Title: Public Transport Booking System

Group No: 1
Lab Section: 02G
Lecturer: Miss Nur Farahaina Binti Idris

	Name	Matric No
1	Lim Joey	SD21038
2	Chong Jia Xin	SD21039
3	Melissa Faqihah binti Rusydan	SD21034
4	Aqila Sofea binti Abd Manaf	SD21006
5	Tean Jin He	SD21063

Project Background

Today's inhabitants prefer to use public transportation thanks to city globalisation. Public transportation provides people with mobility and access to employment, community resources, medical care, and recreational opportunities in communities.

All public transportation, including buses, trains, and subways, is intended to simplify life. It is crucial to the lives of locals and tourists alike in cities and towns all around the world, whether travelling to work, going shopping, popping into town to meet with friends or getting home safely after a good night out, using public transport can be less stressful and more relaxing than driving. At the same time, public transportation also helps to reduce road congestion and travel times, air pollution, and energy and oil consumption, all of which benefit both riders and non-riders alike. This is so people may save time by avoiding traffic jams and money on today's pricey petroleum by using public transportation. Public transportation networks lay the groundwork for communities to become more liveable and wealthier in several ways, from enhanced community health to affordability.

In Malaysia, the traditional method which is purchasing tickets through a physical bus station or terminal station is a more preferred method by citizens. However, there are a few disadvantages if passengers purchase tickets offline. First and foremost, as a passenger, they can’t purchase tickets immediately after the counter closed. This brings a lot of difficulties to those who want to take the earlier bus trip if they have an emergency to settle. Besides, if the public transport has been delayed or cancelled, the passenger that arrived at the waiting area has no way but to book another ticket. Last but not least, if they have any questions to ask after the operating hours of the customer service counter, they have no people to guide them.

That’s why, to guarantee the flawless operation of public transportation, a comprehensive and user-friendly booking system is extremely important. By creating a public transport booking system, passengers that are affected by the disadvantages of traditional physical ticket purchasing systems no longer need to worry about their journey since it operates 24 hours, with a Frequently Asked Questions (FAQ) interface. A good booking system can provide smooth experiences for purchasing tickets, contacting customer services, or comparing public transport routes for customers.

Objectives

1. To propose an effective method of a public transportation Database Management System

2. To provide major improvements in public transport convenience and safety.

3. To collect information regarding the passengers, staff, and companies.

Scopes:

1. Public transporter operators (system administrators):
   - Able to add, edit and retrieve information and generate reports to assist them with their daily operations.
   - Able to provide users with a registration interface for users to retrieve the latest news of public transport.
   - Able to provide different kinds of payment methods that are convenient for the users.
   - Able to provide an interface for passengers who would like to seek help from customer service after operating hours.

2. Passengers
   - Able to utilize the DBMS web portal to perform their transaction of purchasing public transport tickets at their own hassle-free time.

   - Able to receive the latest updates from the booking system.

Case study comparison (Jx)

Manual Process:

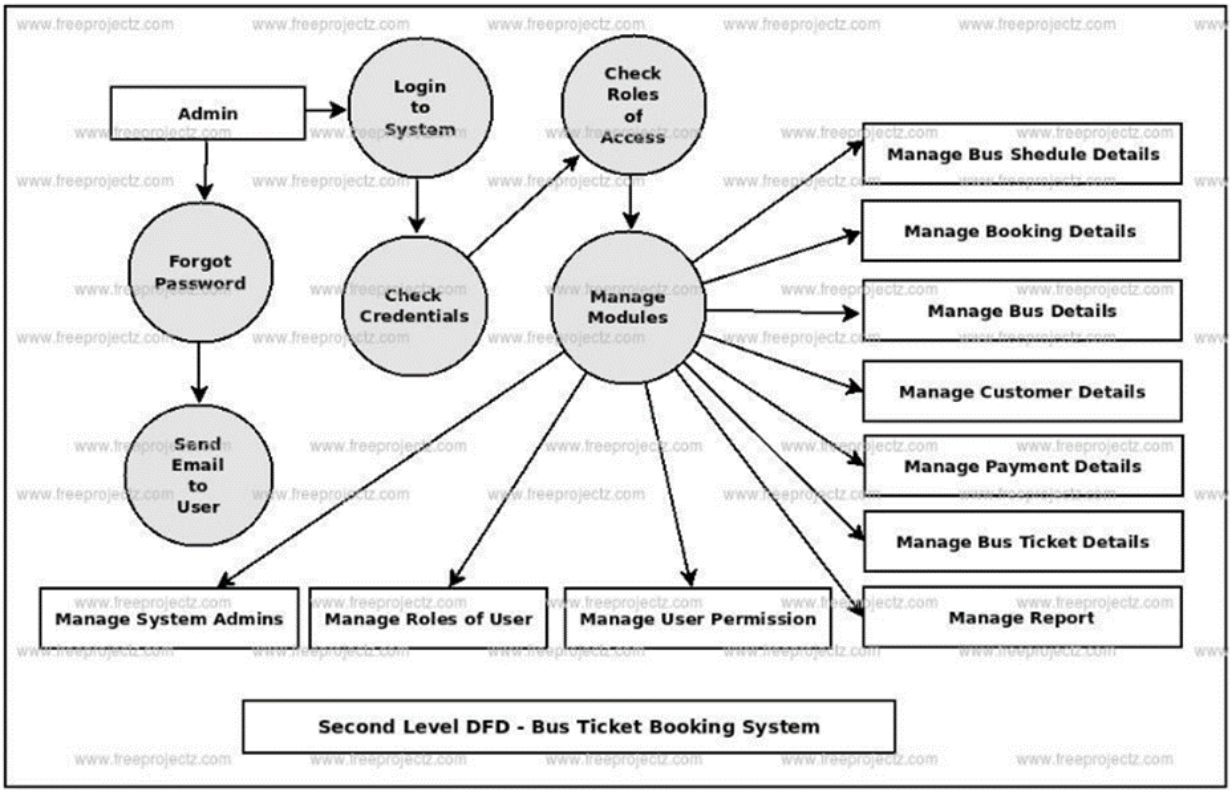


Figure 1: Flowchart Diagram of Bus Ticket Booking System

The Flowchart Diagram of Bus Ticket Booking System shows the process of an admin to accessing the system. This system needs to login as a user and key in the password by the user. If the user forgot his password, the admin of the system will send an e-mail to the user. The temporary password will be given and sent in few seconds in the form of e-mail. The user later may reset another new password that created by himself. Otherwise, if the user can login to the system, the user is able to continue the steps to check the credentials and check the roles of access. After that, the user will enter the manage modules to review or to do the management needed. There are 10 choices of the modules listed, which are System Admins, Roles of User, User Permission, Bus Schedule Details, Booking Details, Bus Details, Customer Details, Payment Details, Bus Ticket Details and Report.

