TrainTicket System Documentation v 1.0

Created March 19, 2020

*1. UML Use Case Sequence*

For simplicity, all TrainTicket Users and Administrators are considered Customers when navigating the Store.

1. UML Use Case Sequence: Search for Tickets
   1. Description: This Use Case details how a TrainTicket customers may utilize TrainTicket Webpage’s search functions to lookup available Tickets.
   2. Actors:
      1. Customer
      2. TrainTicket Webpage and Server
   3. Preconditions:
      1. Customer have TrainTicket Homepage opened on his or her device with active internet connection.
      2. TrainTicket Servers are online and not under maintenance.
      3. Customer have previously registered a User Account with TrainTicket and have successfully signed in.
   4. Use Case Sequence:
      1. Customer navigates to and click on Search Box on TrainTicket Webpage.
      2. Customer selects One-way Trip (default option).
      3. Customer will be prompted to enter Departure Date, Origin, and Destination.
      4. Customer will be prompted to Enter the number of Adult, Senior, Infant and Child Ticket(s).
      5. Customer will press “Search” button.
   5. Alternative Use Case Sequence:
      1. Customer navigates to and click on Search Box on TrainTicket Webpage.
      2. Customer selects Round-Trip option.
      3. Customer will be prompted to enter Departure Date, Origin, Destination and Return Date.
      4. Customer will be prompted to Enter the number of Adult, Senior, Infant and Child Ticket(s).
      5. Customer will press “Search” button.
   6. Post-conditions:
      1. TrainTicket webpage will return a list of search results that matches the Search Criteria.

User Story:

* When the customer/Administrator enter the website, the system should display homepage with find trains sections with one-way option selected by default.
* When the Customer/Administrator enters Origin, Destination, Date of departure, Date of Return( in case of Round-Trip ), Preferred time of travel, number of travelers, coupon/promo code(optional) and clicks "Find Trains", then the system should populate list of all trains matching that condition.
* When the Customer/Administrator selects multi-city option, then the system should redirect to a page with multiple segments input form.
* As a Customer, I want to be able to purchase one-way tickets for my grandmother, kids and myself.
* As a Customer, I want to be able to purchase round-trip tickets for my infant and myself.
* As a Customer, I want to be able to purchase tickets to multiple cities.

1. UML Use Case Sequence: Display Available Itineraries
   1. Description: This Use Case details how a TrainTicket System user may utilize the Website’s Itinerary screen to view list of available trains matching their search criteria.
   2. Actors:
      1. Customer
      2. TrainTicket Webpage and Server
   3. Preconditions:
      1. Customer have TrainTicket Homepage opened on his or her device with active internet connection.
      2. TrainTicket Servers are online and not under maintenance.
      3. Customer have previously registered a User Account with TrainTicket and have successfully signed in.
      4. Customer successfully executed Search Tickets Functionality.
   4. Use Case Sequence:
      1. The website will load list of all itinerary that matches the Search criteria.
      2. For every matched itinerary, the website will display the train name, train timing, duration of travel, cost by coach class and amenities available in that train along with add to cart option.
      3. Customer/Administrator will be able to select the fare they wanted to purchase before selecting Add to Cart.
      4. Customer/Administrator will be able to sort the displayed itineraries based on the coach class, travel duration, fare and departure time.
      5. Customer/Administrator will be able to filter the displayed itineraries time of travel and train name.
   5. Alternative Sequence:
      1. If there is no itinerary available for the search conditions of the Customer/Administrator, then the website will display “No Trains Found on the selected Route” message.
   6. Post-conditions:
      1. When the Customer/Administrator Adds an itinerary to Cart, the website will direct to add cart options.

User Story:

* When the Customer/Administrator clicks on "Find trains", then the system should display the list of all available itinerary that matches the search criteria in a tabular format.
* The displayed itinerary should be grouped by coach. Ticket fare based on coach should be displayed along with any discounted fares and the system should alert if there are very few tickets available at a particular fare under a segment.
* The system should provide sort and filter options on the results and add to cart option.
* As a Customer, I want to view available itineraries based on origin and date or departure.
* As a Customer, I want to view available itineraries based on origin and price.
* As a Customer, I want to view available itineraries along with available amenities and Coach Class.

