



**UNIVERSITATEA  
TEHNICĂ**  
DIN CLUJ-NAPOCA

---

## **Tema 3**

*Proiectare software*

---

Autor: Volcov Sabina  
Grupa: 30231

FACULTATEA DE AUTOMATICĂ  
ȘI CALCULATOARE

aprilie 2024

# Cuprins

<b>1</b>	<b>Problema de rezolvat</b>	<b>2</b>
<b>2</b>	<b>Faza de analiza - Diagrama cazurilor de utilizare</b>	<b>2</b>
2.1	Enuntul problemei	2
2.2	Diagrama cazurilor de utilizare	3
<b>3</b>	<b>Faza de proiectare - Diagrame de clase</b>	<b>4</b>
3.1	Structura bazei de date	4
3.2	Diagrama completă de clase	4
3.3	Pachetul Model	6
3.3.1	Șablonul creațional Factory Method	6
3.4	Subpachetul Repository	7
3.5	Pachetul Controller	7
3.6	Pachetul View	8
3.7	Șablonul Observer	9
<b>4</b>	<b>Faza de implementare - Aplicatia</b>	<b>9</b>
4.1	Instrumentele utilizate	9
4.2	Aplicatia	9

# 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 vor realiza:
  - diagrama de clase respectând arhitectura MVC și folosind:
    - \* șablonul de proiectare comportamental Observer;
    - \* un șablon de proiectare creațional.
  - 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.

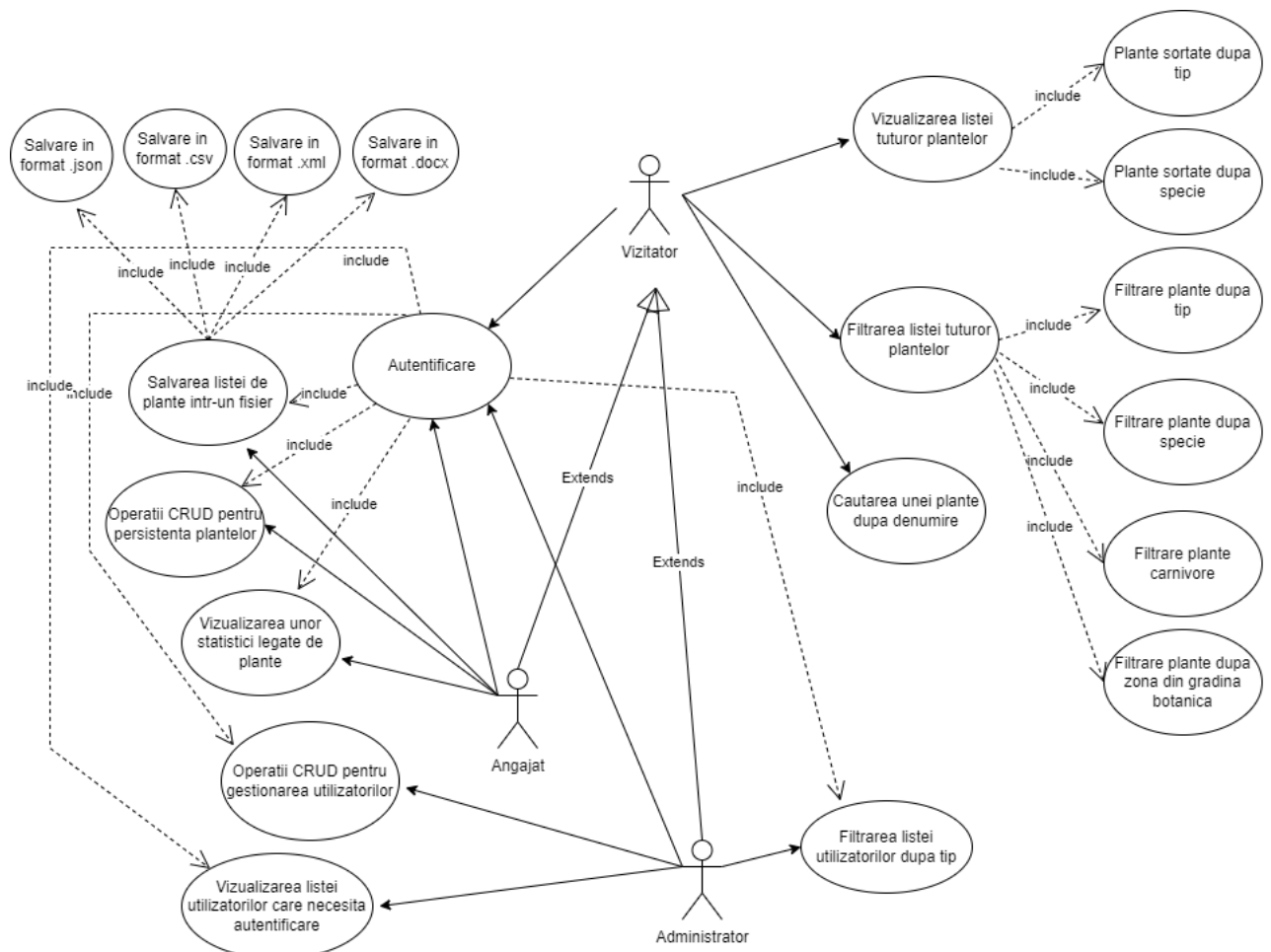
Utilizatorii de tip angajat al grădinii botanice pot efectua următoarele operații după autentificare:

- 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;
- Vizualizarea unor statistici legate de plantele din grădina botanică utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

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;
- Filtrarea listei utilizatorilor care necesită autentificare după tipul utilizatorilor.

## 2.2 Diagrama cazurilor de utilizare



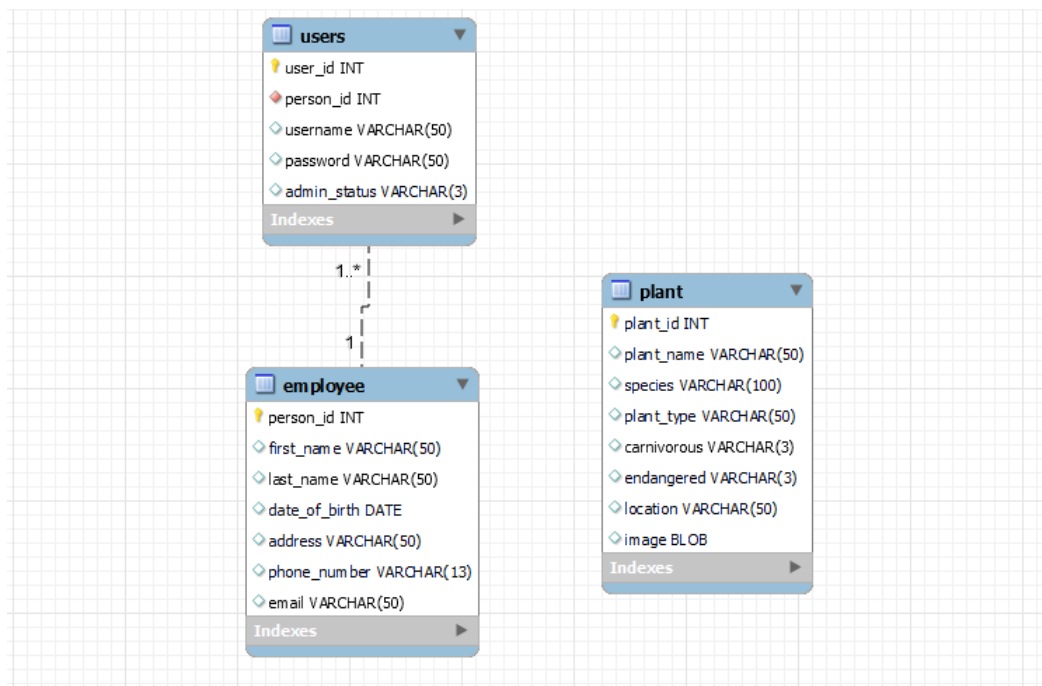
Din diagramă reies cei 3 actori care apar în aplicație, și anume vizitatorii anonimi, care pot vizualiza, sorta, filtra și căuta după denumire plantele din baza de date, angajații autentificați care moștenesc use-case-urile vizitatorii și pot pe deasupra realiza operații CRUD pentru persistența plantelor, salvarea listei de plante într-un fișier în 4 formate și vizualiza statistici legate de plantele din grădina botanică, 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, inclusiv filtrarea acestora după tipul de utilizator.