Case Study 1:

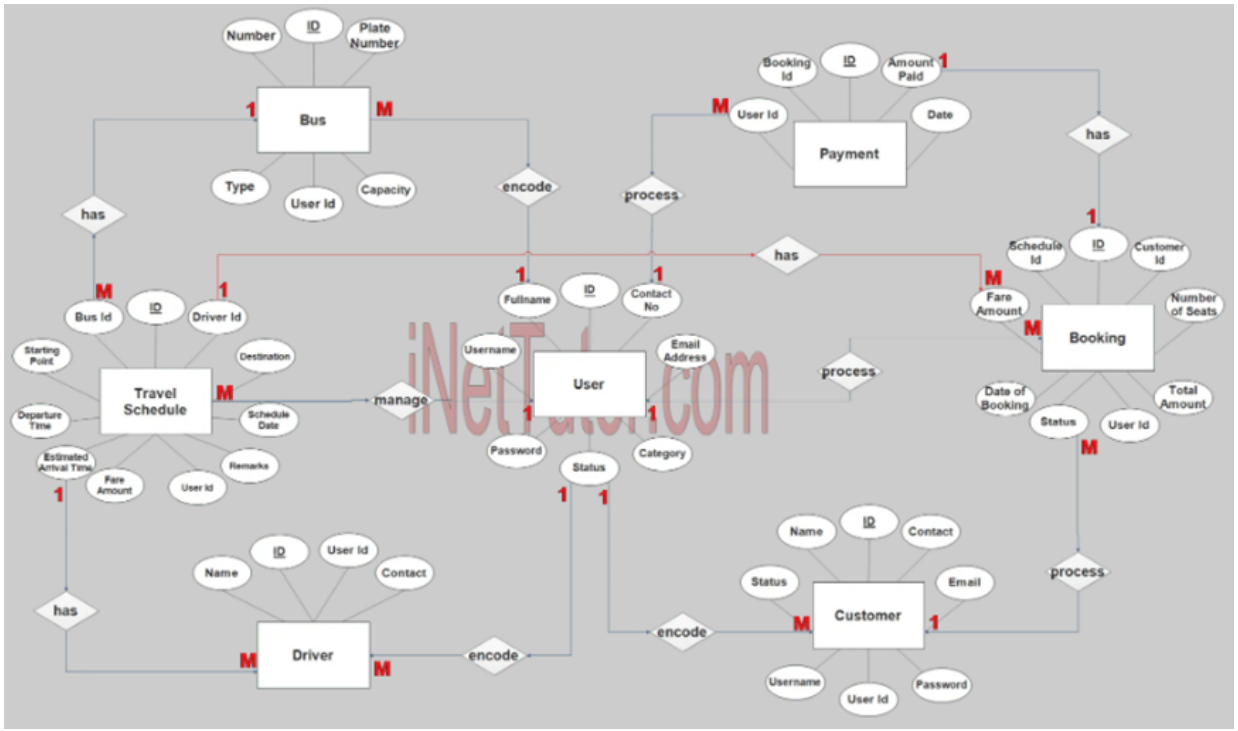


Figure 1: Diagram Case Study 1 ERD of Bus Booking System

The ERD Diagram of Case study 1 shows a database system that manages bus ticket booking. The database consists of a web-based system and a database. The aim of this system is to assist the bus company to manage the data such as travel schedule data, user data, driver data, customer data, booking data, payment data, and bus data. The ERD has 7 entities and 51 attributes including the primary key and foreign key. When the user log into the bus ticket booking system, the user may key in his details such as Fullname, Password, Status, and Contact No (User). The system will set up a User ID by the order. The entity of User is connected to others entities which are Bus, Travel Schedule, Driver, Customer, Booking, and Payment by each foreign key. To book a ticket bus, the user is able to view and choose a date, number of seats, and determine the total amount of the booking set (Booking). The booking entity records the customer id and schedule id. At the same time, the customer's details will be recorded despite one customer or more than one customer in a booking ticket (Customer). After booking the bus ticket, the user needs to make the payment according to the amount paid shown by the system (Payment). The user may do many booking and payment. On the other hand, the system will have the record of the transport tool which is the bus (Bus), bus driver (Driver), and the travel schedule of each bus (Travel Schedule). The bus entity will record the data such as plate number, bus type, and capacity of a bus. The travel schedule entity also records the Driver ID and Bus ID. Each bus has its own travel schedule which included the starting point, destination, estimated

arrival time, departure time, and scheduled date. The user can review the information needed before and after booking the bus ticket.

Case study 2:

