



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

PROIECTARE SOFTWARE

Documentație proiect

Grădina Botanică

Student: Adela Iosif
Grupa: 30233

2023-2024

Cuprins

1.	Introducere	3
1.1.	Obiectiv	3
1.2.	Cerințe	3
1.3.	Enunțul problemei	3
2.	Analiza problemei	4
2.1.	Diagrama cazurilor de utilizare.....	4
3.	Proiectare.....	5
3.1.	Diagrama de pachete.....	5
3.2.	Diagrama entitate-relație	8
4.	Implementare.....	8
4.1.	Instrumente utilizate	8
4.2.	Descrierea aplicației.....	8

1. Introducere

1.1. Obiectiv

Obiectivul acestei teme este familiarizarea cu șablonul architectural **Model-View-Presenter**. 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 **MVP** și principiile **SOLID**, 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.
- ❖ În **faza de testare** se vor implementa teste unitate (proiect de testare) corespunzătoare operațiilor de creare a bazei de date, de realizare a conexiunii la baza de date, de creare a tabelelor și de interogare a tabelelor bazei de date.
- ❖ 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.

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.

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. 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.

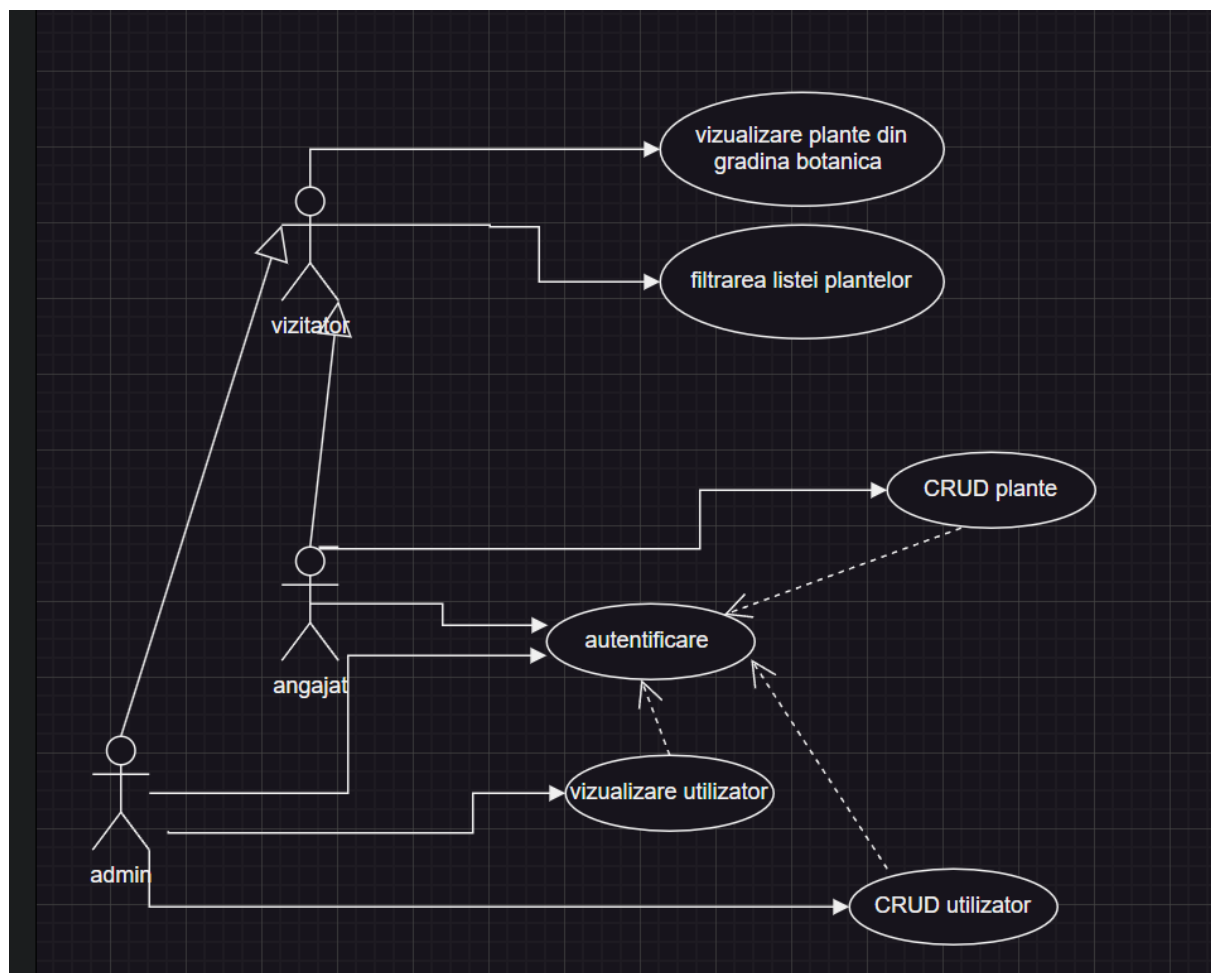


Figura 1. Diagrama de use case

