

Team Name: IBM

Members: Joe Kanagawa

Harkamal Grewal

Yimin Mei

Project Title: Bullet Train Reservation system

Database Schema:

```
DROP DATABASE IF EXISTS BTRS;  
CREATE DATABASE BTRS;  
USE BTRS;
```

```
DROP TABLE IF EXISTS Car  
CREATE TABLE Car (  
    carNumber INT,  
    seatID VARCHAR(3),  
    trainID INT,  
    passengerID INT DEFAULT NULL,  
    PRIMARY KEY(carNumber, seatID, trainID),  
    FOREIGN KEY(trainID) REFERENCES Train(trainID) ON UPDATE CASCADE ON DELETE  
CASCADE,  
    FOREIGN KEY(passengerID) REFERENCES Passenger(passengerID) ON UPDATE CASCADE  
ON DELETE CASCADE  
);
```

```
DROP TABLE IF EXISTS Station  
CREATE TABLE Station (  
    stationID INT PRIMARY KEY,  
    name VARCHAR(64),  
    order INT  
);
```

```
DROP TABLE IF EXISTS Train  
CREATE TABLE Train (  
    trainID INT PRIMARY KEY,  
    deptTime TIME,  
    isFull BOOL  
);
```

```
DROP TABLE IF EXISTS Passenger;  
CREATE TABLE Passenger(  
    ID INT,  
    accountID INT ,  
    startStID INT,  
    endStID INT,
```

```

seatID INT,
dateTime date,
wifi boolean
);

DROP TABLE IF EXISTS Banned;
CREATE TABLE Banned (
    accountID INT,
    PRIMARY KEY(accountID) REFERENCES Account_Holder(accountID) ON UPDATE
CASCADE ON DELETE CASCADE
);

DROP TABLE IF EXISTS Account_Holder;
CREATE TABLE Account_Holder (
    accountID INT,
    fullName VARCHAR(50),
    email VARCHAR(30),
    password VARCHAR(30),
    creditCard INT,
    PRIMARY KEY(accountID)
);

```

### Relations:

#### *Car*

| Car 🚍                 |  |                 |                   |
|-----------------------|--|-----------------|-------------------|
| 🔑 seatID (varchar(3)) | 🔑 carNumber (int,<br>$0 \leq x \leq 3$ ) | 🔑 trainID (int) | passengerID (int) |
|                       |  |                 |                   |

#### Description of each attr:

*seatID* - A string merely a concatenation of an integer ( $1 \leq x \leq 14$ ) denoting row, and an upper case letter  $l \in \{A, B, C, D\}$  denoting the seat. Example: "11A" - 'A' seat of Row #11; kind of like the seat map on this page: [https://www.jrhokkaido.co.jp/train/tr008\\_03.html](https://www.jrhokkaido.co.jp/train/tr008_03.html).

*carNumber* - Specifies which of the 4 cars.

*trainID* - Specifies which train.

*passengerID* - Passenger who reserved the seat, if any. If none, NULL.

#### *Station*

| Station 🚆         |                    |             |
|-------------------|--------------------|-------------|
| 🔑 stationID (int) | name (varchar(64)) | order (int) |
|                   |                    |             |

### Description of each attr:

*stationID* - Identifies the station.

*name* - The name of the station. Example: "San Jose Diridon"

*order* - An integer uniquely given to each station that denotes the order of stops. Important in removing/adding stations (I understand it sounds confusing right now but I have an idea for this which I intend to explain in our next meeting).

### *Train*

| Train                 |                        |                      |
|--|------------------------|----------------------|
|  <i>trainID</i> (int) | <i>deptTime</i> (time) | <i>isFull</i> (bool) |

### Description of each attr:

*trainID* - Identifies the train.

*deptDateTime* - Indicates the time of departure.

*isFull* - If true, all seats in that train are reserved. Default = false.

### *Passenger*

| Passenger                 |                        |                        |                      |                     |                             |                       |
|--|------------------------|------------------------|----------------------|---------------------|-----------------------------|-----------------------|
|  <i>passengerID</i> (int) | <i>accountID</i> (int) | <i>startStID</i> (int) | <i>endStID</i> (int) | <i>seatID</i> (int) | <i>dateTime</i> (date/time) | <i>wifi</i> (boolean) |

### Description of each attr:

*passengerID*: unique identifier given to each passenger regardless of having an account. Must **not** be NULL.

*accountID*: unique identifier for passenger

*startStID*: the starting station ID

*endStID*: the ending station ID

*dateTime*: the date and time of trip

*wifi*: wifi for the passenger?

### *Account\_Holder*

| Account_Holder         |                          |                       |                        |                         |
|------------------------|--------------------------|-----------------------|------------------------|-------------------------|
| <i>accountID</i> (int) | <i>fullName</i> (String) | <i>email</i> (String) | <i>password</i> (hash) | <i>CreditCard</i> (int) |

### Description of each attr:

*accountID*: unique identifier for the account holder

fullName: the name of account holder  
email: the email of account holder  
password: the password to login  
CreditCard: the holder's credit card

*Ban\_list*

|                |
|----------------|
| Banned         |
| accountID(int) |

Description of each attr:

accountID: the unique identifier for account banned

FUNCTIONAL REQUIREMENTS:

1. Sign into customer account: Validating the user and confirming credentials
2. Sign up for a customer account: User can register for account; Information will be stored.
3. Reserve train destination: Sign up for a trip.
4. Change/update train destination: Update a trip destination.
5. Change train time: Update a trip time.
6. Change date information: Update a trip date.
7. Change seat: Update a trip seat.
8. Delete reservation: Delete a trip.
9. Update customer information: Update customer information.
10. Reset customer password: Reset password
11. Ban a customer: Add a customer to the ban list.
12. Adding a train: Add a new train.
13. Adding station: Add a new station.
14. Delete/remove ban: Admin can remove user account and insert user to a Banned DB
15. Sign up for Wifi: Passenger can register for wifi for that specific train ride