The presented class diagram illustrates the changes necessary to model UML constructs that do not exist in the Java language, The changes introduced are highlighted in green. The following section will describe the transformation process of certain UML constructs such as attributes types.

Runtime validation logic:

Using the java custom runtime exception and basic if checks we making validation we could use Java Bean valotion for same purposes and make it faster and simple and more like a c# .net but we dont what u goona do about it

Optional Attributes:

Atributes that may coontain no value they are represented in UML as atribute with [0..1] notaition and inplemented as null values in java with separate constructor so if dont wanna create one u just dont type on while constacting the object examples: Product:description[0..1] :String

Multi-value atributes:

Java is nativly supporting Multi-value atributes as classes inherited from Colection class we using for our purposes ArrayList<E> from java.util this its worth mentioning how and what for we using that collection first of all to store in our customers cart a peroduct that he added there and potentionaly willing to buy we used arraylist generic nature to fill our needs in diffrenct scenarious we save custom class products in our card using that implementation as well as we ussing more in basic wawy in our moderator system which can write and adjast tickets examples:**Cart-**unavailableProducts[0..*]:List<Product>,**Admin-**ArchiveLogs[0..*] :List<String>,**MarketModerator-**suspectedMerchants[0..*] : Map<String,Merchant>,**Support-**TicketsHistory[0..*] : List<String>

Derived Atributes

Derived atributes its such atributes which is not located as field in our implemetation and rather count on the run in java we used regualr get methods as in ecapsulation principle to simulate existing of these atributes for example;TimeLeft() so

```
public int TimeLeft() {
    return dateOfExpiration.getDays() - dateOfEstablishment.getDays();
}
```

Enums(tagged values)

We use it Transaction state in transaction class to deffrinciate from one another and base behaviour of our system based on it we use enums or enuminators for that purposes which is represented in java based libary

```
enum Status implements Serializable{
    FAILED,
    SUCCESSEFUL,
    WAITS
}

public class Transaction implements Serializable{
```

Complex atributes(class atributes)

Complex atributes its such atributes that are contain more then one simple field or could have more complex logic in they behaviour compare to the basic atrubites and field we implemented it using standart java tools they represented as inner classes of the class which have such complex atribute as an example i have here Date class which is inner class of DurationDate and it usses two fields Date one called start of seccond called end and they represent day motyj hour and year

```
public static class Date implements Serializable {
    private int year;
    private int month;
    private int day;
    private int hour;


    Date(int year, int month, int day, int hour) {
        if (month < 1 || month > 12 || day < 1 || day > 31 || hour < 0 || hour > 23) {
            throw new InvalidFormatException("Invalid date format");
        }
        this.year = year;
        this.month = month;
        this.day = day;
        this.hour = hour;
    }


public class DurationDate implements Serializable {
    private static final List<DurationDate> extent = new ArrayList<>();
    private static final String EXT_FILE = "duration_extent.ser";


    private Date dateOfEstablishment;
    private Date dateOfExpiration;
```

Extension classes(how we saving our progress):

Since using databasesis forbiden and some misterious wispers from within the reality dectates us to use an extension classes to save our systems basicly it method of saving objects in the file and initilazie it later in even completly another session of the programs and what was i describing was a Serialization and particulary waht we using in java for such puprposes is Serializable interface which of our classes inherit from to get that functionality also we have a static field which is represented with object of ArayList class which as we remember is a collection and everytime constractor of the class are triggered ArayList get filled up with our class and every point of them system all objects of the specific class can be saved in thorugh a static method also we using inharitance becouse by making that functionality in User all of its child classes like Merchant ,Advertiser,Shopper already have it implementation but its formality we can from any point we wont use any additional frameworks all of this just basic java so that it is how it is

Tests:

Here avtually we using the additional functional to make test we using JUnit testing for java as tool to create test easier and faster and more be more precised in our testing so we test our i would called straight forward normal people called dumb vilation of the atributes for every class we implemented and as well test testing fow our class seralization work or how our Extension calsses system work