3. Proiectare

3.1. Diagrama de pachete

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

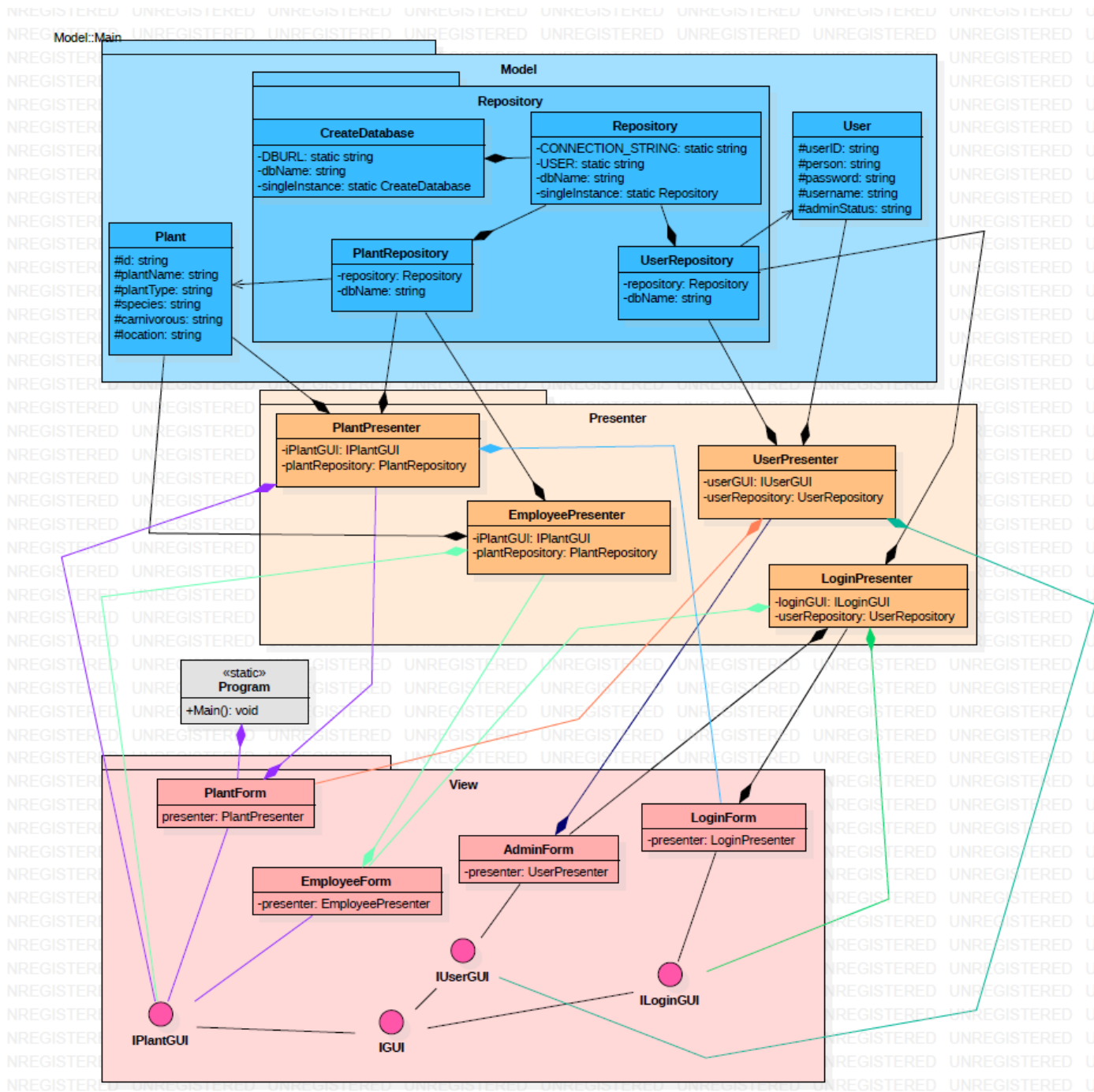


Figura 2. Diagramă de clase restrânsă a arhitecturii MVP

Pachetul **Model** (figura 3) cuprinde două clase: **Plant** și **User** și pachetul **Repository** (patru clase: **CreateDatabase**, **PlantRepository**, **Repository**, **UserRepository**).

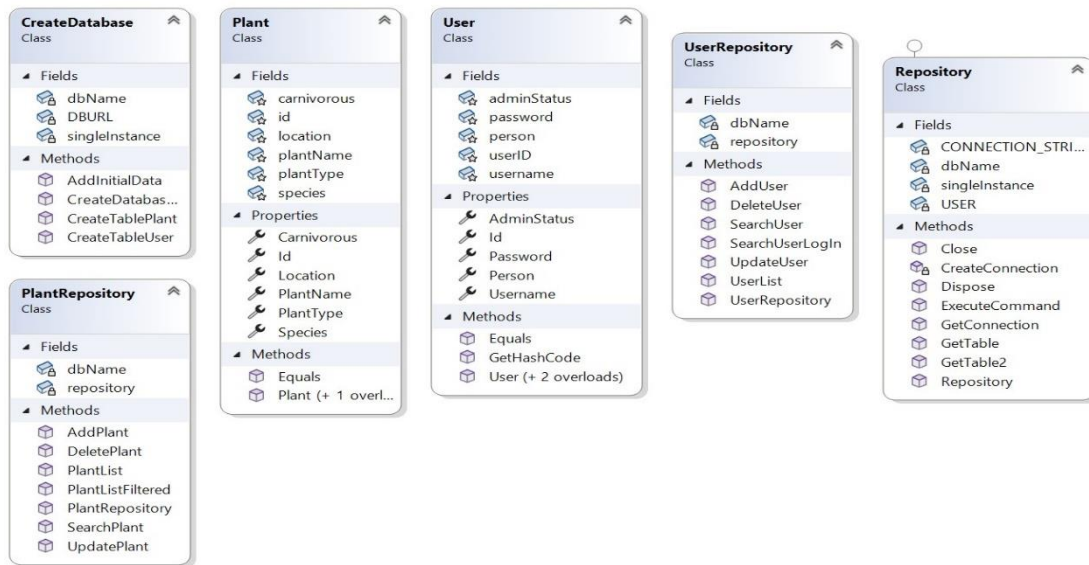


Figura 3. Pachetul **Model**

Pachetul **View** (figura 4) cuprinde patru clase (**AdminForm**, **EmployeeForm**, **LogInForm**, **PlantForm**) și trei interfețe (**ILogInGUI**, **IPlantGUI**, **IUserGUI**).

