

# *Extragerea cerințelor continuare*

*Prof. univ. dr. ing. Florica Moldoveanu*

*Curs Ingineria programelor – UPB, Automatică și Calculatoare*  
2020-2021

# Activități de extragere a cerintelor

- din cursul anterior-

1. **Identificarea actorilor:** entitățile externe sistemului care vor interacționa cu sistemul. Pentru sistemul considerat, SGCB (Sistemul de gestiune a cartilor din mai multe biblioteci), au fost identificați 2 actori: **bibliotecarul și abonatul**.
2. **Identificarea și descrierea scenariilor de utilizare:** se definesc scenarii pentru functionalitățile tipice care vor fi furnizate de viitorul sistem.
3. **Definirea cazurilor de utilizare ale sistemului:** plecând de la scenarii, dezvoltatorii definesc un set de cazuri de utilizare care descriu toate posibilitățile de utilizare a viitorului sistem.
4. **Rafinarea cazurilor de utilizare**
5. **Identificarea și reprezentarea relațiilor dintre cazurile de utilizare**
6. **Identificarea și definirea cerintelor nefunctionale**

# Cazuri de utilizare

- din cursul anterior-

**Descrierea tipică a unui caz de utilizare cuprinde următoarele elemente:**

- **Descrierea comportamentului de bază al sistemului** (fluxul principal de evenimente și operațiile declanșate) **și alternativele** – o descriere pas cu pas a acțiunilor actorului și sistemului.
- **Pre-condiția (opțional)** – o constrângere asupra sistemului la inițierea cazului de utilizare; se specifică în limbaj natural.
- **Post-condiția (opțional)** – o constrângere asupra sistemului la terminarea execuției cazului de utilizare; se specifică în limbaj natural.
- **Cerințe speciale** – cerințe ce nu pot fi descrise cu ușurință în fluxul de evenimente.
- **O schiță a interfeței utilizator** la executia cazului de utilizare **(opțional)** .

*In faza de analiza a cerintelor, cazurile de utilizare se rafinează si formalizează folosind diagrame de secvență UML pentru descrierea scenariilor și diagrame de stari UML pentru descrierea comportamentului in timp al entitatilor din domeniul aplicatiei.*

# Descrierea unui caz de utilizare

- din cursul anterior-

Un caz de utilizare descrie o functionalitate a sistemului în raport cu un actor; este o abstractizare a tuturor scenariilor care descriu acea functionalitate.

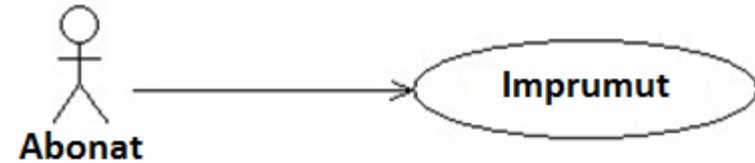
Un caz de utilizare este descris prin secventa tipica de pași (comportamentul de bază) și alternativele la secvența tipică.

**Exemplu: Cazul de utilizare "Împrumut" al SGCB**

**declanșat de Abonat**

**Fluxul de baza (secvența tipică de pași)**

1. Un utilizator accesează interfața web a sistemului în secțiunea pentru împrumut cărți și completează rubricile rezervate numelui de utilizator și parolei de acces, apoi apasă butonul "Submit".
2. Sistemul preia datele și verifică identitatea utilizatorului.
3. Sistemul afișează formularul de împrumut.
4. Abonatul completează formularul de împrumut, cu titlul cărții, numele și prenumele autorului și codul ISBN al cărții apoi apasă butonul "Submit".
5. Sistemul preia datele și caută cartea.
6. Sistemul înregistrează împrumutul.
7. Sistemul afișează mesajul "Luati cartea de la ghiseu".



# Descrierea unui caz de utilizare

- din cursul anterior-

## Alternative:

La pasul 3:

3a) Utilizatorul nu este înregistrat ca abonat și atunci sesiunea este încheiată de sistem cu mesajul: „Nume de utilizator inexistent. Nu sunteți înregistrat ca abonat. Efectuați procedura de înregistrare”.

