THE CATHOLIC UNIVERSITY OF EASTERN AFRICA

UNIT CODE: CMT 302

UNIT TITLE: ADVANCED DATABASE SYSTEMS

PROJECT NAME: ONLINE BUS TICKET BOOKING SYSTEM

SUBMISSION DATE: 22TH NOV 2024

GROUP NO:31 STREAM B

MEMBERS:

1062811- JULIUS MAINGI NJUGUNA

1062793- NJENGA KENDI

1046045- SHARON CHEPKORIR

1045121- LUCY MWONGELI

1049060- PETER KILUNDA

OVERVIEW

Badariah, (2007) emphasised that the online bus ticket booking System which was developed at Politeknik Kota Kuala Terengganu (PKKT) was to make sure that users could make their online booking or reservations to their desired transport companies with facilities provided by the new system. He pointed out that the methodology and technology being used in this new transport system could be applied to other areas of activities. The user who wants to use the transport must make an application to book the transport before boarding.

Similarly, after considering the type of system which Badariah adopted, this project will be designed with the same aim of presenting the customers of Wema Transport Company with the opportunity of making reservations at the comfort of their homes or offices without being faced with the challenges of queuing at counters before embarking on any journey. This project will also enlighten prospective customers and users of the system on the need to patronise the system as it displays more advantages over the old system by providing an easy to use Graphic User interface (GUI) interaction, checking availability of routes before boarding etc.

RATIONALE:

Online bus ticket booking systems can be justified for a variety of reasons, including:

Convenience for customers: Customers can book tickets at any time, from any device, and from anywhere with an internet connection. This eliminates the need for phone calls or physical visits.

Streamlined operations: Online booking systems can automate reservation processes, simplify calendar management, and improve communication with clients.

Reduced risk of human error: Online booking systems can help reduce the risk of human error.

Increased revenue: Online booking systems can help increase revenue to its servicing company.

Superior customer experience: Online booking systems can help deliver a superior customer experience and leave reviews and ratings by customers.

Avoid disappointment of queues: Online ticketing allows visitors to purchase tickets in advance, avoiding the need to wait in long queues.

OBJECTIVES:

The main purpose of this project is to automate the manual procedures of reserving a bus ticket for a road journey facilitated through a transport company.

The specific objectives of this project consists of:

i) To provide a web-based bus booking system where a customer can buy a bus ticket without the need to queue up at the counter.

ii) To enable customers to check the availability of buses online.

iii) To enable customers to choose their seats online.

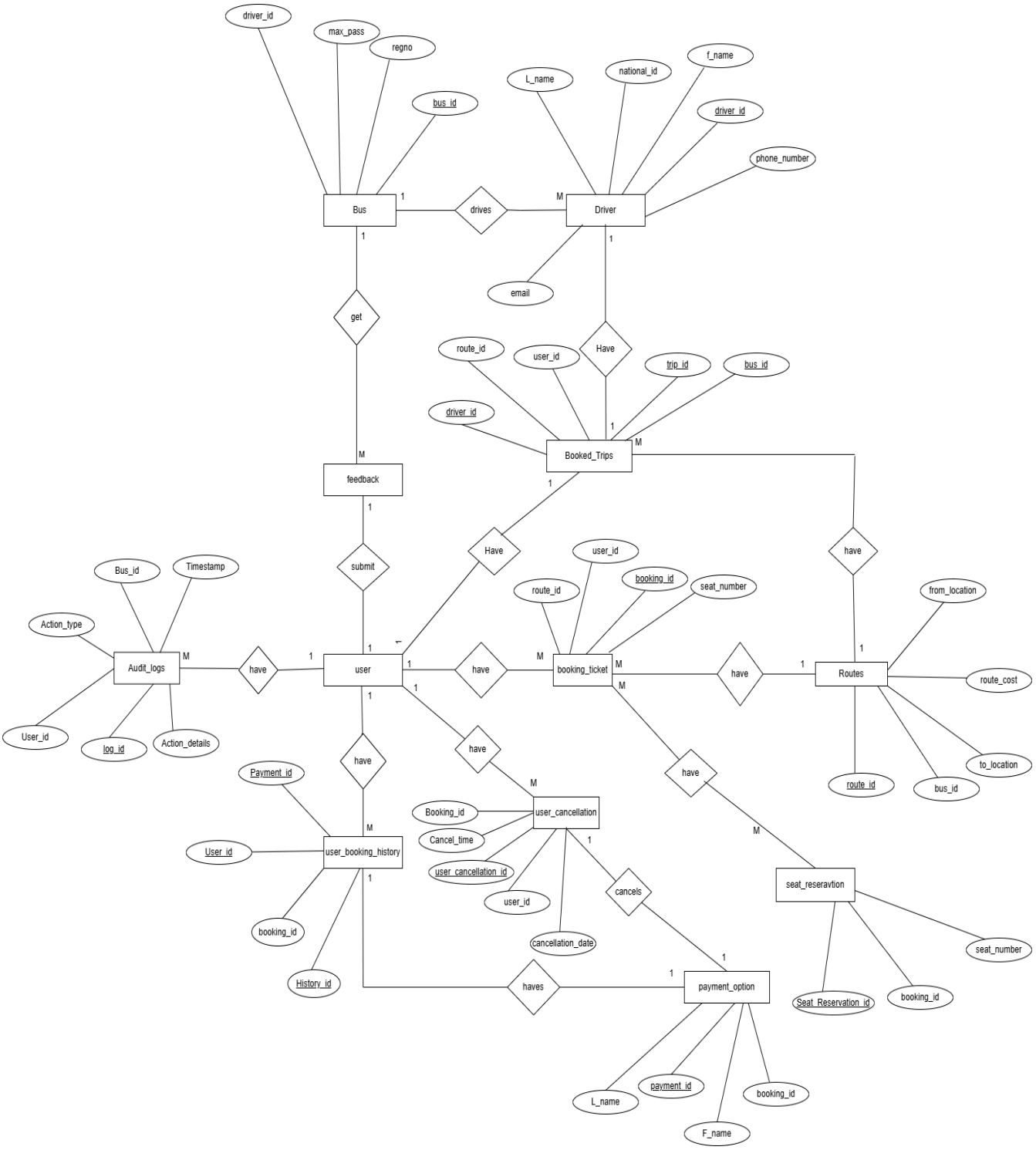
iii) To enable customers to check the time of departure for a bus.

