UTM UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING
UTM Johor Bahru

SECD2523-03 DATABASE

Semester 01, 2025/2026

Phase 3

Lecturer:

HASLINA BINTI HASHIM

Group Members:

No	Name	Matric No
1	ABDURRAFIQ BIN ZAKARIA	A24CS0031
2	NAJMUDDIN BIN KAMARUDIN	A24CS0145
3	DANISH IZZ KHAN BIN AZMAN KHAN	A24CS0243
4	DANIEL IMAN HAQIMIE BIN YUSOFF	A24CS0063

TABLE OF CONTENTS

1.0 INTRODUCTION

The UTM Digital Cafeteria Management System is a strategic initiative developed to address critical operational challenges within the dining facilities at Universiti Teknologi Malaysia. Current issues, such as significant service delays, food wastage, and order inaccuracies, necessitate a transition from traditional manual workflows to a more robust, digitally integrated ecosystem. This project aims to modernize campus dining by implementing features such as online pre-ordering, intelligent inventory management, and real-time data analytics, thereby enhancing operational efficiency and the overall user experience for the university community.

In this Phase 3 of the project, the primary focus is directed toward a comprehensive Requirement Analysis and the definition of Transaction Requirements. This stage involves meticulously mapping the current business processes and identifying the specific data transactions necessary for system functionality, including data entry, updates, and complex queries for key stakeholders such as customers, staff, and administrators. Establishing these requirements is fundamental to ensuring that the resulting database architecture is capable of supporting all planned business logic with high reliability.

It should be noted that this document serves as a preliminary draft and represents an ongoing design process. At this stage, the content does not yet encompass database normalization, as the current objective is to finalize the functional requirements and external structures. Furthermore, the Conceptual Design provided is currently in a correction phase; it will be subject to further refinement in subsequent stages to ensure complete alignment with detailed university business rules and technical specifications.

2.0 OVERVIEW OF PROJECT

The UTM Digital Cafeteria Management System (DCMS) is a technological initiative designed to modernize campus dining by replacing inefficient manual workflows with an integrated digital platform. The core solution centers on a smart ordering platform that facilitates online pre-orders and digital queue management, effectively eliminating long wait times during peak hours. By synchronizing automated kitchen workflows with dynamic menus, the system ensures real-time updates on dish availability and dietary information, thereby significantly reducing customer disappointment and operational miscommunication.

The primary objectives of this project are to achieve a 50% reduction in peak-hour wait times and to optimize resource utilization through automated inventory tracking and cashless payment integration. Furthermore, the system aims to enhance institutional decision-making by providing administrators and staff with a sales analytics dashboard to monitor consumption trends and collect instant customer feedback. These goals collectively support UTM's broader digital transformation and sustainability initiatives by minimizing physical paper use and preventing unnecessary food waste.

The project scope encompasses the development of mobile and web-based ordering interfaces, real-time occupancy monitoring, and centralized kitchen display systems. However, it is important to note that this documentation currently serves as a preliminary design draft for Phase 3. Consequently, the Conceptual ERD is still in a correction stage, and comprehensive database normalization has not yet been applied. The boundaries of the project are strictly focused on internal cafeteria management and do not extend to external food delivery services or physical renovations of the cafeteria facilities.

3.0 DATABASE CONCEPTUAL DESIGN

3.1 Updated Business Rule.

3.1.1. User Hierarchy (Generalization)

- **User Specialization:** A User is the superclass identified by a unique UserID {PK} and includes common attributes such as UserName, UserContact, and UserPassword.
- **Subclasses:** Users are categorized into three mandatory and disjoint subclasses: Customer, Cafeteria Staff, and Admin.
- **Customer:** A Customer is a specific type of user who must provide a physical Address.
- Cafeteria Staff: Cafeteria Staff members are specialized users identified by their specific Role within the cafeteria.

