Ministry of Education, Culture and Research of the Republic of Moldova

Technical University of Moldova

Department of Software and Automation Engineering

**REPORT**

Laboratory work No. 5

Discipline: AMS

Topic: Study and analysis of OO abstractions and classes in UML (class diagrams).

Analysis and modeling of an online store (e-commerce)

Done by: Lupascu Felicia

st.gr. FAF212

Verified by: univ.lect.  
 Sava Nina  
Melnic Radu

Chișinău 2022

**Objective**: studying the concepts of class, attribute and function, inheritance, composition, aggregation, association, dependency

**Task:** to make 3-4 class diagrams for the chosen information system

**Theoretical considerations:**

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.Class diagrams are the most popular UML diagrams used for the construction of software applications. It is very important to learn the drawing procedure of class diagram.

Class diagrams have a lot of properties to consider while drawing but here the diagram will be considered from a top level view. Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. A collection of class diagrams represent the whole system.

**The purpose of the class diagram can be summarized as**

* Analysis and design of the static view of an application.
* Describe the responsibilities of a system.
* Base for component and deployment diagrams.
* Forward and reverse engineering.

**Basic components of a class diagram**

The standard class diagram is composed of three sections:

* **Upper section:** Contains the name of the class. This section is always required, whether you are talking about the classifier or an object.
* **Middle section:** Contains the attributes of the class. Use this section to describe the qualities of the class. This is only required when describing a specific instance of a class.
* **Bottom section:** Includes class operations (methods). Displayed in list format, each operation takes up its own line. The operations describe how a class interacts with data.

**Where to Use Class Diagrams?**

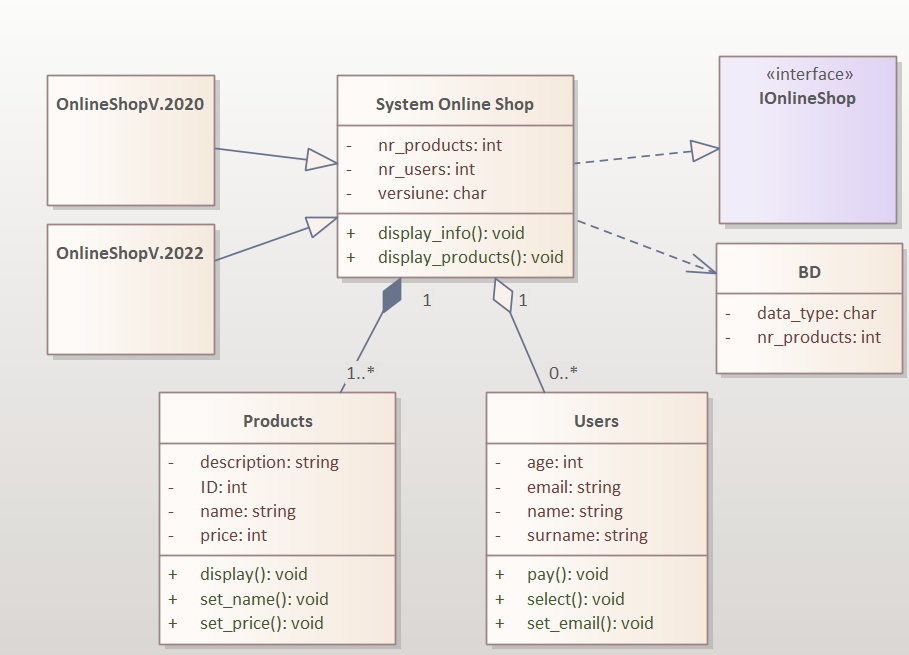
Class diagram is a static diagram and it is used to model the static view of a system. The static view describes the vocabulary of the system. Class diagram is also considered as the foundation for component and deployment diagrams. Class diagrams are not only used to visualize the static view of the system but they are also used to construct the executable code for forward and reverse engineering of any system. Generally, UML diagrams are not directly mapped with any object-oriented programming languages, but the class diagram is an exception.