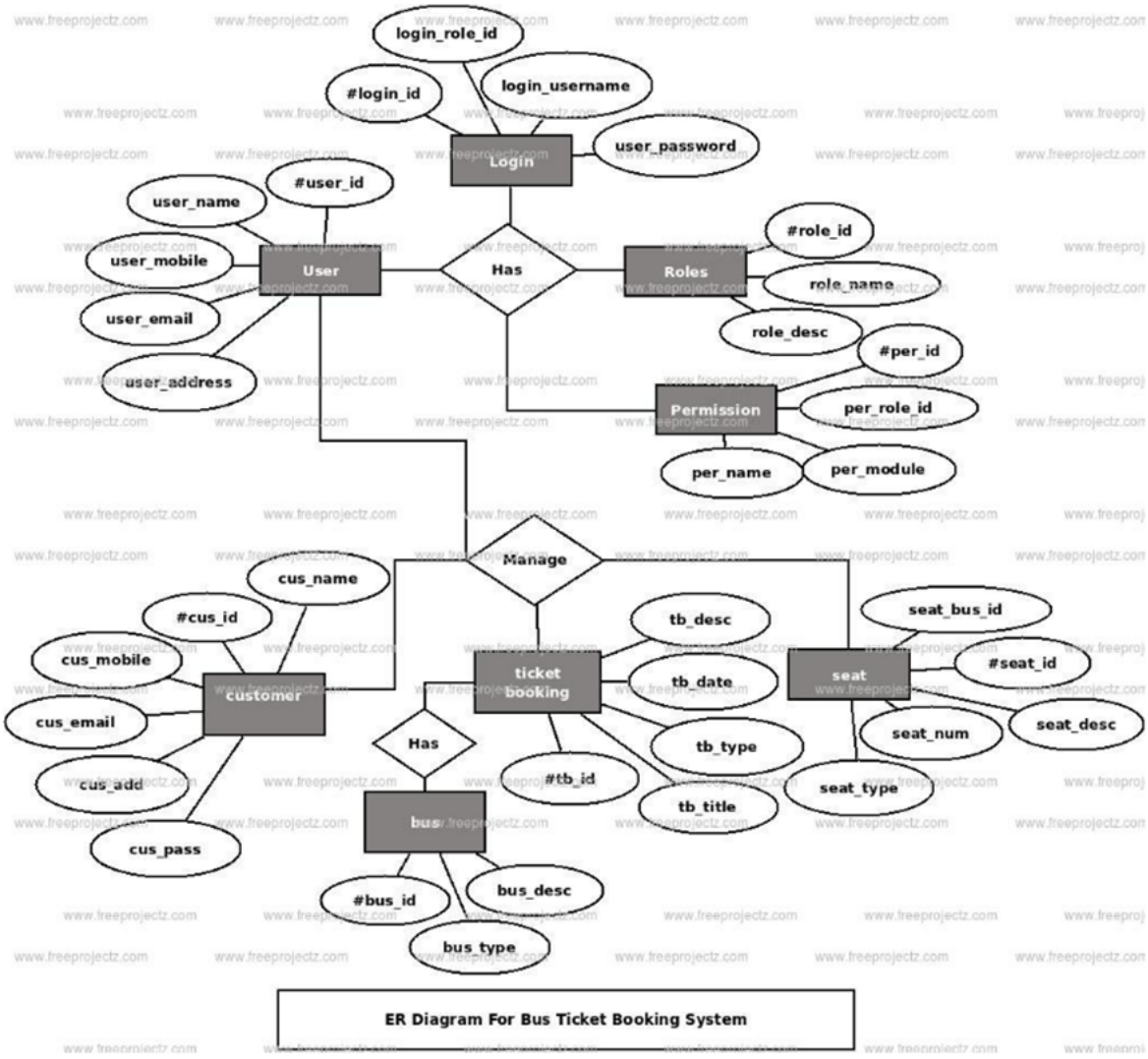


Figure 2: Diagram Case Study 2 ERD of Bus Ticket Booking System

The ERD Diagram of Case study 2 also shows a database system that managing the bus ticket booking. The aim of this system is to assist the bus company to manage the data such as user, login, roles, permission, customer, bus, ticket, and seat. The ERD has 8 entities, and 35 attributes included the primary key and foreign key. The difference of both ERD Diagrams is the Case Study 1 pays attention on the travel schedule of the buses while the Case Study 2 pays attention on the seats booked of the bus. To enter the Bus Ticket Booking System, the user needs

to log in by enter his username and password and the record will be saved in the entity Login (Login). The booking system records the user's details such as name, mobile, email, and address in the User entity (User). Other than that, the roles (Roles) and permission (Permission) data also recorded. Next, to manage the all the seats booked by the customers, the number of seats, seat type, and bus id have been recorded(seat). The customer(customer) and bus(bus) data stored in each entity. Finally, the entity ticket booking created to store the booking date, booking type and so on (ticket booking).

	Manual Process	Case Study 1	Data System 2
Process/entities	1. 'Admin' to 'Forget Password' to 'Send Email to User' 2. 'Admin' to 'Login to System' to 'Check Credentials' to 'Check Roles of Access' to 'Manage Modules'	1. User 2. Bus 3. Travel Schedule 4. Driver 5. Booking 6. Payment 7. Customer	1. User 2. Login 3. Roles 4. Permission 5. customer 6. bus 7. ticket booking 8. seat

Attributes	Manage Modules 1. Manage System Admins 2. Manage User Permission 3. Manage Roles of User 4. Manage Bus Schedule Details 5. Manage Booking Details 6. Manage Bus Details 7. Manage Customer Details 8. Manage Payment Details 9. Manage Bus Ticket Details 10. Manage Report	1. User ● ID ● Fullname ● Contact ● Email address ● Username ● Password ● Category ● Status 2. Bus ● ID ● Number ● Plate Number ● Type ● Capacity ● User ID 3. Travel Schedule ● ID ● Driver ID ● Bus ID ● Starting Point ● Destination ● Departure Time ● Estimated Arrival Time ● Schedule Date ● Fare Amount ● Remarks ● User ID 4. Driver ● ID ● Name ● Contact ● User ID	1. User ● user_id ● user_name ● user_mobile ● user_email ● user_address 2. Login ● login_id ● login_role_id ● login_username ● user_password 3. Roles ● role_id ● role_name ● role_desc 4. Permission ● per_id ● per_role_id ● per_module ● per_name 5. customer ● cus_id ● cus_name ● cus_mobile ● cus_email ● cus_add ● cus_pass 6. ticket booking ● tb_id ● tb_title ● tb_type ● tb_data ● tb_desc

		5. Booking ● ID ● Customer ID ● Schedule ID ● Number of Seats ● Fare Amount ● Date of Booking ● Total Amount ● Status ● User ID 6. Payment ● ID ● Booking ID ● Amount Paid ● Date ● User ID 7. Customer ● ID ● Name ● Contact ● Status ● Email ● Username ● Password ● User ID	7. bus ● bus_id ● bus_type ● bus_desc 8. seat ● seat_id ● seat_bus_id ● seat_desc ● seat_num ● seat_type

Primary keys/Foreign keys		1. User ● PK: ID 2. Bus ● PK: ID ● FK: User ID 3. Travel Schedule ● PK: ID ● FK: Driver ID ● FK: Bus ID ● FK: User ID 4. Driver ● PK: ID ● FK: User ID 5. Booking ● PK: ID ● FK: Customer ID ● FK: Schedule ID ● FK: User ID 6. Payment ● PK: ID ● FK: Booking ID ● FK: User ID 7. Customer ● PK: ID ● FK: User ID	1. User ● PK: user_id 2. Login ● PK: login_id ● FK: user_password 3. Roles ● PK: role_id 4. Permission ● PK: per_id 5. customer ● PK: cus_id 6. ticket booking ● PK: tb_id 7. bus ● PK: bus_id 8. seat ● PK: seat_id

References	https://www.freeprojectz.com/dfd/bus-ticket-booking-system-dataflow-diagram	https://www.inettutor.com/diagrams/bus-booking-system-er-diagram/	https://www.freeprojectz.com/entity-relationship/bus-ticket-booking-system-er-diagram

Attributes and Primary Key (Joey)

Entity	**Attributes**	**Primary Key**
User	user_name user_contactnumber user_address user_email paymentID	userID
Service	service_type service_contactnumber service_email service_description bookingID userID	serviceID
Booking Ticket	booking_date booking_type booking_destination booking_description userID paymentID	bookingID

Payment	payment_transactiondate payment_totalprice payment_unit payment_description payment_method	paymentID
Staff	staff_name staff_contactnumber staff_email transportID companyID	staffID
Company	company_name company_contactnumber company_description	companyID
Transport	transport_num transport_type transport_location transport_description companyID	transportID