3.1.2. Ordering and Processing

- **Placing Orders:** A Customer can place zero to many (0..*) Orders, but each Order must be placed by exactly one (1..1) Customer.
- **Processing Orders:** A Cafeteria Staff member can process zero to many (0..*) Orders, while each Order must be handled by exactly one (1..1) staff member.
- **Order Tracking:** An Order is uniquely identified by an OrderID {PK} and records the OrderDate, OrderTime, OrderStatus, and a TotalAmount.
- **Payment Details:** Every Order must store transaction information, specifically the PaymentMethod used and the current PaymentStatus.

3.1.3. Menu and Order Composition

- **Order Composition:** An Order must consist of at least one or more (1..*) OrderDetails.
- **Associative Link:** OrderDetails acts as a bridge between an Order and a MenuItem, tracking the specific OrderQty and Subtotal for each item included.
- **MenuItem Details:** A MenuItem is uniquely identified by a MenuID {PK} and includes the ItemName, Category, Price, and its current Availability.
- **Item Inclusion:** Each OrderDetails entry must include exactly one (1..1) MenuItem, whereas a MenuItem can be included in zero to many (0..*) order detail records.

3.1.4. Feedback System

- **Optional Feedback:** Providing feedback is optional for the Customer (0..1) and for the Order (0..1).
- **Feedback Identification:** Feedback is uniquely identified by a FeedbackID {PK} and contains a Rating, Comment, and the Date it was submitted.
- **Relationships:** Each Feedback entry is written by exactly one (1..1) Customer and is received by exactly one (1..1) Order.

