB086
Moodu Roopa	2022CSB087
Jitendra Kumar Dodwadiya	2022CSB089
Deep Sutariya	2022CSB090

1. Introduction

Purpose:
This SRS document describes the functional and non-functional requirements for an automated fee collection system at an institute.

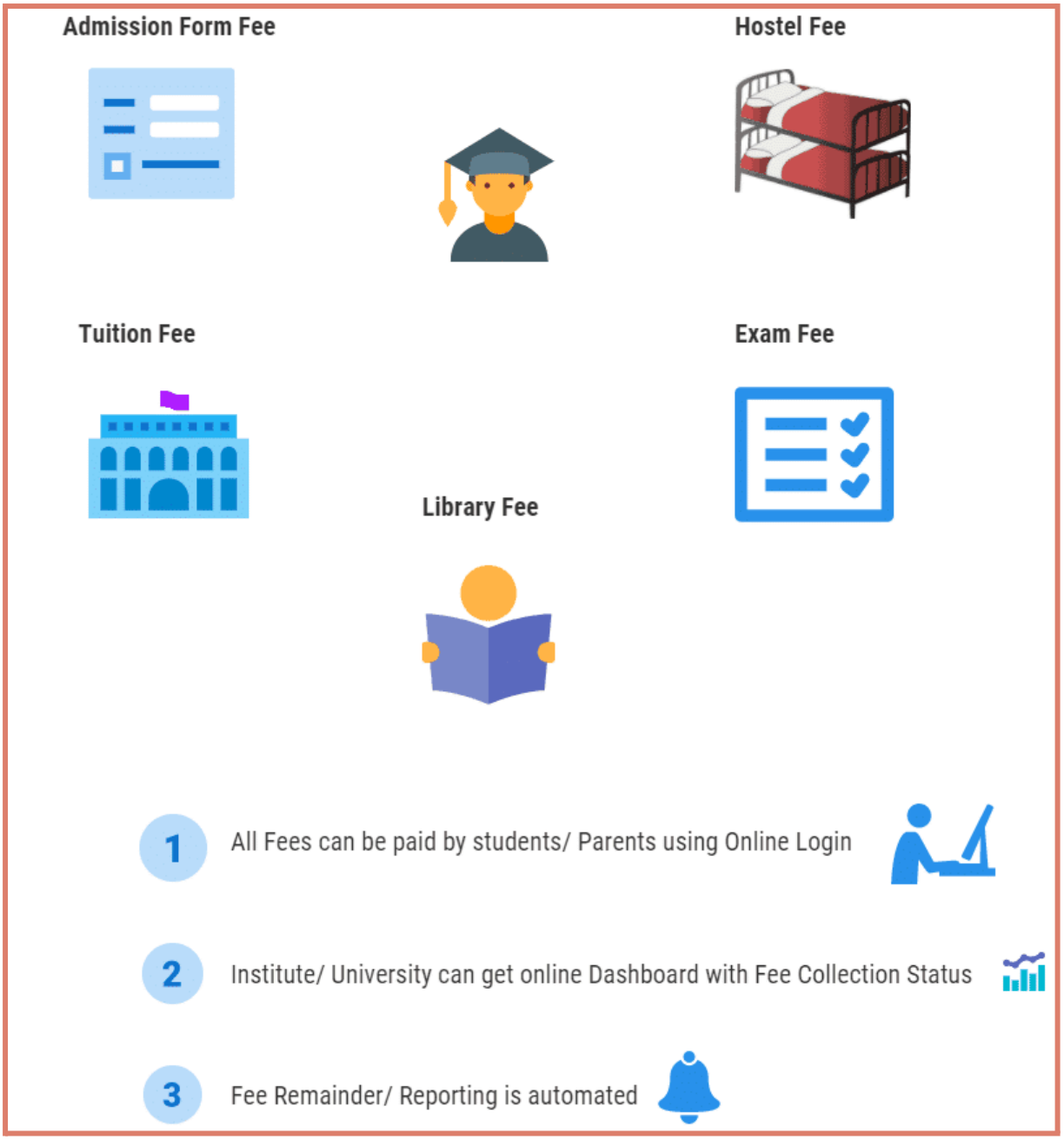
The system is designed to replace manual fee collection with an integrated solution that manages fee calculation, invoicing, payment processing, and reporting.

