PHASE 3 DATABASE LOGICAL DESIGN & SQL

UTM SHUTTLE BUS SYSTEM

SECD2523

DATABASE

SECTION 03

PREPARED BY:

NO.	STUDENT NAME	STUDENT ID
1	NUR ALYA IZZATI BINTI MOHD ROZILI	A24CS0156
2	HANIS SOFIYA BINTI BADRUL EHSAN	A24CS0080
3	SITI MAISARAH BINTI AHMAD ARZMY	A24CS0190
4	NUR RIHHADATUL ARIFAH BINTI MOHD NAZARUDDIN	A24CS0166

PREPARED TO:

DR. HASLINA BINTI HASHIM

SUBMISSION DATE: 26/12/2025

Table of Content

1.0 Introduction

The UTM shuttle bus plays a vital role in transporting students and staff across various locations within the campus. It serves as a convenient, cost effective and eco friendly solution for daily commuting needs especially for those without personal transportation. As the university population continues to grow, the shuttle service becomes increasingly important in maintaining smooth campus operations and ensuring punctuality for students to attend their classes, meetings and other related activities.

Recent feedback from students and staff highlights the growing dissatisfaction with the current shuttle system. Especially regarding the long waiting time as bus schedules tend to be unsure. Since the demand for transportation within the university increases due to increase in student and staff, these issues are becoming more prominent so it is necessary and timely for technology interventions to maintain efficiency.

Despite its importance, the systems will tend to face several operational challenges that will reduce its efficiency, reliability and convenience. Common issues that the current system face are unreliable bus schedules, overcrowding especially during peak hours and frequent misplacement of personal belongings. These problems not only cause frustration among users but also undermine the reliability and credibility of the bus service. Absence of real-time tracking and effective communication further limits the system's ability to respond to changing demands and emergencies.

Our team has embarked on a project planning approach to develop innovative and technology-based solutions so that all of these problems can be addressed immediately. This project proposes a technology-based solution to improve the shuttle service through a better tracking system by adding a real-time system, digital lost and found reporting system and seat reservation capabilities. By modernizing the shuttle bus system, this project aims to streamline operations,facilitate the recovery of lost items, reduce waiting time and overcrowding and ultimately enhance overall user satisfaction and experience.

2.0 Overview of Project

In Phase 3, the conceptual ERD developed in Phase 2 is converted into a logical ERD that can be implemented in a relational database. At this stage, several modifications are made to simplify the design by removing complex relationships and restructuring entities to better represent the shuttle bus reservation system. Each entity in the logical ERD is then transformed into a relational schema by defining attributes, primary keys, and foreign keys. This is important for the shuttle bus system as entities such as user, bus, schedule, seat reservation, route, and administration must be clearly connected to support system operations.

Normalization is carried out up to BCNF to reduce data redundancy and prevent data inconsistency. This ensures that information such as seat reservations, bus schedules, and user details are stored efficiently and correctly related. The data dictionary is updated accordingly to reflect the final normalized design. Lastly, SQL statements are proposed and matched with the interface design to validate that the logical ERD supports the system's transaction requirements. This confirms that basic functions such as viewing schedules, reserving seats, and managing bus information can be performed effectively.

3.0 Database Conceptual Design

3.1 Updated Business Rule

Administrator:

- Manage and update shuttle bus schedules in the system.
- Oversee and handle digital lost-and-found reports submitted by users or drivers.
- Monitor bus tracking data, seat reservations, and overall system activity.
- View and respond to feedback submitted by students.
- Generate reports on bus performance, passenger capacity, and feedback statistics.

