UTM

UNIVERSITI TEKNOLOGI MALAYSIA

SECD2523

(DATABASE)

SECTION 10

PROJECT PHASE 3: DATABASE LOGICAL DESIGN

TITLE: GRAB STUDENT PREBET

GROUP NAME : QUATRON

PREPARED BY :

MUHAMMAD ALIF IZUDDIN BIN AZMAN	**A22EC0202**
MUHAMAD HAFIZ BIN MOHD FAUZI	**A22EC0196**
NIDA QISTINA BINTI ABD WAHID	**A22EC0231**
NURAISYA SALSABILA BINTI MOHD FADZAL	**A22EC0249**

TABLE OF CONTENT

1.0 INTRODUCTION

The title that we have proposed for our project is the Grab Student Prebet System. Grab Student Prebet services are well known among UTM students. This system has been a trusted companion for students throughout their ride, especially the first year students and those who have no transportation. However, as time flies, we realise the need for innovation to improve the user experience, leading to higher student satisfaction.

This Grab Student Prebet system offers services such as booking a car for a ride, selecting a desired driver that specialises in women, choosing the price range, picking the number of seats, and promoting a job application as a driver for the students who are interested in providing this service.

This Grab Student Prebet System started in a Telegram group where the admin of the group shared the link to the group among the students. It allows the students to find a ride from one place to another in the group. Not all students have their own transportation, especially first year students who are prohibited from bringing their own transportation to the campus. Since the price of the ride in the external grabcar depends on the peak hour and the weather conditions, the price of the ride provided by Grab Student Prebet is fixed up to the driver's desire and is still affordable for students. However, this system is not fully computerised, resulting in the disorganisation of the data.

Thus, the focus of this project is to suggest some improvements to the system to improve the user experience, leading to student and driver satisfaction and easing the process of car booking.

2.0 OVERVIEW OF PROJECT

The goal of the Grab Student Prebet System project is to transform the ride-sharing service system that is currently in use by many UTM students. The current system is effective in offering mobility options manually, especially to the first year students and those who do not have personal transportation.

This project seeks to improve the user experience by implementing a new system (the proposed system). The existing one is a manually controlled system, started in a Telegram group but it has had trouble keeping up with the increasing demand for ride services and efficiently managing data. This project is proposed to install a computerized data management system, enhance student-driver communication, and create an automated booking system. It also aims to improve overall happiness for drivers and students by customizing options for trips, ensuring no order is overlooked and expediting the booking process. The objective of the new system is to introduce a new level of efficiency and convenience in student transportation services.

3.0 DATABASE CONCEPTUAL DESIGN

3.1 UPDATED BUSINESS RULE

Registration & authentication

- All users must register with valid personal information, including a verified phone number to use the system.
- User authentication is mandatory before accessing any of the services in the system.

Booking processes

- Users must provide accurate and up-to-date information when booking a ride, including pickup location and destination.
- Booking information and driver details should be provided to the user in real-time.
- Users and drivers should adhere to the cancellation policy, with applicable charges for cancellations made after a certain timeframe.

Payment, invoicing, & fare

- All payments for the services must be processed through the system. Supporting multiple payment methods.
- Invoices should be generated automatically and made available to users after completing an order.
- Fare calculation must be transparent and based on a predetermined pricing model, including factors such as distance, time, and demand.

Driver standards

- Drivers must undergo a thorough background check and provide valid documentation, including driver's license and vehicle registration.
- Drivers are required to maintain a clean and well-maintained vehicle in compliance with local regulations.

Feedback & ratings

- Users and drivers should provide honest and constructive feedback after ride order has completed to maintain service quality
- Consistently low ratings or multiple negative feedbacks may result in account reviews or suspensions.

