

PROIECT

Termen predare: 25 mai 2021

Obiectiv:

Obiectivul acestui proiect este familiarizarea cu șablonul architectural *Client/Server* și cu șabloanele de proiectare.

Restricție: Pentru persistență se va utiliza fie o bază de date relațională (SQL Server, MySQL, Oracle, etc.), fie un fișier XML.

Cerințe:

Transformați aplicația implementată la **tema 3** într-o aplicație client-server astfel încât să utilizați un șablon de proiectare creațional, un șablon de proiectare comportamental și un șablon de proiectare structural. Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională (implicit limba română).

- ❖ În **faza de analiză** se vor realiza:
 - diagrama cazurilor de utilizare corespunzătoare aplicației *server*;
 - diagrama cazurilor de utilizare corespunzătoare aplicației *client*;
 - diagramele de activități corespunzătoare tuturor cazurilor de utilizare.
- ❖ În **faza de proiectare** se vor realiza:
 - diagrama de clase corespunzătoare aplicației *server*;
 - diagrama de clase corespunzătoare aplicației *client*;
 - diagrama de relaționare a entităților (doar dacă s-a utilizat o bază de date pentru persistență);
 - diagrame de secvență corespunzătoare tuturor cazurilor de utilizare.
- ❖ În **faza de implementare** se va scrie cod pentru îndeplinirea tuturor funcționalităților utilizând unul dintre următoarele limbaje de programare: C#, C++, Java, Python.
- ❖ Finalizarea temei va consta în predarea unui director ce va cuprinde:
 - Un fișier word care cuprinde:
 - numele studentului, grupa, enunțul problemei;
 - enumerarea instrumentelor utilizate în dezvoltarea aplicației;
 - justificarea limbajului de programare ales;
 - justificarea modului de realizare a persistenței;
 - justificarea alegerii șablonului de proiectare creațional utilizat;
 - justificarea alegerii șablonului de proiectare comportamental utilizat;
 - justificarea alegerii șablonului de proiectare structural utilizat;
 - prezentarea etapelor de dezvoltare a aplicației informatice:
 - etapa de analiză: descriere diagrame;
 - etapa de proiectare: descriere diagrame;
 - etapa de implementare: prezentarea interfețelor grafice și a funcționalităților implementate.
 - Un fișier cu diagramele UML realizate.
 - Directorul cu aplicația implementată.

Nr_crt	Lista studentilor	Număr problemă
1	Alexe Cristina - Gabriela	10
2	Alexuc Tudor	12
3	Alungei Paul - Catalin	18
4	Anechitei Matei	1
5	Bonta Sebastiano - Florin	35
6	Brinzan Elena - Denisa	3
7	Campean Vasile - Laurentiu	8
8	Ciure Raluca - Dana	33
9	Cosan Cosmin	16
10	Filip Vlad - Ilie	36
11	Gherman Miruna	19
12	Luncan Bogdan - Rares	28
13	Malita Teodora Loredana	
14	Marginean Ovidiu Adrian	34
15	Mesaros Viviana - Sara	6
16	Metes Miruna	7
17	Muresan Raul Cristian	
18	Muresan Denisa Bianca	17
19	Muresan Andreea - Ioana	5
20	Osan Tudor	
21	Pati Diana Georgeta	2
22	Petrascu Andrei	15
23	Popa Sergiu - Nicolae	13
24	Risa Alexandru - Iosif	31
25	Roncea Natalia	
26	Rus Mihai - Tudorel	14
27	Sas Bianca - Alexandra	4
28	Sasaran Andrei - Paul	11
29	Stoian Iulia - Tia	30
30	Vanyi Vivien	9

Problema 1

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o *aplicație client/server* care poate fi utilizată în instituții unde pot fi expuse opere de artă: galerii de artă sau muzee. Conceptul de **operă de artă plastică** este caracterizată de următoarele caracteristici: titlul, numele artistului și anul realizării. Din această clasă se vor deriva clasele corespunzătoare conceptelor **tablou** și **sculptură**. Sculptura va avea ca atribut tipul de sculptură (altorelief, basorelief, relief, statuie, statuie ecvestră, etc.), iar tabloul va avea ca atribute genul picturii (peisaj, portret, etc.) și tehnica utilizată (ulei, acuarelă, ceară, frescă, etc.).

Aplicația va avea 3 tipuri de utilizatori: vizitator al instituției (galerie/muzeu) de artă, angajat al lanțului de instituții de artă și administrator.