### 3 Faza de proiectare - Diagrame de clase

#### 3.1 Structura bazei de date

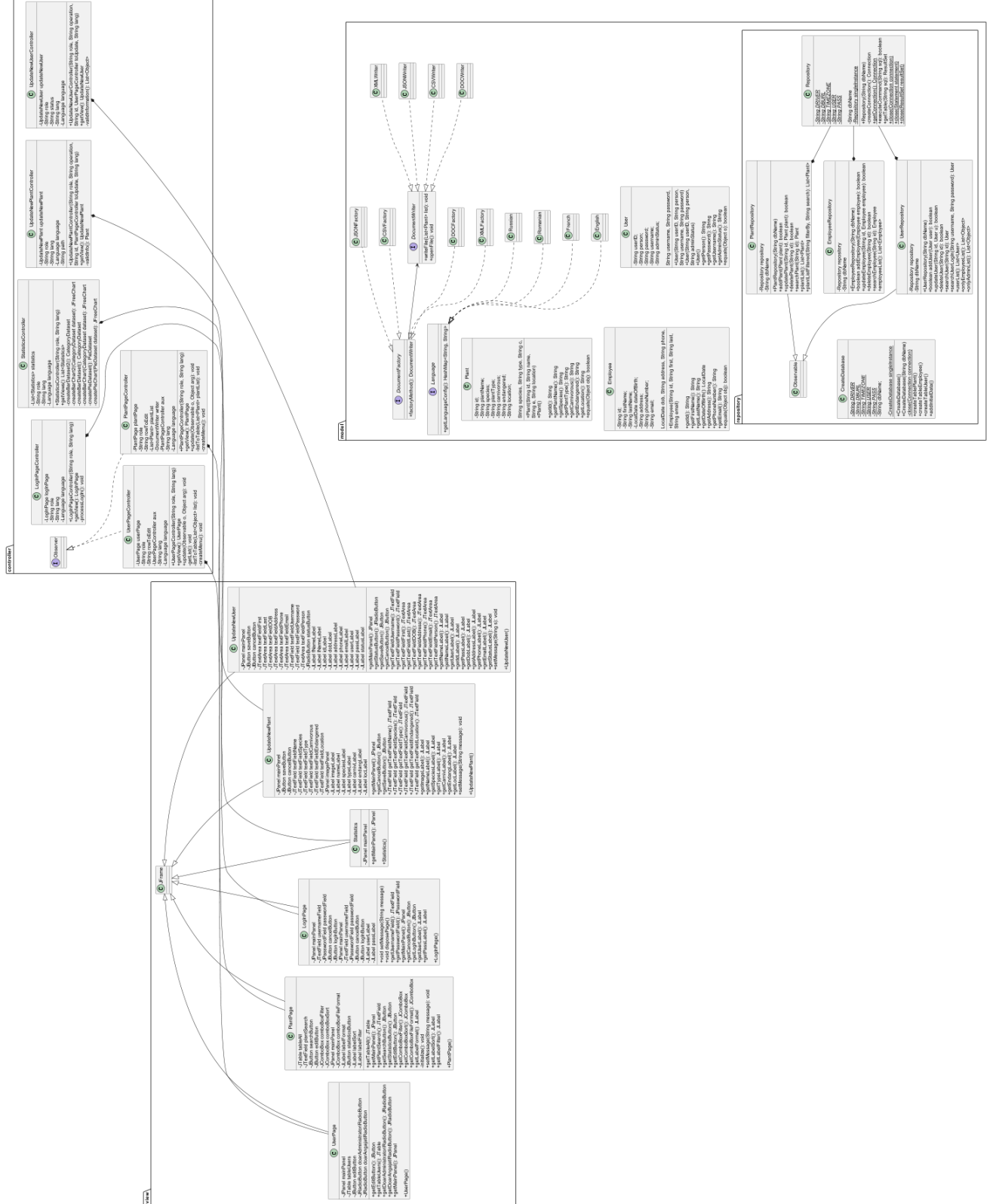
Sistemul de gestionare a bazei de date folosit este MySQL, conectarea cu mediul Java realizându-se prin intermediul driver-ului jdbc.

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, tip de alimentație și imagine reprezentativă), "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).