1. UML Use Case Sequence: Add to cart
   1. Description: This Use Case details how a TrainTicket System will allow a Customer to add an itinerary to the Cart.
   2. Actors:
      1. Customer
      2. TrainTicket Webpage and Server
   3. Preconditions:
      1. Customer have TrainTicket Homepage opened on his or her device with active internet connection.
      2. TrainTicket Servers are online and not under maintenance.
      3. Customer have previously registered a User Account with TrainTicket and have successfully signed in.
      4. Display Available Itineraries use case is executed and TrainTicket returned a list of available itineraries.
   4. Use Case Sequence:
      1. The Customer/Administrator navigates to any of the listed Itineraries and view the total fares for the travelers selected.
      2. The Customer/Administrator will select a fare of their choice and click on “Add to Cart” Button.
      3. The Website will provide an option for the Customer to select Add-Ons based on the amenities available in that segment.
      4. If the Customer selects an Add-On, then the fare will reflect to include new fare.
   5. Alternative Sequence:
      1. The Customer/Administrator will exit the website without adding an item to Cart after viewing the displayed itineraries.
   6. Post-conditions:
      1. Customer’s cart will be updated to reflect additions.
      2. The Website will allow the Customer to check out the Item(s) added to Cart.

User Story:

* When the Customer/Administrator selects a ticket and clicks on "Add to Cart”, then the system should add the ticket to the cart.
* When the Customer/Administrator clicks on continue, then the system should redirect to add-on’s page.
* When the Customer/Administrator selects any add-ons, then The system should calculate and display new cost of the ticket based on the selection.
* As a Customer, I want to save itineraries by adding selected itineraries to Cart.
* As a Customer, I want to be able to add amenities to Cart.
* As a Customer, I want to view items added to Cart.

1. UML Use Case Sequence: Check out
   1. Description: This Use Case details how a TrainTicket Customer may order Train Tickets online.
   2. Actors:
      1. Customer
      2. TrainTicket Webpage and Server
   3. Preconditions:
      1. Customer have TrainTicket Webpage opened on his or her device with active internet connection.
      2. TrainTicket Servers are online and not under maintenance.
      3. Customer have previously selected a valid Itinerary from the Webpage’s Search Function and added it to Cart.
   4. Use Case Sequence:
      1. Customer navigates to Cart and clicks on the “Checkout” Option.
      2. Customer will be directed to Billing Page.
      3. Customer will be prompted to Sign In, Register, or use Guest Check Out.
      4. If Customer have previously Registered and Signed In, Customer Personal Information and Billing Information will be automatically filled out from previously saved Profile Information.
      5. Customer navigates to and clicks Confirm Order.
   5. Alternative Sequence:
      1. If Customer selects “Guest Check Out”, Customer will be prompted to enter valid Personal Information and Billing Information.
      2. Customer navigates to and clicks Confirm Order.
   6. Post-conditions:
      1. TrainTicket Servers will update Ticket and Train Seat availability.
      2. TrainTicket Webpage will return a printable Ticket for Customer.
      3. TrainTicket Servers will email Customer with a printable Ticket and Itinerary.

User Story:

* When the customer/administrator clicks on "Checkout", then the system should redirect to billing and checkout page, where the personal information of the user will be asked for guest users and will be displayed for registered users.
* When the customer/administrator enters /reviews personal information and clicks on "Continue", then the system will redirect to payment page where the user can enter the payment information and will display the cost of the purchase.
* When the customer/administrator enters payment information and clicks on "Review", then the system will redirect to review page where the user can click on "Purchase" to reserve a ticket or "Cancel" to terminate the transaction.
* As a Customer, I want to pay my train fare via Credit Card.
* As a Customer, I want to pay my train ticket via Electronic Check.
* As a Customer, I want to review my train fare transaction before confirming the payment.

