

ADDIS ABABA UNIVERSITY

SCHOOL OF INFORMATION SCIENCE

**PROJECT TITLE:** FBE CAMPUS CAFTERIA MANAGEMENT SYSTEM

**Group members Id**

Afework Mulugeta UGR/8435/16

Beamlak Amare UGR/4446/16

Doii Hailu UGR/7039/16

Fahmi Ahmed UGR/9284/16

Honelign Yohannes UGR/1424/16

Natinael Boda UGR/1426/16

Sophonyas Bewuketu UGR/3280/16

Jan 26, 2025

Acknowledgement

We would like to extend our heartfelt gratitude to everyone who has supported us since the inception of this cafeteria project. First and foremost, we thank God for guiding us and blessing this journey. We would also like to extend our heartfelt appreciation to Addis Ababa University, FBE Campus, for providing us with the resources and opportunity to carry out this project.

We would like to extend our heartfelt appreciation to our mentor, Mr. Aminu Mohamed, for his valuable advice, support, and constructive criticism throughout the project. His knowledge and insight were of great importance in shaping our ideas and making the project a success.

We would also like to thank our peers and team members for their relentless effort, support, and teamwork in bringing our vision into reality.

Contents

[Chapter 1 Introduction 4](#_Toc188831587)

[Background 4](#_Toc188831588)

[Purpose of the system 4](#_Toc188831589)

[Statements of the problem 5](#_Toc188831590)

[Scope of the project 6](#_Toc188831591)

[Objective of the project 7](#_Toc188831592)

[General objective 7](#_Toc188831593)

[Specific objectives 7](#_Toc188831594)

[Database Development Methodology 8](#_Toc188831595)

[Data Sources & Collection Methods 9](#_Toc188831596)

[DB Analysis and Design Methods 10](#_Toc188831597)

[Deliverables of the Project 11](#_Toc188831598)

[Development Tools, Platforms and Technologies 13](#_Toc188831599)

[Development Tools 13](#_Toc188831600)

[Platforms 14](#_Toc188831601)

[Technologies 15](#_Toc188831602)

[Project Time Plan 15](#_Toc188831603)

[Chapter Two Requirement Specification 22](#_Toc188831604)

[2.1. Data Requirements 22](#_Toc188831605)

[2.2. Transaction Requirements 22](#_Toc188831606)

[Chapter 3 Database Design 23](#_Toc188831607)

[3.1 Conceptual database design 23](#_Toc188831608)

[3.1.1 Entities 23](#_Toc188831609)

[3.1.2 Attributes 24](#_Toc188831610)

[3.1.3 Relationships Between Entities 25](#_Toc188831611)

[3.1.4 Entity Relationship diagram 26](#_Toc188831612)

[3.2. Logical Database Design 27](#_Toc188831613)

[3.2.1. ER-Relation Mapping 27](#_Toc188831614)

[3.2.2. Normalization 27](#_Toc188831615)

[3.2.3 Relational Schema with referential integrity after normalization 31](#_Toc188831616)

[3.3 Physical database design 31](#_Toc188831617)

[3.3.1 Physical database strategy 31](#_Toc188831618)

[3.3.2 Database deployment details 33](#_Toc188831619)

[Chapter 4 Implementation and Testing 36](#_Toc188831620)

[4.1. SQL script for creating database 36](#_Toc188831621)

[4.2. SQL Scripts for creating the tables, view, indexes 36](#_Toc188831622)

[4.3. Testing 41](#_Toc188831623)

# Chapter 1 Introduction

## Background

The FBE-campus cafeteria plays a pivotal role in providing students with easy access to meals. One of the main functions of the university is to provide meal services that meet the needs of its students. However, with the rise in the number of users of the cafeteria, traditional and manual cafeteria operations often fall short in terms of efficiency, accuracy, and integrity. Food services as one of the major function and responsibilities of the university shall be given a bigger concern.

Traditional paper-based recognition method has been used across various processes. However, these methods are open to inefficiencies, including:

* Risk of damage, forgery, loss.
* Time-consuming manual verification processes.
* Environmental concerns due to excessive paper usage.
* High costs associated with printing and distribution.

With the growing acceptance and integration of technology in campus facilities, the need for a digitalized and modernized cafeteria management system has to be essential. This system will integrate technology and database management system to enhance operational efficiency, provide better service, and improve students’ satisfaction while reducing overhead costs.

In an ever-changing digital world, the adoption of modern and efficient recognition methods is essential; QR codes have emerged as a reliable, cost-effective, and flexible solution, capable of addressing these challenges while improving students’ experience.

## Purpose of the system

The purpose to develop this project is to replace the current traditional system by providing faster, accurate, and efficient system. With this new system, it can eliminate some problems such as:

* Long lines of Students for getting their meal cards also for their foods.
* Feeding of non-Authorized Students on the Campus Cafeteria.
* Errors on reporting the daily records to determine the number of Students who have taken their meal.
* Slow manual ticking of Students on papers.

The purpose of transitioning to a **QR code-based recognition system** is to modernize and streamline operations by leveraging digital technology. The proposed system aims to replace traditional paper-based methods with QR codes to achieve the following objectives:

1. **Efficient Verification**: QR codes can be scanned quickly using smartphones or dedicated scanners, enabling instant recognition and verification
2. **Enhanced Security**: QR codes can be dynamically generated and encrypted, reducing the risk of forgery and unauthorized duplication.
3. **Environmental Friendliness**: Reducing or eliminating paper usage aligns with environmental sustainability goals and minimizes waste.
4. **Cost Savings**: Lower costs associated with printing, distribution, and physical storage of paper-based materials.
5. **Convenience and Accessibility**: QR codes can be distributed digitally via email, apps, or websites, making them accessible anywhere and reducing logistical complexities.

**Real-Time Data Integration**: QR codes can be linked to backend systems, allowing seamless integration with databases for attendance tracking, payments, or certificate validation.

## Statements of the problem

In the current cafeteria system, there are large problems associated with manual record keeping for the meal card and attendance in the cafeteria. This old-fashioned concept allows the students to easily exploit the system by using other students meal card which leads to inequitable distribution of resources and costs the cafeteria a lot of money. Further, with manual preparation of records of meal card numbers, paperwork gets unmanageable and bulky creating a problem while dealing with the attendance records, meal usage statistics and reports. The process is quite tiring which causes congestion during rush meal periods. Worsening these problems, there is no way for the student to record any special diets or allergies; this is a major health hazard because a

student may get served a meal that he was allergic to. Moreover, it eliminates the chance for cafeteria staff to analyze consumption habits or inefficiencies of the cafeteria without extensive reporting methods.

Through QR code-based Database management information system, the preceding problems related to the cafeteria management process will be solved by using some automation and information security continuously in a proper manner. The system can automatically record participants’ attendance by assigning each meal card its own unique QR code linked to the card, thus reducing and or eliminating delays and errors associated with the manual processes. It helps avoid some of meal card fraud since students’ ids are linked uniquely with the QR code to enable them access the cafeteria services. Traditional paper-based records on attendance and meal list are less practical because they consume a lot of space and can easily be misarranged or even lost while the electronic version makes it easier to organize, search and evaluate data for efficient decision making. Furthermore, persons benefit from a safer environment due to an added option to input their dietary and allergy preferences that the system can verify against the menu to avoid dangerous food combinations. By introducing these features, the system eases working processes, cuts down on doing work by hand, and increases the standards of food services for the learners.

