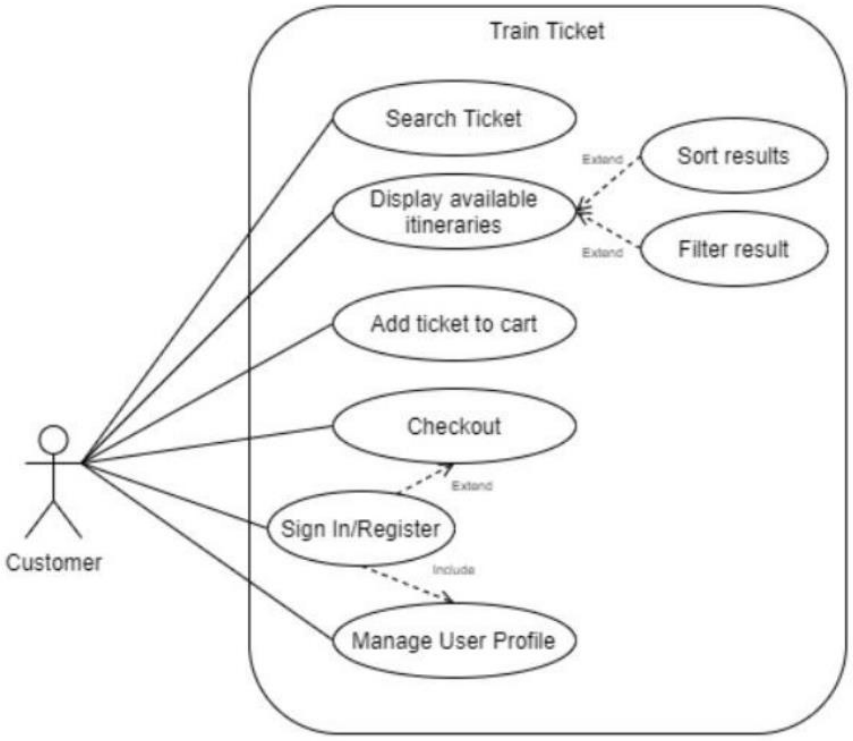
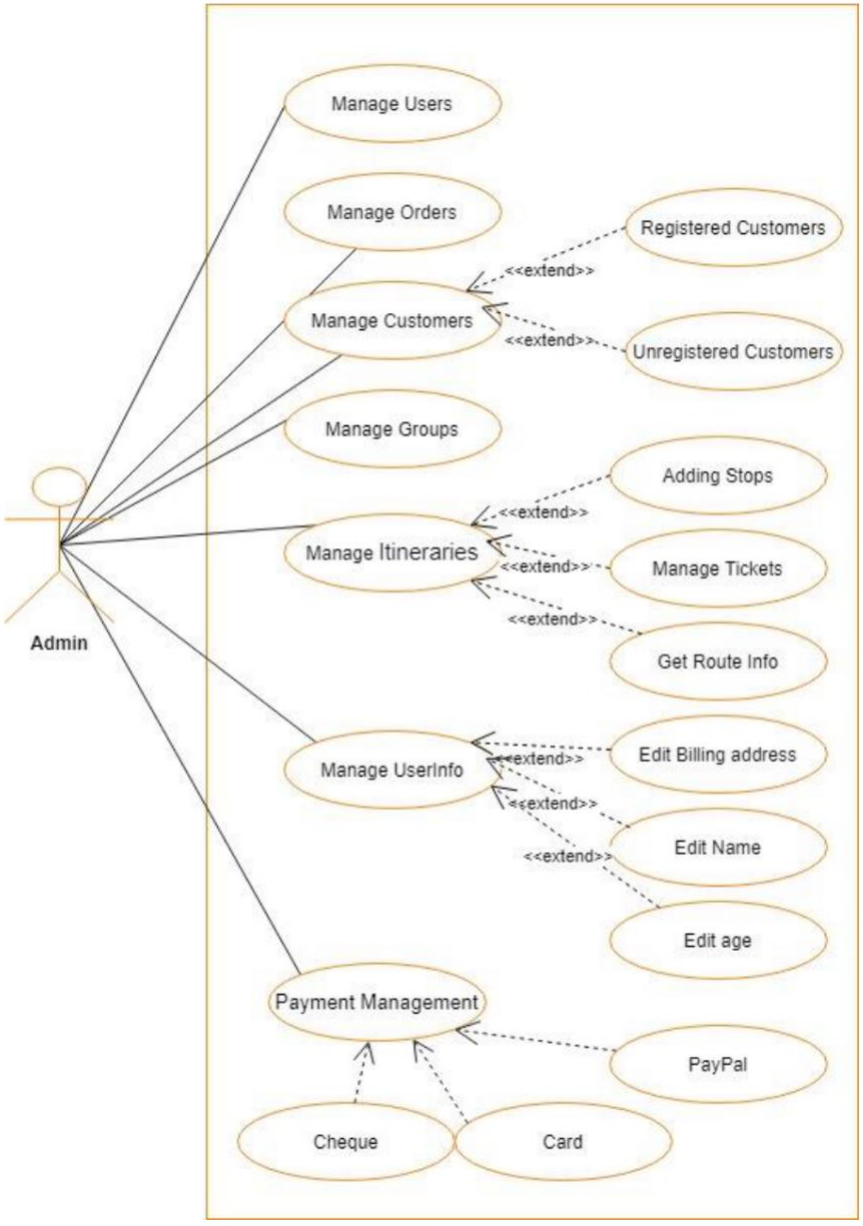
User Stories, Use Case, Component and Deployment Diagram of Train Ticketing System