3.2 CONCEPTUAL ERD

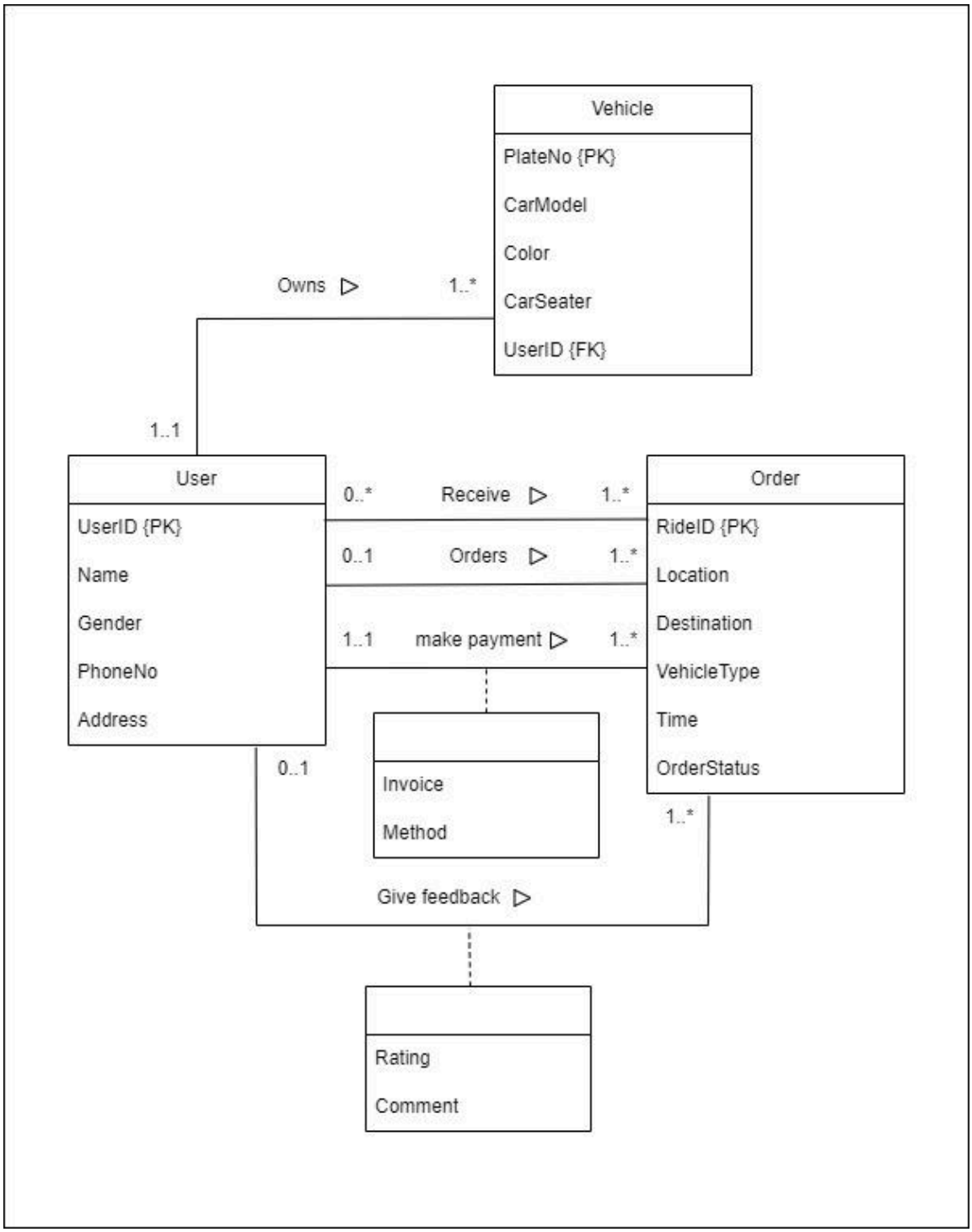


FIGURE 1 : CONCEPTUAL ERD

4.0 DATABASE LOGICAL DESIGN

4.1 LOGICAL ERD

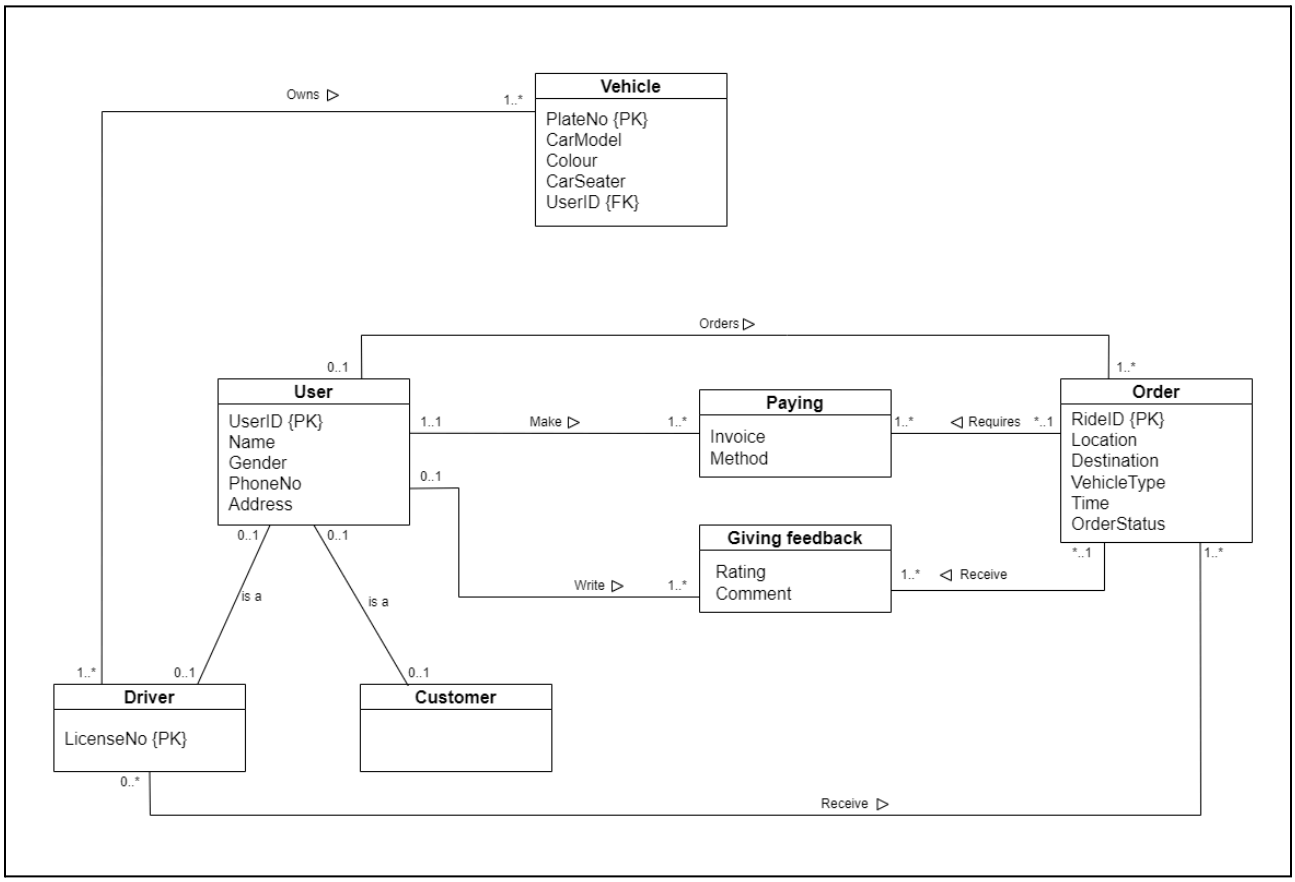


FIGURE 2 : LOGICAL ERD

4.2 UPDATED DATA DICTIONARY

4.2.1 Data dictionary that documents the entities

Entity name	Description	Occurrence
User	• A superclass to Driver and Customer entities • Hold both driver and customer data	Each user has a role where it can be identified from their ID.

Driver	• A subclass to User Entity. • Hold the driver's license number.	Driver has a license number. Driver owns a car. Driver receives the order.
Customer	A subclass to User Entity.	Customer orders Order Customer make payment Customer can write Feedback
Vehicle	Hold data of the driver's vehicle.	Every vehicle can be identified from the driver's vehicle plate number.
Order	Hold the data of the customer's booking.	Every order can be identified from the ride ID.
Paying	• A relationship entity. • Hold data of payment.	Every payment is made by the customer once the ride has finished.
Giving Feedback	• A relationship entity. • Hold the data of customer's feedback.	Every customer gives feedback once the ride has finished.

4.2.2 Data dictionary showing the description of relationship

Entity name	**Multiplicity**	**Relationship**	**Entity name**	**Multiplicity**
User	0..1	As a	Customer	0..1

	0..1	As a	Driver	0..1
	0..1	Orders	Order	1..*
	1..1	Make	Paying	1..*
	0..1	Write	Giving Feedback	1..*
Order	*..1	Requires	Paying	1..*