3b) Utilizatorul este înregistrat dar a depășit numărul maxim admis de cărți împrumutate. Sesiunea este încheiată de sistem cu mesajul: „Ați depășit numărul maxim de cărți împrumutate. Restituiți o parte dintre ele”.

La pasul 6:

6a) Cartea nu este găsită. Sistemul afișează mesajul: „Cartea nu există în bibliotecile noastre”.

La fluxul de bază se pot adăuga o *precondiție* și o *postcondiție*.

# Rafinarea cazurilor de utilizare

În acest pas se adaugă detalii la cazurile de utilizare, care descriu comportarea sistemului în prezența defectelor și a condițiilor excepționale.

De exemplu, la cazul de utilizare **"Împrumut"** se pot adauga cazurile de excepție:

## Excepții

La pasul 2: Sesiunea este încheiata de sistem cu mesajul "Eroare de acces la lista de utilizatori".

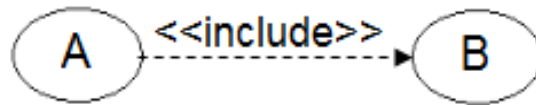
La pasul 6: Sesiunea este încheiata de sistem cu mesajul "Eroare de acces la lista de abonati".

# Relații între cazurile de utilizare - (1)

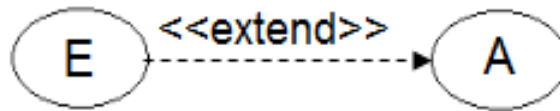
## - reprezentare în UML -

Între cazurile de utilizare se pot stabili relații de:

- **Includere** – un caz de utilizare A include, în unul sau mai mulți pași, comportamentul descris într-un alt caz de utilizare, B.

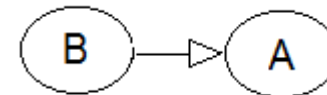


- **Extindere** – un caz de utilizare E extinde un alt caz de utilizare, A, dacă A include comportamentul descris în E, numai în anumite condiții.



- **Generalizare/specializare:** similară conceptual cu relația de generalizare dintre clase

B este o specializare a cazului de utilizare A:



În UML, reprezentarea unui set de cazuri de utilizare și a relațiilor dintre ele se numește **diagramă de cazuri de utilizare**.

# Relatii între cazurile de utilizare (2)

## Relatia de includere - exemplificare

Orice utilizator al SGCB ar trebui ca, înainte de orice operatie cu sistemul, să se autentifice:

- operatia de autentificare trebuie sa fie inclusa in orice caz de utilizare al sistemului.

Autentificarea consta dintr-o secventa de pasi, care:

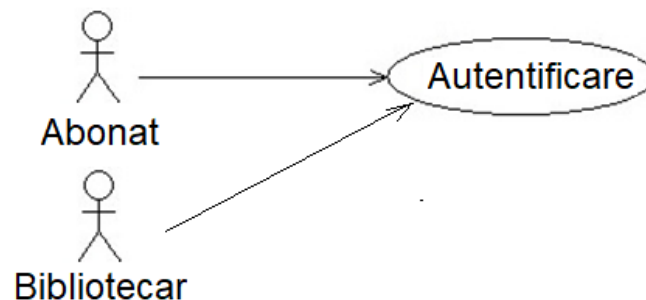
- poate fi inclusa in fiecare caz de utilizare, sau
- descrisa intr-un caz de utilizare separat care va fi inclus in toate celelalte cazuri de utilizare

## Cazul de utilizare “Autentificare”

### Fluxul de baza

1. Un utilizator completează rubricile rezervate numelui de utilizator și parolei de acces în interfața Web a sistemului, apoi apasa butonul “Submit”.
2. Sistemul preia datele și verifică identitatea utilizatorului.
3. Numele de utilizator si parola sunt corecte si utilizatorul este autentificat in sistem.