Use case diagram for Traveler:



Use Case diagram for Admin:



USE CASE: SEARCH TICKET

ACTOR: Customer

FLOW OF EVENTS:

- The use case begins when customer connect to the train ticket homepage. From the search section of the website, customer enters the start location and the destination. Customer selects date and time. Customer selects the quantity of tickets, and finally clicks
  "Search" button.

USER STORIES:

1. Search One-Way trip:
   As a user, when I select a one-way trip along with from, to location and choose the departure date and arrival date and the number of tickets needed, the system should display all the available itineraries for the details I have provided including the date and time.

2. Search Round trip:
   As a user, when I select a round trip along with from, to location and choose the departure date and arrival date and number of tickets needed, the system should display all the available itineraries for the details I have provided including the date and time for both to and from trips.

3. Search Box autocomplete/auto-recognize:
   As a user, when I type in the city name or the station code, the system should be able to suggest the matching city or stations, which will then auto fill the search box upon a click.

USE CASE: DISPLAY AVAILABLE ITINERARIES

ACTOR: Customer

FLOW OF EVENTS:

- The use case begins when customer click search. Customer view the provided list of available itineraries that matches or close to customer's preferences in different price ranges, classes date and time. Customer filters the list by price, and then customer sorts the list from the cheapest to most expensive price. Customer then filters the list by both price and exact selected day and hours. After finding the desirable ticket, customer selects the ticket check box and add to cart.

USER STORIES:

1. One-way trip or round trip:

PRE-CONDITION: The customer has already entered all the necessary information to search for a trip

USER STORIES: As a customer, I am able to browse through the list of available itineraries, depending on whether I chose to search for one way or round trip, the system must be able to display one trip or two trips respectively.

2. Ticket Type:

PRE-CONDITION: The customer has already entered all the necessary information to search for a trip.

USER STORIES: As a customer, I am able to browse through the list of available itineraries, choose the ticket class that I want and view the total price.

3. Sort Ticket:

PRE-CONDITION: The customer has already entered all the necessary information to search for a trip

USER STORIES: As a customer, I am able to browse through the list of available itineraries, then sort the ticket either by date or time.

USE CASE: ADD TO CART

ACTOR: Customer

FLOW OF EVENTS:

- The use case begins when customer press “add ticket to cart” button. Customer presses the cart button and views the selected ticket. Customer then proceeds to checkout.