Utilizatorii de tip vizitator pot efectua următoarele operații fără autentificare:

- ❖ Vizualizarea tuturor operelor de artă expuse în aceste instituții de artă;
- ❖ Filtrarea listei operelor de artă plastică după următoarele criterii: instituția de artă, artist, tipul operei de artă;
- ❖ Căutarea unei opere de artă după titlu;
- ❖ Vizualizarea unor statistici legate de operele de artă din aceste instituții de artă: procente după instituția unde sunt expuse operele de artă, tipul de opere de artă expuse în aceeași instituție utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

Utilizatorii de tip angajat al lanțului de instituții de artă pot efectua următoarele operații după autentificare:

- ❖ Toate operațiile permise utilizatorilor de tip vizitator;
- ❖ Operații CRUD în ceea ce privește persistența operelor de artă expuse;
- ❖ Salvare rapoarte / liste cu operele de artă în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip angajat.

Problema 2

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată într-un lanț de farmacii / plafare. Aplicația va avea două tipuri de utilizatori: angajat al lanțului de farmacii / plafare și administrator.

Utilizatorii de tip angajat al lanțului de farmacii/plafare pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea tuturor produselor dintr-o farmacie selectată / dintr-un plafar selectat;
- ❖ Filtrarea produselor după următoarele criterii: disponibilitate, valabilitate, preț, producător;
- ❖ Căutarea unui produs după denumire;
- ❖ Vizualizarea unor statistici legate de produse: procente după disponibilitate, termen de valabilitate, categorii, pret utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).
- ❖ Operații CRUD în ceea ce privește persistența produselor;
- ❖ Salvare rapoarte / liste cu produse în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip angajat.

Problema 3

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată într-un lanț de magazine de parfumuri. Aplicația va avea două tipuri de utilizatori: angajat al lanțului de magazine de parfumuri și administrator.

Utilizatorii de tip angajat al lanțului de magazine de parfumuri pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea tuturor parfumurilor dintr-un magazin selectat;
- ❖ Filtrarea parfumurilor după următoarele criterii: producător, disponibilitate, preț;

- ❖ Căutarea unui parfum după denumire;
- ❖ Vizualizarea unor statistici legate de parfumuri: procente după disponibilitate, preț, producători utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).
- ❖ Operații CRUD în ceea ce privește persistența parfumurilor;
- ❖ Salvare rapoarte / liste cu parfumuri în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip angajat.

Problema 4

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată într-un lanț de cofetării. Aplicația va avea două tipuri de utilizatori: angajat al lanțului de cofetării și administrator.

Utilizatorii de tip angajat al lanțului de cofetării pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea tuturor prăjiturilor dintr-o cofetărie selectată;
- ❖ Filtrarea prăjiturilor după următoarele criterii: disponibilitate, preț, valabilitate;
- ❖ Căutarea unei prăjituri după denumire;
- ❖ Vizualizarea unor statistici legate de prăjituri: procente după disponibilitate, preț, valabilitate utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).
- ❖ Operații CRUD în ceea ce privește persistența prăjiturilor;
- ❖ Salvare rapoarte / liste cu prăjituri în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip angajat.

Problema 5

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată într-un lanț de florării. Aplicația va avea două tipuri de utilizatori: angajat al lanțului de florării și administrator.

Utilizatorii de tip angajat al lanțului de florării pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea tuturor florilor dintr-o florărie selectată;
- ❖ Filtrarea florilor după următoarele criterii: disponibilitate, preț, culoare, cantitate;
- ❖ Căutarea unei flori după denumire;
- ❖ Vizualizarea unor statistici legate de flori: procente după disponibilitate, preț, culoare, cantitate utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).
- ❖ Operații CRUD în ceea ce privește persistența florilor;
- ❖ Salvare rapoarte / liste cu flori în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip angajat.

Problema 6

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată în bibliotecile unei universități. Aplicația va avea 3 tipuri de utilizatori: abonat al bibliotecii, bibliotecar și administrator.

Utilizatorii de tip abonat pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea tuturor cărților dintr-o bibliotecă selectată;
- ❖ Filtrarea listei cărților după următoarele criterii: domeniu, disponibilitate, editura, autor;
- ❖ Căutarea unei cărți după titlu;
- ❖ Vizualizarea unor statistici legate de cărțile din aceste biblioteci: procente după domeniu, disponibilitate, editura, autor utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

Utilizatorii de tip bibliotecar pot efectua următoarele operații după autentificare:

- ❖ Toate operațiile permise utilizatorilor de tip abonat;
- ❖ Împrumutarea/returnarea unei cărți;
- ❖ Operații CRUD în ceea ce privește persistența cărților;
- ❖ Salvare rapoarte / liste cu cărțile în mai multe formate: csv, json, txt;
- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip abonat.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip bibliotecar.

