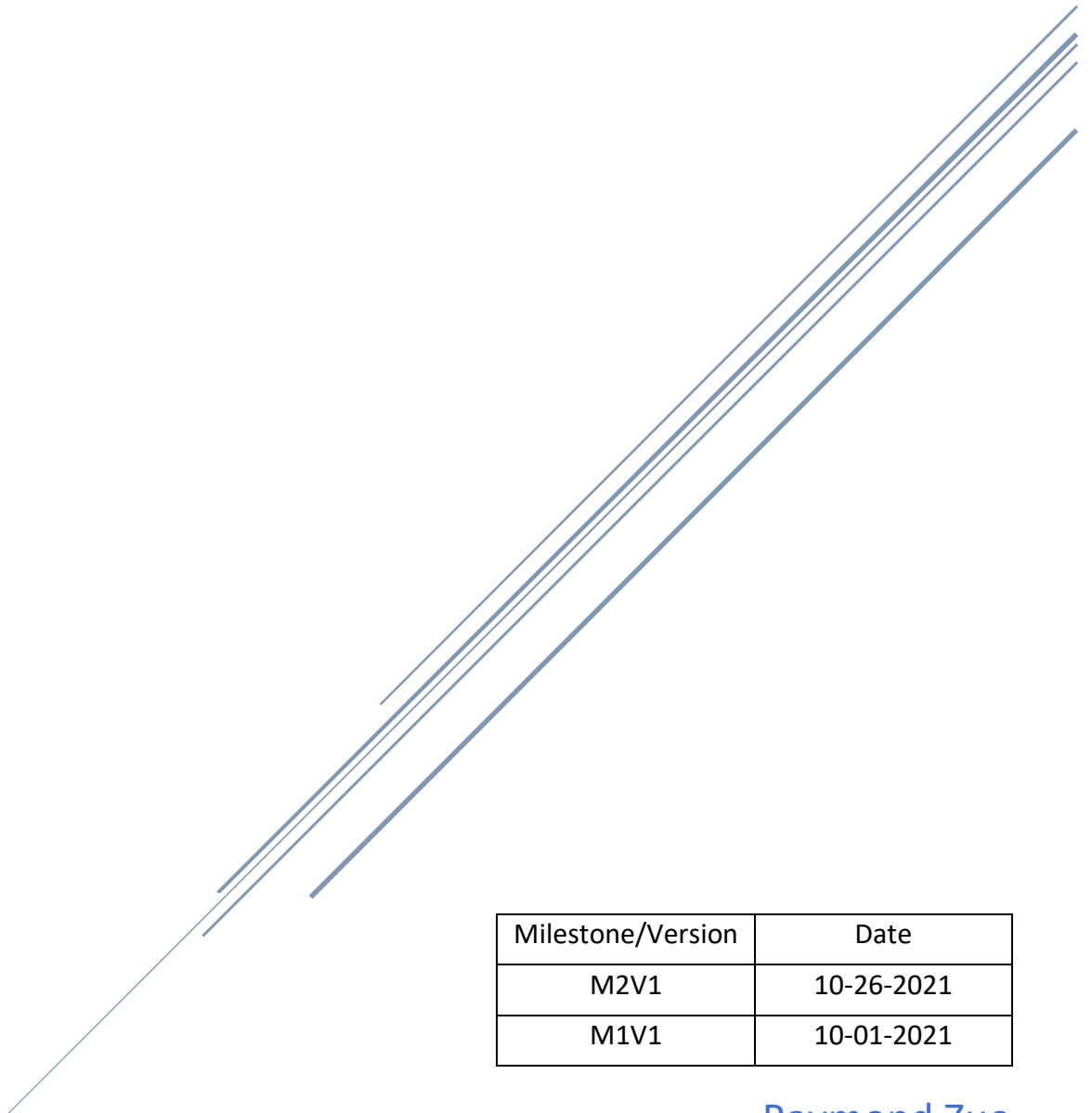


RAILWAY SYSTEM DATABASE

Github: [Zellineous](#)



Milestone/Version	Date
M2V1	10-26-2021
M1V1	10-01-2021

Raymond Zuo
918295032

Table of Contents

Project Description	2
Use Case.....	3
Database Requirements.....	4
Main Entities, Attributes and Keys	7
ERD.....	9
Testing Table	10

Project Description

The database that is being created in this project is a Railway System Database. The name of the app is RailX

This railway system will be an app that allows the passengers to look up train availability, the destination of the train, booking and cancelling of tickets, etc. This system will also allow the passengers to look at the records of the train which includes the name, number, seat availability, and whether the seats are already booked. It will also show what days the train will be available for booking. A Passenger will also be able to look at their own ticket and see what they booked. This system will also be good for the admins of the app. They can use the app to maintain the trains and resolve any issues people may have when booking their trips.

