UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

PROIECTARE SOFTWARE
Documentație proiect
Grădina Botanică
Arhitectura MVVM și principiile GRASP

Student: Adela Iosif
Grupa: 30233

2023-2024

Cuprins

1. Introducere

1.1. Obiectiv

Obiectivul acestei teme este familiarizarea cu șablonul architectural ***Model-View-ViewModel.***

Pentru persistența informației se va utiliza o bază de date relațională (SQL Server, MySQL, etc.).

1.2. Cerințe

❖ În ***faza de analiză*** se va realiza **diagrama cazurilor de utilizare**.

❖ În faza de proiectare se va realiza diagrama de clase respectând arhitectura MVVM (care utilizează șablonul de proiectare comportamental ***Command***) și principiile GRASP, dar și diagrama entitate-relație corespunzătoare bazei de date.

❖ În ***faza de implementare*** se va scrie cod pentru îndeplinirea tuturor funcționalităților precizate de diagrama cazurilor de utilizare utilizând:

- proiectarea dată de diagrama de clase;
- unul dintre următoarele limbaje de programare: C#, C++, Java, Python.

❖ Finalizarea temei va consta în predarea unui director ce va cuprinde:

- Un fișier cu diagramele UML realizate;
- Baza de date;
- Aplicația soft;
- Documentația (minim 10 pagini) - un fișier care cuprinde: ▪ numele studentului, grupa;
  - enunțul problemei;
  - instrumente utilizate;
  - justificarea limbajului de programare ales;
  - descrierea diagramelor UML;
  - descrierea aplicației.

1.3. Enunțul problemei

Dezvoltați o aplicație care poate fi utilizată într-o **grădină botanică**. Aplicația va avea 3 tipuri de utilizatori: vizitator al grădinii botanice, angajat al grădinii botanice și administrator.

Utilizatorii de tip **vizitator** pot efectua următoarele operații fără autentificare:

❖ Vizualizarea listei tuturor plantelor din grădina botanică sortată după tip și specie;

❖ Filtrarea listei plantelor după următoarele criterii: tip, specie, plante carnivore, zona grădină botanică.

❖ Căutarea unei plante după denumire.

Utilizatorii de tip **angajat** al grădinii botanice pot efectua următoarele operații după autentificare:

❖ Toate operațiile permise utilizatorilor de tip vizitator;

❖ Operații CRUD în ceea ce privește persistența plantelor din grădina botanică.

❖ Salvare liste cu plantele în mai multe formate: csv, json, xml, doc.

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

❖ Toate operațiile permise utilizatorilor de tip vizitator;

❖ Operații CRUD pentru informațiile legate de utilizatorii care necesită autentificare;

❖ Vizualizarea listei utilizatorilor care necesită autentificare.

2. Analiza problemei

2.1. Diagrama cazurilor de utilizare

În diagrama de use case (figura 1) se pot identifica cei trei actori: vizitator, angajat și admin. Astfel, vizitatorii sunt persoane nelogate care pot accesa aplicația și pot vizualiza lista plantelor din grădina botanică sortată după tip și specie, dar pot opta și pentru filtrarea listei plantelor după anumite criterii: tip, specie, plante carnivore, zonă grădină, având posibilitatea și de a căuta o plantă după nume. Mai mult, actorii de tip angajat și admin se pot loga, apoi având posibilitatea de efectuare a operațiilor corespunzătoare vizitatorului, alături de altele suplimentare specifice fiecărui rol. Astfel, utilizatorul de tip angajat are ca operații specifice salvarea listei cu plante în diferite formate (csv, json, xml, doc) și operațiile CRUD pe plantele din grădina botanică. Administratorul are ca acțiuni specifice rolului operațiile CRUD pe utilizatori și vizualizarea listei acestora.