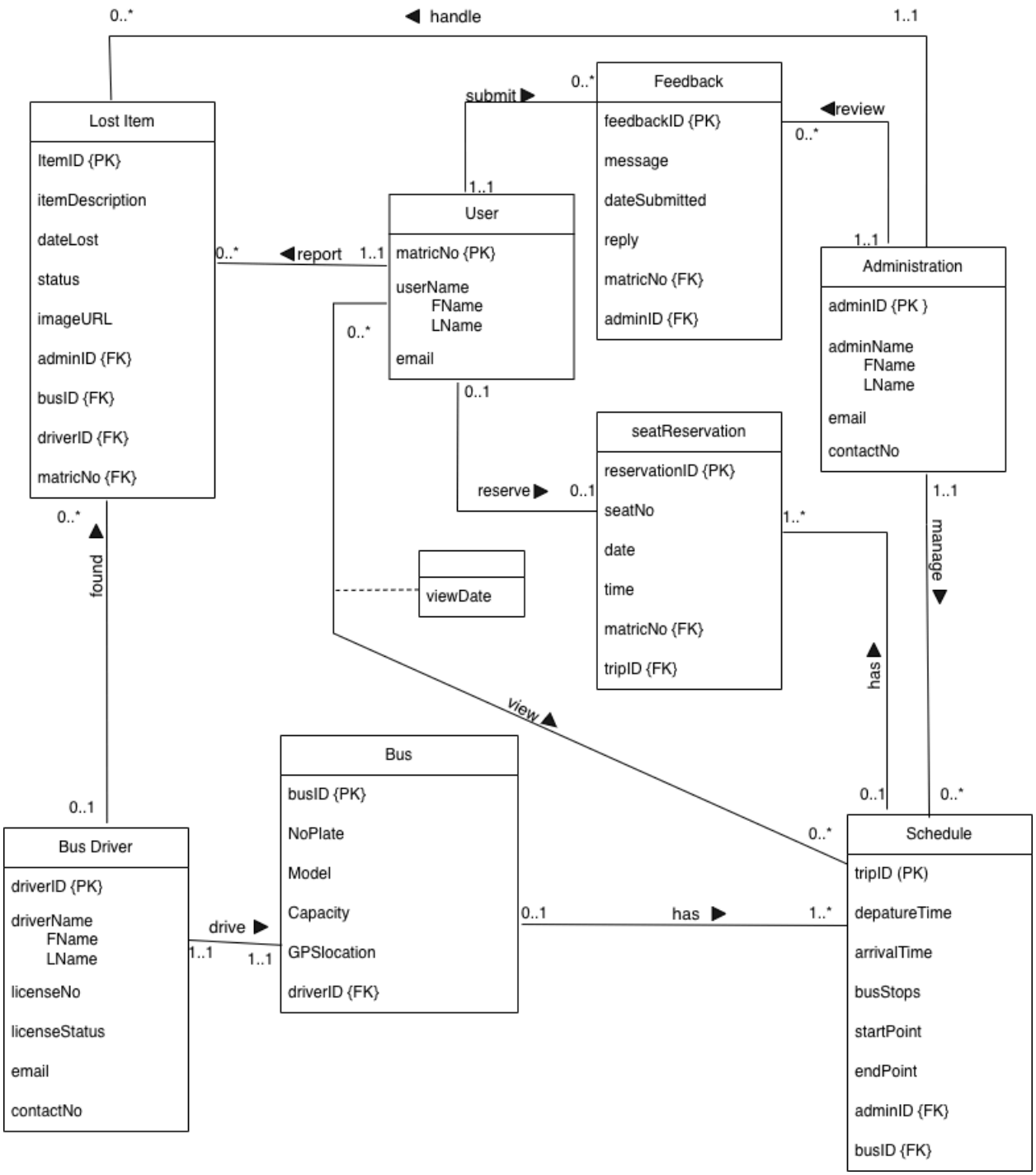
Bus Driver:

- Log into the system to update current bus status and report found items.
- Submit details of lost items discovered during trips.
- View assigned bus routes and schedules managed by the administrator.
- Update trip completion status after each route.