Problema 7

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată într-un lanț de librării. Aplicația va avea 2 tipuri de utilizatori: angajat al lanțului de librării și administrator.

Utilizatorii de tip angajat pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea tuturor cărților dintr-o librărie selectată;
- ❖ Filtrarea listei cărților după următoarele criterii: domeniu, disponibilitate, editura, autor, preț;
- ❖ Căutarea unei cărți după titlu;
- ❖ Vizualizarea unor statistici legate de cărțile din acest lanț de librării: procente după domeniu, disponibilitate, editura, autor utilizând grafice (structură radială, structură inelară, de tip coloană, etc.);
- ❖ Vânzarea unei cărți (decrementarea cu 1 a numărului de exemplare disponibile din aceea carte);
- ❖ Operații CRUD în ceea ce privește persistența cărților;
- ❖ Salvare rapoarte / liste cu cărțile în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip angajat.

Problema 8

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** 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 tuturor plantelor din grădina botanică;
- ❖ Filtrarea listei plantelor după următoarele criterii: tip, specie, plante carnivore, zona grădină botanică;
- ❖ Căutarea unei plante după denumire;
- ❖ Vizualizarea unor statistici legate de plantele din grădina botanică: procente după specie, tip, zona grădina botanică utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

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 rapoarte / liste cu plantele în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip angajat.

Problema 9

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată într-un cabinet medical. Aplicația va avea 3 tipuri de utilizatori: medic, asistent și administrator.

Utilizatorii de tip medic pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea și actualizarea (simptome, diagnostic, tratament) fișelor medicale ale propriilor pacienți;
- ❖ Filtrarea propriilor pacienți după următoarele criterii: diagnostic, tratament;
- ❖ Căutarea unui pacient după nume;
- ❖ Vizualizarea unor statistici legate de pacienți: procente după diagnostic, simptome utilizând grafice (structură radială, structură inelară, de tip coloană, etc.);
- ❖ Specificarea propriului program de lucru și vizualizarea propriului program de consultații.

Utilizatorii de tip asistent pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD în ceea ce privește persistența pacienților (mai puțin actualizări legate de simptome, diagnostic, tratament);
- ❖ Filtrarea pacienților după următoarele criterii: medic, diagnostic, varstă;
- ❖ Căutarea unui pacient după nume;
- ❖ Planificarea pacienților pentru consultații;
- ❖ Salvare rapoarte / liste cu pacienți sau medici în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip medic și asistent.

Problema 10

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată într-un cabinet veterinar. Aplicația va avea 3 tipuri de utilizatori: medic, asistent și administrator.

Utilizatorii de tip medic pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea și actualizarea (simptome, diagnostic, tratament) fișelor medicale ale animalelor;
- ❖ Filtrarea animalelor după următoarele criterii: diagnostic, tratament;

- ❖ Vizualizarea unor statistici legate de animale: procente după diagnostic, simptome utilizând grafice (structură radială, structură inelară, de tip coloană, etc.);

- ❖ Specificarea propriului program de lucru și vizualizarea propriului program de consultații.

Utilizatorii de tip asistent pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD în ceea ce privește persistența animalelor (mai puțin actualizari legate de simptome, diagnostic, tratament);

- ❖ Filtrarea animalelor după următoarele criterii: medic, diagnostic, specie;

- ❖ Planificarea animalelor pentru consultații;

- ❖ Salvare rapoarte / liste cu animale sau medici în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip medic și asistent.

Problema 11

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată într-un oraș pentru determinarea traseului optim utilizând transportul în comun. Aplicația va avea 3 tipuri de utilizatori: călător, angajat al firmei de transport în comun și administrator.

Utilizatorii de tip călător pot efectua următoarele operații fără autentificare:

- ❖ Vizualizarea tuturor liniilor de transport în comun;

- ❖ Vizualizarea traseului optim după specificare stației de plecare și a stației de sosire (se va modela prin utilizarea grafurilor);

- ❖ Căutarea unei linii de transport în comun după număr.

Utilizatorii de tip angajat pot efectua următoarele operații după autentificare:

- ❖ Toate operațiile permise utilizatorilor de tip călător;

- ❖ Operații CRUD în ceea ce privește persistența liniilor de transport în comun;

- ❖ Salvare rapoarte / liste cu trasee optime în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip angajat.

