

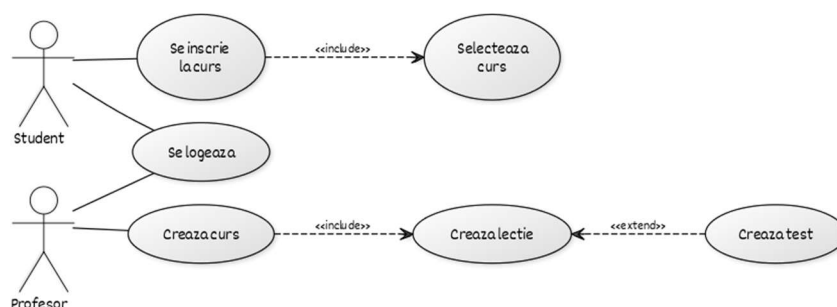
LL#1 Use Case Diagram. Use Case Scenario

1. Alegeți un sistem informatic (SI) existent.
2. Faceți o descriere succintă a SI ales (denumirea, adresa URL, domeniul de aplicație, scopul, utilizatori, funcționalități de bază).
3. Alegeti si descrieti un set de reguli / cerinte din domeniul dat, pe care for fi bazate urmatoarele artefacte.
4. Elaborați diagrama cazurilor de utilizare (Use Case Diagram - UCD).
5. Descrieți cateva cazuri de utilizare: identificator unic, denumirea (ca în diagramă), descrierea succintă; identificați actori; descrieți pre-condiții și post-condiții; elaborați scenarii de utilizare (un scenariu de baza, scenarii alternative și excepționale la necesitate). Adaugați referințe la reguli sau cerinte corespunzatoare.
6. Observați similarități între scenarii de utilizare și scenarii de testare.
7. Întocmiți darea de seamă cu sarcina, artefacte elaborate, concluzii.

Diagrama cazurilor de utilizare

Diagrama cazurilor de utilizare (Use Case Diagram) - este o reprezentare grafică a funcționalităților sistemului, semnificative pentru utilizatorii lui .

Pe aceasta diagrama sunt reprezentate grafic *cazurile de utilizare* si *actorii* (Figura 1).



CREATED WITH YUML

Figura 1: Diagrama cazurilor de utilizare a unui sistem de instruire la distanță*

Actor reprezintă o categorie de utilizatori (un rol care joacă un grup de utilizatori în sistem). Se reprezintă grafic ca un omuleț. Ex.: Student, Profesor, etc. (Figura 1)

Caz de utilizare este o funcționalitate a sistemului informatic din punct de vedere al utilizatorului. Practic el descrie ce poate face un utilizator în cadrul sistemului dat (Ex.: să se înscrie la curs, să creeze o lecție, să caute curs, etc). Grafic se reprezintă ca o elipsă cu denumire, care se începe cu verb (Figura 1). Fiecare caz de utilizare va fi completat cu o descriere, care include un scenariu de bază și câteva scenarii alternative și excepționale.

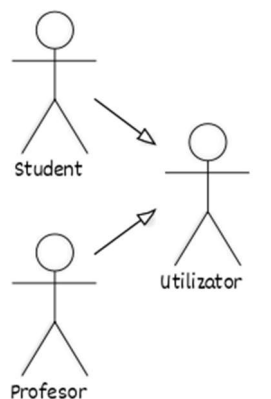
De asemenea pe diagrama sunt reprezentate *relatii*:

Asociere - relație dintre actor și caz de utilizare. Ea înseamnă, că actorul dat efectuează pașii scenariului cazului de utilizare respectiv. Este reprezentată ca o linie neîntreruptă fără săgeți, care leagă actorul și cazul de utilizare (Figura 1).