	*..1	Receive	Giving Feedback	1..*
Driver	1..*	Owns	Vehicle	1..*
	0..*	Receive	Order	1..*

4.2.3 Data dictionary showing the description of attributes

Entity name	**Attributes**	**Description**	**Data type & length**	**Nulls**	**Multi-valued**
User	UserID	Uniquely identifies user	VARCHAR2 (9)	No	No
	Name	Name of user	VARCHAR2 (100)	No	No
	Gender	Gender of user	CHAR(1)	No	No

	Phone No	Telephone number of user	VARCHAR2 (11)	No	No
	Address	College address of User	VARCHAR2 (250	No	No
Order	RideID	Uniquely identifies order	VARCHAR2 (6)	No	No
	Location	Location of user	VARCHAR2 (100)	No	No
	Destination	Destination of customer	VARCHAR2 (100)	No	No
	VehicleType	Vehicle type of driver	NUMBER (1)	No	No
	Time	Time of order	TIMESTAMP	No	No
	OrderStatus	Status of customer's order	VARCHAR2 (10)	No	No
Vehicle	PlateNo	Plate number of vehicle	VARCHAR2 (8)	No	No
	CarModel	Car model of vehicle	VARCHAR2 (20)	No	No
	Color	Color of vehicle	VARCHAR2 (15)	No	No
	CarSeater	Number of car seat of	NUMBER (1)	No	No

		vehicle			
	UserID	Uniquely identifies driver	VARCHAR2 (9)	No	No
Payment	Invoice	Invoice of payment	VARCHAR2 (100)	No	No
	Method	Method of payment	VARCHAR2 (20)	No	No
	UserID	Uniquely identifies driver	VARCHAR2 (9)	No	No
	RideID	Uniquely identifies order	VARCHAR2 (6)	No	No
Feedback	Rating	Rating of ride feedback	NUMBER(5)	Yes	No
	UComment	Comment of ride feedback	VARCHAR2 (50)	Yes	No
	UserID	Uniquely identifies driver	VARCHAR2 (9)	No	No
	RideID	Uniquely identifies order	VARCHAR2 (6)	No	No