iv) To enable customers to pay for their bus ticket online.

iv) To provide the ability of customers to cancel their reservation online incase of change of plans.

SYSTEM DESIGN:

(i) ER- DIAGRAMS-



(ii)TABLE STRUCTURES

1.Users table

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
user_id	int	NO	PRI	NULL	auto_increment
first_name	varchar(255)	YES		NULL	
last_name	varchar(255)	YES		NULL	
email	varchar(255)	YES	UNI	NULL	
password	varchar(255)	YES		NULL	

Result 2

2.Routes table

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
route_id	int	NO	PRI	NULL	auto_increment
bus_id	int	YES	MUL	NULL	
from_location	varchar(255)	YES	MUL	NULL	
to_location	varchar(255)	YES		NULL	
max_passengers	int	YES		NULL	
route_name	varchar(255)	YES		NULL	
route_cost	float	YES		NULL	

3.Buses table

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
bus_id	int	NO	PRI	NULL	auto_increment
reg_number	varchar(50)	YES	UNI	NULL	
max_passengers	int	YES		NULL	
driver_id	int	YES	MUL	NULL	

4.Drivers table

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
driver_id	int	NO	PRI	NULL	auto_increment
national_id	varchar(20)	YES	UNI	NULL	
first_name	varchar(255)	YES		NULL	
last_name	varchar(255)	YES		NULL	
phone_number	varchar(20)	YES		NULL	
email	varchar(255)	YES	UNI	NULL	

---

5 Admin table

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
admin_id	int	NO	PRI	NULL	auto_increment
first_name	varchar(255)	YES		NULL	
last_name	varchar(255)	YES		NULL	
email	varchar(255)	YES	UNI	NULL	
password	varchar(255)	YES		NULL	

---

6 Booking Ticket

Result Grid | Filter Rows: | Export: | Wrap Cell Conter

Field	Type	Null	Key	Default	Extra
booking_id	int	NO	PRI	NULL	auto_increment
user_id	int	YES	MUL	NULL	
route_id	int	YES	MUL	NULL	
date	date	YES		NULL	
time	time	YES		NULL	
price	decimal(10,2)	YES		NULL	
seat_number	varchar(10)	YES	UNI	NULL	

7.Payment_option

Field	Type	Null	Key	Default	Extra
payment_id	int	NO	PRI	NULL	auto_increment
first_name	varchar(255)	YES		NULL	
last_name	varchar(255)	YES		NULL	
booking_id	int	YES	MUL	NULL	
payment_date	datetime	YES		NULL	
payment_status	enum('confirmed','pending','failed')	YES		NULL	
mpesa_receipt_no	varchar(255)	YES		NULL	
MerchantRequestID	varchar(222)	YES		NULL	
CheckoutRequestID	varchar(222)	YES		NULL	

8.Seat_Reservation

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
reservation_id	int	NO	PRI	NULL	auto_increment
booking_id	int	YES	MUL	NULL	
seat_number	varchar(10)	YES	MUL	NULL	

9.User_booking_history

Field	Type	Null	Key	Default	Extra
history_id	int	NO	PRI	NULL	auto_increment
user_id	int	YES	MUL	NULL	
booking_id	int	YES	MUL	NULL	
day	date	YES		NULL	
time	time	YES		NULL	
payment_id	int	YES	MUL	NULL	

10.Booked_trips

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
trip_id	int	NO	PRI	NULL	auto_increment
user_id	int	YES	MUL	NULL	
route_id	int	YES	MUL	NULL	
bus_id	int	YES	MUL	NULL	
driver_id	int	YES	MUL	NULL	
trip_date	date	YES		NULL	
trip_time	time	YES		NULL	

11.Feedback table

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
feedback_id	int	NO	PRI	NULL	auto_increment
user_id	int	YES	MUL	NULL	
bus_id	int	YES	MUL	NULL	
rating	int	YES		NULL	
comments	varchar(255)	YES		NULL	
feedback_date	datetime	YES		NULL	

12.Audit_Logs

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
log_id	int	NO	PRI	NULL	auto_increment
action_type	enum('booking','cancellation','payment','update'...	YES		NULL	
user_id	int	YES	MUL	NULL	
action_details	text	YES		NULL	
timestamp	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

13.User_cancellation_table

Field	Type	Null	Key	Default	Extra
cancellation_id	int	NO	PRI	NULL	auto_increment
user_id	int	YES	MUL	NULL	
booking_id	int	YES	MUL	NULL	
cancellation_date	date	YES		NULL	
cancellation_time	time	YES		NULL	
payment_id	int	YES	MUL	NULL	
cancellation_status	enum('Cancelled','Pending')	YES		NULL	

