



**UNIVERSITATEA
TEHNICĂ**
DIN CLUJ-NAPOCA

Tema 1

Proiectare software

Autor: Volcov Sabina
Grupa: 30231

FACULTATEA DE AUTOMATICĂ
ȘI CALCULATOARE

martie 2024

Cuprins

1	Problema de rezolvat	2
2	Faza de analiza - Diagrama cazurilor de utilizare	2
2.1	Enuntul problemei	2
2.2	Diagrama cazurilor de utilizare	3
3	Faza de proiectare - Diagrame de clase	4
3.1	Structura bazei de date	4
3.2	Diagrama completă de clase	4
3.3	Pachetul Model	6
3.4	Subpachetul Repository	6
3.5	Pachetul Presenter	7
3.6	Pachetul View	8
3.7	Pachetul Test	9
4	Faza de implementare - Aplicatia	10
4.1	Instrumentele utilizate	10
4.2	Aplicatia	10

1 Problema de rezolvat

Se cere dezvoltarea unei aplicații care să respecte următoarele 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:
 1. proiectarea dată de diagrama de clase;
 2. 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 tabelor și de interogare a tabelor bazei de date.
- Finalizarea temei va consta în predarea unui director ce va cuprinde:
 1. Un fișier cu diagramele UML realizate;
 2. Baza de date;
 3. Aplicația soft;
 4. 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.

2 Faza de analiza - Diagrama cazurilor de utilizare

2.1 Enunțul problemei

Problema 11

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.

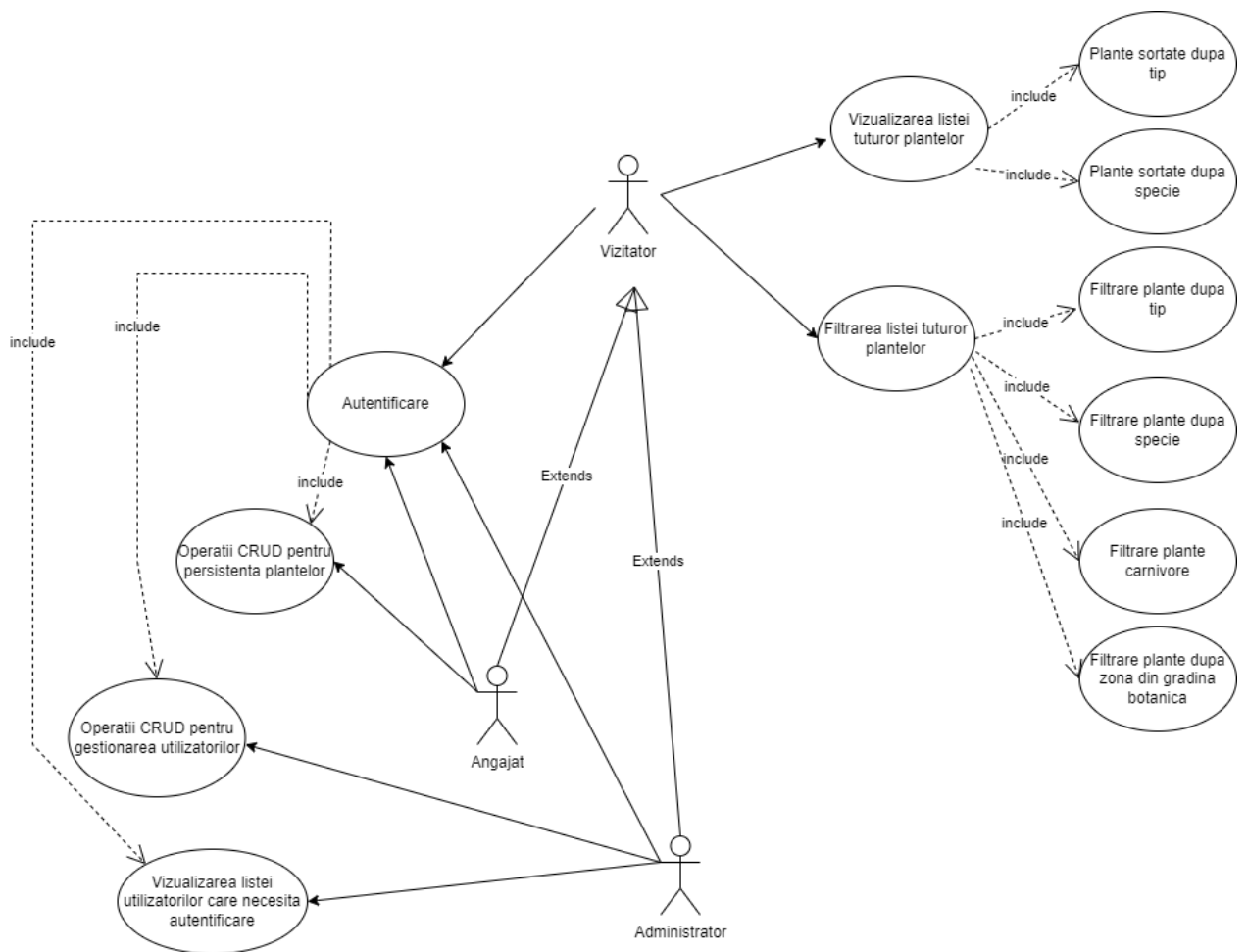
Utilizatorii de tip angajat al grădinii botanice pot efectua următoarele operații după autentificare:

