Tema

1.Limbajul de Modelare Unificat (UML) este o consolidare a celor mai bune practici unificate de modelare care au fost stabilite de-a lungul anilor în utilizarea limbajelor de modelare. UML ne permite să prezentăm aspectele foarte variate ale unui sistem software (de exemplu, cerințe, structuri de date, fluxuri de date și fluxuri de informații) într-un singur cadru folosind concepte orientate pe obiecte.

2.

Noțiunea de model este importantă nu numai în tehnologia informației, ci și în multe alte discipline științifice (matematică, filozofie, psihologie, economie, etc.). Derivată din latinul "modulus", care desemnează o scară în arhitectură, în timpul Renașterii cuvântul "modello" era folosit în Italia pentru un obiect ilustrativ destinat să prezinte forma și designul unei clădiri planificate unui client și să clarifice întrebările de design și arhitectură. În decursul secolelor următoare, noțiunea de "model" a fost utilizată în diverse ramuri ale științei pentru o descriere simplificată a faptelor complexe din realitate. În 1973, Herbert Stachowiak a propus o teorie a modelului care se deosebește prin trei caracteristici:

1. Mapping: un model este întotdeauna o imagine (mapare) a ceva, o reprezentare a originalelor naturale sau artificiale care pot fi ele însele modele.

2. Reduction: un model nu capturează toate atributele originale, ci doar pe cele care par relevante pentru modelator sau utilizatorul modelului.

3. Pragmatism: pragmatismul înseamnă orientare către utilitate. Un model este atribuit unui original pe baza următoarelor întrebări:

Pentru cine? De ce? Pentru ce? Un model este folosit de modelator sau de utilizator în locul originalului într-un cadru de timp specific și pentru un scop specific. Modelele susțin o reprezentare a unui sistem redus la esențial pentru a minimiza complexitatea sistemului la aspecte gestionabile. Un sistem este de obicei descris nu printr-o singură vedere, ci printr-un număr de vederi care împreună produc o imagine generală unificată. Astfel, o vedere ar putea descrie obiectele implicate și relația lor între ele; o altă vedere ar putea descrie comportamentul unui grup de obiecte sau prezenta interacțiunile între diferite obiecte.

3.

1. **Diagrama de clase (Class Diagram)**:

* + Asemănări: Ambele diagrame sunt utilizate pentru a reprezenta structura statică a sistemului, inclusiv clasele din sistem și relațiile între ele.
  + Deosebiri: Diagrama de clase se concentrează pe structura statică a sistemului, în timp ce diagrama de obiecte se concentrează pe instanțele de obiecte specifice la un moment dat.

1. **Diagrama de obiecte (Object Diagram)**:
   * Asemănări: Ambele diagrame sunt utilizate pentru a reprezenta instanțele de obiecte și relațiile între acestea.
   * Deosebiri: Diagrama de obiecte este o instanță particulară a sistemului la un anumit moment, în timp ce diagrama de clase reprezintă structura generală a claselor din sistem.
2. **Diagrama de secvență (Sequence Diagram)**:
   * Asemănări: Ambele diagrame sunt utilizate pentru a reprezenta interacțiunile între obiectele din sistem într-o anumită secvență temporală.
   * Deosebiri: Diagrama de secvență se concentrează pe secvența de mesaje trimise între obiecte, în timp ce diagrama de activitate se concentrează pe fluxurile de control și procese.
3. **Diagrama de activitate (Activity Diagram)**:
   * Asemănări: Ambele diagrame sunt utilizate pentru a modela fluxurile de control și procese în cadrul sistemului.
   * Deosebiri: Diagrama de activitate se concentrează pe procese și fluxuri de control, în timp ce diagrama de secvență se concentrează pe interacțiunile între obiecte.
4. The Communication Diagram

Asemenea diagramelor de secvență, diagrama de comunicare (vezi Capitoulul 6) descrie comunicarea între diferite obiecte. Aici, accentul este pus pe relațiile de comunicare între partenerii de interacțiune în loc să fie pe ordinea cronologică a schimbului de mesaje. Structurile de control complexe nu sunt disponibile. Această diagramă arată în mod clar cine interacționează cu cine.

1. The Timing Diagram

Diagrama de temporizare arată în mod explicit schimbările de stare ale partenerilor de interacțiune care pot apărea datorită evenimentelor de timp sau ca rezultat al schimbului de mesaje. De exemplu, o persoană se află în starea de conectat atunci când primește mesajul de la sistemul de administrare al universității că parola trimisă este validă.

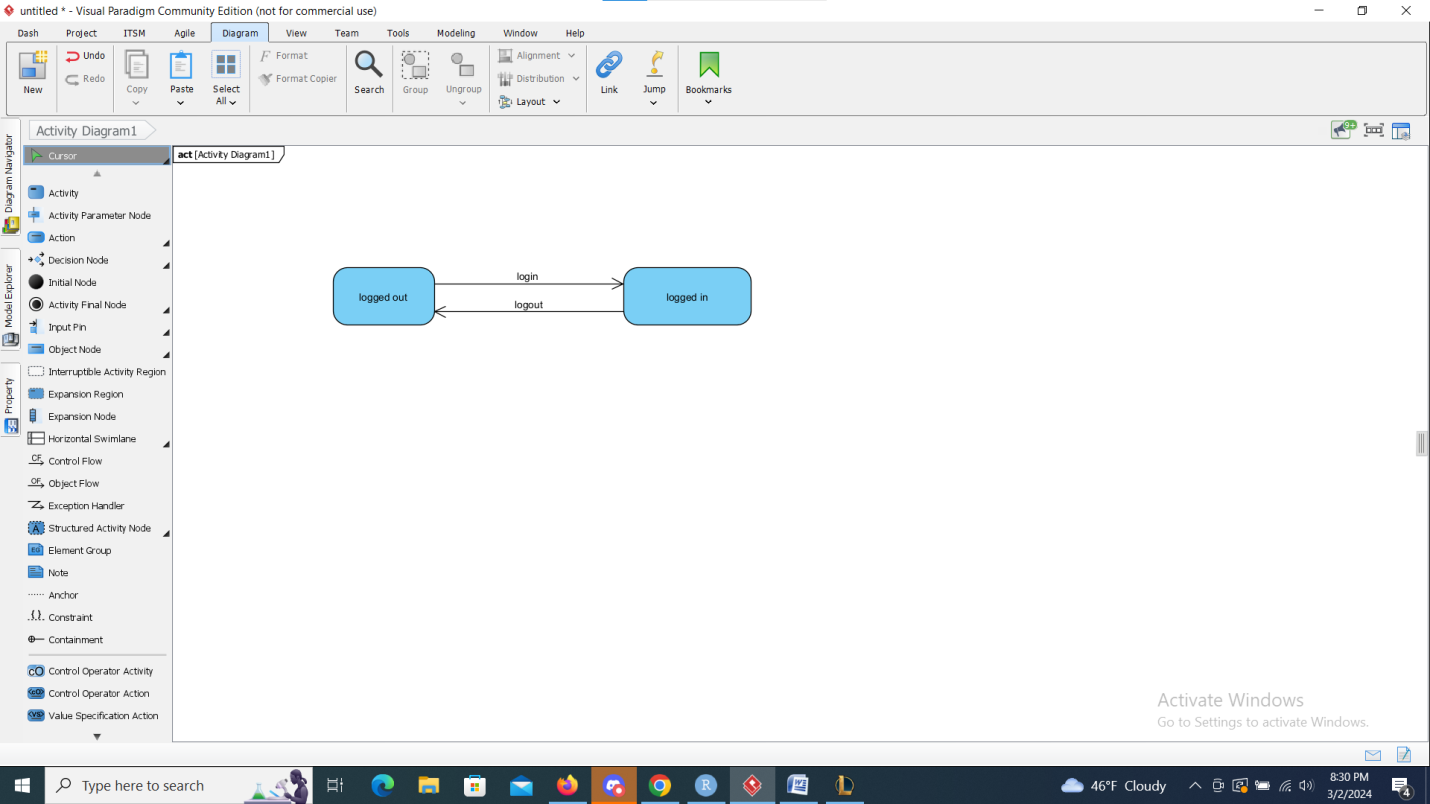
1. The Interaction Overview Diagram

Diagrama de supraveghere a interacțiunii modelează conexiunea dintre diferitele procese de interacțiune prin stabilirea unor diagrame individuale de interacțiune (adică diagrama de secvență, diagrama de comunicare, diagrama de temporizare și alte diagrame de supraveghere a interacțiunii) într-o secvență bazată pe timp și cauzalitate. De asemenea, specifică condițiile în care procesele de interacțiune sunt permise să aibă loc. Pentru a modela fluxul de control, sunt utilizate conceptele din diagrama de activitate. De exemplu, un utilizator al sistemului de administrare a universității trebuie să se autentifice mai întâi (care deja reprezintă o interacțiune separată cu sistemul) înainte de a i se permite să folosească funcționalitățile ulterioare.

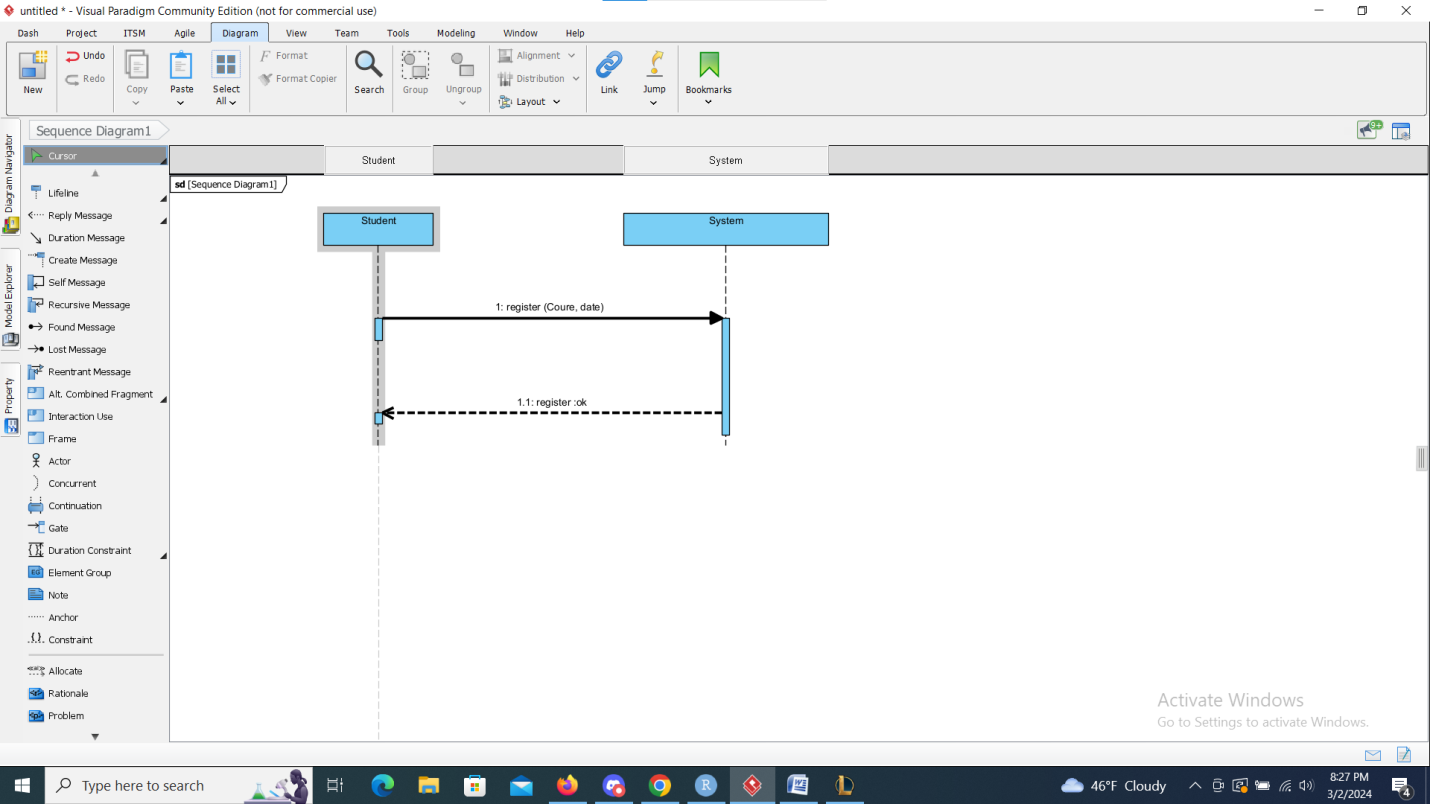
4

1. Diagrama de Clasă:
   * Folosită pentru a reprezenta structura statică a sistemului, inclusiv clasele și relațiile între ele.
2. Diagrama de Obiect:
   * Arată o imagine concretă a stării sistemului la un anumit moment, folosind instanțe de obiecte.
3. Diagrama de Pachete:
   * Grupă elementele sau diagramele conform proprietăților comune, precum coeziunea funcțională.
4. Diagrama de Componente:
   * Folosită pentru a modela componente independente, care furnizează sau utilizează serviciile altor componente.
5. Diagrama de Structură a Compoziției:
   * Permite o decompunere ierarhică a părților sistemului, oferind detalii despre structura internă a claselor sau componentelor.

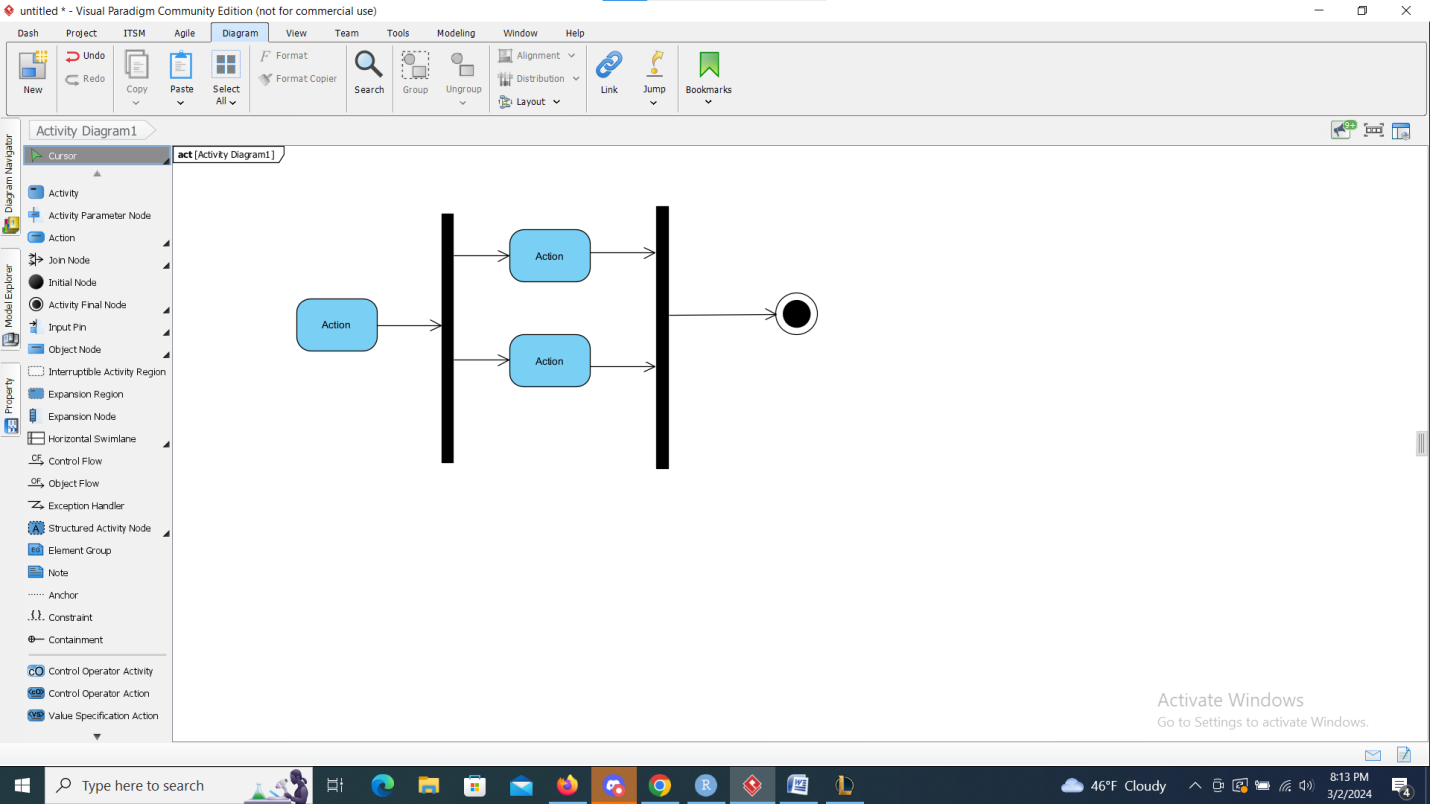
State machine diagram



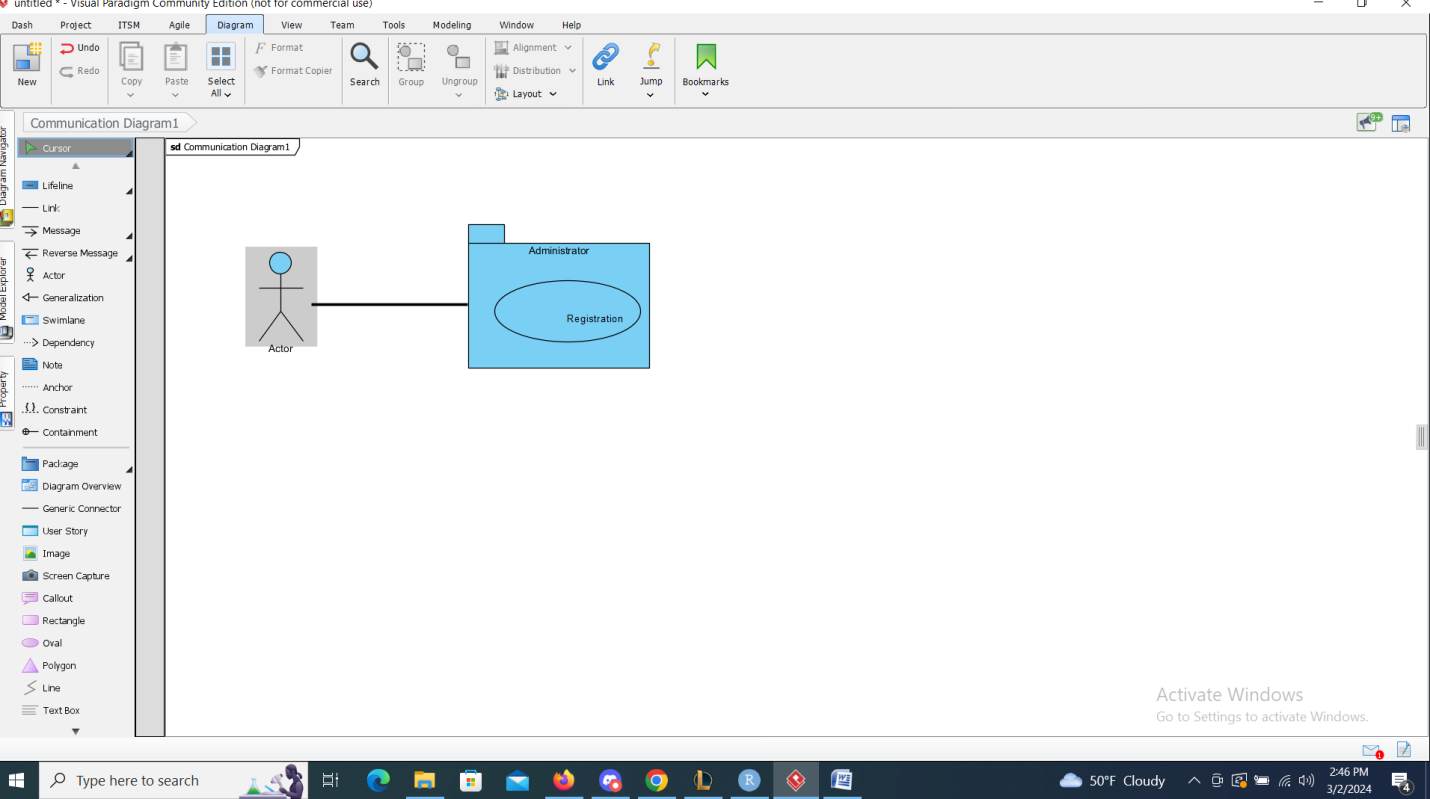
Sequence diagram



Activity diagram



Use case diagram



Class diagram

