Railway System Database

Masen Beacham

Student ID: 918724721

GitHub username: masenbeacham

Checkpoint #	**Date Submitted**
Checkpoint I	02/21/2023
Checkpoint II	02/28/2023
Checkpoint III	03/14/2023
Checkpoint IV	4/4/2023
Checkpoint V	4/4/2023
Checkpoint VI	
Checkpoint VII	

TABLE OF CONTENTS

PROJECT DESCRIPTION

- Railway Management System

We are motivated to create a Railway Systems Database to solve the problems related to managing and organizing information related to railway transportation. Some problems that occur in the current system are, outdated schedules, inefficient use of train routes, long wait times and poor communication between the staff and passengers.

This database aims to provide several features that will make it more efficient for the people taking the railway system. Algorithms will be used by our database system to improve train timetables and routes while also thinking about travel time, distance, and passenger demand. Passenger wait times will be cut down, and railroad operations will run more smoothly. Customers will be able to purchase tickets online, eliminating the need for long waits at railway stations. This will increase the precision and effectiveness of ticketing operations. The communication technologies in our database system will enable staff to interact with passengers more efficiently by giving them real time updates on train times, delays, and other important information. This will result in less frustration from both passengers and staff.

USE CASES

1. **Use Case:** Overbooking- Staff Shortage in Railway Management System

   Actor: Railway Ticketing Staff (Caleb), Passengers, Train Conductors

   Description: Caleb is a railway ticketing staff member who handles the reservation and booking of train tickets. Due to sometimes being short on staff or other circumstances, there are times when train conductors are not available during their scheduled time, which results in overbooking of the train. This leads to overcrowding and inconvenience for passengers. To improve this situation Caleb will need a system that stores the availability status of train conductors and their schedules.

2. **Use Case:** Train Delay Notification in Railway Management System

   Actor: Train Passengers, Train Conductors, Railway Management System

   Description: Train passengers often face delays due to many reasons such as weather, technical issues, or other circumstances. To provide better customer service and improve passenger satisfaction, the Railway Management System needs a feature that notifies passengers of train delays in a timely and accurate manner.

3. **Use Case:** Train Schedule Management in Railway Management System

   Actor: Train Schedulers, Train Conductors, Railway Management System

   Description: The Railway Management System needs a feature that enables schedulers to manage train schedules effectively. The train schedules needs to

run correctly in order to ensure that train services are on time and that there are no problems in train schedules. This feature will help in the overall management of train schedules.

4. **Use Case:** Fares and Ticketing in Railway Management System

   Actor: Passengers, Ticketing Agents, Railway Management System

   Description: Passengers need to purchase train tickets for wherever they are going, and the Railway Management System needs a feature that enables fare calculation and ticketing. This feature will help in the efficient calculation of fares and enable passengers to purchase tickets easily.

5. **Use Case:** Train Maintenance Scheduling in Railway Management System

   Actor: Train maintenance Crew, Railway Management System

   Description: Trains need to be maintained on a regular basis to make sure they are safe and efficient for operation. The Railway Management System needs a feature that enables train maintenance crews to schedule and manage train maintenance effectively.

FUNCTIONAL DATABASE REQUIREMENTS

1. **General User**

    1.1. A general user shall be able to create an account.
    1.2. A general user shall be able to view ticket pricing.
    1.3. A general user shall be able to see travel reviews.

2. **Registered User**

    2.1. A registered user shall be able to log into the system from multiple devices.
    2.2. A registered user shall be able to view their personal information, including their name, contact details, and travel history.
    2.3. A registered user shall be able to reset their password.
    2.4. A registered user shall be able to rate their travel experience.

3. **Account**

    3.1. An account shall be created by only one user.
    3.2. An account shall have an email address associated with it.
    3.3. An account shall have a unique account number.
    3.4. An account shall have a user role assigned to it.

4. **Role**

    4.1. A role shall be linked to many users.
    4.2. A role shall be assigned to a user upon account creation.
    4.3. A role shall determine a user's permissions within the system.