Includere - relație între cazuri de utilizare, care înseamnă că un caz de utilizare este o parte obligatorie a scenariului altui caz de utilizare. Se desenează ca o linie întreruptă cu săgeată deschisă, orientată spre cazul de utilizare inclus și decorată cu stereotipul <<include>>. De exemplu, pentru a se înscrie la curs Studentul **obligatoriu trebuie** să selecteze curs pentru înscriere (Figura 1)

Extindere - relatie intre cazuri de utilizare, care inseamna ca un caz de utilizare este o parte optionala a scenariului altui caz de utilizare. Se deseneaza ca o linie intrerupta cu sageata deschisa, orientata spre cazul de utilizare extins si decorata cu stereotipul <<extend>>. De exemplu, creand o lectie, Profesorul **optional poate** adauga in aceasta lectie un test (Figura 1)

Generalizare - relatie dintre un element specific si un element generalizat. Elementul specific are toate proprietatile celui generalizat plus unele specifice. Relatia de generalizare dintre elemente A si B are loc atunci, cand putem spune, ca "B este A" sau ca "B este un tip de A". In acest caz, A este element generalizat, iar B - cel specific. Se reprezinta cu ajutorul unei linii neintrerupte cu in triunghi gol la capat, orientat spre element mai general. De exemplu, Student este Utilizator, Profesor este Utilizator (Figura 2)



CREATED WITH YUML

Figura 1: Relatia de generalizare intre actori**

Diagrama cazurilor de utilizare impreuna cu specificatiile ale cazurilor de utilizare pot fi utilizate drept sursa pentru proiectarea, realizarea si testarea sistemului

* Codul yUML corespunzator diagramei: [Student]-(Se logeaza), [Student]-(Se inscrie la curs), [Profesor]-(Se logeaza), [Profesor]-(Creaza curs), (Creaza curs)>(Creaza lectie), (Creaza lectie)<(Creaza test), (Se inscrie la curs)>(Selecteaza curs)

** Codul yUML corespunzator diagramei: [Student]^[Utilizator], [Profesor]^[Utilizator]

Caz de utilizare

Caz de utilizare (Use case) reprezintă o interacțiune a utilizatorului cu sistemul cu un anumit scop semnificativ pentru utilizator.

Caz de utilizare este o succesiune de acțiuni care produc un rezultat măsurabil pentru un utilizator.

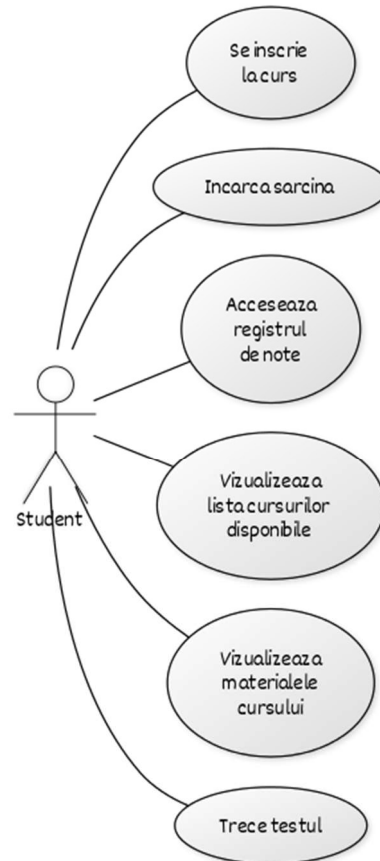
Pentru a identifica cazurile de utilizare trebuie să punem următoarele întrebări:

- Ce trebuie să facă un actor in sistemul dat?
- Care sunt sarcinile principale ale actorului?
- Ce actorul dorește să vadă, să creeze, să schimbe?
- Despre ce actorul dorește să fie informat din partea sistemului?
- Despre ce actorul trebuie să informeze sistemul?

De exemplu, Student, ca un utilizator al sistemului informatic de studii la distanță are nevoie:

- Să vadă lista cursurilor disponibile
- Să se înscrie la un curs

- Să cunoască informații de bază despre curs, cum ar fi descrierea și cerințele
- Să vizualizeze materialele cursului
- Să treacă testul și să obțină nota
- Să încarce lucrul individual și să obțină nota
- Să cunoască nota medie pentru curs



CREATED WITH YUML

Figura 3. Cazuri de utilizare pentru actorul Student*

Cazuri de utilizare au reprezentare grafică, dar pot avea și o descriere textuală, care conține printre altele și așa numite scenarii. Un **scenariu al cazului de utilizare** descrie pas cu pas procesul de interacțiune a utilizatorului cu sistemul.