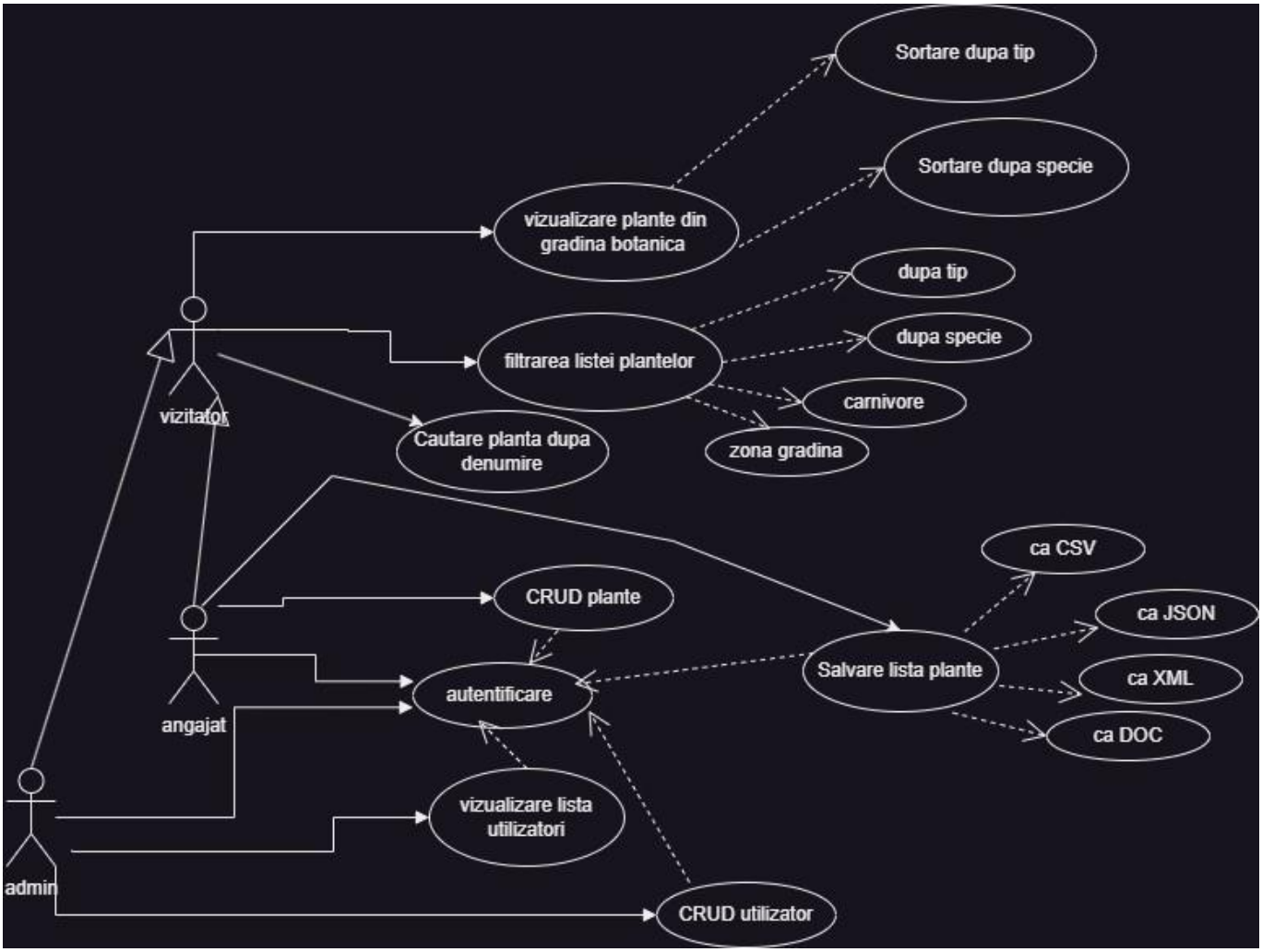

Figura 1. Diagrama de use case

3. Proiectare

Aplicația desktop proiectată de mine respectă arhitectura **MVVM** (care folosește șablonul de proiectare comportamental ***Command***) și principiile **GRASP**. Astfel, GRASP reprezintă principii de asignare a responsabilității claselor.

3.1. Diagrama de pachete

Soluția prezentată (figura 2) utilizează varianta șablonului arhitectural MVVM în care pachetele ***View*** și ***Model*** sunt decuplate.

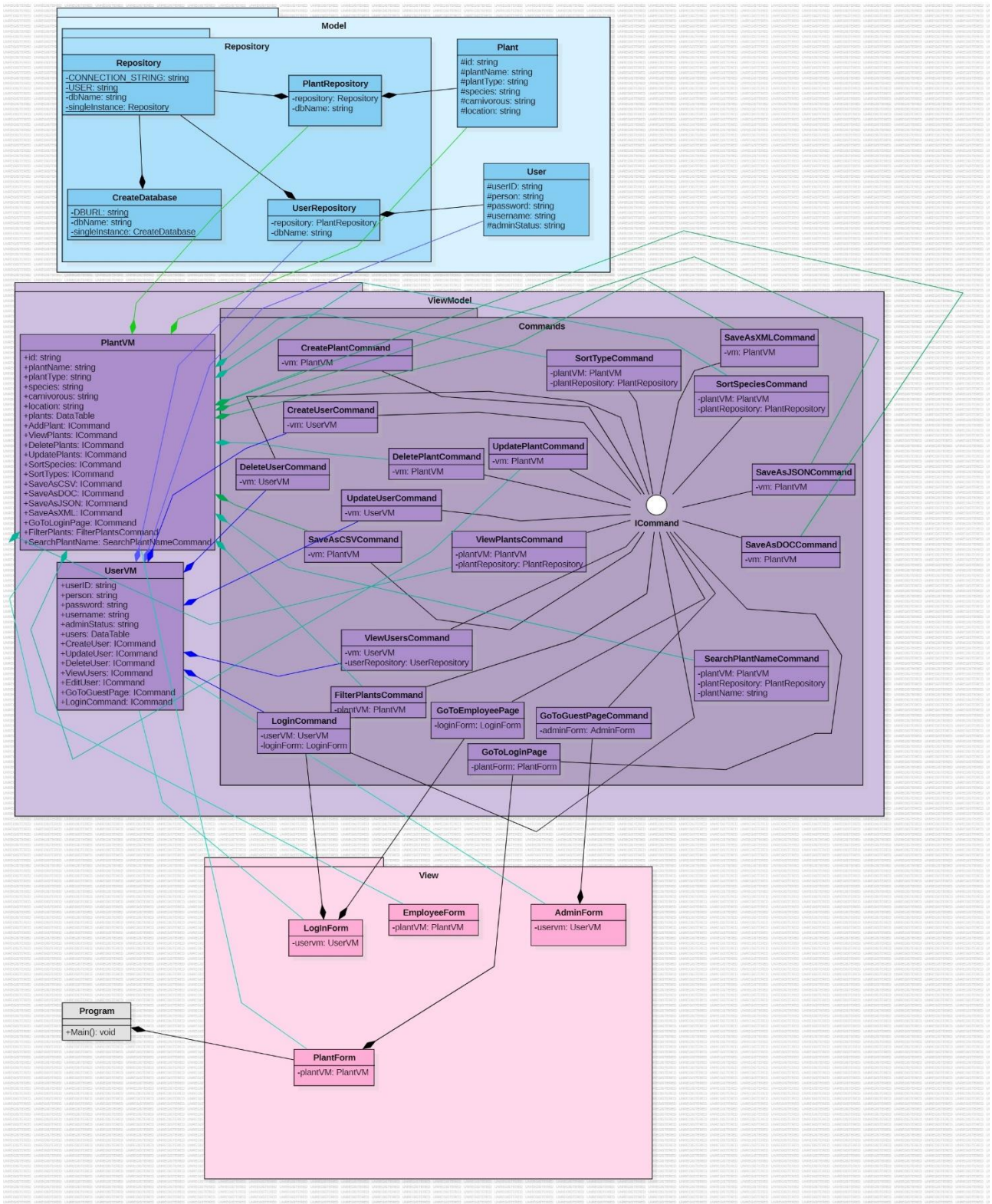

Figura 2. Diagramă de clase restrânsă a arhitecturii MVVM

Pachetul ***Model*** cuprinde 2 clase și un subpachet: ***Plant, User*** (figura 3) și pachetul ***Repository*** (figura 4; acesta având patru clase: ***CreateDatabase***, ***PlantRepository***, ***Repository***, ***UserRepository***). Astfel, pachetul ***Model*** stochează date și logica aferentă.