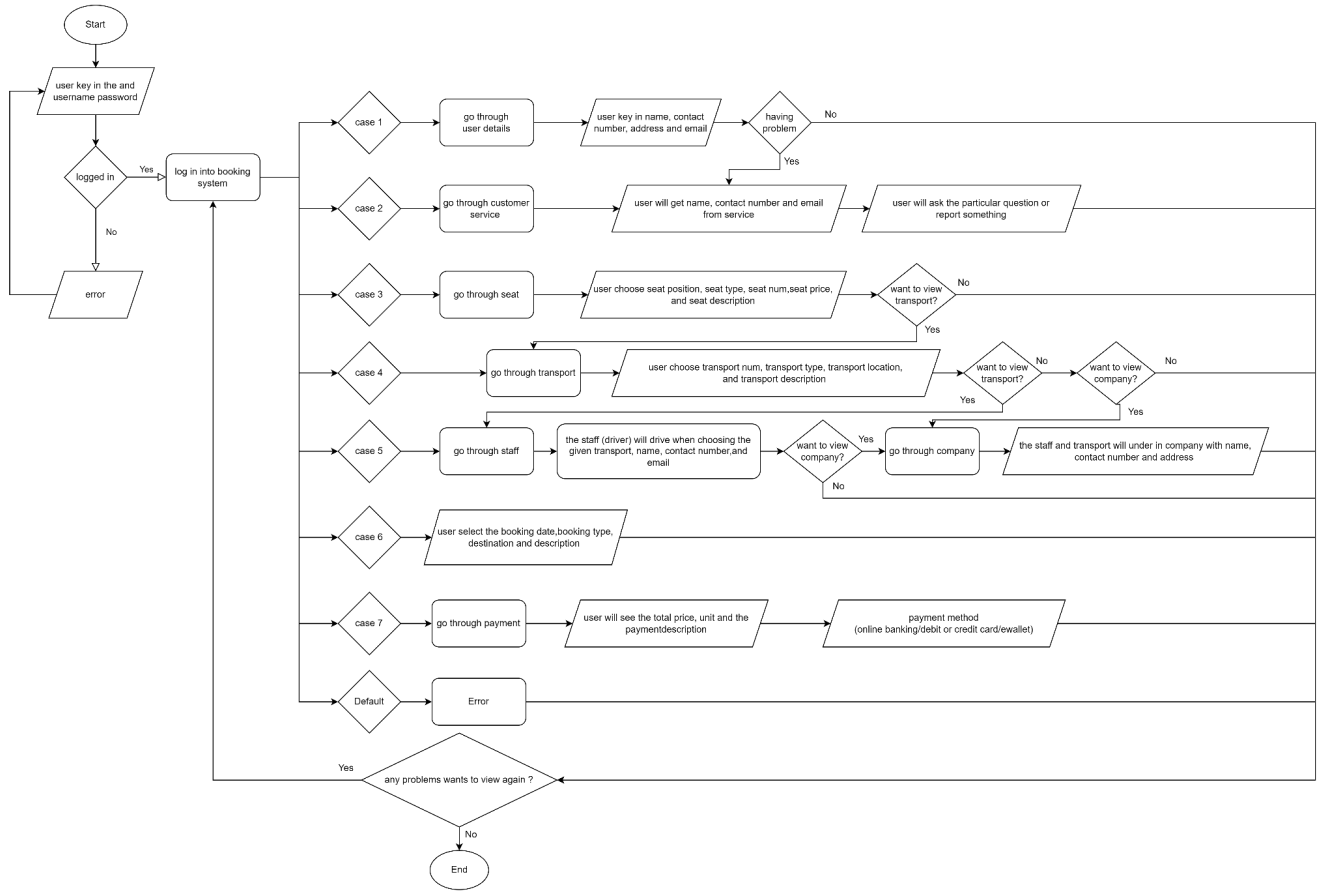
Seat	seat_position seat_num seat_type seat_price seat_description bookingID transportID	seatID
Details	working_time	bookingID staffID
Class	operation_hours	companyID bookingID
Orders	count_journey	seatID transportID

Business Rule and Relationship (Joey)

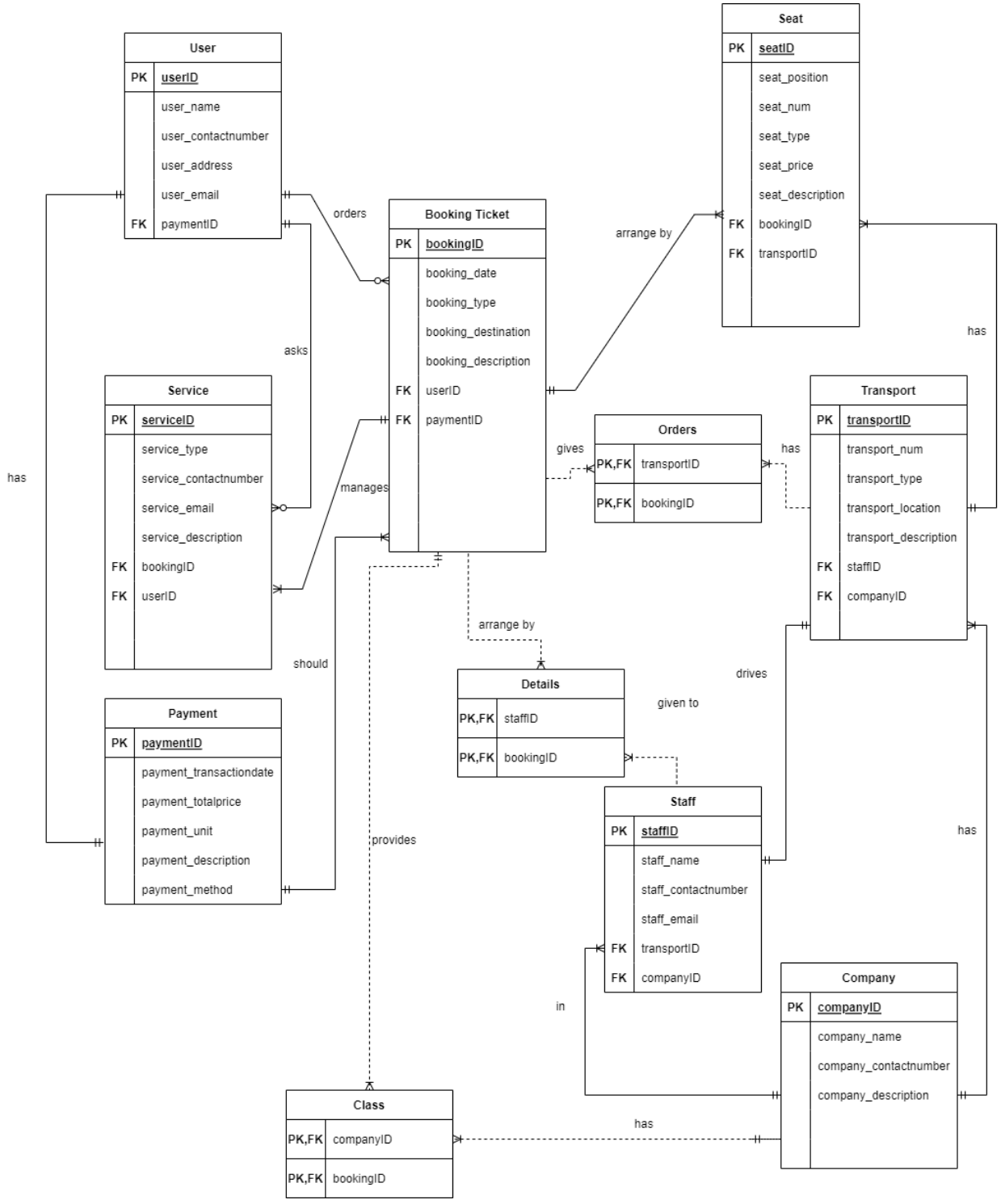
Entity (Data type)	Entity	Relationship	Business Rules
User	Booking Ticket	1:M	- One user can order many booking tickets - One booking ticket can only ordered by one user
User	Payment	1:1	- One user can only have one payment at a time - A payment can proceed by one user at a time
User	Service	1:M	- One user can ask for many services
Booking Ticket	Service	1:M	- One booking ticket managed by many services
Booking Ticket	Seat	1:M	- One seat can only arrange for one booking ticket
Transport	Orders	M:1	- One transport can have many orders
Orders	Seat	1:M	- One order can give many seats.
Transport	Staff	1:1	- One transport can drive by one staff at a time