1. UML Use Case Sequence: Sign in/Register
   1. Description: This Use Case details how Customers may create a User Account with TrainTicket system.
   2. Actors:
      1. Customer
      2. TrainTicket Webpage and Server
   3. Preconditions:
      1. Customer has the TrainTicket Webpage opened on his or her device with active internet connection.
      2. Customer have not previously registered for an account with TrainTicket.
      3. TrainTicket Servers are online and not under maintenance.
   4. Use Case Sequence:
      1. Customer navigates to and select the “New User” Option.
      2. Customer must enter valid Personal Information and Sign-In Information (Username and Password) and click Create Account.
      3. After Account Creation, Customer may sign in by navigating to and selecting the “Sign In” Option.
   5. Alternative Sequence:
      1. Customer navigates to the My Account tab and select the Sign In Option without first registering for an account.
      2. A Register Option will be provided for the Customer to create new account.
   6. Alternative Sequence:
      1. Customer entered incorrect email address and or password.
      2. The webpage shall display “Login Failed. Username or Password is incorrect” along with the option to reenter login credentials again.
   7. Post-Conditions:
      1. Customer successfully authenticated and signed into his or her account.

User Story:

* When the customer clicks on "join", then the system should show form that needs to be filled by customer to register. When customer clicks on "register", then system should add the customer to the existing user’s list and should redirect to login page.
* When the customer enters username & password and clicks on "login", then the system should authenticate the credentials and loads the home page for the signed in user.
* When the customer clicks on "reset password", then the system should send password reset link to the customer’s email address.
* As a Customer, I want to create an Account with TrainTicket to save profile and billing information.
* As a Customer, I want to be able to sign into a previously registered TrainTicket Account.
* As a Customer, I do not want unauthorized Users to log into my account.

1. UML Use Case Sequence: Manage user profile for registered users
   1. Description: This Use Case details how an TrainTicket customer may modify his or her TrainTicket Account Profile.
   2. Actors:
      1. Customer
      2. TrainTicket Webpage and Server
   3. Preconditions:
      1. Customer have TrainTicket Webpage opened on his or her device with active internet connection.
      2. TrainTicket Servers are online and not under maintenance.
      3. Customer have previously Registered and Create an Account with TrainTicket.
      4. Customer successfully authenticated and signed into his or her account.
   4. Use Case Sequence:
      1. Customer navigates to and clicks on the My Profile Tab in his or her account page.
      2. Customer may change Username, Password, Personal Information, Billing, and Address.
   5. Post-conditions:
      1. TrainTicket servers will be updated to reflect changes to Customer Account Profile.

User Story:

* When the logged in user clicks on manage profile, then the system should allow load the profile page of the user.
* The system should allow the user to change the Billing Address, to modify password, email address and mobile number. When the user clicks on "Save", then the system should save the changes made.
* When the user clicks on "manage my reservations", then the system should display editable upcoming reservations and view only past reservations.
* As a Customer, I want to modify my User Profile to reflect new billing address.
* As a Customer, I want to change my Password.
* As a Customer, I want to update my email address.

Administrator UML Use case Sequence

Administrators are a Subclass of Users and have the following additional functionalities.

1. UML Use Case Sequence: Manage orders
   1. Description: This Use Case details how an Administrator can examine and modify Ticket orders.
   2. Actors:
      1. Administrator
      2. TrainTicket Webpage and Server
   3. Preconditions:
      1. Administrator has the TrainTicket Administrator Webpage opened on his or her device with active internet connection.
      2. TrainTicket Servers are online and not under maintenance.
      3. Administrator have successfully logged in and authenticated Administrator Account.
   4. Use Case Sequence:
      1. Administrator navigates to the Orders Tab in the Admin Webpage.
      2. The Administrator may view the list of orders and select specific orders from the List or Orders tab.
      3. The Administrator will be able to cancel a customer’s reserved tickets and issue a refund.
      4. The Administrator may modify billing information, schedule, train, origin, and destination (itinerary information).
   5. Post-conditions:
      1. The Customer’s itinerary and Ticket will be updated to reflect changes.
      2. The Customer’s Ticket refund status will be updated to reflect changes.
      3. TrainTicket’s Servers will be updated to reflect changes to passengers and seat availability.