Use Cases

Case 1: Dave is a passenger who wants to get a ticket from RailX. He is always late to work because he buys the ticket from the station right before the trains depart. He decides to use the railway app to book tickets ahead of time. He first registers with the into the system. He then logs in to his account. He then looks at all the train times and books his ticket. While booking ahead of time he realized that he is only able to book one ticket at a time. The system does not handle multiple tickets at a time.

Case 2: Tom is a person who never rode in the railway system before. He decides that he needs to use the railway to get to work. Since Tom is a very organized person, he wants to know everything ahead of time. He uses the Rail app to plan his trips. He finds the train number and the destination of the trains. Since he is not buying tickets from the app and is going to buy the tickets from the station clerk, he does not want to register. This is a problem because the system only allows registered users to view the times of the train.

Case 3: The manager of a station Bob wants to create a way for people in the station to see all the times of when the trains arrive. First, he registers with the app. Then he will display the times on the monitor. He sees that he is only able to display the time and the destination of one train at a time. The app does not have a section that shows all trains and times all at once currently.

Case 4: Rick is a police officer who must investigate in an area near the railway. He wants to investigate a case with the information of specific trains. To view this information, he must register with the app. This information is only presented for registered users. The information he was able to get is the trains number, schedule, how many passengers went in and out of the train on specific dates, and which passenger was in the train.

Case 5: Sam is a firefighter who is responding to a call about a situation in the train station. The fire station has access to admin permission of the database. While on the way to the situation the firefighters will be able to access all the information about the person who needs help. When the call is put in and a description of the person who is in need is given, the firefighters can get all the information they need before they even arrive on the scene.

Database Requirements

1) Station

- a) A station shall have zero or more passengers.
- b) A station shall have zero or more employees.
- c) A station shall be at only one region.
- d) A station shall report to zero or more police department
- e) A station shall report to zero or more fire department
- f) A station shall have zero or more train yards
- g) A station shall have zero or more repair yards
- h) A station shall have one manager which is also an employee

2) Manager

- a) A manager is an employee.

3) Passengers

- a) A passenger shall have one and only one account.
- b) A passenger shall be at one or more stations.
- c) A passenger shall buy one or many tickets

4) User

- a) Many users shall have one account.

5) Account

- a) An account shall have one or more user
- b) An account shall have one unique account id
- c) An account shall have one user encrypted password
- d) An account shall have one unique user email
- e) An account shall have only one passenger
- f) An account shall be accessed by a device

6) Basic Account

- a) A basic account shall have one unique account id

7) Subscription Account

- a) A premium account shall have one unique account id
- b) A premium account shall have one expiration date

8) Employee

- a) An employee shall consist of one or more booking clerks
- b) An employee shall consist of zero or more guides
- c) An employee shall have a manager

- d) An employee shall work at least at one station
- e) An employee shall consist of zero or more drivers

9) Booking clerk

- a) A booking clerk is an employee
- b) A booking clerk shall sell many tickets

10) Train

- a) A train shall run zero or more trips.
- b) A train shall board many tickets holders

11) Ticket

- a) A ticket shall be owned by one and only one passenger
- b) A ticket shall allow the boarding of one train
- c) A ticket shall be sold by the booking clerk
- d) A ticket shall have an expiration date

12) Trip

- a) A trip shall be taken by zero or more trains
- b) A trip shall have one or more routes

13) Train yard

- a) A train yard shall reside in zero or more stations

14) Guide

- a) A guide is an employee.