Problema 12

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată de către o firmă de curierat rapid. Aplicația va avea 3 tipuri de utilizatori: poștaş, coordonator activitate și administrator.

Utilizatorii de tip poștaş pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea listei coletelor de distribuit, inclusiv locațiile unde trebuie distribuite;

- ❖ Vizualizarea traseului optim în funcție de lista coletelor (se va modela prin utilizarea grafurilor);

- ❖ Căutarea unui colet după număr.

Utilizatorii de tip coordonator activitate pot efectua următoarele operații după autentificare:

- ❖ Toate operațiile permise utilizatorilor de tip poștaş;

- ❖ Operații CRUD în ceea ce privește persistența coletelor;

- ❖ Alocarea unui colet către un poștaş;
- ❖ Salvare rapoarte / liste cu informații despre colete în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip poștaş și coordonator activitate.

Problema 13

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată de către o agenție turistică. Aplicația va avea 3 tipuri de utilizatori: client, angajat și administrator.

Utilizatorii de tip client pot efectua următoarele operații fără autentificare:

- ❖ Vizualizarea listei pachetelor oferite de agenție;
- ❖ Filtrarea listei pachetelor după destinație, perioadă, preț;
- ❖ Vizualizarea unor statistici despre cele mai vizitate stațiuni utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

Utilizatorii de tip angajat pot efectua următoarele operații după autentificare:

- ❖ Toate operațiile permise utilizatorilor de tip client;
- ❖ Operații CRUD în ceea ce privește persistența pachetelor oferite de agenție;
- ❖ Rezervarea unui pachet de către un client;
- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip client;
- ❖ Salvare rapoarte cu informații despre pachetele oferite de agenție în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip angajat.

Problema 14

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată într-un cabinet de nutriție. Aplicația va avea 3 tipuri de utilizatori: dietetician, secretar și administrator.

Utilizatorii de tip dietetician pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea propriilor clienți;
- ❖ Stabilirea dietei unui client;
- ❖ Filtrarea propriilor clienți după următoarele criterii: vârstă, greutate, dietă;
- ❖ Căutarea unui client după nume;
- ❖ Vizualizarea unor statistici legate de clienți: procente după vârstă, greutate utilizând grafice (structură radială, structură inelară, de tip coloană, etc.);
- ❖ Specificarea propriului program de lucru și vizualizarea propriului program de consultații.

Utilizatorii de tip secretar pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD în ceea ce privește persistența clienților (mai puțin actualizări legate de dietă);
- ❖ Filtrarea clienților după următoarele criterii: dietetician, dietă, vârstă, greutate;
- ❖ Căutarea unui client după nume;
- ❖ Planificarea clienților pentru consultații;

❖ Salvare rapoarte / liste cu clienți în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

❖ Operații CRUD pentru informațiile legate de utilizatorii de tip dietetician și secretar.

Problema 15

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată de către o agenție de rezervare bilete de tren. Aplicația va avea 3 tipuri de utilizatori: călător, angajat și administrator.

Utilizatorii de tip călător pot efectua următoarele operații fără autentificare:

- ❖ Vizualizarea listei trenurilor după stație de plecare, destinație, durată;
- ❖ Vizualizarea listei trenurilor dintre 2 locații, inclusiv preț și disponibilitate locuri libere;
- ❖ Căutarea unui tren după număr.

Utilizatorii de tip angajat pot efectua următoarele operații după autentificare:

- ❖ Toate operațiile permise utilizatorilor de tip călător;
- ❖ Operații CRUD în ceea ce privește persistența trenurilor și biletelor vândute;
- ❖ Vânzarea unui bilet către un călător;
- ❖ Vizualizarea unor statistici legate de biletele vândute: procente după stație de plecare, destinație, preț utilizând grafice (structură radială, structură inelară, de tip coloană, etc.);
- ❖ Salvare rapoarte / liste cu informații despre trenuri în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip angajat.

Problema 16

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată de către o agenție imobiliară pentru gestionarea închirierilor de locuințe. Aplicația va avea 3 tipuri de utilizatori: client, angajat și administrator. Utilizatorii de tip client pot efectua următoarele operații fără autentificare:

- ❖ Vizualizarea listei proprietăților disponibile pentru închiriat;
- ❖ Filtrarea listei proprietăților disponibile pentru închiriat după locație, preț, tip locuință, număr camere;
- ❖ Vizualizarea unor statistici legate de locuințele de închiriat disponibile: procente după locație, preț, tip locuință, număr camere utilizând grafice (structură radială, structură inelară, de tip coloană, etc.);

Utilizatorii de tip angajat pot efectua următoarele operații după autentificare:

- ❖ Toate operațiile permise utilizatorilor de tip client;
- ❖ Operații CRUD în ceea ce privește persistența locuințelor de închiriat și a clienților;
- ❖ Închirierea unei locuințe de către un client;
- ❖ Salvare rapoarte / liste cu informații despre locuințele de închiriat în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip angajat.