- 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.

2.2 Diagrama cazurilor de utilizare



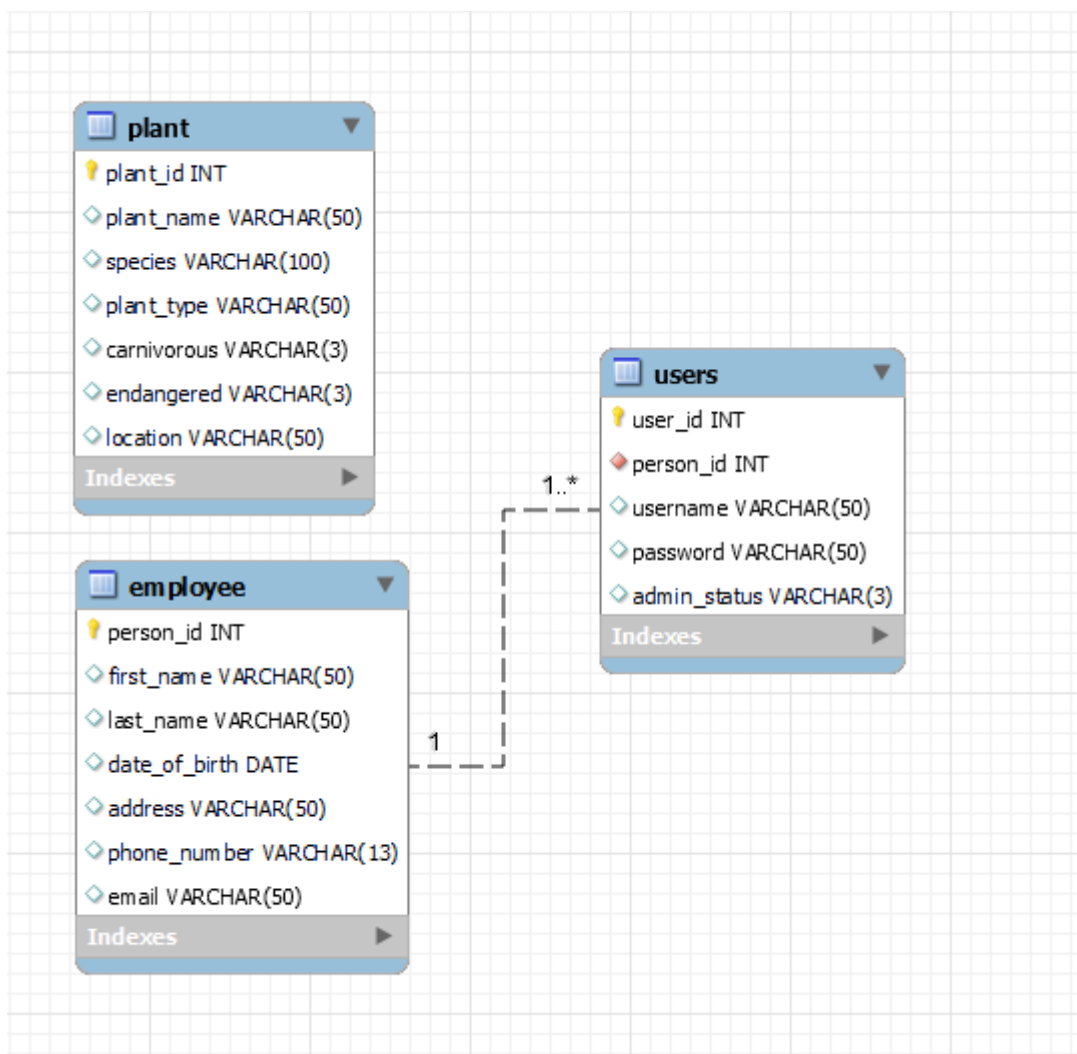
Din diagramă reies cei 3 actori care apar în aplicație, și anume vizitatorii anonimi, care pot vizualiza, sorta, filtra și căuta plantele din baza de date, angajații autentificați care pot moștenesc use-case-urile vizitatorii și pot pe deasupra realiza operații CRUD pentru persistența plantelor, și administratorii, care la fel moștenesc cazurile de utilizare ale vizitatorilor, dar spre deosebire de angajați pot efectua operații CRUD și vizualiza utilizatorii care necesită autentificare.

