

Inginerie Software 2024

Laboratoarele 2-3

Modelarea sistemului în context: funcții și interacțiuni *

Mădălina Erașcu, Alexandru Munteanu, Cristian Pal, Ionica Puiu

February 2024

Obiective:

1. Utilizarea diagramei cazurilor de utilizare în diferite contexte.
2. Descrierea cazului de utilizare.
3. Utilizarea diagramei de secvențe la nivel de sistem.

1 Introducere

Un **sistem software** oferă un **set de funcții**. Acestea sunt utilizate de **entități din contextul** său (actori) care **interacționează** cu sistemul.

Primul nivel de modelare se referă la:

- funcțiile oferite de sistem
- interacțiunile entităților din context cu sistemul pentru a obține aceste funcții. Abordarea se bazează pe **cazuri de utilizare**.

Un **caz de utilizare** definește interacțiunile dintre actorii externi și sistem în scopul realizării unui anumit obiectiv.

Actor specifică un rol ¹ jucat de o persoană sau o altă entitate (un alt sistem, un dispozitiv) atunci când interacționează cu sistemul.

“Use case is a behaviourally related sequence of transactions performed by an actor in a dialogue with the system to provide some measurable value to the actor” [1].

Actorii există în afara sistemului studiat și participă la o secvență de activități în *dialog cu sistemul*, în scopul îndeplinirii unui anumit *obiectiv*. *Modelul sistemului* conține:

*Bazat pe resursele de laborator ale Conf. Dr. Cristina Mândruță

¹ Aceeași persoană poate utiliza sistemul ca actori diferiți deoarece aceasta poate juca roluri diferite la momente de timp diferite.

- Diagrama cazurilor de utilizare
- Detaliile pentru fiecare caz de utilizare

Remarcă. *Analiza cazurilor de utilizare este forma primară de culegere a cerințelor de utilizare pentru un nou program software sau sarcină de îndeplinit.*

Obiectivele primare ale analizei cazurilor de utilizare sunt:

- proiectarea unui sistem din perspectiva utilizatorului
- comunicare comportamentului sistemului în termenii utilizatorului
- specificarea tuturor interacțiunilor vizibile din exterior.

2 Diagrama Cazurilor de Utilizare

- Scop: **Ce** trebuie să facă sistemul.
- Descrie **funcționalitatea** sistemului din punctul de vedere al utilizatorului.
- Nivel superior de abstractizare.
- Modelare centrată pe obiectiv: scopul (**ce** face sistemul) nu soluția (**cum**).

2.1 Elemente de modelare

Actor – entitate ce interacționează cu sistemul; rol jucat de o persoană, de un alt sistem sau de un dispozitiv hardware.

Sistem – sistemul modelat.

Caz de utilizare (Use Case) – un serviciu pe care sistemul știe să-l furnizeze; funcție cheie a sistemului.

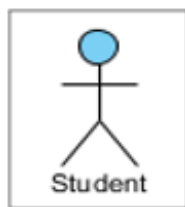
Tipuri de relații:

- **asociere** – interacțiune între actor și caz de utilizare.
- **«include»** – între două cazuri de utilizare; cazul de utilizare inclus este un caz de utilizare reutilizabil care este incorporat necondiționat în executarea cazului de utilizare ce îl include; cazul de utilizare apelant decide când și de ce utilizează cazul de utilizare inclus.
- **«extend»** – între două cazuri de utilizare; cazul de utilizare care extinde este un caz de utilizare reutilizabil care întrerupe condiționat execuția cazului de utilizare extins pentru a-i extinde funcția; cazul de utilizare care extinde decide când va fi folosit.
- **generalizare** – relație de moștenire între actori sau între cazuri de utilizare.

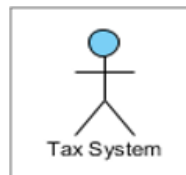
2.2 Notății

2.2.1 Actor

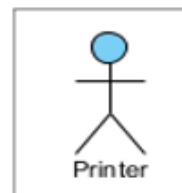
Vezi Figura 1.



(a) O persoană



(b) Un alt sistem



(c) Un dispozitiv extern

Figure 1: Tipuri de Actori



Figure 2: Caz de utilizare

2.2.2 Caz de utilizare (Use case)

Vezi Figura 2.

Numele cazului de utilizare trebuie să înceapă cu un verb!

2.2.3 Relații

Pentru un sumar al relațiilor, consultați Tabelul 1.