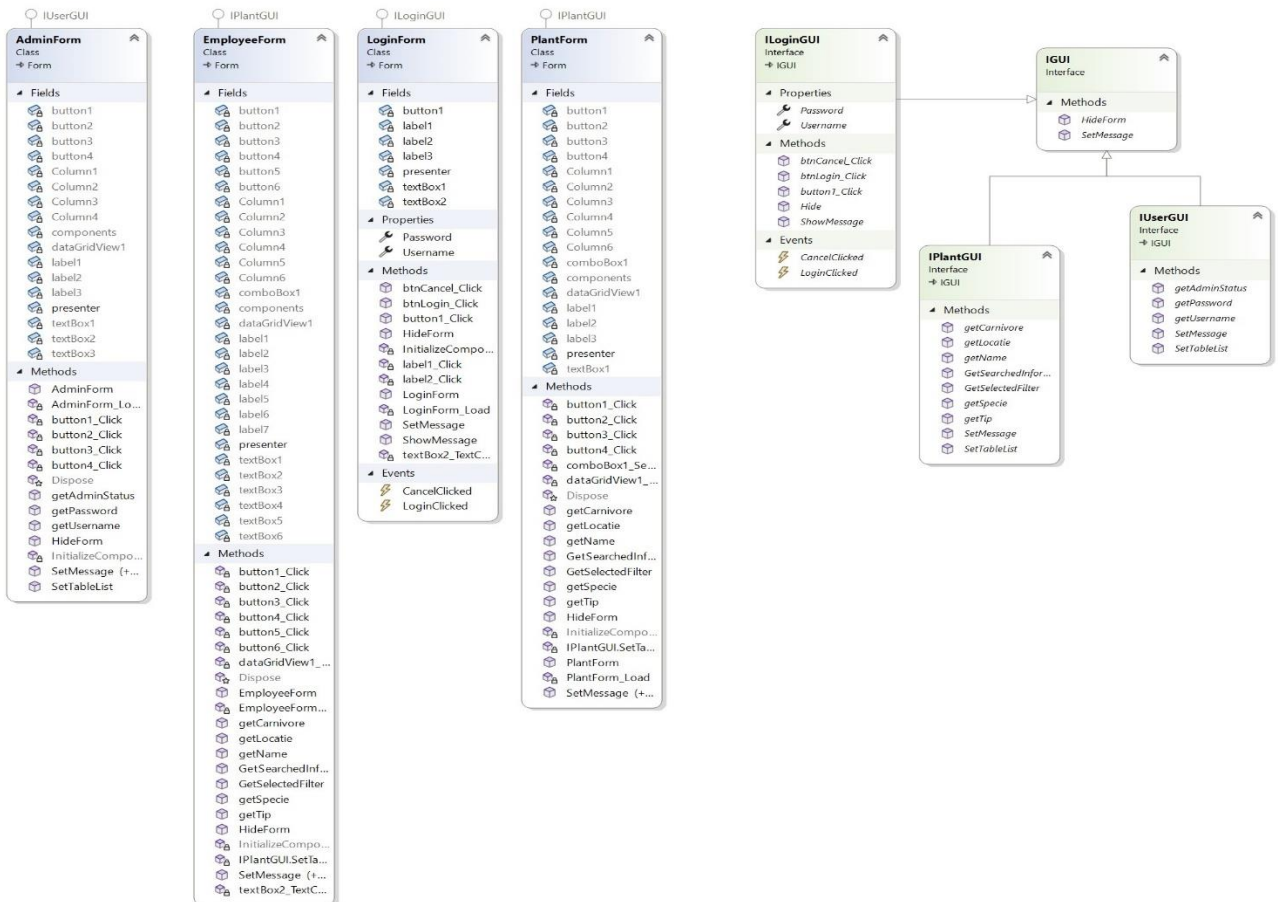


Figura 4. Pachetul **View**

Pachetul **Presenter** (figura 5) cuprinde patru clase (*EmployeePresenter*, *LoginPresenter*, *PlantPresenter*, *UserPresenter*).