User:

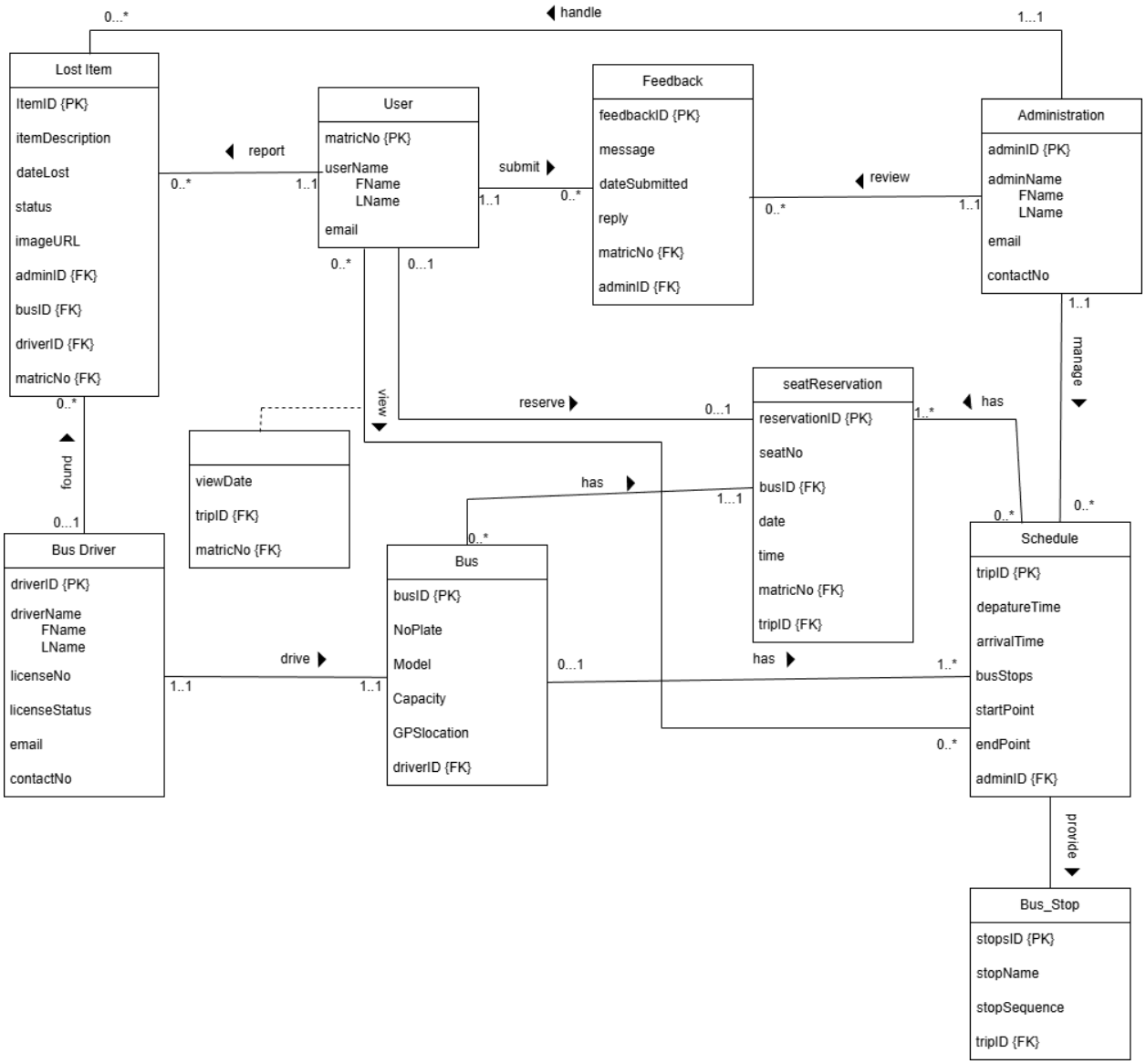
- Log into the shuttle bus system using UTM credentials.
- View real-time bus locations and shuttle schedules.
- Reserve seats for specific trips during peak hours.
- Submit lost item reports or check lost-and-found listings.
- Provide service feedback through the digital feedback form.