Problema 17

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată de către un lanț hotelier. Aplicația va avea 3 tipuri de utilizatori: client, angajat și administrator.

Utilizatorii de tip client pot efectua următoarele operații fără autentificare:

- ❖ Vizualizarea listei camerelor dintr-un hotel selectat;
- ❖ Filtrarea listei camerelor după locație, disponibilitate, preț, poziție, facilități;
- ❖ Vizualizarea unor statistici despre cele mai rezervate camere din lanțul hotelier utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

Utilizatorii de tip angajat pot efectua următoarele operații după autentificare:

- ❖ Toate operațiile permise utilizatorilor de tip client;
- ❖ Operații CRUD în ceea ce privește persistența camerelor din lanțul hotelier;
- ❖ Rezervarea unei camere de către un client;
- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip client;
- ❖ Salvare rapoarte / liste cu informații despre camerele rezervate în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Toate operațiile permise utilizatorilor de tip client;
- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip angajat.

Problema 18

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată de către o firmă organizatoare de evenimente. Aplicația va avea 3 tipuri de utilizatori: client, coordonator eveniment și administrator.

Utilizatorii de tip client pot efectua următoarele operații fără autentificare:

- ❖ Vizualizarea listei tipurilor de evenimente coordonate;
- ❖ Vizualizarea unor statistici legate de evenimentele deja organizate: procente după scopul evenimentului, locație, număr persoane utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

Utilizatorii de tip coordonator eveniment pot efectua următoarele operații după autentificare:

- ❖ Toate operațiile permise utilizatorilor de tip client;
- ❖ Operații CRUD în ceea ce privește persistența evenimentelor și a clienților;
- ❖ Filtrarea listei de evenimente după locație, număr persoane, scop;
- ❖ Salvare rapoarte / liste cu informații despre evenimentele ce urmează a fi organizate în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Toate operațiile permise utilizatorilor de tip client;
- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip coordonator eveniment.

Problema 19

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată într-un lanț de magazine cu produse cosmetice. Aplicația va avea două tipuri de utilizatori: angajat al lanțului de magazine cu produse cosmetice și administrator.

Utilizatorii de tip angajat al lanțului de magazine cu produse cosmetice pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea tuturor produselor cosmetice disponibile într-un magazin selectat;
- ❖ Filtrarea produselor cosmetice după următoarele criterii: producător, disponibilitate, preț;
- ❖ Căutarea unui produs cosmetic după denumire;
- ❖ Vizualizarea unor statistici legate de produsele cosmetice: procente după disponibilitate, preț, producători utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).
- ❖ Operații CRUD în ceea ce privește persistența produselor cosmetice;
- ❖ Salvare rapoarte / liste cu produse cosmetice în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip angajat.

Problema 20

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată ca soft educațional pentru studiul geometriei patrulaterului. Aplicația va avea un singur tip de utilizator care va putea efectua următoarele operații fără autentificare:

- ❖ desenarea interactivă a patrulaterelor prin înlocuirea creionului și a riglei cu *mouse*-ul și alegerea stilului de desenare (inclusiv culoarea);
- ❖ verificarea și afișarea unor caracteristici: patrulater convex/concav, patrulater înscritibil, patrulater circumscriptibil;
- ❖ calcularea și afișarea unor proprietăți: lungimile laturilor, măsurile unghiurilor, perimetrul, aria, raza cercului înscris (dacă patrulaterul este circumscriptibil), raza cercului circumscris (dacă patrulaterul este înscritibil);
- ❖ vizualizarea unor elemente specifice unui patrulater:
 - puncte importante într-un patrulater: punctul lui Newton al unui patrulater circumscriptibil, punctul lui Miquel al unui patrulater convex, punctul lui Mathot al unui patrulater înscritibil;
 - linii importante într-un patrulater convex: diagonale, bimediane, bisectoare, dreapta lui Newton corespunzătoare unui patrulater circumscriptibil, dreapta lui Gauss corespunzătoare unui patrulater complet, dreapta lui Aubert corespunzătoare unui patrulater complet;
 - ceruri speciale: cercul circumscris (dacă patrulaterul este înscritibil), cercul înscris (dacă patrulaterul este circumscriptibil);
- ❖ salvarea / încărcarea unui patrulater într-un/dintr-un fișier xml.