Fiecare caz de utilizare are un scenariu de baza și poate avea mai multe scenarii alternative și excepționale.

Scenariu de baza descrie flux de evenimente normal, care permite atingerea scopului cazului de utilizare (actorul procedează în modul corect, recomandat).

Scenariu alternativ descrie o deviere de la scenariu normal, care totuși este acceptabil, legal, și permite atingerea scopului.

Scenariu excepțional descrie situații, care nu permit atingerea scopului, posibil din cauza acțiunilor nerecomandate actorului (introducerea datelor incorecte, utilizator neautorizat, etc).

* Codul yUML corespunzător diagramei: [Student]-(Se înscrie la curs),[Student]-(Încarcă sarcină),[Student]-(Accesează registrul de note),[Student]-(Vizualizează lista cursurilor disponibile),[Student]-(Vizualizează materialele cursului),[Student]-(Trece testul)

Descrierea cazului de utilizare

Descrierea fiecărui caz de utilizare conține de obicei următoarele elemente:

- **Denumirea** – un verb, sau o combinație de cuvinte care se începe cu verb, care comunică clar scopul acestui caz de utilizare
- **Identificatorul** - un identificator unic, care permite referirea la acest caz de utilizare din alte documente. Un exemplu de identificator: UC002
- **Descrierea succintă** – o propoziție/paragraf care descrie explicit scopul acestui caz de utilizare.
- **Actori** – lista de actori (rolurile utilizatorilor), care pot participa în această interacțiune cu sistem
- **Precondiții** – toate condițiile, care trebuie să fie îndeplinite, când se începe cazul de utilizare
- **Postcondiții** – orice condiție, care trebuie să fie îndeplinită când se termină ultimul pas al scenariului.
- **Un scenariu de bază (Basic Flow)** – un set de pași de bază, care execută actorii pentru a atinge scopul cazului de utilizare. Se recomandă descrierea clară a răspunsului din partea sistemului pentru fiecare acțiune a actorului
- **Scenarii alternative (Alternate Flows)** – descriu pașii, executate mai rar, dar totuși admisibile. Un scenariu alternativ permite atingerea scopului prin alte căi, decât calea de bază.
- **Scenarii excepționale (Exception Flows)** – pașii, care au loc în situații excepționale, când utilizatorul încearcă acțiuni inadmisibile, de exemplu introducerea datelor incorecte. Aceste scenarii nu permit atingerea scopului (postcondițiile nu vor fi îndeplinite)

Exemplu de descriere a unui caz de utilizare

Denumirea cazului de utilizare: "Se înscrie la curs"

Identificator: UC011

Descrierea: Înscrierea studentului la cursul universitar disponibil (pentru care studentul este eligibil).

Precondiții: Studentul este înregistrat în sistemul universitar de studii online (*UC001: Se înregistrează*) și este logat în sistem (*UC002: Se logează*).

Postcondiții: Studentul va fi înscris la cursul dorit, pentru care este eligibil și va avea acces la materialele cursului.

Scenariu de bază:

1. Cazul de utilizare se începe când utilizatorul dorește să se înscrie la curs
2. Studentul selectează opțiunea Cursuri universitare din meniul principal în fereastra *UI123 Fereastra principală*.
3. Sistemul afișează fereastra de dialog *UI032 Selectează curs*, care conține lista cursurilor disponibile.
4. Studentul selectează cursul la care dorește să se înscrie [**Scenariu alternativ A: Studentul decide să nu se înscrie la curs**].
5. Sistemul verifică eligibilitatea studentului pentru înscriere la curs conform regulii de business *BR111 Studentul nu poate fi înscris la curs dacă nu a plătit contractul* [**Scenariu excepțional B: Studentul nu a plătit contractul de studii**]

