JINKA UNIVERSITY

COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCE
DEPARTMENT OF COMPUTER SCIENCE

Course Title: Software Engineering

Course Code: Cosc3061

Project Title: JKU student cafeteria management system

Advisor Name: Lencho Desalegn (Msc)

Submited To: Lencho Desalegn (Msc)

Prepared By Group: 5

No	Name	Id
1	Solomon Shifera……	Ugr/20994/16
2	Sister Botola……..	Rgr/20091/15
3	Turiset Tadie…….	Rgr/20259/15
4	Getu Arega ……...	Uga/0295/16

JINKA ETHIOPIA

December: 2025

Declaration

We, the undersigned members of Group 5 in the Department of Computer Science at Jinka University, hereby solemnly declare that this project documentation titled "Design and Development of a Web-Based Automated Cafeteria Management System for Jinka University" represents our original work conducted as a partial fulfillment for the course Software Engineering (COSC3061) during the academic year 2023/2024.

This work has not been submitted in whole or in part for any other degree, diploma, or qualification at Jinka University or any other institution of higher learning. All sources of information, data, and literature used in this document have been properly acknowledged and cited according to academic standards. We take full responsibility for the content presented herein and affirm that all contributions have been accurately represented.

We understand that any false claim or misrepresentation in this declaration may result in academic penalties according to Jinka University's regulations.

No	Name	signature
1	**Solomon Shifera**	
2	**Sister Botola**	
3	**Tiruset Tadie**	
4	**Getu Arega**	

This is to certify that we have read this project and that in our opinion it is fully adequate, in scope and quality, as a project for the degree of computer of Science.

Name of Advisor: **signature:**

Mr: Lencho Desalegn _____

It is decleared that this project has been written in compliance with the formatting rules laid down by the college of the university

Acknowledgment

We begin by expressing our profound gratitude to the Almighty God for granting us the wisdom, strength, and perseverance to undertake and complete this challenging academic endeavor. His divine guidance has been our constant companion throughout this journey.

Our deepest and most sincere appreciation is extended to our project advisor, **Mr. Lencho Desalegn (MSc)**, whose exceptional mentorship, unwavering support, and invaluable technical guidance were instrumental in shaping this project from its conceptualization to completion. His profound expertise in software engineering, coupled with his patience in providing critical feedback during our weekly consultations, significantly enhanced the quality of our work and deepened our understanding of software development methodologies.

Our appreciation extends to the **Department of Computer Science** faculty members for imparting the technical knowledge and analytical skills that enabled us to undertake this project. The department's provision of academic resources and learning facilities created an environment conducive to successful project execution.

Acronym	Definition
JKU	Jinka University
SNNPR	Southern Nations, Nationalities, and Peoples' Region
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
PHP	PHP: Hypertext Preprocessor
DBMS	Database Management System
SQL	Structured Query Language
UML	Unified Modeling Language
UI	User Interface
UX	User Experience
ERD	Entity-Relationship Diagram
WBS	Work Breakdown Structure
CRUD	Create, Read, Update, Delete
MVC	Model-View-Controller
AJAX	Asynchronous JavaScript and XML
UAT	User Acceptance Testing
SIS	Student Information System
ROI	Return on Investment
LAMP	Linux, Apache, MySQL, PHP

Table1.1: List of Acronyms

Abstract

The efficient management of student cafeteria services represents a critical operational function within university administration, directly impacting student welfare, satisfaction, and institutional operational costs. **Jinka University** has been operating using an **automatic ID card scanning system** to verify student meal access. However, over time this system has not received the necessary upgrades or maintenance, resulting in recurring scanning errors that disrupt the smooth delivery of services.

This comprehensive project, formally titled **"Design and Development of a Web-Based Automated Cafeteria Management System for Jinka University,"** aims to holistically address these multifaceted challenges through the strategic application of modern web technologies and software engineering principles. The proposed solution is a robust, scalable, three-tier web application architecture that will systematically digitize the entire cafeteria management workflow. Its core technological functionalities include a centralized digital repository for comprehensive student data management, automated generation of unique finger scan student identification credentials, an intuitive **finger-scanning** interface for instantaneous meal verification and logging, and a sophisticated administrative dashboard for real-time operational monitoring and advanced reporting capabilities.

Table of Contents

List of Table

List of figure

CHAPTER 1:

1.1 INTRODUCTION

This document outlines the comprehensive design and development of a **Web-Based Automated Cafeteria Management System** for Jinka University. The project aims to address critical operational inefficiencies in the university's current cafeteria service, which relies on an outdated and error-prone ID card scanning system. Persistent issues such as unauthorized ID usage, frequent scanning failures, manual record-keeping, and lack of real-time reporting have led to service delays, student dissatisfaction, and resource wastage (1).

To resolve these challenges, this project proposes a modern, secure, and scalable web-based solution that integrates fingerprint-based student identification, automated meal verification, and a centralized administrative dashboard. Developed using a structured software engineering approach and the Incremental Model, the system is designed to enhance service efficiency, ensure data accuracy, and improve the overall dining experience for students and staff.

1.2. Background of the Organization

Jinka University (JKU) stands as a prominent public higher education institution established in the Southern Nations, Nationalities, and Peoples' Region (SNNPR) of Ethiopia, specifically situated in the Ari Zone. The university maintains a strong commitment to providing comprehensive quality education while simultaneously focusing on supporting overall student welfare through various auxiliary services. Among these essential services, the student cafeteria represents a vital component of campus life, serving as a primary source of daily nutrition for a significant portion of the student population. (2)

The cafeteria currently operates from a central building strategically located on the main campus, housing two primary functional sections: a dedicated kitchen facility for food preparation and a multi-purpose hall that serves dual functions as both a dining area for students and a venue for university meetings, seminars, and events. This dual-purpose nature of the facility underscores the critical need for an exceptionally efficient and rapid service system to minimize disruptions

during transition between different operational modes. The cafeteria currently serves approximately [insert approximate number] students daily across three meal sessions: breakfast, lunch, and dinner.

Historically, the cafeteria service at Jinka University has been operating using an **automatic ID card scanning system** to verify student meal access. However, over time this system has not received the necessary upgrades or maintenance, resulting in recurring scanning errors that disrupt the smooth delivery of services. These errors often slow down the verification process, create long queues, and reduce overall service efficiency.

Additionally, the university cafeteria has long suffered from a **shortage of dining room space**. As the student population has continued to grow, the limited room capacity has become insufficient to comfortably accommodate all students during peak meal hours. This has led to overcrowding, long waiting times, and reduced satisfaction with the cafeteria environment.

These two long-standing issues **Unauthorized ID Usage and frequent ID scan errors** have collectively posed significant challenges to effective service delivery. Their persistent nature highlights the need for modernization and improved infrastructure to enhance the overall cafeteria management system.

1.3. Problem Statement.

The current cafeteria service system at Jinka University, although partially automated through the use of ID card scanning, still faces significant operational challenges that reduce efficiency, delay service delivery, and negatively affect both students and cafeteria staff. The major problems identified are outlined below:

1. Unauthorized ID Usage:

a student can potentially use another student's ID to access cafeteria services. In practice, this means that a student could take another student's ID, change only the profile picture, and still use the system to obtain meals. This flaw undermines the integrity of the cafeteria management system, enabling unauthorized meal consumption, increasing food wastage, and making it difficult to maintain accurate records of student participation. The proposed system aims to eliminate this vulnerability by implementing a more secure, automated verification process that ensures each student can only use their own ID, potentially leveraging additional authentication markers or indicators. By correcting this loophole, the system will prevent fraudulent access, enhance operational accuracy, and improve fairness in meal distribution. (3)

2. Frequent ID Card Scanning Errors.

Although the university uses an **automatic ID card scanning system**, it often experiences technical errors such as:

- Failure to read or detect ID cards
- Intermittent connection issues between the scanner and the server
- Slow response time during peak hours

These recurring scanning errors lead to long queues, delays in service, and frequent dissatisfaction among students. Therefore, improving the reliability, speed, and accuracy of the scanning system is essential to ensure smooth meal distribution.

The justification for this project is therefore compelling from multiple perspectives—operational, economic, and strategic. By transitioning to an automated, digitally-driven system, the university can achieve transformative improvements in operational efficiency, realize significant and recurring cost savqings, enhance service quality and student satisfaction, and establish a technological foundation for data-informed management decisions and continuous improvement.

1.4. Objective

1.4.1. General Objective

The overarching general objective of this project is to systematically design, develop, implement, and deploy a secure, reliable, scalable, and user-friendly web-based cafeteria management system that will comprehensively automate the core operational processes of student registration, meal verification, and reporting. This technological solution will fundamentally streamline cafeteria operations, achieve substantial cost reductions, and dramatically improve the overall service experience for the entire Jinka University community, including students, cafeteria staff, and administrative personnel. (4)

1.4.2. Specific Objectives

To successfully achieve the general objective of the project, the following specific, measurable, achievable, relevant, and time-bound (SMART) objectives are defined:

1. **To build a fast and responsive fingerprint-based meal verification interface** integrated with the web system for accurate meal attendance logging, including automatic prevention of duplicate meal attempts through real-time validation.
2. **To implement a secure, role-based authentication and authorization system** that controls access rights for different user groups (administrator, cafeteria staff, and optionally students) to ensure data protection and proper system use.

3. **To design a well-structured and modular system architecture** that includes three core operational modules (Student Management Module, Staff Operations Module, and Administrative Dashboard Module), supported by a secure, normalized, and efficient relational database design.
4. **To conduct a complete and detailed requirement analysis** by actively consulting all key stakeholders (cafeteria staff, students, and university management) in order to clearly identify and document all functional and non-functional system requirements, including workflows, data needs, and performance expectations.
5. **To design a well-structured and modular system architecture** that includes three core operational modules (Student Management Module, Staff Operations Module, and Administrative Dashboard Module), supported by a secure, normalized, and efficient relational database design.
6. **To develop a user-friendly student registration portal** equipped with strong data validation features for capturing full student information, including automatic generation of a fingerprint and a machine-readable barcode for each registered student.

1.5. Methodology

This project will be executed using a structured, disciplined software development methodology. The **Incremental Model** has been strategically selected for its iterative development approach, which allows the system to be built and delivered in successive increments. This methodology is particularly well-suited for accommodating user feedback and evolving requirements while progressively delivering functional modules of the cafeteria management system. The methodology encompasses the following detailed phases:

1. Requirements Gathering and Analysis:

1. Direct Observation

Researchers observed operations in the student cafeteria during breakfast, lunch, and dinner. The focus was on:

- **Service Flow:** Tracking the process from when a student enters to when they receive a meal.
- **ID Verification Process:** Monitoring how efficiently and reliably the automated ID checkers work.
- **Queue Management:** Identifying bottlenecks and measuring wait times, especially during busy hours.

- **Staff Workflow:** Observing how staff managed their tasks during high demand.

2. Interviews

In-depth interviews were conducted with key stakeholders to gather detailed insights. Interviewees included:

- **Cafeteria Staff:** ID checkers who interact with the system daily.
- **Students:** A selection of students from different faculties to understand their experiences.

