



ALBUKHARY INTERNATIONAL UNIVERSITY

## SCHOOL OF COMPUTING AND INFORMATICS

**CCC2133 : Database Management System**

**Semester I, Academic Session: 2023/2024**

### **SYSTEM IMPLEMENTATION (Final Report)**

*Group Number 19*

*Lecturer | Assoc. Prof. Dr. Zurinahni Zainol*

*Project Title | Bus Booking Management System*

NAME	MATRIC NO	EMAIL ID	MODULE	ROLE
Ugyen Tshering	AIU22102222	<a href="mailto:ugyen.tshering@student.aiu.edu.my">ugyen.tshering@student.aiu.edu.my</a>	Admin	Leader
Md Farman Ali	AIU20092185	<a href="mailto:farman.ali@student.aiu.edu.my">farman.ali@student.aiu.edu.my</a>	Ticket	Member
Abdullahi Adewole Zakariyah	AIU221023260	<a href="mailto:abdullahi.zakariyah@student.aiu.edu.my">abdullahi.zakariyah@student.aiu.edu.my</a>	Driver	Member
Areyan Muhemed Ali Hussein	AIU21102179	<a href="mailto:areyan.hussein@student.aiu.edu.my">areyan.hussein@student.aiu.edu.my</a>	Passenger	Member
Aidarus Abdirahman Egal	AIU21102202	<a href="mailto:aidarus.egal@student.aiu.edu.my">aidarus.egal@student.aiu.edu.my</a>	Rating and Feedbacks	Member
Nima Yoezer	AIU22102221	<a href="mailto:nima.yozer@student.aiu.edu.my">nima.yozer@student.aiu.edu.my</a>	Refund and Cancellation	Member

\

## TABLE OF CONTENTS

<b>1.</b>	<b>INTRODUCTION</b>	
1.1.	<b>Background Information.....</b>	<b>3</b>
1.2.	<b>Current System and its problems.....</b>	<b>3</b>
1.3.	<b>Project Scope.....</b>	<b>4</b>
1.4.	<b>Motivation.....</b>	<b>4</b>
1.5.	<b>How will the system benefit the organization.....</b>	<b>4</b>
1.6.	<b>Modules.....</b>	<b>5</b>
<b>2.</b>	<b>USER REQUIREMENTS AND BUSINESS RULES</b>	
2.1.	<b>Administration Module.....</b>	<b>6</b>
2.2.	<b>Passenger Module.....</b>	<b>7</b>
2.3.	<b>Ticket Module.....</b>	<b>8</b>
2.4.	<b>Driver Module.....</b>	<b>9</b>
2.5.	<b>Cancellation and Refund Module:.....</b>	<b>9</b>
2.6.	<b>Feedback and Rating Module.....</b>	<b>10</b>
2.7.	<b>Cross-Module.....</b>	<b>11</b>
<b>3.</b>	<b>Entity Relationship Modelling</b>	
3.1.	<b>Admin Module.....</b>	<b>12</b>
3.2.	<b>Passenger Module.....</b>	<b>15</b>
3.3.	<b>Ticket Module.....</b>	<b>18</b>
3.4.	<b>Driver Module.....</b>	<b>21</b>
3.5.	<b>Cancellation and Refund Module.....</b>	<b>24</b>
3.6.	<b>Feedback and Rating Module.....</b>	<b>27</b>
3.7.	<b>Combined-ERD.....</b>	<b>29</b>

<b>4.</b>	<b>Normalization</b>	
4.1.	<b>Admin Module</b>	32
4.2.	<b>Passenger Module</b>	34
4.3.	<b>Ticket Module</b>	35
4.4.	<b>Driver Module</b>	36
4.5.	<b>Cancellation and Refund Module</b>	37
4.6.	<b>Feedback and Rating Module</b>	38
<b>5.</b>	<b>Data Dictionary</b>	
5.1.	<b>Admin Module</b>	39
5.2.	<b>Passenger Module</b>	42
5.3.	<b>Ticket Module</b>	45
5.4.	<b>Driver Module</b>	47
5.5.	<b>Cancellation and Refund Module</b>	50
5.6.	<b>Feedback and Rating Module</b>	53
<b>6.</b>	<b>Data Implementation</b>	54
<b>7.</b>	<b>Front-End System Design and Implementation</b>	66
<b>8.</b>	<b>Project Problems and Pitfalls</b>	72
<b>9.</b>	<b>Conclusion/Recommendations/Future Work</b>	72
<b>10.</b>	<b>References</b>	74

## **1.1 BACKGROUND INFORMATION**

Before deciding on the organization we are going to build the database for, we carried out some extensive research to look for the most suitable organization. Surprisingly, we found out that all the transport companies here in Malaysia have databases, and their databases are in perfect order, requiring no adjustment or amendment. This led us to choose a district-level transport company in *Bhutan* called *Karma Transport*. We discovered that the company lacks a database as well as an official website to book tickets. In addition, the company currently adopts a traditional method of booking tickets, like through a phone call, or going to the bus reservation office which customers find inconvenient and time consuming.

After discovering all this, we decided to build a database system, as well as a decent website, for the company to facilitate the growth of their business. Through the database system, we want to address those issues faced by both the business owner in managing passengers details/tickets and for the convenience of passengers.

## **1.2 CURRENT SYSTEM AND ITS PROBLEMS**

The current system of Karma Transport does not have a database for bus booking and adheres with the conventional method of booking a ticket. This lack of a database imposes several threats and problems to the company. On the other hand, our database system is aimed to provide the best and most effective solutions to those problems.

1. Lack of Record Keeping: Without a database, the company struggles to maintain records of passenger details, such as passenger names, contact information, ticket information and many more. This can lead to difficulties in managing bookings, tracking passenger information, and providing efficient passenger service.
2. Dependence on Manual Task: The absence of a database means that the company has to rely on manual methods to organize and arrange data. This can be time-consuming and prone to errors, as manual data entry and retrieval processes are more susceptible to mistakes.
3. Limited Access for Passengers: Without a database, passengers may face challenges in accessing information about available buses, seats available, routes, and ticket availability. This lack of easy access can be inconvenient to passengers and deter them from using the company's services.
4. Payment and Refund Issues: The company faces problems related to payment methods and refund processes. Without a database management system, it may be difficult to offer a variety of payment options, track payments accurately, and handle cancellations and refunds efficiently.

5. Security Concerns: The absence of a database management system can also raise security concerns. Sensitive passenger information, such as personal and payment details, may not be adequately protected without proper encryption and security measures in place.

By carefully addressing these considerations, we aim to develop a bus booking system that is efficient, secure, and user-friendly. Regular maintenance and updates will further contribute to the long-term success of the system.

### **1.3 PROJECT SCOPE**

Databases are designed in such a way that it will keep an individual's record and make the information available anytime it is in need. Databases also serve as reference, as it shows data that has been kept as record purposes, which can clarify ambiguity between clients and company.

Our aim in designing this database is to keep record of the company in a structured manner, the database we will design will also perform some activities including; data integrity, computerized data storage, data security, availability of data to the client and the company.

Our database is limited to only structured data and restricted from unstructured or semi structured data, we will make a database for Karma transport based on structured data only. Nowadays organizations are based mainly on structured data as it is easy to maintain and it helps on data retrieval.

### **1.4 MOTIVATION**

The motivation behind the proposed bus booking management system is to automate and optimize the bus booking reservation process for Karma Transport company, which was initially done manually. By implementing and integrating a streamline database system, it aims to help passengers access the details (such as their desired routes, buses, tickets, departure) more easily and enhance the overall booking experience.

### **1.5 HOW WILL THE SYSTEM BENEFIT THE ORGANIZATION**

Our database system involves addressing the issues mentioned above (1.2).

- Firstly, our database system can help eliminate manual and mundane tasks which are time consuming. Our online system is available to use anytime and the user does not need to visit the office inorder to book a ticket or inquire about something.

- Secondly, it will maintain records of passenger details, bus details, ticket details and help to organize and arrange those data efficiently, making it accessible to all the end-users including drivers, admin and passengers.
- Thirdly, our database system can help passengers access any information associated with bus, drivers, departure, like which bus is going from which station and where. Moreover, passengers can look up about their tickets, whether it's confirmed or not, and the information about refunds if they wish to cancel their reservation.
- Lastly, unlike the conventional file system that was implemented in Karma Transport previously, our database system comes with builtin security features that protects passengers information and all transactions from unauthorized users.

## 1.6 MODULES

For the Database Management System, we have 6 modules, which includes; – Administration, Passenger, Ticket, Driver, Cancellation and Refund Module, Feedback and Rating Module. These modules work together to give functionalities to the overall database system.

1. **Administration Module:** This module is solely for administrators, to update the information or details on bus routes, tickets, offering discounts, launching updates and all. It will be part of the backend view of this application and passengers won't be able to access it.
2. **Passenger Module:** This module is designed to allow passengers to create their own account and update their information respectively. In this database, it will save passengers information like;, passenger's name, passenger's age, passenger's gender, username and password. Subsequently, passengers can choose their departure and final destination along with specific dates and time and reserve it online. Shortly after they have made the payment, they will receive their e-tickets in their respective emails.
3. **Ticket Module:** This module is designed to show available buses and tickets in a particular route to the passengers. This will serve as the main page/home page of the website, where customers can have a look at bus routes, tickets, number of buses available, or seats available. It will also include the specified price for each ticket. Not only that, it will give details about the bus as well, so that customers can choose the bus type up to their comfort. It includes, bus route, no of buses available, bus type, bus color, number plate, bus driver contact details, available seats, departure date and time, arrival date and time.

4. **Driver Module:** This module is designed to store information about drivers and which bus they should drive on which date. It is aimed to minimize the confusion between drivers. Moreover, the company owner can have access to these details and make changes if required. Like adding new bus drivers or kicking out old drivers from the company. In this database, the module will store the driver's name, driver's license, bus they are driving, bus number plates, shifts(day or night).
  
5. **Cancellation/Refund Module:** Unlike other company's databases, we have a separate module for passengers who wish to cancel or make a refund. Those passengers will be stored in this module, where it stores, passenger's name, passenger's id, seat no booked, passenger's reason for refund. The 'reason for refund' will help the company analyze the weakness and improve its services. In addition, Passengers can only apply for a refund before 3 days of departure, so that seats can be made available for other passengers during that time.
  
6. **Rating and Feedbacks Module:** To ensure operational sustainability of the business, it's crucial for business to have an interactive interface where passengers can rate and give feedback to improve its quality of services. That's why, this module is designed to accumulate ratings and feedback given by passengers during their journey with Karma transport. Here, passengers will be marked 'anonymous' but the travel details including the bus route, time, bus driver, bus type and all the necessary information will be saved.

## 2. USER REQUIREMENTS AND BUSINESS RULES

### 2.1 Admin Module

*Table 1: User Requirements and Business Rules of Admin Module*

User Requirement	Business Rule
1.0) Admin should regulate the operation of the entire database. They should delete previous and update new details on the tickets / buses available on a particular route. So that customers can look for the available buses and seats.	1.0) An admin can manage many tickets. A ticket can only be managed by one admin at a time.

1.1) Admin should regulate passengers' accounts, like usernames, emails and passwords. They should make sure that no duplicate account for a specific username/email appears anywhere.	1.1) An admin can administer zero or many accounts. An account can be administered by zero or only one admin.
1.2) Admin should have access to the staff/bus driver profiles. They should constantly keep records of bus drivers and their status.	1.2) An admin should have access to any staff (drivers) profiles. Each driver profile can be managed by only one admin.
1.3) Admin should handle the ticket canceling and refunding process with the passengers. They should be able to retrieve details from passengers who wish to cancel and take necessary steps to make refunds to the passengers.	1.3) An admin can handle many refund / cancellation requests from passengers. Refund / cancellation requests can be processed by only one admin.
1.4) Admin should also manage feedback and rating from the passengers and generate reports monthly.	1.4) An admin will receive a lot of feedback and ratings from the passengers. Feedbacks and ratings will be analyzed by one admin.

## 2.2 Passenger Module

Table 2: User Requirements and Business Rules of Passenger Module

User Requirement	Business Rule
2.1) The system allows passengers to create an account along with strong passwords. Password and the username allows the passenger to log in to their accounts	2.1) A passenger can have zero or legally be registered to one account. An account can only be owned by one passenger only.
2.2) The system allows passengers with accounts to apply for membership where they will receive special offers and discounts.	2.2) An account can have zero or only one membership, membership can be owned by one or many accounts.
2.3) Passengers who applied for membership are	2.3) One membership is entitled to one or many

entitled to a lot of benefits such as getting a discount on long distance travel, getting full refund payment if a ticket is canceled and so on.	benefits depending on the availability , each benefit is entitled to one or many memberships.
--	---

### 2.3 Ticket Module

*Table 3: User Requirements and Business Rules of Ticket Module*

User Requirement	Business Rule
3.1) It allows passengers to view details about buses like; bus driver, bus color, number plate. It even shows different types of bus, such as, single deck, double-deck and coach bus.	3.1) A ticket will only contain info about ONE bus. Info about the bus will be visible on zero or many tickets.
3.2 Passengers can choose any tickets and the price for each ticket will be shown.	3.2) A passenger can book zero or many tickets. A ticket can only belong to one passenger.
3.3) The system will allow the passengers to purchase tickets online using the account number. It will NOT provide an inbuilt transaction page. Passengers will have to use their own banking app and send the screenshot of transaction	3.3) A passenger can send only one transaction screenshot. Each transaction screenshot can be made by zero or many passengers.
3.4) Tickets will also show bus drivers information such as their name, phone number and other relevant information.	3.4) A ticket can have details about two or more bus drivers if there are morning and night 'shifts' involved. Bus drivers' information can be shown in zero or only one ticket.
3.5) Passengers will receive their e-tickets through their respective emails. In addition, it will also provide any updates or important notice, if any, before the departure.	3.5) A passenger can receive many e-tickets in their respective emails. An e-tickets will be sent to only one email / passenger.

## 2.4 Driver Module

*Table 4: User Requirements and Business Rules of Driver Module*

User Requirement	Business Rule
4.1) The system allows drivers to know about the bus they will drive and where they will drive the passengers to.	4.1) A driver can only drive one bus at a time. A bus can be driven by one driver only.
4.2) Driver can have access to the tickets of the passengers, it can help in the verification process when the passengers are boarding the bus	4.2) Driver can view one or many tickets, A ticket can be viewed by the only driver on board.
4.4) Each Bus will be able locate their pick up points, which will make the trip easier for the driver as they know each stops and pick up areas	4.4) A bus can have one or many pick up points. Each pick up point can have only one bus.
4.5) Drivers can constantly update the status about their trip to indicate whether they have reached their location or not.	4.5) A driver can update one or many trip statuses. trip status can only be updated by the driver.
4.6) Drivers can be able to know his or her shift schedule in a day or week , shift types include; morning shift and afternoon shift.	4.6) A Driver can have zero or many shifts in a day, a shift can only have one driver.
4.7) Driver updates the trip status on each pick up point, on each pick up point driver updates trip status so that administrator will be updated about the journey.	4.7) A pick up points can have one trip status, trip status can consist of many pick up points.

## 2.5 Cancellation and Refund Module:

*Table 5: User Requirements and Business Rules of Cancellation and Refund Module*

User Requirement	Business Rule
5.1) If passengers wish to cancel their tickets and	5.1) A passenger can make multiple cancellations

apply for refund, they can apply for the refund and send the cancellation request.	requests. A cancellation request can only be made by one passenger. 5.2) Refunds will only be eligible before 3 days of departure.
5.2) Passengers have to provide their banking credentials along with other ticket details that will be used for refund purposes only.	5.2) A passenger can only have one account. An account can only be owned by one passenger.
5.3) In addition, passengers will have to provide reasons for the refund, so that the company could collect those data to improve its services in the future.	5.3) A passenger applying for a refund can write any number of reasons. A reason must be stated by only one passenger applying for a refund.
5.4) Passengers will have to provide their ticket details including their, ticket id, bus id, seat no and other relevant information	5.4) A passenger can only apply for one ticket's refund. Ticket's refund can be made by only one passenger.
5.5) Refunds will be made through online payment only and the passengers should bear any tax or service fees. On the other hand, passengers have the right to select a convenient payment method.	5.5) A refund can only include one transaction for each request. A transaction can only be made once for each refund request.

## 2.6 Feedback and Rating Module

*Table 6: User Requirements and Business Rules of Feedback and Rating Module*

User Requirement	Business Rule
6.1) Design a user-friendly interface for passengers to submit feedback. It allows passengers to provide feedback.	6.1) A passenger can generate zero or many feedback. Feedbacks can be submitted by one or many passengers.
6.2) Passengers are given a separate platform to	6.2) A passenger can choose not to rate or rate

rate their overall journey including how comfortable the bus is and overall services.	once for every departure. Each rating will be given by only one passenger.
6.3) An admin should be able to generate reports on average ratings, common issues, and other relevant metrics.	6.3) An admin can generate only one single report from all the ratings and feedback. A report can be generated by an admin.

## 2.7 Cross-Module User Requirement and Business Rules

Table 7: User Requirements and Business Rules of Cross-Module

User Requirement	Business Rule
7.1) Drivers have access to passenger's details as well as their ticket details such as ticket_id, passengers gender, passenger age, seat no and others.	7.1) A driver can view details of one or many passengers traveling. A passenger's details can only be accessed by only one bus driver.
7.2) Admin should process e-tickets and send via email or whatsapp to the respective passengers. So that way, passengers don't have to travel all the way to the reservation office to get paper tickets.	7.2) An admin can process zero or many e-tickets. An e-tickets can only be processed by one admin.
7.3) Tickets should have details on pickup points, like when and where the bus will stop. So that passengers can wait at the nearest pickup station.	7.3) A ticket will consist of zero or many pickup points but pickup points will only appear one ticket, meaning only one bus will handle one pickup point.
7.4) During the refund process, an admin needs to have Bank credentials of the passenger so that it can proceed with the transaction. During that time, the admin can access the cancellation module where passengers give bank credentials.	7.4) An admin can access one or many bank credentials of passengers. A passenger's bank credential can be accessed by only one admin.
7.5) Passengers who have applied for the membership are entitled for the discounts. So the	7.5) An admin should be able to view and manage many passengers who applied for membership.

admin should know who all have applied for membership accounts and under what conditions they should be given discounts.