Problema 21

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată ca soft educațional pentru studiul conicelor. Aplicația va avea un singur tip de utilizator care va putea efectua următoarele operații fără autentificare:

- ❖ desenarea interactivă a conicelor (elipsă, hiperbolă, parabolă, cerc) prin înlocuirea creionului și a riglei cu *mouse*-ul și alegerea stilului de desenare (inclusiv culoarea);
- ❖ determinarea și desenarea tangentei și normalei într-un punct la o conică;
- ❖ determinarea și desenarea conicei obținute prin simetrie față de o dreaptă;
- ❖ salvarea / încărcare unei conice într-un/dintr-un fișier xml.

Problema 22

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată ca soft educațional pentru studiul poliedrelor. Aplicația va avea un singur tip de utilizator care va putea efectua următoarele operații fără autentificare:

- ❖ desenarea interactivă a poliedrelor (prismă, paralelipiped dreptunghic, cub, piramidă, tetraedru, trunchi de piramidă) prin înlocuirea creionului și a riglei cu *mouse*-ul și alegerea stilului de desenare (inclusiv culoarea);
- ❖ calcularea și afișarea unor proprietăți: aria laterală, aria bazei, aria totală, volum;
- ❖ determinarea și desenarea poliedrului obținut prin simetrie față de un plan;
- ❖ salvarea / încărcare unui poliedru într-un/dintr-un fișier xml.

Problema 23

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată ca soft educațional pentru studiul corpurilor rotunde. Aplicația va avea un singur tip de utilizator care va putea efectua următoarele operații fără autentificare:

- ❖ desenarea interactivă a poliedrelor (cilindru, con, trunchi de con, sferă) prin înlocuirea creionului și a riglei cu *mouse*-ul și alegerea stilului de desenare (inclusiv culoarea);
- ❖ calcularea și afișarea unor proprietăți: aria laterală, aria bazei, aria totală, volum;
- ❖ determinarea și desenarea corpului rotund obținut prin simetrie față de un plan;
- ❖ salvarea / încărcare unui corp rotund într-un/dintr-un fișier xml.

Problema 24

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** care poate fi utilizată de către o întreprindere horticolă. Aplicația va avea 2 tipuri de utilizatori: angajat și administrator.

Utilizatorii de tip angajat pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea listei tuturor serelor existente în întreprinderea horticolă;

- ❖ Filtrarea listei serelor după dimensiune, temperatura interioară;
 - ❖ Vizualizarea rețelei de termoficare optimă după poziție (se va modela prin utilizarea grafurilor);
 - ❖ Vizualizarea unor statistici legate de sere: procente după dimensiune, temperatură interioară utilizând grafice (structură radială, structură inelară, de tip coloană, etc.);
 - ❖ Operații CRUD în ceea ce privește persistența serelor;
 - ❖ Salvare rapoarte / liste cu informații despre sere în mai multe formate: csv, json, txt.
- Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:
- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip angajat.

Problema 28

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o *aplicație client/server* care poate fi utilizată de către o întreprindere pentru optimizarea procesului de utilizare a liniilor automate de fabricație. Aplicația va avea 2 tipuri de utilizatori: angajat și administrator. Utilizatorii de tip angajat pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea listei tuturor probuselor fabricate în întreprindere;
- ❖ Vizualizare listei cu etapele de trebuie parcurse pentru fabricarea lui;
- ❖ Vizualizarea listei liniilor de fabricație din întreprindere (inclusiv specificul acestora);
- ❖ Filtrarea listei liniilor de fabricație din întreprindere după scop, disponibilitate, poziție;
- ❖ Vizualizarea listei optime de utilizare a liniilor de fabricație pentru fabricarea unui produs selectat (se va modela prin utilizarea grafurilor, nodurile în graf fiind date de liniile de fabricație);
- ❖ Operații CRUD în ceea ce privește persistența produselor și a liniilor de fabricație;
- ❖ Salvare rapoarte / liste cu informații despre liniile de fabricație în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip angajat.

Problema 29

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o *aplicație client/server* care poate fi utilizată de către o firmă de colectare deșeurilor. Aplicația va avea 3 tipuri de utilizatori: angajat, coordonator activitate și administrator.