Scope:
The system will serve as a centralized platform that allows:

- Students can view fee details, make payments, and receive notifications.
- Finance Department to monitor fee collection, generate reports, reconcile accounts, and handle refunds.
- Hostel Wardens to access fee information related to hostel residents.
- Institute Administrators to register new students, update fee structures, and manage overall operations.

Overview:
The document outlines system functionalities including user authentication, fee calculation, payment gateway integration, invoice generation, and real-time notifications.

It also covers system constraints, performance criteria, and design guidelines that ensure ease of use and maintainability.

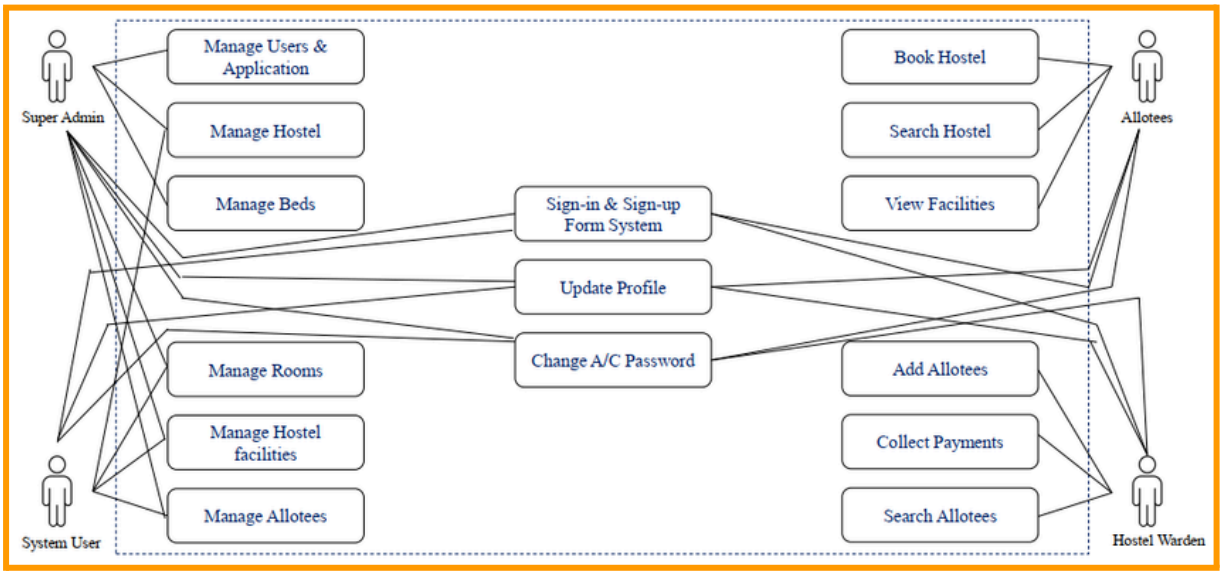


This diagram shows all the necessary components of a student's fees.