The goal was to collect first-hand accounts of issues like meal attendance errors, technical problems with ID machines, and challenges in tracking usage.

2. Incremental System Design:

Each increment undergoes focused design activities, including modular architectural planning, normalized database schema design (Entity-Relationship Diagram), user interface (UI) mockups and wireframes, user experience (UX) flow diagrams, and relevant UML diagrams (use cases, sequences, activities, classes) specific to the increment being developed. (5)

3. Incremental Implementation (Coding):

Development is performed incrementally, with each module or functional component coded and integrated into the system progressively. The frontend presentation layer, backend business logic layer, and database persistence layer are developed concurrently for each increment, with continuous integration and frequent testing to ensure stability and functionality.

4. Incremental Testing:

Each increment undergoes thorough testing before integration with the existing system. The testing strategy includes Unit Testing (for individual components and functions), Integration Testing (for interfaces between modules), System Testing (for the workflow of the increment), and User Acceptance Testing (UAT) conducted with cafeteria staff to verify operational suitability and gather feedback for subsequent increments.

5.Deployment and Maintenance:

Each successfully tested increment is deployed to the university server, enabling early use and feedback from the cafeteria staff. Data migration, user training, and support mechanisms are implemented incrementally to ensure smooth adoption. Post-deployment maintenance continues for each increment, including bug fixes, enhancements, and updates, while subsequent increments are developed and integrated until the full cafeteria management system is complete.

1.6. Tools and Technologies

The following tools and technologies have been carefully selected based on technical requirements, development team expertise, scalability considerations, and institutional resource constraints (6):

1. **Design & Modeling Tools:**

- **Figma:** For creating high-fidelity UI/UX prototypes, interactive wireframes, and design system components.
- **/ Lucidchart:** For creating professional UML diagrams, process flowcharts, and Entity-Relationship Diagrams (ERD).

2. **Frontend Development Stack:**

- **HTML5:** To structure web pages semantically with accessibility considerations.
- **CSS3 with Flexbox/Grid:** For advanced styling, responsive layout design, and creating mobile-friendly interfaces.
- **JavaScript (ES6+):** To implement client-side interactivity, dynamic form validation, asynchronous operations (AJAX), and real-time interface updates.

3. **Backend Development Stack:**

- **PHP (v8.x):** As the server-side scripting language to handle business logic, session management, authentication, and database interactions.

4. **Database Management System:**

- **MySQL (v8.x):** As the relational database management system (RDBMS) to securely store all application data with proper integrity constraints and relationships.

5. **Development Environment:**

- **XAMPP / WAMP Stack:** Provides an integrated local server environment with Apache HTTP Server, PHP interpreter, and MySQL database for development and testing.
- **Visual Studio Code:** The primary code editor with extensive extensions for PHP IntelliSense, HTML/CSS support, JavaScript debugging, and version control integration.

6. **Hardware Integration Components:**

- **USB finger Scanners:** Configured in keyboard emulation mode for seamless integration with web input fields without requiring custom drivers.

7. **Version Control:**

- **Git with GitHub/GitLab:** For source code version control, collaboration management, and change tracking throughout the development lifecycle.

1.7. Scope and Limitation of the Project

In-Scope (System Deliverables):

- ✓ A fully functional, web-based cafeteria management application with three distinct user roles: Student (as data subjects), Cafeteria Staff (operational users), and Administrator (system managers).
- ✓ Automated **finger generation and printing** functionality integrated with student records.
- ✓ Real-time **finger printing** interface for meal verification with immediate duplicate prevention.
- ✓ Comprehensive reporting module for generating daily, weekly, monthly, and custom date range meal consumption reports with export capabilities.
- ✓ Secure role-based user authentication and authorization system with session management.
- ✓ Responsive web design ensuring usability across different device types (desktop, tablet).

Out-of-Scope (Current Limitations):

- ✓ **Inventory Management Functionality:** Features for tracking raw material stock levels, supplier management, purchase order processing, or kitchen inventory control are explicitly excluded from this version.
- ✓ **External System Integration:** Automated integration with the university's central student information system (SIS), financial management system, or other campus administrative systems is not planned for the initial release (though API endpoints may be designed for future integration).
- ✓ **Advanced Analytics and Forecasting:** Sophisticated predictive analytics, machine learning-based demand forecasting, or advanced business intelligence features are beyond the scope of this initial implementation.

1.8. Scheduling

The project is scheduled to be completed within one academic semester, as outlined in the Gantt chart below

Task Name	**Start Date**	**End Date**	**Duration**	Oct 2018	Jun 2018
				12/02/2018	08/05/2018
Project Initiation and Planning	12/02/2018	19/02/18	1 Week		
Requirements Gathering and Analysis	20/02/2018	04/03/2018	2 Weeks		
System Design and Modeling	06/03/2018	20/03/2018	2 Weeks		
System Implementation (Coding)	22/03/2018	29/03/2018	1Weeks		
System Testing	01/04/2018	08/04/2018	1Weeks		
Deployment and User Training	12/04/2018	19/04/2018	1 Week		
Documentation Finalization	01/05/2018	08/05/2018	1Weeks		

Figure1.1:Time Scheduling by gant chart

1.9. Significance of the Project

This project holds substantial multi-faceted significance across various stakeholder groups within the university ecosystem:

- ✓ **For Students:** It promises a faster, more reliable, and equitable meal service experience, significantly reducing time spent in queues and ensuring fair access to allocated meals according to eligibility criteria. The transparency and consistency of the automated system will enhance student satisfaction with campus services.
- ✓ **For Cafeteria Staff:** It dramatically reduces their manual, repetitive administrative workload, minimizes human errors in record-keeping, and allows them to refocus their efforts on improved food service delivery and customer interaction rather than administrative tasks.
- ✓ **For University Management:** It provides unprecedented data transparency and operational visibility, enables precise cost control by reducing food waste and optimizing labor allocation, and supports evidence-based strategic planning with accurate consumption analytics and reporting.
- ✓ **For Jinka University as an Institution:** It represents a tangible, impactful step towards digitizing campus services, improving operational modernism and administrative efficiency, and demonstrating institutional commitment to leveraging technology for service improvement.

1.10. Feasibility Study

1.10.1. Technical Feasibility

The project is highly technically feasible based on multiple factors. The selected technology stack (HTML5, CSS3, JavaScript, PHP, MySQL) is mature, widely supported, well-documented, and entirely open-source, eliminating licensing costs and proprietary dependencies. All development team members possess foundational to intermediate knowledge in these core technologies, and extensive learning resources (online tutorials, documentation, community forums) are readily accessible for skill enhancement. The hardware requirements (standard computers, commercially available USB finger scanners) are commonplace, affordable, and easily procurable. The proposed system architecture follows established web development patterns that are well-understood and proven in similar institutional contexts. (7)

1.10.2. Economic Feasibility

The project demonstrates strong economic feasibility with a compelling return on investment (ROI) profile. The primary costs are limited to one-time procurement of finger scanners (estimated at ETB [insert amount] per unit) and potential minimal server hosting fees if university infrastructure is unavailable. These costs are negligible compared to the anticipated recurring annual savings from reduced labor costs (estimated 30-40% reduction in administrative staffing requirements) and decreased food waste (estimated 15-25% reduction through duplicate meal prevention). The use of free, open-source software throughout the development stack keeps development costs minimal. A preliminary cost-benefit analysis indicates the system would likely pay for itself within [insert timeframe] of operation through efficiency gains alone.

1.10.3. Operational Feasibility

The system is deliberately designed for operational simplicity and ease of adoption. The primary interface for cafeteria staff is streamlined to a single scanning action with clear visual feedback, requiring minimal training. The administrative dashboard employs familiar web interfaces and intuitive data visualization. Given the basic computer literacy of most staff members and the planned comprehensive training program, resistance to change is expected to be manageable, and the new system will be rapidly adopted as it directly alleviates their most pressing operational pain points (manual record-keeping, queue management, dispute resolution).

1.10.4. Legal Feasibility

The system will be developed in full compliance with Jinka University's internal policies regarding student data privacy, IT system usage, and administrative procedures. It will adhere to fundamental principles of data protection by implementing appropriate access controls, audit trails, and data encryption. Since the system does not handle sensitive financial information or highly confidential personal data beyond basic student information, its legal and regulatory compliance requirements are manageable within the university's existing governance framework.

1.11. Work Breakdown Structure

Phase	Major Tasks	Duration	Key Deliverables
1. Initiation & Planning	Project Charter Finalization, Team Formation & Role Assignment, Initial Stakeholder Identification, Feasibility Analysis, Preliminary Scheduling & Resource Planning.	1 Week	Project Charter, Initial Project Plan
2. Requirement	Stakeholder Interviews (Cafeteria Staff,	2 Weeks	Requirements

Analysis	Management), Process Observation & Documentation, Functional Requirements Specification, Non-Functional Requirements Specification, Requirements Validation.		Specification Document, Process Flow Diagrams
3. System Design	System Architecture Design, Database Schema Design (ERD), UI/UX Wireframing & Prototyping, Detailed UML Diagram Creation (Use Cases, Sequences, Activities, Classes), Technology Stack Finalization.	2 Weeks	System Design Document, UI Prototypes, Database Schema
4. Implementation	Frontend Development (All Pages & Components), Backend Development (PHP Business Logic), Database Implementation & Population, Module Integration, Barcode Scanner Integration.	1 Weeks	Functional System Codebase, Integrated Modules
5. Testing	Unit Testing (PHPUnit), Integration Testing, System Testing (End-to-End Workflows), Performance & Security Testing, User Acceptance Testing (UAT) with Cafeteria Staff.	1 Weeks	Test Reports, Bug Fix Log, UAT Sign-off
6. Deployment	Production Server Configuration & Optimization, System Installation & Configuration, Data Migration (if applicable), User Training Program Delivery, Go-Live Transition Support.	1 Week	Deployed System, Training Materials, User Guides
7. Documentation & Closure	Final Technical Documentation Compilation, User Manual Preparation, Final Project Report Writing, Project Presentation Preparation, Project Closure Formalities.	1Weeks	Comprehensive Project Documentation, Final Presentation

Table2.1: Detailed Work Breakdown Structure (WBS) with Timeline

Resource Type	Specific Allocation	Responsibilities
Human Resources	4 computer science Students (Group 5) 1 Project Advisor (Mr. Lencho Desalegn)	Design, Development, Testing, Documentation Guidance, Review, Evaluation

Hardware Resources	4 Personal Development Computers 2-3 USB finger Scanners (testing & deployment) University Server Access	Development Environment System Integration Production Hosting
Software Resources	VS Code, XAMPP, Figma, Git, Web Browsers Windows/Linux OS	Development, Design, Version Control, Testing
Financial Resources	Estimated ETB 35,160 (as per proposal budget) Potential University Funding	Hardware Procurement, Printing, Documentation

Table3.1: Comprehensive Project Resource Allocation Matrix

CHAPTER 2: Requirement Analysis And Specification

2.1 Current System

The current cafeteria management system at Jinka University operates as a **hybrid manual-digital process**. While an **automatic ID card scanning system** exists, it has not been updated or properly maintained, leading to operational inefficiencies. (8)

2.1.1 Problems of the Existing System

The current cafeteria management system at Jinka University suffers from significant operational, technical, and administrative deficiencies that hinder efficient service delivery. These problems have been identified through direct observation, stakeholder interviews, and analysis of historical service records:

1. Unauthorized Access and Identity Fraud:

- ✓ **Photo Substitution Vulnerability:** The current ID card system allows students to replace the profile picture on another student's card and use it to access cafeteria services. This loophole enables multiple meal consumptions per student and results in substantial food wastage.
- ✓ **Lack of Biometric Verification:** The absence of unique biological identifiers makes it impossible to definitively confirm that the person presenting the ID card is its legitimate owner.