## Scope of the project

This project focuses on the development and implementation of a QR code-based database management system for the FBE cafeteria. The scope encompasses solving key operational challenges, such as automating attendance tracking and ensuring secure and fair usage of meal cards. By integrating QR code technology, the system will enhance efficiency, minimize manual effort, and improve overall cafeteria operations. It will also include features for managing dietary restrictions and allergies to prioritize student safety.

The project will address the following aspects:

**Designing a relational database** to store and organize data related to students, staff, meals, QR codes, and attendance.

**Automating attendance tracking** by replacing the manual paper-based process with QR code scanning linked to the database. Students will scan their unique QR codes to validate their attendance in real-time.

**Preventing meal card misuse** by associating each QR code with a specific student ID, ensuring that only authorized students can access cafeteria services.

**Introducing a menu management system** to log daily meal options and integrate them with students' dietary preferences and allergy information.

**Providing tools for reporting and analysis** to generate insights on attendance patterns, meal consumption trends, and operational inefficiencies.

## Objective of the project

### General objective

Develop and implement a QR-code based DBMS that improves the security, accuracy and

efficieny of meal tracking at AAU

### Specific objectives

1. Facilitate Menu Management

Develop a feature-rich menu management module that allows cafeteria staff to update daily menus, display available food items, and include prices in real-time. This ensures transparency and reduces confusion among customers.

1. Enhance Order Processing

Introduce an intuitive order management system that supports both in-person and online orders, reducing wait times and ensuring accuracy. The system should include functionalities for tracking order statuses and notifying customers upon completion.

1. Improve Inventory Control

Build an inventory management system to monitor stock levels of raw materials and supplies. This will help prevent shortages or overstocking and facilitate timely reordering.

1. Enable Payment Flexibility

Integrate multiple payment options, including cash, card, and digital wallets, to provide convenience and cater to diverse user preferences.

1. Generate Detailed Reports

Create a reporting module to generate detailed analytics on sales, inventory usage, and peak dining hours. These insights will assist administrators in decision-making and forecasting demand.

1. Ensure User Accessibility and Security

Implement user-friendly interfaces for both customers and staff, ensuring accessibility for non- technical users. Additionally, robust security measures will protect user data and prevent unauthorized access.

1. Minimize Wastage

Introduce features that track food production versus consumption to minimize food wastage and promote sustainable operations.

1. Support Scalability

Design the system to accommodate future expansions, such as adding new cafeterias or integrating advanced technologies like AI-powered recommendations or IoT-based inventory tracking.

By achieving these objectives, the proposed cafeteria management system will not only address the current inefficiencies but also lay the foundation for a modernized, customer-centric dining ecosystem.

## Database Development Methodology

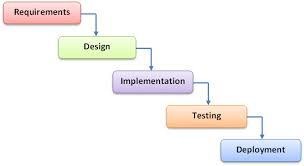
In developing the AAU FBE Campus Cafeteria Management System, we will adhere to a

structured approach to ensure the efficiency, accuracy, and relevance of the database system. The methodology adopted for this project will combine elements of Agile and Waterfall methodologies to allow for iterative development while maintaining a clear plan and

requirements framework.



Methodology Highlights

1. Agile Elements: Iterative development cycles, frequent stakeholder feedback, and adaptability to changing requirements.
2. Waterfall Elements: Defined project phases, clear milestones, and structured documentation for each stage of development.

### Data Sources & Collection Methods

To populate the database for the AAU FBE Campus Cafeteria Management System, we will employ a combination of data collection methods to ensure comprehensive and accurate information retrieval.

Data collection methodologies

1. Interviews:

Conducting interviews with cafeteria staff, management, and students to gather insights on current processes, requirements, and expectations for the new system.

1. Observation:

Observing the daily operations within the cafeteria, including order processing, inventory management, and customer interactions, to identify data points and workflow patterns.

1. Existing Data Sources:

Utilizing data from the current cafeteria management system, including transaction records, menu items, inventory levels, to inform the design and implementation of the new database.

By combining insights gathered through interviews, observations, and existing data sources, we aim to create a robust and user-centric database management system for the AAU FBE Campus Cafeteria. This approach will ensure that the system not only meets the operational needs of the cafeteria but also enhances efficiency, customer experience, and overall management

effectiveness.

### DB Analysis and Design Methods

#### DB analysis methods

1. Requirement Gathering:

Interviews will be carried out with the cafeteria staff, management, and students to extract information on the current way of working and the expectations about the new system.

Surveys and questionnaires would be used to collect data related to user needs and preferences.

1. Use Case Analysis:

Identify and document the use cases for different types of users, such as students, staff, and administrators.

Define how users will interact with the system to capture all functional requirements.

1. Entity-Relationship Modeling:

Use ERDs to represent the data entities visually that will take part in the problem, including students, meals, orders, and their relationships.

Identify primary keys and foreign keys to ensure data integrity and establish relationships between tables.

1. Data Flow Diagrams:

Elaborate on data flow diagrams to depict the flow of data within the system.

Identify processes, data stores, and data sources to understand how information will be processed and utilized.

1. Normalization:

Apply normalization techniques to avoid data redundancy and efficiently organize the data.

Design the database tables in accordance with the rules of normalization (1NF, 2NF, 3NF) for better integrity of data.

#### DB design techniques

1. Logical Design:

Transform the conceptual model (ERD) into a logical form that SQL Server can understand. Schema definition: define tables, columns, data types, and constraints.

1. Physical Design:

Design the physical storage of the database, including indexing strategies to optimize query performance.

Determine the hardware and software requirements for the SQL Server environment.

1. Security Design:

Implement user roles and permissions based on user needs to ensure data security and access control.

Plan for data encryption and backup strategies to protect sensitive information.

1. Testing and Validation:

Design test cases to validate database functionalities with respect to requirements. Test the performance and reliability of the system using sample data.

1. Documentation:

Always have proper documentation regarding the whole design process, including but not limited to data dictionaries, schema designs, user guides.

Ensuring Documentation is clear, readable and would always be easy to access with respect to future reference/maintenance.

## Deliverables of the Project

1. Project Documentation:

A comprehensive project report on system requirements, objectives, methodologies, and design decisions.

Documentation of interviews, questionnaires, and data collection methods used.

1. Database Schema:

A fully designed and implemented database schema with all tables, relationships, and constraints. ERD and data flow diagrams showing how data is organized and flows.

1. User Interface Design:

User Interface: prototypes or wireframes for the cafeteria management system. User experience and accessibility considerations that are to be documented.

1. Functional System:

A functional cafeteria management system implemented in SQL Server Management Studio. The modules will include menu management, order processing, inventory control, and reporting.

1. Testing Reports:

Documentation of test processes: test cases, results, and problems detected.

Summary of fixed bugs and system optimizations made based on the results of testing.

1. Training Materials:

User Manuals and Training Guides for cafeteria staff and students.

Training conducted in order for the users to get familiar with the new system.

1. Final Presentation:

A PowerPoint presentation summarizing the project outcomes, including the features, benefits, and future expansion possibilities of the system.