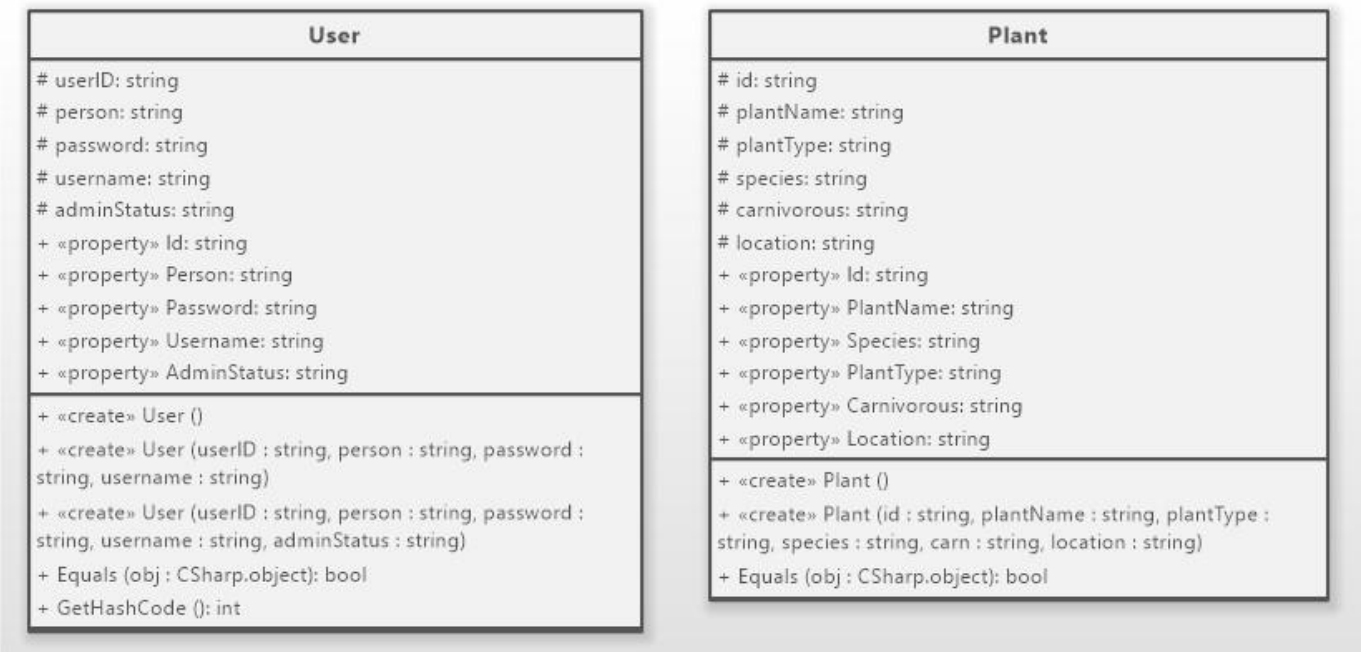

Figura 3. Clasele ***User*** **și** ***Plant*** **din** pachetul ***Model***

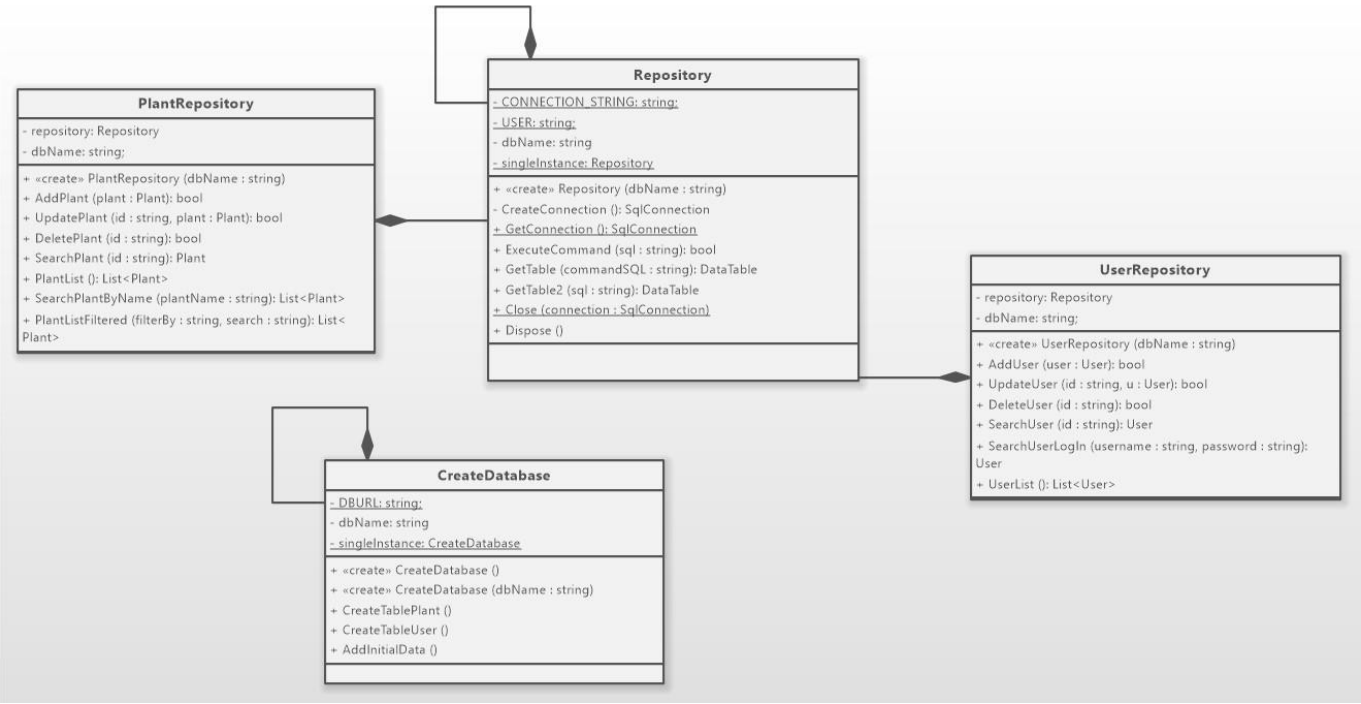

Figura 4. Subpachetul ***Repository*** din pachetul ***Model***

Pachetul ***View*** (figura 5) cuprinde patru clase (***AdminForm***, ***EmployeeForm***, ***LogInForm***, ***PlantForm***) care sunt responsabile cu afișarea unei interfețe prietenoase și cu gestionarea acțiunilor făcute de utilizator.

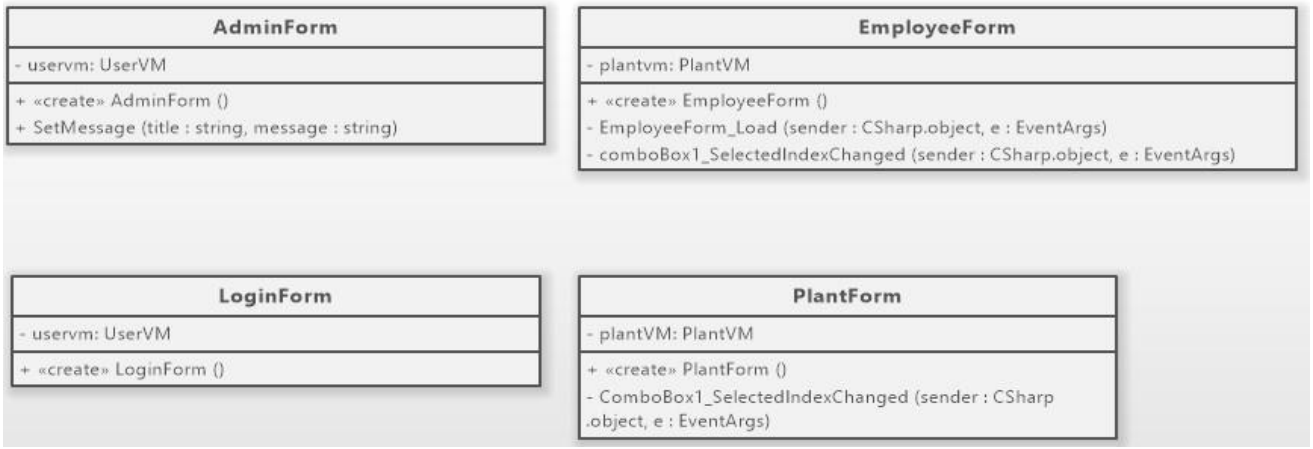

Figura 5. Pachetul ***View***

Pachetul ***ViewModel*** cuprinde 2 clase (***PlantVM*** și ***UserVM*** – figura 6) și subpachetul ***Commands*** (figura 7), acesta având 20 de clase (***CreatePlantCommand***, ***CreateUserCommand***, ***DeletePlantCommand***, ***DeleteUserCommand***, ***FilterPlantsCommand***, ***GoToEmployeePage***, ***GoToGuestPageCommand***, ***GoToLoginPage***, ***LoginCommand***, ***SaveAsCSVCommand***, ***SaveAsDOCCommand***, ***SaveAsJSONCommand***, ***SortTypeCommand,*** ***UpdatePlantCommand*** ***SaveAsXMLCommand***, ***SearchPlantNameCommand***, ***SortSpeciesCommand***, ***UpdateUserCommand***, ***ViewPlantsCommand***, ***ViewUsersCommand***) și interfața ***ICommand***. Astfel, pachetul ***ViewModel*** este responsabil cu implementarea propriu-zisă a acțiunilor la care are acces un utilizator; prin intermediul acestui pachet se face legătura între ***View*** și ***Model***.