2.Frequent ID Card Scanning Errors:

- ✓ The scanning system often fails to read cards, experiences connection drops, or slows down during peak hours.
- ✓ This causes long queues, delays, and student dissatisfaction.

3.Lack of Real-Time Monitoring and Reporting:

- ✓ Administrators cannot access real-time data on meal consumption, leading to reactive rather than proactive management.

2.1.2 Strengths of the Existing System

Despite its shortcomings, the current system has the following strengths:

1. **Simplicity and Low Technical Barrier:**

- ✓ Requires minimal training for staff to operate.
- ✓ No dependency on continuous power or internet connectivity.

2. **Immediate Fallback to Manual Process:**

- ✓ In case of technical failure, staff can revert to manual verification without halting service entirely.

3. **Direct Human Interaction:**

- ✓ Allows for personal verification and discretionary service in special cases (e.g., forgotten ID).

2.2 Business Rules

The following operational and technical rules govern the cafeteria service:

1. **Eligibility Rules:**

- ✓ Only students with "cafe active" status are permitted meal access.
- ✓ Each student is entitled to **one meal per session** (breakfast, lunch, dinner).

2. **Meal Session Rules:**

- ✓ Meals are served only within fixed time windows.
- ✓ No carry-over of unused meals to other sessions or days.

3. **Technical Constraints:**

- ✓ Must use existing USB finger scanners in keyboard emulation mode.
- ✓ Must operate within the university's existing network and server infrastructure.

4. **Budgetary and Timeline Constraints:**

- ✓ Total project cost must not exceed ETB 35,000.
- ✓ System must be fully deployed within one academic semester.

5. **Data Management Rules:**

- ✓ Student data must be retained for at least one academic year.

- ✓ Access to data must be role-based and audited.

2.3 Proposed System

2.3.1 Overview

The proposed system is a **web-based automated cafeteria management system** designed to replace the current hybrid process. It will introduce a **finger-based student identification system** to eliminate ID misuse and scanning errors. The system will be built using a **three-tier architecture**:

- **Presentation Layer:** HTML5, CSS3, JavaScript – responsive web interfaces for admins, staff, and students.
- **Application Layer:** PHP 8.x – handles business logic, authentication, and real-time processing.
- **Data Layer:** MySQL 8.x – secure, normalized database for student, meal, and user data.

- Key features include:

- Automated finger generation and printing for students.
- Real-time meal verification with duplicate prevention.
- Admin dashboard for monitoring, reporting, and user management.
- Role-based access control (Admin, Staff, Student).

2.4. System Requirements Specification

2.4.1. Functional Requirements

1. Student Management Module:

- The system shall allow administrators to register new students with comprehensive profile information including name, department, academic year, and cafe status.
- The system shall automatically generate a unique finger scan for each registered student and associate it with their digital profile.
- The system shall provide search, filter, and bulk operation capabilities for student record management.
- The system shall allow administrators to update student cafe status (active/inactive) with effective date control.

2. Meal Processing

- The system shall provide a dedicated scanning interface for cafeteria staff to process student barcodes.
- The system shall display student name and photo upon successful finger scan for visual verification.
- The system shall check in real-time if a student has already consumed a meal in the current session and prevent duplicate meal logging.
- The system shall successfully log meal transactions with timestamp, finger scan, and meal type when validation passes.
- The system shall provide clear visual and auditory feedback for successful and failed transaction attempts.

2. Reporting and Analytics

- The system shall generate real-time dashboards showing daily meal counts, peak hour analytics, and consumption trends.
- The system shall allow administrators to generate custom reports for specific date ranges with export capabilities (PDF, Excel).
- The system shall provide graphical visualizations of meal consumption patterns and operational metrics.

3. User Management and Security

- The system shall provide secure role-based authentication for administrators and cafeteria staff
- The system shall maintain audit logs of all system activities including logins, data modifications, and meal transactions.
- The system shall enforce session timeouts and password complexity requirements.

Requirement	Priority	Description
Student Registration	High	System shall allow admin to register students with complete details
Finger printed Generation	High	System shall generate unique cafeteria with finger printed
Meal Verification	High	System shall verify student eligibility using finger printed
Attendance	High	System shall record all meal transactions in real-time

Tracking		
Admin Dashboard	High	System shall provide comprehensive admin interface
Report Generation	Medium	System shall generate daily, weekly, monthly reports
User Management	Medium	System shall manage different user roles and permissions
Data Backup	Low	System shall perform automated data backups

Table4.1: of Functional Requirements

2.4.2. Non-Functional Requirements

1.Performance Requirements:

- The system shall process meal transactions in less than 3 seconds under normal load conditions
- The system shall support concurrent access by at least 5 verification stations without performance degradation.
- The system dashboard shall load completely within 5 seconds on standard university network connections.
- The database shall efficiently handle at least 3,000 student records and 300,000 monthly transaction records.

2.Reliability and Availability Requirements:

- The system shall maintain 99% uptime during cafeteria operational hours (6:00 AM - 8:00 PM)
- The system shall include graceful degradation features to operate in limited functionality during network outages
- Critical system functions shall have automatic failover mechanisms
- Data backup procedures shall ensure no more than 1 hour of data loss in case of system failure

3.Security Requirements:

- User passwords shall be stored using strong hashing algorithms (bcrypt with appropriate cost factors)
- All system access shall be logged with timestamp, user identification, and action performed
- The system shall implement protection against common web vulnerabilities (SQL injection, XSS, CSRF)
- Student personal data shall be encrypted both in transit (SSL/TLS) and at rest in the database

4.Scalability and Maintainability Requirements:

- The system architecture shall support a 50% increase in student population without significant re-engineering
- The codebase shall be modular with comprehensive documentation for future maintenance
- The system shall provide comprehensive logging for troubleshooting and performance monitoring

Category	Requirement	Metric
Performance	Response Time	< 2 seconds for all operations
Availability	Uptime	99.5% during meal hours
Security	Data Protection	Role-based access control, encrypted passwords
Usability	User Interface	Intuitive, requires less than 30 minutes training
Scalability	User Load	Support up to 5000 students
Reliability	System Stability	Maximum 1 hour downtime per month

Table5.1: non Functional Requirements

CHAPTER 3: System Models and Design

3.1 Introduction

This chapter presents the detailed design and modeling of the **Web-Based Automated Cafeteria Management System for Jinka University**. Building upon the requirements analysis documented in Chapter 2, this chapter translates functional and non-functional specifications into structured visual and architectural representations using the **Unified Modeling Language (UML)**. These models serve as an essential blueprint for the development team, ensuring a clear, consistent, and systematic approach to system construction. (9)

The design phase bridges the gap between *what* the system should do and *how* it will be implemented. It encompasses the definition of system architecture, interaction workflows, data structures, and user interfaces. By employing UML diagrams including **use case diagrams,**

sequence diagrams, activity diagrams, and class diagrams this chapter provides a multi-view perspective of the system, addressing both static structure and dynamic behavior.

Actors

An actor is a person, organization, or external system that interacts with the system.

1. **Administrator:** The primary user responsible for overall system management, user management, configuration, and accessing comprehensive reports.
2. **Cafeteria Staff:** The operational user at the serving counter who uses the system to verify students and log meals. Their interaction is typically limited to the scanning interface.
3. **Student:** A beneficiary of the system who is the subject of the data. Their interactions are primarily passive (being printed) but may include limited active use via a self-service portal.
4. **System:** This is a non-human actor that performs automated tasks, such as sending notifications, generating reports, and running backups. It acts on behalf of other users or on a scheduled basis.

Use Case identification and description

Here are the use cases identified for each actor, derived directly from the functional requirements and system description.

Use Case Name	Description
Manage Student Records	Register new students, update their profiles (name, department, status), and deactivate records.
Manage User Accounts	Create, modify, and deactivate accounts for Cafeteria Staff and other Administrators.
Configure System Settings	Set and modify operational parameters such as meal session timings (Breakfast, Lunch, Dinner).
Perform System Backup & Restore	Manually initiate or schedule database backups and restore data from a backup if needed.

Table6.1: Use Cases for the Administrator

Use Case Name	Description
View Student Details (on print)	Upon a successful print, see the student's name and photo for visual verification.
Log Meal Transaction	The system automatically records a successful meal transaction after a valid print.
Handle Invalid/Duplicate print	Receive clear visual/auditory feedback when a print is invalid or a duplicate meal attempt is detected.
View Personal Transaction History	(Optional) View a log of meal transactions they have processed during their shift.

Table7.1: Use Cases for the Cafeteria Staff

Use Case Name	Description
View Personal Profile	Log in to a self-service portal to view their own cafeteria-related information (e.g., meal plan status).
View Meal History	See a personal history of their own meal consumption over a defined period.
Present finger for print	This is the primary interaction: presenting their finger printed by staff.

Table8.1: Use Cases for the Student

Use Case Name	Description
Generate Unique fingerprint	Automatically create a unique, machine-readable fingerprint when a new student is registered.
Prevent Duplicate Meal	Check in real-time against the database to ensure a student hasn't already eaten in the current meal session.
Log Meal Transaction	Automatically record the timestamp, fingerpint, and meal type upon a successful verification.
Perform Automated Backup	Execute scheduled backups of the database without manual intervention.
Send System	Generate alerts for system errors, low disk space, or suspicious

Notifications	activities.

Table9.1: Use Cases for the System (Automated)

Definition of a Use Case Model

A **Use Case Model** is a fundamental UML representation that describes the **functional requirements** of a system by illustrating how **external actors** interact with the system to achieve specific goals. It focuses on **what the system should do**, rather than how it is internally implemented.

In the context of the *Web-Based Automated Cafeteria Management System for Jinka University*, the use case model visually captures all major activities such as student registration, fingerprint generation, meal verification, report generation, and system backup and shows how each user type participates in these activities.