Visual aids and demonstrations of how the new system is efficient and provides improvements.

1. Maintenance Plan:

Document describing the maintenance procedures to be followed, such as regular backups, updates, and user support.

Future strategies for enhancement and scalability of the system.

## Development Tools, Platforms and Technologies

### Development Tools

**SQL Server Management Studio (SSMS):**

This is used for design the database structure: creating tables, relationships, indexes. Writing and testing SQL queries for data entry, retrieval, and updates. Database security management and user permissions management.

* + Purpose: This is used to design, manage, and interact with the SQL Server database, create tables, write SQL queries, and optimize performance.
  + Selection Reason: It is user-friendly in interface for database management and integrated with SQL Server to develop databases efficiently.

**Microsoft Word:**

It helps to document all the designing and development processes of the database system. Helps write project reports, including a system requirement, database structure, objectives, and results. Helps maintain documentation throughout in order for all steps to be well-recorded.

* + Purpose: To document the project, including the system design, functionality, and objectives.
  + Reason for Choice: This is a widely used tool for writing project reports and documentation.

**Microsoft PowerPoint**:

This is utilized for making presentations to communicate the progress and final outcomes of the project. It helps in presenting the database design, functionalities,

and how the system solves the problems of the cafeteria in a clear and professional manner

* + Purpose: To create presentations that describe the project to the class.
  + Reason for Selection: Offers professional templates and is widely used for presentations in academic and business settings.

**Gantt Chart:**

This is used for project planning and tracking. It helps in visualizing the project timeline, managing tasks, and ensuring that milestones are met on schedule. It is a useful tool for monitoring the progress of the project and keeping the development process on track.

* + - Purpose: For the planning and tracking of project tasks, timelines, and progress.
* Reason for Selection: A visual tool used to manage project deadlines, making sure all tasks are done in due time.

### This may contain: the logo for microsoft's new server 2008 program, which is now available on windowsPlatforms

**SQL Server :**

This is the management system to be used to effectively create, manage, and store all cafeteria-related information and data such as inventory and sales records, and also a record of the students in the cafeteria.

* + Purpose: Main platform where the cafeteria information will be stored, managed, and queried.
  + Reason for Selection: This is scalable database management system which provides the ability to support complex queries with large datasets.

**Browsers:**

* (Internet Explorer, Mozilla Firefox, Google Chrome…)

### Technologies

**Relational Database Technology:**

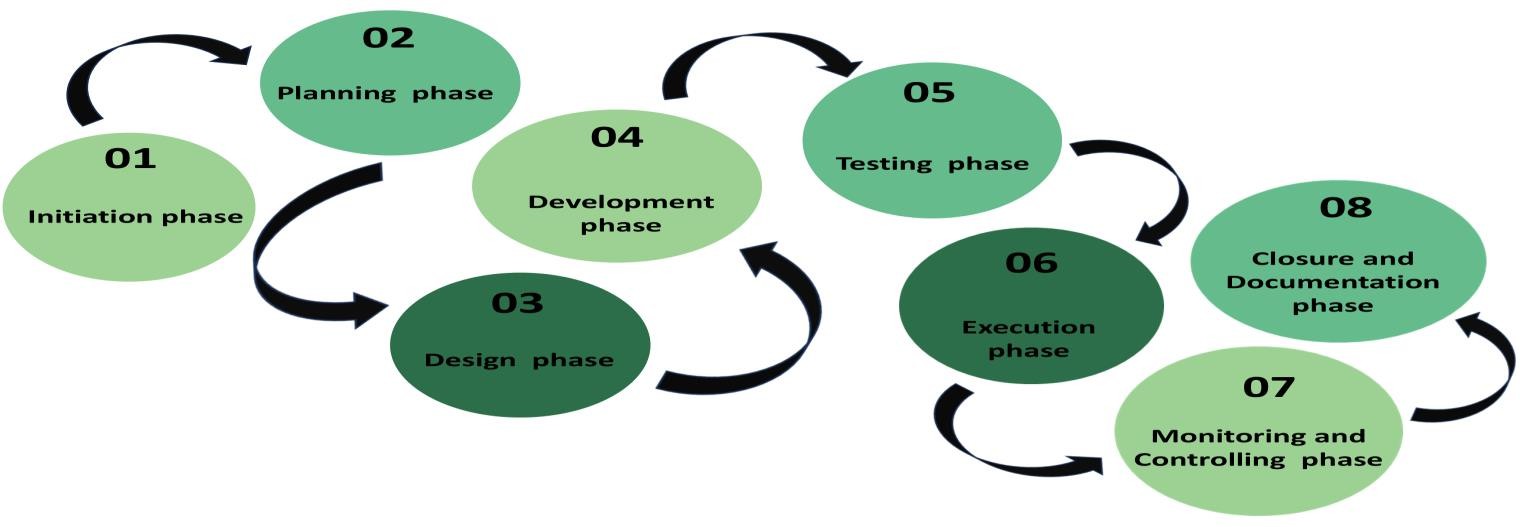
This involves structuring data in tables and managing the relationships between these tables, using SQL Server.

Reason for Selection: It provides for efficient data organization, ensuring the consistency, security, and integrity of the data.

## Project Time Plan

**General Overview**

The timeline, activities, milestones and outputs for the effective introduction of the “AAU FBE Campus Cafeteria Management System” are described in the project time plan. The plan guarantees the successful execution of all activities in the allotted time and place and that the project’s goals are ensured to be achieved. In order to overcome the age-old issues associated with cafeteria service efficiency, customer satisfaction and management in the Campus Cafeteria, it is of our greatest need to employ eight (8) core phases to conduct the project with clarity, structure, scalability and flexibility.



1. Initiation phase:

Objective:

* + Define the prime scope, requirements and initial research for the system. Tasks undertaken:
  + Identify key problems or opportunities that require immediate solution.
  + Conduct a research or study to evaluate the feasibility or viability of the project.
  + Outline objectives and outcomes.
  + Identify key personnels and their roles.
  + Secure approvals and resources necessary for the initiation of the project. Duration:
  + 1week Deadline:
  + December 5, 2024 Participating team members:
  + Afework Mulugeta
  + Natinael Boda
  + Sophonyas Bewuketu

1. Planning phase:

Objective:

* + Define the prime scope, requirements and initial research for the system. Tasks undertaken:
  + Document and understand the cafeteria’s current systems, points and processes.
  + Determine the resources and tools that are required.
  + Finalize on the lists of tools and requirements.
  + Create a detailed and concise project plan.
  + Develop project timeline and schedule. Duration:
  + 1week Deadline:
  + December 12, 2024 Participating team members:
  + Afework Mulugeta (Lead)
  + Beamlak Amare
  + Fahmi Ahmed

1. Design phase:

Objective:

* + Develop a detailed database schema and system design in accordance with the project at hand. Tasks undertaken:
  + Develop the database schema.
    - Gather data that needs to be stored.
    - Identify relationships.
    - Identify entities and attributes.
    - Define primary key.
    - Create relationships with entities.
    - Draw entity- relationship diagram.
    - Include constraints.
    - Write SQL statements to create the tables and relationships.
    - Implement, test and optimize the schema.
  + Create the user interface of the QR-based system. Duration:
  + 11 days Deadline:
  + December 23, 2024 Participating team members:
  + Honelign Yohannes
  + Fahmi Ahmed
  + Natinael Boda
  + Doii Hailu

1. Development phase:

Objective:

* + Build QR code and integrate it with the database system created. Tasks undertaken:
  + Implement the integrated database and connect the system to support the QR code system.
  + Develop a unique set of QR code generation for each user and the scanning module.
  + Develop a user-friendly interface and integrate it with the system. Duration:
  + 2weeks Deadline:
  + January 9, 2025 Participating team members:
  + Honelign Yohannes
  + Sophonyas Bewuketu
  + Beamiak Amare
  + Doii Hailu
  + Natinael Boda

1. Testing phase:

Objective:

* + Put the system under use and ensure the system functions as intended and fix any issues encountered.

