CHAPTER 1

INTRODUCTION

The Bus Ticket Management System is a web-based platform designed to simplify the process of booking and managing bus tickets. It enables users to search for available buses, select travel routes, and book tickets effortlessly. The system eliminates the need for traditional paper-based ticketing and provides a digital solution that enhances user convenience and efficiency. By integrating a structured database and an intuitive user interface, it ensures a seamless experience for both passengers and administrators.

This system consists of two main modules: the User Module and the Admin Module. In the User Module, passengers can register, log in, view available buses, book tickets, make payments (mock transactions), and receive confirmation notifications. Users can also check their booking history, cancel tickets if needed, and provide feedback on their travel experience. The Admin Module allows operators to manage bus details, update routes and schedules, monitor bookings, and oversee user transactions, ensuring smooth and organized operations.

Built using Java, Hibernate and MySQL, the system ensures data security and efficient handling of reservations. Hibernate ORM simplifies database interactions, while JDBC facilitates smooth connectivity with MySQL. The use of relational databases ensures consistency, preventing overbooking and maintaining an accurate record of transactions. Additionally, an authentication mechanism protects user information and prevents unauthorized access.

The Bus Ticket Management System enhances the efficiency of the public transportation industry by automating booking processes and reducing human errors. It eliminates long queues at ticket counters, making travel planning more convenient for passengers. Moreover, the feedback system allows users to rate their travel experiences, helping service providers improve their operations. This system is a step toward digital transformation, providing a smart and reliable solution for modern bus ticketing.

CHAPTER 2

OBJECTIVES

1. **Simplify Online Ticket Booking** – The primary objective of this system is to provide an easy-to-use platform for passengers to book bus tickets online. It allows users to search for available buses, choose routes, select seats, and make payments efficiently, reducing dependency on physical ticket counters.

2. **Automate Bus and Ticket Management** – The system automates the management of bus schedules, ticket availability, and reservations. By integrating an organized database, it prevents overbooking, ensures real-time seat availability updates, and provides instant ticket confirmation.

3. **Enhance User and Admin Experience** – The system is designed to offer a smooth and intuitive interface for both passengers and administrators. Users can easily view their booking history, cancel tickets, and provide feedback, while admins can manage bus details, monitor bookings, and oversee customer interactions.

4. **Ensure Secure Transactions and Data Management** – With the integration of authentication mechanisms and database security measures, the system ensures that user data, payment details, and booking records remain protected. Role-based access control allows only authorized users to manage critical system functions.

5. **Improve Efficiency and Reduce Errors** – By digitizing the ticketing process, the system minimizes human errors, eliminates paperwork, and speeds up the booking process. This ensures better efficiency in bus service management, leading to improved customer satisfaction.

6. **Facilitate Feedback and Service Improvement** – The system includes a feedback module where passengers can rate their travel experience. This helps bus operators analyze user feedback and make necessary improvements to enhance service quality.

7. **Support Scalability and Future Enhancements** – The system is designed to be scalable, allowing for the addition of new features like dynamic pricing, seat selection, and integration with third-party payment gateways. It ensures long-term usability and adaptability to future needs.