(iii) SQL SCHEMA

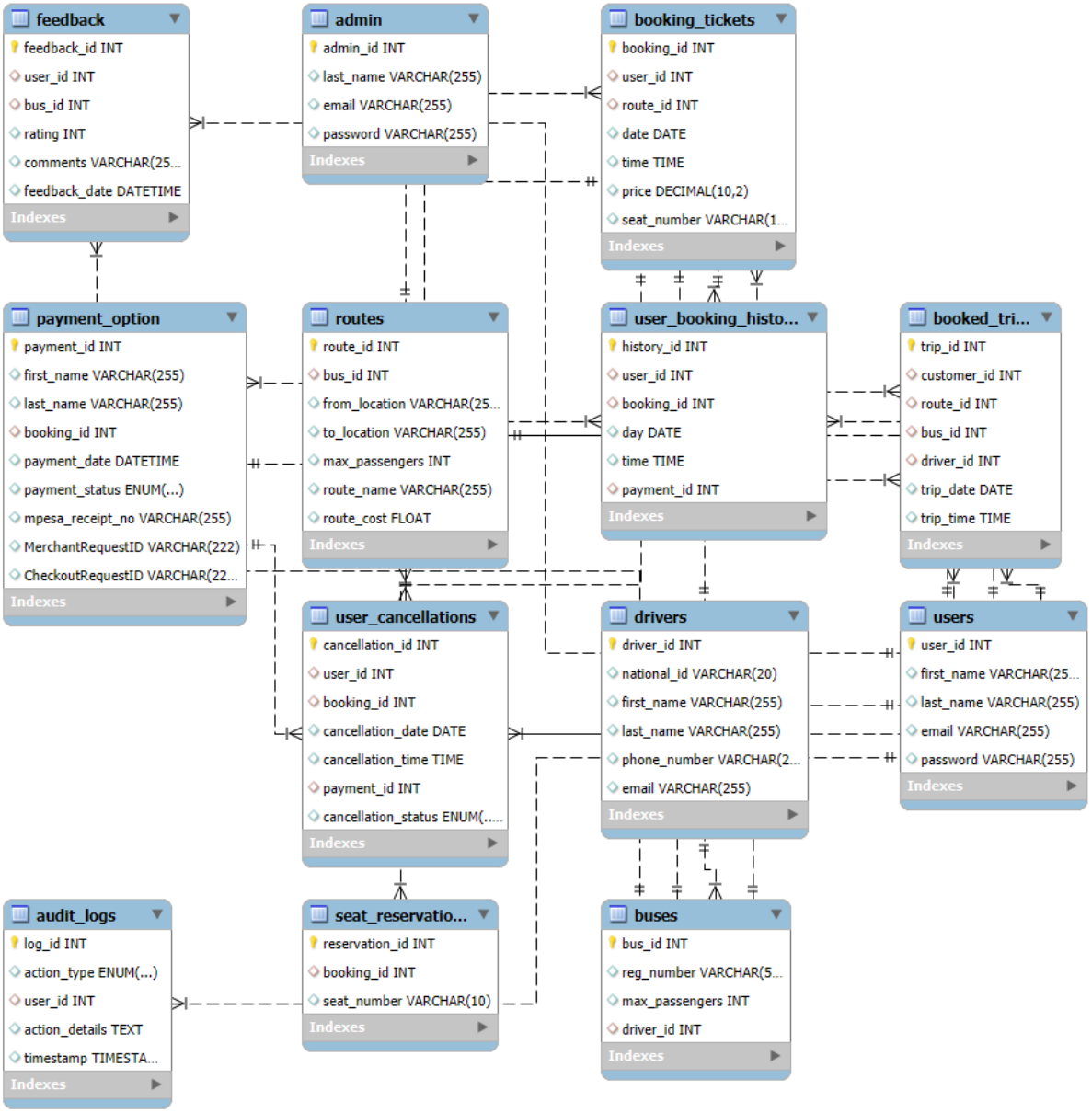


TABLE CREATION SCRIPTS

Tables

```sql
CREATE database PROJECT_rev;

USE PROJECT_rev;
CREATE TABLE users (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(255),
    last_name VARCHAR(255),
    email VARCHAR(255) UNIQUE,
    password VARCHAR(255)
);
desc users;


-- Creating Drivers Table
CREATE TABLE drivers (
    driver_id INT AUTO_INCREMENT PRIMARY KEY,
    national_id VARCHAR(20) UNIQUE,
    first_name VARCHAR(255),
    last_name VARCHAR(255),
    phone_number VARCHAR(20),
    email VARCHAR(255) UNIQUE
);
desc drivers;
```

```sql
-- Creating booking tickets table
CREATE TABLE booking_tickets (
    booking_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    route_id INT,
    date DATE,
    time TIME,
    price DECIMAL(10, 2),
    seat_number VARCHAR(10),
    unique(seat_number),
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (route_id) REFERENCES routes(route_id)
);
desc booking_tickets;

-- Creating Payment Option Table
CREATE TABLE payment_option (
    payment_id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(255),
    last_name VARCHAR(255),
    booking_id INT,
    payment_date DATETIME,
    payment_status ENUM('confirmed', 'pending', 'failed'),
    mpesa_receipt_no VARCHAR(255),
    MerchantRequestID VARCHAR(222),
    CheckoutRequestID VARCHAR(222),
    FOREIGN KEY (booking_id) REFERENCES booking_tickets(booking_id));
```

```sql
-- Creating User Booking History Table
CREATE TABLE user_booking_history (
    history_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    booking_id INT,
    day DATE,
    time TIME,
    payment_id INT,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (booking_id) REFERENCES booking_tickets(booking_id),
    FOREIGN KEY (payment_id) REFERENCES payment_option(payment_id)
);
desc user_booking_history;

-- Creating User Cancellations Table
CREATE TABLE user_cancellations (
    cancellation_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    booking_id INT,
    cancellation_date DATE,
    cancellation_time TIME,
    payment_id INT,
    cancellation_status ENUM('Cancelled', 'Pending'),
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (booking_id) REFERENCES booking_tickets(booking_id),
    FOREIGN KEY (payment_id) REFERENCES payment_option(payment_id)
);
```

```sql
-- Creating Booked Trips Table
CREATE TABLE booked_trips (
    trip_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    route_id INT,
    bus_id INT,
    driver_id INT,
    trip_date DATE,
    trip_time TIME,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (route_id) REFERENCES routes(route_id),
    FOREIGN KEY (bus_id) REFERENCES buses(bus_id),
    FOREIGN KEY (driver_id) REFERENCES drivers(driver_id)
);
desc booked_trips;

-- Creating Admin Table
CREATE TABLE admin (
    admin_id INT AUTO_INCREMENT PRIMARY KEY,
    first_name varchar(255),
    last_name VARCHAR(255),
    email VARCHAR(255) UNIQUE,
    password VARCHAR(255)
);
desc admin;
```

```sql
-- Creating Seat Reservations Table
CREATE TABLE seat_reservations (
    reservation_id INT AUTO_INCREMENT PRIMARY KEY,
    booking_id INT,
    seat_number VARCHAR(10),
    foreign key (seat_number) references booking_tickets(seat_number),
    FOREIGN KEY (booking_id) REFERENCES booking_tickets(booking_id)
);
desc seat_reservations;
-- Creating Audit Logs Table
CREATE TABLE audit_logs (
    log_id INT AUTO_INCREMENT PRIMARY KEY,
    action_type ENUM('booking', 'cancellation', 'payment', 'update', '
    user_id INT,
    action_details TEXT,
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);
desc audit_logs;
```

```sql
CREATE TABLE feedback (
    feedback_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    bus_id INT,
    rating INT,
    comments VARCHAR(255),
    feedback_date DATETIME,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (bus_id) REFERENCES buses(bus_id)
);
desc feedback;
```

```sql
-- Creating Buses Table
CREATE TABLE buses (
    bus_id INT AUTO_INCREMENT PRIMARY KEY,
    reg_number VARCHAR(50) UNIQUE,
    max_passengers INT,
    driver_id INT,
    FOREIGN KEY (driver_id) REFERENCES drivers(driver_id)
);
desc buses;

-- Creating Routes Table
CREATE TABLE routes (
    route_id INT AUTO_INCREMENT PRIMARY KEY,
    bus_id INT,
    from_location VARCHAR(255),
    to_location VARCHAR(255),
    max_passengers INT,
    route_name VARCHAR(255),
    route_cost FLOAT,
    FOREIGN KEY (bus_id) REFERENCES buses(bus_id),
    UNIQUE (from_location, to_location)
);
desc routes;
```