1. Generalizare (vz. Figura 3)

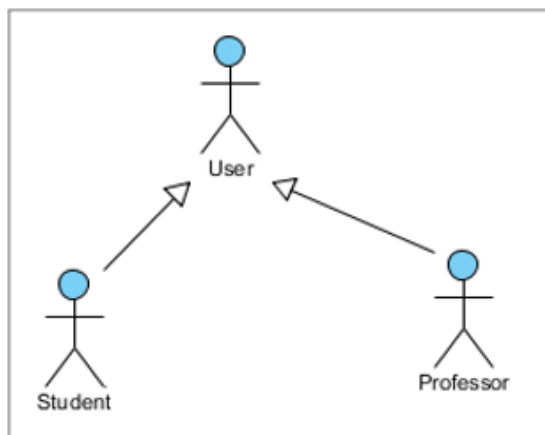


Figure 3: Exemplu de relație de generalizare: Actorii "Student" și "Profesor" sunt specializări ale actorului "User" care este mai general.

2. Incluziune ($\ll include \gg$) (vz. Figura 4)

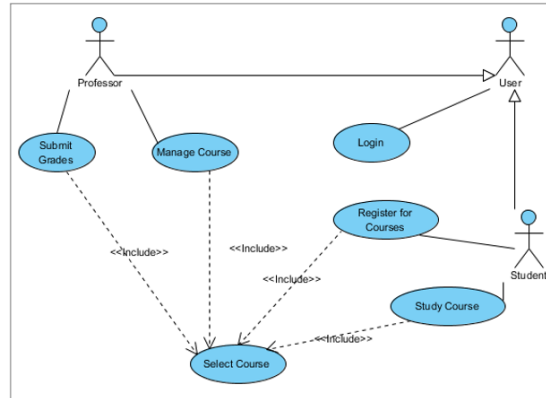


Figure 4: Exemplu de relație de incluziune: Cazul de utilizare "Select Course" va fi executat de fiecare dată când va fi lansat "Register for Courses" sau "Study Course".

3. Extindere ($\ll extend \gg$) (vz. Figura 5)

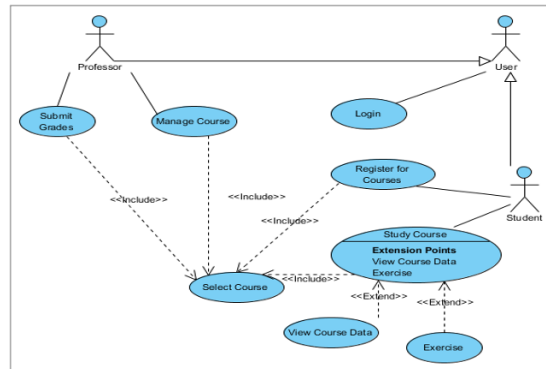


Figure 5: Exemplu de relație de extindere: Cazul de utilizare "View Course Data" va fi executat doar în punctul de extensie "View Course Data" al cazului de utilizare "Study Course". Similar pentru cazul de utilizare "Exercise".

2.3 Marcarea limitelor sistemului (opțional)

Cazurile de utilizare pot fi incluse într-un dreptunghi care va delimita sistemul de contextul său. Tot ce e conținut în acest dreptunghi reprezintă funcționalitatea aflată în zona sistemului și tot ce este în afara acestuia se află în afara sistemului.

Remarcă. Reprezentarea limitelor poate fi utilizată și pentru a identifica ce cazuri de utilizare vor fi livrate în fiecare versiune majoră a sistemului.

include	extend
Extinde comportamentul cazului de utilizare de bază.	Extinde comportamentul cazului de utilizare de bază.
Cazul de utilizare inclus este folosit întotdeauna pentru a extinde cazul de utilizare în curs de execuție.	Cazul de utilizare care extinde ar putea fi folosit (uneori) pentru a extinde cazul de de utilizare în curs de execuție.
Cazul de utilizare în curs de execuție decide când apelează cazul de utilizare inclus. Cazul de utilizare inclus nu este conștient de cazul de utilizare de bază.	Cazul de utilizare care extinde decide când se va insera în execuția cazului de utilizare de bază. Cazul de utilizare de bază nu este conștient de cazul de utilizare care extinde.
Săgeata este desenată către cazul de utilizare inclus.	Săgeata este desenată către cazul de utilizare extins.

Table 1: Sumar al relațiilor

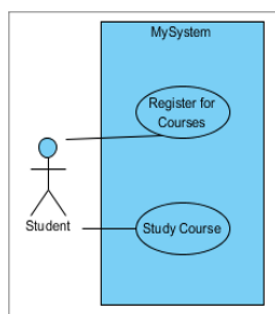


Figure 6: Delimitare sistem de context

3 Reguli

- Un caz de utilizare definește un serviciu, nu secvențe de operații.
- Toți actorii comunică cu un singur sistem.
- Actorii nu comunică direct între ei.
- Trebuie identificate toate funcțiile de nivel înalt ale sistemului, adică ceea ce sistemul știe să facă pentru actorii săi.

