



**UNIVERSITATEA
TEHNICĂ**
DIN CLUJ-NAPOCA

Tema 2

Proiectare software

Autor: Volcov Sabina

Grupa: 30231

FACULTATEA DE AUTOMATICĂ
ȘI CALCULATOARE

aprilie 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	7
3.5	Pachetul ViewModel	7
3.6	Pachetul View	9
4	Faza de implementare - Aplicatia	9
4.1	Instrumentele utilizate	9
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 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:
 1. proiectarea dată de diagrama de clase;
 2. unul dintre următoarele limbaje de programare: C#, C++, Java, Python.
- 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;
- Căutarea unei plante după denumire.

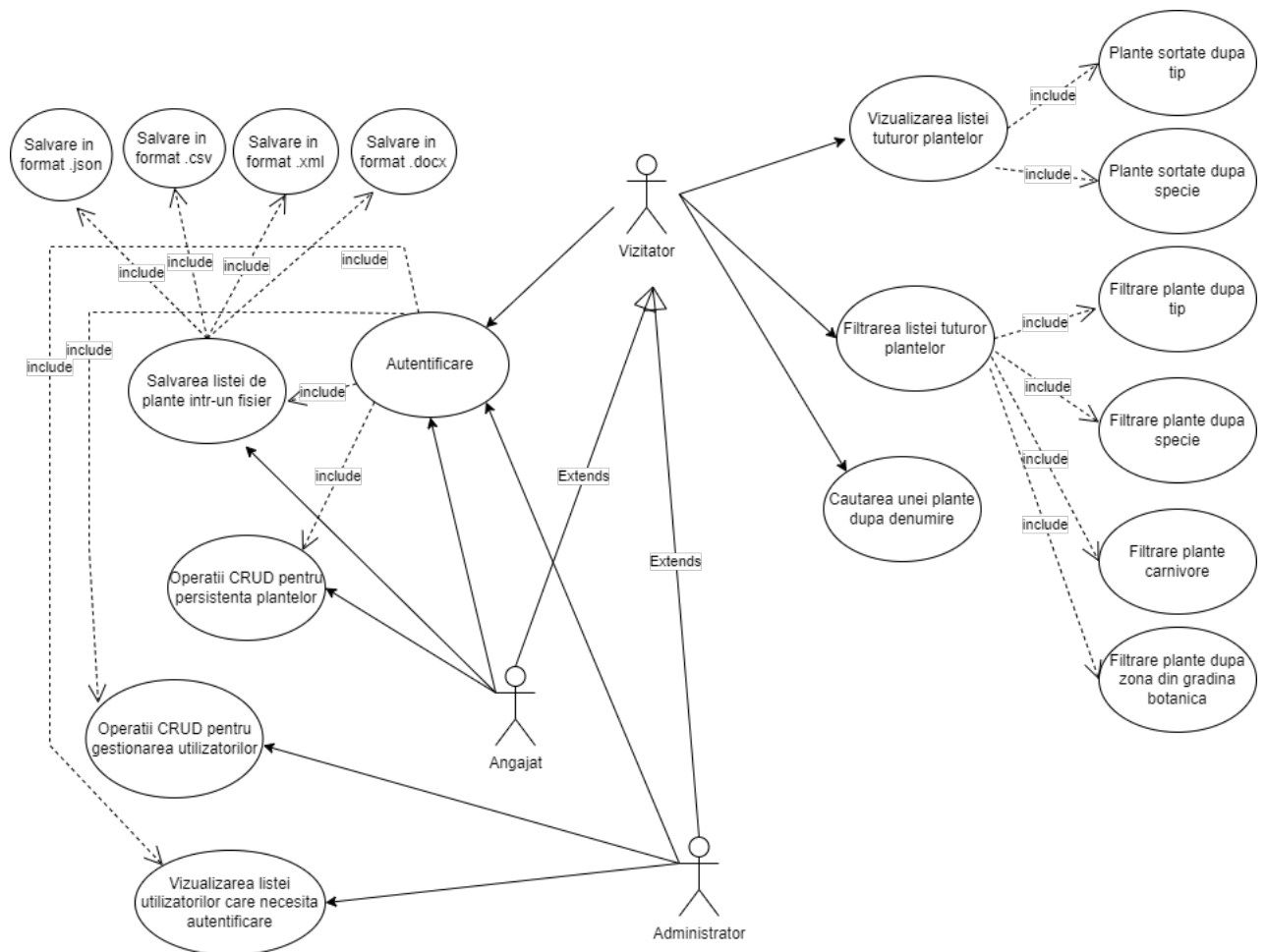
Utilizatorii de tip angajat al grădinii botanice pot efectua următoarele operații după autentificare:

- 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.