INSERTION

```sql
INSERT INTO users (first_name, last_name, email, password) VALUES
('John', 'mutuku', 'johnmutuku@gmail.com', 'rt455'),
('Jane', 'Maina', 'janemaina@gmail.com', 'hidfghjkl'),
('George', 'Simon', 'georgesimon@gmail.com', 'sdfghjkl'),
('Mary', 'Kenga', 'marykenga@gmail.com', 'asdfgh'),
('Chris', 'Njunguna', 'chrisnjugu@gmail.com', 'vbnm'),
('Patricia', 'Mwende', 'patriciamwende@gmail.com', '83456jl'),
('Michael', 'Odieng', 'michaelodieng@gmail.com', 'zxcvb'),
('Sarah', 'Otieno', 'sarahotieno@gmail.com', 'mkio98'),
('David', 'Malonza', 'davidmalonza@gmail.com', '4rtyui'),
('Linda', 'mary', 'lindamarya@gmail.com', 'wertyui9');


INSERT INTO drivers (national_id, first_name, last_name, phone_number, email) VALUES
('12345678901', 'Joseph', 'Karanja', '0712345678', 'joseph.karanja@bus.com'),
('23456789012', 'Amina', 'Mohamed', '0723456789', 'amina.mohamed@bus.com'),
('34567890123', 'Abdi', 'Omar', '0734567890', 'abdi.omar@bus.com'),
('45678901234', 'Kimani', 'Mwangi', '0745678901', 'kimani.mwangi@bus.com'),
('56789012345', 'Musa', 'Ali', '0756789012', 'musa.ali@bus.com'),
('67890123456', 'Nia', 'Njoroge', '0767890123', 'nia.njoroge@bus.com'),
('78901234567', 'Samuel', 'Wangui', '0778901234', 'samuel.wangui@bus.com'),
('89012345678', 'Emily', 'Kiplangat', '0789012345', 'emily.kiplangat@bus.com'),
('90123456789', 'Abigail', 'Ochieng', '0790123456', 'abigail.ochieng@bus.com'),
('01234567890', 'Eric', 'Mutua', '0801234567', 'eric.mutua@bus.com');
```

```sql
INSERT INTO buses (reg_number, max_passengers, driver_id) VALUES
('KBG 123X', 50, 1),
('KBS 456Y', 40, 2),
('KBT 789Z', 60, 3),
('KBU 012A', 55, 4),
('KBC 345B', 45, 5),
('KBD 678C', 50, 6),
('KBE 901D', 40, 7),
('KBF 234E', 60, 8),
('KBG 567F', 45, 9),
('KBH 890G', 55, 10);
select * from buses;


INSERT INTO routes (bus_id, from_location, to_location, max_passengers, route_name, route_cost) VALUES
(1, 'Nairobi', 'Mombasa', 50, 'Nairobi-Mombasa Express', 1500.00),
(2, 'Nairobi', 'Kisumu', 40, 'Nairobi-Kisumu Highway', 1200.00),
(3, 'Mombasa', 'Nairobi', 60, 'Mombasa-Nairobi Direct', 1600.00),
(4, 'Nairobi', 'Kisii', 55, 'Nairobi-Kisii Route', 1000.00),
(5, 'Nairobi', 'Eldoret', 45, 'Nairobi-Eldoret Connection', 1300.00),
(6, 'Mombasa', 'Kisumu', 50, 'Mombasa-Kisumu Link', 1250.00),
(7, 'Nairobi', 'Nyeri', 40, 'Nairobi-Nyeri', 800.00),
(8, 'Mombasa', 'Kakamega', 60, 'Mombasa-Kakamega Road', 1450.00),
(9, 'Kisumu', 'Nairobi', 45, 'Kisumu-Nairobi Express', 1100.00),
(10, 'Eldoret', 'Nairobi', 55, 'Eldoret-Nairobi Route', 1350.00);
```

```sql
INSERT INTO booking_tickets (user_id, route_id, date, time, price, seat_number) VALUES
(1, 1, '2024-11-19', '10:00:00', 1500.00, 'A1'),
(2, 2, '2024-11-19', '11:00:00', 1200.00, 'B1'),
(3, 3, '2024-11-19', '12:00:00', 1600.00, 'C1'),
(4, 4, '2024-11-19', '13:00:00', 1000.00, 'D1'),
(5, 5, '2024-11-19', '14:00:00', 1300.00, 'E1'),
(6, 6, '2024-11-19', '15:00:00', 1250.00, 'F1'),
(7, 7, '2024-11-19', '16:00:00', 800.00, 'G1'),
(8, 8, '2024-11-19', '17:00:00', 1400.00, 'H1'),
(9, 9, '2024-11-19', '18:00:00', 1100.00, 'I1'),
(10, 10, '2024-11-19', '19:00:00', 1350.00, 'J1')
;


INSERT INTO payment_option (first_name, last_name, booking_id, payment_date, payment_status, mpesa_receipt_no, MerchantRequestID, CheckoutRequestID) VALUES
('John', 'Mutuku', 1, '2024-11-19 10:05:00', 'confirmed', 'MPESA123456', 'MR123', 'CR123'),
('Jane', 'Maina', 2, '2024-11-19 11:05:00', 'confirmed', 'MPESA123457', 'MR124', 'CR124'),
('George', 'Simon', 3, '2024-11-19 12:05:00', 'confirmed', 'MPESA123458', 'MR125', 'CR125'),
('Mary', 'Kenga', 4, '2024-11-19 13:05:00', 'confirmed', 'MPESA123459', 'MR126', 'CR126'),
('Chris', 'Njunguna', 5, '2024-11-19 14:05:00', 'confirmed', 'MPESA123460', 'MR127', 'CR127'),
('Patricia', 'Mwende', 6, '2024-11-19 15:05:00', 'confirmed', 'MPESA123461', 'MR128', 'CR128'),