3.2 Conceptual ERD

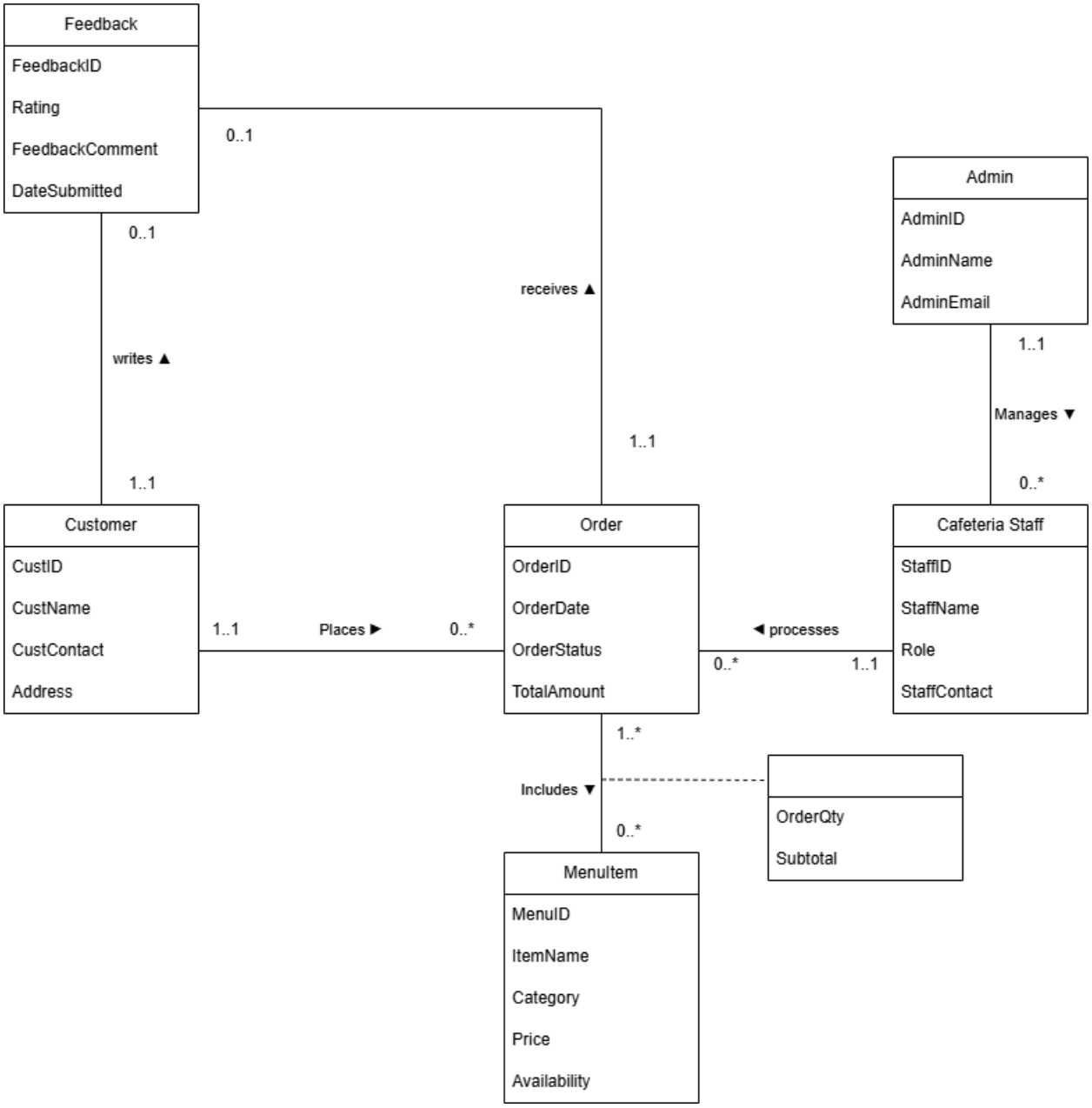

Figure 1.4 Conceptual Design

4.0 DB LOGICAL DESIGN

4.1 Logical ERD

1. **USER(<u>UserID</u>, Fname, Lname, UserContact)**

2. **CUSTOMER(<u>UserID</u>, Address, Street, City, Postcode, State)**
   FK: UserID references USER(UserID)

3. **ADMIN(<u>UserID</u>)**
   FK: UserID references USER(UserID)

4. **CAFETERIA_STAFF(<u>UserID</u>, Role, AdminID)**
   FK1: UserID references USER(UserID)
   FK2: AdminID references ADMIN(UserID)

5. **MENU_ITEM(<u>MenuID</u>, ItemName, Category, Price, Availability)**

6. **ORDER(<u>OrderID</u>, OrderDate, OrderStatus, TotalAmount, CustomerID)**
   FK: CustomerID references CUSTOMER(UserID)

7. **ORDER_DETAILS(<u>OrderID</u>, <u>MenuID</u>, OrderQty, Subtotal)**
   FK1: OrderID references ORDER(OrderID)
   FK2: MenuID references MENU_ITEM(MenuID)

8. **FEEDBACK(<u>FeedbackID</u>, Rating, FeedbackComment, DateSubmitted, CustomerID, OrderID)**
   FK1: CustomerID references CUSTOMER(UserID)
   FK2: OrderID references ORDER(OrderID)