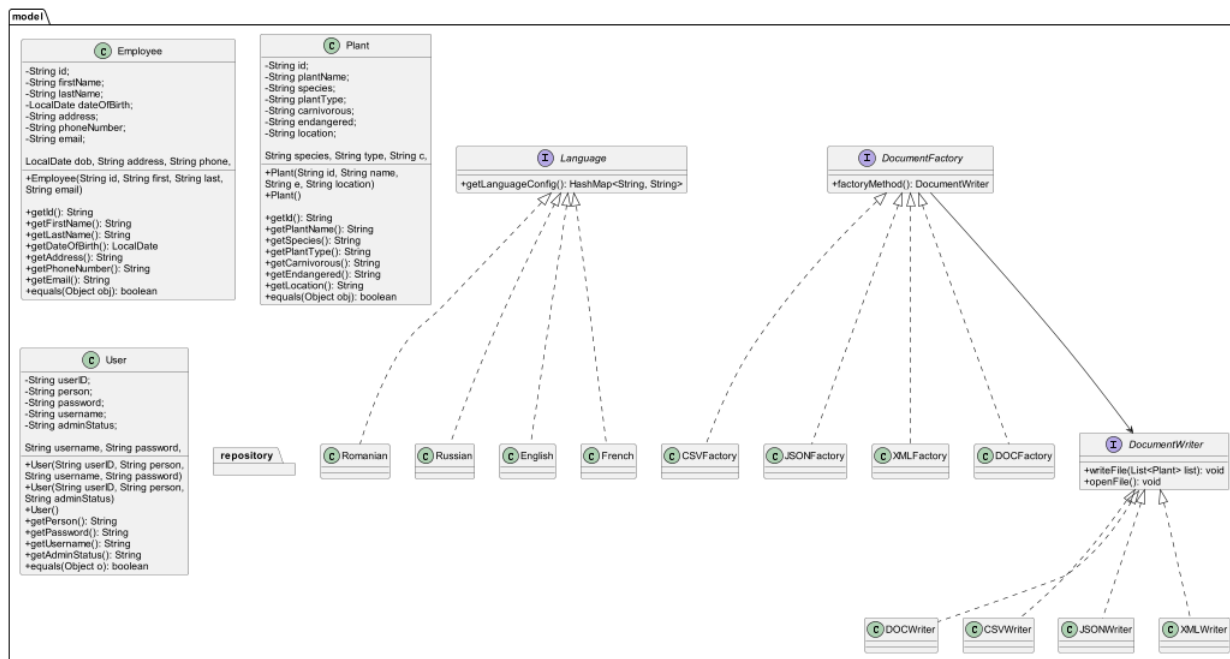
#### 3.2 Diagrama completă de clase

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 conține clasele Plant, 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.

Pentru a disponibiliza interfața grafică a aplicației în 4 limbi diferite, pachetul model include de asemenea interfața Language, cu metoda getLanguageConfig(), care va crea un dicționar de tip HashMap<String, String>, cu perechi cheie-valoare care vor reprezenta traducerile pentru elementele interfeței în fiecare limbă.

Clasele Romanian, English, French și Russian sunt cele care implementează interfața Language și vor fi responsabile de crearea dicționarului pornind de la fișierul .json corespunzător.

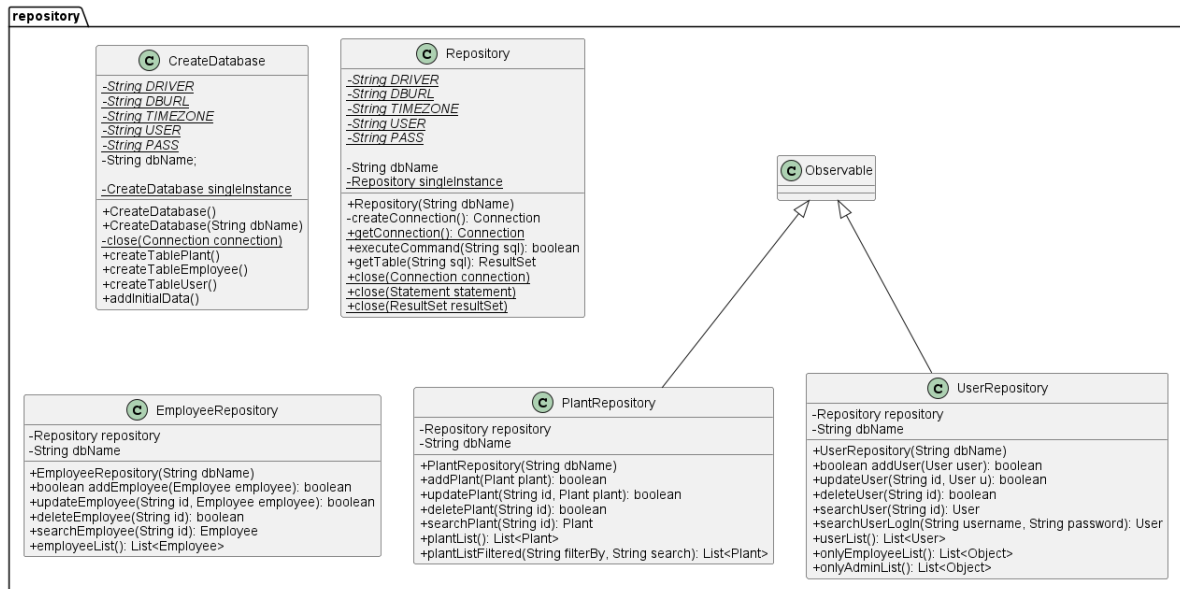
Interfețele DocumentWriter și DocumentFactory definite în pachetul model devin utile în momentul salvării listei de plante într-un fișier, conținând metode care mai apoi sunt implementate de fiecare clasă în parte. Această structură va respecta principiul Open-Closed, adăugarea unui format nou fiind posibilă prin simpla creare a unor clase noi care să implementeze DocumentWriter cu metodele writeFile() și openFile() și DocumentFactory.

#### 3.3.1 Șablonul creațional Factory Method

Clasele și interfețele din model utilizate la salvarea listei de plante în fișier respectă șablonul creațional FactoryMethod. Interfața DocumentFactory și clasele DOCFactory, XMLFactory, CSVFactory și JSONFactory reprezintă partea de Factory, care creează obiecte noi de tipul

produsului (DocumentWriter), iar interfața DocumentWriter și clasele DOCWriter, XMLWriter, CSVWriter și JSONWriter - partea de produs, cu metodele care definesc logica necesară scrierii în fișier.