Each passenger with a membership account can be managed by only one admin.

### **3.0 Entity Relationship Modelling**

All the entity relationship diagrams are drawn with Crow's foot notation

#### **3.1 ADMIN MODULE**

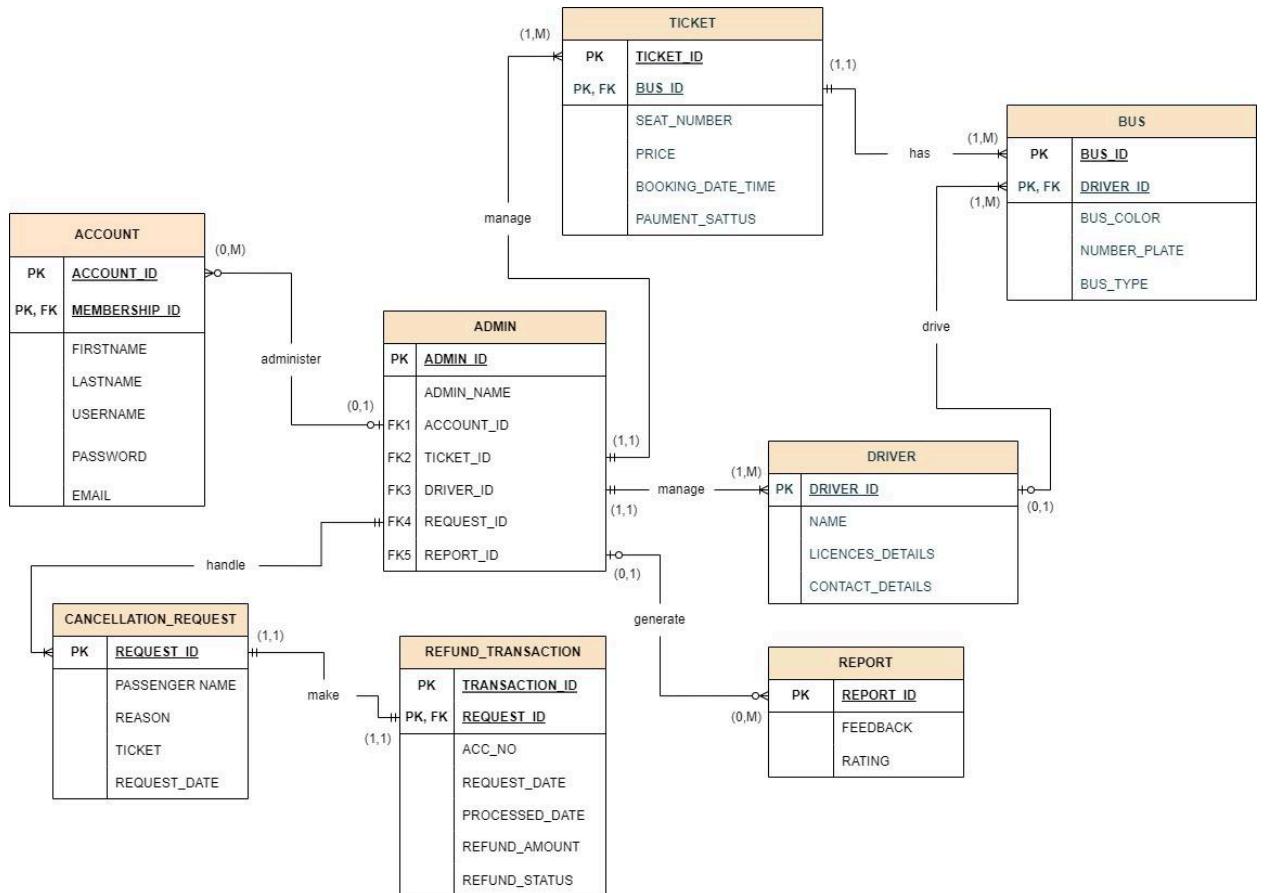


Figure 1: ERD Diagram for ADMIN Module

### ***Explanation***

Admin module contain 9 entities, namely ADMIN, ACCOUNT, MEMBERSHIP\_STATUS, TICKET, BUS, DRIVER, REPORT, CANCELLATION\_REQUEST and REFUND\_TRANSACTION.

### ***Relationship***

- ADMIN and ACCOUNT entities have one to many relationships, both being optional to one another (refer to business rule 1.1). It is a weak relationship because the ACCOUNT entity does not inherit the primary key of the ADMIN entity.
- ADMIN and TICKET entities have one to many relationships, where ADMIN entities being mandatory. They have a weak relationship.
- TICKET and BUS entities have one to many relationships, where TICKET entities are mandatory. It is a strong relationship as the TICKET entity contains the primary key of the BUS entity called BUS\_ID.
- ADMIN and DRIVER entities have one to many relationships, where ADMIN entity being mandatory (refer to business rule 1.2). It formed a weak relationship as the DRIVER entity does not contain any primary keys of the ADMIN entity.
- ADMIN and REPORT entities have one to many relationships, both being optional (refer to business rule 1.4). Again, it formed a weak relationship as one entity does not inherit another entity's primary key.
- ADMIN and CANCELLATION\_REQUEST form one to many relationships, where the ADMIN entity is mandatory (refer to business rule 1.3). It formed a weak relationship.
- CANCELLATION\_REQUEST and REFUND\_TRANSACTION entity forms one to one relationship, both being mandatory. It formed strong relationships, as REFUND\_TRANSACTION uses REQUEST\_ID, as the primary key of CANCELLATION\_REQUEST.

### ***Key Constraints***

- ADMIN entity has ADMIN\_ID as the primary key, followed by 5 foreign keys, namely, ACCOUNT\_ID which refers to ACCOUNT entity, TICKET\_ID which refers to TICKET entity, DRIVER\_ID refers to DRIVER entity, REQUEST\_ID refers to REQUEST entity and REPORT\_ID refers to REFER entity.
- ACCOUNT entity has two primary keys, ACCOUNT\_ID and MEMBERSHIP\_ID, latter being foreign key as well. It is a strong entity as its existence does not depend on other entities.

- A TICKET entity has TICKET\_ID and BUS\_ID as their primary key, the latter being foreign key as well. It is a strong entity as its existence is independent.
- The BUS entity has two primary keys, BUS\_ID and DRIVER\_ID, the latter being foreign key as well. It is a strong entity.
- DRIVER entity has DRIVER\_ID as their primary key, which uniquely identifies all other attributes. Due to its independent existence, it is a strong entity.
- The REPORT entity has REPORT\_ID as the primary key.
- CANCELLATION\_REQUEST entity has REQUEST\_ID as primary key. It is a strong entity.
- The REFUND\_TRANSACTION entity has two composite primary keys, TRANSACTION\_ID and REQUEST\_ID, the latter being foreign key as well which references the CANCELLATION\_REQUEST entity.

### ***Clarification***

- In the ADMIN entity, ADMIN\_ID uniquely identifies each administrator managing the system. It has following attributes; ADMIN\_NAME which stores admin's name, ACCOUNT\_ID which is a foreign key referring ACCOUNT entity, TICKET\_ID which is foreign key, referring TICKET entity, DRIVER\_ID referring driver entity, REQUEST\_ID and REPORT\_ID which are both foreign key referring CANCELLATION\_REQUEST and REPORT entity respectively.
- In ACCOUNT entity, ACCOUNT\_ID and MEMBERSHIP\_ID which uniquely identifies other attributes. FIRSTNAME and LASTNAME attributes storing names of the passengers, USERNAME, PASSWORD and EMAIL attributes storing passenger's login credentials.
- In the TICKET entity, TICKET\_ID and BUS\_ID are the two composite primary keys which identifies the entity. It has SEAT\_NUMBER, PRICE, BOOKING\_DATE/TIME and PAYMENT\_STATUS, which stores all the necessary details.
- In the BUS entity, it has two primary keys called BUS\_ID which uniquely identifies each bus. It has DRIVER\_ID as a composite primary key which is also a foreign key that refers to the DRIVER entity. It includes BUS\_COLOR, NUMBER\_PLATE and BUS\_TYPE which gives information about the color of the bus, its number plate and its type (which ranges from single-deck, double-deck and coach bus).
- In the DRIVER entity, DRIVER\_ID is the primary key, followed by driver's NAME, LICENSE\_DETAILS like license number, issue dates, renewal dates, and CONTACT\_DETAILS of the driver.
- In the REPORT entity, REPORT\_ID uniquely identifies each report. Which also consists of FEEDBACK and RATING given by passengers.

- In CANCELLATION\_REQUEST entity, it has REQUEST\_ID as primary key. Followed by PASSENGERS\_NAME, REASON, TICKET details and their REQUEST DATE.
- Lastly, the REFUND\_TRANSACTION which include two primary keys, TRANSACTION\_ID and REQUEST\_ID, latter being the foreign key referring to the REQUEST entity. It consists of after attributes like ACC\_NO of the passenger, REQUEST\_DATE, refund PROCESSED\_DATE, REFUND\_AMOUNT and REFUND\_STATUS to indicate whether the refund has been processed or not.

### 3.2 PASSENGER MODULE

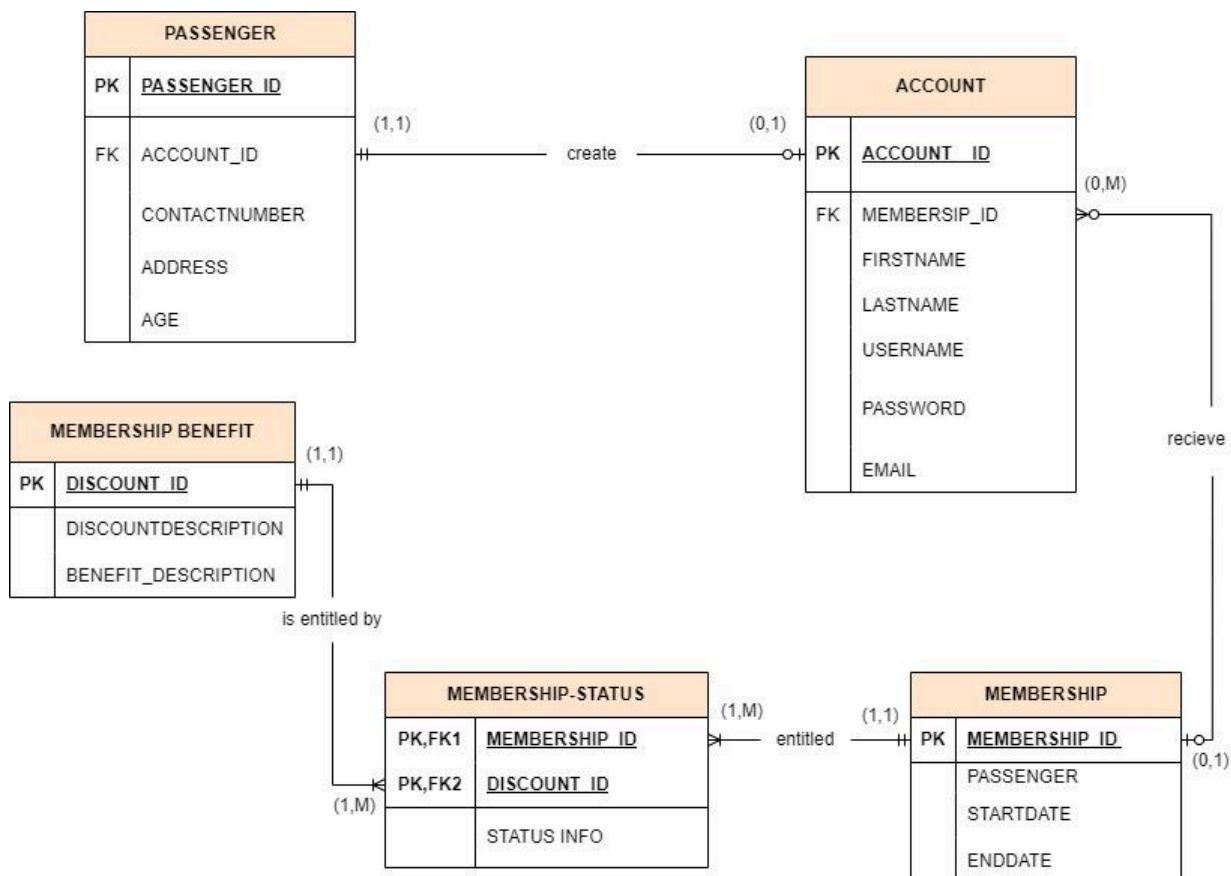


Figure 2: ERD Diagram for PASSENGER Module

### ***Explanation***

Passenger module contain 5 entities, namely PASSENGER\_ID, ACCOUNT, MEMBERSHIP\_ID, MEMBERSHIP\_STATUS, MEMBERSHIP BENEFIT

### ***Relationships:***

- PASSENGER entity and ACCOUNT entity has zero to one relationship(refer to business rule 3.1) which is a weak relationship as primary key of ACCOUNT entity do not consist of the primary key of PASSENGER entity
- ACCOUNT entity and MEMBERSHIP entity has one to more relationship (refer to business rule 3.2) which is a weak relationship as primary key of ACCOUNT entity do not consist primary key of MEMBERSHIP
- MEMBERSHIP entity and MEMBERSHIP\_BENEFIT entity has a one-to many relationship (refer to business rule 3.3). In order to resolve this one-to-many relationships in this ERD, MEMBERSHIP\_STATUS entity acts as a bridge entity between the two entities
- MEMBERSHIP\_STATUS entity has a strong relationship between both MEMBERSHIP entity and MEMBERSHIP\_BENEFIT entity as it inherits both primary key from both MEMBERSHIP entity and MEMBERSHIP\_BENEFIT entity. At the same time, MEMBERSHIP\_STATUS entity is known to be a weak entity as it is existence dependent to MEMBERSHIP entity and MEMBERSHIP\_BENEFIT entity.

### ***Key Constraints:***

- PASSENGER entity has a primary key named PASSENGER\_ID and a foreign key from ACCOUNT entity which is ACCOUNT\_ID.
- The ACCOUNT entity has a primary key named ACCOUNT\_ID and a foreign key from the MEMBERSHIP entity which is MEMBERSHIP\_ID.
- MEMBERSHIP entity has a primary key named MEMBERSHIP\_ID
- MEMBERSHIP\_STATUS entity is a weak entity as it inherits two primary and foreign keys namely MEMBERSHIP\_ID from MEMBERSHIP entity and DISCOUNT\_ID from MEMBERSHIP\_BENEFIT entity. It cannot uniquely identify using its own attributes.
- MEMBERSHIP\_BENEFIT entity has DISCOUNT\_ID as its primary key

### ***Clarification***

- In the PASSENGER entity, PASSENGER\_ID is a primary key which is a unique ID after login to the system . Additionally ACCOUNT\_ID is a foreign key to check the account of passengers, The other attribute is ACCOUNT NUMBER and ADDRESS.
- In an ACCOUNT entity, ACCOUNT\_ID is a primary key which is a unique ID . foreign key which is a MEMBERSHIP-ID is created in order to sign in into the application after that the passenger to completes some important details such as FIRSTNAME, LASTNAME, USERNAME, PASSWORD, EMAIL and REGISTRATION.
- In the MEMBERSHIP entity, MEMBERSHIP\_ID which is the primary key which is actually an ID generated by the system after the registration in account has successfully applied for a membership.
- In the MEMBERSHIP\_STATUS entity, the MEMBERSHIP-ID and DISCOUNT-ID both have primary key and foreign key, the STATUS\_INFO attribute shows whether or not the membership owned by the passenger is active.
- In the MEMBERSHIP\_BENEFIT entity, the primary key, DISCOUNT\_ID is the discount code entitled for passengers to get good service. The other attributes are BENEFIT\_DESCRIPTION and DISCOUNT\_DESCRIPTION which is the description of each type of discount like the benefit of discount and percentage of discount.

### 3.3 TICKET MODULE

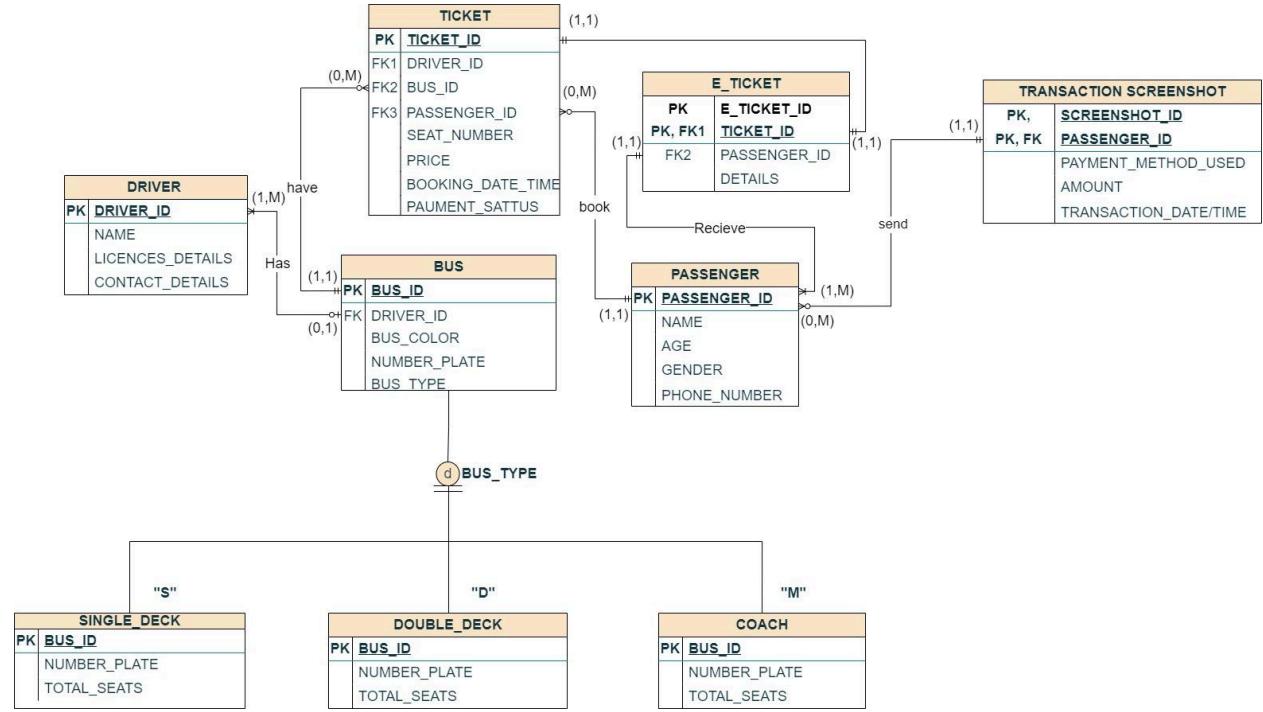


Figure 3: ERD Diagram for TICKET Module

#### **Explanation:**

Ticket module contains 9 entities which includes, TICKET, E-TICKET, TRANSACTION\_SCREENSHOTS, DRIVER, BUS, PASSENGER. BUS entity is a super type with three subtypes, namely, SINGLE\_DECK, DOUBLE\_DECK and COACH.