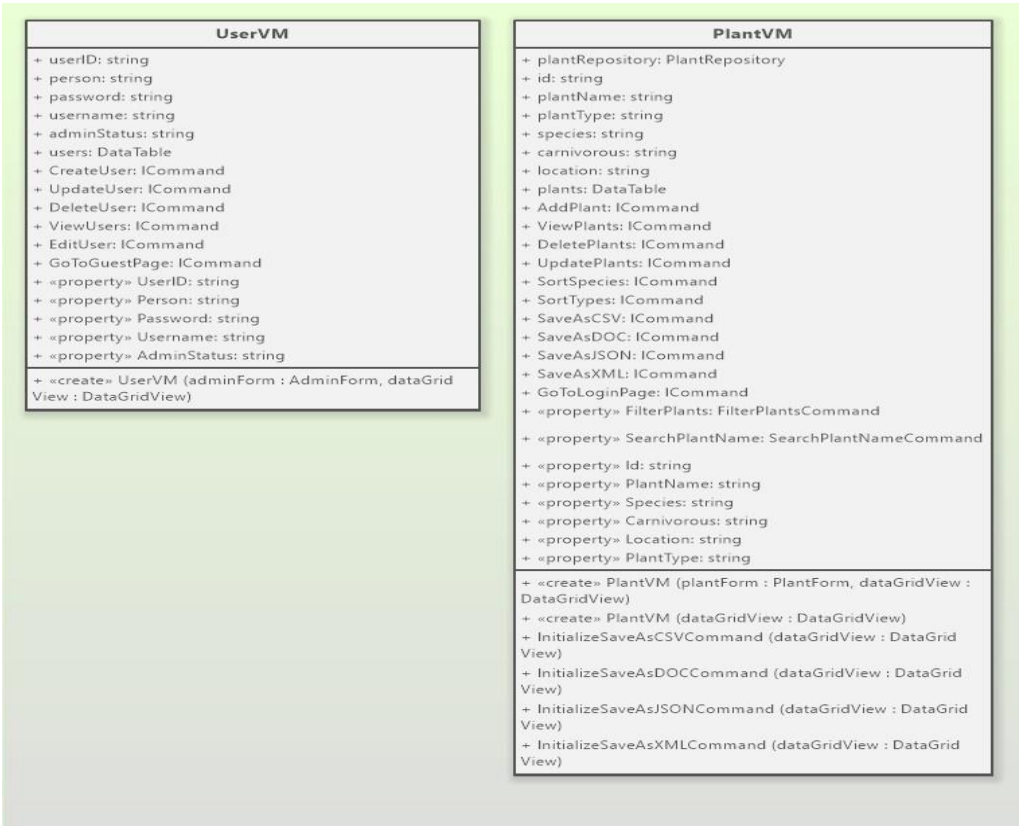

Figura 6. Clasele ***UserVM*** **și** ***PlantVM*** din pachetul ***ViewModel***

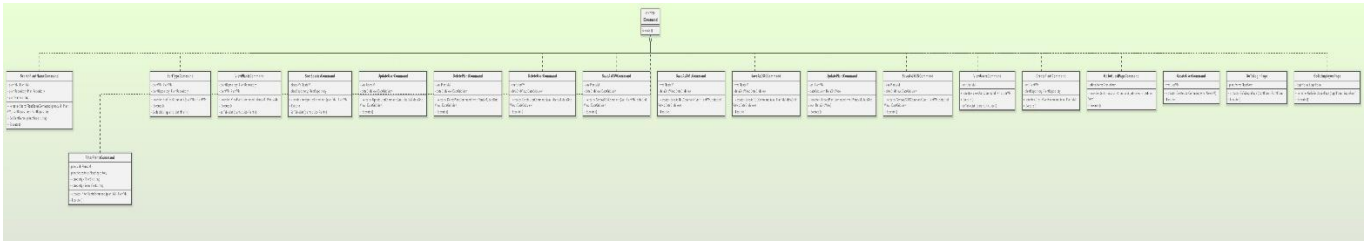

Figura 7. Subpachetul ***Commands*** din pachetul ***ViewModel***

3.2. Diagrama entitate-relație

Diagrama bazei de date este similară cu cea de la tema precedentă, unde am utilizat șablonul MVP (nu a fost necesar să o modific). Acest tip de diagramă (figura 8) este un model vizual care arată conexiunile entităților din baza de date. Astfel, baza de date este formată din două tabele: Users și Plant. Tabela Users este folosită pentru păstrarea datelor fiecărui utilizator (nume de utilizator, parolă, atribut pentru verificarea statutului de administrator). De asemenea, tabela Plant stochează toate plantele prezente în grădina botanică, cu atributele lor: nume, specie, tipul de plantă, atribut care reține dacă planta este carnivoră sau nu și locația plantei în cadrul grădinii.

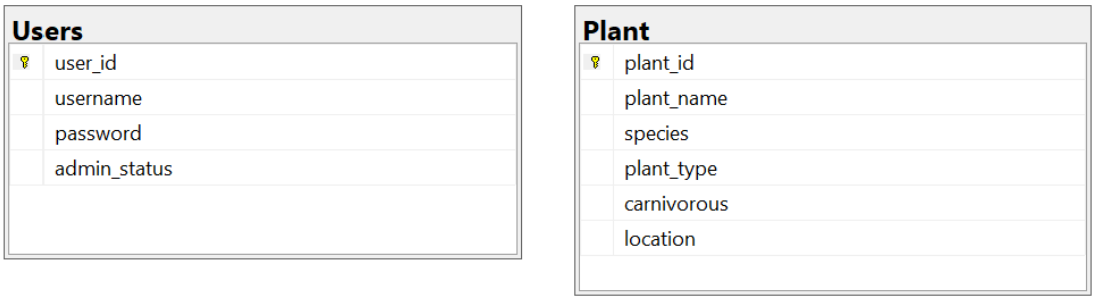

Figura 8. Diagrama entitate-relație

4. Implementare

4.1. Instrumente utilizate

În soluția prezentată se utilizează:

- Limbajul de programare C# (în IDE-ul Visual Studio)
- Sistemul de gestionare de baze de date relaționale SQL Server
- StarUML și Software Ideas Modeler (pentru crearea diagramei de pachete, clase)
- draw.io (pentru diagrama cazurilor de utilizare)

Am ales limbajul C#, deoarece este un limbaj de programare orientat-obiect care simplifică crearea interfeței grafice prin folosirea bibliotecii de clase grafice Windows Forms. Mai mult, C# este un limbaj de programare modern, de uz general, regăsindu-se pe primele poziții ale topurilor limbajelor de programare.

4.2. Descrierea aplicației

La deschiderea aplicației se va afișa pagina de start (figura 9), mai exact cea în care utilizatorul este doar vizitator, acesta nefiind înregistrat în cont. Astfel, un vizitator poate să sorteze plantele după tip sau specie, apăsând unul din butoanele corespunzătoare. Mai mult, plantele din grădina botanică pot fi vizualizate în listă.