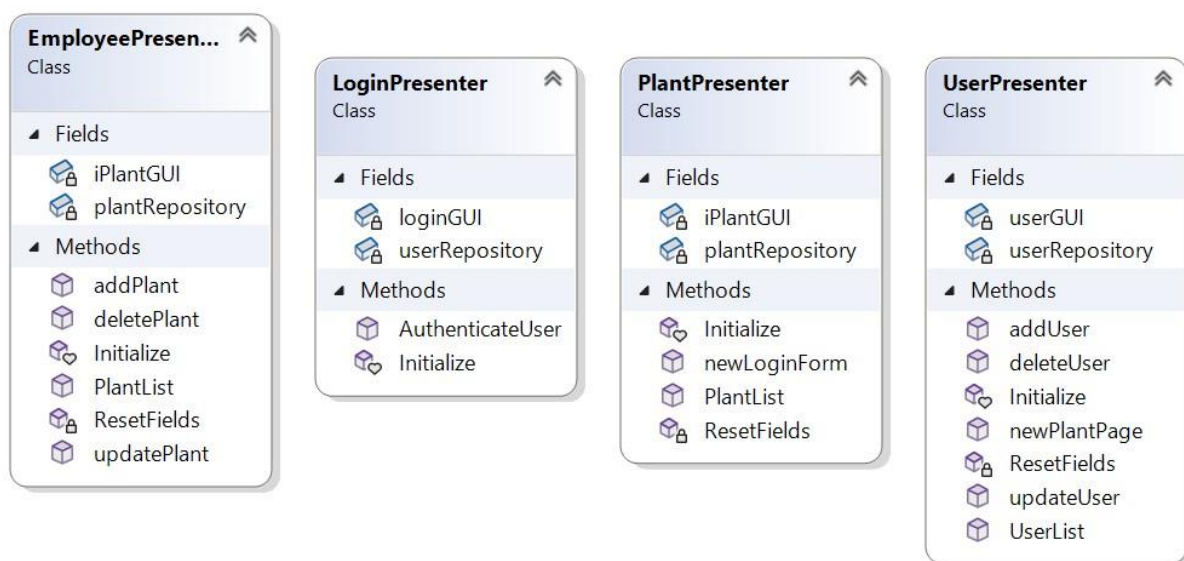


Figura 5. Pachetul *Presenter*

Pachetul **Teste** (figura 6) cuprinde patru clase (*CreateDatabaseTest*, *PlantRepositoryTest*, *RepositoryTest*, *UserRepositoryTest*) în care am testat principalele funcționalități ale claselor din pachetul *Repository*.

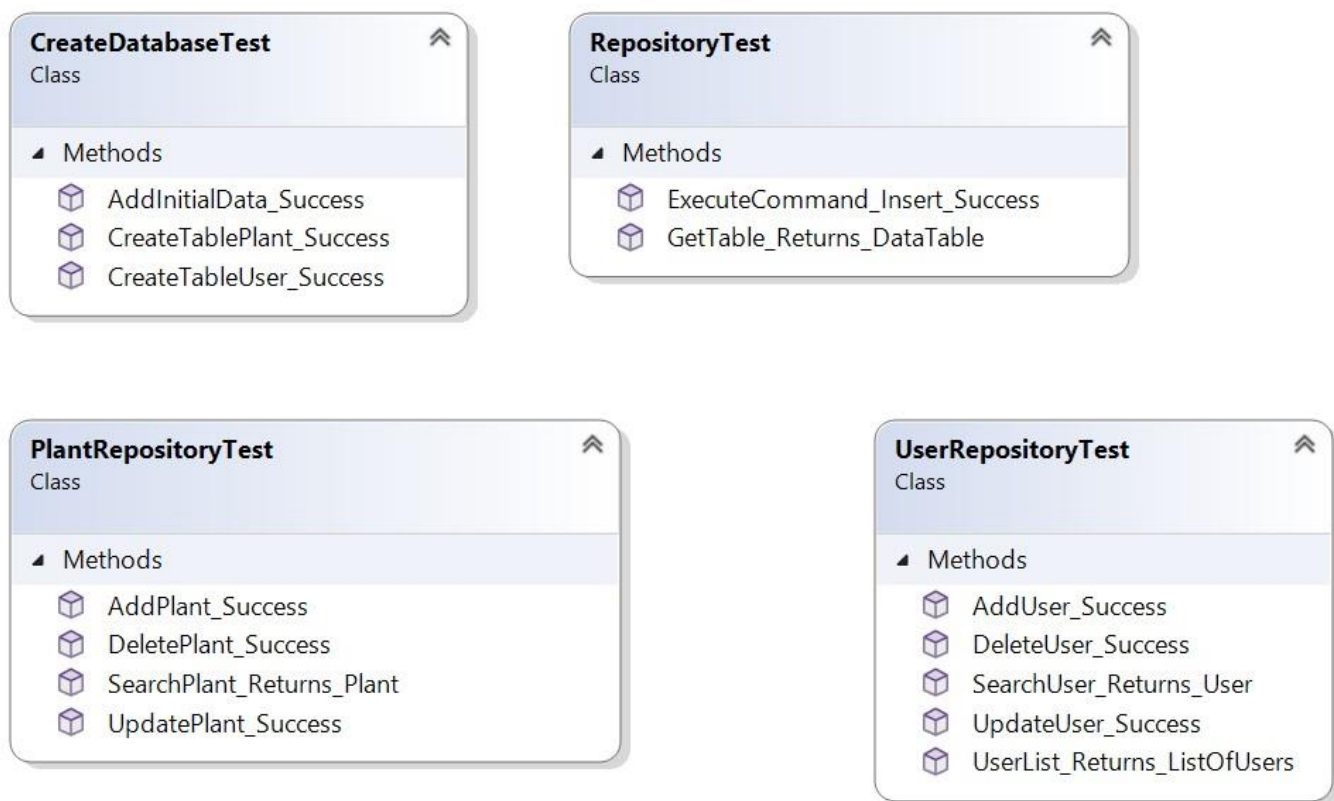


Figura 6. Pachetul *Teste*

3.2. Diagrama entitate-relație

Acest tip de diagramă (figura 7) 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.