CHAPTER 3

DATABASE DESIGN

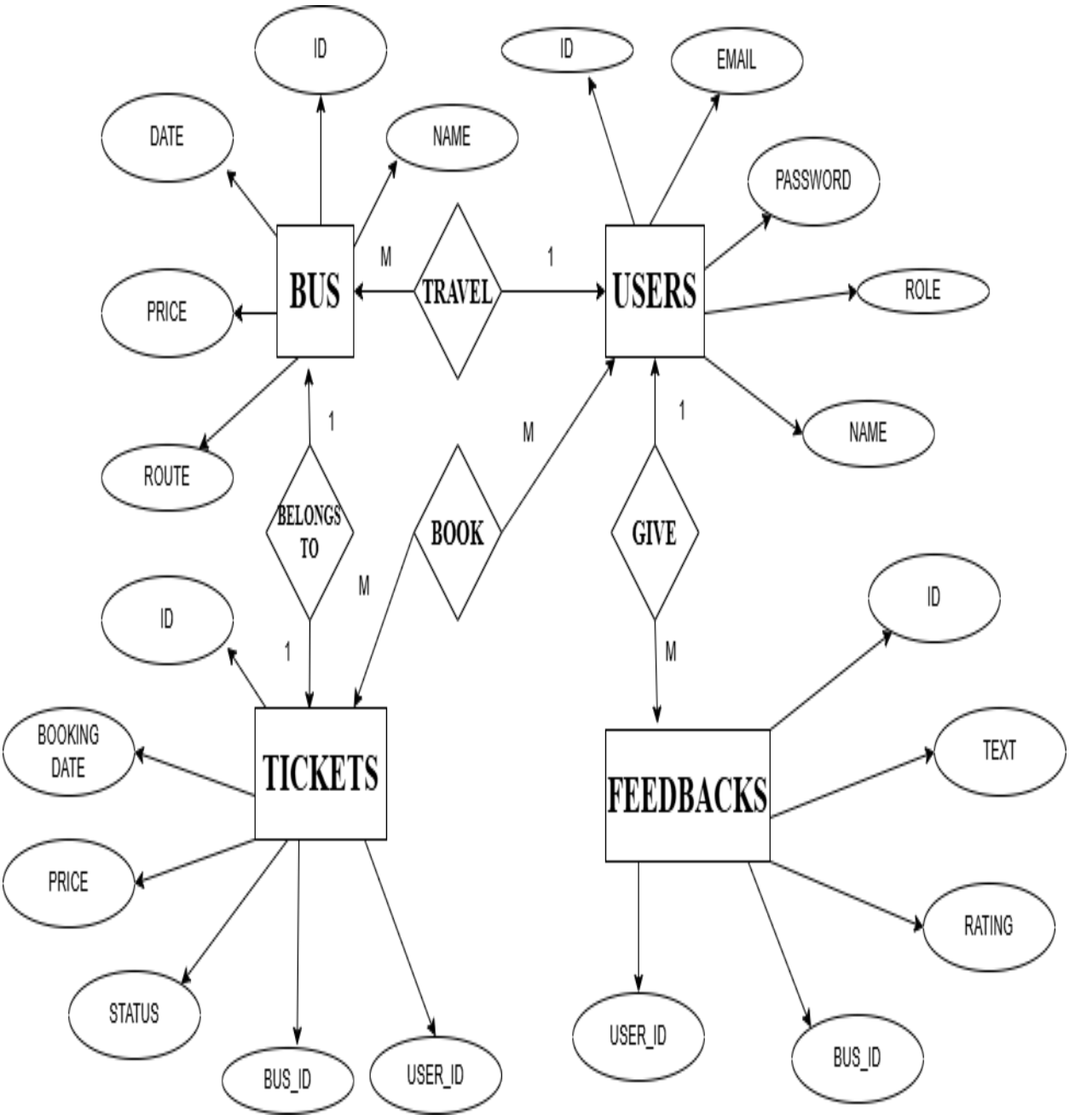
```
MariaDB [bus_ticket]> desc bus;
+---------+--------------+------+-----+---------+----------------+
| Field   | Type         | Null | Key | Default | Extra          |
+---------+--------------+------+-----+---------+----------------+
| busId   | int(11)      | NO   | PRI | NULL    | auto_increment |
| busName | varchar(255) | NO   |     | NULL    |                |
| date    | varchar(255) | NO   |     | NULL    |                |
| price   | double       | NO   |     | NULL    |                |
| route   | varchar(255) | NO   |     | NULL    |                |
+---------+--------------+------+-----+---------+----------------+
5 rows in set (0.034 sec)

MariaDB [bus_ticket]> desc users;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| userId   | int(11)      | NO   | PRI | NULL    | auto_increment |
| email    | varchar(255) | YES  |     | NULL    |                |
| password | varchar(255) | YES  |     | NULL    |                |
| role     | varchar(255) | YES  |     | NULL    |                |
| username | varchar(255) | YES  |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
5 rows in set (0.032 sec)

MariaDB [bus_ticket]> desc feedbacks;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| feedbackId   | int(11)      | NO   | PRI | NULL    | auto_increment |
| feedbackText | varchar(255) | YES  |     | NULL    |                |
| rating       | int(11)      | NO   |     | NULL    |                |
| bus_busId    | int(11)      | YES  | MUL | NULL    |                |
| user_userId  | int(11)      | YES  | MUL | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
5 rows in set (0.031 sec)

MariaDB [bus_ticket]> desc tickets;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| ticketId    | int(11)      | NO   | PRI | NULL    | auto_increment |
| bookingDate | datetime(6)  | YES  |     | NULL    |                |
| price       | double       | NO   |     | NULL    |                |
| status      | varchar(255) | YES  |     | NULL    |                |
| bus_busId   | int(11)      | YES  | MUL | NULL    |                |
| user_userId | int(11)      | YES  | MUL | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
6 rows in set (0.044 sec)
```