3 Faza de proiectare - Diagrame de clase

3.1 Structura bazei de date

Sistemul de gestionare a bazei de date folosit este MySQL, conectarea cu mediul Java realizându-se prin intermediul driver-ului jdbc.

Baza de date conține 3 tabele, și anume "plant", pentru persistența datelor despre plantele din grădina botanică cu toate atributele lor (denumire, specie, tip, localizare etc.), "employee", pentru păstrarea și gestionarea datelor despre angajații din sistem (cu datele aferente: nume, prenume, data nașterii, adresă, număr de telefon, email) și "users", pentru datele fiecărui utilizator asociat unui angajat în parte (cu referință la persoana de care aparține, nume de utilizator, parolă și un atribut care determină rolul de administrator).

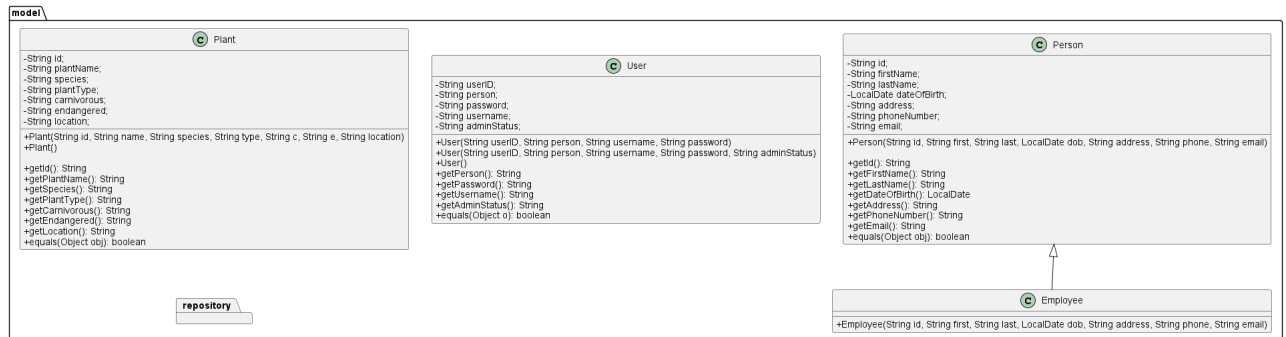


3.2 Diagrama completă de clase

Din motive de vizibilitate, voi analiza în continuare pe rând fiecare pachet din diagrama completă:

3.3 Pachetul Model

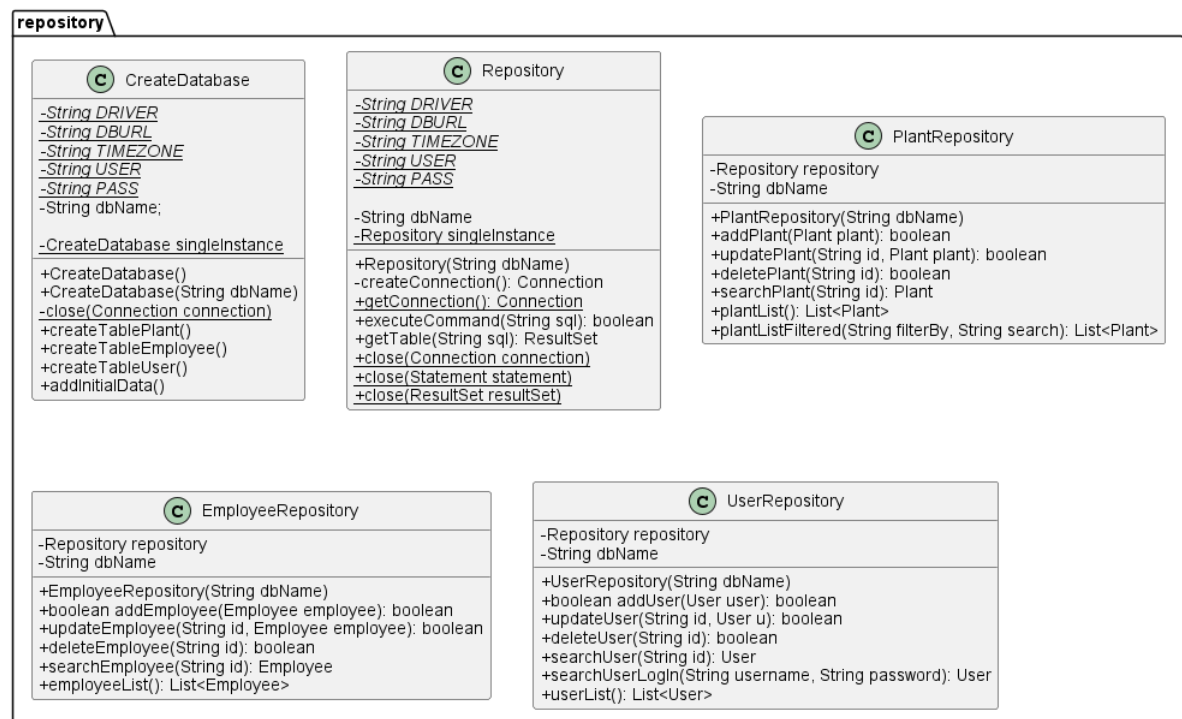
Definirea claselor care modelează lumea aplicației s-a efectuat în pachetul model, care pe lângă clasele principale conține și un subpachet repository, care gestionează legătura dintre baza de date și aplicație.