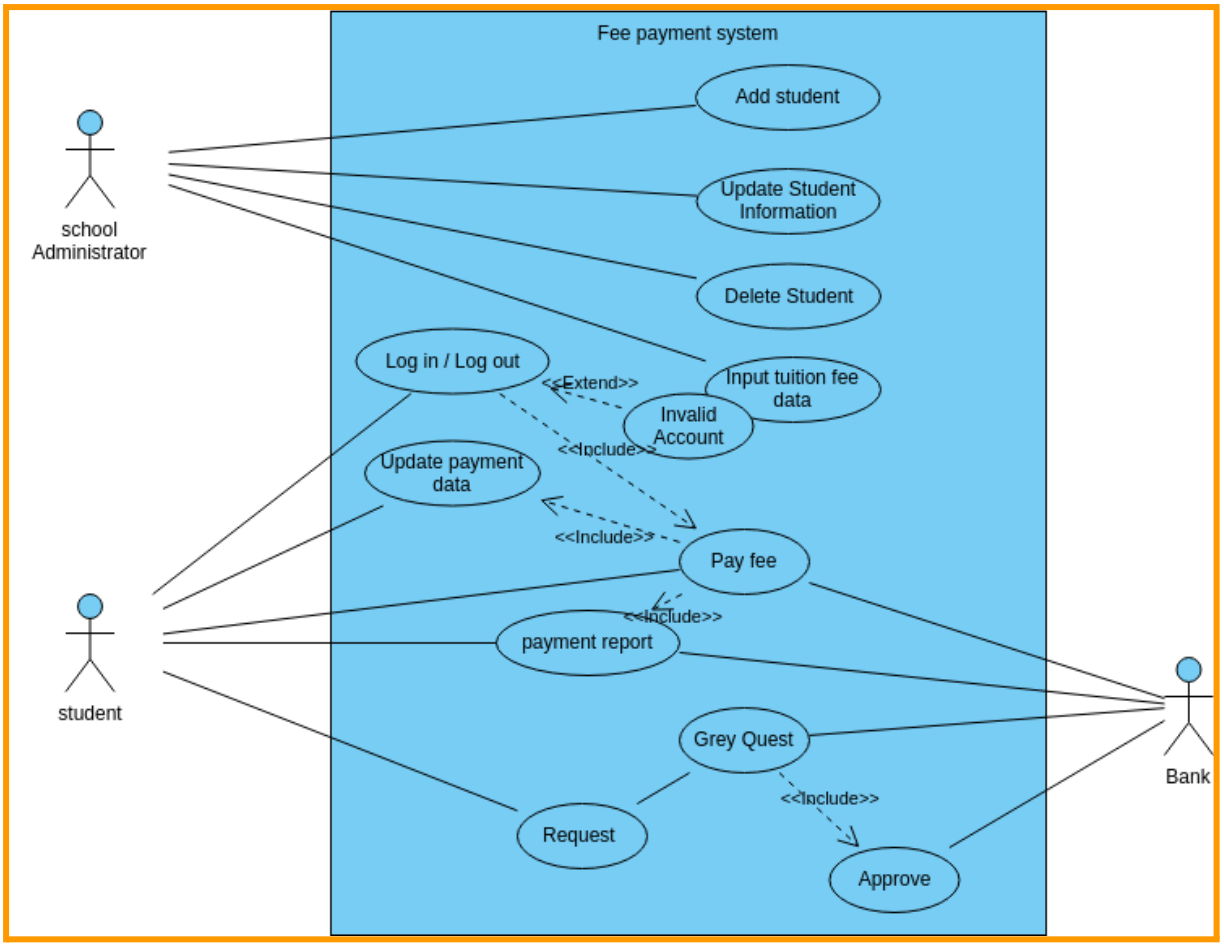
2. General Description

User Profiles & Roles:

- Student: Can log in, view fee statements, make online payments, and download receipts.
- Finance Department: Monitors fee collection status, performs account reconciliation, and generates financial reports.
- Hostel Warden: Accesses fee payment records specific to hostel residents and assists with resolving discrepancies. After the payment of hostel and mess fees they will allocate rooms to each student.
- Administrator: Manages student registration, configures fee structures, updates fee categories, and oversees system administration.