2.2 Diagrama cazurilor de utilizare



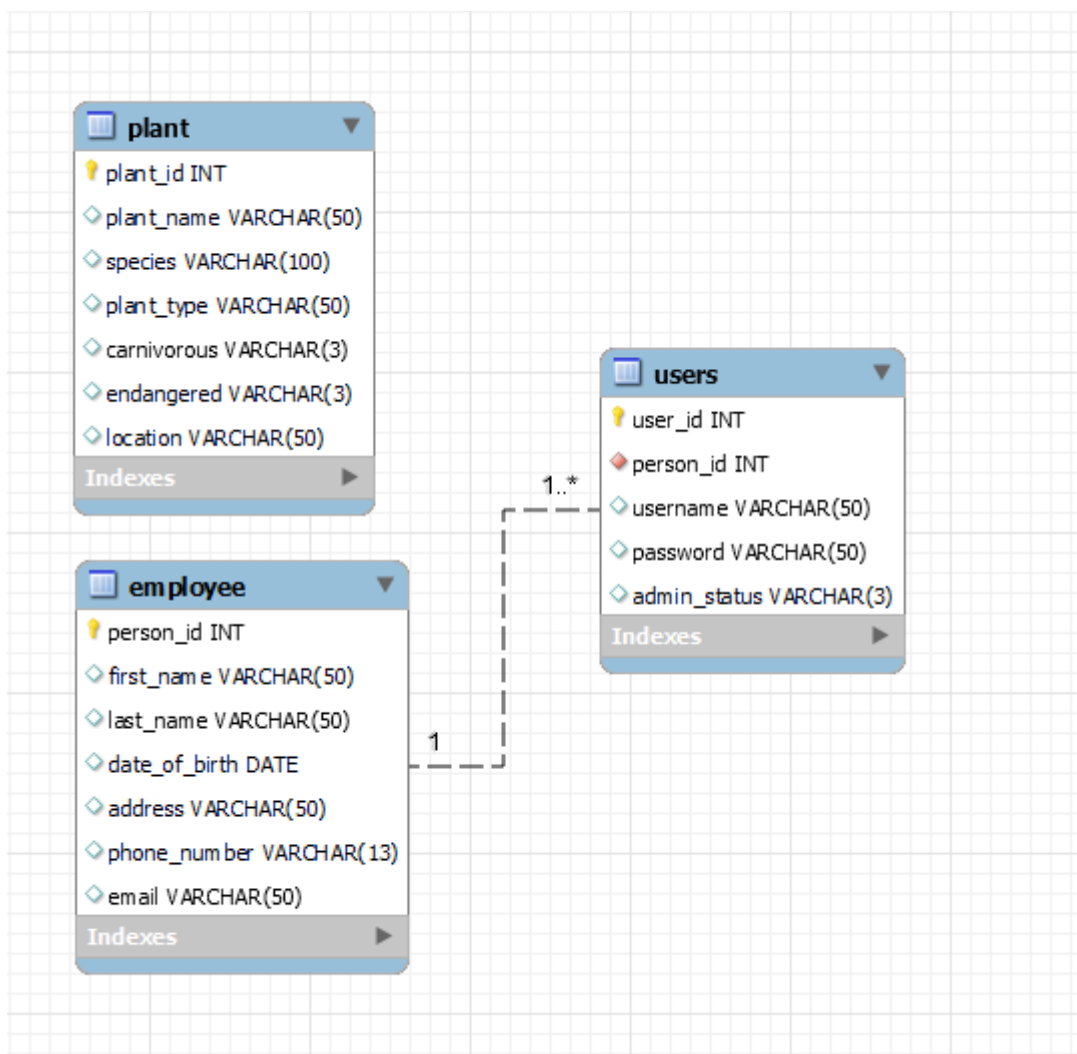
Din diagramă reies cei 3 actori care apar în aplicație, și anume vizitatorii anonimi, care pot vizualiza, sorta, filtra și căuta după denumire plantele din baza de date, angajații autentificați care moștesc use-case-urile vizitatorii și pot pe deasupra realiza operații CRUD pentru persistența plantelor și salvarea listei de plante într-un fișier în 4 formate, și administratorii, care la fel moștesc 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

