

# LABORATOR 1

## 1. Ce este UML și pentru ce este utilizat?

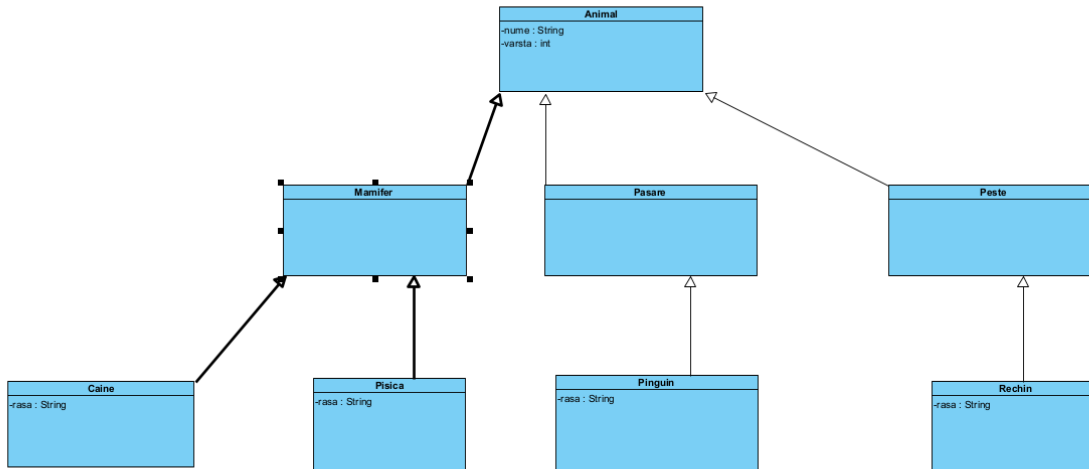
Uml, Unified Modeling Language (sau Limbajul de Modelare Unificat) este un limbaj de modelare al sistemelor software și al altor sisteme complexe. Acesta ne oferă un set de diagrame diferite pentru diferite aspecte ale unui sistem software.

## 2. Ce sunt modelele și care este utilitatea lor?

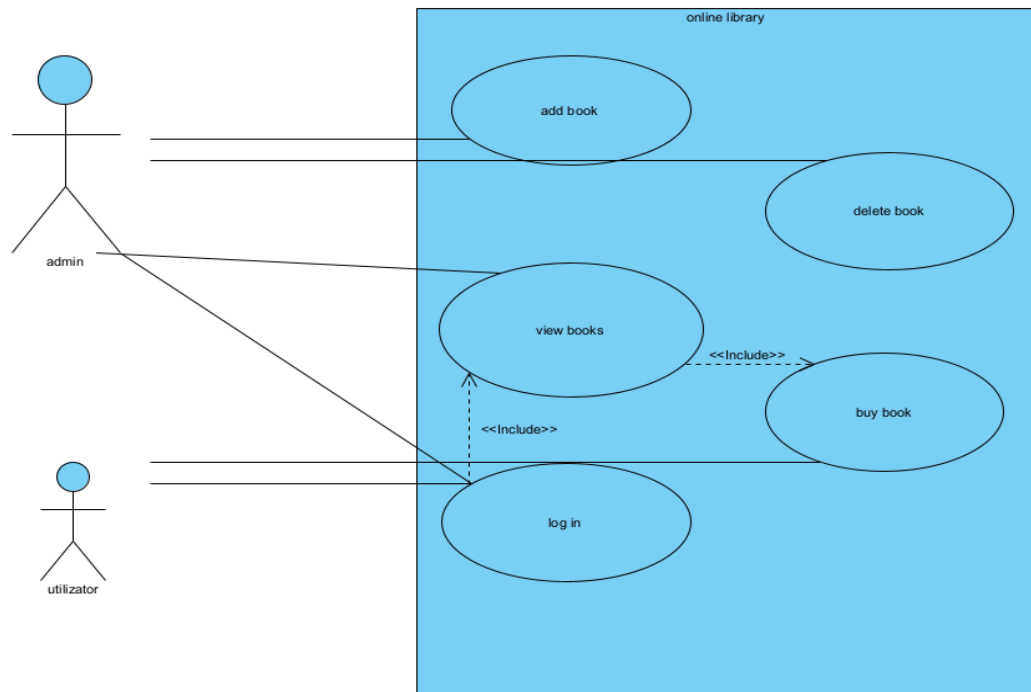
Modelele UML sunt reprezentări abstracte ale unui sistem. Acestea pot descrie structura sistemului, procesele acceptate sau interzise, interacțiuni dintre utilizatori și funcționalități.