User Story:

* When the logged in administrator clicks on manage orders, then the system should display list of active orders.
* The system should allow the administrator to reschedule a Ticket, to modify pricing for Train Ticket, to change the billing information for a Ticket Order.
* The system should allow the administrator to cancel the booked tickets of the customer and to issue a refund.
* As an Administrator, I want to view and manage the list of active Orders.
* As an Administrator, I want to cancel booked Tickets and issue refunds should the need arise.
* As an Administrator, I want to reschedule or modify pricing on active Ticket Orders.

1. UML Use Case Sequence: Manage Users (of the system)
   1. Description: This Use Case details how an Administrator can modify, create, or delete Users.
   2. Actors:
      1. Administrator
      2. TrainTicket Webpage and Server
   3. Preconditions:
      1. Administrator has the TrainTicket Administrator Webpage opened on his or her device with active internet connection.
      2. TrainTicket Servers are online and not under maintenance.
      3. Administrator have successfully logged in and authenticated Administrator Account.
   4. Use Case Sequence:
      1. Administrator may navigate to the Users Tab on the TrainTicket Administrator Webpage.
      2. The webpage will return list of Users along with level of Administrative Privileges.
      3. Administrator may select the (-) button next to a User’s name to delete user.
      4. The Webpage will prompt Administrator to Confirm Deletion.
   5. Alternative Use Case:
      1. The Administrator may select the Create User Option under Users Tab to generate new User(s)
      2. Webpage will prompt Administrator to enter new User profile information and Administrative Level.
   6. Post-conditions:
      1. TrainTicket’s Servers will update to reflect User Changes.

User Story:

* When the logged in administrator clicks on manage users, then the system should display list of all existing users.
* The system should allow the administrator to create/update existing users of various categories.
* The system should allow the administrator to delete existing users.
* As an Administrator, I want to delete inactive Users.
* As an Administrator, I want to create new Users.
* As an Administrator, I want to modify User information and Administrative privileges.

1. UML Use Case Sequence: Manage Customers (registered or unregistered travelers)
   1. Description: This Use Case details how an Administrator can manage Customers and Customer Options.
   2. Actors:
      1. Administrator
      2. TrainTicket Webpage and Server
   3. Preconditions:
      1. Administrator has the TrainTicket Administrator Webpage opened on his or her device with active internet connection.
      2. TrainTicket Servers are online and not under maintenance.
      3. Administrator have successfully logged in and authenticated Administrator Account.
   4. Use Case Sequence:
      1. The Administrator may navigate to the Customers tab in the TrainTicket Admin Webpage.
      2. Selecting the Customers Tab will return a List of Customer IDs for both registered and unregistered travelers.
      3. The Administrator may select a Customer ID from the list and modify or view Customer Options.
   5. Post-conditions:
      1. TrainTicket’s customer list and options will be updated to reflect changes.

User Story:

* When the logged in administrator clicks on manage customers, then the system should display list of all existing customers.
* The system should allow the administrator to create/update existing customers
* The system should allow the administrator to view transactions made by each customer.
* As an Administrator, I want to create Customer profiles.
* As an Administrator, I want to modify Customer profiles.
* As an Administrator, I want to view transactions made by Customers.

1. UML Use Case Sequence: Manage Groups
2. Description: This Use Case details how an Administrator can manage different user groups
3. Actors:
4. Administrator
5. TrainTicket Webpage and Server
6. Preconditions:
7. Servers are online and not under maintenance.
8. Administrator has the Administrator Webpage opened on his or her device with active internet connection.
9. Administrator has successfully logged in and authenticated with Administrator Account.
10. Use Case Sequence:
11. Administrator navigates to the Groups tab in TrainTicket System’s Admin webpage which will return a list of different security groups available.
12. Administrator will be able to view, modify, or delete users from any existing groups by clicking on that group.
13. Administrator will be able to create new security group and specify access rights of that group by filling out the appropriate fields.
14. Administrator will be able to modify or delete security groups except ‘Administrator’ group.
15. Post-conditions:
16. Train Ticket’s Security groups will be updated to reflect the changes.