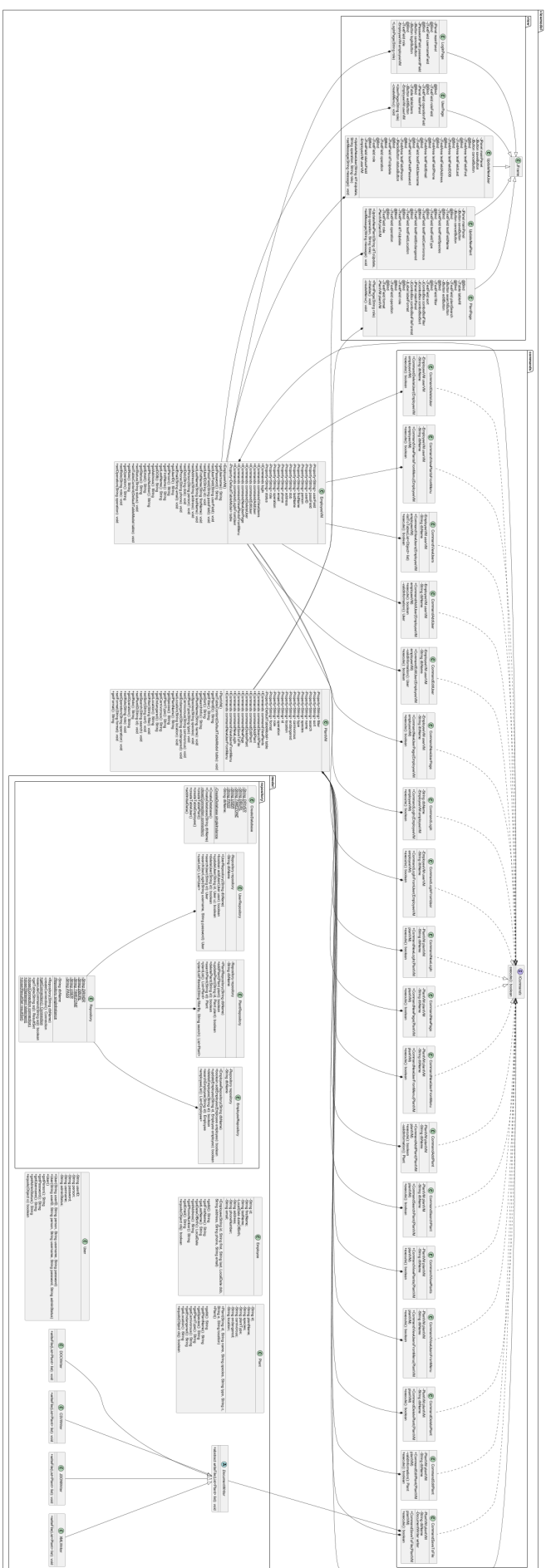
Sistemul de gestionare a bazei de date folosit este MySQL, conectarea cu mediul Java realizându-se prin intermediul driver-ului jdbc.

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).