Tasks undertaken:

* + Conduct system testing both with and without the user to ensure the system works at all times.
  + After testing, identify any bugs and fix any inefficiencies.
  + Ensure data security, use and reliability. Duration:
  + 2weeks Deadline:
  + January 23, 2025 Participating team members:
  + Fahmi Ahmed
  + Doii Hailu
  + Natinael Boda

1. Execution:

Objective:

* + Implement the developed system at the AAU FBE campus cafeteria. Tasks undertaken:
  + Successfully install the system on devices.
  + Give training to staff and cafeteria users on using the system.
  + Prepare user manuals for staff and users.
  + Monitor the progress of the system.
  + Manage the project team.

Duration:

* + 2 days Deadline:
  + January 24, 2025 Participating team members:
  + Afework Mulugeta
  + Beamiak Amare
  + Sophonyas Bewuketu

1. Monitoring and Controlling phase:

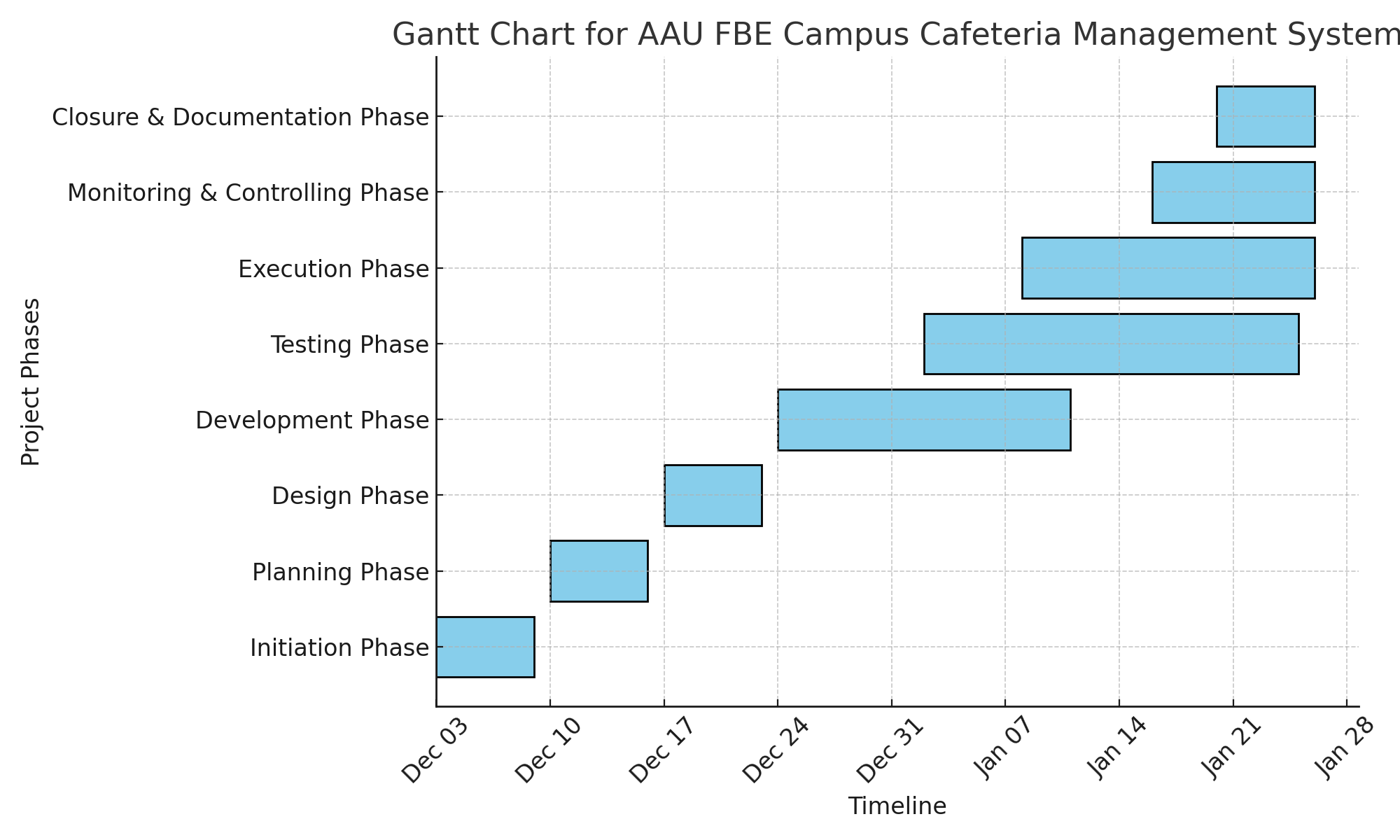
Objective:

* + Ensuring the system works as it should still align with its goals. Tasks undertaken:
  + Track project performance using tools eg. Gnatt charts.
  + Track resource usage and wastage.
  + Identify any gaps between the systems that hinders the satisfaction of the user.
  + Ensure quality control by conducting timely meetings. Duration:
  + 1 day Deadline:
  + January 25, 2025 Participating team members:
  + Honelign Yohannes
  + Fahmi Ahmed
  + Beamiak Amare

1. Closure and Documentation:

Objective:

* + Document data for later system development and maintenance. Tasks undertaken:
  + Create effective ways of collecting, processing and storing data in a form of technical document for system development and maintenance.
  + Conduct project review and analysis.
  + Document lessons learned. Duration:
  + 1 day Deadline:
  + January 26, 2025 Participating team members:
  + Afework Mulugeta
  + Doii Hailu
  + Sophonyas Bewuketu



# Chapter Two Requirement Specification

## 2.1. Data Requirements

The database system must handle the following types of data:

1. **Student Data**

* Full name, student ID, meal card number, QR code, gender, department,
* Data integrity and validation to ensure only registered students are included.

1. **Attendance Data**

* Meal attendance records: timestamps, meal types (breakfast, lunch, dinner), and QR code scans.

1. **Menu Data**
   * Daily menu details: dish names, ingredients, and meal types.
   * Information on potential allergens must also be stored.
2. **Staff Data**
   * Staff information for cafeteria operations (e.g., who records attendance or manages reports).
3. **Reports and Logs**
   * System-generated reports for meal usage trends, attendance patterns, and allergy conflicts.