			- One staff can drive one transport at a time
Transport	Company	M:1	- One transport can only have one company - One company can have many transports
Staff	Company	M:1	- One staff can only be in one company - One company can have many staffs
Orders	Transport	M:1	- One transport has many orders
Details	Staff	M:1	- One staff gives many details
Details	Booking Ticket	M:1	- One booking ticket is arranged by many details
Details	Staff	M:1	- One staff has given many details
Class	Booking Ticket	M:1	- One class provides many booking tickets
Class	Company	1:M	- One class has one company - One company has many classes

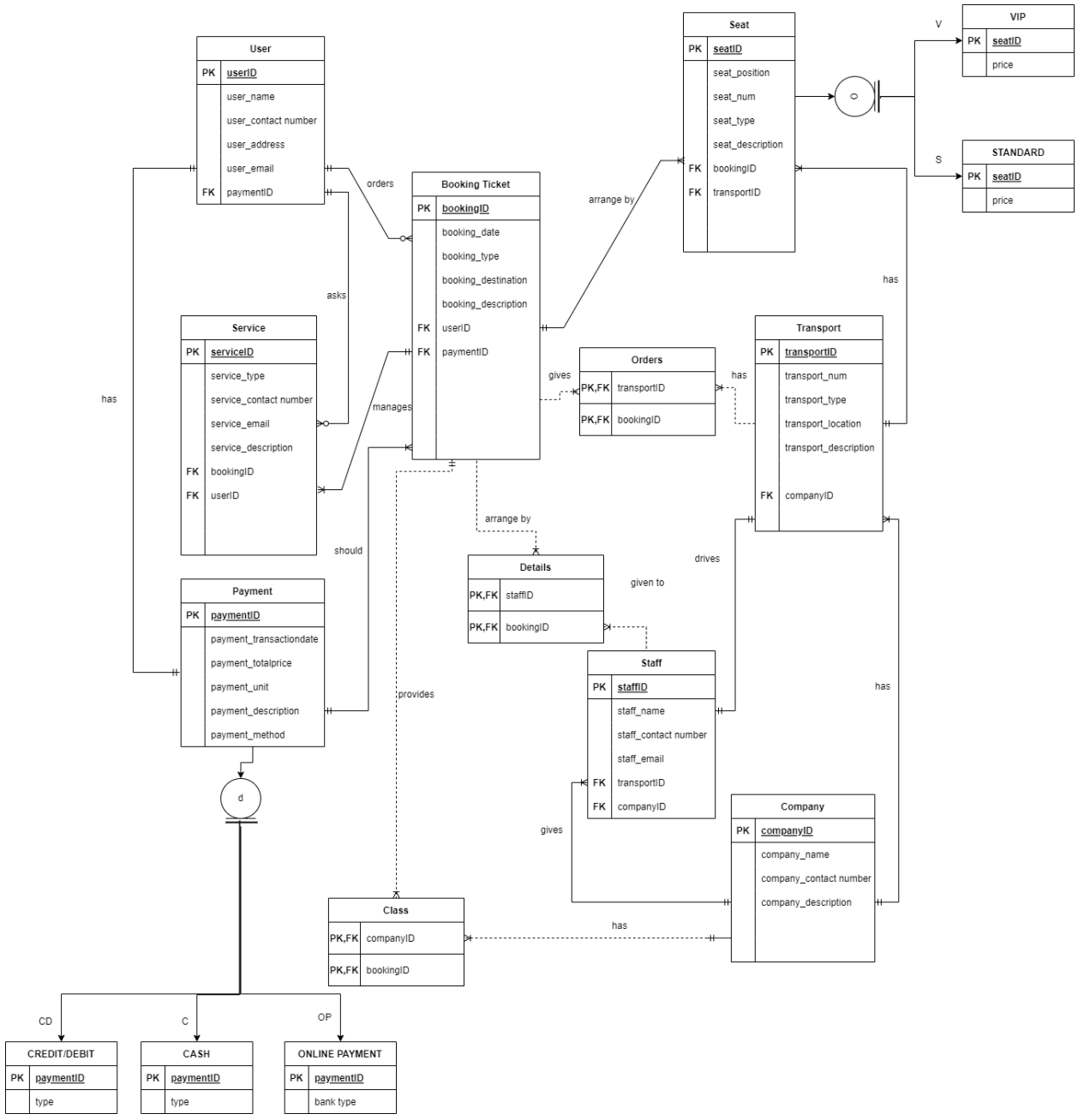
Manual Process (Flowchart) (Jinhe)
https://app.diagrams.net/#G19T7XV9fceigOyJ4rAkZzE0gI30XcFWS6



ERD (Jinhe) https://app.diagrams.net/#G1mskF9Q-HA9vPtRfbLIOadzlKBcwtgsAU



EERD (Jinhe) https://app.diagrams.net/#G1FSQl_plpNAybAPhY56rEG7Xl7KdACyrI



Dataset (Jx)

create database transportbooking;

use transportbooking;

```
create table payment
(
paymentID varchar(3) not null primary key,
payment_transactiondate date,
payment_totalprice decimal(5,2),
payment_unit int,
payment_description varchar(50),
payment_method varchar(50)
);

insert into payment values
('P01', '2022-12-20','20.00', '1', 'Payment by cash', 'cash'),
('P02', '2022-12-25','10.00', '1', 'Payment by cash', 'cash'),
('P03', '2022-12-21','20.00', '2', 'Payment by online banking', 'online banking'),
('P04', '2022-12-26','20.00', '1', 'Payment by online banking', 'online banking'),
('P05', '2022-12-18','20.00', '1', 'Payment by credit/debit', 'credit/debit'),
('P06', '2022-12-18','15.00', '1', 'Payment by credit/debit', 'credit/debit'),
('P07', '2022-12-19','45.00', '3', 'Payment by credit/debit', 'credit/debit'),
('P08', '2022-12-22','25.00', '1', 'Payment by TnG e-wallet', 'TnG e-wallet'),
('P09', '2022-12-23','25.00', '1', 'Payment by TnG e-wallet', 'TnG e-wallet'),
('P10', '2022-12-24','25.00', '1', 'Payment by TnG e-wallet', 'TnG e-wallet');
```

create table user

(

userID varchar(3) not null primary key,

user_name varchar(50),

user_contactnumber varchar(20),

user_address varchar(50),

user_email varchar(50),

paymentID varchar(3), foreign key (paymentID) references payment(paymentID)