6. Sistemul verifică eligibilitatea studentului pentru înscriere la curs conform regulii de business BR112 *Studentul nu poate fi înscris la curs dacă nu a trecut cu nota pozitivă toate prerechizite* **[Scenariu excepțional C: Studentul nu a trecut toate cursuri prerechizite]**
7. Sistemul afișează informații generale despre cursul în fereastra UI033 *Înscriere curs*.
8. Studentul apasă butonul "Înscriere".
9. Sistemul înscrie studentul la curs și informează ca înscrierea a avut succes în fereastra UI088 *Înscriere la curs cu succes*.
10. Cazul de utilizare se termină, când studentul apasă butonul „OK”

Scenariu alternativ A: Studentul decide să nu se înscrie la curs

1. Studentul apasă pe butonul „Cancel”
2. Sistemul afișează fereastra principală UI123 *Fereastra principală*
3. Scenariul continue cu pasul 2 din scenariu de bază

Scenariu excepțional B: Studentul nu a plătit contractul de studii

1. Sistemul determină că studentul nu a plătit contractul de studii.
2. Sistemul afișează mesajul „Nu aveți acces la acest curs deoarece nu ați plătit contractul de studii”.
3. Cazul de utilizare se termină, când studentul apasă butonul „OK”

Scenariu excepțional C: Studentul nu a trecut toate cursuri prerechizite

1. Sistemul determină că studentul nu a trecut unul sau mai multe cursuri necesare.
2. Sistemul afișează mesajul „Pentru a se înscrie la acest curs trebuie să studiați următoarele cursuri-prerechizite: <lista cursurilor necesare>”.
3. Cazul de utilizare se termină, când studentul apasă butonul „OK”

LL#2 State Transition Diagram

1. Faceți o descriere succintă a SI ales (denumirea, adresa URL, domeniul de aplicație, scopul, utilizatori, funcționalități de bază).
2. Alegeti si descrieti un set de reguli/cerinte din domeniul dat.
3. Elaborați diagrama de stări (State Transition Diagram - STD), care descrie stările, în care poate afla sistemul/subsistemul/obiectul, condiții și acțiuni care aduc la schimbarea acestor stări, conform regulilor descrise mai sus.
4. Identificati scenarii de testare in baza diagramei elaborate.
5. Întocmiți darea de seamă cu sarcina, toate artefacte elaborate, concluzii

Diagrama de stari

Diagrama de stări descrie comportamentul unui element de model (obiect, subsistem etc.). Ea reprezintă un automat (*state machine*), care definește o succesiune de stări, prin care trece acest element de-a lungul vieții, răspunzând la diferite evenimente, și reacțiile lui la aceste evenimente.

Elemente principale ale diagramei de stări sunt: *stări*, *evenimente*, *tranziții* și *acțiuni*.

O stare (*state*) reprezintă o situație din viața unui obiect în timpul căreia: se satisfac anumite condiții, se efectuează o activitate sau se așteaptă apariția unui eveniment. Exemplu de stări de obiecte: obiectul *bec* poate avea stările: *aprins* și *stins* (Figura 1); obiectul *factură* poate avea stările: *plătit* și *neplătit*;

Un eveniment (*event*) este ceva ce se produce asupra unui element. Un eveniment poate fi:

- 1) Receptionarea unui semnal, cum ar fi o excepție, o notificare, un eveniment generat de interacțiunea cu utilizatorul .
- 2) Receptionarea unui apel, adica invocarea unei operatii a clasei obiectului.
- 3) Recunoasterea unei conditii in mediul extern sau in obiectul insusi (intern):
 - condiție predefinită, care este îndeplinită la un moment dat, eveniment numit "*change event*" ("*condiție*").
 - trecerea unei perioade de timp desemnate, eveniment numit "*elapsed-time event*"("*dupa o perioada de timp*").

Apariția unui eveniment duce la schimbarea stării - trecerea de la o stare (starea-sursă) spre alta (starea țintă), care se numește tranziție (*transition*). Tranziția reprezintă o relație între 2 stări, indicând faptul ca un obiect aflat in prima stare, în rezultatul producerii unui anumit eveniment va efectua o *acțiune* si apoi va intra in starea a doua.