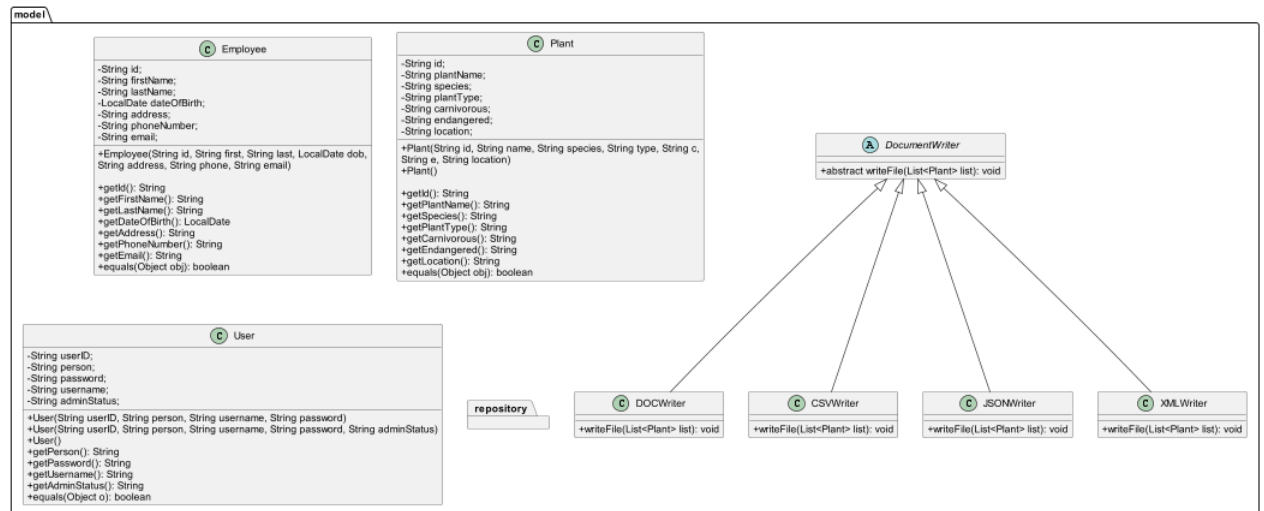
3.2 Diagrama completă de clase

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.

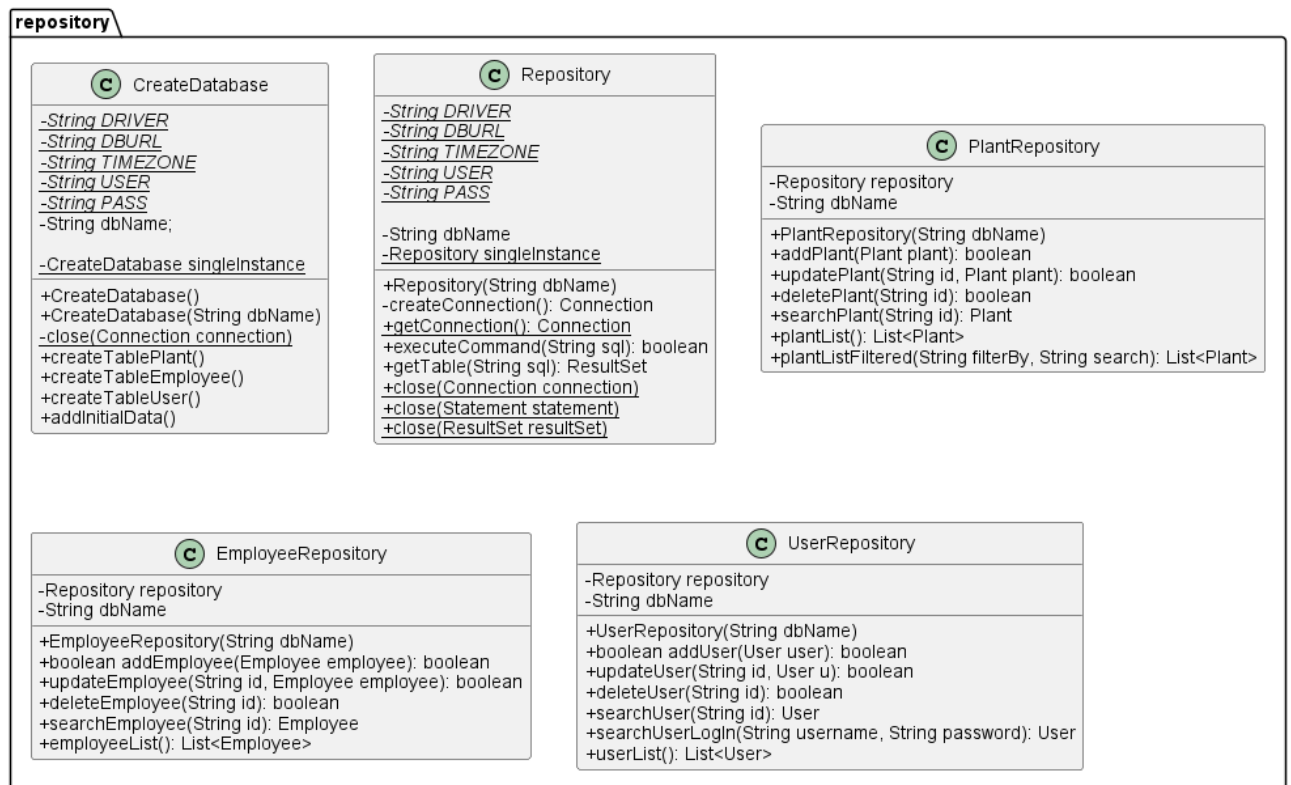


Pachetul model este format din clasele Plant, Employee și User, fiecare cu propriii constructori pentru instanțele care vor corespunde intrărilor din tabelele respective din baza de date și vor modela o grădină botanică reală.