## 2.2. Transaction Requirements

These are divided into four main categories:

**1. Data Entry Requirements**

* Students must be registered into the database with details like name, ID, meal card number, and QR code.
* Staff can input daily menu information, including ingredients and meal types.
* Attendance records must be automatically entered upon scanning a student’s QR code.

**2. Data Retrieval Requirements**

* Retrieve student details when scanning a QR code to validate meal access.
* Allow cafeteria staff to access daily menus and ingredient details quickly.
* Generate reports such as:
  + Daily attendance summary.
  + Meal consumption trends.
  + Allergy conflicts for specific students.

**3. Data Updating Requirements**

* Update student information (e.g., dietary restrictions, lost meal cards, or QR code reassignment).
* Modify menu details for a specific day if there are changes.
* Update attendance records in case of scanning errors.

**4. Data Removal Requirements**

* Remove outdated menu records after a defined retention period.
* Delete or archive data for graduated students.
* Remove invalid QR code records due to system updates or replacements.

# Chapter 3 Database Design

## 3.1 Conceptual database design

### 3.1.1 Entities

1. **Student**
   * Represents students who use the cafeteria services.
   * Each student is assigned a unique QR code for attendance tracking.
2. **Attendance**
   * Tracks meal usage by students.
   * Links students to specific meal times (breakfast, lunch, dinner).
3. **Menu**
   * Stores information about the daily meals served at the cafeteria.
   * Includes details about ingredients and potential allergens.
4. **Staff**
   * Represents cafeteria staff managing attendance and system operations.
5. **Allergy**
   * Tracks dietary restrictions and allergies for students.

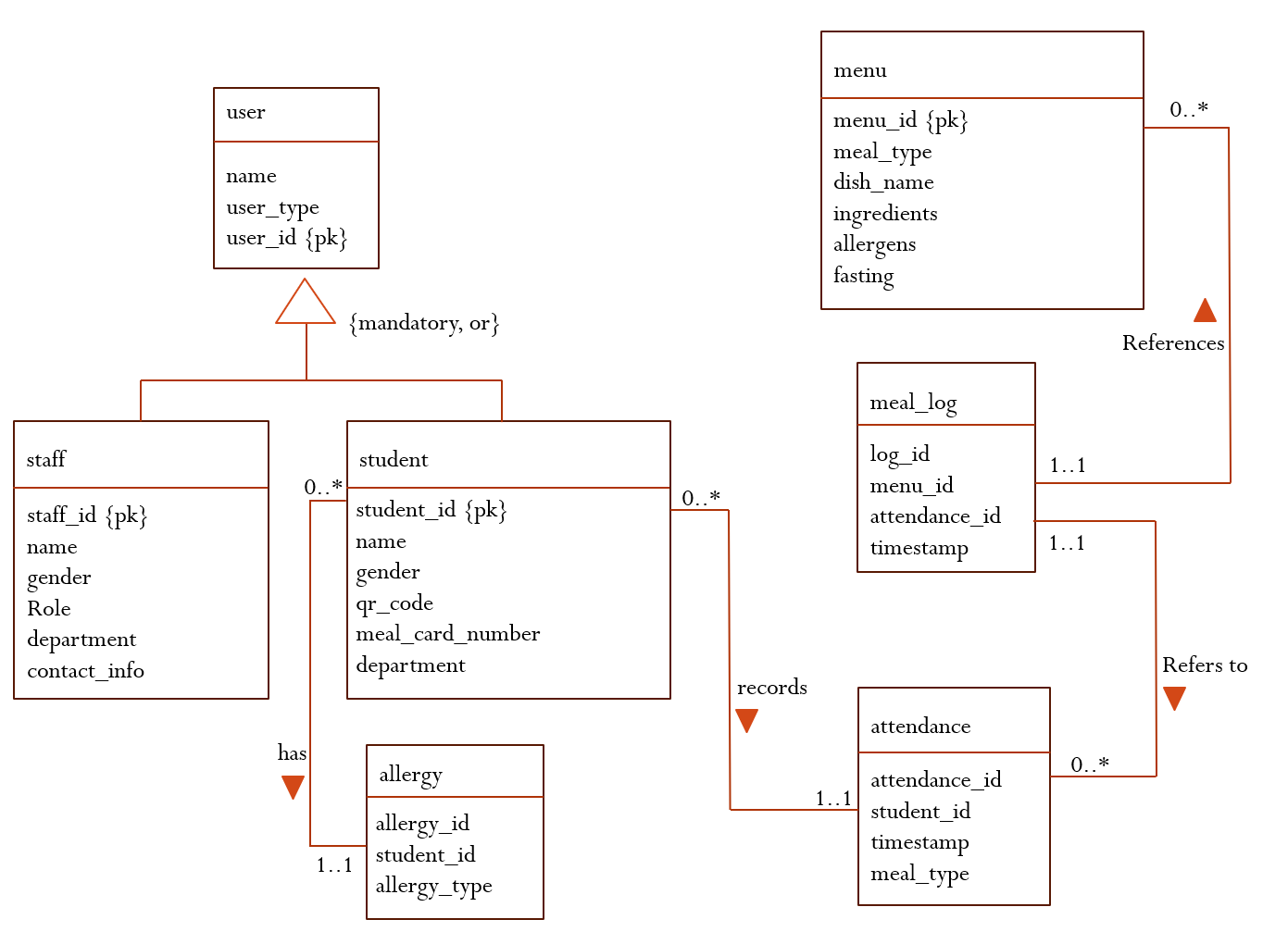
### 3.1.2 Attributes

1. **Student**
   * **Student\_ID**: Unique identifier for the student (primary key).
   * **Name**: Full name of the student.
   * **QR\_Code**: Unique QR code for meal access.
   * **Gender**: Gender of the student.
   * **Meal\_Card\_Number**: Meal card associated with the student.
   * **Department:** Department of the student.
2. **Attendance**
   * **Attendance\_ID**: Unique identifier for the record (primary key).
   * **Student\_ID**: Foreign key linking to the student.
   * **Timestamp**: Date and time the QR code was scanned.
   * **Meal\_Type**: Type of meal accessed (breakfast, lunch, dinner).
3. **Menu**
   * **Menu\_ID**: Unique identifier for the menu (primary key).
   * **Meal\_Type**: Type of meal (breakfast, lunch, dinner).
   * **Dish\_Name**: Name of the dish.
   * **Ingredients**: List of ingredients used in the dish.
   * **Allergens**: List of potential allergens in the dish.
4. **Staff**
   * **Staff\_ID**: Unique identifier for the staff member (primary key).
   * **Name**: Full name of the staff member.
   * **Role**: Role in cafeteria management.
   * **Gender**: Gender of the staff member.
   * **Department**: Department of the staff member.
   * **Contact\_info**: contact information of the staff member
5. **Allergy**
   * **Allergy\_ID**: Unique identifier for the allergy record (primary key).
   * **Student\_ID**: Foreign key linking to the student.
   * **Allergy\_Type**: Type of allergy (e.g., nuts, gluten).

### 3.1.3 Relationships Between Entities

1. **Student → Attendance**
   * One-to-Many relationship: Each student can have multiple attendance records.
2. **Student → Allergy**
   * One-to-Many relationship: Each student can have multiple allergies.
3. **Menu → Attendance**
   * One-to-Many relationship: Each meal type in the menu can be linked to multiple attendance records.
4. **Staff → Attendance**
   * One-to-Many relationship: Each staff member can oversee multiple attendance records.