*Postconditie:* utilizatorul este autentificat in sistem



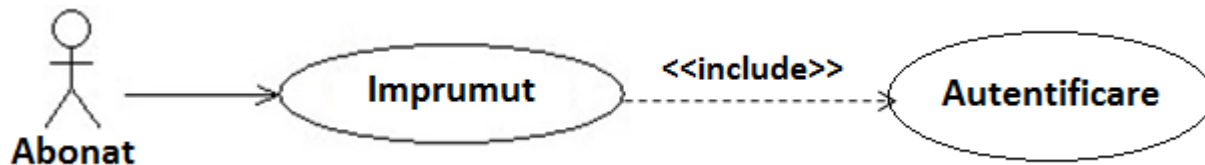


# Relatii între cazurile de utilizare (3)

## Alternative:

La pasul 3: Sistemul afiseaza mesajul: „Nume de utilizator inexistent sau combinatia nume\_utilizator-parola este incorecta” și încheie sesiunea.

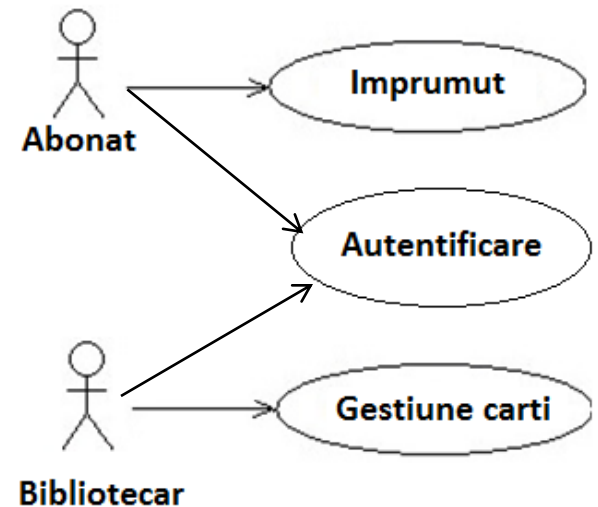
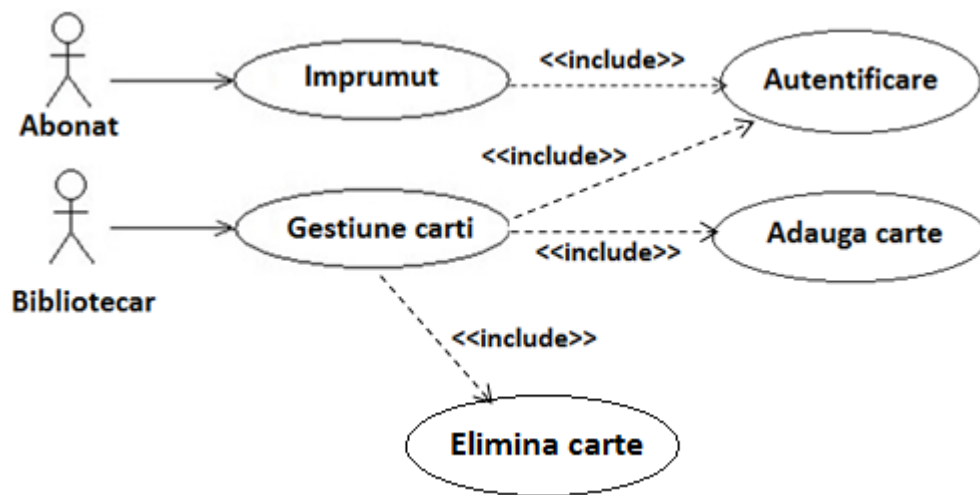
Cazul de utilizare “autentificare” va fi inclus in cazul de utilizare “Imprumut” si în toate celelalte cazuri de utilizare:



## Cazul de utilizare “Imprumut”

1. Utilizatorul executa cazul de utilizare “Autentificare”.
2. Daca utilizatorul nu este autentificat, atunci cazul de utilizare se incheie.
3. Sistemul afișează formularul de împrumut.
4. ....