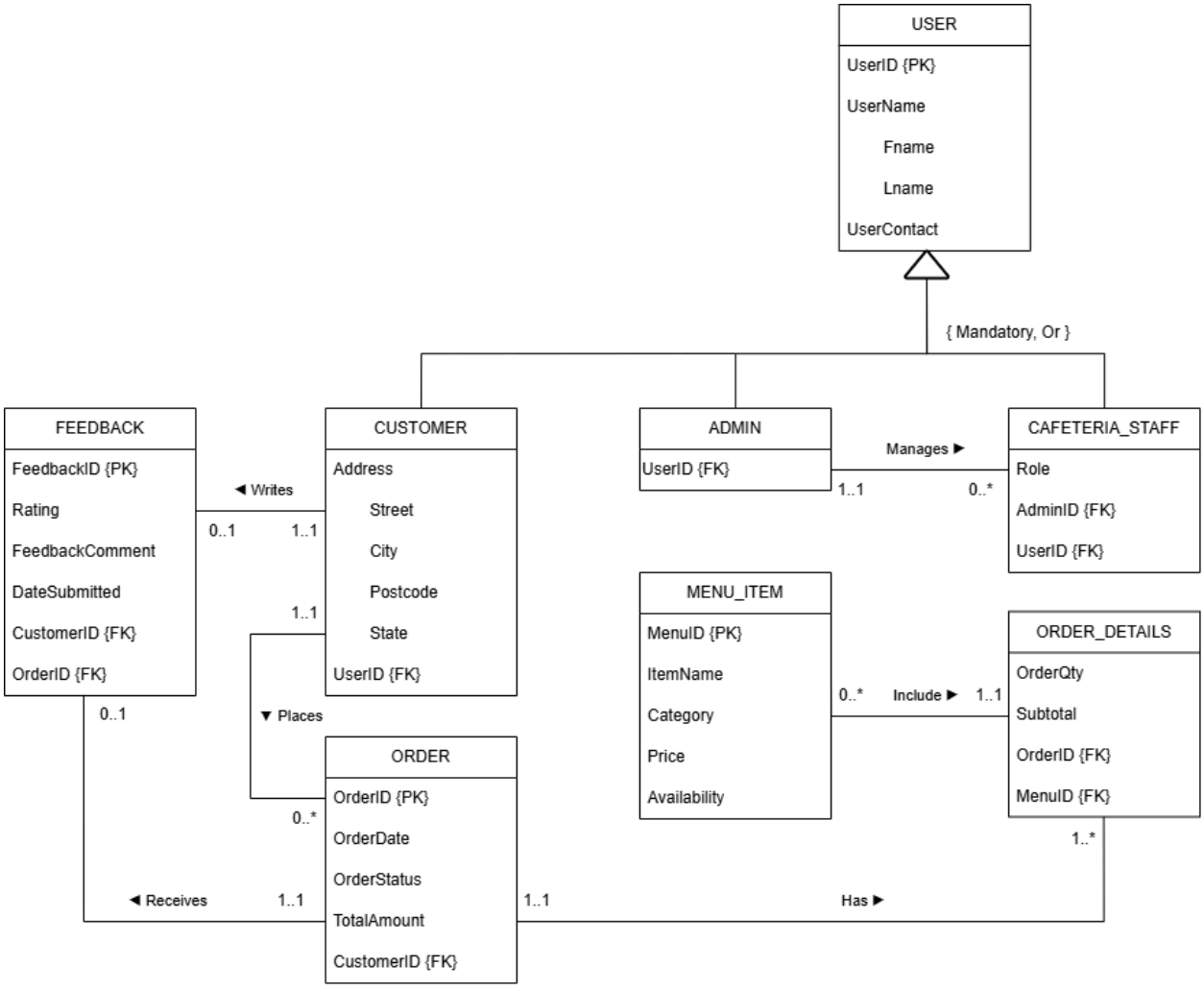

Figure 2.1 Logical ERD (global data model) of Cafeteria Management System

4.2 Updated Data Dictionary

Table Name	Field Name	Data Type	Description
USER	UserID (PK)	VARCHAR(10)	Unique identifier for each user
	Fname	VARCHAR(50)	First name of the user
	Lname	VARCHAR(50)	Last name of the user
	UserContact	VARCHAR(20)	Customer number for login.
CUSTOMER	UserID (PK, FK)	VARCHAR(10)	References USER(UserID)
	Address	VARCHAR(150)	Customer address
	Street	VARCHAR(100)	Street name
	City	VARCHAR(50)	City name
	Postcode	VARCHAR(10)	Postal code
	State	VARCHAR(50)	State
ADMIN	AdminID (PK, FK)	VARCHAR(10)	References USER(UserID)
CAFETERIA_STAFF	UserID (PK, FK)	VARCHAR(10)	References USER(UserID)
	Role	VARCHAR(50)	Staff role (e.g., cashier, cook)
	AdminID (FK)	VARCHAR(10)	References ADMIN(UserID)
MENU_ITEM	MenuID (PK)	VARCHAR(10)	Unique menu item ID
	ItemName	VARCHAR(100)	Name of food/drink
	Category	VARCHAR(50)	Food category
	Price	DECIMAL(6,2)	Price of item
	Availability	BOOLEAN	Item availability
ORDER	OrderID (PK)	VARCHAR(10)	Unique order ID