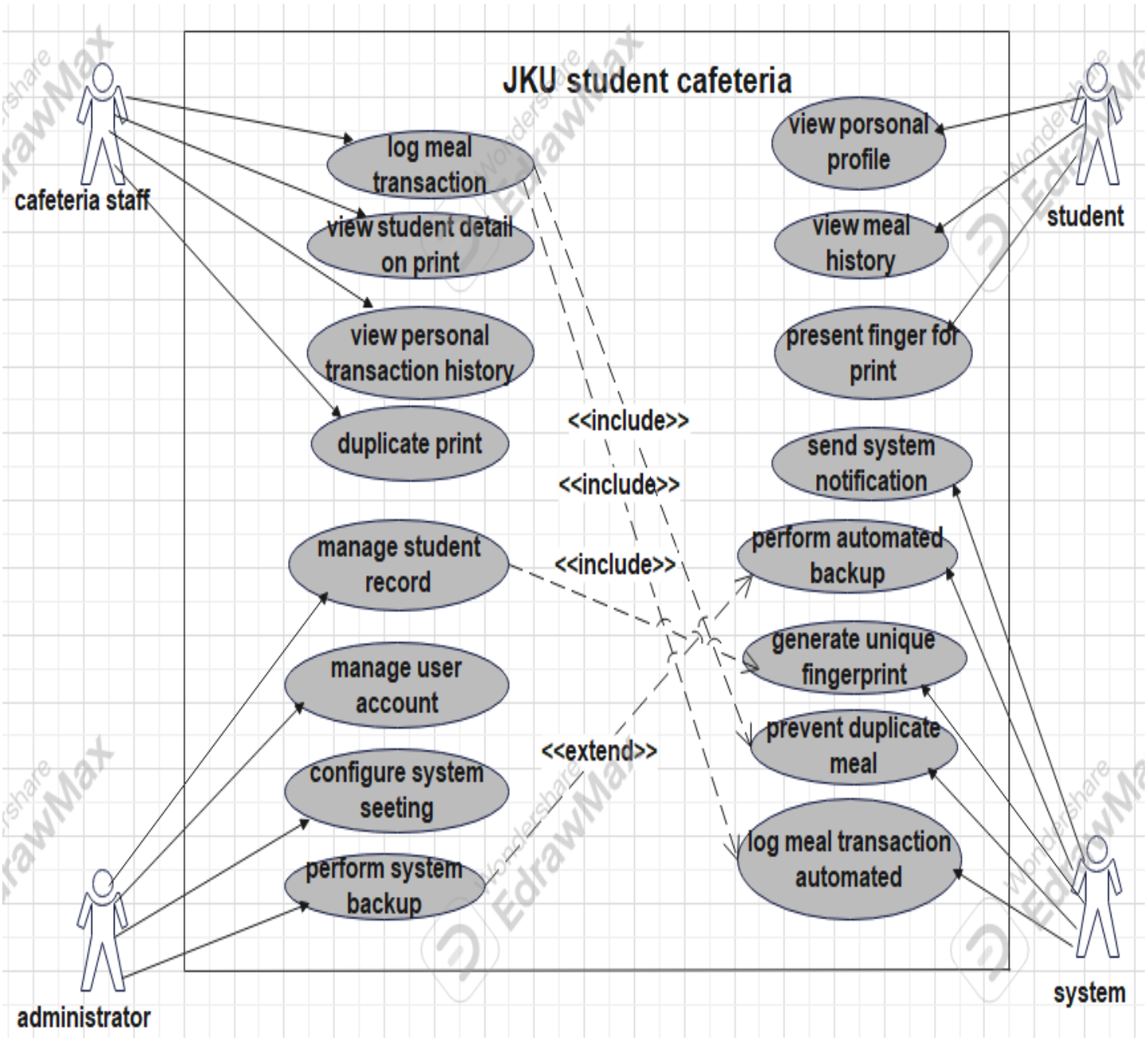

Figure2.1: use case model

Use Case	Generate Unique Fingerprint	
Precondition	New student record has been created	
Actors	System	
Description	Automatically creates unique fingerprint identifier for new students	
Flow of Events	**Actor action**	**System response**
	1. Student record is saved to database	2. System generates unique fingerprint code 3. System associates fingerprint with student record 4. System logs fingerprint generation
Alternation action	**If fingerprint generation fails** 4. System logs critical error 5. Alerts administrator	
Post condition	Unique fingerprint is generated and stored	

Table10.1: Description of Use Cse Generate Unique Fingerprint

Use Case	Meal Verification Process	
Precondition	Student presents finger at cafeteria counter during meal session	
Actors	Student, Cafeteria Staff, System	
Description	Used to verify student eligibility and log meal transaction	
Flow of Events	**Actor action**	**System response**
	1. Student presents finger for scanning 3. System checks student eligibility 5. System displays student name and photo 7. Staff approves transaction	2. Scanner reads fingerprint 4. System verifies student has not eaten in current session 6. Staff visually confirms identity 8. System logs meal transaction
Alternation action	**If duplicate meal detected** 4. System displays "Duplicate Meal" error 5. Transaction is blocked	
Post condition	Valid meal transaction is recorded	

Table11.1: Description of Use Case Meal Verification Process

Use Case	Manage Student Records	
Precondition	The user must browse the website of organization	
	The user must have username and password	
Actors	Administrator	
Description	Used to register new students, update their profiles, and deactivate records	
Flow of Events	**Actor action**	**System response**
	1. Presses the login button on the homepage 3. Enters username and password then click login button 6. The user clicks "Manage Students" button 8. Performs an action (Add/Edit/Remove student)	2. Displays a login form 4. Verifies using login Information 5. Displays user's main page 7. Displays the student management page 9. Processes the request and confirms
Alternation action	**If the user enters an incorrect username and password** 5. The system displays an error message 6. The use case returns to step 2	
Post condition	The administrator manages student records successfully	

Table12.1: Description of Use Case Manage Student Records

Use Case	Prevent Duplicate Meal	
Precondition	Student's fingerprint has been scanned	
Actors	System	
Description	Automatically checks if student has already eaten in current meal session	
	Actor action	**System response**
Flow of Events	1. Fingerprint is scanned	2. System queries database 3. System checks for existing meal record 4. System determines eligibility
Alternation action	**If duplicate found** 4. System blocks transaction 5. Triggers error message	
Post condition	Duplicate meal attempt is prevented	

Table13.1: Description of Use Case Prevent Duplicate Meal

Use Case	Manage User Accounts	
Precondition	The user must browse the website of organization	
	The user must have username and password	
Actors	Administrator	
Description	Used to create, modify, and deactivate accounts for Cafeteria Staff and other Administrators	
Flow of Events	**Actor action**	**System response**
	1. Presses the login button on the homepage 3. Enters username and password then click login button 6. The user clicks "Manage User Accounts" button 8. Performs an action (Add/Edit/Deactivate user)	2. Displays a login form 4. Verifies using login Information 5. Displays user's main page 7. Displays the user account management page 9. Processes the request and confirms
Alternation action	**If the user enters an incorrect username and password** 5. The system displays an error message 6. The use case returns to step 2	
Post condition	User accounts are managed successfully	

Table14.1: Description of Use Case Manage User Accounts

Use Case	Configure System Settings	
Precondition	The user must browse the website of organization	
	The user must have username and password	
Actors	Administrator	
Description	Used to set and modify operational parameters such as meal session timings	
Flow of Events	**Actor action**	**System response**
	1. Presses the login button on the homepage	2. Displays a login form
	3. Enters username and password then click login button	4. Verifies using login Information
		5. Displays user's main page
	6. The user clicks "System Settings" button	7. Displays the system configuration page
	8. Modifies settings and saves	9. Saves new settings and confirms
Alternation action	**If the user enters an incorrect username and password** 5. The system displays an error message 6. The use case returns to step 2	
Post condition	System settings are updated successfully	

Table15.1: Description of Use Case Configure System Settings

Use Case	View Student Details	
Precondition	The user must browse the website of organization	
	The user must have username and password	
Actors	Cafeteria Staff	
Description	Used to see student's name and photo for visual verification after fingerprint scan	
Flow of Events	**Actor action**	**System response**
	1. Presses the login button on the homepage 3. Enters username and password then click login button 6. Scans student's fingerprint	2. Displays a login form 4. Verifies using login Information 5. Displays staff's main scanning page 7. Displays student's name and photo
Alternation action	**If the user enters an incorrect username and password** 5. The system displays an error message 6. The use case returns to step 2	
Post condition	Student details are displayed for verification	

Table16.1: Description of Use Case View Student Details

Use Case	Perform System Backup	
Precondition	System is operational	
Actors	Administrator, System	
Description	Used to backup database manually or automatically	
Flow of Events	**Actor action**	**System response**
	1. Administrator initiates backup	2. System prepares backup process 3. System creates database snapshot 4. System saves backup file 5. System confirms completion
Alternation action	**If backup fails** 5. System displays error 6. Notifies administrator	
Post condition	Backup file is created and stored	

Table17.1: Description of Use Case Perform System Backup

Use Case	View Meal History	
Precondition	The user must browse the website of organization	
	The user must have username and password	
Actors	Student	
Description	Used by student to see their personal meal consumption history	
Flow of Events	**Actor action**	**System response**
	1. Presses the login button on the homepage 3. Enters username and password then click login button 6. The user clicks "Meal History" button	2. Displays a login form 4. Verifies using login Information 5. Displays student's personal dashboard 7. Displays the student's meal history page
Alternation action	**If the user enters an incorrect username and password** 5. The system displays an error message 6. The use case returns to step 2	
Post condition	Student views their meal consumption history	

Table18.1: Description of Use Case View Meal History

Definition of a Sequence Diagram:

A Sequence Diagram is a UML behavioral diagram used to model the dynamic behavior of a system by showing the sequence of interactions between actors and system components to accomplish a particular process. It visually represents the flow of messages in chronological order, starting from the initiation of the interaction to the final response. The diagram highlights how different system elements collaborate over time and is essential for understanding system logic, validating functional requirements, and guiding developers during implementation.