USER STORIES:

1. Trip Summary:

PRE-CONDITION: A list of available tickets is displayed. The customer has already found a suitable ticket.

USER STORIES: As a customer, after I click add to cart, I should be able to view the cart summary, such as train name, travel from and to locations along with date and time of travel, total price, in the checkout page.

2. One Way Trip Total Price:

PRE-CONDITION: A list of available tickets is displayed. The customer has already found a suitable ticket.

USER STORIES: As a customer, I will be able to confirm the price for the one-way trip that I selected and display the total price in each page after selecting price and class.

3. Round Trip Total Price:

PRE-CONDITION: A list of available tickets is displayed. The customer has already found a suitable ticket.

USER STORIES: As a customer, I will be able to confirm the price for the departing trip, and then select the ticket and confirm the total price for both ways.

USE CASE: CHECKOUT

ACTOR: Customer

FLOW OF EVENTS:

- The use case begins when customer check and make sure that they add the right tickets to cart. She then enters Name, Billing Address, Travelers’ information, phone number, email address. Customer reviews the orders and finally places orders.

USER STORIES:

1. All Passengers' Info:

PRE-CONDITION: There a ticket in the cart and the customer chooses to checkout.

USER STORIES: As a customer, I am able to input information such as name, email, and age for all of the passengers for a particular trip

POST-CONDITION: After the customer hit the "place order" button, the system will save all the customer's information into the database.

2. Billing Adddress:

PRE-CONDITION: There a ticket in the cart and the customer chooses to checkout.

USER STORIES: As a customer, while checking out the order I shall be able to add the billing address in check out page before processing payment and it should be available for viewing in later stages.

POST-CONDITION: After the customer hit the "place order" button, the system will save all the customer's information into the database.

3. Payment Method:

PRE-CONDITION: There a ticket in the cart and the customer chooses to checkout.

USER STORIES: As a customer, I am able to choose my preferred payment method from credit card or debit card or other alternate payments such as Stripe, Bean stream, Braintree, money order and PayPal. and view the total price before placing my order.

POST-CONDITION: After the customer hit the "place order" button, the system will save all the customer's information into the database.

USE CASE: SIGN IN/REGISTER

ACTOR: Customer

FLOW OF EVENTS:

- The use case begins when the customer chooses to sign into her account. She clicks on the sign in button and choose to sign in using either her Google Account or GitHub account.

USER STORIES:

1. Register:

As a customer, I am able to register for a new account using my own GitHub account or Google Account.

2. Sign in:

As a customer, I am able to sign in using my own GitHub account or Google Account.

3. Sign out:

PRE-CONDITION: A customer had already signed in

USER STORIES: As a customer, from any page, I should be able to sign out of my account by using sign out option provided at right corner of the page.

USE CASE: MANAGE USER'S PROFILE

ACTOR: Customer

FLOW OF EVENTS:

- The use case begins when customer click on their account setting. Customer views their profile setting. Customer changes password. She changes her billing address and information since she just moved. Customer then views her payment method. Customer choose to save all the changes.

USER STORIES:

1. View Profile:

PRE-CONDITION: A customer has already had an account

USER STORIES: As a customer, from my profile page, I am able to view all of the information such as name, email, address and preferred payment methods on my profile page.

2. Edit Profile:

PRE-CONDITION: A customer has already had created an account

USER STORIES: As a customer, from my profile page, I must be able to edit all of my information such as name, address, email, age, and preferred payment method.

POST-CONDITION: The system database will store the new customer's information.

USE CASE: MANAGE ADMIN USER

ACTOR: Admin

FLOW OF EVENTS:

- The event starts when an Admin can log in to the Administration page and view the user's tab. Admin can create new user, update the user’s information and he can able to delete the user and unlock the user account, if it is locked.

USER STORIES:

1. View Admin List:

PRE-CONDITION: The system's database must have at least one admin user

USER STORIES: As an Admin when I click on the manage users, a list of all the admin users shall be populated in a table. each record will have a username, password, email.

2. Add/Remove an Admin

PRE-CONDITION: The admin logins to the administration page.