('Michael', 'Odieng', 7, '2024-11-19 16:05:00', 'confirmed', 'MPESA123462', 'MR129', 'CR129'),
('Sarah', 'Otieno', 8, '2024-11-19 17:05:00', 'confirmed', 'MPESA123463', 'MR130', 'CR130'),
('David', 'Malonza', 9, '2024-11-19 18:05:00', 'confirmed', 'MPESA123464', 'MR131', 'CR131'),
('Linda', 'Mary', 10, '2024-11-19 19:05:00', 'confirmed', 'MPESA123465', 'MR132', 'CR132');
```

```sql
INSERT INTO user_booking_history (user_id, booking_id, day, time, payment_id) VALUES
(1, 1, '2024-11-19', '10:00:00', 1),
(2, 2, '2024-11-19', '11:00:00', 2),
(3, 3, '2024-11-19', '12:00:00', 3),
(4, 4, '2024-11-19', '13:00:00', 4),
(5, 5, '2024-11-19', '14:00:00', 5),
(6, 6, '2024-11-19', '15:00:00', 6),
(7, 7, '2024-11-19', '16:00:00', 7),
(8, 8, '2024-11-19', '17:00:00', 8),
(9, 9, '2024-11-19', '18:00:00', 9),
(10, 10, '2024-11-19', '19:00:00', 10);
```

```sql
INSERT INTO user_cancellations (user_id, booking_id, cancellation_date, cancellation_time, payment_id, cancellation_status) VALUES
(1, 1, '2024-11-20', '10:00:00', 1, 'Cancelled'),
(2, 2, '2024-11-20', '11:00:00', 2, 'Pending'),
(3, 3, '2024-11-20', '12:00:00', 3, 'Cancelled'),
(4, 4, '2024-11-20', '13:00:00', 4, 'Pending'),
(5, 5, '2024-11-20', '14:00:00', 5, 'Cancelled'),
(6, 6, '2024-11-20', '15:00:00', 6, 'Pending'),
(7, 7, '2024-11-20', '16:00:00', 7, 'Cancelled'),
(8, 8, '2024-11-20', '17:00:00', 8, 'Pending'),
(9, 9, '2024-11-20', '18:00:00', 9, 'Cancelled'),
(10, 10, '2024-11-20', '19:00:00', 10, 'Pending');
```

```sql
INSERT INTO booked_trips (user_id, route_id, bus_id, driver_id, trip_date, trip_time) VALUES
(1, 1, 1, 1, '2024-11-19', '10:00:00'),
(2, 2, 2, 2, '2024-11-19', '11:00:00'),
(3, 3, 3, 3, '2024-11-19', '12:00:00'),
(4, 4, 4, 4, '2024-11-19', '13:00:00'),
(5, 5, 5, 5, '2024-11-19', '14:00:00'),
(6, 6, 6, 6, '2024-11-19', '15:00:00'),
(7, 7, 7, 7, '2024-11-19', '16:00:00'),
(8, 8, 8, 8, '2024-11-19', '17:00:00'),
(9, 9, 9, 9, '2024-11-19', '18:00:00'),
(10, 10, 10, 10, '2024-11-19', '19:00:00');

INSERT INTO admin (first_name,last_name, email, password) VALUES
('Kennedy','Musyoka', 'kenmusyoka5678@gmail.com', 'aertyu654432');
```

```sql
INSERT INTO seat_reservations (booking_id, seat_number) VALUES
(1, 'A1'),
(2, 'B1'),
(3, 'C1'),
(4, 'D1'),
(5, 'E1'),
(6, 'F1'),
(7, 'G1'),
(8, 'H1'),
(9, 'I1'),
(10, 'J1');
```

```sql
INSERT INTO audit_logs (action_type, user_id, action_details) VALUES
('booking', 1, 'Booked a ticket for Nairobi-Mombasa Express'),
('payment', 1, 'Payment completed for Nairobi-Mombasa Express'),
('cancellation', 1, 'Cancelled booking for Nairobi-Mombasa Express'),
('booking', 2, 'Booked a ticket for Nairobi-Kisumu Highway'),
('payment', 2, 'Payment completed for Nairobi-Kisumu Highway'),
('cancellation', 2, 'Cancelled booking for Nairobi-Kisumu Highway'),
('update', 3, 'Updated booking for Mombasa-Nairobi Direct'),
('login', 4, 'Logged in to the system'),
('logout', 4, 'Logged out of the system');
```

```sql
INSERT INTO feedback (user_id, bus_id, rating, comments, feedback_date) VALUES
(1, 1, 5, 'Great trip, very comfortable.', '2024-11-19 10:30:00'),
(2, 2, 4, 'Good service, but could improve timing.', '2024-11-19 11:30:00'),
(3, 3, 5, 'Excellent trip, enjoyed the ride!', '2024-11-19 12:30:00'),
(4, 4, 3, 'Decent trip, but the bus was a bit delayed.', '2024-11-19 13:30:00'),
(5, 5, 5, 'Very smooth and comfortable journey.', '2024-11-19 14:30:00'),
(6, 6, 4, 'Good trip, though the bus could have been cleaner.', '2024-11-19 15:30:00'),
(7, 7, 5, 'Fantastic ride, highly recommend!', '2024-11-19 16:30:00'),
(8, 8, 3, 'The trip was okay, but seating arrangements were poor.', '2024-11-19 17:30:00'),
(9, 9, 5, 'Great service, will book again!', '2024-11-19 18:30:00'),
(10, 10, 4, 'Good trip, though the bus could have had more legroom.', '2024-11-19 19:30:00');
```