15) Police department

- a) Many police departments shall respond to many stations

16) Fire department

- a) Many fire departments shall respond to many stations

17) Routes

- a) A route shall be in one or more trips

18) Repair Yard

- a) A repair yard shall reside in zero or more stations

19) Expiration date

- a) An expiration date shall handle tickets
- b) An expiration date shall handle premium accounts

20) Region

- a) A region will have only one station

21) Driver

- a) A Driver is an employee

22) Device

- a) A device shall access an account

Main Entities, Attributes and Keys

1) User: Strong

- a) user_id: key, numeric
- b) name: multi-value, alphanumeric
- c) last_name: multi-value, alphanumeric
- d) email: key, alphanumeric

2) Account: Weak

- a) account_id: key, numeric
- b) email: key, alphanumeric
- c) password: alphanumeric

3) Train: Strong

- a) train_id: key, numeric
- b) model: alphanumeric

4) Employee: Weak

- a) employee_id: key, numeric
- b) station_id: numeric

5) Police department: Strong

- a) police_id: key, numeric
- b) p_station_id: key, alphanumeric

6) Fire department: Strong

- a) police_id: key, numeric
- c) f_station_id: key, alphanumeric

7) Station: Strong

- a) station_id: key, numeric
- b) name: alphanumeric

8) Train yard: Strong

- a) t_yard_id: key, numeric
- b) station_id: numeric

9) Repair yard: Strong

- a) r_yard_id: key, numeric
- b) station_id: numeric

10) Trip: Strong

- a) trip_id: key, numeric
- b) name: alphanumeric

11) Routes: Strong

- a) route_id: key, numeric
- b) name: alphanumeric

12) Ticket: Strong

- a) ticket_id: key, numeric
- b) expiration_id: numeric

13) Passenger: Strong

- a) passenger_id: key, numeric
- b) name: alphanumeric

14) Driver: Weak

- a) driver_id: key, numeric
- b) name: alphanumeric

15) Manager: Weak

- a) manager_id: key, numeric
- b) name: alphanumeric

16) Payment Type: Strong

- a) type_id: key, numeric
- b) billing_address: alphanumeric

17) Account Type: Strong

- a) account_typeid: key, numeric
- b) description: alphanumeric

18) Region: Strong

- a) region_id: key, numeric
- b) description: alphanumeric

19) Supplier: Strong

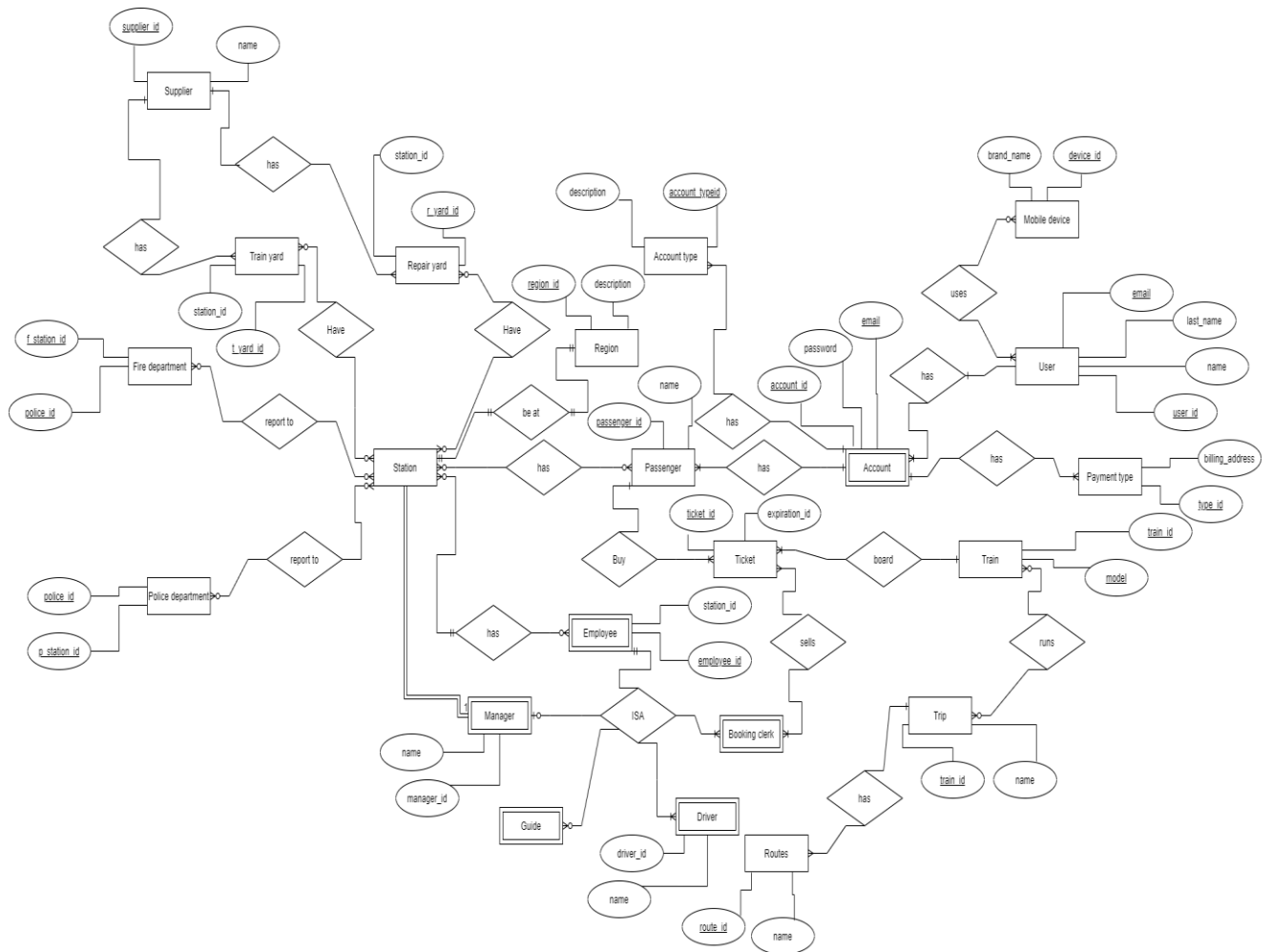
- a) supplier_id: key, numeric
- b) name: alphanumeric