# Relatii între cazurile de utilizare (4)



O alta posibilitate de descriere a cazului de utilizare “Imprumut”, si a celorlalte, care corespunde diagramei din partea dreaptă, este:

## Cazul de utilizare “Imprumut”

*Preconditie:* utilizatorul este autentificat în sistem

1. Sistemul afișează formularul de împrumut.
- 2.....

Aceasta presupune că înainte de orice utilizare a sistemului, utilizatorul trebuie sa execute cazul de utilizare **Autentificare** terminat prin autentificarea sa în sistem.

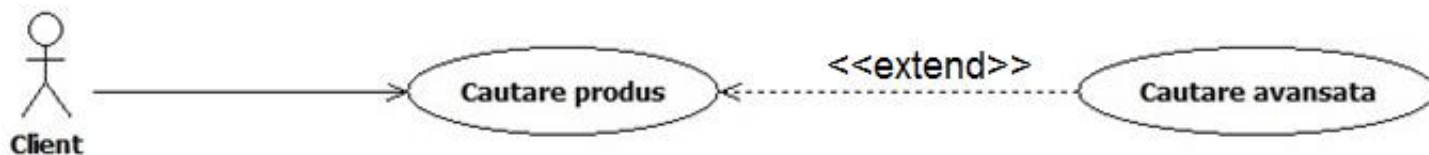
# Relatii între cazurile de utilizare (5)

## Relatia de extindere - exemplificare

Utilizatorul unui site dorește să găsească un anumit produs. Dacă funcționalitatea de căutare nu îi este suficientă pentru găsirea produsului dorit, îi este oferită opțiunea de căutare avansată.

Operația de cautare avansata poate fi descrisa printr-un caz de utilizare separat,

**“Căutare avansată”,** care **extinde cazul de utilizare “Cautare produs” atunci cand cautarea normala nu are rezultat**



**Un posibil flux de evenimente pentru cazul descris este:**

1. Utilizatorul introduce un criteriu de căutare.
2. Site-ul întoarce mesajul “nici un rezultat” și utilizatorului îi este oferită opțiunea de căutare avansată.
3. **Daca** utilizatorul selectează opțiunea de căutare avansată **atunci** se continua cu pasul 4 **altfel** se continua cu pasul 6
4. Se executa cazul de utilizare “Cautare avansata”
5. Site-ul afișează rezultatele căutării avansate.