3.2 Conceptual ERD



4.0 DB Logical Design

4.1 Logical ERD



4.2 Updated Data Dictionary

4.2.1 Entity Description

Entity	Description	Occurrence
User	User's information	Users submit feedback for experience and improvement and report their lost items to administration. Users check their selected bus schedule.
Administration	Administration's information	Administrators review reports, manage feedback replies and oversee lost item handling. Administrators also manage bus schedules.
Bus Driver	Bus Driver's information	Drivers are linked to buses and may report or return lost items on the bus.
Bus	Bus's information	Each bus is assigned to a driver and may have multiple trip schedules.
Schedule	Schedule's information	Each schedule records planned bus trips managed by the administration and linked to a specific bus.
Lost Item	Lost Item's information	Each record is created when a user reports a lost item or when a driver submits a found item, later updated once handled by the administrator.
Feedback	Feedback's information	Each feedback entry is submitted by a user and reviewed or replied to by the administrators to improve service quality.
Seat Reservation	Reservation's information	User book seats for available trips. Each reservation is linked to one user and one trip. A seat can only be

		reserved once per trip.
Bus Stop	Bus Stop's information	Each stop belongs to a specific trip schedule and includes stop name and sequence in the route.

4.2.2 Relationships Description

Entity	Multiplicity	Relationship	Multiplicity	Entity
User	1..1	report	0..*	Lost Item
	0..*	view	0..*	Schedule
	1..1	submit	0..*	Feedback
	1..1	reserve	0..*	Seat Reservation
Administration	1..1	handle	0..*	Lost Item
	1..1	manage	0..*	Schedule
	1..1	review	0..*	Feedback
Bus Driver	1..1	drive	1..1	Bus
	0..1	found	0..*	Lost Item
Bus	0..1	has	1..*	Schedule
	0..*	has	1..*	Seat Reservation
Schedule	1..1	has	0..*	Seat Reservation
	1..1	provide	1..*	Bus Stop

4.2.3 Attribute Relationship

Entity	Attributes	Description	Data Type	Null	Multi-Valued
User	matricNo	Uniquely identify a user (PK)	VARCHAR2(10)	No	No
	userName	Name of user	VARCHAR2(60)	No	No

	FName	First name of user	VARCHAR2(30)	No	No
	LName	Last name of user	VARCHAR2(30)	No	No
	email	Email of user	VARCHAR	No	No
Administration	adminID	Uniquely identify a administration (PK)	VARCHAR2(10)	No	No
	adminName	Name of administrator	VARCHAR2(60)	No	No
	FName	First name of administrator	VARCHAR2(30)	No	No
	LName	Last name of administrator	VARCHAR2(30)	No	No
	email	Email of administrator	VARCHAR	No	No
	contactNo	Contact number of administrator	VARCHAR2(13)	No	Yes
Bus Driver	driverID	Uniquely identify a driver (PK)	VARCHAR2(10)	No	No
	driverName	Name of driver	VARCHAR2(60)	No	No
	FName	First name of user	VARCHAR2(30)	No	No
	LName	Last name of user	VARCHAR2(30)	No	No
	licenseNo	Driving license number of driver	VARCHAR2(20)	No	No
	licenseStatus	Status of license	VARCHAR2(20)	No	No
	email	Email of driver	VARCHAR	Yes	No
	contactNo	Contact number of driver	VARCHAR2(20)	No	Yes
Bus	busID	Uniquely identify a bus (PK)	VARCHAR2(10)	No	No
	NoPlate	Plate number of a bus	VARCHAR2(15)	No	No
	Model	Model of bus	VARCHAR2(40)	No	No
	Capacity	Capacity of bus	INT	No	No

	GPSlocation	Current or latest location of a bus	VARCHAR2(50)	Yes	No
	driverID	A foreign key that identify the bus driver assigned to drive the bus (FK)	VARCHAR2(10)	No	No
Schedule	tripID	Uniquely identify a bus schedule (PK)	VARCHAR2(10)	No	No
	departure Time	Scheduled time at which the bus is planned to depart	TIMESTAMP	No	No
	arrivalTime	Scheduled time at which the bus is planned to arrive	TIMESTAMP	No	No
	startPoint	The starting location of the bus route	VARCHAR2(30)	No	No
	endPoint	The final location of the bus route	VARCHAR2(30)	No	No
	adminID	A foreign key that identify the administrator responsible for managing this schedule (FK)	VARCHAR2(10)	No	No
Lost Item	itemID	Uniquely identify a lost item (PK)	VARCHAR2(10)	No	No
	itemDescription	Details about the lost item	VARCHAR2 (200)	No	No
	dateLost	Date on which the item was reported lost	DATE	No	No
	status	Status of lost item such as ‘Found’, ‘Unclaimed’ or ‘Returned’	VARCHAR2(20)	No	No
	imageURL	The URL or web link to an image of the lost item	VARCHAR2 (250)	Yes	No