USER STORIES: As an Admin, I can add a new admin to the system by entering all the user information, and choose the appropriate Admin's permission for that user. Or I can remove an admin from the system.

POST-CONDITION: The system will save the new admin's information into the database, including the list of permission provided.

3. Edit an Admin

PRE-CONDITION: The admin logins to the administration page.

USER STORIES: As an Admin, I can edit an admin by changing his password, permissions...

POST-CONDITION: The system will save the new admin's information into the database, including the list of permission provided.

USE CASE: MANAGE ORDERS/BOOKINGS

ACTOR: Admin

FLOW OF EVENTS:

- The event starts when an Admin can log in to the Administration page and view the order's tab. Admin can manage the user's order, cancel order and he can able to generate the user's orders history report.

USER STORIES:

1. View orders/bookings list

PRE-CONDITION: There must be at least an order in the system database

USER STORIES: As an admin, when I click on the Manage Orders menu, all the orders from the database shall be populated in the page.

2. Add or Remove Orders/Bookings

PRE-CONDITION: There must be at least an order in the system database

USER STORIES: As an admin, I am able to add a new order into the system or remove the order from the system.

POST-CONDITION: The database will change the order based on what was edited.

3. Edit an order/booking

PRE-CONDITION: There must be at least an order in the system database

USER STORIES: As an admin, I am able to edit some information like destinations, quantity... of an already existed order upon customer requests or other circumstances.

POST-CONDITION: The database will change the order based on what was edited.

USE CASE: MANAGE CUSTOMERS

ACTOR: Admin
FLOW OF EVENTS:

- The event starts when an Admin can log in to the Administration page and view the customer's tab. Admin can able to manage registered and unregistered customers activities.

USER STORIES:

1. View Customer List

PRE-CONDITION: The database must contain at least a customer

USER STORIES: As an Admin, when I click on the Manage Customer menu, customer list which contains customer id, customer name, email, contact number shall be populated on the page.

2. Add or Remove a Customer

PRE-CONDITION: The database must contain at least a customer

USER STORIES: As an Admin, when I click on the add customer submenu, the form shall be displayed. when I enter customer name, email, age, and contact number then hit the save button, the customer record shall be saved.

POST-CONDITION: The system will save customer information in the customer table

3. Edit a Customer's info

PRE-CONDITION: The database must contain at least a customer

USER STORIES: As an Admin when I click on the customer record, in the customer list, the system shall navigate to the edit form with prefilled customer data. when I click on update, the latest customer details have to be updated.

POST-CONDITION: The system will save all the new changes to customer table

---

USE CASE: MANAGE GROUPS

ACTOR: admin

FLOW OF EVENTS:

- The event starts when an Admin can log in to the Administration page and view the group's tab. The admin can manage groups of admin users by adding or dropping admin to or from one group, setting the group permission.

USER STORIES:

1. View all groups:

PRE-CONDITION: The database must have at least many different admin users

USER STORIES: As an Admin, I view all the created group in the system

2. Add/Remove a group:

PRE-CONDITION: The database must have at least many different admin users.

USER STORIES: As an Admin, I can create a new group and set all of its permission or remove an already existed group from the system.

POST-CONDITION: The system will save the new changes to the group into the database

3. Edit Group:

PRE-CONDITION: The database must have at least many different admin users

USER STORIES: As an Admin, I can change the Group setting such as set the permission for each group, as well as add/remove admin to or from the group.

POST-CONDITION: The system will save the new changes to the group into database

---

USE CASE: MANAGE ITINERARIES

ACTOR: admin

FLOW OF EVENTS:

- The event starts when an Admin can log in to the Administration page and view the itineraries' tab. Admin can able to manage to add the stops, remove the stops, manage the tickets, and route information.

USER STORIES:

1. View destination list:

PRE-CONDITION: There must be stations information in the database

USER STORIES: As an Admin, I can view all existed stations as well as train operates time.

2. Add/Remove destinations:

PRE-CONDITION: There must be stations information in the database

USER STORIES: As an Admin, I can add a new destination as well as all of its configuration or remove a destination from the system database.