5. **Ticket**

    5.1. A ticket shall be associated with a user account.
    5.2. A ticket shall have a special ticket number.

5.3. A ticket shall be associated with a specific train schedule.

5.4. A ticket shall have a seat assigned to it.

5.5. A ticket shall be valid for a limited time period.

5.6. A ticket shall have a QR code for easy scanning.

5.7. Multiple tickets may be associated with one passenger.

5.8. A ticket shall be able to be canceled and refunded.

6. Train

6.1. A train shall have a train number.

6.2. A train shall have a schedule and route.

6.3. A train shall have a set capacity.

6.4. A train shall be associated with a specific station.

6.5. A train shall be able to have multiple classes of seats.

6.6 A train shall have a current speed.

6.7. A train shall have one or more carriages.

6.8. One train may have multiple carriages.

7. **Train Station**

7.1. A train station shall have a name.

7.2. A train station shall have a location and address.

7.3. A train station may be an intermediate stop on multiple routes.

7.4. A train station shall be able to have multiple trains arriving and departing at the same time.

8. Train Schedule

8.1. A train schedule shall be associated with each train.

8.2. A train schedule shall have a departure time.

8.3. A train schedule shall have an arrival time.

8.4. A train schedule shall have one origin station.

8.5. A train schedule shall have one destination station.

8.6. A train schedule shall be able to be updated or canceled.

9. Seat

9.1. A seat shall have a seat number.

9.2. A seat shall be linked with a specific train.

9.3. A seat shall have a certain class assigned to it.

9.4. A seat shall have a specific location on the train.

10. **Route**

10.1. A route shall have at least two stations.

10.2. A route shall have a distance.

10.3. A route shall have a duration.

10.4. One route may have multiple trains.

11. Payment

11.1. A payment shall be associated with a ticket.

11.2. A payment shall have a payment method.

11.3. A payment shall have a certain amount.

12. Carriage

12.1. A carriage shall have a maximum capacity limit.

12.2. A carriage shall have maintenance records, including inspection dates and repair history.

12.3. Multiple carriages may be assigned to one train.