#### **Relationship:**

- Ticket entity and BUS entity has a one-to-many relationship (refer to business rule 4.1) which is a weak relationship as primary keys of TICKET entity do not consist of the primary key of BUS entity.
- TICKET entity and PASSENGER entity has a one-to-many relationship (refer to business rule 4.6) which is also a weak relationship as primary keys of TICKET entity do not consist of the primary key of PASSENGER entity.
- TICKET entity and E-TICKET entity has a one-to-one relationship which is a strong relationship as primary keys of TICKET entity consist of the primary key of E\_TICKET.
- PASSENGER entity and E-TICKET has a one-to-many relationship which is a weak relationship as primary keys of PASSENGER entity do not consist of the primary key of E\_TICKET entity.

- PASSENGER entity and TRANSACTION\_SCREENSHOT entity has a one-to-many relationship (refer to business rule 4.3) which is a strong relationship as primary keys of PASSENGER entity consist as the primary key of TRANSACTION\_SCREENSHOT entity.
- A BUS entity and a DRIVER entity has a one-to-many relationship (refer to business rules 4.5) which a weak relationship as primary keys of BUS entities do not consist of the primary key of DRIVER entity.
- Also BUS entities have three BUS\_TYPE which are SINGLE\_DECK, DOUBLE\_DECK and Medium\_COCH.

***Key Constraints:***

- In the TICKET entity, the TICKET entity has 8 attributes. TICKET\_ID and E\_TICKET are the composite primary keys of the TICKET entity which is a strong relationship between them. TICKET entity is a strong entity as its composite primary key uniquely identifies each entity and it is existence independent. TICKET entity contains DRIVER\_ID, BUS\_ID, PASSENGER\_ID which are foreign keys of the TICKET entity.
- In the E\_TICKET entity, the E\_TICKET entity has 6 attributes. E\_TICKET\_ID is the primary key while TICKET\_ID is both the primary and foreign key which has a strong relationship with the TICKET\_ID. Another one is PASSENGER\_ID which is foreign key and has a weak relationship with the passenger.
- In the BUS entity, the BUS entity has 5 attributes. BUS\_ID is the primary key which is a weak relationship with the ticket entity. BUS\_TYPE has three different entities namely SINGLE\_DECK, DOUBLE\_DECK, and COACH. Each subtype entity has 3 attributes, with BUS\_ID being a primary key.
- In the DRIVER entity, DRIVER entity has 4 attributes. DRIVER\_ID is the primary key which is a weak relationship with the bus entity.
- The PASSENGER entity consists of 5 attributes. PASSENGER\_ID is the primary key which is a strong relationship SCRENSHOT\_ID.
- IN TRANSACTION\_SCREENSHOT entity, SCREENSHOT\_ID and PASSENGER\_ID are composite primary keys, the latter being foreign key as well.

***Clarification:***

- In the TICKET entity, TICKET\_ID is a primary key which uniquely identifies the ticket module which is referred to as E\_TICKET\_ID. Where DRIVER\_ID, BUS\_ID and PASSENGER\_ID are the foreign keys. SEAT\_NUMBER which is represented by the bus seat number. PRICE is

described as bus ticket price. BOOKING\_DATE\_TIME is represented to ticket booking date and time as well as PAYMENT\_STATUS is used to update the status of payment.

- In E\_TICKET entity, E\_TICKET\_ID and TICKET\_ID are primary keys, which uniquely identifies the entity. TICKT\_ID and PASSENGER\_ID are the foreign keys as well. PASSENGER\_ID refers to a particular passenger. DETAILS which are described to E\_TICKET details or everything.
- In the PASSENGER entity, PASSENGER\_ID is the primary key which uniquely refers to the TRANSACTION\_SCREENSHOT entity. NAME, AGE, GENDER, PHONE\_NUMBER stores the details about the passengers.
- In the TRANSACTION\_SCREENSHOT entity, TRANSACTION\_SCREENSHOT\_ID and PASSENGER\_ID are the primary keys where PASSENGER\_ID is also the foreign key. PAYMENT\_METHOD\_USED represents what payment method is used and what AMOUNT is transferred, along with some details like, PAYMENT\_DATE/TIME.
- In the DRIVER entity, DRIVER\_ID is the primary key which is a uniquely identified driver entity. DRIVER\_ID refers to the BUS entity. NAME is represented by the driver name. LICENCES\_DETAILS is described to the driver licenses and CONTACT\_DETAILS is described to the driver contact details as well.
- In a BUS entity, BUS\_ID is the primary key which uniquely identifies each row in the entity. BUS entity refers to the DRIVER entity, where DRIVER\_ID is used as foreign key. BUS\_COLOR is a supertype used to represent the color of the bus. NUMBER\_PLATE which describes the number plate for the bus. BUS\_TYPE represents the type of buses which are single, double and medium(coach).
- Subtypes SINGLE\_DECK, DOUBLE\_DECK and COACH\_DECK have BUS\_ID each, which is the primary key which uniquely identifies each row in the entity. NUMBER\_PLATE which is described to the number plate of the single type buses. TOTAL\_SEATS is the total available seat of the single type buses.

### **3.4 DRIVER MODULE**

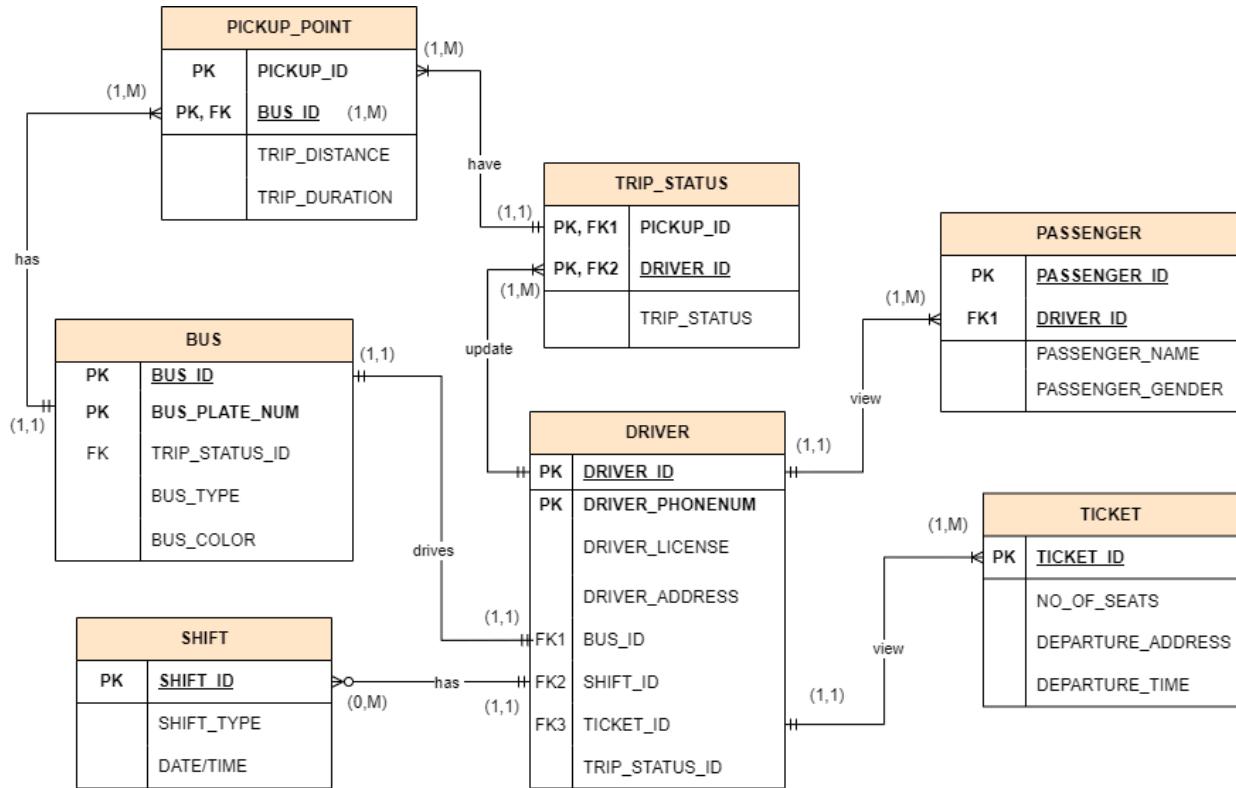


Figure 4: ERD Diagram for DRIVER Module

#### ***Explanation***

Driver module contains seven entities namely; DRIVER, PASSENGER, BUS, SHIFT, TRIP\_STATUS, PICKUP\_POINT, TICKET,.

#### ***Relationship;***

- DRIVER entity and BUS entity has one-to-one relationship (refer to business rule 5.1) which is refer to a weak relationship as the DRIVER entity contain of primary key (DRIVER\_ID ) and of BUS entity which is primary key is (BUS\_ID)
- DRIVER entity and PASSENGER entity has one-to-many relationship (refer to business rule 5.3) which is considered as a weak relationship as the DRIVER entity contain of primary key (DRIVER\_ID) and PASSENGER entity contain of primary key (PASSENGER\_ID)

- DRIVER entity and SHIFT entity has one-to-many relationship (refer to business rule 5.6) which is considered as weak relationship. SHIFT entity has a primary key (SHIFT\_ID) .
- DRIVER entity and TICKET entity has one-to-many relationship which is considered as weak relationship, DRIVER entity has a primary key (DRIVER\_ID) and the TICKET entity has a primary key (TICKET\_ID)
- BUS entity and PICKUP\_POINT entity has a one-to-many relationship (refer to business rule 5.4) which is considered as strong relationship as BUS\_ID from BUS entity serves as primary in PICKUP\_POINT entity. PICKUP\_POINT entity has a primary key (PICKUP\_ID) and (BUS\_ID). foreign key (BUS\_ID)
- DRIVER entity and TRIP\_STATUS entity has one-to-many relationship (refer to business rule 5.5) which is considered a strong relationship as DRIVER\_ID from DRIVER entity serves as primary in DRIVER entity and also serves as foreign key.
- PICKUP\_POINT entity and TRIP\_STATUS entity has one-to-many relationship (refer to business rule 5.6) which is considered as strong relationship as PICKUP\_ID from PICKUP\_POINT serves as primary key in TRIP\_STATUS and also serves as foreign key. Other primary key is DRIVER\_ID and it also foreign key

#### ***Key constraint***

- DRIVER\_ID and DRIVER\_PHONENUM are the composite primary key of the DRIVER entity. A DRIVER entity is a strong entity that has its composite primary key uniquely identifying each entity. DRIVER entity contains BUS\_ID from BUS entity, TICKET\_ID from ticket entity, SHFT\_ID from SHIFT entity.
- PICKUP\_POINT entity is a weak entity as it depends on BUS entity, BUS\_ID serves as its primary key and also foreign key as it is from BUS entity, the other primary key is PICKUP\_ID.
- SHIFT entity is a weak entity as it depends on DRIVER entity, SHFT\_ID serves as its primary key.
- BUS\_ID and BUS\_PLATE\_NUM are the composite primary keys of the BUS entity. A BUS entity is a strong entity as its composite primary key uniquely identifies each entity. BUS entity contain of TRIP\_STATUS\_ID from trip status entity, BUS\_TYPE and BUS\_COLOR
- TRIP\_STATUS entity contains PICKUP\_ID from PICKUP\_POINT entity as primary and foreign key. DRIVER\_ID from DRIVER entity also serves as primary key and foreign key. TRIP\_STATUS relationship is considered a weak entity.
- A PASSENGER entity is a strong entity, as it does not depend on any entity. The primary key is PASSENGER entity and the foreign key is DRIVER\_ID.

### ***Clarification***

- In the DRIVER entity, each driver has their own DRIVER\_ID, their DRIVER\_ID is given to them by the company so that each driver can be identified by the system and their records can be kept accordingly. DRIVER\_ID and DRIVER\_PHONENUM are the primary keys which uniquely identify the entity, other attributes involves in the entity involves DRIVER\_ADDRESS, SHIFT\_ID, BUS\_ID, TICKET\_ID, TRIP\_STATUS\_ID.
- In PASSENGER entity, Passengers are given ID after booking ticket for bus, PASSENGER\_ID uniquely identify each passengers in the database, other attributes involves are PASSENGER\_ID, PASSENGER\_NAME, PASSENGER\_GENDER and DRIVER\_ID
- In the BUS entity, BUS\_ID is given to the company's buses to be able to identify each bus and to describe the buses easily. BUS\_ID and BUS\_PLATE\_NUM are the primary keys which uniquely identify the entity, other attributes involves TRIP\_STATUS\_ID,BUS\_TYPE,BUS\_COLOR
- In the TICKET entity, tickets are generated after passengers book a bus for their trip, tickets serve as evidence of payment and will contain all the information the passengers need to know. TICKET\_ID uniquely identifies each passenger's ticket from each other. Other attributes involved in the TICKET entity are NO\_OF\_SEATS, DEPATURE\_ADDRESS, DEPATURE\_TIME.

### 3.5 CANCELLATION AND REFUND MODULE

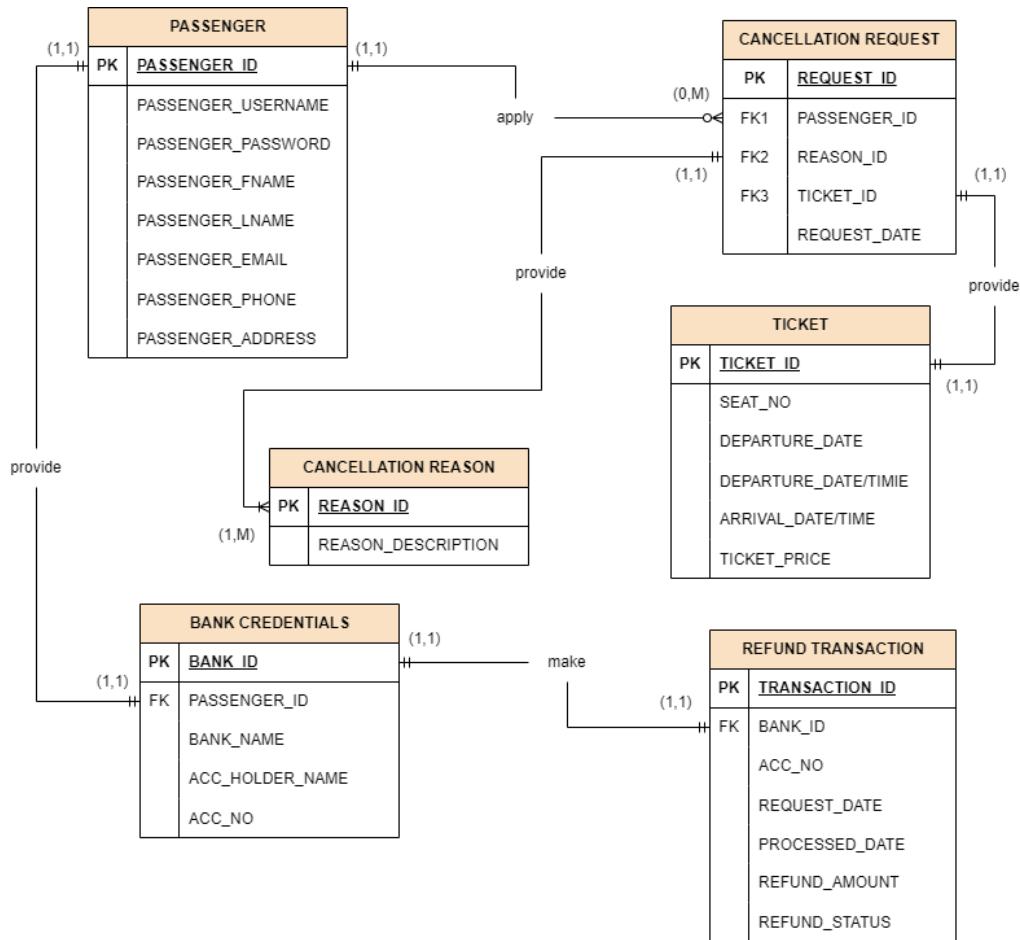


Figure 5: ERD Diagram for CANCELLATION AND REFUND Module

#### **Explanation**

Cancellation and Refund model has six entities namely; PASSENGER, CANCELLATION REQUEST, CANCELLATION REASON, TICKET, BANK CREDENTIALS, REFUND TRANSACTION.

#### **Relationship**

- PASSENGER and CANCELLATION REQUEST have one to many relationship, PASSENGER being mandatory and CANCELLATION\_REQUEST being optional (refer business rule 5.1). It forms a strong relationship, as both the entities use the same primary key, PASSENGER\_ID.

- CANCELLATION REQUEST and CANCELLATION REASON has one to many relationship. CANCELLATION REQUEST being mandatory (refer business rule 5.3). It forms weak relationships because they do not inherit primary keys from each other.
- CANCELLATION REQUEST and TICKET has a one to one relationship. Both of them are mandatory (refer business rule 5.4). It forms weak relationships because they do not inherit primary keys from each other.
- PASSENGER and BANK CREDENTIALS have a one to one relationship. Both of them are mandatory (refer business rule 5.2). It forms weak relationships because they do not inherit primary keys from each other.
- BANK CREDENTIALS and REFUND TRANSACTION have a one to one relationship (refer business rule 5.5). Both of them are mandatory. It forms a strong relationship, as both the entities use the same primary key, BANK\_ID.