# Relatii între cazurile de utilizare (6)

**Euristici privind relațiile de includere și extindere:**

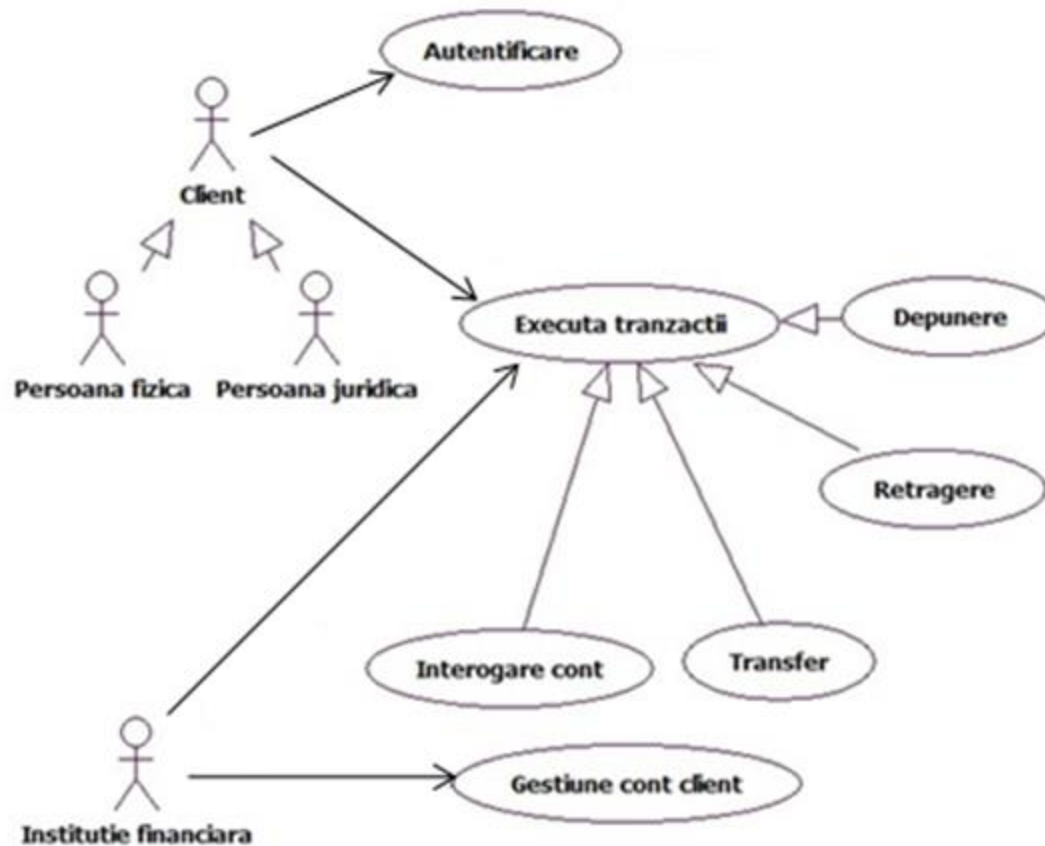
- **Relatia de extindere** se utilizează **pentru a evidenția** comportamente ce corespund unor cazuri optionale, exceptionale, sau care se produc rar.
- **Relatia de includere** se utilizează atunci când **un comportament este comun mai multor cazuri de utilizare sau apare în mai mulți pași ai unui caz de utilizare.**
- Ambele relații trebuie utilizate cu discreție deoarece prea multe relații între cazurile de utilizare pot îngreuna înțelegerea acestora.

## **Generalizarea**

- Atât între actori, cât și între cazuri de utilizare, se pot stabili relații de *generalizare*, care au aceeași semnificație cu relația de generalizare dintre clase: mai mulți actori/cazuri de utilizare, având caracteristici comune, sunt specializări ale unui actor/caz de utilizare mai general.

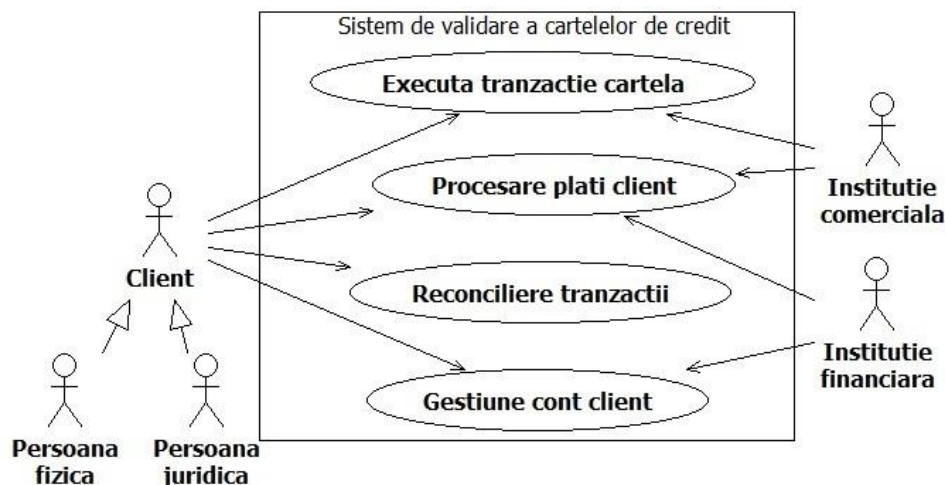
# Relatii între cazurile de utilizare (7)

## Relatia de generalizare



# Diagrama de context

- Este o diagrama care cuprinde toate cazurile de utilizare ale unui sistem și actorii.
- Are rolul de a delimita sistemul de mediul său de operare.