);

insert into user values

('U01', 'Siti Sarah', '0112333232', 'Lot 114, Kampung Peramu Jaya, Pekan', 'sarah@gmail.com', 'P01'),

('U02', 'Ahmad Rizal', '0192332265', '37, Indera Mahkota Jaya, Kuantan', 'rizal@gmail.com', 'P02'),

('U03', 'Ong Tai Chee', '0114214576','31, Peramu Permai, Pekan', 'taichee@hotmail.com', 'P03'),

('U04', 'Sridevi Chopra', '0153442785', 'Lot 3323, Bukit Setongkol, Kuantan', 'sridevi@gmail.com','P04'),

('U05', 'Kamal Azri', '0131433241', '44, Taman Balok Jaya, Kuantan', 'kamal@gmail.com', 'P05'),

('U06', 'Cathy Ong', '0122115321', 'Lot 1141, Bukit Istana, Kuantan', 'cathy@gmail.com', 'P06'),

('U07', 'Gurmit Singh', '0114215436', '213, Taman Permai Jaya, Kuantan','gurmit@hotmail.com', 'P07'),

('U08', 'Lim Pei Ling', '0132124322', '32, Tanjung Lumpur Indah, Kuantan', 'peiling@gmail.com', 'P08'),

('U09', 'Anna Cameron', '0121170985', 'Lot 912, Taman Kuala Pahang, Pekan','anna@gmail.com', 'P09'),

('U10', 'Sally Ong', '0112252455', 'Lot 123, Taman Permai Jaya, Kuantan', 'sally@gmail.com', 'P10');

create table bookingticket

(

bookingID varchar(3) not null primary key,

booking_date date,

booking_type varchar(20),

booking_destination varchar(50),

booking_description varchar(50),

userID varchar(3), foreign key (userID) references user(userID),

paymentID varchar(3), foreign key (paymentID) references payment(paymentID)

);