### ***Key Constraints***

- PASSENGER entity has PASSENGER\_ID as primary key which uniquely identifies all the attributes.
- CANCELLATION\_REQUEST entity has composite primary key, namely, REQUEST\_ID and PASSENGER\_ID. Latter being the foreign key, which refers to the PASSENGER entity. There are another two foreign keys namely REASON\_ID and TICKET\_ID where former refers to CANCELLATION\_REASON entity and latter refers to TICKET entity.
- BANK\_CREDENTIALS has BANK\_ID as a primary key.
- TICKET has a primary key called TICKET\_ID.
- The BANK\_CREDENTIALS entity has a primary key called BANK\_ID.
- The REFUND\_TRANSACTION entity has a composite primary key called TRANSACTION\_ID and BANK\_ID. It has foreign key called BANK\_ID which refers to BANK\_CREDENTIALS.

### ***Clarification***

- In a PASSENGER entity, PASSENGER\_ID is a primary key which is a unique identity after they are registered in a system. The other attributes are PASSENGER\_USERNAME, PASSENGER\_PASSWORD, PASSENGER\_FNAME, PASSENGER\_LNAME, PASSENGER\_EMAIL, PASSENGER\_PHONE, PASSENGER\_ADDRESS. They represent the passenger's user name, passenger's password, passenger's first name, passenger's last name, passenger's email address, passenger's phone number, passenger's address respectively.

- In CANCELLATION\_REQUEST entity, REQUEST\_ID is a primary key which is uniquely identifies each and every requests done by passengers and PASSENGER\_ID is a primary key defining ID of a passenger, it is also a foreign key referring to PASSENGER entity. The other attributes are REASON\_ID, TICKET\_ID, REQUEST\_DATE. They represent the passenger's reason for canceling the ticket, the ticket's ID that they are going to cancel, and the date that they are going to cancel.
- In the TICKET entity, TICKET\_ID is a primary key which uniquely identifies each ticket from other tickets. The other attributes are SEAT\_NO( the seats number that passenger booked), DEPARTURE\_DATE(the date they are going to move), DEPARTURE\_TIME( the exact time that they are going to depart from destination), ARRIVAL\_TIME( time that they are going to reach the destination), TICKET\_PRICE( the price for the ticket).
- In CANCELLATION\_REASON entity,REASON\_ID is a primary key which uniquely identifies the bank used for ticket booking. There is one attribute namely REASON\_DESCRIPTION which states the reason for canceling the ticket.
- In the BANK\_CREDENTIALS entity, BANK\_ID is a primary key which uniquely identifies bank details that passengers put in. The other attributes are PASSENGER\_ID( the ID used for booking), BANK\_NAME( the bank name used of a passenger), ACC HOLDER\_NAME( the name of bank holder), ACC\_NO(account number of a passenger).
- REFUND\_TRANSACTIONS entity, it has a composite key namely, TRANSACTION\_ID which defines the ID of a transaction and BANK\_ID which uniquely identifies the bank ID, it is also a foreign reference to BANK\_CREDENTIALS entity. The other attributes are ACC\_NO( the account number of a bank), REQUEST\_DATE(the date requested by passenger), PROCESSED\_DATE( the date refund being processed), REFUND\_AMOUNT( the refund amount ), REFUND\_STATUS( the status of a refund).

### **3.6 FEEDBACK AND RATING MODULE**

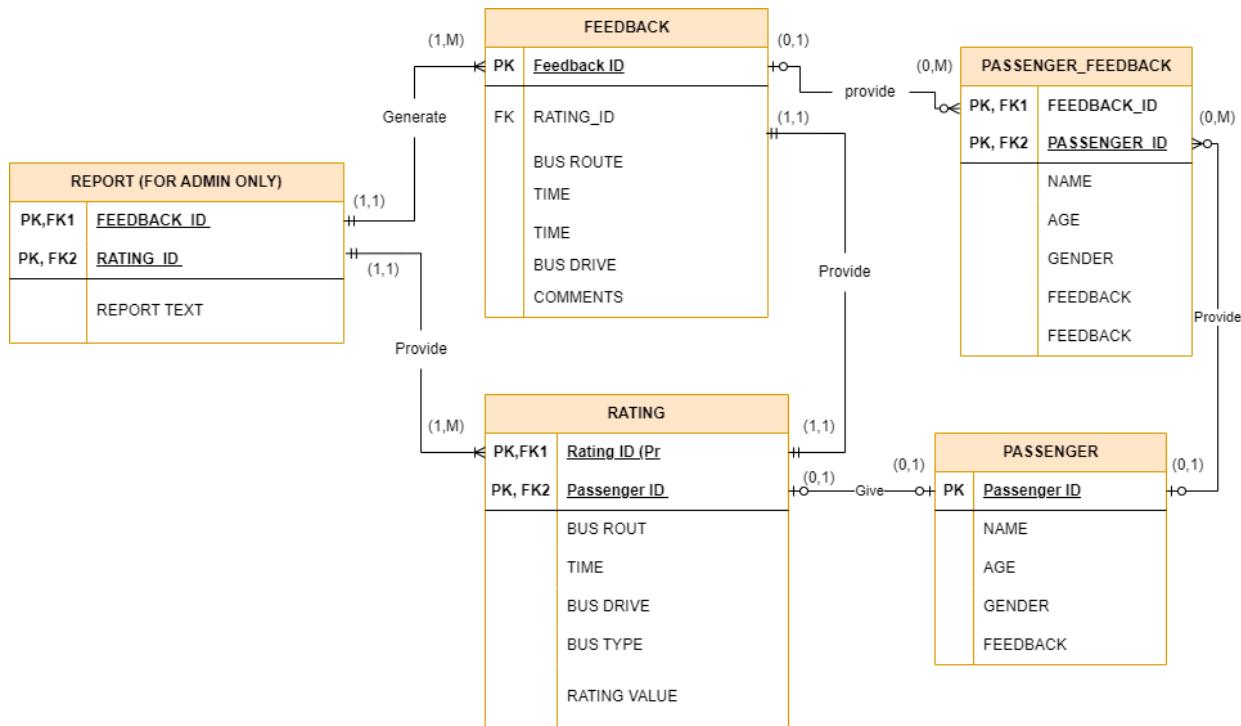


Figure 6: ERD Diagram for FEEDBACK and RATING Module

#### **Explanation**

Feedback and Rating Module contain 5 entities, which are Feedback, Passenger, Feedback, Report for admin , Rating, and Passenger.

#### **Relationships:**

- FEEDBACK entity and REPORT entity has one-to-many relationship, REPORT being mandatory. (refer to business rule 6.3) This is a STRONG relationship because both entities share the same primary key.
- REPORT entity and RaATING entity has a one-to-many relationship (refer to business rule 6.2) which is also a weak relationship.
- FEEDBACK entity and PASSENGER\_FEEDBACK entity has a one-to-many relationship, both entities being optional (refer to business rule (6.1). It is a strong relationship because PASSENGER\_FEEDBACK inherited the primary key of the FEEDBACK entity.

- PASSENGER\_FEEDBCK entity and PASSENGER entity has a one-to many relationship (refer to business rule (6.1).PASSENGER entity acts as a bridge entity between the two entities.
- PASSENGER entity has a weak relationship between both PASSENGER\_FEEDBACK entity and RATING entity .

### ***Key Constraints***

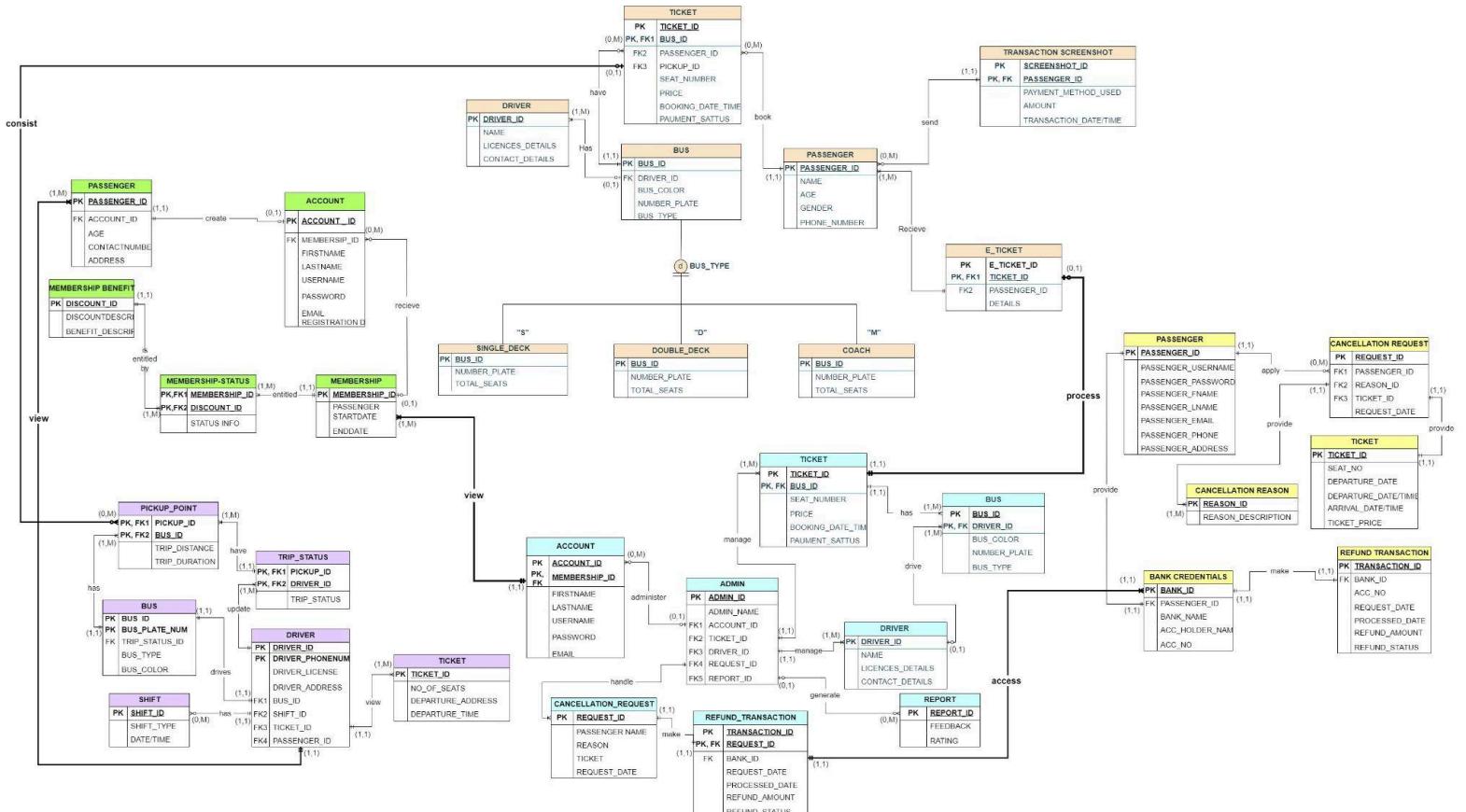
- In the FEEDBACK entity, the primary key is FEEDBACK\_ID which uniquely identifies all other attributes. It contains foreign keys called RATING\_ID which refers to a RATING entity.
- In the PASSENGER\_FEEDBACK entity, FEEDBACK\_ID and PASSENGER\_ID are composite primary keys. This composite primary key are also foreign keys from FEEDBACK\_ID and PASSENGER\_ID respectively. to determine NAME, AGE, GENDER, AND FEEDBACK.
- PASSENGER entity has primary key PASSENGER\_ID to determine \_NAME, \_GENDER,\_FEEDBACK,\_AND AGE
- RATING entity, RATING\_ID(Pr) and PASSENGER\_ID are composite primary keys. This composite primary key is also foreign keys from FEEDBACK\_ID and PASSENGER\_ID. to determine BUS ROUTE, BUS DRIVE, TIME, BUS TYPE.
- In the REPORT entity, FEEDBACK\_ID and RATING\_ID are composite primary keys to determine REPORT TEXT. These composite primary keys are also foreign keys from REPORT entity and FEEDBACK entity .

### ***Clarification***

- In the PASSENGER entity, the primary key PASSENGER\_ID is a unique ID after they are recorded into the system. The other attributes such as NAME, AGE, FEEDBACK and, GENDER.
- In the REPORT ( FOR ADMIN ONLY) entity, REPORT TEXT Refers to submit a complaint to resolve recurring complaints.
- In an FEEDBACK entity, FEEDBACK\_ID is a primary key which is a unique ID. foreign key which is a RATING\_ID is created in order to sign in into the application after that the passenger completes some important details such as BUS ROUTE. TIME, BUS DRIVE, COMMENTS.
- In the RATING entity,the RATING\_ID and PASSENGER\_ID both have primary key and foreign key,
- The other attributes such as BUS ROUTE . TIME, BUS DRIVE, BUS TYPE, RANTIG VALUE .

- In the PASSENGER entity, PASSENGER\_ID is the ID of the Passenger status whereas PASSENGER\_NAME, AGE, GENDER AND FEEDBACK to describe details of the passenger.

## 3.7 COMBINED-ERD



*Figure 7: ERD Diagram for entire KARMA TRANSPORT database*

## *Explanation*

This ERD is the combination of six modules. It contains 36 entities and 32 relationships in total. After the combination, there are four relationships that overlap each other

### ***Relationships***

- The relationship between DRIVER entity and PASSENGER entity has one to many relationships, PASSENGER entity being mandatory to the other entity (refer business rule 7.1). This is a weak relationship since one entity does not inherit the primary key of another entity.
- The relationship between TICKET and PICKUP\_POINT entities have one to many relationships, both being optional to each other (refer business rule 7.3). It is a weak relationship.
- The relationship between two TICKET and E\_TICKET entities have one to one relationship, the initial being optional and latter being mandatory (refer business rule 7.2). It is a strong relationship because both entities have common primary keys, called TICKET\_ID.
- REFUND\_TRANSACTION and BANK\_CREDENTIALS have one to many relationship, the initial being mandatory towards the latter (refer business rule 7.4). It is a weak relationship as it does not inherit the primary key of each other.
- ACCOUNT and MEMBERSHIP entities have one to many relationships, where ACCOUNT entities are mandatory (refer to business rule 7.5). They have a strong relationship as they inherit primary keys from each other.

## 4. NORMALIZATION

Normalization is practically a database schema design approach that involves the objectives of modifying an existing schema to effectively reduce data redundancies and reliance. In order to improve the clarity of data organization, a normalization approach is used to separate a bigger table into several smaller tables at which relationships are defined between them. The forms that are lower than that of 3NF are considered to have data redundancy that leads to anomalies such as insertion, deletion or update ones. Moreover, partial and transitive dependencies do not exist in 3NF tables. However, the ones that consist of more than one candidate key are not defined as BCNF. Furthermore, any forms that are specifically higher than 3NF are considered as unnecessary as those tables are only required to obtain specific results.

### 4.1 Admin Module

Most of the tables are in 3NF, which do not contain partial or transitive dependencies. But in the ACCOUNT table, there's a partial dependency where password only depends on ACCOUNT\_ID and not the MEMBERSHIP\_ID. Moreover, the TICKET table has partial dependency as well, where bus info are not dependent on DRIVER\_ID.

TABLE: ADMIN (3NF)

Functional Dependency

ADMIN_ID	ADMIN_NAME	ACCOUNT_ID	TICKET_ID	DRIVER_ID	REQUEST_ID	REPORT_ID	PASSWORD
----------	------------	------------	-----------	-----------	------------	-----------	----------

TABLE: ACCOUNT (2NF)

Functional Dependency

ACCOUNT_ID	MEMBERSHIP_ID	FIRST_NAME	LAST_NAME	USERNAME	EMAIL	PASSWORD
↑ Partial Dependency						

TABLE: ACCOUNT (3NF)

Functional Dependency

ACCOUNT_ID	FIRST_NAME	LAST_NAME	EMAIL	PASSWORD
------------	------------	-----------	-------	----------

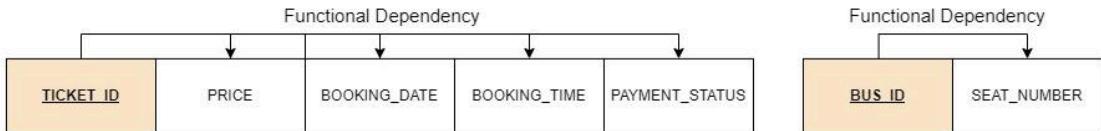
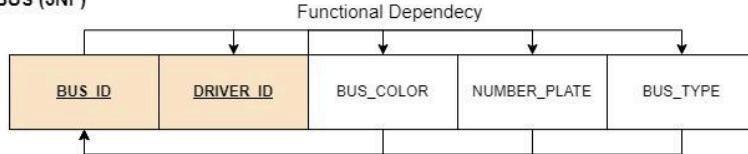
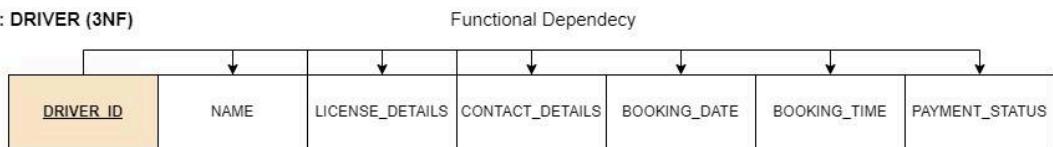
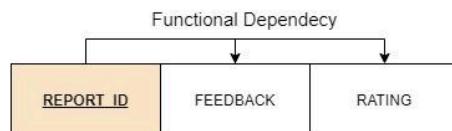
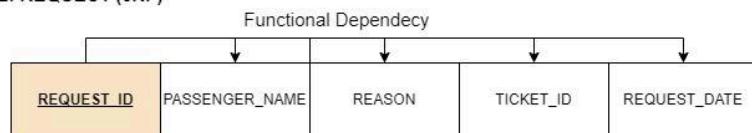
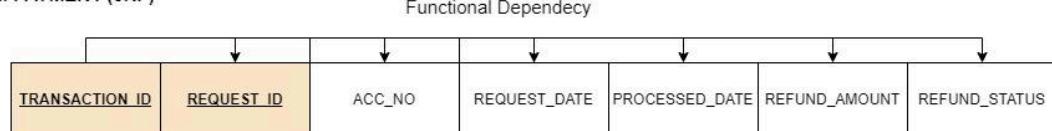
Functional Dependency