### 3.4 Subpachetul Repository



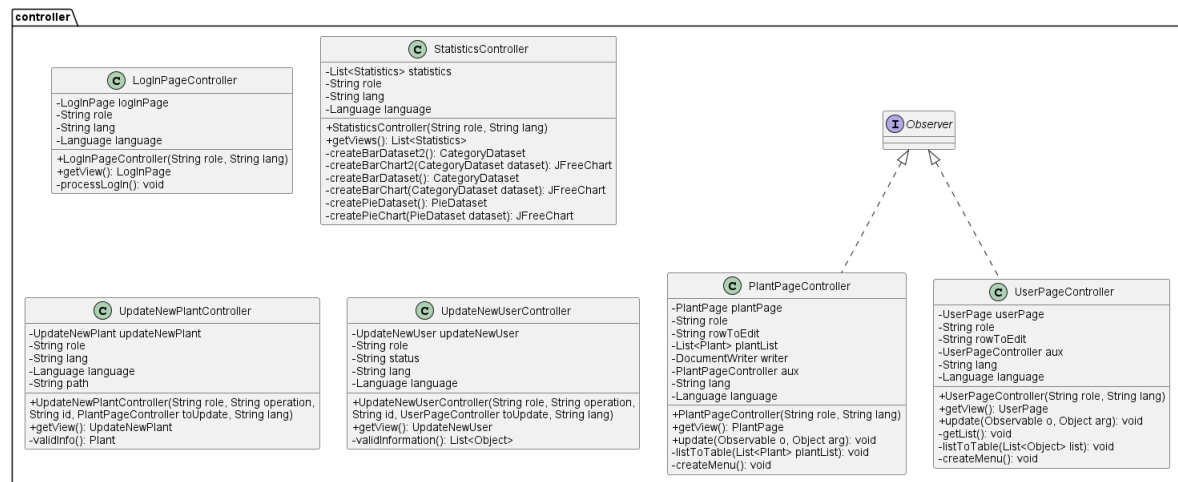
Subpachetul repository din cadrul modelului are rolul de a crea baza de date si de a gestiona operatiile CRUD care se pot efectua pe obiectele din aplicatie.

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 Controller

Pachetul controller este cel responsabil de executarea tuturor operațiilor și de procesarea evenimentelor din interfața grafică, comunicând cu modelul prin șablonul Observer și cu view-ul prin apelarea unor gettere. Fiecare clasă din view are un corespondent în controller, astfel încât să poată fi gestionată atât afișarea, cât și procesarea operațiilor. Clasele din acest pachet sunt PlantPageController, UserPageController, LogInController, UpdateNewUserController, UpdateNewPlantController și StatisticsController.