E R DIAGRAM


ID
DATE
NAME
PRICE
BUS
M
TRAVEL
1
USERS
ID
EMAIL
PASSWORD
ROLE
NAME
ROUTE
1
BELONGS TO
M
1
M
BOOK
1
GIVE
M
ID
BOOKING DATE
TICKETS
PRICE
STATUS
BUS_ID
USER_ID
ID
TEXT
FEEDBACKS
RATING
USER_ID
BUS_ID

CHAPTER 4

IMPLEMENTATION

ENTITY CLASS

1. User Entity (User Management)

- Stores user details (username, password, email, role).
- Defines user roles (passenger/admin).
- Linked to tickets (users book multiple tickets).
- Linked to feedback (users give reviews).

2. Bus Entity (Bus Details)

- Stores bus name, route, price, and travel date.
- Linked to tickets (multiple bookings per bus).
- Linked to feedback (multiple users can review).

3. Ticket Entity (Booking Management)

- Tracks ticket details (price, booking date, status).
- Linked to users (each ticket belongs to a user).
- Linked to buses (each ticket is for a bus).

4. Feedback Entity (User Reviews)

- Stores feedback text, rating, and date.
- Linked to users (each user can submit reviews).
- Linked to buses (each bus can have multiple reviews)

DAO CLASS

1. **UserDAO (User Management)**

- Functions: Add users, fetch users by ID, validate login, list users, delete users.

- How It Works: Handles user sign-up, login verification, and admin user management.

2. **BusDAO (Bus Data Handling)**

- Functions: Add, fetch, list, update, and delete buses.

- How It Works: Admin adds/modifies buses; users view available options.

3. **TicketDAO (Booking Management)**

- Functions: Book, fetch, list, cancel tickets, and check bus occupancy.

- How It Works: Manages ticket bookings, cancellations, and tracking.

4. **FeedbackDAO (User Reviews)**

- Functions: Add, fetch, list, and delete feedback.

- How It Works: Stores and retrieves user reviews, with admin moderation.

MAIN CLASS

The Main class contains the main() method that initializes and runs the program, calling DAO methods to perform operations.

How It Works

This class is the starting point of the application.

Calls the UserDAO to save a user into the database.

Calls getAllUsers() to retrieve and print all users.

HibernateUtil Class

The HibernateUtil class is a helper class that creates and manages the Hibernate SessionFactory

How It Works

Loads Hibernate configurations from hibernate.cfg.xml.

Creates a SessionFactory, which is used to create Session instances for database interactions.

Provides a getSessionFactory() method to get a session for executing queries.

Hibernate.cfg.xml File

The hibernate.cfg.xml file is the Hibernate configuration file that defines database connection settings and Hibernate properties. It includes:

1. Database Connection – Specifies the database URL, username, password, and driver class.

2. Hibernate Properties – Defines dialect, SQL logging, and transaction handling.

3. Mapping Configuration – Lists entity classes mapped to database tables.

This file is essential for Hibernate to connect with the database and manage ORM (Object-Relational Mapping) efficiently.

pom.xml file

The pom.xml file contains dependencies for Hibernate, MySQL, and other required libraries.

Adds Hibernate, MySQL driver, and Jakarta Persistence dependencies.

Ensures required libraries are downloaded and linked automatically.

CONCLUSION

The Bus Ticket Reservation System is a robust and efficient web-based platform designed to streamline the process of booking bus tickets, managing schedules, and handling user interactions. By integrating Hibernate, MySQL, the system ensures smooth database operations, while its structured entity-relationship model enables seamless user authentication, ticket reservations, and feedback management. Admins can efficiently manage buses, users, and bookings, ensuring system reliability and service quality. Overall, this project enhances the convenience of bus travel by providing an automated, user-friendly, and well-organized booking experience.