ACCOUNT_ID	USERNAME	PASSWORD
------------	----------	----------

TABLE: TICKET (3NF)

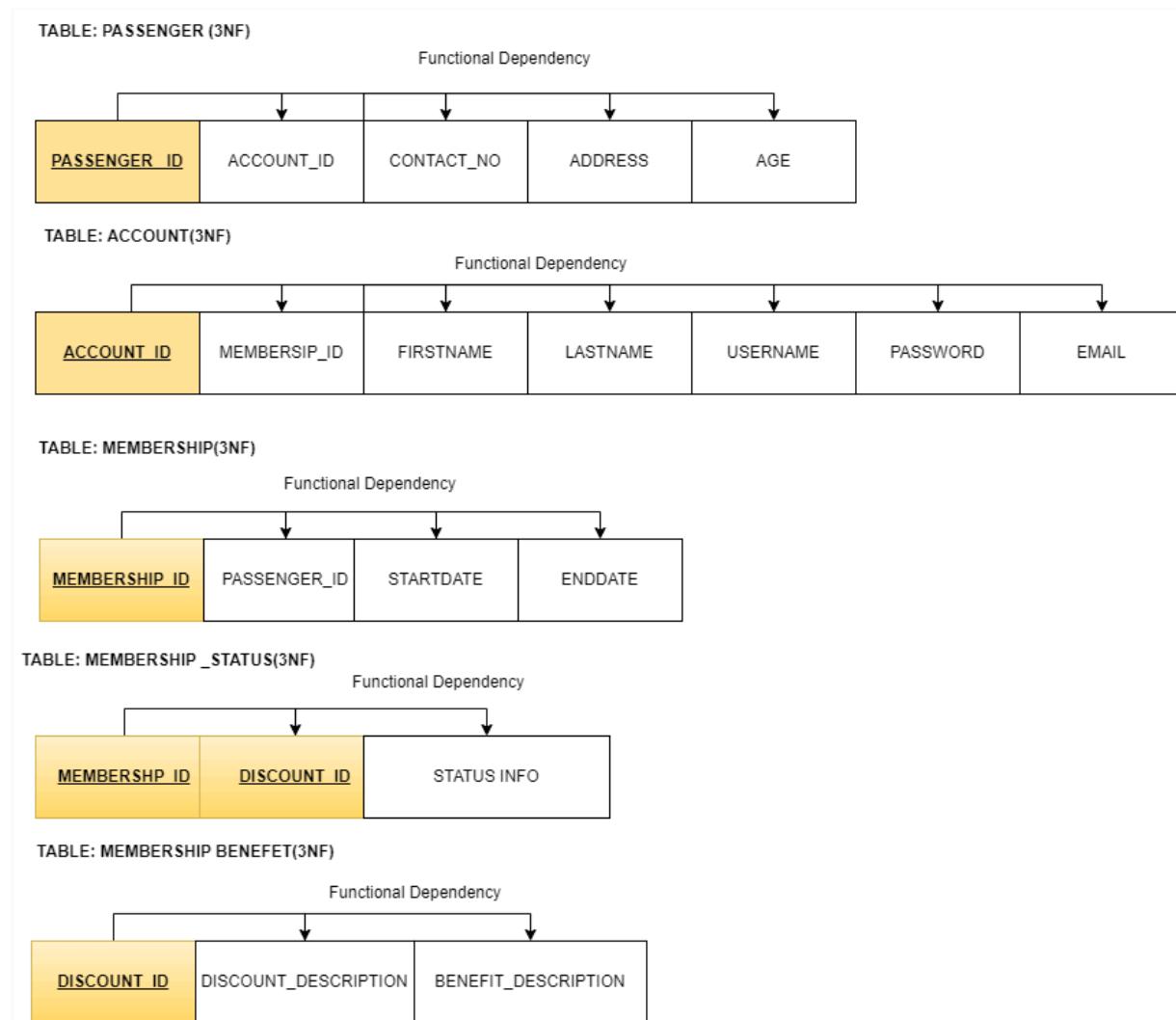
Functional Dependency

TICKET_ID	BUS_ID	SEAT_NUMBER	PRICE	BOOKING_DATE	BOOKING_TIME	PAYMENT_STATUS
-----------	--------	-------------	-------	--------------	--------------	----------------

**TABLE: BUS (3NF)****TABLE: BUS (3NF)****TABLE: DRIVER (3NF)****TABLE: REPORT (3NF)****TABLE: REQUEST (3NF)****TABLE: PAYMENT (3NF)**

## **4.2 Passenger Module**

All five tables in the Ticket Module are in 3NF due to no partial and transitive dependencies. The below shows the dependency diagrams of the tables. Partial and transitive dependencies do not exist.



## **4.3 Ticket Module**

All nine tables in the Ticket Module are in 3NF due to no partial and transitive dependencies. The below shows the dependency diagrams of the tables. Partial and transitive dependencies do not exist.

TABLE: TICKET(3NF)

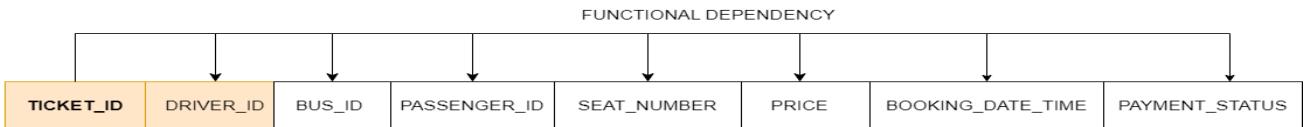


TABLE: E\_TICKET(3NF)

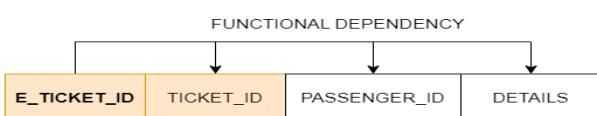


TABLE: BUS(3NF)

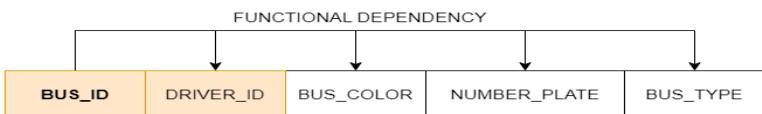


TABLE: DRIVER (3NF)

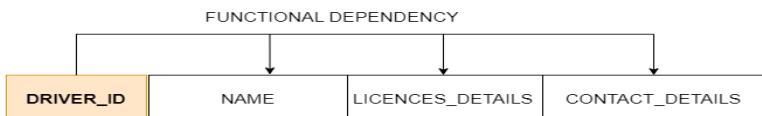


TABLE: PASSENGER(3NF)

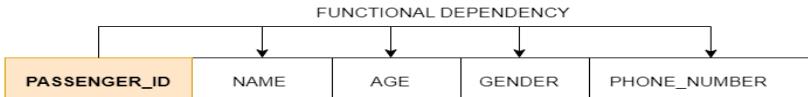


TABLE: TRANSACTION\_SCREENSHOT (3NF)

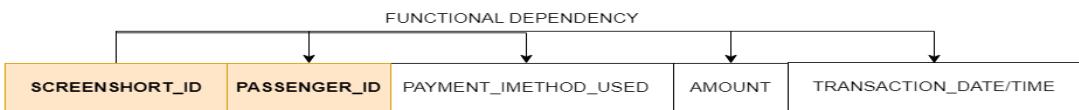


TABLE: SINGLE\_DECK (3NF)

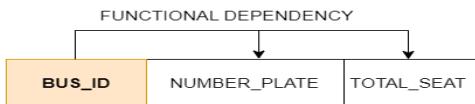


TABLE: DOUBLE\_DECK (3NF)

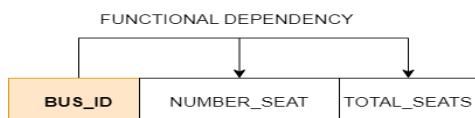
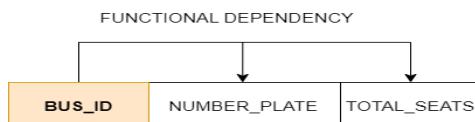


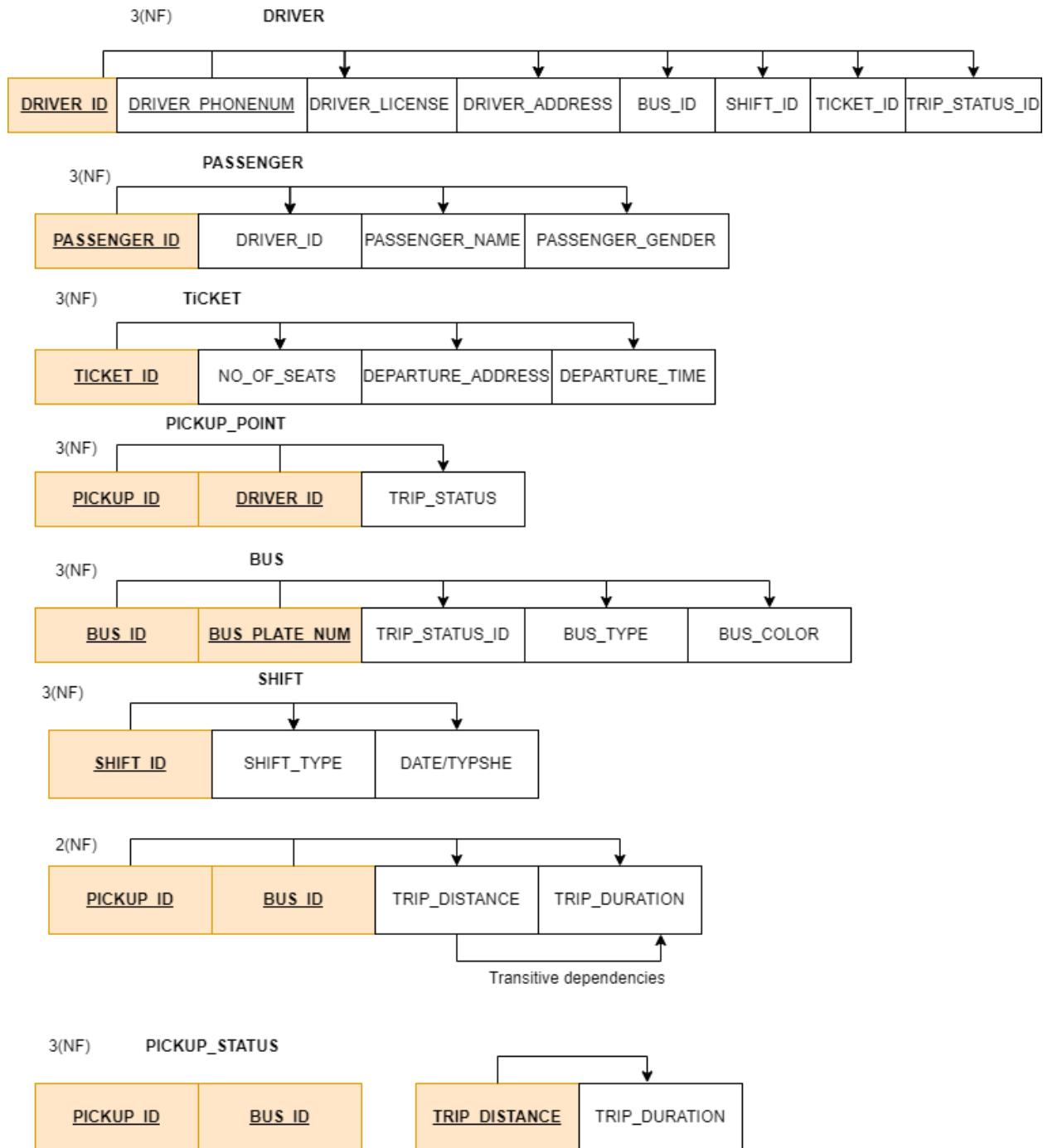
TABLE: COACH(3NF)



#### **4.4 Driver Module**

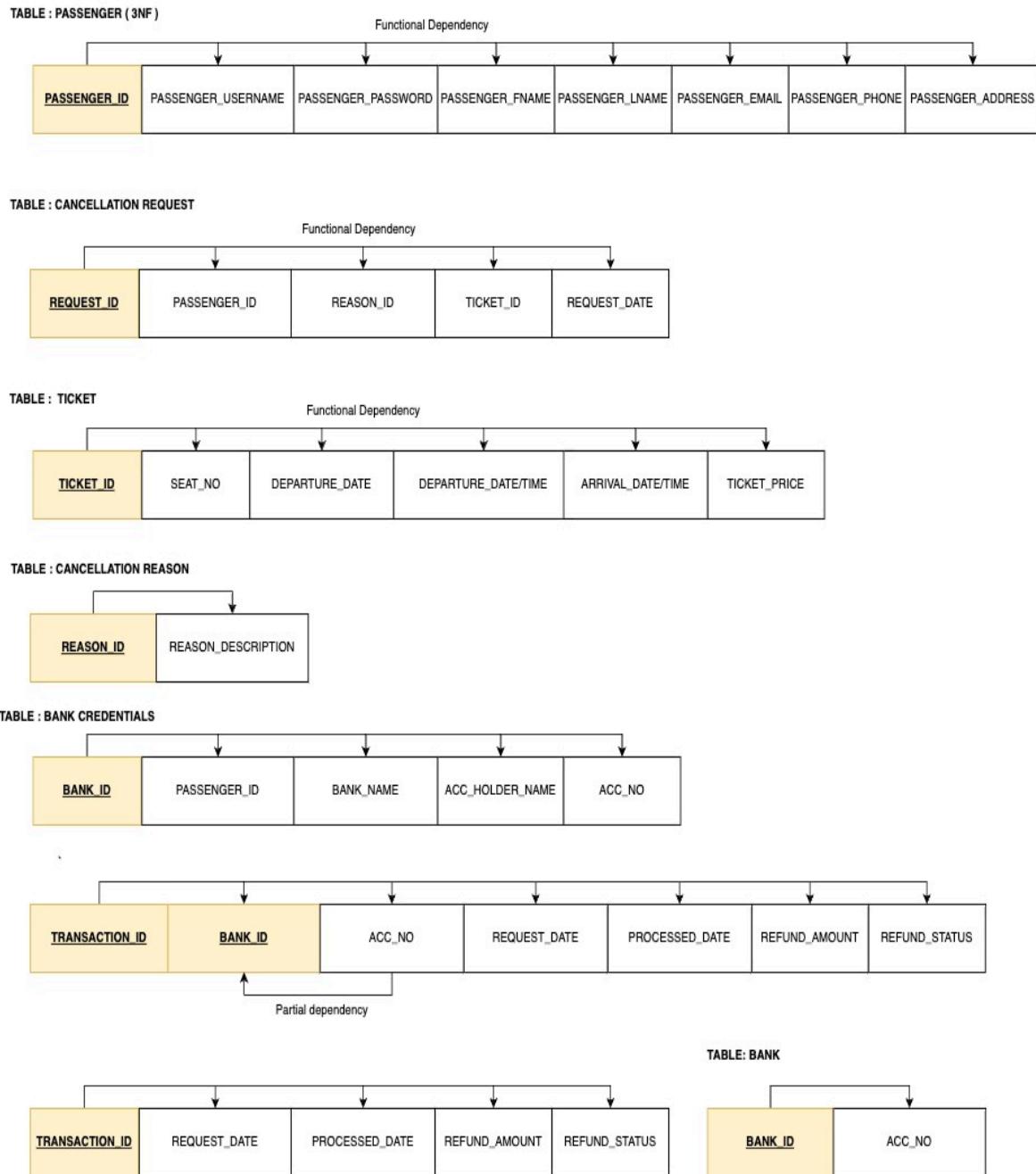
All tables are in 3NF except the second to the last table which is 2NF because it contain of transitive dependency which was later made to 3NF as we name the table TRANSPORT\_INFO.

After making a lot of changes there are 9 tables in total for the DRIVER MODULE.



#### 4.5 Cancellation Module and Refund Module

Four tables are in 3NF due to the functional dependencies on all its attributes. **Bank Credentials** table has partial dependencies on it since one attribute partially depends on the primary key rather fully depending on it. So we broke it down and made it in the third normal form.

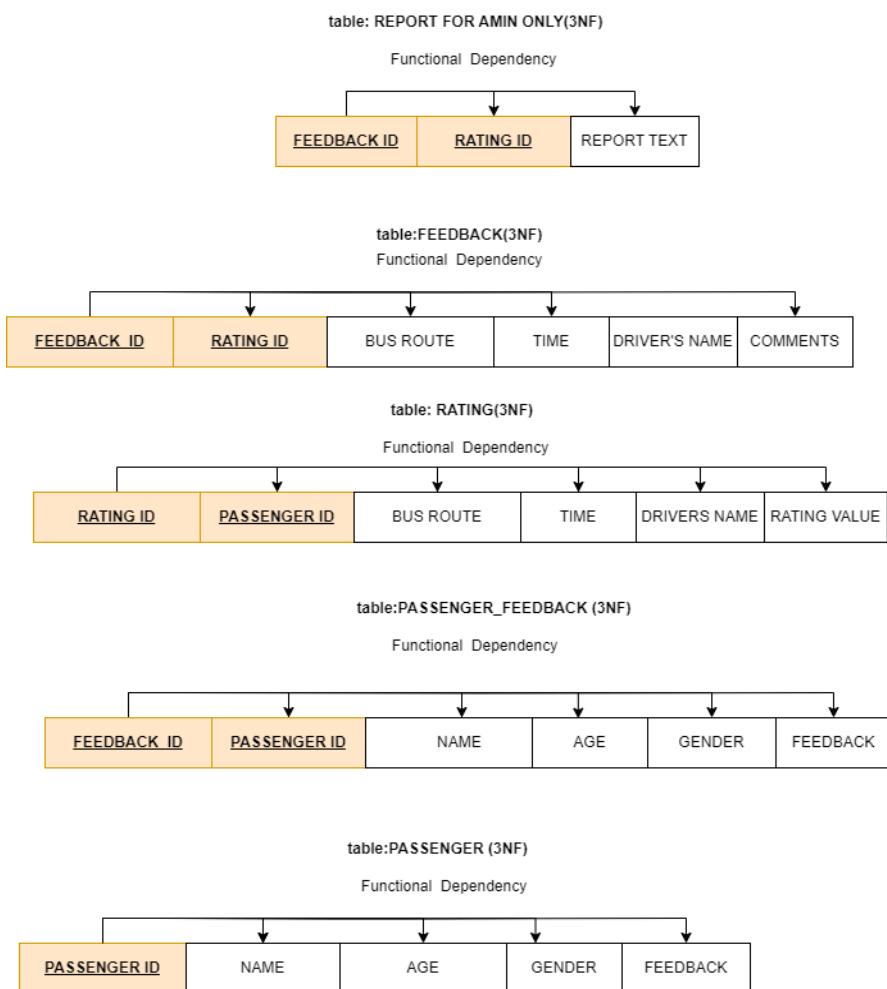


#### **4.6 Feedback and Rating Module.**

All six tables in the Rating module are in 3NF due to no partial and transitive dependencies

Below shows the dependency diagrams of the tables.