Pachetul model este format din clasele Plant, Person din care moștenește clasa Employee și User, fiecare cu proprii constructori pentru instanțele care vor corespunde intrărilor din tabelele respective din baza de date și vor modela o grădină botanică reală.

În afară de attribute, clasele din model au definite și gettere pentru acestea, necesare la reflecția datelor în momentul creării tabelelor din interfața grafică a aplicației.

3.4 Subpachetul Repository



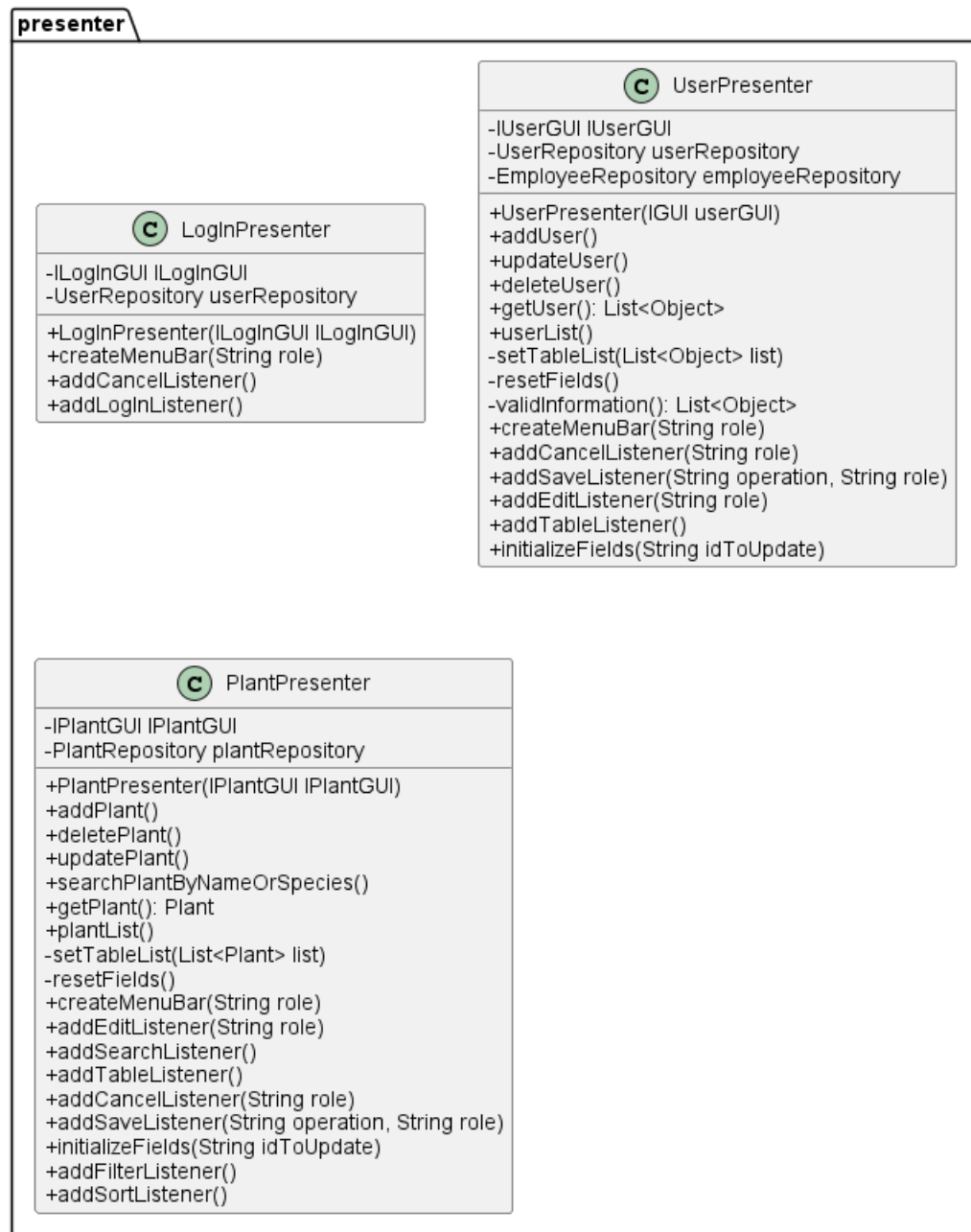
Subpachetul repository din cadrul modelului are rolul de a crea baza de date si de a gestiona operatiile CRUD care se pot efectua pe obiectele din aplicatie.