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

Clasa abstractă DocumentWriter definită în pachetul model devine utilă în momentul salvării listei de plante într-un fișier, conținând o singură metodă, `writeFile()`, care mai apoi este implementată de fiecare clasă copil în parte: DOCWriter, CSVWriter, XMLWriter și JSONWriter. Această structură va respecta principiul Open-Closed, adăugarea unui format nou fiind posibilă prin simpla creare a unei clase noi care să extindă DocumentWriter.

3.4 Subpachetul Repository



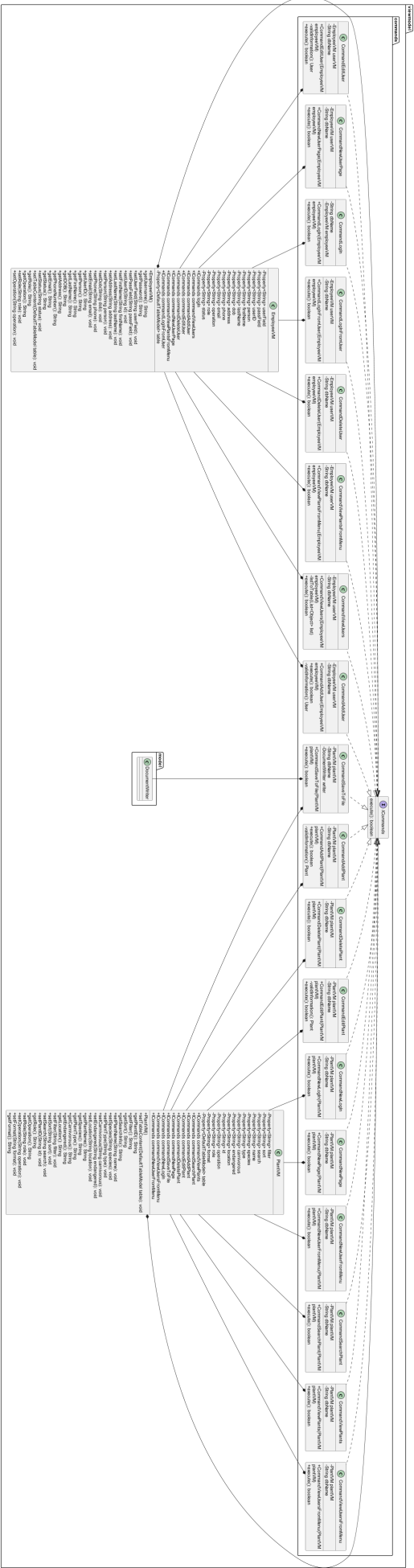
Subpachetul repository din cadrul modelului are rolul de a crea baza de date si de a gestiona operatiile CRUD care se pot efectua pe obiectele din aplicatie.

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 ViewModel

Pachetul viewmodel este cel responsabil de logica de business, cuprinzând un subpachet "commands", unde clasele implementează interfața ICommands și unica metodă corespunzătoare acesteia, "execute". Fiecare clasă comandă este asociată unei acțiuni precum adăugarea, editarea, ștergerea sau afișarea datelor, astfel încât varianta finală a pachetului include următoarele clase: CommandAddPlant, CommandEditPlant, CommandDeletePlant, CommandSaveToFile, CommandSearchPlant, CommandViewPlants, CommandLogIn, CommandNewPage, CommandAddUser, CommandEditUser, CommandDeleteUser, CommandViewUsers etc..

În afară de acestea, viewmodel are și 2 clase EmployeeVM și PlantVM, necesare pentru a face legătura între interfața grafică și evenimente, efectuându-se un data binding implicit între câmpurile de date din iview și proprietățile folosite în cadrul comenzilor din viewmodel.