# Modelul cazurilor de utilizare (Use case model)

## Cuprinde:

- Descrierea actorilor
- Descrieri ale principalelor scenarii de utilizare
- Descrierile tuturor cazurilor de utilizare
- Diagrame de cazuri de utilizare
- Diagrama de context

## Se folosește pentru:

- Specificarea cerințelor funcționale
- Validarea arhitecturii sistemului
- Generarea cazurilor de test

# Cerinte nefunctionale (1)

## Identificarea cerintelor nefunctionale

Cerintele nefunctionale sunt cerinte care nu sunt asociate cu un caz de utilizare specific.

Pot fi:

- **Constrangeri:**
  - De comunicare hardware-software, compatibilitate cu alte sisteme software, interfata utilizator, constrangeri de operare, legislative.
- **Cerinte de calitate a produsului:** fiabilitate, portabilitate, adaptabilitate, performanță, disponibilitate, securitate, siguranta in functionare, standarde.
- **Cerinte de planificare a proiectului:** termene, produse livrate, resurse necesare, cerinte de verificare a produsului final, cerinte de asigurare a calitatii produsului.



# Cerinte nefunctionale (2)

## Cerinte de performanta

- **Valori numerice atasate unor parametri masurabili** cum ar fi: viteza, capacitatea, precizia, frecventa.

Exemplu: sistemul masoara temperatura cu o precizie de 1 grad Celsius.

- Pot fi reprezentate ca un domeniu de valori sau valori individuale:

valoare acceptabila: [min-max]; valoare ideala.

Ex. timpul de raspuns:  $< 1$  sec

## Cerinte de interfata

- Interfetele noului produs cu diferite componente hardware/software
- Interfete interne / externe ale produsului software
- Se definesc prin:
  - Protocoalele de comunicatie de utilizat
  - Fluxul de date prin interfata
  - Evenimentele care declanseaza schimburi de date prin interfata, etc

# Cerinte nefunctionale (3)

## Cerinte de operare

- Modul de comunicare a sistemului cu operatorii umani, aspecte fizice si ergonomice ale interfetei utilizator:
  - Dialogul operator-sistem
  - Aspectul ecranului in timpul dialogului
  - Stilul limbajului de comenzi
  - Usurinta de învățare a modului de operare cu sistemul (perioada invatarii): stilul interfetei, scheme de culoare, help online, documentatia de utilizare.

## Cerinte impuse resurselor fizice

- Puterea de prelucrare
- Memoria necesara
- Spatiul pe disc

# Cerinte nefunctionale (4)

## Cerinte de intretinere

- Usurinta de reparare a defectelor, de îmbunatatire sau adaptare la schimbarea cerintelor.  
Exemplu: "Timpul de reparare a unui defect nu va depasi niciodata o saptamana“.

## Cerinte de fiabilitate

- Frecventa acceptata a caderilor software (cadere: o comportare a sistemului neconforma cu specificatia); poate fi specificata pe categorii/tipuri de caderi:
- Exprimate folosind metrici ca Mean Time Between Failures (MTBF) si Mean Time To Repair (MTTR).
- Exemplu: "Timpul minim intre doua caderi severe va fi mai mare de 3 luni“.

## Cerinte de portabilitate

- "Sa poata rula pe calculatorul X/sub sistemul de operare Y fara modificarea codului sau modificand cel mult 2% din codul sursa“.
- "Nici o parte a software-ului nu trebuie scrisa in assembler“.