4.3 Normalization

1. **USER** (matricNo, userName, FName, LName, email)
   fd1: matricNo → userName, FName, LName, email
   1NF & 2NF & 3NF & BCNF:
   USER (matricNo, userName, FName, LName, email)

2. **ADMINISTRATION** (adminID, adminName, FName, LName, email, contactNo)
   fd1: adminID → adminName, FName, LName, email, contactNo
   1NF & 2NF & 3NF & BCNF:
   ADMINISTRATION (adminID, adminName, FName, LName, email, contactNo)

3. **BUSDRIVER** (driverID, driverName, FName, LName, licenseNo, licenseStatus, email, contactNo)
   fd1: driverID → driverName, FName, LName, licenseNo, licenseStatus, email, contactNo
   1NF & 2NF & 3NF & BCNF:
   BUSDRIVER (driverID, driverName, FName, LName, licenseNo, licenseStatus, email, contactNo)

4. **BUS** (busID, NoPlate, Model, Capacity, GPSlocation, driverID)
   fd1: busID → NoPlate, Model, Capacity, GPSlocation, driverID
   1NF & 2NF & 3NF & BCNF:
   BUS (busID, NoPlate, Model, Capacity, GPSlocation, driverID)

5. **SCHEDULE** (tripID, departureTime, arrivalTime, startPoint, endPoint, adminID, busID)
   fd1: tripID → departureTime, arrivalTime, startPoint, endPoint, adminID, busID
   1NF & 2NF & 3NF & BCNF:
   SCHEDULE (tripID, departureTime, arrivalTime, startPoint, endPoint, adminID, busID)

6. **SEATRESERVATION** (reservationID, seatNo, date, time, matricNo, tripID)
   fd1: reservationID → seatNo, date, time, matricNo, tripID

1NF & 2NF & 3NF & BCNF:

SEATRESERVATION (reservationID, seatNo, date, time, matricNo, tripID)

7. **LOSTITEM** (itemID, itemDescription, dateLost, status, imageURL, adminID, busID, driverID, matricNo)

   fd1: itemID → itemDescription, dateLost, status, imageURL, adminID, busID, driverID, matricNo

   1NF & 2NF & 3NF & BCNF:

   LOSTITEM (itemID, itemDescription, dateLost, status, imageURL, adminID, busID, driverID, matricNo)

8. **FEEDBACK** (feedbackID, message, dateSubmitted, reply, matricNo, adminID)

   fd1: feedbackID → message, dateSubmitted, reply, matricNo, adminID

   1NF & 2NF & 3NF & BCNF:

   FEEDBACK (feedbackID {PK}, message, dateSubmitted, reply, matricNo, adminID)

9. **BUS_STOP** (stopID, stopName, stopSequence, tripID)

   fd1: stopID → stopName, stopSequence, tripID

   1NF & 2NF & 3NF & BCNF:

   BUS_STOP (stopID, stopName, stopSequence, tripID)

*Remark: Underline word is primary key.

5.0 Relational DB Schemas (After Normalization)

The relational database schema for the UTM Shuttle Bus System consists of the following relations:

User (matricNo, userName, FName, LName, email)
Administration (adminID, adminName, FName, LName, email, contactNo)
BusDriver (driverID, driverName, FName, LName, licenseNo, licenseStatus, email, contactNo)
Bus (busID, NoPlate, Model, Capacity, GPSlocation, driverID)
Schedule (tripID, departureTime, arrivalTime, startPoint, endPoint, adminID, busID)
SeatReservation (reservationID, seatNo, date, time, matricNo, tripID)
LostItem (itemID, itemDescription, dateLost, status, imageURL, adminID, busID, driverID, matricNo)
Feedback (feedbackID, message, dateSubmitted, reply, matricNo, adminID)
Bus_Stop (stopID, stopName, stopSequence, tripID)

*Remark: Underline word is primary key.