Use case diagram for Hostel management

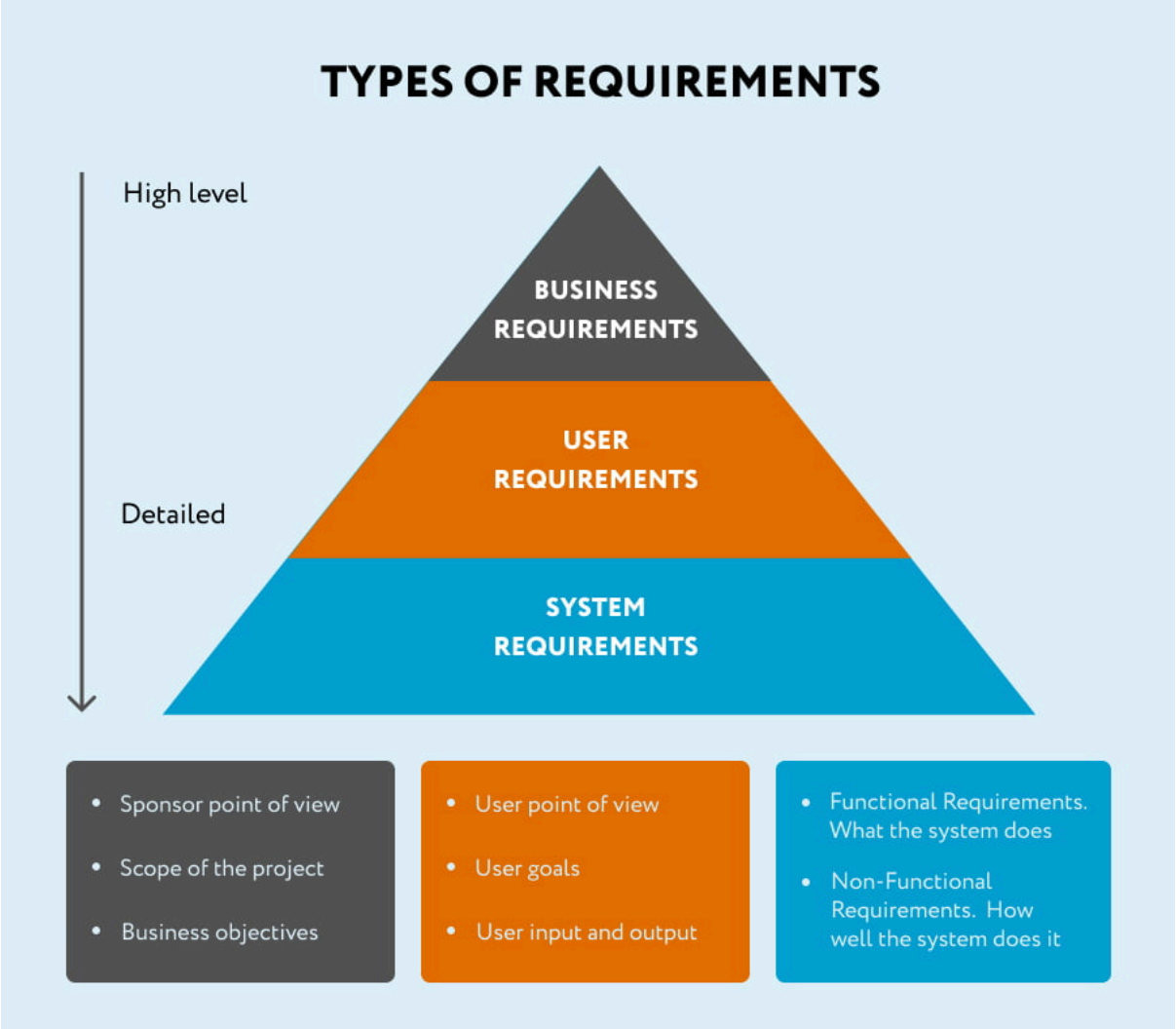