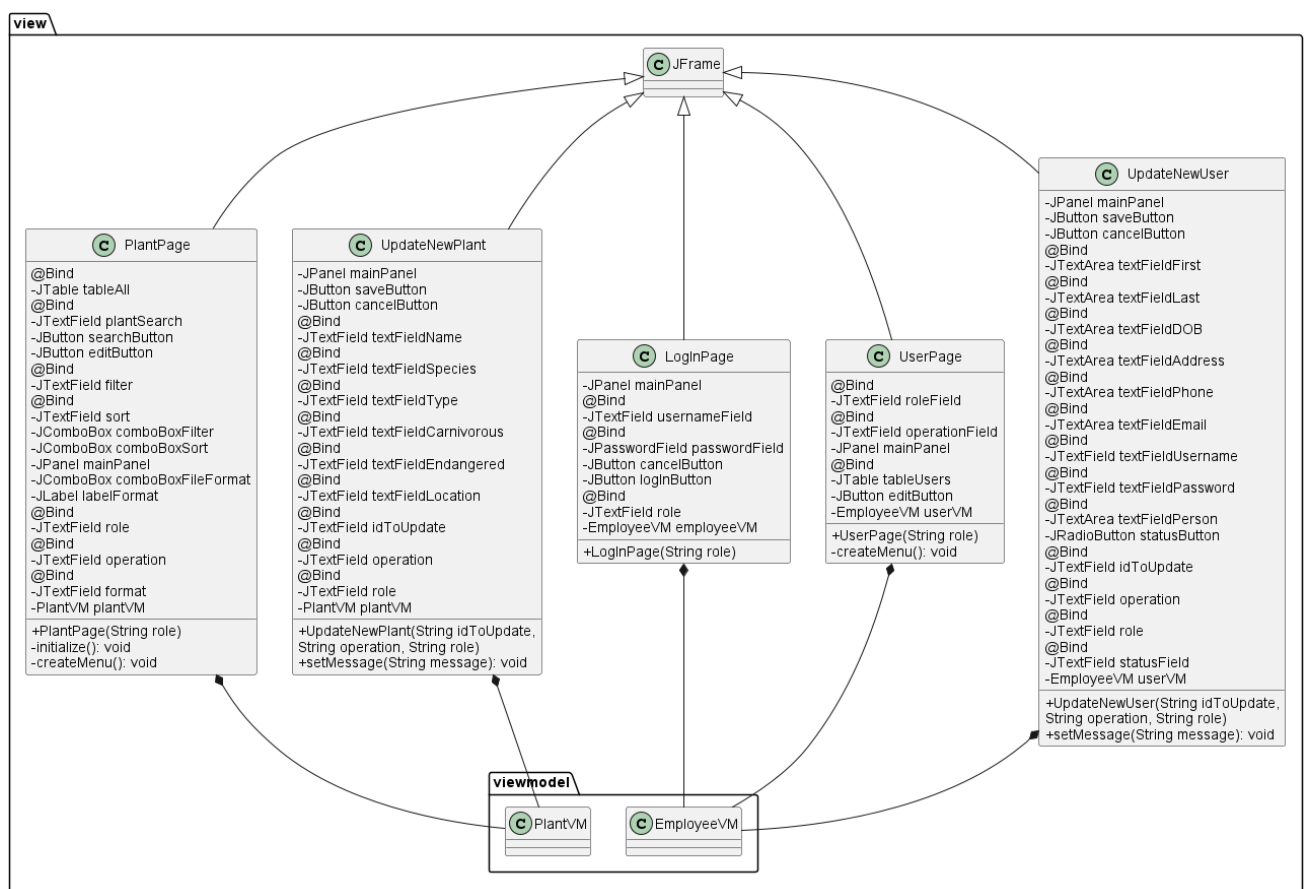


3.6 Pachetul View

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

Pachetul constă din clasele PlantPage (construiește pagina pentru 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).

Obiectivul pachetului este procesarea evenimentelor generate de interacțiunea utilizatorului cu elementele din interfața grafică și crearea ferestrei care apare pe ecran la rulare. Atunci când se dorește executarea unei acțiuni, clasa din view apelează o comandă din viewmodel, care se ocupă de procesarea acesteia.