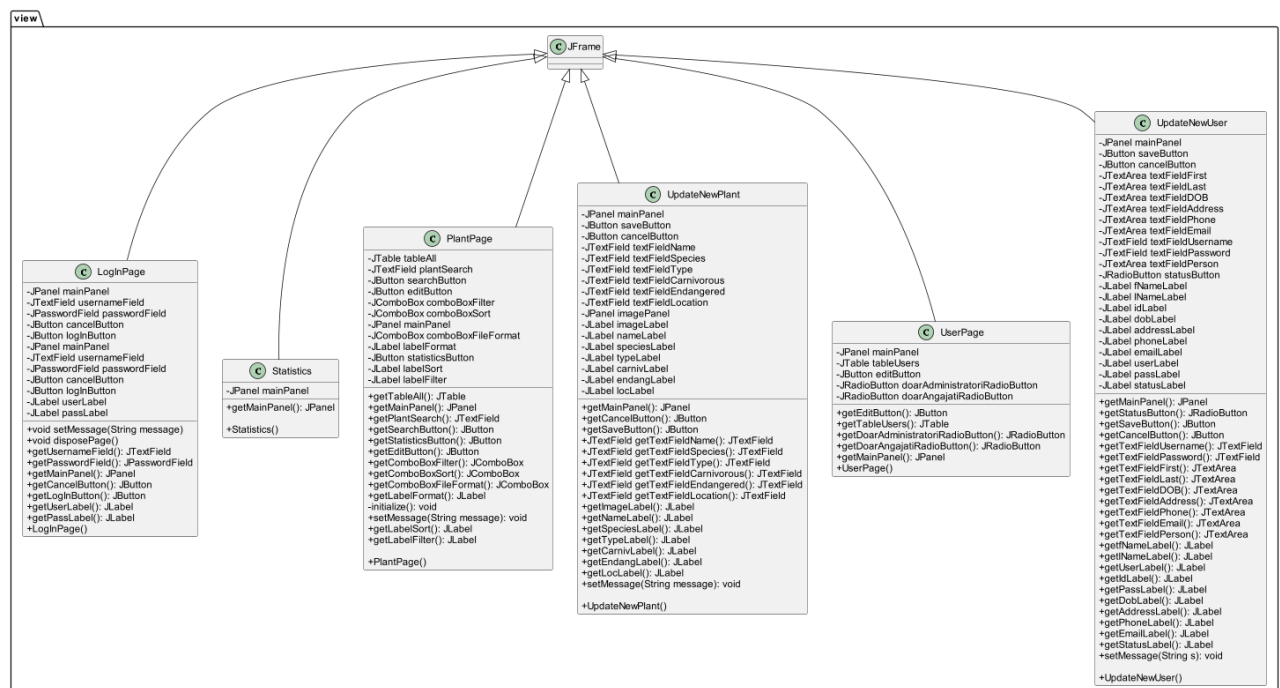


### 3.6 Pachetul View

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), Statistics (pagina în care se vor afișa grafice) și LoginPage (autentificare).

Rolul pachetului este doar de a crea interfața grafică și de a defini gettere pentru elementele acestea, astfel încât controller-ul fiecărei clase să aibă acces la ele și să poată gestiona diversele evenimente care pot să apară.



### 3.7 Șablonul Observer

Aplicația folosește șablonul Observer pentru a permite o interacțiune mai rapidă și mai eficientă între pachete, fără cuplare excesivă. Clasele `PlantRepository` și `UserRepository` din model funcționează ca clase observabile, care notifică atunci când are loc o schimbare în setul de date, iar clasele `PlantPageController` și `UserPageController` din controller reprezintă observatorii care actualizează tabelul cu date din interfața grafică cu modificările din model.

Clasele din model extind clasa `Observable`, iar cele din controller implementează interfața `Observer`, care le permite să folosească metoda `update` pentru actualizarea interfeței grafice.

Așadar, pachetul `view` și `controller`, respectiv `view` și `model` sunt decuplate, dar funcționalitatea dorită rămâne disponibilă.

## 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. Pentru a obține statisticile reprezentate sub formă de grafice am utilizat pachetul Maven JFreeChart, disponibil online.

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 atributele fiecărei intrări din baza de date: id-ul, denumirea, specia, tipul, dacă este carnivoră sau nu, dacă este pe cale de dispariție, o imagine reprezentativă și zona din grădina botanică unde se găsește.









În partea de sus a paginii există 2 meniuri, unul din care vor fi disponibile funcționalitățile permise utilizatorului curent și unul din care se poate alege limba: română, engleză, franceză sau rusă.

Pagina plante

Meniu
Limba

Caută

Sortare după:
Filtrare după:

id	plantName	species	plantType	carnivorous	endangered	location	img
1	lalea	tulipa gesneri...	magnoliophyta	no	no	right entrance	
2	trandafir	rosa chinensis	magnoliophyta	no	no	left entrance	
3	capcana lui Ve...	dionaea musc...	magnoliophyta	yes	no	left entrance	
4	bujor	paeonia pregi...	magnoliophyta	no	yes	center	
5	garofita	dianthus calliz...	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	

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, un buton pentru vizualizarea statisticilor și un buton care permite editarea intrării selectate din tabelul de plante.

Pagina plante

Meniu Limba

Caută


Salvare în format: Sortare după: Filtrare după:

id	plantName	species	plantType	carnivorous	endangered	location	img
1	lalea	tulipa gesneri...	magnoliophyta	no	no	right entrance	
2	trandafir	rosa chinensis	magnoliophyta	no	no	left entrance	
3	capcana lui Ve...	dionaea musc...	magnoliophyta	yes	no	left entrance	
4	bujor	paeonia pregi...	magnoliophyta	no	yes	center	
5	garofita	dianthus calliz...	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	

Statistici Editare

Pagina de editare și cea de creare sunt asemănătoare, permițând utilizatorului să introducă date noi peste cele existente sau ca intrare nouă, dar și să aleagă o imagine de pe dispozitiv în caz de adăugare.

Pagina plante



Denumire planta:

Specie:

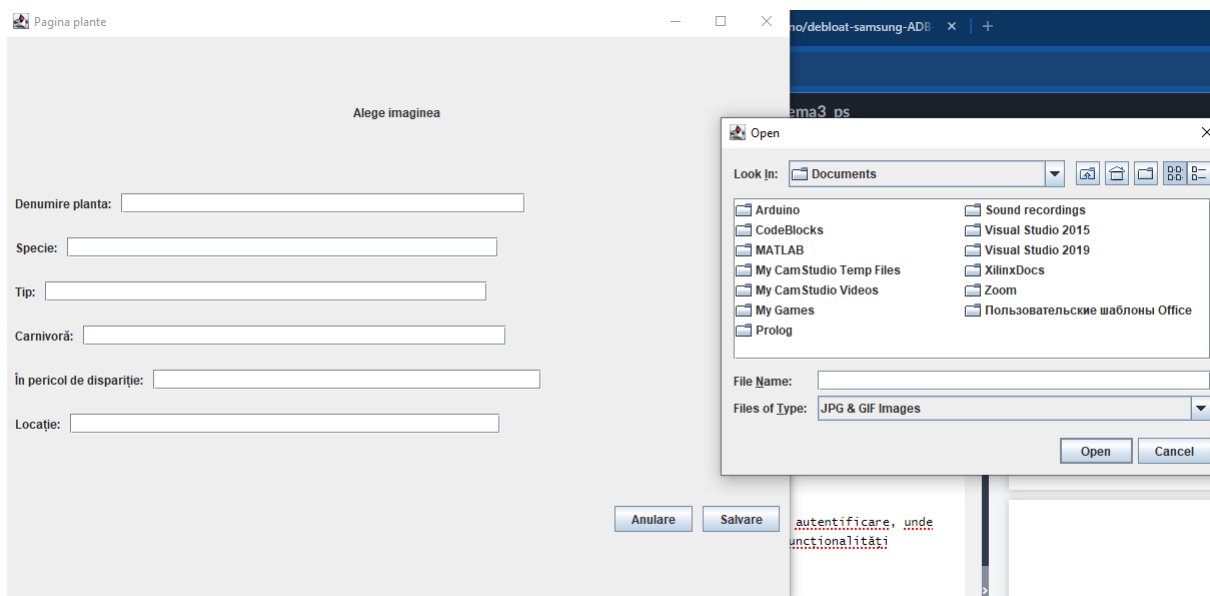
Tip:

Carnivoră:

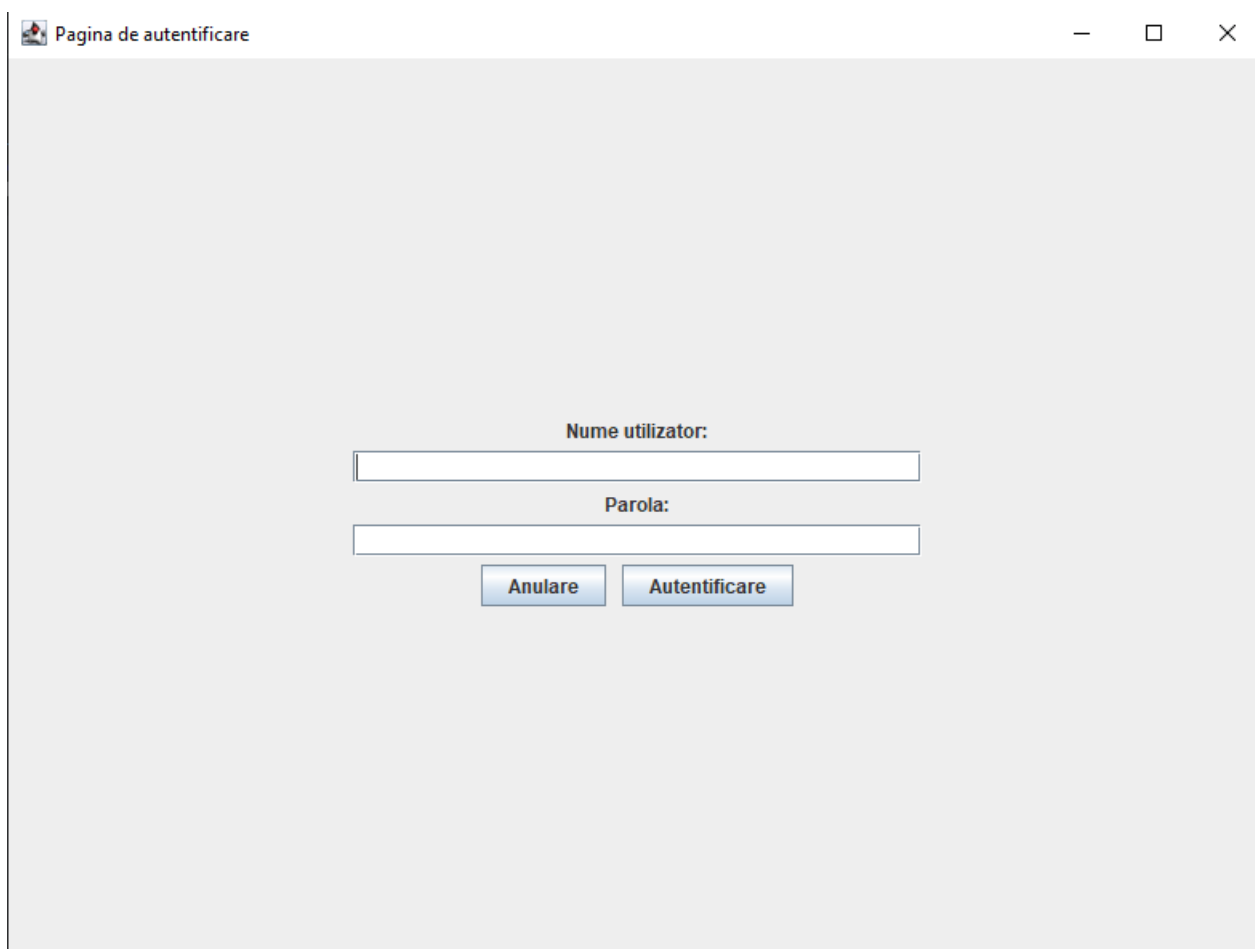
În pericol de dispariție:

Locație:

Anulare Salvare



De pe prima pagină, 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.