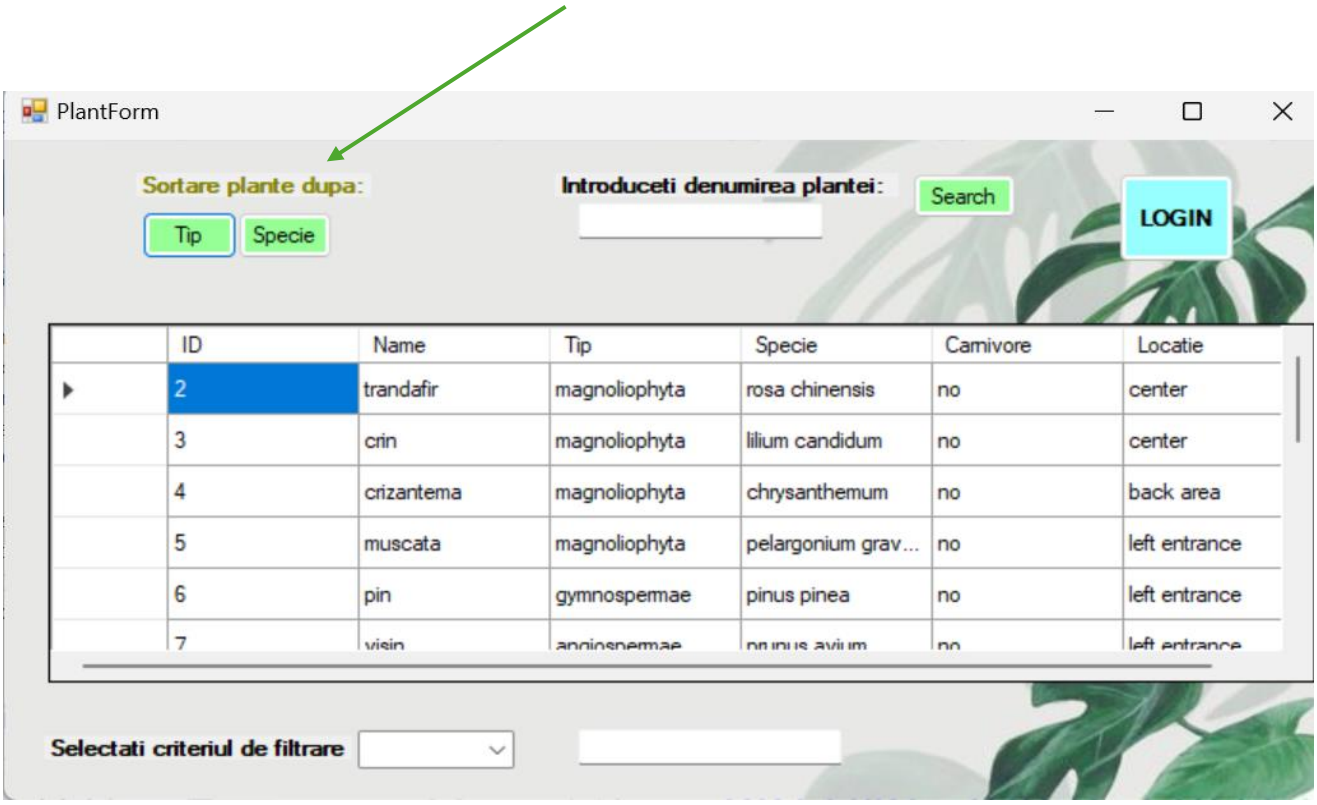

Figura 9. Pagina de pornire a aplicației

De asemenea, un utilizator nelogat poate filtra lista de plante după diverse criterii (tip, specie, carnivore, locație; figura 10), sau poate căuta o plantă după denumire (figura 11).

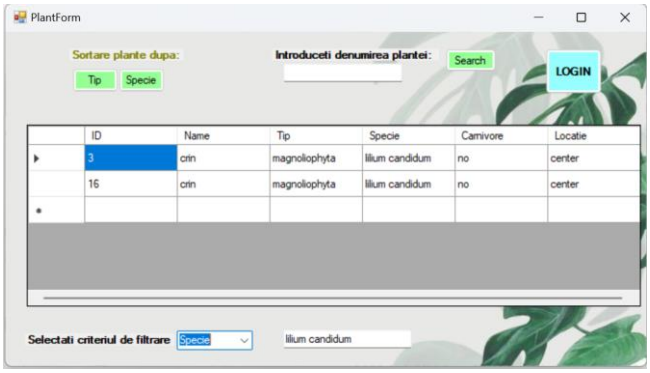

Figura 10. Filtrarea listei de plante după specie

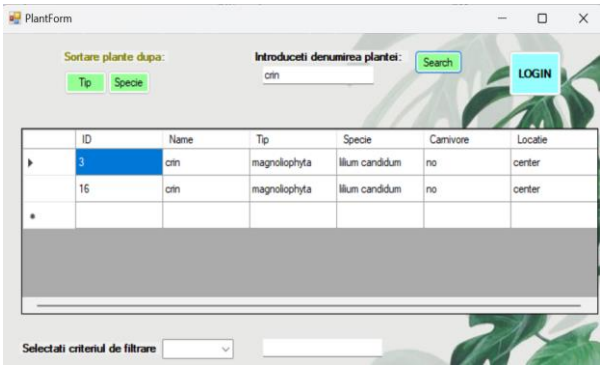

Figura 11. Căutare plantă după nume

În pagina de start a aplicației desktop există și opțiunea de Login, care va deschide o filă nouă (figura 12), unde se poate face autentificarea (în rolul de angajat sau administrator).

