Ministerul Educaţiei, Culturii și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Departamentul Ingineria Software și Automatică

**RAPORT**

Lucrare de laborator Nr.6

Disciplina: Analiza Modelarea Sistemelor

Tema: Dezvoltarea elaborărilor prin intermediul diagramelor de stare și diagramelor de activităţi.

A efectuat: st.gr. Vlașițchi Ștefan TI-216,

A verificat : asist.univ.   
 Sava Nina

lect.univ.  
Melnic Radu

Chișinău 2023

**Scopul**:Studierea noțiunilor de stare, stare compusă cu substari concurente, depuse și disjuncte, activitate, condiție de gardă, bară de sincronizare (fork, join), bloc de decizie.

**Sarcina:** de realizat 3 diagrame de stare și 3 diagrame de activitate pentru sistemul informațional ales

**Considerații teoretice**

O diagramă de stare este un instrument utilizat pentru a descrie comportamentul unui sistem din punctul de vedere al stărilor pe care le poate avea și a tranzițiilor între acestea. Ea este formată din stări, tranziții și evenimente care declanșează tranzițiile.

Diagrama începe cu starea inițială, reprezentată de un punct, și apoi definește toate stările posibile și tranzițiile între acestea. În general, o stare este reprezentată printr-un pătrat cu numele stării, iar tranzițiile sunt reprezentate printr-o săgeată care pornește de la o stare și se termină în alta. Săgeata este etichetată cu evenimentul care declanșează tranziția.

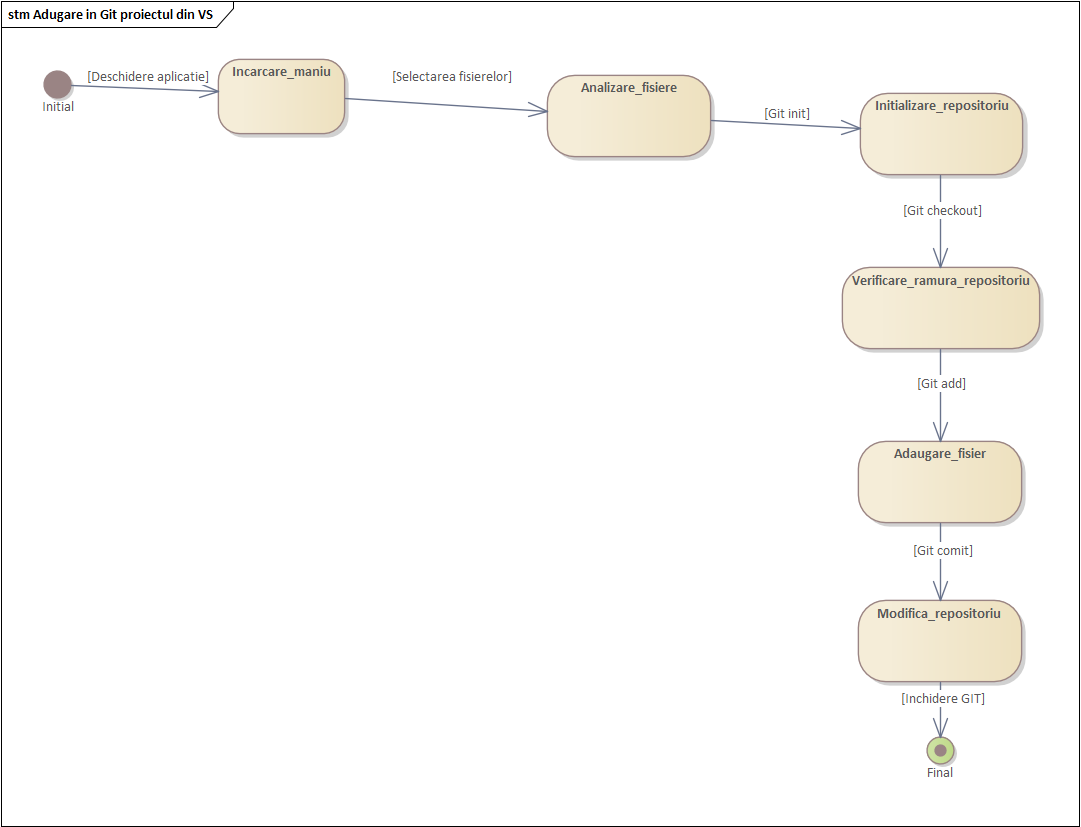
În cazul unui sistem complex, diagrama de stare poate fi foarte mare și poate conține sub-stări și tranziții. De asemenea, pot fi definite acțiuni care se întâmplă în cadrul unei stări sau tranziții, cum ar fi apelul unei funcții sau modificarea unei variabile.