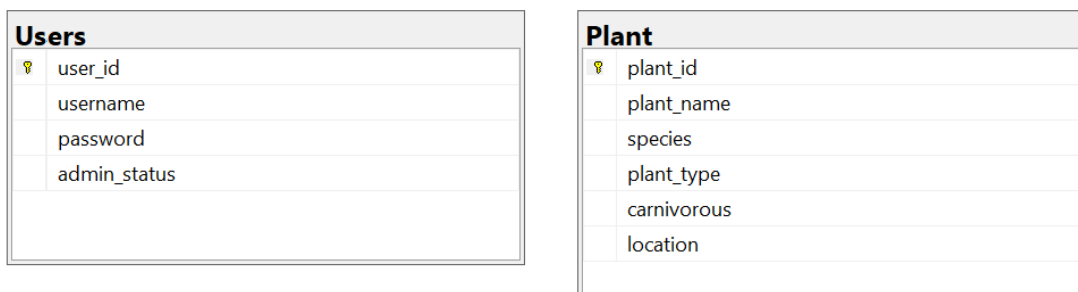


Figura 7. Diagramă entitate-relație

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
- Framework-ul NUnit pentru implementarea testelor unitare
- StarUML (pentru crearea diagramei de pachete)

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.

4.2. Descrierea aplicației

La deschiderea aplicației se va afișa pagina de start (figura 8), 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.

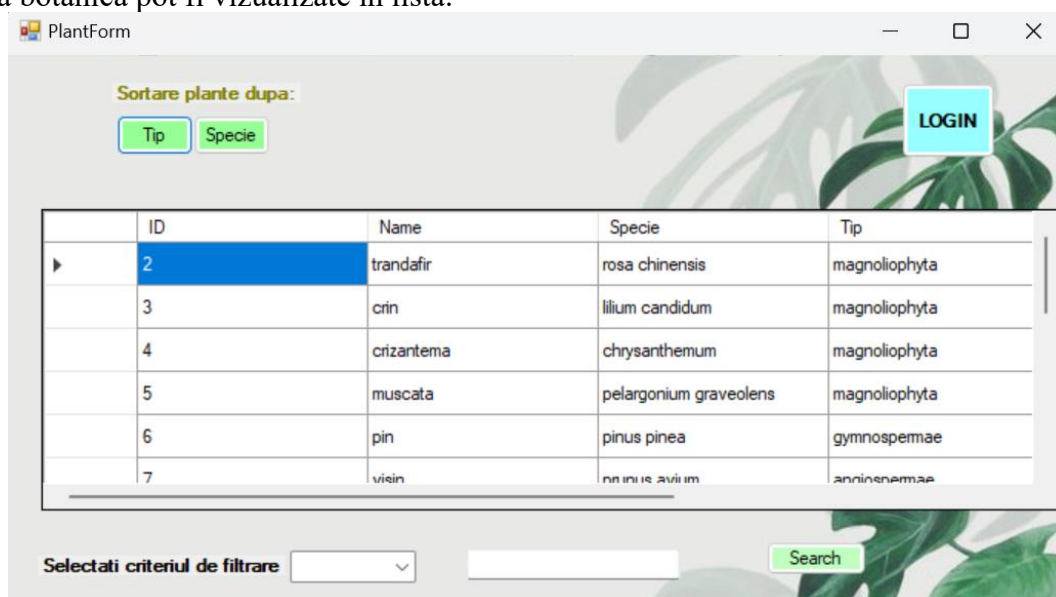


Figura 8. Pagina de pornire a aplicației

De asemenea, tot în pagina inițială (figura 9) există opțiunea de a filtra plantele după diferite criterii. Vizitatorul va alege un criteriu din lista oferită, va introduce denumirea exactă a criteriului pe care îl caută, iar apoi apăsând butonul **Search**, se vor returna în listă câmpurile corespunzătoare căutării (dacă există). În plus, avem opțiunea de resetare a listei (se alege **Reset List** și apoi se apasă butonul **Search**).