### 3.1.4 Entity Relationship diagram

****

## 3.2. Logical Database Design

### 3.2.1. ER-Relation Mapping

|  |  |
| --- | --- |
| **User** (user\_id, name, user\_type)  **Primary key** user\_id | **Menu** (menu\_id, meal\_type, dish\_name, fasting, ingredients, allergens)  **Primary key**: menu\_id |
| **Staff** (staff\_id, name, gender, role, department, contact\_info)  **Primary key** staff\_id | **Attendance** (attendance\_id, student\_id, timestamp, meal\_type)  **Primary key**: attendance\_id  **Foreign key**: student\_id references student(student\_id) |
| **Student** (student\_id, name, gender, role, department)  **Primary key** student\_id | **Meal\_log** (log\_id, menu\_id, attendance\_id, timestamp)  **Primary key** log\_id  **Alternate key** attendance\_id  **Foreign key** menu\_id references menu(menu\_id)  **Foreign key** attendance\_id references attendance(attendance\_id) |
| **Allergy** (allergy\_id, student\_id, allergy\_type)  **Primary key** allergy\_id  **Foreign key**: student\_id references student(student\_id) |  |

### 3.2.2. Normalization

**Normalization Process**

The normalization process is a technique used in database design to organize data in a way that reduces redundancy and prevents update anomalies. It involves a series of steps that transform data into relations that conform to specific normal forms

**First Normal Form (1NF)**

1. **Objective**:
   * Eliminate multivalued attributes and repeating groups.
   * Ensure that each column contains atomic values.

**Menu Table**:

* + - ingredients and allergens were multivalued attributes.
    - Split into two separate tables:
      * **Menu\_Ingredients**: Contains menu\_id and individual ingredient values.
      * **Menu\_Allergens**: Contains menu\_id and individual allergen values.

**Before 1NF**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| menu\_id | meal\_type | dish\_name | Ingredients | Allergens | Fasting |
| 1 | Breakfast | Misir | Injera, Bread, Onions, oil, pepper, salt, tomato, Misir | Misir, Tomato | No |

**After 1NF**:

* + - **Menu Table**:

|  |  |  |  |
| --- | --- | --- | --- |
| Menu\_id | Meal\_type | Dish\_name | Fasting |
| 1 | Lunch | Misir | No |

* + - **Menu\_Ingredients Table**:

|  |  |
| --- | --- |
| Menu\_Id | Ingredient |
| 1 | Injera |
| 1 | onion |
| 1 | Bread |
| 1 | pepper |
| 1 | Salt |
| 1 | Misir |
| 1 | Tomato |
| 1 | Oil |

* + - **Menu\_Allergens Table**:

|  |  |
| --- | --- |
| Menu\_id | Allergen |
| 1 | Tomato |
| 1 | Misir |

**Second Normal Form (2NF)**

1. **Objective**:
   * Eliminate partial dependencies (non-primary attributes must depend on the whole primary key).
   * **Meal\_Log Table**:
     + timestamp depended only on attendance\_id and not on the composite key 1
     + Moved timestamp to **Attendance**.

**Before 2NF**:

* **Meal\_log Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Log\_id | Menu\_id | Attendance\_id | timestamp |
| 1 | 1 | 301 | 2024-10-30 12:00 |

**After 2NF**:

* + - **Meal\_Log Table**:

|  |  |  |
| --- | --- | --- |
| Log\_ig | Menu\_id | Attendance\_id |
| 1 | 1 | 301 |

* + - **Attendance Table**:

|  |  |  |  |
| --- | --- | --- | --- |
| Attendance\_id | Student\_id | Meal\_type | timestamp |
| 301 | 201 | Lunch | 2024-10-30 12:00 |

**Third Normal Form (3NF)**

1. **Objective**:
   * Eliminate transitive dependencies (non-primary attributes should not depend on other non-primary attributes).
   * **Student Table**:
     + meal\_card\_number depended on qr\_code instead of the primary key (student\_id).
     + Moved meal\_card\_number to a new **Meal\_Card** table.

**Before 3NF**:

|  |  |  |  |
| --- | --- | --- | --- |
| Student\_id | Name | Qr\_code | Meal\_card\_number |
| 201 | Abebe | QR123 | 3213 |

**After 3NF**:

|  |  |  |  |
| --- | --- | --- | --- |
| Student\_id | Name | Qr\_code | Department |
| 201 | Abebe | QR123 | IT |

* + - **Student Table**:

|  |  |  |  |
| --- | --- | --- | --- |
| Student\_id | Name | Qr\_code | Department |
| 201 | Abebe | QR123 | IT |

* + - **Meal\_Card Table**:

|  |  |
| --- | --- |
| QR\_code | Meal\_card\_number |
| QR123 | 3213 |

### 3.2.3 Relational Schema with referential integrity after normalization

1. **User Table**:

(user\_id, name, user\_type)

1. **Staff Table**:

(staff\_id, name, gender, role, department, contact\_info,user\_id)

1. **Student Table**:

(student\_id, name, gender, qr\_code, department, user\_id)

1. **Meal\_Card Table**:

(qr\_code, meal\_card\_number)

1. **Menu Table**:

(menu\_id, meal\_type, dish\_name, fasting)

1. **Menu\_Ingredients Table**:

(menu\_id, ingredient)

1. **Menu\_Allergens Table**:

(menu\_id, allergen)

1. **Allergy Table**:

(allergy\_id, student\_id, allergy\_type)

1. **Attendance Table**:

(attendance\_id, student\_id, meal\_type, timestamp)

1. **Meal\_Log Table**:

(log\_id, menu\_id, attendance\_id)

## 3.3 Physical database design

### 3.3.1 Physical database strategy

Physical database design represents a vital phase in the creation of a database specifically tailored to fulfill the unique requirements of a given application. For example, when developing a cafeteria database system for the FBE campus of Addis Ababa University, the primary emphasis is placed on the efficient storage and access of data. This design phase necessitates critical decisions regarding the organization of tables, the types of data each table will house, and the interrelations among these tables. Implementing strategies such as normalization is crucial, as it aids in the removal of redundant data, thus ensuring the database operates at peak efficiency. Additionally, the team must evaluate various indexing methods that can greatly improve data retrieval speed, ultimately enhancing the overall performance of the system.

Our group engaged in close collaboration during the planning phase to effectively implement the physical database design. Each member brought their distinct skills and viewpoints, which enhances the analysis of the cafeteria’s operational requirements. The discussions encompass a broad range of topics, including the various transactions that will occur, such as food ordering, inventory management, and customer feedback handling, all of which are essential for determining the optimal organization of the data. Additionally, the group prioritizes the assessment of security measures to protect sensitive information, including payment details and customer personal data. This proactive security approach is crucial in the current digital landscape, where data breaches can lead to significant repercussions.

By cultivating a cooperative atmosphere and exchanging their perspectives, it was able to develop a strong and scalable database that not only facilitates the operations of the cafeteria but also accommodates future requirements. This thorough design process guarantees that the database remains reliable and efficient, even as transaction volumes and data grow over time. Ultimately, the effective physical design of the database will enhance service delivery in the cafeteria, thereby improving the experience for both students and staff. Through meticulous planning and collaboration, the team is prepared to create a solution that meets and surpasses the expectations of the FBE campus community.