IMPLEMENTATION:

(i)CRUD OPERATIONS

(a) users_side_operations

```sql
use PROJECT_rev;

-- USER SIDE
/*
Main System Transaction
*/

-- Register
INSERT INTO users (first_name, last_name, email, password)
VALUES ('Lucy', 'Chepkemoi', 'lucychepkemoi@gmail.com', 'rtyuio');

-- Login
SELECT * FROM users
WHERE email = 'lucychepkemoi@gmail.com' AND password = 'rtyuio';

-- Reset Password
UPDATE users
SET password = 'fghjkl3'
WHERE email = 'lucychepkemoi@gmail.com';

-- Insert a reservation for Booking ID 1, Seat A2
INSERT INTO seat_reservations (booking_id, seat_number)
VALUES (2, 'A4');

-- Enter Passsenger(s) Details
INSERT INTO booking_tickets (user_id, route_id, date, time, price, seat_number)
VALUES (1, 1, '2024-11-20', '10:00:00', 1000.00, 'A2');
```

```sql
-- Pay For The Selected seats to be Booked
INSERT INTO payment_option (first_name, last_name, booking_id, payment_date, payment_status, mpesa_receipt_no)
VALUES ('Sharon', 'Chepkorir', 1, NOW(), 'confirmed', 'MPESA12345');

/*
Other User System Functionalities
*/

-- Edit User Profile
UPDATE users
SET first_name = 'Linda', last_name = 'mary', email = 'lindamarya@gmail.com'
WHERE user_id = 10;

-- Insert user feedback
INSERT INTO feedback (user_id, bus_id, rating, comments, feedback_date)
VALUES (1, 3, 5, 'The ride was very smooth, and the driver was professional.', NOW());


-- Cancel a Booking
-- Check if the booking exists and is valid
SELECT bt.booking_id, bt.user_id, bt.date, bt.price
FROM booking_tickets bt
WHERE bt.booking_id = 123 AND bt.user_id = 1;
```

```sql
-- Record the cancellation
INSERT INTO user_cancellations (user_id, booking_id, cancellation_date, cancellation_time, payment_id, cancellation_status)
SELECT
    1 AS user_id,
    123 AS booking_id,
    CURDATE() AS cancellation_date,
    CURTIME() AS cancellation_time,
    po.payment_id,
    'Cancelled' AS cancellation_status
FROM payment_option po
WHERE po.booking_id = 123;
select * from user_cancellations;

-- Search Bus
SELECT r.route_id, r.route_name, r.from_location, r.to_location, r.route_cost, b.reg_number
FROM routes r
JOIN buses b ON r.bus_id = b.bus_id
WHERE r.from_location = 'Nairobi' AND r.to_location = 'Mombasa';


-- Select Seat On the Bus Found
-- Check available seats on the selected route If no rows are returned, the seat is available.
SELECT sr.seat_number
FROM seat_reservations sr
JOIN booking_tickets bt ON sr.booking_id = bt.booking_id
WHERE bt.route_id = 1
  AND sr.seat_number = 'A2';
```

```sql
-- Print Ticket
SELECT bt.booking_id, u.first_name, u.last_name, r.route_name, r.from_location, r.to_location, r.route_cost, sr.seat_number
FROM booking_tickets bt
JOIN users u ON bt.user_id = u.user_id
JOIN routes r ON bt.route_id = r.route_id
JOIN seat_reservations sr ON bt.booking_id = sr.booking_id
WHERE bt.booking_id = 10;


-- View Booking History
SELECT bh.history_id, r.route_name, bh.day, bh.time, p.payment_status
FROM user_booking_history bh
JOIN booking_tickets bt ON bh.booking_id = bt.booking_id
JOIN routes r ON bt.route_id = r.route_id
JOIN payment_option p ON bh.payment_id = p.payment_id
WHERE bh.user_id = 1;

-- View Booked Trips
SELECT bt.trip_id, u.first_name, u.last_name, r.route_name, b.reg_number, d.first_name AS driver_name, bt.trip_date, bt.trip_time
FROM booked_trips bt
JOIN users u ON bt.user_id = u.user_id
JOIN routes r ON bt.route_id = r.route_id
JOIN buses b ON bt.bus_id = b.bus_id
JOIN drivers d ON bt.driver_id = d.driver_id;
```