20) Mobile device: Strong

- a) device_id: key, numeric
- b) brand_name: alphanumeric

ERD

Zoom in to see better.



Testing Table for EER

Rule	Entity A	Relation	Entity B	Cardinality	P/F	Description
1	Station	has	Passenger	1-to-M	Fail	Can have zero passengers
2	Passenger	be at	Station	0-to-M	Pass	None
3	Station	has	Employee	1-to-M	Pass	None
4	Employee	works	Station	1-to-1	Fail	Employee can work at another station
5	Station	be at	Region	1-to-M	Fail	Station can only be in one region
6	Region	Has	Station	1-to1	Pass	None
7	Station	report to	Police department	1-to-M	Pass	None
8	Station	report to	Fire department	1-to-M	Pass	None
9	Police department	respond to	Station	M-to-M	Pass	None
10	Fire department	respond to	Station	M-to-M	Pass	None
11	Station	have	Train yard	1-to-M	Pass	None
12	Station	have	Repair yard	1-to-M	Pass	None
13	Repair yard	in	Station	1-to-M	Pass	None
14	Train yard	in	Station	1-to-M	Pass	None
15	Booking clerk	is a	Employee	ISA	Pass	None
16	Driver	is a	Employee	ISA	Pass	None
17	Guide	is a	Employee	ISA	Pass	None
18	Manager	is a	Employee	ISA (should be Recursive)	Fail	This is a Recursive relationship one manager will manage one station
19	Passenger	has	Account	1-to-M	Fail	A passenger should only have one account

20	Account	represent	Passenger	1-to-1	Fail	A passenger could make more than one account
21	User	has	Account	1-to-M	Pass	None
22	Account	represent	User	M-to-1	Pass	None
23	Passenger	buys	Ticket	1-to-M	Pass	None
24	Ticket	owned by	Passenger	1-to-1	Pass	None
25	Ticket	sold by	Booking clerk	1-to-M	Fail	Many tickets can be sold
26	Booking clerk	sells	Ticket	1-to-M	Pass	None
27	Train	has	Ticket	1-to-M	Pass	None
28	Ticket	board	Train	M-to-1	Pass	None
29	Train	runs	Trip	M-to-M	Pass	None
30	Trip	has	Train	1-to-M	Pass	None
31	Trip	has	Routes	1-to-M	Pass	None
32	Routes	is	Trip	1-to-M	Fail	There are many routes to one trip
33	Train yard	has	Supplier	1-to-1	Pass	None
34	Supplier	supplies	Train yard	1-to-M	Pass	None
35	Repair yard	has	Supplier	1-to-1	Pass	None
36	Supplier	supplies	Repair yard	1-to-M	Pass	None
37	Account	has	Account type	1-to-M	Pass	None
38	Account type	consist of	Account	M-to-1	Pass	None
39	Account	has	Payment Type	1-to-M	Pass	None
41	Payment Type	of	Account	M-to-1	Pass	None
42	User	has	Mobile device	1-to-M	Pass	None
43	Mobile device	used by	User	M-to-1	Pass	None