Clasele pe care le include acest pachet sunt: CreateDatabase (care construiește baza de date și tabelele), Repository (care gestionează legătura cu baza de date, deschizând și închizând

conexiunea la cerere), PlantRepository prin intermediul căreia se realizează operațiile CRUD pentru tabelul "plant", UserRepository pentru operațiile CRUD caracteristice tabelului "users" și EmployeeRepository pentru operațiile CRUD pentru tabelul "employee".

3.5 Pachetul Presenter

Pachetul presenter are rolul de a face legătura dintre model și interacțiunile utilizatorului cu interfața grafică a aplicației, astfel încât în cadrul acestuia are loc procesarea evenimentelor care pot avea loc în sistem.



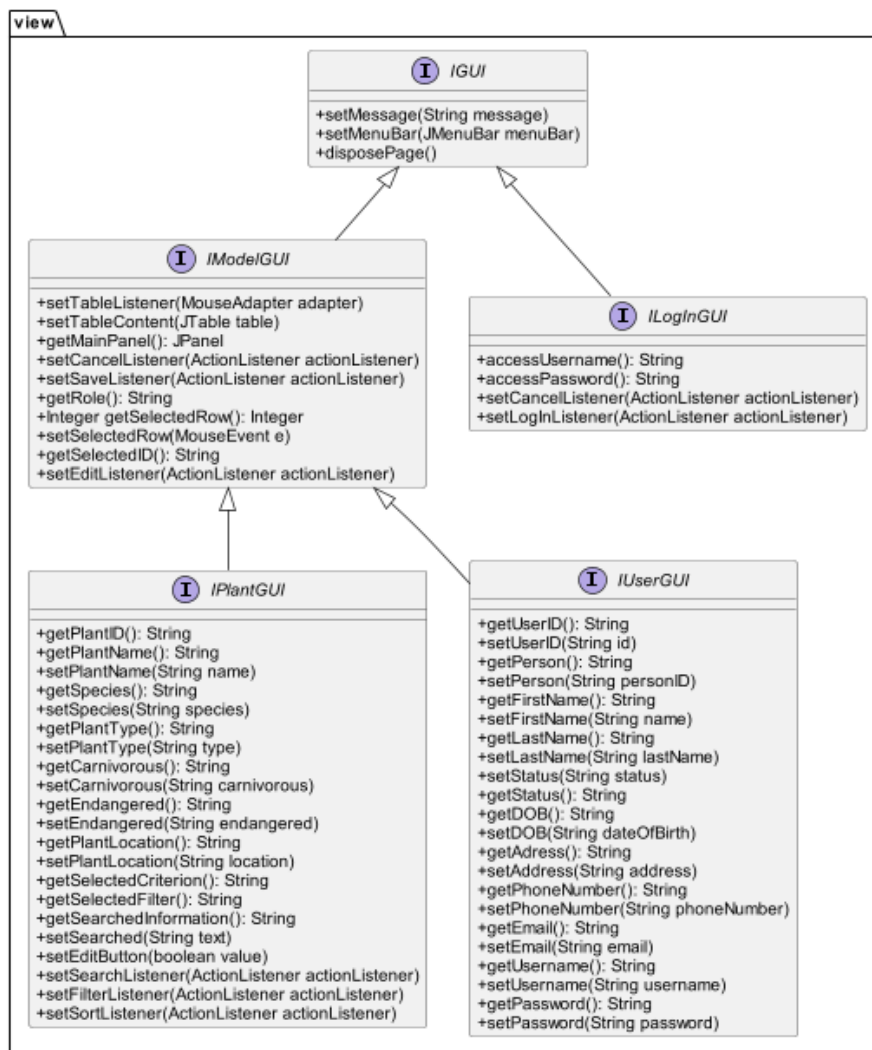
De exemplu, clasa PlantPresenter este responsabilă de evenimentele care au loc în pagina PlantPage sau UpdateNewPlant din view, realizând adăugarea, actualizarea, căutarea și ștergerea plantelor din baza de date, adăugarea listenerilor pentru butoane, setarea listei de plante în interfață sub forma unui tabel.