(b)admin_side_operations

```sql
use PROJECT_rev;

-- ADMIN SIDE FUNCTIONALITIES
/*
Main Operations
*/
delete from users
where user_id=11;

-- Add a new bus
INSERT INTO buses (reg_number, max_passengers, driver_id)
VALUES ('KBC456k', 50, 2);

-- Update existing bus information
UPDATE buses
SET max_passengers = 60
WHERE bus_id = 1;


-- Add a new driver
INSERT INTO drivers (national_id, first_name, last_name, phone_number, email)
VALUES ('12345678', 'Alice', 'Brown', '0712345678', 'alice.brown@example.com');

-- Update driver details
UPDATE drivers
SET phone_number = '0723456789'
WHERE driver_id = 1;
```

```sql
-- View Booked Trips
SELECT bt.trip_id, u.first_name, u.last_name, r.route_name, b.reg_number, d.first_name AS driver_name, bt.trip_date, bt.trip_time
FROM booked_trips bt
JOIN users u ON bt.user_id = u.user_id
JOIN routes r ON bt.route_id = r.route_id
JOIN buses b ON bt.bus_id = b.bus_id
JOIN drivers d ON bt.driver_id = d.driver_id;
```

```sql
-- View all payments
SELECT p.payment_id, p.first_name, p.last_name, p.booking_id, p.payment_status, p.payment_date
FROM payment_option p;

-- View and Respond to User Feedbacks
SELECT f.feedback_id, u.first_name, u.last_name, b.reg_number, f.rating, f.comments, f.feedback_date
FROM feedback f
JOIN users u ON f.user_id = u.user_id
JOIN buses b ON f.bus_id = b.bus_id;

-- Audit System Logs
SELECT log_id, action_type, user_id, action_details, timestamp
FROM audit_logs
ORDER BY timestamp DESC;
```

```sql
/*
Other Operations
*/

-- Cancel Booking
-- Mark a booking as cancelled
INSERT INTO user_cancellations (user_id, booking_id, cancellation_date, cancellation_time, payment_id, cancellation_status)
VALUES (1, 123, CURDATE(), CURTIME(), 10, 'Cancelled');
-- Update payment status
UPDATE payment_option
SET payment_status = 'pending'
WHERE payment_id = 10;

--  View Available Routes
SELECT r.route_id, r.route_name, r.from_location, r.to_location, r.route_cost, b.reg_number
FROM routes r
JOIN buses b ON r.bus_id = b.bus_id;

--  Register New Admin
INSERT INTO admin (first_name,last_name, email, password)
VALUES ('Irene','Kilunda', 'ireene08@gmail.com', 'rtyui8');
select * from admin;

-- Assign Drivers to Buses
UPDATE buses
SET driver_id = 2
WHERE bus_id = 1;
```

```sql
-- Monitor Bus Ratings
-- Calculate average ratings for each bus
SELECT b.reg_number, AVG(f.rating) AS avg_rating
FROM feedback f
JOIN buses b ON f.bus_id = b.bus_id
GROUP BY b.bus_id
ORDER BY avg_rating DESC;


-- View and Respond to User Feedbacks
SELECT f.feedback_id, u.first_name, u.last_name, b.reg_number, f.rating, f.comments, f.feedback_date
FROM feedback f
JOIN users u ON f.user_id = u.user_id
JOIN buses b ON f.bus_id = b.bus_id;

-- Cancel Booking
-- Mark a booking as cancelled
INSERT INTO user_cancellations (user_id, booking_id, cancellation_date, cancellation_time, payment_id, cancellation_status)
VALUES (1, 123, CURDATE(), CURTIME(), 10, 'Cancelled');
-- Update payment status
UPDATE payment_option
SET payment_status = 'pending'
WHERE payment_id = 10;
```

```sql
--  View Available Routes
SELECT r.route_id, r.route_name, r.from_location, r.to_location, r.route_cost, b.reg_number
FROM routes r
JOIN buses b ON r.bus_id = b.bus_id;

-- Monitor Bus Ratings
-- Calculate average ratings for each bus
SELECT b.reg_number, AVG(f.rating) AS avg_rating
FROM feedback f
JOIN buses b ON f.bus_id = b.bus_id
GROUP BY b.bus_id
ORDER BY avg_rating DESC;
-- Cancel a Booking
-- Check if the booking exists and is valid
SELECT bt.booking_id, bt.user_id, bt.date, bt.price
FROM booking_tickets bt
WHERE bt.booking_id = 123 AND bt.user_id = 1;
```