NOTE: All attributes in table PASSENGER FEEDBACK which is PASSENGER\_ID and FEEDBACK\_ID are primary keys.



## 5. DATA DICTIONARY

### 5.1 Admin Module

Table Name	Attribute Name	Contents	Data Types	Format	NULL available	PK/FK	Reference Table
ADMIN	ADMIN_ID	Admin's id	varchar(5)	A1111	No	PK	
	ACCOUNT_ID	Passenger's Account id	varchar(10)	KT00000000	No	FK	ACCOUNT
	TICKET_ID	Passengers ticket id	varchar(7)	AA12345	No	FK	TICKET
	DRIVER_ID	Driver's id	varchar(5)	D1234	No	FK	DRIVER
	REQUEST_ID	Cancellation request date	varchar(5)	R1234	No	FK	CANCELLATION REQUEST
	REPORT_ID	Report id	varchar(7)	RR12345	No	FK	REPORT
ACCOUNT	ACCOUNT_ID	Passengers account id	varchar(10)	KT00000000	No	PK	
	MEMBERSHIP_ID	Passengers membership id	varchar(7)	MM12345	Yes	FK	
	FIRST_NAME	Passengers first name	text(20)	Karma	No		
	LAST_NAME	Passengers last name	text(20)	Karma	Yes		
	USERNAME	username	varchar(20)	karma123	No		
	PASSWORD	password	varchar(20)	123abc	No		

Table Name	Attribute Name	Contents	Data Types	Format	NULL available	PK/FK	Reference Table
	EMAIL	email	varchar(20)	karma123@gmail.com	No		
TICKET	TICKET_ID	Ticket id	varchar(7)	AA12345	No	PK	
	BUS_ID	Bus id	varchar(10)	B123	No	PK, FK	BUS
	TICKET_PRICE	Ticket price in (MYR)	varchar(5)	RM120	NO		
	DEPARTURE_DATE	Bus departure date	date	2024/01/24	NO		
	DEPARTURE_TIME	Bus departure time	time	00:00:00	NO		
	ARRIVAL_DATE	Bus arrival date	date	2024/01/24	NO		
	ARRIVAL_TIME	Bus arrival time	time	00:00:00	NO		
	PICKUP_ID	Bus pickup id	varchar(6)	BP1234	NO		
BUS	BUS_ID	Bus id	varchar(10)	B123	NO	PK	
	DRIVER_ID	Driver id	varchar(5)	D1234	NO	PK, FK	DRIVER
	BUS_COLOR	Bus color	char(10)	red	YES		
	NUMBER_PLATE	Bus number plate	varchar(10)	ABC1234	YES		

Table Name	Attribute Name	Contents	Data Types	Format	NULL available	PK/FK	Reference Table
	BUS_TYPE	Bus type whether single, double deck or coach	varchar(7)	single-deck	YES		
DRIVER	DRIVER_ID	Driver id	varchar(5)	D1234	NO	PK	
	DRIVER_NAME	Driver's name	text(20)	Karma	YES		
	DRIVER_LICENSE	Driver's license	varchar(10)	XXX1234	YES	PK	
	DRIVER_ADDRES S	Driver's address	varchar(20)	Jalan Tun Abdul Razak, Alor Setar, 05200	YES		
	CONTACT_NO	Drivers contact number	Number (15,0)	+60 1122334455	NO		
CANCELLATION_REQUEST	REQUEST_ID	Cancellation request id	varchar(5)	R1234	NO	PK	
	PASSENGER_NAME	Passengers name	text(20)	Karma	NO		
	REASON	Cancellation reason	text	Hello world	NO		
	TICKET_ID	Passengers ticket id	varchar(7)	AA12345	NO	FK	
	REQUEST_DATE	Request date	date	2024/01/24	NO		
	TRANSA	Transaction	varchar( )	ABC1234	NO	PK	

Table Name	Attribute Name	Contents	Data Types	Format	NULL available	PK/ FK	Reference Table
REFUND_TRANS ACTION	CTION_ID	id	20)				
	REQUEST_ID	Request id	varchar(5)	R1234	NO	PK, FK	CANCELLATION_REQUEST
	ACC_NO	Passengers account number	number(20,0)	02114523	NO		
	REQUEST_DATE	Passengers request date	date	2024/01/24	NO		
	PROCESS_ED_DATE	Refund process date	date	2024/01/24	NO		
	REFUND_AMOUNT	Refund amount	varchar(5)	RM120	NO		
	REFUND_STATUS	Whether refunded or not	char(3)	YES	NO		

## 5.2 Passenger Module

Table Name	Attribut e Name	Contents	Data Type	Format	Null availabl e	PKOr FK	Reference Table
PASSENGER	PASSENGER_ID	passenger's id	VARCHA R(10)	KT0000	No	PK	
	ACCOUNT_ID	account's_id	VARCHA R(10)	KT00000000	No	FK	Account_id

	CONTACT NUMBER	Contact number	NUMBER(15,0)	+601122334455	yes		
	ADDRESS	Passenger's addresses	VARCHAR (20)	Jalan Tun Abdul Razak,05200 , Alor Setar	yes		
	AGE	Passenger's age	NUMBER (*,0)	12	Yes		

ACCOUNT	Account_ID	Account's	VARCHAR (10)	KT00000000	No	PK	
	MEMBERSHIP_ID	membership	NUMBER (*,0)	1	No	FK	Membership
	FIRSTNAME	firstname	TEXT (20)	KARMA	No		
	LASTNAME	lastname	TEXT (20)	KARMA	No		
	USERNAME	username	VARCHAR (10)	KARMA123	No		
	PASSWORD	password	VARCHAR (20)	123ABC	No		
	EMAIL	Email	VARCHAR (50)	KARMA123@gmail.COM	Yes		

MEMBERSHIP	MEMBERSHIP_P_ID	passenger's membership id	VARCHAR(7)	MM12345	No	PK	
	PASSENGER	passenger's	NUMBER(*,0)	1	No		
	STARTDATE	startdate	date	1	No		
	ENDDATE	enddate	date	2024/01/24	No		
MEMBERSHIP_STATUS	MEMBERSHIP_P_ID	Passenger's membership id	VARCHAR(7)	MM12345	No	PK, FK	MEMBERSHIP
	DISCOUNT_ID	Passenger's Discount code	VARCHAR(20)	ABC123	No	PK, FK	MEMBERSHIP_BENEFIT
	STATUS_INFO	Membership Status	TEXT(3)	NO	No		
MEMBERSHIP_BENEFIT	DISCOUNT_ID	Discount ID	VARCHAR(20)	D12345	No	PK	
	DISCOUNT_DESCRIPTION	Description of discount	VARCHAR(50)	TICKET PRICE 10% OFF	No		

### **5.3 Ticket Module:**

Table Name	Attribute Name	Contents	Data Types	Format	NULL available	PK/FK	Reference Table
TICKET	TICKET_ID	Ticket	Varchar(7)	AA123 45	No	PK	
	DRIVER_ID	Driver	Varchar(5)	D1234	No	FK1	Driver
	BUS_ID	Bus's	Varchar(10)	B123	No	FK2	Bus's
	PASSENGER_ID	Passenger's	Varchar(10)	KT000 0	No	FK3	Passenger's
	SEAT_NUMBER	Seat numbers	varchar(5)	M12	No		
	PRICE	Price	Varchar(5)	RM120	No		
	BOOKING_DATE	Booking date and time	Date	2024/0 1/24	No		
	BOOKING_TIME	Booking time	Time	00:00:0 0	No		
	PAYMENT_STATUS	Payment status	text(3)	NO	No		
E_TICKET	E_TICKET_ID	Electronic ticket id	Varchar(7)	AA123 45	No	PK	
	TICKET_ID	Ticket	Varchar(7)	AA123 45	No	PK, FK1	Ticket
	PASSENGER_ID	Passenger	Varchar(10)	KT000 0	No	FK2	Passenger
BUS	BUS_ID	Bus's	Varchar(10)	B123	No	PK	
	DRIVER_ID	Driver's	Varchar(5)	D1234	No	FK	Driver's
	NUMBER_PLATE	Number plate	Varchar(10)	ABC12 34	No		
	BUS_COLOR	Bus color	char(10)	Read	Yes		

	BUS_TYPE	Bus type	Varchar(7)	Single; Deck	No		
	CONTACT_NUMBE R	Contact Details	Number(15, 0)	112233 4455	No		
PASSENG ER	PASSENGER_ID	Passenger's	Varchar(10)	KT000 0	No	PK	
	NAME	Name	Text(20)	Karma	No		
	AGE	Age	int(5)	34	No		
	GENDER	Gender	Text(10)	Male	No		
	PHONE_NUMBER	Phone number	int(15)	112233 4455	No		
TRANSA CTION_S CREENS HORT	SCREENSHOT_ID	Screenshots	varchar(7)		No	PK	
	PASSENGER_ID	Passenger's	Varchar(10)	KT000 0	No	PK, FK	Passeng er's
	PAYMENT_METHO D_USED	Payment method used	text(10)	Online transfer	No		
	AMOUNT	Amount	Varchar(5)	RM120	No		
	TRANSACTION_DA TE	Transaction date	Date	2024/0 1/24	No		
	TRANSACTION_TIM E	Transaction time	Time	00:00:0 0	No		
DRIVER	DRIVER_ID	Driver's	Varchar(5)	D1234	No	PK	
	NAME	Name's	Text(20)	Karma	No		
	LICENCES_NUMBE R	Licenses number	int(15)	112233 4455	No		
	CONTACT_NUMBE R	Contact details	int(15)	112233 4455	No		
SINGLE_DESK	BUS_ID	Bus's	Varchar(10)	B123	No	PK	
	NUMBER_PLATE	Number	Varchar(10)	ABC12	Yes		

		Plate		34			
	TOTAL_SEAT	Total seat	int(5)	12	Yes		
DOUBLE _DECK	BUS_ID	Bus's	Varchar(10)	B123	No	PK	
	NUMBER_PLATE	Number plate	Varchar(10)	ABC1234	Yes		
	TOTAL_SEAT	Total seat	int(5)	12	Yes		
COACH	BUS_ID	Bus's	Varchar(10)	B123	No	PK	
	NUMBER_PLATE	Number plate	Varchar(10)	ABC1234	Yes		
	TOTAL_SEAT	Total seat	int(5)	12	Yes		

#### **5.4 Driver module:**

Table Name	Attribute Name	Content s	Data Types	Format	NULL available	PK or FK	Reference Table
DRIVER	DRIVER_ID	Driver id	varchar(5)	D1234	NO	PK	
	DRIVER_NAME	Driver's name	text(20)	karma	NO		
	DRIVER_EMAIL	Driver's email	varchar(20)	Karma@123@gmail.com			
	DRIVER_LICENSE	Driver's license	VARCHAR (10)	XXX1234	NO		
	DRIVER_ADDRESS	Driver's Address	VARCHAR (20)	Jalan Tun Abdulrasak, Alor setar, 05200	NO		

	CONTACT_NO	Driver's phone number	NUMBER(15,0)	+601122334455	NO		
	BUS_ID	Bus id	varchar(10)	B123	NO	FK	BUS
	SHIFT_ID	Shift id	NUMBER(*0)		NO	FK	SHIFT
	TICKET_ID	Ticket's id	INT(15)	AA12345	NO	FK	TICKET
	TRIP_STATUS_ID	Trip status id	INT(15)		NO	FK	TRIP_STATUS
TICKET	TICKET_ID		varchar(7)	AA12345	NO	PK	
	SEAT_NO		varchar(5)	M12	NO		
	DESTINATION		text(10)		NO		
	DEPATURE_TIME		time	00:00:00	NO		
	DEAPRTURE_DATE		date	2024/01/30	NO		
	ARRIVAL_DATE	Arrival date	date	2024/01/24	YES		
	ARRIVAL_TIME	Arrival time	time	00:00:00	YES		
SHIFT	SHIFT_ID	Shift's id	varchar(10)	DS000	NO	PK	
	SHIFT_TYPE	Shift type	text(10)	morning	NO		
	DATE	Shift date	date	2024/01/24	YES		
	TIME	Shift time	time	00:00:00			
BUS	BUS_ID	Bus id	varchar(10)		NO	PK	
	NUM_PLATE	Plate	varchar(10)	abc1234	NO		

		number					
	TRIP_STATUS_ID	Trip status id	INT(15)		NO	FK	TRIP_STATUS
	BUS_TYPE	Bus type	varchar(7)	single-deck	YES		
	BUS_COLOR	Bus color	char(10)	red	YES		
PASSENGER	PASSENGER_ID	Passenger's id	varchar(10)	KT0000	NO	PK	
	DRIVER_ID	Driver's id	varchar(5)	D1234	NO	FK	DRIVER
	PASSENGER_NAME	Passengers name	varchar(10)		NO		
	PASSENGER_GENDER	gender	text(10)	Male	NO		
TRIP_STATUS	PICKUP_ID	Pickup's id	Varchar(6)	BP1234	NO	PK	
	DRIVER_ID	Driver's Id	varchar(5)		NO	FK	DRIVER
	TRIP_STATUS	Trip status	text(15)		YES		
PICKUP_POINT	PICKUP_ID	Pick up id	Varchar(6)		NO		
	BUS_ID	Bus id	varchar(10)		NO		
TRIP_INFO	TRIP_DISTANCE	Trip distance	text (20)		NO		
	TRIP_DURATION	Trip duration	varchar(10)		YES		

## **5.5 CANCELLATION AND REFUND MODULE**

Table Name	Attribute Name	Contents	Data Types	Format	NULL available	PK/FK	Reference Table
PASSENGER	PASSENGER_ID	Passenger's identity name	VARCHAR(10)	KT0000	NO	PK	
	PASSENGER_USERNAME	Passenger user name	VARCHAR(10)	KARMA123	NO		
	PASSENGER_PASSWORD	Assigned password	VARCHAR(20)	123ABC	NO		
	PASSENGER_FNAME	Passenger's first name	TEXT(20)	KARMA	NO		
	PASSENGER_LNAME	Passenger's Last name	TEXT(20)	KARMA	YES		
	PASSENGER_EMAIL	Passenger's email	VARCHAR(20)	karma@123gmail.com	NO		
	PASSENGER_PHONE	Passenger's phone number	NUMBER(15,0)	1122334455	NO		
	PASSENGER_ADDRESS	Passenger's address	VARCHAR(20)	JALAN TUN	YES		
CANCELLATION REQUEST	REQUEST_ID	Passenger's Request ID	VARCHAR(5)	R1234	NO	PK	
	PASSENGER_ID	Passenger's ID	VARCHAR(10)	KT0000	NO	FK	PASSENGER
	REASON_ID	Reason identity	VARCHAR(10)	CR0000	NO	FK	CANCELLATION REASON
	TICKET_ID	Ticket Identity	VARCHAR(7)	AA12345	NO	FK	TICKET

		)					
	REQUEST_DATE	Cancellation request date	DATE	2024/01/24	NO		
CANCELLATION_REASON	REASON_ID	Cancelation reason ID	TEXT	HELLO WORLD	NO	PK	

TICKET	TICKET_ID	Ticket ID	VARCHAR(7)	AA12345	NO	PK	
	SEAT_NO	Seat number	VARCHAR(5)	M12	NO		
	DEPARTURE_DATE	Departing date	DATE	2024/01/24	NO		
	DEPARTURE_TIME	Departure time	TIME	00:00:00	NO		
	ARRIVAL_DATE	Arrival date	DATE	2024/01/28	NO		
	ARRIVAL_TIME	Arrival time	TIME	00:00:00	NO		
	TICKET_PRICE	Ticket price	VARCHAR(5)	RM120	NO		
BANK_CREDENTIALS	BANK_ID	Bank Id	VARCHAR(10)	ABCD1234	NO	PK	
	PASSENGER_ID	Passenger Id	VARCHAR(10)	KT0000	NO		
	BANK_NAME	Bank name	TEXT	BANK MUAMA LAT	NO		
	ACC_HOLDER_NAME	Account holder name	TEXT(20)	KARMA	NO		
	ACC_NO	Account	NUMBER(20,	02114523	NO		

TICKET	TICKET_ID	Ticket ID	VARCHAR(7)	AA12345	NO	PK	
	SEAT_NO	Seat number	VARCHAR(5)	M12	NO		
	DEPARTURE_DATE	Departing date	DATE	2024/01/24	NO		
	DEPARTURE_TIME	Departure time	TIME	00:00:00	NO		
	ARRIVAL_DATE	Arrival date	DATE	2024/01/28	NO		
	ARRIVAL_TIME	Arrival time	TIME	00:00:00	NO		
	TICKET_PRICE	Ticket price	VARCHAR(5)	RM120	NO		
		number	0)				
REFUND TRANSACTION	TRANSACTION_ID	Transaction Id	VARCHAR(20)	ABC1234	NO	PK	
	ACC_NO	Bank account number	NUMBER(20,0)	02114523	NO		
	REQUEST_DATE	Requested date for transaction	DATE	2024/01/24	NO		
	PROCESSED_DATE	Processed date	DATE	2024/01/24	NO		
	REFUND_AMOUNT	Amount to be refund	VARCHAR(5)	RM120	NO		
	REFUND_STATUS	Refund status	CHAR(3)	YES	NO		
BANK	ACC_NO	Account number	NUMBER(20,0)	02114523	NO		
	BANK_ID	Bank ID	VARCHAR(10)	ABCD1234	NO		