13. Maintenance

13.1. A maintenance shall be associated with a train.

13.2. A maintenance shall have a start date.

13.3. A maintenance shall have an end date.

13.4. A maintenance shall have a description of the problem.

14. Employee

14.1. An employee shall have a name.

14.2. An employee shall have a phone number.

14.3. An employee shall have a job title.

14.4. An employee shall have a salary.

NON FUNCTIONAL DATABASE REQUIREMENTS

1. **Scalability**

   1.1. The database system shall support horizontal scaling to handle increasing data loads.

   1.2 The database system should have the ability to add additional nodes to the cluster without any downtime.

2. **Accessibility**

   2.1. The database system shall be accessible through a secure network connection.

   2.2 The database system should support access control to limit user access based on their role and permissions.

3. **Maintainability**

   3.1 The database system should provide tools for monitoring database health and performance.

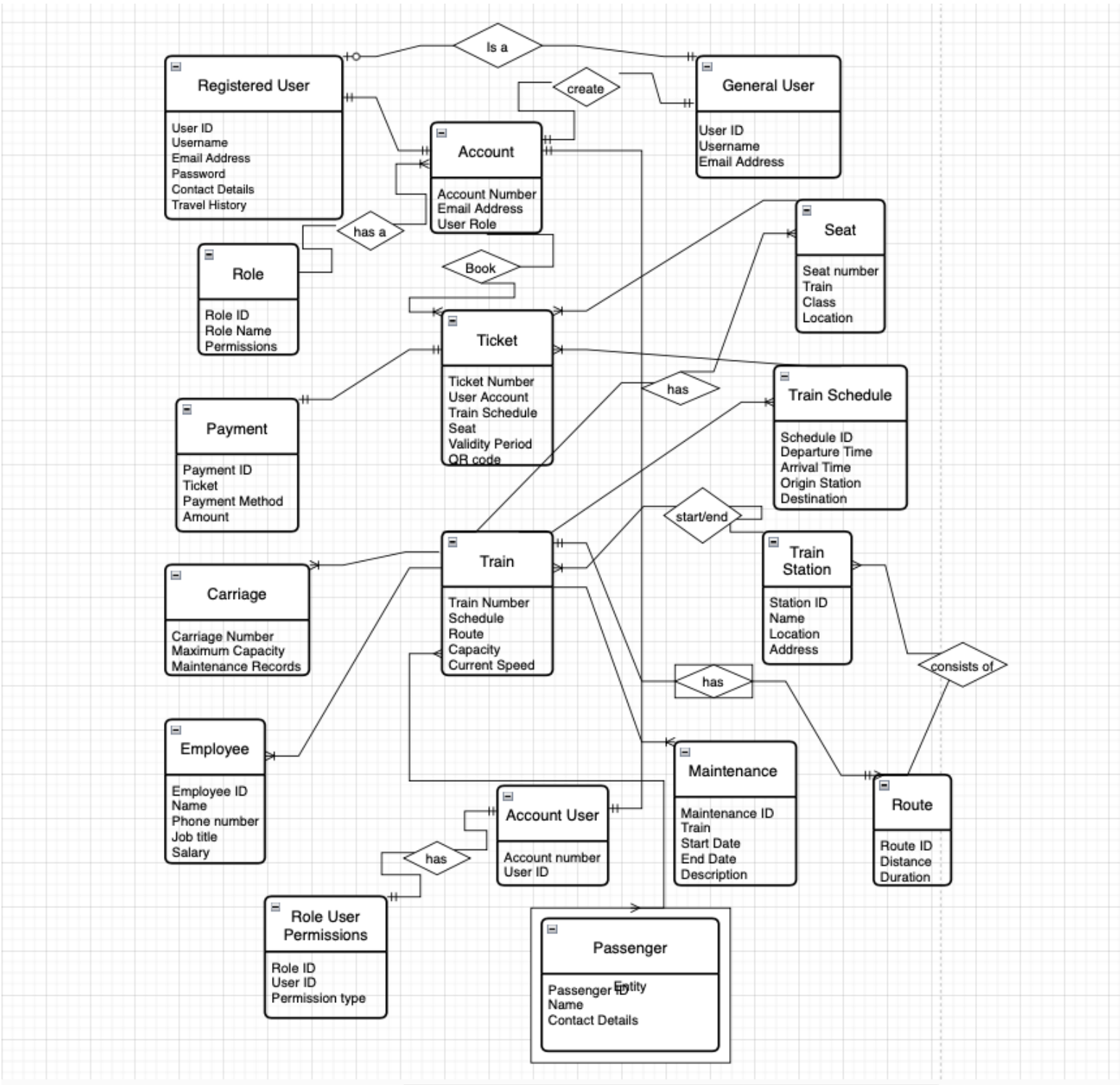
   3.2 The database system should support automated backups and database schema migrations.

4. **Performance**

   4.1. The database system shall support multiple transactions.

5. **Security**
6. **Storage**

ERD DIAGRAM


Is a
Registered User
User ID
Username
Email Address
Password
Contact Details
Travel History
create
General User
User ID
Username
Email Address
Account
Account Number
Email Address
User Role
has a
Role
Role ID
Role Name
Permissions
Book
Seat
Seat number
Train
Class
Location
Ticket
Ticket Number
User Account
Train Schedule
Seat
Validity Period
QR code
has
Train Schedule
Schedule ID
Departure Time
Arrival Time
Origin Station
Destination
Payment
Payment ID
Ticket
Payment Method
Amount
start/end
Train
Train Number
Schedule
Route
Capacity
Current Speed
Train Station
Station ID
Name
Location
Address
consists of
Carriage
Carriage Number
Maximum Capacity
Maintenance Records
has
Employee
Employee ID
Name
Phone number
Job title
Salary
Maintenance
Maintenance ID
Train
Start Date
End Date
Description
Route
Route ID
Distance
Duration
Account User
Account number
User ID
has
Role User Permissions
Role ID
User ID
Permission type
Passenger
Passenger ID
Name
Contact Details
Entity