All classes have different access levels depending on the access modifier (visibility). Here are the access levels with their corresponding symbols:

* Public (+)
* Private (-)
* Protected (#)
* Packages (~)
* derived (/)
* Static (underlined)

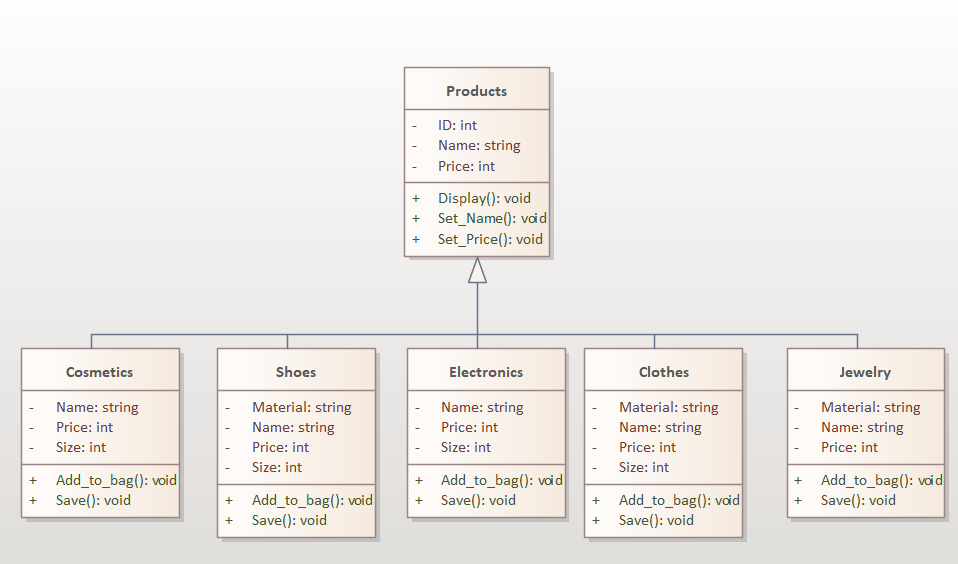
The chosen topic is **Analysis and modeling of an online store (e-commerce**). In recent times, the advancement of wireless technology and the growth of market potentials have led to an increase in the number of mobile device users. The emergence of this technology gave rise to the rapid development of mobile e-commerce technologies. It brings on-the-go Internet access to the general online market, without geographical and time constraints.

**Implementation, practical results:**



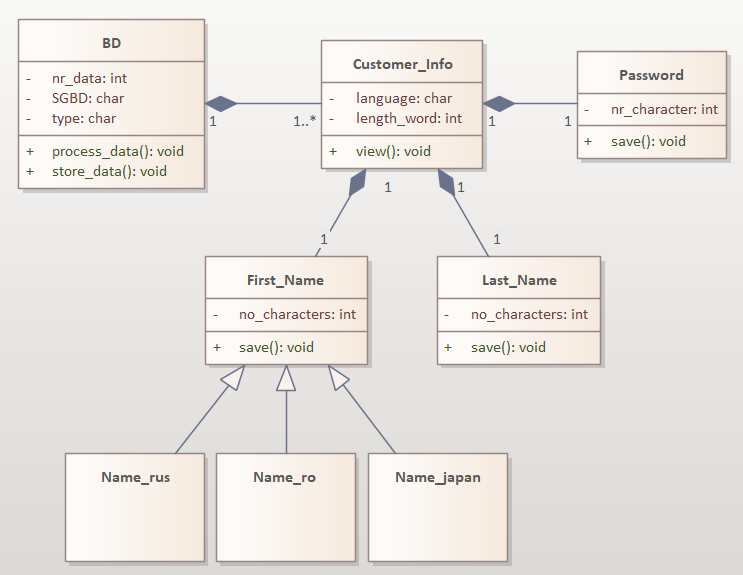
***Figure 1.*** *System Online Shop Diagram*

Class diagram number 1 shows next classes :OnlineShop Versions, SystemOnlineShop, BD, Products and the interface IonlineShop. The class System Online Shop is connected with interface through Realize relationship (Realize relationships is always used to connect interface with class). The class BD is connected with Class SystemOnlineShop through dependency relationship. Versions of Online Shop are connected with System Online Shop through Generalization Relationship. This Online Shop can exist without customers(users). And the last one, Products and System Online Shop are connected through Compose Relationship, because The online Shop can’t exist without Products. 1..\* means At least one object. 0..\* means No object or at least one.