ADVANCED QUERIES

```sql
-- Count Total Trips by Each Driver
SELECT
d.driver_id,
CONCAT(d.first_name, ' ', d.last_name) AS driver_name,
COUNT(bt.trip_id) AS total_trips
FROM
drivers d
JOIN
booked_trips bt ON d.driver_id = bt.driver_id
GROUP BY
d.driver_id, driver_name
ORDER BY
total_trips DESC;
```

```sql
-- List Buses with Average Feedback Ratings
SELECT
b.bus_id,
b.reg_number AS bus_registration,
COUNT(f.feedback_id) AS total_feedbacks,
AVG(f.rating) AS avg_rating
FROM
buses b
LEFT JOIN
feedback f ON b.bus_id = f.bus_id
GROUP BY
b.bus_id, b.reg_number
ORDER BY
avg_rating DESC;
```

```sql
    -- Find Users with Pending or Failed Payments
    SELECT
    u.user_id,
    CONCAT(u.first_name, ' ', u.last_name) AS full_name,
    po.payment_id,
    po.payment_status,
    po.payment_date
FROM
    users u
JOIN
    payment_option po ON u.user_id = po.payment_id
WHERE
    po.payment_status IN ('pending', 'failed')
ORDER BY
    po.payment_date DESC;
```

```sql
    -- Generate Cancellation Statistics by Route
    SELECT
    r.route_id,
    r.route_name,
    COUNT(uc.cancellation_id) AS total_cancellations,
    SUM(bt.price) AS total_loss_revenue
FROM
    routes r
JOIN
    booking_tickets bt ON r.route_id = bt.route_id
JOIN
    user_cancellations uc ON bt.booking_id = uc.booking_id
WHERE
    uc.cancellation_status = 'Cancelled'
GROUP BY
    r.route_id, r.route_name
ORDER BY
    total_cancellations DESC;
```

```sql
-- View Feedback with User and Driver Details
SELECT
f.feedback_id,
CONCAT(u.first_name, ' ', u.last_name) AS user_name,
b.reg_number AS bus_registration,
CONCAT(d.first_name, ' ', d.last_name) AS driver_name,
f.rating,
f.comments,
f.feedback_date
FROM
feedback f
JOIN
users u ON f.user_id = u.user_id
JOIN
buses b ON f.bus_id = b.bus_id
JOIN
drivers d ON b.driver_id = d.driver_id
ORDER BY
f.feedback_date DESC;
```

TESTING AND VALIDATION:

1. Unit Testing:

We tested the UNIQUE (email) in the users and drivers table by inserting both unique and duplicate email records and our results were successful for unique emails and fail for duplicate email hence maintaining data integrity as expected.

2.Integration Testing:

We confirmed the relationships between the users and booking table specifically focusing on Foreign Key constraints. We concluded that only existing users can make bookings, while non-existing users are rejected to make bookings.

This confirms that there is referential integrity constraint.

3.Load Testing:

We tested for load balancing (bottleneck control) by generating 1000 booking records simultaneously. The system performed well generally with an average response time of 0.8 seconds but we also observed minor performance degradation beyond 800 records, where response times increased up to 1.5 seconds.

CONCLUSIONS AND RECOMMENDATIONS:

Conclusions:

This online bus ticket booking system will provide numerous advantages such as enhancing customer satisfaction, smooth operations and increased revenue. This is because it allows customers to book tickets, select seats, and make payments online, at the comfort of their homes or offices. The system will also reduce the dependency of customers going to physical booking counters in order to buy a ticket hence making the process more convenient and reliable. Additionally, viewing of seat availability and booking data enables better resource management by reducing operational errors like overbooking.

Overall, the system meets modern customer expectations and will also strengthen the W Transport competitive position in the market.

Future recommendations:

Offer more Integrated Payment Options: such as credit cards, mobile wallets, and bank transfers, to cater to varied customer preferences.

Real-Time Notifications: Implement notification systems that update customers on booking confirmations, trip reminders, and any changes or delays in schedules.

Offer Promotions: Offer promotional discounts on different seasons to encourage more bookings in the future.

Incorporate Data Analytics: This will enable admins to track and understand the booking trends, popular routes.

REFERENCES:

Cosmas, N. I., Etus, C., Ajere, I. U., & Godswill, A. U. (2015). Online bus ticket reservation system. Int J Comput Sci Stat, 1(2).

Oloyede, M. O., Alaya, S. M., & Adewole, K. S. (2014). Development of an online bus ticket reservation system for a transportation service in Nigeria. Development, 5(12), 40-2.

Adam, T. (2019). Automated bus ticket reservation system for ethiopian bus transport system. IOSR Journal of Computer Engineering (IOSR-JCE), 21(3), 22-27.