ENTITY & ATTRIBUTES DESCRIPTION

1. General User (Strong)
    - general_user_id: key, numeric
    - username: alphanumeric
2. Registered User (Strong)
    - registered_user_id: key, numeric
    - username: alphanumeric
    - email_address: alphanumeric
    - password: alphanumeric
    - contact_details: alphanumeric
    - travel_history: alphanumeric
3. Account (Weak)
    - account_number: key, alphanumeric
    - user_id: key, numeric
    - user_role: alphanumeric
4. Role (Strong)
    - role_id: key, numeric
    - role_name: alphanumeric
    - permissions: alphanumeric
5. Train (Strong)
    - train_number: key, alphanumeric
    - schedule: alphanumeric
    - route: alphanumeric
    - capacity: numeric
    - current_speed: numeric
6. Train Schedule (Strong)
    - schedule_id: key, numeric
    - departure_time: timestamp
    - arrival_time: timestamp

- origin_station: alphanumeric
- destination: alphanumeric

7. Train Station (Strong)
    - station_id: key, numeric
    - name: alphanumeric
    - location: alphanumeric
    - address: alphanumeric
8. Carriage (Strong)
    - carriage_number: key, alphanumeric
    - maximum_capacity: numeric
    - maintenance_records: alphanumeric
9. Employee (Strong)
    - employee_id: key, numeric
    - name: alphanumeric
    - phone_number: alphanumeric
    - job_title: alphanumeric
    - salary: numeric
10. Maintenance (Weak)
    - maintenance_id: key, numeric
    - train_number: key, alphanumeric
    - start_date: timestamp
    - end_date: timestamp
    - description: alphanumeric
11. Route (Strong)
    - route_id: key, numeric
    - distance: numeric
    - duration: alphanumeric
12. Seat (Strong)
    - seat_number: alphanumeric
    - train_number: alphanumeric
    - class: alphanumeric

- location: alphanumeric

13. Passenger (Strong)
    - passenger_id: key, numeric
    - name: alphanumeric
    - contact_details: alphanumeric
14. Ticket (Strong)
    - ticket_id: key, alphanumeric
    - name: numeric
    - schedule_id: numeric
    - seat_number: alphanumeric
    - validity_period: timestamp
15. Payment (Strong)
    - payment_id: key, numeric
    - ticket_number: alphanumeric
    - payment_method: alphanumeric
    - amount: numeric

CONSTRAINTS DESCRIPTION

Table	FK	ON DELETE	ON UPDATE	Comment
Account	General user	On CASCADE	ON CASCADE	If a general user is deleted, then the account from that user must be deleted as well.
Account	Registered User	ON CASCADE	ON CASCADE	If a registered user is deleted, then the account from that user must be deleted as well.
Account	role	SET NULL	ON CASCADE	If a role is removed from a specific user, the user that was holding that role will hold no role until a new one is assigned.
Ticket	seat	SET NULL	ON CASCADE	If a seat is deleted, all tickets associated with that seat will have their seat field set to NULL.
Ticket	train	SET NULL	NO ACTION	If a train is deleted, all tickets associated with that train will have their train field set to NULL. The train field cannot be updated.
Ticket	payment	SET NULL	ON CASCADE	If a payment is deleted, all tickets associated with that payment will have their payment field set to NULL
Ticket	account	SET NULL	ON CASCADE	If an account is deleted, all tickets associated with that account will have their account field set to NULL
Seat	number	ON CASCADE	ON CASCADE	If a seat is deleted, then all tickets associated with that seat will have their seat_number field set to NULL
Train Schedule	train	ON CASCADE	ON CASCADE	If a train is deleted, then all of its schedules must be deleted as well.
Train	passenger	SET NULL	ON CASCADE	If a train is deleted, all passengers on that train will have their train field set to NULL
Train	train station	SET NULL	NO ACTION	If the current station of a train is deleted,

				then the train's current_station field will be set to NULL. Updating the station is not allowed
Route	train	ON CASCADE	ON CASCADE	If a route is deleted, then all stations that belong to that route must be deleted as well.
Employee	train	SET NULL	ON CASCADE	If a train is deleted, the employee's train field will be set to NULL until a new train is assigned.
Maintenance	train	ON CASCADE	ON CASCADE	If a train is deleted, then all of its maintenance records must be deleted as well
Carriage	train	ON CASCADE	ON CASCADE	If a train is deleted, then all of its carriages must be deleted as well