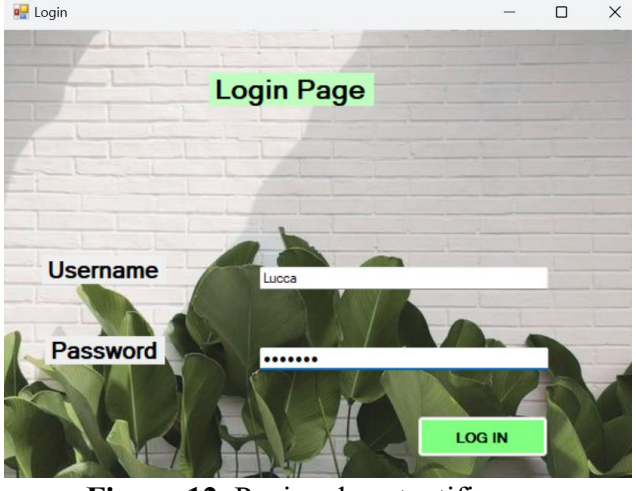

Figura 12. Pagina de autentificare

Logarea ca angajat (figura 13) oferă toate funcționalitățile disponibile și unui vizitator, la care se adaugă operații de creare, citire, actualizare, ștergere a plantelor din grădina botanică sau posibilitatea de a salva listele cu plante în mai multe formate (csv, json, xml, doc).

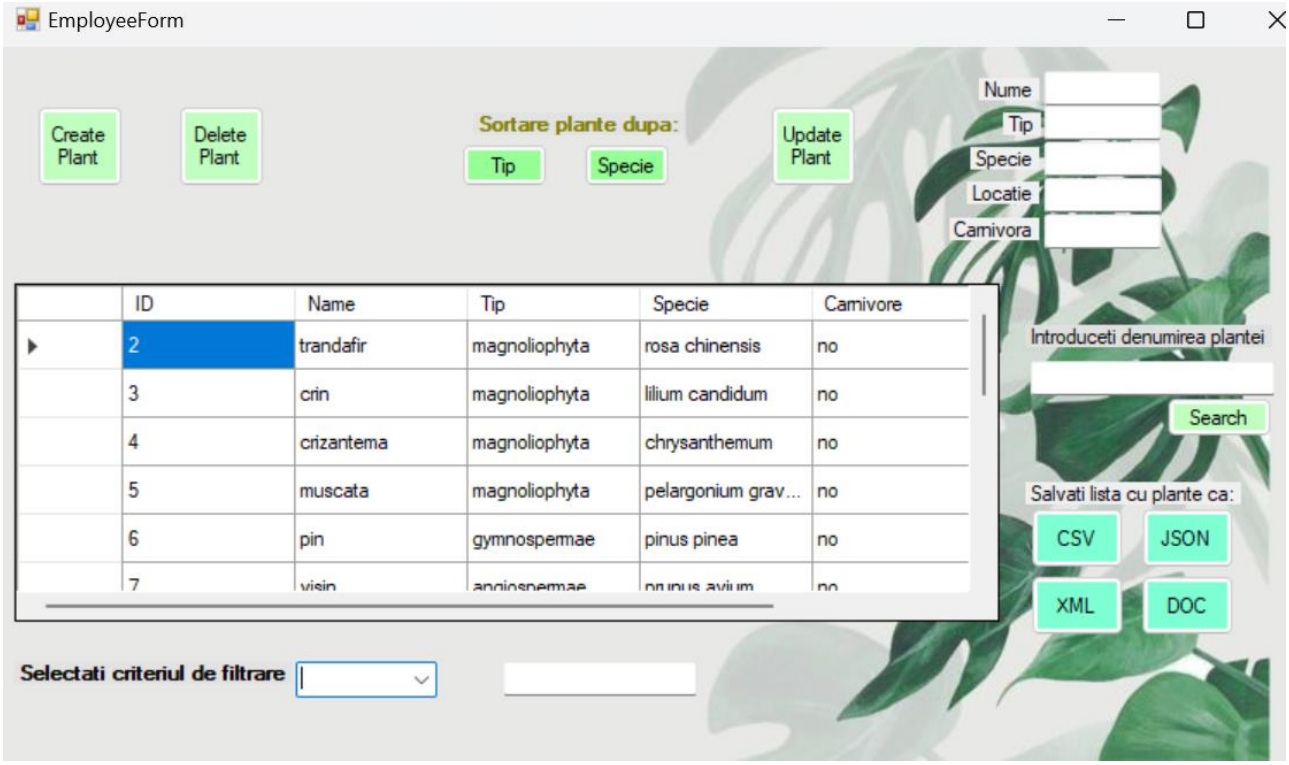

Figura 13. Pagina de angajat

Pentru operațiile de salvare a listei cu plante (figura 14), se alege formatul fișierului dorit la ieșire (prin click pe unul din butoanele: CSV, JSON, XML, DOC); apoi, se va deschide File Explorer, oferind posibilitatea salvării fișierului generat, în locația dorită.

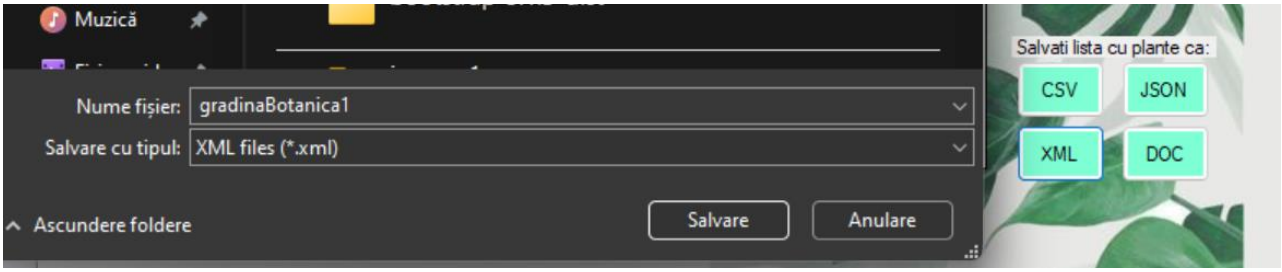

Figura 14. Salvarea listei cu plante în formatul dorit

Pentru a șterge o plantă din listă, se selectează planta dorită și se apasă butonul Delete Plant. Dacă se dorește crearea unei plante, se introduc detaliile corespunzătoare plantei în câmpurile din dreapta-sus (Nume, Tip, Specie, Locatie, Carnivora – cu opțiunile yes/no) și se apasă butonul Create Plant. Asemănător și pentru operația de actualizare a unei plante, se selectează prima dată planta dorită, se adaugă informații în câmpurile corespunzătoare (toate câmpurile trebuie completate, chiar dacă informația rămâne neschimbată), iar apoi se apasă butonul Update Plant.