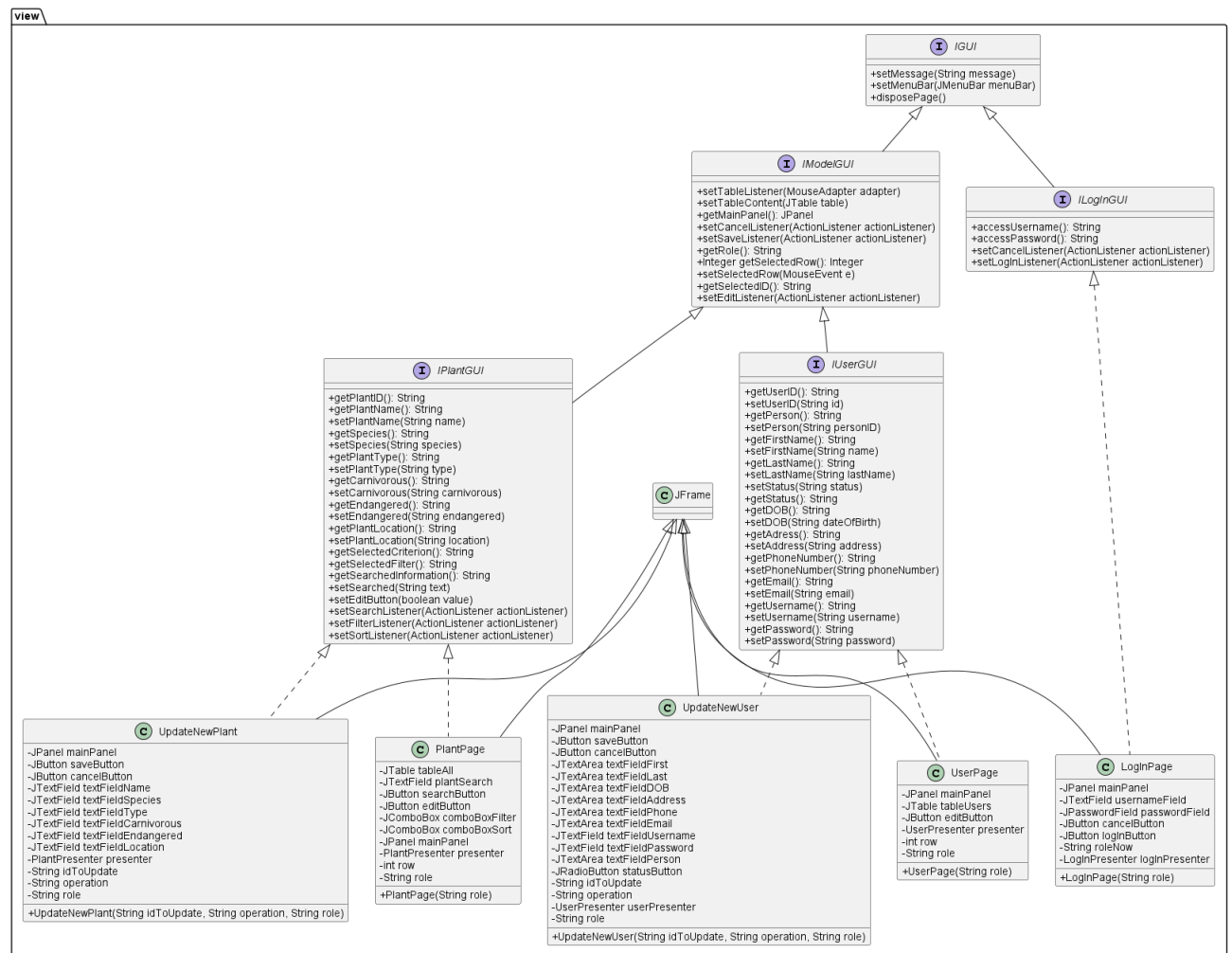
La fel UserPresenter realizează operațiile CRUD pentru utilizatori conform cu acțiunile din paginile UserPage și UpdateNewUser, iar LogInPresenter are funcția de a realiza autentificarea unui utilizator în sistem din pagina de log in.

3.6 Pachetul View

Pachetul view este folosit pentru a construi interfața grafică a aplicației, cu elementele senzitive care transmit spre procesare în presenter diferite semnale: butoane, checkbox-uri, opțiuni din meniu și tabele de date.

Pachetul constă dintr-o ierarhie de interfețe și clasele care le implementează metodele, pentru setarea câmpurilor textuale ale aplicației, adăugarea listenerilor pentru evenimente din pachetul presenter, și crearea ferestrei care apare pe ecran la rulare.