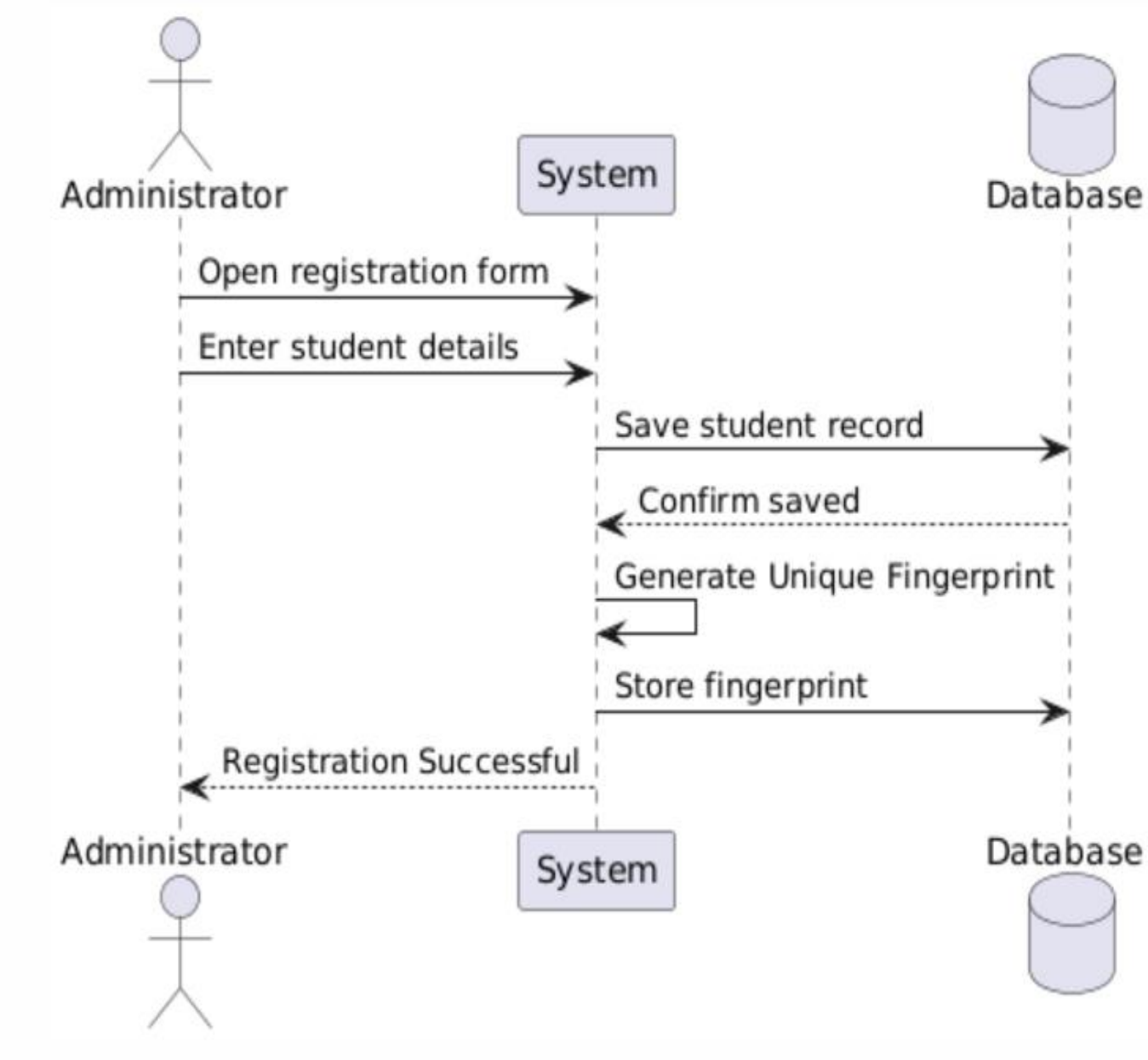

figure3.1:Student registration process sequence diagram

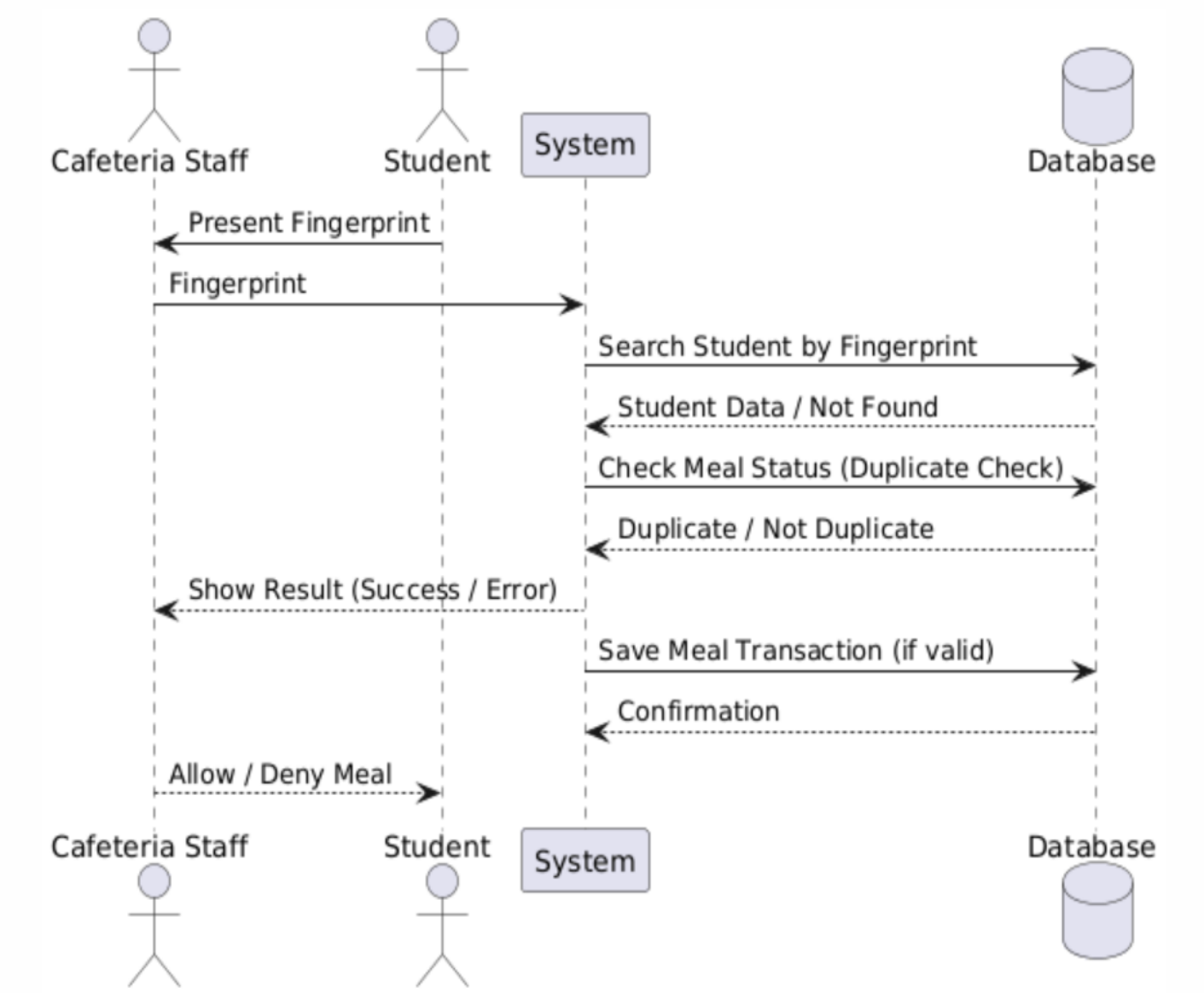

figure4.1: Meal Verification Process

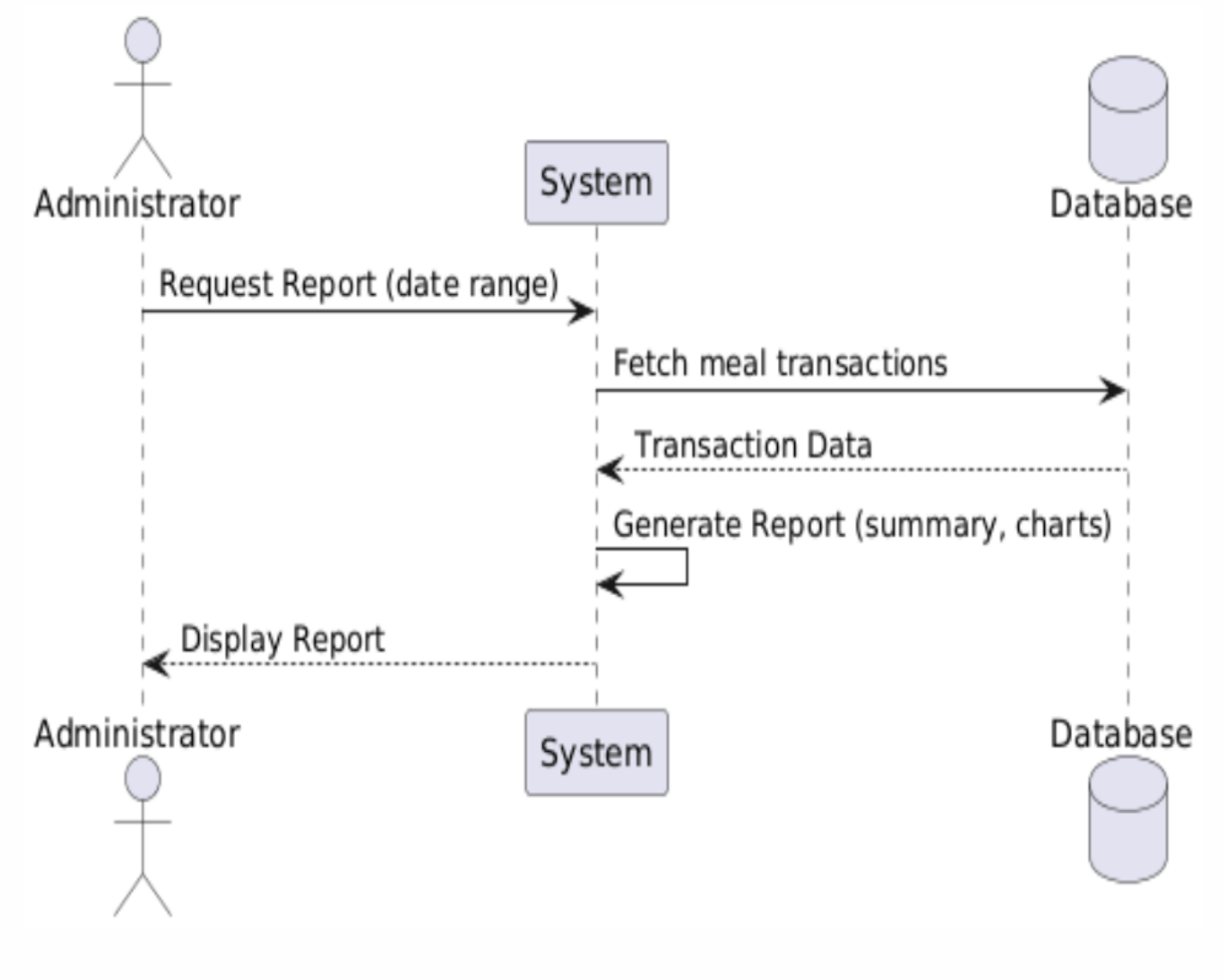

figure5.1: Report Generation

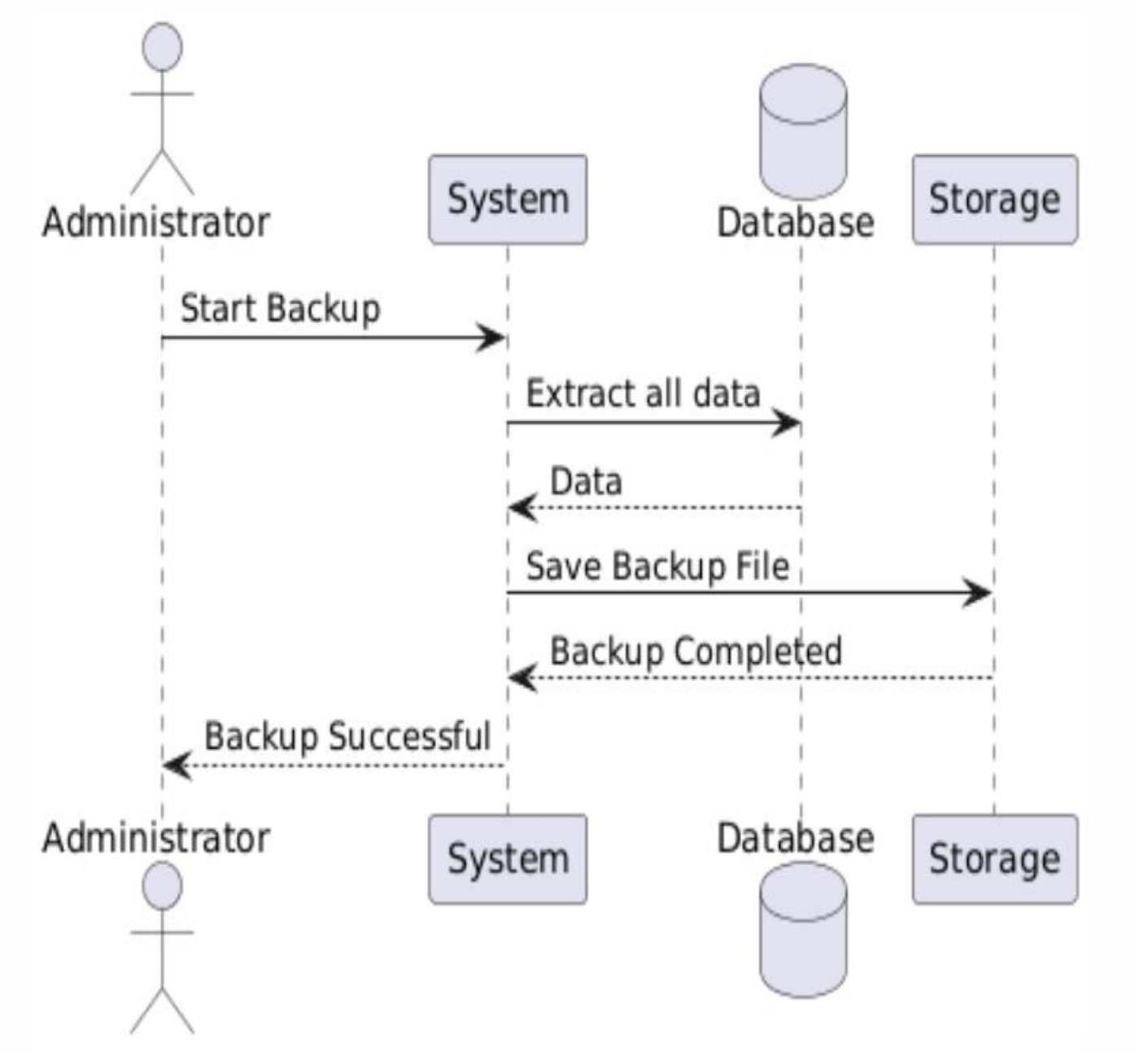

figure6.1: System Backup Process

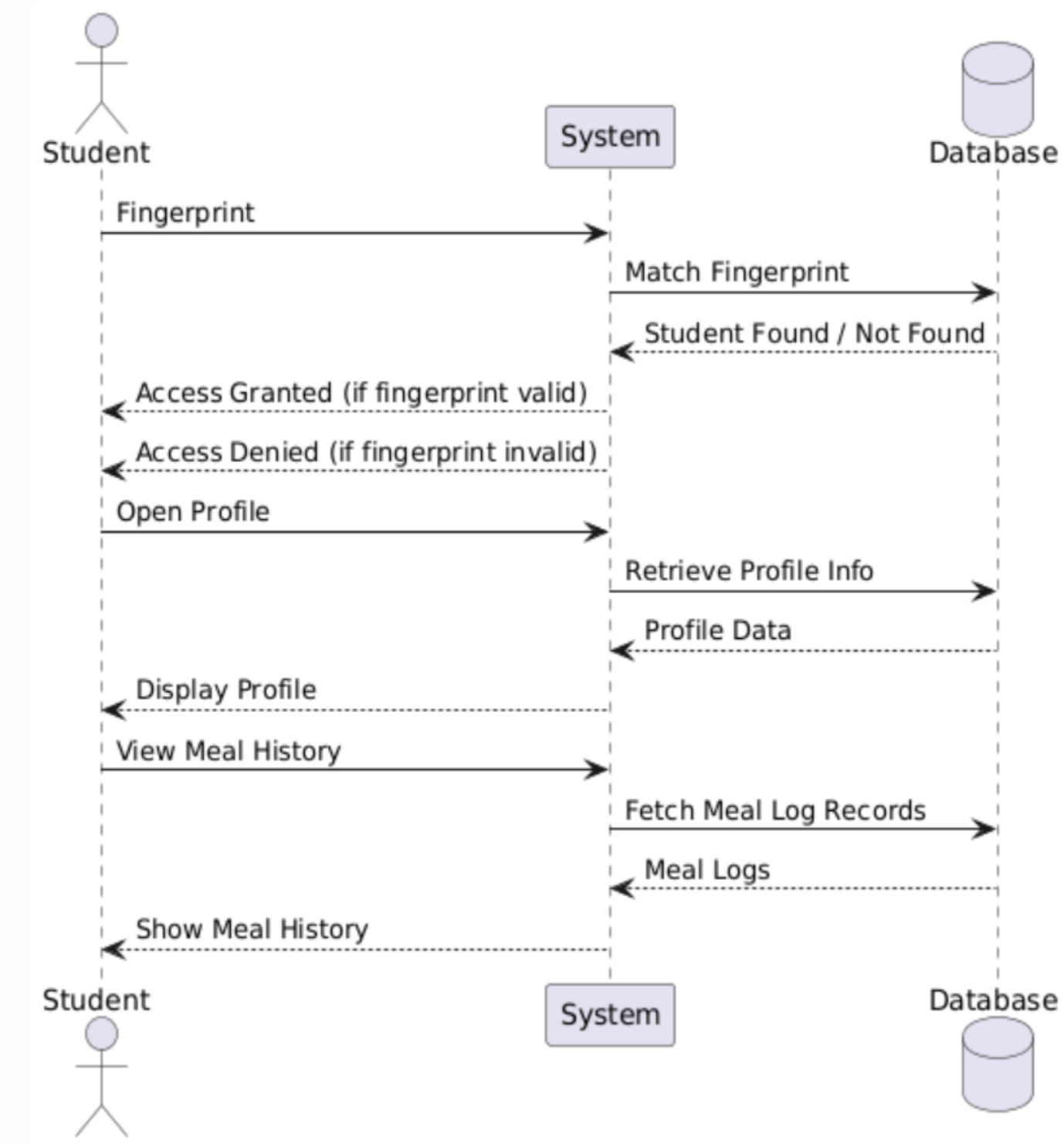

figure7.1: Student Portal Access

Definition of an Activity Diagram

An Activity Diagram is a UML behavioral diagram that illustrates the sequential and logical flow of activities within a specific process or system operation. It models the dynamic workflow by showing actions, decisions, parallel processes, start and end points, and transitions between activities. Activity diagrams help analysts and developers understand complex operational processes, validate functional requirements, and design efficient system workflows. They are widely used to represent processes such as student registration, meal verification, report generation, and system backup in the cafeteria management system.