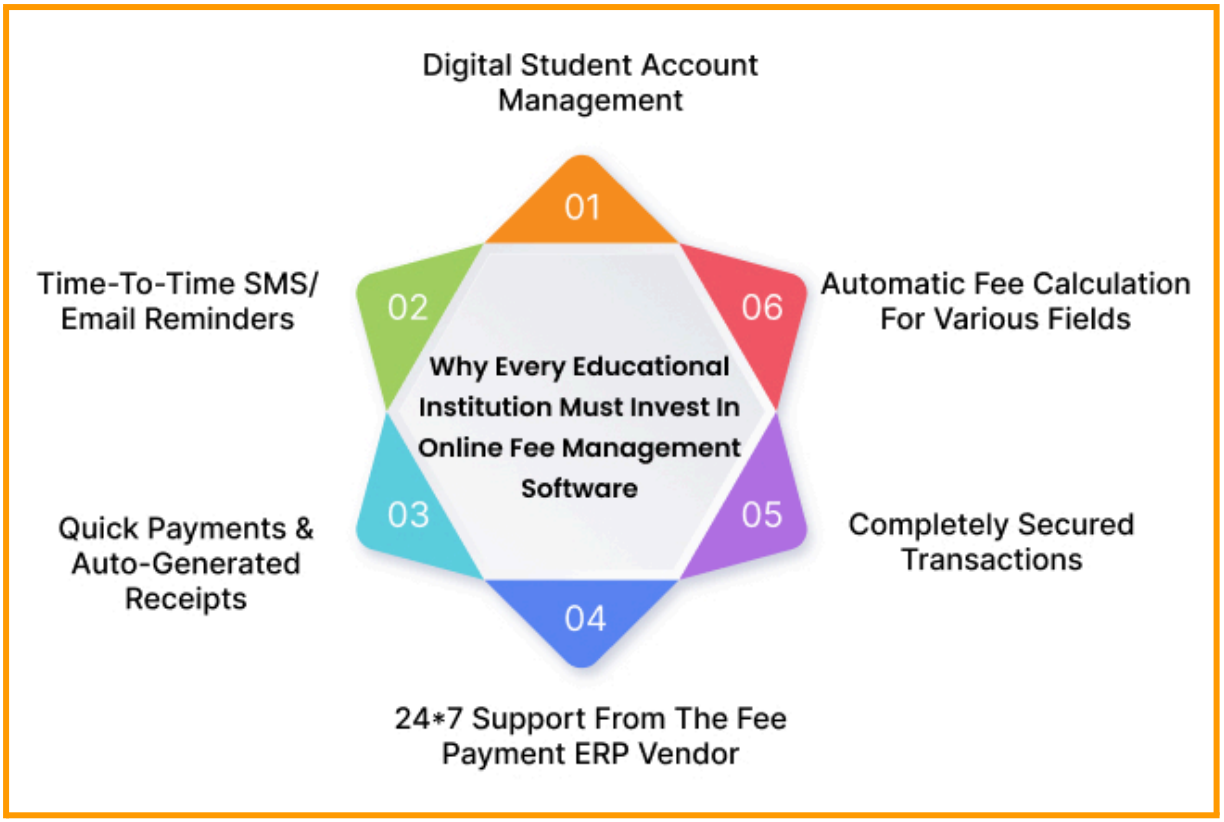
Use case diagram for Institute fees management system