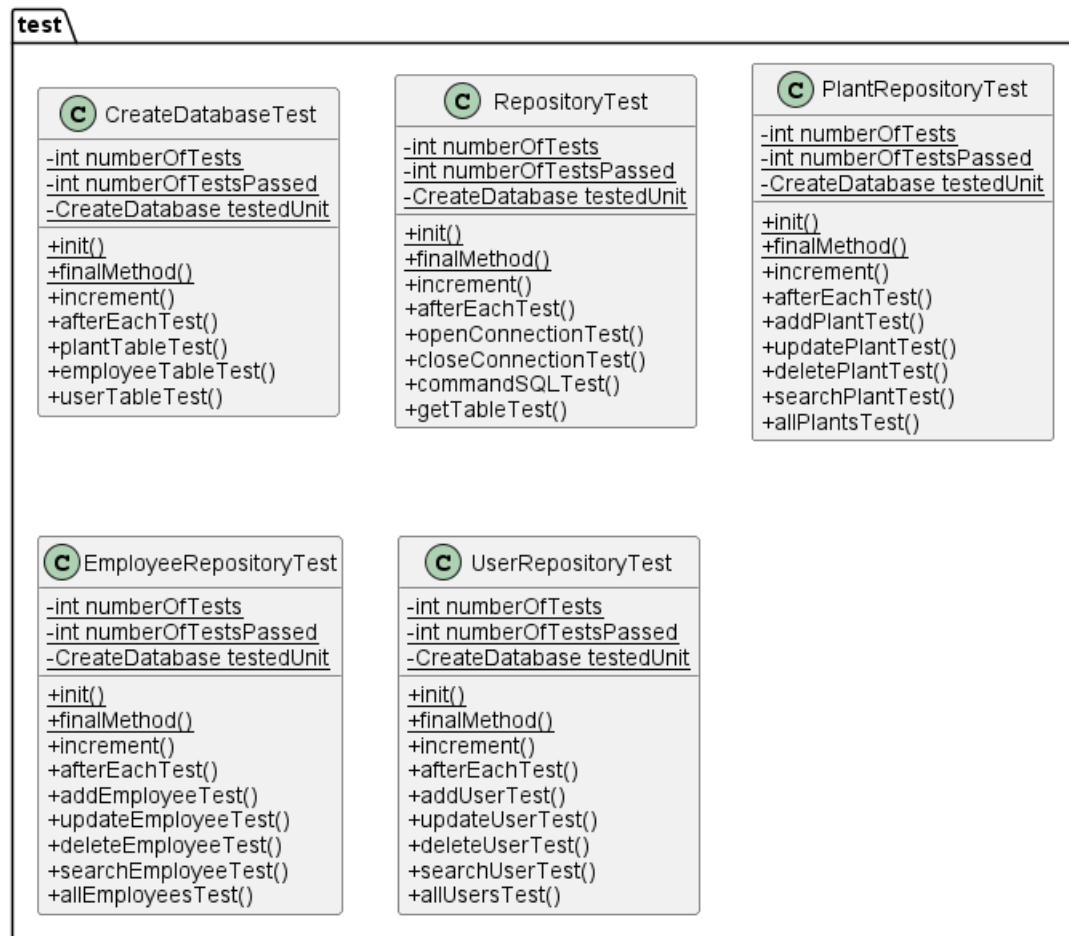


Interfețele sunt: IGUI (cea mai abstractă), extinsă de ILoginGUI (pentru pagina de autentificare) și IModelGUI, ultima fiind la rândul ei extinsă de interfețele IPlantGUI și IUserGUI (pentru clasele din model).

Clasele care implementează interfețele sunt chiar paginile care se deschid în fereastra aplicației, și anume PlantPage (afișarea tuturor plantelor), UpdateNewPlant (editare/adăugare plantă nouă), UserPage (afișarea tuturor utilizatorilor și datelor despre angajații respectivi), UpdateNewUser (editare/adăugare utilizator nou în sistem) și LoginPage (autentificare).

3.7 Pachetul Test

Pachetul test a fost utilizat pentru a proiecta teste unitare JUnit pentru clasele modelului repository, importante în efectuarea persistenței datelor din aplicație.



Clasele de test sunt CreateDatabaseTest, RepositoryTest, PlantRepositoryTest, UserRepositoryTest și EmployeeRepositoryTest, fiecare cu testele specifice clasei căreia îi corespund și conținând câte o instanță a obiectului pe care testează adăugarea, editarea, ștergerea și listarea datelor din și în baza de date.