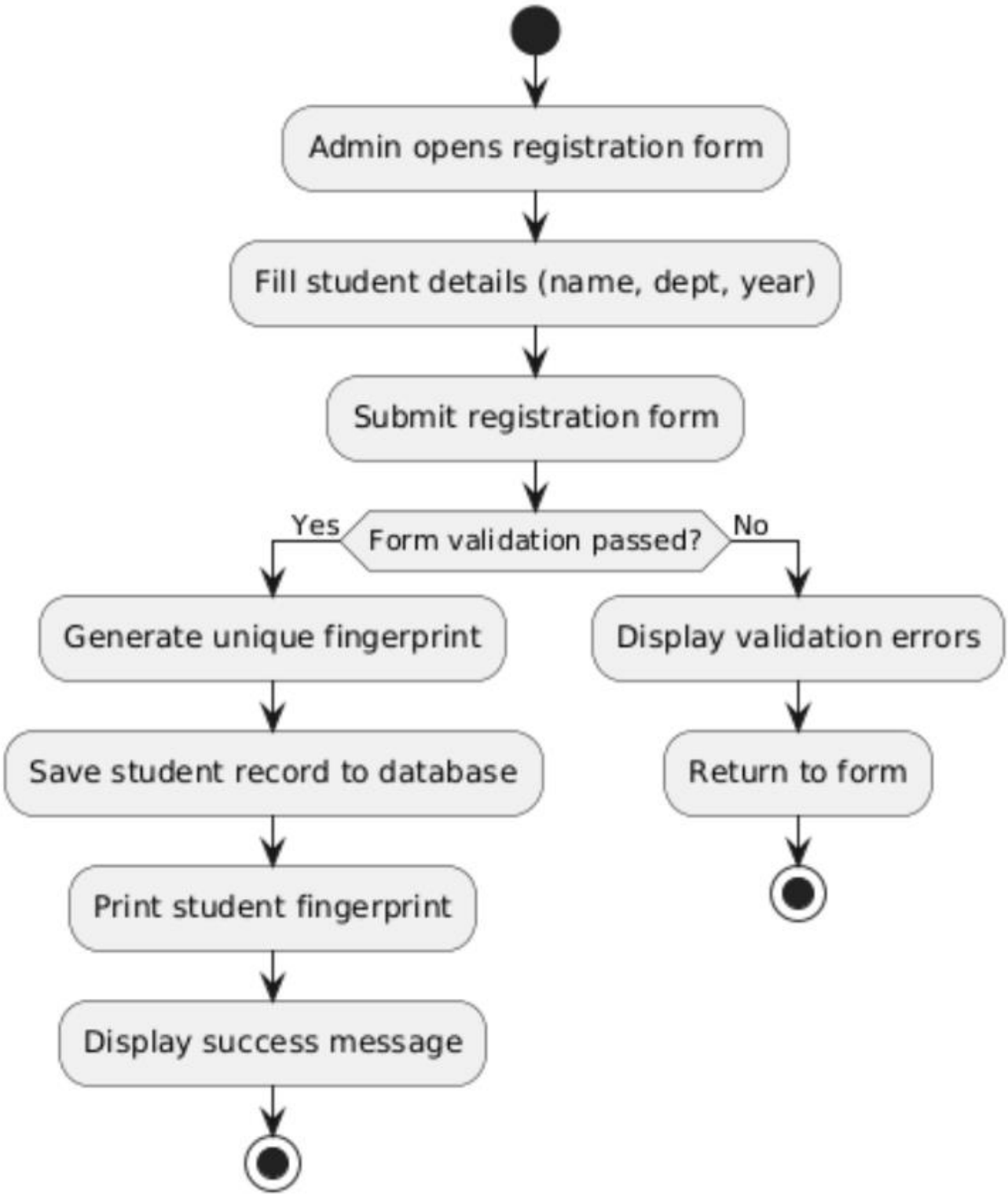

Figure8.1: Student Registration Activity Diagram

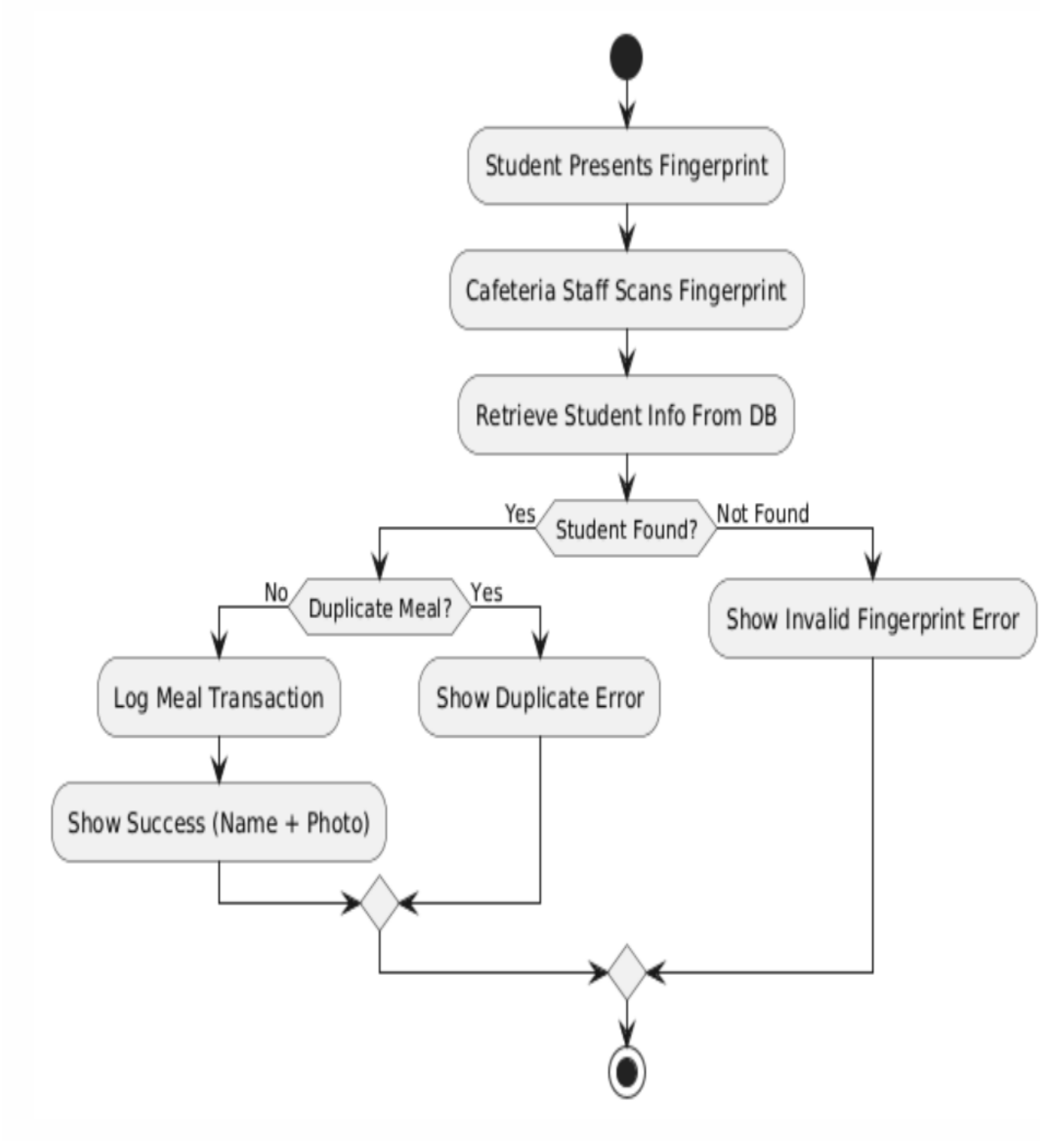

figure9.1: Meal Verification Activity Diagram

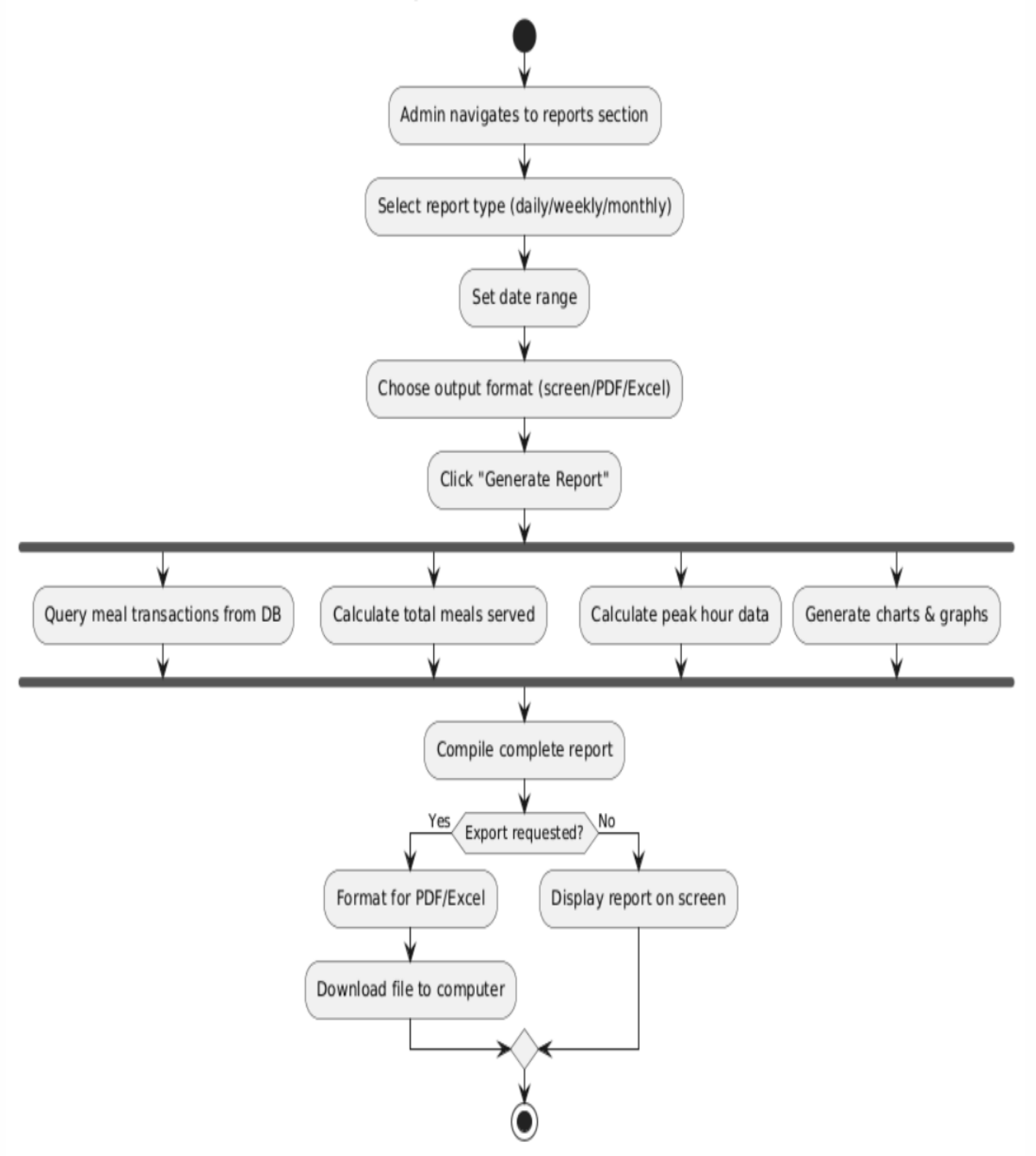

Figure10.1: Report Generation Activity Diagram

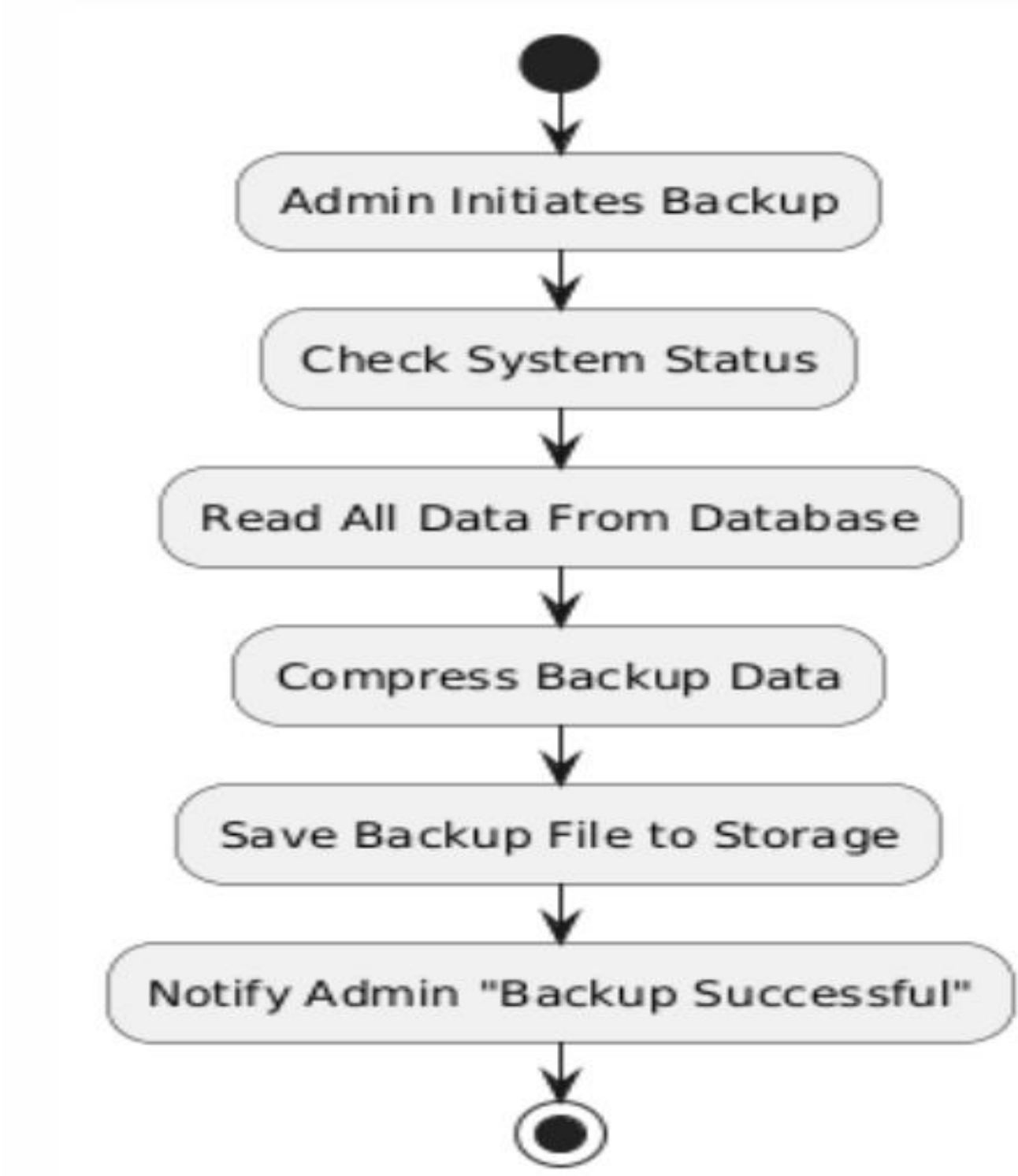

figure11.1: Activity Diagram – System Backup

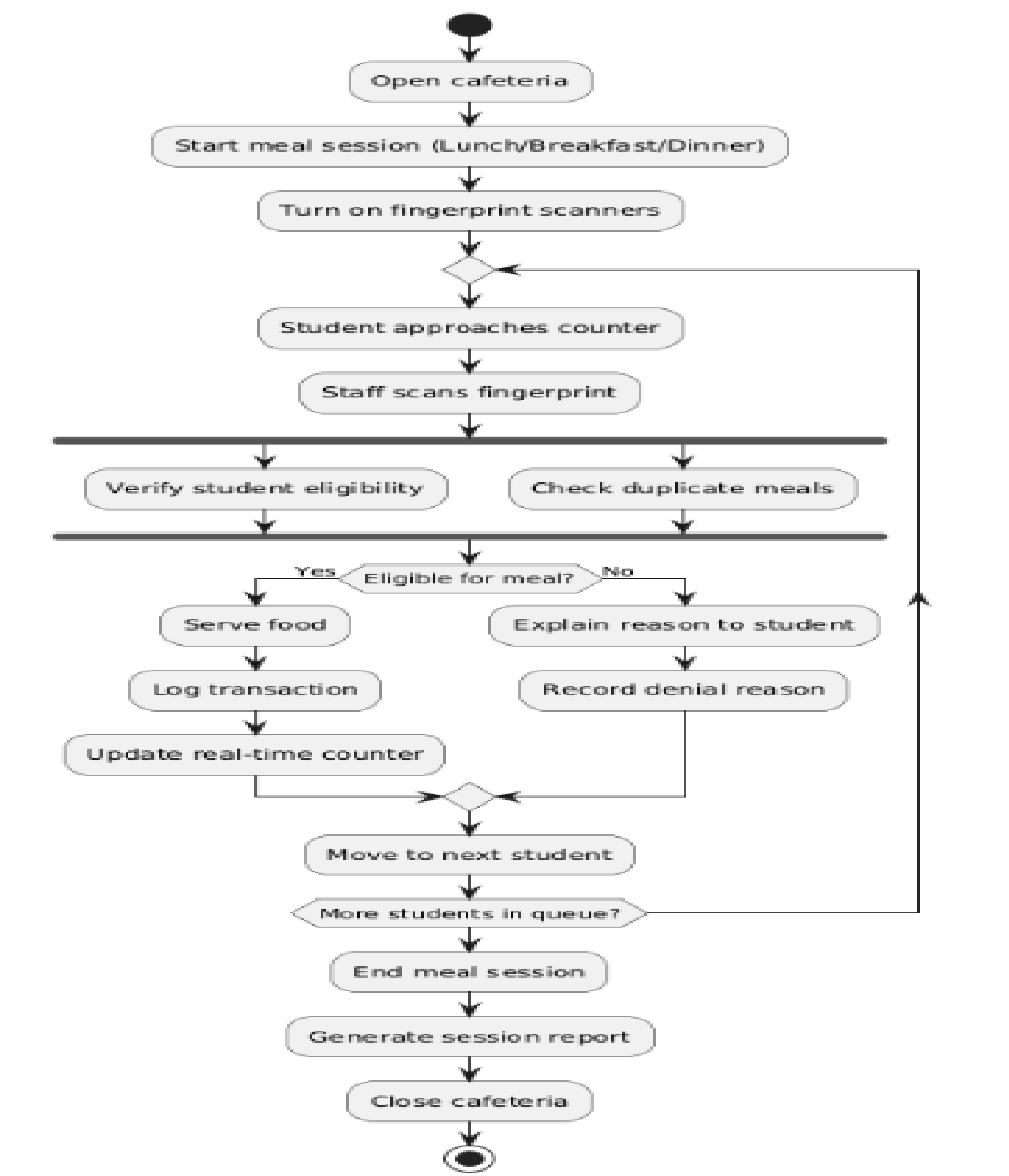

figure12.1: Daily Cafeteria Operations Activity Diagram

Definition of a Class Diagram

A Class Diagram is a UML structural diagram that describes the static architecture of a system by modeling its classes, their attributes, operations, and the relationships among them. It serves as the foundational blueprint for system design, defining how data is organized and how different system components interact structurally. Class diagrams support developers in building a consistent and maintainable software architecture and are essential for database design, object-oriented programming, and system implementation. (10)

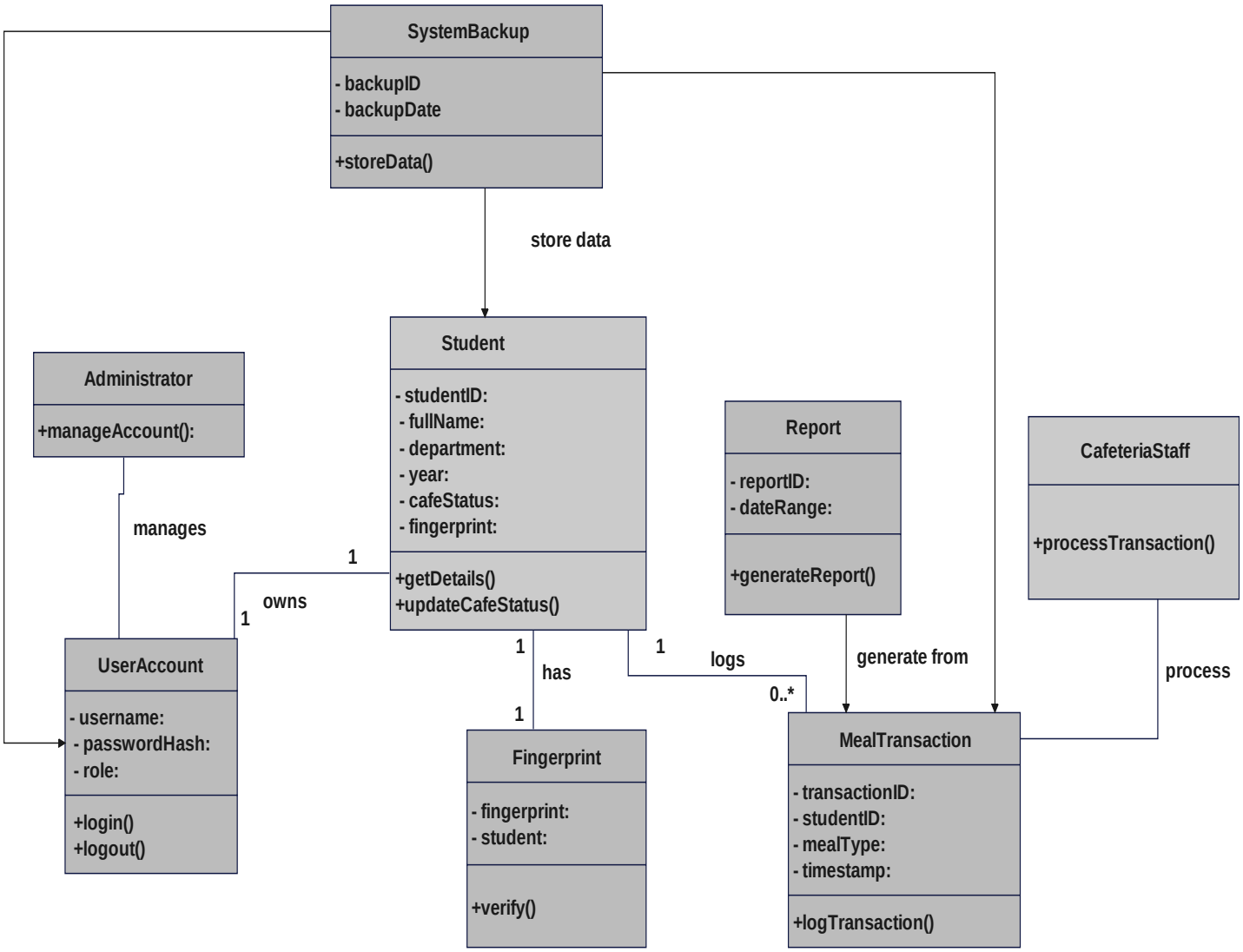

Figure13.1: class diagram

Conclusion

The design and development of the *Web-Based Automated Cafeteria Management System for Jinka University* represents a significant advancement toward modernizing and improving the efficiency of cafeteria operations within the university. Throughout this documentation, the major challenges of the existing system including unauthorized ID usage, frequent scanning errors, lack of real-time reporting, long queues, and inaccurate manual record-keeping were thoroughly analyzed and addressed using appropriate software engineering principles and a structured methodology.

By replacing the old ID-based system with a secure, fingerprint-based verification mechanism, the proposed system eliminates identity fraud, reduces duplicate meal attempts, and ensures fair, transparent service delivery for all students. The integration of automated fingerprint generation, real-time meal verification, role-based access control, and a comprehensive administrative dashboard introduces a level of operational accuracy and reliability that the previous system could not provide.