On a more concise manner, here are some key strategies for the physical database design:

* **Normalization**: Facilitates efficient data storage by removing duplicate information and structuring the database into organized tables. This process enhances data consistency and minimizes the risk of anomalies during data operations.
* **Indexing Methods**: Enhances the efficiency of data retrieval by implementing structures like B-trees or hash indexes, which facilitate faster searches and lookups within the database.
* **Assessment of Security Measures**: Focuses on safeguarding sensitive information, including payment details and personal customer data, to avert data breaches and ensure adherence to privacy regulations.
* **Scalability Planning**: Constructs the database to accommodate growing transaction volumes and larger datasets as the system expands, thereby maintaining consistent performance over time.
* **Data Partitioning**: This method partitions extensive tables into smaller, more manageable sections, either along horizontal or vertical lines, to improve performance and facilitate maintenance. Such a strategy is especially advantageous for handling significant transaction volumes.
* **Backup and Recovery Planning**: Establishes automated backup protocols and recovery plans to safeguard against data loss caused by system malfunctions or human mistakes, thereby ensuring the continuity of business operations during emergencies.
* **Optimization of Query Performance**: Evaluates and refines frequently executed queries, utilizing tools such as EXPLAIN plans in SQL, to decrease execution time and enhance system responsiveness.
* **Monitoring and Performance Tuning**: Employs database monitoring tools to identify performance issues and implement necessary adjustments.

### 3.3.2 Database deployment details

Establishing a database for the cafeteria system at the FBE campus of Addis Ababa University requires a series of essential steps to guarantee its effectiveness and dependability. Initially, the process will commence with the configuration of the database server, preferably utilizing a cloud-based solution such as AWS or Azure to take advantage of scalability and backup options. Subsequently, a detailed design of the database schema is necessary to support various entities, including menu items, orders, customers, and staff. The next phase will involve implementation, which includes the creation of tables, defining relationships, and setting constraints using a reliable database management system (DBMS) like MySQL. A role-based access control (RBAC) system will be established to enhance security measures by limiting access according to the specific roles assigned to users. Additionally, thorough testing will be conducted to confirm data integrity, enhance performance, and guarantee smooth integration with the cafeteria’s system and user interface. Continuous maintenance and updates will be scheduled to ensure the efficient operation of the database system.. This collaborative initiative will ensure that the database fulfills the operational requirements of the cafeteria and improves the overall user experience.

Deployment details for the Cafeteria Database System at the FBE Campus include the following:

* **Database Server Configuration**

Select a cloud-based platform to guarantee scalability, high availability, and robust backup options. Configure the database server with sufficient computing power and storage capacity, automated backups conducted regularly to facilitate disaster recovery, monitoring and logging systems to oversee server performance and identify potential issues and establish secure access to the server through the implementation of firewalls and encrypted connections.

* **Database Schema Design**

Establish the schema to encompass the following entities:

- Menu Items: Information such as item name, description, price, and availability status.

- Orders: Components including order ID, timestamp, customer ID, item details, and current status.

- Customers: Data points comprising customer ID, name, contact details, and preferences.

- Staff: Information including staff ID, roles, contact details, and work schedules.

Create relationships among the tables.

For instance, a one-to-many relationship exists between Orders and Menu Items.

Implement constraints to maintain data integrity.

* **Implementation**

Utilize a dependable Database Management System (DBMS) such as MySQL, or SQL Server. Create tables, indexes, views, and stored procedures as required. Specify triggers to automate processes, including inventory updates following an order.

* **Testing**

Conduct thorough testing to guarantee data integrity to verify the accuracy of Create, Read, Update, and Delete (CRUD) operations, to assess query execution times across different workloads, to confirm smooth interaction with the cafeteria’s user interface and any associated systems and to execute penetration testing and audits to uncover potential vulnerabilities.

* **Integration with Front-End System**

Create APIs to facilitate communication between the front-end and mobile applications to ensure that the API endpoints are both secure and optimized for optimal performance.

* **Monitoring and Maintenance**

Utilize monitoring solutions to assess database performance.

Establish a routine for database maintenance that encompasses:

- Optimization of indexes.

- Removal of obsolete records.

- Strategic planning for resource scaling.

Develop a schedule for regular software updates and enhancements of features.

* **Documentation and Training**

Develop comprehensive user and technical documentation, which encompasses backup and recovery protocols. Provide training for cafeteria personnel and administrators to ensure they can utilize the system efficiently.

* **User Input and Ongoing Enhancement**

Gather insights from users, including both customers and employees, to pinpoint areas for enhancement. Introduce modifications gradually, ensuring that operational continuity is maintained with minimal disruption.

The cafeteria database system’s efficiency and adaptability are further enhanced by several additional considerations. Data analytics offer valuable insights into sales and inventory management, while mobile compatibility facilitates smooth ordering processes and timely notifications. Scalability equips the system to accommodate future growth, and adherence to data protection regulations guarantees responsible management of data. Collectively, these elements improve user experience and ensure the system remains relevant in the future.

# Chapter 4 Implementation and Testing

## 4.1. SQL script for creating database

CREATE DATABASE CafeterialMealManagement;

GO

USE CafeterialMealManagement;

GO

## 4.2. SQL Scripts for creating the tables, view, indexes

CREATE TABLE Users (

user\_id INT IDENTITY(1,1) PRIMARY KEY,

name VARCHAR(100) NOT NULL,

user\_type VARCHAR(10) CHECK (user\_type IN ('Staff', 'Student')) NOT NULL

);

CREATE TABLE Staff (

staff\_id INT IDENTITY(1,1) PRIMARY KEY,

user\_id INT NOT NULL,

name VARCHAR(100) NOT NULL,

gender VARCHAR(6) CHECK (gender IN ('Male', 'Female')) NOT NULL,

role VARCHAR(50) NOT NULL,

department VARCHAR(50) NOT NULL,

contact\_info VARCHAR(150),

FOREIGN KEY (user\_id) REFERENCES Users(user\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE Student (

student\_id VARCHAR(11) PRIMARY KEY,

user\_id INT NOT NULL,

name VARCHAR(100) NOT NULL,

gender VARCHAR(6) CHECK (gender IN ('Male', 'Female')) NOT NULL,

qr\_code VARCHAR(100) UNIQUE NOT NULL,

department VARCHAR(50) NOT NULL,

FOREIGN KEY (user\_id) REFERENCES Users(user\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE Meal\_Card (

qr\_code VARCHAR(100) PRIMARY KEY,

meal\_card\_number CHAR(4) NOT NULL,

FOREIGN KEY (qr\_code) REFERENCES Student(qr\_code) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE Menu (

menu\_id INT IDENTITY(1,1) PRIMARY KEY,

meal\_type VARCHAR(10) CHECK (meal\_type IN ('Breakfast', 'Lunch', 'Dinner')) NOT NULL,

dish\_name VARCHAR(100) NOT NULL,

fasting Bit NOT NULL

);