4 Faza de implementare - Aplicatia

4.1 Instrumentele utilizate

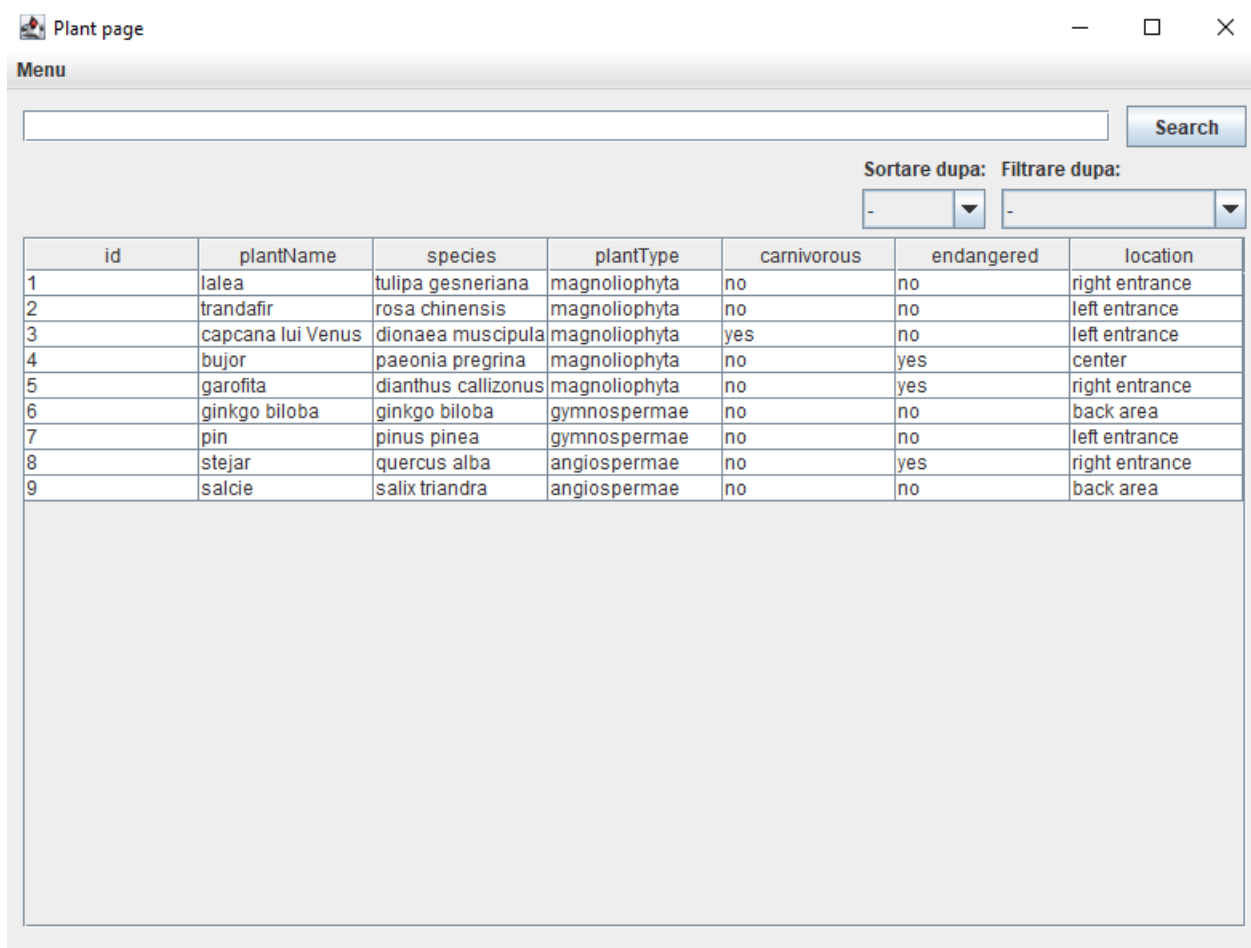
Pentru crearea bazei de date am utilizat sistemul de gestionare de baze de date relaționale MySQL, cu sintaxa SQL clasică și driver-ul jdbc pentru a realiza conexiunea din Java.

Partea de cod Java am implementat-o folosind IDE-ul IntelliJ Idea, care oferă un mediu de dezvoltare intuitiv și rapid inclusiv pentru utilizarea bibliotecii Swing pentru construirea interfeței grafice.

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 asigurat, respectiv primește eticheta de "visitor") îi apare pagina de prezentare a tuturor plantelor din baza de date, într-un tabel care are ca coloane atributele 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.



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.

De asemenea, după autentificare, pe pagina principală apare un nou meniu, pentru salvarea listei de plante în diverse formate: .csv, .json, .xml și .doc ;și un buton care permite editarea intrării selectate din tabelul de plante.