User

matricNo	userName	FName	LName	email

Administration

adminID	adminName	FName	LName	email	contactNo

BusDriver

driverID	driverName	FName	LName	licenseNo	licenseStatus	email	contactNo

Bus

busID	NoPlate	Model	Capacity	GPSlocation	driverID

Schedule

tripID	departureTime	arrivalTime	startPoint	endPoint	adminID	busID

SeatReservation

reservationID	seatNo	date	time	matricNo	tripID

LostItem

itemID	itemDescription	dateLost	status	imageURL	adminID	busID	driverID	matricNo

Feedback

feedbackID	message	dateSubmitted	reply	matricNo	adminID

Bus_Stop

stopID	stopName	stopSequence	tripID

6.0 SQL Statements (DDL & DML)

6.1 DDL:

6.1.1 User Table

```
CREATE TABLE User (
        matricNo VARCHAR(10) PRIMARY KEY,
        userName VARCHAR(50),
        FName VARCHAR(50),
        LName VARCHAR(50),
        email  VARCHAR(100)
 );
```

6.1.2 Administration Table

```
CREATE TABLE Administration (
        adminID INT PRIMARY KEY,
        adminName VARCHAR(50),
        FName VARCHAR(50),
        LName VARCHAR(50),
        email  VARCHAR(100),
        contactNo VARCHAR(15)
 );
```

6.1.3 Bus Driver Table

```
CREATE TABLE Bus_Driver (
```

```
    driverID INT PRIMARY KEY,
    driverName VARCHAR(50),
    FName VARCHAR(50),
    LName VARCHAR(50),
    licenseNo VARCHAR(20),
    licenseStatus VARCHAR(20),
    email  VARCHAR(100),
    contactNo VARCHAR(15)
);
```

6.1.4 Bus Table

```
CREATE TABLE Bus (
    busID INT PRIMARY KEY,
    NoPlate VARCHAR(50),
    Model VARCHAR(50),
    Capacity INT,
    GPSlocation  VARCHAR(100),
    driverID INT,
    FOREIGN KEY (driverID) REFERENCES Bus_Driver(driverID)
);
```

6.1.5 Schedule Table

```
CREATE TABLE Schedule (
    tripID INT PRIMARY KEY,
    departureTime TIME,
    arrivalTime TIME,
```

```
    busStops VARCHAR(50),
    startPoint VARCHAR(100),
    endPoint VARCHAR(100),
    adminID INT,
    FOREIGN KEY (adminID) REFERENCES Administration(adminID)
);
```

6.1.6 Seat Reservation Table

```
CREATE TABLE SeatReservation (
    reservationID INT PRIMARY KEY,
    seatNo VARCHAR(10),
    busID INT,
    date DATE,
    time TIME,
    matricNo VARCHAR(10),
    tripID INT,
    FOREIGN KEY (busID) REFERENCES Bus(busID),
    FOREIGN KEY (matricNo) REFERENCES User(matricNo),
    FOREIGN KEY (tripID) REFERENCES Schedule(tripID)
);
```

6.1.7 Lost Item Table

```
CREATE TABLE Lost_Item (
    itemID INT PRIMARY KEY,
    itemDescription VARCHAR(200),
    dateLost DATE,
    status VARCHAR(20),
```

```
    imageURL VARCHAR(200),
    adminID INT,
    busID INT,
    driverID INT,
    matricNo VARCHAR(10),
    FOREIGN KEY (adminID) REFERENCES Administration(adminID),
    FOREIGN KEY (busID) REFERENCES Bus(busID),
    FOREIGN KEY (driverID) REFERENCES Bus_Driver(driverID),
    FOREIGN KEY (matricNo) REFERENCES User(matricNo)
);
```

6.1.8 Feedback Table

```
CREATE TABLE Feedback (
    feedbackID INT PRIMARY KEY,
    message VARCHAR(300),
    dateSubmitted DATE,
    reply VARCHAR(300),
    matricNo VARCHAR(10),
    adminID INT,
    FOREIGN KEY (matricNo) REFERENCES User(matricNo),
    FOREIGN KEY (adminID) REFERENCES Administration(adminID)
);
```