4.3 NORMALIZATION

UNF

1. Vehicle (PlateNo, CarModel, Colour, CarSeater, UserID)

   PK: PlateNo

   FK1: UserID reference User(UserID)

2. User (UserID, Name, Gender, PhoneNo, Address)

   PK: UserID

3. Paying (Invoice, Methode, UserID, RideID)

   PK: UserID, RideID

   FK1: UserID reference User(UserID)

   FK2: RideID reference Order(RideID)

4. GivingFeedback (Rating, Comment, UserID, RideID)

   PK: UserID, RideID

   FK1: UserID reference User(UserID)

   FK2: RideID reference Order(RideID)

5. Order (RideID, Location, Destination, VehicleType, Time, OrderStatus)

   PK: RideID

6. DriverUser (LicenseNo, UserID, Name, Gender, PhoneNo, Address)

   PK: LicenseNo, UserID

   FK1 : UserID reference User(UserID)

7. CustomerUser (UserID, Name, Gender, PhoneNo, Address)

   PK: UserID

   FK1: UserID reference User(UserID)

Functional Dependency

Vehicle:

1. FD1: PK : PlateNo → CarModel, Colour, CarSeater, UserID

User:

1. FD2: PK : UserID → Name, Gender, PhoneNo, Address

Paying:

1. FD3: PK : UserID, RideID → Invoice, Methode

GivingFeedback:

1. FD5: PK : UserID, RideID → Rating, Comment

Order:

1. FD6: PK : RideID → Location, Destination, VehicleType, Time, OrderStatus

DriverUser:

1. FD8: PK : LicenseNo, UserID → Name, Gender, PhoneNo, Address
2. FD9: Partial Dependency : UserID → Name, Gender, PhoneNo, Address

CustomerUser:

1. FD10: PK : UserID → Name, Gender, PhoneNo, Address

1NF

1. Vehicle (PlateNo, CarModel, Colour, CarSeater, UserID)

   PK: PlateNo

   FK1: UserID reference User(UserID)

2. User (UserID, Name, Gender, PhoneNo, Address)

   PK: UserID

3. Paying (Invoice, Methode, UserID, RideID)

   PK: UserID, RideID

   FK1: UserID reference User(UserID)

   FK2: RideID reference Order(RideID)

4. GivingFeedback (Rating, Comment, UserID, RideID)

   PK: UserID, RideID

   FK1: UserID reference User(UserID)

   FK2: RideID reference Order(RideID)

5. Order (RideID, Location, Destination, VehicleType, Time, OrderStatus)

   PK: RideID

6. DriverUser (LicenseNo, UserID, Name, Gender, PhoneNo, Address)

   PK: LicenseNo, UserID

   FK1 : UserID reference User(UserID)

7. CustomerUser (UserID, Name, Gender, PhoneNo, Address)

   PK: UserID

   FK1: UserID reference User(UserID)

2NF

1. Vehicle (PlateNo, CarModel, Colour, CarSeater, UserID)

   PK: PlateNo

   FK1: UserID reference User(UserID)

2. User (UserID, Name, Gender, PhoneNo, Address)

PK: UserID

3. Paying (Invoice, Methode, UserID, RideID)

   PK: UserID, RideID

   FK1: UserID reference User(UserID)

   FK2: RideID reference Order(RideID)

4. GivingFeedback (Rating, Comment, UserID, RideID)

   PK: UserID, RideID

   FK1: UserID reference User(UserID)

   FK2: RideID reference Order(RideID)

5. Order (RideID, Location, Destination, VehicleType, Time, OrderStatus)

   PK: RideID

6. DriverUser (LicenseNo, UserID, Name, Gender, PhoneNo, Address)

   6.1 From FD9: UserDetails (UserID, Name, Gender, PhoneNo, Address)

      PK: UserID

      FK1: UserID reference User(UserID)

   6.2 The rest: DriverUser (LicenseNo, UserID)

      PK: LicenseNo, UserID

      FK1 : UserID reference User(UserID)

7. CustomerUser (UserID, Name, Gender, PhoneNo, Address)

   PK: UserID

   FK1: UserID reference User(UserID)

BCNF

1. Vehicle (PlateNo, CarModel, Colour, CarSeater, UserID)

   PK: PlateNo

   FK1: UserID reference User(UserID)

2. User (UserID, Name, Gender, PhoneNo, Address)

   PK: UserID

3. Paying (Invoice, Method, UserID, RideID)

   PK: UserID, RideID

   FK1: UserID reference User(UserID)

   FK2: RideID reference Order(RideID)

4. GivingFeedback (Rating, Comment, UserID, RideID)

   PK: UserID, RideID

   FK1: UserID reference User(UserID)