Remarcă. *Explicați erorile din Figurile 7 and 8.*

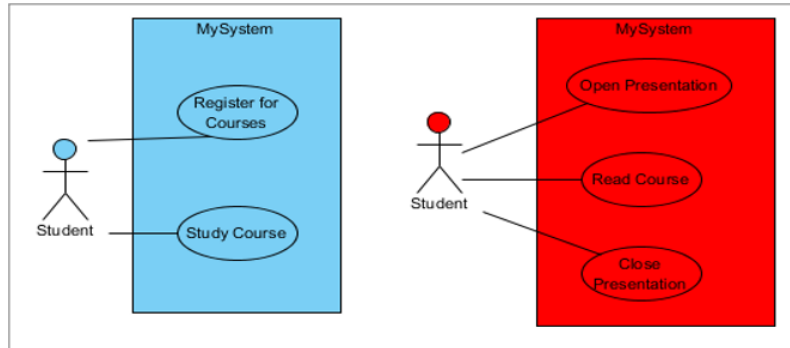


Figure 7: Exemplu 1. Corect - Gresit

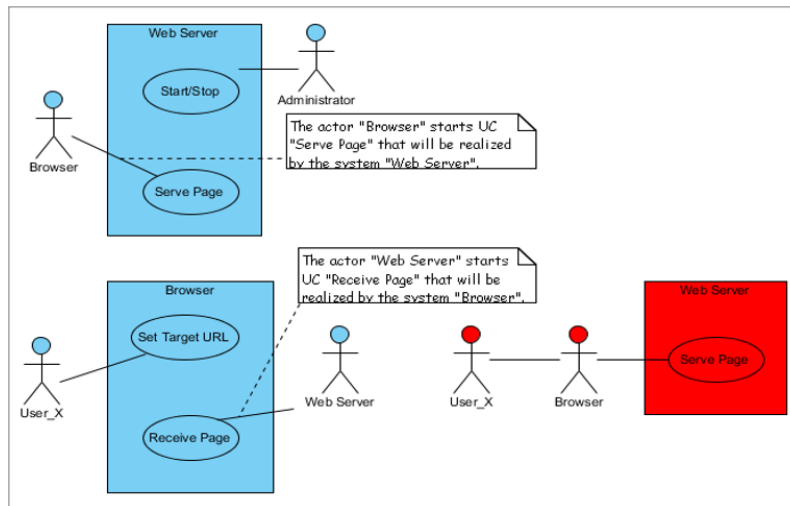


Figure 8: Exemplu 2. Corect - Gresit

4 Detaliile Cazului de Utilizare

Diagrama cazurilor de utilizare ilustrează actori, cazuri de utilizare și relațiile dintre aceste entități, fără detalii referitoare la comportament.

Detaliile cazului de utilizare este un document structurat ce descrie cazul de utilizare sub forma unui set de interacțiuni ale sistemului cu actorii, având un început (declanșator), un cuprins (dialog) și un final.

Detaliile cazului de utilizare reprezintă o serie completă de interacțiuni, descrise din punctul de vedere al Actorului.

Detaliile² descriu modul în care Actorul va interacționa cu sistemul pentru

²Recomandare referitoare la limbaj: evitați jargonul tehnic; se va prefera limbajul utilizatorului final sau al expertului domeniului aplicației.

a îndeplini un anumit obiectiv.

Dintr-un caz de utilizare se pot genera unul sau mai multe scenarii, cu detalii pentru fiecare variantă posibilă de realizare a acestuia.

Elementele unei descrieri a detaliilor unui caz de utilizare includ (dar nu se reduc la):