4 Faza de implementare - Aplicatia

4.1 Instrumentele utilizate

Pentru crearea bazei de date am utilizat sistemul de gestionare de baze de date relaționale MySQL, cu sintaxa SQL clasică și driver-ul jdbc pentru a realiza conexiunea din Java.

Partea de cod Java am implementat-o folosind IDE-ul IntelliJ Idea, care integrează și partea de testing to framework-ul JUnit.

Am ales limbajul Java, deoarece este orientat pe obiect, ceea ce este esențial pentru a modela utilizatorii și plantele din baza de date, dar și din motive de simplitate în ceea ce privește testarea și dezvoltarea unei interfețe grafice interactive (biblioteca Java Swing).

4.2 Aplicatia

La deschiderea aplicației, utilizatorului (care încă nu are niciun rol asignat, respectiv primește eticheta de "visitor") îi apare pagina de prezentare a tuturor plantelor din baza de date, într-un

tabel care are ca coloane attributele fiecărei intrări din baza de date: id-ul, denumirea, specia, tipul, dacă este carnivoră sau nu, dacă este pe cale de dispariție și zona din grădina botanică unde se găsește.

Plant page

Menu

Search

Sortare dupa: Filtare dupa:

id	plantName	species	plantType	carnivorous	endangered	location
1	lalea	tulipa gesneriana	magnoliophyta	no	no	right entrance
2	trandafir	rosa chinensis	magnoliophyta	no	no	left entrance
3	capcana lui Venus	dionaea muscipula	magnoliophyta	yes	no	left entrance
4	bujor	paeonia peprina	magnoliophyta	no	yes	center
5	garofita	dianthus callizonus	magnoliophyta	no	yes	right entrance
6	ginkgo biloba	ginkgo biloba	gymnospermae	no	no	back area
7	pin	pinus pinea	gymnospermae	no	no	left entrance
8	stejar	quercus alba	angiospermae	no	yes	right entrance
9	salcie	salix triandra	angiospermae	no	no	back area

Pagina conține și opțiuni de sortare (după specie și după tip), opțiuni de filtrare conform cu textul introdus în bara de căutare (după tip, specie, locație etc.) și un buton de căutare care permite returnarea plantei cu denumirea sau specia indicată în bara de căutare. Dacă pe orice intrare din tabel se dă dublu click, iar utilizatorul este un angajat autentificat, va apărea o fereastră de dialog din care se va putea efectua ștergerea intrării respective.

Username:

Password:

Cancel Log in

Din meniul aplicației, un vizitator poate intra pe pagina de autentificare, unde își va introduce numele de utilizator și parola pentru a accesa funcționalități specifice rolului său în sistem.

Atunci când se detectează apăsarea butonului "Log in", se începe verificarea credențialelor. Se caută utilizatorul, iar dacă acesta nu există sau a indicat parola greșită interfața generează un mesaj de eroare. În caz că autentificarea a avut loc cu succes, se verifică atributul de "admin_status" al utilizatorului și se identifică rolul acestuia: "employee" sau "administrator".

Se deschide iarăși pagina de prezentare a plantelor din aplicație, dar de data aceasta meniul este actualizat cu funcțiile specifice rolului utilizatorului autentificat. De asemenea, dacă utilizatorul este un angajat, acesta are acces la butonul de "edit" din josul paginii, de unde la selectarea unei linii din tabel, se deschide pagina de editare a respectivei intrări.

User administration page

Menu

userID	person	password	username	adminSta...	id	firstName	lastName	dateOfBirth	address	phoneNu...	email
1	1	pass	dv_one	no	1	D	V	2000-03-...	str. Unu, 5	0733222...	d@yaho...
2	2	1234	flor	no	2	F	R	1998-06-...	str. Cires...	0764442...	flor@gm...
3	3	admin	admin	yes	3	A	M	2001-11-...	str. Avg, 10	07999333	a.m@ya...
4	4	SeCuRe	user_lk	no	4	L	K	1999-01-...	str. Doi, 76	0711112...	lk@gmai...

Edit

Dacă utilizatorul autentificat are însă rolul de "administrator", în meniu apare și opțiunea de vizualizare a tuturor utilizatorilor din sistem. Pagina UserPage afișează sub forma unui nou tabel datele despre utilizatori, dar și despre angajații cărora le corespund: nume, prenume, data nașterii, adresa, numărul de telefon și email-ul.

Un administrator poate realiza editarea utilizatorilor, modificând username-ul, parola sau statutul de administrator al fiecăruia. Datele despre angajat sunt imutabile, decât dacă se creează un utilizator nou, în pagina de creare existând opțiunea de a adăuga date și pentru un angajat nou.

O altă operație CRUD care se poate face este ștergerea unui utilizator și respectiv a angajatului asociat, prin apăsarea dublă a oricărei intrări din tabel.