Diagrama de stare este utilă pentru a înțelege comportamentul unui sistem și pentru a identifica eventuale probleme sau incoerențe în comportamentul acestuia. Ea poate fi utilizată în procesul de dezvoltare software pentru a ajuta la proiectarea și implementarea sistemului, dar și pentru a testa și depana aplicația.

O diagramă de activitate este o reprezentare grafică a fluxului de lucru, arătând activitățile și acțiunile care sunt efectuate și deciziile luate în timpul procesului. Aceasta este formată din forme geometrice, cum ar fi ovalul, rombul, dreptunghiul și săgeți, care reprezintă acțiuni, decizii, bifurcări, îmbinări și fluxuri de date. Diagrama de activitate poate fi utilizată pentru a modela diverse procese de afaceri, fluxuri de lucru și algoritmi.

Într-o diagramă de activitate, activitățile sunt reprezentate de obicei de dreptunghiuri, iar săgețile indică direcția fluxului de lucru. Deciziile sunt reprezentate de obicei de romburi și pot fi utilizate pentru a modela bifurcări în proces, în funcție de anumite condiții. Bifurcările și îmbinările pot fi reprezentate de obicei de diamante și indică fluxurile alternative și fuziunea lor. Fluxurile de date pot fi reprezentate prin săgeți cu linie punctată, care arată transmiterea datelor între activități.

Diagrama de activitate poate fi folosită pentru a modela fluxurile de lucru și procesele de afaceri, pentru a identifica și rezolva problemele într-un proces, pentru a îmbunătăți eficiența procesului și pentru a îmbunătăți comunicarea între departamente și echipe. Este important să se acorde atenție detaliilor și să se asigure că diagrama de activitate este ușor de înțeles și de utilizat de către toți cei implicați în procesul modelat.

**Implementare, rezultate practice:**

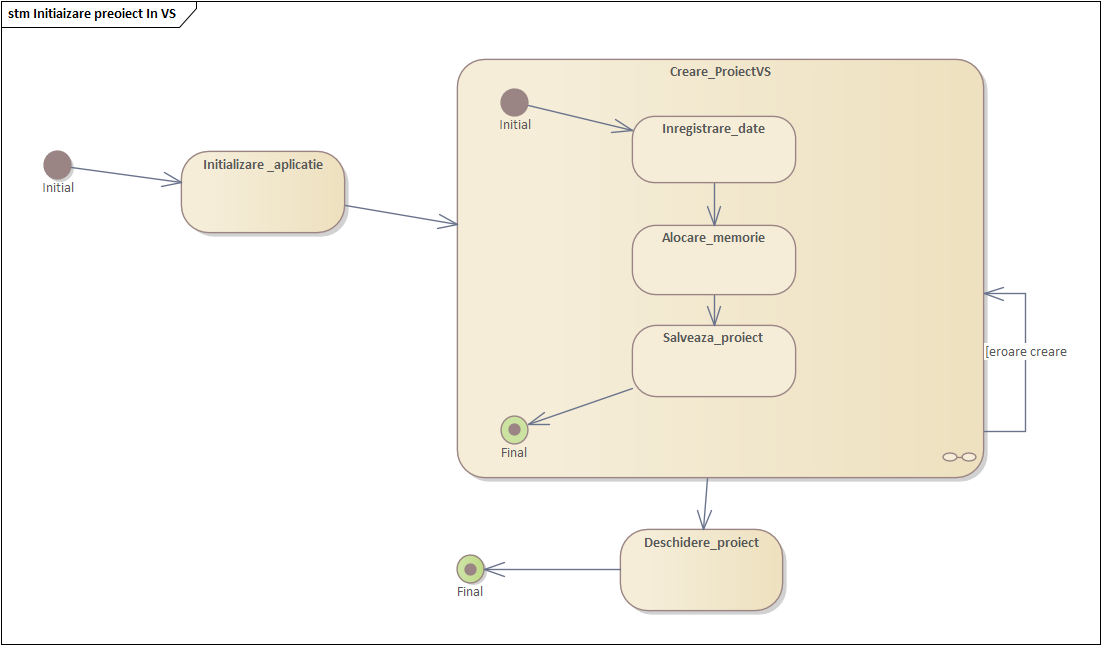
***Figura 1.*** Reprezentarea functionalitaii Adugare in GIT a proiectului din Visual Studio

Diagrama de stare reprezintă fluxul procesului de adăugare a unui proiect în Git. În momentul în care utilizatorul deschide aplicația, acesta va vedea meniul principal, care îi oferă opțiunea de a selecta fișierele pe care dorește să le adauge în Git.

După ce utilizatorul a selectat fișierele, aplicația va începe analiza acestora. În continuare, aplicația va inițializa repositoriul Git, creând astfel un mediu pentru stocarea și gestionarea versiunilor fișierelor.

Următorul pas este verificarea ramurilor repositoriului Git, pentru a asigura faptul că adăugarea de fișiere nu va afecta alte ramuri existente. După ce a fost confirmată siguranța adăugării fișierelor, acestea sunt adăugate la repositoriu.

În final, Git modifică repositoriul, astfel încât fișierele adăugate să fie gestionate și să poată fi accesate și editate de alți utilizatori. Procesul de adăugare a fișierelor în Git se încheie astfel.

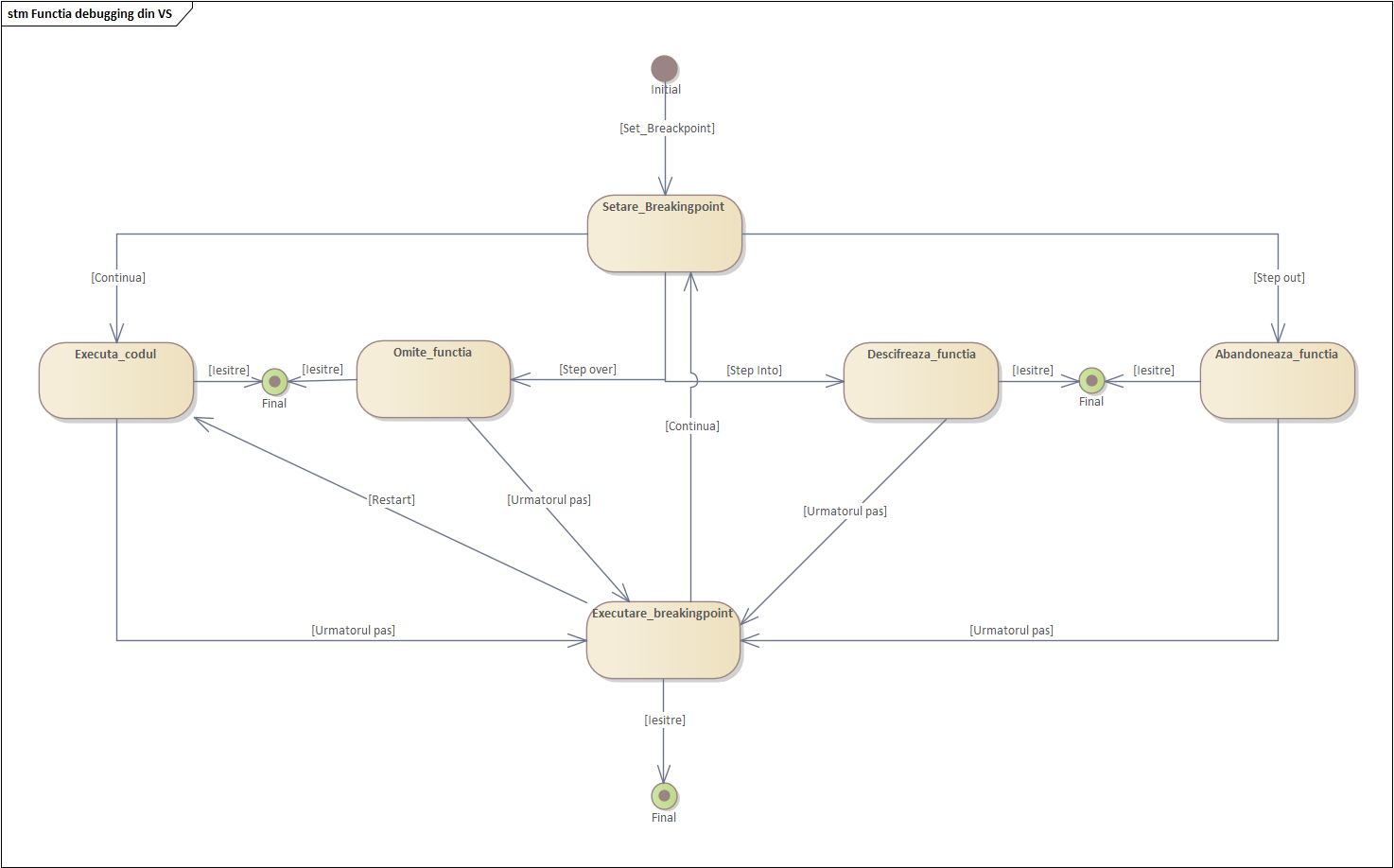


***Figura 2.*** Reprezentare functionalitaii “Initializare proiect”

În figura numărul 2 este reprezentată o diagramă de stare a funcției de inițializare a proiectului în Visual Studio. În momentul în care aplicația este deschisă, se inițializează meniul principal și funcțiile acestuia.

Pentru a crea un proiect nou, utilizatorul va deschide o nouă fereastră în care va introduce datele necesare, precum numele proiectului, locația de stocare, etc. Aplicația va verifica apoi dacă datele introduse corespund cerințelor de creare a proiectului, cum ar fi existența numelor de fișiere sau alocarea suficientă de spațiu de stocare.

În cazul în care datele sunt valide, aplicația va deschide proiectul creat, permițând utilizatorului să lucreze asupra acestuia. În caz contrar, proiectul nu va fi creat și procesul de creare va începe din nou.

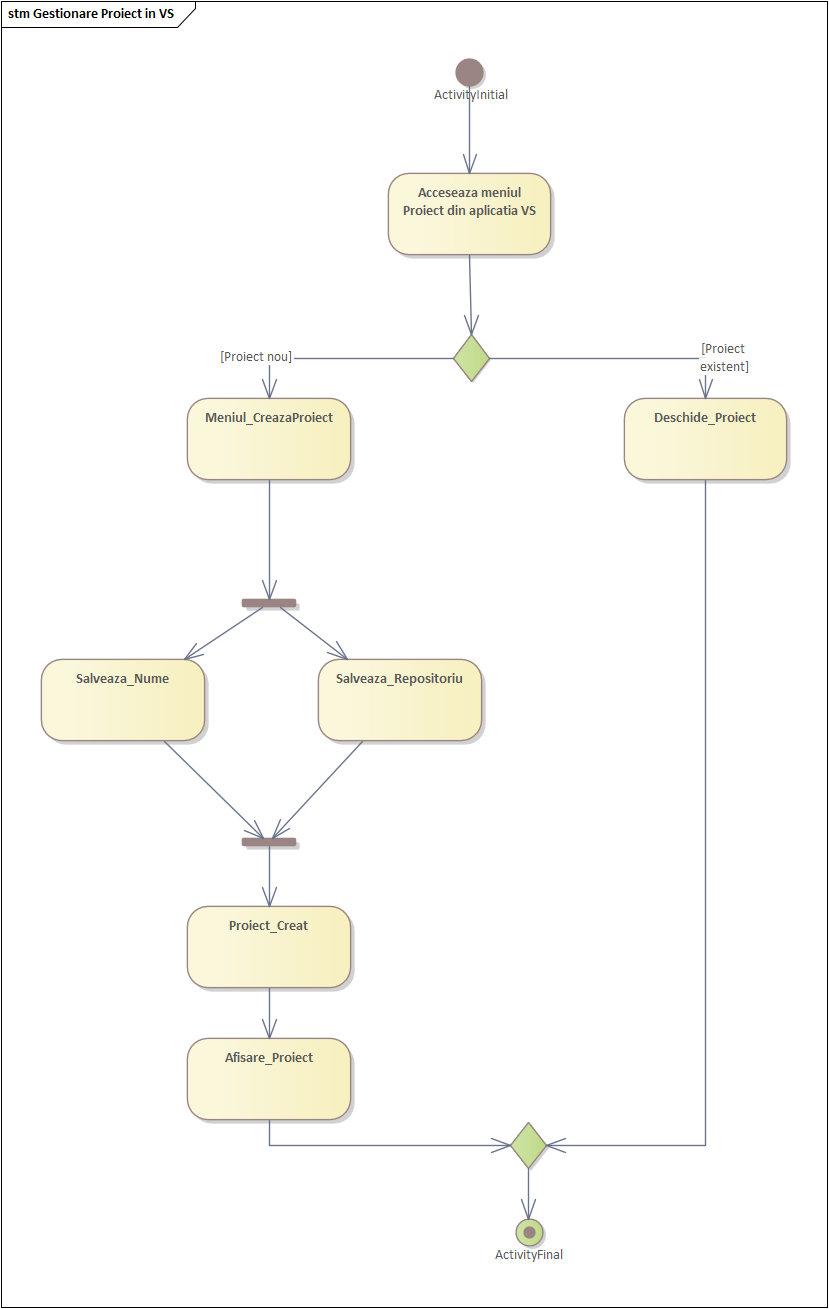


***Figura 3.*** Reprezentare functionalitaii “Debuging in Visual Studio”

În cea de-a treia imagine este reprezentat un proces mai complex, și anume procesul de debugging în Visual Studio. În momentul în care utilizatorul setează punctele de breakpoint, acesta are mai multe opțiuni pentru a continua procesul.

Dacă utilizatorul alege funcția "Continue", procesul va continua până la următorul breakpoint setat, unde utilizatorul poate alege alte funcții precum "Step over", "Step into" sau "Step out". Aceste funcții permit utilizatorului să parcurgă codul și să inspecteze valorile variabilelor în timpul execuției acestuia. Utilizatorul poate opri procesul în orice moment, închizând funcțiile sau alegând să oprească întregul proces de debugging.

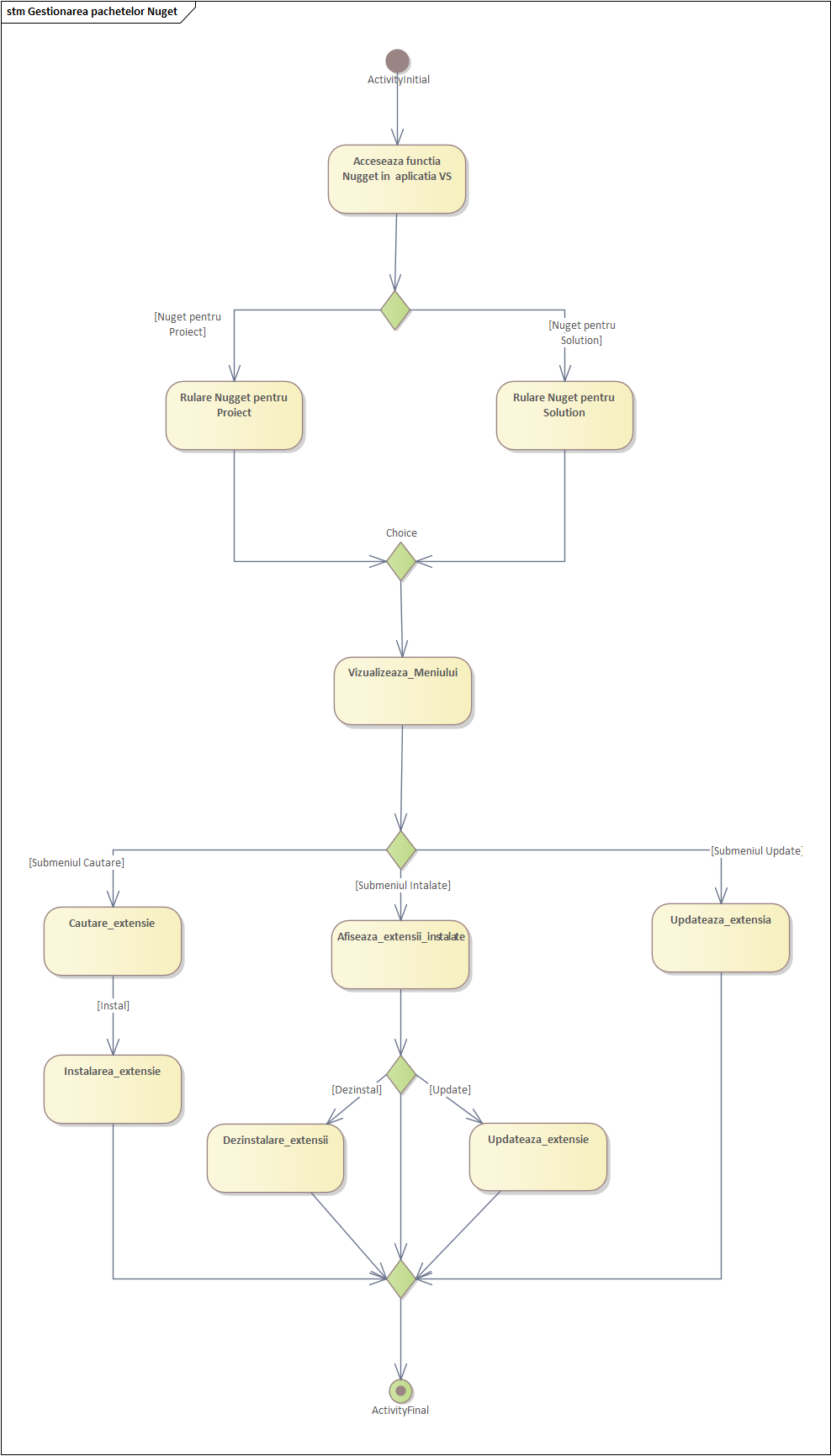
Procesul de debugging poate fi repetitiv, fiind necesară parcurgerea codului de mai multe ori până când se găsește problema și aceasta este remediată



***Figura 4.*** Reprezentare functionalitaii “Gestionarea proiectelor in Visual Studio”

În figura 4 este reprezentată o diagramă de activitate a funcției de gestionare a proiectelor în aplicația Visual Studio. În prima instanță, utilizatorul trebuie să aleagă dacă dorește să creeze un proiect nou sau să deschidă unul existent.

Dacă utilizatorul alege să creeze un proiect nou, două activități vor avea loc în paralel. În primul rând, numele și locația repositoriului vor fi salvate, după care proiectul va fi creat și în cele din urmă va fi afișat. Dacă utilizatorul alege să deschidă un proiect existent, acesta va fi afișat imediat în aplicație.

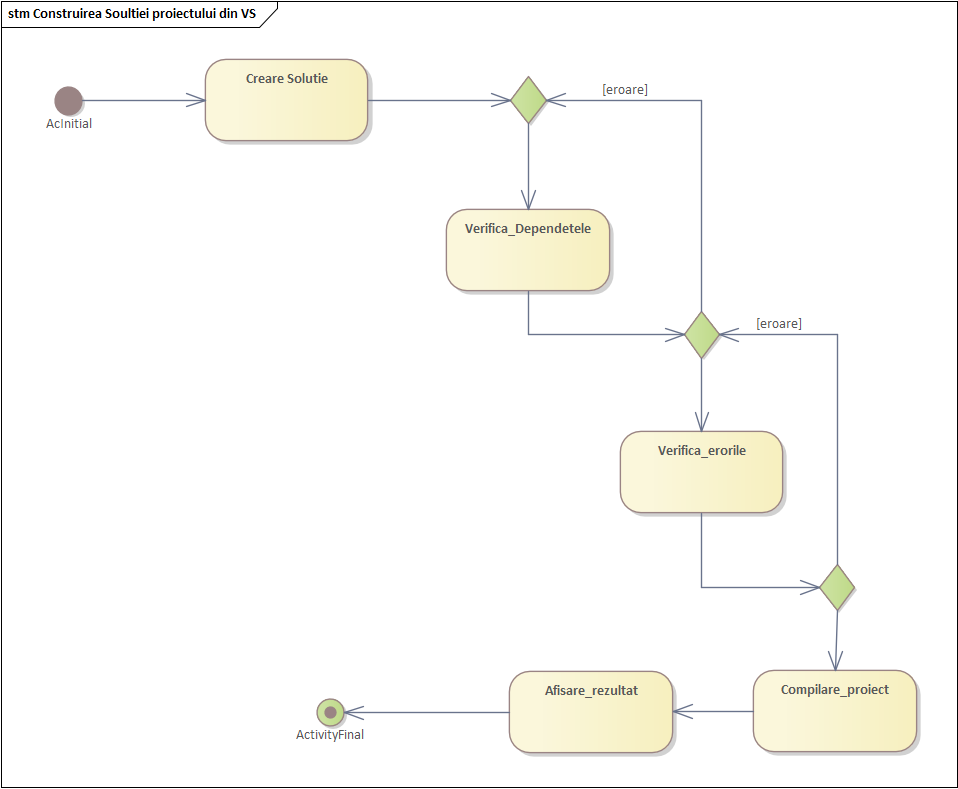


***Figura 5.*** Reprezentare functionalitaii “Gestionarea pachetelor Nugget in Visual Studio”

În figura 5 este prezentat un proces complex de gestionare a pachetelor utilizând funcția Nugget în Visual Studio. Utilizatorul are posibilitatea să instaleze pachete pentru un proiect specific sau pentru întreaga soluție.

Pentru a efectua această operație, utilizatorul trebuie să acceseze meniul funcției Nugget, unde poate alege să caute un pachet. În acest caz, se introduce numele pachetului și se afișează rezultatele căutării. Dacă utilizatorul alege submeniul "Instalate", acesta va vedea o listă cu extensiile instalate și va putea să dezinstaleze, să actualizeze sau să vizualizeze doar extensiile.

În cele din urmă, utilizatorul poate accesa și submeniul care indică dacă există actualizări disponibile pentru extensiile instalate. În acest fel, utilizatorul poate gestiona eficient pachetele utilizate în proiectele sale și poate asigura un mediu de lucru actualizat și funcțional.



***Figura 6.*** Reprezentare functionalitaii “Crearea Solutiei pentru un proiect in Visual Studio”

In figura 6 este reprezentata diagrama de activitati a functionalitatii de creare a solutiei pentru un proiect in aplicatia Visual Studio. La inceputul procesului, se verifica daca toate dependentele sunt corect inregistrate si daca referintele sunt valide. In cazul in care exista o eroare, procesul se repeta pana cand problema este rezolvata. Similar, daca exista erori in codul sursa, procesul se opreste pana cand acestea sunt corectate. Daca nu exista erori, se compileaza proiectul si se afiseaza rezultatul final.

**Concluzii:**

Diagrama de stare și diagrama de activități sunt două instrumente importante în dezvoltarea elaborărilor de software. Diagrama de stare prezintă stările prin care trece un obiect sau sistem în timpul funcționării sale, împreună cu tranzțiile între aceste stări. Această diagramă este utilă în a înțelege modul în care sistemul se comportă și poate fi folosită pentru a identifica problemele și a îmbunătăți performanța sistemului.

Diagrama de activități este un alt instrument important în dezvoltarea elaborărilor de software. Aceasta prezintă fluxurile de activități și decizii pe care le urmează sistemul sau obiectul în timpul funcționării sale. Această diagramă poate fi folosită pentru a identifica punctele cheie în procesul de dezvoltare și pentru a îmbunătăți performanța acestuia.

În ansamblu, diagrama de stare și diagrama de activități sunt instrumente esențiale pentru dezvoltarea elaborărilor de software, permitând dezvoltatorilor să înțeleagă și să optimizeze funcționarea sistemului. Utilizarea acestor instrumente poate contribui semnificativ la îmbunătățirea calității software-ului și la reducerea timpului și costurilor de dezvoltare.

**Bibliografie**

1. **Melnic R., Sava N.** Indrumar metodic “Analiza si modelarea sistemelor informationale”.
2. **Introduction to Visual Studio :** <https://www.geeksforgeeks.org/introduction-to-visual-studio/>
3. **Aplicatia GIT:** <https://github.com/>
4. **Crearea unei diagrame de colaborare UML** : <https://support.microsoft.com/ro-ro/office/crearea-unei-diagrame-de-colaborare-uml-6978fee5-3319-4fbf-8a45-7cfd1062a6d4>
5. **Install NuGet client tools** : <https://learn.microsoft.com/en-us/nuget/install-nuget-client-tools>
6. **State diagram:** [https://www.educba.com/state-diagram/](%20https:/www.educba.com/state-diagram/)
7. **What is Activity Diagram? :** [**https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/**](https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/)
8. **Diagrama de activitate:** <https://creately.com/diagram/example/jqpiv3ht1/diagrama-de-activitate-classic>