**UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE**

**Team Inspector:** A-Byte Wizards

**Team Inspected:** ODS

**INSPECTION**

***Requirements***

***Requirements - Description***

The requirement document of this project is completely and also it has the main purpose of the requirements has to express. But there’re points with the comparison with the early description paragraph. The paragraph is summarized as Edison says something short, but the description is shorter than we expect, it isn’t implying with greater clarity what the system could do. To the rest comparison of the requirements and the paragraph is okay.

***Use Case Diagram***

The use case diagram is completely correct, it begins with verbs and the body is a complement, using the actors as a noun.

***Class Diagram***

Checking the class diagram with the code and obviously the classes on the code, as inspector and my co-inspectors (team), we could see the correct use and implementation of classes in the system, but the code has mistakes on the association part and the omitted attributes on diagram that the code has.

Observations and Possible Errors:

* The relationship between Ticket and Payment is 1..\* to 1. This indicates that a Ticket can have one or more Payments, and a Payment belongs to a single Ticket. This seems reasonable, although it could be clearer if it is indicated that a Ticket can have zero or more Payments.
* The relationship between Ticket and Baggage is unclear in terms of multiplicity. Indicates 0..\* for Baggage without clearly specifying how it relates to Ticket.

Consistency:

* The PaymentManager class has the method makePayment(paymentMethod : PaymentMethod, amount : double), but it is not clear how it is associated with PaymentMethod.

Conclusion:

The multiplicity and relationships in the diagram are sometimes not clear. It is important to ensure that all associations, aggregations and compositions are correctly specified.

**Class Diagram VS Use Case Diagram**

Reserve Flight:

In FlightManager, there should be a createFlight() or similar method, as well as methods for selecting flights and creating tickets.

Select Payment Method:

In TicketManager, there should be a sendTicketByEmail(customer : Customer, ticket : Ticket) method.

Imagen que contiene Diagrama

Descripción generada automáticamente