CREATE TABLE Menu\_Ingredients (

menu\_id INT NOT NULL,

ingredient VARCHAR(100) NOT NULL,

PRIMARY KEY (menu\_id, ingredient),

FOREIGN KEY (menu\_id) REFERENCES Menu(menu\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE Menu\_Allergens (

menu\_id INT NOT NULL,

allergen VARCHAR(100) NOT NULL,

PRIMARY KEY (menu\_id, allergen),

FOREIGN KEY (menu\_id) REFERENCES Menu(menu\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE Allergy (

allergy\_id INT IDENTITY(1,1) PRIMARY KEY,

student\_id VARCHAR(11) NOT NULL,

allergy\_type VARCHAR(100) NOT NULL,

FOREIGN KEY (student\_id) REFERENCES Student(student\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE Attendance (

attendance\_id INT IDENTITY(1,1) PRIMARY KEY,

student\_id VARCHAR(11) NOT NULL,

meal\_type VARCHAR(10) CHECK (meal\_type IN ('Breakfast', 'Lunch', 'Dinner')) NOT NULL,

timestamp DATETIME NOT NULL,

FOREIGN KEY (student\_id) REFERENCES Student(student\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE Meal\_Log (

log\_id INT IDENTITY(1,1) PRIMARY KEY,

menu\_id INT NOT NULL,

attendance\_id INT NOT NULL,

FOREIGN KEY (menu\_id) REFERENCES Menu(menu\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (attendance\_id) REFERENCES Attendance(attendance\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

-- View to list all students with their meal card information

CREATE VIEW vw\_StudentMealCard AS

SELECT

s.student\_id,

s.name AS student\_name,

s.department,

m.qr\_code,

m.meal\_card\_number

FROM

Student s

LEFT JOIN

Meal\_Card m

ON

s.qr\_code = m.qr\_code;

-- View to list all menu items with their ingredients and allergens

CREATE VIEW vw\_MenuDetails AS

SELECT

m.menu\_id,

m.meal\_type,

m.dish\_name,

m.fasting,

mi.ingredient,

ma.allergen

FROM

Menu m

LEFT JOIN

Menu\_Ingredients mi ON m.menu\_id = mi.menu\_id

LEFT JOIN

Menu\_Allergens ma ON m.menu\_id = ma.menu\_id;

-- View to get student allergies and matching menu allergens

CREATE VIEW vw\_StudentAllergies AS

SELECT

s.student\_id,

s.name AS student\_name,

s.department,

a.allergy\_type,

ma.allergen,

CASE

WHEN a.allergy\_type = ma.allergen THEN 'Match'

ELSE 'No Match'

END AS allergy\_match\_status

FROM

Allergy a

INNER JOIN

Student s ON a.student\_id = s.student\_id

LEFT JOIN

Menu\_Allergens ma ON a.allergy\_type = ma.allergen;

-- View to list attendance with meal log details

CREATE VIEW vw\_AttendanceLog AS

SELECT

a.attendance\_id,

a.student\_id,

s.name AS student\_name,

a.meal\_type,

a.timestamp,

ml.menu\_id,

m.dish\_name

FROM

Attendance a

INNER JOIN

Student s ON a.student\_id = s.student\_id

LEFT JOIN

Meal\_Log ml ON a.attendance\_id = ml.attendance\_id

LEFT JOIN

Menu m ON ml.menu\_id = m.menu\_id;

-- Index for faster lookup of students by their QR code

CREATE INDEX idx\_StudentQRCode ON Student (qr\_code);

-- Index for faster retrieval of menu items based on meal type

CREATE INDEX idx\_MenuMealType ON Menu (meal\_type);

-- Index for faster join operations on Student ID in the Allergy table

CREATE INDEX idx\_AllergyStudentID ON Allergy (student\_id);

-- Index for faster retrieval of attendance records by student and timestamp

CREATE INDEX idx\_AttendanceStudentTime ON Attendance (student\_id, timestamp);

-- Index for meal card lookup by QR code

CREATE INDEX idx\_MealCardQRCode ON Meal\_Card (qr\_code);

## 4.3. Testing

-- Insert users

INSERT INTO Users (name, user\_type)

VALUES

('Abebe Kebede', 'Student'),

('Tigist Alemu', 'Student'),

('Dereje Mekonnen', 'Staff'),

('Alemnesh Getachew', 'Staff');

-- Insert students

INSERT INTO Student (user\_id, student\_id, name, gender, qr\_code, department)

VALUES

(1, 'ugr/0001/16', 'Abebe Kebede', 'Male', 'QR001', 'Information Systems'),

(2, 'ugr/0002/16', 'Tigist Alemu', 'Female', 'QR002', 'Accounting');

-- Insert staff members

INSERT INTO Staff (user\_id, name, gender, role, department, contact\_info)

VALUES

(3, 'Dereje Mekonnen', 'Male', 'Cafeteria Manager', 'Cafeteria', 'dereje.mekonnen@university.et'),

(4, 'Alemnesh Getachew', 'Female', 'Cook', 'Cafeteria', 'alemnesh.getachew@university.et');

-- Link students with their meal cards

INSERT INTO Meal\_Card (qr\_code, meal\_card\_number)

VALUES

('QR001', 'MC01'),

('QR002', 'MC02');

-- Add some meals to the menu

INSERT INTO Menu (meal\_type, dish\_name, fasting)

VALUES

('Breakfast', 'Firfir with Injera', 0),

('Lunch', 'Shiro and Rice', 1),

('Dinner', 'Minchet', 0);

-- Add ingredients for meals

INSERT INTO Menu\_Ingredients (menu\_id, ingredient)

VALUES

(1, 'Injera'),

(1, 'Berbere'),

(2, 'Rice'),

(2, 'Shiro'),

(3, 'Meat'),

(3, 'Injera');

-- Add allergens for meals

INSERT INTO Menu\_Allergens (menu\_id, allergen)

VALUES

(1, 'Gluten'),

(2, 'Legumes'),

(3, 'Meat');

-- Add student allergies

INSERT INTO Allergy (student\_id, allergy\_type)

VALUES

('UGR/0001/16', 'Gluten'),

('UGR/0002/16', 'Legumes');

-- Log student attendance

INSERT INTO Attendance (student\_id, meal\_type, timestamp)

VALUES

('UGR/0001/16', 'Breakfast', GETDATE()),

('UGR/0002/16', 'Lunch', GETDATE());

-- Link attendance with menu

INSERT INTO Meal\_Log (menu\_id, attendance\_id)

VALUES

(1, 1), -- Abebe Kebede, Breakfast

(2, 2); -- Tigist Alemu, Lunch

--testing the queries

SELECT \* FROM vw\_StudentMealCard;

SELECT \* FROM vw\_MenuDetails;

SELECT \* FROM vw\_StudentAllergies;

SELECT \* FROM vw\_AttendanceLog;

# References

* 1. Connolly, T., & Begg, C. (2015). Database Systems: A Practical Approach to Design, Implementation, and Management (6th Edition). Pearson Education.
  2. TeamGantt. (n.d.). Create online Gantt charts. Retrieved from <https://www.teamgantt.com>
  3. Microsoft SQL Server Documentation. (n.d.). Retrieved from

<https://learn.microsoft.com/en-us/sql/?view=sql-server-ver16>

# Appendix