Dacă ne înregistrăm ca admin (figura 15) vom avea acces, de asemenea, la toate operațiile permise și unui vizitator (prin apăsarea butonului **Go to Guest page** se deschide chiar pagina inițială corespunzătoare vizitatorului), la care se adaugă operații de creare, citire, actualizare și ștergere a utilizatorilor din baza de date. De asemenea, administratorul va putea vizualiza lista tuturor utilizatorilor.

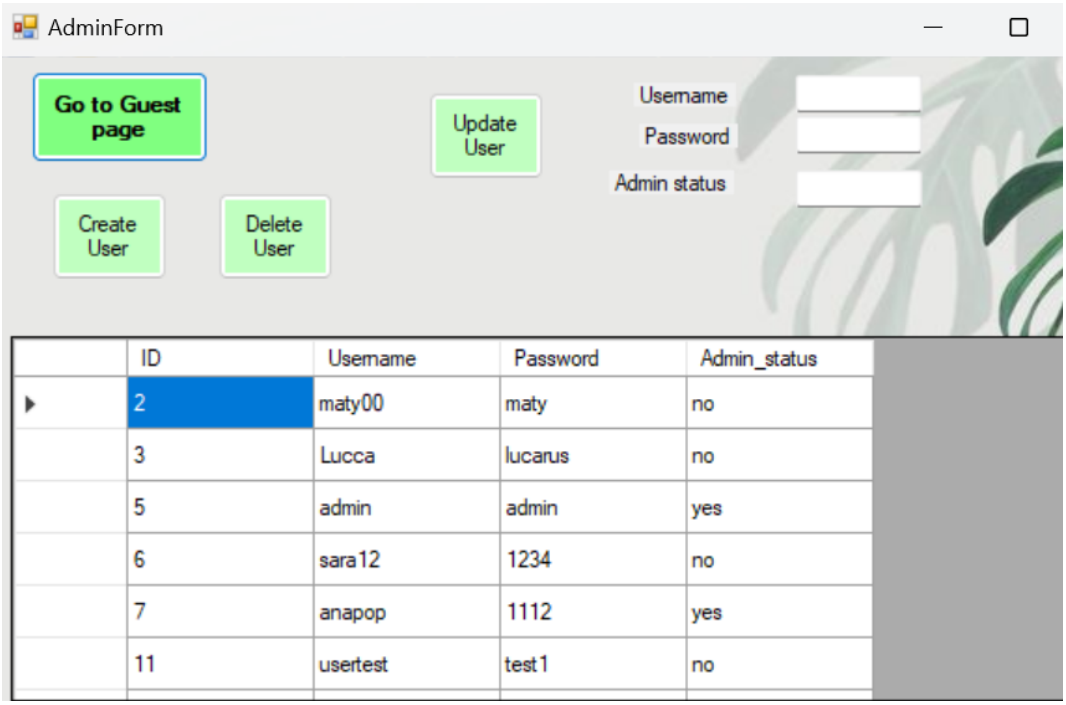

Figura 15. Pagina de administrator

Astfel, pentru a adăuga un utilizator nou, administratorul trebuie să completeze cele 3 câmpuri din dreapta-sus (Username, Password, Admin status – yes/no), iar apoi prin apăsarea butonului Create User, se va salva automat noul user în listă (dacă datele sunt corect introduse). Pentru ștergere și actualizare, trebuie să se facă prima dată selecția unui utilizator (figura 16).

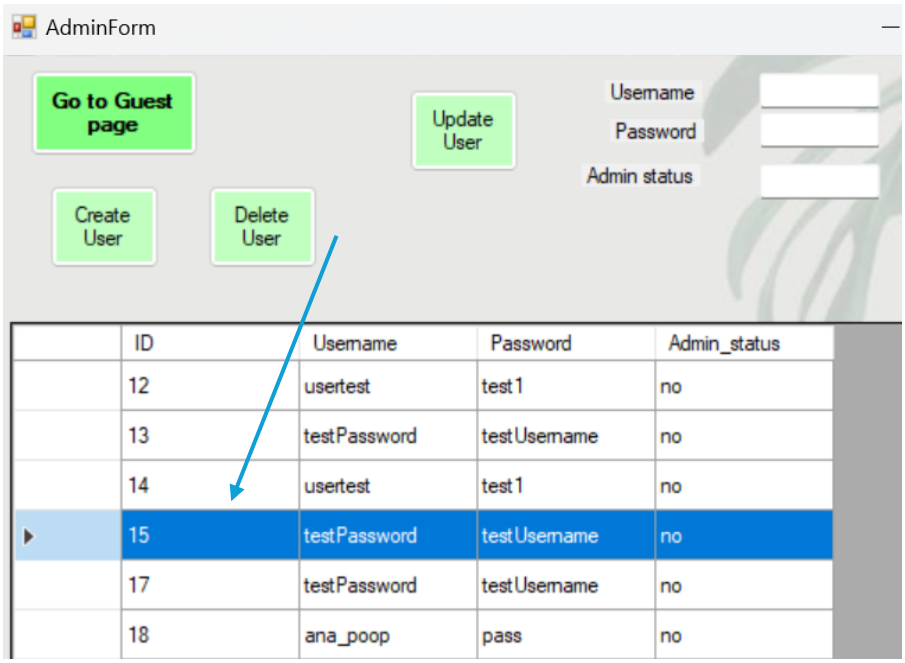

Figura 16. Selecție utilizator pentru update/delete