   FK2: RideID reference Order(RideID)

5. Order (RideID, Location, Destination, VehicleType, Time, OrderStatus)

   PK: RideID

6. UserDetails (UserID, Name, Gender, PhoneNo, Address)
7. DriverUser (LicenseNo, UserID)

   PK: LicenseNo, UserID

   FK1 : UserID reference User(UserID)

8. CustomerUser (UserID, Name, Gender, PhoneNo, Address)

   PK: UserID

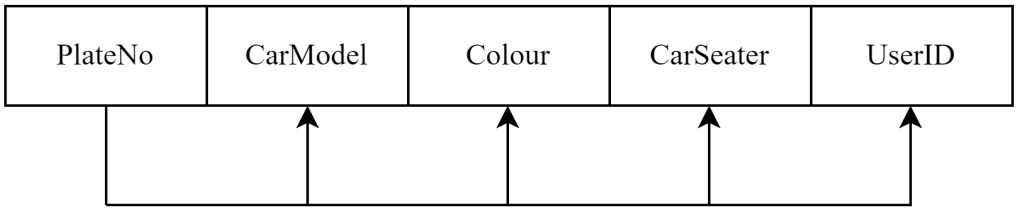
   FK1: UserID reference User(UserID)

5.0 RELATIONAL DATABASE SCHEMAS

Since there is only partial dependency (PD) and no transitive dependency (TD) in all relations, therefore, the relations Vehicle, User, Paying, GivingFeedback, Order, UserDetails, DriverUser and CustomerUser are already in BCNF relations**.**

1. **VEHICLE (<u>PlateNo</u>, CarModel, Colour, CarSeater, <u>UserID</u>)**

   VEHICLE



   PK : PlateNo

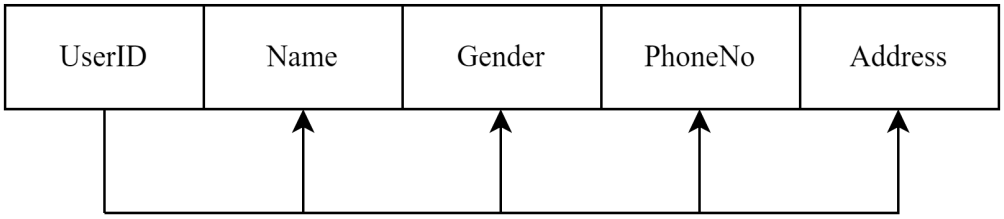
   FK1 : UserID reference USER(UserID)

   FD1 : PlateNo → CarModel, Colour, CarSeater, UserID

   FD1 is full functional dependency. Relation VEHICLE is in BCNF

2. **USER (<u>UserID</u>, Name, Gender, PhoneNo, Address)**

   USER



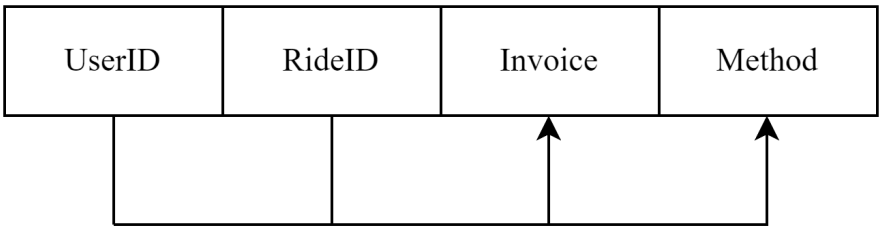
PK : UserID

FD2 : UserID → Name, Gender, PhoneNo, Address

FD2 is full functional dependency. Relation USER is in BCNF

3. **PAYING (Invoice, Method, <u>UserID</u>, <u>RideID</u>)**

PAYING



PK : UserID, RideID

FK1 : UserID reference to USER (UserID)

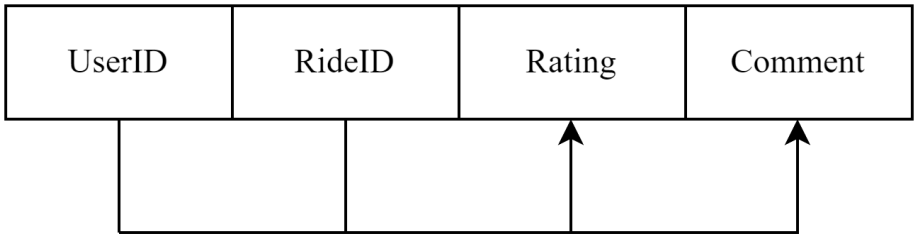
FK2 : RideID reference to ORDER (RideID)

FD3 : UserID, RideID → Invoice, Method

FD3 is full functional dependency. Relation PAYING is in BCNF

4. **GIVINGFEEDBACK (Rating, Comment, <u>UserID</u>, <u>RideID</u>)**

GIVINGFEEDBACK



PK : UserID, RideID

FK1 : UserID reference USER (UserID)