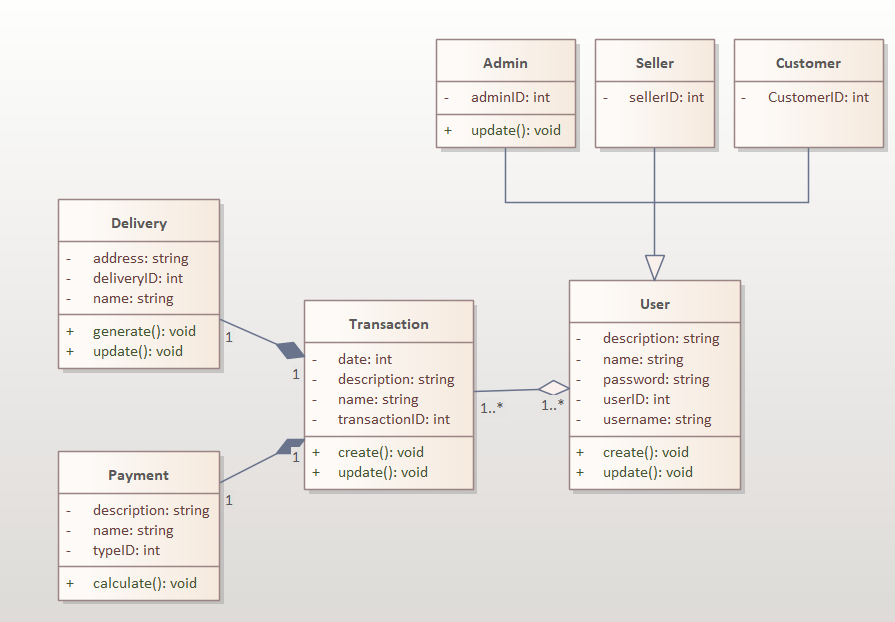
***Figure 2.*** *Products*

Figure 2 shows the class diagram Products. There are some categories of Products: Cosmetics, Shoes, electronics, Clothes and Jewelry. The are all conected with the class Products through Generalization relatioships and I used line style to make these kind of lines.



***Figure 3.*** *Customer Info Diagram*

Class diagram number 3 shows a summary of Customer\_Info. The Class Customer Info is connected with BD, Password, First\_Name, Last\_Name classes through Composition relatioship. And different types of names are connected with the class First\_Name with Realization Relationship.



***Figure 4.*** *User Diagram*

Figure 4 User Diagram shows us some different types of Users : Admin, Seller and Customer. Their attribuite is repesctivelly, AdminID:int, Seller:ID:int and CustomerID:int. User can make transactions, this is not obligatory. But if the User decides to make a transaction, Delivery and Payment are necessary, so they use composition relationships. Because they are mandatory within Transaction.

**Conclusions:** The purpose of this lab work is to understand and familiarize ourselves with Class diagrams, the connections between them and the actors. In this lab work I created some class diagrams. These diagrams allow us to demonstrate how the inner workings of our system work. Class diagrams are one of the most useful types of diagrams in UML as they clearly map out the structure of a particular system by modeling its classes, attributes, operations, and relationships between objects. With our UML diagramming software, creating these diagrams is not as overwhelming as it might appear.

**Bibliography**

1. **Melnic R., Sava N.** Indrumar metodic “Analiza si modelarea sistemelor informationale”.
2. **Моделирование бизнес процессов|CASE средства|Rational Rose**, [Электронный ресурс].-Режим доступа: <http://www.kpms.ru/Automatization/Rational_Rose.htm>