User Story:

* When the logged in administrator clicks on manage groups, then the system should display list of existing security groups.
* The system should allow the Administrator to create/update security groups, add, update or delete users from existing security groups.
* When the logged in administrator clicks on ‘Administrator group’ from the list of groups displayed, then the system should display read-only details of that group.
* As an Administrator, I want to view security groups associated with TrainTicket Systems.
* As an Administrator, I want to delete security groups associated with TrainTicket Systems.
* As an Administrator, I want to create new security groups in TrainTicket Systems with access specifications.

1. UML Use Case Sequence: Manage Itineraries
   1. Description: This Use Case details how an Administrator can configure Itineraries.
   2. Actors:
      1. Administrator
      2. TrainTicket Webpage and Server
   3. Preconditions:
      1. Administrator has the TrainTicket Administrator Webpage opened on his or her device with active internet connection.
      2. TrainTicket Servers are online and not under maintenance.
      3. Administrator have successfully logged in and authenticated Administrator Account.
   4. Use Case Sequence:
      1. Administrator may navigate to the Itineraries Tab in the TrainTicket Admin Webpage which returns a list of all possible Itineraries.
      2. Administrator may add new Departure/Arrival location by selecting “Add Itinerary” option under the Itineraries Tab.
   5. Alternative Use Case:
      1. Administrator may modify existing by selecting an Itinerary from the List and selecting “Modify” which returns information regarding the Itinerary.
      2. Administrator may modify stops, pricing, availability, and route information from each respective field.
   6. Post-conditions:
      1. TrainTicket’s Server shall be updated to reflect changes in Itinerary.

User Story:

* When the logged in administrator clicks on manage itineraries, then the system should display list of all existing itineraries.
* The system should allow the administrator to add new itinerary or modify the existing itinerary
* The system should allow the administrator to add new stops to the system.
* As an Administrator, I want to add new Arrival or Departure destinations.
* As an Administrator, I want update availability for a specific route.
* As an Administrator, I want to update pricing for a specific route.

1. UML Use Case Sequence: Manage Customer Information.
   1. Description: This Use Case details how an Administrator can manage customer information.
   2. Actors:
      1. Administrator
      2. TrainTicket Webpage and Server
   3. Preconditions:
      1. TrainTicket Servers are online and not under maintenance.
      2. Administrator has the TrainTicket Administrator Webpage opened on his or her device with active internet connection.
      3. Administrator have successfully logged in and authenticated Administrator Account.
   4. Use Case Sequence:
      1. The Administrator may navigate to the Customers tab in the TrainTicket Admin Webpage.
      2. Selecting the Customers Tab will return a List of Customer IDs for both registered and unregistered travelers.
      3. The Administrator may select a Customer ID from the list and modify or view Customer Information by selecting the appropriate action.
      4. Selecting View will return the Customer Profile information.
      5. Selecting Modify will return the Customer Profile with changeable fields.
   5. Post-conditions:
      1. TrainTicket’s Servers shall be updated to reflect Customer Profile changes.

User Story:

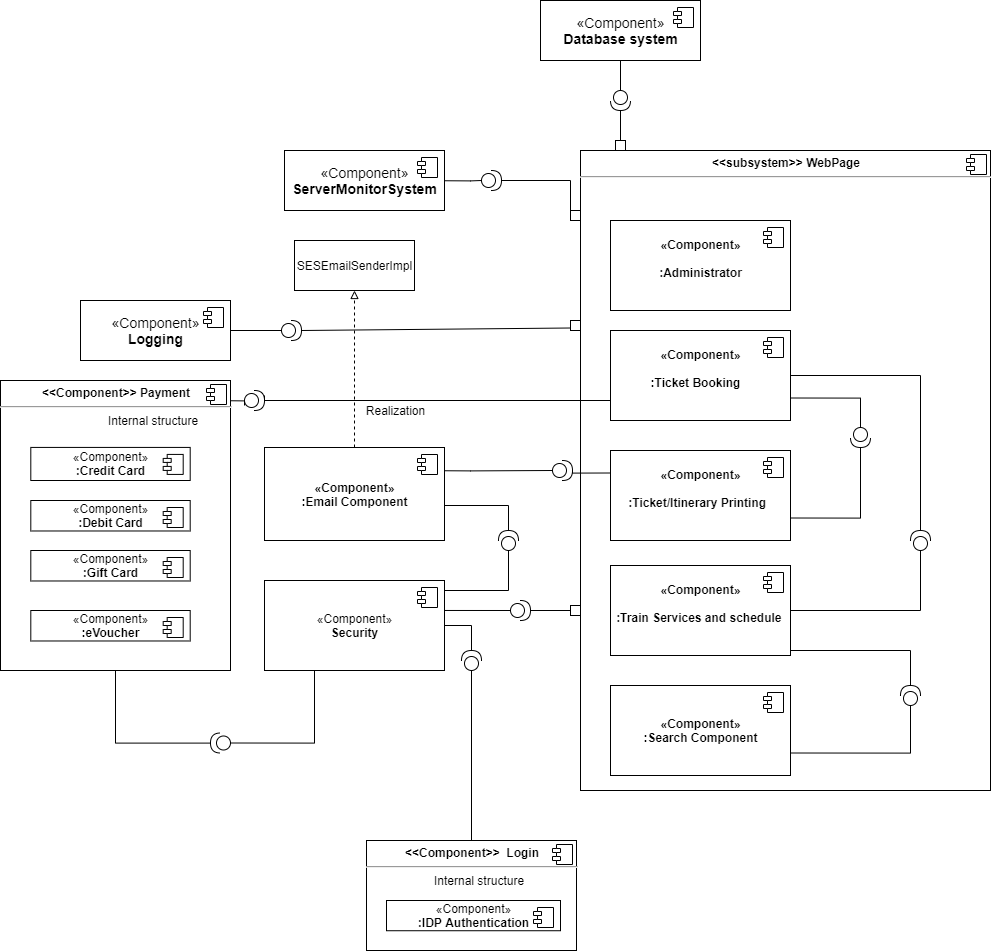
* When the logged in administrator clicks on manage customer information, then the system should display list of all existing registered/unregistered travelers and their customer ID.
* When the administrator clicks on a customer ID, then the system should load that traveler's profile for view or modification.
* When the administrator updates the information and clicks on save, then the system should save the updates.
* As an Administrator, I want to update Customer billing addresses.
* As an Administrator, I want to update Customer email addresses.
* As an Administrator, I want to reset Customer passwords should a Customer make a reset password request.

1. UML Use Case Sequence: Payment management
   1. Description: This Use Case details how an Administrator can configure payment options.
   2. Actors:
      1. Administrator
      2. TrainTicket Webpage and Server
   3. Preconditions:
      1. TrainTicket Servers are online and not under maintenance.
      2. Administrator has the TrainTicket Administrator Webpage opened on his or her device with active internet connection.
      3. Administrator have successfully logged in and authenticated Administrator Account.
   4. Use Case Sequence:
      1. The Administrator may navigate to the Payment tab in the TrainTicket’s Admin Webpage which returns a list of Payment and Payment Processing Options.
      2. The Administrator may add a new payment option by clicking on the “Add Payment Method” button
      3. The Administrator may Enable/Disable Money Order, PayPal, Beanstream, Stripe, and or Braintree payment processing options by toggling each on or off.
   5. Post-conditions:
      1. The TrainTicket’s Servers shall update to reflect payment processing changes.

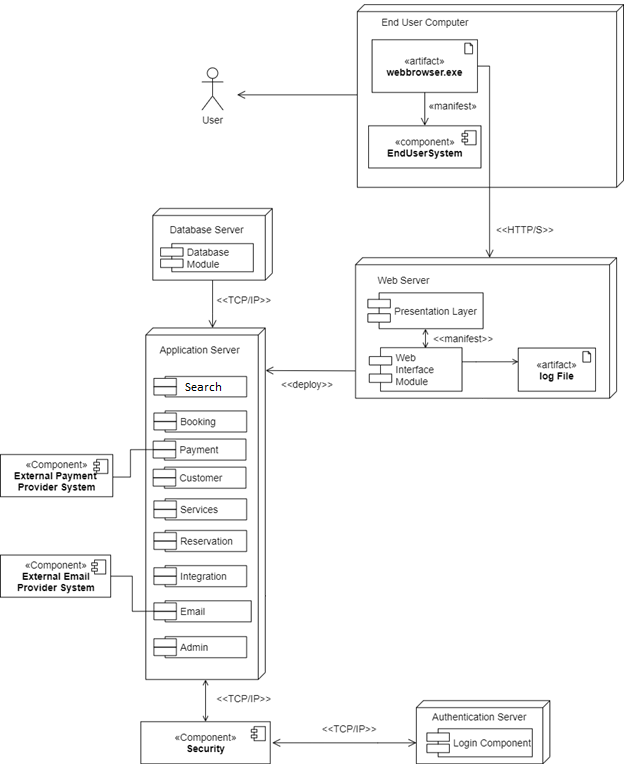
User Story:

* When the logged in administrator clicks on payment management, then the system should display
* existing modes of payment along with their status if they are enabled or disabled
* When the logged in administrator clicks on add new payment, then the system should allow the administrator to add a new mode of payment by asking for payment details.
* When the logged in administrator clicks on any existing new payment, then the system should allow the administrator to disable/enable it.
* As an Administrator, I want to change payment options to include Money Orders.
* As an Administrator, I want to change payment processing services to PayPal.
* As an Administrator, I want to disable Stripe Payment Processing Services.

*2. TrainTicket UML Component Diagram:*



*3. TrainTicket UML Deployment Diagram:*



*4. TrainTicket UML Sequence Diagram:*

(1) Search tickets:



(2) Check out:



(3) Manage itineraries: To be Implemented.

*5. TrainTicket UML Class Diagram:*

(1) Search tickets:



(2) Check out:



(3) Manage itineraries: To be Implemented.

*6. Quality Attributes:*

1. TrainTicket Security: Customer Login (Not Implemented)
   1. Source: Customer who had previously registered with TrainTicket
   2. Stimulus: 5 consecutive Invalid User Password Entered over a 20-minute period.
   3. Artifact: TrainTicket User Login System
   4. Environment: Normal Operations
   5. Response:
      1. Notify customer that credentials are incorrect every time invalid credentials are entered.
      2. Log Invalid login attempt and timestamp.
      3. Warn customer of imminent account lockout after 4 invalid login attempts.
      4. Notify customer via email (associated with Username) of multiple invalid login attempts and prompt customer to lock account/change password.
      5. Lock account for 24 hours after 5th failed password attempt.
   6. Response Measure:
      1. Block login attempt with incorrect user credentials 100% of the time.
      2. Warn customer of imminent account lockout after 4 invalid login attempts 100% of the time.
      3. Notify Customer via email (associated with Username) 100% of the time.
      4. Lock account for 24 hours after 5th failed password attempt 100% of the time.
   7. Associated Classes/Files: (Not Implemented)
      1. User.java
      2. UserController.java
      3. UserService.java
      4. IUserService.java
      5. SecurityConfiguration.java
2. TrainTicket Modifiability: Add New Table/Update Table in Database
   1. Source: TrainTicket Developer
   2. Stimulus: TrainTicket Developer wishes to update/add new Table to Database
   3. Artifact: TrainTicket Database (PostgreSQL)
   4. Environment: Design, compile and build environment
   5. Response: Developer may edit init\_table.sql and init\_data.sql in Amtrak\_sql folder and adjust code related to the SQL files to accommodate. Alternatively, Developer may update and create tables via pgAdmin4 App.
   6. Response Measure: Database can be updated and redeployed with new data within 3 hours.
   7. Associated Classes/Files:
      1. Init\_table.sql
      2. Init\_data.sql
3. TrainTicket Interoperability: Payment Processing
   1. Source: TrainTicket Billing and Payments system
   2. Stimulus: Valid Credit Card Information Sent (to Stripe Servers)
   3. Artifact: Stripe Payment Processing System
   4. Environment: Systems known prior to runtime
   5. Response: Stripe Payment Processing Validates Payment and Billing Information, stores payment information, and returns valid payment and billing information to TrainTicket.
   6. Response Measure: Billing and Payment Information correct 99.99% of the time.
   7. Associated Classes/Files:
      1. StripeService
      2. IStripeService
      3. stripPayment.html