Assumptions:

- Internet connectivity is available for online payment transactions.
- Users are provided with secure login credentials.
- Integration with third-party payment gateways (such as bank APIs or UPI services) is feasible.
- The institute's existing data (student records, fee structures) is available in a digital format for import.

3. Functional Requirements





User Requirements

1. User Registration & Authentication
    - The system shall allow new student registration via a secure form.
    - The system shall provide login functionality for all user roles.
    - Passwords and sensitive data must be stored using secure encryption.
2. Fee Management
    - The system shall automatically calculate fees based on predefined criteria (e.g., tuition, hostel, miscellaneous charges).
    - Administrators shall be able to define, update, and remove fee categories.

- The system shall generate fee invoices for each student.

3. Payment Processing

    - The system shall integrate with payment gateways to support multiple payment methods (e.g., credit/debit cards, UPI, net banking).

    - Students shall be able to make partial or full fee payments.

    - A receipt shall be automatically generated and emailed/downloaded post-payment.

4. Notifications and Reminders

    - The system shall send automated notifications (via email/SMS/app notifications) for payment confirmations and overdue fee reminders.

    - Hostel wardens and finance staff shall receive alerts about pending or overdue fees.

5. Reporting and Reconciliation

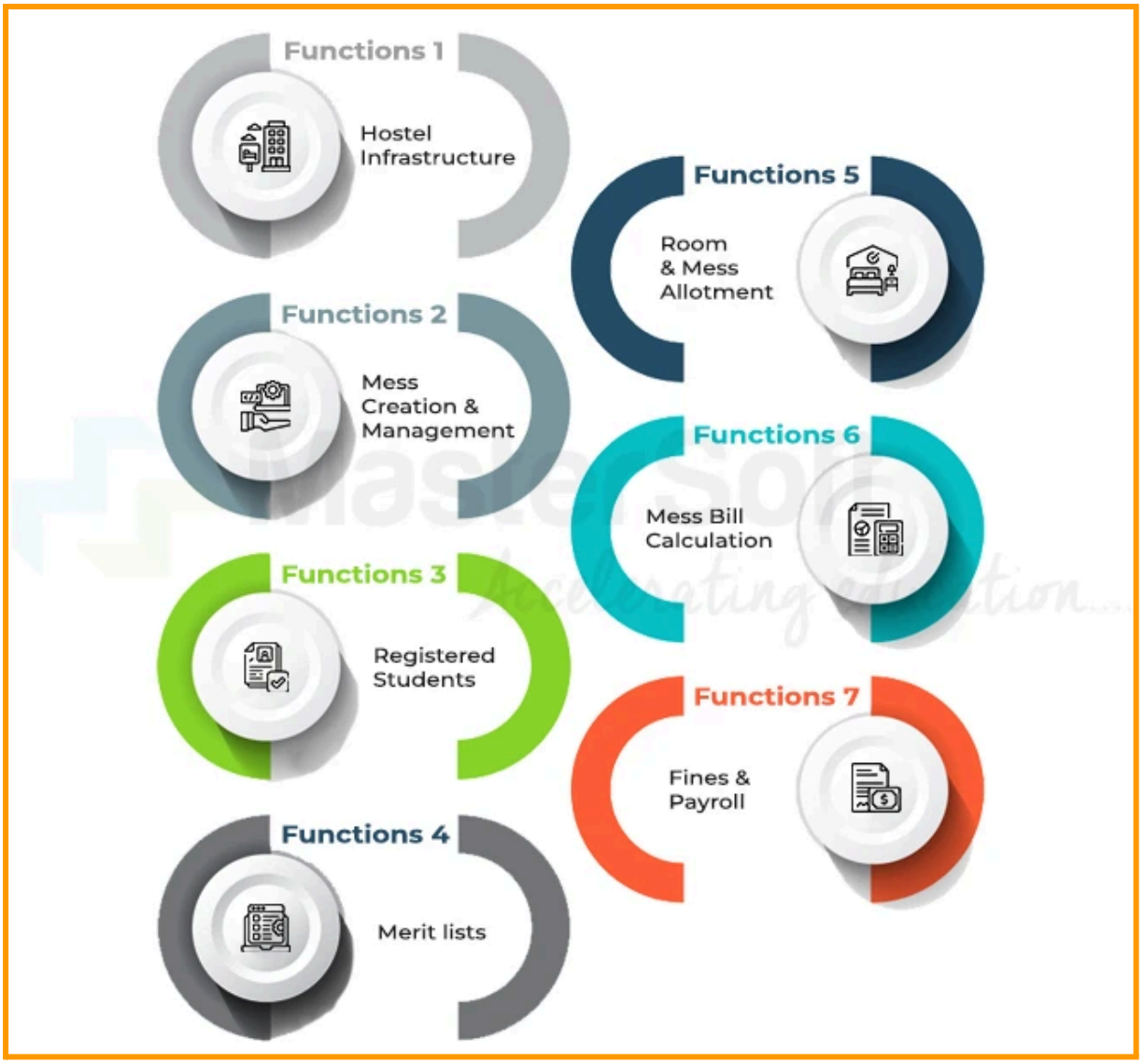
    - The finance department shall have access to dashboards displaying real-time fee collection metrics.

    - The system shall provide detailed financial reports (daily, monthly, annual) and support data export.

6. Access Control and Audit Trail

    - The system shall enforce role-based access controls.

    - All critical actions (e.g., fee updates, payments) shall be logged for audit purposes.



