E-Commerce Platform Model Implementation using EMF

1. Introduction

This report outlines the development of an E-Commerce platform metamodel using the Eclipse Modeling Framework (EMF). The model is designed to capture the core components of an e-commerce system, including customers, products, orders, and promotions. It enforces data consistency through constraints, dynamically computes key metrics with derived fields, and supports business logic through operations. A UML diagram (Figure 1) provides a visual representation of the model's structure and relationships.

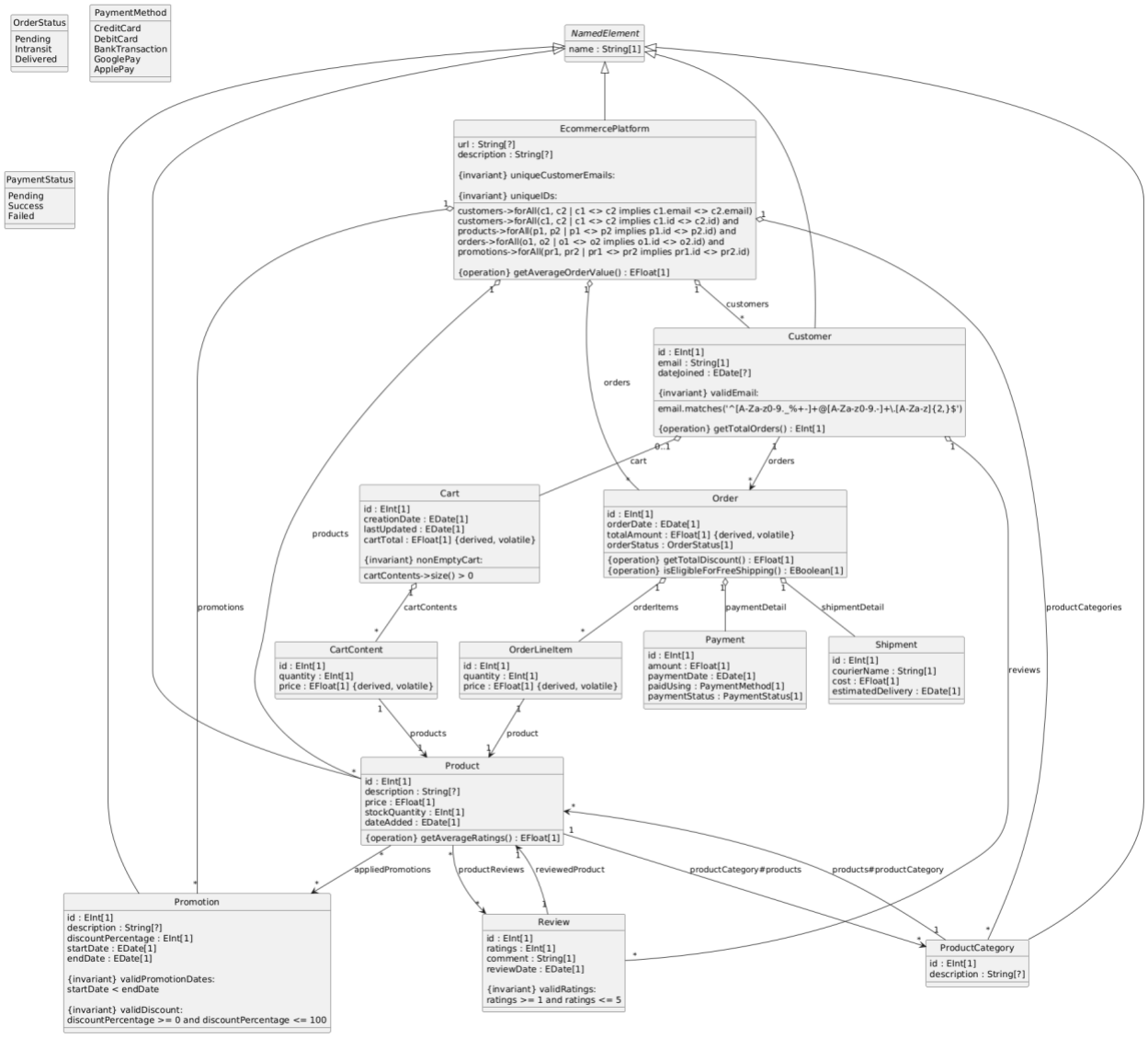

Figure 1: Figure 1 E-commerce Metamodel UML(generated using plantuml)