## **5.6 Feedback and Rating module**

Table Name	Attribute Name	Contents	Data types	Format	NULL available	PK or FK	Reference Table
REPORT (FOR ADMIN ONLY)	FEEDBACK_ID	Feedback's id	varchar(10)	FB0000	NO	PK, FK	
	RATING_ID	Rating id	varchar(10)	FR0000		FK	
	REPORT	Passengers overall Report	VARCHAR(7)	RR12345			
FEEDBACK	FEEDBACK_ID	Feedback's id	Varchar(10)	FB0000		PK	
	RATING_ID	Rating id	Varchar(10)	FR0000		FK	
	BUS_ROUTE	Bus route	varchar(20)	From KL terminal			
	TIME	Time	time	00:00:00			
	DRIVERS NAME	Bus Drive	text(20)	Karma			
	COMMENTS	Comments	text	Hello World			
RATING	RATING ID	Rating id	Varchar(10)	FB0000		PK, FK	
	PASSENGER ID	Passenger's id	Varchar(10)	Kt0000		PK, FK	
	BUS_ROUTE	Bus Route	varchar(20)	From KL terminal			
	TIME	Time	time	00:00:00			
	BUS DRIVE	Bus Drive	Text(20)	karma			

	BUS TYPE	Bus type	varchar(7 )	Single-deck	00,00,00		
	RATING VALUE	Rating Value	Varchar	FR0000			
PASSENGER_FEEDBACK	FEEDBACK_ID	Feedback's id	Varchar(	FB0000		PK , FK	
	PASSENGER_ID	Passenger's id	Varchar(10)	Kt0000		PK , FK	
	NAME	Name	Text(20)	Karma	Yes		
	AGE	Age	number(3 ,0)	34			
	GENDER	Gender	text(10)	Male			
	FEEDBACK	Feedback	text	Hello world			

## **6. DATA/SQL IMPLEMENTATION**

CREATE TABLE PASSENGER(

    PASSENGER\_ID VARCHAR(10) NOT NULL,

    ACCOUNT\_ID VARCHAR(10) NOT NULL,

    CONTACT\_NUMBER INT(15),

    ADDRESS VARCHAR(20),

    AGE INT,

    PRIMARY KEY (PASSENGER\_ID)

);

```
CREATE TABLE ACCOUNT(
    ACCOUNT_ID VARCHAR(10) NOT NULL,
    MEMBERSHIP_ID INT(14) NOT NULL,
    FIRSTNAME TEXT(20) NOT NULL,
    LASTNAME TEXT(20) NOT NULL,
    USERNAME VARCHAR(20) NOT NULL,
    PASSWORD VARCHAR(20) NOT NULL,
    EMAIL VARCHAR(20),
    PRIMARY KEY (ACCOUNT_ID)
);

CREATE TABLE MEMBERSHIP (
    MEMBERSHIP_ID VARCHAR(7) PRIMARY KEY,
    PASSENGER INT(5) NOT NULL,
    START_DATE DATE NOT NULL,
    END_DATE DATE NOT NULL
);

CREATE TABLE MEMBERSHIP_STATUS(
    MEMBERSHIP_ID VARCHAR(7) NOT NULL,
    DISCOUNT_ID VARCHAR(20) NOT NULL,
    STATUS_INFO TEXT(3),
    FOREIGN KEY (MEMBERSHIP_ID) REFERENCES MEMBERSHIP(MEMBERS

```

```
HIP_ID),  
    FOREIGN KEY (DISCOUNT_ID) REFERENCES MEMBERSHIP_BENEFIT(D  
ISCOUNT_ID),  
    PRIMARY KEY (MEMBERSHIP_ID,DISCOUNT_ID)  
);
```

```
CREATE TABLE MEMBERSHIP_BENEFIT(  
DISCOUNT_ID VARCHAR(20) PRIMARY KEY,  
DISCOUNT_DESCRIPTION VARCHAR(50) NOT NULL,  
);
```

```
CREATE TABLE SHIFT(  
SHIFT_ID VARCHAR(10) PRIMARY KEY,  
SHIFT_TYPE TEXT(10) NOT NULL,  
DATE DATE,  
TIME TIME  
);
```

```
CREATE TABLE BUS(  
BUS_ID VARCHAR(10) NOT NULL,  
NUM_PLATE VARCHAR(10) NOT NULL,  
TRIP_STATUS_ID INT(15) NOT NULL,  
BUS_TYPE VARCHAR(7),  
BUS_COLOR CHAR(10),
```

```
FOREIGN KEY (TRIP_STATUS_ID) REFERENCES TRIP_STATUS(TRIP_STATUS_ID),  
PRIMARY KEY (BUS_ID,NUM_PLATE)  
);
```

```
CREATE TABLE PICKUP_POINT(  
    PICKUP_ID VARCHAR(6) NOT NULL,  
    BUS_ID VARCHAR(10) NOT NULL,  
    TRIP_DISTANCE TEXT(20) NOT NULL,  
    TRIP_DURATION VARCHAR(10),  
    FOREIGN KEY (BUS_ID) REFERENCES BUS(BUS_ID),  
    PRIMARY KEY (PICKUP_ID, BUS_ID)  
);
```

```
CREATE TABLE DRIVER(  
    DRIVER_ID VARCHAR(5) PRIMARY KEY,  
    DRIVER_NAME TEXT(20) NOT NULL,  
    DRIVER_EMAIL VARCHAR(20) NOT NULL,  
    DRIVER_LICENSE VARCHAR(10) NOT NULL,  
    DRIVER_ADDRESS VARCHAR(20) NOT NULL,  
    CONTACT_NO INT(15) NOT NULL,  
    BUS_ID VARCHAR(10) NOT NULL,  
    SHIFT_ID INT(15) NOT NULL,  
    TICKET_ID INT(15) NOT NULL,
```

```
    TRIP_STATUS_ID INT(15) NOT NULL  
    FOREIGN KEY (BUS_ID) REFERENCES BUS(BUS_ID),  
    FOREIGN KEY (SHIFT_ID) REFERENCES SHIFT(SHIFT_ID),  
    FOREIGN KEY(TICKET_ID) REFERENECES TICKET(TICKET_ID)  
);
```

```
CREATE TABLE TRIP_STATUS(  
  
    PICKUP_ID VARCHAR(6) NOT NULL,  
  
    DRIVER_ID VARCHAR(5) NOT NULL,  
  
    TRIP_STATUS TEXT(15),  
  
    FOREIGN KEY (PICKUP_ID) REFERENCES PICKUP_POINT(PICKUP_ID),  
  
    FOREIGN KEY (DRIVER_ID) REFERENCES DRIVER (DRIVER_ID),  
  
    PRIMARY KEY (PICKUP_ID, DRIVER_ID)  
  
);
```

```
CREATE TABLE BUS(  
  
    BUS_ID VARCHAR (10) PRIMARY KEY,  
  
    DRIVER_ID VARCHAR(5),  
  
    NUMBER_PLATE VARCHAR(10),  
  
    BUS_COLOR CHAR(10),  
  
    BUS_TYPE VARCHAR(7) NOT NULL,  
  
    CONTACT_NUMBER INT(15) NOT NULL,  
  
    FOREIGN KEY (DRIVER_ID) REFERENCES DRIVER(DRIVER_ID)
```

);

```
CREATE TABLE DOUBLE_DECK(  
    BUS_ID VARCHAR(10) PRIMARY KEY,  
    NUMBER_PLATE VARCHAR(10) NOT NULL,  
    TOTAL_SEAT INT(5)  
);
```

```
CREATE TABLE COACH (  
    BUS_ID VARCHAR(10) PRIMARY KEY,  
    NUMBER_PLATE VARCHAR(10) NOT NULL,  
    TOTAL SEAT INT(5)  
);
```

```
CREATE TABLE TICKET(  
    TICKET_ID VARCHAR(7) PRIMARY KEY,  
    DRIVER_ID VARCHAR(5),  
    BUS_ID VARCHAR(10),  
    PASSENGER_ID VARCHAR(10),  
    SEAT_NUMBER VARCHAR(5) NOT NULL,  
    PRICE VARCHAR(5) NOT NULL,  
    BOOKING_DATE DATE NOT NULL,  
    BOOKING_TIME TIME NOT NULL,  
    PAYMENT_STATUS TEXT(3) NOT NULL,
```

FOREIGN KEY (DRIVER\_ID) REFERENCES DRIVER(DRIVER\_ID),  
FOREIGN KEY (BUS\_ID) REFERENCES BUS(BUS\_ID),  
FOREIGN KEY (PASSENGER\_ID) REFERENCES PASSENGER(PASSENGER\_ID)

CREATE TABLE E\_TICKET(  
E\_TICKET\_ID VARCHAR(7) NOT NULL,  
TICKET\_ID VARCHAR(7) NOT NULL,  
PASSENGER\_ID VARCHAR(10) NOT NULL,  
FOREIGN KEY (TICKET\_ID) REFERENCES TICKET(TICKET\_ID),  
FOREIGN KEY (PASSENGER\_ID) REFERENCES PASSENGER(PASSENGER\_ID),  
PRIMARY KEY (E\_TICKET\_ID,TICKET\_ID)  
);

CREATE TABLE TRANSACTION\_SCREENSHOT(  
SCREENSHOT\_ID VARCHAR(7) NOT NULL,  
PASSENGER\_ID VARCHAR(10) NOT NULL,  
PAYMENT\_METHOD TEXT(10),  
AMOUNT VARCHAR(5) NOT NULL,  
TRANSACTION\_DATE DATE NOT NULL,  
TRANSACTION\_TIME TIME NOT NULL,  
FOREIGN KEY (PASSENGER\_ID) REFERENCES PASSENGER(PASSENGER\_ID),

```
PRIMARY KEY (SCREENSHOT_ID,PASSENGER_ID)
```

```
);
```

```
CREATE TABLE CANCELLATION_REASON(
```

```
REASON_ID TEXT,
```

```
PRIMARY KEY(REASON_ID)
```

```
);
```

```
CREATE TABLE CANCELLATION_REQUEST(
```

```
REQUEST_ID VARCHAR(5),
```

```
PASSENGER_ID VARCHAR(10),
```

```
REASON_ID VARCHAR(10),
```

```
TICKET_ID VARCHAR(7),
```

```
REQUEST_DATE DATE,
```

```
PRIMARY KEY(REQUEST_ID),
```

```
FOREIGN KEY (PASSENGER_ID) REFERENCES PASSENGER (PASSENEGER_ID),
```

```
FOREIGNKEY(REASON_ID) REFERENCES CANACELLATION_REASON (REASON_ID),
```

```
FOREIGN KEY (TICKET_ID) REFERENCES TICKET(TICKET_ID)
```

```
);
```

```
CREATE TABLE BANK_CREDENTIALS(
```

```
BANK_ID VARCHAR(10),
```

```
PASSENEGR_ID VARCHAR(10),  
BANK_NAME TEXT,  
ACC HOLDER_NAME TEXT(20),  
ACC_NO NUMBER(20,0),  
PRIMARY KEY(BANK_ID),  
FOREIGN KEY (PASSENGER_ID) REFERENCES PASSENGER (PASSENGER_ID)  
);
```

```
CREATE TABLE REFUND_TRANSACTION(  
TRANSACTION_ID VARCHAR(20),  
ACC_NO NUMBER(20,0),  
REQUEST_DATE DATE,  
PROCESSED_DATE DATE,  
REFUND_AMOUNT VARCHAR(5),  
REFUND_STATUS CHAR(3),  
BANK_ID,  
PRIMARY KEY(TRANSACTION_ID),  
FOREIGN KEY (BANK_ID) REFERENCES BANK CREDENTIALS (BANK_ID)  
);
```

```
CREATE TABLE RATING(  
RATING_ID VARCHAR(10),
```

```
PASSENGER_ID VARCHAR(10),  
BUS_ROUTE VARCHAR(20),  
TIME TIME,  
BUS_DRIVE TEXT(20),  
BUS_TYPE VARCHAR(7),  
RATING_VALUE VARCHAR,  
PRIMARY KEY(RATING_ID, PASSENGER_ID),  
FOREIGN KEY (RATING_ID) REFERENCES RATING (RATING_ID),  
FOREIGN KEY (PASSENGER_ID) REFERENCES PASSENGER (PASSENGER_ID)  
);
```

```
CREATE TABLE FEEDBACK(  
FEEDBACK_ID VARCHAR(10),  
RATING_ID VARCHAR(10),  
BUS_ROUTE VARCHAR(20),  
TIME TIME,  
DRIVERS_NAME TEXT(20),  
COMMENTS TEXT,  
PRIMARY KEY(FEEDBACK_ID),  
FOREIGN KEY (PASSENGER_ID) REFERENCES PASSENGER (PASSENGER_ID),  
FOREIGN KEY (RATING_ID) REFERENCES RATING (RATING_ID)
```

);

```
CREATE TABLE REPORT(
    FEEDBACK_ID VARCHAR(10),
    RATING_ID VARCHAR(10),
    REPORT VARCHAR(7),
    PRIMARY KEY(FEEDBACK_ID, RATING_ID),
    FOREIGN KEY (FEEDBACK_ID) REFERENCES FEEDBACK ( FEEDBACK_ID),
    FOREIGN KEY (RATING_ID) REFERENCES RATING (RATING_ID)
);
```

```
CREATE TABLE PASSENGER_FEEDBACK(
```

```
    FEEDBACK_ID VARCHAR,
    PASSENGER_ID VARCHAR(10),
    NAME TEXT(20),
    AGE INT(5),
    GENDER TEXT(10),
    FEEDBACK TEXT,
    PRIMARY KEY(FEEDBACK_ID, PASSENGER_ID),
    FOREIGN KEY (FEEDBACK_ID) REFERENCES FEEDBACK (FEEDBACK_ID),
    FOREIGN KEY (PASSENGER_ID) REFERENCES PASSENGER (PASSENGER_ID)
```

);

```
CREATE TABLE DRIVER(  
    DRIVER_ID VARCHAR(5),  
    DRIVER_NAME TEXT(20),  
    DRIVER_LICENSE VARCHAR(10),  
    DRIVER_ADDRESS VARCHAR(20),  
    CONTACT_NO NUMBER(20,0),  
    PRIMARY KEY(DRIVER_ID)  
);
```

```
CREATE TABLE CANCELLATION_REQUEST(  
    REQUEST_ID VARCHAR(5),  
    PASSENGER_NAME TEXT(20),  
    REASON TEXT,  
    TICKET_ID VARCHAR(7),  
    REQUEST_DATE DATE,  
    PRIMARY KEY(REQUEST_ID),  
    FOREIGN KEY (TICKET_ID) REFERENCES TICKET(TICKET_ID)  
);
```

## 7. FRONT-END SYSTEM DESIGN AND IMPLEMENTATION

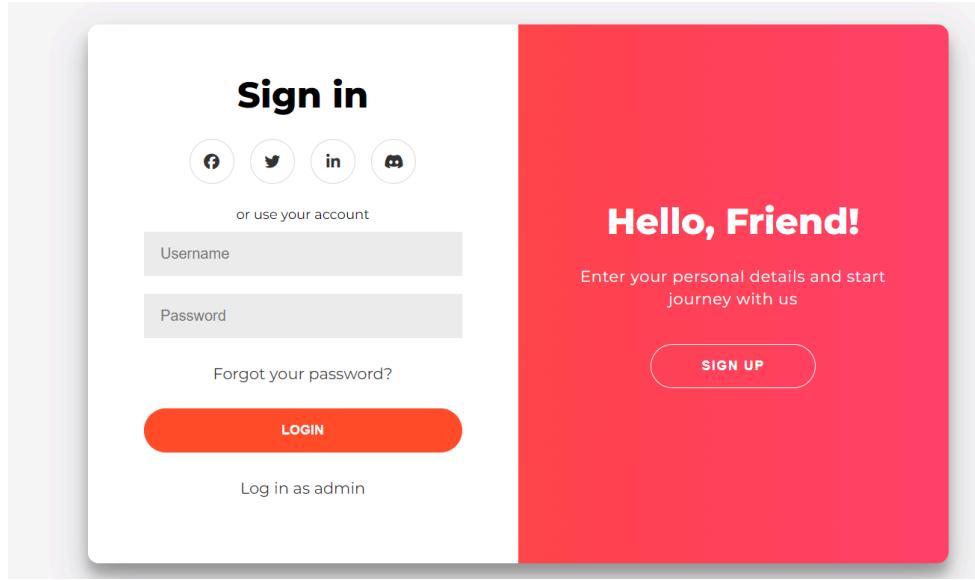


Figure 1: User Login Page

The figure above is the login page of our database. If passengers have already created an account, they can log in using their username and password. When they log in using their username and password, our system will check the database to validate the user and then only they can enter.