Utilizatorii de tip angajat pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea listei cu pozițiile deșeurilor ce urmează a fi colectate;
- ❖ Vizualizarea traseului optim în funcție de poziția deșeurilor (se va modela prin utilizarea grafurilor).

Utilizatorii de tip coordonator activitate pot efectua următoarele operații după autentificare:

- ❖ Toate operațiile permise utilizatorilor de tip angajat;
- ❖ Operații CRUD în ceea ce privește persistența locațiilor cu deșeurilor;
- ❖ Alocarea unei locații cu deșeurilor către un angajat în vederea colectării;
- ❖ Salvare rapoarte / liste cu informații despre listele cu pozițiile deșeurilor alocate angajaților în mai multe formate: csv, json, txt.

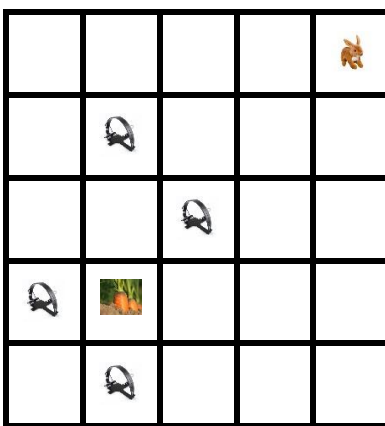
Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip angajat și coordonator activitate.

Problema 30

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** pentru următorul joc: se dă un labirint de dimensiune $n \times n$ căsuțe în care pot să existe capcane pentru animale sălbatice și o “comoară” de morcovi. Într-un colț al labirintului se află un iepuraș flămând. Ajutați-l pe iepuraș să ajungă la “comoara” de morcovi astfel încât să nu fie prins într-una din capcane, știind că el se poate deplasa prin sărituri pe diagonală în căsuțele care au un vârf comun cu căsuța curentă.

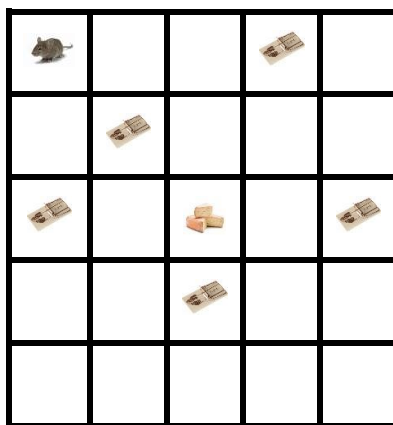


Jocul va fi dezvoltat pe mai multe niveluri ($n=4,5,6,7,\dots$). După ce un jucător a obținut soluția, se va afișa numărul de deplasări utilizat de către acesta în soluția furnizată și i se va comunica dacă a obținut soluția optimă. Dacă nu a obținut soluția optimă, aceasta va fi afișată pas cu pas. Soluția optimă se va determina utilizând un algoritm euristic (de exemplu A*).

Problema 31

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o **aplicație client/server** pentru următorul joc: se consideră un labirint de dimensiune $n \times n$ căsuțe în care pot să existe curse pentru șoricea și o “comoară” din brânză. Într-un colț al labirintului se află un șoricel flămând care simte mirosul brânzei, dar nu știe unde este poziționată aceasta. Ajutați-l pe șoricel să ajungă la “comoara” de brânză astfel încât să nu fie prins într-una din curse, știind că el se poate deplasa în căsuțele alăturate fie pe orizontală, fie pe verticală. Jocul va fi dezvoltat pe mai multe niveluri ($n=4,5,6,7,\dots$). După ce un jucător a obținut soluția, se va afișa numărul de deplasări utilizat de către acesta în soluția furnizată și i se va comunica dacă a obținut soluția optimă. Dacă nu a obținut soluția optimă, aceasta va fi afișată pas cu pas. Soluția optimă se va determina utilizând un algoritm euristic (de exemplu A*).