O acțiune (*action*) reprezintă o prelucrare executată de un obiect și provocată de apariția unui eveniment. O acțiune este considerată ca instantanee, adică are un timp de execuție neglijabil în raport cu dinamica sistemului. Deci, o acțiune nu poate fi întreruptă de evenimente.

LL#3 Decision Table

1. Faceți o descriere succintă a SI ales (denumirea, adresa URL, domeniul de aplicație, scopul, utilizatori, funcționalități de bază, reguli de afacere/cerinte)
2. Elaborați tabelul de decizii (Decision Table - DT), care descrie regulile (business-logica) sistemului cercetat
3. Întocmiți darea de seamă cu sarcina, artefacte elaborate, concluzii

Tabel de decizii

Tabel de decizii este un instrument util pentru specificarea cerințelor și testarea sistemului realizat în baza acestor cerințe. Tabel de decizii este o reprezentare formalizată a regulilor de afacere complexe.

Reguli de afacere de obicei se formulează în formatul următor: "Având <unele condiții>, sistemul trebuie <sa faca unele actiuni>"

Într-un sistem complicat mai multe condiții se suprapun, astfel rezultatul (actiunile) se schimbă. Atunci, pentru a asigura acoperirea completa cu teste a tuturor regulilor de afacere, se recomandă utilizarea tabelului de decizii, care conține toate combinațiile posibile de condiții și toate acțiunile efectuate în aceste condiții.

Decision Table Template		Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8
Conditions									
Condition 1		choice 1a	choice 1b	choice 1a	choice 1b	choice 1a	choice 1b	choice 1a	choice 1b
Condition 2		choice 2a	choice 2a	choice 2b	choice 2b	choice 2a	choice 2a	choice 2b	choice 2b
Condition 3		choice 3a	choice 3a	choice 3a	choice 3a	choice 3b	choice 3b	choice 3b	choice 3b
Outcomes									
OC001 Outcome 1		-	-	X	-	-	-	-	X
OC002 Outcome 2		X	-	X	-	-	-	-	-
OC003 Outcome 3		X	-	X	-	X	-	X	X
OC004 Outcome 4		X	X	-	-	X	X	-	-
OC005 Outcome 5		-	X	-	X	-	-	-	-

Tabelul de decizii

Pasii de creare a tabelului de decizii

Pasul 1. Analizați cerințele și completați prima coloana: **condiții** (partea de sus) / **acțiuni** (partea de jos)

Pasul 2. Adăugați coloanele în partea dreapta a tabelului.

- Numărul de coloane depinde de numărul de condiții și numărul de valori posibile pentru fiecare condiție. În caz general, trebuie să înmulțiți numărul de valori pentru fiecare condiție.

- Dacă presupunem, ca fiecare condiție poate avea două valori*: adevărat (T) și fals (F), atunci numărul total de combinații posibile va fi 2^N , unde N - numărul de condiții (Ex.: având 3 condiții, trebuie să adăugăm în tabel $2^3 = 8$ coloane).

Pasul 3. Completați coloanele adăugate cu valori ale condițiilor

- În partea dreapta de sus coloanele se completează cu combinații de valori ale **condițiilor**. Pentru a evita greșeli, repetări sau pierderi se recomandă următoarea metodă de completare: divizați prima linie în 2 părți și completați toate celulele a părții de stânga cu T, iar toate celulele din partea dreaptă - cu F (ex.: TTTTFFFF). Pentru completarea liniei a doua ambele părți divizați în două păstrând același principiu de completare (ex.: TTFF). Continuați până atunci, când fiecare parte conține câte o celulă, cea din stânga completată cu T, iar cea din dreapta - cu F (ex.: TF). Astfel, fiecare coloană va conține o combinație unică de valori ale condițiilor.

Pasul 4. Determinați acțiunile, care au loc în anumite condiții.

- Fiecare coloană acum conține un set unic de valori ale condițiilor. Pentru a nota faptul că în aceste condiții are loc o acțiune, puneți bifă (cruce) în celula, care se află la intersecția coloanei corespunzătoare a setului de condiții și liniei, care corespunde acțiunii.