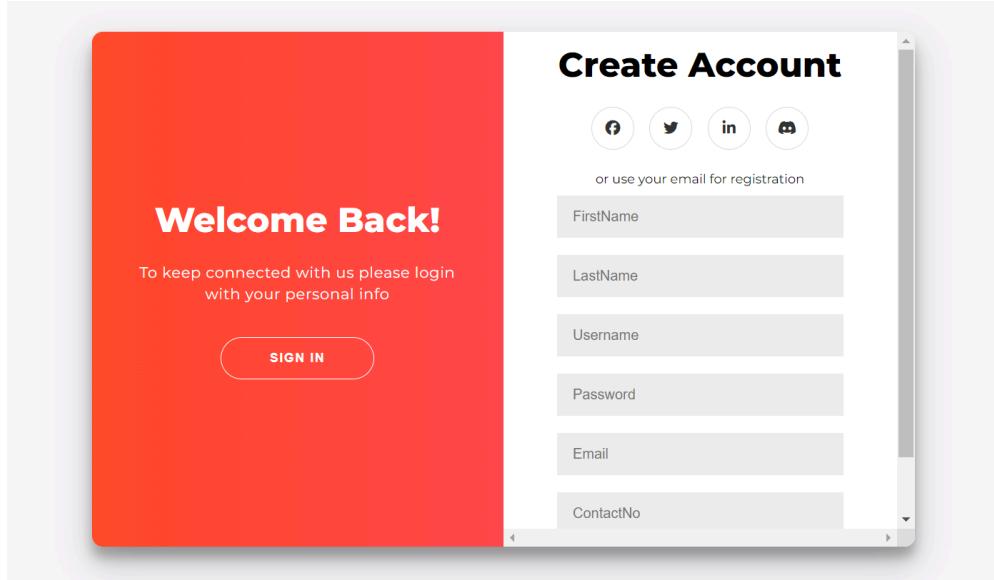


Figure 2: User Sign Up Page

Passengers who have not yet created their account, can click, 'SIGN UP' button. Then their information such as, FirstName, LastName, etc will be stored in our database.

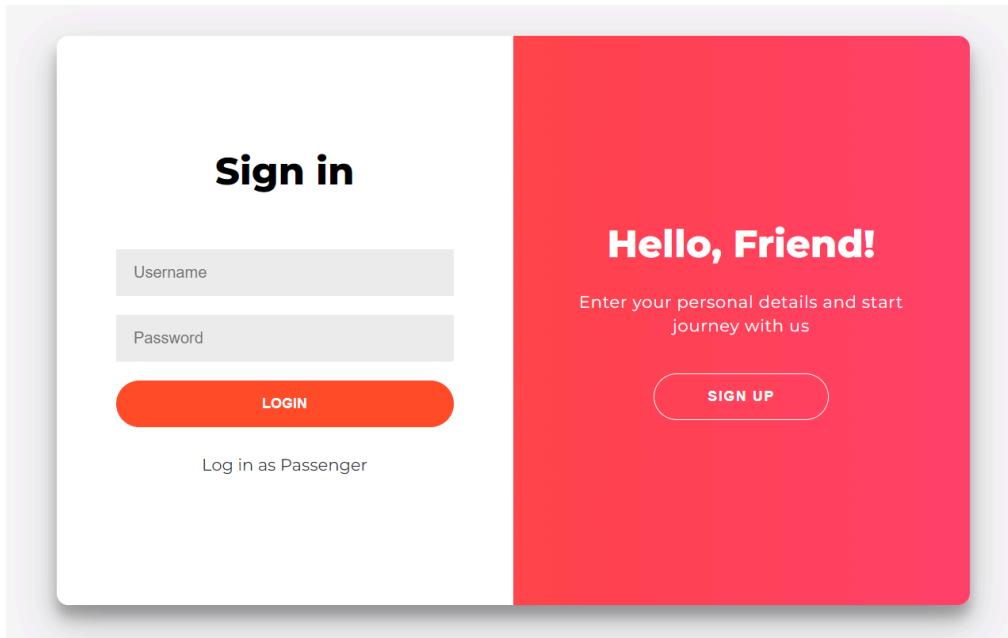


Figure 3: Admin Sign In Page

This is for the admin login. Only the admin can login with their correct credentials and have access to the whole system.

The image shows the Admin Dashboard. On the left is a dark sidebar with the 'Karma Transport' logo at the top. Below it are five menu items: 'Dashboard', 'User Account', 'Route', 'Bus', and 'Driver', each with an icon. At the bottom of the sidebar are a user profile picture, the name 'admin', and a 'log out' button. The main area of the dashboard is light blue and displays three green-bordered boxes showing statistics: 'Total No. of Users: 7', 'Total No. of Buses: 4', and 'Total No. of Drivers: 5', each preceded by an icon.

Figure 4: Admin Dashboard

After admin has successfully logged in, they will be directed to the Admin Dashboard page, where they can see the total number of passengers who registered into the system, total number of driver, and buses as well.

The screenshot shows a web-based application interface for managing passenger accounts. On the left, there is a vertical sidebar with icons for navigation. The main content area has a title "Manage Passengers' Account" at the top, followed by a red "EDIT" button. Below this is a table with columns: AccountID, FirstName, LastName, Username, Password, Email, and ContactNo. The table contains four rows of data. At the bottom right of the table is a red "SUBMIT" button. Below the table, there is a section titled "Add New User" with input fields for FirstName, LastName, Username, Password, Email, and ContactNo, followed by a red "ADD USER" button. Another section titled "Delete Old User" includes a "User ID:" input field and a red "DELETE USER" button.

AccountID	FirstName	LastName	Username	Password	Email	ContactNo
39	Alice	Johnson	alicej	*****	alicej@example.com	1234567890
40	Bob	Smith	bobsmith	*****	bob@example.com	2147483647
41	Emma	Brown	emmabrown	*****	emma@example.com	2147483647
42	Jack	Wilson	jackw	*****	jack@example.com	2147483647

Figure 5: Manage Passengers' Account

This page is only accessible to the admin user. If they wish to edit some information about the passenger, they can click the ‘edit’ button, located at the top and then click, ‘submit’ button to update the changes in the database. They can also add new users or delete existing ones too.

The screenshot shows a web-based application interface for managing bus routes. On the left, there is a vertical sidebar with icons for navigation. The main content area has a title "Manage Bus Routes" at the top, followed by a red "EDIT" button. Below this is a table with columns: DATE, BUS\_TYPE, DEPARTURE\_PLACE, ARRIVAL\_PLACE, DEPARTURE\_TIME, BUSARRIVAL\_TIME, PRICE, and SEATS\_AVAILABILITY. The table contains three rows of data. At the bottom right of the table is a red "SUBMIT" button. Below the table, there is a section titled "Add New Trip" with input fields for Date, Bus\_Type, Departure\_Place, Arrival\_Place, Departure\_Time, Arrival\_Time, Price (\$), and Seats Availability, followed by a red "ADD USER" button. Another section titled "Delete Old Trip" includes a "User ID:" input field and a red "DELETE USER" button.

DATE	BUS_TYPE	DEPARTURE_PLACE	ARRIVAL_PLACE	DEPARTURE_TIME	BUSARRIVAL_TIME	PRICE	SEATS_AVAILABILITY
2024-02-10	Coach	City A	City B	08:00:00	16:00:00	50.00	50
2024-02-11	Sing-Deck	City B	City C	09:00:00	17:00:00	75.50	30
2024-02-12	Double_Dec	City C	City D	10:00:00	18:00:00	45.75	20

Figure 6: Manage Bus Routes page

This page is only accessible to the admin user. If the admin wish to edit some information about the specific bus route, they can click the ‘edit’ button, located at the top and then click, ‘submit’ button to update the changes in the database. They can also add new routes/trips or delete existing ones too.

**Manage Buses**

**Bus ID** **Driver ID** **Bus Color** **Number Plate** **Bus Type**

KTB0001	D1001	Red	ABC 4567 A	Single Deck
KTB0002	D1002	White	WD 4567 C	Single Deck
KTB0003	D1003	Red	QAA 4567 C	Double Deck
KTB0004	D1004	White	SAB 4567 C	Single Deck

**Add New Bus**

**Bus ID** **Driver ID** **Bus Color** **Number Plate** **Bus Type** **ADD BUS**

**Delete Bus**

**Bus ID:**  **DELETE BUS**

**SUBMIT**

Figure 7:Manage Buses

This page is only accessible to the admin user. Here also, admin have the privilege to edit information on buses, add new buses or delete the existing ones.

**Manage Drivers**

**Driver ID** **Driver Name** **Email** **License** **Address** **Contact No** **Driver ID**

D001	John Doe	john@example.com	DL12345	123 Main St	1234567890	B001
D002	Jane Smith	jane@example.com	DL54321	456 Elm St	2147483647	B002
D003	Michael Johnson	michael@example.com	DL67890	789 Oak St	2147483647	B003
D004	Emily Brown	emily@example.com	DL09876	321 Pine St	2147483647	B004
D005	David Lee	david@example.com	DL24680	654 Cedar St	2147483647	B005

**Add New Driver**

**Driver Name** **Email** **License** **Address** **Contact No** **ADD DRIVER**

**Delete Driver**

**Driver ID:**  **DELETE DRIVER**

**SUBMIT**

Figure 8:Manage Drivers

Again, this page is only accessible to the admin user. Here also, admin have the privilege to edit information on drivers, add new drivers or delete the existing ones.

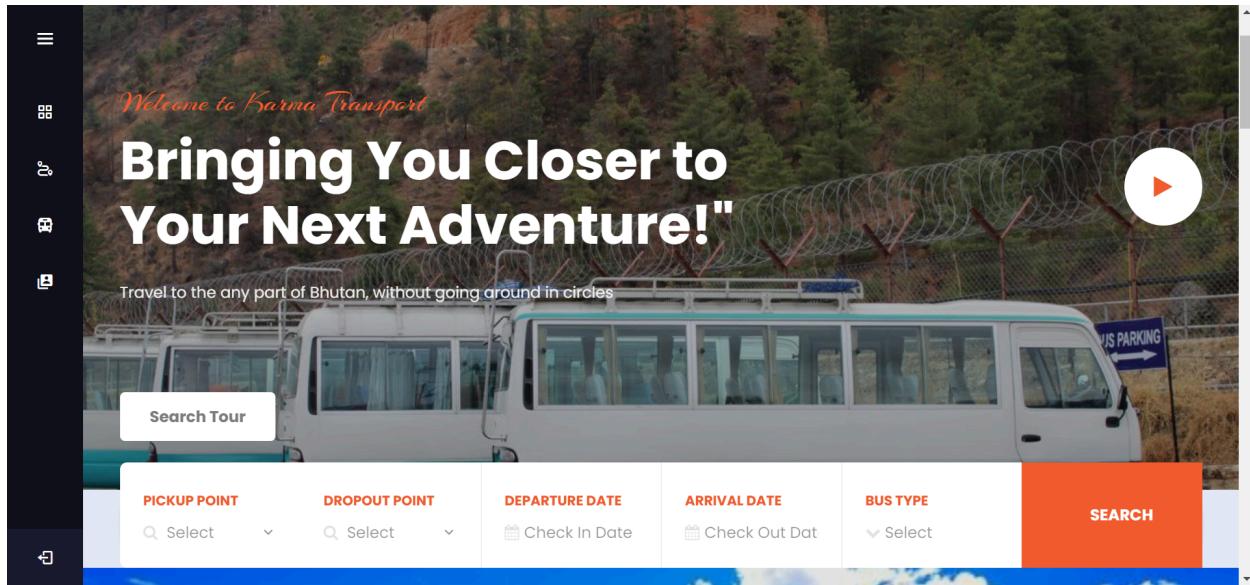


Figure 9: Passenger's Dashboard

After the user has successfully logged in, they are directed to this page, dashboard. Here, they can enter their pickup point, dropout point, and some details like bus type, and then they can click, ‘search’ button to scan our database and look for the desired bus routes.

DATE	BUS_TYPE	DEPARTURE_PLACE	ARRIVAL_PLACE	DEPARTURE_TIME	BUSARRIVAL_TIME	PRICE	SEATS	ACTION
2024-02-10	Coach	City A	City B	08:00:00	16:00:00	50.00	50	<button>BOOK</button>
2024-02-11	Sing-Deck	City B	City C	09:00:00	17:00:00	75.50	30	<button>BOOK</button>
2024-02-12	Double_Dec	City C	City D	10:00:00	18:00:00	45.75	20	<button>BOOK</button>
2024-02-10	Single-Dec	City A	City B	08:00:00	12:00:00	50.00	50	<button>BOOK</button>
2024-02-11	Double-Dec	City B	City C	09:00:00	13:00:00	60.00	40	<button>BOOK</button>
2024-02-12	Coach	City C	City D	10:00:00	14:00:00	70.00	30	<button>BOOK</button>
2024-02-13	Single-Dec	City D	City E	11:00:00	15:00:00	80.00	20	<button>BOOK</button>
2024-02-14	Double-Dec	City E	City F	12:00:00	16:00:00	90.00	10	<button>BOOK</button>
2024-02-15	Coach	City F	City G	13:00:00	17:00:00	100.00	5	<button>BOOK</button>

Figure 10: Passengers Trip/Routes list (view-only)

This page shows all the routes/trips that will take place in a week. So passengers can view and they can book the ticket by clicking on the ‘book’ button, located at the end of each trip.

User Account Info Details

AccountID	FirstName	LastName	Username	Password	Email	ContactNo
39	Alice	Johnson	alicej	*****	alicej@example.com	1234567890

**SUBMIT**

Figure 11: Passenger's Account

This page will display the information about the passenger who has logged in, and they can edit their information and save it to the database by clicking on, ‘submit’ button.

Available Buses at Karma Transport

Bus_ID	Driver_ID	Bus_Color	Number_Plate	Bus_Type
KTB0001	D1001	Red	ABC 4567 A	Single Deck
KTB0002	D1002	White	WD 4567 C	Single Deck
KTB0003	D1003	Red	QAA 4567 C	Double Deck
KTB0004	D1004	White	SAB 4567 C	Single Deck

Figure 12: Bus Details

Passengers can also view the details on buses as it may affect their choice of trip or ticket.

## **8. PROJECT PROBLEM AND PITFALLS**

At first, we struggled to decide on a single topic for the project since many of us suggested various interesting project topics and all seemed doable. Later by considering all the advantages and disadvantages, we managed to decide on one topic, which is the online bus booking management system. Since none of us had any idea about the database, we had a hard time coming up with user requirements and business rules. But after some support from lecturers and friends, we managed to come up with very well defined business rules for our chosen company, Karma Transport.

For our project, we used many languages like MySql, php, javascript, css and html. Although we had some knowledge about it, it cost us a lot of time learning and implementing them. Using XAMPP and implementing mysql was not difficult since we learned in our Database classes and during lab work. But learning php and implementing them in our website was very challenging, as php was new to all of us and it took so many days to learn. After having learnt the basics, we faced unlimited issues with the program functionally and structure. So that time, we used ChatGPT to debug and organize our programs in order.

## **9. CONCLUSION/ RECOMMENDATIONS/FUTURE WORK**

### **Conclusion:**

To sum up, Karma Transport's bus booking management system development is a major project designed to solve important issues the company is now facing with its manual booking procedures. By carefully arranging and putting different pieces into place, the project hopes to completely transform the way both passengers and the organization make reservations. A complete database solution is needed given the reported problems with the current system, which include limited record-keeping, dependence on manual activities, restricted passenger access, problems with payments and cancellations, and safety issues. By implementing computerized procedures, improved security measures, and structured data management, the suggested solution aims to do away with these difficulties.

The system provides a comprehensive approach to improving the effectiveness, accessibility, and security of the booking process by focussing on particular modules such as Administration, Passenger, Ticket, Driver, Cancellation and Refund, and Rating and Feedback. Each module contributes significantly to improving the system's overall functioning, from giving administrators easy access to control routes for buses and tickets to providing passengers with attractive booking and feedback interfaces. The project's aims also include improving user experience, optimizing operations, and supporting Karma Transport's

long-term growth. By employing cutting-edge technologies and database management best practices, the system aims to put the business in a competitive and more digitized market.

We have learned an incredible amount of new knowledge from this project. For example, in order to accomplish the primary goal of our project, we created our database system utilizing SQL Database, which we have only learned about in early January. While working on our project, we learn how to design ERDs and how to identify and learn strong and weak relationships between entities throughout different modules. In addition, we are able to carry out ERD normalization, Data dictionary, as well as implementation. Despite all of the difficulties we have faced, due to everyone's support and collaboration, we are still able to overcome them. We are able to collaborate effectively and support one another through challenging times. We can put the abilities we've learned throughout this course to the test with this assignment. As the project draws to a close, it serves as evidence of the revolutionary potential of technology and cooperative teamwork in fostering organizational success.

### **Future Work**

Here are some potential future work that we can incorporate into our systems:

**Mobile App:** Currently, we just have a website for our system, but we can proceed to make a fully functional app for both android and OS systems.

**Payment Services:** Currently, we allow passengers to make use of their own banking app to make transactions and submit the screenshots. But in the future, we can add more options like credit cards, digital wallets or payment service. That way it will be much more convenient and easy.

**Bus Location:** We can add a feature so passengers can see where their bus is in real time, which would be really helpful for the passengers.

**Reward Events:** If passengers ride the bus often we will initiate reward events, where they can earn points and get discounts on the future tickets.

**Social Media:** We will take our system into social media and spread the words and draw as many interested customers as possible.

**Multi-Lingual:** We will make sure the system works in different languages, so no matter what language passengers speak they can easily book their tickets and use the system without any problems.

Future works on the Bus Booking Management System can focus on expanding its functionality to support multiple transport companies and integration with external APIs to provide a comprehensive travel experience. Finally, ongoing updates and maintenance can ensure the system remains up to date with evolving technologies and user requirements.

## 10. REFERENCES

*Bus Ticket Booking System Project - Download project source code and database | Academic Projects.*  
(n.d.).

<https://www.freeprojectz.com/project-source-code-database-download/bus-ticket-booking-system-project>

*Online Bus Booking System Project using PHP/MySQL.* (n.d.). SourceCodester.  
<https://www.sourcecodester.com/php/14438/online-bus-booking-system-project-using-phpmysql.html>

*Bus Reservation System* | *Bus Ticketing System* | *PHPJabbers.* (n.d.).

<https://www.phpjacobs.com/bus-reservation-system>  
*Ticket Booking System Project - Download project source code and database | Academic Projects.*  
(n.d.).

<https://www.freeprojectz.com/project-source-code-database-download/bus-ticket-booking-system-project>

*Project report on Bus Ticket Booking System | Academic Projects.* (n.d.).  
<https://www.freeprojectz.com/project-report/12114>

Torkut, D. (2023, August 22). How to build an online bus booking system: key features and benefits.  
*Ascendix* *Tech.*  
<https://ascendixtech.com/online-bus-ticket-booking-reservation-system-development/>

*Project report on Bus Ticket Booking System | Academic Projects.* (n.d.-b).  
<https://www.freeprojectz.com/project-report/3047>