The use of UML diagrams including use case diagrams, sequence diagrams, activity diagrams, and class diagrams ensured a clear understanding of system requirements and interactions before implementation. The Incremental Model supported progressive development, continuous stakeholder feedback, and improved adaptability throughout the project lifecycle. The chosen technology stack (HTML, CSS, JavaScript, PHP, My**No table of figures entries found.**SQL) and affordable hardware components (USB fingerprint scanners) made the solution both technically feasible and economically sustainable for Jinka University.

Reference

1. **Author), Jinka University (Corporate.** *About Jinka University.* s.l. : Jinka University Official Website, 2024. N/A (Webpage).

2. **Jinka University.** *Jinka University Institutional Profile 2023/2024*.* s.l. : Jinka University Press, 2024. 5.

3. **Directorate, Student Services.** *Annual Report on Student Welfare Services.* 2 14, 2025.

4. **Sommerville, I.** *Software engineering.* s.l. : Pearson Education, 2023. N/A (Complete book).

5. **Pressman, R. S., & Maxim, B. R.** *Software engineering: A practitioner's approach.* s.l. : McGraw-Hill Education, 2020. N/A (Complete book).

6. **Author), W3Schools (Corporate.** PHP tutorial. [Online] 4 8, 2015.

7. **Connolly, T. M., & Begg, C. E.** A practical approach to design, implementation, and management. s.l. : Pearson, 2020.

8. **student.** *about of jinka university student cafeteria management system.* 2 25, 2025.

9. **Group, Object Management.** Unified Modeling Language . [Online] 2017.

10. **Fowler, M.** *A brief guide to the standard object modeling language.* s.l. : Addison-Wesley, 2004.