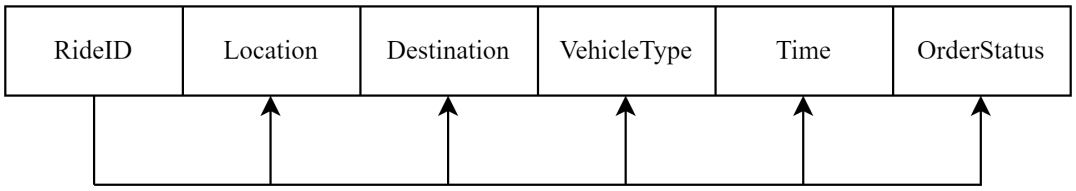
FK2 : RideID reference to ORDER (RideID)

FD5 : UserID, RideID → Rating, Comment

FD5 is full functional dependency. Relation GIVINGFEEDBACK is in BCNF

5. **ORDER (<u>RideID</u>, Location, Destination, VehicleType, Time, OrderStatus)**

ORDER



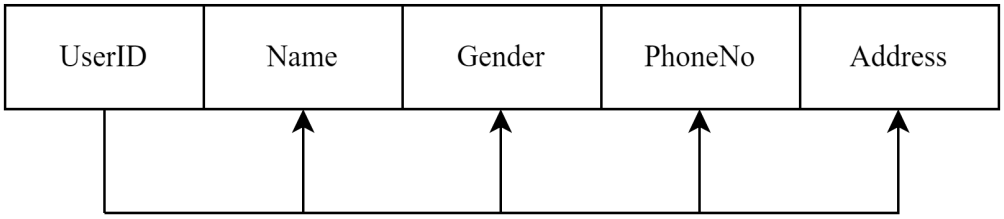
PK : RideID

FD6 : RideID → Location, Destination, VehicleType, Time, OrderStatus

FD6 is full functional dependency. Relation ORDER is in BCNF

6. **USERDETAILS (<u>UserID</u>, Name, Gender, PhoneNo, Address)**

USERDETAILS



PK : UserID

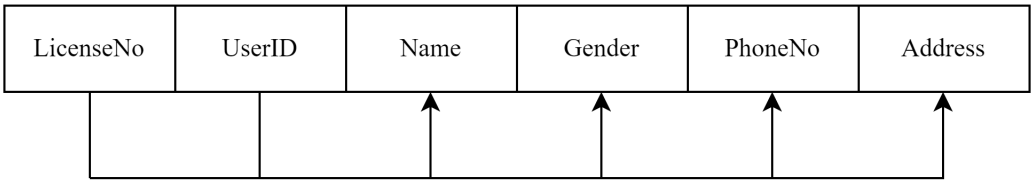
FK1 : UserID reference User(UserID)

FD2 : UserID → Name, Gender, PhoneNo, Address

FD2 is full functional dependency. Relation USER is in BCNF

7. **DRIVERUSER (<u>LicenseNo</u>, <u>UserID</u>)**

DRIVERUSER



PK : LicenseNo, UserID

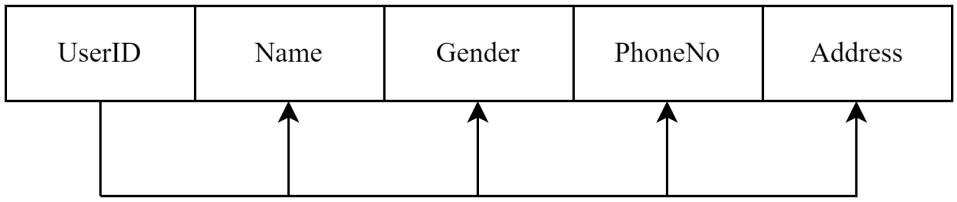
FK1 : UserID reference User(UserID)

FD8 : LicenseNo, UserID → Name, Gender, PhoneNo, Address

FD9 : Partial Dependency (PD) : UserID → Name, Gender, PhoneNo, Address

8. **CUSTOMERUSER (<u>UserID</u>, Name, Gender, PhoneNo, Address)**

CUSTOMERUSER



PK : UserID

FK1 : UserID reference User(UserID)