1. *Numele cazului de utilizare* are rolul de identificator unic pentru acesta. Trebuie scris într-un format verb-substantiv (e.g., Împrumut cărți, Extragere bani), trebuie descris un obiectiv ce poate fi realizat și trebuie să fie suficient pentru ca utilizatorul final să poată înțelege la ce se referă cazul de utilizare. Analiza cazurilor de utilizare condusă pe bază de obiectiv va denumi cazurile de utilizare conform obiectivelor actorului, asigurând astfel centrarea pe utilizator a cazurilor de utilizare. Două sau trei cuvinte este optim.
2. *Obiectivul* descrie pe scurt ce anume dorește utilizatorul să obțină de la sistem în cadrul acestui caz de utilizare.
3. *Actorii implicați*. Un actor este o entitate din afara sistemului care fie acționează asupra sistemului – actor primar – fie sistemul acționează asupra lui – actor secundar. Un actor poate fi o persoană, un dispozitiv, un alt sistem sau sub-sistem, sau timpul. Actorii reprezintă diferitele roluri pe care entități din afara sistemului le joacă în relația cu sistemul ale cărui cerințe funcționale sunt specificate. Un individ din lumea reală poate fi reprezentat de mai mulți actori dacă acesta joacă mai multe roluri în utilizarea sistemului.
4. *Precondiții* definesc toate condițiile referitoare la starea sistemului ce trebuie să fie îndeplinite astfel încât să aibă sens inițierea cazului de utilizare de către declanșator. Dacă sistemul nu se află în starea descrisă de preconditionii comportamentul cazului de utilizare este nedeterminat. Pe de altă parte, simplul fapt că preconditioniile sunt îndeplinite nu inițiază cazul de utilizare.
5. *Declanșatoare*. Această secțiune descrie evenimentul care produce inițierea cazului de utilizare. Acest eveniment poate fi extern, temporal sau intern. Dacă declanșatorul nu este un eveniment trivial (e.g., clientul apasă un buton), ci reprezintă îndeplinirea unui set de condiții, este necesar un proces de declanșare care se execută continuu (sau periodic) pentru a testa îndeplinirea condițiilor de declanșare. Evenimentul declanșator este un semnal de la acest proces care indică îndeplinirea tuturor condițiilor din set.
6. *Fluxul de bază*. Fiecare caz de utilizare trebuie să transmită un scenariu primar (un curs tipic de evenimente), numit și "flux de bază", "happy flow" sau "happy path". Cursul de bază al evenimentelor este reprezentat deseori ca un set de pași numerotați.

Exemplu. *Sistemul cere utilizatorului să se conecteze. Utilizatorul introduce numele și parola. Sistemul verifică informația introdusă. Sistemul conectează utilizatorul la sistem.*

7. *Fluxuri alternative*. Cazurile de utilizare pot conține căi secundare (sce-

narii alternative) care sunt variații ale temei principale. Fiecare regulă testată poate conduce la o cale alternativă, iar dacă sunt mai multe reguli atunci numărul căilor crește rapid, ceea ce conduce la documente foarte complexe. Uneori este mai bine să se utilizeze logica condițională sau diagrame de activitate pentru descrierea cazurilor de utilizare cu multe reguli și condiții.

8. *Excepții*. Excepțiile, sau ceea ce se întâmplă dacă lucrurile merg prost la nivelul sistemului, pot fi descrise, de asemenea, nu prin utilizarea acestei secțiuni ci într-o secțiune dedicată lor.

Căile alternative utilizează numerotarea cursului de bază al evenimentelor pentru a indica punctele în care diferă de scenariul de bază, și, dacă e cazul, unde se reunifică cu acesta.

Pe *diagrama de activitate* (se va studia în laboratoarele viitoare) se pot reprezenta simultan fluxul de bază, fluxurile alternative și tratarea excepțiilor.

9. *Punctele de extensie*. Condițiile de lansare a cazurilor de utilizare care extind acest caz de utilizare.
10. *Postcondiții*. Secțiunea post-condiții descrie starea sistemului după încheierea cazului de utilizare. Postcondițiile sunt garantate a fi îndeplinite la terminarea cazului de utilizare.
11. *Note*. Secțiune ce permite înregistrarea de informații, mai puțin structurate dar necesare, ce nu au loc în alte secțiuni ale șablonului.
12. *Versiune, autor și dată*. Conține identificatorul versiunii, data creării acestei versiuni a cazului de utilizare și autorul (autorii). documentului. Trebuie să conțină și lista tuturor versiunilor anterioare care fac parte din documentația curentă a sistemului.

5 Metodologie

La scrierea efectivă a cazurilor de utilizare nu se urmărește obținerea descrierii perfecte de prima dată. Dezvoltarea cazurilor de utilizare trebuie privită ca un process iterativ cu rafinări successive ale detaliilor.

5.1 Pașii dezvoltării unei diagrame de cazuri de utilizare

1. Definirea contextului sistemului: identificarea actorilor și a responsabilităților lor.
2. Identificarea cazurilor de utilizare: funcțiile sistemului în termeni de obiective specifice și/sau rezultate ce trebuie produse.
3. Evaluarea actorilor și cazurilor de utilizare pentru găsirea oportunităților de rafinare (cum ar fi divizarea sau gruparea definițiilor).
 - Evaluarea cazurilor de utilizare pentru găsirea relațiilor «include».
 - Evaluarea cazurilor de utilizare pentru găsirea relațiilor «extend».
 - Evaluarea actorilor și cazurilor de utilizare pentru a detecta oportunitățile de generalizare (proprietăți comune).

5.1.1 Identificarea actorilor

Remarcă. Trebuie identificate roluri reprezentative, nu persoane individuale! (vz. Figura 9)

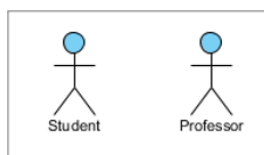


Figure 9: Identificare actorilor

5.1.2 Identificarea cazurilor de utilizare

Procedura este următoarea:

1. Identificarea serviciilor potențiale punând următoarele întrebări d.p.d.v. al fiecărui actor:
 - Ce încearcă să obțină utilizatorii acestui rol?
 - Ce trebuie să poată face utilizatorii pentru a îndeplini acest rol?
 - Care sunt principalele sarcini ale utilizatorilor acestui rol?
 - Ce informații trebuie să examineze, să creeze sau să modifice utilizatorii acestui rol?
 - Ce informații trebuie să primească de la sistem utilizatorii acestui rol?
 - Ce informații trebuie să transmită sistemului utilizatorii acestui rol?
2. Schițarea obiectivelor fiecărui actor. În urma stabilirii actorilor și obiectivelor fiecăruia rezultă o listă inițială cu cazurile de utilizare de nivel înalt (vz. Figura 10).

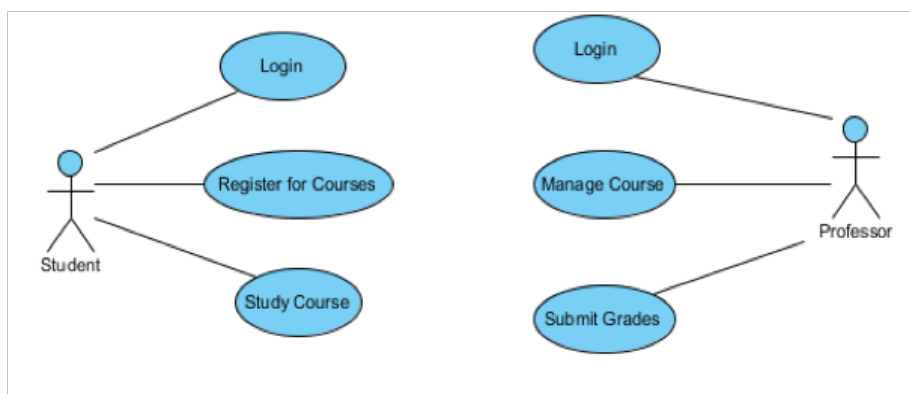


Figure 10: O listă de cazuri de utilizare inițiale

5.1.3 Evaluarea actorilor și a cazurilor de utilizare pentru găsirea oportunităților de rafinare

Identificarea oportunităților de reutilizare a cazurilor de utilizare În prima versiune a diagramei cazurilor de utilizare din exemplul nostru apare un caz de utilizare duplicat atât la Student cât și la Professor care include "Login". Putem introduce un actor mai general care are acest caz de utilizare iar cei doi actori inițiali îl vor "moșteni" de la noul actor.

Figura 4 ilustrează faptul că un utilizator general se conectează la sistem iar studentul și profesorul posedă cazuri de utilizare proprii dar și cazul de utilizare al utilizatorului general.

Beneficiile generalizării constau în eliminarea duplicării, ceea ce va conduce la un sistem mai ușor de înțeles și totodată mai flexibil.

Generalizarea se poate aplica atât la actori cât și la cazuri de utilizare.

Evaluarea cazurilor de utilizare pentru găsirea relațiilor «include»

Pentru a opera cu un curs, atât profesorul cât și studentul trebuie ca în prealabil să selecteze unul din cele pe care le predă, respectiv la care este înscris.

Presupunem că selectarea unui curs este suficient de complexă pentru a deveni un caz de utilizare. Selectarea unui curs trebuie făcută pentru înregistrarea notelor studenților, pentru înregistrarea studentului la curs, pentru gestionarea cursului sau pentru studiul acestuia.

În consecință, cazul de utilizare "Select Course" va fi executat întotdeauna ca sprijin pentru fiecare din cazurile de utilizare "Submit Grades", "Manage Course", "Register for Courses" sau "Study Course".

Aceasta se reprezintă prin relația «include» în diagrama cazurilor de utilizare, ca în Figura 4.

Evaluarea cazurilor de utilizare pentru găsirea relațiilor «extend»

Atunci când studiază un curs, studentul poate fie să citească documentația cursului fie să exerseze folosind temele și testele oferite împreună cu acest curs.

În consecință, fie cazul de utilizare "View Course Data" fie cazul de utilizare "Exercise" va fi executat, conform unei selecții făcută de student.

Aceasta se reprezintă prin relația «extend» în diagrama cazurilor de utilizare, ca în Figura 5.

5.2 Exemplu aplicare metodologie de dezvoltare a cazurilor de utilizare

1. **Identificarea componentelor cheie ale cazului de utilizare.** Cazul de utilizare este reprezentat textual ilustrându-se o secvență de evenimente. Fie cazul de utilizare "Submit grades".
2. **Alocarea unui nume și descrierea pe scurt a cazului de utilizare**

Nume UC: Submit grades

Actor: Professor

Descriere: Profesorul înregistrează notele obținute de studenți.

3. **Identificare precondiții:** Actorul trebuie să fi fost autentificat în sistem ca Professor.
4. **Crearea fluxului principal al UC:** Fluxul principal al UC reprezintă fluxul cel mai important al evenimentelor, care are loc de cele mai multe ori (numit și "Happy Day Scenario" deoarece are loc atunci când totul decurge normal, fără erori sau excepții). Cursul principal este important și ca referință pentru înțelegerea excepțiilor. De asemenea, deoarece reprezintă în medie 70% din sistem, este mai probabil ca echipa de dezvoltare să implementeze codul corect încă din primele etape.

Flux principal:

- 4.1. Profesorul selectează opțiunea de înregistrare note.
- 4.2. Sistemul afișează lista cursurilor predate de profesor.
- 4.3. Profesorul selectează un curs.
- 4.4. Sistemul afișează lista studenților înregistrați la cursul respectiv.
- 4.5. Pentru fiecare student din listă
 - 4.5.1. Profesorul indică nota
 - 4.5.2. Sistemul memorează nota
5. **Crearea fluxurilor alternative ale UC** Fluxurile alternative pot fi:
 - 5.1. Flux de excepție sau de eroare pentru orice linie din fluxul de bază.
 - 5.2. Flux adițional, bazat pe condiții ce pot să apară în decursul interacțiunii.Flux alternativ 1 : Dacă la pasul 4.5.1 profesorul indică o notă în afara intervalului 1-10 sistemul sistemul afișează un mesaj și se reia pasul 4.5.1.
6. **Identificare postcondiții**

Notele obținute de studenți la cursul selectat există memorate în sistem.
7. **Utilizarea diagramei de secvențe la nivel de sistem**

Pentru fiecare caz de utilizare se pot reprezenta cu diagrame de secvențe la nivel de sistem (vz. Figura 11) interacțiunile între actori și sistem, adică interacțiuni descrise în fluxul principal și fluxurile alternative. Într-o astfel de diagramă sistemul este văzut ca obiect unic în dialog cu unul sau mai mulți actori.

Pe diagrama de secvențe la nivel de sistem se reprezintă actorii și sistemul sub formă de "*life lines*". Interacțiunile actorilor cu sistemul se realizează prin mesaje reprezentate cu săgeți.

Fiecare flux (principal și alternativ) se poate reprezenta pe câte o diagramă de secvențe. Fluxurile simple se pot combina pe o singură diagramă. Pentru aceasta este necesară reprezentarea de structuri logice folosind fragmente speciale (combined fragment) de tip "*loop*" pentru structura repetitivă și "*alt*" pentru structura decizională.

6 Studiu

1. Studiați crearea de diagrame de cazuri de utilizare de la https://www.visual-paradigm.com/support/documents/vpuserguide/94/2575/6362_drawinguseca.html.

sd Submit Grades UC

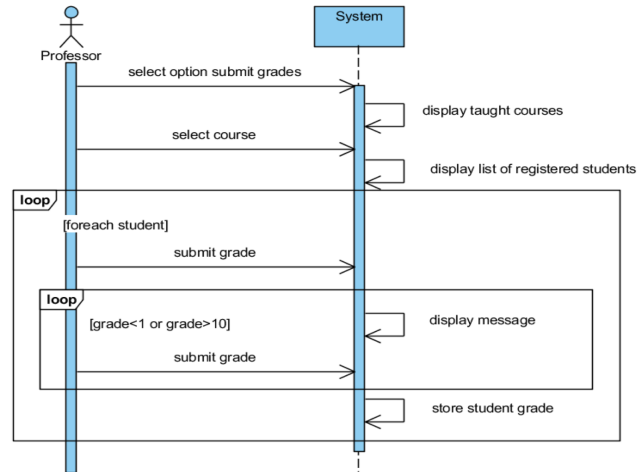


Figure 11: Diagrama de secvențe la nivel de sistem

2. Studiați crearea de diagrame de secvențe https://www.visual-paradigm.com/support/documents/vpuserguide/94/2577/7025_drawingseque.html.

7 Temă

1. Desenați diagrama cazurilor de utilizare (abreviată UC) pentru următorul robot telefonic.
 - UC: Leave a Message
 - Actor: Caller
 - Pre-Condiție: Room on Tape
 - Post-Condiție: New Message
 - UC: Review Messages
 - Actor: Owner
 - Cale principală: Review New
 - Cale alternativă: No New Messages
 - UC: Review Messages Locally
 - Cale principală: Review New Messages
 - Cale alternativă: No New Messages
 - Extinde: Review Messages
 - UC: Review Messages Remotely
 - Cale principală: Review New Messages
 - Cale alternativă: No New Messages
 - Extinde: Review Messages
 - Include: Authorize Access

- UC: Authorize Access
 - Cale principală: User Authorized
 - Cale alternativă: User Not Authorized
2. Creați o diagramă UC pe baza următoarei descrieri. Dezvoltăm un sistem cu care profesorii pot înregistra și actualiza notele studenților. Profesorii trebuie să poată distribui rapoarte cu notele. Iată lista completă a cerințelor sistemului:
- Un profesor poate înregistra note. De câte ori sunt înregistrate note, acestea sunt salvate pe disc.
 - Un profesor poate actualiza note. De câte ori sunt actualizate note, nota existentă este încărcată. După modificare noua notă este salvată pe disc.
 - Profesorul, secretara și studentul pot vizualiza note.
 - Pentru a vizualiza note, solicitantul trebuie să se conecteze la sistem. Dacă eșuează conectarea, solicitantul trebuie să se re-autentifice indicând nume și parolă.
 - Un student cu taxă este un tip de student.
 - O secretară poate genera rapoarte cu notele.
 - Un profesor poate distribui rapoarte cu notele.

Cerințe

- Identificați actorii.
 - Există actori care sunt specializări ale altor actori mai generali? Dacă da, identificați care este actorul general și care este actorul specializat. Ce tip de relație trebuie reprezentată între actorul general și specializarea sa?
 - Identificați cazurile de utilizare.
 - Există cazuri de utilizare folosite întotdeauna de alte cazuri de utilizare? Dacă da, ce tip de relație există între acestea? Identificați cazurile de utilizare folosite întotdeauna și cazurile de utilizare ce le folosesc.
 - Există cazuri de utilizare folosite uneori de către un alt caz de utilizare? Dacă da, ce tip de relație există între acestea? Identificați cazurile de utilizare folosite uneori și cazurile de utilizare ce le folosesc.
 - Desenați diagrama UC corespunzătoare, incluzând toți actorii, cazurile de utilizare, și relațiile. Atenție la folosirea notațiilor corespunzătoare și a etichetelor pentru actori, cazuri de utilizare și relații.
3. Desenați o diagramă UC pentru următoarea aplicație. O bibliotecă conține cărți și jurnale. Se cere dezvoltarea unui sistem software pentru împrumutul de cărți. Pentru a împrumuta o carte, clientul trebuie să fie membru al bibliotecii. Există o limită a numărului de cărți ce pot fi simultan împrumutate de un membru al bibliotecii. Biblioteca poate deține mai multe exemplare ale unei anumite cărți. O carte se poate rezerva. Unele cărți pot fi împrumutate doar pe termen scurt. Alte cărți pot fi împrumutate pentru 3 săptămâni. Utilizatorii pot extinde împrumuturile.

Apoi detaliați textual următoarele cazuri de utilizare:

- Împrumut exemplar carte.
- Extinderea împrumutului.

Remarcă. *Dialogurile între actori și sistem vor fi reprezentate cu o diagramă de secvențe la nivel de sistem care va fi apoi exportată ca fișier grafic. Importați apoi documentul grafic în fișierul text cu restul descrierii fiecărui caz de utilizare.*

4. Fie următoarea descriere a cerințelor unei aplicații software pentru o firmă de închiriere mașini. Firma are mai multe oficii de unde se pot închiria mașinile. Identificați actorii și utilizați descrierile de mai jos ca bază pentru definirea cazurilor de utilizare. Desenați un model al cazurilor de utilizare ilustrând toate relațiile dintre cazurile de utilizare. Reprezentați apoi în VP for UML diagrama cazurilor de utilizare, **editați detaliile și reprezentați fluxurile de evenimente cu diagrame de secvențe la nivel de sistem.**

REZERVARE

Clientul accesează site-ul firmei cu scopul de a face o rezervare. Pe site i se va afișa un formular în care va trebui să indice data inițială și data finală, vehiculul preferat și oficiul de închiriere de unde dorește să închirieze mașina. După ce datele sunt trimise sistemul verifică dacă este disponibil un vehicul corespunzător solicitării clientului. Dacă vehiculul solicitat este disponibil, atunci sistemul calculează prețul și îl afișează clientului. Clientul acceptă prețul afișat. Sistemul înregistrează rezervarea iar clientul primește un număr de închiriere. Sistemul crează un acord de închiriere, incluzând numărul de închiriere, perioada închirierii, tipul vehiculului și oficiul de închiriere.

Excepții

- Nu este disponibil un vehicul corepunzător la oficiul de închiriere ales. Clientului i se oferă posibilitatea de a indica un alt set de preferințe sau poate anula solicitarea.
- Clientul nu este de acord cu prețul. Clientul poate indica un alt set de cerințe sau poate anula solicitarea.

VERIFICARE DISPONIBILITATE

Sistemul verifică disponibilitatea pentru a vedea dacă este disponibil un vehicul de un anumit tip, la un anumit oficiu de închiriere, pentru o perioadă dată de timp. Pentru fiecare vehicul sistemul cunoaște intervalele de timp când este disponibil și când nu. Dacă este disponibil, vehiculul este rezervat pentru perioada solicitată.

Excepții

- Dacă nu se poate face o rezervare datorită lipsei de vehicule sistemul generează un raport ce va fi utilizat la planificarea activității firmei.

INIȚIERE ÎNCHIRIERE

Clientul ajunge la oficiul de închiriere și indică angajatului numărul de închiriere. Angajatul introduce numărul de închiriere în sistem. Sistemul

caută acordul de închiriere corespunzător și îi afișează pentru a fi discutat cu clientul. Dacă clientul acceptă acordul de închiriere atunci acesta este imprimat pentru a fi semnat de către client. Apoi sistemul afișează o listă de opțiuni de asigurare existente. Clientul indică opțiunea preferată. Angajatul introduce în sistem preferința clientului. Sistemul imprimă formularul poliței de asigurare pentru a fi semnată de către client și atașată la acordul semnat.

Excepții

- Clientul nu are o rezervare. În acest caz angajatul introduce preferințele clientului. Sistemul verifică disponibilitatea unui vehicul corespunzător. Dacă vehiculul este disponibil, atunci clientului i se afișează și prețul. Dacă clientul acceptă prețul atunci este creat acordul de închiriere și este inițiată închirierea.
- Dacă nu este disponibil un vehicul de tipul celui rezervat (din cauza întârzierii unei returnări) atunci clientului i se oferă posibilitatea de a indica un alt set de preferințe.

PROCESARE RETURNARE VEHICOL

Clientul înregistrează kilometrajul și nivelul de combustibil și le indică angajatului care le introduce în sistem. Sistemul calculează cantitatea de combustibil consumată de client și o adaugă la contul închirierii din acordul de închiriere. Contul închirierii, care include costurile totale ale închirierii, este afișat de sistem și verificat de către client. Clientul plătește costurile închirierii. Angajatul înregistrează în sistem faptul că plata a fost făcută.

Excepții

- Vehiculul returnat este avariata, necesitând completarea unei revendicări a asigurării. Angajatul înregistrează în sistem faptul că vehiculul este avariata. Sistemul imprimă formularul de revendicare pentru a fi completat și semnat de către client.
- Clientul contestă contul. Angajatul înregistrează în sistem acest lucru.

CREARE RAPOARTE MANAGEMENT

Sistemul poate genera mai multe tipuri de rapoarte. Managerul firmei selectează un tip de raport. Sistemul generează și afișează raportul solicitat. Dacă managerul solicită imprimare, sistemul imprimă raportul.

OPTIONAL: O EXTENSIE ULTERIOARĂ

Extindeți proiectul prin adăugarea de caracteristici suplimentare suport pentru închirieri regulate de către firme client. În acest caz vor fi disponibile zilnic un număr precizat de vehicule pentru angajații companiei client, la oficiile de închiriere precizate. Firma client poate nominaliza și oferi o listă cu angajații autorizați ce pot ridica vehiculele. Firmei client i se va prezenta lunar un cont de plată. Pentru aceasta va trebui ca mai întâi să scrieți un nou caz de utilizare și/sau să adaptați cazurile de utilizare existente.

References

- [1] Ivar Jacobson, Maria Ericsson, and Agneta Jacobson. *The object advantage: business process reengineering with object technology*. ACM Press/Addison-Wesley Publishing Co., 1994.