# Cerinte nefunctionale (5)

## Cerinte de securitate

- Cum sa fie securizat sistemul împotriva pericolelor:
  - Erori utilizator (distrugerea accidentala a software-ului sau datelor)
  - Hazarduri fizice (foc)
  - Access ne-autorizat
  - Virusi, securitatea comunicarii in retea

## Cerinte de siguranta in functionare (Safety)

- Protectia împotriva distrugerilor cauzate de caderile software. De exemplu, ce trebuie sa se întâmple în cazul producerii unei caderi software:
  - Degradare treptata
  - Continuare dintr-un anumit punct
  - Sa fie implementate tehnici de izolare a caderilor

# Cerinte nefunctionale (6)

## Cerinte de verificare a produsului final

- Cerinte impuse mediului de testare
- Posibilitati de diagnosticare
- Cerinte pentru testarea de acceptare
- Efectuarea unor simulari (atunci cand sistemul nu poate fi testat in mediul operational inainte de testarea de acceptare)

## Cerinte de asigurare a calitatii

- Utilizarea anumitor standarde de produs sau de proces
- Utilizarea de personal extern pentru asigurarea calitatii

## Cerinte legislative

- Conditii de certificare, de licentiere, etc

# Documentarea cerintelor utilizator(1)

Cerintele extrase sunt de regula definite într-un **document general de Specificare a cerintelor software**, care include atât cerințele utilizator cât și rezultatul analizei cerintelor.

**In cadrul documentului sunt definite:**

- 1. Obiectivele generale ale sistemului**, granitele si constrangerile generale.
- 2. Domeniul (mediul) de operare al viitorului sistem, principalele entitati ale domeniului, fluxul informational.**
- 3. Cerintele de sistem:** configuratia hardware, siguranta in functionare, instalarea, punerea in functiune, comunicarea cu sisteme externe, s.a.

# Documentarea cerintelor utilizator(2)

4. **Actorii:** rolurile entitatilor externe care interactioneaza direct cu sistemul (om sau alta entitate externa).

5. **Cerintele functionale ale sistemului, specificate folosind:**

- **limbajul natural;** exemple:

- *masoara temperatura si o afiseaza in grade Celsius*
- *cere codul PIN, il verifica si afiseaza un mesaj de acceptare sau rejectare*

- **principalele scenarii de utilizare** a sistemului de catre diferiti actori.

- **schite ale interfeței utilizator** pentru diferite scenarii de utilizare

- **cazurile de utilizare ale sistemului și diagrame de cazuri de utilizare**

6. **Cerintele nefunctionale**

# Documentul cerintelor software (Software Requirements Document)

## Continutul general al documentului (finalizat dupa analiza cerintelor)

### 1. Introducere

- 1.1. Scopul documentului
- 1.2. Domeniul aplicatiei (in care se incadreaza/va functiona noul produs)
- 1.3. Scopul noului produs
- 1.4. Obiectivele și criteriile de succes ale proiectului. Produse similare.
- 1.5. Referințe (la sisteme existente, studii de fezabilitate, etc.) – daca este cazul
- 1.6. Lista de definitii si abrevieri

### 2. Sistemul curent (daca sistemul propus va inlocui un sistem)

### 3. Sistemul propus (noul produs)

- 3.1. Descriere generala
- 3.2. Descrierea categoriilor de utilizatori directi/indirecti ai sistemului
- 3.3. Mediul de operare – daca este cazul
- 3.4. Cerinte de sistem (echipamente de calcul, dispozitive, comunicatia in retea)
- 3.5. Cerinte functionale (in limbaj natural)
- 3.6. Cerinte nefunctionale
- 3.7. Modele ale sistemului
  - 3.7.1. Actorii si cazurile de utilizare prin care interactioneaza
  - 3.7.2. Descrierea principalelor scenarii de utilizare
  - 3.7.3. Descrierea cazurilor de utilizare ale sistemului (interfata utilizator la executia CU)
  - 3.7.4. Diagrama de context
  - 3.7.5. Modelul dinamic – definit in etapa de analiza a cerintelor
  - 3.7.6. Modelul obiect – definit in etapa de analiza a cerintelor