FD10 : UserID → Name, Gender, PhoneNo, Address

FD10 is full functional dependency, Relation CUSTOMERUSER is in BCNF

6.0 SQL STATEMENTS

6.1 CREATE DATABASE AND TABLE

```
CREATE TABLE USER_INFO
(
	UserID VARCHAR2(9) PRIMARY KEY,
	Name VARCHAR2(100) NOT NULL,
	PhoneNo VARCHAR2(11) NOT NULL,
	Address VARCHAR2(250) NOT NULL,
	Gender CHAR(1) NOT NULL
);


CREATE TABLE ORDER_INFO
(
	RideID VARCHAR2(6) PRIMARY KEY,
	Location VARCHAR2(100) NOT NULL,
	Destination VARCHAR2(100) NOT NULL,
	VehicleType NUMBER(1) NOT NULL,
	Time TIMESTAMP NOT NULL,
	OrderStatus VARCHAR2(10) NOT NULL
);
```

```
CREATE TABLE VEHICLE
(
      PlateNo VARCHAR2(8) PRIMARY KEY,
      CarModel VARCHAR2(20) NOT NULL,
      Color VARCHAR2(15) NOT NULL,
      CarSeater NUMBER(1) NOT NULL,
      UserID VARCHAR2(9),
      FOREIGN KEY (UserID) REFERENCES USER_INFO (UserID)
);

CREATE TABLE PAYMENT
(
      UserID VARCHAR2(9),
      RideID VARCHAR2(6),
      Invoice VARCHAR2(100) NOT NULL,
      Method VARCHAR2(20) NOT NULL,
      PRIMARY KEY (UserID, RideID),
      FOREIGN KEY (UserID) REFERENCES USER_INFO (UserID),
      FOREIGN KEY (RideID) REFERENCES ORDER_INFO (RideID)
);
```

```
CREATE TABLE FEEDBACK
(
     UserID VARCHAR2(9),
     RideID VARCHAR2(6),
     Rating NUMBER(5),
     UComment VARCHAR2(50),
     PRIMARY KEY (UserID, RideID),
     FOREIGN KEY (UserID) REFERENCES USER_INFO (UserID),
     FOREIGN KEY (RideID) REFERENCES ORDER_INFO (RideID)
);


CREATE TABLE DRIVER
(
    UserID VARCHAR2(9) PRIMARY KEY,
    LicenseNumber VARCHAR2(20) NOT NULL,
    FOREIGN KEY (UserID) REFERENCES USER_INFO(UserID)
);
```

6.2 TEST DATA (SQL-DML)

INSERT USER_INFO - DRIVER

```
INSERT INTO USER_INFO VALUES ('A22EC0345', 'ALI JOHN',
'0112233445', 'MA1 KTDI UTM', 'M');

INSERT INTO USER_INFO VALUES ('A22EM0290', 'NATALIA ABU',
'0162232445', 'L12 KTHO UTM', 'F');

INSERT INTO USER_INFO VALUES ('A22EM2546', 'AIMAN HAKIMI',
'0135564736', 'K10 UTM', 'M');

INSERT INTO USER_INFO VALUES ('A22EE1196', 'FARAH NAJIHA',
'0197654434', 'MA7 KTDI UTM', 'F');

INSERT INTO USER_INFO VALUES ('A22EM4325', 'JOHAN HUZAIRI',
'01734224564', 'M23 KTDI UTM', 'M');
```

INSERT USER_INFO – CUSTOMER

```
INSERT INTO USER_INFO VALUES ('A23SC0221', 'NUR QAMALIA',
'01139879876', 'M27 KTDI UTM', 'F');

INSERT INTO USER_INFO VALUES ('B21EC0998', 'ALIF IZUDDIN',
'0145671234', 'UB2 KOLEJ 10 UTM', 'M');

INSERT INTO USER_INFO VALUES ('A23EE0876', 'FEKRE FADHLEE',
'0187654321', 'MA1 KTDI UTM', 'M');

INSERT INTO USER_INFO VALUES ('A22EA1453', 'RUBI ABU',
'0155678345', 'K04 KTR UTM', 'F');

INSERT INTO USER_INFO VALUES ('B21EC0021', 'SITI IZZAH',
'0120987890', 'UA2 KOLEJ 9 UTM', 'F');

INSERT INTO USER_INFO VALUES ('B20SC0300', 'NUR AMALINA',
'01142345643', 'S12 KTC UTM', 'F');
```

```
INSERT INTO USER_INFO VALUES ('A23SS0091', 'ABUJA MOHD', '0129876543', 'S05 KTC UTM', 'M');

INSERT INTO USER_INFO VALUES ('A21EM0167', 'ADAM RAZI', '0104328769', 'G17 KRP UTM', 'M');

INSERT INTO USER_INFO VALUES ('A22EA0042', 'NUR MAISARAH', '01117652348', 'UA2 KOLEJ 9 UTM', 'F');

INSERT INTO USER_INFO VALUES ('A21EC0005', 'HAFIZ HARITH', '0156879009', 'G18 KRP UTM', 'M');
```

INSERT VEHICLE

```
INSERT INTO VEHICLE VALUES ('SAC7877', 'PERODUA MYVI', 'SILVER', '4', 'A22EC0345');

INSERT INTO VEHICLE VALUES ('RAF4356', 'PERODUA BEZZA', 'BLUE', '4', 'A22EM0290');

INSERT INTO VEHICLE VALUES ('WJS5643', 'PROTON EXORA', 'WHITE', '6', 'A22EM2546');

INSERT INTO VEHICLE VALUES ('BJC6755', 'HONDA CITY', 'RED', '4', 'A22EE1196');

INSERT INTO VEHICLE VALUES ('JTY5554', 'HYUNDAI STAREX', 'BLACK', '6', 'A22EM4325');
```