	OrderDate	DATE	Date of order
	OrderStatus	VARCHAR(30)	Order status
	TotalAmount	DECIMAL(8,2)	Total price
	CustomerID (FK)	VARCHAR(10)	References CUSTOMER(UserID)
ORDER_DETAILS	OrderID (PK, FK)	VARCHAR(10)	References ORDER(OrderID)
	MenuID (PK, FK)	VARCHAR(10)	References MENU_ITEM(MenuID)
	OrderQty	INT	Quantity ordered
	Subtotal	DECIMAL(8,2)	Price × quantity
FEEDBACK	FeedbackID (PK)	VARCHAR(10)	Unique feedback ID
	Rating	INT	Rating from 1 to 5.
	FeedbackComment	VARCHAR(255)	Customer’s written comment..
	DateSubmitted	DATE	Feedback date.
	CustomerID (FK)	VARCHAR(10)	References CUSTOMER(UserID)
	OrderID (FK)	VARCHAR(10)	References ORDER(OrderID)

4.3 Normalization

5.0 RELATIONAL DB SCHEMAS (AFTER NORMALIZATION)

6.0 SQL STATEMENTS (DDL & DML)

6.1 Logical DDL: Schema Definition

6.1.1 USER Table

```
CREATE TABLE USER (
    UserID VARCHAR(10) PRIMARY KEY,
    Fname VARCHAR(50),
    Lname VARCHAR(50),
    UserContact VARCHAR(20)
);
```

Explanation: This DDL statement creates the base entity with a PRIMARY KEY to uniquely identify each person. VARCHAR is used for flexible text storage while ensuring entity integrity.

6.1.2 CUSTOMER Table

```
CREATE TABLE CUSTOMER (
    UserID VARCHAR(10) PRIMARY KEY,
    Address VARCHAR(150),
    Street VARCHAR(100),
    City VARCHAR(50),
    Postcode VARCHAR(10),
    State VARCHAR(50),
    CONSTRAINT fk_customer_user FOREIGN KEY (UserID) REFERENCES
USER(UserID)
);
```

Explanation: This table uses a shared PRIMARY KEY that also acts as a FOREIGN KEY to link specific address data to a user. This setup enforces referential integrity by ensuring a customer record must exist in the parent USER table first.

6.1.3 CAFETERIA_STAFF Table

```
CREATE TABLE CAFETERIA_STAFF (
    UserID VARCHAR(10) PRIMARY KEY,
    Role VARCHAR(50),
    AdminID VARCHAR(10),
    FOREIGN KEY (UserID) REFERENCES USER(UserID),
    FOREIGN KEY (AdminID) REFERENCES ADMIN(AdminID)
);
```

Explanation: This command establishes staff roles while maintaining a hierarchical link to an administrator via a FOREIGN KEY. It ensures that every staff member is properly categorized and supervised within the system.

6.1.4 MENU_ITEM Table

```
CREATE TABLE MENU_ITEM (
    MenuID VARCHAR(10) PRIMARY KEY,
    ItemName VARCHAR(100),
    Category VARCHAR(50),
    Price DECIMAL(6,2),
    Availability BOOLEAN
);
```

Explanation: DECIMAL is used for the price to provide exact numeric precision required for financial calculations. The BOOLEAN type allows for a simple true/false flag to track item availability.