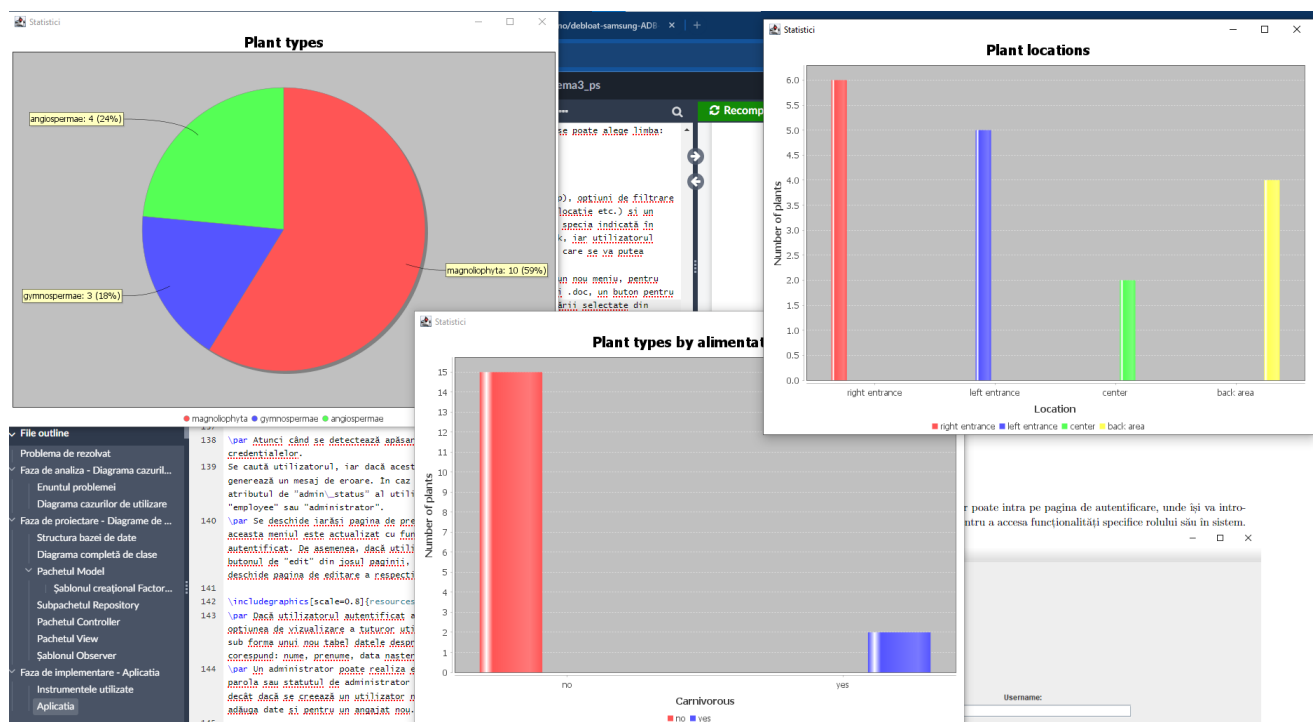
Atunci când se detectează apăsarea butonului "Log in", se începe verificarea credențialelor. Se caută utilizatorul în baza de date, iar dacă acesta nu există sau a indicat parola greșită interfața generează un mesaj de eroare. În caz că autentificarea a avut loc cu succes, se veri-

fică 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" și "statistics" din josul paginii.

Apăsarea butonului "statistics" va determina deschiderea a 3 noi ferestre cu grafice:

1. O statistică despre numărul plantelor de fiecare tip;
2. O statistică despre numărul plantelor după tipul de alimentație;
3. O statistică despre numărul plantelor după localizarea în grădina botanică;



Dacă utilizatorul autentificat are 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.

 Administrare utilizatori

Meniu
 Limba

☒ Doar administratori
 ☐ Doar angajați

userID	person	password	username	adminStatus
1	1	pass	dv_one	no
2	2	1234	flor	no
3	3	admin	admin	yes
4	4	SeCuRe	user_1k	no
5	1	pass	dv_one	no
6	2	1234	flor	no
7	3	admin	admin	yes
8	4	SeCuRe	user_1k	no

Editare

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.