INSERT ORDER_INFO

```
INSERT INTO ORDER_INFO VALUES ('A12345', 'L50 UTM', 'ARKED LESTARI', 4, SYSTIMESTAMP, 'ACCEPTED');

INSERT INTO ORDER_INFO VALUES ('A11234', 'P19 UTM', 'ARKED ANGKASA', 6, SYSTIMESTAMP, 'ACCEPTED');

INSERT INTO ORDER_INFO VALUES ('A22012', 'C25 UTM', 'AEON TAMAN UNIVERSITI', 4, SYSTIMESTAMP, 'REJECTED');

INSERT INTO ORDER_INFO VALUES ('A87654', 'MSI', 'M46', 4, SYSTIMESTAMP, 'REJECTED');

INSERT INTO ORDER_INFO VALUES ('A00001', 'KDSE', 'P19', 6, SYSTIMESTAMP, 'ACCEPTED');

INSERT INTO ORDER_INFO VALUES ('A02023', 'PKU UTM', 'TAMAN UNIVERSITI', 4, SYSTIMESTAMP, 'REJECTED');

INSERT INTO ORDER_INFO VALUES ('A22223', 'K9', 'L50 UTM', 4, SYSTIMESTAMP, 'ACCEPTED');
```

INSERT FEEDBACK

```
INSERT INTO FEEDBACK VALUES ('A23SC0221', 'A12345', 4, NULL);

INSERT INTO FEEDBACK VALUES ('B21EC0998', 'A11234', 5, NULL);

INSERT INTO FEEDBACK VALUES ('A21EC0005', 'A00001', 3, 'The ride was awful');

INSERT INTO FEEDBACK VALUES ('B20SC0300', 'A22223', 5, 'Good service and friendly', );
```

INSERT PAYMENT

```
INSERT INTO PAYMENT VALUES (UserID, RideID, Invoice, Method)
SELECT
'A23SC0221',
'A12345',
'FROM' || Location || 'TO' || Destination || ', RM6.50',
'Online Transfer'
FROM ORDER_INFO
WHERE RideID = 'A12345';
```

```
INSERT INTO PAYMENT VALUES (UserID, RideID, Invoice, Method)
SELECT
'B21EC0998',
'A11234',
'FROM' || Location || 'TO' || Destination || ', RM7.00',
'QR Pay'
FROM ORDER_INFO
WHERE RideID = 'A11234';
```

```
INSERT INTO PAYMENT VALUES (UserID, RideID, Invoice, Method)
SELECT
'A21EC0005',
'A00001',
'FROM' || Location || 'TO' || Destination || ', RM10.00',
```

```
'QR Pay'
FROM ORDER_INFO
WHERE RideID = 'A00001';

INSERT INTO PAYMENT VALUES (UserID, RideID, Invoice, Method)
SELECT
'B20SC0300',
'A22223',
'FROM' || Location || 'TO' || Destination || ', RM7.30',
'QR Pay'
FROM ORDER_INFO
WHERE RideID = 'A22223';

INSERT DRIVER
INSERT INTO DRIVER (UserID, LicenseNumber) VALUES ('A22EC0345', '55647897');
INSERT INTO DRIVER (UserID, LicenseNumber) VALUES ('A22EM0290', '12345678');
INSERT INTO DRIVER (UserID, LicenseNumber) VALUES ('A22EM2546', '98765432');
INSERT INTO DRIVER (UserID, LicenseNumber) VALUES ('A22EE1196', '87654321');
INSERT INTO DRIVER (UserID, LicenseNumber) VALUES ('A22EM4325', '45678901');
```

Retrieve records with USING clause

```
SELECT O.LOCATION, O.DESTINATION, F.RATING, F.UCOMMENT
FROM FEEDBACK F JOIN ORDER_INFO O
ON F.RIDEID = O.RIDEID;
```

Three-Way Joins with ON clause

```
SELECT O.LOCATION, O.DESTINATION, F.RATING, F.UCOMMENT, P.METHOD
FROM FEEDBACK F JOIN ORDER_INFO O
ON F.RIDEID = O.RIDEID
JOIN PAYMENT P
ON O.RIDEID = P.RIDEID;
```

7.0 SUMMARY

The problems that the UTM PREBET service on the Telegram platform was having can be fully resolved by using the Grab Student PREBET system. The system attempts to provide an effective, economical, and user-friendly environment for both drivers and riders through automation, real-time communication features, and careful consideration of operational, technical, and economic factors.

In addition to dealing with the manual processing of customer requests, the system offers advantages like an improved user experience, increased operational effectiveness, and possible cost savings. To optimise the beneficial effects on the UTM PREBET service, it is critical to follow a clearly defined project plan, keep an eye on the adoption process, and guarantee ongoing improvement based on user feedback as the suggested solution is put into practice.