6.1.9 Bus Stop Table

```
CREATE TABLE Bus_Stop (
    stopID INT PRIMARY KEY,
    stopName VARCHAR(100),
```

```
        stopSequence INT,
        tripID INT,
        FOREIGN KEY (tripID) REFERENCES Schedule(tripID)
);
```

6.2 Data Manipulation Language (DML)

6.2.1 Insert User Data

```
    INSERT INTO User
    VALUES ('A22CS0123', 'amira22', 'Nurul' , 'Amira',
'nurulamira@graduate.utm.my');
```

6.2.2 Insert Administration Data

```
    INSERT INTO Administration
    VALUES ( 1, 'Admin1', 'Ahmad', 'Adam', 'admin1@utm.my',  '0112233445' );
```

6.2.3 Insert Bus Driver Data

```
    INSERT INTO Bus_Driver
    VALUES ( 1, 'Razak', 'Abdul', 'Razak', 'D12345', 'Active',  'razak@gmail.com',
'0136789543' );
```

6.2.4 Insert Bus Data

```
    INSERT INTO Bus
    VALUES ( 10, 'JRQ1122', 'Hino', 40, 'UTM CentrePoint',  1 );
```

6.2.5 Insert Schedule Data

```
INSERT INTO Schedule
VALUES ( 1001, '08:00', '09:00', 'KTR,KTDI' , 'Kolej' , 'Library' , 1 );
```

6.2.6 Insert Seat Reservation

```
INSERT INTO SeatReservation
VALUES ( 501, 'A3', 10, '2025-12-27', '08:00', 'A22CS0123', 1001 );
```

6.2.7 Insert Lost Item Report

```
INSERT INTO Lost_Item
VALUES ( 301, 'Black Purse', '2025-12-25', 'Pending', 'purse.jpg', 1, 10, 1,
'A22CS0123' );
```

6.2.8 Insert Feedback

```
INSERT INTO Feedback
VALUES ( 701, 'Bus arrived late', '2025-12-26', NULL, 'A22CS0123', 1);
```

6.2.9 Update Lost Item Status

```
UPDATE Lost_Item
SET status = 'Found'
WHERE itemID = 301;
```

6.2.10 Update Feedback Reply

```
UPDATE Feedback
SET reply = ‘Thank you for your feedback. We will improve our service.’
WHERE feedbackID = 701;
```

6.2.11 Retrieve Bus Schedule

```
SELECT * FROM Schedule;
```

6.2.12 Retrieve User Seat Reservation

```
SELECT *
FROM SeatReservation
WHERE matricNo =  ‘A22CS0123’;
```

7.0 Summary

During this phase, we focused on transforming our Conceptual ERD from Phase 2 into a logical ERD suitable for a relational database. We learned how to simplify complex relationships, remove non-relational features, many to many relationships and structure the entities according to relational database principles. This process helped us understand how to translate conceptual ideas into a clear and organized design that can be implemented in SQL.

Each entity such as user, bus, schedule, seat reservation, bus stop, and administration, was converted into a relational schema. We defined attributes, primary keys, and foreign keys for each table to ensure proper relationships and data integrity. Normalization was also performed up to Boyce-Codd Normal Form (BCNF) to eliminate redundancy and maintain accurate and consistent data, especially for seat reservations, bus schedules, and user information.

The final Logical ERD provided a clear visual representation of how all entities are connected, while the updated data dictionary helped document the tables and their attributes properly. We also reviewed and adjusted business rules to make sure the system operations, such as seat reservations, schedule management, bus tracking, lost item reporting, and feedback handling, are supported correctly.

Overall, we learned that normalization and proper structuring of entities and relationships help prevent data errors, reduce redundancy, and make the system more efficient. These skills are directly useful for the UTM shuttle bus system. By applying what we learned, the shuttle bus system can operate more smoothly, making it easier for users to book seats, for drivers to manage trips, and for administrators to oversee the overall system.