6.1.5 ORDER_DETAILS Table

```
CREATE TABLE ORDER_DETAILS (
    OrderID VARCHAR(10),
    MenuID VARCHAR(10),
    OrderQty INT,
    Subtotal DECIMAL(8,2),
    CONSTRAINT pk_order_details PRIMARY KEY (OrderID, MenuID),
    FOREIGN KEY (OrderID) REFERENCES `ORDER`(OrderID),
    FOREIGN KEY (MenuID) REFERENCES MENU_ITEM(MenuID)
);
```

Explanation: This table uses a composite primary key to uniquely identify each item line within a specific order. The foreign keys maintain the relationship between the transaction and the items purchased.

6.2 Logical DML: Data Population

6.2.1 Data Population (Requirement: 20 Records)

The following INSERT INTO statements populate the USER table to establish the minimum required data volume.

```
INSERT INTO USER (UserID, Fname, Lname, UserContact) VALUES
('U001', 'Ahmad', 'Zaki', '011-22334455'), ('U002', 'Siti', 'Aminah',
'012-33445566'),
('U003', 'John', 'Tan', '013-44556677'), ('U004', 'Mary', 'Lim',
'014-55667788'),
('U005', 'Ravi', 'Kumar', '015-66778899'), ('U006', 'Fatimah', 'Aziz',
'016-77889900'),
('U007', 'Chong', 'Wei', '017-88990011'), ('U008', 'Sarah', 'Connor',
'018-99001122'),
('U009', 'Ali', 'Baba', '019-00112233'), ('U010', 'Lee', 'Min',
'011-11223344'),
('U011', 'Zulkifli', 'Musa', '012-22334455'), ('U012', 'Grace', 'Ho',
'013-33445566'),
('U013', 'Badrul', 'Hisham', '014-44556677'), ('U014', 'Nor', 'Aini',
'015-55667788'),
('U015', 'Karthik', 'S', '016-66778899'), ('U016', 'Sophia', 'Luo',
'017-77889900'),
('U017', 'Daniel', 'Iskandar', '018-88990011'), ('U018', 'Aisha', 'Bakri',
'019-99001122'),
('U019', 'William', 'Tan', '011-22112233'), ('U020', 'Jessica', 'Wong',
'012-33223344');
```

Explanation: This DML command uses the INSERT INTO statement to add new rows of data into the USER table. By populating 20 unique records, the database provides a sufficient dataset for meaningful querying and reporting.

6.2.2 Advanced Data Retrieval

These queries demonstrate proficiency in retrieving data using specific comparison and logical operators.

6.2.2.1 Skill 1: Range Filtering (BETWEEN)

```
SELECT * FROM MENU_ITEM WHERE Price BETWEEN 5.00 AND 15.00;
```

Explanation: This query retrieves items within a specific price range using the BETWEEN operator, allowing for efficient financial filtering.

6.2.2.2 Skill 2: Set Membership (IN)

```
SELECT * FROM USER WHERE UserID IN ('U001', 'U005', 'U010');
```

Explanation: The IN operator is utilized here to search for values within a predefined list set, simplifying multiple-record selection.

6.2.2.3 Skill 3: Pattern Matching (LIKE)

```
SELECT Fname, UserContact FROM USER WHERE Fname LIKE 'A%';
```

Explanation: This skill uses the % wildcard to perform a search for names starting with "A," demonstrating flexible text retrieval.

6.2.2.4 Skill 4: Handling Nulls (IS NULL)

```
SELECT * FROM FEEDBACK WHERE FeedbackComment IS NULL;
```

Explanation: This query identifies records where columns contain no data, which is essential for identifying incomplete customer feedback.

6.2.2.5 Skill 5: Multi-Condition Logic (AND)

```
SELECT * FROM MENU_ITEM WHERE Category = 'Food' AND Availability = TRUE;
```

Explanation: The AND logical operator ensures that both conditions must be true, providing precise control over the result set.

7.0 SUMMARY