Problema 32

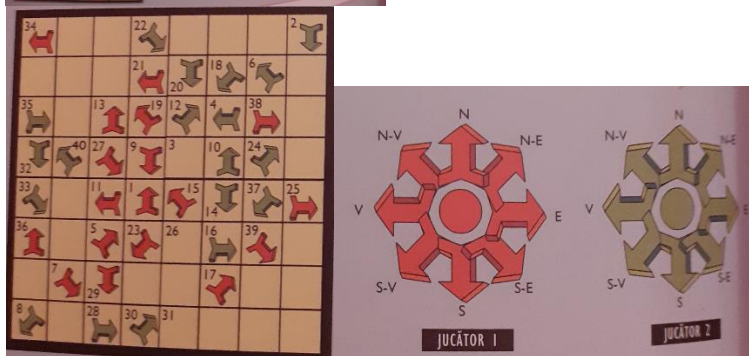
Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o *aplicație client/server* pentru următorul joc cu 2 jucători: Se consideră săgeți de 2 culori, iar scopul jocului este de a completa zonele din pătrat astfel încât pe aceeași linie, pe aceeași coloană și pe aceeași diagonală să nu se găsească 2 săgeți orientate în aceeași direcție, indiferent de culoarea acestora. Pierde jucătorul care nu mai are nicio posibilitate de a așeza săgeți în zonele libere ale pătratului. Vor exista 2 niveluri ale jocului conform imaginilor de mai jos. Implementarea se va realiza astfel încât un utilizator al aplicației (jocului) să joace cu calculatorul. Se va utiliza o variantă a algoritmului MINIMAX.

Nivel 1



Nivel 2



Problema 33

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o *aplicație client/server* care poate fi utilizată într-un lanț de magazine de îmbrăcăminte. Aplicația va avea două tipuri de utilizatori: angajat al lanțului de magazine de îmbrăcăminte și administrator. Utilizatorii de tip angajat al lanțului de magazine de îmbrăcăminte pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea tuturor produselor de îmbrăcăminte disponibile într-un magazin selectat;
- ❖ Filtrarea produselor de îmbrăcăminte după următoarele criterii: producător, disponibilitate, preț;
- ❖ Căutarea unui produs de îmbrăcăminte după denumire și/sau producător;
- ❖ Vizualizarea unor statistici legate de produsele de îmbrăcăminte: procente după disponibilitate, preț, producători utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).
- ❖ Operații CRUD în ceea ce privește persistența produselor de îmbrăcăminte;
- ❖ Salvare rapoarte / liste cu produse de îmbrăcăminte în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip angajat.

Problema 34

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o *aplicație client/server* care poate fi utilizată pentru gestionarea de anunțuri. Aplicația va avea 3 tipuri de utilizatori: client, editor și administrator.

Utilizatorii de tip client pot efectua următoarele operații fără autentificare:

- ❖ Vizualizarea listei anunțurilor grupate pe categorii;
- ❖ Vizualizarea unor statistici legate de anunțuri: procente după categorie, preț utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

Utilizatorii de tip editor pot efectua următoarele operații după autentificare:

- ❖ Toate operațiile permise utilizatorilor de tip client;
- ❖ Operații CRUD în ceea ce privește persistența anunțurilor;
- ❖ Filtrarea listei de anunțuri după categorie;
- ❖ Salvare rapoarte / liste cu informații despre anunțuri în mai multe formate: csv, json, txt.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Toate operațiile permise utilizatorilor de tip client;
- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip editor.

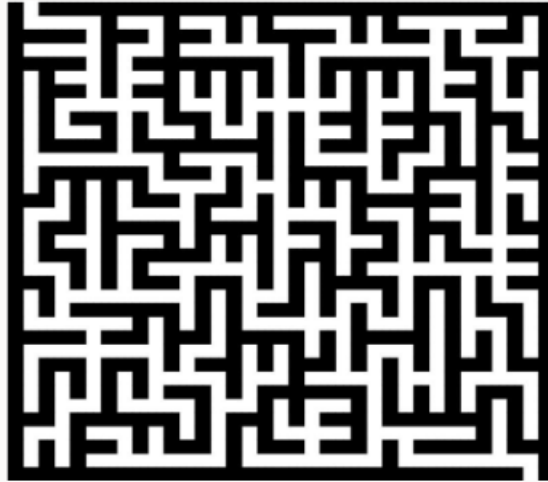
Problema 35

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o *aplicație client/server* pentru un joc de șah. Implementarea se va realiza astfel încât un utilizator al aplicației (jocului) să joace cu calculatorul. Se va utiliza o variantă a algoritmului MINIMAX.

Problema 36**Descrierea aplicației:**

Dezvoltați (analiză, proiectare, implementare) o *aplicație client/server* pentru următorul joc de tipul labirintului prezentat în continuare.



Jocul va fi dezvoltat pe mai multe niveluri. După ce un jucător a obținut soluția, se va afișa numărul de deplasări utilizat de către acesta în soluția furnizată și i se va comunica dacă a obținut soluția optimă. Dacă nu a obținut soluția optimă, aceasta va fi afișată pas cu pas. Soluția optimă se va determina utilizând un algoritm euristic (de exemplu A*).