Database Model/EER

	Table	FK	ON DELETE	ON UPDATE	Comment
1	Station	Police Department	CASCADE	SET NULL	If police dept is deleted, then the station will have a open slot for another police station
2	Station	Fire Department	CASCADE	SET NULL	If fire dept is deleted, then the station will have a open slot for another fire station
3	Station	Train Yard	CASCADE	SET NULL	If train yard is deleted, then the station will have a open slot for another train yard
4	Station	Repair Yard	CASCADE	SET NULL	If repair yard is deleted, then the station will have a open slot for another repair yard
5	Station	Region	CASCADE	CASCADE	If region is deleted the station that is in the region must be deleted
6	Station	Passenger	CASCADE	SET NULL	If passenger is deleted, then the station will have a open slot for another passenger
7	Train Yard	Supplier	CASCADE	SET NULL	If supplier is deleted, then the train yard will have a open slot for another supplier
8	Repair Yard	Supplier	CASCADE	SET NULL	If supplier is deleted, then the repair yard will have a open slot for another supplier
9	Passenger	Ticket	CASCADE	SET NULL	If ticket is deleted, then the passenger will have a open slot for another ticket

10	Ticket	Train	CASCADE	CASCADE	If train is deleted the ticket that is tied to the train is deleted
11	Train	Trip	CASCADE	SET NULL	If trip is deleted, then the train will have a open slot for another trip
12	Trip	Routes	CASCADE	CASCADE	If routes is deleted the trip that is tied to the routes is deleted
13	Account	Payment Type	CASCADE	SET NULL	If payment type is deleted, then the account will have a open slot for another payment type
14	Account	Account Type	CASCADE	CASCADE	If account type is deleted the account that is tied to the account type is deleted
15	Account	User	CASCADE	CASCADE	If user is deleted the account that is tied to the user is deleted
16	Account	Passenger	CASCADE	SET NULL	If passenger is deleted, then the account will have a open slot for another passenger
17	User	Mobile device	CASCADE	SET NULL	If mobile device is deleted, then the user will have a open slot for another mobile device
18	Employee	Station	CASCADE	CASCADE	If station is deleted the employee that is tied to the station is deleted
19	Employee	Manager	CASCADE	SET NULL	If manager is deleted, then the employee will have a open slot for another manager
20	Employee	Driver	CASCADE	CASCADE	If driver is deleted the employee that is tied

					to the driver is deleted
21	Station	Manager	CASCADE	SET NULL	If manager is deleted, then the station will have a open slot for another manager

Testing Table X

Entity	SQLQuery	Pass/Fail	Error Description	Possible Solution
Passenger	Delete	Pass	None	None
Passenger	Update	Pass	None	None
AccountType	Delete	Pass	None	None
AccountType	Update	Pass	None	None
Region	Delete	Pass	None	None
Region	Update	Fail	region_id had a dupe	Make sure there are no dupes in update
Station	Delete	Pass	None	None
Station	Update	Fail	Wrong where identification	check spelling
User	Delete	Pass	None	None
User	Update	Pass	None	None
Account	Delete	Fail	Based on previous quarry it will make it null	Change what is deleted
Account	Update	Fail	Key constraint fails	Try to have more things inputted
Train	Delete	Pass	None	None
Train	Update	Pass	None	None
Employee	Delete	Fail	Nothing left to delete	Change what was deleted before.
Employee	Update	Pass	None	None
PoliceDepartment	Delete	Pass	None	None
PoliceDepartment	Update	Pass	None	None
FireDepartment	Delete	Pass	None	None
FireDepartment	Update	Pass	None	None
TrainYard	Delete	Pass	None	None
TrainYard	Update	Pass	None	None
RepairYard	Delete	Pass	None	None
RepairYard	Update	Pass	None	None
Trip	Delete	Pass	None	None
Trip	Update	Fail	Dupes for id is not allowed	Remove dupe
Routes	Delete	Pass	None	None
Routes	Update	Pass	None	None
Ticket	Delete	Pass	None	None
Ticket	Update	Pass	None	None
Driver	Delete	Pass	None	None

Driver	Update	Fail	Had another dupe when updating	Remove dupe
Manager	Delete	Pass	None	None
Manager	Update	Pass	None	None
PaymentType	Delete	Pass	None	None
PaymentType	Update	Pass	None	None
Supplier	Delete	Pass	None	None
Supplier	Update	Pass	None	None
MobileDevice	Delete	Pass	None	None
MobileDevice	Update	Pass	None	None