POST-CONDITION: The system's database will reflect the new changes.

3. Edit a destination:

PRE-CONDITION: There must be stations information in the database

USER STORIES: As an Admin, I can add new changes to the destination as well as all of its configuration

POST-CONDITION: The system's database will reflect the new changes

---

USE CASE: PAYMENT MANAGEMENT

ACTOR: admin
FLOW OF EVENTS:

- The event starts when an Admin can log in to the Administration page and view the payment's tab. Admin manage all the payment's configurations

USER STORIES:

1. View payment's method list:

PRE-CONDITION: The admin logins to the administration page.

USER STORIES: As an Admin, I can view the list of all payment method which system is supporting right now. such as Stripe, Braintree, bean steam, PayPal and Money Order

2. Enable/Disable Payment methods:

PRE-CONDITION: The admin logins to the administration page.

USER STORIES: As an Admin, I can edit the payment method by enabling or disabling a payment method. I can also configure the payment method.

POST-CONDITION: The system database will be updated with the new changes in payment

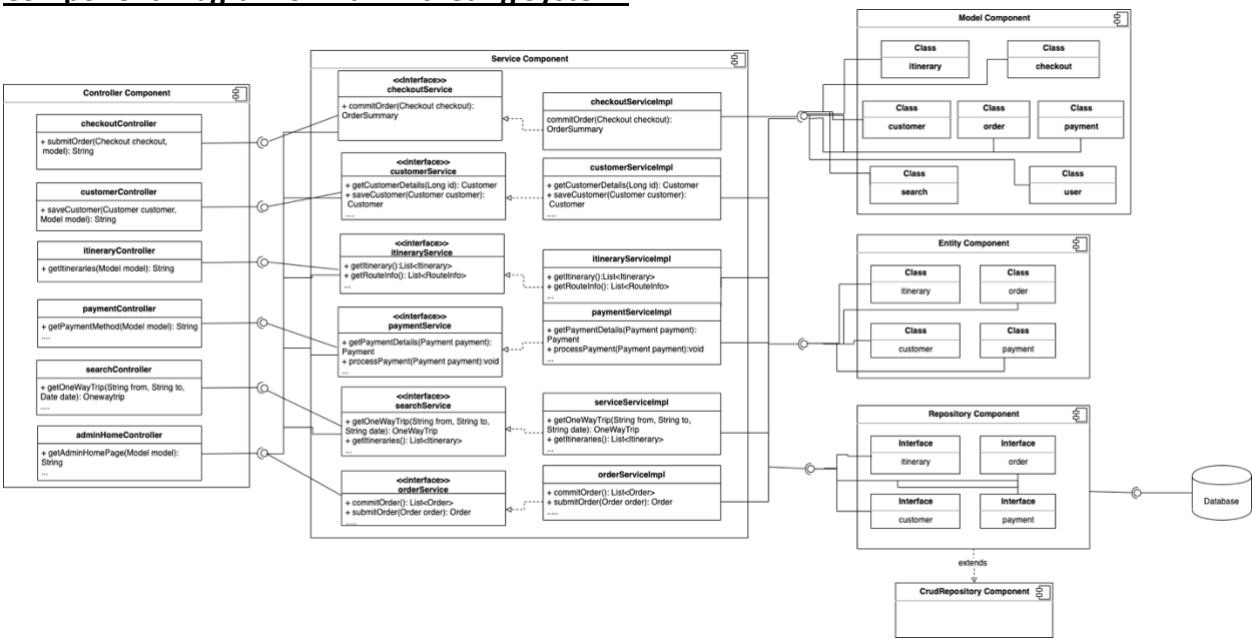
3. Add new payment method:

PRE-CONDITION: The admin logins to the administration page.

USER STORIES: As an Admin, I can add a new payment method, along with configuration the API key for the newly added payment method.

POST-CONDITION: The system database will be updated with the new changes in payment

Component Diagram of Train Ticketing System:



Deployment Diagram for Train Ticketing System: