Master SIA, anul I, zi

Disciplina Analiza şi proiectarea orientate obiect

**Sistem de gestionare a internshipurilor**

**1. Descrierea cerinţelor sistemului**

În sfera organizației se urmărește dezvoltarea unei soluții pentru administrarea programelor de stagii, care să permită angajaților și departamentelor firmei să acceseze cu ușurință informații despre perioadele de internship, înregistrarea completă a activităților specifice fiecărui stagiu, resursele umane și materiale alocate pentru a facilita condițiile stagiilor. Grupurile de utilizatori ai acestei aplicații sunt: administrator, coordonator al programului de internship, mentor în echipa de stagiu și stagiari. Principalele funcționalități solicitate ale aplicației sunt organizate în patru module distincte: Internship, Echipa, Activitati si Echipamente.

* 1. **Modulul Internship**

În cadrul acestui modul, se vor putea adauga noi stagii, urmari pe cele în desfăşurare şi analiza pe cele finalizate deja. Un internship este realizat atunci cand un post este eliberat, este coordonat de un manager de stagiu (PM) și presupune realizarea mai multor activități. Există mai multe tipuri posibile de stagii. Acestea pot fi pentru studenti sau absolventi, cu norma intreaga sau cu jumate de norma, pe diferite arii de lucru si cu sau fără plată.

 **Adaugă stagiu** - la definirea unui nou stagiu, se vor completa câmpurile: codul şi numele pozitiei de internship, tipul internshipului (platit sau neplatit), numarul de ore saptamanal, id-ul coordornatorului de internship, id-ul echipei, data de început, data de sfârşit, observaţii şi starea internshipului. Un internship se poate afla în una din stările “interview”, “în execuţie”, “abandonat”, „incomplet” şi “finalizat”. În momentul creării internshipului, va fi în starea „interview”. După salvarea internshipului, utilizatorul ar putea continua cu adaugarea de activități pentru acesta.

 **Modifică internship** - pe parcursul ciclului de viaţă a internshipului, coordonatorul poate să modifice datele, inclusiv să schimbe starea acestuia. La crearea sa, internshipul va trece în starea „interview”; el va trece automat în starea „în execuţie” dupa ce este adaugat un intern la acest internship (doar un intern este permis pentru fiecare internship); proiectul poate trece în starea „finalizat” numai dacă toate activitatile alocate sunt în starea „finalizată” data curenta este mai tarzie decat data de sfarsit; un internship poate fi anulat numai dacă se află în una din sterile „interview” sau „în execuţie”, caz în care vor fi anulate toate activitatile aflate „în execuţie”. In cazul in care un internship ajunge are activitati nefinalizate si s-a depasit data de sfarsit, internshipul trece automat in starea „incomplet”. La finalizarea internshipului, coordonatorul trebuie să completeze o lista de observatii in legatura cu activitatile si sa ofere un feedback ce va ajuta compania sa decida pro sau contra angajarii permanente a internului.

 **Anulează internship** – permite abandonarea unui internship din varii motive. Internshipul va trece în starea „abandonat”, iar toate activitățile nefinalizate vor fi anulate (trecute în starea „abandonată”).

 **Vizualizează internshipuri** – se afişează date generale pentru fiecare inernship. Se permite filtrarea rezultatelor in functie de starea proiectului, coordonator, mentor, pozitie, etc. La selectarea unui internship, se vor afisa si detaliile in legatura cu internul asignat la acel internship cat si activitatile pe care le deruleaza.

**1. 2) Modulul Activitati**

În acest modul se gestionează activitatile aferente internshipurilor create în sistem. Aceste operaţiuni pot fi realizate doar de către coordonatorul de internship. O activitate se poate afla in urmatoarele stari: „propunere”, „in executie”, „abandonat”, „finalizat” „acceptată”, „respinsă”.

 **Creare activitate** – crearea unei activitati noi poate fi realizată pe parcursul derulării internshipului, atât timp cât el se află „în execuţie”. La crearea unei activitati trebuie introduse denumirea ei, o descriere, data de creare, termenul de finalizare, codul internshipului şi numarul orelor-om estimate pentru realizarea ei. La momentul creării o activitate va intra în starea „propunere”. Prin codul internshipului, aceasta activitate este repartizata internului asignat pe acel internship.

 **Asumare activitate** – Internul are responsabilitatea de a consulta informatiile despre activitatile propuse in cadrul internshipului si a completa toate activitatile intr-un mod eficient pentru a respecta termenele de finalizare. Internul are responsabilitate de a-si gestiona singur timpul. In momentul in care o activitate este inceputa, internul va consemna data de inceput a activitatii.

 **Modificare activitate** – coordonatorul de internship poate modifica datele privind activitatile create. Numărul orelor-om estimat nu mai poate fi modificat după începerea realizării activității. Dupa finalizare, o activitate este supusă unui process de acceptare de către coordonatorul de internship care poate valida rezultatele sau le poate respinge. La finalizarea unei activitati, managerul proiectului va completa rubrica de observatii din cadrul activitatii, ce va fi utilizata la evaluarea performantei internului.

 **Vizualizare activitati** – vizualizarea activitatilor se face dupa selectarea unui internship. La deschiderea paginii vor fi afisate doar activitatile curente (nefinalizate), ordonate dupa termenul de finalizare. In functie de optiunile utilizatorului, vor putea fi afisate toate activitatile sau filtrate in functie de stare. La vizualizarea activitatilor, se va colora in mod intuitiv pentru a evidentia starea activitatilor.

**1.3) Modulul Echipa**

În acest modul se vor gestiona datele privind angajaţii care au primit in echipa lor internul. Se va facilita lucrul in echipa prin prezentarea colegilor, a modalitatii de contact preferata si a cadrului lor de activitate si expertiza. Astfel, internul va avea acces facil la resursele umane din cadrul echipei.

 **Adaugă in echipa** – se adaugă datele pentru un membru nou, precum numele, funcţia, stare functionala (intern, coordonator de internship, membru), modalitatea de contact preferata, cadrul de activitate, expertiza, cunostinte utile, observatii. De asemenea, se poate adauga un membru in echipa folosind datele deja existente, doar prin legarea cu id-ul angajatului.

 **Modifică date** – permite actualizarea datelor privind angajaţii.

 **Şterge** – permite eliminarea unui angajat din cadrul echipei.

 **Vizualizează date membru** – se afişează o listă cu angajatii din echipa și datele despre aceștia. La cererea utilizatorului se pot afișa detalii privind activitățile repartizate unei persoane.

**1.4) Modulul echipamente**

În acest modul se realizează gestiunea echipamentelor necesare pentru derularea fiecărui internship. Sunt incluse echipamente precum: laptop, pc, monitor, incarcator, docking station, mouse, tastatura, casti, scaun de birou.

 **Adăugă echipament** – se adauga datele pentru echipamentul respectiva precum tip, nume, serial number, responsabil (null daca nu a fost asignat nimanui).

 **Atribuie echipament** – se permiteatribuirea echipamentului unei persoane responsabile.

 **Modifică date echipament** – se permite schimbarea datelor legate de resurse.

 **Vizualizare echipamente** – dupa selectarea unui internship se afișează toate echipamentele asignate acelui internship. Astfel, la sfarsitul internshipului se pot verifica toate echipamentele folosite.

1. **Arhitectura de referinta selectată pentru acest sistem**

Arhitectura care va fi utilizată la implementarea aplicaţiei conţine următoarele straturi (fig. 1)

1. Presentation Layer (UI cu .NET framework winforms application) – afisare informatii, interactiuni cu utilizatorul, transformare evenimente UI in apeluri catre celelalte straturi ale aplicatiei. Acesta este stratul vizual al aplicației în care utilizatorii interacționează cu aplicația.
2. Domain Layer (separate .NET project) – Acest strat conține logica de afaceri a aplicației și este responsabil pentru procesarea și manipularea datelor. Este separat de stratul UI. Aici sunt gestionate cererile și răspunsurile către/dinspre API-ul REST și bază de date.
3. Data Layer (API REST cu Java Spring Boot) – contine clasele cu rol de asigurare a accesului la stratul de persistenta.
4. Data Layer (Baza de Date PostgreSQL) – Acesta este stratul de stocare a datelor care păstrează datele aplicației. PostgreSQL este utilizat ca sistem de gestionare a bazelor de date și stochează datele în tabele structurate.

Interactiuni intre straturi (fig. 2):

1. Utilizatorul interacționează cu aplicația WinForms, efectuând cereri sau acțiuni, cum ar fi obținerea sau modificarea datelor.
2. Interfața WinForms trimite cererile către stratul de Logică de Afaceri pentru procesare.
3. Stratul de Logică de Afaceri utilizează API-ul REST pentru a efectua cereri către Java Spring Boot, care gestionează datele din bază.
4. Java Spring Boot interacționează cu baza de date PostgreSQL pentru a obține sau actualiza datele necesare.
5. Datele obținute sunt apoi procesate și trimise înapoi către stratul de Logică de Afaceri, care le returnează stratului UI.
6. Interfața WinForms afișează datele actualizate sau răspunsurile utilizatorului.

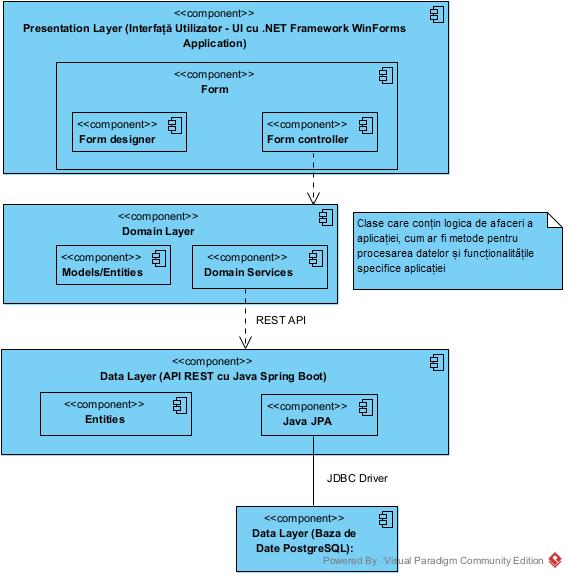
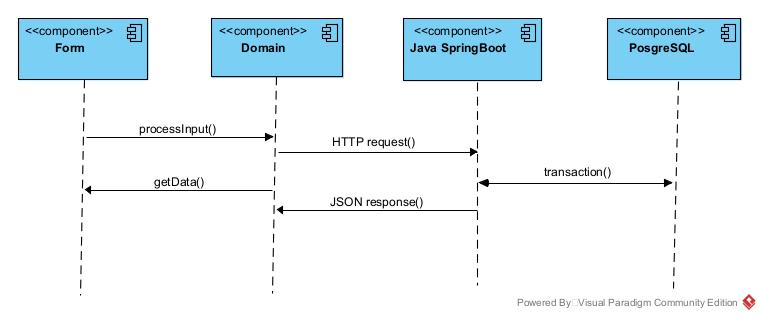


Figura 1 Arhitectura de referinţă a aplicaţiei

Figura 2 Interacţiunile dintre straturile arhitecturii de referinţă

1. **Structurarea cerinţelor funcţionale ale aplicației**

Pentru structurarea cerinţelor funcţionale se vor utiliza diagrama de activitiăţi şi diagrama cazurilor de utilizare. Diagrama de activităţi poate fi folosită pentru descrierea logicii proceselor economice, respectiv a principalelor funcţionalități ale aplicaţiei și a succesiunii în care ele sunt realizate (fig. 3); această diagramă va sta la baza identificării principalelor cazuri de utilizare. De asemenea, diagrama de activități poate fi utilizată pentru descrierea logicii cazurilor de utilizare sau chiar scenariilor de lucru ceva mai complexe.

**3.1) Diagrama de activităţi pentru descrierea procesului**

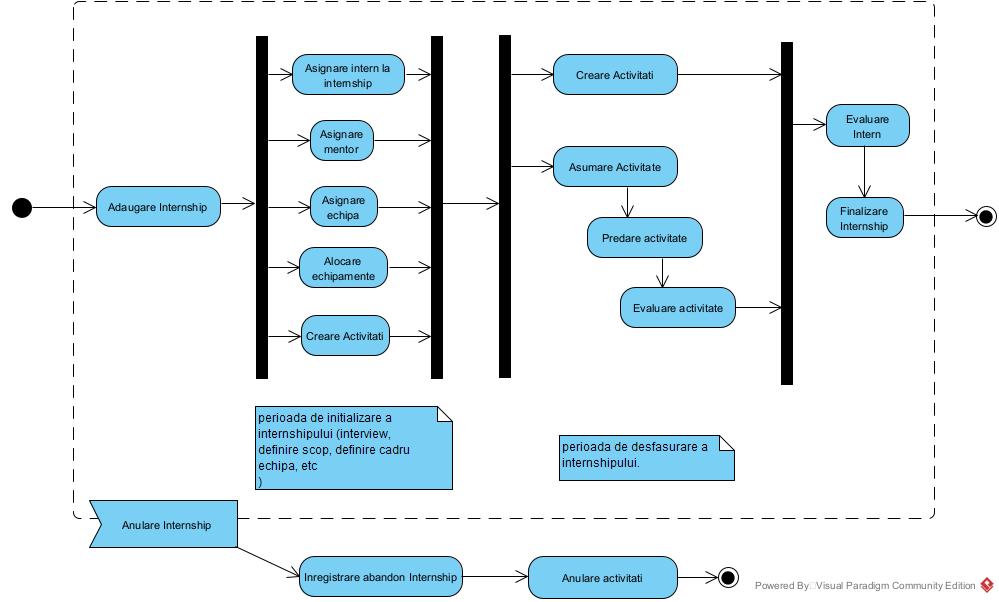


Figura 3 Diagrama de proces pentru managementul proiectelor

**3.2) Diagramele cazurilor de utilizare**

Avand în vedere complexitatea funcţională a aplicaţiei, s-a pornit cu crearea unei diagrame de pachete (fig. 4), în care se realizează partiționarea funcționala a aplicației. Diagrama prezintă cele 4 funcţiuni principale ale aplicaţiei, sub forma de pachete. Crearea ei a fost necesară pentru a controla complexitatea aplicaţiei, dat fiind că diagramele cazurilor de utilizare nu pot fi descompuse ca şi diagramele fluxurilor de date.

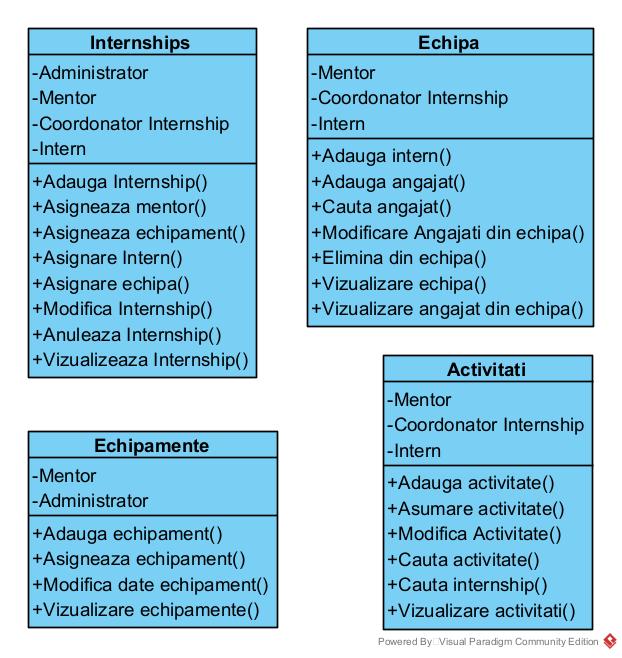
**()()()(pt internships „cauta internship”**

Figura 4 Funcţiunile aplicaţiei

Pentru fiecare dintre cele patru funcţiuni principale descrise în primul capitol și evidențiate în diagrama de pachete se va construi câte o diagramă a cazurilor de utilizare. În continuare vor fi prezentate cele 4 diagrame (fig. 5-8).

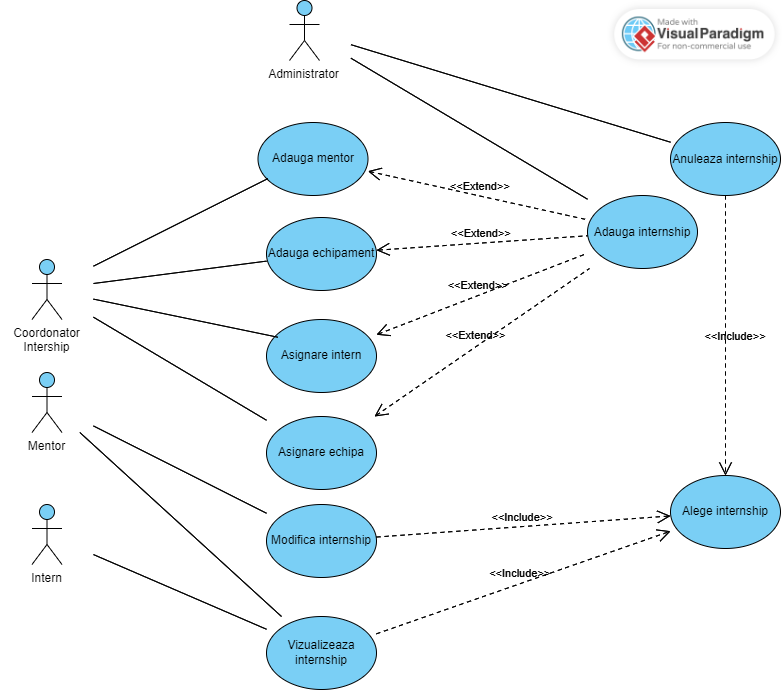
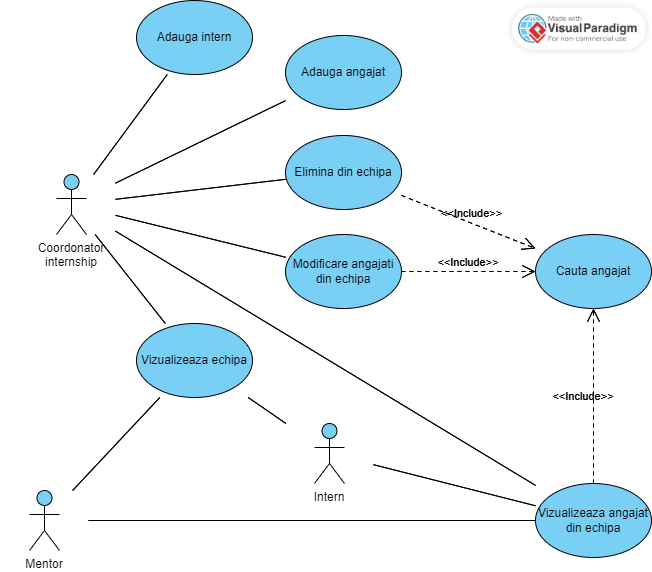
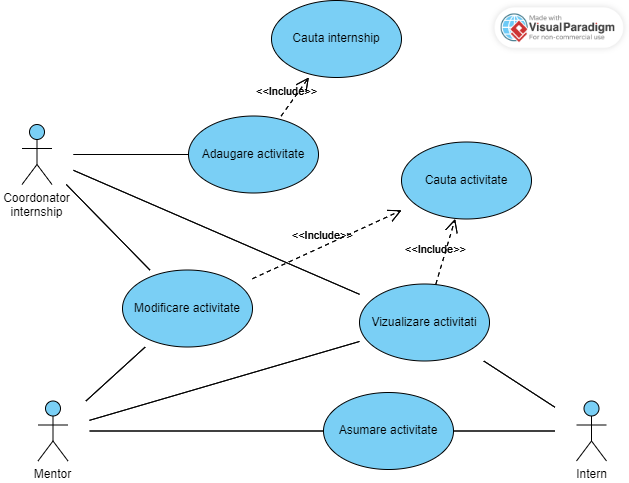
****

Figura 5 DCU pentru modulul Internships



Include adaug ang in cauta++++++++++++++++++++

Figura 6 DCU pentru modulul Echipe



La fel+++++++++++++++++++++

Figura 7 DCU pentru modulul Activitati

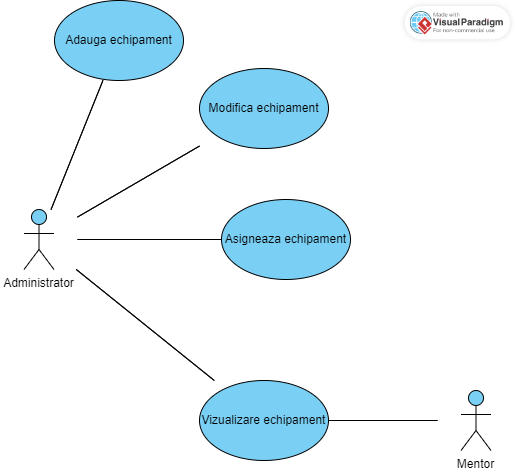


Figura 8 DCU pentru modulul Echipamente

3.3) Descrierea narativă a cazurilor de utilizare

**Numerotare**: CU1

**Nume CU**: Adauga Internship

**Obiectiv**: Inregistrarea cu succes in sistem a unui internship nou

**Actorii**: Administratorul

**Prioritate**: 1

**Preconditii**: Utilizatorul este conectat cu un cont care are drepturi de administrator

**Post-conditii**: Internshipul este inregistrat in sistem

**Scenariul principal**: Inregistrarea cu succes a noului internship, continand un cod unic si detalii specifice pentru internship(nume pozitiei, tip internship, numar de ore, data inceput, data sfarsit)

**Scenarii de exceptie**: Abandonarea adaugarii de internship

**Pasii scenariului principal:**

1. Utilizatorul initiaza operatiunea de adaugare a internshipului

2. Utilizatorul introduce numele pozitiei, tipul de internship, numarul de ore, data de inceput si data de sfarsit pentru internship

3. Utilizatorul initiaza procesul de salvare a internshipului

4. Sistemul valideaza informatiile (daca datele introduse nu sunt corecte, se reia de la pasul 2)

5. Un cod unic pentru internship este generat

6. Sistemul salveaza informatiile in baza de date

7. Sistemul afiseaza datele internshipului salvat si un mesaj ce confirma salvarea

**Pasii scenariului principal:**

Daca in oricare moment inainte de pasul 6 al scenariului principal, utilizatorul abandoneaza operatiunea de adaugare, atunci:

1. Sistemul avertizeaza utilizatorul ca datele introduse nu vor fi salvaate.
2. Cazul de utilizare se incheie, fara salvarea datelor introduse.

**Numerotare**: CU2

**Nume CU**: Vizualizare activitate

**Obiectiv**: Vizualizarea unei activitati de catre intern

**Actorii**: Intern

**Prioritate**: 1

**Preconditii**: 1. Utilizatorul este conectat cu un cont care are drepturi de intern

2. Activitatea ce se doreste a fi vizualizata este asignata internului conectat

**Post-conditii**: Internul vizualizeaza informatii despre activitate

**Scenariul principal**: Vizualizarea cu succes a informatiilor specifice unei activitati.

**Scenarii de exceptie**: 1. Abandonarea procesului

2. Activitatea nu poate fi gasita in sistem

**Pasii scenariului principal:**

1. Utilizatorul deschide modulul de activitati al sistemului

2. Utilizatorul initiaza cautarea unei activitati

3. Utilizatorul introduce numele activitatii in zona de cautare

4. Sistemul cauta in, baza de date, activitatea care corespunde acelui nume

5. Sistemul afiseaza informatii despre activitatea gasita

**Pasii scenariului exceptie 1:**

Daca in oricare moment inainte de pasul 4 al scenariului principal, utilizatorul abandoneaza operatiunea de vizualizare, atunci:

1. Cazul de utilizare se incheie, fara vizualizarea datelor despre activitate

**Pasii scenariului exceptie 2:**

Daca in pasul 3 al scenariului principal, activitatea nu poate fi gasita, atunci:

1. Sistemul afiseaza un mesaj ce mentioneaza ca activitatea nu a fost gasita.
2. Se reia procesul de la pasul 2 al scenariului principal.

**3.4 Descrierea cazurilor de utilizare (comportamentului aplicației) folosind limbajul Gherkin**

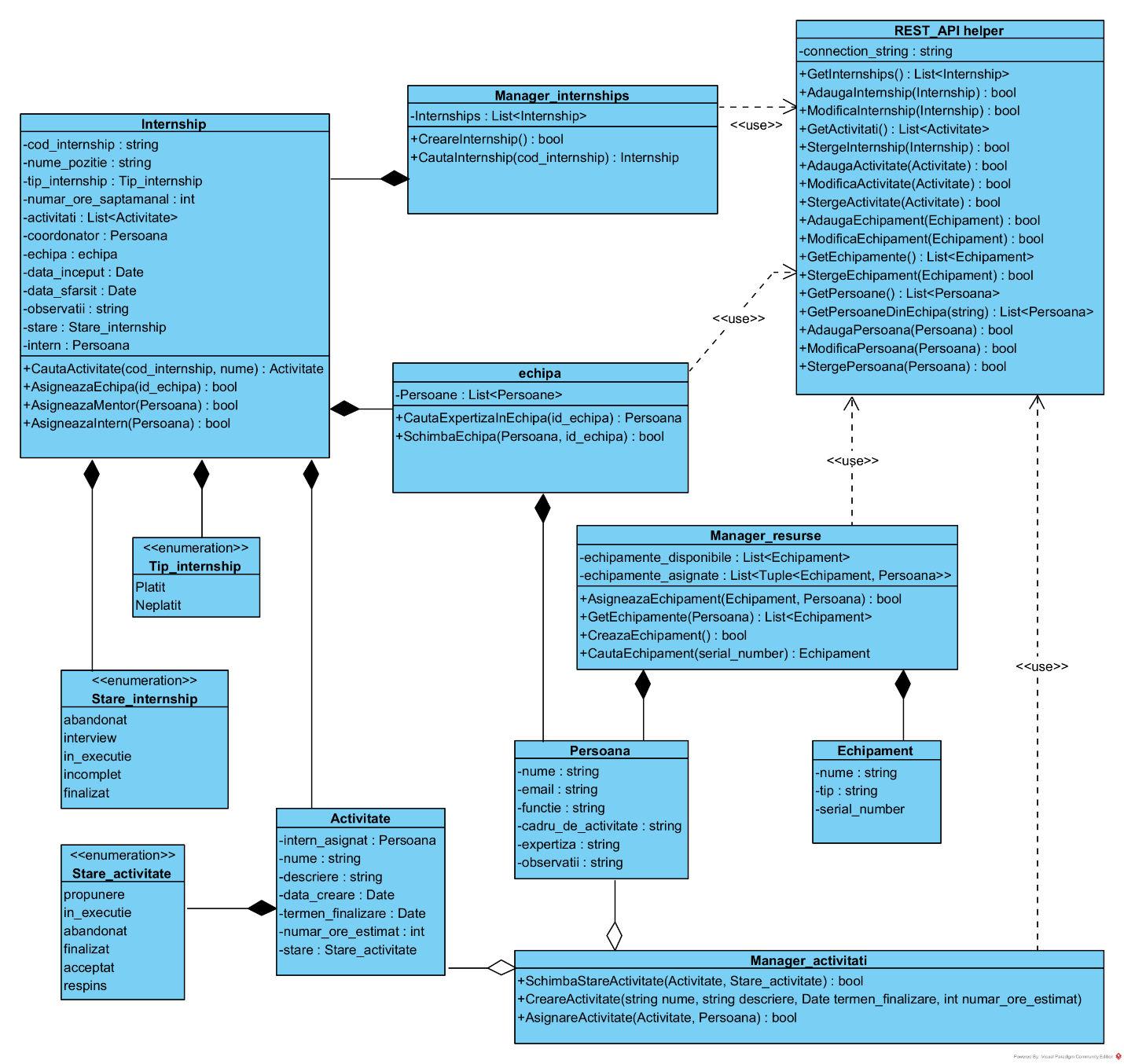
|  |
| --- |
| **Scenario**: Adauga Internship  **Given** Utilizatorul este conectat cu un cont care are drepturi de administrator  **When** Utilizatorul initiaza operatiunea de adaugare a internshipului  **And** Utilizatorul introduce numele pozitiei, tipul de internship, numarul de ore, data de inceput si data de sfarsit pentru internship  **And** Utilizatorul initiaza procesul de salvare a internshipului  **And** Sistemul valideaza informatiile (daca datele introduse nu sunt corecte, se reia de la pasul 2)  **And** Un cod unic pentru internship este generat  **And** Sistemul salveaza informatiile in baza de date  **Then** Sistemul afiseaza datele internshipului salvat si un mesaj ce confirma salvarea |

|  |
| --- |
| **Scenario**: Vizualizeaza activitate  **Given** Utilizatorul este conectat cu un cont care are drepturi de intern  **And** Activitatea ce se doreste a fi vizualizata este asignata internului conectat  **When** Utilizatorul deschide modulul de activitati al sistemului  **And** Utilizatorul initiaza cautarea unei activitati  **And** Utilizatorul introduce numele activitatii in zona de cautare  **And** Sistemul cauta in, baza de date, activitatea care corespunde acelui nume  **Then** Sistemul afiseaza informatii despre activitatea gasita |

**4. Structura aplicaţiei. Diagrame de clase**

**Domain Model (Business Layer)**

Este prezentată diagrama de clase specifică domeniului problemei (domain model). Aici au fost incluse următoarele grupe de clase:

* Clasele de tip Entity care mapează obiectele din domeniul problemei (business entities) (clase precum Internship, Echipa, Activitati). Aceste clase vor cuprinde doar atributele (starea) şi metodele care gestionează valorile atributelor (get-uri, set-uri şi metodele pentru gestionarea atributelor de tip colecţie). De dragul simplității, metodele get și set nu au fost incluse în model.
* Clasele care implementează funcţionalitatea specifică domeniului problemei (business logic sau service objects în Domain Driven Design). Astfel de clase sunt necesare atunci când există unele operațiuni (comportament) specifice domeniului problemei (unele verbe) care nu aparțin strict unor entități (clase de tip Entity), deci nu pot fi incluse în acestea. Așadar, scopul claselor din acest grup este doar de a oferi funcționalități specifice domeniului și vor face referire la clasele de tip Entity care conțin datele necesare în implementarea funcționalităților. La modul general, clasele din acest grup vor implementa operațiuni de prelucrarea, transformarea și managementul datelor aplicației, implementarea regulilor afacerii (business rules) sau asigurarea validității datelor din aplicație
* Clasele de tip enumeration – StareActivitate și StareProiect, care conţine valorile posibile pentru atributul stare din clasele Activitate și Proiect. 
* Figura 9: Diagrama de clase pentru modulul echipamente din Presentation Layer

Presentation Layer

Este prezentată diagrama de clase care conține clasele specifice stratului interfeței utilizator (Presentation Layer). În acest strat sunt incluse clasele care implementează interfata potrivita pentru winforms application. Folosind winforms, view-urile sunt primele ce clase ce sunt create, urmand sa fie create controller si model pentru view. Pentru fiecare actiune a utilizatorului in view, aceasta este trimisa controllerului pentru a fi procesata si actualizata in model. Controllerul are rolul de a incarca date si a completa task-uri consumatoare de timp.

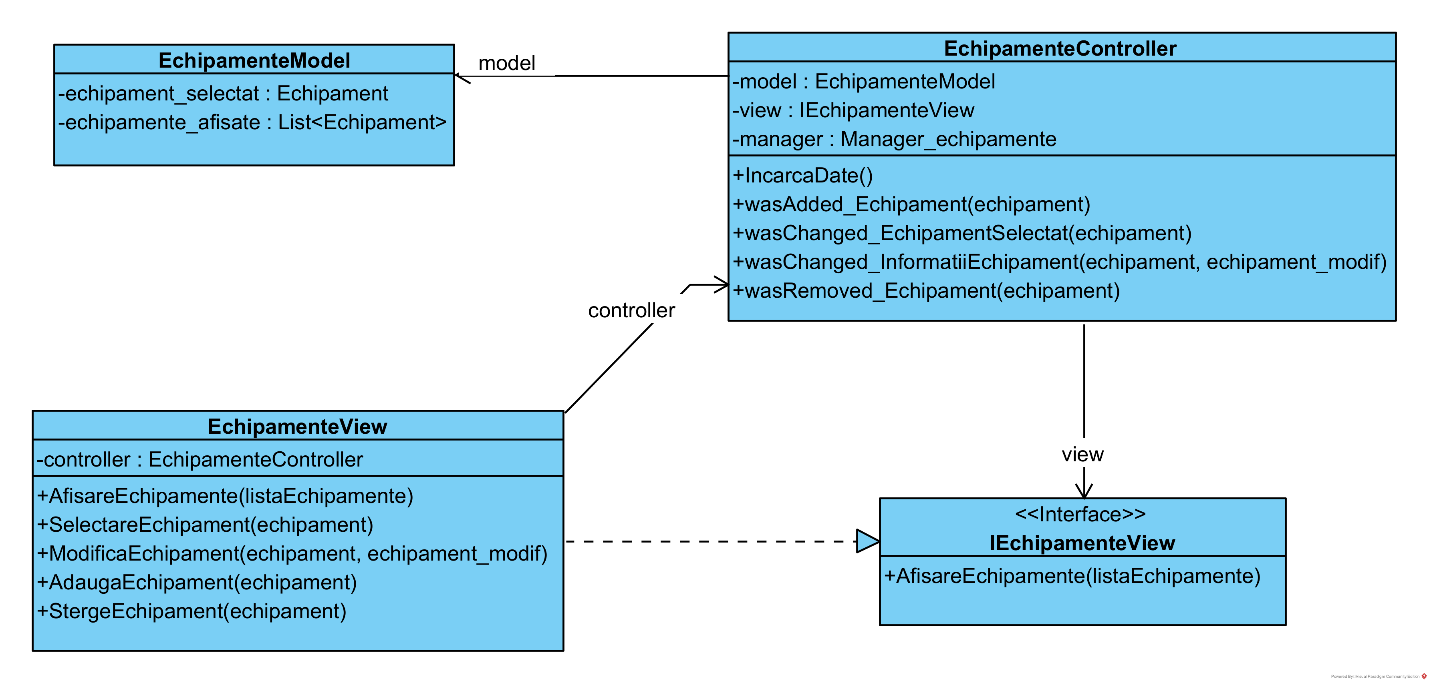


Figura 10: Diagrama de clase pentru modulul echipamente din Presentation Layer

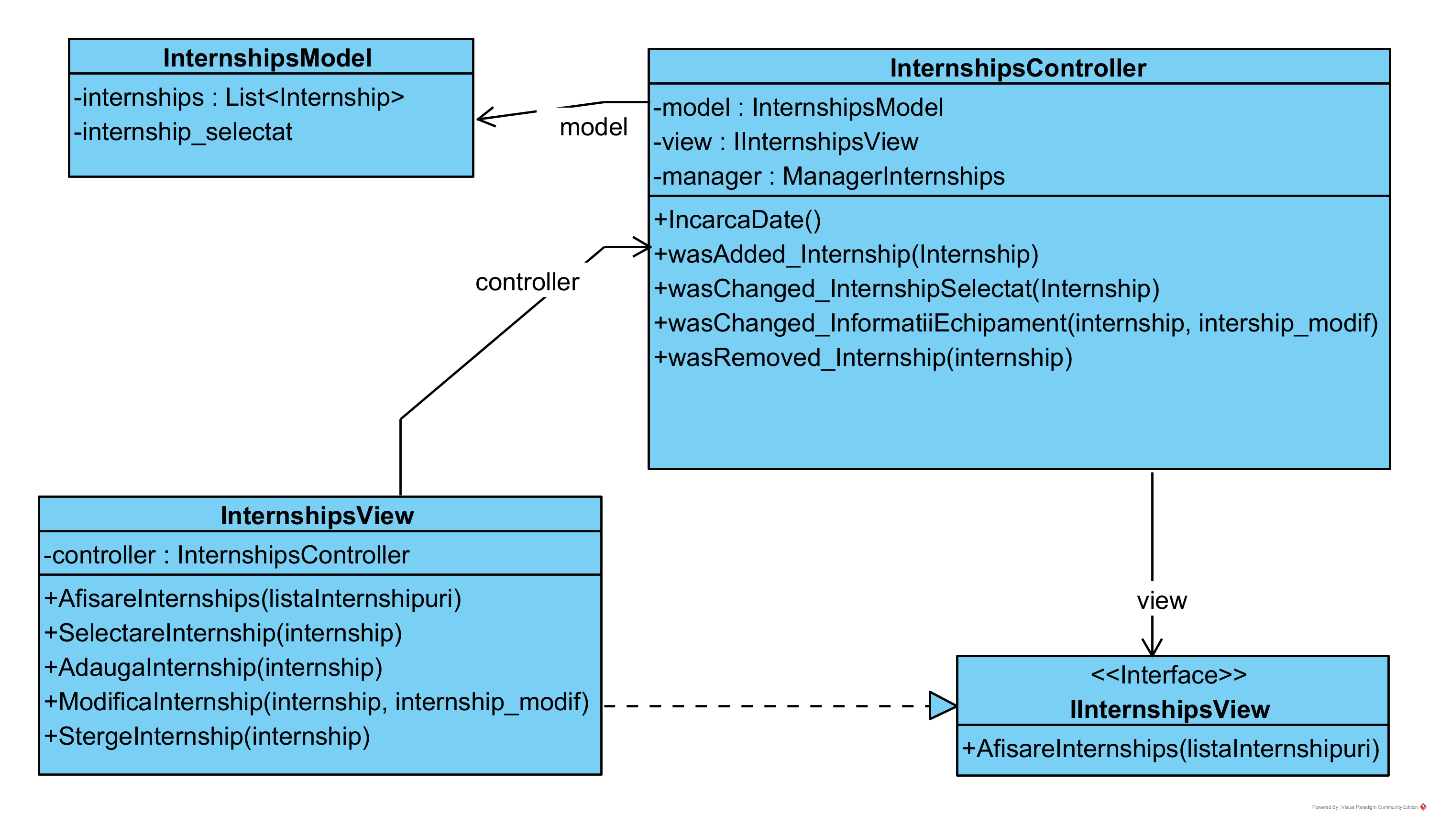


Figura 11: Diagrama de clase pentru modulul Internship din Presentation Layer

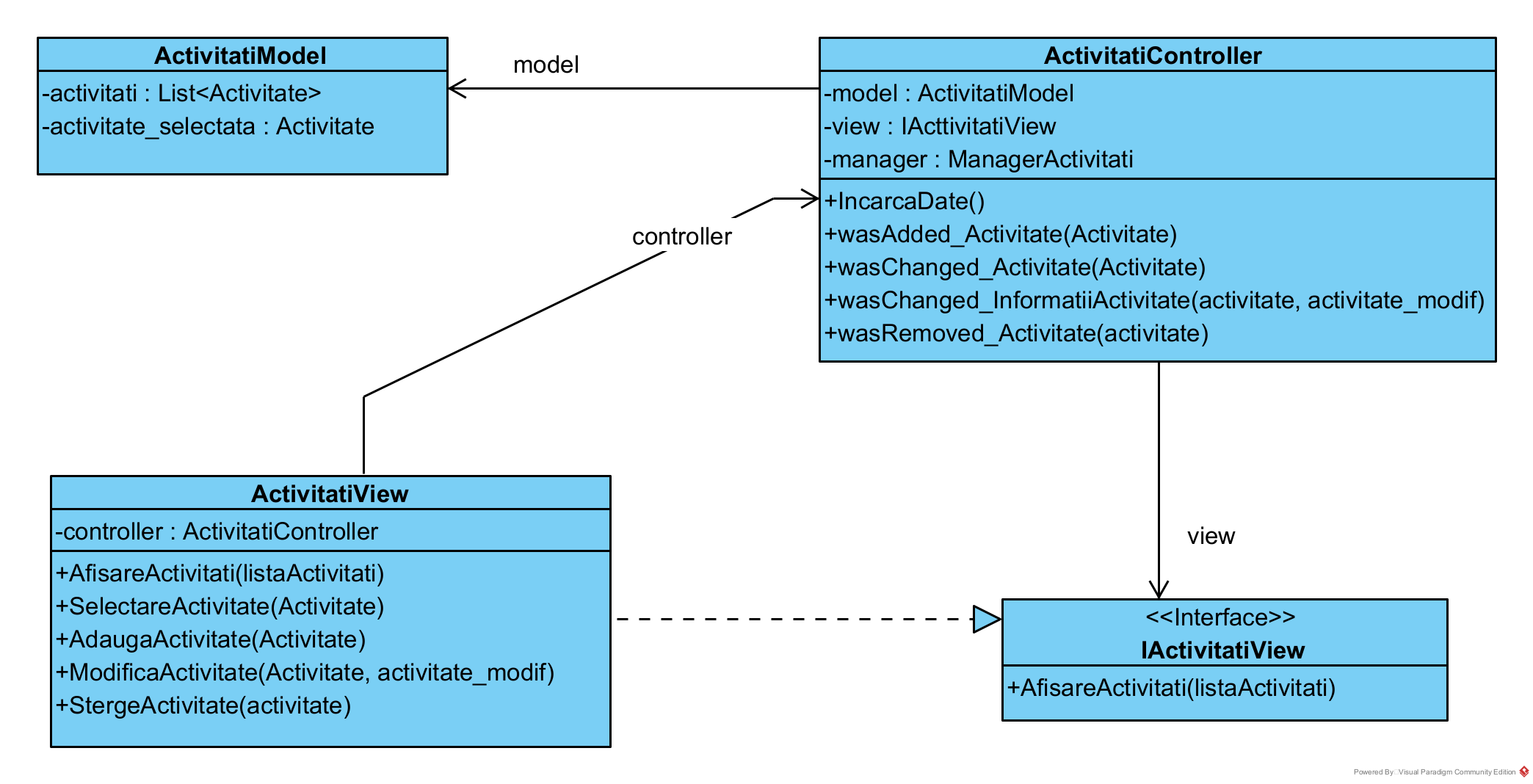
ewqeqwe1241212312312312rf23f3223rg

Figura 12: Diagrama de clase pentru modulul Activitati din Presentation Layer