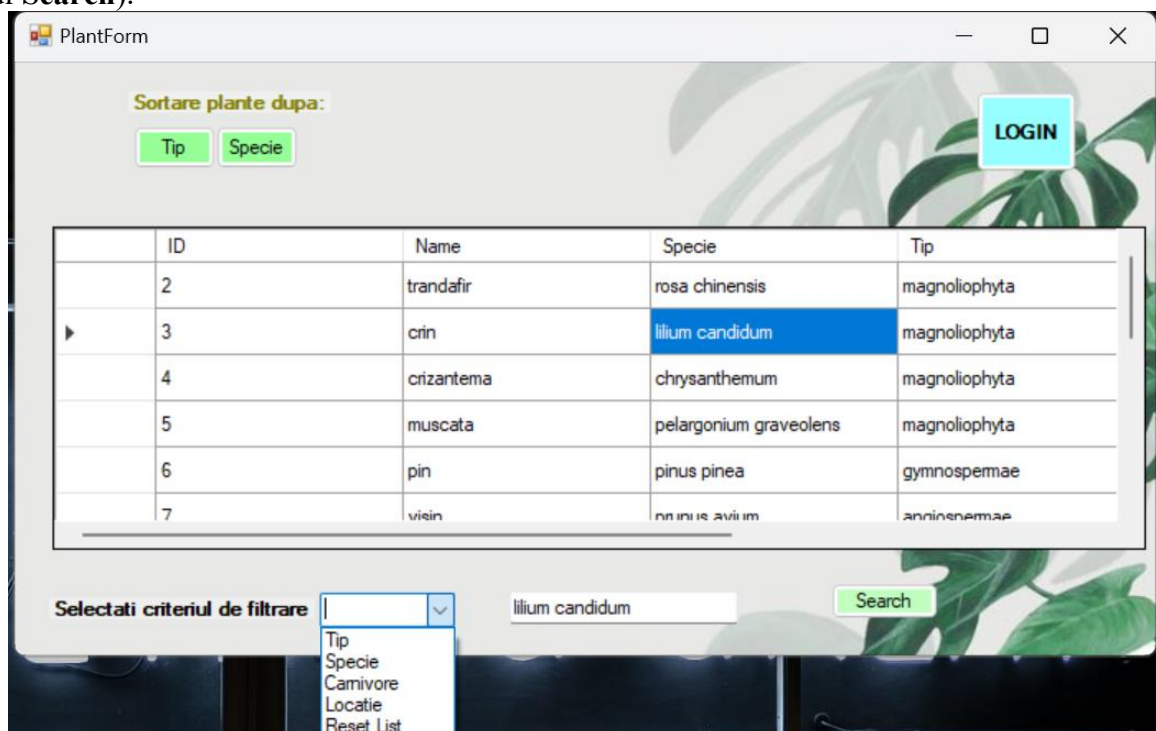


Figura 9. Pagina de pornire a aplicației, exemplificarea filtrării

Pagina conține și opțiunea de Login, care va deschide o pagină nouă (figura 10), unde se poate face autentificarea (în rolul de angajat sau administrator).