2. Model Description

The **EcommercePlatform** serves as the core class, aggregating customers, products, product categories, orders, and promotions. It has operations to calculate the average value of all orders placed on the platform.

The **Customer** class represents users, identified by unique IDs and email addresses. Customers can place orders, leave reviews, and maintain a shopping cart. The model ensures that all customer emails follow proper format rules.

ProductCategory groups products into logical categories such as electronics or clothing.

The **Product** class defines items available for sale, with details like price, stock quantity, and date added. It supports an operation to calculate the average rating of the product based on its reviews. Products can be part of one category and may have multiple promotions applied.

The **Order** class tracks customer purchases, including their associated products, payment, and shipment details. The order's total amount is automatically calculated from the price and quantity of the products included. It has operations to compute the total discount and determine eligibility for free shipping.

The **Payment** class records payment details such as the amount, payment date, payment method, and status.

The **Shipment** class includes shipment details like courier name, shipping cost, and estimated delivery date.

The **Cart** class temporarily stores items before checkout. It dynamically computes the total value of the items in the cart and ensures it is not empty.

The **CartContent** class represents individual items in the cart, specifying the product, quantity, and price. The price takes into account any applicable promotions on the product.

The **Review** class allows customers to provide feedback on products. Reviews include ratings, comments, and review dates. Ratings are restricted to a range of 1 to 5.

The **OrderLineItem** class represents individual items in an order, recording their quantity and derived price.

The **Promotion** class defines discounts and special offers that can be applied to products. Constraints ensure that promotions have valid dates and their discount percentages are within the range of 0–100%.

3. **Constraints**

The model enforces the following constraints to ensure data consistency:

- All IDs across customers, products, orders, and promotions are unique.
- Customer email addresses must follow a valid format.
- Promotions must have a start date earlier than the end date.
- Discounts must be between 0 and 100%.
- Shopping carts cannot be empty.
- Product reviews must have ratings within the range of 1 to 5.

4. **Derived Fields**

Certain fields in the model are automatically computed:

- **Order Total Amount**: Sum of the prices and quantities of products in the order.
- **Cart Total**: Aggregate value of the items in the shopping cart.
- **Cart Content Price**: Dynamically calculated for each cart item, considering promotions.
- **Order Line Item Price**: Derived from the quantity and price of the product.

5. **Operations**

Key operations in the model include:

- **EcommercePlatform**: Calculates the average value of all orders.
- **Customer**: Computes the total number of orders placed.
- **Product**: Calculates the average rating of a product based on reviews.
- **Order**: Computes total discounts and determines eligibility for free shipping.

6. **Plugin Project:** *it.univaq.disim.mde.course.assignment.four.business*

   The implementation began by creating a modeling project to define the e-commerce metamodel. The *genmodel* was then used to generate Java APIs, transforming the modeling project into a plugin project. A second plugin project, ***it.univaq.disim.mde.course.assignment.four.business***, was created which imports the modeling project and interact with the generated APIs programatically.

   The business plugin in this implements the following, as per the assignment requirements:

   - **Create**: Builds a fully instantiated EcommercePlatform model with realistic data, including customers, products, orders, and promotions.
   - **Serialize**: Saves the model to an `.xmi` file for persistence.
   - **Load**: Reloads the model from the `.xmi` file to verify its correctness.
   - **Validate**: Ensures the model adheres to business rules using OCL constraints, such as unique IDs and valid email formats.