Pasul 5. Reduceți tabelul (daca este necesar)

- Pot exista situații când într-o combinație particulară unele condiții își pierd importanța. Analizați cerințele pentru a găsi aceste situații. Dacă ele există, marcați valorile nesemnificative cu "-". Observați că unele coloane au devenit identice. Ștergeți toate duplicatele.

Pasul 6. Dați denumirile coloanelor. Pot fi denumite scurt Rule1, Rule2, etc., sau TC01, TC02 etc., sau utilizați alte denumiri mai semnificative.

Pasul 7: Creați cazuri de testare (test cases)

- De fapt, fiecare coloană completată din tabel de decizii reprezintă o regulă de afacere (cerință funcțională) sau, din alt punct de vedere, ea poate fi privită ca un caz de testare, care verifică această regulă/cerință (cazul de testare verifică, dacă în <anumite condiții>, au loc <anumite acțiuni/rezultate>).

- Creați cel puțin un caz de testare pentru fiecare coloană. Acesta va fi numărul minim de teste, care asigură acoperirea cu teste a funcționalității analizate (cerințelor funcționale în baza cărora a fost creat tabelul).

Exemplu de tabel de decizie

Să analizăm **Sistem de vânzare a biletelor la cinematograful**.

În acest domeniu există următoarele **reguli de afacere (business rules)** pentru calcularea pretului biletului:

1. Locurile între rândurile 10 și 20 inclusiv se consideră locuri premium, iar celelalte - locuri simple.
2. Pentru locuri premium se aplică tariful Premium = 60 lei
3. Pentru locuri simple se aplică tariful Simplu = 40 lei
4. În zile de odihnă la pretul biletului se adaugă 10 lei
5. Pentru filme 3D la pretul biletului se adaugă 5 lei

Analizând reguli de afacere putem detecta 3 condiții, fiecare cu două valori:

1. Loc Premium (adevărat sau fals)
2. Zi de odihnă (adevărat sau fals)
3. Film 3D (adevărat sau fals)

Acțiunile detectate sunt:

1. Aplică tariful Premium
2. Aplică tariful Simplu
3. Adaugă 10 lei
4. Adaugă 5 lei

Creăm tabel de decizii și completăm condiții, acțiuni și valorile lor, utilizând metoda recomandată:

	Conditions	TC1	TC2	TC3	TC4	TC5	TC6	TC7	TC8
1	Loc Premium	T	T	T	T	F	F	F	F
2	Zi de odihna	T	T	F	F	T	T	F	F
3	Film 3D	T	F	T	F	T	F	T	F
	Actions								
1	Aplica tariful Premium = 60 lei	X	X	X	X	-	-	-	-
2	Aplica tariful Simplu = 40 lei	-	-	-	-	X	X	X	X
3	Adauga 10 lei	X	X	-	-	X	X	-	-
4	Adauga 5 lei	X	-	X	-	X	-	X	-
		75	70	65	60	55	50	45	40

Tabel 1. Tabel de decizii pentru calcularea pretului biletului la cinematograful

Concluzii

Tabelele de decizii sunt o modalitate bună de a descrie cerințele atunci, când există mai multe reguli de afaceri care interacționează împreună.

Utilizând tabelele de decizii un business-analist poate mai ușor analiza cerințele funcționale, iar un software tester poate mai ușor să creeze seturi complete de cazuri de testare pentru o funcționalitate. Un avantaj al utilizării tabelelor de decizie este că acestea permit detectarea combinațiilor de condiții care altfel nu ar fi fost găsite și, prin urmare, nu ar fi fost testate sau dezvoltate. Cerințele devin mult mai clare și uneori pot fi depistate unele cerințe imposibile, ceea ce este greu de văzut atunci, când cerințele sunt exprimate doar în text.

Totuși, un tabel de decizie nu este atât de detaliat cum sunt specificații complete ale cazurilor de testare. Elaborarea tabelelor de decizie reprezintă o tehnică de detectare a cazurilor de testare, care specifică numai datele de intrare și rezultatele așteptate. Cazuri de testare mai detaliate vor fi create în baza tabelului de decizii. Pe lângă datele de intrare și rezultatul așteptat, ele vor conține și instrucțiuni de testare pas cu pas.