Functional Diagram for Hostel management system

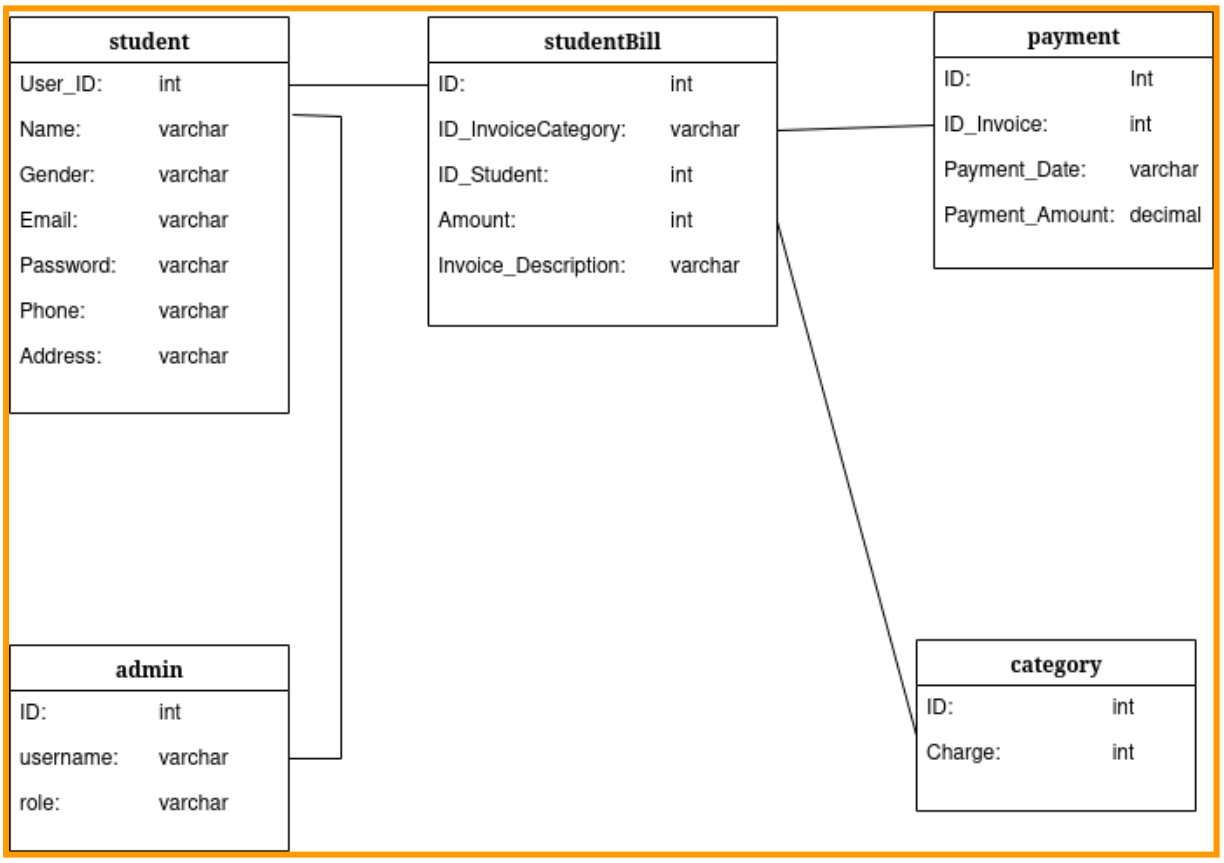
---

4. Non-Functional Requirements

1. Usability:
    - The user interface shall be intuitive and require minimal training.
    - Multi-language support can be provided as per institute requirements.

2. Performance:
    - The system shall respond to user actions (e.g., payment processing, invoice generation) within 2–3 seconds.
    - It should support at least 1,000 concurrent users during peak periods (scalability considerations for larger institutes can be added).
3. Security:
    - Data encryption must be implemented for storage and transmission.
    - The system shall comply with industry standards (e.g., PCI-DSS for payment processing).
4. Maintainability and Scalability:
    - The system design should support easy updates and addition of new features.
    - A modular architecture is preferred to allow scaling up (e.g., using microservices for payment and notification modules).
5. Availability and Reliability:
    - The system shall be available 99.9% of the time during working hours.
    - Data backup and disaster recovery procedures must be implemented.

5. System Architecture (High-Level Overview)



- Client Application:
  Web and mobile interfaces for students, finance personnel, and hostel wardens.

- Application Server:
  Hosts business logic including fee calculation, user authentication, payment processing, and notification services.

- Database:
  A relational database (e.g., MySQL or PostgreSQL) stores student records, fee structures, invoices, and transaction logs.

- Integration Layer:
  APIs for integrating with third-party payment gateways and external notification systems.

For further details on SRS structure and best practices, see resources such as the GeeksforGeeks SRS Format guide

geeksforgeeks.org
and Jama Software's guide on writing SRS documents
jamasoftware.com

---

6. Constraints, Assumptions, and Dependencies

- Constraints:
    - The system must comply with relevant financial and data security regulations.
    - Integration with existing student information systems is required.
- Dependencies:
    - Dependence on third-party payment gateways and SMS/email providers.
    - Hardware infrastructure (servers, network) provided by the institute.
- Assumptions:
    - Users have basic computer literacy.
    - Internet access is reliable during fee collection periods.

7. Appendices

- Glossary:
  Definitions of key terms (e.g., invoice, fee structure, payment gateway, etc.).

- References:
  Links to standards, guidelines, and best practice documents used to develop the SRS.

- Use Case Diagrams:
  Visual representations of interactions between students, finance staff, hostel wardens, and the system.

---

8. Approval and Revision History

- Document Approval:
  To be reviewed and approved by the institute's IT department, finance department, and administrative board.

- Revision History:
  A section to document changes made during the project lifecycle.