insert into bookingticket values

('B01', '2023-01-11', 'Go', 'Johor Bahru', null,'U01','P01'),

('B02', '2023-01-12', 'Go', 'Seremban', null,'U02','P02'),

('B03', '2023-01-13', 'Go', 'Bandaraya Melaka', null,'U03','P03'),

('B04', '2023-01-13', 'Go', 'Bandaraya Melaka', null,'U03','P03'),

('B05', '2023-01-14', 'Go', 'Shah Alam',null,'U04','P04'),

('B06', '2023-01-15', 'Go', 'Kuantan', null,'U05','P05'),

('B07', '2023-01-16', 'Back', 'Ipoh', null,'U06','P06'),

```
('B08', '2023-01-17', 'Back', 'Alor Setar', null,'U07','P07'),
('B09', '2023-01-17', 'Back', 'Alor Setar', null,'U07','P07'),
('B10', '2023-01-17', 'Back', 'Alor Setar', null,'U07','P07'),
('B11', '2023-01-18', 'Back', 'George Town', null, 'U08','P08'),
('B12', '2023-01-19', 'Back', 'Kota Bharu', null, 'U09','P09'),
('B13', '2023-01-20', 'Back', 'Kangar', null,'U10','P10');

create table service
(
serviceID varchar(3) not null primary key,
service_type varchar(50),
service_contactnumber varchar(20),
service_email varchar(50),
service_description varchar(50),
bookingID varchar(3), foreign key (bookingID) references bookingticket(bookingID),
userID varchar(3), foreign key (userID) references user(userID)
);

insert into service values
('S01','booking error','0112252455', 'sally@gmail.com',null,'B01','U01'),
('S02','booking error','0112252455', 'sally@gmail.com',null,'B10','U10'),
('S03', 'transaction error', '0148872455','aminah@gmail.com', null,'B01','U01'),
('S04', 'transaction error', '0148872455','aminah@gmail.com', null,'B10','U10'),
```

```
('S05', 'change date','0125049889', 'sarra@hotmail.com',null,'B02','U02'),
('S06','change destination ', '0113322055', 'lisa@gmail.com',null,'B03','U03'),
('S07', 'change seat','0152252020', 'adamhakim@hotmail.com',null,'B06','U06'),
('S08', 'change company', '0123308877', 'devi@yahoo.com',null,'B07','U07'),
('S09', 'change staff', '0142275565', 'farah@gmail.com',null,'B08','U08'),
('S10', 'change transport', '0134449776', 'ravi@hotmail.com',null,'B09','U09'),
('S11', 'change user name', '0123192365', 'aiman@gmail.com',null,'B05','U05'),
('S12', 'change user email', '0155592201', 'brianlee@yahoo.com',null,'B05','U05');

 create table company
(
companyID varchar(3) not null primary key,
company_name varchar(50),
company_contactnumber varchar(20),
company_description varchar(50)
);

insert into company values
 ('C01', 'BusA Company', '0123456789', 'provide 1 bus'),
 ('C02', 'BusB Company', '0134567891', 'provide 1 bus'),
 ('C03', 'BusC Company', '0145678912', 'provide 1 bus'),
 ('C04', 'BusD Company', '0156789123', 'provide 1 bus'),
 ('C05', 'VanA Company', '0167891234', 'provide 1 van'),
```

```
('C06', 'VanB Company', '0178912345', 'provide 1 van'),
('C07', 'VanC Company', '0189123456', 'provide 1 van'),
('C08', 'VanD Company', '0191234567', 'provide 1 van'),
('C09', 'TrainA Company', '0101234567', 'provide 1 train'),
('C10', 'TrainB Company', '0123456789', 'provide 1 train');

create table transport
(
transportID varchar(3) not null primary key,
transport_num varchar(7),
transport_type varchar(10),
transport_location varchar(50),
transport_description varchar(50),
companyID varchar(3), foreign key (companyID) references company(companyID)
);

insert into transport values
('T01', 'JJH8888', 'Bus', 'Johor Bahru', 'Bus to Johor Bahru', 'C01'),
('T02', 'NJH8888', 'Bus', 'Seremban', 'Bus to Seremban', 'C02'),
('T03', 'MJH8888', 'Bus', 'Bandaraya Melaka', 'Bus to Bandaraya Melaka', 'C03'),
('T04', 'BJH8888', 'Bus', 'Shah Alam', 'Bus to Shah Alam', 'C04'),
('T05', 'CJH8888', 'Van', 'Kuantan', 'Van to Kuantan', 'C05'),
('T06', 'AJH8888', 'Van', 'Ipoh', 'Van to Ipoh', 'C06'),
```

```
('T07', 'KJH8888', 'Van', 'Alor Setar', 'Van to Alor Setar', 'C07'),
('T08', 'PJH8888', 'Van', 'George Town', 'Van to George Town', 'C08'),
('T09', 'DJH8888', 'Train', 'Kota Bharu', 'Train to Kota Bharu', 'C09'),
('T10', 'RJH8888', 'Train', 'Kangar', 'Train to Kangar', 'C10');

create table staff
(
staffID varchar(3) not null primary key,
staff_name varchar(50),
staff_contactnumber varchar(20),
staff_email varchar(50),
transportID varchar(3), foreign key (transportID) references transport(transportID),
companyID varchar(3), foreign key (companyID) references company(companyID)
);

insert into staff values
('F01', 'Ali', '0112333232', 'ali@gmail.com', 'T01','C01'),
('F02', 'Abu', '0192332265', 'abu@gmail.com', 'T02','C02'),
('F03', 'Muthu', '0114214576', 'muthu@hotmail.com', 'T03','C03'),
('F04', 'Kabou', '0153442785', 'kabou@gmail.com', 'T04','C04'),
('F05', 'Keely', '0131433241', 'keely@gmail.com', 'T05','C05'),
('F06', 'Stacy', '0122115321', 'stacy@gmail.com', 'T06','C06'),
('F07', 'Eliane', '0114215436', 'eliane@hotmail.com',null,null),
```

```
('F08', 'Joe', '0132124322', 'joe@gmail.com', 'T08','C08'),
('F09', 'Joel', '0121170985', 'joel@gmail.com','T09','C09'),
('F10', 'Zoey', '0112333232', 'zoey@gmail.com', 'T10','C10');

create table seat
(
seatID varchar(3) not null primary key,
seat_position varchar(20),
seat_num varchar(5),
seat_type varchar(20),
seat_price decimal(5,2),
seat_description varchar(50),
bookingID varchar(3), foreign key (bookingID) references bookingticket(bookingID),
transportID varchar(3), foreign key (transportID) references transport(transportID)
);

insert into seat values
('E01', 'Right', 'BA001', 'Standard', '20.00', 'Window seat and no snack','B01', 'T01'),
('E02', 'Middle', 'BB003', 'Standard', '10.00', 'no snack','B02', 'T02'),
('E03', 'Middle', 'BC003', 'Standard', '10.00', 'no snack','B03', 'T03'),
('E04', 'Middle Left', 'BC004', 'Standard', '10.00', 'no snack','B04', 'T03'),
('E05', 'Left', 'VB005', 'Standard', '20.00', 'Window seat and no snack','B05', 'T06'),
('E06', 'Left', 'VC005', 'Standard', '20.00', 'Window seat and no snack','B06', 'T07'),
```

```
('E07', 'Middle Right', 'VD002', 'VIP', '15.00', 'provide snack','B07', 'T08'),
('E08', 'Middle Right', 'VC002', 'VIP', '15.00', 'provide snack','B08', 'T07'),
('E09', 'Middle ', 'VC003', 'VIP', '15.00', 'provide snack','B09', 'T07'),
('E10', 'Middle Left', 'VC004', 'VIP', '15.00', ' provide snack','B10', 'T07'),
('E11', 'Left', 'VD005', 'VIP', '25.00', 'Window seat and provide snack','B11', 'T08'),
('E12', 'Left', 'TA005', 'VIP', '25.00', 'Window seat and provide snack','B12', 'T09'),
('E13', 'Left', 'TB005', 'VIP', '25.00', 'Window seat and provide snack','B13', 'T10');

create table orders
(
seatID varchar(3), foreign key (seatID) references seat(seatID),
transportID varchar(3),  foreign key (transportID) references transport(transportID),
count_journey int
);

insert into orders values
('E01','T01','10'),
('E02','T02','12'),
('E03','T03','16'),
('E04','T03','16'),
('E05','T06','18'),
('E06','T07','17'),
('E07','T08','15'),
```

('E08','T07','11'),

('E09','T07','11'),

('E10','T07','11'),

('E11','T08','13'),

('E12','T09','14'),

('E13','T10','20');

create table class

(

bookingID varchar(3), foreign key (bookingID) references bookingticket(bookingID),

companyID varchar(3), foreign key (companyID) references company(companyID),

operation_hour int

);

insert into class values

('B01','C01','22'),

('B02','C02','18'),

('B03','C03','20'),

('B04','C03','20'),

('B05','C04','18'),

('B06','C05','17'),

('B07','C06','12'),

('B08','C07','10'),

('B09','C07','10'),

('B10','C07','10'),

('B11','C08','11'),

('B12','C09','15'),

('B13','C10','15');

create table details

(

bookingID varchar(3), foreign key (bookingID) references bookingticket(bookingID),

staffID varchar(3), foreign key (staffID) references staff(staffID),

working_time varchar(20)

);

insert into details values

('B01','F01','9:00a.m.-5:00p.m.'),

('B02','F02','12:00a.m.-4:00a.m.'),

('B03','F03','4:00p.m.-9:00p.m.'),

('B04','F03','4:00p.m.-9:00p.m.'),

('B05','F04','12:00a.m.-4:00a.m.'),

('B06','F05','1:00p.m.-5:00p.m.'),

('B07','F06','12:00a.m.-5:00p.m.'),

('B08','F07','8:00a.m.-6:00p.m.'),

('B09','F07','8:00a.m.-6:00p.m.'),

('B10','F07','8:00a.m.-6:00p.m.'),

('B11','F08','5:00p.m.-9:00p.m.'),

('B12','F09','12:00p.m.-5:00p.m.'),

('B13','F10','12:00p.m.-5:00p.m.');

commit;

SQL command (Melissa, Sofea, Joey)

- Create table
- Insert Data
- SQL for query

Normalization (Sofea)

1NF:

Transportbooking(paymentID, payment_transactiondate, payment_totalprice, payment_unit, payment_description, payment_method, userID, user_name, user_contactnumber, user_address, user_email, bookingID, booking_date, booking_type, booking_destination, booking_description, serviceID, service_type, service_contactnumber, service_email, service_description, companyId, company_name, company_contactnumber, company_description, transportID, transport_num, transport_type, transport_location, transport_description, staffID, staff_name, staff_contactnumber, staff_email, seatID, seat_position, seat_num, seat_type, seat_price, seat_description)

Functional Dependency:

1. Partial Dependency

userID-->user_name, user_contactnumber, user_address, user_email, bookingID, booking_date, booking_type, booking_destination, booking_description, paymentID,payment_transactiondate, payment_totalprice, payment_unit, payment_description, payment_method, serviceID, service_type, service_contactnumber, service_email, service_description

companyID-->companyID, company_name, company_contactnumber, company_description, transportID, transport_num, transport_type, transport_location, transport_description, staffID, staff_name, staff_contactnumber, staff_email, seatID, seat_position, seat_num, seat_type,seat_price,seat_description

2. Transitive Dependency

transportID-->seatID

3. Primary Key

userID,companyID-->bookingID

2NF:

user(userID, user_name, user_contactnumber, user_address, user_email, paymentID, payment_transactiondate, payment_totalprice, payment_unit, payment_description, payment_method, serviceID, service_type, service_contactnumber, service_email, service_descriptionseatID, seat_position, seat_num, seat_type, seat_price,seat_description)

company(companyID, company_name, company_contactnumber, company_description, transportID, transport_num, transport_type, transport_location, transport_description, staffID, staff_name, staff_contactnumber, staff_email)

bookingticket(userID, companyID, bookingID, booking_date, booking_type, booking_destination, booking_description)

1. Transitive dependency

transportID-->seatID

3NF:

bookingticket(userID, companyID, bookingID, booking_date, booking_type, booking_destination, booking_description)

user(userID, user_name, user_contactnumber, user_address, user_email, paymentID, serviceID)

payment(paymentID, payment_transactiondate, payment_totalprice, payment_unit, payment_description, payment_method)

service(serviceID, service_type,service_contactnumber, service_email, service_description)

company(companyID, company_name, company_contactnumber, company_description, staffID)

staff(staffID, staff_name, staff_contactnumber, staff_email, transportID)

transport(transportID, transport_num, transport_type, transport_location, transport_description, seatID)

seat(seatID, seat_position, seat_num, seat_type, seat_price, seat_description)

Data Dictionary (Melissa)

PK:Primary Key

FK:Foreign Key

1. Payment

No	Data	Description	Type	Length	PK/FK
1	paymentID	Define payment ID	Varchar	3	PK
2	payment_transactiondate	Define payment transaction date	Date		
3	payment_totalprice	Define payment total price	Decimal	5,2	
4	payment_unit	Define payment unit	Interger		
5	payment_description	Define payment description	Varchar	50	
6	payment_method	Define payment method	Varchar	50	

2. User

No	Data	Description	Type	Length	PK/FK
1	userID	Define user ID	Varchar	3	PK
2	user_name	Define user name	Varchar	50	

3	user_contactnumber	Define contact number	Varchar	20	
4	user_address	Define user address	Varchar	50	
5	user_email	Define user email	Varchar	50	
6	paymentID	Define payment ID	Varchar	3	FK

3. Booking Ticket

No	Data	Description	Type	Length	PK/FK
1	bookingID	Define booking ID	Varchar	3	PK
2	booking_date	Define booking date	Date		
3	booking_type	Define booking type	Varchar	20	
4	booking_destination	Define booking destination	Varchar	50	
5	booking_description	Define booking description	Varchar	50	
6	userID	Define user ID	Varchar	3	FK
7	paymentID	Define payment ID	Varchar	3	FK

4. Service

No	Data	Description	Type	Length	PK/FK
1	serviceID	Define service ID	Varchar	3	PK
2	service_type	Define service type	Varchar	50	
3	service_contactnumber	Define service contact number	Varchar	20	
4	service_email	Define service email	Varchar	50	
5	service_description	Define service description	Varchar	50	
6	bookingID	Define booking ID	Varchar	3	FK
7	userID	Define user ID	Varchar	3	FK

5. Staff

No	Data	Description	Type	Length	PK/FK
1	staffID	Define staff ID	Varchar	3	PK
2	staff_name	Define staff name	Varchar	50	
3	staff_contactnumber	Define staff contact number	Integer		
4	staff_email	Define staff email	Varchar	50	
5	transportID	Define transport ID	Varchar	3	FK

6	companyID	Define company ID	Varchar	3	FK

6. Company

No	Data	Description	Type	Length	PK/FK
1	companyID	Define company ID	Varchar	3	PK
2	company_name	Define company name	Varchar	50	
3	company_contactnumber	Define company contact number	Varchar		
4	company_description	Define company description	Varchar	50	

7. Transport

No	Data	Description	Type	Length	PK/FK
1	transportID	Define transport ID	Varchar	3	PK
2	transport_num	Define transport number	Varchar	7	
3	transport_type	Define transport type	Varchar	10	
4	transport_location	Define transport location	Varchar	50	
5	transport_de	Define	Varchar	50	

	scription	transport description			
6	companyID	Define company ID	Varchar	3	FK

8. Seat

No	Data	Description	Type	Length	PK/FK
1	seatID	Define seat ID	Varchar	20	PK
2	seat_position	Define seat position	Varchar	5	
3	seat_num	Define seat num	Varchar	5	
4	seat_type	Define seat type	Varchar	20	
5	seat_price	Define seat price	Decimal	5,2	
6	seat_description	Define seat description	Varchar	50	
7	bookingID	Define booking ID	Varchar	3	FK
8	transportID	Define transport ID	Varchar	3	FK

9. Orders

No	Data	Description	Type	Length	PK/FK
1	seatID	Define seat ID	Varchar	3	FK
2	trasnportID	Define trasport ID	Varchar	3	FK

3	count_journey	Define count journey	Integer		

10. Class

No	Data	Description	Type	Length	PK/FK
1	companyID	Define company ID	Varchar	3	FK
2	bookingID	Define booking ID	Varchar	3	FK
3	operation_hour	Define operation hour	Integer		

11. Details

No	Data	Description	Type	Length	PK/FK
1	staffID	Define staff ID	Varchar	3	FK
2	bookingID	Define booking ID	Varchar	3	FK
3	working_time	Define working time	Varchar	20	

References (Jiaxin)

- freeproject. (2017, April 14). Bus Ticket Booking System ER Diagram. free Projectz. https://www.freeprojectz.com/dfd/bus-ticket-booking-system-dataflow-diagram
- oretnom23. (2020, September 9). Online Bus Booking System Project Using PHP/MySQL. Sourcecodester. https://www.sourcecodester.com/php/14438/online-bus-booking-system-project-using-phpmysql.html
- Ravikiran A S. (2022, Nov 18). What is Normalization in SQL? 1NF, 2NF, 3NF and BCNF in DBMS. simplilearn. https://www.simplilearn.com/tutorials/sql-tutorial/what-is-normalization-in-sql

- Edward Pollack. (2021, April 1). Building a SQL Server data dictionary. Redgate Hub.https://www.red-gate.com/simple-talk/databases/sql-server/database-administration-sql-server/building-a-sql-server-data-dictionary/
- Cacoo. (2022, Feb 9). ER diagrams vs. EER diagrams: What's the difference? nulab.https://nulab.com/learn/software-development/er-diagrams-vs-eer-diagrams-whats-the-difference/

Work Distributions (Joey)

Appendix (Joey)