Plant page

Menu

Salvare in format:

Sortare dupa:

Filtrare dupa:

-

▼

-

▼

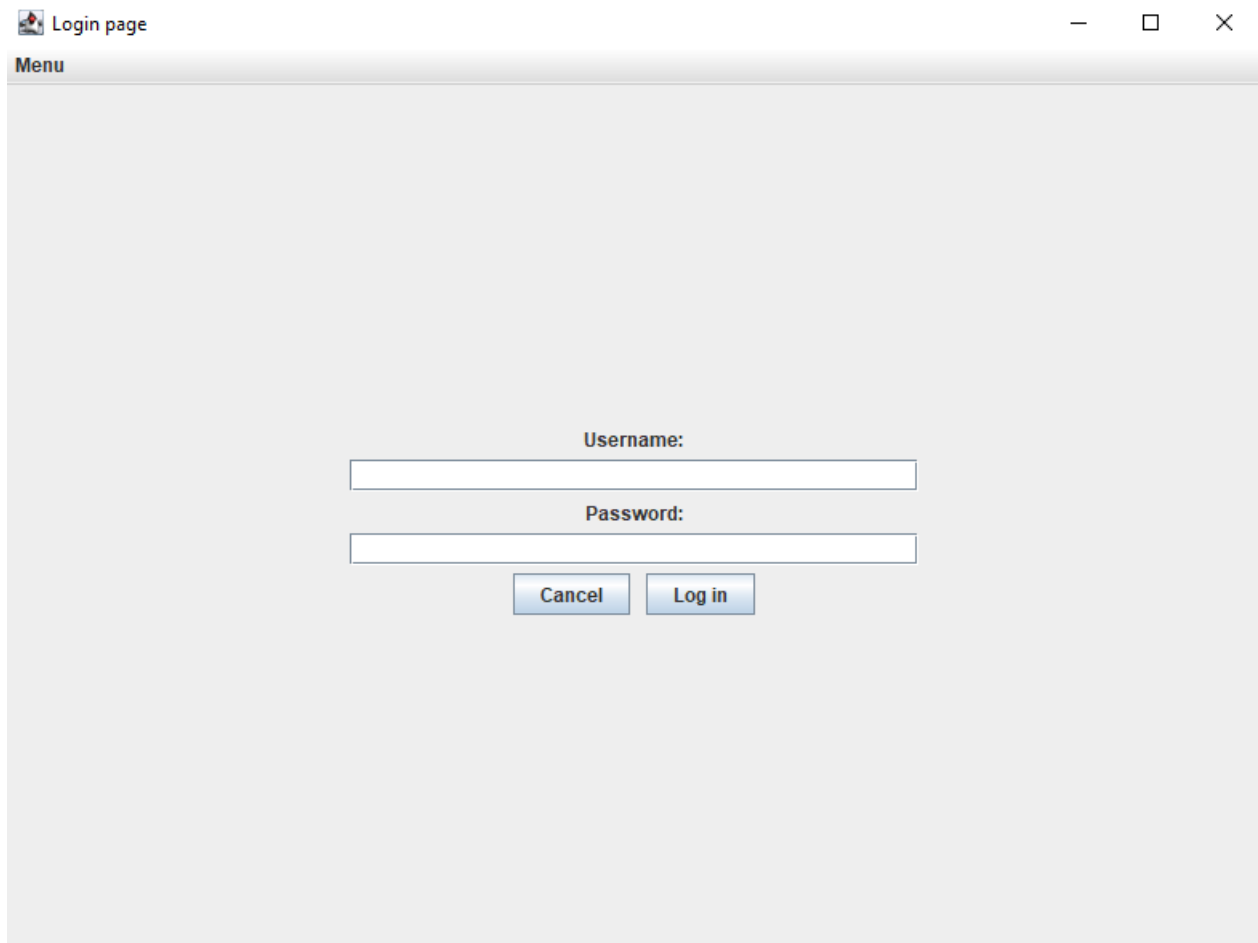
-

▼

id	plantName	species	plantType	is	endangered	location
1	lalea	tulipa gesneriana	magnoliophyta		no	right entrance
2	trandafir	rosa chinensis	magnoliophyta		no	left entrance
3	capcana lui Venus	dionaea muscipula	magnoliophyta		no	left entrance
4	bujor	paeonia pregrina	magnoliophyta		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

Edit

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.



The image shows a web browser window titled "Login page". The window has a standard title bar with minimize, maximize, and close buttons. Below the title bar is a "Menu" bar. The main content area is light gray and contains a login form. The form has two input fields: "Username:" and "Password:". Below the "Password:" field are two buttons: "Cancel" and "Log in".

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.

User administration page

First name: A

Last name: M

Person id: 3

Date of birth: 2001-11-19

Address: str. Avg, 10

Phone number: 07999333

Email: a.m@yahoo.com

Username: admin

Password: admin

Admin status: ☒ activate

Cancel Save

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.