## 3. Familiarizarea cu mediul de lucru mai sus amintit prin crearea anumitor diagrame din secțiunile 2.3.1 și 2.3.2.

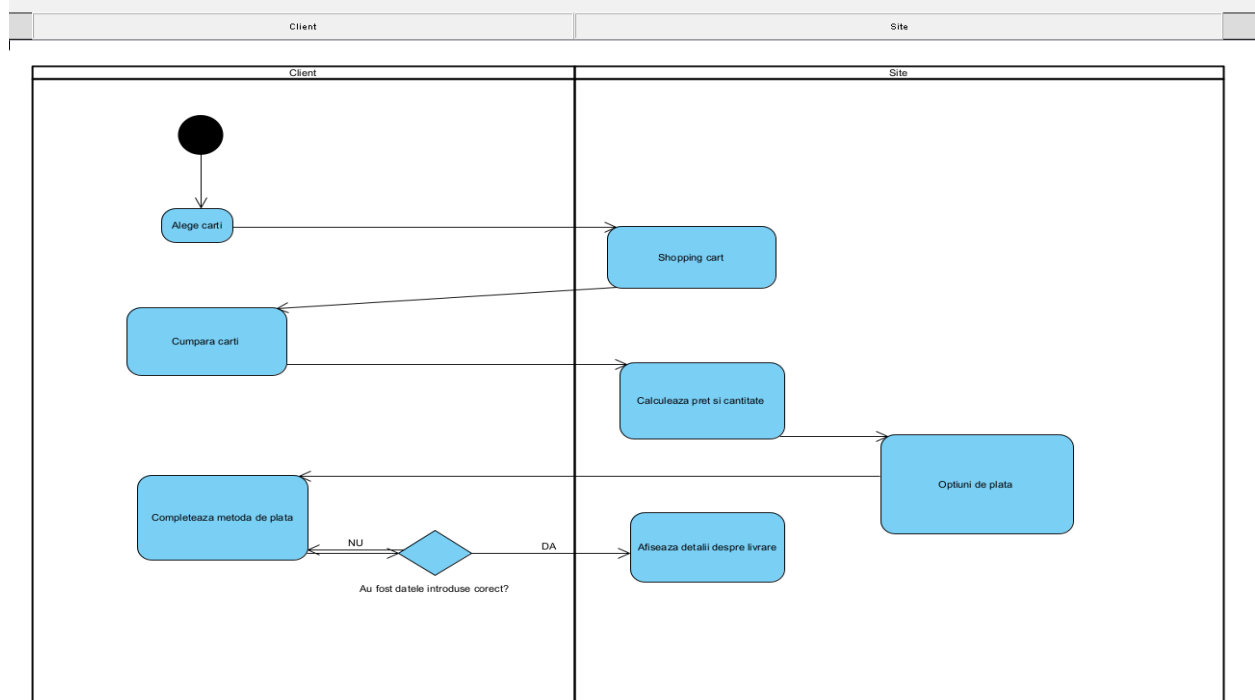
### -Class diagram



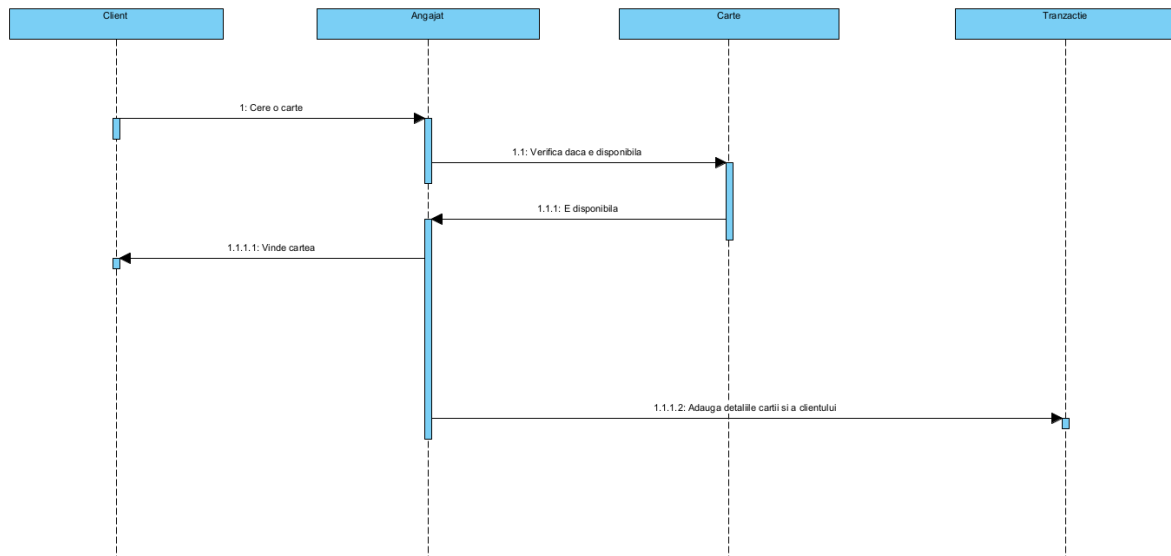
## -Use case diagram



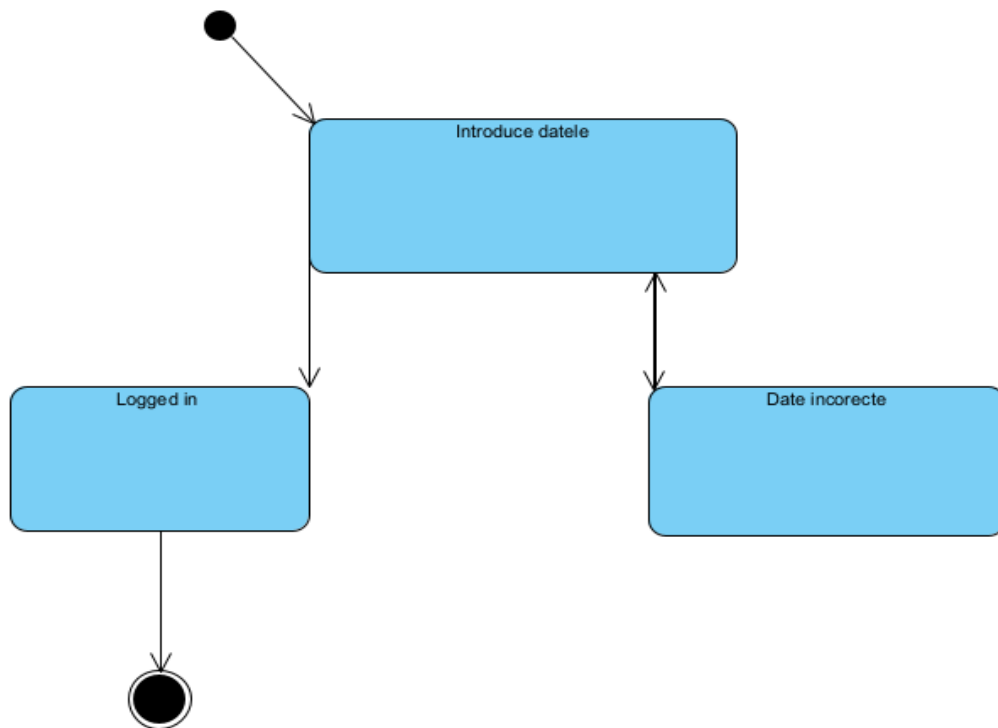
## -Activity diagram



## -Sequence diagram



## -State machine diagram



#### **4. Ce tipuri de diagrame sunt prezentate în [1] – Capitolul 2? Care sunt asemănările și deosebirile între acestea?**

Toate diagramele prezentate în capitolul 2 sunt : class diagram, object diagram, package diagram, component diagram, composition structure diagram, deployment diagram, profile diagram, use case diagram, state machine diagram, activity diagram, sequence diagram, communication diagram, timing diagram, interaction overview diagram.

Toate diagramele sunt folosite pentru a reprezenta structura și anumite acțiuni ale unui sistem.

Diferențele dintre ele sunt legate chiar de ce anume reprezintă. De exemplu, class diagram descrie structura sistemului, use case diagram ilustrează interacțiunile dintre sistem și actori, activity diagram descrie procesele și acțiunile din sistem, sequence diagram descrie interacțiunile dintre obiecte, iar state machine diagram modelează starea unui obiect sau a unui sistem.

#### **5. Caracterizați succint fiecare diagrama prezentată în [1] – Capitolul 2 (cele 5 menționate mai sus)**

**-class diagram:** oferă o imagine a stării sistemului la un timp specific.

Diagrama este bazată în principiu pe conceptele de clasă, generalizare și asociere.

**-use case diagram:** evidențiază care utilizatori pot folosi anumite funcționalități ale sistemului, dar nu prezintă detalii specifice ale implementării.

**-activity diagram:** cu ajutorul lor se pot modela procesele de orice fel: procese de business și procese software

**-sequence diagram:** descrie interacțiunile dintre obiecte pentru a îndeplini o sarcină anume. Se axează pe ordinea cronologică a mesajelor dintre partenerii ce interacționează.

**-state machine diagram:** obiectele trec prin diferite stări în timpul vieții lor. Aceste diagrame arată multiplele stări pe care un obiect le poate avea și cum trece de la o stare la alta.

## 6. Folosind Visual Paradigm, realizați diagramele de clasă din Figurile 1 și 2.