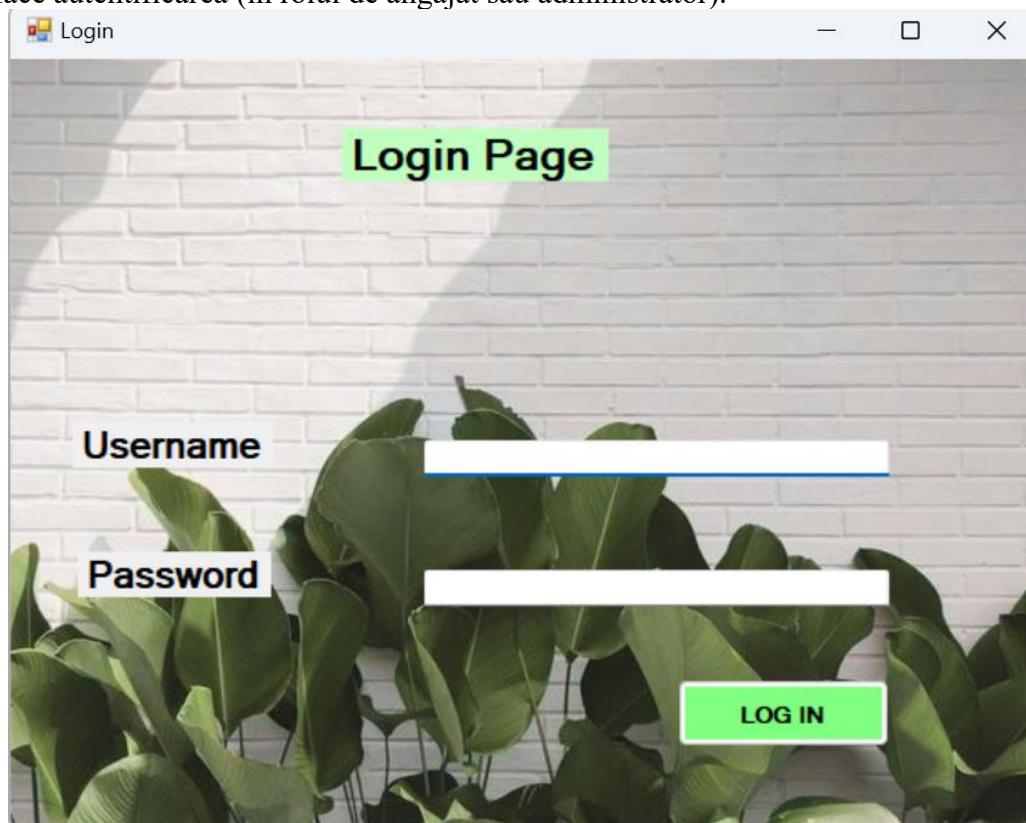


Figura 10. Pagina de Login

Logarea ca angajat (figura 11) oferă toate funcționalitățile disponibile și unui vizitator, la care se adaugă operații de creare, citire, actualizare și ștergere a plantelor din grădina botanică.

EmployeeForm

Create Plant Delete Plant

Sortare plante dupa: Tip Specie Update Plant

Nume
Tip
Specie
Locatie
Camivora

	ID	Name	Specie	Tip
▶	2	trandafir	rosa chinensis	magnoliophyta
	3	crin	lilium candidum	magnoliophyta
	4	crizantema	chrysanthemum	magnoliophyta
	5	muscata	pelargonium graveolens	magnoliophyta
	6	pin	pinus pinea	gymnospermae
	7	visin	ornithogalum	angiospermae

Selectati criteriul de filtrare Search

Figura 11. Pagina de Angajat

Dacă ne înregistrăm ca admin (figura 12) 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.

AdminForm

Go to Guest page Update User

Create User Delete User

Username
Password
Admin status

	ID	Username	Password	Admin_status
▶	2	maty00	maty	no
	3	Lucca	lucarus	no
	5	admin	admin	yes
	6	sara12	1234	no
	7	anapop	1112	yes
	11	usertest	test1	no

Figura 12. Pagina de Admin

Pentru a crea un nou utilizator (figura 13) trebuie să introducem date în câmpurile **Username**, **Password** și **Admin status**, apoi vom apăsa butonul **Create User**.

	ID	Username	Password	Admin_status
	17	testPassword	testUsername	no
	18	ana_poop	pass	no
	19	maty00	maty	no
	21	lavy	11	no
*				

Figura 13. Exemplificarea adăugării unui nou utilizator

Se poate observa că s-a creat (figura 14) un nou utilizator cu datele introduse, acestuia fiindu-i atribuit id-ul corespunzător.

	ID	Username	Password	Admin_status
	18	ana_poop	pass	no
	19	maty00	maty	no
	21	lavy	11	no
	26	userNou	new123	no
*				

Figura 14. Adăugarea cu succes a unui nou utilizator

Mai mult, pentru ștergerea unui utilizator, se selectează user-ul dorit din panoul de selecție din partea stângă, și se apasă butonul **Delete User**. Pentru actualizare, se selectează utilizatorul dorit, se introduc datele noi în câmpurile corespunzătoare și apoi se apasă **Update User**.